Application Service Mesh

FAQs

Issue 02

Date 2025-05-28

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

1 Service Mesh Cluster

1.1 Why Does an Enabled Service Mesh Remain in the Installing State for a Long Time?

Symptom

After I enable a service mesh (buy a service mesh) for a CCE cluster, it remains in the installing state for a long time and a message is displayed indicating that the Istio-based service mesh is being enabled and the 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 service mesh again.

1.2 Why Does an Uninstalled Service Mesh Remain in the Unready State for a Long Time?

Symptom

On the ASM console, after I uninstall a service mesh, it remains in the unready state 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, 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 service mesh again.

----End

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

Symptom

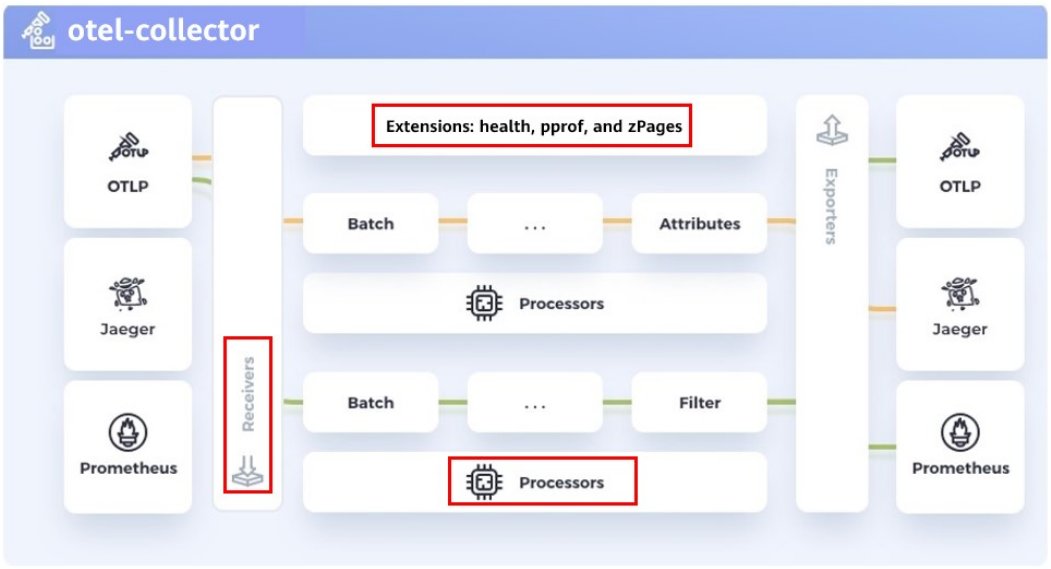
An otel-collector workload is automatically created when a service mesh is created.

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@ ~]# 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** specifies that telemetry data can be obtained from Envoy proxies using **zipkin** and **prometheus**. **prometheus** specifies 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.

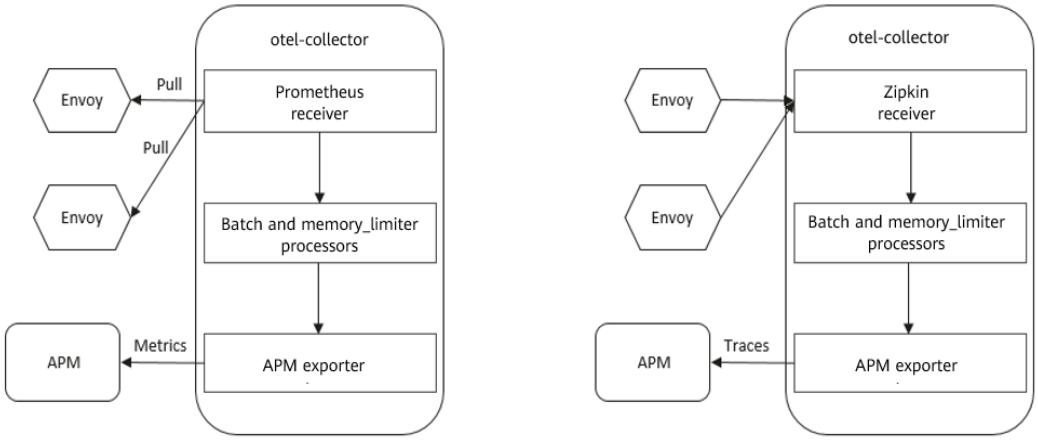
```
extensions:
  health_check:
    endpoint: 127.0.0.1:13133
```

- **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 Service Mesh for My Cluster?

