



OpenShift Container Platform 4.20

Workloads APIs

Reference guide for workloads APIs

OpenShift Container Platform 4.20 Workloads APIs

Reference guide for workloads APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes the OpenShift Container Platform workloads API objects and their detailed specifications.

Table of Contents

CHAPTER 1. WORKLOADS APIS	24
1.1. BUILDCONFIG [BUILD.OPENSIFT.IO/V1]	24
1.2. BUILD [BUILD.OPENSIFT.IO/V1]	24
1.3. BUILDLOG [BUILD.OPENSIFT.IO/V1]	24
1.4. BUILDREQUEST [BUILD.OPENSIFT.IO/V1]	24
1.5. CRONJOB [BATCH/V1]	25
1.6. DAEMONSET [APPS/V1]	25
1.7. DEPLOYMENT [APPS/V1]	25
1.8. DEPLOYMENTCONFIG [APPS.OPENSIFT.IO/V1]	25
1.9. DEPLOYMENTCONFIGROLLBACK [APPS.OPENSIFT.IO/V1]	25
1.10. DEPLOYMENTLOG [APPS.OPENSIFT.IO/V1]	26
1.11. DEPLOYMENTREQUEST [APPS.OPENSIFT.IO/V1]	26
1.12. JOB [BATCH/V1]	26
1.13. POD [V1]	26
1.14. REPLICATIONCONTROLLER [V1]	26
1.15. REPLICASET [APPS/V1]	27
1.16. STATEFULSET [APPS/V1]	27
 CHAPTER 2. BUILDCONFIG [BUILD.OPENSIFT.IO/V1]	 28
2.1. SPECIFICATION	28
2.1.1. .spec	29
2.1.2. .spec.output	33
2.1.3. .spec.output.imageLabels	34
2.1.4. .spec.output.imageLabels[]	34
2.1.5. .spec.postCommit	34
2.1.6. .spec.revision	37
2.1.7. .spec.revision.git	37
2.1.8. .spec.revision.git.author	37
2.1.9. .spec.revision.git.committer	38
2.1.10. .spec.source	38
2.1.11. .spec.source.binary	40
2.1.12. .spec.source.configMaps	41
2.1.13. .spec.source.configMaps[]	41
2.1.14. .spec.source.git	42
2.1.15. .spec.source.images	42
2.1.16. .spec.source.images[]	43
2.1.17. .spec.source.images[].paths	44
2.1.18. .spec.source.images[].paths[]	44
2.1.19. .spec.source.secrets	45
2.1.20. .spec.source.secrets[]	45
2.1.21. .spec.strategy	46
2.1.22. .spec.strategy.customStrategy	47
2.1.23. .spec.strategy.customStrategy.secrets	48
2.1.24. .spec.strategy.customStrategy.secrets[]	48
2.1.25. .spec.strategy.dockerStrategy	48
2.1.26. .spec.strategy.dockerStrategy.volumes	51
2.1.27. .spec.strategy.dockerStrategy.volumes[]	51
2.1.28. .spec.strategy.dockerStrategy.volumes[].mounts	52
2.1.29. .spec.strategy.dockerStrategy.volumes[].mounts[]	52
2.1.30. .spec.strategy.dockerStrategy.volumes[].source	52
2.1.31. .spec.strategy.jenkinsPipelineStrategy	53

2.1.32. .spec.strategy.sourceStrategy	54
2.1.33. .spec.strategy.sourceStrategy.volumes	55
2.1.34. .spec.strategy.sourceStrategy.volumes[]	55
2.1.35. .spec.strategy.sourceStrategy.volumes[].mounts	56
2.1.36. .spec.strategy.sourceStrategy.volumes[].mounts[]	56
2.1.37. .spec.strategy.sourceStrategy.volumes[].source	57
2.1.38. .spec.triggers	58
2.1.39. .spec.triggers[]	58
2.1.40. .spec.triggers[].bitbucket	59
2.1.41. .spec.triggers[].bitbucket.secretReference	60
2.1.42. .spec.triggers[].generic	60
2.1.43. .spec.triggers[].generic.secretReference	61
2.1.44. .spec.triggers[].github	61
2.1.45. .spec.triggers[].github.secretReference	62
2.1.46. .spec.triggers[].gitlab	62
2.1.47. .spec.triggers[].gitlab.secretReference	63
2.1.48. .spec.triggers[].imageChange	63
2.1.49. .status	64
2.1.50. .status.imageChangeTriggers	65
2.1.51. .status.imageChangeTriggers[]	65
2.1.52. .status.imageChangeTriggers[].from	66
2.2. API ENDPOINTS	66
2.2.1. /apis/build.openshift.io/v1/buildconfigs	67
2.2.2. /apis/build.openshift.io/v1/watch/buildconfigs	67
2.2.3. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs	67
2.2.4. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs	69
2.2.5. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}	70
2.2.6. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs/{name}	73
CHAPTER 3. BUILD [BUILD.OPENSIFT.IO/V1]	74
3.1. SPECIFICATION	74
3.1.1. .spec	75
3.1.2. .spec.output	78
3.1.3. .spec.output.imageLabels	79
3.1.4. .spec.output.imageLabels[]	79
3.1.5. .spec.postCommit	80
3.1.6. .spec.revision	82
3.1.7. .spec.revision.git	82
3.1.8. .spec.revision.git.author	83
3.1.9. .spec.revision.git.committer	83
3.1.10. .spec.source	83
3.1.11. .spec.source.binary	86
3.1.12. .spec.source.configMaps	86
3.1.13. .spec.source.configMaps[]	86
3.1.14. .spec.source.git	87
3.1.15. .spec.source.images	88
3.1.16. .spec.source.images[]	88
3.1.17. .spec.source.images[].paths	89
3.1.18. .spec.source.images[].paths[]	89
3.1.19. .spec.source.secrets	90
3.1.20. .spec.source.secrets[]	90
3.1.21. .spec.strategy	91
3.1.22. .spec.strategy.customStrategy	92

3.1.23. .spec.strategy.customStrategy.secrets	93
3.1.24. .spec.strategy.customStrategy.secrets[]	93
3.1.25. .spec.strategy.dockerStrategy	93
3.1.26. .spec.strategy.dockerStrategy.volumes	96
3.1.27. .spec.strategy.dockerStrategy.volumes[]	96
3.1.28. .spec.strategy.dockerStrategy.volumes[].mounts	97
3.1.29. .spec.strategy.dockerStrategy.volumes[].mounts[]	97
3.1.30. .spec.strategy.dockerStrategy.volumes[].source	97
3.1.31. .spec.strategy.jenkinsPipelineStrategy	98
3.1.32. .spec.strategy.sourceStrategy	99
3.1.33. .spec.strategy.sourceStrategy.volumes	100
3.1.34. .spec.strategy.sourceStrategy.volumes[]	100
3.1.35. .spec.strategy.sourceStrategy.volumes[].mounts	101
3.1.36. .spec.strategy.sourceStrategy.volumes[].mounts[]	101
3.1.37. .spec.strategy.sourceStrategy.volumes[].source	102
3.1.38. .spec.triggeredBy	103
3.1.39. .spec.triggeredBy[]	103
3.1.40. .spec.triggeredBy[].bitbucketWebHook	104
3.1.41. .spec.triggeredBy[].bitbucketWebHook.revision	104
3.1.42. .spec.triggeredBy[].bitbucketWebHook.revision.git	104
3.1.43. .spec.triggeredBy[].bitbucketWebHook.revision.git.author	105
3.1.44. .spec.triggeredBy[].bitbucketWebHook.revision.git.committer	105
3.1.45. .spec.triggeredBy[].genericWebHook	106
3.1.46. .spec.triggeredBy[].genericWebHook.revision	106
3.1.47. .spec.triggeredBy[].genericWebHook.revision.git	106
3.1.48. .spec.triggeredBy[].genericWebHook.revision.git.author	107
3.1.49. .spec.triggeredBy[].genericWebHook.revision.git.committer	107
3.1.50. .spec.triggeredBy[].githubWebHook	107
3.1.51. .spec.triggeredBy[].githubWebHook.revision	108
3.1.52. .spec.triggeredBy[].githubWebHook.revision.git	108
3.1.53. .spec.triggeredBy[].githubWebHook.revision.git.author	109
3.1.54. .spec.triggeredBy[].githubWebHook.revision.git.committer	109
3.1.55. .spec.triggeredBy[].gitlabWebHook	109
3.1.56. .spec.triggeredBy[].gitlabWebHook.revision	110
3.1.57. .spec.triggeredBy[].gitlabWebHook.revision.git	110
3.1.58. .spec.triggeredBy[].gitlabWebHook.revision.git.author	111
3.1.59. .spec.triggeredBy[].gitlabWebHook.revision.git.committer	111
3.1.60. .spec.triggeredBy[].imageChangeBuild	111
3.1.61. .status	112
3.1.62. .status.conditions	113
3.1.63. .status.conditions[]	114
3.1.64. .status.output	114
3.1.65. .status.output.to	115
3.1.66. .status.stages	115
3.1.67. .status.stages[]	115
3.1.68. .status.stages[].steps	116
3.1.69. .status.stages[].steps[]	116
3.2. API ENDPOINTS	117
3.2.1. /apis/build.openshift.io/v1/builds	118
3.2.2. /apis/build.openshift.io/v1/watch/builds	118
3.2.3. /apis/build.openshift.io/v1/namespaces/{namespace}/builds	118
3.2.4. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds	120
3.2.5. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}	121

3.2.6. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds/{name}	124
3.2.7. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/details	124
3.2.8. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/webhooks	126
3.2.9. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/webhooks/{path}	126
CHAPTER 4. BUILDLOG [BUILD.OPENSIFT.IO/V1]	127
4.1. SPECIFICATION	127
4.2. API ENDPOINTS	127
4.2.1. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/log	127
CHAPTER 5. BUILDREQUEST [BUILD.OPENSIFT.IO/V1]	129
5.1. SPECIFICATION	129
5.1.1. .binary	131
5.1.2. .dockerStrategyOptions	131
5.1.3. .revision	132
5.1.4. .revision.git	132
5.1.5. .revision.git.author	133
5.1.6. .revision.git.committer	133
5.1.7. .sourceStrategyOptions	133
5.1.8. .triggeredBy	134
5.1.9. .triggeredBy[]	134
5.1.10. .triggeredBy[].bitbucketWebHook	134
5.1.11. .triggeredBy[].bitbucketWebHook.revision	135
5.1.12. .triggeredBy[].bitbucketWebHook.revision.git	135
5.1.13. .triggeredBy[].bitbucketWebHook.revision.git.author	136
5.1.14. .triggeredBy[].bitbucketWebHook.revision.git.committer	136
5.1.15. .triggeredBy[].genericWebHook	136
5.1.16. .triggeredBy[].genericWebHook.revision	137
5.1.17. .triggeredBy[].genericWebHook.revision.git	137
5.1.18. .triggeredBy[].genericWebHook.revision.git.author	138
5.1.19. .triggeredBy[].genericWebHook.revision.git.committer	138
5.1.20. .triggeredBy[].githubWebHook	138
5.1.21. .triggeredBy[].githubWebHook.revision	139
5.1.22. .triggeredBy[].githubWebHook.revision.git	139
5.1.23. .triggeredBy[].githubWebHook.revision.git.author	140
5.1.24. .triggeredBy[].githubWebHook.revision.git.committer	140
5.1.25. .triggeredBy[].gitlabWebHook	140
5.1.26. .triggeredBy[].gitlabWebHook.revision	141
5.1.27. .triggeredBy[].gitlabWebHook.revision.git	141
5.1.28. .triggeredBy[].gitlabWebHook.revision.git.author	142
5.1.29. .triggeredBy[].gitlabWebHook.revision.git.committer	142
5.1.30. .triggeredBy[].imageChangeBuild	142
5.2. API ENDPOINTS	143
5.2.1. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/clone	143
5.2.2. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/instantiate	144
CHAPTER 6. CRONJOB [BATCH/V1]	147
6.1. SPECIFICATION	147
6.1.1. .spec	147
6.1.2. .spec.jobTemplate	149
6.1.3. .spec.jobTemplate.spec	150
6.1.4. .spec.jobTemplate.spec.podFailurePolicy	156
6.1.5. .spec.jobTemplate.spec.podFailurePolicy.rules	157
6.1.6. .spec.jobTemplate.spec.podFailurePolicy.rules[]	157

6.1.7. .spec.jobTemplate.spec.podFailurePolicy.rules[].onExitCodes	159
6.1.8. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions	160
6.1.9. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions[]	161
6.1.10. .spec.jobTemplate.spec.successPolicy	161
6.1.11. .spec.jobTemplate.spec.successPolicy.rules	162
6.1.12. .spec.jobTemplate.spec.successPolicy.rules[]	162
6.1.13. .status	163
6.2. API ENDPOINTS	164
6.2.1. /apis/batch/v1/cronjobs	165
6.2.2. /apis/batch/v1/watch/cronjobs	165
6.2.3. /apis/batch/v1/namespaces/{namespace}/cronjobs	165
6.2.4. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs	167
6.2.5. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}	168
6.2.6. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs/{name}	171
6.2.7. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status	171
CHAPTER 7. DAEMONSET [APPS/V1]	175
7.1. SPECIFICATION	175
7.1.1. .spec	175
7.1.2. .spec.updateStrategy	177
7.1.3. .spec.updateStrategy.rollingUpdate	177
7.1.4. .status	179
7.1.5. .status.conditions	181
7.1.6. .status.conditions[]	181
7.2. API ENDPOINTS	182
7.2.1. /apis/apps/v1/daemonsets	183
7.2.2. /apis/apps/v1/watch/daemonsets	183
7.2.3. /apis/apps/v1/namespaces/{namespace}/daemonsets	183
7.2.4. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets	185
7.2.5. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}	186
7.2.6. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets/{name}	189
7.2.7. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status	189
CHAPTER 8. DEPLOYMENT [APPS/V1]	193
8.1. SPECIFICATION	193
8.1.1. .spec	193
8.1.2. .spec.strategy	195
8.1.3. .spec.strategy.rollingUpdate	195
8.1.4. .status	196
8.1.5. .status.conditions	198
8.1.6. .status.conditions[]	198
8.2. API ENDPOINTS	199
8.2.1. /apis/apps/v1/deployments	200
8.2.2. /apis/apps/v1/watch/deployments	200
8.2.3. /apis/apps/v1/namespaces/{namespace}/deployments	200
8.2.4. /apis/apps/v1/watch/namespaces/{namespace}/deployments	202
8.2.5. /apis/apps/v1/namespaces/{namespace}/deployments/{name}	203
8.2.6. /apis/apps/v1/watch/namespaces/{namespace}/deployments/{name}	206
8.2.7. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status	206
CHAPTER 9. DEPLOYMENTCONFIG [APPS.OPENSIFT.IO/V1]	210
9.1. SPECIFICATION	210
9.1.1. .spec	211
9.1.2. .spec.strategy	213

9.1.3. .spec.strategy.customParams	214
9.1.4. .spec.strategy.recreateParams	214
9.1.5. .spec.strategy.recreateParams.mid	215
9.1.6. .spec.strategy.recreateParams.mid.execNewPod	215
9.1.7. .spec.strategy.recreateParams.mid.tagImages	216
9.1.8. .spec.strategy.recreateParams.mid.tagImages[]	216
9.1.9. .spec.strategy.recreateParams.post	217
9.1.10. .spec.strategy.recreateParams.post.execNewPod	217
9.1.11. .spec.strategy.recreateParams.post.tagImages	218
9.1.12. .spec.strategy.recreateParams.post.tagImages[]	218
9.1.13. .spec.strategy.recreateParams.pre	219
9.1.14. .spec.strategy.recreateParams.pre.execNewPod	220
9.1.15. .spec.strategy.recreateParams.pre.tagImages	220
9.1.16. .spec.strategy.recreateParams.pre.tagImages[]	220
9.1.17. .spec.strategy.rollingParams	221
9.1.18. .spec.strategy.rollingParams.post	223
9.1.19. .spec.strategy.rollingParams.post.execNewPod	224
9.1.20. .spec.strategy.rollingParams.post.tagImages	224
9.1.21. .spec.strategy.rollingParams.post.tagImages[]	224
9.1.22. .spec.strategy.rollingParams.pre	225
9.1.23. .spec.strategy.rollingParams.pre.execNewPod	226
9.1.24. .spec.strategy.rollingParams.pre.tagImages	226
9.1.25. .spec.strategy.rollingParams.pre.tagImages[]	227
9.1.26. .spec.triggers	227
9.1.27. .spec.triggers[]	227
9.1.28. .spec.triggers[].imageChangeParams	228
9.1.29. .status	229
9.1.30. .status.conditions	230
9.1.31. .status.conditions[]	230
9.1.32. .status.details	231
9.1.33. .status.details.causes	232
9.1.34. .status.details.causes[]	232
9.1.35. .status.details.causes[].imageTrigger	232
9.2. API ENDPOINTS	233
9.2.1. /apis/apps.openshift.io/v1/deploymentconfigs	234
9.2.2. /apis/apps.openshift.io/v1/watch/deploymentconfigs	234
9.2.3. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs	234
9.2.4. /apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs	236
9.2.5. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}	237
9.2.6. /apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs/{name}	240
9.2.7. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/status	240
CHAPTER 10. DEPLOYMENTCONFIGROLLBACK [APPS.OPENSIFT.IO/V1]	244
10.1. SPECIFICATION	244
10.1.1. .spec	245
10.2. API ENDPOINTS	246
10.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/rollback	246
CHAPTER 11. DEPLOYMENTLOG [APPS.OPENSIFT.IO/V1]	248
11.1. SPECIFICATION	248
11.2. API ENDPOINTS	248
11.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/log	248
CHAPTER 12. DEPLOYMENTREQUEST [APPS.OPENSIFT.IO/V1]	250

12.1. SPECIFICATION	250
12.2. API ENDPOINTS	251
12.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/instantiate	251
CHAPTER 13. JOB [BATCH/V1]	253
13.1. SPECIFICATION	253
13.1.1. .spec	253
13.1.2. .spec.podFailurePolicy	259
13.1.3. .spec.podFailurePolicy.rules	260
13.1.4. .spec.podFailurePolicy.rules[]	260
13.1.5. .spec.podFailurePolicy.rules[].onExitCodes	262
13.1.6. .spec.podFailurePolicy.rules[].onPodConditions	263
13.1.7. .spec.podFailurePolicy.rules[].onPodConditions[]	264
13.1.8. .spec.successPolicy	264
13.1.9. .spec.successPolicy.rules	265
13.1.10. .spec.successPolicy.rules[]	265
13.1.11. .status	266
13.1.12. .status.conditions	270
13.1.13. .status.conditions[]	270
13.1.14. .status.uncountedTerminatedPods	271
13.2. API ENDPOINTS	271
13.2.1. /apis/batch/v1/jobs	272
13.2.2. /apis/batch/v1/watch/jobs	272
13.2.3. /apis/batch/v1/namespaces/{namespace}/jobs	273
13.2.4. /apis/batch/v1/watch/namespaces/{namespace}/jobs	275
13.2.5. /apis/batch/v1/namespaces/{namespace}/jobs/{name}	275
13.2.6. /apis/batch/v1/watch/namespaces/{namespace}/jobs/{name}	278
13.2.7. /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status	279
CHAPTER 14. POD [V1]	282
14.1. SPECIFICATION	282
14.1.1. .spec	283
14.1.2. .spec.affinity	294
14.1.3. .spec.affinity.nodeAffinity	294
14.1.4. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	295
14.1.5. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	295
14.1.6. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference	296
14.1.7. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions	296
14.1.8. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions	297
14.1.9. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields	297
14.1.10. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]	297
14.1.11. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	298
14.1.12. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	299
14.1.13. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]	299
14.1.14. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions	299
14.1.15. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields	300
14.1.16. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]	300

	300
14.1.17. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matc	301
14.1.18. .spec.affinity.podAffinity	301
14.1.19. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	303
14.1.20. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	303
14.1.21. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm	304
14.1.22. .spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	306
14.1.23. .spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	306
14.1.24. .spec.affinity.podAntiAffinity	309
14.1.25. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	310
14.1.26. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	310
14.1.27. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm	311
14.1.28. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	313
14.1.29. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	313
14.1.30. .spec.containers	316
14.1.31. .spec.containers[]	316
14.1.32. .spec.containers[].env	324
14.1.33. .spec.containers[].env[]	324
14.1.34. .spec.containers[].env[].valueFrom	325
14.1.35. .spec.containers[].env[].valueFrom.configMapKeyRef	325
14.1.36. .spec.containers[].env[].valueFrom.fieldRef	326
14.1.37. .spec.containers[].env[].valueFrom.resourceFieldRef	326
14.1.38. .spec.containers[].env[].valueFrom.secretKeyRef	327
14.1.39. .spec.containers[].envFrom	328
14.1.40. .spec.containers[].envFrom[]	328
14.1.41. .spec.containers[].envFrom[].configMapRef	328
14.1.42. .spec.containers[].envFrom[].secretRef	329
14.1.43. .spec.containers[].lifecycle	329
14.1.44. .spec.containers[].lifecycle.postStart	331
14.1.45. .spec.containers[].lifecycle.postStart.exec	331
14.1.46. .spec.containers[].lifecycle.postStart.httpGet	332
14.1.47. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders	333
14.1.48. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]	333
14.1.49. .spec.containers[].lifecycle.postStart.sleep	333
14.1.50. .spec.containers[].lifecycle.postStart.tcpSocket	334
14.1.51. .spec.containers[].lifecycle.preStop	334
14.1.52. .spec.containers[].lifecycle.preStop.exec	335
14.1.53. .spec.containers[].lifecycle.preStop.httpGet	335
14.1.54. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders	336
14.1.55. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]	336
14.1.56. .spec.containers[].lifecycle.preStop.sleep	337
14.1.57. .spec.containers[].lifecycle.preStop.tcpSocket	337
14.1.58. .spec.containers[].livenessProbe	338
14.1.59. .spec.containers[].livenessProbe.exec	339
14.1.60. .spec.containers[].livenessProbe.grpc	340
14.1.61. .spec.containers[].livenessProbe.httpGet	340
14.1.62. .spec.containers[].livenessProbe.httpGet.httpHeaders	341
14.1.63. .spec.containers[].livenessProbe.httpGet.httpHeaders[]	341
14.1.64. .spec.containers[].livenessProbe.tcpSocket	342
14.1.65. .spec.containers[].ports	342
14.1.66. .spec.containers[].ports[]	343

14.1.67. .spec.containers[].readinessProbe	343
14.1.68. .spec.containers[].readinessProbe.exec	345
14.1.69. .spec.containers[].readinessProbe.grpc	346
14.1.70. .spec.containers[].readinessProbe.httpGet	346
14.1.71. .spec.containers[].readinessProbe.httpGet.httpHeaders	347
14.1.72. .spec.containers[].readinessProbe.httpGet.httpHeaders[]	347
14.1.73. .spec.containers[].readinessProbe.tcpSocket	348
14.1.74. .spec.containers[].resizePolicy	348
14.1.75. .spec.containers[].resizePolicy[]	348
14.1.76. .spec.containers[].resources	349
14.1.77. .spec.containers[].resources.claims	350
14.1.78. .spec.containers[].resources.claims[]	350
14.1.79. .spec.containers[].securityContext	351
14.1.80. .spec.containers[].securityContext.appArmorProfile	353
14.1.81. .spec.containers[].securityContext.capabilities	354
14.1.82. .spec.containers[].securityContext.seLinuxOptions	354
14.1.83. .spec.containers[].securityContext.seccompProfile	355
14.1.84. .spec.containers[].securityContext.windowsOptions	356
14.1.85. .spec.containers[].startupProbe	357
14.1.86. .spec.containers[].startupProbe.exec	359
14.1.87. .spec.containers[].startupProbe.grpc	359
14.1.88. .spec.containers[].startupProbe.httpGet	360
14.1.89. .spec.containers[].startupProbe.httpGet.httpHeaders	361
14.1.90. .spec.containers[].startupProbe.httpGet.httpHeaders[]	361
14.1.91. .spec.containers[].startupProbe.tcpSocket	361
14.1.92. .spec.containers[].volumeDevices	362
14.1.93. .spec.containers[].volumeDevices[]	362
14.1.94. .spec.containers[].volumeMounts	362
14.1.95. .spec.containers[].volumeMounts[]	363
14.1.96. .spec.dnsConfig	365
14.1.97. .spec.dnsConfig.options	366
14.1.98. .spec.dnsConfig.options[]	366
14.1.99. .spec.ephemeralContainers	367
14.1.100. .spec.ephemeralContainers[]	367
14.1.101. .spec.ephemeralContainers[].env	374
14.1.102. .spec.ephemeralContainers[].env[]	374
14.1.103. .spec.ephemeralContainers[].env[].valueFrom	375
14.1.104. .spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef	375
14.1.105. .spec.ephemeralContainers[].env[].valueFrom.fieldRef	376
14.1.106. .spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef	376
14.1.107. .spec.ephemeralContainers[].env[].valueFrom.secretKeyRef	377
14.1.108. .spec.ephemeralContainers[].envFrom	378
14.1.109. .spec.ephemeralContainers[].envFrom[]	378
14.1.110. .spec.ephemeralContainers[].envFrom[].configMapRef	378
14.1.111. .spec.ephemeralContainers[].envFrom[].secretRef	379
14.1.112. .spec.ephemeralContainers[].lifecycle	379
14.1.113. .spec.ephemeralContainers[].lifecycle.postStart	381
14.1.114. .spec.ephemeralContainers[].lifecycle.postStart.exec	381
14.1.115. .spec.ephemeralContainers[].lifecycle.postStart.httpGet	382
14.1.116. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders	383
14.1.117. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]	383
14.1.118. .spec.ephemeralContainers[].lifecycle.postStart.sleep	383
14.1.119. .spec.ephemeralContainers[].lifecycle.postStart.tcpSocket	384

14.1.120. .spec.ephemeralContainers[].lifecycle.preStop	384
14.1.121. .spec.ephemeralContainers[].lifecycle.preStop.exec	385
14.1.122. .spec.ephemeralContainers[].lifecycle.preStop.httpGet	385
14.1.123. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders	386
14.1.124. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]	386
14.1.125. .spec.ephemeralContainers[].lifecycle.preStop.sleep	387
14.1.126. .spec.ephemeralContainers[].lifecycle.preStop.tcpSocket	387
14.1.127. .spec.ephemeralContainers[].livenessProbe	388
14.1.128. .spec.ephemeralContainers[].livenessProbe.exec	389
14.1.129. .spec.ephemeralContainers[].livenessProbe.grpc	390
14.1.130. .spec.ephemeralContainers[].livenessProbe.httpGet	390
14.1.131. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders	391
14.1.132. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]	391
14.1.133. .spec.ephemeralContainers[].livenessProbe.tcpSocket	392
14.1.134. .spec.ephemeralContainers[].ports	392
14.1.135. .spec.ephemeralContainers[].ports[]	392
14.1.136. .spec.ephemeralContainers[].readinessProbe	393
14.1.137. .spec.ephemeralContainers[].readinessProbe.exec	395
14.1.138. .spec.ephemeralContainers[].readinessProbe.grpc	396
14.1.139. .spec.ephemeralContainers[].readinessProbe.httpGet	396
14.1.140. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders	397
14.1.141. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]	397
14.1.142. .spec.ephemeralContainers[].readinessProbe.tcpSocket	398
14.1.143. .spec.ephemeralContainers[].resizePolicy	398
14.1.144. .spec.ephemeralContainers[].resizePolicy[]	398
14.1.145. .spec.ephemeralContainers[].resources	399
14.1.146. .spec.ephemeralContainers[].resources.claims	400
14.1.147. .spec.ephemeralContainers[].resources.claims[]	400
14.1.148. .spec.ephemeralContainers[].securityContext	401
14.1.149. .spec.ephemeralContainers[].securityContext.appArmorProfile	403
14.1.150. .spec.ephemeralContainers[].securityContext.capabilities	404
14.1.151. .spec.ephemeralContainers[].securityContext.seLinuxOptions	404
14.1.152. .spec.ephemeralContainers[].securityContext.seccompProfile	405
14.1.153. .spec.ephemeralContainers[].securityContext.windowsOptions	406
14.1.154. .spec.ephemeralContainers[].startupProbe	407
14.1.155. .spec.ephemeralContainers[].startupProbe.exec	409
14.1.156. .spec.ephemeralContainers[].startupProbe.grpc	409
14.1.157. .spec.ephemeralContainers[].startupProbe.httpGet	410
14.1.158. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders	411
14.1.159. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]	411
14.1.160. .spec.ephemeralContainers[].startupProbe.tcpSocket	411
14.1.161. .spec.ephemeralContainers[].volumeDevices	412
14.1.162. .spec.ephemeralContainers[].volumeDevices[]	412
14.1.163. .spec.ephemeralContainers[].volumeMounts	412
14.1.164. .spec.ephemeralContainers[].volumeMounts[]	413
14.1.165. .spec.hostAliases	415
14.1.166. .spec.hostAliases[]	416
14.1.167. .spec.imagePullSecrets	416
14.1.168. .spec.imagePullSecrets[]	416
14.1.169. .spec.initContainers	417
14.1.170. .spec.initContainers[]	417
14.1.171. .spec.initContainers[].env	425
14.1.172. .spec.initContainers[].env[]	425

14.1.173. .spec.initContainers[].env[].valueFrom	426
14.1.174. .spec.initContainers[].env[].valueFrom.configMapKeyRef	426
14.1.175. .spec.initContainers[].env[].valueFrom.fieldRef	427
14.1.176. .spec.initContainers[].env[].valueFrom.resourceFieldRef	427
14.1.177. .spec.initContainers[].env[].valueFrom.secretKeyRef	428
14.1.178. .spec.initContainers[].envFrom	429
14.1.179. .spec.initContainers[].envFrom[]	429
14.1.180. .spec.initContainers[].envFrom[].configMapRef	429
14.1.181. .spec.initContainers[].envFrom[].secretRef	430
14.1.182. .spec.initContainers[].lifecycle	430
14.1.183. .spec.initContainers[].lifecycle.postStart	432
14.1.184. .spec.initContainers[].lifecycle.postStart.exec	432
14.1.185. .spec.initContainers[].lifecycle.postStart.httpGet	433
14.1.186. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders	434
14.1.187. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]	434
14.1.188. .spec.initContainers[].lifecycle.postStart.sleep	434
14.1.189. .spec.initContainers[].lifecycle.postStart.tcpSocket	435
14.1.190. .spec.initContainers[].lifecycle.preStop	435
14.1.191. .spec.initContainers[].lifecycle.preStop.exec	436
14.1.192. .spec.initContainers[].lifecycle.preStop.httpGet	436
14.1.193. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders	437
14.1.194. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]	437
14.1.195. .spec.initContainers[].lifecycle.preStop.sleep	438
14.1.196. .spec.initContainers[].lifecycle.preStop.tcpSocket	438
14.1.197. .spec.initContainers[].livenessProbe	439
14.1.198. .spec.initContainers[].livenessProbe.exec	440
14.1.199. .spec.initContainers[].livenessProbe.grpc	441
14.1.200. .spec.initContainers[].livenessProbe.httpGet	441
14.1.201. .spec.initContainers[].livenessProbe.httpGet.httpHeaders	442
14.1.202. .spec.initContainers[].livenessProbe.httpGet.httpHeaders[]	442
14.1.203. .spec.initContainers[].livenessProbe.tcpSocket	443
14.1.204. .spec.initContainers[].ports	443
14.1.205. .spec.initContainers[].ports[]	444
14.1.206. .spec.initContainers[].readinessProbe	444
14.1.207. .spec.initContainers[].readinessProbe.exec	446
14.1.208. .spec.initContainers[].readinessProbe.grpc	447
14.1.209. .spec.initContainers[].readinessProbe.httpGet	447
14.1.210. .spec.initContainers[].readinessProbe.httpGet.httpHeaders	448
14.1.211. .spec.initContainers[].readinessProbe.httpGet.httpHeaders[]	448
14.1.212. .spec.initContainers[].readinessProbe.tcpSocket	449
14.1.213. .spec.initContainers[].resizePolicy	449
14.1.214. .spec.initContainers[].resizePolicy[]	449
14.1.215. .spec.initContainers[].resources	450
14.1.216. .spec.initContainers[].resources.claims	451
14.1.217. .spec.initContainers[].resources.claims[]	451
14.1.218. .spec.initContainers[].securityContext	452
14.1.219. .spec.initContainers[].securityContext.appArmorProfile	454
14.1.220. .spec.initContainers[].securityContext.capabilities	455
14.1.221. .spec.initContainers[].securityContext.seLinuxOptions	455
14.1.222. .spec.initContainers[].securityContext.seccompProfile	456
14.1.223. .spec.initContainers[].securityContext.windowsOptions	457
14.1.224. .spec.initContainers[].startupProbe	458
14.1.225. .spec.initContainers[].startupProbe.exec	460

14.1.226. .spec.initContainers[].startupProbe.grpc	460
14.1.227. .spec.initContainers[].startupProbe.httpGet	461
14.1.228. .spec.initContainers[].startupProbe.httpGet.httpHeaders	462
14.1.229. .spec.initContainers[].startupProbe.httpGet.httpHeaders[]	462
14.1.230. .spec.initContainers[].startupProbe.tcpSocket	462
14.1.231. .spec.initContainers[].volumeDevices	463
14.1.232. .spec.initContainers[].volumeDevices[]	463
14.1.233. .spec.initContainers[].volumeMounts	463
14.1.234. .spec.initContainers[].volumeMounts[]	464
14.1.235. .spec.os	466
14.1.236. .spec.readinessGates	467
14.1.237. .spec.readinessGates[]	467
14.1.238. .spec.resourceClaims	468
14.1.239. .spec.resourceClaims[]	468
14.1.240. .spec.resources	469
14.1.241. .spec.resources.claims	470
14.1.242. .spec.resources.claims[]	470
14.1.243. .spec.schedulingGates	471
14.1.244. .spec.schedulingGates[]	471
14.1.245. .spec.securityContext	472
14.1.246. .spec.securityContext.appArmorProfile	477
14.1.247. .spec.securityContext.seLinuxOptions	478
14.1.248. .spec.securityContext.seccompProfile	478
14.1.249. .spec.securityContext.sysctls	479
14.1.250. .spec.securityContext.sysctls[]	480
14.1.251. .spec.securityContext.windowsOptions	480
14.1.252. .spec.tolerations	481
14.1.253. .spec.tolerations[]	481
14.1.254. .spec.topologySpreadConstraints	483
14.1.255. .spec.topologySpreadConstraints[]	483
14.1.256. .spec.volumes	489
14.1.257. .spec.volumes[]	489
14.1.258. .spec.volumes[].awsElasticBlockStore	494
14.1.259. .spec.volumes[].azureDisk	495
14.1.260. .spec.volumes[].azureFile	496
14.1.261. .spec.volumes[].cephfs	497
14.1.262. .spec.volumes[].cephfs.secretRef	498
14.1.263. .spec.volumes[].cinder	498
14.1.264. .spec.volumes[].cinder.secretRef	499
14.1.265. .spec.volumes[].configMap	500
14.1.266. .spec.volumes[].configMap.items	501
14.1.267. .spec.volumes[].configMap.items[]	501
14.1.268. .spec.volumes[].csi	502
14.1.269. .spec.volumes[].csi.nodePublishSecretRef	503
14.1.270. .spec.volumes[].downwardAPI	504
14.1.271. .spec.volumes[].downwardAPI.items	504
14.1.272. .spec.volumes[].downwardAPI.items[]	504
14.1.273. .spec.volumes[].downwardAPI.items[].fieldRef	505
14.1.274. .spec.volumes[].downwardAPI.items[].resourceFieldRef	506
14.1.275. .spec.volumes[].emptyDir	506
14.1.276. .spec.volumes[].ephemeral	507
14.1.277. .spec.volumes[].ephemeral.volumeClaimTemplate	507
14.1.278. .spec.volumes[].ephemeral.volumeClaimTemplate.spec	508

14.1.279. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource	510
14.1.280. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef	511
14.1.281. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources	511
14.1.282. .spec.volumes[].fc	512
14.1.283. .spec.volumes[].flexVolume	513
14.1.284. .spec.volumes[].flexVolume.secretRef	513
14.1.285. .spec.volumes[].flocker	514
14.1.286. .spec.volumes[].gcePersistentDisk	514
14.1.287. .spec.volumes[].gitRepo	515
14.1.288. .spec.volumes[].glusterfs	516
14.1.289. .spec.volumes[].hostPath	517
14.1.290. .spec.volumes[].image	518
14.1.291. .spec.volumes[].iscsi	519
14.1.292. .spec.volumes[].iscsi.secretRef	521
14.1.293. .spec.volumes[].nfs	521
14.1.294. .spec.volumes[].persistentVolumeClaim	522
14.1.295. .spec.volumes[].photonPersistentDisk	523
14.1.296. .spec.volumes[].portworxVolume	523
14.1.297. .spec.volumes[].projected	524
14.1.298. .spec.volumes[].projected.sources	525
14.1.299. .spec.volumes[].projected.sources[]	525
14.1.300. .spec.volumes[].projected.sources[].clusterTrustBundle	526
14.1.301. .spec.volumes[].projected.sources[].configMap	527
14.1.302. .spec.volumes[].projected.sources[].configMap.items	528
14.1.303. .spec.volumes[].projected.sources[].configMap.items[]	528
14.1.304. .spec.volumes[].projected.sources[].downwardAPI	529
14.1.305. .spec.volumes[].projected.sources[].downwardAPI.items	530
14.1.306. .spec.volumes[].projected.sources[].downwardAPI.items[]	530
14.1.307. .spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef	530
14.1.308. .spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef	531
14.1.309. .spec.volumes[].projected.sources[].secret	531
14.1.310. .spec.volumes[].projected.sources[].secret.items	532
14.1.311. .spec.volumes[].projected.sources[].secret.items[]	532
14.1.312. .spec.volumes[].projected.sources[].serviceAccountToken	533
14.1.313. .spec.volumes[].quobyte	534
14.1.314. .spec.volumes[].rbd	535
14.1.315. .spec.volumes[].rbd.secretRef	537
14.1.316. .spec.volumes[].scaleIO	537
14.1.317. .spec.volumes[].scaleIO.secretRef	538
14.1.318. .spec.volumes[].secret	539
14.1.319. .spec.volumes[].secret.items	540
14.1.320. .spec.volumes[].secret.items[]	540
14.1.321. .spec.volumes[].storageos	541
14.1.322. .spec.volumes[].storageos.secretRef	542
14.1.323. .spec.volumes[].vsphereVolume	542
14.1.324. .status	543
14.1.325. .status.conditions	549
14.1.326. .status.conditions[]	549
14.1.327. .status.containerStatuses	550
14.1.328. .status.containerStatuses[]	551
14.1.329. .status.containerStatuses[].allocatedResourcesStatus	555
14.1.330. .status.containerStatuses[].allocatedResourcesStatus[]	555
14.1.331. .status.containerStatuses[].allocatedResourcesStatus[].resources	556

14.1.332. .status.containerStatuses[].allocatedResourcesStatus[].resources[]	556
14.1.333. .status.containerStatuses[].lastState	557
14.1.334. .status.containerStatuses[].lastState.running	557
14.1.335. .status.containerStatuses[].lastState.terminated	558
14.1.336. .status.containerStatuses[].lastState.waiting	558
14.1.337. .status.containerStatuses[].resources	559
14.1.338. .status.containerStatuses[].resources.claims	560
14.1.339. .status.containerStatuses[].resources.claims[]	560
14.1.340. .status.containerStatuses[].state	561
14.1.341. .status.containerStatuses[].state.running	561
14.1.342. .status.containerStatuses[].state.terminated	561
14.1.343. .status.containerStatuses[].state.waiting	562
14.1.344. .status.containerStatuses[].user	562
14.1.345. .status.containerStatuses[].user.linux	563
14.1.346. .status.containerStatuses[].volumeMounts	563
14.1.347. .status.containerStatuses[].volumeMounts[]	563
14.1.348. .status.ephemeralContainerStatuses	564
14.1.349. .status.ephemeralContainerStatuses[]	564
14.1.350. .status.ephemeralContainerStatuses[].allocatedResourcesStatus	569
14.1.351. .status.ephemeralContainerStatuses[].allocatedResourcesStatus[]	569
14.1.352. .status.ephemeralContainerStatuses[].allocatedResourcesStatus[].resources	570
14.1.353. .status.ephemeralContainerStatuses[].allocatedResourcesStatus[].resources[]	570
14.1.354. .status.ephemeralContainerStatuses[].lastState	571
14.1.355. .status.ephemeralContainerStatuses[].lastState.running	571
14.1.356. .status.ephemeralContainerStatuses[].lastState.terminated	572
14.1.357. .status.ephemeralContainerStatuses[].lastState.waiting	572
14.1.358. .status.ephemeralContainerStatuses[].resources	573
14.1.359. .status.ephemeralContainerStatuses[].resources.claims	574
14.1.360. .status.ephemeralContainerStatuses[].resources.claims[]	574
14.1.361. .status.ephemeralContainerStatuses[].state	575
14.1.362. .status.ephemeralContainerStatuses[].state.running	575
14.1.363. .status.ephemeralContainerStatuses[].state.terminated	575
14.1.364. .status.ephemeralContainerStatuses[].state.waiting	576
14.1.365. .status.ephemeralContainerStatuses[].user	576
14.1.366. .status.ephemeralContainerStatuses[].user.linux	577
14.1.367. .status.ephemeralContainerStatuses[].volumeMounts	577
14.1.368. .status.ephemeralContainerStatuses[].volumeMounts[]	577
14.1.369. .status.hostIPs	578
14.1.370. .status.hostIPs[]	578
14.1.371. .status.initContainerStatuses	579
14.1.372. .status.initContainerStatuses[]	579
14.1.373. .status.initContainerStatuses[].allocatedResourcesStatus	583
14.1.374. .status.initContainerStatuses[].allocatedResourcesStatus[]	583
14.1.375. .status.initContainerStatuses[].allocatedResourcesStatus[].resources	584
14.1.376. .status.initContainerStatuses[].allocatedResourcesStatus[].resources[]	584
14.1.377. .status.initContainerStatuses[].lastState	585
14.1.378. .status.initContainerStatuses[].lastState.running	585
14.1.379. .status.initContainerStatuses[].lastState.terminated	586
14.1.380. .status.initContainerStatuses[].lastState.waiting	586
14.1.381. .status.initContainerStatuses[].resources	587
14.1.382. .status.initContainerStatuses[].resources.claims	588
14.1.383. .status.initContainerStatuses[].resources.claims[]	588
14.1.384. .status.initContainerStatuses[].state	589

14.1.385. .status.initContainerStatuses[].state.running	589
14.1.386. .status.initContainerStatuses[].state.terminated	589
14.1.387. .status.initContainerStatuses[].state.waiting	590
14.1.388. .status.initContainerStatuses[].user	590
14.1.389. .status.initContainerStatuses[].user.linux	591
14.1.390. .status.initContainerStatuses[].volumeMounts	591
14.1.391. .status.initContainerStatuses[].volumeMounts[]	591
14.1.392. .status.podIPs	592
14.1.393. .status.podIPs[]	592
14.1.394. .status.resourceClaimStatuses	593
14.1.395. .status.resourceClaimStatuses[]	593
14.2. API ENDPOINTS	593
14.2.1. /api/v1/pods	595
14.2.2. /api/v1/watch/pods	595
14.2.3. /api/v1/namespaces/{namespace}/pods	595
14.2.4. /api/v1/watch/namespaces/{namespace}/pods	597
14.2.5. /api/v1/namespaces/{namespace}/pods/{name}	598
14.2.6. /api/v1/namespaces/{namespace}/pods/{name}/log	601
14.2.7. /api/v1/watch/namespaces/{namespace}/pods/{name}	601
14.2.8. /api/v1/namespaces/{namespace}/pods/{name}/resize	602
14.2.9. /api/v1/namespaces/{namespace}/pods/{name}/status	604
14.2.10. /api/v1/namespaces/{namespace}/pods/{name}/ephemeralcontainers	607
CHAPTER 15. REPLICATIONCONTROLLER [V1]	610
15.1. SPECIFICATION	610
15.1.1. .spec	611
15.1.2. .spec.template	612
15.1.3. .spec.template.spec	612
15.1.4. .spec.template.spec.affinity	624
15.1.5. .spec.template.spec.affinity.nodeAffinity	624
15.1.6. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	625
15.1.7. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	625
15.1.8. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preferredDuringSchedulingIgnoredDuringExecution	626
15.1.9. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preferredDuringSchedulingIgnoredDuringExecution	626
15.1.10. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preferredDuringSchedulingIgnoredDuringExecution	627
15.1.11. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preferredDuringSchedulingIgnoredDuringExecution	627
15.1.12. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preferredDuringSchedulingIgnoredDuringExecution	627
15.1.13. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	628
15.1.14. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	629
15.1.15. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	629
15.1.16. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	629
15.1.17. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	630
15.1.18. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	630
15.1.19. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector	630

