Application Service Mesh

Best Practices

Issue 01

Date 2022-07-01

HUAWEI TECHNOLOGIES CO., LTD.

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

Copyright © Huawei Technologies Co., Ltd.

1 Upgrading Data Plane Sidecars Without Service Interruption

ASM enables you to manage and monitor the traffic of services added into a service mesh. Sidecars are important components in ASM data plane. The upgrade of sidecars involves the re-injection of sidecars into data plane service pods, which requires the pods to be updated.

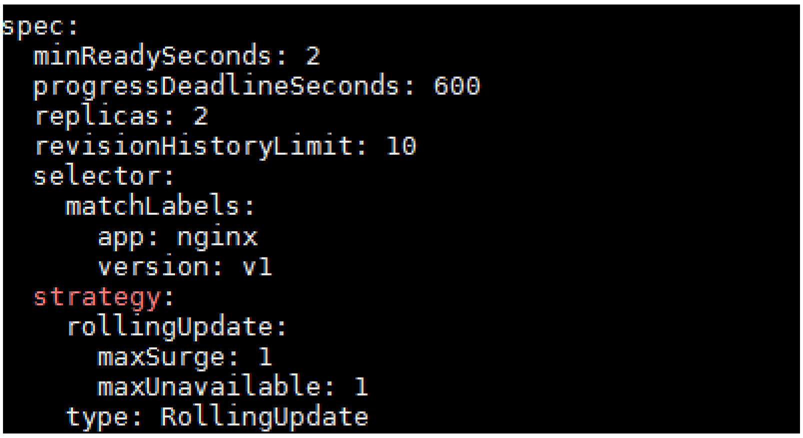
This section describes how to avoid service interruption during sidecar upgrade.

Configuring the Number of Service Pods

Ensure that the number of service pods is greater than or equal to **2** and the upgrade policy is set to **RollingUpdate**.

Sample configurations:

kubectl get deploy nginx -n *namespace_name* **-oyaml | grep strategy -a10**



```
spec:
  minReadySeconds: 2
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
      version: v1
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
```

Configuration description:

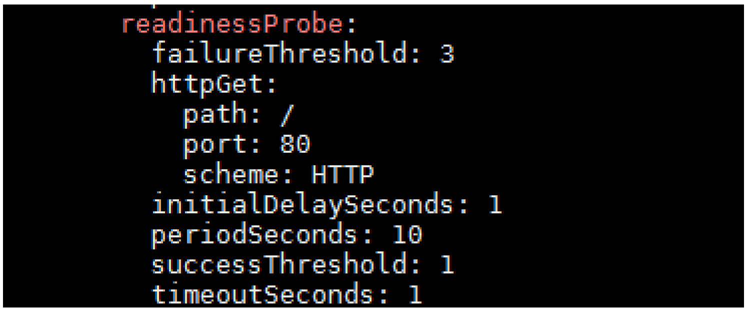
- Number of service pods: deployment.spec.replicas >= 2
- Upgrade policy: deployment.spec.strategy.type == RollingUpdate
- Minimum number of alive pods in rolling upgrade: deployment.spec.replicas - deployment.spec.strategy.maxUnavailable > 0

Adding a Readiness Probe

Readiness probes help you ensure that new service pods take over traffic only when they are ready. This prevents access failures caused by unready new pods.

The configurations are as follows:

kubectl get deploy nginx -n *namespace_name* **-oyaml | grep readinessProbe -a10**



```
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /
            port: 80
            scheme: HTTP
          initialDelaySeconds: 1
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
```

Configuration description:

Configuring a readiness probe: deployment.spec.template.spec.containers[i].readinessProbe

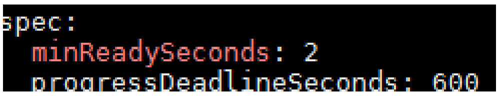
The configuration includes the initial check time, check interval, and timeout duration.

Setting the Service Ready Time

minReadySeconds is used to specify the minimum number of seconds for which a newly created pod should be ready without any of its containers crashing, for it to be considered available.

The configurations are as follows:

kubectl get deploy nginx -n *namespace_name* **-oyaml | grep minReadySeconds -a1**



```
spec:
  minReadySeconds: 2
  progressDeadlineSeconds: 600
```

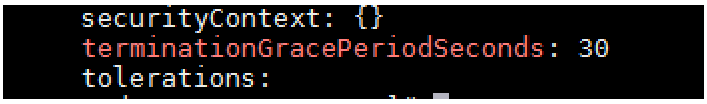
Configuration description:

The service ready time: deployment.spec.minReadySeconds. Configure this parameter based on the live environment.

Configuring a Graceful Shutdown Time

terminationGracePeriodSeconds is used to configure a graceful shutdown time. During a rolling upgrade, the endpoint of an old service pod is removed and the pod status is set to **Terminating**. A SIGTERM signal is then sent to the pod. After the graceful shutdown time you configured, the pod will be forcibly terminated. The graceful deletion time allows the pod to keep processing unfinished requests, if any, to avoid hard termination.

kubectl get deploy nginx -n *namespace_name* **-oyaml | grep terminationGracePeriodSeconds -a1**



```
      securityContext: {}
      terminationGracePeriodSeconds: 30
      tolerations:
```

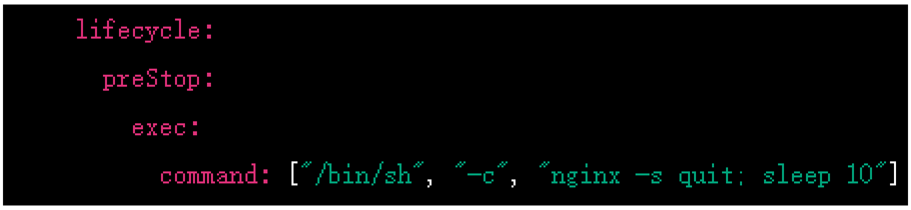
Configuration description:

The graceful shutdown time: deployment.spec.template.spec.terminationGracePeriodSeconds. The default value is 30s. Configure this parameter based on the live environment.

Configuring the preStop

preStop enabled you to perform certain execution before a service pod is stopped. In this way, it helps you gracefully shut down a service pod. Configure this parameter based on service requirements. Nginx is used as an example here.

kubectl get deploy nginx -n *namespace_name* **-oyaml | grep lifec -a10**



```
     lifecycle:
       preStop:
         exec:
           command: ["/bin/sh", "-c", "nginx -s quit; sleep 10"]
```

In the **lifecycle.preStop** field, the **nginx -s quit; sleep 10** command is defined. This command first sends a graceful shutdown signal to the Nginx process and then the pod termination pauses for 10 seconds. In this way, Nginx has enough time to complete the ongoing requests and gracefully close the pod before it terminates.

10 in **sleep 10** is an example value. You can change it based on the actual requirements and application performance. The key should be a proper value so that Nginx has enough time to gracefully shut down the process.

Alternatively, you can run custom commands or scripts to gracefully shut down your service process.

2 Service Governance for Dubbo-based Applications

2.1 Introduction

Dubbo is a special protocol which needs the following supports:

- Envoy on the service mesh data plane supports the parsing and traffic management of the Dubbo protocol.
- The mesh control plane supports the configuration of Dubbo governance rules to manage services such as grayscale release, load balancing, and access authorization.

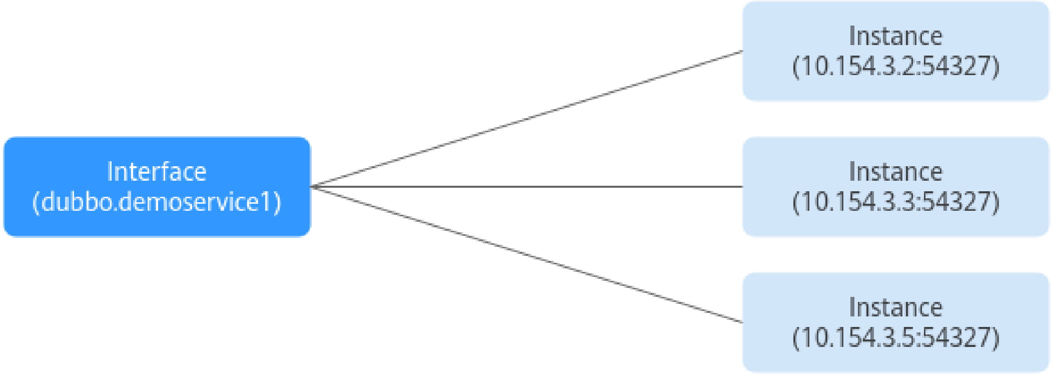
In addition, the service discovery model of Dubbo is different from that of Kubernetes and Spring Cloud. Therefore, additional processing is required.

2.2 Service Discovery Model

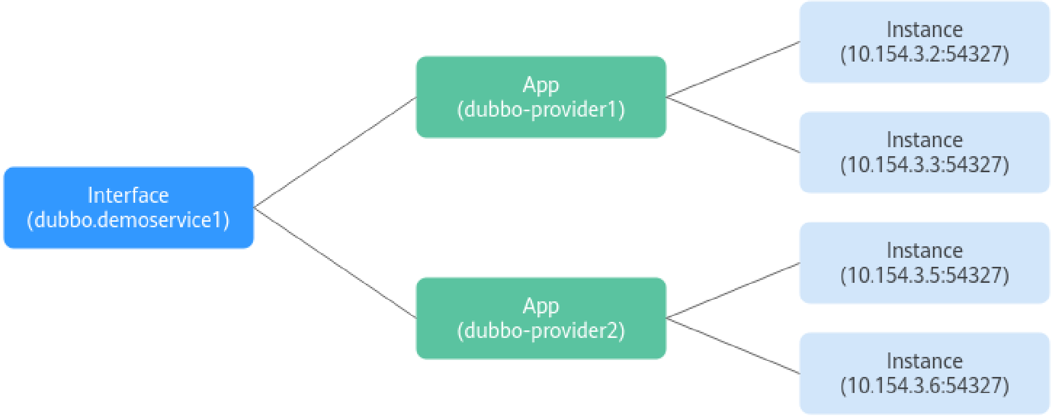
Problems in the existing Dubbo model (summarized from the Dubbo community version 2.7.4):

- In the microservice architecture, the Registry manages applications (services) instead of external service interfaces. However, the Dubbo Registry manages Dubbo service interfaces, contrary to the Spring Cloud or Cloud Native registration mode.
- A Dubbo application (service) allows N Dubbo service interfaces to be registered. The more interfaces, the heavier the load of the Registry.

