

Using VMware Tanzu Service Mesh

VMware Tanzu Service Mesh

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2023 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

- 1 About Using VMware Tanzu Service Mesh 5**
- 2 Support of Istio in Tanzu Service Mesh 6**
- 3 Connect Services Across Clusters with a Global Namespace 8**
 - View the Health Status and Details for a Global Namespace 11
- 4 Manage Services with a Service Group 14**
- 5 Create an External Service 18**
 - External Service Wildcard Support 23
 - Edit an External Service Configuration 25
 - Monitor the Performance of External Services 26
 - Add a Custom Certificate 30
- 6 Create a Public Service 31**
 - Configure Global Load Balancing for Your Application in Tanzu Service Mesh 39
 - Monitor a Public Service 43
 - Edit the Configuration of a Public Service 48
- 7 Headless Service with StatefulSet 50**
- 8 Manage the Tanzu Service Mesh Components with Tanzu Service Mesh Lifecycle Manager Operator 54**
 - Manage Tanzu Service Mesh Updates 55
- 9 View Details of a Tanzu Mission Control Managed Cluster 59**
- 10 View the Topology of Services in a Global Namespace or a Cluster 61**
- 11 View the Proxy Configuration Settings 67**
- 12 Configure the Export of API Audit Logs to Splunk 69**
- 13 GitOps Workflow with Tanzu Service Mesh 74**
 - Integrating Tanzu Service Mesh into a GitOps workflow 74
 - Keeping Git as the Source of Truth for Configuration 75
 - Managing Declarative Manifests 76

14 Tanzu Service Mesh CLI 77

Common CLI Tasks 78

Install the CLI 78

Log in to the Tanzu Service Mesh CLI 79

Get a List of CLI Commands and Get Help on a Command 80

Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification 80

Use an Existing Object or Policy Configuration to Create a Manifest 84

Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI 86

Onboard a Cluster Using the CLI 87

Remove a Cluster from Tanzu Service Mesh Using the CLI 92

Create a Global Namespace 93

Create an Access Control Policy 97

Create a Public Service Using the CLI 98

Add a Certificate to Tanzu Service Mesh Using the CLI 108

Debug Problems with a Manifest File 110

Delete an Object or a Policy from Tanzu Service Mesh SaaS Using the CLI 110

15 Tanzu Service Mesh Administration 112

Manage Integrations 112

Create an AWS Integration Account 112

Create an Avi Integration Account 115

Create a Venafi Integration Account 118

Create a Vault CA Integration Account 121

Create a GitHub Integration Account 128

Manage Domains 130

Manage Keys and Certificates 131

Add Certificates 131

Add Certificate Chains 133

About Using VMware Tanzu Service Mesh

1

The *Using VMware Tanzu Service Mesh* documentation provides information about using VMware Tanzu® Service Mesh™.

This documentation complements [Getting Started with Tanzu Service Mesh](#) and contains information about more advanced tasks, including advanced configuration and administrative tasks.

Intended Audience

This information is intended for the following audiences:

- DevOps engineers, site reliability engineers (SREs), and platform operators who want to use Tanzu Service Mesh to deploy and manage cloud-native microservices applications across clusters, clouds, and platforms.
- Application developers and service owners who want to use Tanzu Service Mesh for application development.
- Security engineers and SecOps engineers who want to use Tanzu Service Mesh to configure user- and data-centric security policies for their microservices applications and benefit from security analytics and compliance capabilities of the product.

The information is written for platform operators, application developers, and security engineers who have a basic understanding of Kubernetes and are familiar with container deployment concepts and service mesh concepts.

Support of Istio in Tanzu Service Mesh

2

Tanzu Service Mesh is a managed service provided by VMware that delivers the open-source Istio technology to client clusters. Tanzu Service Mesh provides an Istio-based solution that is curated by VMware and tested for compatibility with its own features.

Tanzu Service Mesh provides customers with a streamlined and secure way to manage their Istio deployments across multiple clouds and vendors.

Istio is an open-source service mesh that provides features such as traffic management, policy enforcement, and telemetry collection for microservices-based applications. Tanzu Service Mesh takes the core capabilities of Istio and adds additional value on top of it, including:

- Global namespaces. Tanzu Service Mesh provides a unified management interface for Istio deployments across multiple clusters, allowing customers to manage their Istio environment in a centralized manner.
- Advanced security and resiliency features. Tanzu Service Mesh includes features, such as API security, PII DLP, east/west WAF, and access policies, that are designed to enhance the security and resiliency of customer applications.
 - Management capabilities. Tanzu Service Mesh provides several management capabilities to help customers manage their Istio deployments, including Istio Lifecycle Management. Customers can install, upgrade, and manage their Istio deployments centrally through Tanzu Service Mesh.
 - Cluster dashboards. Tanzu Service Mesh provides a view of the topology of applications on the clusters and of how they interact with the Istio components.
 - Inventory, service maps, and performance graphs. Tanzu Service Mesh provides platform operators with the ability to monitor and manage their Istio deployments across clouds.