	631
15.1.20. .spec.template.spec.affinity.podAffinity	631
15.1.21. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	633
15.1.22. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	633
15.1.23. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[],podAffinit	634
15.1.24. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	636
15.1.25. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	636
15.1.26. .spec.template.spec.affinity.podAntiAffinity	639
15.1.27. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	640
15.1.28. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	640
15.1.29. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[],podAf	641
15.1.30. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	643
15.1.31. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	643
15.1.32. .spec.template.spec.containers	646
15.1.33. .spec.template.spec.containers[]	646
15.1.34. .spec.template.spec.containers[].env	654
15.1.35. .spec.template.spec.containers[].env[]	654
15.1.36. .spec.template.spec.containers[].env[].valueFrom	655
15.1.37. .spec.template.spec.containers[].env[].valueFrom.configMapKeyRef	655
15.1.38. .spec.template.spec.containers[].env[].valueFrom.fieldRef	656
15.1.39. .spec.template.spec.containers[].env[].valueFrom.resourceFieldRef	656
15.1.40. .spec.template.spec.containers[].env[].valueFrom.secretKeyRef	657
15.1.41. .spec.template.spec.containers[].envFrom	658
15.1.42. .spec.template.spec.containers[].envFrom[]	658
15.1.43. .spec.template.spec.containers[].envFrom[].configMapRef	658
15.1.44. .spec.template.spec.containers[].envFrom[].secretRef	659
15.1.45. .spec.template.spec.containers[].lifecycle	659
15.1.46. .spec.template.spec.containers[].lifecycle.postStart	661
15.1.47. .spec.template.spec.containers[].lifecycle.postStart.exec	661
15.1.48. .spec.template.spec.containers[].lifecycle.postStart.httpGet	662
15.1.49. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders	663
15.1.50. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]	663
15.1.51. .spec.template.spec.containers[].lifecycle.postStart.sleep	663
15.1.52. .spec.template.spec.containers[].lifecycle.postStart.tcpSocket	664
15.1.53. .spec.template.spec.containers[].lifecycle.preStop	664
15.1.54. .spec.template.spec.containers[].lifecycle.preStop.exec	665
15.1.55. .spec.template.spec.containers[].lifecycle.preStop.httpGet	665
15.1.56. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders	666
15.1.57. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]	666
15.1.58. .spec.template.spec.containers[].lifecycle.preStop.sleep	667
15.1.59. .spec.template.spec.containers[].lifecycle.preStop.tcpSocket	667
15.1.60. .spec.template.spec.containers[].livenessProbe	668
15.1.61. .spec.template.spec.containers[].livenessProbe.exec	669
15.1.62. .spec.template.spec.containers[].livenessProbe.grpc	670
15.1.63. .spec.template.spec.containers[].livenessProbe.httpGet	670
15.1.64. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders	671
15.1.65. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]	671
15.1.66. .spec.template.spec.containers[].livenessProbe.tcpSocket	672
15.1.67. .spec.template.spec.containers[].ports	672
15.1.68. .spec.template.spec.containers[].ports[]	673
15.1.69. .spec.template.spec.containers[].readinessProbe	673

15.1.70. .spec.template.spec.containers[].readinessProbe.exec	675
15.1.71. .spec.template.spec.containers[].readinessProbe.grpc	676
15.1.72. .spec.template.spec.containers[].readinessProbe.httpGet	676
15.1.73. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders	677
15.1.74. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]	677
15.1.75. .spec.template.spec.containers[].readinessProbe.tcpSocket	678
15.1.76. .spec.template.spec.containers[].resizePolicy	678
15.1.77. .spec.template.spec.containers[].resizePolicy[]	678
15.1.78. .spec.template.spec.containers[].resources	679
15.1.79. .spec.template.spec.containers[].resources.claims	680
15.1.80. .spec.template.spec.containers[].resources.claims[]	680
15.1.81. .spec.template.spec.containers[].securityContext	681
15.1.82. .spec.template.spec.containers[].securityContext.appArmorProfile	683
15.1.83. .spec.template.spec.containers[].securityContext.capabilities	684
15.1.84. .spec.template.spec.containers[].securityContext.seLinuxOptions	684
15.1.85. .spec.template.spec.containers[].securityContext.seccompProfile	685
15.1.86. .spec.template.spec.containers[].securityContext.windowsOptions	686
15.1.87. .spec.template.spec.containers[].startupProbe	687
15.1.88. .spec.template.spec.containers[].startupProbe.exec	689
15.1.89. .spec.template.spec.containers[].startupProbe.grpc	689
15.1.90. .spec.template.spec.containers[].startupProbe.httpGet	690
15.1.91. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders	691
15.1.92. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]	691
15.1.93. .spec.template.spec.containers[].startupProbe.tcpSocket	691
15.1.94. .spec.template.spec.containers[].volumeDevices	692
15.1.95. .spec.template.spec.containers[].volumeDevices[]	692
15.1.96. .spec.template.spec.containers[].volumeMounts	692
15.1.97. .spec.template.spec.containers[].volumeMounts[]	693
15.1.98. .spec.template.spec.dnsConfig	695
15.1.99. .spec.template.spec.dnsConfig.options	696
15.1.100. .spec.template.spec.dnsConfig.options[]	696
15.1.101. .spec.template.spec.ephemeralContainers	697
15.1.102. .spec.template.spec.ephemeralContainers[]	697
15.1.103. .spec.template.spec.ephemeralContainers[].env	704
15.1.104. .spec.template.spec.ephemeralContainers[].env[]	704
15.1.105. .spec.template.spec.ephemeralContainers[].env[].valueFrom	705
15.1.106. .spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef	705
15.1.107. .spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef	706
15.1.108. .spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef	706
15.1.109. .spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef	707
15.1.110. .spec.template.spec.ephemeralContainers[].envFrom	708
15.1.111. .spec.template.spec.ephemeralContainers[].envFrom[]	708
15.1.112. .spec.template.spec.ephemeralContainers[].envFrom[].configMapRef	708
15.1.113. .spec.template.spec.ephemeralContainers[].envFrom[].secretRef	709
15.1.114. .spec.template.spec.ephemeralContainers[].lifecycle	709
15.1.115. .spec.template.spec.ephemeralContainers[].lifecycle.postStart	711
15.1.116. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec	711
15.1.117. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet	712
15.1.118. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders	713
15.1.119. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]	713
15.1.120. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.sleep	713
15.1.121. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket	714
15.1.122. .spec.template.spec.ephemeralContainers[].lifecycle.preStop	714

15.1.123. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec	715
15.1.124. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet	715
15.1.125. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders	716
15.1.126. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]	716
15.1.127. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.sleep	717
15.1.128. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket	717
15.1.129. .spec.template.spec.ephemeralContainers[].livenessProbe	718
15.1.130. .spec.template.spec.ephemeralContainers[].livenessProbe.exec	719
15.1.131. .spec.template.spec.ephemeralContainers[].livenessProbe.grpc	720
15.1.132. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet	720
15.1.133. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders	721
15.1.134. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]	721
15.1.135. .spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket	722
15.1.136. .spec.template.spec.ephemeralContainers[].ports	722
15.1.137. .spec.template.spec.ephemeralContainers[].ports[]	722
15.1.138. .spec.template.spec.ephemeralContainers[].readinessProbe	723
15.1.139. .spec.template.spec.ephemeralContainers[].readinessProbe.exec	725
15.1.140. .spec.template.spec.ephemeralContainers[].readinessProbe.grpc	726
15.1.141. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet	726
15.1.142. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders	727
15.1.143. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]	727
15.1.144. .spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket	728
15.1.145. .spec.template.spec.ephemeralContainers[].resizePolicy	728
15.1.146. .spec.template.spec.ephemeralContainers[].resizePolicy[]	728
15.1.147. .spec.template.spec.ephemeralContainers[].resources	729
15.1.148. .spec.template.spec.ephemeralContainers[].resources.claims	730
15.1.149. .spec.template.spec.ephemeralContainers[].resources.claims[]	730
15.1.150. .spec.template.spec.ephemeralContainers[].securityContext	731
15.1.151. .spec.template.spec.ephemeralContainers[].securityContext.appArmorProfile	733
15.1.152. .spec.template.spec.ephemeralContainers[].securityContext.capabilities	734
15.1.153. .spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions	734
15.1.154. .spec.template.spec.ephemeralContainers[].securityContext.seccompProfile	735
15.1.155. .spec.template.spec.ephemeralContainers[].securityContext.windowsOptions	736
15.1.156. .spec.template.spec.ephemeralContainers[].startupProbe	737
15.1.157. .spec.template.spec.ephemeralContainers[].startupProbe.exec	739
15.1.158. .spec.template.spec.ephemeralContainers[].startupProbe.grpc	739
15.1.159. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet	740
15.1.160. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders	741
15.1.161. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]	741
15.1.162. .spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket	741
15.1.163. .spec.template.spec.ephemeralContainers[].volumeDevices	742
15.1.164. .spec.template.spec.ephemeralContainers[].volumeDevices[]	742
15.1.165. .spec.template.spec.ephemeralContainers[].volumeMounts	742
15.1.166. .spec.template.spec.ephemeralContainers[].volumeMounts[]	743
15.1.167. .spec.template.spec.hostAliases	745
15.1.168. .spec.template.spec.hostAliases[]	746
15.1.169. .spec.template.spec.imagePullSecrets	746
15.1.170. .spec.template.spec.imagePullSecrets[]	746
15.1.171. .spec.template.spec.initContainers	747
15.1.172. .spec.template.spec.initContainers[]	747
15.1.173. .spec.template.spec.initContainers[].env	755
15.1.174. .spec.template.spec.initContainers[].env[]	755
15.1.175. .spec.template.spec.initContainers[].env[].valueFrom	756

15.1.176. .spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef	756
15.1.177. .spec.template.spec.initContainers[].env[].valueFrom.fieldRef	757
15.1.178. .spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef	757
15.1.179. .spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef	758
15.1.180. .spec.template.spec.initContainers[].envFrom	759
15.1.181. .spec.template.spec.initContainers[].envFrom[]	759
15.1.182. .spec.template.spec.initContainers[].envFrom[].configMapRef	759
15.1.183. .spec.template.spec.initContainers[].envFrom[].secretRef	760
15.1.184. .spec.template.spec.initContainers[].lifecycle	760
15.1.185. .spec.template.spec.initContainers[].lifecycle.postStart	762
15.1.186. .spec.template.spec.initContainers[].lifecycle.postStart.exec	762
15.1.187. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet	763
15.1.188. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders	764
15.1.189. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]	764
15.1.190. .spec.template.spec.initContainers[].lifecycle.postStart.sleep	764
15.1.191. .spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket	765
15.1.192. .spec.template.spec.initContainers[].lifecycle.preStop	765
15.1.193. .spec.template.spec.initContainers[].lifecycle.preStop.exec	766
15.1.194. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet	766
15.1.195. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders	767
15.1.196. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]	767
15.1.197. .spec.template.spec.initContainers[].lifecycle.preStop.sleep	768
15.1.198. .spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket	768
15.1.199. .spec.template.spec.initContainers[].livenessProbe	769
15.1.200. .spec.template.spec.initContainers[].livenessProbe.exec	770
15.1.201. .spec.template.spec.initContainers[].livenessProbe.grpc	771
15.1.202. .spec.template.spec.initContainers[].livenessProbe.httpGet	771
15.1.203. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders	772
15.1.204. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]	772
15.1.205. .spec.template.spec.initContainers[].livenessProbe.tcpSocket	773
15.1.206. .spec.template.spec.initContainers[].ports	773
15.1.207. .spec.template.spec.initContainers[].ports[]	774
15.1.208. .spec.template.spec.initContainers[].readinessProbe	774
15.1.209. .spec.template.spec.initContainers[].readinessProbe.exec	776
15.1.210. .spec.template.spec.initContainers[].readinessProbe.grpc	777
15.1.211. .spec.template.spec.initContainers[].readinessProbe.httpGet	777
15.1.212. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders	778
15.1.213. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]	778
15.1.214. .spec.template.spec.initContainers[].readinessProbe.tcpSocket	779
15.1.215. .spec.template.spec.initContainers[].resizePolicy	779
15.1.216. .spec.template.spec.initContainers[].resizePolicy[]	779
15.1.217. .spec.template.spec.initContainers[].resources	780
15.1.218. .spec.template.spec.initContainers[].resources.claims	781
15.1.219. .spec.template.spec.initContainers[].resources.claims[]	781
15.1.220. .spec.template.spec.initContainers[].securityContext	782
15.1.221. .spec.template.spec.initContainers[].securityContext.appArmorProfile	784
15.1.222. .spec.template.spec.initContainers[].securityContext.capabilities	785
15.1.223. .spec.template.spec.initContainers[].securityContext.seLinuxOptions	785
15.1.224. .spec.template.spec.initContainers[].securityContext.seccompProfile	786
15.1.225. .spec.template.spec.initContainers[].securityContext.windowsOptions	787
15.1.226. .spec.template.spec.initContainers[].startupProbe	788
15.1.227. .spec.template.spec.initContainers[].startupProbe.exec	790
15.1.228. .spec.template.spec.initContainers[].startupProbe.grpc	790

15.1.229. .spec.template.spec.initContainers[].startupProbe.httpGet	791
15.1.230. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders	792
15.1.231. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]	792
15.1.232. .spec.template.spec.initContainers[].startupProbe.tcpSocket	792
15.1.233. .spec.template.spec.initContainers[].volumeDevices	793
15.1.234. .spec.template.spec.initContainers[].volumeDevices[]	793
15.1.235. .spec.template.spec.initContainers[].volumeMounts	793
15.1.236. .spec.template.spec.initContainers[].volumeMounts[]	794
15.1.237. .spec.template.spec.os	796
15.1.238. .spec.template.spec.readinessGates	797
15.1.239. .spec.template.spec.readinessGates[]	797
15.1.240. .spec.template.spec.resourceClaims	798
15.1.241. .spec.template.spec.resourceClaims[]	798
15.1.242. .spec.template.spec.resources	799
15.1.243. .spec.template.spec.resources.claims	800
15.1.244. .spec.template.spec.resources.claims[]	800
15.1.245. .spec.template.spec.schedulingGates	801
15.1.246. .spec.template.spec.schedulingGates[]	801
15.1.247. .spec.template.spec.securityContext	802
15.1.248. .spec.template.spec.securityContext.appArmorProfile	807
15.1.249. .spec.template.spec.securityContext.seLinuxOptions	808
15.1.250. .spec.template.spec.securityContext.seccompProfile	808
15.1.251. .spec.template.spec.securityContext.sysctls	809
15.1.252. .spec.template.spec.securityContext.sysctls[]	810
15.1.253. .spec.template.spec.securityContext.windowsOptions	810
15.1.254. .spec.template.spec.tolerations	811
15.1.255. .spec.template.spec.tolerations[]	811
15.1.256. .spec.template.spec.topologySpreadConstraints	813
15.1.257. .spec.template.spec.topologySpreadConstraints[]	813
15.1.258. .spec.template.spec.volumes	819
15.1.259. .spec.template.spec.volumes[]	819
15.1.260. .spec.template.spec.volumes[].awsElasticBlockStore	824
15.1.261. .spec.template.spec.volumes[].azureDisk	825
15.1.262. .spec.template.spec.volumes[].azureFile	826
15.1.263. .spec.template.spec.volumes[].cephfs	827
15.1.264. .spec.template.spec.volumes[].cephfs.secretRef	828
15.1.265. .spec.template.spec.volumes[].cinder	828
15.1.266. .spec.template.spec.volumes[].cinder.secretRef	829
15.1.267. .spec.template.spec.volumes[].configMap	830
15.1.268. .spec.template.spec.volumes[].configMap.items	831
15.1.269. .spec.template.spec.volumes[].configMap.items[]	831
15.1.270. .spec.template.spec.volumes[].csi	832
15.1.271. .spec.template.spec.volumes[].csi.nodePublishSecretRef	833
15.1.272. .spec.template.spec.volumes[].downwardAPI	834
15.1.273. .spec.template.spec.volumes[].downwardAPI.items	834
15.1.274. .spec.template.spec.volumes[].downwardAPI.items[]	834
15.1.275. .spec.template.spec.volumes[].downwardAPI.items[].fieldRef	835
15.1.276. .spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef	836
15.1.277. .spec.template.spec.volumes[].emptyDir	836
15.1.278. .spec.template.spec.volumes[].ephemeral	837
15.1.279. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate	837
15.1.280. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec	838
15.1.281. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource	840

15.1.282. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef	841
15.1.283. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources	841
15.1.284. .spec.template.spec.volumes[].fc	842
15.1.285. .spec.template.spec.volumes[].flexVolume	843
15.1.286. .spec.template.spec.volumes[].flexVolume.secretRef	843
15.1.287. .spec.template.spec.volumes[].flocker	844
15.1.288. .spec.template.spec.volumes[].gcePersistentDisk	844
15.1.289. .spec.template.spec.volumes[].gitRepo	845
15.1.290. .spec.template.spec.volumes[].glusterfs	846
15.1.291. .spec.template.spec.volumes[].hostPath	847
15.1.292. .spec.template.spec.volumes[].image	848
15.1.293. .spec.template.spec.volumes[].iscsi	849
15.1.294. .spec.template.spec.volumes[].iscsi.secretRef	851
15.1.295. .spec.template.spec.volumes[].nfs	851
15.1.296. .spec.template.spec.volumes[].persistentVolumeClaim	852
15.1.297. .spec.template.spec.volumes[].photonPersistentDisk	853
15.1.298. .spec.template.spec.volumes[].portworxVolume	853
15.1.299. .spec.template.spec.volumes[].projected	854
15.1.300. .spec.template.spec.volumes[].projected.sources	855
15.1.301. .spec.template.spec.volumes[].projected.sources[]	855
15.1.302. .spec.template.spec.volumes[].projected.sources[].clusterTrustBundle	856
15.1.303. .spec.template.spec.volumes[].projected.sources[].configMap	857
15.1.304. .spec.template.spec.volumes[].projected.sources[].configMap.items	858
15.1.305. .spec.template.spec.volumes[].projected.sources[].configMap.items[]	858
15.1.306. .spec.template.spec.volumes[].projected.sources[].downwardAPI	859
15.1.307. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items	860
15.1.308. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]	860
15.1.309. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef	860
15.1.310. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef	861
15.1.311. .spec.template.spec.volumes[].projected.sources[].secret	861
15.1.312. .spec.template.spec.volumes[].projected.sources[].secret.items	862
15.1.313. .spec.template.spec.volumes[].projected.sources[].secret.items[]	862
15.1.314. .spec.template.spec.volumes[].projected.sources[].serviceAccountToken	863
15.1.315. .spec.template.spec.volumes[].quobyte	864
15.1.316. .spec.template.spec.volumes[].rbd	865
15.1.317. .spec.template.spec.volumes[].rbd.secretRef	867
15.1.318. .spec.template.spec.volumes[].scaleIO	867
15.1.319. .spec.template.spec.volumes[].scaleIO.secretRef	868
15.1.320. .spec.template.spec.volumes[].secret	869
15.1.321. .spec.template.spec.volumes[].secret.items	870
15.1.322. .spec.template.spec.volumes[].secret.items[]	870
15.1.323. .spec.template.spec.volumes[].storageos	871
15.1.324. .spec.template.spec.volumes[].storageos.secretRef	872
15.1.325. .spec.template.spec.volumes[].vsphereVolume	872
15.1.326. .status	873
15.1.327. .status.conditions	874
15.1.328. .status.conditions[]	874
15.2. API ENDPOINTS	875
15.2.1. /api/v1/replicationcontrollers	876
15.2.2. /api/v1/watch/replicationcontrollers	876
15.2.3. /api/v1/namespaces/{namespace}/replicationcontrollers	877
15.2.4. /api/v1/watch/namespaces/{namespace}/replicationcontrollers	879
15.2.5. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}	879

15.2.6. /api/v1/watch/namespaces/{namespace}/replicationcontrollers/{name}	882
15.2.7. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status	883
CHAPTER 16. REPLICASET [APPS/V1]	886
16.1. SPECIFICATION	886
16.1.1. .spec	887
16.1.2. .status	887
16.1.3. .status.conditions	889
16.1.4. .status.conditions[]	889
16.2. API ENDPOINTS	890
16.2.1. /apis/apps/v1/replicasets	891
16.2.2. /apis/apps/v1/watch/replicasets	891
16.2.3. /apis/apps/v1/namespaces/{namespace}/replicasets	891
16.2.4. /apis/apps/v1/watch/namespaces/{namespace}/replicasets	893
16.2.5. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}	894
16.2.6. /apis/apps/v1/watch/namespaces/{namespace}/replicasets/{name}	897
16.2.7. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}/status	897
CHAPTER 17. STATEFULSET [APPS/V1]	901
17.1. SPECIFICATION	901
17.1.1. .spec	902
17.1.2. .spec.ordinals	905
17.1.3. .spec.persistentVolumeClaimRetentionPolicy	905
17.1.4. .spec.updateStrategy	906
17.1.5. .spec.updateStrategy.rollingUpdate	907
17.1.6. .status	908
17.1.7. .status.conditions	910
17.1.8. .status.conditions[]	910
17.2. API ENDPOINTS	910
17.2.1. /apis/apps/v1/statefulsets	911
17.2.2. /apis/apps/v1/watch/statefulsets	912
17.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets	912
17.2.4. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets	914
17.2.5. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}	914
17.2.6. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets/{name}	917
17.2.7. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status	918

CHAPTER 1. WORKLOADS APIS

1.1. BUILDCONFIG [BUILD.OPENSIFT.IO/V1]

Description

Build configurations define a build process for new container images. There are three types of builds possible – a container image build using a Dockerfile, a Source-to-Image build that uses a specially prepared base image that accepts source code that it can make runnable, and a custom build that can run // arbitrary container images as a base and accept the build parameters. Builds run on the cluster and on completion are pushed to the container image registry specified in the "output" section. A build can be triggered via a webhook, when the base image changes, or when a user manually requests a new build be // created.

Each build created by a build configuration is numbered and refers back to its parent configuration. Multiple builds can be triggered at once. Builds that do not have "output" set can be used to test code or run a verification build.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.2. BUILD [BUILD.OPENSIFT.IO/V1]

Description

Build encapsulates the inputs needed to produce a new deployable image, as well as the status of the execution and a reference to the Pod which executed the build.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.3. BUILDLOG [BUILD.OPENSIFT.IO/V1]

Description

BuildLog is the (unused) resource associated with the build log redirector

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.4. BUILDREQUEST [BUILD.OPENSIFT.IO/V1]

Description

BuildRequest is the resource used to pass parameters to build generator

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.5. CRONJOB [BATCH/V1]

Description

CronJob represents the configuration of a single cron job.

Type

object

1.6. DAEMONSET [APPS/V1]

Description

DaemonSet represents the configuration of a daemon set.

Type

object

1.7. DEPLOYMENT [APPS/V1]

Description

Deployment enables declarative updates for Pods and ReplicaSets.

Type

object

1.8. DEPLOYMENTCONFIG [APPS.OPENSIFT.IO/V1]

Description

Deployment Configs define the template for a pod and manages deploying new images or configuration changes. A single deployment configuration is usually analogous to a single micro-service. Can support many different deployment patterns, including full restart, customizable rolling updates, and fully custom behaviors, as well as pre- and post- deployment hooks. Each individual deployment is represented as a replication controller.

A deployment is "triggered" when its configuration is changed or a tag in an Image Stream is changed. Triggers can be disabled to allow manual control over a deployment. The "strategy" determines how the deployment is carried out and may be changed at any time. The **latestVersion** field is updated when a new deployment is triggered by any means.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer). Deprecated: Use deployments or other means for declarative updates for pods instead.

Type

object

1.9. DEPLOYMENTCONFIGROLLBACK [APPS.OPENSIFT.IO/V1]

Description

DeploymentConfigRollback provides the input to rollback generation.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.10. DEPLOYMENTLOG [APPS.OPENSIFT.IO/V1]

Description

DeploymentLog represents the logs for a deployment

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.11. DEPLOYMENTREQUEST [APPS.OPENSIFT.IO/V1]

Description

DeploymentRequest is a request to a deployment config for a new deployment.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.12. JOB [BATCH/V1]

Description

Job represents the configuration of a single job.

Type

object

1.13. POD [V1]

Description

Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.

Type

object

1.14. REPLICATIONCONTROLLER [V1]

Description

ReplicationController represents the configuration of a replication controller.

Type

object

1.15. REPLICASET [APPS/V1]

Description

ReplicaSet ensures that a specified number of pod replicas are running at any given time.

Type

object

1.16. STATEFULSET [APPS/V1]

Description

StatefulSet represents a set of pods with consistent identities. Identities are defined as: - Network: A single stable DNS and hostname. - Storage: As many VolumeClaims as requested.

The StatefulSet guarantees that a given network identity will always map to the same storage identity.

Type

object

CHAPTER 2. BUILDCONFIG [BUILD.OPENSIFT.IO/V1]

Description

Build configurations define a build process for new container images. There are three types of builds possible – a container image build using a Dockerfile, a Source-to-Image build that uses a specially prepared base image that accepts source code that it can make runnable, and a custom build that can run // arbitrary container images as a base and accept the build parameters. Builds run on the cluster and on completion are pushed to the container image registry specified in the "output" section. A build can be triggered via a webhook, when the base image changes, or when a user manually requests a new build be // created.

Each build created by a build configuration is numbered and refers back to its parent configuration. Multiple builds can be triggered at once. Builds that do not have "output" set can be used to test code or run a verification build.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

2.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	BuildConfigSpec describes when and how builds are created
status	object	BuildConfigStatus contains current state of the build config object.

2.1.1. .spec

Description

BuildConfigSpec describes when and how builds are created

Type

object

Required

- **strategy**

Property	Type	Description
completionDeadlineSeconds	integer	completionDeadlineSeconds is an optional duration in seconds, counted from the time when a build pod gets scheduled in the system, that the build may be active on a node before the system actively tries to terminate the build; value must be positive integer
failedBuildsHistoryLimit	integer	failedBuildsHistoryLimit is the number of old failed builds to retain. When a BuildConfig is created, the 5 most recent failed builds are retained unless this value is set. If removed after the BuildConfig has been created, all failed builds are retained.

Property	Type	Description
mountTrustedCA	boolean	mountTrustedCA bind mounts the cluster's trusted certificate authorities, as defined in the cluster's proxy configuration, into the build. This lets processes within a build trust components signed by custom PKI certificate authorities, such as private artifact repositories and HTTPS proxies. When this field is set to true, the contents of /etc/pki/ca-trust within the build are managed by the build container, and any changes to this directory or its subdirectories (for example - within a Dockerfile RUN instruction) are not persisted in the build's output image.
nodeSelector	object (string)	nodeSelector is a selector which must be true for the build pod to fit on a node. If nil, it can be overridden by default build nodeselector values for the cluster. If set to an empty map or a map with any values, default build nodeselector values are ignored.
output	object	BuildOutput is input to a build strategy and describes the container image that the strategy should produce.
postCommit	object	A BuildPostCommitSpec holds a build post commit hook specification. The hook executes a command in a temporary container running the build output image, immediately after the last layer of the image is committed and before the image is pushed to a registry. The command is executed with the current working directory (\$PWD) set to the image's WORKDIR. The build will be marked as failed if the hook execution fails. It will

Property	Type	Description
		fail if the script or command return a non-zero exit code, or if there is any other error related to starting the temporary container. There are five different ways to configure the hook. As an example, all forms below are equivalent and will execute rake test --verbose. 1. Shell script: <pre>"postCommit": { "script": "rake test --verbose", }</pre> The above is a convenient form which is equivalent to: <pre>"postCommit": { "command": ["/bin/sh", "-ic"], "args": ["rake test --verbose"] }</pre> 2. A command as the image entrypoint: <pre>"postCommit": { "commit": ["rake", "test", "--verbose"] }</pre> Command overrides the image entrypoint in the exec form, as documented in Docker: https://docs.docker.com/engine/reference/builder/#entrypoint. 3. Pass arguments to the default entrypoint: <pre>"postCommit": { "args": ["rake", "test", "--verbose"] }</pre> This form is only useful if the image entrypoint can handle arguments. 4. Shell script with arguments: <pre>"postCommit": { "script": "rake test \$1", "args": ["--verbose"] }</pre> This form is useful if you need to pass arguments that would otherwise be hard to quote properly in the shell script. In the script, \$0 will be "/bin/sh" and \$1, \$2, etc, are the positional arguments from Args.

Property	Type	5. Command with arguments: Description
		<pre>"postCommit": { "command": ["rake", "test"], "args": ["-- verbose"] }</pre> This form is equivalent to appending the arguments to the Command slice. It is invalid to provide both Script and Command simultaneously. If none of the fields are specified, the hook is not executed.
resources	ResourceRequirements	resources computes resource requirements to execute the build.
revision	object	SourceRevision is the revision or commit information from the source for the build
runPolicy	string	RunPolicy describes how the new build created from this build configuration will be scheduled for execution. This is optional, if not specified we default to "Serial".
serviceAccount	string	serviceAccount is the name of the ServiceAccount to use to run the pod created by this build. The pod will be allowed to use secrets referenced by the ServiceAccount
source	object	BuildSource is the SCM used for the build.
strategy	object	BuildStrategy contains the details of how to perform a build.
successfulBuildsHistoryLimit	integer	successfulBuildsHistoryLimit is the number of old successful builds to retain. When a BuildConfig is created, the 5 most recent successful builds are retained unless this value is set. If removed after the BuildConfig has been created, all successful builds are retained.

Property	Type	Description
triggers	array	triggers determine how new Builds can be launched from a BuildConfig. If no triggers are defined, a new build can only occur as a result of an explicit client build creation.
triggers[]	object	BuildTriggerPolicy describes a policy for a single trigger that results in a new Build.

2.1.2. .spec.output

Description

BuildOutput is input to a build strategy and describes the container image that the strategy should produce.

Type

object

Property	Type	Description
imageLabels	array	imageLabels define a list of labels that are applied to the resulting image. If there are multiple labels with the same name then the last one in the list is used.
imageLabels[]	object	ImageLabel represents a label applied to the resulting image.
pushSecret	LocalObjectReference_v2	PushSecret is the name of a Secret that would be used for setting up the authentication for executing the Docker push to authentication enabled Docker Registry (or Docker Hub).

Property	Type	Description
to	ObjectReference	to defines an optional location to push the output of this build to. Kind must be one of 'ImageStreamTag' or 'DockerImage'. This value will be used to look up a container image repository to push to. In the case of an ImageStreamTag, the ImageStreamTag will be looked for in the namespace of the build unless Namespace is specified.

2.1.3. .spec.output.imageLabels

Description

imageLabels define a list of labels that are applied to the resulting image. If there are multiple labels with the same name then the last one in the list is used.

Type

array

2.1.4. .spec.output.imageLabels[]

Description

ImageLabel represents a label applied to the resulting image.

Type

object

Required

- **name**

Property	Type	Description
name	string	name defines the name of the label. It must have non-zero length.
value	string	value defines the literal value of the label.

2.1.5. .spec.postCommit

Description

A `BuildPostCommitSpec` holds a build post commit hook specification. The hook executes a command in a temporary container running the build output image, immediately after the last layer of the image is committed and before the image is pushed to a registry. The command is executed with the current working directory (`$PWD`) set to the image's `WORKDIR`.

The build will be marked as failed if the hook execution fails. It will fail if the script or command return a non-zero exit code, or if there is any other error related to starting the temporary container.

There are five different ways to configure the hook. As an example, all forms below are equivalent and will execute **rake test --verbose**.

1. Shell script:

```
"postCommit": {
  "script": "rake test --verbose",
}
```

The above is a convenient form which is equivalent to:

```
"postCommit": {
  "command": ["/bin/sh", "-ic"],
  "args": ["rake test --verbose"]
}
```

2. A command as the image entrypoint:

```
"postCommit": {
  "commit": ["rake", "test", "--verbose"]
}
```

Command overrides the image entrypoint in the exec form, as documented in Docker: <https://docs.docker.com/engine/reference/builder/#entrypoint>.

3. Pass arguments to the default entrypoint:

```
"postCommit": {
  "args": ["rake", "test", "--verbose"]
}
```

This form is only useful if the image entrypoint can handle arguments.

4. Shell script with arguments:

```
"postCommit": {
  "script": "rake test $1",
  "args": ["--verbose"]
}
```

This form is useful if you need to pass arguments that would otherwise be hard to quote properly in the shell script. In the script, `$0` will be `/bin/sh` and `$1`, `$2`, etc, are the positional arguments from `Args`.

5. Command with arguments:

```
"postCommit": {
  "command": ["rake", "test"],
  "args":  ["--verbose"]
}
```

This form is equivalent to appending the arguments to the Command slice.

It is invalid to provide both Script and Command simultaneously. If none of the fields are specified, the hook is not executed.

Type object

Property	Type	Description
args	array (string)	args is a list of arguments that are provided to either Command, Script or the container image's default entrypoint. The arguments are placed immediately after the command to be run.
command	array (string)	command is the command to run. It may not be specified with Script. This might be needed if the image doesn't have /bin/sh , or if you do not want to use a shell. In all other cases, using Script might be more convenient.
script	string	script is a shell script to be run with /bin/sh -ic . It may not be specified with Command. Use Script when a shell script is appropriate to execute the post build hook, for example for running unit tests with rake test . If you need control over the image entrypoint, or if the image does not have /bin/sh , use Command and/or Args. The -i flag is needed to support CentOS and RHEL images that use Software Collections (SCL), in order to have the appropriate collections enabled in the shell. E.g., in the Ruby image, this is necessary to make ruby , bundle and other binaries available in the PATH.

2.1.6. .spec.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

2.1.7. .spec.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

2.1.8. .spec.revision.git.author

Description

SourceControlUser defines the identity of a user of source control

Type
object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

2.1.9. .spec.revision.git.committer

Description

SourceControlUser defines the identity of a user of source control

Type
object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

2.1.10. .spec.source

Description

BuildSource is the SCM used for the build.

Type
object

Property	Type	Description
binary	object	BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.
configMaps	array	configMaps represents a list of configMaps and their destinations that will be used for the build.

Property	Type	Description
configMaps[]	object	ConfigMapBuildSource describes a configmap and its destination directory that will be used only at the build time. The content of the configmap referenced here will be copied into the destination directory instead of mounting.
contextDir	string	contextDir specifies the sub-directory where the source code for the application exists. This allows to have buildable sources in directory other than root of repository.
dockerfile	string	dockerfile is the raw contents of a Dockerfile which should be built. When this option is specified, the FROM may be modified based on your strategy base image and additional ENV stanzas from your strategy environment will be added after the FROM, but before the rest of your Dockerfile stanzas. The Dockerfile source type may be used with other options like git - in those cases the Git repo will have any innate Dockerfile replaced in the context dir.
git	object	GitBuildSource defines the parameters of a Git SCM
images	array	images describes a set of images to be used to provide source for the build

Property	Type	Description
images[]	object	ImageSource is used to describe build source that will be extracted from an image or used during a multi stage build. A reference of type ImageStreamTag, ImageStreamImage or DockerImage may be used. A pull secret can be specified to pull the image from an external registry or override the default service account secret if pulling from the internal registry. Image sources can either be used to extract content from an image and place it into the build context along with the repository source, or used directly during a multi-stage container image build to allow content to be copied without overwriting the contents of the repository source (see the 'paths' and 'as' fields).
secrets	array	secrets represents a list of secrets and their destinations that will be used only for the build.
secrets[]	object	SecretBuildSource describes a secret and its destination directory that will be used only at the build time. The content of the secret referenced here will be copied into the destination directory instead of mounting.
sourceSecret	LocalObjectReference_v2	sourceSecret is the name of a Secret that would be used for setting up the authentication for cloning private repository. The secret contains valid credentials for remote repository, where the data's key represent the authentication method to be used and value is the base64 encoded credentials. Supported auth methods are: ssh-privatekey.
type	string	type of build input to accept

2.1.11. .spec.source.binary

Description

BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.

Type

object

Property	Type	Description
asFile	string	asFile indicates that the provided binary input should be considered a single file within the build input. For example, specifying "webapp.war" would place the provided binary as /webapp.war for the builder. If left empty, the Docker and Source build strategies assume this file is a zip, tar, or tar.gz file and extract it as the source. The custom strategy receives this binary as standard input. This filename may not contain slashes or be '..' or '!'.

2.1.12. .spec.source.configMaps**Description**

configMaps represents a list of configMaps and their destinations that will be used for the build.

Type

array

2.1.13. .spec.source.configMaps[]**Description**

ConfigMapBuildSource describes a configmap and its destination directory that will be used only at the build time. The content of the configmap referenced here will be copied into the destination directory instead of mounting.

Type

object

Required

- **configMap**

Property	Type	Description
configMap	LocalObjectReference_v2	configMap is a reference to an existing configmap that you want to use in your build.

Property	Type	Description
destinationDir	string	destinationDir is the directory where the files from the configmap should be available for the build time. For the Source build strategy, these will be injected into a container where the assemble script runs. For the container image build strategy, these will be copied into the build directory, where the Dockerfile is located, so users can ADD or COPY them during container image build.

2.1.14. .spec.source.git

Description

GitBuildSource defines the parameters of a Git SCM

Type

object

Required

- **uri**

Property	Type	Description
httpProxy	string	httpProxy is a proxy used to reach the git repository over http
httpsProxy	string	httpsProxy is a proxy used to reach the git repository over https
noProxy	string	noProxy is the list of domains for which the proxy should not be used
ref	string	ref is the branch/tag/ref to build.
uri	string	uri points to the source that will be built. The structure of the source will depend on the type of build to run

2.1.15. .spec.source.images

Description

images describes a set of images to be used to provide source for the build

Type

array

2.1.16. .spec.source.images[]**Description**

ImageSource is used to describe build source that will be extracted from an image or used during a multi stage build. A reference of type ImageStreamTag, ImageStreamImage or DockerImage may be used. A pull secret can be specified to pull the image from an external registry or override the default service account secret if pulling from the internal registry. Image sources can either be used to extract content from an image and place it into the build context along with the repository source, or used directly during a multi-stage container image build to allow content to be copied without overwriting the contents of the repository source (see the 'paths' and 'as' fields).

Type

object

Required

- **from**

Property	Type	Description
as	array (string)	A list of image names that this source will be used in place of during a multi-stage container image build. For instance, a Dockerfile that uses "COPY --from=nginx:latest" will first check for an image source that has "nginx:latest" in this field before attempting to pull directly. If the Dockerfile does not reference an image source it is ignored. This field and paths may both be set, in which case the contents will be used twice.
from	ObjectReference	from is a reference to an ImageStreamTag, ImageStreamImage, or DockerImage to copy source from.

Property	Type	Description
paths	array	paths is a list of source and destination paths to copy from the image. This content will be copied into the build context prior to starting the build. If no paths are set, the build context will not be altered.
paths[]	object	ImageSourcePath describes a path to be copied from a source image and its destination within the build directory.
pullSecret	LocalObjectReference_v2	pullSecret is a reference to a secret to be used to pull the image from a registry. If the image is pulled from the OpenShift registry, this field does not need to be set.

2.1.17. .spec.source.images[].paths

Description

paths is a list of source and destination paths to copy from the image. This content will be copied into the build context prior to starting the build. If no paths are set, the build context will not be altered.

Type

array

2.1.18. .spec.source.images[].paths[]

Description

ImageSourcePath describes a path to be copied from a source image and its destination within the build directory.

Type

object

Required

- **sourcePath**
- **destinationDir**

Property	Type	Description

Property	Type	Description
destinationDir	string	destinationDir is the relative directory within the build directory where files copied from the image are placed.
sourcePath	string	sourcePath is the absolute path of the file or directory inside the image to copy to the build directory. If the source path ends in /. then the content of the directory will be copied, but the directory itself will not be created at the destination.

2.1.19. .spec.source.secrets

Description

secrets represents a list of secrets and their destinations that will be used only for the build.

Type

array

2.1.20. .spec.source.secrets[]

Description

SecretBuildSource describes a secret and its destination directory that will be used only at the build time. The content of the secret referenced here will be copied into the destination directory instead of mounting.

Type

object

Required

- **secret**

Property	Type	Description

Property	Type	Description
destinationDir	string	destinationDir is the directory where the files from the secret should be available for the build time. For the Source build strategy, these will be injected into a container where the assemble script runs. Later, when the script finishes, all files injected will be truncated to zero length. For the container image build strategy, these will be copied into the build directory, where the Dockerfile is located, so users can ADD or COPY them during container image build.
secret	LocalObjectReference_v2	secret is a reference to an existing secret that you want to use in your build.

2.1.21. .spec.strategy

Description

BuildStrategy contains the details of how to perform a build.

Type

object

Property	Type	Description
customStrategy	object	CustomBuildStrategy defines input parameters specific to Custom build.
dockerStrategy	object	DockerBuildStrategy defines input parameters specific to container image build.
jenkinsPipelineStrategy	object	JenkinsPipelineBuildStrategy holds parameters specific to a Jenkins Pipeline build. Deprecated: use OpenShift Pipelines
sourceStrategy	object	SourceBuildStrategy defines input parameters specific to an Source build.

Property	Type	Description
type	string	type is the kind of build strategy.

2.1.22. .spec.strategy.customStrategy

Description

CustomBuildStrategy defines input parameters specific to Custom build.

Type

object

Required

- **from**

Property	Type	Description
buildAPIVersion	string	buildAPIVersion is the requested API version for the Build object serialized and passed to the custom builder
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
exposeDockerSocket	boolean	exposeDockerSocket will allow running Docker commands (and build container images) from inside the container.
forcePull	boolean	forcePull describes if the controller should configure the build pod to always pull the images for the builder or only pull if it is not present locally
from	ObjectReference	from is reference to an DockerImage, ImageStreamTag, or ImageStreamImage from which the container image should be pulled
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries

Property	Type	Description
secrets	array	secrets is a list of additional secrets that will be included in the build pod
secrets[]	object	SecretSpec specifies a secret to be included in a build pod and its corresponding mount point

2.1.23. .spec.strategy.customStrategy.secrets

Description

secrets is a list of additional secrets that will be included in the build pod

Type

array

2.1.24. .spec.strategy.customStrategy.secrets[]

Description

SecretSpec specifies a secret to be included in a build pod and its corresponding mount point

Type

object

Required

- **secretSource**
- **mountPath**

Property	Type	Description
mountPath	string	mountPath is the path at which to mount the secret
secretSource	LocalObjectReference_v2	secretSource is a reference to the secret

2.1.25. .spec.strategy.dockerStrategy

Description

DockerBuildStrategy defines input parameters specific to container image build.

