OpenShift Container Platform 3.11

Service Mesh Release Notes

OpenShift Container Platform 3.11 Service Mesh Release Notes

Last Updated: 2018-10-12

OpenShift Container Platform 3.11 Service Mesh Release Notes

OpenShift Container Platform 3.11 Service Mesh Release Notes

Legal Notice

Abstract

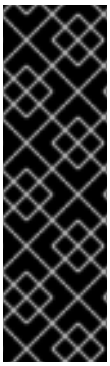
Release Notes for Service Mesh

Table of Contents

CHAPTER 1. RED HAT OPENSHIFT SERVICE MESH RELEASE NOTES

1.1. INTRODUCTION TO RED HAT OPENSHIFT SERVICE MESH 0.2.TECHPREVIEW

1.1.1. Red Hat OpenShift Service Mesh overview



IMPORTANT

This release of Red Hat OpenShift Service Mesh is a Technology Preview release only. Technology Preview releases are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete, and Red Hat does NOT recommend using them for production. Using Red Hat OpenShift Service Mesh on a cluster renders the whole OpenShift cluster as a technology preview, that is, in an unsupported state. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information see Red Hat Technology Preview Features Support Scope.

Red Hat OpenShift Service Mesh is a platform that provides behavioral insight and operational control over the service mesh, providing a uniform way to connect, secure, and monitor microservice applications.

The term *service mesh* describes the network of microservices that make up applications in a distributed microservice architecture and the interactions between those microservices. As a service mesh grows in size and complexity, it can become harder to understand and manage.

Based on the open source Istio project, Red Hat OpenShift Service Mesh adds a transparent layer on existing distributed applications without requiring any changes to the service code. You add Red Hat OpenShift Service Mesh support to services by deploying a special sidecar proxy throughout your environment that intercepts all network communication between microservices. You configure and manage the service mesh using the control plane features.

Red Hat OpenShift Service Mesh provides an easy way to create a network of deployed services that provides discovery, load balancing, service-to-service authentication, failure recovery, metrics, and monitoring. A service mesh also provides more complex operational functionality, including A/B testing, canary releases, rate limiting, access control, and end-to-end authentication.

1.1.2. Getting support

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. Through the customer portal, you can:

- Search or browse through the Red Hat Knowledgebase of technical support articles about Red Hat products
- Submit a support case to Red Hat Global Support Services (GSS)
- Access other product documentation

If you have a suggestion for improving this guide or have found an error, please submit a Bugzilla report at http://bugzilla.redhat.com against **Product** for the **Documentation** component. Please provide specific details, such as the section number, guide name, and {product-title_short} version so we can

easily locate the content.

1.2. NEW AND CHANGED FEATURES

1.2.1. New features Technology Preview 2

The release adds the Kiali observability console to Red Hat OpenShift Service Mesh. Kiali provides a number of graphs that you can use to view the topography and health of the microservices that make up your service mesh. You can view predefined dashboards that provide detailed request and response metrics (volume, duration, size, TCP traffic) per inbound and outbound traffic. You can also browse your service mesh by application, workloads, and services to view the health of each element.

1.2.2. New features Technology Preview 1

Red Hat OpenShift Service Mesh provides a number of key capabilities uniformly across a network of services:

- **Traffic Management** - Control the flow of traffic and API calls between services, make calls more reliable, and make the network more robust in the face of adverse conditions.
- **Service Identity and Security** - Provide services in the mesh with a verifiable identity and provide the ability to protect service traffic as it flows over networks of varying degrees of trustworthiness.
- **Policy Enforcement** - Apply organizational policy to the interaction between services, ensure access policies are enforced and resources are fairly distributed among consumers. Policy changes are made by configuring the mesh, not by changing application code.
- **Telemetry** - Gain understanding of the dependencies between services and the nature and flow of traffic between them, providing the ability to quickly identify issues.