When customers onboard their clusters to Tanzu Service Mesh, Tanzu Service Mesh will deploy OSS Istio to the customer's client clusters on any cloud and vendor. The supported Kubernetes platforms are listed on the [Tanzu Service Mesh Environment Requirements and Supported Platforms page](#). Once Istio is deployed in the client clusters, customers can interact with Istio

through the Kubernetes API and Istio CRDs, as they would with the open-source version of Istio. However, Tanzu Service Mesh features, such as global namespaces and policies, are managed centrally through the Tanzu Service Mesh SaaS (UI, Tanzu Service Mesh CLI, or REST API). The demarcation line between Istio and Tanzu Service Mesh features is as follows:

- Istio single-cluster features. These are managed through the Kubernetes cluster API and Istio CRDs.
- Tanzu Service Mesh features: These are managed centrally through the Tanzu Service Mesh SaaS through the Tanzu Service Mesh CLI, UI, or REST API.

When interacting with Istio through Tanzu Service Mesh, consider several things:

- Tanzu Service Mesh supports Istio deployments managed through the Tanzu Service Mesh lifecycle manager only.

Note Attaching an Istio control plane that was not deployed by Tanzu Service Mesh is currently not supported.

- Tanzu Service Mesh does not support Istio multiclustering with global namespaces. Customers who want to use global namespaces should do so for federated multiclustering.
- Customer gateways and configurations to Istio policies must be made in the app namespace because Tanzu Service Mesh will overwrite any changes made to the istio-system namespace.
- When using global namespaces, you may need to update some Istio policies or move them a global namespace to avoid conflicts. Make sure that the policies do not overlap or collide with Tanzu Service Mesh access policies.

In conclusion, Tanzu Service Mesh is a managed service that provides customers with a streamlined and secure way to manage their Istio deployments. Tanzu Service Mesh adds additional value to Istio through features, such as global namespaces and advanced security and resiliency capabilities, while providing management capabilities to help customers manage their Istio deployments. To get the most out of Tanzu Service Mesh, customers should consider the points mentioned above and follow the Tanzu Service Mesh Environment Requirements and Supported Platforms page for deploying Istio on their client clusters.

Connect Services Across Clusters with a Global Namespace

3

With global namespaces in Tanzu Service Mesh, you can easily connect and secure the services in your application across clusters. You can learn how to add the services in your application to a global namespace to have them automatically discovered and connected across the clusters. A global namespace can be shared across a single cluster, multiple clusters, or even clusters in different clouds.

Where appropriate, an example of a sample e-commerce application is used to show you how to connect services across clusters by adding them to a [global namespace](#). The sample application is made up of 12 services and is configured to have most of the services deployed on one cluster and the catalog service deployed on the other cluster.

Prerequisites

Verify the following prerequisites:

- [Onboard the clusters where your services are deployed to Tanzu Service Mesh.](#)
- Know the Kubernetes namespaces in your clusters that hold the services of your application.
- [Access the Tanzu Service Mesh Console](#)
- Learn more about Tanzu Service Mesh and global namespace in [About Tanzu Service Mesh Concepts](#).

Procedure

- 1 In the Tanzu Service Mesh Console, create a global namespace for your application services:
 - a In the navigation panel on the left, click **Inventory** and then click **Global Namespaces**.
 - b On the **Global Namespaces** page, click **New Global Namespace**.
 - c On the **General Details** page of the **New Global Namespace** wizard, enter a unique name and a domain name for the global namespace.

The name of a global namespace and its domain name together forms a fully qualified domain name (FQDN) that uniquely identifies that global namespace and makes it possible for the services in the global namespace to communicate with each other across clusters.

In the example, you must enter a name of *sample-gns* and a domain name of *sample.app.com*.

- d On the **Namespace Mapping** page, to add the services in your application to the global namespace, specify their Kubernetes namespace-cluster pairs. Under **Namespace Mapping Rule**, in the left drop-down menu, select the namespace on one of your clusters that holds some of the services and in the right drop-down menu, select the name of the cluster. Click **Add Mapping Rule** to create multiple namespace mapping rules for the same or different clusters.

Important Kubernetes namespaces with different names in one or more clusters can be mapped into a single global namespace. However, you cannot assign a specific namespace in a cluster to more than one global namespace at the same time. Multiple namespace supports public service and cross cluster routing; however, traffic management, SLOs and autoscaling are not supported. For more information on multitenant support in a single global namespace, see [Global Namespaces](#).

The namespace-cluster pairs you specify here define *namespace mapping rules* that are used to select services for a global namespace. Click **Service Preview** under each namespace-mapping rule to see the names of the selected services from each cluster.

The sample application has services running on two clusters. For most of the services running in one cluster, select the **default** namespace in the left drop-down menu and the *prod-cluster1* cluster in the right drop-down menu. Then click **Add Namespace Mapping** and select the **default** namespace and the *prod-cluster2* cluster for the catalog service on the other cluster.

- e (Optional) On the **External Services** page, configure external services in the global namespace.

For more information about creating external services, see [Create an External Service](#).

- f (Optional) On the **Public Services** page, select one of the following options.
 - To configure GSLB-enabled or non-GSLB public services in the global namespace, click **Configure Public Service/s** and click **Next**. For an explanation of GSLB-enabled and non-GSLB public services and information about configuring public services, see [Chapter 6 Create a Public Service](#).
 - To create a global namespace without public services, click **No Public Services** and click **Next**.

