



OpenShift Container Platform 4.20

Autoscale APIs

Reference guide for autoscale APIs

OpenShift Container Platform 4.20 Autoscale APIs

Reference guide for autoscale 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 autoscale API objects and their detailed specifications.

Table of Contents

CHAPTER 1. AUTOSCALE APIS	4
1.1. CLUSTERAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1]	4
1.2. MACHINEAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1BETA1]	4
1.3. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]	4
1.4. SCALE [AUTOSCALING/V1]	4
CHAPTER 2. CLUSTERAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1]	5
2.1. SPECIFICATION	5
2.1.1. .spec	5
2.1.2. .spec.resourceLimits	8
2.1.3. .spec.resourceLimits.cores	9
2.1.4. .spec.resourceLimits.gpus	10
2.1.5. .spec.resourceLimits.gpus[]	10
2.1.6. .spec.resourceLimits.memory	11
2.1.7. .spec.scaleDown	11
2.1.8. .spec.scaleUp	12
2.1.9. .status	12
2.2. API ENDPOINTS	12
2.2.1. /apis/autoscaling.openshift.io/v1/clusterautoscalers	13
2.2.2. /apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}	15
2.2.3. /apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}/status	17
CHAPTER 3. MACHINEAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1BETA1]	21
3.1. SPECIFICATION	21
3.1.1. .spec	21
3.1.2. .spec.scaleTargetRef	22
3.1.3. .status	23
3.1.4. .status.lastTargetRef	23
3.2. API ENDPOINTS	24
3.2.1. /apis/autoscaling.openshift.io/v1beta1/machineautoscalers	25
3.2.2. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers	25
3.2.3. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers/{name}	27
3.2.4. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers/{name}/status	30
CHAPTER 4. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]	34
4.1. SPECIFICATION	34
4.1.1. .spec	35
4.1.2. .spec.behavior	36
4.1.3. .spec.behavior.scaleDown	38
4.1.4. .spec.behavior.scaleDown.policies	39
4.1.5. .spec.behavior.scaleDown.policies[]	39
4.1.6. .spec.behavior.scaleUp	40
4.1.7. .spec.behavior.scaleUp.policies	41
4.1.8. .spec.behavior.scaleUp.policies[]	42
4.1.9. .spec.metrics	42
4.1.10. .spec.metrics[]	42
4.1.11. .spec.metrics[].containerResource	44
4.1.12. .spec.metrics[].containerResource.target	45
4.1.13. .spec.metrics[].external	45
4.1.14. .spec.metrics[].external.metric	46
4.1.15. .spec.metrics[].external.target	46

4.1.16. .spec.metrics[].object	47
4.1.17. .spec.metrics[].object.describedObject	48
4.1.18. .spec.metrics[].object.metric	49
4.1.19. .spec.metrics[].object.target	49
4.1.20. .spec.metrics[].pods	50
4.1.21. .spec.metrics[].pods.metric	50
4.1.22. .spec.metrics[].pods.target	51
4.1.23. .spec.metrics[].resource	52
4.1.24. .spec.metrics[].resource.target	52
4.1.25. .spec.scaleTargetRef	53
4.1.26. .status	54
4.1.27. .status.conditions	55
4.1.28. .status.conditions[]	55
4.1.29. .status.currentMetrics	56
4.1.30. .status.currentMetrics[]	56
4.1.31. .status.currentMetrics[].containerResource	57
4.1.32. .status.currentMetrics[].containerResource.current	58
4.1.33. .status.currentMetrics[].external	58
4.1.34. .status.currentMetrics[].external.current	59
4.1.35. .status.currentMetrics[].external.metric	59
4.1.36. .status.currentMetrics[].object	60
4.1.37. .status.currentMetrics[].object.current	60
4.1.38. .status.currentMetrics[].object.describedObject	61
4.1.39. .status.currentMetrics[].object.metric	62
4.1.40. .status.currentMetrics[].pods	62
4.1.41. .status.currentMetrics[].pods.current	63
4.1.42. .status.currentMetrics[].pods.metric	63
4.1.43. .status.currentMetrics[].resource	64
4.1.44. .status.currentMetrics[].resource.current	64
4.2. API ENDPOINTS	65
4.2.1. /apis/autoscaling/v2/horizontalpodautoscalers	66
4.2.2. /apis/autoscaling/v2/watch/horizontalpodautoscalers	66
4.2.3. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers	66
4.2.4. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers	69
4.2.5. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}	69
4.2.6. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers/{name}	72
4.2.7. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/status	73
CHAPTER 5. SCALE [AUTOSCALING/V1]	76
5.1. SPECIFICATION	76
5.1.1. .spec	76
5.1.2. .status	77
5.2. API ENDPOINTS	77
5.2.1. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale	78
5.2.2. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}/scale	80
5.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale	83
5.2.4. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale	86

CHAPTER 1. AUTOSCALE APIS

1.1. CLUSTERAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1]

Description

ClusterAutoscaler is the Schema for the clusterautoscalers API

Type

object

1.2. MACHINEAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1BETA1]

Description

MachineAutoscaler is the Schema for the machineautoscalers API

Type

object

1.3. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]

Description

HorizontalPodAutoscaler is the configuration for a horizontal pod autoscaler, which automatically manages the replica count of any resource implementing the scale subresource based on the metrics specified.

Type

object

1.4. SCALE [AUTOSCALING/V1]

Description

Scale represents a scaling request for a resource.