The existing Dubbo service model searches for service instances based on the Dubbo interface.



The Dubbo Cloud Native service discovery model adds an app layer to search for instances.

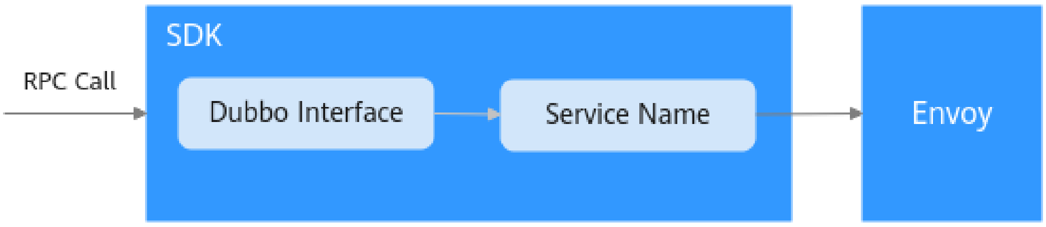


2.3 SDK Adaptation Mode

2.3.1 PASSTHROUGH Solution

Introduction

When the client in the SDK calls the target service by an interface, the client accesses the service name, instead of the service instance.



Description

Cases are different based on the Dubbo protocol versions:

- 2.7.4 and later versions: Cloud Native 2.7.4 and later versions have reconstructed the service discovery model which is consistent with that of Kubernetes. Service information can be directly obtained via interfaces.

- 2.7.3 and earlier versions: The Dubbo community versions do not provide the level-2 relationship between interfaces and services. The SDK needs to maintain the mapping from interfaces to services based on the actual usage mode. For example, information such as a service name may be provided in extended information during service registration.

You can select a processing mode based on your SDK. The SDK of an earlier version can perform the following operations in the existing service registration and discovery processes:

1. Extend the definition of **Service** in the registration information. During service deployment, service metadata can be injected into the SDK as environment variables, including **appname** and **namespace**, which indicate the name and namespace of the deployed service, respectively.
2. When the service is started, the relationship between the Dubbo interface and Kubernetes service name and namespace is registered in the Registry.
3. When a client initiates an access request, the service metadata is queried by the interface according to the original service discovery process, and the corresponding service information is used to assemble an RPC request. The extended field **Attachment** is advised to be used to store the **appname** and **namespace** information in the Dubbo request.

2.3.2 Static Target Service

Introduction

Use dubbo:reference to configure the referenced service provider in the service consumer of the Dubbo service. Use the **url** option to define the address of the point-to-point direct connection service provider to bypass the Registry and directly call the target service.

Description

If the original Dubbo service uses the **.xml** configuration file, only the configuration file needs to be modified.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans>
    <!-- Interfaces that can be called -->
    <dubbo:reference id="helloService " interface="com.dubbo.service.HelloService " url = "dubbo://
helloService:20880" />
</beans>
```

If an annotation is used to define the referenced target service, only the annotation of the target service in the code needs to be modified.

```
@Reference(url = "dubbo://helloService:20880")
HelloService helloService;
```

3 Reserving Source IP Address for Gateway Access

⚠ CAUTION

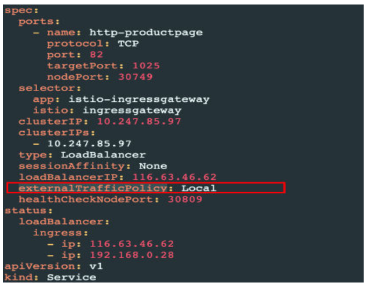
If node affinity is configured, the ELB address cannot be used to access workloads in some scenarios. For details, see **Why the ELB Address Cannot Be Used to Access Workloads in a Cluster?**

Scenario

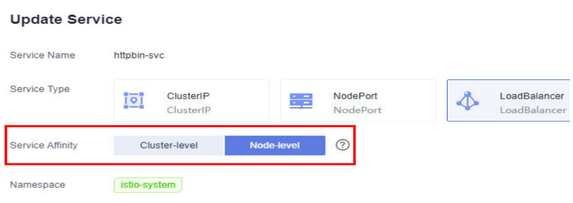
When a service is accessed through a gateway, the source IP address displayed in the target container is not the source IP address of the client by default. To retain the source IP address, perform the operations described in this section.

Configuration

Log in to the CCE console. On the **Networking** page, select the **istio-system** namespace, update the gateway service associated with the service, and change **Service Affinity** to **Node-level**. The prerequisite is that the function of obtaining the client IP address of the ELB has been enabled (this function is enabled by default).