- g On the **Summary** page, review the configuration of the global namespace and click **Finish**.

- 2 (Optional) To enable the cross-cluster communication between the services, edit the Kubernetes deployment manifest for the appropriate service on one cluster to specify the domain name of the global namespace, prefixing the domain name with the name of the service on the other cluster.

Important Make sure that you prefix the domain name with the name of the service that you want the service being edited to communicate with. See the following example.

In the sample application, the shopping service on one cluster must communicate with the catalog service on the other cluster. To edit the deployment manifest of the shopping service, run the following `kubectl` command.

```
kubectl --context=prod-cluster1.local edit deployment shopping
```

In the deployment manifest, set the appropriate variable to **catalog.sample.app.com**. The `catalog` prefix is required for the shopping service to communicate with the catalog service.

Important If you are using your custom application instead of the sample application, make sure to use the `appProtocol` field to define your port in the Kubernetes service manifest for your application. This is needed for the services running on one cluster to communicate with the services running on the other clusters.

The following example of a service manifest shows `appProtocol` for HTTP under `ports`.

```
apiVersion: v1
kind: Service
metadata:
  name: order-service
spec:
  ports:
    - appProtocol: http
      number: 3000
```

3 Verify the cross-cluster communication between the services in Tanzu Service Mesh.

- a On the navigation pane on the left, click **Inventory** and then **Global Namespaces**.
- b On the **Global Namespaces** page, click the name of the global namespace that you created (*sample-gns* in the example).

The global namespace details page displays the summary information about the global namespace, including its overall health state. The global namespace can have of the following health statuses:

- **Healthy.** The configuration of the global namespace is synced and applied to the clusters that make up the global namespace. There is connectivity between the Tanzu Service Mesh SaaS and the clusters in the global namespace.
- **Syncing.** A temporary status. The configuration of the global namespace is being synced to the clusters that are in the global namespace. When the synchronization is complete, the status will change to Healthy or Error.
- **Error.** There was a problem syncing the configuration of the global namespace to the clusters. For example, the global namespace is incorrectly configured, or something is missing from the configuration. To resolve the problem, open a support request with VMware.

- c Click the **GNS Topology** tab.

The service topology graph shows the connections between the services in the different clusters. The line between the services indicates that traffic flows between them. The number of requests per seconds (RPS) or other specified service metrics are shown.

What to do next

For information about how to specify metrics to show in the service topology graph and other details about using the topology graph, see [View the Summary Infrastructure and Service Information](#).

View the Health Status and Details for a Global Namespace

You can view the overall health status of a global namespace to monitor the health of the services in the global namespace. You can also view the different configuration details and the performance metrics for its services.

You can view the overall health status of a global namespace, the performance metrics for its services, and the different details about the global namespace on the global namespace details page.

The top of the page displays the health status of the global namespace. A global namespace can have of the following health statuses:

- **Healthy.** The configuration of the global namespace is synced and applied to the clusters that make up the global namespace. There is connectivity between the Tanzu Service Mesh SaaS and the clusters in the global namespace.
- **Syncing.** A temporary status. The configuration of the global namespace is being synced to the clusters that are in the global namespace. When the synchronization is complete, the status changes to **Healthy** or **Error**.
- **Error.** There was a problem syncing the configuration of the global namespace to the clusters. For example, the global namespace is incorrectly configured, or something is missing from the configuration. To resolve the problem, open a support request with VMware.

The top of the page also displays the aggregated metrics for the services in the global namespace and its general details. You can also edit the configuration of the global namespace from this page by clicking **Edit Configuration** in the upper-right corner. You can delete the global namespace by clicking **More** and then **Delete** in the upper-right corner. For more information about the general global namespace details, click **Support** on the right side of the page and then click **View the health status and summary metrics for a global namespace** in the Support panel.

The following tabs on the page display the different kinds of information for the global namespace:

- **GNS Topology.** View a graph showing the topology of the services in the global namespace and the performance metrics for the services. A rectangle is displayed for each cluster in the global namespace. For more information about the **GNS Topology** tab, click **Support** on the right side of the page and then click **View a topology graph of the services in a global namespace** in the Support panel.
- **Performance.** View the performance metric charts for the services in the global namespace. For information about customizing the charts to display the metrics you want and customizing the time range in the charts, click **Support** on the right side of the page and then click **Customize the metric charts** and **Customize the metric charts** in the Support panel.
- **Public Services.** View the details about the public services configured in the global namespace. To view the details page for a public service, click its name in the **Public Service** column.
- **Services.** View the details about the services in the global namespace. For more information about the details that are displayed, click **Support** on the right side of the page and then click **View the services in a global namespace** in the Support panel.
- **Infrastructure.** View the infrastructure details about the services in the global namespace, including the clusters and nodes that contain the services and the infrastructure metrics. For more information about the details that are displayed, click **Support** on the right side of the page and then click

View the clusters and nodes for the services in a global namespace in the Support panel.

- **Configuration.** View the different sections that make up the configuration of the global namespace. To edit the details you want in the **Edit Global Namespace** window, click **Edit** next to the name of the appropriate configuration section. For example, to edit the service mappings of the global namespace, click **Edit** next to **Service Mappings**.

