



OpenShift Container Platform 4.20

Observability overview

Contains information about observability for OpenShift Container Platform

OpenShift Container Platform 4.20 Observability overview

Contains information about observability for OpenShift Container Platform

Legal Notice

Copyright © Red Hat.

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, 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 Software Collections 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

Red Hat OpenShift Observability provides real-time visibility, monitoring, and analysis of various system metrics, logs, traces, and events to help users quickly diagnose and troubleshoot issues before they impact systems or applications.

Table of Contents

CHAPTER 1. ABOUT OBSERVABILITY 3

1.1. MONITORING 3

1.2. LOGGING 3

1.3. DISTRIBUTED TRACING 3

1.4. RED HAT BUILD OF OPENTELEMETRY 4

1.5. NETWORK OBSERVABILITY 4

1.6. POWER MONITORING 4

CHAPTER 1. ABOUT OBSERVABILITY

Red Hat OpenShift Observability provides real-time visibility, monitoring, and analysis of various system metrics, logs, traces, and events to help users quickly diagnose and troubleshoot issues before they impact systems or applications. To help ensure the reliability, performance, and security of your applications and infrastructure, OpenShift Container Platform offers the following Observability components:

- Monitoring
- Logging
- Distributed tracing
- Red Hat build of OpenTelemetry
- Network Observability
- Power monitoring

Red Hat OpenShift Observability connects open-source observability tools and technologies to create a unified Observability solution. The components of Red Hat OpenShift Observability work together to help you collect, store, deliver, analyze, and visualize data.



NOTE

With the exception of monitoring, Red Hat OpenShift Observability components have distinct release cycles separate from the core OpenShift Container Platform release cycles. See the Red Hat [OpenShift Operator Life Cycles](#) page for their release compatibility.

1.1. MONITORING

Monitor the in-cluster health and performance of your applications running on OpenShift Container Platform with metrics and customized alerts for CPU and memory usage, network connectivity, and other resource usage. Monitoring stack components are deployed and managed by the Cluster Monitoring Operator.

Monitoring stack components are deployed by default in every OpenShift Container Platform installation and are managed by the Cluster Monitoring Operator (CMO). These components include Prometheus, Alertmanager, Thanos Querier, and others. The CMO also deploys the Telemeter Client, which sends a subset of data from platform Prometheus instances to Red Hat to facilitate Remote Health Monitoring for clusters.

For more information, see [About OpenShift Container Platform monitoring](#) and [About remote health monitoring](#).

1.2. LOGGING

Collect, visualize, forward, and store log data to troubleshoot issues, identify performance bottlenecks, and detect security threats. In logging 5.7 and later versions, users can configure the LokiStack deployment to produce customized alerts and recorded metrics.

1.3. DISTRIBUTED TRACING

Store and visualize large volumes of requests passing through distributed systems, across the whole stack of microservices, and under heavy loads. Use it for monitoring distributed transactions, gathering insights into your instrumented services, network profiling, performance and latency optimization, root cause analysis, and troubleshooting the interaction between components in modern cloud-native microservices-based applications.

For more information, see [Distributed tracing architecture](#).

1.4. RED HAT BUILD OF OPENTELEMETRY

Instrument, generate, collect, and export telemetry traces, metrics, and logs to analyze and understand your software's performance and behavior. Use open-source back ends like Tempo or Prometheus, or use commercial offerings. Learn a single set of APIs and conventions, and own the data that you generate.

For more information, see [Red Hat build of OpenTelemetry](#).

1.5. NETWORK OBSERVABILITY

Observe the network traffic for OpenShift Container Platform clusters and create network flows with the Network Observability Operator. View and analyze the stored network flows information in the OpenShift Container Platform console for further insight and troubleshooting.

For more information, see [Network Observability overview](#).

1.6. POWER MONITORING

Monitor the power usage of workloads and identify the most power-consuming namespaces running in a cluster with key power consumption metrics, such as CPU or DRAM measured at the container level. Visualize energy-related system statistics with the Power Monitoring Operator.

For more information, see [Power monitoring overview](#).