ALIBABA CLOUD

Alibaba Cloud

Alibaba Cloud Service Mesh

FAQ

Document Version: 20220525

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.What is the difference between namespaces in an ASM instance and those of the clusters in the data plane managed by the ASM instance?

Alibaba Cloud Service Mesh (ASM) is a managed service mesh platform. It decouples the control plane of a service mesh from the data plane that the control plane manages. You can separately manage the lifecycle of the control plane and data plane. The data plane is where Container Service for Kubernetes (ACK) clusters reside. In the ASM console, you can create, define, and delete namespaces in an ASM instance to manage resources in the data plane. This topic describes the difference between namespaces in an ASM instance and those of the clusters in the data plane managed by the ASM instance. This topic also describes how to enable automatic sidecar injection for a namespace in the ASM console.

Difference

The namespaces that you create in an ASM instance, whether in the ASM console or by using the kubectl client, belong only to the ASM instance. They are independent of the Kubernetes clusters in the data plane that are managed by the ASM instance. Therefore, the namespaces in the control plane of the ASM instance may be different from the namespaces of the Kubernetes clusters in the data plane. When you create or delete namespaces for the ASM instance, the namespaces of the Kubernetes clusters in the data plane are not affected.

Enable automatic sidecar injection for a namespace

You can enable automatic sidecar injection for a namespace of a Kubernetes cluster. After automatic sidecar injection is enabled, the istio-injection=enabled tag is added to the namespace. An Envoy proxy is automatically injected as a sidecar into each pod that is created in the namespace. You can also disable automatic sidecar injection for the namespace. After automatic sidecar injection is disabled, the istio-injection=disabled tag is added to the namespace.

To enable automatic sidecar injection for a namespace of a Kubernetes cluster in the ASM console, perform the following steps:

1.
2.
3.
4.
5. On the **Namespaces** page, find the namespace for which you want to enable automatic sidecar injection and click **Enable Automatic Sidecar Injection** in the **Automatic Sidecar Injection** column.

> **Note** If automatic sidecar injection is already enabled for the namespace, **Disable Automatic Sidecar Injection** appears in the **Automatic Sidecar Injection** column. If you want to disable this feature, click Disable Automatic Sidecar Injection. In the **Confirm** message, click **OK**.

6. In the **Confirm** message, click **OK**.

> **Note** After you enable or disable automatic sidecar injection for namespaces in the ASM console, automatic sidecar injection is automatically enabled or disabled for the same namespaces of the Kubernetes clusters in the data plane. However, the creation and deletion of namespaces in the control plane are not synchronized to the Kubernetes clusters in the data plane. This ensures the stability of the resources used by the Kubernetes clusters in the data plane.

2.How do I delete a namespace in the terminating state?

After you try to delete a namespace in a Kubernetes cluster, the namespace may be stuck in the terminating state for a long time. This topic describes how to delete a namespace in the terminating state.

Problem description

After you try to delete a namespace in a Kubernetes cluster, the namespace may be stuck in the terminating state for a long time.

```
$ kubectl delete ns <namespacename>
Error from server (Conflict): Operation cannot be fulfilled on namespaces "<namespacename>"
: The system is ensuring all content is removed from this namespace. Upon completion, this
namespace will automatically be purged by the system.
$ kubectl describe ns <namespacename>
Name: <namespacename>
Labels: <none>
Annotations: kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Na
mespace","metadata":{"annotations":{},"name":"<namespacename>","namespace":""}}
Status: Terminating
```

Cause

Residual resources exist in the namespace that you want to delete from the cluster.

Solution

To delete a namespace in the terminating state, you can delete the finalizers field in the namespace configuration.

This method can clear a namespace that is stuck in the terminating state. However, resources that belong to the namespace cannot be automatically deleted and thus left in the cluster. You must manually delete the residual resources. After the residual resources are deleted, you can perform the following steps to clear the array of the finalizers field and delete the finalizers field. Then, Kubernetes deletes the namespace in the terminating state.