Prerequisites

- [Chapter 3 Connect Services Across Clusters with a Global Namespace.](#)
- [Access the Tanzu Service Mesh Console.](#)

Procedure

- 1 On the Home page, on the **GNS Overview** tab, click the name of the global namespace in its card.
- 2 On the global namespace details page, click the tab for the kind of information you want to view about the global namespace.

Manage Services with a Service Group

4

You can create a service group to manage a group of services collectively. You can observe aggregated metrics for the services in the group or consistently enforce policies across the service group.

A service group is a type of [resource groups](#). A service group is a collection of [services](#) that share certain characteristics. A service group defines one or more conditions that a service must satisfy to be included in that group. For example, a service group can define a condition for the service name to begin with *shopping*.

Service groups serve two main purposes:

- You can create a service group to monitor relevant metrics (such as requests per seconds, latency, and error rate) for the services in the group collectively.
- You can also define and apply consistent policies to the entire service group.

You can use a sample service group to create a service group in Tanzu Service Mesh. The service group is used to collectively manage all cart services deployed in clusters located in California. The sample service group defines conditions for the service name to begin with *cart*, for the cluster name to be *prod-cluster-ca*, and for the services to be in the default or sample namespace in the cluster.

You can access and observe metrics for the services in a service group.

Prerequisites

- Verify that you have onboarded the [clusters](#) where your services are deployed. For more information about onboarding a cluster, see [Onboard a Cluster to Tanzu Service Mesh](#).
- [Access the Tanzu Service Mesh Console](#).

Procedure

- 1 In Tanzu Service Mesh Console, in the navigation panel on the left, click **Inventory > Service Groups**.

The **Service Groups** tab displays a list of existing service groups.

- 2 If you want to view information about a service group, perform these steps.
 - a To view the membership conditions defined for a service group, click the arrow to the left of the service group name.
 - b To view details about the service group on the service group details page, click its name on the list.

- 3 In the upper-right corner, click **New Service Group** and provide the required information.


Property	Description
Group Name	Enter a descriptive name for the group to help distinguish it from other service groups.
Description	Optionally, provide a description for the group.

- 4 Define conditions for membership in the group.
 - a Under **Membership Conditions**, click **Add Condition** and select a service attribute or metric for which you want to define a condition (for example, **Service Name** or **Namespace**).
 - b In the new condition row, select an operator (for example, **Is Exactly** or **Starts with**) in the second drop-down menu and select or enter a value in the third drop-down menu.
 - c To define additional conditions, repeat steps a–b.

If you have multiple conditions, by default the conditions are joined with the **AND** operator. A service is matched if all the conditions are true.

- d To join the conditions with the **OR** operator to specify that any of the conditions can be true for a match, click the **AND** button on the left.

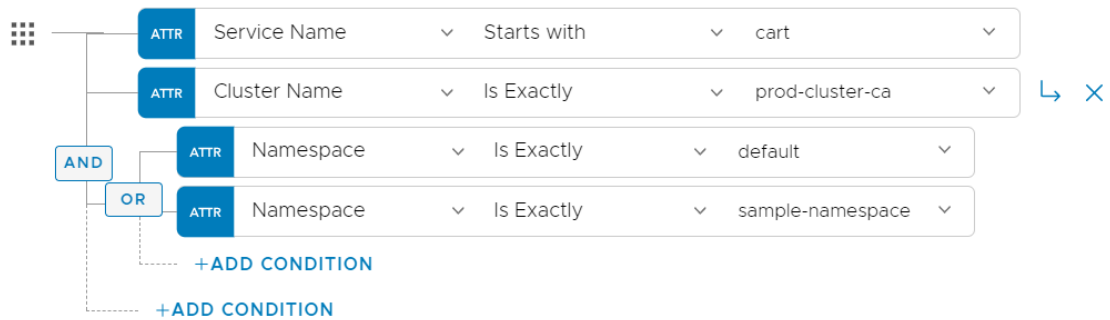
The label on the button changes to **OR** to indicate that the conditions are joined with the **OR** operator.

- e To combine two or more conditions in a complex condition, point to one of the condition rows that you want to combine and click  at the right end of the row and click **Add Condition** to add another condition to combine.

For the example, the following conditions for membership are defined in the service group. The conditions include a complex condition for the default and sample-namespace namespaces where the conditions are joined with **OR**.

Membership Conditions

Include a service in the group if the following is **TRUE**:



According to these conditions, a service is included in the service group if it meets these requirements:

- If the name of a service starts with cart.
 - The service is located in the prod-cluster-ca cluster.
 - The service is contained in the default namespace or the sample-namespace namespace.
- 5 To view a list of the services that satisfy the membership conditions, click **Membership Preview**.
 - 6 Click **Save**.

7 View the metrics for the services in a service group.

- a On the **Service Groups** tab, click the name of the service group.

At the top of the service group details page, next to the service group name, a summary of aggregated key metrics, such as requests per second, 99th percentile latency, and percentage of errors, is displayed for the group.

- b Click the **Performance** tab.

The charts on the **Performance** tab show metrics, such as requests per seconds, 99th percentile latency and error percentage, aggregated for the services in the group.

The metrics are shown for the time period selected in the **Metric Time Range** drop-down menu in the upper-right corner.