```
spec:
  ports:
    - name: http-productpage
      protocol: TCP
      port: 82
      targetPort: 1025
      nodePort: 30749
  selector:
    app: istio-ingressgateway
    istio: ingressgateway
  clusterIP: 10.247.85.97
  clusterIPs:
    - 10.247.85.97
  type: LoadBalancer
  sessionAffinity: None
  loadBalancerIP: 116.63.46.62
  externalTrafficPolicy: Local
  healthCheckNodePort: 30809
status:
  loadBalancer:
    ingress:
      - ip: 116.63.46.62
      - ip: 192.168.0.28
apiVersion: v1
kind: Service
```



Update Service

Service Name httpbin-svc

Service Type: ClusterIP | NodePort | [✓ LoadBalancer]

Service Affinity: Cluster-level [✓ Node-level]

Namespace istio-system

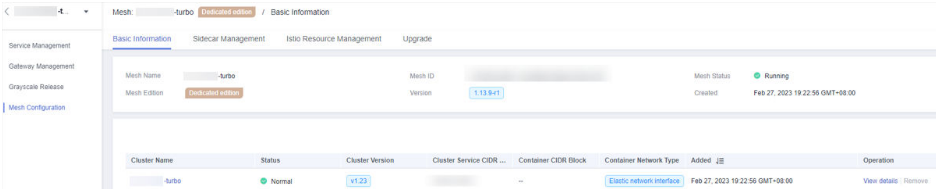
externalTrafficPolicy: indicates whether the Service wants to route external traffic to the local nodes or cluster-wide endpoints. Two options are available: Cluster (default) and Local. **Cluster** hides the client IP address, which may cause the second hop to another node, but has a good overall load distribution. **Local** retains the source IP address of the client and avoids the second hop of

LoadBalancer and **NodePort** services. However, there is a potential risk of unbalanced traffic transmission.

Authentication Method

The **x-forward-for** field of the **httpbin** image displays the source IP address. The **httpbin** service is an HTTP Request & Response Service that can send requests to the **httpbin** image. The **httpbin** image will return the requests based on the specified rules. You can search for the **httpbin** image in SWR. Before using the **httpbin** image for verification, ensure that the mesh function has been enabled for the cluster.

1. Log in to the ASM console and click the name of an available test mesh to go to its details page.
2. In the navigation pane, choose **Mesh Configuration**. Then, view the associated cluster.



Mesh Name: -turbo
Mesh Edition: Dedicated edition
Mesh ID:
Version: 1.13.9-r1
Mesh Status: Running
Created: Feb 27, 2023 19:22:56 GMT+08:00

Cluster Name	Status	Cluster Version	Cluster Service CIDR ...	Container CIDR Block	Container Network Type	Added	Operation
-turbo	Normal	v1.23		--	Elastic network interface	Feb 27, 2023 19:22:56 GMT+08:00	View details Remove

3. Click the cluster name to go to the cluster console. In the navigation pane, choose **Workloads**. On the **Deployments** tab, click **Create Workload** in the upper right corner.
4. Configure workload information.

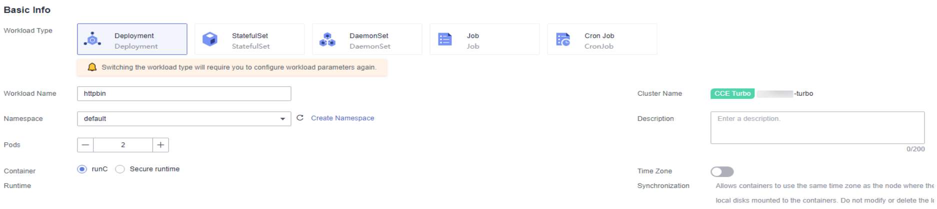
   Basic Information

   – **Workload Type**: Select **Deployment**.

   – **Workload Name**: Enter **httpbin**.

   – **Namespace**: Select the namespace of the workload. The default value is **default**.

   – Retain the default values of other parameters.



Workload Type: Deployment
Switching the workload type will require you to configure workload parameters again.
Workload Name: httpbin
Namespace: default
Pods: 2
Container Runtime: ● runC ○ Secure runtime
Cluster Name: CCE Turbo -turbo
Description: Enter a description. 0/200
Time Zone Synchronization: Allows containers to use the same time zone as the node where the local disks mounted to the containers. Do not modify or delete the l

   Container Configuration

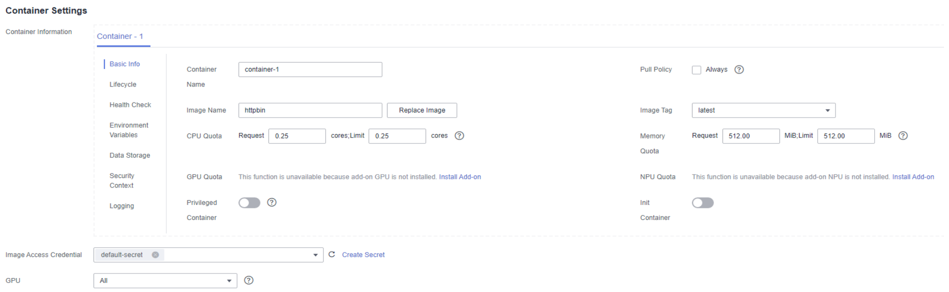
   – **Basic Information**

   ■ **Container Name**: Customize a name.

   ■ **Image Name**: Click **Select Image**, search for the **httpbin** image, select the image, and click **OK**.

   ■ **Image Tag**: Select an image tag.

   ■ Retain the default values of other parameters.



Container Settings

Container Information: Container - 1

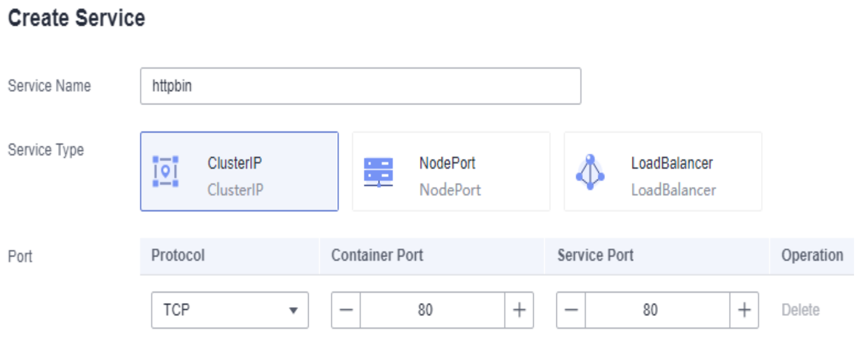
Container Name: container-1
Pull Policy: □ Always
Image Name: httpbin
Image Tag: latest
CPU Quota: Request 0.25 cores;Limit 0.25 cores
Memory Quota: Request 512.00 MiB;Limit 512.00 MiB
GPU Quota: This function is unavailable because add-on GPU is not installed. Install Add-on
NPU Quota: This function is unavailable because add-on NPU is not installed. Install Add-on
Privileged Container: ○
Init Container: ○
Image Access Credential: default-secret
GPU: All

Service Settings

A Service solves the pod access problems. With a fixed IP address, a Service forwards access traffic to pods and performs load balancing for these pods.

Click **+** under the service configuration parameter to go to the **Create Service** page.

- **Service Name**: Enter the workload name.
- **Service Type**: Select **ClusterIP**.
- Port parameters:
  - **Protocol**: Select **TCP**.
  - **Container Port**: 80 (Use the actual port number.)
  - **Service Port**: 80 (Use the actual port number.)



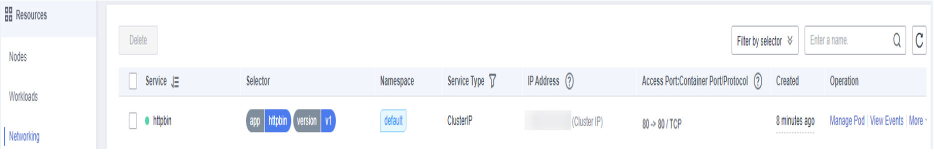
Create Service

Service Name: httpbin
Service Type: ClusterIP

Port:

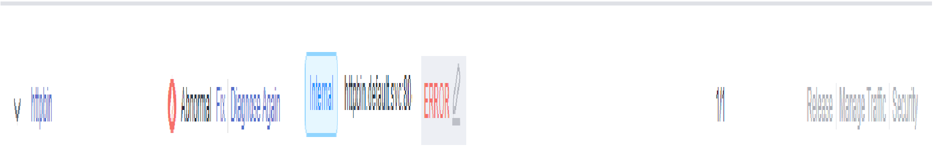
Protocol	Container Port	Service Port	Operation
TCP	80	80	Delete

5. Click **OK** in the lower right corner.
6. Click **Create Workload** in the lower right corner.
7. On the cluster console, click **Networking** in the navigation pane. The created **httpbin** service is displayed in the service list.



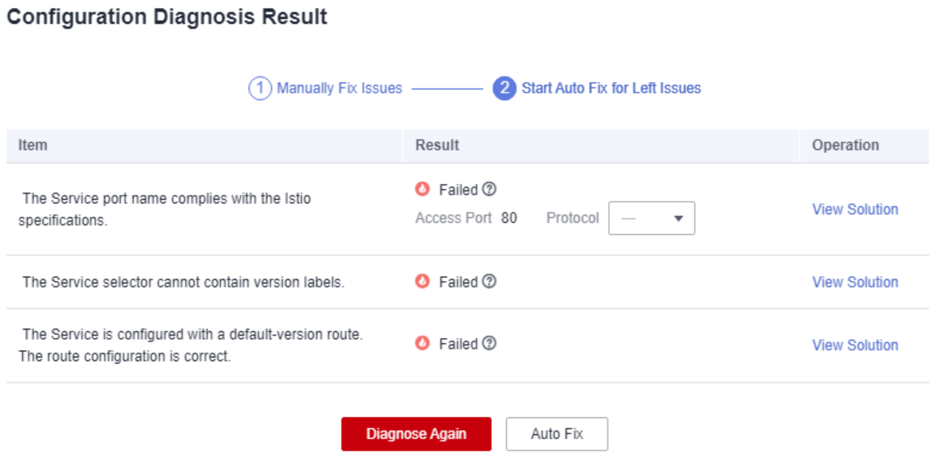
Service	Selector	Namespace	Service Type	IP Address	Access Port:Container Port/Protocol	Created	Operation
httpbin	app httpbin version v1	default	ClusterIP	[redacted] (Cluster IP)	80 -> 80 / TCP	8 minutes ago	Manage Pod | View Events | More

8. Return to the ASM console and click **Service Management**. The **Configuration Diagnosis Result** of the **httpbin** service is **Abnormal**.



httpbin | Abnormal | Internal | httpbin.default.svc:80 | ERROR | 1/1

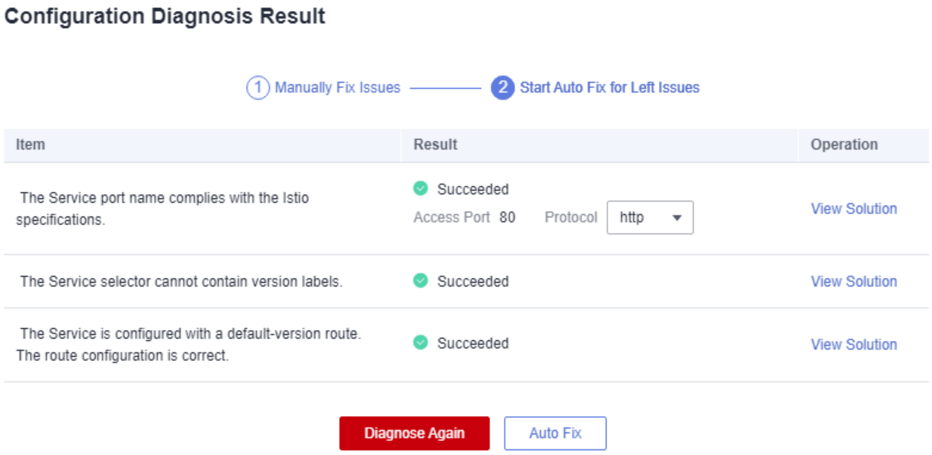
9. Click **Fix**. In **Configuration Diagnosis Result**, rectify the fault by following the solution.



Configuration Diagnosis Result

Item	Result	Operation
The Service port name complies with the Istio specifications.	Failed Access Port 80 Protocol --	View Solution
The Service selector cannot contain version labels.	Failed	View Solution
The Service is configured with a default-version route. The route configuration is correct.	Failed	View Solution

Take "The Service port name complies with the Istio specifications" as an example. Select **http** and click **Auto Fix**.



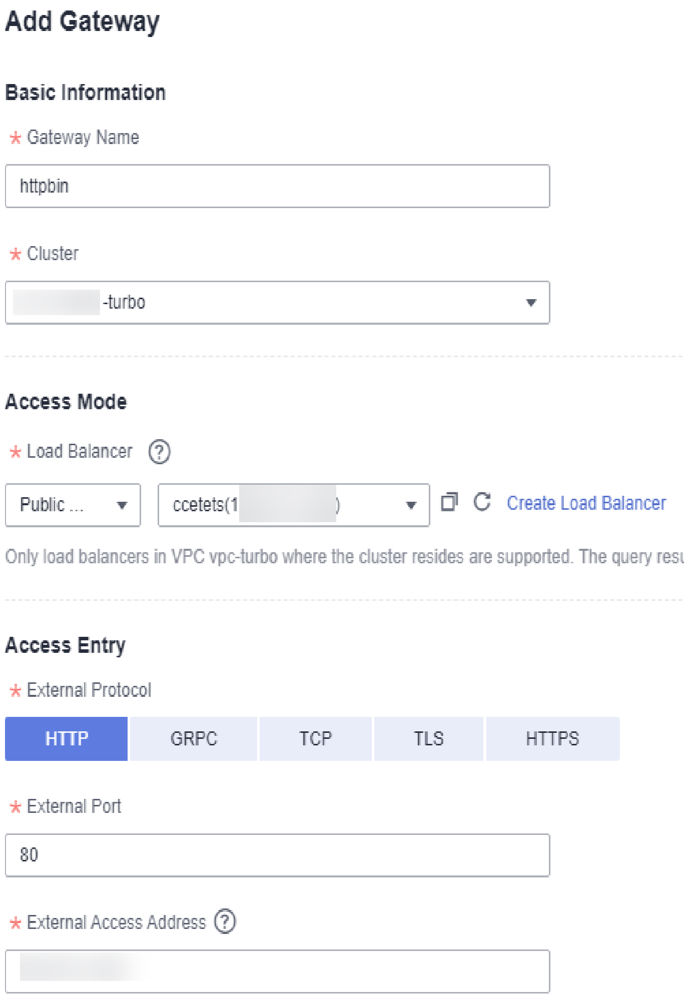
Configuration Diagnosis Result

Item	Result	Operation
The Service port name complies with the Istio specifications.	Succeeded Access Port 80 Protocol http	View Solution
The Service selector cannot contain version labels.	Succeeded	View Solution
The Service is configured with a default-version route. The route configuration is correct.	Succeeded	View Solution

10. In the navigation pane, choose **Gateway Management**. Then, click **Add Gateway** in the upper right corner. Configure the parameters in the window that slides out from the right.

Configuration Information

- **Gateway Name**: Enter **httpbin**.
- **Cluster**: Select the cluster associated with the mesh.
- **Load Balancer**: Select Public network and select a load balancer.
- **External Protocol**: Select **HTTP**.
- **External Port**: Specify a port.
- **External Access Address**: Enter the public IP address of the load balancer selected for the **Load Balancer**.



Add Gateway

Basic Information

* Gateway Name

httpbin

* Cluster

[redacted]-turbo

Access Mode

* Load Balancer

Public ... ccetets(1[redacted]) Create Load Balancer

Only load balancers in VPC vpc-turbo where the cluster resides are supported. The query resu

Access Entry

* External Protocol

HTTP GRPC TCP TLS HTTPS

* External Port

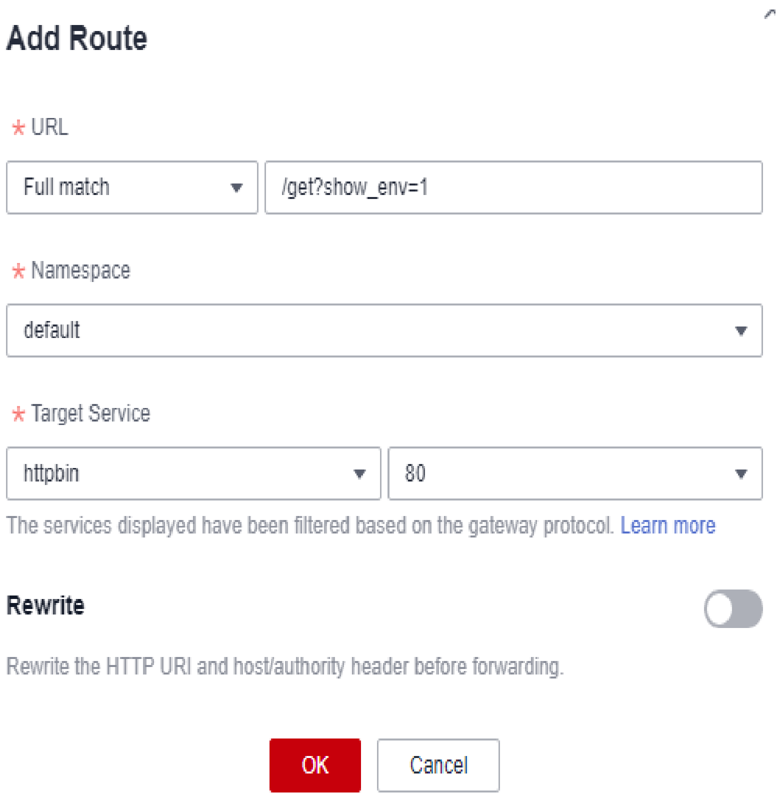
80

* External Access Address

[redacted]

– Click **+** under **Routing**. In the **Add Route** dialog box displayed, add a route.

  - **URL**: Select **Full match** and enter a mapping.
  - **Namespace**: Select the namespace to which the service belongs.
  - **Target Service**: Retain the default value.



Add Route

* URL

Full match ▾ /get?show_env=1

* Namespace

default ▾

* Target Service

httpbin ▾ 80 ▾

The services displayed have been filtered based on the gateway protocol. Learn more

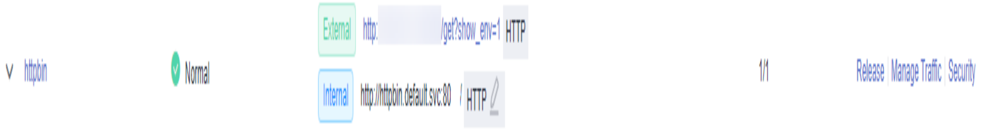
Rewrite ○

Rewrite the HTTP URI and host/authority header before forwarding.

OK Cancel

When the configuration is complete, click **OK**.

11. Click **OK**.
12. Click **Service Management** on the left. You can view the external access address of the created route in the **Access Address** column.



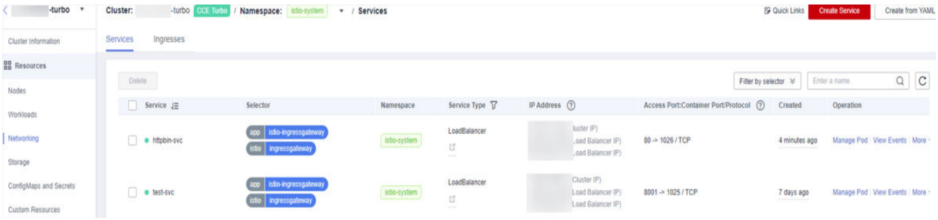
∨ httpbin	Normal	External http: /get?show_env=1 HTTP Internal http://httpbin.default.svc:80 / HTTP	1/1	Release Manage Traffic Security

13. Add **?show_env=1** to the mapped external access address, for example, **http://xxx.xxx.xxx:80/get?show_env=1**. You can view the value of the **x-forward-for** field. The IP address obtained by the gateway is the container IP address.