Type

object

CHAPTER 2. CLUSTERAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1]

Description

ClusterAutoscaler is the Schema for the clusterautoscalers API

Type

object

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
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	Desired state of ClusterAutoscaler resource
status	object	Most recently observed status of ClusterAutoscaler resource

2.1.1. .spec

Description

Desired state of ClusterAutoscaler resource

Type

object

Property	Type	Description
balanceSimilarNodeGroups	boolean	BalanceSimilarNodeGroups enables/disables the --balance-similar-node-groups cluster-autoscaler feature. This feature will automatically identify node groups with the same instance type and the same set of labels and try to keep the respective sizes of those node groups balanced.
balancingIgnoredLabels	array (string)	BalancingIgnoredLabels sets "--balancing-ignore-label <label name>" flag on cluster-autoscaler for each listed label. This option specifies labels that cluster autoscaler should ignore when considering node group similarity. For example, if you have nodes with "topology.ebs.csi.aws.com/zone" label, you can add name of this label here to prevent cluster autoscaler from splitting nodes into different node groups based on its value.

Property	Type	Description
expanders	array (string)	Sets the type and order of expanders to be used during scale out operations. This option specifies an ordered list, highest priority first, of expanders that will be used by the cluster autoscaler to select node groups for expansion when scaling out. Expanders instruct the autoscaler on how to choose node groups when scaling out the cluster. They can be specified in order so that the result from the first expander is used as the input to the second, and so forth. For example, if set to [LeastWaste, Random] the autoscaler will first evaluate node groups to determine which will have the least resource waste, if multiple groups are selected the autoscaler will then randomly choose between those groups to determine the group for scaling. The following expanders are available: * LeastWaste - selects the node group that will have the least idle CPU (if tied, unused memory) after scale-up. * Priority - selects the node group that has the highest priority assigned by the user. For details, please see https://github.com/openshift/kubernetes-autoscaler/blob/master/cluster-autoscaler/expander/priority/readme.md * Random - selects the node group randomly. If not specified, the default value is Random, available options are: LeastWaste, Priority, Random.
ignoreDaemonsetsUtilization	boolean	Enables/Disables --ignore-daemonsets-utilization CA feature flag. Should CA ignore DaemonSet pods when calculating resource utilization for scaling down. false by default

Property	Type	Description
logVerbosity	integer	Sets the autoscaler log level. Default value is 1, level 4 is recommended for DEBUGGING and level 6 will enable almost everything. This option has priority over log level set by the CLUSTER_AUTOSCALER_VERBOSITY environment variable.
maxNodeProvisionTime	string	Maximum time CA waits for node to be provisioned
maxPodGracePeriod	integer	Gives pods graceful termination time before scaling down
podPriorityThreshold	integer	To allow users to schedule "best-effort" pods, which shouldn't trigger Cluster Autoscaler actions, but only run when there are spare resources available, More info: https://github.com/kubernetes/autoscaler/blob/master/cluster-autoscaler/FAQ.md#how-does-cluster-autoscaler-work-with-pod-priority-and-preemption
resourceLimits	object	Constraints of autoscaling resources
scaleDown	object	Configuration of scale down operation
scaleUp	object	Configuration of scale up operation
skipNodesWithLocalStorage	boolean	Enables/Disables --skip-nodes-with-local-storage CA feature flag. If true cluster autoscaler will never delete nodes with pods with local storage, e.g. EmptyDir or HostPath. true by default at autoscaler

2.1.2. .spec.resourceLimits

Description

Constraints of autoscaling resources

Type

object

Property	Type	Description
cores	object	Minimum and maximum number of cores in cluster, in the format <min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers.
gpus	array	Minimum and maximum number of different GPUs in cluster, in the format <gpu_type>:<min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers. Can be passed multiple times.
gpus[]	object	
maxNodesTotal	integer	Maximum number of nodes in all node groups. Cluster autoscaler will not grow the cluster beyond this number.
memory	object	Minimum and maximum number of GiB of memory in cluster, in the format <min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers.

2.1.3. .spec.resourceLimits.cores**Description**

Minimum and maximum number of cores in cluster, in the format <min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers.

Type

object

Required

- **max**
- **min**

Property	Type	Description
max	integer	
min	integer	

2.1.4. .spec.resourceLimits.gpus

Description

Minimum and maximum number of different GPUs in cluster, in the format <gpu_type>:<min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers. Can be passed multiple times.

Type

array

2.1.5. .spec.resourceLimits.gpus[]

Description

Type

object

Required

- **max**
- **min**
- **type**

Property	Type	Description
max	integer	
min	integer	
type	string	The type of GPU to associate with the minimum and maximum limits. This value is used by the Cluster Autoscaler to identify Nodes that will have GPU capacity by searching for it as a label value on the Node objects. For example, Nodes that carry the label key cluster-api/accelerator with the label value being the same as the Type field will be counted towards the resource limits by the Cluster Autoscaler.

2.1.6. .spec.resourceLimits.memory

Description

Minimum and maximum number of GiB of memory in cluster, in the format <min>:<max>. Cluster autoscaler will not scale the cluster beyond these numbers.

Type

object

Required

- **max**
- **min**

Property	Type	Description
max	integer	
min	integer	

2.1.7. .spec.scaleDown

Description

Configuration of scale down operation

Type

object

Required

- **enabled**

Property	Type	Description
delayAfterAdd	string	How long after scale up that scale down evaluation resumes
delayAfterDelete	string	How long after node deletion that scale down evaluation resumes, defaults to scan-interval
delayAfterFailure	string	How long after scale down failure that scale down evaluation resumes
enabled	boolean	Should CA scale down the cluster

Property	Type	Description
unneededTime	string	How long a node should be unneeded before it is eligible for scale down
utilizationThreshold	string	Node utilization level, defined as sum of requested resources divided by capacity, below which a node can be considered for scale down

2.1.8. .spec.scaleUp

Description

Configuration of scale up operation

Type

object

Property	Type	Description
newPodScaleUpDelay	string	Scale up delay for new pods, if omitted defaults to 0 seconds

2.1.9. .status

Description

Most recently observed status of ClusterAutoscaler resource

Type

object

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/autoscaling.openshift.io/v1/clusterautoscalers**
 - **DELETE**: delete collection of ClusterAutoscaler
 - **GET**: list objects of kind ClusterAutoscaler
 - **POST**: create a ClusterAutoscaler
- **/apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}**
 - **DELETE**: delete a ClusterAutoscaler
 - **GET**: read the specified ClusterAutoscaler