Symptom

I cannot create a service mesh for my cluster.

Analysis

Currently, clusters earlier than v1.21 cannot be managed by service meshes.

Solution

Step 1 Check the cluster version. Currently, only clusters v1.21 or later can be managed by service meshes.

Step 2 Check your browser. Chrome is recommended. The button for service 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              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
```


CAUTION

Perform the following operations only in Istio 1.18.7-r4 or later.

After running the **kubectl edit iop** command to edit the parameter to be modified, change the value of **install.istio.io/ignoreReconcile** to **false**, save the modification, and exit.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  annotations:
    asm/patch: ""
    asm/post: ""
    asm/updateTimestamp: "2025-03-12T08:23:39Z"
    install.istio.io/ignoreReconcile: "false"
  creationTimestamp: "2025-03-12T08:22:35Z"
  finalizers:
  - istio-finalizer.install.istio.io
```

Run the **kubectl get iop -n istio-system** command to check the IOP status. Wait until the value of **STATUS** changes to **HEALTHY**.

```
[root@whtest-cluster-131-17185-dorgy ~]#
[root@whtest-cluster-131-17185-dorgy ~]# kubectl get iop -n istio-system
NAME                          REVISION    STATUS    AGE
installed-state-eastwest                            22h
private-data-plane-1-18-7-r4  1-18-7-r4   HEALTHY   23h
[root@whtest-cluster-131-17185-dorgy ~]#
```

Change the value of **install.istio.io/ignoreReconcile** to **true**.

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, 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

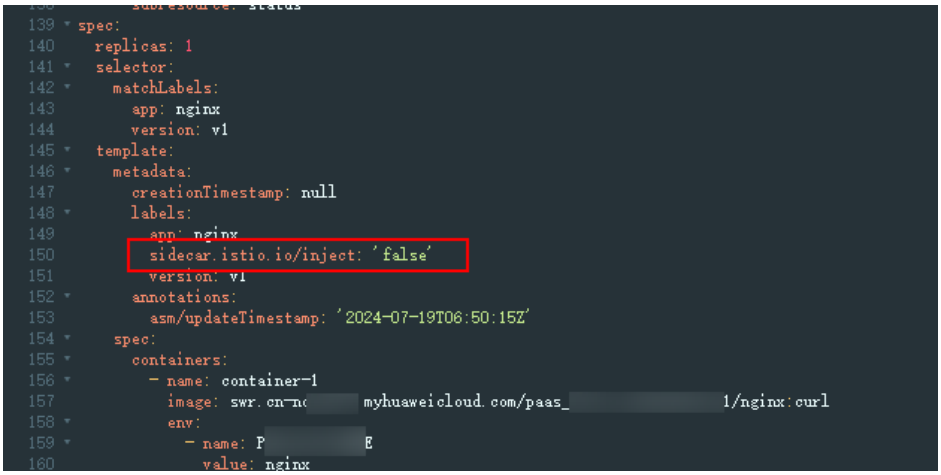
Locate the **spec.template.metadata.annotations** field and add **sidecar.istio.io/inject: 'false'**.

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

- 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'
  ```



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?

Symptom

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-crital
    multiCluster:
      clusterName: test-yyl-multi
    proxy:
      autoInject: enabled
      holdApplicationUntilProxyStarts: true
```


CAUTION

Perform the following operations only in Istio 1.18.7-r4 or later.

After running the **kubectl edit iop** command to edit the parameter to be modified, change the value of **install.istio.io/ignoreReconcile** to **false**, save the modification, and exit.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  annotations:
    asm/patch: ""
    asm/post: ""
    asm/updateTimestamp: "2025-03-12T08:23:39Z"
    install.istio.io/ignoreReconcile: "false"
  creationTimestamp: "2025-03-12T08:22:35Z"
  finalizers:
  - istio-finalizer.install.istio.io
```

Run the **kubectl get iop -n istio-system** command to check the IOP status. Wait until the value of **STATUS** changes to **HEALTHY**.