```
...
...
...
x-forward-for: xxxx
```

14. Return to the cluster console, click **Networking** in the navigation pane, and modify the configuration of the gateway service associated with the service as follows:

    Select the **istio-system** namespace.



[redacted]-turbo ▾ | Cluster: [redacted]-turbo CCE Turbo / Namespace: istio-system ▾ / Services

Quick Links | Create Service | Create from YAML

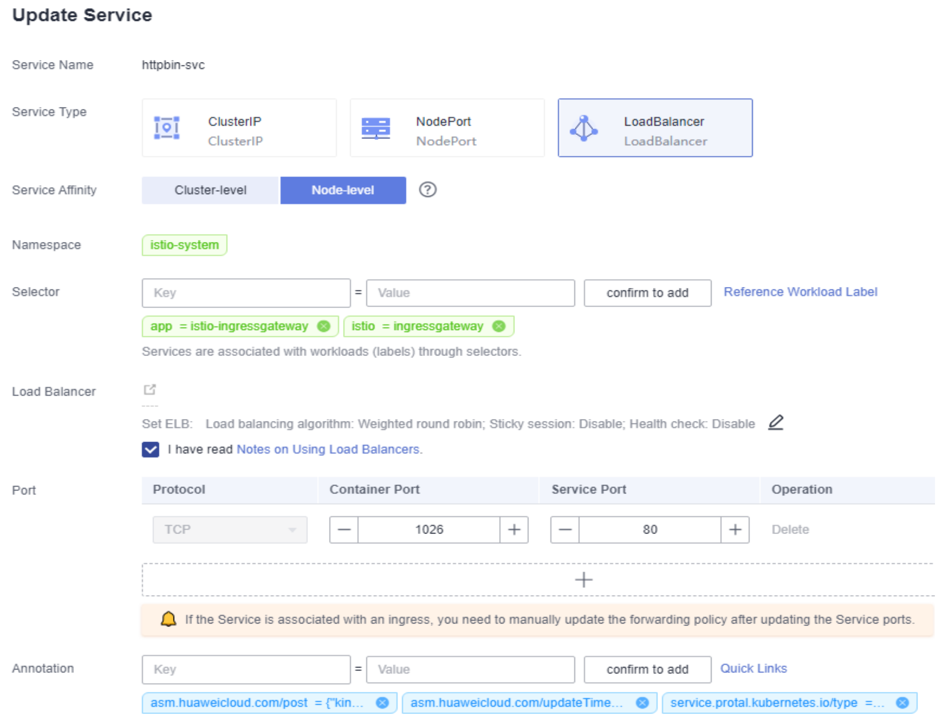
Cluster Information
Resources
Nodes
Workloads
Networking
Storage
ConfigMaps and Secrets
Custom Resources

Services | Ingresses

Delete | Filter by selector | Enter a name

	Service	Selector	Namespace	Service Type	IP Address	Access Port:Container Port/Protocol	Created	Operation
□	● httpbin-svc	app istio-ingressgateway istio ingressgateway	istio-system	LoadBalancer	[redacted] luster IP) .oad Balancer IP) .oad Balancer IP)	80 -> 1026 / TCP	4 minutes ago	Manage Pod View Events More
□	● test-svc	app istio-ingressgateway istio ingressgateway	istio-system	LoadBalancer	[redacted] Cluster IP) Load Balancer IP) Load Balancer IP)	8001 -> 1025 / TCP	7 days ago	Manage Pod View Events More

Click **More** > **Update** in the **Operation** column. On the **Update Service** page displayed, change **Service Affinity** to **Node-level**, select **I have read Notes on Using Load Balancers**, and click **OK**.



Update Service

Service Name httpbin-svc

Service Type: ClusterIP ClusterIP | NodePort NodePort | LoadBalancer LoadBalancer

Service Affinity: Cluster-level | Node-level

Namespace: istio-system

Selector: Key = Value | confirm to add | Reference Workload Label

app = istio-ingressgateway | istio = ingressgateway

Services are associated with workloads (labels) through selectors.

Load Balancer

Set ELB: Load balancing algorithm: Weighted round robin; Sticky session: Disable; Health check: Disable

☑ I have read Notes on Using Load Balancers.

Port

Protocol	Container Port	Service Port	Operation
TCP	1026	80	Delete

If the Service is associated with an ingress, you need to manually update the forwarding policy after updating the Service ports.

Annotation: Key = Value | confirm to add | Quick Links

asm.huaweicloud.com/post = {"kin... | asm.huaweicloud.com/updateTime... | service.protal.kubernetes.io/type =...

15. Return and refresh the external IP address accessed in 13. If the IP address obtained by the gateway in the **x-forward-for** field is the source IP address of the local host, the verification is complete.