- **PATCH**: partially update the specified ClusterAutoscaler
- **PUT**: replace the specified ClusterAutoscaler
- **/apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}/status**
 - **GET**: read status of the specified ClusterAutoscaler
 - **PATCH**: partially update status of the specified ClusterAutoscaler
 - **PUT**: replace status of the specified ClusterAutoscaler

2.2.1. /apis/autoscaling.openshift.io/v1/clusterautoscalers

HTTP method

DELETE

Description

delete collection of ClusterAutoscaler

Table 2.1. HTTP responses

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

HTTP method

GET

Description

list objects of kind ClusterAutoscaler

Table 2.2. HTTP responses

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

HTTP method

POST

Description

create a ClusterAutoscaler

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
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 2.4. Body parameters

Parameter	Type	Description
body	ClusterAutoscaler schema	

Table 2.5. HTTP responses

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

2.2.2. /apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}

Table 2.6. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterAutoscaler

HTTP method

DELETE

Description

delete a ClusterAutoscaler

Table 2.7. 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 2.8. HTTP responses

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

HTTP method

GET

Description

read the specified ClusterAutoscaler

Table 2.9. HTTP responses

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

HTTP method

PATCH**Description**

partially update the specified ClusterAutoscaler

Table 2.10. 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 2.11. HTTP responses

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

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

replace the specified ClusterAutoscaler

Table 2.12. 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 2.13. Body parameters

Parameter	Type	Description
body	ClusterAutoscaler schema	

Table 2.14. HTTP responses

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

2.2.3. /apis/autoscaling.openshift.io/v1/clusterautoscalers/{name}/status

Table 2.15. Global path parameters

Parameter	Type	Description
name	string	name of the ClusterAutoscaler

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

read status of the specified ClusterAutoscaler

Table 2.16. HTTP responses

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

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

partially update status of the specified ClusterAutoscaler

Table 2.17. 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 2.18. HTTP responses

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

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

replace status of the specified ClusterAutoscaler

Table 2.19. 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 2.20. Body parameters

Parameter	Type	Description
body	ClusterAutoscaler schema	

Table 2.21. HTTP responses

HTTP code	Response body
200 - OK	ClusterAutoscaler schema
201 - Created	ClusterAutoscaler schema
401 - Unauthorized	Empty

CHAPTER 3. MACHINEAUTOSCALER [AUTOSCALING.OPENSIFT.IO/V1BETA1]

Description

MachineAutoscaler is the Schema for the machineautoscalers API

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	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	Specification of constraints of a scalable resource
status	object	Most recently observed status of a scalable resource

3.1.1. .spec

Description

Specification of constraints of a scalable resource

Type

object

Required

- **maxReplicas**
- **minReplicas**
- **scaleTargetRef**

Property	Type	Description
maxReplicas	integer	MaxReplicas constrains the maximal number of replicas of a scalable resource
minReplicas	integer	MinReplicas constrains the minimal number of replicas of a scalable resource
scaleTargetRef	object	ScaleTargetRef holds reference to a scalable resource

3.1.2. .spec.scaleTargetRef

Description

ScaleTargetRef holds reference to a scalable resource

Type

object

Required

- **kind**
- **name**

Property	Type	Description
----------	------	-------------

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 specifies a name of an object, e.g. worker-us-east-1a. Scalable resources are expected to exist under a single namespace.

3.1.3. .status

Description

Most recently observed status of a scalable resource

Type

object

Property	Type	Description
lastTargetRef	object	LastTargetRef holds reference to the recently observed scalable resource

3.1.4. .status.lastTargetRef

Description

LastTargetRef holds reference to the recently observed scalable resource

Type

object

Required

- **kind**
- **name**

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 specifies a name of an object, e.g. worker-us-east-1a. Scalable resources are expected to exist under a single namespace.

3.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/autoscaling.openshift.io/v1beta1/machineautoscalers**
 - **GET**: list objects of kind MachineAutoscaler
- **/apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers**
 - **DELETE**: delete collection of MachineAutoscaler

- **GET**: list objects of kind MachineAutoscaler
- **POST**: create a MachineAutoscaler
- **/apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers/{name}**
 - **DELETE**: delete a MachineAutoscaler
 - **GET**: read the specified MachineAutoscaler
 - **PATCH**: partially update the specified MachineAutoscaler
 - **PUT**: replace the specified MachineAutoscaler
- **/apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoscalers/{name}/status**
 - **GET**: read status of the specified MachineAutoscaler
 - **PATCH**: partially update status of the specified MachineAutoscaler
 - **PUT**: replace status of the specified MachineAutoscaler

3.2.1. /apis/autoscaling.openshift.io/v1beta1/machineautoscalers

HTTP method

GET

Description

list objects of kind MachineAutoscaler

Table 3.1. HTTP responses

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

3.2.2. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineauto

HTTP method

DELETE

Description

delete collection of MachineAutoscaler

Table 3.2. HTTP responses

HTTP code	Reponse body
-----------	--------------

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

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

list objects of kind MachineAutoscaler

Table 3.3. HTTP responses

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

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

create a MachineAutoscaler

Table 3.4. 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 3.5. Body parameters

Parameter	Type	Description
body	MachineAutoscaler schema	

Table 3.6. HTTP responses

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

3.2.3. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautc

Table 3.7. Global path parameters

Parameter	Type	Description
name	string	name of the MachineAutoscaler

HTTP method**DELETE****Description**

delete a MachineAutoscaler

Table 3.8. 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 3.9. HTTP responses

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

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

read the specified MachineAutoscaler

Table 3.10. HTTP responses

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

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

partially update the specified MachineAutoscaler

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
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 3.12. HTTP responses

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

HTTP method

PUT

Description

replace the specified MachineAutoscaler

Table 3.13. 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 3.14. Body parameters