1.3. RESOLVED ISSUES

1.3.1. Fixed issues

MASITRA-21 - The default in the installer of 512Mi was too low for tracing. Increased default Elasticsearch memory from 512 MB to 1GB.

1.4. KNOWN ISSUES

1.4.1. Known issues

NOTE

While Kafka publisher is included in the release as part of Jaeger, it is not supported.

These known issues or limitations exist in Red Hat OpenShift Service Mesh at this time:

- Red Hat OpenShift Service Mesh does not support multi-tenancy.
- Red Hat OpenShift Service Mesh does not support IPv6, as it it not supported by the upstream Istio project, nor fully supported by OpenShift.

- The istio-init container requires a privileged security context or at least to run as root and to have the NET_ADMIN capability. The istio-init container needs to be privileged because it needs to properly configure the iptables rules in the pod in order to intercept network connections. The team is currently investigating ideas for ways to reduce the privileges required by Istio.

MAISTRA-4 - The uninstall does not remove all the files, and as a result, when you re-install the istio-operator installation fails because **`customresourcedefinitions.apiextensions.k8s.io "installations.istio.openshift.com"`** already exists.

Workaround - In order to cleanly remove the operator execute the following command:

```
oc process -f istio_product_operator_template.yaml | oc delete -f -
```

MAISTRA-5 - **`openshift-ansible-istio-installer-job`** pod tries to start but with errors.

Graph layout - The layout for the Kiali graph can render differently, depending on your application architecture and the data to display (number of graph nodes and their interactions). Because it is difficult if not impossible to create a single layout that renders nicely for every situation, Kiali offers a choice of several different layouts. To choose a different layout, you can choose a different **Layout Schema** from the **Graph Settings** menu.

MAISTRA-13 - Installer should determine release version from installed components. At present, the installer queries the yaml file, however if the yaml has been modified, the installer is not able to remove an older version.

Workaround - If the installation fails because of a previous, older installation, then use the following workaround:

```
$ oc delete namespace devex istio-operator istio-system bookinfo logging
foo bar

$ oc delete csr istio-sidecar-injector.istio-system

$ oc get crd  | grep istio | awk '{print $1}' | xargs oc delete crd

$ oc get mutatingwebhookconfigurations  | grep istio | awk '{print $1}' |
xargs oc delete mutatingwebhookconfigurations

$ oc get validatingwebhookconfiguration  | grep istio | awk '{print $1}' |
xargs oc delete validatingwebhookconfiguration

$ oc get clusterroles  | grep istio | awk '{print $1}' | xargs oc delete
clusterroles

$ oc get clusterroles  | grep kiali | awk '{print $1}' | xargs oc delete
clusterroles

$ oc get clusterrolebindings  | grep istio | awk '{print $1}' | xargs oc
delete clusterrolebindings

$ oc get clusterrolebindings  | grep kiali | awk '{print $1}' | xargs oc
delete clusterrolebindings
```

Kiali-507 Kiali does not support Internet Explorer 11. This is because the underlying frameworks do not support Internet Explorer. In order to access the Kiali console, use one of the two most recent versions of the Chrome, Edge, Firefox or Safari browser.

KIALI-1284 In Istio, a Workload can be any pod or group of pods, regardless where they originate from. They may come from Kubernetes Deployments, Replica Sets or even as a single "orphan" pod. In Kiali the current assumption is that a Workload comes from a Deployment. This should represent the vast majority of the cases.

KIALI-1570 When a graph is loading in the Kiali console, a message that the graph is empty is displayed instead of a message that the graph is loading.

KIALI-1572 If you see this ERROR message when you view the Kiali logs, you can ignore it: **`Failed to determine console version from file [/opt/kiali/console/version.txt]. error=open /opt/kiali/console/version.txt: no such file or directory Kiali: Console version: unknown`**

KIALI-1609 When dealing with very small values (for example, less than 0.0.1 rps) you might encounter some inconsistencies in the graph. We are working on making changes to have this function better when dealing with very small values.