```
...
...
...
x-forward-for: xxxx
```

4 Creating a Service Mesh with IPv4/IPv6 Dual Stack Enabled

You can create a CCE cluster with IPv4/IPv6 dual stack enabled and enable IPv4/IPv6 dual stack for the service mesh that the cluster is added to. IPv4/IPv6 dual stack allows services in the service mesh to use both IPv4 and IPv6 addresses for service-to-service interactions. After an IPv4/IPv6 dual-stack gateway is added for the service mesh, you can provide services for users using an IPv6 client. This section describes how you can create a service mesh with IPv4/IPv6 dual stack, so that services in the service mesh can communicate with each other using IPv6 addresses.

Application Scenarios

- If an IPv6 address is required for service access and traffic management, you can enable IPv4/IPv6 dual stack.
- If you provide services for users who use IPv6 clients, you can create a gateway for a service mesh with IPv4/IPv6 dual stack enabled.

Constraints

- Constraints on enabling IPv4/IPv6 dual stack for a service mesh

Service Mesh Edition	Istio Version	Cluster Type	Cluster Network Type	Remarks
Basic	1.18 or later	CCE Turbo clusters	Cloud native network 2.0	IPv6 needs to be enabled for the clusters. For details, see **Creating an IPv4/IPv6 Dual-Stack Cluster in CCE**.

- Constraints on creating an IPv4/IPv6 dual-stack gateway

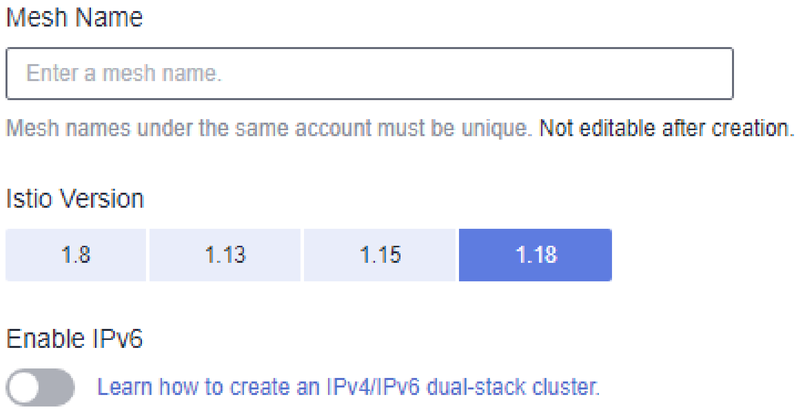
Service Mesh Edition	Istio Version	Load Balancer Type	Load Balancer Specification	Remarks
Basic	1.18 or later	Dedicated	Network load balancing (Layer 4)	The load balancer has an IPv6 address.

- IPv4/IPv6 dual stack cannot be disabled once it is enabled for a service mesh. IPv4/IPv6 dual stack cannot be enabled for an existing service mesh.
- IPv4/IPv6 dual stack is only available for service meshes of v1.18 or later, but it cannot be enabled for a service mesh that is upgraded to v1.18 or later.

Creating a Service Mesh with IPv6 Addresses

Step 1 Log in to the ASM console, purchase a service mesh, and configure the parameters as follows:

- **Mesh Edition**: Select **Basic edition**.
- **Mesh Name**: Enter a service mesh name.
- **Istio Version**: Select 1.18 or later.
- **Enable IPv6**: If this option is enabled, CCE clusters that meet the conditions will be displayed.



Mesh Name

Enter a mesh name.

Mesh names under the same account must be unique. Not editable after creation.

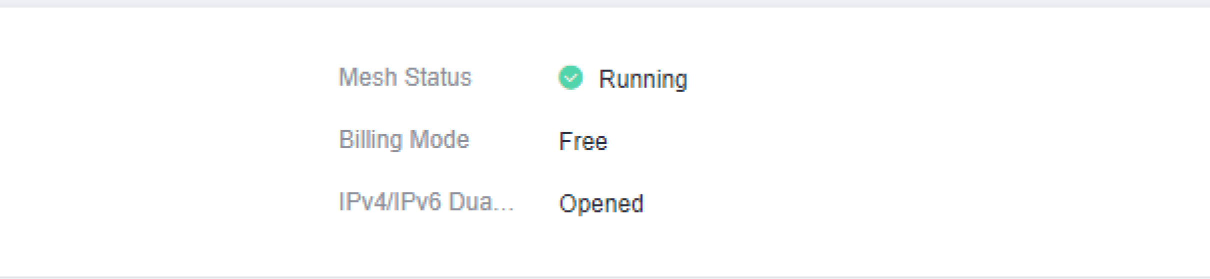
Istio Version

Enable IPv6

Configure other parameters based on site requirements.

Step 2 Click the service mesh name to access the details page.

On the **Mesh Configuration** > **Basic Information** tab, you can see that IPv4/IPv6 dual stack has been enabled.



Mesh Status ● Running

Billing Mode Free

IPv4/IPv6 Dua... Opened

----End

Adding an IPv4/IPv6 Dual-Stack Gateway

Step 1 Log in to the ASM console. On the service mesh list page, click the name of the service mesh with IPv4/IPv6 dual stack enabled. In the navigation pane, choose **Gateway Management**. Click **Add Gateway** and configure the parameters as follows:

- **Access Mode**: Select IPv4/IPv6 dual stack.
- **Load Balancer**: Select **Dedicated**. The dedicated load balancer must have an IPv6 address.

Configure other parameters based on site requirements.

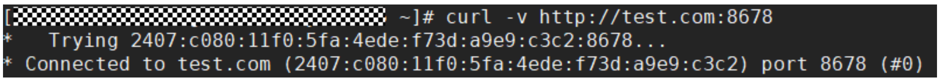
NOTE

If IPv4/IPv6 dual stack is enabled, only domain names are allowed to access the gateway.

----End

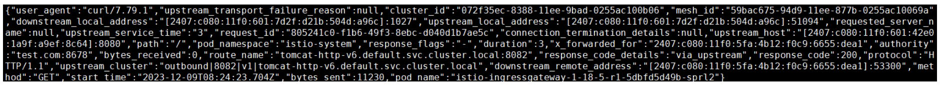
Verification

1. Configure domain name resolution for the client, so that the domain name is mapped to the IPv6 address of the gateway. This way, the client can access the gateway using the domain name.



```
[[redacted] ~]# curl -v http://test.com:8678
*   Trying 2407:c080:11f0:5fa:4ede:f73d:a9e9:c3c2:8678...
* Connected to test.com (2407:c080:11f0:5fa:4ede:f73d:a9e9:c3c2) port 8678 (#0)
```

2. View the IPv6 request information in the ingressgateway log.



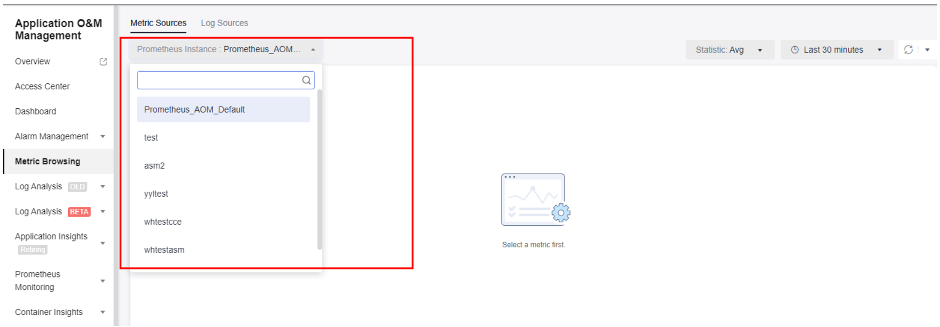
```
{"user_agent":"curl/7.79.1","upstream_transport_failure_reason":null,"cluster_id":"072f35ec-8388-11ee-9bad-0255ac100b06","mesh_id":"59bac675-94d9-11ee-877b-0255ac10069a"
,"downstream_local_address":"[2407:c080:11f0:601:7d2f:d21b:504d:a96c]:1027","upstream_local_address":"[2407:c080:11f0:601:7d2f:d21b:504d:a96c]:51094","requested_server_n
ame":null,"upstream_service_time":"3","request_id":"805241c0-f1b6-49f3-8ebc-d040d1b7ae5c","connection_termination_details":null,"upstream_host":"[2407:c080:11f0:601:42e0
:1a9f:a9ef:8c64]:8080","path":"/","pod_namespace":"istio-system","response_flags":"-","duration":3,"x_forwarded_for":"2407:c080:11f0:5fa:4b12:f0c9:6655:dea1","authority"
:"test.com:8678","bytes_received":0,"route_name":"tomcat-http-v6.default.svc.cluster.local:8082","response_code_details":"via_upstream","response_code":200,"protocol":"H
TTP/1.1","upstream_cluster":"outbound|8082|v1|tomcat-http-v6.default.svc.cluster.local","downstream_remote_address":"[2407:c080:11f0:5fa:4b12:f0c9:6655:dea1]:53300","met
hod":"GET","start_time":"2023-12-09T08:24:23.704Z","bytes_sent":11230,"pod_name":"istio-ingressgateway-1-18-5-r1-5dbfd5d49b-sprl2"}
```

5 How Do I Query Application Metrics in AOM?

After application metrics are enabled for a service mesh and the service mesh is connected to Huawei Cloud Application Operations Management (AOM), you can query the application metrics in AOM.

Step 1 Log in to the AOM console and choose **Metric Browsing**.

Step 2 Click the **Metric Sources** tab and select the Prometheus instance whose metrics you want to query.



Step 3 Query metrics reported to AOM.

- All metrics

You can enter a metric name to query metrics.

In addition, you can add conditions to query related information.

- Prometheus statement

Enter the PromQL statement to query metrics.

----End

6 Reducing the Agency Permissions of ASM Users

Background

ASM permission management is implemented through IAM agencies. However, users authorized prior to July 2024 may have excessive agency permissions. For security purposes, you are advised to reduce the agency permissions.

Procedure

Step 1 Log in to the **IAM console**.

Step 2 In the navigation pane, choose **Agencies**. Then, search for **asm_admin_trust** and click its name.

Step 3 Click the **Permissions** tab and delete all permissions.

Step 4 Click **Authorize**, search for and select the **CCE Administrator** policy, and click **Next**. Set **Scope** to **Region-specific projects**, select the region where the ASM service is to be used, and click **OK**.

Step 5 Click **Authorize**, search for and select the **Tenant Guest** policy, and click **Next**. Set **Scope** to **Region-specific projects**, select the region where the ASM service is to be used, and click **OK**.

----End

📖 NOTE

Grant permissions that deleted in **Step 3** again. Otherwise, an exception may occur.

7 Istio-ingressgateway HA Configuration

7.1 Gateway Workloads

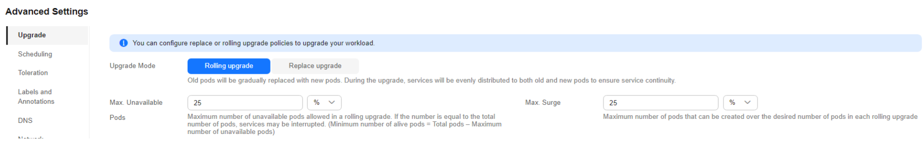
7.1.1 Rolling Upgrade Policy

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
spec:
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 10%
```

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



You can configure replace or rolling upgrade policies to upgrade your workload.

Upgrade Mode: Rolling upgrade | Replace upgrade

Old pods will be gradually replaced with new pods. During the upgrade, services will be evenly distributed to both old and new pods to ensure service continuity.

Max. Unavailable Pods: 25 %

Maximum number of unavailable pods allowed in a rolling upgrade. If the number is equal to the total number of pods, services may be interrupted. (Minimum number of alive pods = Total pods – Maximum number of unavailable pods)

Max. Surge: 25 %

Maximum number of pods that can be created over the desired number of pods in each rolling upgrade

Parameter description

Parameter	Description
Max. Unavailable Pods (maxUnavailable)	The maximum number or percentage of pods that can be deleted based on the value of **spec.replicas**. The recommended value is **0**. For example, if **spec.replicas** is set to **3**, there should be at least three pods during the upgrade.

Parameter	Description
Max. Surge (maxSurge)	The percentage of pods that are allowed based on the value of **spec.replicas**. The recommended value is **10**. For example, if **spec.replicas** is set to **3**, there should be a maximum of four pods during the upgrade, and 10% of pods can be added each time. During the upgrade, the value is converted into a number and rounded up. (10% of pods in this example is one pod.) So, you can also set this parameter to the maximum number of pods.