Type

object

Property	Type	Description
buildArgs	array (EnvVar)	buildArgs contains build arguments that will be resolved in the Dockerfile. See https://docs.docker.com/engine/reference/builder/#/arg for more details. NOTE: Only the 'name' and 'value' fields are supported. Any settings on the 'valueFrom' field are ignored.
dockerfilePath	string	dockerfilePath is the path of the Dockerfile that will be used to build the container image, relative to the root of the context (contextDir). Defaults to Dockerfile if unset.
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
forcePull	boolean	forcePull describes if the builder should pull the images from registry prior to building.
from	ObjectReference	from is a reference to an DockerImage, ImageStreamTag, or ImageStreamImage which overrides the FROM image in the Dockerfile for the build. If the Dockerfile uses multi-stage builds, this will replace the image in the last FROM directive of the file.

Property	Type	Description
imageOptimizationPolicy	string	imageOptimizationPolicy describes what optimizations the system can use when building images to reduce the final size or time spent building the image. The default policy is 'None' which means the final build image will be equivalent to an image created by the container image build API. The experimental policy 'SkipLayers' will avoid committing new layers in between each image step, and will fail if the Dockerfile cannot provide compatibility with the 'None' policy. An additional experimental policy 'SkipLayersAndWarn' is the same as 'SkipLayers' but simply warns if compatibility cannot be preserved.
noCache	boolean	noCache if set to true indicates that the container image build must be executed with the --no-cache=true flag
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries
volumes	array	volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

2.1.26. .spec.strategy.dockerStrategy.volumes

Description

volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info:

<https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

2.1.27. .spec.strategy.dockerStrategy.volumes[]

Description

BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

Type

object

Required

- **name**
- **source**
- **mounts**

Property	Type	Description
mounts	array	mounts represents the location of the volume in the image build container
mounts[]	object	BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.
name	string	name is a unique identifier for this BuildVolume. It must conform to the Kubernetes DNS label standard and be unique within the pod. Names that collide with those added by the build controller will result in a failed build with an error message detailing which name caused the error. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
source	object	BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

2.1.28. .spec.strategy.dockerStrategy.volumes[].mounts

Description

mounts represents the location of the volume in the image build container

Type

array

2.1.29. .spec.strategy.dockerStrategy.volumes[].mounts[]

Description

BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.

Type

object

Required

- **destinationPath**

Property	Type	Description
destinationPath	string	destinationPath is the path within the buildah runtime environment at which the volume should be mounted. The transient mount within the build image and the backing volume will both be mounted read only. Must be an absolute path, must not contain '..' or ':', and must not collide with a destination path generated by the builder process. Paths that collide with those added by the build controller will result in a failed build with an error message detailing which path caused the error.

2.1.30. .spec.strategy.dockerStrategy.volumes[].source

Description

BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

Type

object

Required

- **type**

Property	Type	Description
configMap	ConfigMapVolumeSource_v2	configMap represents a ConfigMap that should populate this volume
csi	CSIVolumeSource	csi represents ephemeral storage provided by external CSI drivers which support this capability
secret	SecretVolumeSource_v2	secret represents a Secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
type	string	type is the BuildVolumeSourceType for the volume source. Type must match the populated volume source. Valid types are: Secret, ConfigMap

2.1.31. .spec.strategy.jenkinsPipelineStrategy

Description

JenkinsPipelineBuildStrategy holds parameters specific to a Jenkins Pipeline build. Deprecated: use OpenShift Pipelines

Type

object

Property	Type	Description
env	array (EnvVar)	env contains additional environment variables you want to pass into a build pipeline.

Property	Type	Description
jenkinsfile	string	Jenkinsfile defines the optional raw contents of a Jenkinsfile which defines a Jenkins pipeline build.
jenkinsfilePath	string	JenkinsfilePath is the optional path of the Jenkinsfile that will be used to configure the pipeline relative to the root of the context (contextDir). If both JenkinsfilePath & Jenkinsfile are both not specified, this defaults to Jenkinsfile in the root of the specified contextDir.

2.1.32. .spec.strategy.sourceStrategy

Description

SourceBuildStrategy defines input parameters specific to an Source build.

Type

object

Required

- **from**

Property	Type	Description
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
forcePull	boolean	forcePull describes if the builder should pull the images from registry prior to building.
from	ObjectReference	from is reference to an DockerImage, ImageStreamTag, or ImageStreamImage from which the container image should be pulled
incremental	boolean	incremental flag forces the Source build to do incremental builds if true.

Property	Type	Description
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries
scripts	string	scripts is the location of Source scripts
volumes	array	volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

2.1.33. .spec.strategy.sourceStrategy.volumes

Description

volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info:

<https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

2.1.34. .spec.strategy.sourceStrategy.volumes[]

Description

BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

Type

object

Required

- **name**

- **source**
- **mounts**

Property	Type	Description
mounts	array	mounts represents the location of the volume in the image build container
mounts[]	object	BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.
name	string	name is a unique identifier for this BuildVolume. It must conform to the Kubernetes DNS label standard and be unique within the pod. Names that collide with those added by the build controller will result in a failed build with an error message detailing which name caused the error. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
source	object	BuildVolumeSource represents the source of a volume to mount Only one of its supported types may be specified at any given time.

2.1.35. .spec.strategy.sourceStrategy.volumes[].mounts

Description

mounts represents the location of the volume in the image build container

Type

array

2.1.36. .spec.strategy.sourceStrategy.volumes[].mounts[]

Description

BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.

Type

object

Required

- **destinationPath**

Property	Type	Description
destinationPath	string	destinationPath is the path within the buildah runtime environment at which the volume should be mounted. The transient mount within the build image and the backing volume will both be mounted read only. Must be an absolute path, must not contain '..' or '.', and must not collide with a destination path generated by the builder process. Paths that collide with those added by the build controller will result in a failed build with an error message detailing which path caused the error.

2.1.37. .spec.strategy.sourceStrategy.volumes[].source

Description

BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

Type

object

Required

- **type**

Property	Type	Description
configMap	ConfigMapVolumeSource_v2	configMap represents a ConfigMap that should populate this volume
csi	CSIVolumeSource	csi represents ephemeral storage provided by external CSI drivers which support this capability
secret	SecretVolumeSource_v2	secret represents a Secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Property	Type	Description
type	string	type is the BuildVolumeSourceType for the volume source. Type must match the populated volume source. Valid types are: Secret, ConfigMap

2.1.38. .spec.triggers

Description

triggers determine how new Builds can be launched from a BuildConfig. If no triggers are defined, a new build can only occur as a result of an explicit client build creation.

Type

array

2.1.39. .spec.triggers[]

Description

BuildTriggerPolicy describes a policy for a single trigger that results in a new Build.

Type

object

Required

- **type**

Property	Type	Description
bitbucket	object	WebHookTrigger is a trigger that gets invoked using a webhook type of post
generic	object	WebHookTrigger is a trigger that gets invoked using a webhook type of post
github	object	WebHookTrigger is a trigger that gets invoked using a webhook type of post
gitlab	object	WebHookTrigger is a trigger that gets invoked using a webhook type of post

Property	Type	Description
imageChange	object	ImageChangeTrigger allows builds to be triggered when an ImageStream changes
type	string	type is the type of build trigger. Valid values: - GitHub GitHubWebHookBuildTriggerType represents a trigger that launches builds on GitHub webhook invocations - Generic GenericWebHookBuildTriggerType represents a trigger that launches builds on generic webhook invocations - GitLab GitLabWebHookBuildTriggerType represents a trigger that launches builds on GitLab webhook invocations - Bitbucket BitbucketWebHookBuildTriggerType represents a trigger that launches builds on Bitbucket webhook invocations - ImageChange ImageChangeBuildTriggerType represents a trigger that launches builds on availability of a new version of an image - ConfigChange ConfigChangeBuildTriggerType will trigger a build on an initial build config creation WARNING: In the future the behavior will change to trigger a build on any config change

2.1.40. .spec.triggers[].bitbucket

Description

WebHookTrigger is a trigger that gets invoked using a webhook type of post

Type

object

Property	Type	Description
allowEnv	boolean	allowEnv determines whether the webhook can set environment variables; can only be set to true for GenericWebHook.
secret	string	secret used to validate requests. Deprecated: use SecretReference instead.
secretReference	object	SecretLocalReference contains information that points to the local secret being used

2.1.41. `.spec.triggers[].bitbucket.secretReference`

Description

SecretLocalReference contains information that points to the local secret being used

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the resource in the same namespace being referenced

2.1.42. `.spec.triggers[].generic`

Description

WebHookTrigger is a trigger that gets invoked using a webhook type of post

Type

object

Property	Type	Description

Property	Type	Description
allowEnv	boolean	allowEnv determines whether the webhook can set environment variables; can only be set to true for GenericWebHook.
secret	string	secret used to validate requests. Deprecated: use SecretReference instead.
secretReference	object	SecretLocalReference contains information that points to the local secret being used

2.1.43. .spec.triggers[].generic.secretReference

Description

SecretLocalReference contains information that points to the local secret being used

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the resource in the same namespace being referenced

2.1.44. .spec.triggers[].github

Description

WebHookTrigger is a trigger that gets invoked using a webhook type of post

Type

object

Property	Type	Description
allowEnv	boolean	allowEnv determines whether the webhook can set environment variables; can only be set to true for GenericWebHook.

Property	Type	Description
secret	string	secret used to validate requests. Deprecated: use SecretReference instead.
secretReference	object	SecretLocalReference contains information that points to the local secret being used

2.1.45. .spec.triggers[].github.secretReference

Description

SecretLocalReference contains information that points to the local secret being used

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the resource in the same namespace being referenced

2.1.46. .spec.triggers[].gitlab

Description

WebHookTrigger is a trigger that gets invoked using a webhook type of post

Type

object

Property	Type	Description
allowEnv	boolean	allowEnv determines whether the webhook can set environment variables; can only be set to true for GenericWebHook.
secret	string	secret used to validate requests. Deprecated: use SecretReference instead.

Property	Type	Description
secretReference	object	SecretLocalReference contains information that points to the local secret being used

2.1.47. .spec.triggers[].gitlab.secretReference

Description

SecretLocalReference contains information that points to the local secret being used

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the resource in the same namespace being referenced

2.1.48. .spec.triggers[].imageChange

Description

ImageChangeTrigger allows builds to be triggered when an ImageStream changes

Type

object

Property	Type	Description
from	ObjectReference	from is a reference to an ImageStreamTag that will trigger a build when updated It is optional. If no From is specified, the From image from the build strategy will be used. Only one ImageChangeTrigger with an empty From reference is allowed in a build configuration.

Property	Type	Description
lastTriggeredImageID	string	lastTriggeredImageID is used internally by the ImageChangeController to save last used image ID for build This field is deprecated and will be removed in a future release. Deprecated
paused	boolean	paused is true if this trigger is temporarily disabled. Optional.

2.1.49. .status

Description

BuildConfigStatus contains current state of the build config object.

Type

object

Required

- **lastVersion**

Property	Type	Description
imageChangeTriggers	array	ImageChangeTriggers captures the runtime state of any ImageChangeTrigger specified in the BuildConfigSpec, including the value reconciled by the OpenShift APIServer for the lastTriggeredImageID. There is a single entry in this array for each image change trigger in spec. Each trigger status references the ImageStreamTag that acts as the source of the trigger.
imageChangeTriggers[]	object	ImageChangeTriggerStatus tracks the latest resolved status of the associated ImageChangeTrigger policy specified in the BuildConfigSpec.Triggers struct.

Property	Type	Description
lastVersion	integer	lastVersion is used to inform about number of last triggered build.

2.1.50. .status.imageChangeTriggers

Description

ImageChangeTriggers captures the runtime state of any ImageChangeTrigger specified in the BuildConfigSpec, including the value reconciled by the OpenShift APIServer for the lastTriggeredImageID. There is a single entry in this array for each image change trigger in spec. Each trigger status references the ImageStreamTag that acts as the source of the trigger.

Type

array

2.1.51. .status.imageChangeTriggers[]

Description

ImageChangeTriggerStatus tracks the latest resolved status of the associated ImageChangeTrigger policy specified in the BuildConfigSpec.Triggers struct.

Type

object

Property	Type	Description
from	object	ImageStreamTagReference references the ImageStreamTag in an image change trigger by namespace and name.
lastTriggerTime	Time	lastTriggerTime is the last time this particular ImageStreamTag triggered a Build to start. This field is only updated when this trigger specifically started a Build.
lastTriggeredImageID	string	lastTriggeredImageID represents the sha/id of the ImageStreamTag when a Build for this BuildConfig was started. The lastTriggeredImageID is updated each time a Build for this BuildConfig is started, even if this ImageStreamTag is not the reason the Build is started.

2.1.52. `.status.imageChangeTriggers[].from`

Description

ImageStreamTagReference references the ImageStreamTag in an image change trigger by namespace and name.

Type

object

Property	Type	Description
name	string	name is the name of the ImageStreamTag for an ImageChangeTrigger
namespace	string	namespace is the namespace where the ImageStreamTag for an ImageChangeTrigger is located

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/build.openshift.io/v1/buildconfigs**
 - **GET**: list or watch objects of kind BuildConfig
- **/apis/build.openshift.io/v1/watch/buildconfigs**
 - **GET**: watch individual changes to a list of BuildConfig. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs**
 - **DELETE**: delete collection of BuildConfig
 - **GET**: list or watch objects of kind BuildConfig
 - **POST**: create a BuildConfig
- **/apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs**
 - **GET**: watch individual changes to a list of BuildConfig. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}**
 - **DELETE**: delete a BuildConfig
 - **GET**: read the specified BuildConfig
 - **PATCH**: partially update the specified BuildConfig
 - **PUT**: replace the specified BuildConfig

- **/apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs/{name}**
 - **GET**: watch changes to an object of kind BuildConfig. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

2.2.1. /apis/build.openshift.io/v1/buildconfigs

HTTP method

GET

Description

list or watch objects of kind BuildConfig

Table 2.1. HTTP responses

HTTP code	Reponse body
200 - OK	BuildConfigList schema
401 - Unauthorized	Empty

2.2.2. /apis/build.openshift.io/v1/watch/buildconfigs

HTTP method

GET

Description

watch individual changes to a list of BuildConfig. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

2.2.3. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs

HTTP method

DELETE

Description

delete collection of BuildConfig

Table 2.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v4 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind BuildConfig

Table 2.5. HTTP responses

HTTP code	Reponse body
200 - OK	BuildConfigList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a BuildConfig

Table 2.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.7. Body parameters

Parameter	Type	Description
body	BuildConfig schema	

Table 2.8. HTTP responses

HTTP code	Response body
200 - OK	BuildConfig schema
201 - Created	BuildConfig schema
202 - Accepted	BuildConfig schema
401 - Unauthorized	Empty

2.2.4. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs

HTTP method

GET

Description

watch individual changes to a list of BuildConfig. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

2.2.5. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}

Table 2.10. Global path parameters

Parameter	Type	Description
name	string	name of the BuildConfig

HTTP method

DELETE

Description

delete a BuildConfig

Table 2.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v4 schema
202 - Accepted	Status_v4 schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified BuildConfig

Table 2.13. HTTP responses

HTTP code	Reponse body
200 - OK	BuildConfig schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified BuildConfig

Table 2.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.15. HTTP responses

HTTP code	Reponse body
200 - OK	BuildConfig schema
201 - Created	BuildConfig schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified BuildConfig

Table 2.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.17. Body parameters

Parameter	Type	Description
body	BuildConfig schema	

Table 2.18. HTTP responses

HTTP code	Reponse body
200 - OK	BuildConfig schema
201 - Created	BuildConfig schema
401 - Unauthorized	Empty

2.2.6. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/buildconfigs/{name}

Table 2.19. Global path parameters

Parameter	Type	Description
name	string	name of the BuildConfig

HTTP method

GET

Description

watch changes to an object of kind BuildConfig. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 2.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 3. BUILD [BUILD.OPENSIFT.IO/V1]

Description

Build encapsulates the inputs needed to produce a new deployable image, as well as the status of the execution and a reference to the Pod which executed the build.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

3.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	BuildSpec has the information to represent a build and also additional information about a build

Property	Type	Description
status	object	BuildStatus contains the status of a build

3.1.1. .spec

Description

BuildSpec has the information to represent a build and also additional information about a build

Type

object

Required

- **strategy**

Property	Type	Description
completionDeadlineSeconds	integer	completionDeadlineSeconds is an optional duration in seconds, counted from the time when a build pod gets scheduled in the system, that the build may be active on a node before the system actively tries to terminate the build; value must be positive integer
mountTrustedCA	boolean	mountTrustedCA bind mounts the cluster's trusted certificate authorities, as defined in the cluster's proxy configuration, into the build. This lets processes within a build trust components signed by custom PKI certificate authorities, such as private artifact repositories and HTTPS proxies. When this field is set to true, the contents of /etc/pki/ca-trust within the build are managed by the build container, and any changes to this directory or its subdirectories (for example - within a Dockerfile RUN instruction) are not persisted in the build's output image.

Property	Type	Description
nodeSelector	object (string)	nodeSelector is a selector which must be true for the build pod to fit on a node. If nil, it can be overridden by default build nodeSelector values for the cluster. If set to an empty map or a map with any values, default build nodeSelector values are ignored.
output	object	BuildOutput is input to a build strategy and describes the container image that the strategy should produce.
postCommit	object	A BuildPostCommitSpec holds a build post commit hook specification. The hook executes a command in a temporary container running the build output image, immediately after the last layer of the image is committed and before the image is pushed to a registry. The command is executed with the current working directory (\$PWD) set to the image's WORKDIR. The build will be marked as failed if the hook execution fails. It will fail if the script or command return a non-zero exit code, or if there is any other error related to starting the temporary container. There are five different ways to configure the hook. As an example, all forms below are equivalent and will execute rake test --verbose. 1. Shell script: <pre>"postCommit": { "script": "rake test --verbose", }</pre> The above is a convenient form which is equivalent to:

Property	Type	Description
		<code>"postCommit": { "command": ["/bin/sh", "-ic"], "args": ["rake test --verbose"] }</code> 2. A command as the image entrypoint: <code>"postCommit": { "commit": ["rake", "test", "--verbose"] }</code> Command overrides the image entrypoint in the exec form, as documented in Docker: https://docs.docker.com/engine/reference/builder/#entrypoint. 3. Pass arguments to the default entrypoint: <code>"postCommit": { "args": ["rake", "test", "--verbose"] }</code> This form is only useful if the image entrypoint can handle arguments. 4. Shell script with arguments: <code>"postCommit": { "script": "rake test \$1", "args": ["--verbose"] }</code> This form is useful if you need to pass arguments that would otherwise be hard to quote properly in the shell script. In the script, \$0 will be "/bin/sh" and \$1, \$2, etc, are the positional arguments from Args. 5. Command with arguments: <code>"postCommit": { "command": ["rake", "test"], "args": ["--verbose"] }</code> This form is equivalent to appending the arguments to the Command slice. It is invalid to provide both Script and Command simultaneously. If none of the fields are specified, the hook is not executed.
resources	ResourceRequirements	resources computes resource requirements to execute the build.

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
serviceAccount	string	serviceAccount is the name of the ServiceAccount to use to run the pod created by this build. The pod will be allowed to use secrets referenced by the ServiceAccount
source	object	BuildSource is the SCM used for the build.
strategy	object	BuildStrategy contains the details of how to perform a build.
triggeredBy	array	triggeredBy describes which triggers started the most recent update to the build configuration and contains information about those triggers.
triggeredBy[]	object	BuildTriggerCause holds information about a triggered build. It is used for displaying build trigger data for each build and build configuration in oc describe. It is also used to describe which triggers led to the most recent update in the build configuration.

3.1.2. .spec.output

Description

BuildOutput is input to a build strategy and describes the container image that the strategy should produce.

Type

object

Property	Type	Description
imageLabels	array	imageLabels define a list of labels that are applied to the resulting image. If there are multiple labels with the same name then the last one in the list is used.

Property	Type	Description
imageLabels[]	object	ImageLabel represents a label applied to the resulting image.
pushSecret	LocalObjectReference_v2	PushSecret is the name of a Secret that would be used for setting up the authentication for executing the Docker push to authentication enabled Docker Registry (or Docker Hub).
to	ObjectReference	to defines an optional location to push the output of this build to. Kind must be one of 'ImageStreamTag' or 'DockerImage'. This value will be used to look up a container image repository to push to. In the case of an ImageStreamTag, the ImageStreamTag will be looked for in the namespace of the build unless Namespace is specified.

3.1.3. .spec.output.imageLabels

Description

imageLabels define a list of labels that are applied to the resulting image. If there are multiple labels with the same name then the last one in the list is used.

Type

array

3.1.4. .spec.output.imageLabels[]

Description

ImageLabel represents a label applied to the resulting image.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	name defines the name of the label. It must have non-zero length.
value	string	value defines the literal value of the label.

3.1.5. .spec.postCommit

Description

A BuildPostCommitSpec holds a build post commit hook specification. The hook executes a command in a temporary container running the build output image, immediately after the last layer of the image is committed and before the image is pushed to a registry. The command is executed with the current working directory (\$PWD) set to the image's WORKDIR.

The build will be marked as failed if the hook execution fails. It will fail if the script or command return a non-zero exit code, or if there is any other error related to starting the temporary container.

There are five different ways to configure the hook. As an example, all forms below are equivalent and will execute **rake test --verbose**.

1. Shell script:

```
"postCommit": {
  "script": "rake test --verbose",
}
```

The above is a convenient form which is equivalent to:

```
"postCommit": {
  "command": ["/bin/sh", "-ic"],
  "args": ["rake test --verbose"]
}
```

2. A command as the image entrypoint:

```
"postCommit": {
  "commit": ["rake", "test", "--verbose"]
}
```

Command overrides the image entrypoint in the exec form, as documented in Docker: <https://docs.docker.com/engine/reference/builder/#entrypoint>.

3. Pass arguments to the default entrypoint:

```
"postCommit": {
  "args": ["rake", "test", "--verbose"]
}
```

This form is only useful if the image entrypoint can handle arguments.

4. Shell script with arguments:

```
"postCommit": {
  "script": "rake test $1",
  "args": ["--verbose"]
}
```

This form is useful if you need to pass arguments that would otherwise be hard to quote properly in the shell script. In the script, \$0 will be `/bin/sh` and \$1, \$2, etc, are the positional arguments from Args.

5. Command with arguments:

```
"postCommit": {
  "command": ["rake", "test"],
  "args": ["--verbose"]
}
```

This form is equivalent to appending the arguments to the Command slice.

It is invalid to provide both Script and Command simultaneously. If none of the fields are specified, the hook is not executed.

Type object

Property	Type	Description
args	array (string)	args is a list of arguments that are provided to either Command, Script or the container image's default entrypoint. The arguments are placed immediately after the command to be run.
command	array (string)	command is the command to run. It may not be specified with Script. This might be needed if the image doesn't have /bin/sh , or if you do not want to use a shell. In all other cases, using Script might be more convenient.

Property	Type	Description
script	string	script is a shell script to be run with /bin/sh -ic . It may not be specified with Command. Use Script when a shell script is appropriate to execute the post build hook, for example for running unit tests with rake test . If you need control over the image entrypoint, or if the image does not have /bin/sh , use Command and/or Args. The -i flag is needed to support CentOS and RHEL images that use Software Collections (SCL), in order to have the appropriate collections enabled in the shell. E.g., in the Ruby image, this is necessary to make ruby, bundle and other binaries available in the PATH.

3.1.6. .spec.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

3.1.7. .spec.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

3.1.8. .spec.revision.git.author**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.9. .spec.revision.git.committer**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.10. .spec.source

Description

BuildSource is the SCM used for the build.

Type

object

Property	Type	Description
binary	object	BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.
configMaps	array	configMaps represents a list of configMaps and their destinations that will be used for the build.
configMaps[]	object	ConfigMapBuildSource describes a configmap and its destination directory that will be used only at the build time. The content of the configmap referenced here will be copied into the destination directory instead of mounting.
contextDir	string	contextDir specifies the sub-directory where the source code for the application exists. This allows to have buildable sources in directory other than root of repository.
dockerfile	string	dockerfile is the raw contents of a Dockerfile which should be built. When this option is specified, the FROM may be modified based on your strategy base image and additional ENV stanzas from your strategy environment will be added after the FROM, but before the rest of your Dockerfile stanzas. The Dockerfile source type may be used with other options like git - in those cases the Git repo will have any innate Dockerfile replaced in the context dir.
git	object	GitBuildSource defines the parameters of a Git SCM

Property	Type	Description
images	array	images describes a set of images to be used to provide source for the build
images[]	object	ImageSource is used to describe build source that will be extracted from an image or used during a multi stage build. A reference of type ImageStreamTag, ImageStreamImage or DockerImage may be used. A pull secret can be specified to pull the image from an external registry or override the default service account secret if pulling from the internal registry. Image sources can either be used to extract content from an image and place it into the build context along with the repository source, or used directly during a multi-stage container image build to allow content to be copied without overwriting the contents of the repository source (see the 'paths' and 'as' fields).
secrets	array	secrets represents a list of secrets and their destinations that will be used only for the build.
secrets[]	object	SecretBuildSource describes a secret and its destination directory that will be used only at the build time. The content of the secret referenced here will be copied into the destination directory instead of mounting.
sourceSecret	LocalObjectReference_v2	sourceSecret is the name of a Secret that would be used for setting up the authentication for cloning private repository. The secret contains valid credentials for remote repository, where the data's key represent the authentication method to be used and value is the base64 encoded credentials. Supported auth methods are: ssh-privatekey.

Property	Type	Description
type	string	type of build input to accept

3.1.11. .spec.source.binary

Description

BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.

Type

object

Property	Type	Description
asFile	string	asFile indicates that the provided binary input should be considered a single file within the build input. For example, specifying "webapp.war" would place the provided binary as /webapp.war for the builder. If left empty, the Docker and Source build strategies assume this file is a zip, tar, or tar.gz file and extract it as the source. The custom strategy receives this binary as standard input. This filename may not contain slashes or be '..' or '..'.

3.1.12. .spec.source.configMaps

Description

configMaps represents a list of configMaps and their destinations that will be used for the build.

Type

array

3.1.13. .spec.source.configMaps[]

Description

ConfigMapBuildSource describes a configmap and its destination directory that will be used only at the build time. The content of the configmap referenced here will be copied into the destination directory instead of mounting.

Type

object

Required

- **configMap**

Property	Type	Description
configMap	LocalObjectReference_v2	configMap is a reference to an existing configmap that you want to use in your build.
destinationDir	string	destinationDir is the directory where the files from the configmap should be available for the build time. For the Source build strategy, these will be injected into a container where the assemble script runs. For the container image build strategy, these will be copied into the build directory, where the Dockerfile is located, so users can ADD or COPY them during container image build.

3.1.14. .spec.source.git

Description

GitBuildSource defines the parameters of a Git SCM

Type

object

Required

- **uri**

Property	Type	Description
httpProxy	string	httpProxy is a proxy used to reach the git repository over http
httpsProxy	string	httpsProxy is a proxy used to reach the git repository over https
noProxy	string	noProxy is the list of domains for which the proxy should not be used
ref	string	ref is the branch/tag/ref to build.
uri	string	uri points to the source that will be built. The structure of the source will depend on the type of build to run

3.1.15. .spec.source.images

Description

images describes a set of images to be used to provide source for the build

Type

array

3.1.16. .spec.source.images[]

Description

ImageSource is used to describe build source that will be extracted from an image or used during a multi stage build. A reference of type ImageStreamTag, ImageStreamImage or DockerImage may be used. A pull secret can be specified to pull the image from an external registry or override the default service account secret if pulling from the internal registry. Image sources can either be used to extract content from an image and place it into the build context along with the repository source, or used directly during a multi-stage container image build to allow content to be copied without overwriting the contents of the repository source (see the 'paths' and 'as' fields).

Type

object

Required

- from

Property	Type	Description
as	array (string)	A list of image names that this source will be used in place of during a multi-stage container image build. For instance, a Dockerfile that uses "COPY --from=nginx:latest" will first check for an image source that has "nginx:latest" in this field before attempting to pull directly. If the Dockerfile does not reference an image source it is ignored. This field and paths may both be set, in which case the contents will be used twice.
from	ObjectReference	from is a reference to an ImageStreamTag, ImageStreamImage, or DockerImage to copy source from.

Property	Type	Description
paths	array	paths is a list of source and destination paths to copy from the image. This content will be copied into the build context prior to starting the build. If no paths are set, the build context will not be altered.
paths[]	object	ImageSourcePath describes a path to be copied from a source image and its destination within the build directory.
pullSecret	LocalObjectReference_v2	pullSecret is a reference to a secret to be used to pull the image from a registry. If the image is pulled from the OpenShift registry, this field does not need to be set.

3.1.17. .spec.source.images[].paths

Description

paths is a list of source and destination paths to copy from the image. This content will be copied into the build context prior to starting the build. If no paths are set, the build context will not be altered.

Type

array

3.1.18. .spec.source.images[].paths[]

Description

ImageSourcePath describes a path to be copied from a source image and its destination within the build directory.

Type

object

Required

- **sourcePath**
- **destinationDir**

Property	Type	Description
destinationDir	string	destinationDir is the relative directory within the build directory where files copied from the image are placed.
sourcePath	string	sourcePath is the absolute path of the file or directory inside the image to copy to the build directory. If the source path ends in /. then the content of the directory will be copied, but the directory itself will not be created at the destination.

3.1.19. .spec.source.secrets

Description

secrets represents a list of secrets and their destinations that will be used only for the build.

Type

array

3.1.20. .spec.source.secrets[]

Description

SecretBuildSource describes a secret and its destination directory that will be used only at the build time. The content of the secret referenced here will be copied into the destination directory instead of mounting.

Type

object

Required

- **secret**

Property	Type	Description

Property	Type	Description
destinationDir	string	destinationDir is the directory where the files from the secret should be available for the build time. For the Source build strategy, these will be injected into a container where the assemble script runs. Later, when the script finishes, all files injected will be truncated to zero length. For the container image build strategy, these will be copied into the build directory, where the Dockerfile is located, so users can ADD or COPY them during container image build.
secret	LocalObjectReference_v2	secret is a reference to an existing secret that you want to use in your build.

3.1.21. .spec.strategy

Description

BuildStrategy contains the details of how to perform a build.

Type

object

Property	Type	Description
customStrategy	object	CustomBuildStrategy defines input parameters specific to Custom build.
dockerStrategy	object	DockerBuildStrategy defines input parameters specific to container image build.
jenkinsPipelineStrategy	object	JenkinsPipelineBuildStrategy holds parameters specific to a Jenkins Pipeline build. Deprecated: use OpenShift Pipelines
sourceStrategy	object	SourceBuildStrategy defines input parameters specific to an Source build.

Property	Type	Description
type	string	type is the kind of build strategy.

3.1.22. .spec.strategy.customStrategy

Description

CustomBuildStrategy defines input parameters specific to Custom build.

Type

object

Required

- **from**

Property	Type	Description
buildAPIVersion	string	buildAPIVersion is the requested API version for the Build object serialized and passed to the custom builder
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
exposeDockerSocket	boolean	exposeDockerSocket will allow running Docker commands (and build container images) from inside the container.
forcePull	boolean	forcePull describes if the controller should configure the build pod to always pull the images for the builder or only pull if it is not present locally
from	ObjectReference	from is reference to an DockerImage, ImageStreamTag, or ImageStreamImage from which the container image should be pulled
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries

Property	Type	Description
secrets	array	secrets is a list of additional secrets that will be included in the build pod
secrets[]	object	SecretSpec specifies a secret to be included in a build pod and its corresponding mount point

3.1.23. .spec.strategy.customStrategy.secrets

Description

secrets is a list of additional secrets that will be included in the build pod

Type

array

3.1.24. .spec.strategy.customStrategy.secrets[]

Description

SecretSpec specifies a secret to be included in a build pod and its corresponding mount point

Type

object

Required

- **secretSource**
- **mountPath**

Property	Type	Description
mountPath	string	mountPath is the path at which to mount the secret
secretSource	LocalObjectReference_v2	secretSource is a reference to the secret

3.1.25. .spec.strategy.dockerStrategy

Description

DockerBuildStrategy defines input parameters specific to container image build.

Type

object

Property	Type	Description
buildArgs	array (EnvVar)	buildArgs contains build arguments that will be resolved in the Dockerfile. See https://docs.docker.com/engine/reference/builder/#/arg for more details. NOTE: Only the 'name' and 'value' fields are supported. Any settings on the 'valueFrom' field are ignored.
dockerfilePath	string	dockerfilePath is the path of the Dockerfile that will be used to build the container image, relative to the root of the context (contextDir). Defaults to Dockerfile if unset.
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
forcePull	boolean	forcePull describes if the builder should pull the images from registry prior to building.
from	ObjectReference	from is a reference to an DockerImage, ImageStreamTag, or ImageStreamImage which overrides the FROM image in the Dockerfile for the build. If the Dockerfile uses multi-stage builds, this will replace the image in the last FROM directive of the file.

Property	Type	Description
imageOptimizationPolicy	string	imageOptimizationPolicy describes what optimizations the system can use when building images to reduce the final size or time spent building the image. The default policy is 'None' which means the final build image will be equivalent to an image created by the container image build API. The experimental policy 'SkipLayers' will avoid committing new layers in between each image step, and will fail if the Dockerfile cannot provide compatibility with the 'None' policy. An additional experimental policy 'SkipLayersAndWarn' is the same as 'SkipLayers' but simply warns if compatibility cannot be preserved.
noCache	boolean	noCache if set to true indicates that the container image build must be executed with the --no-cache=true flag
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries
volumes	array	volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

3.1.26. .spec.strategy.dockerStrategy.volumes

Description

volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info:

<https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

3.1.27. .spec.strategy.dockerStrategy.volumes[]

Description

BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

Type

object

Required

- **name**
- **source**
- **mounts**

Property	Type	Description
mounts	array	mounts represents the location of the volume in the image build container
mounts[]	object	BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.
name	string	name is a unique identifier for this BuildVolume. It must conform to the Kubernetes DNS label standard and be unique within the pod. Names that collide with those added by the build controller will result in a failed build with an error message detailing which name caused the error. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

Property	Type	Description
source	object	BuildVolumeSource represents the source of a volume to mount Only one of its supported types may be specified at any given time.

3.1.28. .spec.strategy.dockerStrategy.volumes[].mounts

Description

mounts represents the location of the volume in the image build container

Type

array

3.1.29. .spec.strategy.dockerStrategy.volumes[].mounts[]

Description

BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.

Type

object

Required

- **destinationPath**

Property	Type	Description
destinationPath	string	destinationPath is the path within the buildah runtime environment at which the volume should be mounted. The transient mount within the build image and the backing volume will both be mounted read only. Must be an absolute path, must not contain '..' or ':', and must not collide with a destination path generated by the builder process Paths that collide with those added by the build controller will result in a failed build with an error message detailing which path caused the error.

3.1.30. .spec.strategy.dockerStrategy.volumes[].source

Description

BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

Type

object

Required

- **type**

Property	Type	Description
configMap	ConfigMapVolumeSource_v2	configMap represents a ConfigMap that should populate this volume
csi	CSIVolumeSource	csi represents ephemeral storage provided by external CSI drivers which support this capability
secret	SecretVolumeSource_v2	secret represents a Secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret
type	string	type is the BuildVolumeSourceType for the volume source. Type must match the populated volume source. Valid types are: Secret, ConfigMap

3.1.31. .spec.strategy.jenkinsPipelineStrategy

Description

JenkinsPipelineBuildStrategy holds parameters specific to a Jenkins Pipeline build. Deprecated: use OpenShift Pipelines

Type

object

Property	Type	Description
env	array (EnvVar)	env contains additional environment variables you want to pass into a build pipeline.

Property	Type	Description
jenkinsfile	string	Jenkinsfile defines the optional raw contents of a Jenkinsfile which defines a Jenkins pipeline build.
jenkinsfilePath	string	JenkinsfilePath is the optional path of the Jenkinsfile that will be used to configure the pipeline relative to the root of the context (contextDir). If both JenkinsfilePath & Jenkinsfile are both not specified, this defaults to Jenkinsfile in the root of the specified contextDir.

3.1.32. .spec.strategy.sourceStrategy

Description

SourceBuildStrategy defines input parameters specific to an Source build.

Type

object

Required

- **from**

Property	Type	Description
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
forcePull	boolean	forcePull describes if the builder should pull the images from registry prior to building.
from	ObjectReference	from is reference to an DockerImage, ImageStreamTag, or ImageStreamImage from which the container image should be pulled
incremental	boolean	incremental flag forces the Source build to do incremental builds if true.

Property	Type	Description
pullSecret	LocalObjectReference_v2	pullSecret is the name of a Secret that would be used for setting up the authentication for pulling the container images from the private Docker registries
scripts	string	scripts is the location of Source scripts
volumes	array	volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

3.1.33. .spec.strategy.sourceStrategy.volumes

Description

volumes is a list of input volumes that can be mounted into the builds runtime environment. Only a subset of Kubernetes Volume sources are supported by builds. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

3.1.34. .spec.strategy.sourceStrategy.volumes[]

Description

BuildVolume describes a volume that is made available to build pods, such that it can be mounted into buildah's runtime environment. Only a subset of Kubernetes Volume sources are supported.

Type

object

Required

- **name**

- **source**
- **mounts**

Property	Type	Description
mounts	array	mounts represents the location of the volume in the image build container
mounts[]	object	BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.
name	string	name is a unique identifier for this BuildVolume. It must conform to the Kubernetes DNS label standard and be unique within the pod. Names that collide with those added by the build controller will result in a failed build with an error message detailing which name caused the error. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
source	object	BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

3.1.35. .spec.strategy.sourceStrategy.volumes[].mounts

Description

mounts represents the location of the volume in the image build container

Type

array

3.1.36. .spec.strategy.sourceStrategy.volumes[].mounts[]

Description

BuildVolumeMount describes the mounting of a Volume within buildah's runtime environment.

Type

object

Required

- **destinationPath**

Property	Type	Description
destinationPath	string	destinationPath is the path within the buildah runtime environment at which the volume should be mounted. The transient mount within the build image and the backing volume will both be mounted read only. Must be an absolute path, must not contain '..' or '.', and must not collide with a destination path generated by the builder process. Paths that collide with those added by the build controller will result in a failed build with an error message detailing which path caused the error.

3.1.37. .spec.strategy.sourceStrategy.volumes[].source

Description

BuildVolumeSource represents the source of a volume to mount. Only one of its supported types may be specified at any given time.

Type

object

Required

- **type**

Property	Type	Description
configMap	ConfigMapVolumeSource_v2	configMap represents a ConfigMap that should populate this volume
csi	CSIVolumeSource	csi represents ephemeral storage provided by external CSI drivers which support this capability
secret	SecretVolumeSource_v2	secret represents a Secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Property	Type	Description
type	string	type is the BuildVolumeSourceType for the volume source. Type must match the populated volume source. Valid types are: Secret, ConfigMap

3.1.38. .spec.triggeredBy

Description

triggeredBy describes which triggers started the most recent update to the build configuration and contains information about those triggers.

Type

array

3.1.39. .spec.triggeredBy[]

Description

BuildTriggerCause holds information about a triggered build. It is used for displaying build trigger data for each build and build configuration in oc describe. It is also used to describe which triggers led to the most recent update in the build configuration.

Type

object

Property	Type	Description
bitbucketWebHook	object	BitbucketWebHookCause has information about a Bitbucket webhook that triggered a build.
genericWebHook	object	GenericWebHookCause holds information about a generic WebHook that triggered a build.
githubWebHook	object	GitHubWebHookCause has information about a GitHub webhook that triggered a build.
gitlabWebHook	object	GitLabWebHookCause has information about a GitLab webhook that triggered a build.
imageChangeBuild	object	ImageChangeCause contains information about the image that triggered a build

Property	Type	Description
message	string	message is used to store a human readable message for why the build was triggered. E.g.: "Manually triggered by user", "Configuration change", etc.

3.1.40. .spec.triggeredBy[].bitbucketWebHook

Description

BitbucketWebHookCause has information about a Bitbucket webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	Secret is the obfuscated webhook secret that triggered a build.

3.1.41. .spec.triggeredBy[].bitbucketWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

3.1.42. .spec.triggeredBy[].bitbucketWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

3.1.43. .spec.triggeredBy[].bitbucketWebHook.revision.git.author**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.44. .spec.triggeredBy[].bitbucketWebHook.revision.git.committer**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user

Property	Type	Description
name	string	name of the source control user

3.1.45. .spec.triggeredBy[].genericWebHook

Description

GenericWebHookCause holds information about a generic WebHook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	secret is the obfuscated webhook secret that triggered a build.

3.1.46. .spec.triggeredBy[].genericWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

3.1.47. .spec.triggeredBy[].genericWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

3.1.48. .spec.triggeredBy[].genericWebHook.revision.git.author**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.49. .spec.triggeredBy[].genericWebHook.revision.git.committer**Description**

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.50. .spec.triggeredBy[].githubWebHook

Description

GitHubWebHookCause has information about a GitHub webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	secret is the obfuscated webhook secret that triggered a build.

3.1.51. .spec.triggeredBy[].githubWebHook.revision**Description**

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

3.1.52. .spec.triggeredBy[].githubWebHook.revision.git**Description**

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

3.1.53. .spec.triggeredBy[].githubWebHook.revision.git.author

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.54. .spec.triggeredBy[].githubWebHook.revision.git.committer

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.55. .spec.triggeredBy[].gitlabWebHook

Description

GitLabWebHookCause has information about a GitLab webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	Secret is the obfuscated webhook secret that triggered a build.

3.1.56. .spec.triggeredBy[].gitlabWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

3.1.57. .spec.triggeredBy[].gitlabWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control

Property	Type	Description
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

3.1.58. .spec.triggeredBy[].gitlabWebHook.revision.git.author

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.59. .spec.triggeredBy[].gitlabWebHook.revision.git.committer

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

3.1.60. .spec.triggeredBy[].imageChangeBuild

Description

ImageChangeCause contains information about the image that triggered a build

Type

object

Property	Type	Description
fromRef	ObjectReference	fromRef contains detailed information about an image that triggered a build.
imageID	string	imageID is the ID of the image that triggered a new build.

3.1.61. .status

Description

BuildStatus contains the status of a build

Type

object

Required

- **phase**

Property	Type	Description
cancelled	boolean	cancelled describes if a cancel event was triggered for the build.
completionTimestamp	Time	completionTimestamp is a timestamp representing the server time when this Build was finished, whether that build failed or succeeded. It reflects the time at which the Pod running the Build terminated. It is represented in RFC3339 form and is in UTC.
conditions	array	Conditions represents the latest available observations of a build's current state.
conditions[]	object	BuildCondition describes the state of a build at a certain point.
config	ObjectReference	config is an ObjectReference to the BuildConfig this Build is based on.
duration	integer	duration contains time.Duration object describing build time.

Property	Type	Description
logSnippet	string	logSnippet is the last few lines of the build log. This value is only set for builds that failed.
message	string	message is a human-readable message indicating details about why the build has this status.
output	object	BuildStatusOutput contains the status of the built image.
outputDockerImageReference	string	outputDockerImageReference contains a reference to the container image that will be built by this build. Its value is computed from Build.Spec.Output.To, and should include the registry address, so that it can be used to push and pull the image.
phase	string	phase is the point in the build lifecycle. Possible values are "New", "Pending", "Running", "Complete", "Failed", "Error", and "Cancelled".
reason	string	reason is a brief CamelCase string that describes any failure and is meant for machine parsing and tidy display in the CLI.
stages	array	stages contains details about each stage that occurs during the build including start time, duration (in milliseconds), and the steps that occurred within each stage.
stages[]	object	StageInfo contains details about a build stage.
startTimestamp	Time	startTimestamp is a timestamp representing the server time when this Build started running in a Pod. It is represented in RFC3339 form and is in UTC.

3.1.62. .status.conditions

Description

Conditions represents the latest available observations of a build's current state.