Parameter	Type	Description
body	MachineAutoscaler schema	

Table 3.15. HTTP responses

HTTP code	Response body
200 - OK	MachineAutoscaler schema
201 - Created	MachineAutoscaler schema
401 - Unauthorized	Empty

3.2.4. /apis/autoscaling.openshift.io/v1beta1/namespaces/{namespace}/machineautoc

Table 3.16. Global path parameters

Parameter	Type	Description
name	string	name of the MachineAutoscaler

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

read status of the specified MachineAutoscaler

Table 3.17. HTTP responses

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

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

partially update status of the specified MachineAutoscaler

Table 3.18. 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 3.19. HTTP responses

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

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

replace status of the specified MachineAutoscaler

Table 3.20. 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 3.21. Body parameters

Parameter	Type	Description
body	MachineAutoscaler schema	

Table 3.22. HTTP responses

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

CHAPTER 4. HORIZONTALPODAUTOSCALER [AUTOSCALING/V2]

Description

HorizontalPodAutoscaler is the configuration for a horizontal pod autoscaler, which automatically manages the replica count of any resource implementing the scale subresource based on the metrics specified.

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
metadata	ObjectMeta	metadata is the standard object metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	HorizontalPodAutoscalerSpec describes the desired functionality of the HorizontalPodAutoscaler.

Property	Type	Description
status	object	HorizontalPodAutoscalerStatus describes the current status of a horizontal pod autoscaler.

4.1.1. .spec

Description

HorizontalPodAutoscalerSpec describes the desired functionality of the HorizontalPodAutoscaler.

Type

object

Required

- **scaleTargetRef**
- **maxReplicas**

Property	Type	Description
behavior	object	HorizontalPodAutoscalerBehavior configures the scaling behavior of the target in both Up and Down directions (scaleUp and scaleDown fields respectively).
maxReplicas	integer	maxReplicas is the upper limit for the number of replicas to which the autoscaler can scale up. It cannot be less than minReplicas.
metrics	array	metrics contains the specifications for which to use to calculate the desired replica count (the maximum replica count across all metrics will be used). The desired replica count is calculated multiplying the ratio between the target value and the current value by the current number of pods. Ergo, metrics used must decrease as the pod count is increased, and vice-versa. See the individual metric source types for more information about how each type of metric must respond. If not set, the default metric will be set to 80% average CPU utilization.

Property	Type	Description
metrics[]	object	MetricSpec specifies how to scale based on a single metric (only type and one other matching field should be set at once).
minReplicas	integer	minReplicas is the lower limit for the number of replicas to which the autoscaler can scale down. It defaults to 1 pod. minReplicas is allowed to be 0 if the alpha feature gate HPAScaleToZero is enabled and at least one Object or External metric is configured. Scaling is active as long as at least one metric value is available.
scaleTargetRef	object	CrossVersionObjectReference contains enough information to let you identify the referred resource.

4.1.2. .spec.behavior

Description

HorizontalPodAutoscalerBehavior configures the scaling behavior of the target in both Up and Down directions (scaleUp and scaleDown fields respectively).

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
scaleDown	object	HPAScalingRules configures the scaling behavior for one direction via scaling Policy Rules and a configurable metric tolerance. Scaling Policy Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen. The tolerance is applied to the metric values and prevents scaling too eagerly for small metric variations. (Note that setting a tolerance requires enabling the alpha HPAConfigurableTolerance feature gate.)
scaleUp	object	HPAScalingRules configures the scaling behavior for one direction via scaling Policy Rules and a configurable metric tolerance. Scaling Policy Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen. The tolerance is applied to the metric values and prevents scaling too eagerly for small metric variations. (Note that setting a tolerance requires enabling the alpha HPAConfigurableTolerance feature gate.)

4.1.3. .spec.behavior.scaleDown

Description

HPAScalingRules configures the scaling behavior for one direction via scaling Policy Rules and a configurable metric tolerance.

Scaling Policy Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.

The tolerance is applied to the metric values and prevents scaling too eagerly for small metric variations. (Note that setting a tolerance requires enabling the alpha HPAConfigurableTolerance feature gate.)

Type

object

Property	Type	Description
policies	array	policies is a list of potential scaling policies which can be used during scaling. If not set, use the default values: - For scale up: allow doubling the number of pods, or an absolute change of 4 pods in a 15s window. - For scale down: allow all pods to be removed in a 15s window.
policies[]	object	HPAScalingPolicy is a single policy which must hold true for a specified past interval.
selectPolicy	string	selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
stabilizationWindowSeconds	integer	stabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: - For scale up: 0 (i.e. no stabilization is done). - For scale down: 300 (i.e. the stabilization window is 300 seconds long).

Property	Type	Description
tolerance	Quantity	tolerance is the tolerance on the ratio between the current and desired metric value under which no updates are made to the desired number of replicas (e.g. 0.01 for 1%). Must be greater than or equal to zero. If not set, the default cluster-wide tolerance is applied (by default 10%). For example, if autoscaling is configured with a memory consumption target of 100Mi, and scale-down and scale-up tolerances of 5% and 1% respectively, scaling will be triggered when the actual consumption falls below 95Mi or exceeds 101Mi. This is an alpha field and requires enabling the <code>HPAConfigurableTolerance</code> feature gate.

4.1.4. .spec.behavior.scaleDown.policies

Description

policies is a list of potential scaling policies which can be used during scaling. If not set, use the default values: – For scale up: allow doubling the number of pods, or an absolute change of 4 pods in a 15s window. – For scale down: allow all pods to be removed in a 15s window.

Type

array

4.1.5. .spec.behavior.scaleDown.policies[]

Description

HPAScalingPolicy is a single policy which must hold true for a specified past interval.