----End

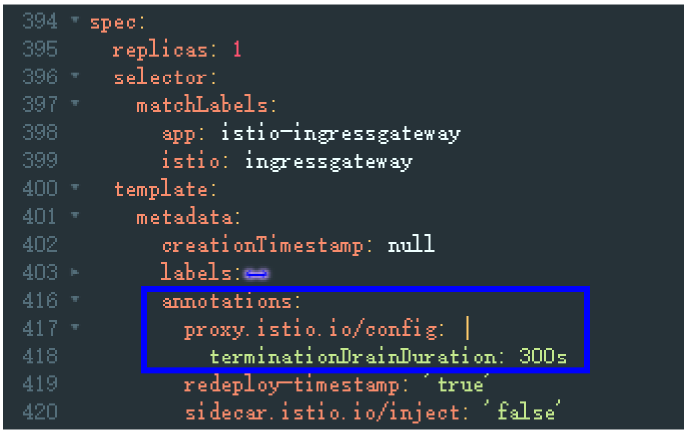
NOTE

- If only **maxUnavailable** and **maxSurge** are configured, the workload rolling upgrade will not be triggered. This means pods will not restart. The new values take effect in the next rolling upgrade.
- When you redeploy a workload on the CCE console, you are performing a workload rolling upgrade. The workload version does not change.

7.1.2 Graceful Pod Deletion

7.1.2.1 Termination Drain Duration of the Istio-proxy Container

Step 1 Log in to the CCE Console console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.



```
spec:
  replicas: 1
  selector:
    matchLabels:
      app: istio-ingressgateway
      istio: ingressgateway
  template:
    metadata:
      creationTimestamp: null
      labels:
      annotations:
        proxy.istio.io/config: |
          terminationDrainDuration: 300s
        redeploy-timestamp: 'true'
        sidecar.istio.io/inject: 'false'
```

Step 2 Configure the following parameters in YAML:

```
spec:
  template:
metadata:
      annotations:
        proxy.istio.io/config: |
          terminationDrainDuration: 300s
```

Parameter description

Parameter	Description
terminationDrainDuration	The amount of time that the istio-proxy container waits for before it is killed or shut down when the applications in the container are terminated. The recommended value is **300s** when the Istio-ingressgateway is used.

----End

NOTE

Configuring this parameter will trigger a workload rolling upgrade. This means pods will restart immediately. If an old pod is deleted, the old value (**5s**) will be used.

7.1.2.2 Pod Scale-In Time Window

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

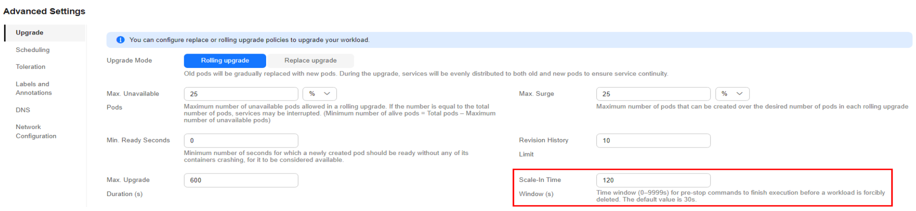
Step 2 Configure the following parameters in YAML:

```
spec:
  template:
    spec:
      terminationGracePeriodSeconds: 301
```

Parameter description

Parameter	Description
Scale-In Time Window (s) (terminationGracePeriodSeconds)	Graceful deletion time. The default value is **301s** (greater than the termination drain time of the Istio-proxy container). When a pod is deleted, a SIGTERM signal is sent and the system waits for the applications in the container to terminate. If the applications are not terminated within the time specified by **terminationGracePeriodSeconds**, a SIGKILL signal is sent to forcibly terminate the pod.

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



Advanced Settings

You can configure replace or rolling upgrade policies to upgrade your workload.

Upgrade Mode: Rolling upgrade | Replace upgrade

Old pods will be gradually replaced with new pods. During the upgrade, services will be evenly distributed to both old and new pods to ensure service continuity.

Max. Unavailable Pods: 25 %

Maximum number of unavailable pods allowed in a rolling upgrade. If the number is equal to the total number of pods, services may be interrupted. (Minimum number of alive pods = Total pods – Maximum number of unavailable pods)

Max. Surge: 25 %

Maximum number of pods that can be created over the desired number of pods in each rolling upgrade

Min. Ready Seconds: 0

Minimum number of seconds for which a newly created pod should be ready without any of its containers crashing, for it to be considered available.

Revision History Limit: 10

Max. Upgrade Duration (s): 600

Scale-In Time Window (s): 120

Time window (0–9999s) for pre-stop commands to finish execution before a workload is forcibly deleted. The default value is 30s.

----End

NOTE

- Configuring this parameter will trigger a workload rolling upgrade. This means pods will restart immediately. If an old pod is deleted, the old value (**30s**) will be used.
- For workloads injected with sidecars, the larger value between the termination drain duration of the Istio-proxy container and the graceful deletion time of the service container is used.

7.1.3 Scheduling Policies

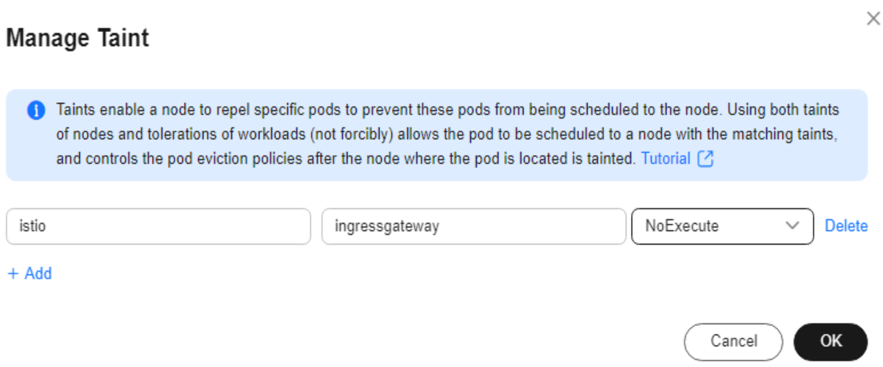
7.1.3.1 Taints and Tolerations

Step 1 Log in to the CCE Console console and click the cluster name to access the cluster console. In the navigation pane, choose **Nodes** > **Nodes**. Then, locate the node and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
spec:
  taints:
    - key: istio
      value: ingressgateway
      effect: NoExecute
```

You can also choose **More** > **Manage Taint** in the **Operation** column of the node to configure the parameters.



Manage Taint

Taints enable a node to repel specific pods to prevent these pods from being scheduled to the node. Using both taints of nodes and tolerations of workloads (not forcibly) allows the pod to be scheduled to a node with the matching taints, and controls the pod eviction policies after the node where the pod is located is tainted. Tutorial

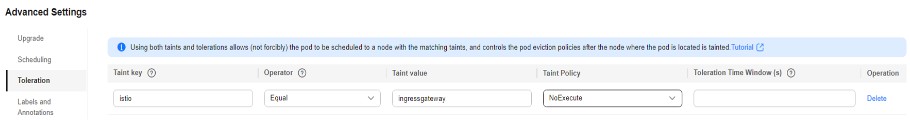
istio | ingressgateway | NoExecute | Delete

Step 3 Log in to the CCE Console console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

Step 4 Configure the following parameters in YAML:

```
spec:
  template:
spec:
  tolerations:
   - key: istio
     operator: Equal
     value: ingressgateway
     effect: NoExecute
```

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



Using both taints and tolerations allows (not forcibly) the pod to be scheduled to a node with the matching taints, and controls the pod eviction policies after the node where the pod is located is tainted.

Taint key	Operator	Taint value	Taint Policy	Toleration Time Window (s)	Operation
istio	Equal	ingressgateway	NoExecute		Delete

Parameters for configuring a taint tolerance policy

Parameter	Description
Taint Key	Key of a node taint.
Operator	- **Equal**: matches the nodes with the specified taint key (mandatory) and value. If the taint value is left blank, all taints with the key the same as the specified taint key will be matched. - **Exists**: matches the nodes with the specified taint key. In this case, the taint value cannot be specified. If the taint key is left blank, all taints will be tolerated.
Taint Value	Taint value specified if **Operator** is set to **Equal**.
Taint Policy	- **All**: All taint policies are matched. - **NoSchedule**: Only the **NoSchedule** taint is matched. - **PreferNoSchedule**: Only the **PreferNoSchedule** taint is matched. - **NoExecute**: Only the **NoExecute** taint is matched.
Toleration Time Window (s)	**tolerationSeconds**, which is configurable only when **Taint Policy** is set to **NoExecute**. Within the tolerance time window, pods still run on the node with taints. After the time expires, the pods will be evicted.

----End

NOTE

Configuring taint tolerance parameters will trigger a workload rolling upgrade. This means pods will restart immediately.

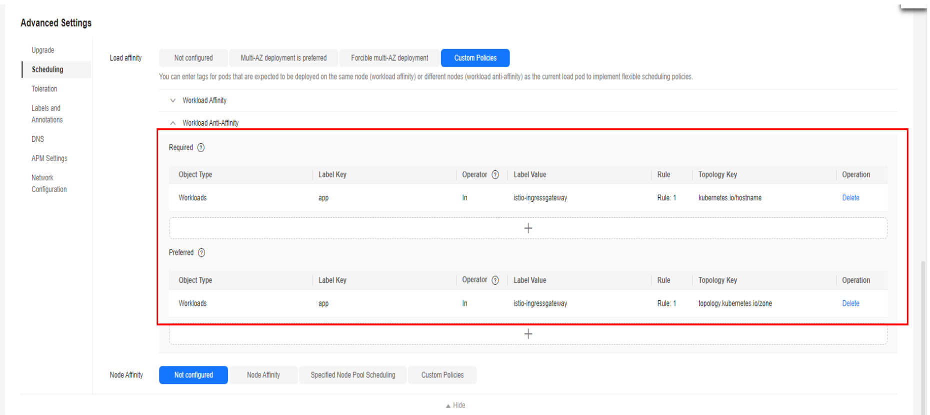
7.1.3.2 Load Affinity

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
spec:
  template:
spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - istio-ingressgateway
              namespaces:
                - istio-system
              topologyKey: kubernetes.io/hostname
          preferredDuringSchedulingIgnoredDuringExecution:
            - weight: 1
              podAffinityTerm:
                labelSelector:
                  matchExpressions:
                    - key: app
                      operator: In
                      values:
                        - istio-ingressgateway
                namespaces:
                  - istio-system
                topologyKey: topology.kubernetes.io/zone
```

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



You can enter tags for pods that are expected to be deployed on the same node (workload affinity) or different nodes (workload anti-affinity) as the current load pod to implement flexible scheduling policies.