Type

array

3.1.63. .status.conditions[]

Description

BuildCondition describes the state of a build at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	The last time the condition transitioned from one status to another.
lastUpdateTime	Time	The last time this condition was updated.
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of build condition.

3.1.64. .status.output

Description

BuildStatusOutput contains the status of the built image.

Type

object

Property	Type	Description
to	object	BuildStatusOutputTo describes the status of the built image with regards to image registry to which it was supposed to be pushed.

3.1.65. .status.output.to

Description

BuildStatusOutputTo describes the status of the built image with regards to image registry to which it was supposed to be pushed.

Type

object

Property	Type	Description
imageDigest	string	imageDigest is the digest of the built container image. The digest uniquely identifies the image in the registry to which it was pushed. Please note that this field may not always be set even if the push completes successfully - e.g. when the registry returns no digest or returns it in a format that the builder doesn't understand.

3.1.66. .status.stages

Description

stages contains details about each stage that occurs during the build including start time, duration (in milliseconds), and the steps that occurred within each stage.

Type

array

3.1.67. .status.stages[]

Description

StageInfo contains details about a build stage.

Type

object

Property	Type	Description
durationMilliseconds	integer	durationMilliseconds identifies how long the stage took to complete in milliseconds. Note: the duration of a stage can exceed the sum of the duration of the steps within the stage as not all actions are accounted for in explicit build steps.
name	string	name is a unique identifier for each build stage that occurs.
startTime	Time	startTime is a timestamp representing the server time when this Stage started. It is represented in RFC3339 form and is in UTC.
steps	array	steps contains details about each step that occurs during a build stage including start time and duration in milliseconds.
steps[]	object	StepInfo contains details about a build step.

3.1.68. .status.stages[].steps

Description

steps contains details about each step that occurs during a build stage including start time and duration in milliseconds.

Type

array

3.1.69. .status.stages[].steps[]

Description

StepInfo contains details about a build step.

Type

object

Property	Type	Description
durationMilliseconds	integer	durationMilliseconds identifies how long the step took to complete in milliseconds.

Property	Type	Description
name	string	name is a unique identifier for each build step.
startTime	Time	startTime is a timestamp representing the server time when this Step started. it is represented in RFC3339 form and is in UTC.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/build.openshift.io/v1/builds**
 - **GET**: list or watch objects of kind Build
- **/apis/build.openshift.io/v1/watch/builds**
 - **GET**: watch individual changes to a list of Build. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/build.openshift.io/v1/namespaces/{namespace}/builds**
 - **DELETE**: delete collection of Build
 - **GET**: list or watch objects of kind Build
 - **POST**: create a Build
- **/apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds**
 - **GET**: watch individual changes to a list of Build. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}**
 - **DELETE**: delete a Build
 - **GET**: read the specified Build
 - **PATCH**: partially update the specified Build
 - **PUT**: replace the specified Build
- **/apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds/{name}**
 - **GET**: watch changes to an object of kind Build. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/details**
 - **PUT**: replace details of the specified Build

- **/apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/webhooks**
 - **POST**: connect POST requests to webhooks of BuildConfig
- **/apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/webhooks/{path}**
 - **POST**: connect POST requests to webhooks of BuildConfig

3.2.1. /apis/build.openshift.io/v1/builds

HTTP method

GET

Description

list or watch objects of kind Build

Table 3.1. HTTP responses

HTTP code	Reponse body
200 - OK	BuildList schema
401 - Unauthorized	Empty

3.2.2. /apis/build.openshift.io/v1/watch/builds

HTTP method

GET

Description

watch individual changes to a list of Build. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.3. /apis/build.openshift.io/v1/namespaces/{namespace}/builds

HTTP method

DELETE

Description

delete collection of Build

Table 3.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v4 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Build

Table 3.5. HTTP responses

HTTP code	Reponse body
200 - OK	BuildList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Build

Table 3.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.7. Body parameters

Parameter	Type	Description
body	Build schema	

Table 3.8. HTTP responses

HTTP code	Response body
200 - OK	Build schema
201 - Created	Build schema
202 - Accepted	Build schema
401 - Unauthorized	Empty

3.2.4. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds

HTTP method

GET

Description

watch individual changes to a list of Build. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.5. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}

Table 3.10. Global path parameters

Parameter	Type	Description
name	string	name of the Build

HTTP method

DELETE

Description

delete a Build

Table 3.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v4 schema
202 - Accepted	Status_v4 schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Build

Table 3.13. HTTP responses

HTTP code	Response body
200 - OK	Build schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Build

Table 3.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.15. HTTP responses

HTTP code	Response body
200 - OK	Build schema
201 - Created	Build schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Build

Table 3.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.17. Body parameters

Parameter	Type	Description
body	Build schema	

Table 3.18. HTTP responses

HTTP code	Reponse body
200 - OK	Build schema
201 - Created	Build schema
401 - Unauthorized	Empty

3.2.6. /apis/build.openshift.io/v1/watch/namespaces/{namespace}/builds/{name}

Table 3.19. Global path parameters

Parameter	Type	Description
name	string	name of the Build

HTTP method

GET

Description

watch changes to an object of kind Build. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 3.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.7. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/details

Table 3.21. Global path parameters

Parameter	Type	Description
name	string	name of the Build

Table 3.22. Global query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method**PUT****Description**

replace details of the specified Build

Table 3.23. Body parameters

Parameter	Type	Description
body	Build schema	

Table 3.24. HTTP responses

HTTP code	Reponse body
200 - OK	Build schema
201 - Created	Build schema

HTTP code	Reponse body
401 - Unauthorized	Empty

3.2.8. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/we

Table 3.25. Global path parameters

Parameter	Type	Description
name	string	name of the Build

HTTP method

POST

Description

connect POST requests to webhooks of BuildConfig

Table 3.26. HTTP responses

HTTP code	Reponse body
200 - OK	string
401 - Unauthorized	Empty

3.2.9. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/we

Table 3.27. Global path parameters

Parameter	Type	Description
name	string	name of the Build

HTTP method

POST

Description

connect POST requests to webhooks of BuildConfig

Table 3.28. HTTP responses

HTTP code	Reponse body
200 - OK	string
401 - Unauthorized	Empty

CHAPTER 4. BUILDLOG [BUILD.OPENSIFT.IO/V1]

Description

BuildLog is the (unused) resource associated with the build log redirector

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

4.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/log**
 - **GET**: read log of the specified Build

4.2.1. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/log

Table 4.1. Global path parameters

Parameter	Type	Description
name	string	name of the BuildLog

HTTP method**GET****Description**

read log of the specified Build

Table 4.2. HTTP responses

HTTP code	Reponse body
200 - OK	BuildLog schema
401 - Unauthorized	Empty

CHAPTER 5. BUILDREQUEST [BUILD.OPENSIFT.IO/V1]

Description

BuildRequest is the resource used to pass parameters to build generator

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

5.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
binary	object	BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.
dockerStrategyOptions	object	DockerStrategyOptions contains extra strategy options for container image builds
env	array (EnvVar)	env contains additional environment variables you want to pass into a builder container.
from	ObjectReference	from is the reference to the ImageStreamTag that triggered the build.

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
lastVersion	integer	lastVersion (optional) is the LastVersion of the BuildConfig that was used to generate the build. If the BuildConfig in the generator doesn't match, a build will not be generated.
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
revision	object	SourceRevision is the revision or commit information from the source for the build
sourceStrategyOptions	object	SourceStrategyOptions contains extra strategy options for Source builds
triggeredBy	array	triggeredBy describes which triggers started the most recent update to the build configuration and contains information about those triggers.
triggeredBy[]	object	BuildTriggerCause holds information about a triggered build. It is used for displaying build trigger data for each build and build configuration in oc describe. It is also used to describe which triggers led to the most recent update in the build configuration.

Property	Type	Description
triggeredByImage	ObjectReference	triggeredByImage is the Image that triggered this build.

5.1.1. .binary

Description

BinaryBuildSource describes a binary file to be used for the Docker and Source build strategies, where the file will be extracted and used as the build source.

Type

object

Property	Type	Description
asFile	string	asFile indicates that the provided binary input should be considered a single file within the build input. For example, specifying "webapp.war" would place the provided binary as /webapp.war for the builder. If left empty, the Docker and Source build strategies assume this file is a zip, tar, or tar.gz file and extract it as the source. The custom strategy receives this binary as standard input. This filename may not contain slashes or be '..' or '..'.

5.1.2. .dockerStrategyOptions

Description

DockerStrategyOptions contains extra strategy options for container image builds

Type

object

Property	Type	Description
buildArgs	array (EnvVar)	Args contains any build arguments that are to be passed to Docker. See https://docs.docker.com/engine/reference/builder/#/arg for more details

Property	Type	Description
noCache	boolean	noCache overrides the docker-strategy noCache option in the build config

5.1.3. .revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

5.1.4. .revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control

Property	Type	Description
message	string	message is the description of a specific commit

5.1.5. .revision.git.author

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.6. .revision.git.committer

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.7. .sourceStrategyOptions

Description

SourceStrategyOptions contains extra strategy options for Source builds

Type

object

Property	Type	Description
incremental	boolean	incremental overrides the source-strategy incremental option in the build config

5.1.8. .triggeredBy

Description

triggeredBy describes which triggers started the most recent update to the build configuration and contains information about those triggers.

Type

array

5.1.9. .triggeredBy[]

Description

BuildTriggerCause holds information about a triggered build. It is used for displaying build trigger data for each build and build configuration in oc describe. It is also used to describe which triggers led to the most recent update in the build configuration.

Type

object

Property	Type	Description
bitbucketWebHook	object	BitbucketWebHookCause has information about a Bitbucket webhook that triggered a build.
genericWebHook	object	GenericWebHookCause holds information about a generic WebHook that triggered a build.
githubWebHook	object	GitHubWebHookCause has information about a GitHub webhook that triggered a build.
gitlabWebHook	object	GitLabWebHookCause has information about a GitLab webhook that triggered a build.
imageChangeBuild	object	ImageChangeCause contains information about the image that triggered a build
message	string	message is used to store a human readable message for why the build was triggered. E.g.: "Manually triggered by user", "Configuration change",etc.

5.1.10. .triggeredBy[].bitbucketWebHook

Description

BitbucketWebHookCause has information about a Bitbucket webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	Secret is the obfuscated webhook secret that triggered a build.

5.1.11. .triggeredBy[].bitbucketWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

5.1.12. .triggeredBy[].bitbucketWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control

Property	Type	Description
commit	string	commit is the commit hash identifying a specific commit
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

5.1.13. `.triggeredBy[].bitbucketWebHook.revision.git.author`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.14. `.triggeredBy[].bitbucketWebHook.revision.git.committer`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.15. `.triggeredBy[].genericWebHook`

Description

GenericWebHookCause holds information about a generic WebHook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	secret is the obfuscated webhook secret that triggered a build.

5.1.16. .triggeredBy[].genericWebHook.revision**Description**

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

5.1.17. .triggeredBy[].genericWebHook.revision.git**Description**

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit

Property	Type	Description
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

5.1.18. .triggeredBy[].genericWebHook.revision.git.author

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.19. .triggeredBy[].genericWebHook.revision.git.committer

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.20. .triggeredBy[].githubWebHook

Description

GitHubWebHookCause has information about a GitHub webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	secret is the obfuscated webhook secret that triggered a build.

5.1.21. .triggeredBy[].githubWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

5.1.22. .triggeredBy[].githubWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit

Property	Type	Description
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

5.1.23. `.triggeredBy[].githubWebHook.revision.git.author`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.24. `.triggeredBy[].githubWebHook.revision.git.committer`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.25. `.triggeredBy[].gitlabWebHook`

Description

GitLabWebHookCause has information about a GitLab webhook that triggered a build.

Type

object

Property	Type	Description
revision	object	SourceRevision is the revision or commit information from the source for the build
secret	string	Secret is the obfuscated webhook secret that triggered a build.

5.1.26. .triggeredBy[].gitlabWebHook.revision

Description

SourceRevision is the revision or commit information from the source for the build

Type

object

Required

- **type**

Property	Type	Description
git	object	GitSourceRevision is the commit information from a git source for a build
type	string	type of the build source, may be one of 'Source', 'Dockerfile', 'Binary', or 'Images'

5.1.27. .triggeredBy[].gitlabWebHook.revision.git

Description

GitSourceRevision is the commit information from a git source for a build

Type

object

Property	Type	Description
author	object	SourceControlUser defines the identity of a user of source control
commit	string	commit is the commit hash identifying a specific commit

Property	Type	Description
committer	object	SourceControlUser defines the identity of a user of source control
message	string	message is the description of a specific commit

5.1.28. `.triggeredBy[].gitlabWebHook.revision.git.author`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.29. `.triggeredBy[].gitlabWebHook.revision.git.committer`

Description

SourceControlUser defines the identity of a user of source control

Type

object

Property	Type	Description
email	string	email of the source control user
name	string	name of the source control user

5.1.30. `.triggeredBy[].imageChangeBuild`

Description

ImageChangeCause contains information about the image that triggered a build

Type

object

Property	Type	Description
fromRef	ObjectReference	fromRef contains detailed information about an image that triggered a build.
imageID	string	imageID is the ID of the image that triggered a new build.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/clone**
 - **POST**: create clone of a Build
- **/apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/instantiate**
 - **POST**: create instantiate of a BuildConfig

5.2.1. /apis/build.openshift.io/v1/namespaces/{namespace}/builds/{name}/clone

Table 5.1. Global path parameters

Parameter	Type	Description
name	string	name of the BuildRequest

Table 5.2. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method**POST****Description**

create clone of a Build

Table 5.3. Body parameters

Parameter	Type	Description
body	BuildRequest schema	

Table 5.4. HTTP responses

HTTP code	Reponse body
200 - OK	BuildRequest schema
201 - Created	BuildRequest schema
202 - Accepted	BuildRequest schema
401 - Unauthorized	Empty

5.2.2. /apis/build.openshift.io/v1/namespaces/{namespace}/buildconfigs/{name}/ins**Table 5.5. Global path parameters**

Parameter	Type	Description
name	string	name of the BuildRequest

Table 5.6. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method**POST****Description**

create instantiate of a BuildConfig

Table 5.7. Body parameters

Parameter	Type	Description
body	BuildRequest schema	

Table 5.8. HTTP responses

HTTP code	Reponse body
200 - OK	Build schema
201 - Created	Build schema
202 - Accepted	Build schema
401 - Unauthorized	Empty

CHAPTER 6. CRONJOB [BATCH/V1]

Description

CronJob represents the configuration of a single cron job.

Type

object

6.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	CronJobSpec describes how the job execution will look like and when it will actually run.
status	object	CronJobStatus represents the current state of a cron job.

6.1.1. .spec

Description

CronJobSpec describes how the job execution will look like and when it will actually run.

Type

object

Required

- **schedule**
- **jobTemplate**

Property	Type	Description
concurrencyPolicy	string	Specifies how to treat concurrent executions of a Job. Valid values are: - "Allow" (default): allows CronJobs to run concurrently; - "Forbid": forbids concurrent runs, skipping next run if previous run hasn't finished yet; - "Replace": cancels currently running job and replaces it with a new one Possible enum values: - "Allow" allows CronJobs to run concurrently. - "Forbid" forbids concurrent runs, skipping next run if previous hasn't finished yet. - "Replace" cancels currently running job and replaces it with a new one.
failedJobsHistoryLimit	integer	The number of failed finished jobs to retain. Value must be non-negative integer. Defaults to 1.
jobTemplate	object	JobTemplateSpec describes the data a Job should have when created from a template
schedule	string	The schedule in Cron format, see https://en.wikipedia.org/wiki/Cron .
startingDeadlineSeconds	integer	Optional deadline in seconds for starting the job if it misses scheduled time for any reason. Missed jobs executions will be counted as failed ones.

Property	Type	Description
successfulJobsHistoryLimit	integer	The number of successful finished jobs to retain. Value must be non-negative integer. Defaults to 3.
suspend	boolean	This flag tells the controller to suspend subsequent executions, it does not apply to already started executions. Defaults to false.
timeZone	string	The time zone name for the given schedule, see https://en.wikipedia.org/wiki/List_of_tz_database_time_zones . If not specified, this will default to the time zone of the kube-controller-manager process. The set of valid time zone names and the time zone offset is loaded from the system-wide time zone database by the API server during CronJob validation and the controller manager during execution. If no system-wide time zone database can be found a bundled version of the database is used instead. If the time zone name becomes invalid during the lifetime of a CronJob or due to a change in host configuration, the controller will stop creating new new Jobs and will create a system event with the reason UnknownTimeZone. More information can be found in https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/#time-zones

6.1.2. .spec.jobTemplate

Description

JobTemplateSpec describes the data a Job should have when created from a template

Type

object

Property	Type	Description

Property	Type	Description
metadata	ObjectMeta	Standard object's metadata of the jobs created from this template. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	JobSpec describes how the job execution will look like.

6.1.3. .spec.jobTemplate.spec

Description

JobSpec describes how the job execution will look like.

Type

object

Required

- **template**

Property	Type	Description
activeDeadlineSeconds	integer	Specifies the duration in seconds relative to the startTime that the job may be continuously active before the system tries to terminate it; value must be positive integer. If a Job is suspended (at creation or through an update), this timer will effectively be stopped and reset when the Job is resumed again.
backoffLimit	integer	Specifies the number of retries before marking this job failed. Defaults to 6

Property	Type	Description
backoffLimitPerIndex	integer	Specifies the limit for the number of retries within an index before marking this index as failed. When enabled the number of failures per index is kept in the pod's <code>batch.kubernetes.io/job-index-failure-count</code> annotation. It can only be set when Job's <code>completionMode=Indexed</code> , and the Pod's restart policy is <code>Never</code> . The field is immutable.
completionMode	string	<code>completionMode</code> specifies how Pod completions are tracked. It can be NonIndexed (default) or Indexed. NonIndexed means that the Job is considered complete when there have been <code>.spec.completions</code> successfully completed Pods. Each Pod completion is homologous to each other. Indexed means that the Pods of a Job get an associated completion index from 0 to <code>(.spec.completions - 1)</code>, available in the annotation <code>batch.kubernetes.io/job-completion-index</code>. The Job is considered complete when there is one successfully completed Pod for each index. When value is Indexed, <code>.spec.completions</code> must be specified and <code>.spec.parallelism</code> must be less than or equal to 10^5. In addition, The Pod name takes the form <code>\$(job-name)-\$(index)-\$(random-string)</code>, the Pod hostname takes the form <code>\$(job-name)-\$(index)</code>. More completion modes can be added in the future. If the Job controller observes a mode that it doesn't recognize, which is possible during upgrades due to version skew, the controller skips updates for the Job.

Property	Type	Possible enum values: - Description
		mode. In this mode, the Pods of a Job get an associated completion index from 0 to (.spec.completions - 1). The Job is considered complete when a Pod completes for each completion index. - "NonIndexed" is a Job completion mode. In this mode, the Job is considered complete when there have been .spec.completions successfully completed Pods. Pod completions are homologous to each other.
completions	integer	Specifies the desired number of successfully finished pods the job should be run with. Setting to null means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Property	Type	Description
managedBy	string	ManagedBy field indicates the controller that manages a Job. The k8s Job controller reconciles jobs which don't have this field at all or the field value is the reserved string kubernetes.io/job-controller, but skips reconciling Jobs with a custom value for this field. The value must be a valid domain-prefixed path (e.g. acme.io/foo) - all characters before the first "/" must be a valid subdomain as defined by RFC 1123. All characters trailing the first "/" must be valid HTTP Path characters as defined by RFC 3986. The value cannot exceed 63 characters. This field is immutable. This field is beta-level. The job controller accepts setting the field when the feature gate JobManagedBy is enabled (enabled by default).
manualSelector	boolean	manualSelector controls generation of pod labels and pod selectors. Leave manualSelector unset unless you are certain what you are doing. When false or unset, the system pick labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see manualSelector=true in jobs that were created with the old extensions/v1beta1 API. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector

Property	Type	Description
maxFailedIndexes	integer	Specifies the maximal number of failed indexes before marking the Job as failed, when <code>backoffLimitPerIndex</code> is set. Once the number of failed indexes exceeds this number the entire Job is marked as Failed and its execution is terminated. When left as null the job continues execution of all of its indexes and is marked with the Complete Job condition. It can only be specified when <code>backoffLimitPerIndex</code> is set. It can be null or up to completions. It is required and must be less than or equal to 10^4 when is completions greater than 10^5 .
parallelism	integer	Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when $((\text{.spec.completions} - \text{.status.successful}) < \text{.spec.parallelism})$, i.e. when the work left to do is less than max parallelism. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
podFailurePolicy	object	PodFailurePolicy describes how failed pods influence the <code>backoffLimit</code> .

Property	Type	Description
podReplacementPolicy	string	podReplacementPolicy specifies when to create replacement Pods. Possible values are: - TerminatingOrFailed means that we recreate pods when they are terminating (has a metadata.deletionTimestamp) or failed. - Failed means to wait until a previously created Pod is fully terminated (has phase Failed or Succeeded) before creating a replacement Pod. When using podFailurePolicy, Failed is the the only allowed value. TerminatingOrFailed and Failed are allowed values when podFailurePolicy is not in use. This is an beta field. To use this, enable the JobPodReplacementPolicy feature toggle. This is on by default. Possible enum values: - "Failed" means to wait until a previously created Pod is fully terminated (has phase Failed or Succeeded) before creating a replacement Pod. - "TerminatingOrFailed" means that we recreate pods when they are terminating (has a metadata.deletionTimestamp) or failed.
selector	LabelSelector	A label query over pods that should match the pod count. Normally, the system sets this field for you. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
successPolicy	object	SuccessPolicy describes when a Job can be declared as succeeded based on the success of some indexes.

Property	Type	Description
suspend	boolean	suspend specifies whether the Job controller should create Pods or not. If a Job is created with suspend set to true, no Pods are created by the Job controller. If a Job is suspended after creation (i.e. the flag goes from false to true), the Job controller will delete all active Pods associated with this Job. Users must design their workload to gracefully handle this. Suspending a Job will reset the StartTime field of the Job, effectively resetting the ActiveDeadlineSeconds timer too. Defaults to false.
template	PodTemplateSpec	Describes the pod that will be created when executing a job. The only allowed template.spec.restartPolicy values are "Never" or "OnFailure". More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
ttlSecondsAfterFinished	integer	ttlSecondsAfterFinished limits the lifetime of a Job that has finished execution (either Complete or Failed). If this field is set, ttlSecondsAfterFinished after the Job finishes, it is eligible to be automatically deleted. When the Job is being deleted, its lifecycle guarantees (e.g. finalizers) will be honored. If this field is unset, the Job won't be automatically deleted. If this field is set to zero, the Job becomes eligible to be deleted immediately after it finishes.

6.1.4. .spec.jobTemplate.spec.podFailurePolicy

Description

PodFailurePolicy describes how failed pods influence the backoffLimit.

Type

object

Required

- **rules**

Property	Type	Description
rules	array	A list of pod failure policy rules. The rules are evaluated in order. Once a rule matches a Pod failure, the remaining of the rules are ignored. When no rule matches the Pod failure, the default handling applies - the counter of pod failures is incremented and it is checked against the backoffLimit. At most 20 elements are allowed.
rules[]	object	PodFailurePolicyRule describes how a pod failure is handled when the requirements are met. One of onExitCodes and onPodConditions, but not both, can be used in each rule.

6.1.5. .spec.jobTemplate.spec.podFailurePolicy.rules

Description

A list of pod failure policy rules. The rules are evaluated in order. Once a rule matches a Pod failure, the remaining of the rules are ignored. When no rule matches the Pod failure, the default handling applies - the counter of pod failures is incremented and it is checked against the backoffLimit. At most 20 elements are allowed.

Type

array

6.1.6. .spec.jobTemplate.spec.podFailurePolicy.rules[]

Description

PodFailurePolicyRule describes how a pod failure is handled when the requirements are met. One of onExitCodes and onPodConditions, but not both, can be used in each rule.

Type

object

Required

- **action**

Property	Type	Description

Property	Type	Description
action	string	Specifies the action taken on a pod failure when the requirements are satisfied. Possible values are: - FailJob: indicates that the pod's job is marked as Failed and all running pods are terminated. - FailIndex: indicates that the pod's index is marked as Failed and will not be restarted. - Ignore: indicates that the counter towards the .backoffLimit is not incremented and a replacement pod is created. - Count: indicates that the pod is handled in the default way - the counter towards the .backoffLimit is incremented. Additional values are considered to be added in the future. Clients should react to an unknown action by skipping the rule. Possible enum values: - "Count" This is an action which might be taken on a pod failure - the pod failure is handled in the default way - the counter towards .backoffLimit, represented by the job's .status.failed field, is incremented. - "FailIndex" This is an action which might be taken on a pod failure - mark the Job's index as failed to avoid restarts within this index. This action can only be used when backoffLimitPerIndex is set. - "FailJob" This is an action which might be taken on a pod failure - mark the pod's job as Failed and terminate all running pods. - "Ignore" This is an action which might be taken on a pod failure - the counter towards .backoffLimit, represented by the job's .status.failed field, is not incremented and a replacement pod is created.

Property	Type	Description
onExitCodes	object	PodFailurePolicyOnExitCodesRequirement describes the requirement for handling a failed pod based on its container exit codes. In particular, it lookups the <code>.state.terminated.exitCode</code> for each app container and init container status, represented by the <code>.status.containerStatuses</code> and <code>.status.initContainerStatuses</code> fields in the Pod status, respectively. Containers completed with success (exit code 0) are excluded from the requirement check.
onPodConditions	array	Represents the requirement on the pod conditions. The requirement is represented as a list of pod condition patterns. The requirement is satisfied if at least one pattern matches an actual pod condition. At most 20 elements are allowed.
onPodConditions[]	object	PodFailurePolicyOnPodConditionPattern describes a pattern for matching an actual pod condition type.

6.1.7. `.spec.jobTemplate.spec.podFailurePolicy.rules[].onExitCodes`

Description

PodFailurePolicyOnExitCodesRequirement describes the requirement for handling a failed pod based on its container exit codes. In particular, it lookups the `.state.terminated.exitCode` for each app container and init container status, represented by the `.status.containerStatuses` and `.status.initContainerStatuses` fields in the Pod status, respectively. Containers completed with success (exit code 0) are excluded from the requirement check.

Type

object

Required

- **operator**
- **values**

Property	Type	Description
containerName	string	Restricts the check for exit codes to the container with the specified name. When null, the rule applies to all containers. When specified, it should match one the container or initContainer names in the pod template.
operator	string	Represents the relationship between the container exit code(s) and the specified values. Containers completed with success (exit code 0) are excluded from the requirement check. Possible values are: - In: the requirement is satisfied if at least one container exit code (might be multiple if there are multiple containers not restricted by the 'containerName' field) is in the set of specified values. - NotIn: the requirement is satisfied if at least one container exit code (might be multiple if there are multiple containers not restricted by the 'containerName' field) is not in the set of specified values. Additional values are considered to be added in the future. Clients should react to an unknown operator by assuming the requirement is not satisfied. Possible enum values: - "In" - "NotIn"
values	array (integer)	Specifies the set of values. Each returned container exit code (might be multiple in case of multiple containers) is checked against this set of values with respect to the operator. The list of values must be ordered and must not contain duplicates. Value '0' cannot be used for the In operator. At least one element is required. At most 255 elements are allowed.

6.1.8. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions

Description

Represents the requirement on the pod conditions. The requirement is represented as a list of pod condition patterns. The requirement is satisfied if at least one pattern matches an actual pod condition. At most 20 elements are allowed.

Type

array

6.1.9. .spec.jobTemplate.spec.podFailurePolicy.rules[].onPodConditions[]**Description**

PodFailurePolicyOnPodConditionsPattern describes a pattern for matching an actual pod condition type.

Type

object

Required

- **type**
- **status**

Property	Type	Description
status	string	Specifies the required Pod condition status. To match a pod condition it is required that the specified status equals the pod condition status. Defaults to True.
type	string	Specifies the required Pod condition type. To match a pod condition it is required that specified type equals the pod condition type.

6.1.10. .spec.jobTemplate.spec.successPolicy**Description**

SuccessPolicy describes when a Job can be declared as succeeded based on the success of some indexes.

Type

object

Required

- **rules**

Property	Type	Description
rules	array	rules represents the list of alternative rules for the declaring the Jobs as successful before .status.succeeded >= .spec.completions . Once any of the rules are met, the "SucceededCriteriaMet" condition is added, and the lingering pods are removed. The terminal state for such a Job has the "Complete" condition. Additionally, these rules are evaluated in order; Once the Job meets one of the rules, other rules are ignored. At most 20 elements are allowed.
rules[]	object	SuccessPolicyRule describes rule for declaring a Job as succeeded. Each rule must have at least one of the "succeededIndexes" or "succeededCount" specified.

6.1.11. .spec.jobTemplate.spec.successPolicy.rules

Description

rules represents the list of alternative rules for the declaring the Jobs as successful before **.status.succeeded >= .spec.completions**. Once any of the rules are met, the "SucceededCriteriaMet" condition is added, and the lingering pods are removed. The terminal state for such a Job has the "Complete" condition. Additionally, these rules are evaluated in order; Once the Job meets one of the rules, other rules are ignored. At most 20 elements are allowed.

Type

array

6.1.12. .spec.jobTemplate.spec.successPolicy.rules[]

Description

SuccessPolicyRule describes rule for declaring a Job as succeeded. Each rule must have at least one of the "succeededIndexes" or "succeededCount" specified.

Type

object

Property	Type	Description

Property	Type	Description
succeededCount	integer	succeededCount specifies the minimal required size of the actual set of the succeeded indexes for the Job. When succeededCount is used along with succeededIndexes, the check is constrained only to the set of indexes specified by succeededIndexes. For example, given that succeededIndexes is "1-4", succeededCount is "3", and completed indexes are "1", "3", and "5", the Job isn't declared as succeeded because only "1" and "3" indexes are considered in that rules. When this field is null, this doesn't default to any value and is never evaluated at any time. When specified it needs to be a positive integer.
succeededIndexes	string	succeededIndexes specifies the set of indexes which need to be contained in the actual set of the succeeded indexes for the Job. The list of indexes must be within 0 to ".spec.completions-1" and must not contain duplicates. At least one element is required. The indexes are represented as intervals separated by commas. The intervals can be a decimal integer or a pair of decimal integers separated by a hyphen. The number are listed in represented by the first and last element of the series, separated by a hyphen. For example, if the completed indexes are 1, 3, 4, 5 and 7, they are represented as "1,3-5,7". When this field is null, this field doesn't default to any value and is never evaluated at any time.

6.1.13. .status

Description

CronJobStatus represents the current state of a cron job.

Type

object

Property	Type	Description
active	array (ObjectReference)	A list of pointers to currently running jobs.
lastScheduleTime	Time	Information when was the last time the job was successfully scheduled.
lastSuccessfulTime	Time	Information when was the last time the job successfully completed.

6.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/batch/v1/cronjobs**
 - **GET**: list or watch objects of kind CronJob
- **/apis/batch/v1/watch/cronjobs**
 - **GET**: watch individual changes to a list of CronJob. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/batch/v1/namespaces/{namespace}/cronjobs**
 - **DELETE**: delete collection of CronJob
 - **GET**: list or watch objects of kind CronJob
 - **POST**: create a CronJob
- **/apis/batch/v1/watch/namespaces/{namespace}/cronjobs**
 - **GET**: watch individual changes to a list of CronJob. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/batch/v1/namespaces/{namespace}/cronjobs/{name}**
 - **DELETE**: delete a CronJob
 - **GET**: read the specified CronJob
 - **PATCH**: partially update the specified CronJob
 - **PUT**: replace the specified CronJob
- **/apis/batch/v1/watch/namespaces/{namespace}/cronjobs/{name}**
 - **GET**: watch changes to an object of kind CronJob. deprecated: use the 'watch' parameter with a list operation instead.

with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

- **/apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status**
 - **GET**: read status of the specified CronJob
 - **PATCH**: partially update status of the specified CronJob
 - **PUT**: replace status of the specified CronJob

6.2.1. /apis/batch/v1/cronjobs

HTTP method

GET

Description

list or watch objects of kind CronJob

Table 6.1. HTTP responses

HTTP code	Reponse body
200 - OK	CronJobList schema
401 - Unauthorized	Empty

6.2.2. /apis/batch/v1/watch/cronjobs

HTTP method

GET

Description

watch individual changes to a list of CronJob. deprecated: use the 'watch' parameter with a list operation instead.

Table 6.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

6.2.3. /apis/batch/v1/namespaces/{namespace}/cronjobs

HTTP method

DELETE

Description

delete collection of CronJob

Table 6.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 6.4. HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind CronJob

Table 6.5. HTTP responses

HTTP code	Response body
200 - OK	CronJobList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a CronJob

Table 6.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.7. Body parameters

Parameter	Type	Description
body	CronJob schema	

Table 6.8. HTTP responses

HTTP code	Response body
200 - OK	CronJob schema
201 - Created	CronJob schema
202 - Accepted	CronJob schema
401 - Unauthorized	Empty

6.2.4. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs

HTTP method

GET

Description

watch individual changes to a list of CronJob. deprecated: use the 'watch' parameter with a list operation instead.

Table 6.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

6.2.5. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}

Table 6.10. Global path parameters

Parameter	Type	Description
name	string	name of the CronJob

HTTP method

DELETE

Description

delete a CronJob

Table 6.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 6.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified CronJob

Table 6.13. HTTP responses

HTTP code	Reponse body
200 - OK	CronJob schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified CronJob

Table 6.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.15. HTTP responses

HTTP code	Reponse body
200 - OK	CronJob schema
201 - Created	CronJob schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified CronJob

Table 6.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.17. Body parameters

Parameter	Type	Description
body	CronJob schema	

Table 6.18. HTTP responses

HTTP code	Reponse body
200 - OK	CronJob schema
201 - Created	CronJob schema
401 - Unauthorized	Empty

6.2.6. /apis/batch/v1/watch/namespaces/{namespace}/cronjobs/{name}

Table 6.19. Global path parameters

Parameter	Type	Description
name	string	name of the CronJob

HTTP method

GET

Description

watch changes to an object of kind CronJob. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 6.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

6.2.7. /apis/batch/v1/namespaces/{namespace}/cronjobs/{name}/status

Table 6.21. Global path parameters

Parameter	Type	Description
name	string	name of the CronJob

HTTP method

GET

Description

read status of the specified CronJob

Table 6.22. HTTP responses

HTTP code	Response body
200 - OK	CronJob schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified CronJob

Table 6.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.24. HTTP responses

HTTP code	Response body
200 - OK	CronJob schema
201 - Created	CronJob schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified CronJob

Table 6.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 6.26. Body parameters

Parameter	Type	Description
body	CronJob schema	

Table 6.27. HTTP responses

HTTP code	Reponse body
200 - OK	CronJob schema
201 - Created	CronJob schema
401 - Unauthorized	Empty

CHAPTER 7. DAEMONSET [APPS/V1]

Description

DaemonSet represents the configuration of a daemon set.

Type

object

7.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	DaemonSetSpec is the specification of a daemon set.
status	object	DaemonSetStatus represents the current status of a daemon set.

7.1.1. .spec

Description

DaemonSetSpec is the specification of a daemon set.

Type

object

Required

- **selector**
- **template**

Property	Type	Description
minReadySeconds	integer	The minimum number of seconds for which a newly created DaemonSet pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready).
revisionHistoryLimit	integer	The number of old history to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.
selector	LabelSelector	A label query over pods that are managed by the daemon set. Must match in order to be controlled. It must match the pod template's labels. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
template	PodTemplateSpec	An object that describes the pod that will be created. The DaemonSet will create exactly one copy of this pod on every node that matches the template's node selector (or on every node if no node selector is specified). The only allowed template.spec.restartPolicy value is "Always". More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#pod-template

Property	Type	Description
updateStrategy	object	DaemonSetUpdateStrategy is a struct used to control the update strategy for a DaemonSet.

7.1.2. .spec.updateStrategy

Description

DaemonSetUpdateStrategy is a struct used to control the update strategy for a DaemonSet.

Type

object

Property	Type	Description
rollingUpdate	object	Spec to control the desired behavior of daemon set rolling update.
type	string	Type of daemon set update. Can be "RollingUpdate" or "OnDelete". Default is RollingUpdate. Possible enum values: - "OnDelete" Replace the old daemons only when it's killed - "RollingUpdate" Replace the old daemons by new ones using rolling update i.e replace them on each node one after the other.

7.1.3. .spec.updateStrategy.rollingUpdate

Description

Spec to control the desired behavior of daemon set rolling update.

Type

object

Property	Type	Description

Property	Type	Description
maxSurge	IntOrString	The maximum number of nodes with an existing available DaemonSet pod that can have an updated DaemonSet pod during an update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up to a minimum of 1. Default value is 0. Example: when this is set to 30%, at most 30% of the total number of nodes that should be running the daemon pod (i.e. <code>status.desiredNumberScheduled</code>) can have their a new pod created before the old pod is marked as deleted. The update starts by launching new pods on 30% of nodes. Once an updated pod is available (Ready for at least <code>minReadySeconds</code>) the old DaemonSet pod on that node is marked deleted. If the old pod becomes unavailable for any reason (Ready transitions to false, is evicted, or is drained) an updated pod is immediately created on that node without considering surge limits. Allowing surge implies the possibility that the resources consumed by the daemonset on any given node can double if the readiness check fails, and so resource intensive daemonsets should take into account that they may cause evictions during disruption.

Property	Type	Description
maxUnavailable	IntOrString	The maximum number of DaemonSet pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of total number of DaemonSet pods at the start of the update (ex: 10%). Absolute number is calculated from percentage by rounding up. This cannot be 0 if MaxSurge is 0. Default value is 1. Example: when this is set to 30%, at most 30% of the total number of nodes that should be running the daemon pod (i.e. <code>status.desiredNumberScheduled</code>) can have their pods stopped for an update at any given time. The update starts by stopping at most 30% of those DaemonSet pods and then brings up new DaemonSet pods in their place. Once the new pods are available, it then proceeds onto other DaemonSet pods, thus ensuring that at least 70% of original number of DaemonSet pods are available at all times during the update.

7.1.4. .status

Description

DaemonSetStatus represents the current status of a daemon set.

Type

object

Required

- **currentNumberScheduled**
- **numberMisscheduled**
- **desiredNumberScheduled**
- **numberReady**

Property	Type	Description
collisionCount	integer	Count of hash collisions for the DaemonSet. The DaemonSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision.
conditions	array	Represents the latest available observations of a DaemonSet's current state.
conditions[]	object	DaemonSetCondition describes the state of a DaemonSet at a certain point.
currentNumberScheduled	integer	The number of nodes that are running at least 1 daemon pod and are supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
desiredNumberScheduled	integer	The total number of nodes that should be running the daemon pod (including nodes correctly running the daemon pod). More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
numberAvailable	integer	The number of nodes that should be running the daemon pod and have one or more of the daemon pod running and available (ready for at least spec.minReadySeconds)
numberMisscheduled	integer	The number of nodes that are running the daemon pod, but are not supposed to run the daemon pod. More info: https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

Property	Type	Description
numberReady	integer	numberReady is the number of nodes that should be running the daemon pod and have one or more of the daemon pod running with a Ready Condition.
numberUnavailable	integer	The number of nodes that should be running the daemon pod and have none of the daemon pod running and available (ready for at least spec.minReadySeconds)
observedGeneration	integer	The most recent generation observed by the daemon set controller.
updatedNumberScheduled	integer	The total number of nodes that are running updated daemon pod

7.1.5. .status.conditions

Description

Represents the latest available observations of a DaemonSet's current state.

Type

array

7.1.6. .status.conditions[]

Description

DaemonSetCondition describes the state of a DaemonSet at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	Last time the condition transitioned from one status to another.

Property	Type	Description
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of DaemonSet condition.

7.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps/v1/daemonsets**
 - **GET**: list or watch objects of kind DaemonSet
- **/apis/apps/v1/watch/daemonsets**
 - **GET**: watch individual changes to a list of DaemonSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/daemonsets**
 - **DELETE**: delete collection of DaemonSet
 - **GET**: list or watch objects of kind DaemonSet
 - **POST**: create a DaemonSet
- **/apis/apps/v1/watch/namespaces/{namespace}/daemonsets**
 - **GET**: watch individual changes to a list of DaemonSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}**
 - **DELETE**: delete a DaemonSet
 - **GET**: read the specified DaemonSet
 - **PATCH**: partially update the specified DaemonSet
 - **PUT**: replace the specified DaemonSet
- **/apis/apps/v1/watch/namespaces/{namespace}/daemonsets/{name}**

- **GET**: watch changes to an object of kind DaemonSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status**
 - **GET**: read status of the specified DaemonSet
 - **PATCH**: partially update status of the specified DaemonSet
 - **PUT**: replace status of the specified DaemonSet

7.2.1. /apis/apps/v1/daemonsets

HTTP method

GET

Description

list or watch objects of kind DaemonSet

Table 7.1. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSetList schema
401 - Unauthorized	Empty

7.2.2. /apis/apps/v1/watch/daemonsets

HTTP method

GET

Description

watch individual changes to a list of DaemonSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 7.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

7.2.3. /apis/apps/v1/namespaces/{namespace}/daemonsets

HTTP method

DELETE

Description

delete collection of DaemonSet

Table 7.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 7.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind DaemonSet

Table 7.5. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSetList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a DaemonSet

Table 7.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.7. Body parameters

Parameter	Type	Description
body	DaemonSet schema	

Table 7.8. HTTP responses

HTTP code	Response body
200 - OK	DaemonSet schema
201 - Created	DaemonSet schema
202 - Accepted	DaemonSet schema
401 - Unauthorized	Empty

7.2.4. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets

HTTP method

GET

Description

watch individual changes to a list of DaemonSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 7.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

7.2.5. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}

Table 7.10. Global path parameters

Parameter	Type	Description
name	string	name of the DaemonSet

HTTP method

DELETE

Description

delete a DaemonSet

Table 7.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 7.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified DaemonSet

Table 7.13. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSet schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified DaemonSet

Table 7.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.15. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSet schema
201 - Created	DaemonSet schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified DaemonSet

Table 7.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.17. Body parameters

Parameter	Type	Description
body	DaemonSet schema	

Table 7.18. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSet schema
201 - Created	DaemonSet schema
401 - Unauthorized	Empty

7.2.6. /apis/apps/v1/watch/namespaces/{namespace}/daemonsets/{name}

Table 7.19. Global path parameters

Parameter	Type	Description
name	string	name of the DaemonSet

HTTP method

GET

Description

watch changes to an object of kind DaemonSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 7.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

7.2.7. /apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status

Table 7.21. Global path parameters

Parameter	Type	Description
name	string	name of the DaemonSet

HTTP method

GET

Description

read status of the specified DaemonSet

Table 7.22. HTTP responses

HTTP code	Response body
200 - OK	DaemonSet schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified DaemonSet

Table 7.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.24. HTTP responses

HTTP code	Response body
200 - OK	DaemonSet schema
201 - Created	DaemonSet schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified DaemonSet

Table 7.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 7.26. Body parameters

Parameter	Type	Description
body	DaemonSet schema	

Table 7.27. HTTP responses

HTTP code	Reponse body
200 - OK	DaemonSet schema
201 - Created	DaemonSet schema
401 - Unauthorized	Empty

CHAPTER 8. DEPLOYMENT [APPS/V1]

Description

Deployment enables declarative updates for Pods and ReplicaSets.

Type

object

8.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	DeploymentSpec is the specification of the desired behavior of the Deployment.
status	object	DeploymentStatus is the most recently observed status of the Deployment.

8.1.1. .spec

Description

DeploymentSpec is the specification of the desired behavior of the Deployment.

Type

object

Required

- **selector**
- **template**

Property	Type	Description
minReadySeconds	integer	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
paused	boolean	Indicates that the deployment is paused.
progressDeadlineSeconds	integer	The maximum time in seconds for a deployment to make progress before it is considered to be failed. The deployment controller will continue to process failed deployments and a condition with a ProgressDeadlineExceeded reason will be surfaced in the deployment status. Note that progress will not be estimated during the time a deployment is paused. Defaults to 600s.
replicas	integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
revisionHistoryLimit	integer	The number of old ReplicaSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.