- c To specify which metrics to display, click **Chart Settings** and select the check box next to each metric to display.

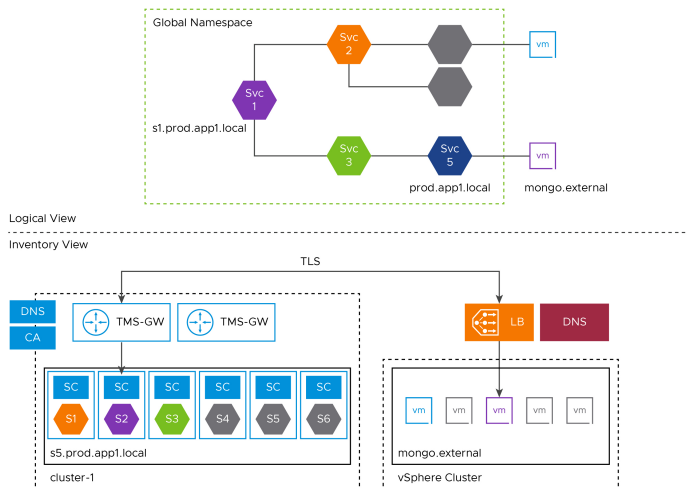
You can select a maximum of four metrics to be displayed.

Create an External Service

5

External services are services that exist outside the VMware Tanzu Service Mesh (for example, third-party database services) but are made accessible by services within a global namespace of the VMware Tanzu Service Mesh. Services can run on virtual machines, external Kubernetes clusters, Tanzu Application Service environments, lambda functions or even on bare metal, and can be accessed over TCP, TLS, HTTP, or HTTPS.

As an example, a service in a global namespace of Tanzu Service Mesh requires access to an external database or web service that is running outside the mesh. When the internal service attempts to communicate with the external service, the communication flows through the envoy proxy as a sidecar, which allows all traffic by default. Tanzu Service Mesh, on the other hand, defines service-layer policies, and these policies are transmitted to the envoy proxy, which allows or denies egress traffic as necessary. Tanzu Service Mesh also automates the creation of routing rules, destination rules, and gateway configuration for these external services.



To connect to an external service in Tanzu Service Mesh, you need to describe its configuration in the global namespace. Currently, the external services can be accessed over protocols such as HTTP, HTTPS, TCP, and TLS.

Note The following steps explain how to configure an external service from the Tanzu Service Mesh UI:

Prerequisites

- Create a global namespace to which you can add the external service. For information about creating a global namespace and adding services to it, see [Connect Services Across Clusters with a Global Namespace](#).
- If you want to use an HTTPS or TLS protocol to connect to the external service, add a TLS certificate. The certificate must match the domain you specify for the external service. This certificate will be used to encrypt incoming and outgoing traffic for the service. Private Keys are optional, and only Public Keys are mandatory when adding a certificate to External Services. For more information, see [Add Certificates](#).

Note Adding a TLS certificate is not mandatory. For external sites like google.com and amazon.com that use HTTPS, public certificates are already trusted by the client, so it is not necessary to add certificates. For

Procedure

- 1 On the **Home** page, on the **GNS Overview** tab, in the card for the global namespace that contains the external service, click the name or topology of the global namespace.
- 2 Click **Edit Configuration**, go to the **External Services** page by clicking **Next** on the **General Details** page and then on the **Namespace Mapping** page.
- 3 On the **External Services** page, click **Add External Service(s)** and perform these steps:
 - a In **External Service Name**, enter a unique name for the external service you are connecting to so that it can be identified. Once created, the external service name cannot be edited.
 - b In **Service Port**, specify the port through which the service can be accessed.
 - c Next to **External URL**, specify the parts of the URL at which the service will be accessible: the protocol, domain, and port number {protocol + FQDN hostname + monolith port number}.

Attention When using HTTPS or TLS protocols, the Service port and the Port for the External URL shouldn't be the same.

- d (Optional) If you specify **HTTPS** or **TLS** in **Protocol**, you can select the name of the certificate that you want to use for the service. You can select from the certificates that your administrator has defined. For more information, see [Add a Custom Certificate](#).

Important

- For a TLS or HTTPS external service, the service port must not match any of the endpoints or external URL ports.
- A service inside a Tanzu Service Mesh global namespace can only access TLS external service via TCP (even though the external server is using TLS).
- A service inside a Tanzu Service Mesh global namespace can only access HTTPS external Service through HTTP protocol (even though the external server is HTTPS).

Table 5-1. Custom TLS Version Support

TLS version	TLS version option
TLS version 1.0	TLSV1_0
TLS version 1.1	TLSV1_1
TLS version 1.2	TLSV1_2
TLS version 1.3	TLSV1_3

TLS versions 1.2 and 1.3 are enabled by default. Envoy filter can be edited to enable the other two TLS versions if desired.

- e (Optional) To define an internal URL to access the external service, click **Connect using an alias** and enter the domain name. The host name would be used if there is no alias provided.

Note It is important that the alias name is a Fully Qualified Domain Name (FQDN).

- f (Optional) In **Gateway Addresses**, select **Connect using gateway(s)**, click **Add Endpoint**, and provide the endpoint (IP address or domain name) if you wish to connect the external service using a gateway to manage the egress traffic.