```
[root@whtest-cluster-131-17185-dorgy ~]#
[root@whtest-cluster-131-17185-dorgy ~]# kubectl get iop -n istio-system
NAME                          REVISION    STATUS    AGE
installed-state-eastwest                            22h
private-data-plane-1-18-7-r4  1-18-7-r4   HEALTHY   23h
[root@whtest-cluster-131-17185-dorgy ~]#
```

Change the value of **install.istio.io/ignoreReconcile** to **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                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-6b4959fffb-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-yyl-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:
        kubectl.kubernetes.io/restartedAt: "2022-11-25T10:35:02+08:00"
        proxy.istio.io/config: |
          holdApplicationUntilProxyStarts: true
      creationTimestamp: null
```

2.7 Why Does the Owner Group of the File Mounted to the Service Container Change After Configuring fsgroup?

Symptom

When a sidecar is injected into the service pod and **fsgroup** is set to **1337**, the owner group of the file mounted to the service container is changed to **1337**.

Analysis

A Kubernetes version bug:

https://github.com/kubernetes/kubernetes/issues/57923

https://github.com/istio/istio/pull/27367

In versions earlier than 1.8.6-r2, **fsgroup** is automatically set to **1337** during sidecar injection. This setting will change the owner group of the file mounted to the service container to **1337**.

Solution

This problem is resolved in Kubernetes v1.19 and later versions. For meshes of v1.8.6-r2 and later, if the cluster version is 1.19 or later, ASM automatically sets **EnableLegacyFSGroupInjection** to **false**. This configuration prevents **fsgroup** from being set to **1337** during sidecar injection, and then the owner group of the file mounted to the service container will not be changed to **1337**. If adaptation is performed in the early stage of the business, the adaptation needs to be corrected.

2.8 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.

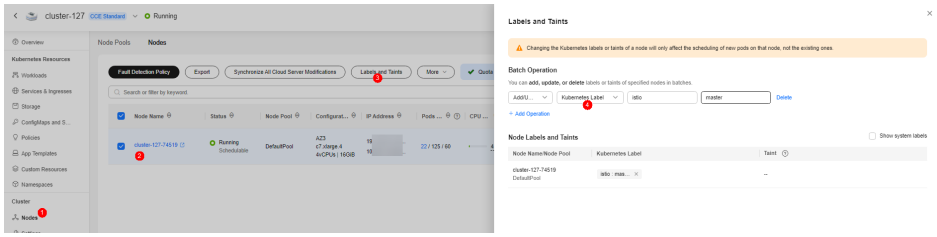
   Solution: Use the cluster version listed in the following table.

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, 1.27, and 1.28
1.18	1.25, 1.27, and 1.28, as well as 1.29, 1.30, and 1.31

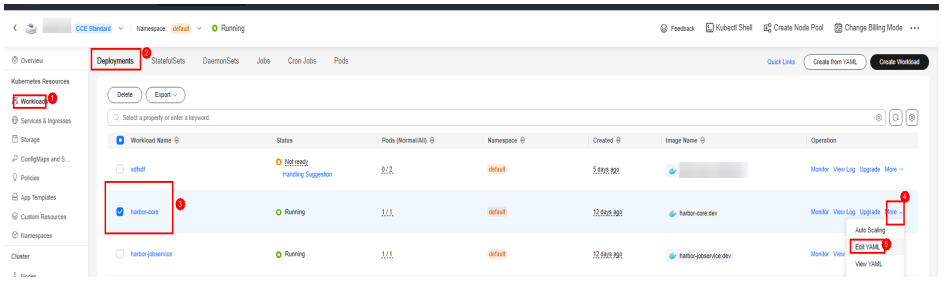
7. Failed to check the component affinity before the upgrade.

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.