1. Open a shell terminal. Run the following command to create a reverse proxy for your Kubernetes cluster:

   ```
   kubectl proxy
   ```

   A command output similar to the following one appears:

   ```
   Starting to serve on 127.0.0.1:8001
   ```

2. Open a new shell terminal. Define environment variables to connect to the Kubernetes cluster. Then, run the curl command to check the connectivity and authorization.

```
export TOKEN=$(kubectl describe secret $(kubectl get secrets | grep default | cut -f1
-d ' ') | grep -E '^token' | cut -f2 -d':' | tr -d '\t')
curl http://localhost:8001/api/v1/namespaces --header "Authorization: Bearer $TOKEN"
--insecure
```

3. Query the configuration of a namespace, for example, istio-system.

```
kubectl get namespace istio-system -o json > istio-system.json
```

4. Clear the array of the finalizers field and save the configuration.

```
    "spec": {
        "finalizers": [
        ]
    },
```

5. Run the following command to delete the finalizers field of the istio-system namespace:

```
curl -X PUT --data-binary @istio-system.json http://localhost:8001/api/v1/namespaces/
istio-system/finalize -H "Content-Type: application/json" --header "Authorization: Be
arer $TOKEN" --insecure
```

3.What can I do if the pods of a Kubernetes cluster on the data plane cannot access the IP address of the SLB instance that is configured in an ingress gateway?

This topic shows you how to resolve the issue where the pods of a Kubernetes cluster on the data plane cannot access the IP address of the Server Load Balancer (SLB) instance that is configured in an ingress gateway.

Problem description

A Kubernetes cluster is added to your Alibaba Cloud Service Mesh (ASM) instance. An SLB instance whose externalTrafficPolicy parameter is set to Local is configured in an ingress gateway for the ASM instance. When the pods of the Kubernetes cluster access the IP address of the SLB instance that is configured in the ingress gateway, the following issue occurs:

- The Pod on some specific nodes can access the SLB address exposed by the entry gateway.
- The Pod on some specific nodes can not access the SLB address exposed by the entry gateway.

Causes

If the SLB instance whose `externalTrafficPolicy` parameter is set to Local is specified for the ingress gateway service of the Kubernetes cluster, only the backend pods where the service is deployed can access the IP address of the SLB instance. This is because the IP address of the SLB instance is regarded as an external IP address of the service and is used to access the ingress gateway from outside the Kubernetes cluster. If the nodes and pods in the Kubernetes cluster cannot directly access the IP address of the SLB instance, the system does not route requests to the SLB instance. Instead, the requests are forwarded by kube-proxy in iptables or IP Virtual Server (IPVS) mode.

If no backend pods of the service are deployed on the nodes of the Kubernetes cluster or the nodes where the pods that send requests reside, the IP address of the SLB instance cannot be accessed. If the backend pods of the service are deployed, the IP address of the SLB instance can be accessed. For more information, see Why kube-proxy add external-lb's address to node local iptables rule?.

Solutions

- You can use the IP address of the Kubernetes cluster or the name of the ingress gateway service to access the IP address of the SLB instance within the Kubernetes cluster. The name of the ingress gateway service is istio-ingressgateway.istio-system.

  > **Note** We recommend that you use this solution.

- If you do not require source IP addresses, you can use the following solution:

  Change the value of the `externalTrafficPolicy` parameter of the ingress gateway to `Cluster`. In this case, you cannot obtain source IP addresses when you access the IP address of the SLB instance. For more information, see Modify an ingress gateway service.

```
apiVersion: istio.alibabacloud.com/v1beta1
kind: IstioGateway
metadata:
  name: ingressgateway
  namespace: istio-system
  ....
spec:
  externalTrafficPolicy: Cluster
....
```

- If you use elastic network interfaces (ENIs) of Terway or your clusters are in inclusive ENI mode, you can use the following solution: This solution allows you to access the IP address of the SLB instance within the Kubernetes cluster without losing source IP addresses.

  Change the value of the `externalTrafficPolicy` parameter of the ingress gateway to `Cluster` and add an annotation, such as `serviceAnnotations: service.beta.kubernetes.io/backend-type: "eni"`, to directly connect to ENIs. For more information, see Modify an ingress gateway service.

```
apiVersion: istio.alibabacloud.com/v1beta1
kind: IstioGateway
metadata:
  name: ingressgateway
  namespace: istio-system
  ....
spec:
  externalTrafficPolicy: Cluster
  maxReplicas: 5
  minReplicas: 2
  ports:
    - name: status-port
      port: 15020
      targetPort: 15020
    - name: http2
      port: 80
      targetPort: 80
    - name: https
      port: 443
      targetPort: 443
    - name: tls
      port: 15443
      targetPort: 15443
  replicaCount: 2
  resources:
    limits:
      cpu: '2'
      memory: 2G
    requests:
      cpu: 200m
      memory: 256Mi
  runAsRoot: false
  serviceAnnotations:
    service.beta.kubernetes.io/backend-type: eni
  serviceType: LoadBalancer
```

4.How can I retain the SLB instance configured for an ASM gateway when I delete the ASM gateway?