Type

object

Required

- **type**
- **value**
- **periodSeconds**

Property	Type	Description
periodSeconds	integer	periodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
type	string	type is used to specify the scaling policy.
value	integer	value contains the amount of change which is permitted by the policy. It must be greater than zero

4.1.6. .spec.behavior.scaleUp

Description

HPAScalingRules configures the scaling behavior for one direction via scaling Policy Rules and a configurable metric tolerance.

Scaling Policy Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.

The tolerance is applied to the metric values and prevents scaling too eagerly for small metric variations. (Note that setting a tolerance requires enabling the alpha HPAConfigurableTolerance feature gate.)

Type

object

Property	Type	Description
policies	array	policies is a list of potential scaling polices which can be used during scaling. If not set, use the default values: - For scale up: allow doubling the number of pods, or an absolute change of 4 pods in a 15s window. - For scale down: allow all pods to be removed in a 15s window.
policies[]	object	HPAScalingPolicy is a single policy which must hold true for a specified past interval.

Property	Type	Description
selectPolicy	string	selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
stabilizationWindowSeconds	integer	stabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: - For scale up: 0 (i.e. no stabilization is done). - For scale down: 300 (i.e. the stabilization window is 300 seconds long).
tolerance	Quantity	tolerance is the tolerance on the ratio between the current and desired metric value under which no updates are made to the desired number of replicas (e.g. 0.01 for 1%). Must be greater than or equal to zero. If not set, the default cluster-wide tolerance is applied (by default 10%). For example, if autoscaling is configured with a memory consumption target of 100Mi, and scale-down and scale-up tolerances of 5% and 1% respectively, scaling will be triggered when the actual consumption falls below 95Mi or exceeds 101Mi. This is an alpha field and requires enabling the HPAConfigurableTolerance feature gate.

4.1.7. .spec.behavior.scaleUp.policies

Description

policies is a list of potential scaling policies which can be used during scaling. If not set, use the default values: - For scale up: allow doubling the number of pods, or an absolute change of 4 pods in a 15s window. - For scale down: allow all pods to be removed in a 15s window.

Type

array

4.1.8. .spec.behavior.scaleUp.policies[]

Description

HPAScalingPolicy is a single policy which must hold true for a specified past interval.

Type

object

Required

- **type**
- **value**
- **periodSeconds**

Property	Type	Description
periodSeconds	integer	periodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
type	string	type is used to specify the scaling policy.
value	integer	value contains the amount of change which is permitted by the policy. It must be greater than zero

4.1.9. .spec.metrics

Description

metrics contains the specifications for which to use to calculate the desired replica count (the maximum replica count across all metrics will be used). The desired replica count is calculated multiplying the ratio between the target value and the current value by the current number of pods. Ergo, metrics used must decrease as the pod count is increased, and vice-versa. See the individual metric source types for more information about how each type of metric must respond. If not set, the default metric will be set to 80% average CPU utilization.

Type

array

4.1.10. .spec.metrics[]

Description

MetricSpec specifies how to scale based on a single metric (only **type** and one other matching field should be set at once).

Type

object

Required

- **type**

Property	Type	Description
containerResource	object	ContainerResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.
external	object	ExternalMetricSource indicates how to scale on a metric not associated with any Kubernetes object (for example length of queue in cloud messaging service, or QPS from loadbalancer running outside of cluster).
object	object	ObjectMetricSource indicates how to scale on a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).
pods	object	PodsMetricSource indicates how to scale on a metric describing each pod in the current scale target (for example, transactions-processed-per-second). The values will be averaged together before being compared to the target value.

Property	Type	Description
resource	object	ResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.
type	string	type is the type of metric source. It should be one of "ContainerResource", "External", "Object", "Pods" or "Resource", each mapping to a matching field in the object.

4.1.11. .spec.metrics[].containerResource

Description

ContainerResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.

Type

object

Required

- **name**
- **target**
- **container**

Property	Type	Description
container	string	container is the name of the container in the pods of the scaling target

Property	Type	Description
name	string	name is the name of the resource in question.
target	object	MetricTarget defines the target value, average value, or average utilization of a specific metric

4.1.12. .spec.metrics[].containerResource.target

Description

MetricTarget defines the target value, average value, or average utilization of a specific metric

Type

object

Required

- **type**

Property	Type	Description
averageUtilization	integer	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods. Currently only valid for Resource metric source type
averageValue	Quantity	averageValue is the target value of the average of the metric across all relevant pods (as a quantity)
type	string	type represents whether the metric type is Utilization, Value, or AverageValue
value	Quantity	value is the target value of the metric (as a quantity).

4.1.13. .spec.metrics[].external

Description

ExternalMetricSource indicates how to scale on a metric not associated with any Kubernetes object (for example length of queue in cloud messaging service, or QPS from loadbalancer running outside of cluster).

Type

object

Required

- **metric**
- **target**

Property	Type	Description
metric	object	MetricIdentifier defines the name and optionally selector for a metric
target	object	MetricTarget defines the target value, average value, or average utilization of a specific metric

4.1.14. .spec.metrics[].external.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.15. .spec.metrics[].external.target

Description

MetricTarget defines the target value, average value, or average utilization of a specific metric

Type**object****Required**

- **type**

Property	Type	Description
averageUtilization	integer	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods. Currently only valid for Resource metric source type
averageValue	Quantity	averageValue is the target value of the average of the metric across all relevant pods (as a quantity)
type	string	type represents whether the metric type is Utilization, Value, or AverageValue
value	Quantity	value is the target value of the metric (as a quantity).

4.1.16. .spec.metrics[].object

Description

ObjectMetricSource indicates how to scale on a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).