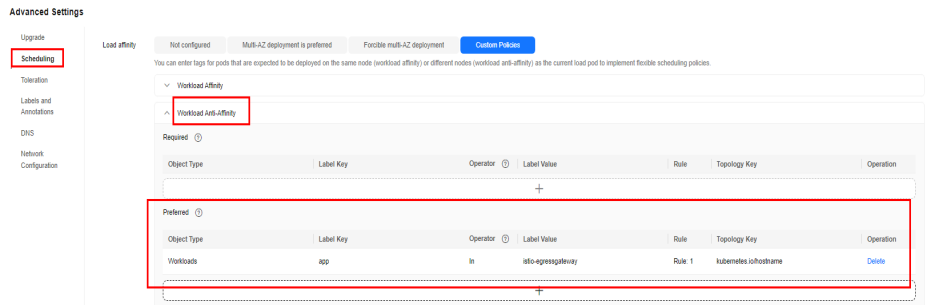


– **Solution 2**: Edit the YAML file to modify the anti-affinity policy on the CCE console.



```
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.

2.9 Possible Causes of Sidecar Injection Failures

Common scenarios and solutions:

- **Possible cause**: The number of pods managed by a service mesh has reached the limit, so no more sidecars can be injected.

  Check method: Check whether the number of pods injected into the service mesh reaches the limit.

  Log in to the **ASM console** and check whether the number of pods displayed in your service mesh card reaches the service mesh scale. If yes, the number of pods injected to the service mesh has reached the limit.

  Solution: Contact O&M engineers or **submit a service ticket**.

- **Possible cause**: The control plane component istiod is abnormal.

  Check method: Check whether the istiod component in the **istio-system** namespace is abnormal.

  Log in to the **CCE console** and click the cluster name to go to the cluster console. In the navigation pane, choose **Workloads**. Then, select the **istio-system** namespace, and check whether the status of **istiod-1-18-7-r4** is **Running** in the **Status** column. If the status is not **Running**, the component may be abnormal.

  NOTE

  1-18-7-r4 indicates the service mesh version. The service mesh version is specified by **Version** on the **Basic Information** tab (**Mesh Configuration** > **Basic Information**). The displayed version number here is combined with hyphens (-).

  Solution: If the component is abnormal, contact O&M engineers or **submit a service ticket**.

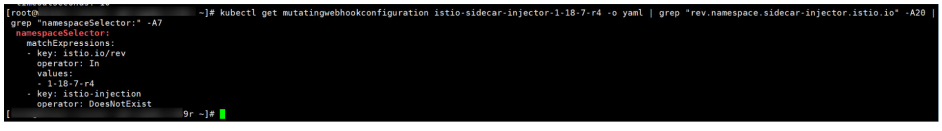
- **Possible cause**: The automatic injection label is not added to the namespace.

  Check method: Run the following command to view the webhook's namespaceSelector:

  ```
  kubectl get mutatingwebhookconfiguration istio-sidecar-injector-1-18-7-r4 -o yaml | grep "rev.namespace.sidecar-injector.istio.io" -A20 | grep "namespaceSelector:" -A7
  ```

  NOTE

  Replace **1-18-7-r4** in the preceding command with the service mesh version in the current format (combined with hyphens). The service mesh version is specified by **Version** on the **Basic Information** tab (**Mesh Configuration** > **Basic Information**).

  For example, the command output in the preceding figure indicates that the **istio.io/rev=1-18-7-r4** label is required and the **istio-injection** label is not allowed.

  Run the following command to check whether the target namespace is included in the webhook:


```
kubectl get {namespace} --show-labels
```

Solution: Add the obtained injection label to the namespace.

- **Possible cause**: The default injection policy is not set to **enabled** for the namespace.

  Check method: Run the following command to check the default policy, and ensure that the policy value is **enabled**.

  ```
  kubectl -n istio-system get configmap istio-sidecar-injector-1-18-7-r4 -o jsonpath='{.data.config}' | grep policy:
  ```

  NOTE

  Replace **1-18-7-r4** in the preceding command with the service mesh version in the current format (combined with hyphens). The service mesh version is specified by **Version** on the **Basic Information** tab (**Mesh Configuration** > **Basic Information**).

  Solution: Refer to **How Do I Enable Namespace Injection for a Cluster?**

- **Possible cause**: There is the **sidecar.istio.io/inject: 'false'** label.

  Check method: Check the workload label.

  Log in to the **CCE console** and click the cluster name to go to the cluster console. In the navigation pane, choose **Workloads**. Then, select the corresponding namespace, locate your workload, and choose **More** > **View YAML** in the **Operation** column. Find the **spec.template.metadata.labels** field and ensure that the **sidecar.istio.io/inject: 'false'** label does not exist.

  NOTE

  Replace **1-18-7-r4** in the preceding command with the service mesh version in the current format (combined with hyphens). The service mesh version is specified by **Version** on the **Basic Information** tab (**Mesh Configuration** > **Basic Information**).

  Solution: If there is the **sidecar.istio.io/inject: 'false'** label, delete it. For details, see **How Do I Disable Sidecar Injection for Some Workloads?**

- **Possible cause**: The pod cannot be created.

  Solution: Run the following command to check the error information and rectify the fault based on the error information:

  ```
  kubectl describe -n {namespace} {deployment name}
  ```

- **Possible cause**: Ensure that your pod is not in the **kube-system** or **kube-public** namespace.

  NOTE

  Automatic sidecar injection will be ignored for pods in these namespaces.

- **Possible cause**: Ensure that the pod specification does not contain **hostNetwork: true**.

  NOTE

  Automatic sidecar injection will be ignored for pods with **hostNetwork: true**.

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 Deployed Demo Application?

Symptom

The page of the deployed demo application cannot be accessed.

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 Does Error Code 500 Is Displayed When I Create a Gateway?

Symptom

When I deploy Bookinfo in one click, a message is displayed, indicating that the gateway fails to be created.

Fault Location

Log in to the ASM console, press **F12**, and switch to the **Network** tab to view APIs. Error code 500 is returned when a POST request is sent to create a Gateway. The returned information is as follows:

```
IP is not the same with LoadBalancerIP
```

Analysis

Residual **gatewayservice** exists in the **istio-system** namespace. This is because the added gateways are not deleted before you delete the release.

Solution

Run the following command to delete the **gatewayservice** service remaining in the **istio-system** namespace.

kubectl delete svc *<svc-name>* **-n namespace**

<svc-name> indicates the service name.

3.5 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.

3.6 What Should I Do If the Application Data Fails to Be Obtained?

Symptom

After a service is added to a mesh, the service is not displayed on the **Service List** page. A message is displayed indicating that the application data fails to be obtained.

Fault Location

Log in to the ASM console, press **F12**, and switch to the **Network** tab to view APIs. Error code 200 is returned for all APIs. The following error information is displayed in the Console output:

```
TypeError: Cannot read property 'slice' of undefined
```

Analysis

The service port number is left blank.

Solution

Step 1 Check the service port.

kubectl get svc --all-namespaces

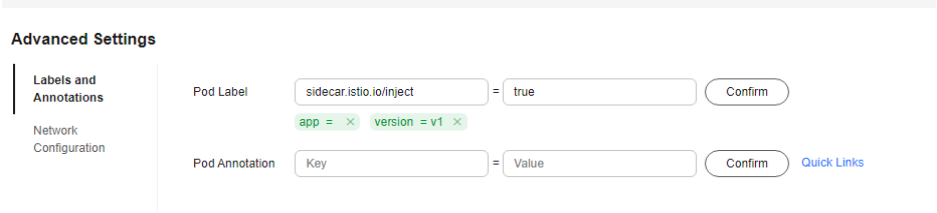
Step 2 Configure a port for the service that does not have a port.

----End

3.7 How Do I Inject a Sidecar for the Pod Created Using a Job or Cron Job?

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 cron job. If sidecar injection is required, choose **Labels and Annotations** in **Advanced Settings** and set **sidecar.istio.io/inject** to **true** for **Pod Label**.



The following is an example cron job:

```
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 cron job, you need to run a specific command in the target container to disable the sidecar.

Disabling a Sidecar After a Job or Cron Job Is Complete

Call **curl -sf -XPOST http://127.0.0.1:15000/quitquitquit** to disable istio-proxy after a job or cron job is complete.

The following is an example cron job:

```
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 a Grayscale Release?

Symptom

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

Analysis

When performing the grayscale release on a service, you can only change the tags of the image used by the service.

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 a 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.

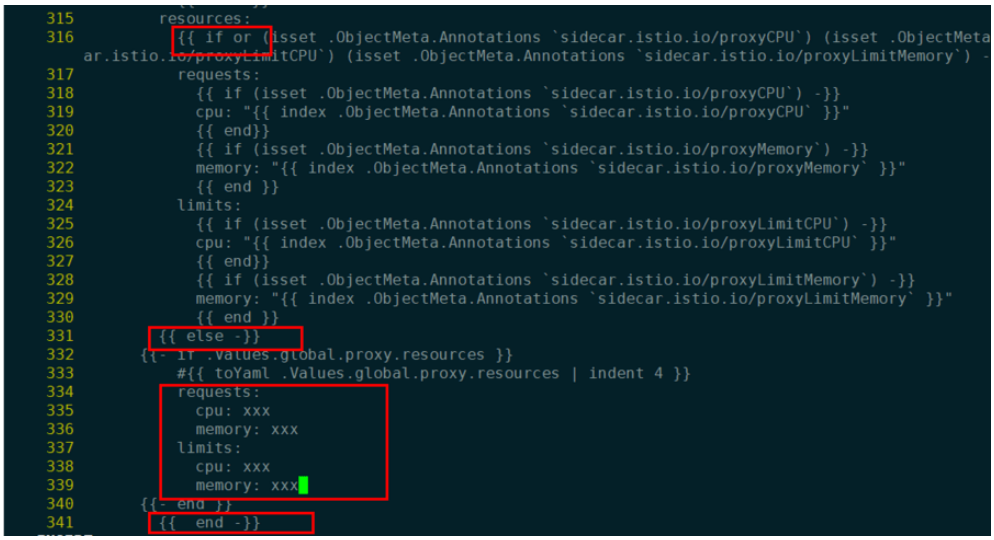
```
resources:
  limits:
    cpu: "2"
    memory: 512Mi
  requests:
    cpu: "1"
    memory: 512Mi
```

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



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 **spec.template.spec.containers.env.PILOT_HTTP10** to **1** in the Istiod Deployment.