When you delete an Alibaba Cloud Service Mesh (ASM) gateway, the Server Load Balancer (SLB) instance configured for the ASM gateway is also deleted. This topic describes the cause of the issue and provides a solution.

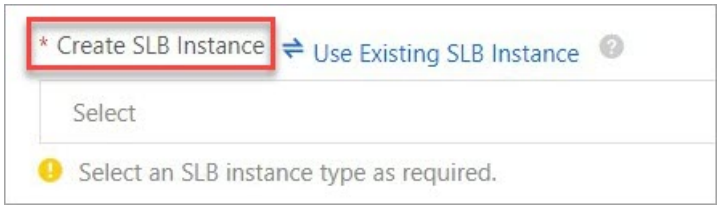
Problem description

When you delete an ASM gateway, the SLB instance configured for the ASM gateway is also deleted.

Cause

When you create an ASM gateway, if you specify **Create SLB Instance**, an SLB instance is automatically created. In this case, when you delete the ASM gateway, the SLB instance that is automatically created is also deleted.

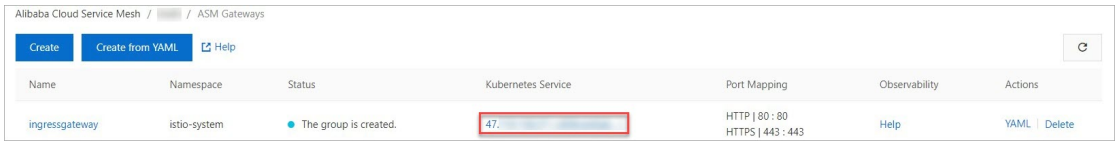
> **Note** If you specify **Use Existing SLB Instance** when you create an ASM gateway, the SLB instance that you use is retained after you delete the ASM gateway.



Solution

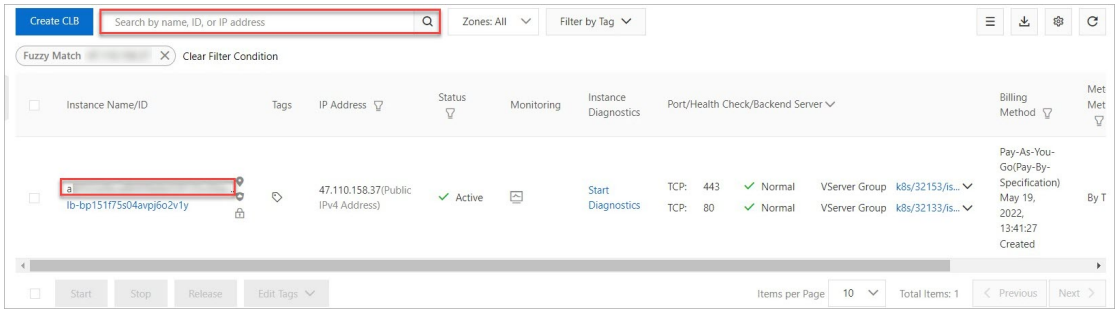
To ensure that the SLB instance that is automatically created for an ASM gateway is retained when you delete the ASM gateway, perform the following steps:

1. Obtain the IP address of the ASM gateway.
   i.
   ii.
   iii.
   iv.
   v. On the **ASM Gateways** page, obtain the IP address of the ASM gateway whose SLB instance you want to retain in the Kubernetes Service column.



2. Obtain the ID of the SLB instance.
   i. Log on to the SLB console.
   ii. In the left-side navigation pane, choose **CLB (FKA SLB) > Instances**.

iii. On the **Instances** page, enter the ID of the ASM gateway obtained in Step 1 in the search box. In the search results, you can obtain the ID of the SLB instance.



3. Modify the YAML file of the ASM gateway.

   i. On the **ASM Gateways** page, click **YAML** in the **Actions** column of the ASM gateway that you want to manage.

   ii. Add the following content to the `serviceAnnotations` parameter in the YAML file. Then, click **OK**.

