

# Application Service Mesh

## FAQs

**Issue** 02  
**Date** 2022-10-09



**Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

# Security Declaration

## Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

---

# Contents

---

<b>1 Service Mesh Cluster</b>	<b>1</b>
1.1 Why Does a Service Mesh Remain in the Installing Status for a Long Time After I Enable It for a Cluster?	1
1.2 Why Does a Service Mesh Remain in the Unready Status for a Long Time After I Uninstall It?	1
1.3 Why Is an otel-collector Workload Created Alongside a Service Mesh?	2
<b>2 Mesh Management</b>	<b>7</b>
2.1 Why Cannot I Create a Mesh for My Cluster?	7
2.2 Why Are Exclusive Nodes Still Exist After Istio Is Uninstalled?	7
2.3 How Do I Upgrade ICAgent?	8
2.4 How Do I Enable Namespace Injection for a Cluster?	8
2.5 How Do I Disable Sidecar Injection for Workloads?	9
2.6 What Can I Do If A Pod Cannot Be Started Due to Unready Sidecar	10
2.7 How Do I Handle a Canary Upgrade Failure?	13
<b>3 Adding a Service</b>	<b>16</b>
3.1 What Do I Do If an Added Gateway Does Not Take Effect?	16
3.2 Why Does It Take a Long Time to Start the Demo Application in Experiencing Service Mesh in One Click?	16
3.3 Why Cannot I Access the page of the Demo Application After It Is Successfully Deployed?	17
3.4 Why Cannot I Select the Corresponding Service When Adding a Route?	17
3.5 How Do I Inject a Sidecar for the Pod Created Using a Job or CronJob?	17
<b>4 Performing Grayscale Release</b>	<b>19</b>
4.1 Why Can't I Change the Image Used for the Grayscale Version When Performing Grayscale Release?	19
4.2 Why Does Not a Grayscale Policy that Based on Request Content Take Effect for Some Services?	19
4.3 InvalidRequestBody Is Reported When a Grayscale Release Task Is Created for a Service with Multiple Ports	20
<b>5 Managing Traffic</b>	<b>21</b>
5.1 Why Are the Created Clusters, Namespaces, and Applications Not Displayed on the Traffic Management Page?	21
5.2 How Do I Change the Resource Requests of the <b>istio-proxy</b> Container?	21
5.3 Does ASM Support HTTP/1.0?	22
5.4 How Can I Block Access from Some IP Address Ranges or Ports for a Service Mesh?	23
5.5 How Do I Configure max_concurrent_streams for a Gateway?	26

---

5.6 How Do I Fix Compatibility Issues Between Istio CNI and Init Containers?.....	27
<b>6 Monitoring Traffic.....</b>	<b>28</b>
6.1 Why Cannot I View Traffic Monitoring Data Immediately After a Pod Is Started?.....	28
6.2 Why Are the Latency Statistics on the Dashboard Page Inaccurate?.....	28
6.3 Why Is the Traffic Ratio Inconsistent with That in the Traffic Monitoring Chart?.....	28
6.4 Why Can't I Find Certain Error Requests in Tracing?.....	28
6.5 Why Cannot I Find My Service in the Traffic Monitoring Topology?.....	29
6.6 How Do I Connect a Service Mesh to Jaeger or Zipkin for Viewing Traces?.....	29

# 1 Service Mesh Cluster

---

## 1.1 Why Does a Service Mesh Remain in the Installing Status for a Long Time After I Enable It for a Cluster?

### Symptom

After I create a service mesh (that is, buy a Dedicated mesh) for a CCE cluster, the mesh remains in the installing status for a long time and a message is displayed, indicating that the user security group rules are successfully enabled.

### Fault Diagnosis

Log in to the CCE console and choose **Resource Management > Namespaces**. Then, check whether the **istio-system** namespace of the cluster exists.

### Analysis

Residual **istio-system** namespaces exist.

### Solution

Delete the residual **istio-system** namespaces and install the mesh again.

## 1.2 Why Does a Service Mesh Remain in the Unready Status for a Long Time After I Uninstall It?

### Symptom

On the ASM console, after I uninstall a service mesh, the mesh remains in the unready status for a long time.

## Fault Diagnosis

**Step 1** Log in to the CCE console and click the cluster name to go to the cluster console. In the navigation pane on the left, choose **O&M > Charts**.

**Step 2** Click **Releases** and check the releases and the latest events about uninstallation failure.

The **Status** of **istio-master** is **Uninstallation Failed**, and the following message is displayed.

```
deletion failed with 1 error(s): clusterroles.rbac.authorization.k8s.io "istio-cleanup-secrets-istio-system" already exists
```

----End

## Analysis

Abnormal operations cause the Helm chart of Istio stuck during uninstallation. Residual resources lead to an uninstallation failure.

## Solution

**Step 1** Connect to the CCE cluster using kubectl.

**Step 2** Run the following commands to clear Istio resources:

```
kubectl delete ServiceAccount -n istio-system `kubectl get ServiceAccount -n istio-system | grep istio | awk '{print $1}'`  
kubectl delete ClusterRole -n istio-system `kubectl get ClusterRole -n istio-system | grep istio | awk '{print $1}'`  
kubectl delete ClusterRoleBinding -n istio-system `kubectl get ClusterRoleBinding -n istio-system | grep istio | awk '{print $1}'`  
kubectl delete job -n istio-system `kubectl get job -n istio-system | grep istio | awk '{print $1}'`  
kubectl delete crd -n istio-system `kubectl get crd -n istio-system | grep istio | awk '{print $1}'`  
kubectl delete mutatingwebhookconfigurations -n istio-system `kubectl get mutatingwebhookconfigurations -n istio-system | grep istio | awk '{print $1}'`
```

**Step 3** Log in to the ASM console and uninstall the mesh again.

----End

## 1.3 Why Is an otel-collector Workload Created Alongside a Service Mesh?

### Symptom

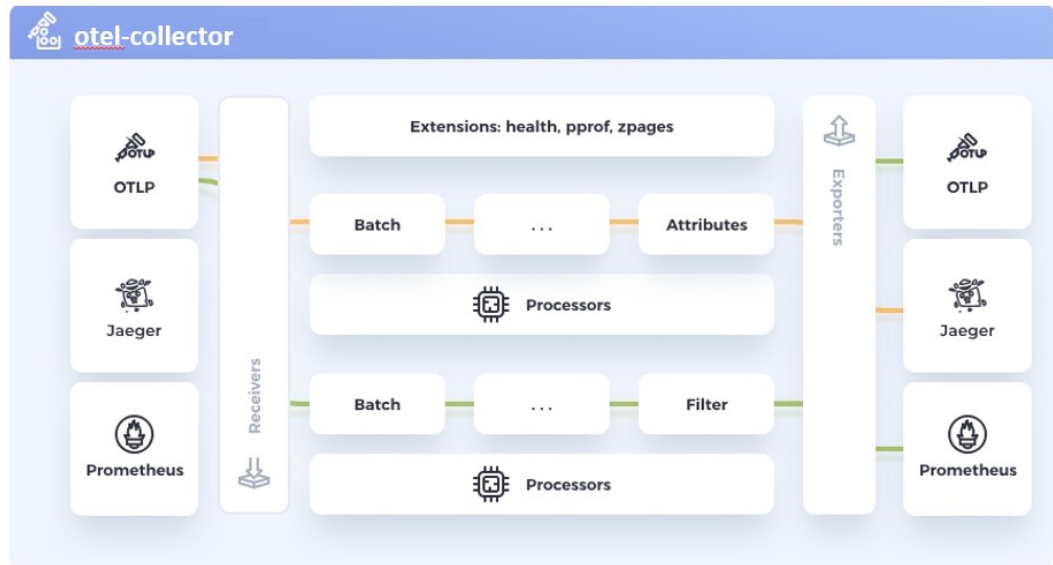
An otel-collector workload is automatically created when a service mesh is created.