```
423         env:
424           - name: REVISION
425             value: 1-18-7-r2
426           - name: JWT_POLICY
427             value: third-party-jwt
428           - name: PILOT_CERT_PROVIDER
429             value: istiod
430           - name: POD_NAME
431             valueFrom:
432               fieldRef:
433                 apiVersion: v1
434                 fieldPath: metadata.name
435           - name: POD_NAMESPACE
436             valueFrom:
437               fieldRef:
438                 apiVersion: v1
439                 fieldPath: metadata.namespace
440           - name: SERVICE_ACCOUNT
441             valueFrom:
442               fieldRef:
443                 apiVersion: v1
444                 fieldPath: spec.serviceAccountName
445           - name: INSTANCE_IP
446             valueFrom:
447               fieldRef:
448                 apiVersion: v1
449                 fieldPath: status.podIP
450           - name: ENABLE_DEBUG_ON_HTTP
451             value: 'false'
452           - name: PAAS_CRYPTO_PATH
453             value: /opt/cloud/asm/secret/kubernetes
454           - name: KUBECONFIG
455             value: /var/run/secrets/remote/config
456           - name: PILOT_HTTP10
457             value: '1'
458           - name: PILOT_TRACE_SAMPLING
459             value: '1'
460           - name: PILOT_ENABLE_PROTOCOL_SNIFFING_FOR_OUTBOUND
461             value: 'true'
462           - name: PILOT_ENABLE_PROTOCOL_SNIFFING_FOR_INBOUND
463             value: 'true'
464           - name: ISTIOD_ADDR
465             value: istiod-1-18-7-r2.istio-system.svc:15012
466           - name: PILOT_ENABLE_ANALYSIS
467             value: 'false'
468           - name: CLUSTER_ID
469             value: wanghai-cluster-129
```

CAUTION

Perform the following operations only in Istio 1.18.7-r4 or later.

After running the **kubectl edit iop** command to edit the parameter to be modified, change the value of **install.istio.io/ignoreReconcile** to **false**, save the modification, and exit.

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  annotations:
    asm/patch: ""
    asm/post: ""
    asm/updateTimestamp: "2025-03-12T08:23:39Z"
    install.istio.io/ignoreReconcile: "false"
  creationTimestamp: "2025-03-12T08:22:35Z"
  finalizers:
  - istio-finalizer.install.istio.io
```

Run the **kubectl get iop -n istio-system** command to check the IOP status. Wait until the value of **STATUS** changes to **HEALTHY**.

```
[root@whtest-cluster-131-17185-dorgy ~]#
[root@whtest-cluster-131-17185-dorgy ~]# kubectl get iop -n istio-system
NAME                          REVISION    STATUS    AGE
installed-state-eastwest                            22h
private-data-plane-1-18-7-r4  1-18-7-r4   HEALTHY   23h
[root@whtest-cluster-131-17185-dorgy ~]#
```

Change the value of **install.istio.io/ignoreReconcile** to **true**.

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

Scenarios

In Istio, to implement transparent traffic management of a service mesh, sidecars are designed to intercept all incoming and outgoing traffic by default. This design prevents service intrusion and ensures security and reliability. However, this design may cause the following problems in actual scenarios:

- When all traffic passes through a sidecar, the sidecar's memory and CPU usages are high. In severe cases, the service pod may be restarted or there is even a service cascading failure.
- In some scenarios, direct access to external services (such as database connection pools) is required, where the default interception mechanism cannot be used.

This section describes how to configure refined traffic interception rules to resolve the problems.