Attention Multiple Endpoint Support

You can specify **multiple endpoints** and **load balance** between them. Tanzu Service Mesh supports load-balancing for multiple endpoints in all four protocols, including HTTPS, HTTP, TLS, and TCP.

Considerations:

- Make sure you configure endpoints of the same protocol only, for example, you shouldn't configure HTTP and HTTPS endpoints under the same external service.
- You can either create a new external service with multiple endpoints or edit an existing external service with an extra endpoint.
- Run the same server on multiple endpoints.

Configuration:

In **Gateway Addresses** section, configure multiple external service endpoints.

Once you enter multiple endpoints under gateway addresses in the UI, round-robin will be set as the default load-balancing scheme. You do not need to configure global load balancing (GSLB) for the external services. It is possible to check the traffic being load balanced between two servers as follows:

```
root@catalog-589b54f8d7-sdp8q:/app# curl http://www.demohttps.com:25000
Thread-2 From HTTPS Server
root@catalog-589b54f8d7-sdp8q:/app# while true
do
curl http://www.demohttps.com:25000
sleep .1
done
Thread-4 From HTTPS Server
Thread-5 From HTTPS Server
Thread-6 From HTTPS Server
Thread-7 From HTTPS Server
Thread-8 From HTTPS Server
Thread-9 From HTTPS Server
Thread-10 From HTTPS Server
Thread-1 From HTTPS Server- 2
Thread-11 From HTTPS Server
Thread-2 From HTTPS Server- 2
Thread-3 From HTTPS Server- 2
Thread-4 From HTTPS Server- 2
Thread-5 From HTTPS Server- 2
Thread-12 From HTTPS Server
Thread-13 From HTTPS Server
Thread-6 From HTTPS Server- 2
Thread-7 From HTTPS Server- 2
Thread-14 From HTTPS Server
Thread-15 From HTTPS Server
Thread-8 From HTTPS Server- 2
Thread-9 From HTTPS Server- 2
Thread-16 From HTTPS Server
Thread-17 From HTTPS Server
Thread-18 From HTTPS Server
Thread-10 From HTTPS Server- 2
Thread-19 From HTTPS Server
Thread-11 From HTTPS Server- 2
Thread-20 From HTTPS Server
Thread-12 From HTTPS Server- 2
Thread-13 From HTTPS Server- 2
```

Warning For multiple service endpoint configurations, the service port and the gateway port must be different.

- g To configure additional external service in the global namespace, click **Add External Service** and repeat steps a–f.

- 4 Click **Next**.
- 5 To save the external services you have configured in the global namespace, click **Finish**.

The **External Services** page displays the names of the configured external services, external URL at which each service is accessible, and other details. To access this page:

- a On the **Home** page, click the **GNS Overview** tab.
- b In the card for the global namespace in which the external service is configured, click the name of the global namespace.
- c Select the **External Services** tab.
- d (Optional) To view the names of the configured external services, the URL and port at which each service is accessible, click **Configuration** tab on the same page.

Results

The external service has been configured successfully. To edit the configuration of an external service or configure additional external services in the global namespace, perform steps 1–5 of this procedure.

Note With future enhancements, users will be able to define traffic management and access control policies for external services.

What to do next

For information about editing the configuration of an external service, see [Edit an External Service Configuration](#).

For information about how to monitor the performance of external services with metric charts and view its details, see [Monitor the Performance of External Services](#).

External Service Wildcard Support

The external service wildcard support in the Tanzu Service Mesh Global Namespace allows services inside Tanzu Service Mesh global namespace to connect to external servers whose hostnames are in wildcard format (e.g. *.google.com, *.wikipedia.com). With wildcard, you can choose exactly which servers to connect to among the set of wildcard servers, while the external service end points option load balances between external service endpoints.

Configuration Rules

- If we configure HTTPS/TLS wildcard external service, the service port should not match any of the endpoint ports or External URL port.
- From a service inside a Tanzu Service Mesh global namespace, an HTTPS wildcard external service can be accessed only through HTTP protocol (even though the External Server is HTTPS).
- Access to TLS wildcard external service is only possible through TCP protocol from a service in a global namespace (even though the External Server is TLS).
- Currently, Tanzu Service Mesh offers the option to use wildcards to match subdomains of external service hostnames.

```
Example hostname : www.wikipedia.org
www - subdomain
wikipedia - second level domain
org - top level domain
```

- In order for a wildcard external service to work, there must be a live www. subdomain server in the list of external servers.

```
Example wildcard server :
www.google.com -----> live www subdomain
translate.google.com
carrers.google.com
```

Configure Wildcard External Service

- 1 Go to the global namespace **External Services** configuration page.
- 2 Specify the subdomain of external service to * to represent the wildcard.
- 3 You may configure certificates (optional) for wildcard external servers if they are HTTPS/TLS servers. Choose the appropriate certificate from the certificate options, or upload a new certificate.

1 General Details

2 Namespace Mapping

3 External Services

4 Public Services

5 GILB & Resiliency

6 Configuration Summary

3. External Services (Optional)

Configure connectivity to external services for this DNS, e.g. web services, or VM/bare-metal servers.