### Analysis

After a cluster is connected to a service mesh, an otel-collector workload is automatically created in the **monitoring** namespace to collect telemetry data (traces, logs, and metrics) from Envoy proxies, process the data, and export the data to one or more backends for mesh observability.

#### otel-collector Architecture

**Figure 1-1** otel-collector architecture



otel-collector consists of four modules:

- **Receivers**  
A receiver, which can be push or pull based, is how data gets into otel-collector. Receivers can receive telemetry data in multiple formats, such as OTLP, Jaeger, and Prometheus in the preceding figure.
- **Processors**  
Processors process data collected by receivers. For example, a common batch processor processes telemetry data in batches.
- **Exporters**  
An exporter is how you send telemetry data to one or more backends. It allows a visual display of telemetry data for data analysis.
- **Extensions**  
Extensions are available primarily for tasks that do not involve processing telemetry data. Extensions are optional. For example, you can add the health\_check extension to check the health of otel-collector.

### Using otel-collector on ASM of the Basic Edition

You can run the following command to obtain the settings of otel-collector.



```
[root@1001-209-0040 ~]# kubectl get cm -n monitoring otel-collector-conf -oyaml
apiVersion: v1
data:
  otel-collector-config: |-
    receivers:
      zipkin: { }
      prometheus:
        config:
          scrape_configs:
            - job_name: 'istio-mesh'
              scrape_interval: 15s
              metrics_path: /stats/prometheus
              kubernetes_sd_configs:
                - role: pod
              relabel_configs:
                - source_labels: [ __meta_kubernetes_pod_container_port_name ]
                  action: keep
                  regex: http-envoy-prom
            metric_relabel_configs:
              - source_labels: [ __name__ ]
                action: keep
                regex: istio.*
              - source_labels: [ __name__ ]
                regex: 'istio_build'
                action: drop
              - source_labels: [ __name__ ]
                regex: 'istio_response_bytes.*'
                action: drop
              - source_labels: [ __name__ ]
                regex: 'istio_request_bytes.*'
                action: drop
    processors:
      batch:
      memory_limiter:
```

The following uses the configuration file obtained from ASM of the basic edition as an example:

- **receivers** defines that telemetry data can be obtained from Envoy proxies using **zipkin** and **prometheus**. **prometheus** defines that data is captured from **/stats/prometheus** every 15 seconds.

```
otel-collector-config: |-
  receivers:
    zipkin: { }
    prometheus:
      config:
        scrape_configs:
          - job_name: 'istio-mesh'
            scrape_interval: 15s
            metrics_path: /stats/prometheus
            kubernetes_sd_configs:
              - role: pod
            relabel_configs:
              - source_labels: [ __meta_kubernetes_pod_container_port_name ]
                action: keep
                regex: http-envoy-prom
          metric_relabel_configs:
            - source_labels: [ __name__ ]
              action: keep
              regex: istio.*
            - source_labels: [ __name__ ]
              regex: 'istio_build'
              action: drop
            - source_labels: [ __name__ ]
              regex: 'istio_response_bytes.*'
              action: drop
            - source_labels: [ __name__ ]
              regex: 'istio_request_bytes.*'
              action: drop
```

- **processors** defines two data processing modes: batch and memory\_limiter.

```
processors:  
  batch:  
  memory_limiter:  
    check_interval: 1s  
    limit_percentage: 80  
    spike_limit_percentage: 20
```

- **exporters** defines how processed telemetry data is exported to the APM server.

```
exporters:  
  apm:  
    address: "100.79.1.215:8923"  
    project_id: 719217bc273743ea8d7ac1ae8bc34480  
    cluster_id: d7491b95-5111-11ee-8779-0255ac100b05
```

- **extensions** defines the health\_check extension, which is used to check the health of otel-collector.

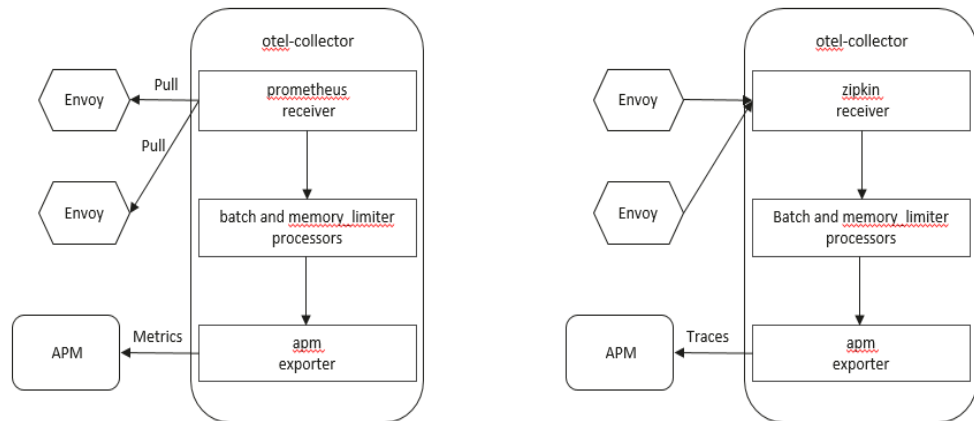
```
extensions:  
  health_check:  
    endpoint: 127.0.0.1:13133
```

- **service** is used to configure the preceding defined configuration items used by otel-collector.

```
service:  
  telemetry:  
    logs:  
      level: info  
    extensions: [ health_check ]  
  pipelines:  
    metrics/apm:  
      receivers: [ prometheus ]  
      processors: [ memory_limiter, batch ]  
      exporters: [ apm ]  
    traces/apm:  
      receivers: [ zipkin ]  
      processors: [ memory_limiter, batch ]  
      exporters: [ apm ]
```

For example, in the preceding configuration file, two pipelines are configured for processing metrics and traces. (A pipeline consists of a receiver, a processor, and an exporter.) The log level is set to INFO or higher. The following figure shows the architecture used for processing metrics and traces.

Figure 1-2 Metrics and traces processing architecture



## Solution

No action is required.