Required

Object Type	Label Key	Operator	Label Value	Rule	Topology Key	Operation
Workloads	app	In	istio-ingressgateway	Rule: 1	kubernetes.io/hostname	Delete

Preferred

Object Type	Label Key	Operator	Label Value	Rule	Topology Key	Operation
Workloads	app	In	istio-ingressgateway	Rule: 1	topology.kubernetes.io/zone	Delete

Recommended value description:

- **Required**: Pods are scheduled to different nodes.
- **Preferred**: Pods are preferentially scheduled to nodes in different AZs.

Load affinity policies

Policy	Trigger Type	Description
Workload Affinity	Required	Hard constraint, which corresponds to **requiredDuringSchedulingIgnoredDuringExecution** in YAML for specifying the conditions that must be met. Select pods that require affinity by label. If such pods already run on a node in the topology key, the scheduler will **forcibly** schedule the created pods to that topology key. **Note:** If multiple affinity rules are configured, multiple labels will be used to filter pods that require affinity, and the newly created pods must have affinity with all pods that meet the label filtering conditions. In this way, all pods that meet the label filtering conditions locate in the same topology key for scheduling.
	Preferred	Soft constraint, which corresponds to **preferredDuringSchedulingIgnoredDuringExecution** in YAML for specifying the conditions that need to be met as much as possible. Select pods that require affinity by label. If such pods already run on a node in the topology key, the scheduler will **preferentially** schedule the created pods to that topology key. **Note:** If multiple affinity rules are configured, multiple labels will be used to filter pods that require affinity, and the newly created pods will be preferentially to have affinity with multiple pods that meet the label filtering conditions. However, even if no pod meets the label filtering conditions, a topology key will be selected for scheduling.

Policy	Trigger Type	Description
Workload Anti-Affinity	Required	Hard constraint, which corresponds to **requiredDuringSchedulingIgnoredDuringExecution** in YAML for specifying the conditions that must be met. Select one or more pods that require anti-affinity by label. If such pods already run on a node in the topology key, the scheduler will **not** schedule the created pods to that topology key. **Note:** If multiple anti-affinity rules are configured, multiple labels will be used to filter pods that require anti-affinity, and the newly created pods must have anti-affinity with all pods that meet the label filtering conditions. In this way, all the topology keys where the pods that meet the label filtering conditions locate will not be scheduled.
	Preferred	Soft constraint, which corresponds to **preferredDuringSchedulingIgnoredDuringExecution** in YAML for specifying the conditions that need to be met as much as possible. Select one or more pods that require anti-affinity by label. If such pods already run on a node in the topology key, the scheduler will **preferentially** schedule the created pods to other topology keys. **Note:** If multiple anti-affinity rules are configured, multiple labels will be used to filter pods that require anti-affinity, and the newly created pods will be preferentially to have anti-affinity with multiple pods that meet the label filtering conditions. However, even if all topology keys involve the pods that require anti-affinity, a topology key will be selected for scheduling.

Parameters for configuring workload affinity/anti-affinity scheduling policies

Parameter	Description
Weight	This parameter is only available in a **preferred** scheduling policy. The weight ranges from **1** to **100**. During scheduling, the scheduler adds the weight to the scores of other priority functions and schedules pods to the node with the largest total score.

Parameter	Description
Namespace	Namespace for which the scheduling policy takes effect.
Topology Key	A topology key (**topologyKey**) determines the range of nodes to be scheduled based on node labels. For example, if the node label is **kubernetes.io/hostname**, the range of nodes is determined by node name. Nodes with different names are in different topology keys. In this case, a topology key contains only one node. If the specified label is **kubernetes.io/os**, the range of nodes is determined by node OS. Nodes running different OSs belong to different topology keys. In this case, a topology key may contain multiple nodes. After the node range is determined using the topology key, configure the policy for scheduling, including the label key, operator, and label value. The minimum unit for scheduling is a topology key. For example, if a node in a topology key meets the load affinity policy, all nodes in the topology key can be scheduled.
Label Key	When configuring a workload affinity or anti-affinity policy, enter the workload label to be matched. Both default labels and custom labels are supported.
Operator	The following operators are supported: ● **In**: The label of the affinity or anti-affinity object is in the label value list (**values** field). ● **NotIn**: The label of the affinity or anti-affinity object is not in the label value list (**values** field). ● **Exists**: The affinity or anti-affinity object has a specified label key. ● **DoesNotExist**: The affinity or anti-affinity object does not have a specified label key.
Label Value	When configuring a workload affinity or anti-affinity policy, enter the value of the workload label.

----End

NOTE

Configuring load affinity parameters will trigger a workload rolling upgrade. This means pods will restart immediately.

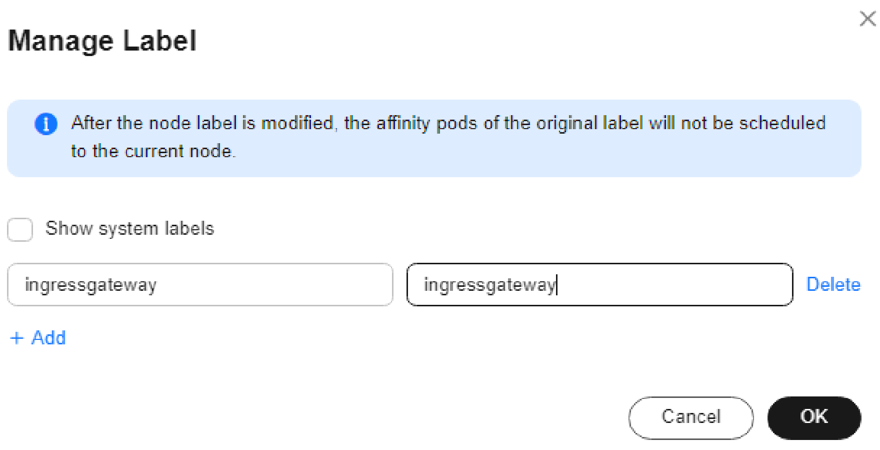
7.1.3.3 Node Affinity

Step 1 Log in to the CCE Console console and click the cluster name to access the cluster console. In the navigation pane, choose **Nodes** > **Nodes**. Then, locate the node and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
metadata:
  labels:
    istio: ingressgateway
```

You can also choose **More** > **Manage Label** in the **Operation** column of the **node** to configure the parameters.



Manage Label

After the node label is modified, the affinity pods of the original label will not be scheduled to the current node.

□ Show system labels

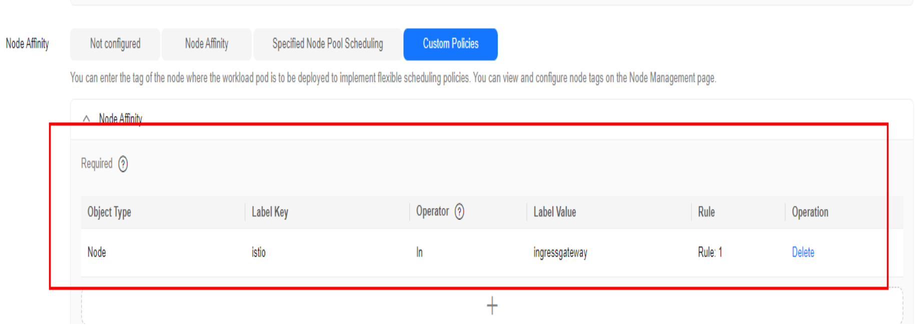
ingressgateway	ingressgateway

Step 3 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

Step 4 Configure the following parameters in YAML:

```
spec:
  template:
spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: istio
                    operator: In
                    values:
                      - ingressgateway
```

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



Node Affinity: Not configured | Node Affinity | Specified Node Pool Scheduling | **Custom Policies**

You can enter the tag of the node where the workload pod is to be deployed to implement flexible scheduling policies. You can view and configure node tags on the Node Management page.

Node Affinity

Required

Object Type	Label Key	Operator	Label Value	Rule	Operation
Node	istio	In	ingressgateway	Rule: 1	Delete

Recommended value description:

Required: Pods are scheduled to the node with the **istio:ingressgateway** label.

Node affinity parameters

Parameter	Description
Required	Hard constraint, which corresponds to **requiredDuringSchedulingIgnoredDuringExecution** for specifying the conditions that must be met. If multiple rules that must be met are added, scheduling will be performed when only one rule is met.
Preferred	Soft constraint, which corresponds to **preferredDuringSchedulingIgnoredDuringExecution** for specifying the conditions that need to be met as much as possible. If multiple rules that are preferentially met are added, scheduling will be performed even if one or none of the rules is met.

Parameters for configuring node affinity scheduling policies

Parameter	Description
Label Key	When configuring node affinity, enter the node label to be matched. Both default labels and custom labels are supported.
Operator	The following operators are supported: ● **In**: The label of the affinity or anti-affinity object is in the label value list (**values** field). ● **NotIn**: The label of the affinity or anti-affinity object is not in the label value list (**values** field). ● **Exists**: The affinity or anti-affinity object has a specified label key. ● **DoesNotExist**: The affinity or anti-affinity object does not have a specified label key. ● **Gt**: (available only for node affinity) The label value of the scheduled node is greater than the list value (string comparison). ● **Lt**: (available only for node affinity) The label value of the scheduled node is less than the list value (string comparison).
Label Value	When configuring node affinity, enter the value of the node label.

----End

NOTE

Configuring node affinity parameters will trigger a workload rolling upgrade. This means pods will restart immediately.

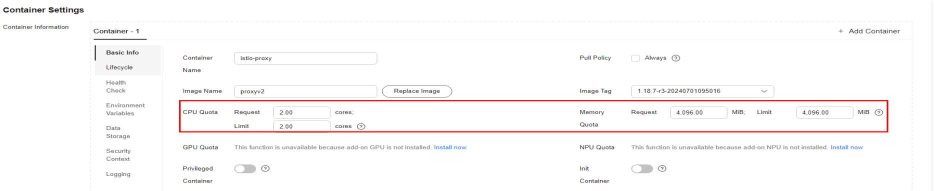
7.1.4 Container Resource Requests and Limits

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Workloads**. Then, locate the workload you want to upgrade and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
spec:
  template:
spec:
      containers:
        - name: istio-proxy
          resources:
            limits:
              cpu: '2'
              memory: 4Gi
            requests:
              cpu: '2'
              memory: 4Gi
```

You can also click **Upgrade** in the **Operation** column of the **workload** to configure the parameters.