Property	Type	Description
selector	LabelSelector	Label selector for pods. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment. It must match the pod template's labels.
strategy	object	DeploymentStrategy describes how to replace existing pods with new ones.
template	PodTemplateSpec	Template describes the pods that will be created. The only allowed template.spec.restartPolicy value is "Always".

8.1.2. .spec.strategy

Description

DeploymentStrategy describes how to replace existing pods with new ones.

Type

object

Property	Type	Description
rollingUpdate	object	Spec to control the desired behavior of rolling update.
type	string	Type of deployment. Can be "Recreate" or "RollingUpdate". Default is RollingUpdate. Possible enum values: - "Recreate" Kill all existing pods before creating new ones. - "RollingUpdate" Replace the old ReplicaSets by new one using rolling update i.e gradually scale down the old ReplicaSets and scale up the new one.

8.1.3. .spec.strategy.rollingUpdate

Description

Spec to control the desired behavior of rolling update.

Type

object

Property	Type	Description
maxSurge	IntOrString	The maximum number of pods that can be scheduled above the desired number of pods. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 25%. Example: when this is set to 30%, the new ReplicaSet can be scaled up immediately when the rolling update starts, such that the total number of old and new pods do not exceed 130% of desired pods. Once old pods have been killed, new ReplicaSet can be scaled up further, ensuring that total number of pods running at any time during the update is at most 130% of desired pods.
maxUnavailable	IntOrString	The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 25%. Example: when this is set to 30%, the old ReplicaSet can be scaled down to 70% of desired pods immediately when the rolling update starts. Once new pods are ready, old ReplicaSet can be scaled down further, followed by scaling up the new ReplicaSet, ensuring that the total number of pods available at all times during the update is at least 70% of desired pods.

8.1.4. .status

Description

DeploymentStatus is the most recently observed status of the Deployment.

Type
object

Property	Type	Description
availableReplicas	integer	Total number of available non-terminating pods (ready for at least minReadySeconds) targeted by this deployment.
collisionCount	integer	Count of hash collisions for the Deployment. The Deployment controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ReplicaSet.
conditions	array	Represents the latest available observations of a deployment's current state.
conditions[]	object	DeploymentCondition describes the state of a deployment at a certain point.
observedGeneration	integer	The generation observed by the deployment controller.
readyReplicas	integer	Total number of non-terminating pods targeted by this Deployment with a Ready Condition.
replicas	integer	Total number of non-terminating pods targeted by this deployment (their labels match the selector).
terminatingReplicas	integer	Total number of terminating pods targeted by this deployment. Terminating pods have a non-null .metadata.deletionTimestamp and have not yet reached the Failed or Succeeded .status.phase. This is an alpha field. Enable DeploymentReplicaSetTerminatingReplicas to be able to use this field.

Property	Type	Description
unavailableReplicas	integer	Total number of unavailable pods targeted by this deployment. This is the total number of pods that are still required for the deployment to have 100% available capacity. They may either be pods that are running but not yet available or pods that still have not been created.
updatedReplicas	integer	Total number of non-terminating pods targeted by this deployment that have the desired template spec.

8.1.5. .status.conditions

Description

Represents the latest available observations of a deployment's current state.

Type

array

8.1.6. .status.conditions[]

Description

DeploymentCondition describes the state of a deployment at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	Last time the condition transitioned from one status to another.
lastUpdateTime	Time	The last time this condition was updated.

Property	Type	Description
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of deployment condition.

8.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps/v1/deployments**
 - **GET**: list or watch objects of kind Deployment
- **/apis/apps/v1/watch/deployments**
 - **GET**: watch individual changes to a list of Deployment. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/deployments**
 - **DELETE**: delete collection of Deployment
 - **GET**: list or watch objects of kind Deployment
 - **POST**: create a Deployment
- **/apis/apps/v1/watch/namespaces/{namespace}/deployments**
 - **GET**: watch individual changes to a list of Deployment. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}**
 - **DELETE**: delete a Deployment
 - **GET**: read the specified Deployment
 - **PATCH**: partially update the specified Deployment
 - **PUT**: replace the specified Deployment
- **/apis/apps/v1/watch/namespaces/{namespace}/deployments/{name}**

- **GET**: watch changes to an object of kind Deployment. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status**
 - **GET**: read status of the specified Deployment
 - **PATCH**: partially update status of the specified Deployment
 - **PUT**: replace status of the specified Deployment

8.2.1. /apis/apps/v1/deployments

HTTP method

GET

Description

list or watch objects of kind Deployment

Table 8.1. HTTP responses

HTTP code	Response body
200 - OK	DeploymentList schema
401 - Unauthorized	Empty

8.2.2. /apis/apps/v1/watch/deployments

HTTP method

GET

Description

watch individual changes to a list of Deployment. deprecated: use the 'watch' parameter with a list operation instead.

Table 8.2. HTTP responses

HTTP code	Response body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

8.2.3. /apis/apps/v1/namespaces/{namespace}/deployments

HTTP method

DELETE

Description

delete collection of Deployment

Table 8.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 8.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Deployment

Table 8.5. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Deployment

Table 8.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.7. Body parameters

Parameter	Type	Description
body	Deployment schema	

Table 8.8. HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
201 - Created	Deployment schema
202 - Accepted	Deployment schema
401 - Unauthorized	Empty

8.2.4. /apis/apps/v1/watch/namespaces/{namespace}/deployments

HTTP method

GET

Description

watch individual changes to a list of Deployment. deprecated: use the 'watch' parameter with a list operation instead.

Table 8.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

8.2.5. /apis/apps/v1/namespaces/{namespace}/deployments/{name}

Table 8.10. Global path parameters

Parameter	Type	Description
name	string	name of the Deployment

HTTP method

DELETE

Description

delete a Deployment

Table 8.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 8.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Deployment

Table 8.13. HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Deployment

Table 8.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.15. HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
201 - Created	Deployment schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Deployment

Table 8.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.17. Body parameters

Parameter	Type	Description
body	Deployment schema	

Table 8.18. HTTP responses

HTTP code	Reponse body
200 - OK	Deployment schema
201 - Created	Deployment schema
401 - Unauthorized	Empty

8.2.6. /apis/apps/v1/watch/namespaces/{namespace}/deployments/{name}

Table 8.19. Global path parameters

Parameter	Type	Description
name	string	name of the Deployment

HTTP method

GET

Description

watch changes to an object of kind Deployment. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 8.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

8.2.7. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/status

Table 8.21. Global path parameters

Parameter	Type	Description
name	string	name of the Deployment

HTTP method

GET

Description

read status of the specified Deployment

Table 8.22. HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified Deployment

Table 8.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.24. HTTP responses

HTTP code	Response body
200 - OK	Deployment schema
201 - Created	Deployment schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified Deployment

Table 8.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 8.26. Body parameters

Parameter	Type	Description
body	Deployment schema	

Table 8.27. HTTP responses

HTTP code	Reponse body
200 - OK	Deployment schema
201 - Created	Deployment schema
401 - Unauthorized	Empty

CHAPTER 9. DEPLOYMENTCONFIG [APPS.OPENSIFT.IO/V1]

Description

Deployment Configs define the template for a pod and manages deploying new images or configuration changes. A single deployment configuration is usually analogous to a single micro-service. Can support many different deployment patterns, including full restart, customizable rolling updates, and fully custom behaviors, as well as pre- and post- deployment hooks. Each individual deployment is represented as a replication controller.

A deployment is "triggered" when its configuration is changed or a tag in an Image Stream is changed. Triggers can be disabled to allow manual control over a deployment. The "strategy" determines how the deployment is carried out and may be changed at any time. The **latestVersion** field is updated when a new deployment is triggered by any means.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer). Deprecated: Use deployments or other means for declarative updates for pods instead.

Type

object

Required

- **spec**

9.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	DeploymentConfigSpec represents the desired state of the deployment.
status	object	DeploymentConfigStatus represents the current deployment state.

9.1.1. .spec

Description

DeploymentConfigSpec represents the desired state of the deployment.

Type

object

Property	Type	Description
minReadySeconds	integer	MinReadySeconds is the minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
paused	boolean	Paused indicates that the deployment config is paused resulting in no new deployments on template changes or changes in the template caused by other triggers.
replicas	integer	Replicas is the number of desired replicas.

Property	Type	Description
revisionHistoryLimit	integer	RevisionHistoryLimit is the number of old ReplicationControllers to retain to allow for rollbacks. This field is a pointer to allow for differentiation between an explicit zero and not specified. Defaults to 10. (This only applies to DeploymentConfigs created via the new group API resource, not the legacy resource.)
selector	object (string)	Selector is a label query over pods that should match the Replicas count.
strategy	object	DeploymentStrategy describes how to perform a deployment.
template	PodTemplateSpec	Template is the object that describes the pod that will be created if insufficient replicas are detected.
test	boolean	Test ensures that this deployment config will have zero replicas except while a deployment is running. This allows the deployment config to be used as a continuous deployment test - triggering on images, running the deployment, and then succeeding or failing. Post strategy hooks and After actions can be used to integrate successful deployment with an action.
triggers	array	Triggers determine how updates to a DeploymentConfig result in new deployments. If no triggers are defined, a new deployment can only occur as a result of an explicit client update to the DeploymentConfig with a new LatestVersion. If null, defaults to having a config change trigger.

Property	Type	Description
triggers[]	object	DeploymentTriggerPolicy describes a policy for a single trigger that results in a new deployment.

9.1.2. .spec.strategy

Description

DeploymentStrategy describes how to perform a deployment.

Type

object

Property	Type	Description
activeDeadlineSeconds	integer	ActiveDeadlineSeconds is the duration in seconds that the deployer pods for this deployment config may be active on a node before the system actively tries to terminate them.
annotations	object (string)	Annotations is a set of key, value pairs added to custom deployer and lifecycle pre/post hook pods.
customParams	object	CustomDeploymentStrategyParams are the input to the Custom deployment strategy.
labels	object (string)	Labels is a set of key, value pairs added to custom deployer and lifecycle pre/post hook pods.
recreateParams	object	RecreateDeploymentStrategyParams are the input to the Recreate deployment strategy.
resources	ResourceRequirements	Resources contains resource requirements to execute the deployment and any hooks.
rollingParams	object	RollingDeploymentStrategyParams are the input to the Rolling deployment strategy.
type	string	Type is the name of a deployment strategy.

9.1.3. .spec.strategy.customParams

Description

CustomDeploymentStrategyParams are the input to the Custom deployment strategy.

Type

object

Property	Type	Description
command	array (string)	Command is optional and overrides CMD in the container Image.
environment	array (EnvVar)	Environment holds the environment which will be given to the container for Image.
image	string	Image specifies a container image which can carry out a deployment.

9.1.4. .spec.strategy.recreateParams

Description

RecreateDeploymentStrategyParams are the input to the Recreate deployment strategy.

Type

object

Property	Type	Description
mid	object	LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.
post	object	LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.
pre	object	LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Property	Type	Description
timeoutSeconds	integer	TimeoutSeconds is the time to wait for updates before giving up. If the value is nil, a default will be used.

9.1.5. .spec.strategy.recreateParams.mid

Description

LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Type

object

Required

- **failurePolicy**

Property	Type	Description
execNewPod	object	ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.
failurePolicy	string	FailurePolicy specifies what action to take if the hook fails.
tagImages	array	TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.
tagImages[]	object	TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

9.1.6. .spec.strategy.recreateParams.mid.execNewPod

Description

ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.

Type

object

Required

- **command**
- **containerName**

Property	Type	Description
command	array (string)	Command is the action command and its arguments.
containerName	string	ContainerName is the name of a container in the deployment pod template whose container image will be used for the hook pod's container.
env	array (EnvVar)	Env is a set of environment variables to supply to the hook pod's container.
volumes	array (string)	Volumes is a list of named volumes from the pod template which should be copied to the hook pod. Volumes names not found in pod spec are ignored. An empty list means no volumes will be copied.

9.1.7. .spec.strategy.recreateParams.mid.tagImages

Description

TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Type

array

9.1.8. .spec.strategy.recreateParams.mid.tagImages[]

Description

TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

Type

object

Required

- **containerName**
- **to**

Property	Type	Description
containerName	string	ContainerName is the name of a container in the deployment config whose image value will be used as the source of the tag. If there is only a single container this value will be defaulted to the name of that container.
to	ObjectReference	To is the target ImageStreamTag to set the container's image onto.

9.1.9. .spec.strategy.recreateParams.post

Description

LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Type

object

Required

- **failurePolicy**

Property	Type	Description
execNewPod	object	ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.
failurePolicy	string	FailurePolicy specifies what action to take if the hook fails.
tagImages	array	TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.
tagImages[]	object	TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

9.1.10. .spec.strategy.recreateParams.post.execNewPod

Description

ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.

Type

object

Required

- **command**
- **containerName**

Property	Type	Description
command	array (string)	Command is the action command and its arguments.
containerName	string	ContainerName is the name of a container in the deployment pod template whose container image will be used for the hook pod's container.
env	array (EnvVar)	Env is a set of environment variables to supply to the hook pod's container.
volumes	array (string)	Volumes is a list of named volumes from the pod template which should be copied to the hook pod. Volumes names not found in pod spec are ignored. An empty list means no volumes will be copied.

9.1.11. .spec.strategy.recreateParams.post.tagImages

Description

TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Type

array

9.1.12. .spec.strategy.recreateParams.post.tagImages[]

Description

TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

Type

object

Required

- **containerName**
- **to**

Property	Type	Description
containerName	string	ContainerName is the name of a container in the deployment config whose image value will be used as the source of the tag. If there is only a single container this value will be defaulted to the name of that container.
to	ObjectReference	To is the target ImageStreamTag to set the container's image onto.

9.1.13. .spec.strategy.recreateParams.pre

Description

LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Type

object

Required

- **failurePolicy**

Property	Type	Description
execNewPod	object	ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.
failurePolicy	string	FailurePolicy specifies what action to take if the hook fails.
tagImages	array	TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.
tagImages[]	object	TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

9.1.14. .spec.strategy.recreateParams.pre.execNewPod

Description

ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.

Type

object

Required

- **command**
- **containerName**

Property	Type	Description
command	array (string)	Command is the action command and its arguments.
containerName	string	ContainerName is the name of a container in the deployment pod template whose container image will be used for the hook pod's container.
env	array (EnvVar)	Env is a set of environment variables to supply to the hook pod's container.
volumes	array (string)	Volumes is a list of named volumes from the pod template which should be copied to the hook pod. Volumes names not found in pod spec are ignored. An empty list means no volumes will be copied.

9.1.15. .spec.strategy.recreateParams.pre.tagImages

Description

TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Type

array

9.1.16. .spec.strategy.recreateParams.pre.tagImages[]

Description

TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

Type

object**Required**

- **containerName**
- **to**

Property	Type	Description
containerName	string	ContainerName is the name of a container in the deployment config whose image value will be used as the source of the tag. If there is only a single container this value will be defaulted to the name of that container.
to	ObjectReference	To is the target ImageStreamTag to set the container's image onto.

9.1.17. .spec.strategy.rollingParams**Description**

RollingDeploymentStrategyParams are the input to the Rolling deployment strategy.

Type**object**

Property	Type	Description
intervalSeconds	integer	IntervalSeconds is the time to wait between polling deployment status after update. If the value is nil, a default will be used.

Property	Type	Description
maxSurge	IntOrString	MaxSurge is the maximum number of pods that can be scheduled above the original number of pods. Value can be an absolute number (ex: 5) or a percentage of total pods at the start of the update (ex: 10%). Absolute number is calculated from percentage by rounding up. This cannot be 0 if MaxUnavailable is 0. By default, 25% is used. Example: when this is set to 30%, the new RC can be scaled up by 30% immediately when the rolling update starts. Once old pods have been killed, new RC can be scaled up further, ensuring that total number of pods running at any time during the update is atmost 130% of original pods.
maxUnavailable	IntOrString	MaxUnavailable is the maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of total pods at the start of update (ex: 10%). Absolute number is calculated from percentage by rounding down. This cannot be 0 if MaxSurge is 0. By default, 25% is used. Example: when this is set to 30%, the old RC can be scaled down by 30% immediately when the rolling update starts. Once new pods are ready, old RC can be scaled down further, followed by scaling up the new RC, ensuring that at least 70% of original number of pods are available at all times during the update.
post	object	LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Property	Type	Description
pre	object	LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.
timeoutSeconds	integer	TimeoutSeconds is the time to wait for updates before giving up. If the value is nil, a default will be used.
updatePeriodSeconds	integer	UpdatePeriodSeconds is the time to wait between individual pod updates. If the value is nil, a default will be used.

9.1.18. .spec.strategy.rollingParams.post

Description

LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Type

object

Required

- **failurePolicy**

Property	Type	Description
execNewPod	object	ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.
failurePolicy	string	FailurePolicy specifies what action to take if the hook fails.
tagImages	array	TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.
tagImages[]	object	TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

9.1.19. .spec.strategy.rollingParams.post.execNewPod

Description

ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.

Type

object

Required

- **command**
- **containerName**

Property	Type	Description
command	array (string)	Command is the action command and its arguments.
containerName	string	ContainerName is the name of a container in the deployment pod template whose container image will be used for the hook pod's container.
env	array (EnvVar)	Env is a set of environment variables to supply to the hook pod's container.
volumes	array (string)	Volumes is a list of named volumes from the pod template which should be copied to the hook pod. Volumes names not found in pod spec are ignored. An empty list means no volumes will be copied.

9.1.20. .spec.strategy.rollingParams.post.tagImages

Description

TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Type

array

9.1.21. .spec.strategy.rollingParams.post.tagImages[]

Description

TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

Type

object

Required

- **containerName**
- **to**

Property	Type	Description
containerName	string	ContainerName is the name of a container in the deployment config whose image value will be used as the source of the tag. If there is only a single container this value will be defaulted to the name of that container.
to	ObjectReference	To is the target ImageStreamTag to set the container's image onto.

9.1.22. .spec.strategy.rollingParams.pre

Description

LifecycleHook defines a specific deployment lifecycle action. Only one type of action may be specified at any time.

Type

object

Required

- **failurePolicy**

Property	Type	Description
execNewPod	object	ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.
failurePolicy	string	FailurePolicy specifies what action to take if the hook fails.
tagImages	array	TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Property	Type	Description
tagImages[]	object	TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

9.1.23. .spec.strategy.rollingParams.pre.execNewPod

Description

ExecNewPodHook is a hook implementation which runs a command in a new pod based on the specified container which is assumed to be part of the deployment template.

Type

object

Required

- **command**
- **containerName**

Property	Type	Description
command	array (string)	Command is the action command and its arguments.
containerName	string	ContainerName is the name of a container in the deployment pod template whose container image will be used for the hook pod's container.
env	array (EnvVar)	Env is a set of environment variables to supply to the hook pod's container.
volumes	array (string)	Volumes is a list of named volumes from the pod template which should be copied to the hook pod. Volumes names not found in pod spec are ignored. An empty list means no volumes will be copied.

9.1.24. .spec.strategy.rollingParams.pre.tagImages

Description

TagImages instructs the deployer to tag the current image referenced under a container onto an image stream tag.

Type

array

9.1.25. .spec.strategy.rollingParams.pre.tagImages[]

Description

TagImageHook is a request to tag the image in a particular container onto an ImageStreamTag.

Type

object

Required

- **containerName**
- **to**

Property	Type	Description
containerName	string	ContainerName is the name of a container in the deployment config whose image value will be used as the source of the tag. If there is only a single container this value will be defaulted to the name of that container.
to	ObjectReference	To is the target ImageStreamTag to set the container's image onto.

9.1.26. .spec.triggers

Description

Triggers determine how updates to a DeploymentConfig result in new deployments. If no triggers are defined, a new deployment can only occur as a result of an explicit client update to the DeploymentConfig with a new LatestVersion. If null, defaults to having a config change trigger.

Type

array

9.1.27. .spec.triggers[]

Description

DeploymentTriggerPolicy describes a policy for a single trigger that results in a new deployment.

Type

object

Property	Type	Description
imageChangeParams	object	DeploymentTriggerImageChangeParams represents the parameters to the ImageChange trigger.
type	string	Type of the trigger

9.1.28. .spec.triggers[].imageChangeParams

Description

DeploymentTriggerImageChangeParams represents the parameters to the ImageChange trigger.

Type

object

Required

- **from**

Property	Type	Description
automatic	boolean	Automatic means that the detection of a new tag value should result in an image update inside the pod template.
containerNames	array (string)	ContainerNames is used to restrict tag updates to the specified set of container names in a pod. If multiple triggers point to the same containers, the resulting behavior is undefined. Future API versions will make this a validation error. If ContainerNames does not point to a valid container, the trigger will be ignored. Future API versions will make this a validation error.
from	ObjectReference	From is a reference to an image stream tag to watch for changes. From.Name is the only required subfield - if From.Namespace is blank, the namespace of the current deployment trigger will be used.

Property	Type	Description
lastTriggeredImage	string	LastTriggeredImage is the last image to be triggered.

9.1.29. .status

Description

DeploymentConfigStatus represents the current deployment state.

Type

object

Required

- **latestVersion**
- **observedGeneration**
- **replicas**
- **updatedReplicas**
- **availableReplicas**
- **unavailableReplicas**

Property	Type	Description
availableReplicas	integer	AvailableReplicas is the total number of available pods targeted by this deployment config.
conditions	array	Conditions represents the latest available observations of a deployment config's current state.
conditions[]	object	DeploymentCondition describes the state of a deployment config at a certain point.
details	object	DeploymentDetails captures information about the causes of a deployment.
latestVersion	integer	LatestVersion is used to determine whether the current deployment associated with a deployment config is out of sync.

Property	Type	Description
observedGeneration	integer	ObservedGeneration is the most recent generation observed by the deployment config controller.
readyReplicas	integer	Total number of ready pods targeted by this deployment.
replicas	integer	Replicas is the total number of pods targeted by this deployment config.
unavailableReplicas	integer	UnavailableReplicas is the total number of unavailable pods targeted by this deployment config.
updatedReplicas	integer	UpdatedReplicas is the total number of non-terminated pods targeted by this deployment config that have the desired template spec.

9.1.30. .status.conditions

Description

Conditions represents the latest available observations of a deployment config's current state.

Type

array

9.1.31. .status.conditions[]

Description

DeploymentCondition describes the state of a deployment config at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description

Property	Type	Description
lastTransitionTime	Time	The last time the condition transitioned from one status to another.
lastUpdateTime	Time	The last time this condition was updated.
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of deployment condition.

9.1.32. .status.details

Description

DeploymentDetails captures information about the causes of a deployment.

Type

object

Required

- **causes**

Property	Type	Description
causes	array	Causes are extended data associated with all the causes for creating a new deployment
causes[]	object	DeploymentCause captures information about a particular cause of a deployment.
message	string	Message is the user specified change message, if this deployment was triggered manually by the user

9.1.33. `.status.details.causes`

Description

Causes are extended data associated with all the causes for creating a new deployment

Type

array

9.1.34. `.status.details.causes[]`

Description

DeploymentCause captures information about a particular cause of a deployment.

Type

object

Required

- **type**

Property	Type	Description
imageTrigger	object	DeploymentCauseImageTrigger represents details about the cause of a deployment originating from an image change trigger
type	string	Type of the trigger that resulted in the creation of a new deployment

9.1.35. `.status.details.causes[].imageTrigger`

Description

DeploymentCauseImageTrigger represents details about the cause of a deployment originating from an image change trigger

Type

object

Required

- **from**

Property	Type	Description

Property	Type	Description
from	ObjectReference	From is a reference to the changed object which triggered a deployment. The field may have the kinds DockerImage, ImageStreamTag, or ImageStreamImage.

9.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps.openshift.io/v1/deploymentconfigs**
 - **GET**: list or watch objects of kind DeploymentConfig
- **/apis/apps.openshift.io/v1/watch/deploymentconfigs**
 - **GET**: watch individual changes to a list of DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs**
 - **DELETE**: delete collection of DeploymentConfig
 - **GET**: list or watch objects of kind DeploymentConfig
 - **POST**: create a DeploymentConfig
- **/apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs**
 - **GET**: watch individual changes to a list of DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}**
 - **DELETE**: delete a DeploymentConfig
 - **GET**: read the specified DeploymentConfig
 - **PATCH**: partially update the specified DeploymentConfig
 - **PUT**: replace the specified DeploymentConfig
- **/apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs/{name}**
 - **GET**: watch changes to an object of kind DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/status**
 - **GET**: read status of the specified DeploymentConfig

- **PATCH**: partially update status of the specified DeploymentConfig
- **PUT**: replace status of the specified DeploymentConfig

9.2.1. /apis/apps.openshift.io/v1/deploymentconfigs

HTTP method

GET

Description

list or watch objects of kind DeploymentConfig

Table 9.1. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfigList schema
401 - Unauthorized	Empty

9.2.2. /apis/apps.openshift.io/v1/watch/deploymentconfigs

HTTP method

GET

Description

watch individual changes to a list of DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead.

Table 9.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

9.2.3. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs

HTTP method

DELETE

Description

delete collection of DeploymentConfig

Table 9.3. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 9.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v2 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind DeploymentConfig

Table 9.5. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfigList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a DeploymentConfig

Table 9.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.7. Body parameters

Parameter	Type	Description
body	DeploymentConfig schema	

Table 9.8. HTTP responses

HTTP code	Response body
200 - OK	DeploymentConfig schema
201 - Created	DeploymentConfig schema
202 - Accepted	DeploymentConfig schema
401 - Unauthorized	Empty

9.2.4. /apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs

HTTP method

GET

Description

watch individual changes to a list of DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead.

Table 9.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

9.2.5. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}

Table 9.10. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentConfig

HTTP method

DELETE

Description

delete a DeploymentConfig

Table 9.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 9.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v2 schema
202 - Accepted	Status_v2 schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified DeploymentConfig

Table 9.13. HTTP responses

HTTP code	Response body
200 - OK	DeploymentConfig schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified DeploymentConfig

Table 9.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.15. HTTP responses

HTTP code	Response body
200 - OK	DeploymentConfig schema

HTTP code	Reponse body
201 - Created	DeploymentConfig schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified DeploymentConfig

Table 9.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.17. Body parameters

Parameter	Type	Description
body	DeploymentConfig schema	

Table 9.18. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfig schema
201 - Created	DeploymentConfig schema
401 - Unauthorized	Empty

9.2.6. /apis/apps.openshift.io/v1/watch/namespaces/{namespace}/deploymentconfigs/{name}

Table 9.19. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentConfig

HTTP method

GET

Description

watch changes to an object of kind DeploymentConfig. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 9.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

9.2.7. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}

Table 9.21. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentConfig

HTTP method

GET

Description

read status of the specified DeploymentConfig

Table 9.22. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfig schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified DeploymentConfig

Table 9.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.24. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfig schema
201 - Created	DeploymentConfig schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified DeploymentConfig

Table 9.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 9.26. Body parameters

Parameter	Type	Description
body	DeploymentConfig schema	

Table 9.27. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfig schema
201 - Created	DeploymentConfig schema
401 - Unauthorized	Empty

CHAPTER 10. DEPLOYMENTCONFIGROLLBACK [APPS.OPENSIFT.IO/V1]

Description

DeploymentConfigRollback provides the input to rollback generation.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **name**
- **spec**

10.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	Name of the deployment config that will be rolled back.
spec	object	DeploymentConfigRollbackSpec represents the options for rollback generation.

Property	Type	Description
updatedAnnotations	object (string)	UpdatedAnnotations is a set of new annotations that will be added in the deployment config.

10.1.1. .spec

Description

DeploymentConfigRollbackSpec represents the options for rollback generation.

Type

object

Required

- **from**
- **includeTriggers**
- **includeTemplate**
- **includeReplicationMeta**
- **includeStrategy**

Property	Type	Description
from	ObjectReference	From points to a ReplicationController which is a deployment.
includeReplicationMeta	boolean	IncludeReplicationMeta specifies whether to include the replica count and selector.
includeStrategy	boolean	IncludeStrategy specifies whether to include the deployment Strategy.
includeTemplate	boolean	IncludeTemplate specifies whether to include the PodTemplateSpec.
includeTriggers	boolean	IncludeTriggers specifies whether to include config Triggers.
revision	integer	Revision to rollback to. If set to 0, rollback to the last revision.

10.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/rollback**
 - **POST**: create rollback of a DeploymentConfig

10.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/rollback

Table 10.1. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentConfigRollback

Table 10.2. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method

POST

Description

create rollback of a DeploymentConfig

Table 10.3. Body parameters

Parameter	Type	Description
body	DeploymentConfigRollback schema	

Table 10.4. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentConfigRollback schema
201 - Created	DeploymentConfigRollback schema
202 - Accepted	DeploymentConfigRollback schema
401 - Unauthorized	Empty

CHAPTER 11. DEPLOYMENTLOG [APPS.OPENSIFT.IO/V1]

Description

DeploymentLog represents the logs for a deployment

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

11.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

11.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/log**
 - **GET**: read log of the specified DeploymentConfig

11.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/log

Table 11.1. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentLog

HTTP method

GET

Description

read log of the specified DeploymentConfig

Table 11.2. HTTP responses

HTTP code	Reponse body
200 - OK	DeploymentLog schema
401 - Unauthorized	Empty

CHAPTER 12. DEPLOYMENTREQUEST

[APPS.OPENSIFT.IO/V1]

Description

DeploymentRequest is a request to a deployment config for a new deployment.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **name**
- **latest**
- **force**

12.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
excludeTriggers	array (string)	ExcludeTriggers instructs the instantiator to avoid processing the specified triggers. This field overrides the triggers from latest and allows clients to control specific logic. This field is ignored if not specified.
force	boolean	Force will try to force a new deployment to run. If the deployment config is paused, then setting this to true will return an Invalid error.

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
latest	boolean	Latest will update the deployment config with the latest state from all triggers.
name	string	Name of the deployment config for requesting a new deployment.

12.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}/instantia**
te
 - **POST**: create instantiate of a DeploymentConfig

12.2.1. /apis/apps.openshift.io/v1/namespaces/{namespace}/deploymentconfigs/{name}

Table 12.1. Global path parameters

Parameter	Type	Description
name	string	name of the DeploymentRequest

Table 12.2. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method**POST****Description**

create instantiate of a DeploymentConfig

Table 12.3. Body parameters

Parameter	Type	Description
body	DeploymentRequest schema	

Table 12.4. HTTP responses

HTTP code	Response body
200 - OK	DeploymentRequest schema
201 - Created	DeploymentRequest schema
202 - Accepted	DeploymentRequest schema
401 - Unauthorized	Empty

CHAPTER 13. JOB [BATCH/V1]

Description

Job represents the configuration of a single job.

Type

object

13.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	JobSpec describes how the job execution will look like.
status	object	JobStatus represents the current state of a Job.

13.1.1. .spec

Description

JobSpec describes how the job execution will look like.

Type

object

Required

- **template**

Property	Type	Description
activeDeadlineSeconds	integer	Specifies the duration in seconds relative to the startTime that the job may be continuously active before the system tries to terminate it; value must be positive integer. If a Job is suspended (at creation or through an update), this timer will effectively be stopped and reset when the Job is resumed again.
backoffLimit	integer	Specifies the number of retries before marking this job failed. Defaults to 6
backoffLimitPerIndex	integer	Specifies the limit for the number of retries within an index before marking this index as failed. When enabled the number of failures per index is kept in the pod's batch.kubernetes.io/job-index-failure-count annotation. It can only be set when Job's completionMode=Indexed, and the Pod's restart policy is Never. The field is immutable.
completionMode	string	completionMode specifies how Pod completions are tracked. It can be NonIndexed (default) or Indexed. NonIndexed means that the Job is considered complete when there have been .spec.completions successfully completed Pods. Each Pod completion is homologous to each other. Indexed means that the Pods of a Job get an associated completion index from 0 to

Property	Type	Description
		(.spec.completions - 1), available in the annotation <code>batch.kubernetes.io/job-completion-index</code>. The Job is considered complete when there is one successfully completed Pod for each index. When value is Indexed, <code>.spec.completions</code> must be specified and <code>.spec.parallelism</code> must be less than or equal to 10^5. In addition, The Pod name takes the form <code>\$(job-name)-\$(index)-\$(random-string)</code>, the Pod hostname takes the form <code>\$(job-name)-\$(index)</code>. More completion modes can be added in the future. If the Job controller observes a mode that it doesn't recognize, which is possible during upgrades due to version skew, the controller skips updates for the Job. Possible enum values: - "Indexed" is a Job completion mode. In this mode, the Pods of a Job get an associated completion index from 0 to <code>(.spec.completions - 1)</code>. The Job is considered complete when a Pod completes for each completion index. - "NonIndexed" is a Job completion mode. In this mode, the Job is considered complete when there have been <code>.spec.completions</code> successfully completed Pods. Pod completions are homologous to each other.
completions	integer	Specifies the desired number of successfully finished pods the job should be run with. Setting to null means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

Property	Type	Description
managedBy	string	ManagedBy field indicates the controller that manages a Job. The k8s Job controller reconciles jobs which don't have this field at all or the field value is the reserved string kubernetes.io/job-controller, but skips reconciling Jobs with a custom value for this field. The value must be a valid domain-prefixed path (e.g. acme.io/foo) - all characters before the first "/" must be a valid subdomain as defined by RFC 1123. All characters trailing the first "/" must be valid HTTP Path characters as defined by RFC 3986. The value cannot exceed 63 characters. This field is immutable. This field is beta-level. The job controller accepts setting the field when the feature gate JobManagedBy is enabled (enabled by default).
manualSelector	boolean	manualSelector controls generation of pod labels and pod selectors. Leave manualSelector unset unless you are certain what you are doing. When false or unset, the system pick labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see manualSelector=true in jobs that were created with the old extensions/v1beta1 API. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector

Property	Type	Description
maxFailedIndexes	integer	Specifies the maximal number of failed indexes before marking the Job as failed, when <code>backoffLimitPerIndex</code> is set. Once the number of failed indexes exceeds this number the entire Job is marked as Failed and its execution is terminated. When left as null the job continues execution of all of its indexes and is marked with the Complete Job condition. It can only be specified when <code>backoffLimitPerIndex</code> is set. It can be null or up to completions. It is required and must be less than or equal to 10^4 when is completions greater than 10^5 .
parallelism	integer	Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when $((\text{spec.completions} - \text{.status.successful}) < \text{.spec.parallelism})$, i.e. when the work left to do is less than max parallelism. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
podFailurePolicy	object	PodFailurePolicy describes how failed pods influence the <code>backoffLimit</code> .

Property	Type	Description
podReplacementPolicy	string	podReplacementPolicy specifies when to create replacement Pods. Possible values are: - TerminatingOrFailed means that we recreate pods when they are terminating (has a metadata.deletionTimestamp) or failed. - Failed means to wait until a previously created Pod is fully terminated (has phase Failed or Succeeded) before creating a replacement Pod. When using podFailurePolicy, Failed is the the only allowed value. TerminatingOrFailed and Failed are allowed values when podFailurePolicy is not in use. This is an beta field. To use this, enable the JobPodReplacementPolicy feature toggle. This is on by default. Possible enum values: - "Failed" means to wait until a previously created Pod is fully terminated (has phase Failed or Succeeded) before creating a replacement Pod. - "TerminatingOrFailed" means that we recreate pods when they are terminating (has a metadata.deletionTimestamp) or failed.
selector	LabelSelector	A label query over pods that should match the pod count. Normally, the system sets this field for you. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
successPolicy	object	SuccessPolicy describes when a Job can be declared as succeeded based on the success of some indexes.

Property	Type	Description
suspend	boolean	suspend specifies whether the Job controller should create Pods or not. If a Job is created with suspend set to true, no Pods are created by the Job controller. If a Job is suspended after creation (i.e. the flag goes from false to true), the Job controller will delete all active Pods associated with this Job. Users must design their workload to gracefully handle this. Suspending a Job will reset the StartTime field of the Job, effectively resetting the ActiveDeadlineSeconds timer too. Defaults to false.
template	PodTemplateSpec	Describes the pod that will be created when executing a job. The only allowed template.spec.restartPolicy values are "Never" or "OnFailure". More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
ttlSecondsAfterFinished	integer	ttlSecondsAfterFinished limits the lifetime of a Job that has finished execution (either Complete or Failed). If this field is set, ttlSecondsAfterFinished after the Job finishes, it is eligible to be automatically deleted. When the Job is being deleted, its lifecycle guarantees (e.g. finalizers) will be honored. If this field is unset, the Job won't be automatically deleted. If this field is set to zero, the Job becomes eligible to be deleted immediately after it finishes.

13.1.2. .spec.podFailurePolicy

Description

PodFailurePolicy describes how failed pods influence the backoffLimit.

Type

object

Required

- **rules**

Property	Type	Description
rules	array	A list of pod failure policy rules. The rules are evaluated in order. Once a rule matches a Pod failure, the remaining of the rules are ignored. When no rule matches the Pod failure, the default handling applies - the counter of pod failures is incremented and it is checked against the backoffLimit. At most 20 elements are allowed.
rules[]	object	PodFailurePolicyRule describes how a pod failure is handled when the requirements are met. One of onExitCodes and onPodConditions, but not both, can be used in each rule.

13.1.3. .spec.podFailurePolicy.rules

Description

A list of pod failure policy rules. The rules are evaluated in order. Once a rule matches a Pod failure, the remaining of the rules are ignored. When no rule matches the Pod failure, the default handling applies - the counter of pod failures is incremented and it is checked against the backoffLimit. At most 20 elements are allowed.

Type

array

13.1.4. .spec.podFailurePolicy.rules[]

Description

PodFailurePolicyRule describes how a pod failure is handled when the requirements are met. One of onExitCodes and onPodConditions, but not both, can be used in each rule.

Type

object

Required

- **action**

Property	Type	Description

Property	Type	Description
action	string	Specifies the action taken on a pod failure when the requirements are satisfied. Possible values are: - FailJob: indicates that the pod's job is marked as Failed and all running pods are terminated. - FailIndex: indicates that the pod's index is marked as Failed and will not be restarted. - Ignore: indicates that the counter towards the .backoffLimit is not incremented and a replacement pod is created. - Count: indicates that the pod is handled in the default way - the counter towards the .backoffLimit is incremented. Additional values are considered to be added in the future. Clients should react to an unknown action by skipping the rule. Possible enum values: - "Count" This is an action which might be taken on a pod failure - the pod failure is handled in the default way - the counter towards .backoffLimit, represented by the job's .status.failed field, is incremented. - "FailIndex" This is an action which might be taken on a pod failure - mark the Job's index as failed to avoid restarts within this index. This action can only be used when backoffLimitPerIndex is set. - "FailJob" This is an action which might be taken on a pod failure - mark the pod's job as Failed and terminate all running pods. - "Ignore" This is an action which might be taken on a pod failure - the counter towards .backoffLimit, represented by the job's .status.failed field, is not incremented and a replacement pod is created.

Property	Type	Description
onExitCodes	object	PodFailurePolicyOnExitCodesRequirement describes the requirement for handling a failed pod based on its container exit codes. In particular, it lookups the <code>.state.terminated.exitCode</code> for each app container and init container status, represented by the <code>.status.containerStatuses</code> and <code>.status.initContainerStatuses</code> fields in the Pod status, respectively. Containers completed with success (exit code 0) are excluded from the requirement check.
onPodConditions	array	Represents the requirement on the pod conditions. The requirement is represented as a list of pod condition patterns. The requirement is satisfied if at least one pattern matches an actual pod condition. At most 20 elements are allowed.
onPodConditions[]	object	PodFailurePolicyOnPodConditionPattern describes a pattern for matching an actual pod condition type.

13.1.5. `.spec.podFailurePolicy.rules[].onExitCodes`

Description

PodFailurePolicyOnExitCodesRequirement describes the requirement for handling a failed pod based on its container exit codes. In particular, it lookups the `.state.terminated.exitCode` for each app container and init container status, represented by the `.status.containerStatuses` and `.status.initContainerStatuses` fields in the Pod status, respectively. Containers completed with success (exit code 0) are excluded from the requirement check.

Type

object

Required

- **operator**
- **values**

Property	Type	Description
containerName	string	Restricts the check for exit codes to the container with the specified name. When null, the rule applies to all containers. When specified, it should match one the container or initContainer names in the pod template.
operator	string	Represents the relationship between the container exit code(s) and the specified values. Containers completed with success (exit code 0) are excluded from the requirement check. Possible values are: - In: the requirement is satisfied if at least one container exit code (might be multiple if there are multiple containers not restricted by the 'containerName' field) is in the set of specified values. - NotIn: the requirement is satisfied if at least one container exit code (might be multiple if there are multiple containers not restricted by the 'containerName' field) is not in the set of specified values. Additional values are considered to be added in the future. Clients should react to an unknown operator by assuming the requirement is not satisfied. Possible enum values: - "In" - "NotIn"
values	array (integer)	Specifies the set of values. Each returned container exit code (might be multiple in case of multiple containers) is checked against this set of values with respect to the operator. The list of values must be ordered and must not contain duplicates. Value '0' cannot be used for the In operator. At least one element is required. At most 255 elements are allowed.

13.1.6. .spec.podFailurePolicy.rules[].onPodConditions

Description

Represents the requirement on the pod conditions. The requirement is represented as a list of pod condition patterns. The requirement is satisfied if at least one pattern matches an actual pod condition. At most 20 elements are allowed.

Type

array

13.1.7. `.spec.podFailurePolicy.rules[].onPodConditions[]`

Description

PodFailurePolicyOnPodConditionsPattern describes a pattern for matching an actual pod condition type.

Type

object

Required

- **type**
- **status**

Property	Type	Description
status	string	Specifies the required Pod condition status. To match a pod condition it is required that the specified status equals the pod condition status. Defaults to True.
type	string	Specifies the required Pod condition type. To match a pod condition it is required that specified type equals the pod condition type.

13.1.8. `.spec.successPolicy`

Description

SuccessPolicy describes when a Job can be declared as succeeded based on the success of some indexes.

Type

object

Required

- **rules**

Property	Type	Description
rules	array	rules represents the list of alternative rules for the declaring the Jobs as successful before .status.succeeded >= .spec.completions . Once any of the rules are met, the "SucceededCriteriaMet" condition is added, and the lingering pods are removed. The terminal state for such a Job has the "Complete" condition. Additionally, these rules are evaluated in order; Once the Job meets one of the rules, other rules are ignored. At most 20 elements are allowed.
rules[]	object	SuccessPolicyRule describes rule for declaring a Job as succeeded. Each rule must have at least one of the "succeededIndexes" or "succeededCount" specified.

13.1.9. .spec.successPolicy.rules

Description

rules represents the list of alternative rules for the declaring the Jobs as successful before **.status.succeeded >= .spec.completions**. Once any of the rules are met, the "SucceededCriteriaMet" condition is added, and the lingering pods are removed. The terminal state for such a Job has the "Complete" condition. Additionally, these rules are evaluated in order; Once the Job meets one of the rules, other rules are ignored. At most 20 elements are allowed.

Type

array

13.1.10. .spec.successPolicy.rules[]

Description

SuccessPolicyRule describes rule for declaring a Job as succeeded. Each rule must have at least one of the "succeededIndexes" or "succeededCount" specified.