# 2 Mesh Management

---

## 2.1 Why Cannot I Create a Mesh for My Cluster?

### Symptom

I cannot create a mesh for my cluster.

### Analysis

Currently, clusters of versions earlier than 1.15 cannot be managed by meshes.

### Solution

- Step 1** Check the cluster version. Currently, only clusters of v1.15, v1.17, v1.19, v1.21, or v1.23 can be managed by meshes.
- Step 2** Check your browser. Chrome is recommended. The button for mesh creation may be unavailable when you are using other browsers, such as Firefox, due to adaptation problems.

----End

## 2.2 Why Are Exclusive Nodes Still Exist After Istio Is Uninstalled?

### Symptom

After Istio is uninstalled, exclusive nodes still exist.

### Analysis

Only Istio control plane workloads will be deleted when you uninstall Istio for a cluster. Node resources will not be deleted automatically.

## Solution

Nodes from which Istio are uninstalled can be used as common nodes. If these nodes are no longer required, log in to the CCE console and click the cluster name to go to the cluster console. Then, choose **Cluster** > **Nodes** to delete the nodes.

## 2.3 How Do I Upgrade ICAgent?

- Step 1** Log in to the ASM console. In the navigation pane, choose **Monitoring Center** to go to the APM console.
- Step 2** On the APM console, choose **Agent** > **Management** in the navigation pane, select the target cluster, and click **Upgrade ICAgent**.

----End

## 2.4 How Do I Enable Namespace Injection for a Cluster?

When injecting a sidecar to the namespace of a cluster, if the namespace injection is not enabled in the cluster, perform the following steps:

- Step 1** Connect to the cluster using `kubectl`.
- Step 2** Run the `kubectl get iop -nistio-system` command to query iop resources.

- If the following information is displayed, the iop resource exists. Go to [Step 3](#).

```
user@dts2fot109u4ymb-machine:~$ kubectl get iop -nistio-system
NAME          REVISION  STATUS  AGE
data-plane    1         HEALTHY 69d
```

- If the following information is displayed, no iop resources exist. Go to [Step 4](#).

```
web-terminal-7b778fc945-9m2hf:~# kubectl get iop -nistio-system
No resources found in istio-system namespace.
```

- Step 3** Run the `kubectl edit iop -nistio-system data-plane` command to modify the `autoInject` configuration item. In the preceding command, `data-plane` indicates the name of the iop resource queried in the previous step. Replace it with the actual value.

```
global:
  defaultPodDisruptionBudget:
    enabled: true
  hub: ***:20202/asm
  logging:
    level: default:info
  meshID: test-payment
  multiCluster:
    clusterName: test-yy
    network: test-yy-network
  proxy:
    autoInject: enabled
  remotePilotAddress: **:*
  tag: 1.8.6-r1-20220512225026
```

- Step 4** Run the `kubectl edit cm -nistio-system istio-sidecar-injector` command to modify the `istio-sidecar-injector` configuration item.

```
data:
  config: |-
    policy: enabled
```

----End

## 2.5 How Do I Disable Sidecar Injection for Workloads?

If sidecar injection is enabled for a namespace of a cluster, sidecars are automatically injected for the pods of all workloads in the namespace. To prevent sidecars from being injected for some workloads, perform the following operations:

- Step 1** Log in to the CCE console and click the cluster name to go to the cluster console. In the navigation pane on the left, choose **Resources > Workloads**.
- Step 2** Locate the workload and click **Edit YAML** in the **Operation** column.
- Step 3** Locate the target field based on the service mesh version and add **sidecar.istio.io/inject: 'false'**.

- For service meshes earlier than 1.13

Locate the **spec.template.metadata.annotations** field and add **sidecar.istio.io/inject: 'false'**.

```
annotations:
  sidecar.istio.io/inject: 'false'
```

```
107 spec:
108   replicas: 1
109   selector:
110     matchLabels:
111       app: reviews
112       version: v1
113   template:
114     metadata:
115       creationTimestamp: null
116     labels:
117       app: reviews
118       release: istio-bookinfo
119       version: v1
120     annotations:
121       sidecar.istio.io/inject: 'false'
```

- For service meshes 1.13 or later:

Locate the **spec.template.metadata.label** field and add **sidecar.istio.io/inject: 'false'**.

```
label:
  sidecar.istio.io/inject: 'false'
```

```
139 * spec:
140   replicas: 1
141 * selector:
142   matchLabels:
143     app: nginx
144     version: v1
145 * template:
146   metadata:
147     creationTimestamp: null
148   labels:
149     app: nginx
150     sidecar.istio.io/inject: 'false'
151     version: v1
152 * annotations:
153   asm/updateTimestamp: '2024-07-19T06:50:15Z'
154 * spec:
155   containers:
156   - name: container-1
157     image: swr.cn-north-4.myhuaweicloud.com/paas_1/nginx:curl
158   env:
159   - name: P
160     value: nginx
```

For more details about sidecar injection, see [Automatic Sidecar Injection](#).

----End

## 2.6 What Can I Do If A Pod Cannot Be Started Due to Unready Sidecar

### Description

Pods of services managed by a mesh may fail to be started and keep restarting. When the service container communicates with external systems, the traffic passes through the **istio-proxy** container. However, the service container is started earlier than the **istio-proxy** container. As a result, the communication with external systems fails and the pod keeps restarting.

### Solution

In Istio 1.7 and later versions, the community adds a switch named **HoldApplicationUntilProxyStarts** to the **istio-injector** injection logic. After the switch is enabled, the proxy is injected to the first container and the **istio-proxy** container is started earlier than the service container.

The switch can be configured globally or locally. The following describes two ways to enable the switch.

---

#### NOTICE

After this switch is enabled, the service container cannot be started until the sidecar is fully ready, which slows down pod startup and reduces scalability for burst traffic. You are advised to evaluate service scenarios and enable this switch only for required services.

---

- **Global Configuration**
  - a. Run the following command to edit the IOP CR resource:  
**kubectl edit iop private-data-plane -n istio-system**

Add the following command to the `spec.values.global.proxy` field:

```
holdApplicationUntilProxyStarts: true
```

```
values:
  gateways:
    istio-egressgateway:
      autoscaleEnabled: false
      labels:
        app: istio-egressgateway
      tolerations:
        - effect: NoExecute
          key: istio
          operator: Exists
    istio-ingressgateway:
      autoscaleEnabled: false
      customService: true
      labels:
        app: istio-ingressgateway
      replicaCount: 1
      tolerations:
        - effect: NoExecute
          key: istio
          operator: Exists
  global:
    defaultPodDisruptionBudget:
      enabled: true
    hub: swr.cn-north-7.myhuaweicloud.com/asm
    logging:
      level: default:info
    meshID: envoy-critical
    multiCluster:
      clusterName: test-yyl-multi
    proxy:
      autoInject: enabled
      holdApplicationUntilProxyStarts: true
```

- b. Run the following command to check whether the latest logs contain no error information:

```
kubectl logs -n istio-operator $(kubectl get po -n istio-operator | awk '{print $1}' | grep -v NAME)
```

- c. Run the following command to check whether the IOP CR is normal:

```
kubectl get iop -n istio-system
```

```
[root@lx666-14467 ~]# kubectl get iop -n istio-system
NAME                REVISION  STATUS  AGE
private-data-plane  1         HEALTHY 6d2h
[root@lx666-14467 ~]#
```

- d. Run the following command to upgrade the services in the mesh in a rolling manner:

```
kubectl rollout restart deployment nginx -n default
```

where, `nginx` is an example service, and `default` is the namespace. Replace them with the actual values.