Type**object****Required**

- **describedObject**
- **target**
- **metric**

Property	Type	Description
----------	------	-------------

Property	Type	Description
describedObject	object	CrossVersionObjectReference contains enough information to let you identify the referred resource.
metric	object	MetricIdentifier defines the name and optionally selector for a metric
target	object	MetricTarget defines the target value, average value, or average utilization of a specific metric

4.1.17. .spec.metrics[].object.describedObject

Description

CrossVersionObjectReference contains enough information to let you identify the referred resource.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiVersion	string	apiVersion is the API version of the referent
kind	string	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
name	string	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

4.1.18. .spec.metrics[].object.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric. When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.19. .spec.metrics[].object.target

Description

MetricTarget defines the target value, average value, or average utilization of a specific metric

Type

object

Required

- **type**

Property	Type	Description
averageUtilization	integer	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods. Currently only valid for Resource metric source type

Property	Type	Description
averageValue	Quantity	averageValue is the target value of the average of the metric across all relevant pods (as a quantity)
type	string	type represents whether the metric type is Utilization, Value, or AverageValue
value	Quantity	value is the target value of the metric (as a quantity).

4.1.20. .spec.metrics[].pods

Description

PodsMetricSource indicates how to scale on a metric describing each pod in the current scale target (for example, transactions-processed-per-second). The values will be averaged together before being compared to the target value.

Type

object

Required

- **metric**
- **target**

Property	Type	Description
metric	object	MetricIdentifier defines the name and optionally selector for a metric
target	object	MetricTarget defines the target value, average value, or average utilization of a specific metric

4.1.21. .spec.metrics[].pods.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.22. .spec.metrics[].pods.target

Description

MetricTarget defines the target value, average value, or average utilization of a specific metric

Type

object

Required

- **type**

Property	Type	Description
averageUtilization	integer	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods. Currently only valid for Resource metric source type
averageValue	Quantity	averageValue is the target value of the average of the metric across all relevant pods (as a quantity)
type	string	type represents whether the metric type is Utilization, Value, or AverageValue

Property	Type	Description
value	Quantity	value is the target value of the metric (as a quantity).

4.1.23. .spec.metrics[].resource

Description

ResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.

Type

object

Required

- **name**
- **target**

Property	Type	Description
name	string	name is the name of the resource in question.
target	object	MetricTarget defines the target value, average value, or average utilization of a specific metric

4.1.24. .spec.metrics[].resource.target

Description

MetricTarget defines the target value, average value, or average utilization of a specific metric

Type

object

Required

- **type**

Property	Type	Description
----------	------	-------------

Property	Type	Description
averageUtilization	integer	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods. Currently only valid for Resource metric source type
averageValue	Quantity	averageValue is the target value of the average of the metric across all relevant pods (as a quantity)
type	string	type represents whether the metric type is Utilization, Value, or AverageValue
value	Quantity	value is the target value of the metric (as a quantity).

4.1.25. .spec.scaleTargetRef

Description

CrossVersionObjectReference contains enough information to let you identify the referred resource.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiVersion	string	apiVersion is the API version of the referent
kind	string	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
name	string	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

4.1.26. .status

Description

HorizontalPodAutoscalerStatus describes the current status of a horizontal pod autoscaler.

Type

object

Required

- **desiredReplicas**

Property	Type	Description
conditions	array	conditions is the set of conditions required for this autoscaler to scale its target, and indicates whether or not those conditions are met.
conditions[]	object	HorizontalPodAutoscalerCondition describes the state of a HorizontalPodAutoscaler at a certain point.
currentMetrics	array	currentMetrics is the last read state of the metrics used by this autoscaler.
currentMetrics[]	object	MetricStatus describes the last-read state of a single metric.
currentReplicas	integer	currentReplicas is current number of replicas of pods managed by this autoscaler, as last seen by the autoscaler.

Property	Type	Description
desiredReplicas	integer	desiredReplicas is the desired number of replicas of pods managed by this autoscaler, as last calculated by the autoscaler.
lastScaleTime	Time	lastScaleTime is the last time the HorizontalPodAutoscaler scaled the number of pods, used by the autoscaler to control how often the number of pods is changed.
observedGeneration	integer	observedGeneration is the most recent generation observed by this autoscaler.

4.1.27. .status.conditions

Description

conditions is the set of conditions required for this autoscaler to scale its target, and indicates whether or not those conditions are met.

Type

array

4.1.28. .status.conditions[]

Description

HorizontalPodAutoscalerCondition describes the state of a HorizontalPodAutoscaler at a certain point.

Type

object

Required

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

Property	Type	Description
lastTransitionTime	Time	lastTransitionTime is the last time the condition transitioned from one status to another
message	string	message is a human-readable explanation containing details about the transition

Property	Type	Description
reason	string	reason is the reason for the condition's last transition.
status	string	status is the status of the condition (True, False, Unknown)
type	string	type describes the current condition

4.1.29. .status.currentMetrics

Description

currentMetrics is the last read state of the metrics used by this autoscaler.

Type

array

4.1.30. .status.currentMetrics[]

Description

MetricStatus describes the last-read state of a single metric.

Type

object

Required

- **type**

Property	Type	Description
containerResource	object	ContainerResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing a single container in each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.
external	object	ExternalMetricStatus indicates the current value of a global metric not associated with any Kubernetes object.

Property	Type	Description
object	object	ObjectMetricStatus indicates the current value of a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).
pods	object	PodsMetricStatus indicates the current value of a metric describing each pod in the current scale target (for example, transactions-processed-per-second).
resource	object	ResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.
type	string	type is the type of metric source. It will be one of "ContainerResource", "External", "Object", "Pods" or "Resource", each corresponds to a matching field in the object.

4.1.31. .status.currentMetrics[].containerResource

Description

ContainerResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing a single container in each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.

Type

object

Required

- **name**
- **current**
- **container**

Property	Type	Description
container	string	container is the name of the container in the pods of the scaling target
current	object	MetricValueStatus holds the current value for a metric
name	string	name is the name of the resource in question.

4.1.32. .status.currentMetrics[].containerResource.current

Description

MetricValueStatus holds the current value for a metric

Type

object

Property	Type	Description
averageUtilization	integer	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods.
averageValue	Quantity	averageValue is the current value of the average of the metric across all relevant pods (as a quantity)
value	Quantity	value is the current value of the metric (as a quantity).