Type

object

Property	Type	Description

Property	Type	Description
succeededCount	integer	succeededCount specifies the minimal required size of the actual set of the succeeded indexes for the Job. When succeededCount is used along with succeededIndexes, the check is constrained only to the set of indexes specified by succeededIndexes. For example, given that succeededIndexes is "1-4", succeededCount is "3", and completed indexes are "1", "3", and "5", the Job isn't declared as succeeded because only "1" and "3" indexes are considered in that rules. When this field is null, this doesn't default to any value and is never evaluated at any time. When specified it needs to be a positive integer.
succeededIndexes	string	succeededIndexes specifies the set of indexes which need to be contained in the actual set of the succeeded indexes for the Job. The list of indexes must be within 0 to ".spec.completions-1" and must not contain duplicates. At least one element is required. The indexes are represented as intervals separated by commas. The intervals can be a decimal integer or a pair of decimal integers separated by a hyphen. The number are listed in represented by the first and last element of the series, separated by a hyphen. For example, if the completed indexes are 1, 3, 4, 5 and 7, they are represented as "1,3-5,7". When this field is null, this field doesn't default to any value and is never evaluated at any time.

13.1.11. .status

Description

JobStatus represents the current state of a Job.

Type

object

Property	Type	Description
active	integer	The number of pending and running pods which are not terminating (without a deletionTimestamp). The value is zero for finished jobs.
completedIndexes	string	completedIndexes holds the completed indexes when .spec.completionMode = "Indexed" in a text format. The indexes are represented as decimal integers separated by commas. The numbers are listed in increasing order. Three or more consecutive numbers are compressed and represented by the first and last element of the series, separated by a hyphen. For example, if the completed indexes are 1, 3, 4, 5 and 7, they are represented as "1,3-5,7".
completionTime	Time	Represents time when the job was completed. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC. The completion time is set when the job finishes successfully, and only then. The value cannot be updated or removed. The value indicates the same or later point in time as the startTime field.

Property	Type	Description
conditions	array	The latest available observations of an object's current state. When a Job fails, one of the conditions will have type "Failed" and status true. When a Job is suspended, one of the conditions will have type "Suspended" and status true; when the Job is resumed, the status of this condition will become false. When a Job is completed, one of the conditions will have type "Complete" and status true. A job is considered finished when it is in a terminal condition, either "Complete" or "Failed". A Job cannot have both the "Complete" and "Failed" conditions. Additionally, it cannot be in the "Complete" and "FailureTarget" conditions. The "Complete", "Failed" and "FailureTarget" conditions cannot be disabled. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
conditions[]	object	JobCondition describes current state of a job.
failed	integer	The number of pods which reached phase Failed. The value increases monotonically.

Property	Type	Description
failedIndexes	string	FailedIndexes holds the failed indexes when spec.backoffLimitPerIndex is set. The indexes are represented in the text format analogous as for the completedIndexes field, ie. they are kept as decimal integers separated by commas. The numbers are listed in increasing order. Three or more consecutive numbers are compressed and represented by the first and last element of the series, separated by a hyphen. For example, if the failed indexes are 1, 3, 4, 5 and 7, they are represented as "1,3-5,7". The set of failed indexes cannot overlap with the set of completed indexes.
ready	integer	The number of active pods which have a Ready condition and are not terminating (without a deletionTimestamp).
startTime	Time	Represents time when the job controller started processing a job. When a Job is created in the suspended state, this field is not set until the first time it is resumed. This field is reset every time a Job is resumed from suspension. It is represented in RFC3339 form and is in UTC. Once set, the field can only be removed when the job is suspended. The field cannot be modified while the job is unsuspended or finished.
succeeded	integer	The number of pods which reached phase Succeeded. The value increases monotonically for a given spec. However, it may decrease in reaction to scale down of elastic indexed jobs.

Property	Type	Description
terminating	integer	The number of pods which are terminating (in phase Pending or Running and have a deletionTimestamp). This field is beta-level. The job controller populates the field when the feature gate JobPodReplacementPolicy is enabled (enabled by default).
uncountedTerminatedPods	object	UncountedTerminatedPods holds UIDs of Pods that have terminated but haven't been accounted in Job status counters.

13.1.12. .status.conditions

Description

The latest available observations of an object's current state. When a Job fails, one of the conditions will have type "Failed" and status true. When a Job is suspended, one of the conditions will have type "Suspended" and status true; when the Job is resumed, the status of this condition will become false. When a Job is completed, one of the conditions will have type "Complete" and status true. A job is considered finished when it is in a terminal condition, either "Complete" or "Failed". A Job cannot have both the "Complete" and "Failed" conditions. Additionally, it cannot be in the "Complete" and "FailureTarget" conditions. The "Complete", "Failed" and "FailureTarget" conditions cannot be disabled.

More info: <https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/>

Type

array

13.1.13. .status.conditions[]

Description

JobCondition describes current state of a job.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastProbeTime	Time	Last time the condition was checked.
lastTransitionTime	Time	Last time the condition transit from one status to another.
message	string	Human readable message indicating details about last transition.
reason	string	(brief) reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of job condition, Complete or Failed.

13.1.14. .status.uncountedTerminatedPods

Description

UncountedTerminatedPods holds UIDs of Pods that have terminated but haven't been accounted in Job status counters.

Type

object

Property	Type	Description
failed	array (string)	failed holds UIDs of failed Pods.
succeeded	array (string)	succeeded holds UIDs of succeeded Pods.

13.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/batch/v1/jobs**
 - **GET**: list or watch objects of kind Job
- **/apis/batch/v1/watch/jobs**
 - **GET**: watch individual changes to a list of Job. deprecated: use the 'watch' parameter with a list operation instead.

- **/apis/batch/v1/namespaces/{namespace}/jobs**
 - **DELETE**: delete collection of Job
 - **GET**: list or watch objects of kind Job
 - **POST**: create a Job
- **/apis/batch/v1/watch/namespaces/{namespace}/jobs**
 - **GET**: watch individual changes to a list of Job. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/batch/v1/namespaces/{namespace}/jobs/{name}**
 - **DELETE**: delete a Job
 - **GET**: read the specified Job
 - **PATCH**: partially update the specified Job
 - **PUT**: replace the specified Job
- **/apis/batch/v1/watch/namespaces/{namespace}/jobs/{name}**
 - **GET**: watch changes to an object of kind Job. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/batch/v1/namespaces/{namespace}/jobs/{name}/status**
 - **GET**: read status of the specified Job
 - **PATCH**: partially update status of the specified Job
 - **PUT**: replace status of the specified Job

13.2.1. /apis/batch/v1/jobs

HTTP method

GET

Description

list or watch objects of kind Job

Table 13.1. HTTP responses

HTTP code	Reponse body
200 - OK	JobList schema
401 - Unauthorized	Empty

13.2.2. /apis/batch/v1/watch/jobs

HTTP method**GET****Description**

watch individual changes to a list of Job. deprecated: use the 'watch' parameter with a list operation instead.

Table 13.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

13.2.3. /apis/batch/v1/namespaces/{namespace}/jobs**HTTP method****DELETE****Description**

delete collection of Job

Table 13.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 13.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Job

Table 13.5. HTTP responses

HTTP code	Response body
200 - OK	JobList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Job

Table 13.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 13.7. Body parameters

Parameter	Type	Description
body	Job schema	

Table 13.8. HTTP responses

HTTP code	Reponse body
200 - OK	Job schema
201 - Created	Job schema
202 - Accepted	Job schema
401 - Unauthorized	Empty

13.2.4. /apis/batch/v1/watch/namespaces/{namespace}/jobs

HTTP method

GET

Description

watch individual changes to a list of Job. deprecated: use the 'watch' parameter with a list operation instead.

Table 13.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

13.2.5. /apis/batch/v1/namespaces/{namespace}/jobs/{name}

Table 13.10. Global path parameters

Parameter	Type	Description
name	string	name of the Job

HTTP method

DELETE

Description

delete a Job

Table 13.11. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 13.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified Job

Table 13.13. HTTP responses

HTTP code	Reponse body
200 - OK	Job schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified Job

Table 13.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 13.15. HTTP responses

HTTP code	Response body
200 - OK	Job schema
201 - Created	Job schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Job

Table 13.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 13.17. Body parameters

Parameter	Type	Description
body	Job schema	

Table 13.18. HTTP responses

HTTP code	Response body
200 - OK	Job schema
201 - Created	Job schema
401 - Unauthorized	Empty

13.2.6. /apis/batch/v1/watch/namespaces/{namespace}/jobs/{name}

Table 13.19. Global path parameters

Parameter	Type	Description
name	string	name of the Job

HTTP method

GET

Description

watch changes to an object of kind Job. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 13.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

13.2.7. /apis/batch/v1/namespaces/{namespace}/jobs/{name}/status**Table 13.21. Global path parameters**

Parameter	Type	Description
name	string	name of the Job

HTTP method**GET****Description**

read status of the specified Job

Table 13.22. HTTP responses

HTTP code	Reponse body
200 - OK	Job schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified Job

Table 13.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 13.24. HTTP responses

HTTP code	Response body
200 - OK	Job schema
201 - Created	Job schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified Job

Table 13.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 13.26. Body parameters

Parameter	Type	Description
body	Job schema	

Table 13.27. HTTP responses

HTTP code	Response body
200 - OK	Job schema
201 - Created	Job schema
401 - Unauthorized	Empty

CHAPTER 14. POD [V1]

Description

Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.

Type

object

14.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec is a description of a pod.
status	object	PodStatus represents information about the status of a pod. Status may trail the actual state of a system, especially if the node that hosts the pod cannot contact the control plane.

14.1.1. .spec

Description

PodSpec is a description of a pod.

Type

object

Required

- **containers**

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
affinity	object	Affinity is a group of affinity scheduling rules.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.
containers	array	List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.
containers[]	object	A single application container that you want to run within a pod.
dnsConfig	object	PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Property	Type	Description
dnsPolicy	string	Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'. Possible enum values: - "ClusterFirst" indicates that the pod should use cluster DNS first unless hostNetwork is true, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "ClusterFirstWithHostNet" indicates that the pod should use cluster DNS first, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "Default" indicates that the pod should use the default (as determined by kubelet) DNS settings. - "None" indicates that the pod should use empty DNS settings. DNS parameters such as nameservers and search paths should be defined via DNSConfig.
enableServiceLinks	boolean	EnableServiceLinks indicates whether information about services should be injected into pod's environment variables, matching the syntax of Docker links. Optional: Defaults to true.

Property	Type	Description
ephemeralContainers	array	List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's <code>ephemeralcontainers</code> subresource.
ephemeralContainers[]	object	An <code>EphemeralContainer</code> is a temporary container that you may add to an existing <code>Pod</code> for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a <code>Pod</code> is removed or restarted. The kubelet may evict a <code>Pod</code> if an ephemeral container causes the <code>Pod</code> to exceed its resource allocation. To add an ephemeral container, use the <code>ephemeralcontainers</code> subresource of an existing <code>Pod</code>. Ephemeral containers may not be removed or restarted.
hostAliases	array	<code>HostAliases</code> is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
hostAliases[]	object	<code>HostAlias</code> holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.
hostIPC	boolean	Use the host's ipc namespace. Optional: Default to false.

Property	Type	Description
hostNetwork	boolean	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.
hostPID	boolean	Use the host's pid namespace. Optional: Default to false.
hostUsers	boolean	Use the host's user namespace. Optional: Default to true. If set to true or not present, the pod will be run in the host user namespace, useful for when the pod needs a feature only available to the host user namespace, such as loading a kernel module with CAP_SYS_MODULE. When set to false, a new users is created for the pod. Setting false is useful for mitigating container breakout vulnerabilities even allowing users to run their containers as root without actually having root privileges on the host. This field is alpha-level and is only honored by servers that enable the UserNamespacesSupport feature.
hostname	string	Specifies the hostname of the Pod. If not specified, the pod's hostname will be set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
initContainers	array	List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
initContainers[]	object	A single application container that you want to run within a pod.
nodeName	string	nodeName indicates in which node this pod is scheduled. If empty, this pod is a candidate for scheduling by the scheduler defined in schedulerName. Once this field is set, the kubelet for this node becomes responsible for the lifecycle of this pod. This field should not be used to express a desire for the pod to be scheduled on a specific node. https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodename

Property	Type	Description
nodeSelector	object (string)	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/
os	object	PodOS defines the OS parameters of a pod.
overhead	object (Quantity)	Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset. Possible enum values: - "Never" means that pod never preempts other pods with lower priority. - "PreemptLowerPriority" means that pod can preempt other pods with lower priority.

Property	Type	Description
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it prevents users from setting this field. The admission controller populates this field from PriorityClassName. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-node-critical" and "system-cluster-critical" are two special keywords which indicate the highest priorities with the former being the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.
readinessGates	array	If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	object	PodReadinessGate contains the reference to a pod condition
resourceClaims	array	ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.

Property	Type	Description
resourceClaims[]	object	PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.
resources	object	ResourceRequirements describes the compute resource requirements.
restartPolicy	string	Restart policy for all containers within the pod. One of Always, OnFailure, Never. In some contexts, only a subset of those values may be permitted. Default to Always. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy Possible enum values: - "Always" - "Never" - "OnFailure"
runtimeClassName	string	RuntimeClassName refers to a RuntimeClass object in the node.k8s.io group, which should be used to run this pod. If no RuntimeClass resource matches the named class, the pod will not be run. If unset or empty, the "legacy" RuntimeClass will be used, which is an implicit class with an empty definition that uses the default runtime handler. More info: https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class

Property	Type	Description
schedulerName	string	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
schedulingGates	array	SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.
schedulingGates[]	object	PodSchedulingGate is associated to a Pod to guard its scheduling.
securityContext	object	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
serviceAccount	string	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
serviceAccountName	string	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Property	Type	Description
setHostnameAsFQDN	boolean	If true the pod's hostname will be configured as the pod's FQDN, rather than the leaf name (the default). In Linux containers, this means setting the FQDN in the hostname field of the kernel (the nodename field of struct utsname). In Windows containers, this means setting the registry value of hostname for the registry key HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters to FQDN. If a pod does not have FQDN, this has no effect. Default to false.
shareProcessNamespace	boolean	Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false.
subdomain	string	If specified, the fully qualified Pod hostname will be "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>". If not specified, the pod will not have a domainname at all.

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.
tolerations	array	If specified, the pod's tolerations.
tolerations[]	object	The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.
topologySpreadConstraints	array	TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.
topologySpreadConstraints[]	object	TopologySpreadConstraint specifies how to spread matching pods among the given topology.
volumes	array	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	Volume represents a named volume in a pod that may be accessed by any container in the pod.

14.1.2. .spec.affinity

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Node affinity is a group of node affinity scheduling rules.
podAffinity	object	Pod affinity is a group of inter pod affinity scheduling rules.
podAntiAffinity	object	Pod anti affinity is a group of inter pod anti affinity scheduling rules.

14.1.3. .spec.affinity.nodeAffinity

Description

Node affinity is a group of node affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution[]	object	An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).
requiredDuringSchedulingIgnoredDuringExecution	object	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

14.1.4. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

14.1.5. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

- **weight**
- **preference**

Property	Type	Description

Property	Type	Description
preference	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

14.1.6. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

14.1.7. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[

Description

A list of node selector requirements by node's labels.

Type

array

14.1.8. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

14.1.9. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[

Description

A list of node selector requirements by node's fields.

Type

array

14.1.10. .spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

14.1.11. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

Type

object

Required

- **nodeSelectorTerms**

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.
nodeSelectorTerms[]	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

14.1.12. `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.r`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

14.1.13. `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.r`

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

14.1.14. `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.i`

Description

A list of node selector requirements by node's labels.

Type

array

14.1.15. `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.r`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

14.1.16. `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.r`

Description

A list of node selector requirements by node's fields.

Type

array

14.1.17. .spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.r

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

14.1.18. .spec.affinity.podAffinity

Description

Pod affinity is a group of inter pod affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	array	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

14.1.19. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

14.1.20. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

14.1.21. .spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

14.1.22. `.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each `podAffinityTerm` are intersected, i.e. all terms must be satisfied.

Type

array

14.1.23. `.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]`

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

14.1.24. .spec.affinity.podAntiAffinity

Description

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution	array	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

14.1.25. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

14.1.26. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

14.1.27. .spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type**object**

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

14.1.28. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecuti

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

14.1.29. .spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecuti

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

14.1.30. .spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Type

array

14.1.31. .spec.containers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.

Property	Type	Description
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

14.1.32. .spec.containers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

14.1.33. .spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

14.1.34. .spec.containers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

14.1.35. .spec.containers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

14.1.36. .spec.containers[].env[].valueFrom.fieldRef**Description**

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

14.1.37. .spec.containers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

14.1.38. .spec.containers[].env[].valueFrom.secretKeyRef**Description**

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

14.1.39. .spec.containers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

14.1.40. .spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

14.1.41. .spec.containers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

14.1.42. .spec.containers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

14.1.43. .spec.containers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

14.1.44. .spec.containers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP socket action describes an action based on opening a socket

14.1.45. .spec.containers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.46. .spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.47. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.48. .spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.49. .spec.containers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.50. .spec.containers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.51. .spec.containers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

14.1.52. .spec.containers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.53. .spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.54. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.55. .spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.56. .spec.containers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.57. .spec.containers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.58. .spec.containers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.59. .spec.containers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.60. .spec.containers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.61. .spec.containers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.62. .spec.containers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.63. .spec.containers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.64. .spec.containers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.65. .spec.containers[].ports**Description**

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

14.1.66. .spec.containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

14.1.67. .spec.containers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.68. .spec.containers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.69. .spec.containers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.70. .spec.containers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.71. .spec.containers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.72. .spec.containers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.73. .spec.containers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.74. .spec.containers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

14.1.75. .spec.containers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

14.1.76. .spec.containers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.77. .spec.containers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.78. .spec.containers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.79. `.spec.containers[].securityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

14.1.80. .spec.containers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

14.1.81. .spec.containers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

14.1.82. .spec.containers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

14.1.83. .spec.containers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

14.1.84. .spec.containers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

14.1.85. .spec.containers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.86. .spec.containers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.87. .spec.containers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.88. .spec.containers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.89. .spec.containers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.90. .spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.91. .spec.containers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

▼ **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.92. .spec.containers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

14.1.93. .spec.containers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

14.1.94. .spec.containers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

14.1.95. .spec.containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

14.1.96. .spec.dnsConfig

Description

PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Type

object

Property	Type	Description
nameservers	array (string)	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
options	array	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
options[]	object	PodDNSConfigOption defines DNS resolver options of a pod.
searches	array (string)	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

14.1.97. .spec.dnsConfig.options**Description**

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

14.1.98. .spec.dnsConfig.options[]**Description**

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
name	string	Name is this DNS resolver option's name. Required.
value	string	Value is this DNS resolver option's value.

14.1.99. .spec.ephemeralContainers

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

array

14.1.100. .spec.ephemeralContainers[]

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present

Property	Type	Description
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the ephemeral container specified as a DNS_LABEL. This name must be unique among all containers, init containers and ephemeral containers.
ports	array	Ports are not allowed for ephemeral containers.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	Restart policy for the container to manage the restart behavior of each container within a pod. This may only be set for init containers. You cannot set this field on ephemeral containers.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Property	Type	Description
targetContainerName	string	If set, the name of the container from PodSpec that this ephemeral container targets. The ephemeral container will be run in the namespaces (IPC, PID, etc) of this container. If not set then the ephemeral container uses the namespaces configured in the Pod spec. The container runtime must implement support for this feature. If the runtime does not support namespace targeting then the result of setting this field is undefined.
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to <code>/dev/termination-log</code>. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Property	Type	Description
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

14.1.101. .spec.ephemeralContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

14.1.102. .spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

14.1.103. .spec.ephemeralContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

14.1.104. .spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

14.1.105. .spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

14.1.106. .spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

14.1.107. .spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

14.1.108. .spec.ephemeralContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

14.1.109. .spec.ephemeralContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

14.1.110. .spec.ephemeralContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

14.1.111. .spec.ephemeralContainers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

14.1.112. .spec.ephemeralContainers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type
object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

14.1.113. .spec.ephemeralContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP socket action describes an action based on opening a socket

14.1.114. .spec.ephemeralContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.115. .spec.ephemeralContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.116. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.117. .spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.118. .spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.119. .spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.120. .spec.ephemeralContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

14.1.121. .spec.ephemeralContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.122. .spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.123. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.124. .spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.125. .spec.ephemeralContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.126. .spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.127. `.spec.ephemeralContainers[].livenessProbe`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.128. .spec.ephemeralContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.129. .spec.ephemeralContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.130. .spec.ephemeralContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.131. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.132. .spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.133. .spec.ephemeralContainers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.134. .spec.ephemeralContainers[].ports**Description**

Ports are not allowed for ephemeral containers.

Type

array

14.1.135. .spec.ephemeralContainers[].ports[]**Description**

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

14.1.136. .spec.ephemeralContainers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.137. .spec.ephemeralContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.138. .spec.ephemeralContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.139. .spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.140. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.141. .spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.142. .spec.ephemeralContainers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.143. .spec.ephemeralContainers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

14.1.144. .spec.ephemeralContainers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

14.1.145. .spec.ephemeralContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.146. .spec.ephemeralContainers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.147. .spec.ephemeralContainers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.148. `.spec.ephemeralContainers[].securityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

14.1.149. .spec.ephemeralContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

14.1.150. .spec.ephemeralContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

14.1.151. .spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

14.1.152. .spec.ephemeralContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

14.1.153. .spec.ephemeralContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

14.1.154. .spec.ephemeralContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's <code>terminationGracePeriodSeconds</code> will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.155. .spec.ephemeralContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.156. .spec.ephemeralContainers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.157. .spec.ephemeralContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.158. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.159. .spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.160. .spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

▼ **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.161. .spec.ephemeralContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

14.1.162. .spec.ephemeralContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

14.1.163. .spec.ephemeralContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

array

14.1.164. `.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

14.1.165. .spec.hostAliases

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

array

14.1.166. .spec.hostAliases[]

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

- **ip**

Property	Type	Description
hostnames	array (string)	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

14.1.167. .spec.imagePullSecrets

Description

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

14.1.168. .spec.imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.169. .spec.initContainers

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

14.1.170. .spec.initContainers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal <code>"\$(VAR_NAME)"</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal <code>"\$(VAR_NAME)"</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.

Property	Type	Description
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

14.1.171. .spec.initContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

14.1.172. .spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

14.1.173. .spec.initContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

14.1.174. .spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

14.1.175. .spec.initContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

14.1.176. .spec.initContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

14.1.177. .spec.initContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

14.1.178. .spec.initContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

14.1.179. .spec.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

14.1.180. .spec.initContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

14.1.181. .spec.initContainers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

14.1.182. .spec.initContainers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type
object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

14.1.183. .spec.initContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP SocketAction describes an action based on opening a socket

14.1.184. .spec.initContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type
object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.185. .spec.initContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type
object

Required

- port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.186. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.187. .spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.188. .spec.initContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.189. .spec.initContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.190. .spec.initContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

14.1.191. .spec.initContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.192. .spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.193. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.194. .spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.195. .spec.initContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

14.1.196. .spec.initContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.197. .spec.initContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.198. .spec.initContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.199. .spec.initContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.200. .spec.initContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.201. .spec.initContainers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.202. .spec.initContainers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.203. .spec.initContainers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.204. .spec.initContainers[].ports**Description**

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

14.1.205. .spec.initContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

14.1.206. .spec.initContainers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.207. .spec.initContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.208. .spec.initContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.209. .spec.initContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.210. .spec.initContainers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.211. .spec.initContainers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.212. .spec.initContainers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.213. .spec.initContainers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

14.1.214. .spec.initContainers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

14.1.215. .spec.initContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.216. .spec.initContainers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.217. .spec.initContainers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.218. `.spec.initContainers[].securityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

14.1.219. .spec.initContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

14.1.220. .spec.initContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

14.1.221. .spec.initContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

14.1.222. .spec.initContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost – a profile defined in a file on the node should be used. RuntimeDefault – the container runtime default profile should be used. Unconfined – no profile should be applied. Possible enum values: – "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. – "RuntimeDefault" represents the default container runtime seccomp profile. – "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

14.1.223. .spec.initContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

14.1.224. .spec.initContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's <code>terminationGracePeriodSeconds</code> will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

14.1.225. .spec.initContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

14.1.226. .spec.initContainers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

14.1.227. .spec.initContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

14.1.228. .spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

14.1.229. .spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

14.1.230. .spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

▼ port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

14.1.231. .spec.initContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

14.1.232. .spec.initContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

14.1.233. .spec.initContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

14.1.234. .spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

14.1.235. .spec.os

Description

PodOS defines the OS parameters of a pod.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration Clients should expect to handle additional values and treat unrecognized values in this field as os: null

14.1.236. .spec.readinessGates**Description**

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True"

More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

14.1.237. .spec.readinessGates[]**Description**

PodReadinessGate contains the reference to a pod condition

Type

object

Required

- **conditionType**

Property	Type	Description
conditionType	string	ConditionType refers to a condition in the pod's condition list with matching type.

14.1.238. .spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name.

This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable.

Type

array

14.1.239. .spec.resourceClaims[]

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod.

It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name uniquely identifies this resource claim inside the pod. This must be a DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

Property	Type	Description
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in <code>pod.status.resourceClaimStatus</code>. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

14.1.240. .spec.resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.241. .spec.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.242. .spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.243. .spec.schedulingGates

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod.

SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

14.1.244. .spec.schedulingGates[]

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name of the scheduling gate. Each scheduling gate must have a unique name field.

14.1.245. .spec.securityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership (and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Always" indicates that volume's ownership and permissions should always be changed whenever volume is mounted inside a Pod. This the default behavior. - "OnRootMismatch" indicates that volume's ownership and permissions will be changed only when permission and ownership of root directory does not match with expected permissions on the volume. This can help shorten the time it takes to change ownership and permissions of a volume.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with -o context mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible

Property	Type	Description
		volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when <code>spec.os.name</code> is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Property	Type	Description
supplementalGroups	array (integer)	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Merge" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be merged with the primary user's groups as defined in the container image (in /etc/group). - "Strict" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be used instead of any groups defined in the container image.

Property	Type	Description
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
sysctls[]	object	Sysctl defines a kernel parameter to be set
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

14.1.246. .spec.securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".

Property	Type	Description
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

14.1.247. .spec.securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

14.1.248. .spec.securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

14.1.249. .spec.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

14.1.250. .spec.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

14.1.251. .spec.securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

14.1.252. .spec.tolerations

Description

If specified, the pod's tolerations.

Type

array

14.1.253. .spec.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute. Possible enum values: - "NoExecute" Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController. - "NoSchedule" Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler. - "PreferNoSchedule" Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Possible enum values: - "Equal" - "Exists"

Property	Type	Description
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	string	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

14.1.254. .spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

14.1.255. .spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

Property	Type	Description

Property	Type	Description
labelSelector	LabelSelector	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the <code>MatchLabelKeysInPodTopologySpread</code> feature gate to be enabled (enabled by default).

Property	Type	Description
maxSkew	integer	MaxSkew describes the degree to which pods may be unevenly distributed. When whenUnsatisfiable=DoNotSchedule, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When whenUnsatisfiable=ScheduleAnyway, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
minDomains	integer	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.

Property	Type	Description
nodeAffinityPolicy	string	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are: - Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.

Property	Type	Description
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket. We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.

Property	Type	Description
whenUnsatisfiable	string	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it more imbalanced. It's a required field. Possible enum values: - "DoNotSchedule" instructs the scheduler not to schedule the pod when constraints are not satisfied. - "ScheduleAnyway" instructs the scheduler to schedule the pod even if constraints are not satisfied.

14.1.256. .spec.volumes

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

14.1.257. .spec.volumes[]

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

object

Required

- **name**

Property	Type	Description
awsElasticBlockStore	object	Represents a Persistent Disk resource in AWS. An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.
azureDisk	object	AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
azureFile	object	AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
cephfs	object	Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.
cinder	object	Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Property	Type	Description
configMap	object	Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.
csi	object	Represents a source location of a volume to mount, managed by an external CSI driver
downwardAPI	object	DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
emptyDir	object	Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
ephemeral	object	Represents an ephemeral volume that is handled by a normal storage driver.
fc	object	Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.
flexVolume	object	FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Property	Type	Description
flocker	object	Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.
gcePersistentDisk	object	Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.
gitRepo	object	Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.
glusterfs	object	Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
hostPath	object	Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
image	object	ImageVolumeSource represents a image volume resource.

Property	Type	Description
iscsi	object	Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.
name	string	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).
photonPersistentDisk	object	Represents a Photon Controller persistent disk resource.
portworxVolume	object	PortworxVolumeSource represents a Portworx volume resource.
projected	object	Represents a projected volume source
quobyte	object	Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Property	Type	Description
rbd	object	Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
scaleIO	object	ScaleIOVolumeSource represents a persistent ScaleIO volume
secret	object	Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
storageos	object	Represents a StorageOS persistent volume resource.
vsphereVolume	object	Represents a vSphere volume resource.

14.1.258. .spec.volumes[].awsElasticBlockStore

Description

Represents a Persistent Disk resource in AWS.

An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

Type

object

Required

- **volumeID**

Property	Type	Description

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

14.1.259. .spec.volumes[].azureDisk

Description

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

- **diskName**
- **diskURI**

Property	Type	Description
cachingMode	string	cachingMode is the Host Caching mode: None, Read Only, Read Write. Possible enum values: - "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskName is the Name of the data disk in the blob storage
diskURI	string	diskURI is the URI of data disk in the blob storage
fsType	string	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	string	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared Possible enum values: - "Dedicated" - "Managed" - "Shared"
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

14.1.260. .spec.volumes[].azureFile

Description

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

object

Required

- **secretName**
- **shareName**

Property	Type	Description
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretName	string	secretName is the name of secret that contains Azure Storage Account Name and Key
shareName	string	shareName is the azure share Name

14.1.261. .spec.volumes[].cephfs

Description

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **monitors**

Property	Type	Description
monitors	array (string)	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Property	Type	Description
secretFile	string	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

14.1.262. .spec.volumes[].cephfs.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.263. .spec.volumes[].cinder

Description

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Type

object**Required**

- **volumeID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeID	string	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

14.1.264. .spec.volumes[].cinder.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.265. .spec.volumes[].configMap

Description

Adapts a ConfigMap into a volume.

The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

14.1.266. .spec.volumes[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

14.1.267. .spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type**object****Required**

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

14.1.268. .spec.volumes[].csi**Description**

Represents a source location of a volume to mount, managed by an external CSI driver

Type**object****Required**

- **driver**

Property	Type	Description

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
volumeAttributes	object (string)	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

14.1.269. .spec.volumes[].csi.nodePublishSecretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.270. .spec.volumes[].downwardAPI

Description

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

14.1.271. .spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

14.1.272. .spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

14.1.273. .spec.volumes[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
fieldPath	string	Path of the field to select in the specified API version.

14.1.274. .spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

14.1.275. .spec.volumes[].emptyDir

Description

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
sizeLimit	Quantity	sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

14.1.276. .spec.volumes[].ephemeral

Description

Represents an ephemeral volume that is handled by a normal storage driver.

Type

object

Property	Type	Description
volumeClaimTemplate	object	PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

14.1.277. .spec.volumes[].ephemeral.volumeClaimTemplate

Description

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

Type

object

Required

- **spec**

Property	Type	Description
metadata	ObjectMeta	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
spec	object	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

14.1.278. .spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

Type

object

Property	Type	Description
accessModes	array (string)	accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.
dataSourceRef	object	TypedObjectReference contains enough information to let you locate the typed referenced object
resources	object	VolumeResourceRequirements describes the storage resource requirements for a volume.
selector	LabelSelector	selector is a label query over volumes to consider for binding.

Property	Type	Description
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).

Property	Type	Description
volumeMode	string	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec. Possible enum values: - "Block" means the volume will not be formatted with a filesystem and will remain a raw block device. - "Filesystem" means the volume will be or is formatted with a filesystem.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

14.1.279. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

14.1.280. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef**Description**

TypedObjectReference contains enough information to let you locate the typed referenced object

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

14.1.281. .spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources**Description**

VolumeResourceRequirements describes the storage resource requirements for a volume.

Type

object

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.282. .spec.volumes[].fc

Description

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	integer	lun is Optional: FC target lun number
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	array (string)	targetWWNs is Optional: FC target worldwide names (WWNs)

Property	Type	Description
wwids	array (string)	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

14.1.283. .spec.volumes[].flexVolume

Description

FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

object

Required

- **driver**

Property	Type	Description
driver	string	driver is the name of the driver to use for this volume.
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	object (string)	options is Optional: this field holds extra command options if any.
readOnly	boolean	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

14.1.284. .spec.volumes[].flexVolume.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.285. .spec.volumes[].flocker

Description

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

Type

object

Property	Type	Description
datasetName	string	datasetName is Name of the dataset stored as metadata → name on the dataset for Flocker should be considered as deprecated
datasetUUID	string	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

14.1.286. .spec.volumes[].gcePersistentDisk

Description

Represents a Persistent Disk resource in Google Compute Engine.

A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

Type

object**Required**

- **pdName**

Property	Type	Description
fsType	string	fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	string	pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

14.1.287. .spec.volumes[].gitRepo**Description**

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

object

Required

- **repository**

Property	Type	Description
directory	string	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
repository	string	repository is the URL
revision	string	revision is the commit hash for the specified revision.

14.1.288. .spec.volumes[].glusterfs

Description

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **endpoints**
- **path**

Property	Type	Description

Property	Type	Description
endpoints	string	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	boolean	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

14.1.289. .spec.volumes[].hostPath

Description

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **path**

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Property	Type	Description
type	string	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath Possible enum values: - "" For backwards compatible, leave it empty if unset - "BlockDevice" A block device must exist at the given path - "CharDevice" A character device must exist at the given path - "Directory" A directory must exist at the given path - "DirectoryOrCreate" If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. - "File" A file must exist at the given path - "FileOrCreate" If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. - "Socket" A UNIX socket must exist at the given path

14.1.290. .spec.volumes[].image

Description

ImageVolumeSource represents a image volume resource.

Type

object

Property	Type	Description

Property	Type	Description
pullPolicy	string	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
reference	string	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Description

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

Type

object

Required

- **targetPortal**
- **iqn**
- **lun**

Property	Type	Description
chapAuthDiscovery	boolean	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	boolean	chapAuthSession defines whether support iSCSI Session CHAP authentication
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.
iqn	string	iqn is the target iSCSI Qualified Name.
iscsiInterface	string	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

Property	Type	Description
lun	integer	lun represents iSCSI Target Lun number.
portals	array (string)	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
targetPortal	string	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

14.1.292. .spec.volumes[].iscsi.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.293. .spec.volumes[].nfs

Description

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **server**
- **path**

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

14.1.294. .spec.volumes[].persistentVolumeClaim**Description**

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A

PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

Type

object

Required

- **claimName**

Property	Type	Description

Property	Type	Description
claimName	string	claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	boolean	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

14.1.295. .spec.volumes[].photonPersistentDisk

Description

Represents a Photon Controller persistent disk resource.

Type

object

Required

- **pdID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

14.1.296. .spec.volumes[].portworxVolume

Description

PortworxVolumeSource represents a Portworx volume resource.

Type

object

Required

- **volumeID**

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

14.1.297. .spec.volumes[].projected

Description

Represents a projected volume source

Type

object

Property	Type	Description
defaultMode	integer	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
sources	array	sources is the list of volume projections. Each entry in this list handles one source.
sources[]	object	Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

14.1.298. .spec.volumes[].projected.sources**Description**

`sources` is the list of volume projections. Each entry in this list handles one source.

Type

array

14.1.299. .spec.volumes[].projected.sources[]**Description**

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

object

Property	Type	Description
clusterTrustBundle	object	ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.
configMap	object	Adapts a ConfigMap into a projected volume. The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.
downwardAPI	object	Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Property	Type	Description
secret	object	Adapts a secret into a projected volume. The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.
serviceAccountToken	object	ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

14.1.300. .spec.volumes[].projected.sources[].clusterTrustBundle

Description

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

Type

object

Required

- **path**

Property	Type	Description
labelSelector	LabelSelector	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
name	string	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	boolean	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	string	Relative path from the volume root to write the bundle.
signerName	string	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

14.1.301. .spec.volumes[].projected.sources[].configMap

Description

Adapts a ConfigMap into a projected volume.

The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

Type

object

Property	Type	Description

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

14.1.302. .spec.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

14.1.303. .spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

14.1.304. .spec.volumes[].projected.sources[].downwardAPI

Description

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Type

object

Property	Type	Description
items	array	Items is a list of DownwardAPIVolume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

14.1.305. .spec.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

array

14.1.306. .spec.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

14.1.307. .spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

14.1.308. .spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef**Description**

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

14.1.309. .spec.volumes[].projected.sources[].secret**Description**

Adapts a secret into a projected volume.

The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional field specify whether the Secret or its key must be defined

14.1.310. .spec.volumes[].projected.sources[].secret.items**Description**

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array**14.1.311. .spec.volumes[].projected.sources[].secret.items[]****Description**

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

14.1.312. .spec.volumes[].projected.sources[].serviceAccountToken

Description

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

Type

object

Required

- **path**

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours. Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

14.1.313. .spec.volumes[].quobyte

Description

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **registry**
- **volume**

Property	Type	Description
group	string	group to map volume access to Default is no group

Property	Type	Description
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

14.1.314. .spec.volumes[].rbd

Description

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

Type

object

Required

- **monitors**
- **image**

Property	Type	Description

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	array (string)	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

14.1.315. .spec.volumes[].rbid.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.316. .spec.volumes[].scaleIO

Description

ScaleIOVolumeSource represents a persistent ScaleIO volume

Type

object

Required

- **gateway**
- **system**
- **secretRef**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	string	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.

Property	Type	Description
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
sslEnabled	boolean	sslEnabled Flag enable/disable SSL communication with Gateway, default false
storageMode	string	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
storagePool	string	storagePool is the ScaleIO Storage Pool associated with the protection domain.
system	string	system is the name of the storage system as configured in ScaleIO.
volumeName	string	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

14.1.317. .spec.volumes[].scaleIO.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.318. .spec.volumes[].secret

Description

Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
optional	boolean	optional field specify whether the Secret or its keys must be defined
secretName	string	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

14.1.319. .spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

14.1.320. .spec.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

14.1.321. .spec.volumes[].storageos

Description

Represents a StorageOS persistent volume resource.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeName	string	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
volumeNamespace	string	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

14.1.322. .spec.volumes[].storageos.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

14.1.323. .spec.volumes[].vsphereVolume

Description

Represents a vSphere volume resource.

Type

object

Required

- **volumePath**

Property	Type	Description
fsType	string	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	string	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	string	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	string	volumePath is the path that identifies vSphere volume vmdk

14.1.324. .status**Description**

PodStatus represents information about the status of a pod. Status may trail the actual state of a system, especially if the node that hosts the pod cannot contact the control plane.

Type

object

Property	Type	Description
conditions	array	Current service state of pod. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions
conditions[]	object	PodCondition contains details for the current condition of this pod.

Property	Type	Description
containerStatuses	array	Statuses of containers in this pod. Each container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various Kubernetes components is not defined and those statuses might be ignored. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status
containerStatuses[]	object	ContainerStatus contains details for the current status of this container.
ephemeralContainerStatuses	array	Statuses for any ephemeral containers that have run in this pod. Each ephemeral container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various Kubernetes components is not defined and those statuses might be ignored. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status
ephemeralContainerStatuses[]	object	ContainerStatus contains details for the current status of this container.

Property	Type	Description
hostIP	string	hostIP holds the IP address of the host to which the pod is assigned. Empty if the pod has not started yet. A pod can be assigned to a node that has a problem in kubelet which in turns mean that HostIP will not be updated even if there is a node is assigned to pod
hostIPs	array	hostIPs holds the IP addresses allocated to the host. If this field is specified, the first entry must match the hostIP field. This list is empty if the pod has not started yet. A pod can be assigned to a node that has a problem in kubelet which in turns means that HostIPs will not be updated even if there is a node is assigned to this pod.
hostIPs[]	object	HostIP represents a single IP address allocated to the host.
initContainerStatuses	array	Statuses of init containers in this pod. The most recent successful non-restartable init container will have ready = true, the most recently started container will have startTime set. Each init container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various Kubernetes components is not defined and those statuses might be ignored. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#pod-and-container-status
initContainerStatuses[]	object	ContainerStatus contains details for the current status of this container.

Property	Type	Description
message	string	A human readable message indicating details about why the pod is in this condition.
nominatedNodeName	string	nominatedNodeName is set only when this pod preempts other pods on the node, but it cannot be scheduled right away as preemption victims receive their graceful termination periods. This field does not guarantee that the pod will be scheduled on this node. Scheduler may decide to place the pod elsewhere if other nodes become available sooner. Scheduler may also decide to give the resources on this node to a higher priority pod that is created after preemption. As a result, this field may be different than PodSpec.nodeName when the pod is scheduled.
observedGeneration	integer	If set, this represents the .metadata.generation that the pod status was set based upon. This is an alpha field. Enable PodObservedGenerationTracking to be able to use this field.
phase	string	The phase of a Pod is a simple, high-level summary of where the Pod is in its lifecycle. The conditions array, the reason and message fields, and the individual container status arrays contain more detail about the pod's status. There are five possible phase values: Pending: The pod has been accepted by the Kubernetes system, but one or more of the container images has not been created. This includes time before being scheduled as well as time spent downloading images over the network, which could take a while. Running: The pod has been bound to a node, and all of the containers have been created. At least one container is still running,

Property	Type	Description
		or is in the process of starting or restarting. Succeeded: All containers in the pod have terminated in success, and will not be restarted. Failed: All containers in the pod have terminated, and at least one container has terminated in failure. The container either exited with non-zero status or was terminated by the system. Unknown: For some reason the state of the pod could not be obtained, typically due to an error in communicating with the host of the pod. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-phase Possible enum values: - "Failed" means that all containers in the pod have terminated, and at least one container has terminated in a failure (exited with a non-zero exit code or was stopped by the system). - "Pending" means the pod has been accepted by the system, but one or more of the containers has not been started. This includes time before being bound to a node, as well as time spent pulling images onto the host. - "Running" means the pod has been bound to a node and all of the containers have been started. At least one container is still running or is in the process of being restarted. - "Succeeded" means that all containers in the pod have voluntarily terminated with a container exit code of 0, and the system is not going to restart any of these containers. - "Unknown" means that for some reason the state of the pod could not be obtained, typically due to an error in communicating with the host of the pod. Deprecated: It isn't being set since 2015 (74da3b14b0c0f658b3bb8d2def5094686d0e9095)

Property	Type	Description
podIP	string	podIP address allocated to the pod. Routable at least within the cluster. Empty if not yet allocated.
podIPs	array	podIPs holds the IP addresses allocated to the pod. If this field is specified, the 0th entry must match the podIP field. Pods may be allocated at most 1 value for each of IPv4 and IPv6. This list is empty if no IPs have been allocated yet.
podIPs[]	object	PodIP represents a single IP address allocated to the pod.
qosClass	string	The Quality of Service (QOS) classification assigned to the pod based on resource requirements See PodQOSClass type for available QOS classes More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-qos/#quality-of-service-classes Possible enum values: - "BestEffort" is the BestEffort qos class. - "Burstable" is the Burstable qos class. - "Guaranteed" is the Guaranteed qos class.
reason	string	A brief CamelCase message indicating details about why the pod is in this state. e.g. 'Evicted'

Property	Type	Description
resize	string	Status of resources resize desired for pod's containers. It is empty if no resources resize is pending. Any changes to container resources will automatically set this to "Proposed" Deprecated: Resize status is moved to two pod conditions PodResizePending and PodResizeInProgress. PodResizePending will track states where the spec has been resized, but the Kubelet has not yet allocated the resources. PodResizeInProgress will track in-progress resizes, and should be present whenever allocated resources != acknowledged resources.
resourceClaimStatuses	array	Status of resource claims.
resourceClaimStatuses[]	object	PodResourceClaimStatus is stored in the PodStatus for each PodResourceClaim which references a ResourceClaimTemplate. It stores the generated name for the corresponding ResourceClaim.
startTime	Time	RFC 3339 date and time at which the object was acknowledged by the Kubelet. This is before the Kubelet pulled the container image(s) for the pod.