- e. Run the following command to check whether the pod is restarted:

```
kubectl get pod -n default | grep nginx
```



```
[root@lx666-14467 ~]# kubectl get pod -n default | grep nginx
nginx-6b4959ffff-pr8t8    2/2    Running    0    14s
[root@lx666-14467 ~]#
```

- f. Run the following command to check whether **postStart lifecycle** is added to the pod and whether the **istio-proxy** container is placed in the first position:

```
kubectl edit pod nginx-7bc96f87b9-l4dbl
```

```
- name: ISTIO_META_CLUSTER_ID
  value: test-yy1-multi
image: swr.cn-north-7.myhuaweicloud.com/asm/proxyv2:1.13.9-r1-20221110212800
imagePullPolicy: IfNotPresent
lifecycle:
  postStart:
    exec:
      command:
        - pilot-agent
        - wait
name: istio-proxy
ports:
```

- **Local Configuration**

For Istio 1.8 or later versions, you can label the pods for which this function needs to be enabled with **proxy.istio.io/config** and set **holdApplicationUntilProxyStarts** to true.

The following uses the **nginx** service in the **default** namespace as an example. The operations for other services are similar.

```
kubectl edit deploy nginx -n default
```

Add the following commands to the **spec.template.metadata.annotations** field:

```
proxy.istio.io/config: |
  holdApplicationUntilProxyStarts: true
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "6"
    description: ""
  creationTimestamp: "2022-11-24T07:55:31Z"
  generation: 6
  labels:
    appgroup: ""
    version: v1
  name: tomcat
  namespace: default
  resourceVersion: "55550644"
  uid: cd5dbfe8-83cc-4964-86fc-f657c85e852d
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: tomcat
      version: v1
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      annotations:
        kubectrl.kubernetes.io/restartedAt: "2022-11-25T10:35:02+08:00"
        proxy.istio.io/config: |
          holdApplicationUntilProxyStarts: true
    creationTimestamp: null
```

## 2.7 How Do I Handle a Canary Upgrade Failure?

There are many reasons for a canary upgrade failure. In case of a canary upgrade failure, you can use the following solutions to handle it.

1. Failed to check custom resource definitions (CRDs) before the upgrade.

**Solution:** New Istio version does not support some CRDs, including ClusterRbacConfigs, ServiceRoles, ServiceRoleBindings, and Policies. If there are resources to be discarded in the current version, delete them before the upgrade.

2. Failed to check Istio gateway labels before the upgrade.

**Solution:** Configure Istio gateway labels (specified by **matchLabels**) in *{app: istio-ingressgateway, istio: ingressgateway}* format.

3. Failed to check add-ons before the upgrade.

**Solution:** ASM 1.8 and later versions do not support the tracing, kiali, grafana, and prometheus add-ons. Uninstall the add-ons before the upgrade. You can install open-source add-ons or use APM.

4. Failed to check the cluster status before the upgrade.

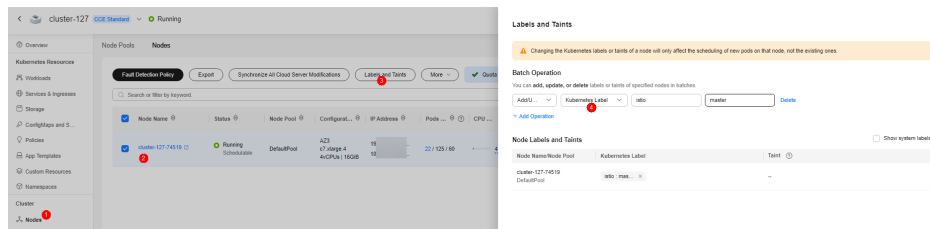
**Solution:** If the cluster is unavailable before the upgrade, do not perform the upgrade.

5. Failed to query resources before the upgrade.  
**Solution:** Prepare the required resources for the canary upgrade.
6. Failed to check the cluster version before the upgrade.  
**Solution:** Use the cluster version listed in the following table.

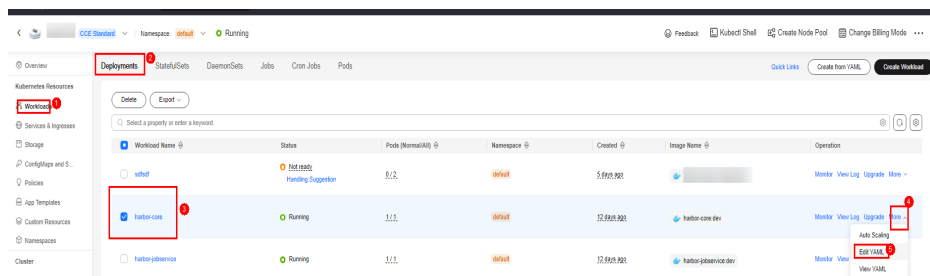
Service Mesh Version	Supported Cluster Version
1.3	1.13, 1.15, 1.17, and 1.19
1.6	1.15 and 1.17
1.8	1.15, 1.17, 1.19, and 1.21
1.13	1.21 and 1.23
1.15	1.21, 1.23, 1.25, and 1.27
1.18	1.25, 1.27, and 1.28

7. Failed to check the component affinity before the upgrade.  
**Solutions:** If you upgrade Istio from a non-canary version to a canary version, ensure that there are at least twice as many nodes labeled with **istio:master** as there are istiod instances, and at least twice as many schedulable nodes as there are istio-ingressgateway or istio-egressgateway instances (depending on which one is larger). If such conditions are not met, add nodes to meet the scheduling requirements or set the anti-affinity of istiod, istio-ingressgateway, and istio-egressgateway to **Preferred**.

- Method 1: Add nodes labeled with **istio:master** on the CCE console.



- **Solution 2:** Edit the YAML file to modify the anti-affinity policy on the CCE console.