Workload Configuration for Blocking or Allowing Traffic from Some IP Address Ranges

Modify the **deployment** file to block the 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
```

CAUTION

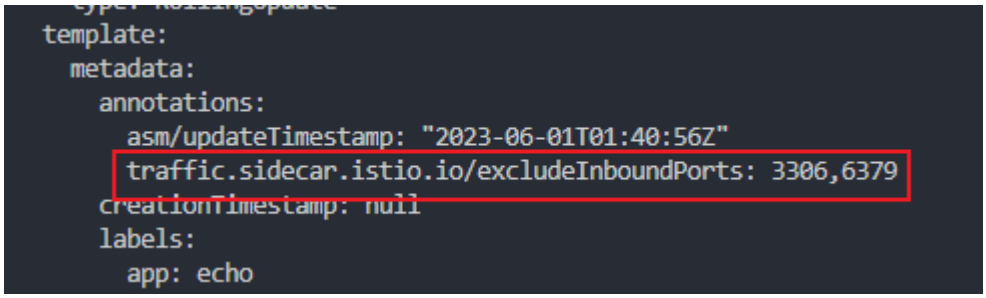
The preceding operations will result in rolling upgrades of service containers.

Workload Configuration for Blocking or Allowing Traffic over Some Ports

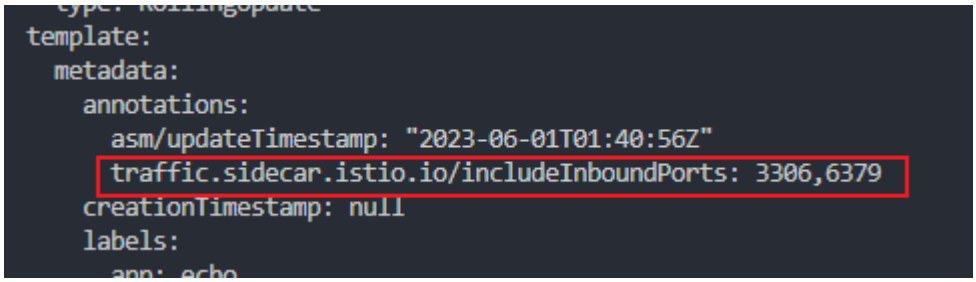
Modify the **deployment** file to block or allow ingress and egress traffic over some ports.

Run the **kubectl edit deploy –n *user_namespace user_deployment*** command.

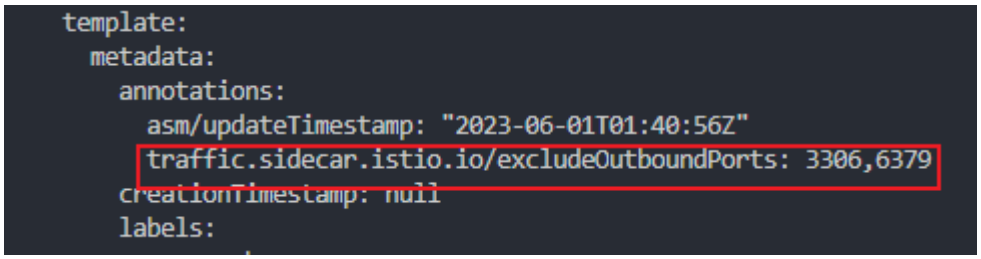
1. In **deployment.spec.template.metadata.annotations**, use **traffic.sidecar.istio.io/excludeInboundPorts** to specify the ports that allow the ingress traffic.



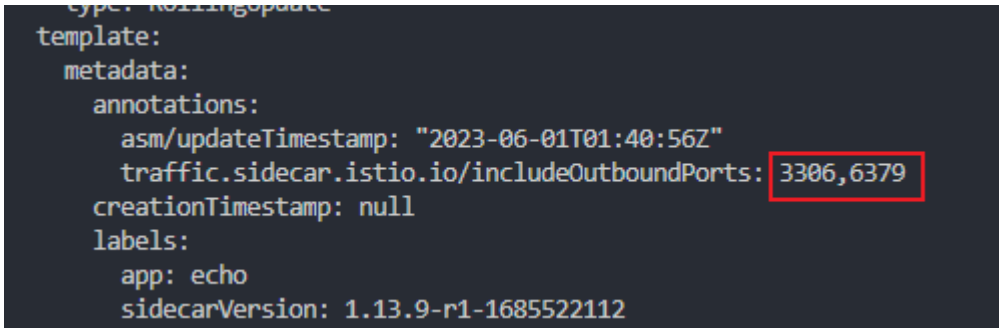
2. In **deployment.spec.template.metadata.annotations**, use **traffic.sidecar.istio.io/includeInboundPorts** to specify the ports that block the ingress traffic.



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



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



CAUTION

The preceding operations will result in 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?

Symptom

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, 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

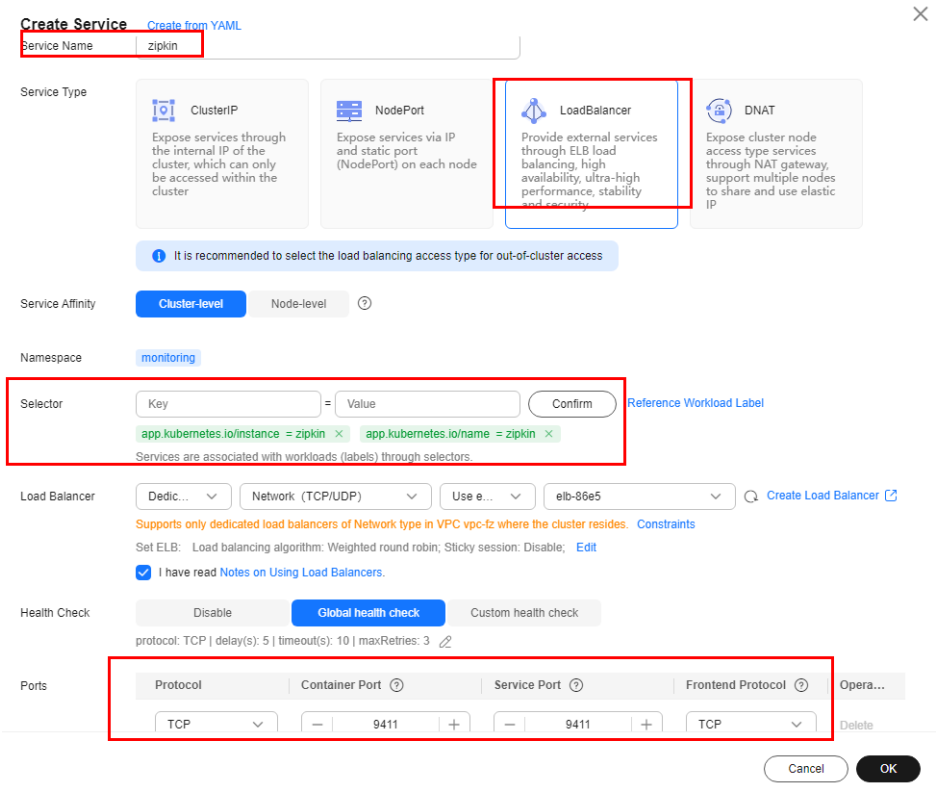