Container Name: istio-proxy

Pull Policy: □ Always

Image Name: proxyv2

Image Tag: 1.18.7-r3-20240701095016

CPU Quota: Request 2.00 cores; Limit 2.00 cores

Memory Quota: Request 4,096.00 MiB; Limit 4,096.00 MiB

GPU Quota: This function is unavailable because add-on GPU is not installed. Install now

NPU Quota: This function is unavailable because add-on NPU is not installed. Install now

Privileged Container

Init Container

Recommended value description:

- The gateway is sensitive to CPU. The recommended ratio of **Request** to **Limit** is 1:1. If the value of **Request** is too small and the value of **Limit** is too large, node resources are overcommitted.
- 2 vCPUs and 4 GiB of memory are only for reference. Adjust the values based on the pressure test result.

CPU quota parameters

Parameter	Description
Request	Minimum number of CPU cores required by a container. Resources are scheduled for the container based on this value. The container can be scheduled to a node only when the total available CPU on the node is greater than or equal to the requested number of CPU cores.
Limit	Maximum number of CPU cores available for a container.

Memory quota parameters

Parameter	Description
Request	Minimum amount of memory required by a container. Resources are scheduled for the container based on this value. The container can be scheduled to a node only when the total available memory on the node is greater than or equal to the requested memory.
Limit	Maximum amount of memory available for a container. When the memory usage exceeds the specified memory limit, the instance may restart, which affects the normal use of the workload.

----End

NOTE

Configuring container resource requests and limits will trigger a workload rolling upgrade. This means pods restart immediately.

7.2 Gateway Services

7.2.1 Service Affinity

CAUTION

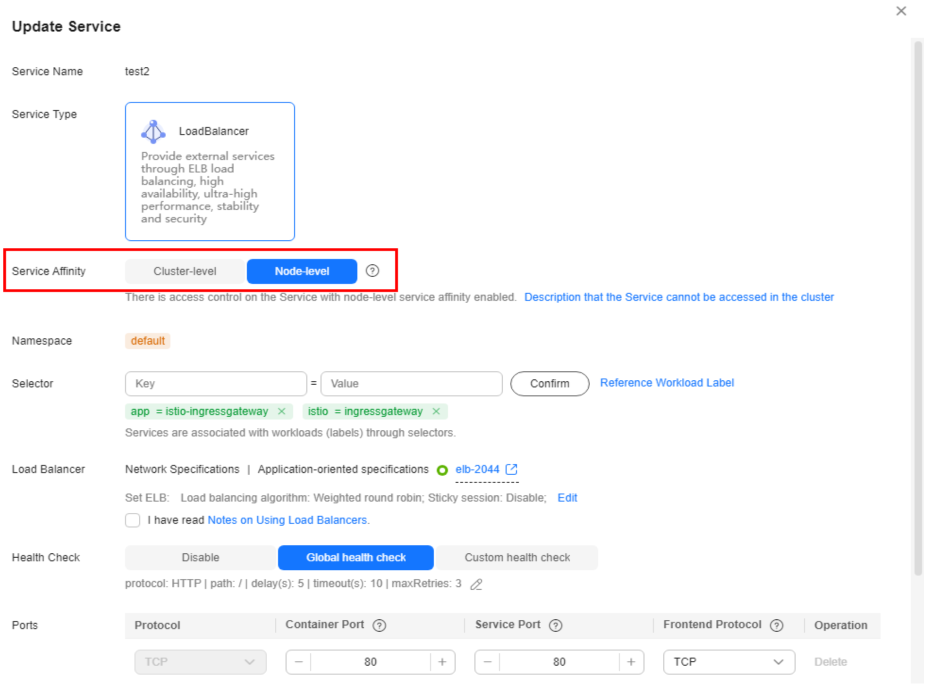
If node affinity is configured, the load balancer may fail to be accessed in some scenarios. For details, see **Why the ELB Address Cannot Be used to Access Workloads in a Cluster?**

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Services & Ingresses** > **Services**. Then, locate the Service you want to operate and choose **More** > **Edit YAML** in the **Operation** column.

Step 2 Configure the following parameters in YAML:

```
spec:
  type: LoadBalancer
  externalTrafficPolicy: Local
  allocateLoadBalancerNodePorts: true
```

You can also choose **More** > **Update** in the **Operation** column of the **Service** to configure the parameters.



Update Service

Service Name test2

Service Type LoadBalancer
Provide external services through ELB load balancing, high availability, ultra-high performance, stability and security

Service Affinity Cluster-level [✓ Node-level]
There is access control on the Service with node-level service affinity enabled. Description that the Service cannot be accessed in the cluster

Namespace default

Selector Key = Value Confirm Reference Workload Label
app = istio-ingressgateway × istio = ingressgateway ×
Services are associated with workloads (labels) through selectors.

Load Balancer Network Specifications | Application-oriented specifications elb-2044
Set ELB: Load balancing algorithm: Weighted round robin; Sticky session: Disable; Edit
□ I have read Notes on Using Load Balancers.

Health Check Disable [✓ Global health check] Custom health check
protocol: HTTP | path: / | delay(s): 5 | timeout(s): 10 | maxRetries: 3

Ports

Protocol	Container Port	Service Port	Frontend Protocol	Operation
TCP	80	80	TCP	Delete

Parameter description

Parameter	Description
Service Affinity (externalTrafficPolicy)	The options are as follows: **Cluster-level** (**Cluster** in YAML): Requests are forwarded within the cluster. The backend workload can be accessed from any node IP address and service port. However, routing hops bring in performance loss, and the client source IP address cannot be obtained. **Node-level** (**Local** in YAML): Requests are forwarded only to the pod on the local node. If there is no pod, the requests will be suspended. If dedicated load balancers are deployed for CCE Turbo clusters, passthrough networking is supported to reduce the network latency and ensure zero performance loss. In this case, the value must be **Cluster-level** (**Cluster** in YAML). In other scenarios, requests are transmitted through a load balancer to a node and then forwarded to the target pod through the Service. In this case, the recommended value is **Node-level** (**Local** in YAML). The traffic is finally forwarded to the pod on the local node instead of being balanced to pods on other nodes. This avoids request exceptions caused by cross-node network faults.

----End

7.2.2 Load Balancer Health Check

Step 1 Log in to the **CCE Console** console and click the cluster name to access the cluster console. In the navigation pane, choose **Services & Ingresses** > **Services**. Then, locate the Service you want to operate and choose **More** > **Edit YAML** in the **Operation** column.

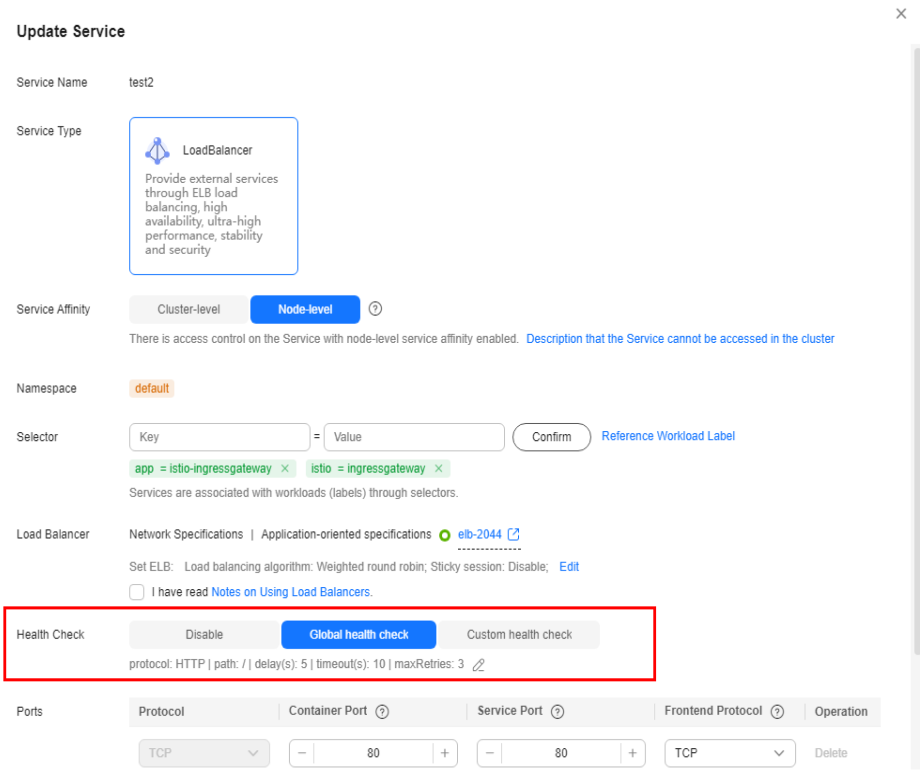
Step 2 Configure the following parameters in YAML:

```
metadata:
  annotations:
    kubernetes.io/elb.health-check-flag: 'on'
    kubernetes.io/elb.health-check-option: '{"protocol":"TCP","delay":"5","timeout":"10","max_retries":"3"}'
```

Parameter description

Parameter	Description
kubernetes.io/elb.health-check-flag	Whether to enable the ELB health check. The recommended value is **on**. If this option is enabled, the **kubernetes.io/elb.health-check-option** field must also be specified at the same time.
kubernetes.io/elb.health-check-option	**protocol**: health check protocol. **delay**: health check interval, in seconds. **timeout**: health check timeout, in seconds. **max_retries**: maximum number of health check retries.

You can also choose **More** > **Update** in the **Operation** column of the **Service** to configure the parameters.



Update Service

Service Name: test2

Service Type: LoadBalancer
Provide external services through ELB load balancing, high availability, ultra-high performance, stability and security

Service Affinity: Cluster-level [✓ Node-level]
There is access control on the Service with node-level service affinity enabled. Description that the Service cannot be accessed in the cluster

Namespace: default

Selector: Key = Value Confirm Reference Workload Label
app = istio-ingressgateway ×   istio = ingressgateway ×
Services are associated with workloads (labels) through selectors.

Load Balancer: Network Specifications | Application-oriented specifications ● elb-2044
Set ELB: Load balancing algorithm: Weighted round robin; Sticky session: Disable; Edit
□ I have read Notes on Using Load Balancers.

Health Check: Disable [✓ Global health check] Custom health check
protocol: HTTP | path: / | delay(s): 5 | timeout(s): 10 | maxRetries: 3

Ports

Protocol	Container Port	Service Port	Frontend Protocol	Operation
TCP	80	80	TCP	Delete

----End