☐ No External Services

☒ Add External Service/s

External Service Name *

googlewec

Service Port *

80

External URL *

https://*google.com:443

Webboard Subdomain Hostname

Certificate

Select certificate

+ NEW CERTIFICATE

Select a Certificate

Gateway Addresses (optional)

☐ Connect using an alias...

☐ Connect using gateway(s)...

+ ADD EXTERNAL SERVICE

CANCEL

BACK

NEXT

Note You can see that alias names and endpoints have been blocked because this will create access conflicts.

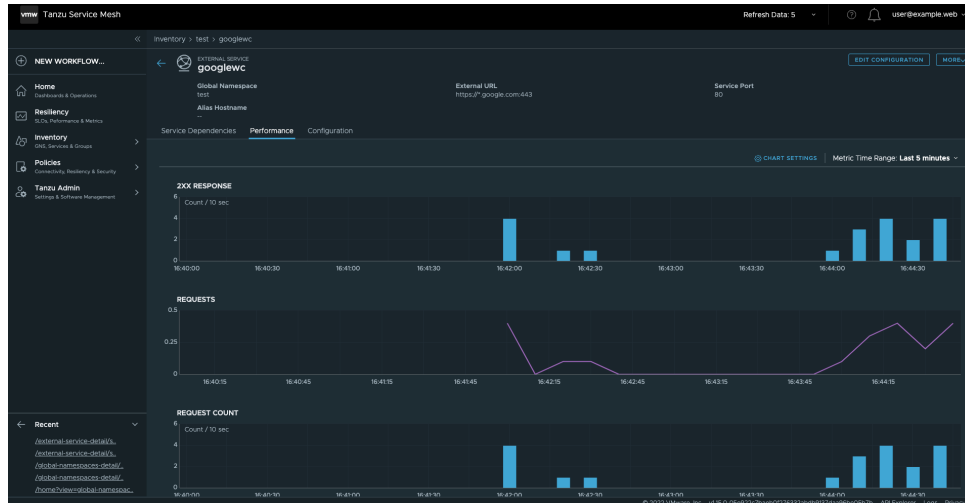
Access wildcard external traffic

Each external server has its own subdomain through which you can access the external service. In the following example subdomain, two distinct Google subdomains are accessed from shopping service inside a global namespace.

[illegible]

Traffic Visibility and Monitoring

External Service traffic can be observed in **GNS Topology** and on the external service **Performance** page.



Edit an External Service Configuration

You can edit the configuration details of an external service. For example, you can change the external service's name, the URL, or the alias name it uses. Additionally, you can add or remove external services or change the TLS certificates.

You can edit the configuration details of an external service from the **External Services** details page.

Note You can also edit the configuration of an external service by editing the [global namespace](#) that contains the public service.

- 1 On the **GNS Overview** tab of the Home page, in the card for the global namespace, click its name.
- 2 In the upper-right corner of the global namespace details page, click **Edit Configuration**.
- 3 In the **Edit Global Namespace** window, on the **External Services** page, make the changes that you want.

Prerequisites

- [Access the Tanzu Service Mesh Console.](#)
- [Chapter 5 Create an External Service.](#)

Procedure

- 1 Open the **External Services** details page.
 - a On the Home page, on the **GNS Overview** tab, in the card for the global namespace that contains the external service, click the name of the global namespace.
 - b On the global namespace details page, select the **External Services** tab.
 - c In the **Service Name** column, click the name of the external service whose configuration you want to edit.

- 2 In the upper-right corner of the external service details page, click **Edit Configuration**.

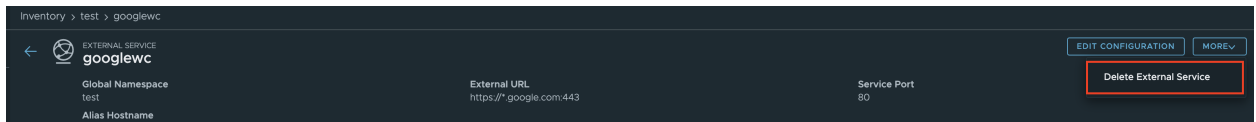


The **External Service Configuration** window displays the name of the external service, the port on which it is accessible, external URL, Connect using an alias, and Gateway addresses. If you selected HTTPs or TLS, you can optionally type in or select the name of the Transport Layer Security (TLS) certificate to use to secure traffic to and from the external service.

- 3 Edit the details that you want.
- 4 Click **Save**.

Results

The **External Services** details page reflects the changes. To delete an external service, select the desired service name from the **External Services** page and click **Delete External Service** from the **More** drop-down in the upper-right corner of the page.



Monitor the Performance of External Services

Tanzu Service Mesh provides detailed information to help you monitor the performance of an external service. This information includes the performance metrics. You can also view the different details about the external service, including its configuration.

You can monitor the performance of an external service and view the details and configuration from its details page in the Tanzu Service Mesh Console UI.

Prerequisites

- [Access the Tanzu Service Mesh Console.](#)
- [Chapter 5 Create an External Service.](#)

Procedure

1 Open the external service details page.

- a On the Home page, on the **GNS Overview** tab, in the card for the global namespace that contains the external service, click the name of the global namespace.
- b On the global namespace details page, click the **GNS Topology** tab. You can view external service traffic on the **GNS Topology** page. The service graph may not appear immediately. Please wait for 30 seconds for the data to be processed.
- c To view external services in detail, click on the **External Services** tab on the **GNS** page. From the **Service Name** column, select the external service you wish to view.