14.1.325. .status.conditions

Description

Current service state of pod. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions>

Type

array

14.1.326. .status.conditions[]

Description

PodCondition contains details for the current condition of this pod.

Type

object**Required**

- **type**
- **status**

Property	Type	Description
lastProbeTime	Time	Last time we probed the condition.
lastTransitionTime	Time	Last time the condition transitioned from one status to another.
message	string	Human-readable message indicating details about last transition.
observedGeneration	integer	If set, this represents the <code>.metadata.generation</code> that the pod condition was set based upon. This is an alpha field. Enable <code>PodObservedGenerationTracking</code> to be able to use this field.
reason	string	Unique, one-word, CamelCase reason for the condition's last transition.
status	string	Status is the status of the condition. Can be True, False, Unknown. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions
type	string	Type is the type of the condition. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions

14.1.327. .status.containerStatuses**Description**

Statuses of containers in this pod. Each container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various

Kubernetes components is not defined and those statuses might be ignored. More info:
<https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status>

Type

array

14.1.328. .status.containerStatuses[]

Description

ContainerStatus contains details for the current status of this container.

Type

object

Required

- **name**
- **ready**
- **restartCount**
- **image**
- **imageID**

Property	Type	Description
allocatedResources	object (Quantity)	AllocatedResources represents the compute resources allocated for this container by the node. Kubelet sets this value to Container.Resources.Requests upon successful pod admission and after successfully admitting desired pod resize.
allocatedResourcesStatus	array	AllocatedResourcesStatus represents the status of various resources allocated for this Pod.
allocatedResourcesStatus[]	object	ResourceStatus represents the status of a single resource allocated to a Pod.
containerID	string	ContainerID is the ID of the container in the format '<type>://<container_id>'. Where type is a container runtime identifier, returned from Version call of CRI API (for example "containerd").

Property	Type	Description
image	string	Image is the name of container image that the container is running. The container image may not match the image used in the PodSpec, as it may have been resolved by the runtime. More info: https://kubernetes.io/docs/concepts/containers/images .
imageID	string	ImageID is the image ID of the container's image. The image ID may not match the image ID of the image used in the PodSpec, as it may have been resolved by the runtime.
lastState	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.
name	string	Name is a DNS_LABEL representing the unique name of the container. Each container in a pod must have a unique name across all container types. Cannot be updated.
ready	boolean	Ready specifies whether the container is currently passing its readiness check. The value will change as readiness probes keep executing. If no readiness probes are specified, this field defaults to true once the container is fully started (see Started field). The value is typically used to determine whether a container is ready to accept traffic.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartCount	integer	RestartCount holds the number of times the container has been restarted. Kubelet makes an effort to always increment the value, but there are cases when the state may be lost due to node restarts and then the value may be reset to 0. The value is never negative.
started	boolean	Started indicates whether the container has finished its postStart lifecycle hook and passed its startup probe. Initialized as false, becomes true after startupProbe is considered successful. Resets to false when the container is restarted, or if kubelet loses state temporarily. In both cases, startup probes will run again. Is always true when no startupProbe is defined and container is running and has passed the postStart lifecycle hook. The null value must be treated the same as false.
state	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Property	Type	Description
stopSignal	string	StopSignal reports the effective stop signal for this container Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" - "SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

Property	Type	Description
		Container user represents user identity information
volumeMounts	array	Status of volume mounts.
volumeMounts[]	object	VolumeMountStatus shows status of volume mounts.

14.1.329. .status.containerStatuses[].allocatedResourcesStatus

Description

AllocatedResourcesStatus represents the status of various resources allocated for this Pod.

Type

array

14.1.330. .status.containerStatuses[].allocatedResourcesStatus[]

Description

ResourceStatus represents the status of a single resource allocated to a Pod.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the resource. Must be unique within the pod and in case of non-DRA resource, match one of the resources from the pod spec. For DRA resources, the value must be "claim: <claim_name>/<request>". When this status is reported about a container, the "claim_name" and "request" must match one of the claims of this container.

Property	Type	Description
resources	array	List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.
resources[]	object	ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP https://kep.k8s.io/4680 .

14.1.331. `.status.containerStatuses[].allocatedResourcesStatus[].resources`

Description

List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.

Type

array

14.1.332. `.status.containerStatuses[].allocatedResourcesStatus[].resources[]`

Description

ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP <https://kep.k8s.io/4680>.

Type

object

Required

- **resourceID**

Property	Type	Description

Property	Type	Description
health	string	Health of the resource. can be one of: - Healthy: operates as normal - Unhealthy: reported unhealthy. We consider this a temporary health issue since we do not have a mechanism today to distinguish temporary and permanent issues. - Unknown: The status cannot be determined. For example, Device Plugin got unregistered and hasn't been re-registered since. In future we may want to introduce the PermanentlyUnhealthy Status.
resourceID	string	ResourceID is the unique identifier of the resource. See the ResourceID type for more information.

14.1.333. .status.containerStatuses[].lastState

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.334. .status.containerStatuses[].lastState.running

Description

ContainerStateRunning is a running state of a container.

Type
object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.335. .status.containerStatuses[].lastState.terminated

Description

ContainerStateTerminated is a terminated state of a container.

Type
object

Required

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.336. .status.containerStatuses[].lastState.waiting

Description

ContainerStateWaiting is a waiting state of a container.

Type

object

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.337. .status.containerStatuses[].resources**Description**

ResourceRequirements describes the compute resource requirements.

Type**object**

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.338. `.status.containerStatuses[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.339. `.status.containerStatuses[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.340. `.status.containerStatuses[].state`

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.341. `.status.containerStatuses[].state.running`

Description

ContainerStateRunning is a running state of a container.

Type

object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.342. `.status.containerStatuses[].state.terminated`

Description

ContainerStateTerminated is a terminated state of a container.

Type

object

Required

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.343. `.status.containerStatuses[].state.waiting`

Description

ContainerStateWaiting is a waiting state of a container.

Type

object

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.344. `.status.containerStatuses[].user`

Description

ContainerUser represents user identity information

Type
object

Property	Type	Description
linux	object	LinuxContainerUser represents user identity information in Linux containers

14.1.345. `.status.containerStatuses[].user.linux`

Description

LinuxContainerUser represents user identity information in Linux containers

Type
object

Required

- **uid**
- **gid**

Property	Type	Description
gid	integer	GID is the primary gid initially attached to the first process in the container
supplementalGroups	array (integer)	SupplementalGroups are the supplemental groups initially attached to the first process in the container
uid	integer	UID is the primary uid initially attached to the first process in the container

14.1.346. `.status.containerStatuses[].volumeMounts`

Description

Status of volume mounts.

Type
array

14.1.347. `.status.containerStatuses[].volumeMounts[]`

Description

VolumeMountStatus shows status of volume mounts.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	MountPath corresponds to the original VolumeMount.
name	string	Name corresponds to the name of the original VolumeMount.
readOnly	boolean	ReadOnly corresponds to the original VolumeMount.
recursiveReadOnly	string	RecursiveReadOnly must be set to Disabled, Enabled, or unspecified (for non-readonly mounts). An IfPossible value in the original VolumeMount must be translated to Disabled or Enabled, depending on the mount result.

14.1.348. .status.ephemeralContainerStatuses

Description

Statuses for any ephemeral containers that have run in this pod. Each ephemeral container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various Kubernetes components is not defined and those statuses might be ignored. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-and-container-status>

Type

array

14.1.349. .status.ephemeralContainerStatuses[]

Description

ContainerStatus contains details for the current status of this container.

Type

object

Required

- **name**

- **ready**
- **restartCount**
- **image**
- **imageID**

Property	Type	Description
allocatedResources	object (Quantity)	AllocatedResources represents the compute resources allocated for this container by the node. Kubelet sets this value to Container.Resources.Requests upon successful pod admission and after successfully admitting desired pod resize.
allocatedResourcesStatus	array	AllocatedResourcesStatus represents the status of various resources allocated for this Pod.
allocatedResourcesStatus[]	object	ResourceStatus represents the status of a single resource allocated to a Pod.
containerID	string	ContainerID is the ID of the container in the format '<type>://<container_id>'. Where type is a container runtime identifier, returned from Version call of CRI API (for example "containerd").
image	string	Image is the name of container image that the container is running. The container image may not match the image used in the PodSpec, as it may have been resolved by the runtime. More info: https://kubernetes.io/docs/concepts/containers/images .
imageID	string	ImageID is the image ID of the container's image. The image ID may not match the image ID of the image used in the PodSpec, as it may have been resolved by the runtime.

Property	Type	Description
lastState	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.
name	string	Name is a DNS_LABEL representing the unique name of the container. Each container in a pod must have a unique name across all container types. Cannot be updated.
ready	boolean	Ready specifies whether the container is currently passing its readiness check. The value will change as readiness probes keep executing. If no readiness probes are specified, this field defaults to true once the container is fully started (see Started field). The value is typically used to determine whether a container is ready to accept traffic.
resources	object	ResourceRequirements describes the compute resource requirements.
restartCount	integer	RestartCount holds the number of times the container has been restarted. Kubelet makes an effort to always increment the value, but there are cases when the state may be lost due to node restarts and then the value may be reset to 0. The value is never negative.

Property	Type	Description
started	boolean	Started indicates whether the container has finished its postStart lifecycle hook and passed its startup probe. Initialized as false, becomes true after startupProbe is considered successful. Resets to false when the container is restarted, or if kubelet loses state temporarily. In both cases, startup probes will run again. Is always true when no startupProbe is defined and container is running and has passed the postStart lifecycle hook. The null value must be treated the same as false.
state	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Property	Type	Description
stopSignal	string	StopSignal reports the effective stop signal for this container Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" - "SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

Property	Type	Description
user	object	ContainerUser represents user identity information
volumeMounts	array	Status of volume mounts.
volumeMounts[]	object	VolumeMountStatus shows status of volume mounts.

14.1.350. .status.ephemeralContainerStatuses[].allocatedResourcesStatus

Description

AllocatedResourcesStatus represents the status of various resources allocated for this Pod.

Type

array

14.1.351. .status.ephemeralContainerStatuses[].allocatedResourcesStatus[]

Description

ResourceStatus represents the status of a single resource allocated to a Pod.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the resource. Must be unique within the pod and in case of non-DRA resource, match one of the resources from the pod spec. For DRA resources, the value must be "claim: <claim_name>/<request>". When this status is reported about a container, the "claim_name" and "request" must match one of the claims of this container.

Property	Type	Description
resources	array	List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.
resources[]	object	ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP https://kep.k8s.io/4680 .

14.1.352. `.status.ephemeralContainerStatuses[].allocatedResourcesStatus[].resources`

Description

List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.

Type

array

14.1.353. `.status.ephemeralContainerStatuses[].allocatedResourcesStatus[].resources`

Description

ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP <https://kep.k8s.io/4680>.

Type

object

Required

- **resourceID**

Property	Type	Description

Property	Type	Description
health	string	Health of the resource. can be one of: - Healthy: operates as normal - Unhealthy: reported unhealthy. We consider this a temporary health issue since we do not have a mechanism today to distinguish temporary and permanent issues. - Unknown: The status cannot be determined. For example, Device Plugin got unregistered and hasn't been re-registered since. In future we may want to introduce the PermanentlyUnhealthy Status.
resourceID	string	ResourceID is the unique identifier of the resource. See the ResourceID type for more information.

14.1.354. .status.ephemeralContainerStatuses[].lastState

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.355. .status.ephemeralContainerStatuses[].lastState.running

Description

ContainerStateRunning is a running state of a container.

Type
object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.356. `.status.ephemeralContainerStatuses[].lastState.terminated`

Description

ContainerStateTerminated is a terminated state of a container.

Type
object

Required

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.357. `.status.ephemeralContainerStatuses[].lastState.waiting`

Description

ContainerStateWaiting is a waiting state of a container.

Type

object

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.358. .status.ephemeralContainerStatuses[].resources**Description**

ResourceRequirements describes the compute resource requirements.

Type**object**

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.359. `.status.ephemeralContainerStatuses[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.360. `.status.ephemeralContainerStatuses[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.361. `.status.ephemeralContainerStatuses[].state`

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.362. `.status.ephemeralContainerStatuses[].state.running`

Description

ContainerStateRunning is a running state of a container.

Type

object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.363. `.status.ephemeralContainerStatuses[].state.terminated`

Description

ContainerStateTerminated is a terminated state of a container.

Type

object

Required

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.364. .status.ephemeralContainerStatuses[].state.waiting

Description

ContainerStateWaiting is a waiting state of a container.

Type

object

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.365. .status.ephemeralContainerStatuses[].user

Description

ContainerUser represents user identity information

Type
object

Property	Type	Description
linux	object	LinuxContainerUser represents user identity information in Linux containers

14.1.366. `.status.ephemeralContainerStatuses[].user.linux`

Description

LinuxContainerUser represents user identity information in Linux containers

Type
object

Required

- **uid**
- **gid**

Property	Type	Description
gid	integer	GID is the primary gid initially attached to the first process in the container
supplementalGroups	array (integer)	SupplementalGroups are the supplemental groups initially attached to the first process in the container
uid	integer	UID is the primary uid initially attached to the first process in the container

14.1.367. `.status.ephemeralContainerStatuses[].volumeMounts`

Description

Status of volume mounts.

Type
array

14.1.368. `.status.ephemeralContainerStatuses[].volumeMounts[]`

Description

VolumeMountStatus shows status of volume mounts.

Type**object****Required**

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	MountPath corresponds to the original VolumeMount.
name	string	Name corresponds to the name of the original VolumeMount.
readOnly	boolean	ReadOnly corresponds to the original VolumeMount.
recursiveReadOnly	string	RecursiveReadOnly must be set to Disabled, Enabled, or unspecified (for non-readonly mounts). An IfPossible value in the original VolumeMount must be translated to Disabled or Enabled, depending on the mount result.

14.1.369. .status.hostIPs**Description**

hostIPs holds the IP addresses allocated to the host. If this field is specified, the first entry must match the hostIP field. This list is empty if the pod has not started yet. A pod can be assigned to a node that has a problem in kubelet which in turns means that HostIPs will not be updated even if there is a node is assigned to this pod.

Type**array****14.1.370. .status.hostIPs[]****Description**

HostIP represents a single IP address allocated to the host.

Type**object****Required**

- **ip**

Property	Type	Description
ip	string	IP is the IP address assigned to the host

14.1.371. .status.initContainerStatuses

Description

Statuses of init containers in this pod. The most recent successful non-restartable init container will have `ready = true`, the most recently started container will have `startTime` set. Each init container in the pod should have at most one status in this list, and all statuses should be for containers in the pod. However this is not enforced. If a status for a non-existent container is present in the list, or the list has duplicate names, the behavior of various Kubernetes components is not defined and those statuses might be ignored. More info: <https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#pod-and-container-status>

Type

array

14.1.372. .status.initContainerStatuses[]

Description

ContainerStatus contains details for the current status of this container.

Type

object

Required

- **name**
- **ready**
- **restartCount**
- **image**
- **imageID**

Property	Type	Description
allocatedResources	object (Quantity)	AllocatedResources represents the compute resources allocated for this container by the node. Kubelet sets this value to <code>Container.Resources.Requests</code> upon successful pod admission and after successfully admitting desired pod resize.

Property	Type	Description
allocatedResourcesStatus	array	AllocatedResourcesStatus represents the status of various resources allocated for this Pod.
allocatedResourcesStatus[]	object	ResourceStatus represents the status of a single resource allocated to a Pod.
containerID	string	ContainerID is the ID of the container in the format '<type>://<container_id>'. Where type is a container runtime identifier, returned from Version call of CRI API (for example "containerd").
image	string	Image is the name of container image that the container is running. The container image may not match the image used in the PodSpec, as it may have been resolved by the runtime. More info: https://kubernetes.io/docs/concepts/containers/images .
imageID	string	ImageID is the image ID of the container's image. The image ID may not match the image ID of the image used in the PodSpec, as it may have been resolved by the runtime.
lastState	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.
name	string	Name is a DNS_LABEL representing the unique name of the container. Each container in a pod must have a unique name across all container types. Cannot be updated.

Property	Type	Description
ready	boolean	Ready specifies whether the container is currently passing its readiness check. The value will change as readiness probes keep executing. If no readiness probes are specified, this field defaults to true once the container is fully started (see Started field). The value is typically used to determine whether a container is ready to accept traffic.
resources	object	ResourceRequirements describes the compute resource requirements.
restartCount	integer	RestartCount holds the number of times the container has been restarted. Kubelet makes an effort to always increment the value, but there are cases when the state may be lost due to node restarts and then the value may be reset to 0. The value is never negative.
started	boolean	Started indicates whether the container has finished its postStart lifecycle hook and passed its startup probe. Initialized as false, becomes true after startupProbe is considered successful. Resets to false when the container is restarted, or if kubelet loses state temporarily. In both cases, startup probes will run again. Is always true when no startupProbe is defined and container is running and has passed the postStart lifecycle hook. The null value must be treated the same as false.
state	object	ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Property	Type	Description
stopSignal	string	StopSignal reports the effective stop signal for this container Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" - "SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

Property	Type	Description
user	object	ContainerUser represents user identity information
volumeMounts	array	Status of volume mounts.
volumeMounts[]	object	VolumeMountStatus shows status of volume mounts.

14.1.373. .status.initContainerStatuses[].allocatedResourcesStatus

Description

AllocatedResourcesStatus represents the status of various resources allocated for this Pod.

Type

array

14.1.374. .status.initContainerStatuses[].allocatedResourcesStatus[]

Description

ResourceStatus represents the status of a single resource allocated to a Pod.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the resource. Must be unique within the pod and in case of non-DRA resource, match one of the resources from the pod spec. For DRA resources, the value must be "claim: <claim_name>/<request>". When this status is reported about a container, the "claim_name" and "request" must match one of the claims of this container.

Property	Type	Description
resources	array	List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.
resources[]	object	ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP https://kep.k8s.io/4680 .

14.1.375. `.status.initContainerStatuses[].allocatedResourcesStatus[].resources`

Description

List of unique resources health. Each element in the list contains an unique resource ID and its health. At a minimum, for the lifetime of a Pod, resource ID must uniquely identify the resource allocated to the Pod on the Node. If other Pod on the same Node reports the status with the same resource ID, it must be the same resource they share. See ResourceID type definition for a specific format it has in various use cases.

Type

array

14.1.376. `.status.initContainerStatuses[].allocatedResourcesStatus[].resources[]`

Description

ResourceHealth represents the health of a resource. It has the latest device health information. This is a part of KEP <https://kep.k8s.io/4680>.

Type

object

Required

- **resourceID**

Property	Type	Description

Property	Type	Description
health	string	Health of the resource. can be one of: - Healthy: operates as normal - Unhealthy: reported unhealthy. We consider this a temporary health issue since we do not have a mechanism today to distinguish temporary and permanent issues. - Unknown: The status cannot be determined. For example, Device Plugin got unregistered and hasn't been re-registered since. In future we may want to introduce the PermanentlyUnhealthy Status.
resourceID	string	ResourceID is the unique identifier of the resource. See the ResourceID type for more information.

14.1.377. `.status.initContainerStatuses[].lastState`

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.378. `.status.initContainerStatuses[].lastState.running`

Description

ContainerStateRunning is a running state of a container.

Type
object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.379. .status.initContainerStatuses[].lastState.terminated

Description

ContainerStateTerminated is a terminated state of a container.

Type
object

Required

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.380. .status.initContainerStatuses[].lastState.waiting

Description

ContainerStateWaiting is a waiting state of a container.

Type

object

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.381. .status.initContainerStatuses[].resources**Description**

ResourceRequirements describes the compute resource requirements.

Type**object**

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

14.1.382. `.status.initContainerStatuses[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate.

This field is immutable. It can only be set for containers.

Type

array

14.1.383. `.status.initContainerStatuses[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

14.1.384. `.status.initContainerStatuses[].state`

Description

ContainerState holds a possible state of container. Only one of its members may be specified. If none of them is specified, the default one is ContainerStateWaiting.

Type

object

Property	Type	Description
running	object	ContainerStateRunning is a running state of a container.
terminated	object	ContainerStateTerminated is a terminated state of a container.
waiting	object	ContainerStateWaiting is a waiting state of a container.

14.1.385. `.status.initContainerStatuses[].state.running`

Description

ContainerStateRunning is a running state of a container.

Type

object

Property	Type	Description
startedAt	Time	Time at which the container was last (re-)started

14.1.386. `.status.initContainerStatuses[].state.terminated`

Description

ContainerStateTerminated is a terminated state of a container.

Type

object**Required**

- **exitCode**

Property	Type	Description
containerID	string	Container's ID in the format '<type>://<container_id>'
exitCode	integer	Exit status from the last termination of the container
finishedAt	Time	Time at which the container last terminated
message	string	Message regarding the last termination of the container
reason	string	(brief) reason from the last termination of the container
signal	integer	Signal from the last termination of the container
startedAt	Time	Time at which previous execution of the container started

14.1.387. .status.initContainerStatuses[].state.waiting**Description**

ContainerStateWaiting is a waiting state of a container.

Type**object**

Property	Type	Description
message	string	Message regarding why the container is not yet running.
reason	string	(brief) reason the container is not yet running.

14.1.388. .status.initContainerStatuses[].user**Description**

ContainerUser represents user identity information

Type
object

Property	Type	Description
linux	object	LinuxContainerUser represents user identity information in Linux containers

14.1.389. `.status.initContainerStatuses[].user.linux`

Description

LinuxContainerUser represents user identity information in Linux containers

Type
object

Required

- **uid**
- **gid**

Property	Type	Description
gid	integer	GID is the primary gid initially attached to the first process in the container
supplementalGroups	array (integer)	SupplementalGroups are the supplemental groups initially attached to the first process in the container
uid	integer	UID is the primary uid initially attached to the first process in the container

14.1.390. `.status.initContainerStatuses[].volumeMounts`

Description

Status of volume mounts.

Type
array

14.1.391. `.status.initContainerStatuses[].volumeMounts[]`

Description

VolumeMountStatus shows status of volume mounts.

Type**object****Required**

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	MountPath corresponds to the original VolumeMount.
name	string	Name corresponds to the name of the original VolumeMount.
readOnly	boolean	ReadOnly corresponds to the original VolumeMount.
recursiveReadOnly	string	RecursiveReadOnly must be set to Disabled, Enabled, or unspecified (for non-readonly mounts). An IfPossible value in the original VolumeMount must be translated to Disabled or Enabled, depending on the mount result.

14.1.392. .status.podIPs**Description**

podIPs holds the IP addresses allocated to the pod. If this field is specified, the 0th entry must match the podIP field. Pods may be allocated at most 1 value for each of IPv4 and IPv6. This list is empty if no IPs have been allocated yet.

Type**array****14.1.393. .status.podIPs[]****Description**

PodIP represents a single IP address allocated to the pod.

Type**object****Required**

- **ip**

Property	Type	Description
ip	string	IP is the IP address assigned to the pod

14.1.394. .status.resourceClaimStatuses

Description

Status of resource claims.

Type

array

14.1.395. .status.resourceClaimStatuses[]

Description

PodResourceClaimStatus is stored in the PodStatus for each PodResourceClaim which references a ResourceClaimTemplate. It stores the generated name for the corresponding ResourceClaim.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name uniquely identifies this resource claim inside the pod. This must match the name of an entry in pod.spec.resourceClaims, which implies that the string must be a DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of the ResourceClaim that was generated for the Pod in the namespace of the Pod. If this is unset, then generating a ResourceClaim was not necessary. The pod.spec.resourceClaims entry can be ignored in this case.

14.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/pods**
 - **GET**: list or watch objects of kind Pod

- **/api/v1/watch/pods**
 - **GET**: watch individual changes to a list of Pod. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/pods**
 - **DELETE**: delete collection of Pod
 - **GET**: list or watch objects of kind Pod
 - **POST**: create a Pod
- **/api/v1/watch/namespaces/{namespace}/pods**
 - **GET**: watch individual changes to a list of Pod. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/pods/{name}**
 - **DELETE**: delete a Pod
 - **GET**: read the specified Pod
 - **PATCH**: partially update the specified Pod
 - **PUT**: replace the specified Pod
- **/api/v1/namespaces/{namespace}/pods/{name}/log**
 - **GET**: read log of the specified Pod
- **/api/v1/watch/namespaces/{namespace}/pods/{name}**
 - **GET**: watch changes to an object of kind Pod. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/api/v1/namespaces/{namespace}/pods/{name}/resize**
 - **GET**: read resize of the specified Pod
 - **PATCH**: partially update resize of the specified Pod
 - **PUT**: replace resize of the specified Pod
- **/api/v1/namespaces/{namespace}/pods/{name}/status**
 - **GET**: read status of the specified Pod
 - **PATCH**: partially update status of the specified Pod
 - **PUT**: replace status of the specified Pod
- **/api/v1/namespaces/{namespace}/pods/{name}/ephemeralcontainers**
 - **GET**: read ephemeralcontainers of the specified Pod

- **PATCH**: partially update ephemeralcontainers of the specified Pod
- **PUT**: replace ephemeralcontainers of the specified Pod

14.2.1. /api/v1/pods

HTTP method

GET

Description

list or watch objects of kind Pod

Table 14.1. HTTP responses

HTTP code	Reponse body
200 - OK	PodList schema
401 - Unauthorized	Empty

14.2.2. /api/v1/watch/pods

HTTP method

GET

Description

watch individual changes to a list of Pod. deprecated: use the 'watch' parameter with a list operation instead.

Table 14.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

14.2.3. /api/v1/namespaces/{namespace}/pods

HTTP method

DELETE

Description

delete collection of Pod

Table 14.3. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 14.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Pod

Table 14.5. HTTP responses

HTTP code	Reponse body
200 - OK	PodList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Pod

Table 14.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.7. Body parameters

Parameter	Type	Description
body	Pod schema	

Table 14.8. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
202 - Accepted	Pod schema
401 - Unauthorized	Empty

14.2.4. /api/v1/watch/namespaces/{namespace}/pods

HTTP method

GET

Description

watch individual changes to a list of Pod. deprecated: use the 'watch' parameter with a list operation instead.

Table 14.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

14.2.5. /api/v1/namespaces/{namespace}/pods/{name}

Table 14.10. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method

DELETE

Description

delete a Pod

Table 14.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 14.12. HTTP responses

HTTP code	Reponse body
200 - OK	Pod schema
202 - Accepted	Pod schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Pod

Table 14.13. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified Pod

Table 14.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.15. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Pod

Table 14.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.17. Body parameters

Parameter	Type	Description
body	Pod schema	

Table 14.18. HTTP responses

HTTP code	Reponse body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

14.2.6. /api/v1/namespaces/{namespace}/pods/{name}/log

Table 14.19. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method

GET

Description

read log of the specified Pod

Table 14.20. HTTP responses

HTTP code	Reponse body
200 - OK	string
401 - Unauthorized	Empty

14.2.7. /api/v1/watch/namespaces/{namespace}/pods/{name}

Table 14.21. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method

GET

Description

watch changes to an object of kind Pod. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 14.22. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

14.2.8. /api/v1/namespaces/{namespace}/pods/{name}/resize

Table 14.23. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method

GET

Description

read resize of the specified Pod

Table 14.24. HTTP responses

HTTP code	Reponse body
200 - OK	Pod schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update resize of the specified Pod

Table 14.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.26. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace resize of the specified Pod

Table 14.27. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.28. Body parameters

Parameter	Type	Description
body	Pod schema	

Table 14.29. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

14.2.9. /api/v1/namespaces/{namespace}/pods/{name}/status

Table 14.30. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method**GET****Description**

read status of the specified Pod

Table 14.31. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified Pod

Table 14.32. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.33. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified Pod

Table 14.34. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.35. Body parameters

Parameter	Type	Description
body	Pod schema	

Table 14.36. HTTP responses

HTTP code	Reponse body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

14.2.10. /api/v1/namespaces/{namespace}/pods/{name}/ephemeralcontainers

Table 14.37. Global path parameters

Parameter	Type	Description
name	string	name of the Pod

HTTP method

GET

Description

read ephemeralcontainers of the specified Pod

Table 14.38. HTTP responses

HTTP code	Reponse body
200 - OK	Pod schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update ephemeralcontainers of the specified Pod

Table 14.39. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.40. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace ephemeralcontainers of the specified Pod

Table 14.41. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 14.42. Body parameters

Parameter	Type	Description
body	Pod schema	

Table 14.43. HTTP responses

HTTP code	Response body
200 - OK	Pod schema
201 - Created	Pod schema
401 - Unauthorized	Empty

CHAPTER 15. REPLICATIONCONTROLLER [V1]

Description

ReplicationController represents the configuration of a replication controller.

Type

object

15.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	If the Labels of a ReplicationController are empty, they are defaulted to be the same as the Pod(s) that the replication controller manages. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	ReplicationControllerSpec is the specification of a replication controller.

Property	Type	Description
status	object	ReplicationControllerStatus represents the current status of a replication controller.

15.1.1. .spec

Description

ReplicationControllerSpec is the specification of a replication controller.

Type

object

Property	Type	Description
minReadySeconds	integer	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
replicas	integer	Replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller
selector	object (string)	Selector is a label query over pods that should match the Replicas count. If Selector is empty, it is defaulted to the labels present on the Pod template. Label keys and values that must match in order to be controlled by this replication controller, if empty defaulted to labels on Pod template. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
template	object	PodTemplateSpec describes the data a pod should have when created from a template

15.1.2. .spec.template

Description

PodTemplateSpec describes the data a pod should have when created from a template

Type

object

Property	Type	Description
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec is a description of a pod.

15.1.3. .spec.template.spec

Description

PodSpec is a description of a pod.

Type

object

Required

- **containers**

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
affinity	object	Affinity is a group of affinity scheduling rules.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.

Property	Type	Description
containers	array	List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.
containers[]	object	A single application container that you want to run within a pod.
dnsConfig	object	PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Property	Type	Description
dnsPolicy	string	Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'. Possible enum values: - "ClusterFirst" indicates that the pod should use cluster DNS first unless hostNetwork is true, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "ClusterFirstWithHostNet" indicates that the pod should use cluster DNS first, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "Default" indicates that the pod should use the default (as determined by kubelet) DNS settings. - "None" indicates that the pod should use empty DNS settings. DNS parameters such as nameservers and search paths should be defined via DNSConfig.
enableServiceLinks	boolean	EnableServiceLinks indicates whether information about services should be injected into pod's environment variables, matching the syntax of Docker links. Optional: Defaults to true.

Property	Type	Description
ephemeralContainers	array	List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's <code>ephemeralcontainers</code> subresource.
ephemeralContainers[]	object	An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the <code>ephemeralcontainers</code> subresource of an existing Pod. Ephemeral containers may not be removed or restarted.
hostAliases	array	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
hostAliases[]	object	HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.
hostIPC	boolean	Use the host's ipc namespace. Optional: Default to false.

Property	Type	Description
hostNetwork	boolean	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.
hostPID	boolean	Use the host's pid namespace. Optional: Default to false.
hostUsers	boolean	Use the host's user namespace. Optional: Default to true. If set to true or not present, the pod will be run in the host user namespace, useful for when the pod needs a feature only available to the host user namespace, such as loading a kernel module with CAP_SYS_MODULE. When set to false, a new users is created for the pod. Setting false is useful for mitigating container breakout vulnerabilities even allowing users to run their containers as root without actually having root privileges on the host. This field is alpha-level and is only honored by servers that enable the UserNamespacesSupport feature.
hostname	string	Specifies the hostname of the Pod. If not specified, the pod's hostname will be set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
initContainers	array	List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/
initContainers[]	object	A single application container that you want to run within a pod.
nodeName	string	nodeName indicates in which node this pod is scheduled. If empty, this pod is a candidate for scheduling by the scheduler defined in schedulerName. Once this field is set, the kubelet for this node becomes responsible for the lifecycle of this pod. This field should not be used to express a desire for the pod to be scheduled on a specific node. https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodename

Property	Type	Description
nodeSelector	object (string)	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/
os	object	PodOS defines the OS parameters of a pod.
overhead	object (Quantity)	Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md
preemptionPolicy	string	PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset. Possible enum values: - "Never" means that pod never preempts other pods with lower priority. - "PreemptLowerPriority" means that pod can preempt other pods with lower priority.

Property	Type	Description
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it prevents users from setting this field. The admission controller populates this field from PriorityClassName. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-node-critical" and "system-cluster-critical" are two special keywords which indicate the highest priorities with the former being the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.
readinessGates	array	If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	object	PodReadinessGate contains the reference to a pod condition

Property	Type	Description
resourceClaims	array	ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable.
resourceClaims[]	object	PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.
resources	object	ResourceRequirements describes the compute resource requirements.
restartPolicy	string	Restart policy for all containers within the pod. One of Always, OnFailure, Never. In some contexts, only a subset of those values may be permitted. Default to Always. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy Possible enum values: - "Always" - "Never" - "OnFailure"

Property	Type	Description
runtimeClassName	string	RuntimeClassName refers to a RuntimeClass object in the node.k8s.io group, which should be used to run this pod. If no RuntimeClass resource matches the named class, the pod will not be run. If unset or empty, the "legacy" RuntimeClass will be used, which is an implicit class with an empty definition that uses the default runtime handler. More info: https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class
schedulerName	string	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
schedulingGates	array	SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.
schedulingGates[]	object	PodSchedulingGate is associated to a Pod to guard its scheduling.
securityContext	object	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Property	Type	Description
serviceAccount	string	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
serviceAccountName	string	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
setHostnameAsFQDN	boolean	If true the pod's hostname will be configured as the pod's FQDN, rather than the leaf name (the default). In Linux containers, this means setting the FQDN in the hostname field of the kernel (the nodename field of struct utsname). In Windows containers, this means setting the registry value of hostname for the registry key HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters to FQDN. If a pod does not have FQDN, this has no effect. Default to false.
shareProcessNamespace	boolean	Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false.
subdomain	string	If specified, the fully qualified Pod hostname will be "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>". If not specified, the pod will not have a domainname at all.

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.
tolerations	array	If specified, the pod's tolerations.
tolerations[]	object	The pod this Tolerant is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.
topologySpreadConstraints	array	TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.
topologySpreadConstraints[]	object	TopologySpreadConstraint specifies how to spread matching pods among the given topology.
volumes	array	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes
volumes[]	object	Volume represents a named volume in a pod that may be accessed by any container in the pod.

15.1.4. .spec.template.spec.affinity

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Node affinity is a group of node affinity scheduling rules.
podAffinity	object	Pod affinity is a group of inter pod affinity scheduling rules.
podAntiAffinity	object	Pod anti affinity is a group of inter pod anti affinity scheduling rules.

15.1.5. .spec.template.spec.affinity.nodeAffinity

Description

Node affinity is a group of node affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution[]	object	An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).
requiredDuringSchedulingIgnoredDuringExecution	object	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

15.1.6. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

15.1.7. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

- **weight**
- **preference**

Property	Type	Description

Property	Type	Description
preference	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

15.1.8. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

15.1.9. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

A list of node selector requirements by node's labels.

Type

array

15.1.10. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredD

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

15.1.11. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDu

Description

A list of node selector requirements by node's fields.

Type

array

15.1.12. .spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredD

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

15.1.13. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

Type

object

Required

- **nodeSelectorTerms**

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.
nodeSelectorTerms[]	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

15.1.14. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

15.1.15. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

15.1.16. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDu

Description

A list of node selector requirements by node's labels.

Type

array

15.1.17. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDui

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

15.1.18. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDui

Description

A list of node selector requirements by node's fields.

Type

array

15.1.19. .spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

15.1.20. .spec.template.spec.affinity.podAffinity

Description

Pod affinity is a group of inter pod affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	array	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

15.1.21. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

15.1.22. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

15.1.23. .spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDu

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

15.1.24. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

15.1.25. .spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDur

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

15.1.26. .spec.template.spec.affinity.podAntiAffinity

Description

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution	array	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

15.1.27. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

15.1.28. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

15.1.29. .spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnore

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

15.1.30. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

15.1.31. .spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set.
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

15.1.32. .spec.template.spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Type

array

15.1.33. .spec.template.spec.containers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.

Property	Type	Description
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

15.1.34. .spec.template.spec.containers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

15.1.35. .spec.template.spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

15.1.36. .spec.template.spec.containers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

15.1.37. .spec.template.spec.containers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

15.1.38. .spec.template.spec.containers[].env[].valueFrom.fieldRef**Description**

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

15.1.39. .spec.template.spec.containers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

15.1.40. .spec.template.spec.containers[].env[].valueFrom.secretKeyRef**Description**

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

15.1.41. .spec.template.spec.containers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

15.1.42. .spec.template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

15.1.43. .spec.template.spec.containers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

15.1.44. .spec.template.spec.containers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

15.1.45. .spec.template.spec.containers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

15.1.46. .spec.template.spec.containers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP socket action describes an action based on opening a socket

15.1.47. .spec.template.spec.containers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.48. .spec.template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.49. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.50. .spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.51. .spec.template.spec.containers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.52. .spec.template.spec.containers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.53. .spec.template.spec.containers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

15.1.54. .spec.template.spec.containers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.55. .spec.template.spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.56. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.57. .spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.58. .spec.template.spec.containers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.59. .spec.template.spec.containers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.60. .spec.template.spec.containers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.61. .spec.template.spec.containers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.62. .spec.template.spec.containers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.63. .spec.template.spec.containers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.64. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.65. .spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.66. .spec.template.spec.containers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.67. .spec.template.spec.containers[].ports**Description**

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

15.1.68. .spec.template.spec.containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

15.1.69. .spec.template.spec.containers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.70. .spec.template.spec.containers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.71. .spec.template.spec.containers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.72. .spec.template.spec.containers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.73. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.74. .spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.75. .spec.template.spec.containers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.76. .spec.template.spec.containers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

15.1.77. .spec.template.spec.containers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

15.1.78. .spec.template.spec.containers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

15.1.79. .spec.template.spec.containers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

15.1.80. .spec.template.spec.containers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

15.1.81. .spec.template.spec.containers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

15.1.82. .spec.template.spec.containers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

15.1.83. .spec.template.spec.containers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

15.1.84. .spec.template.spec.containers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

15.1.85. .spec.template.spec.containers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

15.1.86. .spec.template.spec.containers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

15.1.87. .spec.template.spec.containers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's <code>terminationGracePeriodSeconds</code> will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.88. .spec.template.spec.containers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.89. .spec.template.spec.containers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.90. .spec.template.spec.containers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.91. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.92. .spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.93. .spec.template.spec.containers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

▼ **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.94. .spec.template.spec.containers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

15.1.95. .spec.template.spec.containers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

15.1.96. .spec.template.spec.containers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

15.1.97. .spec.template.spec.containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':'.

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

15.1.98. .spec.template.spec.dnsConfig

Description

PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Type

object

Property	Type	Description
nameservers	array (string)	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
options	array	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
options[]	object	PodDNSConfigOption defines DNS resolver options of a pod.
searches	array (string)	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

15.1.99. .spec.template.spec.dnsConfig.options**Description**

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

array

15.1.100. .spec.template.spec.dnsConfig.options[]**Description**

PodDNSConfigOption defines DNS resolver options of a pod.

Type

object

Property	Type	Description
name	string	Name is this DNS resolver option's name. Required.
value	string	Value is this DNS resolver option's value.

15.1.101. .spec.template.spec.ephemeralContainers

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

array

15.1.102. .spec.template.spec.ephemeralContainers[]

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present

Property	Type	Description
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the ephemeral container specified as a DNS_LABEL. This name must be unique among all containers, init containers and ephemeral containers.
ports	array	Ports are not allowed for ephemeral containers.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	Restart policy for the container to manage the restart behavior of each container within a pod. This may only be set for init containers. You cannot set this field on ephemeral containers.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Property	Type	Description
targetContainerName	string	If set, the name of the container from PodSpec that this ephemeral container targets. The ephemeral container will be run in the namespaces (IPC, PID, etc) of this container. If not set then the ephemeral container uses the namespaces configured in the Pod spec. The container runtime must implement support for this feature. If the runtime does not support namespace targeting then the result of setting this field is undefined.
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to <code>/dev/termination-log</code>. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Property	Type	Description
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

15.1.103. .spec.template.spec.ephemeralContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

15.1.104. .spec.template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

15.1.105. .spec.template.spec.ephemeralContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

15.1.106. .spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyR

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

15.1.107. .spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

15.1.108. .spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRe

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

15.1.109. .spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

15.1.110. .spec.template.spec.ephemeralContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

15.1.111. .spec.template.spec.ephemeralContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

15.1.112. .spec.template.spec.ephemeralContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

15.1.113. .spec.template.spec.ephemeralContainers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

15.1.114. .spec.template.spec.ephemeralContainers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type
object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

15.1.115. .spec.template.spec.ephemeralContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP socket action describes an action based on opening a socket

15.1.116. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.117. `.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet`

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.118. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpH

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.119. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpH

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.120. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.121. .spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.122. .spec.template.spec.ephemeralContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

15.1.123. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.124. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.125. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHe

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.126. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHe

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.127. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.128. .spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.129. `.spec.template.spec.ephemeralContainers[].livenessProbe`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.130. .spec.template.spec.ephemeralContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.131. .spec.template.spec.ephemeralContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.132. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.133. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHead**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.134. .spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHead**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.135. .spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.136. .spec.template.spec.ephemeralContainers[].ports**Description**

Ports are not allowed for ephemeral containers.

Type

array

15.1.137. .spec.template.spec.ephemeralContainers[].ports[]**Description**

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

15.1.138. .spec.template.spec.ephemeralContainers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.139. .spec.template.spec.ephemeralContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.140. .spec.template.spec.ephemeralContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.141. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.142. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHea**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.143. .spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHea**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.144. .spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.145. .spec.template.spec.ephemeralContainers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

15.1.146. .spec.template.spec.ephemeralContainers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

15.1.147. .spec.template.spec.ephemeralContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

15.1.148. `.spec.template.spec.ephemeralContainers[].resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate.

This field is immutable. It can only be set for containers.

Type

array

15.1.149. `.spec.template.spec.ephemeralContainers[].resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

15.1.150. .spec.template.spec.ephemeralContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

15.1.151. .spec.template.spec.ephemeralContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

15.1.152. .spec.template.spec.ephemeralContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

15.1.153. .spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

15.1.154. .spec.template.spec.ephemeralContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

15.1.155. .spec.template.spec.ephemeralContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

15.1.156. .spec.template.spec.ephemeralContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.157. .spec.template.spec.ephemeralContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.158. .spec.template.spec.ephemeralContainers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.159. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.160. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeader

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.161. .spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeader

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.162. .spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

▼ **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.163. .spec.template.spec.ephemeralContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

15.1.164. .spec.template.spec.ephemeralContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

15.1.165. .spec.template.spec.ephemeralContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

array

15.1.166. .spec.template.spec.ephemeralContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

15.1.167. .spec.template.spec.hostAliases

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

array

15.1.168. .spec.template.spec.hostAliases[]

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

- **ip**

Property	Type	Description
hostnames	array (string)	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

15.1.169. .spec.template.spec.imagePullSecrets

Description

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

15.1.170. .spec.template.spec.imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.171. .spec.template.spec.initContainers

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

15.1.172. .spec.template.spec.initContainers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
args	array (string)	Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal <code>"\$(VAR_NAME)"</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
command	array (string)	Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal <code>"\$(VAR_NAME)"</code>. Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell
env	array	List of environment variables to set in the container. Cannot be updated.

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps or Secrets
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated. Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.