4.1.33. .status.currentMetrics[].external

Description

ExternalMetricStatus indicates the current value of a global metric not associated with any Kubernetes object.

Type

object

Required

- **metric**
- **current**

Property	Type	Description
current	object	MetricValueStatus holds the current value for a metric
metric	object	MetricIdentifier defines the name and optionally selector for a metric

4.1.34. .status.currentMetrics[].external.current

Description

MetricValueStatus holds the current value for a metric

Type

object

Property	Type	Description
averageUtilization	integer	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods.
averageValue	Quantity	averageValue is the current value of the average of the metric across all relevant pods (as a quantity)
value	Quantity	value is the current value of the metric (as a quantity).

4.1.35. .status.currentMetrics[].external.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric. When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.36. .status.currentMetrics[].object

Description

ObjectMetricStatus indicates the current value of a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).

Type

object

Required

- **metric**
- **current**
- **describedObject**

Property	Type	Description
current	object	MetricValueStatus holds the current value for a metric
describedObject	object	CrossVersionObjectReference contains enough information to let you identify the referred resource.
metric	object	MetricIdentifier defines the name and optionally selector for a metric

4.1.37. .status.currentMetrics[].object.current

Description

MetricValueStatus holds the current value for a metric

Type
object

Property	Type	Description
averageUtilization	integer	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods.
averageValue	Quantity	averageValue is the current value of the average of the metric across all relevant pods (as a quantity)
value	Quantity	value is the current value of the metric (as a quantity).

4.1.38. .status.currentMetrics[].object.describedObject

Description

CrossVersionObjectReference contains enough information to let you identify the referred resource.

Type
object

Required

- **kind**
- **name**

Property	Type	Description
apiVersion	string	apiVersion is the API version of the referent
kind	string	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
name	string	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

4.1.39. .status.currentMetrics[].object.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.40. .status.currentMetrics[].pods

Description

PodsMetricStatus indicates the current value of a metric describing each pod in the current scale target (for example, transactions-processed-per-second).

Type

object

Required

- **metric**
- **current**

Property	Type	Description
current	object	MetricValueStatus holds the current value for a metric
metric	object	MetricIdentifier defines the name and optionally selector for a metric

4.1.41. .status.currentMetrics[].pods.current

Description

MetricValueStatus holds the current value for a metric

Type

object

Property	Type	Description
averageUtilization	integer	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods.
averageValue	Quantity	averageValue is the current value of the average of the metric across all relevant pods (as a quantity)
value	Quantity	value is the current value of the metric (as a quantity).

4.1.42. .status.currentMetrics[].pods.metric

Description

MetricIdentifier defines the name and optionally selector for a metric

Type

object

Required

- **name**

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	name is the name of the given metric
selector	LabelSelector	selector is the string-encoded form of a standard kubernetes label selector for the given metric. When set, it is passed as an additional parameter to the metrics server for more specific metrics scoping. When unset, just the metricName will be used to gather metrics.

4.1.43. `.status.currentMetrics[].resource`

Description

ResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.

Type

object

Required

- **name**
- **current**

Property	Type	Description
current	object	MetricValueStatus holds the current value for a metric
name	string	name is the name of the resource in question.

4.1.44. `.status.currentMetrics[].resource.current`

Description

MetricValueStatus holds the current value for a metric

Type

object

Property	Type	Description
averageUtilization	integer	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, represented as a percentage of the requested value of the resource for the pods.
averageValue	Quantity	averageValue is the current value of the average of the metric across all relevant pods (as a quantity)
value	Quantity	value is the current value of the metric (as a quantity).

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/autoscaling/v2/horizontalpodautoscalers**
 - **GET**: list or watch objects of kind HorizontalPodAutoscaler
- **/apis/autoscaling/v2/watch/horizontalpodautoscalers**
 - **GET**: watch individual changes to a list of HorizontalPodAutoscaler. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers**
 - **DELETE**: delete collection of HorizontalPodAutoscaler
 - **GET**: list or watch objects of kind HorizontalPodAutoscaler
 - **POST**: create a HorizontalPodAutoscaler
- **/apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers**
 - **GET**: watch individual changes to a list of HorizontalPodAutoscaler. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}**
 - **DELETE**: delete a HorizontalPodAutoscaler
 - **GET**: read the specified HorizontalPodAutoscaler
 - **PATCH**: partially update the specified HorizontalPodAutoscaler
 - **PUT**: replace the specified HorizontalPodAutoscaler

- **/apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers/{name}**
 - **GET**: watch changes to an object of kind HorizontalPodAutoscaler. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/status**
 - **GET**: read status of the specified HorizontalPodAutoscaler
 - **PATCH**: partially update status of the specified HorizontalPodAutoscaler
 - **PUT**: replace status of the specified HorizontalPodAutoscaler

4.2.1. /apis/autoscaling/v2/horizontalpodautoscalers

HTTP method

GET

Description

list or watch objects of kind HorizontalPodAutoscaler

Table 4.1. HTTP responses

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

4.2.2. /apis/autoscaling/v2/watch/horizontalpodautoscalers

HTTP method

GET

Description

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

Table 4.2. HTTP responses

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

4.2.3. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers

HTTP method

DELETE

Description

delete collection of HorizontalPodAutoscaler

Table 4.3. 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 4.4. HTTP responses

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

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

list or watch objects of kind HorizontalPodAutoscaler

Table 4.5. HTTP responses

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

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

create a HorizontalPodAutoscaler

Table 4.6. 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 4.7. Body parameters

Parameter	Type	Description
body	HorizontalPodAutoscaler schema	

Table 4.8. HTTP responses

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

4.2.4. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers/{name}

HTTP method

GET

Description

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

Table 4.9. HTTP responses

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

4.2.5. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}

Table 4.10. Global path parameters

Parameter	Type	Description
name	string	name of the HorizontalPodAutoscaler

HTTP method

DELETE

Description

delete a HorizontalPodAutoscaler

Table 4.11. 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 4.12. HTTP responses