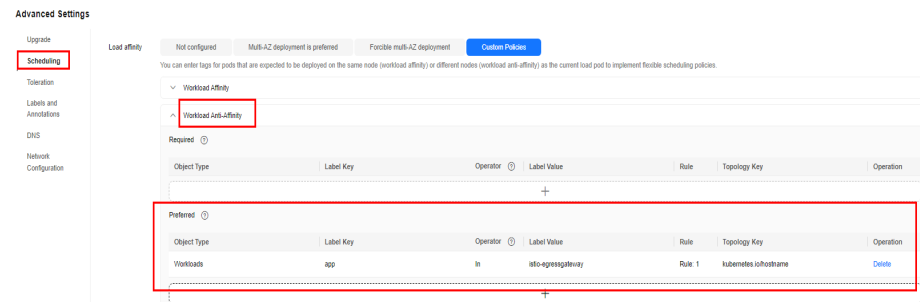


preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1
- podAffinityTerm:
  - labelSelector:
    - matchExpressions:
      - key: app
      - operator: In
      - values:

- istiod (For the anti-affinity policy of istio-ingressgateway, replace the value of **values** with **istio-ingressgateway**. For the anti-affinity policy of istio-egressgateway, replace the value of **values** with **istio-egressgateway**.)  
namespaces:  
- istio-system  
topologyKey: kubernetes.io/hostname

Alternatively, change the anti-affinity from **Required** to **Preferred** on the CCE console.



8. Failed to check the automatic namespace injection before the upgrade.

**Solution:** If there are pods in the namespace when you migrate service mesh data from the Dedicated edition to the Basic edition, enable **automatic injection** for the namespace.

# 3 Adding a Service

---

## 3.1 What Do I Do If an Added Gateway Does Not Take Effect?

The possible cause is that the Gateway-related resource configurations are missing or incorrect. Do as follows to locate the fault:

- Log in to the Elastic Load Balance console, check whether the external port and ECSs are successfully listened by the load balancer.
- Log in to the cluster and run the **kubectl get gateway -n istio-system** command to check whether the IP address, domain name, and port number are configured for the Gateway. Run the **kubectl get svc -n istio-system** command to check whether the ingress Gateway has the corresponding IP address and port and is not in the pending status.
- Check whether the internal access protocol of the service added to the service mesh is consistent with the external access protocol configured for the service's Gateway.
- If the **ERR\_UNSAFE\_PORT** error is displayed when you use a browser to access the service, that is because the port is identified as a dangerous port by the browser. In this case, you need to use another external port.

## 3.2 Why Does It Take a Long Time to Start the Demo Application in Experiencing Service Mesh in One Click?

The demo application contains the productpage, details, ratings, and reviews services. All related workloads and Istio resources including DestinationRule, VirtualService, and Gateway need to be created. Therefore, the creation takes a comparatively long time.

## 3.3 Why Cannot I Access the page of the Demo Application After It Is Successfully Deployed?

### Symptom

The page of the demo application cannot be accessed after the application is successfully deployed.

### Analysis

The load balancer configured for the application does not listen to the port.

### Solution

Log in to the Elastic Load Balance console. Check whether the port listener has been created and whether the health status of the backend server is normal. For details about how to create a load balancer, see [Listener](#).

## 3.4 Why Cannot I Select the Corresponding Service When Adding a Route?

During adding a route, the target service is filtered based on the corresponding gateway protocol. The filtering rules are as follows:

- For an HTTP gateway, select an HTTP service.
- For a TCP gateway, select a TCP service.
- For a gRPC gateway, select a gRPC service.
- For an HTTPS gateway, select either an HTTP or a gRPC service.
- For a TLS gateway which TLS termination is enabled, select a TCP service. If TLS termination for a TLS gateway is disabled, select a TLS service.

## 3.5 How Do I Inject a Sidecar for the Pod Created Using a Job or CronJob?

### Prerequisites

- Ensure that ASM 1.15.5-r3 or later is used to create service meshes.
- By default, the sidecar is not injected for the pod created using a job or CronJob. If sidecar injection is required, choose **Labels and Annotations** in **Advanced Settings** and set **sidecar.istio.io/inject** to **true** for **Pod Label**.

Advanced Settings

Labels and Annotations

Pod Label: sidecar.istio.io/inject = true [Confirm]

Network Configuration

Pod Annotation: Key = Value [Confirm] [Quick Links](#)

The following is an example CronJob:

```
kind: CronJob
apiVersion: batch/v1
metadata:
  name: mycronjob
  namespace: default
spec:
  schedule: */1 * * * *
  jobTemplate:
    spec:
      template:
        metadata:
          creationTimestamp: null
        labels:
          app: mycronjob
          sidecar.istio.io/inject: 'true'
...

```

- Before using a job or CronJob, you need to run a specific command in the target container to exit the sidecar.

## Exiting a Sidecar After a Job or CronJob Is Complete

Call `curl -sf -XPOST http://127.0.0.1:15000/quitquitquit` to exit istio-proxy after a job or CronJob is complete.

The following is an example CronJob:

```
kind: CronJob
apiVersion: batch/v1
metadata:
  name: mycronjob
  namespace: default
spec:
  schedule: */1 * * * *
  concurrencyPolicy: Forbid
  suspend: false
  jobTemplate:
    metadata:
      creationTimestamp: null
    spec:
      template:
        metadata:
          creationTimestamp: null
        labels:
          app: cronjob1
          sidecar.istio.io/inject: 'true'
          version: v1
        spec:
          containers:
            - name: mycronjob-1
              image: 'busybox:latest'
              command:
                - /bin/bash
                - '-c'
              args:
                - |
                  trap "curl --max-time 2 -s -f -XPOST http://127.0.0.1:15000/quitquitquit" EXIT
                  while ! curl -s -f http://127.0.0.1:15020/healthz/ready; do sleep 1;done
                  sleep 2
                  date; echo Hello from the Kubernetes cluster<Your Job command>
...

```

# 4 Performing Grayscale Release

---

## 4.1 Why Can't I Change the Image Used for the Grayscale Version When Performing Grayscale Release?

### Description

When I perform grayscale release, the image used for the grayscale version cannot be changed.

### Analysis

When performing grayscale release on a service, you create a new version of the same service. Therefore, the image used by the service cannot be changed. Only image tags can be changed.

### Solution

Pack the required image into a different tag of the same image and push it to the image repository. Then, select the newly pushed image tag when you perform grayscale release on the service.

## 4.2 Why Does Not a Grayscale Policy that Based on Request Content Take Effect for Some Services?

### Symptom

A grayscale policy that based on request content does not take effect on some services.

### Analysis

A grayscale policy based on request content is valid only for the entry service that is directly accessed.



## Solution

If you want a grayscale policy to be applied to all services in an application, the header information of the HTTP request needs to be transferred in the service code.

## 4.3 InvalidRequestBody Is Reported When a Grayscale Release Task Is Created for a Service with Multiple Ports

### Symptom

When a grayscale release task is created for a service with multiple ports, "ASM.0002 InvalidRequestBody" is reported.

### Fault Location

Log in to the ASM console, press **F12**, and switch to the **Network** tab to view APIs. Error code 400 is returned when each POST request is sent to create a grayscale release task. The returned information is as follows:

```
some ports of the service have been configured with routes, ports=[%v]
```

### Analysis

Some ports are deleted from the service that is properly configured. For example, service01 has two ports 80 and 81, and port 81 is deleted on the CCE console.

### Solution

Restore the deleted service ports.