Property	Type	Description
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

15.1.173. .spec.template.spec.initContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

15.1.174. .spec.template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

15.1.175. .spec.template.spec.initContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

15.1.176. .spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

15.1.177. .spec.template.spec.initContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

15.1.178. .spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

15.1.179. .spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

15.1.180. .spec.template.spec.initContainers[].envFrom**Description**

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

15.1.181. .spec.template.spec.initContainers[].envFrom[]**Description**

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

object

Property	Type	Description
configMapRef	object	ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
prefix	string	Optional text to prepend to the name of each environment variable. Must be a C_IDENTIFIER.
secretRef	object	SecretEnvSource selects a Secret to populate the environment variables with. The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

15.1.182. .spec.template.spec.initContainers[].envFrom[].configMapRef**Description**

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

15.1.183. .spec.template.spec.initContainers[].envFrom[].secretRef**Description**

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type**object**

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

15.1.184. .spec.template.spec.initContainers[].lifecycle**Description**

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type
object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
stopSignal	string	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name Possible enum values: - "SIGABRT" - "SIGALRM" - "SIGBUS" - "SIGCHLD" - "SIGCLD" - "SIGCONT" - "SIGFPE" - "SIGHUP" - "SIGILL" - "SIGINT" - "SIGIO" - "SIGIOT" - "SIGKILL" - "SIGPIPE" - "SIGPOLL" - "SIGPROF" - "SIGPWR" - "SIGQUIT" - "SIGRTMAX" - "SIGRTMAX-1" - "SIGRTMAX-10" - "SIGRTMAX-11" - "SIGRTMAX-12" - "SIGRTMAX-13" - "SIGRTMAX-14" - "SIGRTMAX-2" - "SIGRTMAX-3" - "SIGRTMAX-4" - "SIGRTMAX-5" - "SIGRTMAX-6" - "SIGRTMAX-7" - "SIGRTMAX-8" - "SIGRTMAX-9" - "SIGRTMIN" - "SIGRTMIN+1" - "SIGRTMIN+10" -

Property	Type	Description
		"SIGRTMIN+11" - "SIGRTMIN+12" - "SIGRTMIN+13" - "SIGRTMIN+14" - "SIGRTMIN+15" - "SIGRTMIN+2" - "SIGRTMIN+3" - "SIGRTMIN+4" - "SIGRTMIN+5" - "SIGRTMIN+6" - "SIGRTMIN+7" - "SIGRTMIN+8" - "SIGRTMIN+9" - "SIGSEGV" - "SIGSTKFLT" - "SIGSTOP" - "SIGSYS" - "SIGTERM" - "SIGTRAP" - "SIGTSTP" - "SIGTTIN" - "SIGTTOU" - "SIGURG" - "SIGUSR1" - "SIGUSR2" - "SIGVTALRM" - "SIGWINCH" - "SIGXCPU" - "SIGXFSZ"

15.1.185. .spec.template.spec.initContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP socket action describes an action based on opening a socket

15.1.186. .spec.template.spec.initContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type
object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.187. .spec.template.spec.initContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type
object

Required

- port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.188. `.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.189. `.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.190. `.spec.template.spec.initContainers[].lifecycle.postStart.sleep`

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.191. .spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.192. .spec.template.spec.initContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

15.1.193. .spec.template.spec.initContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.194. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.195. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.196. .spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.197. .spec.template.spec.initContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

15.1.198. .spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.199. .spec.template.spec.initContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.200. .spec.template.spec.initContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.201. .spec.template.spec.initContainers[].livenessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.202. .spec.template.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.203. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.204. .spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.205. .spec.template.spec.initContainers[].livenessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.206. .spec.template.spec.initContainers[].ports**Description**

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

15.1.207. .spec.template.spec.initContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

15.1.208. .spec.template.spec.initContainers[].readinessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.209. .spec.template.spec.initContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.210. .spec.template.spec.initContainers[].readinessProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.211. .spec.template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.212. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.213. .spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.214. .spec.template.spec.initContainers[].readinessProbe.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.215. .spec.template.spec.initContainers[].resizePolicy**Description**

Resources resize policy for the container.

Type

array

15.1.216. .spec.template.spec.initContainers[].resizePolicy[]**Description**

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

15.1.217. .spec.template.spec.initContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

15.1.218. .spec.template.spec.initContainers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

15.1.219. .spec.template.spec.initContainers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

15.1.220. `.spec.template.spec.initContainers[].securityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.

Property	Type	Description
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

15.1.221. .spec.template.spec.initContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

15.1.222. .spec.template.spec.initContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

15.1.223. .spec.template.spec.initContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

15.1.224. .spec.template.spec.initContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

15.1.225. .spec.template.spec.initContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

15.1.226. .spec.template.spec.initContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPCAction specifies an action involving a GRPC service.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

15.1.227. .spec.template.spec.initContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

15.1.228. .spec.template.spec.initContainers[].startupProbe.grpc

Description

GRPCAction specifies an action involving a GRPC service.

Type

object

Required

- **port**

Property	Type	Description

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

15.1.229. .spec.template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

15.1.230. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

15.1.231. .spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

15.1.232. .spec.template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

15.1.233. .spec.template.spec.initContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

15.1.234. .spec.template.spec.initContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

15.1.235. .spec.template.spec.initContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

15.1.236. `.spec.template.spec.initContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None). Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
subPath	string	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
subPathExpr	string	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

15.1.237. .spec.template.spec.os

Description

PodOS defines the OS parameters of a pod.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration Clients should expect to handle additional values and treat unrecognized values in this field as os: null

15.1.238. .spec.template.spec.readinessGates**Description**

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True"

More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type

array

15.1.239. .spec.template.spec.readinessGates[]**Description**

PodReadinessGate contains the reference to a pod condition

Type

object

Required

- **conditionType**

Property	Type	Description
conditionType	string	ConditionType refers to a condition in the pod's condition list with matching type.

15.1.240. .spec.template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name.

This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable.

Type

array

15.1.241. .spec.template.spec.resourceClaims[]

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod.

It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name uniquely identifies this resource claim inside the pod. This must be a DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

Property	Type	Description
resourceClaimTemplateName	string	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in <code>pod.status.resourceClaimStatus</code>. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

15.1.242. .spec.template.spec.resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

15.1.243. .spec.template.spec.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

15.1.244. .spec.template.spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

15.1.245. .spec.template.spec.schedulingGates

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod.

SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

15.1.246. .spec.template.spec.schedulingGates[]

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

- **name**

Property	Type	Description

Property	Type	Description
name	string	Name of the scheduling gate. Each scheduling gate must have a unique name field.

15.1.247. .spec.template.spec.securityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership (and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Always" indicates that volume's ownership and permissions should always be changed whenever volume is mounted inside a Pod. This the default behavior. - "OnRootMismatch" indicates that volume's ownership and permissions will be changed only when permission and ownership of root directory does not match with expected permissions on the volume. This can help shorten the time it takes to change ownership and permissions of a volume.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with -o context mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible

Property	Type	Description
		volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when <code>spec.os.name</code> is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Property	Type	Description
supplementalGroups	array (integer)	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows. Possible enum values: - "Merge" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be merged with the primary user's groups as defined in the container image (in /etc/group). - "Strict" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be used instead of any groups defined in the container image.

Property	Type	Description
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
sysctls[]	object	Sysctl defines a kernel parameter to be set
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

15.1.248. .spec.template.spec.securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".

Property	Type	Description
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement. Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.

15.1.249. .spec.template.spec.securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

15.1.250. .spec.template.spec.securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied. Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).

15.1.251. .spec.template.spec.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

15.1.252. .spec.template.spec.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

15.1.253. .spec.template.spec.securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

15.1.254. .spec.template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

array

15.1.255. .spec.template.spec.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute. Possible enum values: - "NoExecute" Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController. - "NoSchedule" Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler. - "PreferNoSchedule" Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Possible enum values: - "Equal" - "Exists"

Property	Type	Description
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	string	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

15.1.256. .spec.template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

15.1.257. .spec.template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

Property	Type	Description

Property	Type	Description
labelSelector	LabelSelector	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the <code>MatchLabelKeysInPodTopologySpread</code> feature gate to be enabled (enabled by default).

Property	Type	Description
maxSkew	integer	MaxSkew describes the degree to which pods may be unevenly distributed. When whenUnsatisfiable=DoNotSchedule, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When whenUnsatisfiable=ScheduleAnyway, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.

Property	Type	Description
minDomains	integer	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule. For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.

Property	Type	Description
nodeAffinityPolicy	string	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are: - Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy. Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.
nodeTaintsPolicy	string	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included. If this value is nil, the behavior is equivalent to the Ignore policy. Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.

Property	Type	Description
topologyKey	string	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket. We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.

Property	Type	Description
whenUnsatisfiable	string	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it more imbalanced. It's a required field. Possible enum values: - "DoNotSchedule" instructs the scheduler not to schedule the pod when constraints are not satisfied. - "ScheduleAnyway" instructs the scheduler to schedule the pod even if constraints are not satisfied.

15.1.258. .spec.template.spec.volumes

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

15.1.259. .spec.template.spec.volumes[]

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

object

Required

- **name**

Property	Type	Description
awsElasticBlockStore	object	Represents a Persistent Disk resource in AWS. An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.
azureDisk	object	AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
azureFile	object	AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
cephfs	object	Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.
cinder	object	Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Property	Type	Description
configMap	object	Adapts a ConfigMap into a volume. The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.
csi	object	Represents a source location of a volume to mount, managed by an external CSI driver
downwardAPI	object	DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
emptyDir	object	Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
ephemeral	object	Represents an ephemeral volume that is handled by a normal storage driver.
fc	object	Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.
flexVolume	object	FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Property	Type	Description
flocker	object	Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.
gcePersistentDisk	object	Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.
gitRepo	object	Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.
glusterfs	object	Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
hostPath	object	Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
image	object	ImageVolumeSource represents a image volume resource.

Property	Type	Description
iscsi	object	Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.
name	string	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).
photonPersistentDisk	object	Represents a Photon Controller persistent disk resource.
portworxVolume	object	PortworxVolumeSource represents a Portworx volume resource.
projected	object	Represents a projected volume source
quobyte	object	Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Property	Type	Description
rbd	object	Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
scaleIO	object	ScaleIOVolumeSource represents a persistent ScaleIO volume
secret	object	Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
storageos	object	Represents a StorageOS persistent volume resource.
vsphereVolume	object	Represents a vSphere volume resource.

15.1.260. .spec.template.spec.volumes[].awsElasticBlockStore

Description

Represents a Persistent Disk resource in AWS.

An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

Type

object

Required

- **volumeID**

Property	Type	Description

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

15.1.261. .spec.template.spec.volumes[].azureDisk

Description

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

- **diskName**
- **diskURI**

Property	Type	Description
cachingMode	string	cachingMode is the Host Caching mode: None, Read Only, Read Write. Possible enum values: - "None" - "ReadOnly" - "ReadWrite"
diskName	string	diskName is the Name of the data disk in the blob storage
diskURI	string	diskURI is the URI of data disk in the blob storage
fsType	string	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
kind	string	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared Possible enum values: - "Dedicated" - "Managed" - "Shared"
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

15.1.262. .spec.template.spec.volumes[].azureFile

Description

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

object

Required

- **secretName**
- **shareName**

Property	Type	Description
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretName	string	secretName is the name of secret that contains Azure Storage Account Name and Key
shareName	string	shareName is the azure share Name

15.1.263. .spec.template.spec.volumes[].cephfs

Description

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **monitors**

Property	Type	Description
monitors	array (string)	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Property	Type	Description
secretFile	string	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

15.1.264. .spec.template.spec.volumes[].cephfs.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.265. .spec.template.spec.volumes[].cinder

Description

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Type

object

Required

- **volumeID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeID	string	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

15.1.266. .spec.template.spec.volumes[].cinder.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.267. .spec.template.spec.volumes[].configMap

Description

Adapts a ConfigMap into a volume.

The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

15.1.268. .spec.template.spec.volumes[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

15.1.269. .spec.template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type**object****Required**

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

15.1.270. .spec.template.spec.volumes[].csi**Description**

Represents a source location of a volume to mount, managed by an external CSI driver

Type**object****Required**

- **driver**

Property	Type	Description

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
volumeAttributes	object (string)	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

15.1.271. .spec.template.spec.volumes[.csi.nodePublishSecretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.272. .spec.template.spec.volumes[].downwardAPI

Description

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

15.1.273. .spec.template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

15.1.274. .spec.template.spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

15.1.275. .spec.template.spec.volumes[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
fieldPath	string	Path of the field to select in the specified API version.

15.1.276. .spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

15.1.277. .spec.template.spec.volumes[].emptyDir

Description

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
sizeLimit	Quantity	sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

15.1.278. .spec.template.spec.volumes[].ephemeral

Description

Represents an ephemeral volume that is handled by a normal storage driver.

Type

object

Property	Type	Description
volumeClaimTemplate	object	PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

15.1.279. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

Type

object

Required

- **spec**

Property	Type	Description
metadata	ObjectMeta	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
spec	object	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

15.1.280. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

Type

object

Property	Type	Description
accessModes	array (string)	accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.
dataSourceRef	object	TypedObjectReference contains enough information to let you locate the typed referenced object
resources	object	VolumeResourceRequirements describes the storage resource requirements for a volume.
selector	LabelSelector	selector is a label query over volumes to consider for binding.

Property	Type	Description
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).

Property	Type	Description
volumeMode	string	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec. Possible enum values: - "Block" means the volume will not be formatted with a filesystem and will remain a raw block device. - "Filesystem" means the volume will be or is formatted with a filesystem.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

15.1.281. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSo

Description

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

15.1.282. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource**Description**

TypedObjectReference contains enough information to let you locate the typed referenced object

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

15.1.283. .spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resourceRequirements**Description**

VolumeResourceRequirements describes the storage resource requirements for a volume.

Type

object

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

15.1.284. .spec.template.spec.volumes[].fc

Description

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	integer	lun is Optional: FC target lun number
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	array (string)	targetWWNs is Optional: FC target worldwide names (WWNs)

Property	Type	Description
wwids	array (string)	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

15.1.285. .spec.template.spec.volumes[].flexVolume

Description

FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type

object

Required

- **driver**

Property	Type	Description
driver	string	driver is the name of the driver to use for this volume.
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	object (string)	options is Optional: this field holds extra command options if any.
readOnly	boolean	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

15.1.286. .spec.template.spec.volumes[].flexVolume.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.287. .spec.template.spec.volumes[].flocker**Description**

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

Type

object

Property	Type	Description
datasetName	string	datasetName is Name of the dataset stored as metadata → name on the dataset for Flocker should be considered as deprecated
datasetUUID	string	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

15.1.288. .spec.template.spec.volumes[].gcePersistentDisk**Description**

Represents a Persistent Disk resource in Google Compute Engine.

A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

Type

object**Required**

- **pdName**

Property	Type	Description
fsType	string	fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	string	pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

15.1.289. .spec.template.spec.volumes[].gitRepo**Description**

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

object

Required

- **repository**

Property	Type	Description
directory	string	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
repository	string	repository is the URL
revision	string	revision is the commit hash for the specified revision.

15.1.290. .spec.template.spec.volumes[].glusterfs

Description

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **endpoints**
- **path**

Property	Type	Description

Property	Type	Description
endpoints	string	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
readOnly	boolean	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

15.1.291. .spec.template.spec.volumes[].hostPath

Description

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **path**

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Property	Type	Description
type	string	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath Possible enum values: - "" For backwards compatible, leave it empty if unset - "BlockDevice" A block device must exist at the given path - "CharDevice" A character device must exist at the given path - "Directory" A directory must exist at the given path - "DirectoryOrCreate" If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. - "File" A file must exist at the given path - "FileOrCreate" If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. - "Socket" A UNIX socket must exist at the given path

15.1.292. .spec.template.spec.volumes[].image

Description

ImageVolumeSource represents a image volume resource.

Type

object

Property	Type	Description

Property	Type	Description
pullPolicy	string	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present
reference	string	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

15.1.293. .spec.template.spec.volumes[].iscsi

Description

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

Type

object

Required

- **targetPortal**
- **iqn**
- **lun**

Property	Type	Description
chapAuthDiscovery	boolean	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	boolean	chapAuthSession defines whether support iSCSI Session CHAP authentication
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.
iqn	string	iqn is the target iSCSI Qualified Name.
iscsiInterface	string	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

Property	Type	Description
lun	integer	lun represents iSCSI Target Lun number.
portals	array (string)	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
targetPortal	string	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

15.1.294. .spec.template.spec.volumes[].iscsi.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.295. .spec.template.spec.volumes[].nfs

Description

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **server**
- **path**

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

15.1.296. .spec.template.spec.volumes[].persistentVolumeClaim**Description**

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A

PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

Type

object

Required

- **claimName**

Property	Type	Description

Property	Type	Description
claimName	string	claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	boolean	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

15.1.297. .spec.template.spec.volumes[].photonPersistentDisk

Description

Represents a Photon Controller persistent disk resource.

Type

object

Required

- **pdID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

15.1.298. .spec.template.spec.volumes[].portworxVolume

Description

PortworxVolumeSource represents a Portworx volume resource.

Type

object

Required

- **volumeID**

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

15.1.299. .spec.template.spec.volumes[].projected

Description

Represents a projected volume source

Type

object

Property	Type	Description
defaultMode	integer	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
sources	array	sources is the list of volume projections. Each entry in this list handles one source.
sources[]	object	Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

15.1.300. .spec.template.spec.volumes[].projected.sources

Description

sources is the list of volume projections. Each entry in this list handles one source.

Type

array

15.1.301. .spec.template.spec.volumes[].projected.sources[]

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

object

Property	Type	Description
clusterTrustBundle	object	ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.
configMap	object	Adapts a ConfigMap into a projected volume. The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.
downwardAPI	object	Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Property	Type	Description
secret	object	Adapts a secret into a projected volume. The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.
serviceAccountToken	object	ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

15.1.302. .spec.template.spec.volumes[].projected.sources[].clusterTrustBundle

Description

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

Type

object

Required

- **path**

Property	Type	Description
labelSelector	LabelSelector	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

Property	Type	Description
name	string	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	boolean	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	string	Relative path from the volume root to write the bundle.
signerName	string	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

15.1.303. .spec.template.spec.volumes[].projected.sources[].configMap

Description

Adapts a ConfigMap into a projected volume.

The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

Type

object

Property	Type	Description

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

15.1.304. .spec.template.spec.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

15.1.305. .spec.template.spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

15.1.306. .spec.template.spec.volumes[].projected.sources[].downwardAPI

Description

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Type

object

Property	Type	Description
items	array	Items is a list of DownwardAPIVolume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

15.1.307. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items**Description**

Items is a list of DownwardAPIVolume file

Type

array

15.1.308. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]**Description**

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

15.1.309. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

15.1.310. .spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].reso**Description**

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

15.1.311. .spec.template.spec.volumes[].projected.sources[].secret**Description**

Adapts a secret into a projected volume.

The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional field specify whether the Secret or its key must be defined

15.1.312. .spec.template.spec.volumes[].projected.sources[].secret.items**Description**

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array**15.1.313. .spec.template.spec.volumes[].projected.sources[].secret.items[]****Description**

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

15.1.314. .spec.template.spec.volumes[].projected.sources[].serviceAccountToken

Description

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

Type

object

Required

- **path**

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours. Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

15.1.315. .spec.template.spec.volumes[].quobyte

Description

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **registry**
- **volume**

Property	Type	Description
group	string	group to map volume access to Default is no group

Property	Type	Description
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

15.1.316. .spec.template.spec.volumes[].rbd

Description

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

Type

object

Required

- **monitors**
- **image**

Property	Type	Description

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	array (string)	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

15.1.317. .spec.template.spec.volumes[].rbd.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.318. .spec.template.spec.volumes[].scaleIO

Description

ScaleIOVolumeSource represents a persistent ScaleIO volume

Type

object

Required

- **gateway**
- **system**
- **secretRef**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	string	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.

Property	Type	Description
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
sslEnabled	boolean	sslEnabled Flag enable/disable SSL communication with Gateway, default false
storageMode	string	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
storagePool	string	storagePool is the ScaleIO Storage Pool associated with the protection domain.
system	string	system is the name of the storage system as configured in ScaleIO.
volumeName	string	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

15.1.319. .spec.template.spec.volumes[].scaleIO.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.320. .spec.template.spec.volumes[].secret

Description

Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
optional	boolean	optional field specify whether the Secret or its keys must be defined
secretName	string	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

15.1.321. .spec.template.spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

15.1.322. .spec.template.spec.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

15.1.323. .spec.template.spec.volumes[].storageos

Description

Represents a StorageOS persistent volume resource.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeName	string	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
volumeNamespace	string	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

15.1.324. .spec.template.spec.volumes[].storageos.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

15.1.325. .spec.template.spec.volumes[].vsphereVolume

Description

Represents a vSphere volume resource.

Type

object

Required

- **volumePath**

Property	Type	Description
fsType	string	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	string	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	string	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	string	volumePath is the path that identifies vSphere volume vmdk

15.1.326. .status**Description**

ReplicationControllerStatus represents the current status of a replication controller.

Type

object

Required

- **replicas**

Property	Type	Description
availableReplicas	integer	The number of available replicas (ready for at least minReadySeconds) for this replication controller.

Property	Type	Description
conditions	array	Represents the latest available observations of a replication controller's current state.
conditions[]	object	ReplicationControllerCondition describes the state of a replication controller at a certain point.
fullyLabeledReplicas	integer	The number of pods that have labels matching the labels of the pod template of the replication controller.
observedGeneration	integer	ObservedGeneration reflects the generation of the most recently observed replication controller.
readyReplicas	integer	The number of ready replicas for this replication controller.
replicas	integer	Replicas is the most recently observed number of replicas. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller#what-is-a-replicationcontroller

15.1.327. .status.conditions

Description

Represents the latest available observations of a replication controller's current state.

Type

array

15.1.328. .status.conditions[]

Description

ReplicationControllerCondition describes the state of a replication controller at a certain point.

Type

object

Required

- **type**

- **status**

Property	Type	Description
lastTransitionTime	Time	The last time the condition transitioned from one status to another.
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of replication controller condition.

15.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/replicationcontrollers**
 - **GET**: list or watch objects of kind ReplicationController
- **/api/v1/watch/replicationcontrollers**
 - **GET**: watch individual changes to a list of ReplicationController. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/replicationcontrollers**
 - **DELETE**: delete collection of ReplicationController
 - **GET**: list or watch objects of kind ReplicationController
 - **POST**: create a ReplicationController
- **/api/v1/watch/namespaces/{namespace}/replicationcontrollers**
 - **GET**: watch individual changes to a list of ReplicationController. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}**
 - **DELETE**: delete a ReplicationController
 - **GET**: read the specified ReplicationController

- **PATCH**: partially update the specified ReplicationController
- **PUT**: replace the specified ReplicationController
- **/api/v1/watch/namespaces/{namespace}/replicationcontrollers/{name}**
 - **GET**: watch changes to an object of kind ReplicationController. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status**
 - **GET**: read status of the specified ReplicationController
 - **PATCH**: partially update status of the specified ReplicationController
 - **PUT**: replace status of the specified ReplicationController

15.2.1. /api/v1/replicationcontrollers

HTTP method

GET

Description

list or watch objects of kind ReplicationController

Table 15.1. HTTP responses

HTTP code	Response body
200 - OK	ReplicationControllerList schema
401 - Unauthorized	Empty

15.2.2. /api/v1/watch/replicationcontrollers

HTTP method

GET

Description

watch individual changes to a list of ReplicationController. deprecated: use the 'watch' parameter with a list operation instead.

Table 15.2. HTTP responses

HTTP code	Response body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

15.2.3. /api/v1/namespaces/{namespace}/replicationcontrollers

HTTP method

DELETE

Description

delete collection of ReplicationController

Table 15.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 15.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind ReplicationController

Table 15.5. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicationControllerList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a ReplicationController

Table 15.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 15.7. Body parameters

Parameter	Type	Description
body	ReplicationController schema	

Table 15.8. HTTP responses

HTTP code	Response body
200 - OK	ReplicationController schema
201 - Created	ReplicationController schema
202 - Accepted	ReplicationController schema
401 - Unauthorized	Empty

15.2.4. /api/v1/watch/namespaces/{namespace}/replicationcontrollers

HTTP method

GET

Description

watch individual changes to a list of ReplicationController. deprecated: use the 'watch' parameter with a list operation instead.

Table 15.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

15.2.5. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}

Table 15.10. Global path parameters

Parameter	Type	Description
name	string	name of the ReplicationController

HTTP method

DELETE

Description

delete a ReplicationController

Table 15.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 15.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema

HTTP code	Reponse body
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified ReplicationController

Table 15.13. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicationController schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified ReplicationController

Table 15.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 15.15. HTTP responses

HTTP code	Response body
200 - OK	ReplicationController schema
201 - Created	ReplicationController schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified ReplicationController

Table 15.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 15.17. Body parameters

Parameter	Type	Description
body	ReplicationController schema	

Table 15.18. HTTP responses

HTTP code	Response body
200 - OK	ReplicationController schema
201 - Created	ReplicationController schema
401 - Unauthorized	Empty

15.2.6. /api/v1/watch/namespaces/{namespace}/replicationcontrollers/{name}

Table 15.19. Global path parameters

Parameter	Type	Description
name	string	name of the ReplicationController

HTTP method**GET****Description**

watch changes to an object of kind ReplicationController. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 15.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

15.2.7. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status**Table 15.21. Global path parameters**

Parameter	Type	Description
name	string	name of the ReplicationController

HTTP method**GET****Description**

read status of the specified ReplicationController

Table 15.22. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicationController schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified ReplicationController

Table 15.23. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 15.24. HTTP responses

HTTP code	Response body
200 - OK	ReplicationController schema
201 - Created	ReplicationController schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified ReplicationController

Table 15.25. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 15.26. Body parameters

Parameter	Type	Description
body	ReplicationController schema	

Table 15.27. HTTP responses

HTTP code	Response body
200 - OK	ReplicationController schema
201 - Created	ReplicationController schema
401 - Unauthorized	Empty

CHAPTER 16. REPLICASET [APPS/V1]

Description

ReplicaSet ensures that a specified number of pod replicas are running at any given time.

Type

object

16.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	If the Labels of a ReplicaSet are empty, they are defaulted to be the same as the Pod(s) that the ReplicaSet manages. Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	ReplicaSetSpec is the specification of a ReplicaSet.
status	object	ReplicaSetStatus represents the current status of a ReplicaSet.

16.1.1. .spec

Description

ReplicaSetSpec is the specification of a ReplicaSet.

Type

object

Required

- **selector**

Property	Type	Description
minReadySeconds	integer	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
replicas	integer	Replicas is the number of desired pods. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicaset
selector	LabelSelector	Selector is a label query over pods that should match the replica count. Label keys and values that must match in order to be controlled by this replica set. It must match the pod template's labels. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
template	PodTemplateSpec	Template is the object that describes the pod that will be created if insufficient replicas are detected. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/#pod-template

16.1.2. .status

Description

ReplicaSetStatus represents the current status of a ReplicaSet.

Type

object

Required

- **replicas**

Property	Type	Description
availableReplicas	integer	The number of available non-terminating pods (ready for at least minReadySeconds) for this replica set.
conditions	array	Represents the latest available observations of a replica set's current state.
conditions[]	object	ReplicaSetCondition describes the state of a replica set at a certain point.
fullyLabeledReplicas	integer	The number of non-terminating pods that have labels matching the labels of the pod template of the replicaset.
observedGeneration	integer	ObservedGeneration reflects the generation of the most recently observed ReplicaSet.
readyReplicas	integer	The number of non-terminating pods targeted by this ReplicaSet with a Ready Condition.
replicas	integer	Replicas is the most recently observed number of non-terminating pods. More info: https://kubernetes.io/docs/concepts/workloads/controllers/replicaset

Property	Type	Description
terminatingReplicas	integer	The number of terminating pods for this replica set. Terminating pods have a non-null <code>.metadata.deletionTimestamp</code> and have not yet reached the <code>Failed</code> or <code>Succeeded</code> <code>.status.phase</code>. This is an alpha field. Enable <code>DeploymentReplicaSetTerminatingReplicas</code> to be able to use this field.

16.1.3. `.status.conditions`

Description

Represents the latest available observations of a replica set's current state.

Type

array

16.1.4. `.status.conditions[]`

Description

`ReplicaSetCondition` describes the state of a replica set at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	The last time the condition transitioned from one status to another.
message	string	A human readable message indicating details about the transition.

Property	Type	Description
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of replica set condition.

16.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps/v1/replicasets**
 - **GET**: list or watch objects of kind ReplicaSet
- **/apis/apps/v1/watch/replicasets**
 - **GET**: watch individual changes to a list of ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/replicasets**
 - **DELETE**: delete collection of ReplicaSet
 - **GET**: list or watch objects of kind ReplicaSet
 - **POST**: create a ReplicaSet
- **/apis/apps/v1/watch/namespaces/{namespace}/replicasets**
 - **GET**: watch individual changes to a list of ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}**
 - **DELETE**: delete a ReplicaSet
 - **GET**: read the specified ReplicaSet
 - **PATCH**: partially update the specified ReplicaSet
 - **PUT**: replace the specified ReplicaSet
- **/apis/apps/v1/watch/namespaces/{namespace}/replicasets/{name}**
 - **GET**: watch changes to an object of kind ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}/status**
 - **GET**: read status of the specified ReplicaSet

- **PATCH**: partially update status of the specified ReplicaSet
- **PUT**: replace status of the specified ReplicaSet

16.2.1. /apis/apps/v1/replicaset

HTTP method

GET

Description

list or watch objects of kind ReplicaSet

Table 16.1. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicaSetList schema
401 - Unauthorized	Empty

16.2.2. /apis/apps/v1/watch/replicaset

HTTP method

GET

Description

watch individual changes to a list of ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 16.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

16.2.3. /apis/apps/v1/namespaces/{namespace}/replicaset

HTTP method

DELETE

Description

delete collection of ReplicaSet

Table 16.3. Query parameters

Parameter	Type	Description

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 16.4. HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind ReplicaSet

Table 16.5. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSetList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a ReplicaSet

Table 16.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 16.7. Body parameters

Parameter	Type	Description
body	ReplicaSet schema	

Table 16.8. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSet schema
201 - Created	ReplicaSet schema
202 - Accepted	ReplicaSet schema
401 - Unauthorized	Empty

16.2.4. /apis/apps/v1/watch/namespaces/{namespace}/replicasets

HTTP method

GET

Description

watch individual changes to a list of ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 16.9. HTTP responses

HTTP code	Response body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

16.2.5. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}

Table 16.10. Global path parameters

Parameter	Type	Description
name	string	name of the ReplicaSet

HTTP method

DELETE

Description

delete a ReplicaSet

Table 16.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 16.12. HTTP responses

HTTP code	Response body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ReplicaSet

Table 16.13. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSet schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified ReplicaSet

Table 16.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 16.15. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSet schema
201 - Created	ReplicaSet schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified ReplicaSet

Table 16.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 16.17. Body parameters

Parameter	Type	Description
body	ReplicaSet schema	

Table 16.18. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicaSet schema
201 - Created	ReplicaSet schema
401 - Unauthorized	Empty

16.2.6. /apis/apps/v1/watch/namespaces/{namespace}/replicaset/{name}

Table 16.19. Global path parameters

Parameter	Type	Description
name	string	name of the ReplicaSet

HTTP method

GET

Description

watch changes to an object of kind ReplicaSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 16.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

16.2.7. /apis/apps/v1/namespaces/{namespace}/replicaset/{name}/status

Table 16.21. Global path parameters

Parameter	Type	Description
name	string	name of the ReplicaSet

HTTP method

GET

Description

read status of the specified ReplicaSet

Table 16.22. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSet schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified ReplicaSet

Table 16.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 16.24. HTTP responses

HTTP code	Response body
200 - OK	ReplicaSet schema
201 - Created	ReplicaSet schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified ReplicaSet

Table 16.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 16.26. Body parameters

Parameter	Type	Description
body	ReplicaSet schema	

Table 16.27. HTTP responses

HTTP code	Reponse body
200 - OK	ReplicaSet schema
201 - Created	ReplicaSet schema
401 - Unauthorized	Empty

CHAPTER 17. STATEFULSET [APPS/V1]

Description

StatefulSet represents a set of pods with consistent identities. Identities are defined as: – Network: A single stable DNS and hostname. – Storage: As many VolumeClaims as requested.

The StatefulSet guarantees that a given network identity will always map to the same storage identity.

Type

object

17.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	A StatefulSetSpec is the specification of a StatefulSet.
status	object	StatefulSetStatus represents the current state of a StatefulSet.

17.1.1. .spec

Description

A StatefulSetSpec is the specification of a StatefulSet.

Type

object

Required

- **selector**
- **template**

Property	Type	Description
minReadySeconds	integer	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
ordinals	object	StatefulSetOrdinals describes the policy used for replica ordinal assignment in this StatefulSet.
persistentVolumeClaimRetentionPolicy	object	StatefulSetPersistentVolumeClaimRetentionPolicy describes the policy used for PVCs created from the StatefulSet VolumeClaimTemplates.

Property	Type	Description
podManagementPolicy	string	podManagementPolicy controls how pods are created during initial scale up, when replacing pods on nodes, or when scaling down. The default policy is OrderedReady, where pods are created in increasing order (pod-0, then pod-1, etc) and the controller will wait until each pod is ready before continuing. When scaling down, the pods are removed in the opposite order. The alternative policy is Parallel which will create pods in parallel to match the desired scale without waiting, and on scale down will delete all pods at once. Possible enum values: - "OrderedReady" will create pods in strictly increasing order on scale up and strictly decreasing order on scale down, progressing only when the previous pod is ready or terminated. At most one pod will be changed at any time. - "Parallel" will create and delete pods as soon as the stateful set replica count is changed, and will not wait for pods to be ready or complete termination.
replicas	integer	replicas is the desired number of replicas of the given Template. These are replicas in the sense that they are instantiations of the same Template, but individual replicas also have a consistent identity. If unspecified, defaults to 1.
revisionHistoryLimit	integer	revisionHistoryLimit is the maximum number of revisions that will be maintained in the StatefulSet's revision history. The revision history consists of all revisions not represented by a currently applied StatefulSetSpec version. The default value is 10.

Property	Type	Description
selector	LabelSelector	selector is a label query over pods that should match the replica count. It must match the pod template's labels. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
serviceName	string	serviceName is the name of the service that governs this StatefulSet. This service must exist before the StatefulSet, and is responsible for the network identity of the set. Pods get DNS/hostnames that follow the pattern: pod-specific-string.serviceName.default.svc.cluster.local where "pod-specific-string" is managed by the StatefulSet controller.
template	PodTemplateSpec	template is the object that describes the pod that will be created if insufficient replicas are detected. Each pod stamped out by the StatefulSet will fulfill this Template, but have a unique identity from the rest of the StatefulSet. Each pod will be named with the format <statefulsetname>-<podindex>. For example, a pod in a StatefulSet named "web" with index number "3" would be named "web-3". The only allowed template.spec.restartPolicy value is "Always".
updateStrategy	object	StatefulSetUpdateStrategy indicates the strategy that the StatefulSet controller will use to perform updates. It includes any additional parameters necessary to perform the update for the indicated strategy.

Property	Type	Description
volumeClaimTemplates	array (PersistentVolumeClaim)	volumeClaimTemplates is a list of claims that pods are allowed to reference. The StatefulSet controller is responsible for mapping network identities to claims in a way that maintains the identity of a pod. Every claim in this list must have at least one matching (by name) volumeMount in one container in the template. A claim in this list takes precedence over any volumes in the template, with the same name.

17.1.2. .spec.ordinals

Description

StatefulSetOrdinals describes the policy used for replica ordinal assignment in this StatefulSet.

Type

object

Property	Type	Description
start	integer	start is the number representing the first replica's index. It may be used to number replicas from an alternate index (eg: 1-indexed) over the default 0-indexed names, or to orchestrate progressive movement of replicas from one StatefulSet to another. If set, replica indices will be in the range: [.spec.ordinals.start, .spec.ordinals.start + .spec.replicas). If unset, defaults to 0. Replica indices will be in the range: [0, .spec.replicas).

17.1.3. .spec.persistentVolumeClaimRetentionPolicy

Description

StatefulSetPersistentVolumeClaimRetentionPolicy describes the policy used for PVCs created from the StatefulSet VolumeClaimTemplates.

Type

object

Property	Type	Description
whenDeleted	string	WhenDeleted specifies what happens to PVCs created from StatefulSet VolumeClaimTemplates when the StatefulSet is deleted. The default policy of Retain causes PVCs to not be affected by StatefulSet deletion. The Delete policy causes those PVCs to be deleted.
whenScaled	string	WhenScaled specifies what happens to PVCs created from StatefulSet VolumeClaimTemplates when the StatefulSet is scaled down. The default policy of Retain causes PVCs to not be affected by a scaledown. The Delete policy causes the associated PVCs for any excess pods above the replica count to be deleted.

17.1.4. .spec.updateStrategy

Description

StatefulSetUpdateStrategy indicates the strategy that the StatefulSet controller will use to perform updates. It includes any additional parameters necessary to perform the update for the indicated strategy.

Type

object

Property	Type	Description
rollingUpdate	object	RollingUpdateStatefulSetStrategy is used to communicate parameter for RollingUpdateStatefulSetStrategyType.

Property	Type	Description
type	string	Type indicates the type of the StatefulSetUpdateStrategy. Default is RollingUpdate. Possible enum values: - "OnDelete" triggers the legacy behavior. Version tracking and ordered rolling restarts are disabled. Pods are recreated from the StatefulSetSpec when they are manually deleted. When a scale operation is performed with this strategy, specification version indicated by the StatefulSet's currentRevision. - "RollingUpdate" indicates that update will be applied to all Pods in the StatefulSet with respect to the StatefulSet ordering constraints. When a scale operation is performed with this strategy, new Pods will be created from the specification version indicated by the StatefulSet's updateRevision.

17.1.5. .spec.updateStrategy.rollingUpdate

Description

RollingUpdateStatefulSetStrategy is used to communicate parameter for RollingUpdateStatefulSetStrategyType.

Type

object

Property	Type	Description

Property	Type	Description
maxUnavailable	IntOrString	The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding up. This can not be 0. Defaults to 1. This field is alpha-level and is only honored by servers that enable the MaxUnavailableStatefulSet feature. The field applies to all pods in the range 0 to Replicas-1. That means if there is any unavailable pod in the range 0 to Replicas-1, it will be counted towards MaxUnavailable.
partition	integer	Partition indicates the ordinal at which the StatefulSet should be partitioned for updates. During a rolling update, all pods from ordinal Replicas-1 to Partition are updated. All pods from ordinal Partition-1 to 0 remain untouched. This is helpful in being able to do a canary based deployment. The default value is 0.

17.1.6. .status

Description

StatefulSetStatus represents the current state of a StatefulSet.

Type

object

Required

- **replicas**

Property	Type	Description
availableReplicas	integer	Total number of available pods (ready for at least minReadySeconds) targeted by this statefulset.

Property	Type	Description
collisionCount	integer	collisionCount is the count of hash collisions for the StatefulSet. The StatefulSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision.
conditions	array	Represents the latest available observations of a statefulset's current state.
conditions[]	object	StatefulSetCondition describes the state of a statefulset at a certain point.
currentReplicas	integer	currentReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by currentRevision.
currentRevision	string	currentRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [0,currentReplicas).
observedGeneration	integer	observedGeneration is the most recent generation observed for this StatefulSet. It corresponds to the StatefulSet's generation, which is updated on mutation by the API Server.
readyReplicas	integer	readyReplicas is the number of pods created for this StatefulSet with a Ready Condition.
replicas	integer	replicas is the number of Pods created by the StatefulSet controller.
updateRevision	string	updateRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [replicas-updatedReplicas,replicas)

Property	Type	Description
updatedReplicas	integer	updatedReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by updateRevision.

17.1.7. .status.conditions

Description

Represents the latest available observations of a statefulset's current state.

Type

array

17.1.8. .status.conditions[]

Description

StatefulSetCondition describes the state of a statefulset at a certain point.

Type

object

Required

- **type**
- **status**

Property	Type	Description
lastTransitionTime	Time	Last time the condition transitioned from one status to another.
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of statefulset condition.

17.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps/v1/statefulsets**
 - **GET**: list or watch objects of kind StatefulSet
- **/apis/apps/v1/watch/statefulsets**
 - **GET**: watch individual changes to a list of StatefulSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/statefulsets**
 - **DELETE**: delete collection of StatefulSet
 - **GET**: list or watch objects of kind StatefulSet
 - **POST**: create a StatefulSet
- **/apis/apps/v1/watch/namespaces/{namespace}/statefulsets**
 - **GET**: watch individual changes to a list of StatefulSet. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}**
 - **DELETE**: delete a StatefulSet
 - **GET**: read the specified StatefulSet
 - **PATCH**: partially update the specified StatefulSet
 - **PUT**: replace the specified StatefulSet
- **/apis/apps/v1/watch/namespaces/{namespace}/statefulsets/{name}**
 - **GET**: watch changes to an object of kind StatefulSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status**
 - **GET**: read status of the specified StatefulSet
 - **PATCH**: partially update status of the specified StatefulSet
 - **PUT**: replace status of the specified StatefulSet

17.2.1. /apis/apps/v1/statefulsets

HTTP method

GET

Description

list or watch objects of kind StatefulSet

Table 17.1. HTTP responses

HTTP code	Reponse body
200 - OK	StatefulSetList schema
401 - Unauthorized	Empty

17.2.2. /apis/apps/v1/watch/statefulsets

HTTP method

GET

Description

watch individual changes to a list of StatefulSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 17.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

17.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets

HTTP method

DELETE

Description

delete collection of StatefulSet

Table 17.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 17.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind StatefulSet

Table 17.5. HTTP responses

HTTP code	Response body
200 - OK	StatefulSetList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a StatefulSet

Table 17.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 17.7. Body parameters

Parameter	Type	Description
body	StatefulSet schema	

Table 17.8. HTTP responses

HTTP code	Reponse body
200 - OK	StatefulSet schema
201 - Created	StatefulSet schema
202 - Accepted	StatefulSet schema
401 - Unauthorized	Empty

17.2.4. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets

HTTP method

GET

Description

watch individual changes to a list of StatefulSet. deprecated: use the 'watch' parameter with a list operation instead.

Table 17.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

17.2.5. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}

Table 17.10. Global path parameters

Parameter	Type	Description
name	string	name of the StatefulSet

HTTP method

DELETE

Description

delete a StatefulSet

Table 17.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 17.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified StatefulSet

Table 17.13. HTTP responses

HTTP code	Reponse body
200 - OK	StatefulSet schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified StatefulSet

Table 17.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 17.15. HTTP responses

HTTP code	Response body
200 - OK	StatefulSet schema
201 - Created	StatefulSet schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified StatefulSet

Table 17.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 17.17. Body parameters

Parameter	Type	Description
body	StatefulSet schema	

Table 17.18. HTTP responses

HTTP code	Response body
200 - OK	StatefulSet schema
201 - Created	StatefulSet schema
401 - Unauthorized	Empty

17.2.6. /apis/apps/v1/watch/namespaces/{namespace}/statefulsets/{name}

Table 17.19. Global path parameters

Parameter	Type	Description
name	string	name of the StatefulSet

HTTP method

GET

Description

watch changes to an object of kind StatefulSet. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 17.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

17.2.7. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status**Table 17.21. Global path parameters**

Parameter	Type	Description
name	string	name of the StatefulSet

HTTP method**GET****Description**

read status of the specified StatefulSet

Table 17.22. HTTP responses

HTTP code	Reponse body
200 - OK	StatefulSet schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update status of the specified StatefulSet

Table 17.23. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 17.24. HTTP responses

HTTP code	Response body
200 - OK	StatefulSet schema
201 - Created	StatefulSet schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified StatefulSet

Table 17.25. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 17.26. Body parameters

Parameter	Type	Description
body	StatefulSet schema	

Table 17.27. HTTP responses

HTTP code	Response body
200 - OK	StatefulSet schema
201 - Created	StatefulSet schema
401 - Unauthorized	Empty