   Replace `{YourSLBId}` with the ID of the SLB instance obtained in Step 2.

```
service.beta.kubernetes.io/alibaba-cloud-loadbalancer-id: {YourSLBId}
```

```
107   serviceAnnotations:
108     service.beta.kubernetes.io/alibaba-cloud-loadbalance
109     service.beta.kubernetes.io/alicloud-loadbalancer-add
110   serviceType: LoadBalancer
```

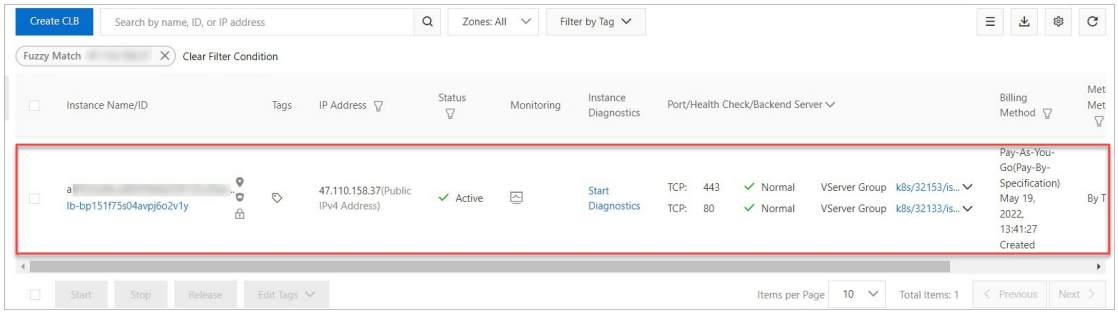
   After you modify the YAML file, the system automatically redeploys the ASM gateway. The ASM gateway enters the **Creating** state. If the status of the ASM gateway becomes **Running**, the ASM gateway is redeployed.

4. Delete the ASM gateway and check whether the SLB instance that is automatically created for the ASM gateway is also deleted.

   i. On the **ASM Gateways** page, find the AMS gateway that you want to delete and click **Delete** in the **Actions** column. In the message that appears, click **OK**.

   ii. Log on to the SLB console. On the **Instances** page, enter the ID of the ASM gateway obtained in Step 1 in the search box.

   If the SLB instance is displayed in the search results on the **Instances** page, the settings in Step 2 take effect. The following figure shows an example.



5.Why are no gateways or non-Istio gateways returned when I run the kubectl command to query gateways?

This topic describes the issue that no gateways or non-Istio gateways are returned when the kubectl command is run to query gateways. This topic also describes the cause of the issue and provides solutions.

Problem description

An Istio gateway is created. After you run the following command, the `No resources found` message is returned or non-Istio gateways are returned.

```
kubectl get gateway --all-namespaces
```

Cause

If you use Alibaba Cloud Service Mesh (ASM) instances whose Istio version is 1.8.6 or later, you may encounter the issue. This is because Kubernetes Gateway API is automatically installed in ASM instances whose Istio version is 1.8.6 or later. For more information, see Use Gateway API to define a routing rule.

Both Kubernetes Gateway API and Istio API provide a resource that is named Gateway. The two Gateway resources have similar features, but they are different resources. The system cannot identify which resource that you specify when you run `kubectl get gateway`. As a result, Kubernetes gateways instead of Istio gateways may be returned. In this case, if no Kubernetes gateways are defined, no gateways are returned. If Kubernetes gateways are defined, the Kubernetes gateways are returned.

Solutions

- Use the ASM console to view Istio gateways.
- Use a full resource name or recognizable abbreviation in the kubectl command.

  For example, you can use gtw to represent Kubernetes gateways, and gw to represent Istio gateways. In this case, you can run `kubectl get gw` or `kubectl get gateways.networking.istio.io` to make sure that Istio gateways can be returned.

6.Why does a destination rule not take effect after it is defined?

This topic describes the issue where a destination rule does not take effect after it is defined. This topic also describes the cause of the issue and provides a solution.

Problem description

When you use a client to call a service for which a destination rule is defined, the call fails because the destination rule does not take effect.