# 5 Managing Traffic

---

## 5.1 Why Are the Created Clusters, Namespaces, and Applications Not Displayed on the Traffic Management Page?

1. Check whether Istio has been enabled for your cluster.
2. Check whether at least one service has been added to the **Service List** page and is in the **Running** state.
3. Check whether you have uninstalled the ICAgent in the cluster.

## 5.2 How Do I Change the Resource Requests of the istio-proxy Container?

The default resources of the **istio-proxy** container are as follows. You can change the resources as required.

```
resources:  
limits:  
  cpu: "2"  
  memory: 512Mi  
requests:  
  cpu: "1"  
  memory: 512Mi
```

### Method 1: Modify the Configuration for All Services in the Mesh

Do as follow to change the resource requests configuration of the **istio-proxy** container for all services in the mesh at a time:

**Step 1** Run the following command to modify the ConfigMap:

```
kubectl edit cm istio-sidecar-injector -n istio-system
```

```

315     resources:
316     {{ if or (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyCPU`) (isset .ObjectMeta
ar.istio.io/proxyLimitCPU`) (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyLimitMemory`) -
317     requests:
318     {{ if (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyCPU`) -}}
319     cpu: "{{ index .ObjectMeta.Annotations `sidecar.istio.io/proxyCPU` }}"
320     {{ end }}
321     {{ if (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyMemory`) -}}
322     memory: "{{ index .ObjectMeta.Annotations `sidecar.istio.io/proxyMemory` }}"
323     {{ end }}
324     limits:
325     {{ if (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyLimitCPU`) -}}
326     cpu: "{{ index .ObjectMeta.Annotations `sidecar.istio.io/proxyLimitCPU` }}"
327     {{ end }}
328     {{ if (isset .ObjectMeta.Annotations `sidecar.istio.io/proxyLimitMemory`) -}}
329     memory: "{{ index .ObjectMeta.Annotations `sidecar.istio.io/proxyLimitMemory` }}"
330     {{ end }}
331     {{ else -}}
332     {{- if .Values.global.proxy.resources }}
333     #{{ toYaml .Values.global.proxy.resources | indent 4 }}
334     requests:
335     cpu: xxx
336     memory: xxx
337     limits:
338     cpu: xxx
339     memory: xxx
340     {{- end }}
341     {{ end -}}

```

**Step 2** Restart the `istio-sidecar-injector` pod in the `istio-system` namespace.

**Step 3** Restart service pods in the rolling upgrade mode to avoid service interruption.

----End

## Method 2: Modify the Configuration for A Specific Service in the Mesh

**Step 1** Modify the YAML file of the service.

```
kubectl edit deploy <nginx> -n <namespace>
```

**Step 2** Add the following configuration lines to `spec.template.metadata.annotations` (you can change the resource size as required).

```

sidecar.istio.io/proxyCPU: 500m
sidecar.istio.io/proxyLimitCPU: 500m
sidecar.istio.io/proxyLimitMemory: 1024Mi
sidecar.istio.io/proxyMemory: 1024Mi

```

For meshes of Istio 1.8, add the following configuration lines:

```

sidecar.istio.io/proxyCPU: 500m
sidecar.istio.io/proxyCPULimit: 500m
sidecar.istio.io/proxyMemoryLimit: 1024Mi
sidecar.istio.io/proxyMemory: 1024Mi

```

**Step 3** After the modification, restart service pods in the rolling upgrade mode to avoid service interruption.

----End

## 5.3 Does ASM Support HTTP/1.0?

### Symptom

By default, Istio does not support HTTP/1.0.

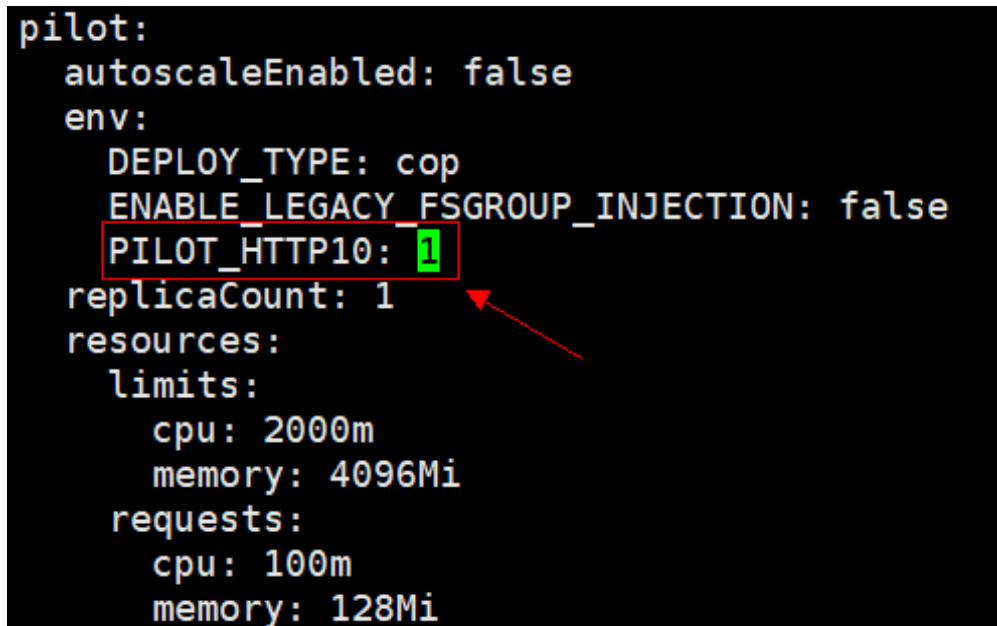
## Analysis

In Istio, Envoy forwards traffic and Pilot allocates rules. **PILOT\_HTTP10** of Pilot is set to **0** by default. This means **HTTP/1.0** is not supported by default.

## Solution

Set `.values.pilot.env.PILOT_HTTP10` to **1** in the `.iop` file and configure **PILOT\_HTTP10** for `pilot`.

```
pilot:
  autoscaleEnabled: false
  env:
    DEPLOY_TYPE: cop
    ENABLE_LEGACY_FSGROUP_INJECTION: false
    PILOT_HTTP10: 1
  replicaCount: 1
  resources:
    limits:
      cpu: 2000m
      memory: 4096Mi
    requests:
      cpu: 100m
      memory: 128Mi
```



## 5.4 How Can I Block Access from Some IP Address Ranges or Ports for a Service Mesh?

### Scenarios

In some scenarios, you may want to specify IP address ranges that need to be blocked by a service mesh proxy. In some scenarios, you may want to specify ports to block requests. This section describes how to block access from some IP address ranges and ports.

### Workload Configuration for Blocking Access from Some IP Address Ranges

Modify the `deployment` file to block some IP address ranges.

Run the `kubectl edit deploy -n user_namespace user_deployment` command.

1. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/includeOutboundIPRanges` to specify IP address ranges to be blocked.

```
type: RollingUpdate
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-03-23T03:49:21Z"
      sidecar.istio.io/proxyCPU: "0.1"
      sidecar.istio.io/proxyCPULimit: "2"
      sidecar.istio.io/proxyMemory: 128Mi
      sidecar.istio.io/proxyMemoryLimit: 2048Mi
      traffic.sidecar.istio.io/includeOutboundIPRanges: 192.168.0.1/24
    creationTimestamp: null
  labels:
    app: nginx
    version: v1
```

2. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/excludeOutboundIPRanges` to specify IP address ranges that are allowed.

```
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-03-23T03:49:21Z"
      sidecar.istio.io/proxyCPU: "0.1"
      sidecar.istio.io/proxyCPULimit: "2"
      sidecar.istio.io/proxyMemory: 128Mi
      sidecar.istio.io/proxyMemoryLimit: 2048Mi
      traffic.sidecar.istio.io/excludeOutboundIPRanges: 192.168.0.1/24
    creationTimestamp: null
  labels:
    app: nginx
    version: v1
```

Note: The preceding operations will cause rolling upgrades of service containers.

## Workload Configuration for Blocking Ingress and Egress Traffic over Some Ports

Modify the `deployment` file to block ingress and egress traffic over some ports.

Run the `kubectl edit deploy -n user_namespace user_deployment` command.

1. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/excludeInboundPorts` to specify the ports that allow the ingress traffic.

```
type: RollingUpdate
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-06-01T01:40:56Z"
      traffic.sidecar.istio.io/excludeInboundPorts: 3306,6379
    creationTimestamp: null
  labels:
    app: echo
```

2. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/includeInboundPorts` to specify the ports that block the ingress traffic.

```
type: RollingUpdate
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-06-01T01:40:56Z"
      traffic.sidecar.istio.io/includeInboundPorts: 3306,6379
    creationTimestamp: null
    labels:
      app: echo
```

3. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/excludeOutboundPorts` to specify the ports that allow the egress traffic.

```
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-06-01T01:40:56Z"
      traffic.sidecar.istio.io/excludeOutboundPorts: 3306,6379
    creationTimestamp: null
    labels:
```

4. In `deployment.spec.template.metadata.annotations`, use `traffic.sidecar.istio.io/includeOutboundPorts` to specify the ports that block the egress traffic.

```
type: RollingUpdate
template:
  metadata:
    annotations:
      asm/updateTimestamp: "2023-06-01T01:40:56Z"
      traffic.sidecar.istio.io/includeOutboundPorts: 3306,6379
    creationTimestamp: null
    labels:
      app: echo
      sidecarVersion: 1.13.9-r1-1685522112
```

Note: The preceding operations will cause rolling upgrades of service containers.

## Verification

The configurations take effect in iptables of containers. Run the following commands to check whether the configurations take effect.

1. Log in to the node where the workload is running and run the `docker ps` command to find the pause container and view the container ID.
2. Run the `docker inspect <CONTAINER_ID> | grep -i pid` command to view the process ID.
3. Run the `nsenter -t <PID> -n bash` command to go to the namespace of the container.
4. Run the `iptables iptables -t nat -L -n -v` command to check whether the configurations take effect for specified IP address ranges and ports.



## 5.5 How Do I Configure max\_concurrent\_streams for a Gateway?

**Step 1** Log in to any node in the cluster where the gateway is located and run the following command to create resources:

```
cat>"stream-limit-envoyfilter.yaml"<<EOF
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: http2-stream-limit
  namespace: istio-system
spec:
  workloadSelector:
    labels:
      istio: ingressgateway
  configPatches:
    - applyTo: NETWORK_FILTER # http connection manager is a filter in Envoy
      match:
        context: GATEWAY
        listener:
          filterChain:
            filter:
              name: "envoy.filters.network.http_connection_manager"
      patch:
        operation: MERGE
        value:
          typed_config:
            "@type": "type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionManager"
            http2_protocol_options:
              max_concurrent_streams: 128
EOF
```

### NOTE

The **max\_concurrent\_streams** parameter indicates the maximum number of concurrent streams of a gateway. You can set this parameter as required.

**Step 2** Run the **kubectl apply -f stream-limit-envoyfilter.yaml** command to create an EnvoyFilter.

```
root@ecs-guobaoqing-0054:~# kubectl get envoyfilter -nistio-system
NAME          AGE
http2-stream-limit 8s
```

----End

## 5.6 How Do I Fix Compatibility Issues Between Istio CNI and Init Containers?

### Context

The Istio CNI plugin may cause network connectivity issues for init containers. When using Istio CNI, kubelet starts a pod with the following steps:

- Step 1** The Istio CNI plugin sets up traffic redirection to the Istio sidecar within the pod.
- Step 2** All init containers execute and complete successfully.
- Step 3** The Istio sidecar starts in the pod along with the pod's other containers.

----End

Init containers execute before the sidecar starts. This means any requests sent by init containers are redirected to the sidecar that is not started. This results in traffic loss during the init containers' execution.

### Solutions

You can use any of the following methods to avoid this traffic loss:

- Set the UID of the init container to **1337** using **runAsUser**. **1337** is the UID used by the sidecar. The traffic sent by this UID is not captured by the Istio's iptables rule. Application container traffic is still be captured as usual.
- Set the **traffic.sidecar.istio.io/excludeOutboundIPRanges** annotation for the CIDR that the init container communicates with to prevent the traffic from being redirected to the sidecar.
- Set the **traffic.sidecar.istio.io/excludeOutboundPorts** annotation for the port that the init container uses to prevent the traffic from being redirected to the sidecar.

---

 **CAUTION**

Use the IP/port exclusion annotations with caution because the annotations apply to both init container traffic and application container traffic. Application traffic sent to the configured IP address or port will bypass the Istio sidecar.

---

For details, visit <https://istio.io/latest/docs/setup/additional-setup/cni/>.



# 6 Monitoring Traffic

---

## 6.1 Why Cannot I View Traffic Monitoring Data Immediately After a Pod Is Started?

1. Check whether APM has been enabled for the cluster.
2. Traffic monitoring aggregates the collected data. Please wait for a minute for the data to be displayed on the **Traffic Monitoring** page.

## 6.2 Why Are the Latency Statistics on the Dashboard Page Inaccurate?

The latency statistics displayed on the **Dashboard** page are data of the services that have the highest latency among all the services in all the clusters of your account within the last one minute. Therefore, ensure that the service has been accessed within the last one minute.

## 6.3 Why Is the Traffic Ratio Inconsistent with That in the Traffic Monitoring Chart?

The traffic ratio data is polled every 10 seconds, while the traffic monitoring data shows the traffic situation of the last 10 seconds.

## 6.4 Why Can't I Find Certain Error Requests in Tracing?

For performance purposes, the sampling rate of tracing is 10%. That is, 10 of your 100 requests are recorded and displayed on the page.

## 6.5 Why Cannot I Find My Service in the Traffic Monitoring Topology?

1. Select a mesh, cluster, and namespace to monitor service traffic.
2. Check whether the ICAgent collector is correctly installed in the cluster.
3. Check whether the service has been added to the service mesh.