HTTP code	Response body
200 - OK	Status schema
202 - Accepted	Status schema

HTTP code	Reponse body
401 - Unauthorized	Empty

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

read the specified HorizontalPodAutoscaler

Table 4.13. HTTP responses

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

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

partially update the specified HorizontalPodAutoscaler

Table 4.14. 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 4.15. HTTP responses

HTTP code	Response body
200 - OK	HorizontalPodAutoscaler schema
201 - Created	HorizontalPodAutoscaler schema
401 - Unauthorized	Empty

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

replace the specified HorizontalPodAutoscaler

Table 4.16. 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 4.17. Body parameters

Parameter	Type	Description
body	HorizontalPodAutoscaler schema	

Table 4.18. HTTP responses

HTTP code	Response body
200 - OK	HorizontalPodAutoscaler schema
201 - Created	HorizontalPodAutoscaler schema
401 - Unauthorized	Empty

4.2.6. /apis/autoscaling/v2/watch/namespaces/{namespace}/horizontalpodautoscalers

Table 4.19. Global path parameters

Parameter	Type	Description
name	string	name of the HorizontalPodAutoscaler

HTTP method

GET**Description**

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

Table 4.20. HTTP responses

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

4.2.7. /apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{na**Table 4.21. Global path parameters**

Parameter	Type	Description
name	string	name of the HorizontalPodAutoscaler

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

read status of the specified HorizontalPodAutoscaler

Table 4.22. HTTP responses

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

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

partially update status of the specified HorizontalPodAutoscaler

Table 4.23. 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 4.24. HTTP responses

HTTP code	Response body
200 - OK	HorizontalPodAutoscaler schema
201 - Created	HorizontalPodAutoscaler schema
401 - Unauthorized	Empty

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

replace status of the specified HorizontalPodAutoscaler

Table 4.25. 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 4.26. Body parameters

Parameter	Type	Description
body	HorizontalPodAutoscaler schema	

Table 4.27. HTTP responses

HTTP code	Response body
200 - OK	HorizontalPodAutoscaler schema
201 - Created	HorizontalPodAutoscaler schema
401 - Unauthorized	Empty

CHAPTER 5. SCALE [AUTOSCALING/V1]

Description

Scale represents a scaling request for a resource.

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
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 metadata; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata .
spec	object	ScaleSpec describes the attributes of a scale subresource.
status	object	ScaleStatus represents the current status of a scale subresource.

5.1.1. .spec

Description

ScaleSpec describes the attributes of a scale subresource.

Type

object

Property	Type	Description
replicas	integer	replicas is the desired number of instances for the scaled object.

5.1.2. .status**Description**

ScaleStatus represents the current status of a scale subresource.

Type

object

Required

- **replicas**

Property	Type	Description
replicas	integer	replicas is the actual number of observed instances of the scaled object.
selector	string	selector is the label query over pods that should match the replicas count. This is same as the label selector but in the string format to avoid introspection by clients. The string will be in the same format as the query-param syntax. More info about label selectors: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale**
 - **GET**: read scale of the specified Deployment
 - **PATCH**: partially update scale of the specified Deployment

- **PUT**: replace scale of the specified Deployment
- **/apis/apps/v1/namespaces/{namespace}/replicasets/{name}/scale**
 - **GET**: read scale of the specified ReplicaSet
 - **PATCH**: partially update scale of the specified ReplicaSet
 - **PUT**: replace scale of the specified ReplicaSet
- **/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale**
 - **GET**: read scale of the specified StatefulSet
 - **PATCH**: partially update scale of the specified StatefulSet
 - **PUT**: replace scale of the specified StatefulSet
- **/api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale**
 - **GET**: read scale of the specified ReplicationController
 - **PATCH**: partially update scale of the specified ReplicationController
 - **PUT**: replace scale of the specified ReplicationController

5.2.1. /apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale

Table 5.1. Global path parameters

Parameter	Type	Description
name	string	name of the Scale

HTTP method

GET

Description

read scale of the specified Deployment

Table 5.2. HTTP responses

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

HTTP method

PATCH

Description

partially update scale of the specified Deployment

Table 5.3. 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
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 5.4. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace scale of the specified Deployment

Table 5.5. 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 5.6. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 5.7. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

5.2.2. /apis/apps/v1/namespaces/{namespace}/replicasets/{name}/scale

Table 5.8. Global path parameters

Parameter	Type	Description
name	string	name of the Scale

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

read scale of the specified ReplicaSet

Table 5.9. HTTP responses

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

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

partially update scale of the specified ReplicaSet

Table 5.10. 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 5.11. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

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

replace scale of the specified ReplicaSet

Table 5.12. 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 5.13. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 5.14. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

5.2.3. /apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale

Table 5.15. Global path parameters

Parameter	Type	Description
name	string	name of the Scale

HTTP method

GET**Description**

read scale of the specified StatefulSet

Table 5.16. HTTP responses

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

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

partially update scale of the specified StatefulSet

Table 5.17. 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 5.18. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

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

replace scale of the specified StatefulSet

Table 5.19. 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 5.20. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 5.21. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

5.2.4. /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale

Table 5.22. Global path parameters

Parameter	Type	Description
name	string	name of the Scale

HTTP method

GET

Description

read scale of the specified ReplicationController

Table 5.23. HTTP responses

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

HTTP method

PATCH

Description

partially update scale of the specified ReplicationController

Table 5.24. 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 5.25. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty

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

replace scale of the specified ReplicationController

Table 5.26. 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 5.27. Body parameters

Parameter	Type	Description
body	Scale schema	

Table 5.28. HTTP responses

HTTP code	Response body
200 - OK	Scale schema
201 - Created	Scale schema
401 - Unauthorized	Empty