Cause

To route a request to a service, Alibaba Cloud Service Mesh (ASM) follows a specific process to search for the destination rule that is defined for the service. If the destination rule is not found in the namespaces that are searched based on the specific process, the destination rule does not take effect. ASM searches for the destination rule based on the following process:

1. ASM checks whether the destination rule exists in the namespace where the client that is used to call the service resides.
2. ASM checks whether the destination rule exists in the namespace where the service that is to be called resides.
3. ASM checks whether the destination rule exists in the root namespace that is named istio-system.

For example, the following YAML file is used to define a destination rule for the myservice service in the ns1 namespace. The `host` field indicates that the myservice service is defined in the default namespace.

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: myservice
spec:
  host: myservice.default.svc.cluster.local
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
```

- If you use a client that resides in the ns1 namespace to call the myservice service, the call is successful because ASM finds the destination rule of the myservice service in the ns1 namespace and uses the destination rule to route the call request.
- If you use a client that resides in the ns2 namespace to call the myservice service, the call fails. ASM searches for the destination rule based on the following process:
  i. The client that is used to call the myservice service resides in the ns2 namespace. ASM searches for the destination rule in the ns2 namespace. However, the destination rule is not found because it does not exist in the ns2 namespace.
  ii. The myservice service to be called resides in the default namespace. ASM searches for the destination rule in the default namespace. However, the destination rule is not found because it does not exist in the default namespace.

iii. The root namespace of ASM is fixed as istio-system. ASM searches for the destination rule in the istio-system namespace. However, the destination rule is not found because it does not exist in the istio-system namespace.

Solution

When you define a destination rule for a service, define the destination rule in one of the following namespaces:

- The root namespace of ASM
- The namespace where the service resides
- The namespace where the client that is used to call the service resides

> **Note** You can use virtual services across namespace boundaries. By default, a virtual service is visible to all namespaces regardless of the namespace where the virtual service is defined. However, you can modify the `exportTo` parameter to change the default setting in the YMAL file that is used to define the virtual service.

7.Why is a pod in the init crash state after a sidecar proxy is injected into the pod?

This topic describes the issue in which a pod is in the `init crash` state after a sidecar proxy is injected into the pod. This topic also describes the cause of the issue and provides a solution.

Problem description

After you run the following command to check the status of pods, you find that a pod into which a sidecar proxy is injected is in the `init crash` state:

```
kubectl get pod
```

The system displays information similar to the following output:

```
NAME                 READY       STATUS                  RESTARTS        AGE
details-v1-u****     0/2         Init:Error              1               12h
productpage-n****    0/2         Init:CrashLoopBackOff   3               12h
```

Then, you run the following command to check the logs of the istio-init container:

```
kubectl --kubeconfig=${USER_KUBECONFIG} -c istio-init logs ${pod}
```

The system displays information similar to the following output:

```
......
......
-A ISTIO_OUTPUT -d 127.0.**.**/32 -j RETURN
-A ISTIO_OUTPUT -d 192.168.0.1/32 -j RETURN
-A ISTIO_OUTPUT -j ISTIO_REDIRECT
COMMIT
2022-03-23T06:42:21.179567Z     info    Running command: iptables-restore --noflush /tmp/ipt
ables-rules-1648017741179373856.txt4205119933
2022-03-23T06:42:21.185698Z     error   Command error output: xtables other problem: line 2
failed
2022-03-23T06:42:21.185720Z     error   Failed to execute: iptables-restore --noflush /tmp/
iptables-rules-1648017741179373856.txt4205119933, exit status 1
```

The `Failed to execute: iptables-restore` error message is recorded in the logs of the istio-init container.

Cause

Check whether you have cleaned up the istio-init container after you exit the istio-init container by running a command such as `docker container rm/docker container prune/docker system prune`, or whether a scheduled task is executed to clean up the istio-init container.

If you clean up the istio-init container after you exit the istio-init container, Kubernetes detects that the istio-init container associated with the pod is removed. In this case, Kubernetes restarts the removed container. However, the newly started istio-init container cannot execute a new iptables rule because an iptables rule has been created before. As a result, no iptables rule can be configured for the newly started istio-init container, and the istio-init container crashes.

Solution

To resolve the issue, recreate the pod. After the pod is recreated, the pod recovers to the normal state.

If you need to run a command or scheduled task to clean up data, take note of the following items:

- If you run a command to batch clean up data, you must filter out the istio-init container in the command to prevent the istio-init container from being cleaned up.