## 6.6 How Do I Connect a Service Mesh to Jaeger or Zipkin for Viewing Traces?

ASM can export traces to Jaeger or Zipkin. You can view them on the Jaeger or Zipkin UI. The following uses Zipkin as an example.

### Prerequisites

The cluster and namespace where Zipkin is to be installed have been specified.

### Procedure

#### Step 1 Create a Deployment named **zipkin**.

Log in to the CCE console and click the cluster name to go to the cluster console. In the navigation pane on the left, choose **Workloads**. On the **Deployments** tab, click **Create from YAML**, and copy the following content to the YAML file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: zipkin
  namespace: monitoring
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/instance: zipkin
      app.kubernetes.io/name: zipkin
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      labels:
        app.kubernetes.io/instance: zipkin
        app.kubernetes.io/name: zipkin
    spec:
      automountServiceAccountToken: false
      containers:
      - env:
        - name: STORAGE_TYPE
          value: mem
        image: openzipkin/zipkin-slim:latest
        # Community Zipkin image path. Ensure
        that you can access this path.
```

```

imagePullPolicy: IfNotPresent
name: zipkin
readinessProbe:
  failureThreshold: 3
  httpGet:
    path: /health
    port: 9411
    scheme: HTTP
  initialDelaySeconds: 5
  periodSeconds: 5
  successThreshold: 1
  timeoutSeconds: 1
resources:
  limits:
    cpu: 500m
    memory: 4Gi
  requests:
    cpu: 100m
    memory: 128Mi
securityContext:
  readOnlyRootFilesystem: true
  runAsNonRoot: true
  runAsUser: 1000
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
terminationGracePeriodSeconds: 30

```

The Deployment named **zipkin** is displayed on the **Deployments** tab. If the status of **zipkin** changes to **Running**, Zipkin has been installed in the **monitoring** namespace of the target cluster.



**NOTE**

You can also refer to [Zipkin official website documentation](#) to complete the installation.

**Step 2** Create a Service of the LoadBalancer type.

On the cluster console, choose **Services & Ingresses** in the navigation pane on the left. On the **Services** tab, click **Create Service**. Then, configure the parameters as follows:

- **Service Name:** Enter a name. **zipkin** is used as an example here.
- **Service Type:** Select **LoadBalancer**.
- **Selector:** Click **Reference Workload Label**. The label is automatically added.
- **Ports:** Configure the container port and Service port. **9411** is used as an example here.

Retain the default values for other parameters.

**Create Service** [Create from YAML](#)

Service Name: zipkin

Service Type:
 

- ClusterIP: Expose services through the internal IP of the cluster, which can only be accessed within the cluster
- NodePort: Expose services via IP and static port (NodePort) on each node
- LoadBalancer: Provide external services through ELB load balancing, high availability, ultra-high performance, stability and security.**
- DNAT: Expose cluster node access type services through NAT gateway, support multiple nodes to share and use elastic IP

It is recommended to select the load balancing access type for out-of-cluster access

Service Affinity: **Cluster-level** Node-level

Namespace: monitoring

Selector:
 

- Key = Value
- app.kubernetes.io/instance = zipkin
- app.kubernetes.io/name = zipkin

 Services are associated with workloads (labels) through selectors.

Load Balancer:
 

- Dedicated Network (TCP/UDP)
- Use e...
- elb-86e5

 Supports only dedicated load balancers of Network type in VPC vpc-fz where the cluster resides.

Health Check:
 

- Disable
- Global health check**
- Custom health check

 protocol: TCP | delay(s): 5 | timeout(s): 10 | maxRetries: 3

Ports:
 

Protocol	Container Port	Service Port	Frontend Protocol
TCP	9411	9411	TCP

Buttons: Cancel, OK

The Service named **zipkin** is displayed on the **Services** tab.

Service	Namespace	Service Type	Access Address	Access Port/Container Port/Protocol	Created	Operation
zipkin	monitoring	LoadBalancer	elb-86e5	9411 - 9411 / TCP	5:59:15.260	Manage Pool View Events More

**CAUTION**

If you do not need to access the Zipkin UI, set **Access Type** to **ClusterIP**.

**Step 3** Buy a service mesh and interconnect it with Zipkin.

Log in to the ASM console and click **Buy Mesh**. In **Cluster Configuration**, select the cluster in **Step 1**. In **Observability Configuration**, enable tracing. Then, select **Third-party Jaeger/Zipkin service** for **Version**, set **Service Address** and **Access Port**, and configure other parameters as required.

Call chain  Enable Call Chain

After call chain tracing is enabled, you can view call chain information in the call chain tracing system.

Sampling Rate  %

Version APM 2.0 Third-party Jaeger/Zipkin service

If Jaeger or Zipkin is not deployed, install the Jaeger or Zipkin plug-in to obtain the service address and port. Ensure that the Jaeger or Zipkin service address is reachable from the service pod to be observed.

Service Address

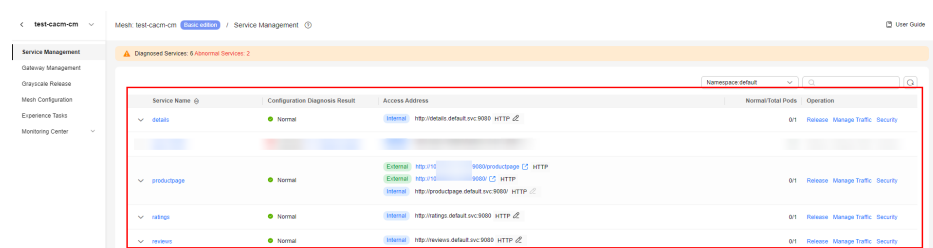
Access Port

**CAUTION**

The value of **Service Address** is *{Service name}.{Namespace}.svc.cluster.local*. Replace *{Service name}* and *{Namespace}* with those specified in **Step 2**.

The value of **Access Port** is that specified in **Step 2**. **9411** is used as an example here.

**Step 4** Deploy an experience service () by referring to [Grayscale Release Practices of Bookinfo](#). After the deployment is complete, the services shown in the following figure are displayed on the **Service Management** page.



**Step 5** Access the productpage details page to trigger tracing.

Go to the service mesh details page. In the navigation pane on the left, choose **Service Management**. On the displayed page, click the external access address **http://{IP address}:{Port number}/productpage** of the productpage service.

**Step 6** View the traces on the Zipkin UI at **http://{Public IP address of the load balancer configured for zipkin}:{Access port of zipkin}/zipkin/**.

 **NOTE**

You can obtain the IP address and port for logging in to the Zipkin client as follows:

- IP address: Go to the console of the cluster where Zipkin is installed. In the navigation pane on the left, choose **Services & Ingresses**. On the **Services** tab, view the public IP address of the load balancer configured for **zipkin**.
- Port: On the **Services** tab, view the access port of **zipkin**.

----**End**