You can also refer to **Zipkin official website documentation** to complete the installation.

Step 2 Create a Service of the LoadBalancer type.

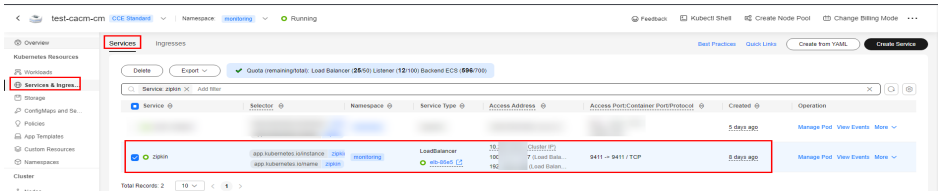
On the cluster console, choose **Services & Ingresses** in the navigation pane. On the **Services** tab, click **Create Service**. Then, configure the parameters as follows:

- **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.



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



CAUTION

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

Step 3 Buy a service mesh and connect it to Zipkin.

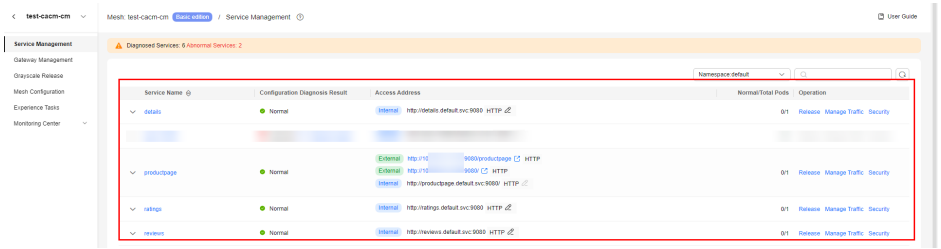
Log in to the ASM console and click **Buy Mesh**. In **Cluster Configuration**, select the cluster in **Step 1**. In **Observability Configuration**, enable tracing. Then, select **Third-party Jaeger/Zipkin service** for **Version**, set **Service Address** and **Access Port**, and configure other parameters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

CAUTION

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

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 **details**, **productpage**, **ratings**, and **reviews** services 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, choose **Service Management**. On the displayed page, click the external 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, 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