  ```
  docker system prune --filter "label!=io.kubernetes.container.name=istio-init"
  ```

- If you run a scheduled task to clean up data, you must replace the `docker system prune` command with the following command in the script of the scheduled task to filter out the istio-init container. This prevents the istio-init container from being cleaned up.

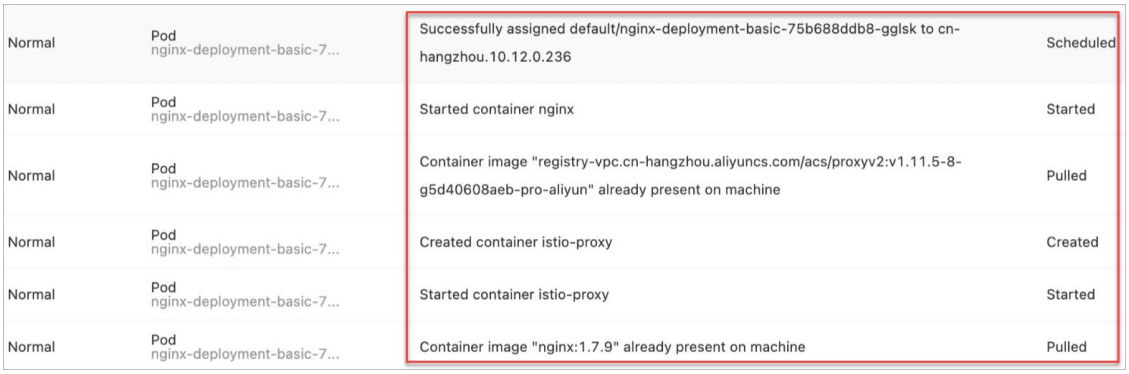
  ```
  docker system prune --filter "label!=io.kubernetes.container.name=istio-init"
  ```

8.Why is no valid health check information displayed after sidecar injection?

This topic describes the issue in which no valid health check information is displayed after sidecar injection. This topic also describes the cause of the issue and provides a solution.

Problem description

No valid health check information is displayed after sidecar injection. In this example, port 8087 is used for TCP health checks. After you enable mutual Transport Layer Security (mTLS), no health check information of port 8087 is displayed on the **Events** tab of the details page of a pod in the Container Service for Kubernetes console.



Normal	Pod nginx-deployment-basic-7...	Successfully assigned default/nginx-deployment-basic-75b688ddb8-gglsk to cn-hangzhou.10.12.0.236	Scheduled
Normal	Pod nginx-deployment-basic-7...	Started container nginx	Started
Normal	Pod nginx-deployment-basic-7...	Container image "registry-vpc.cn-hangzhou.aliyuncs.com/acs/proxyv2:v1.11.5-8-g5d40608aeb-pro-aliyun" already present on machine	Pulled
Normal	Pod nginx-deployment-basic-7...	Created container istio-proxy	Created
Normal	Pod nginx-deployment-basic-7...	Started container istio-proxy	Started
Normal	Pod nginx-deployment-basic-7...	Container image "nginx:1.7.9" already present on machine	Pulled

Cause

After you enable mTLS in Alibaba Cloud Service Mesh (ASM), the requests for health checks sent by the kubelet to the pod are intercepted by the sidecar proxy. If the kubelet cannot provide the required TLS certificate, the health checks fail.

Solution

You can configure settings to allow the traffic of health checks to bypass the sidecar proxy. Perform the following steps:

Allow the traffic of health checks to bypass the sidecar proxy

1.
2.
3.
4.
5. On the **Namespace** tab, select the namespace that you want to manage, click **enable/disable Sidecar proxy by port or address**, and then set the required parameters.

   The following table describes the parameters.

Parameter	Description
Set the port numbers to prevent InboundTraffic from passing through the sidecar proxy	The port on which you want to allow the inbound traffic to bypass the sidecar proxy. In this example, port *8087* is used.
Set the port numbers to prevent OutboundTraffic from passing through the sidecar proxy	The port on which you want to allow the outbound traffic to bypass the sidecar proxy. In this example, port *8087* is used.

6. Click **Update Settings**.

View health check results

1.
2.
3.
4.
5. Click the name of the pod whose details you want to view to go to the details page of the pod. Alternatively, you can click Details in the Actions column that corresponds to the pod.
6. On the details page of the pod, click the **Events** tab.

   The following figure shows the health check results of port 8087.