The top of the external service details page displays the following summary information:

- The external URLs of the external service.
- The name and domain of the global namespace that contains the external service, and the alias name of the external service within the global namespace.

The tabs of the external service details page display the different details and configuration information for the external service. For more information about the details that are displayed on each tab, see the following steps.

2 When we select a specific external service, we will see the **Service Dependencies** page, which will list the services inside global namespace that are contacting it.

Inventory > test > testhttp

EXTERNAL SERVICE

testhttp

Global Namespace

test

Alias Hostname

testhttp

Service Dependencies

Performance

Configuration

External URL

http://ec2-34-211-151-159.us-west-2.compute.amazonaws.com:80

Service Port

80

EDIT CONFIGURATION

MORE

SERVICE DEPENDENCIES

Upstream and Downstream services

```

graph LR
    order[order 0.39 rps] --> istio[istio-egressgateway 0.19 rps]
    istio --> testhttp[testhttp 0.0%]

```

Services

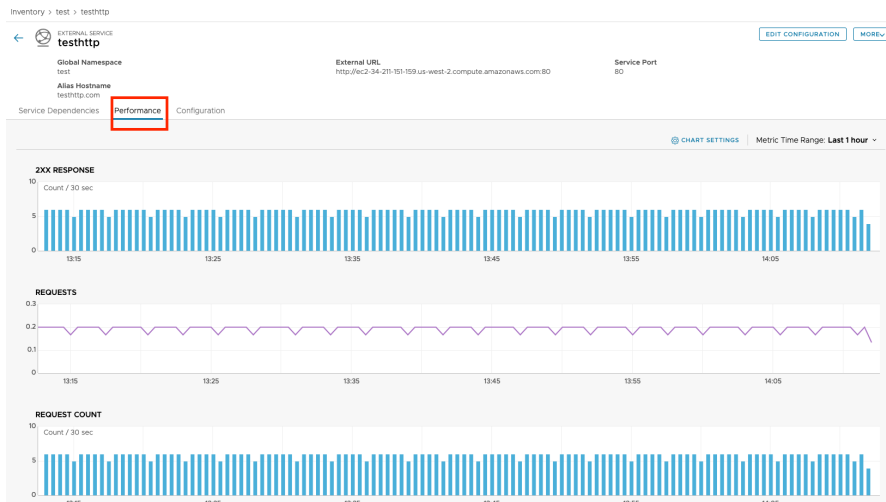
Service Instances

1 Service

COLUMN SETTINGS

Service Name	Service Version	Service Type	Cluster	Service Instances	Requests	Errors
> order	1 version	ServiceDeployment	test	1	--	--

- 3 To monitor the performance of the external service on the different clusters by using metric charts, click the **Performance** tab.

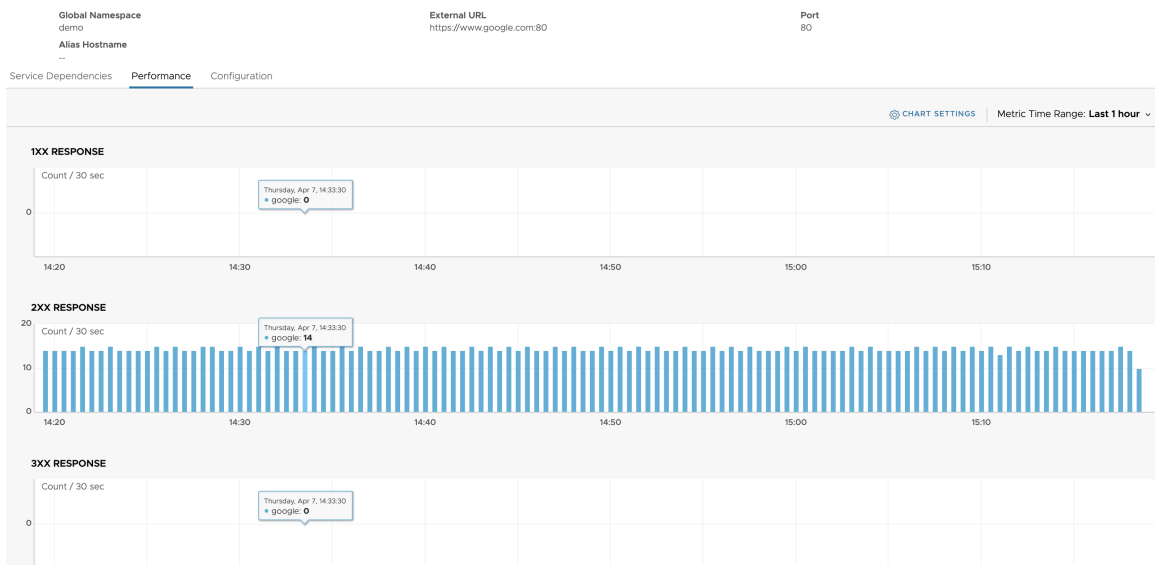


The performance charts on the **Performance** tab displays metrics collected for the external service in the time range selected in the **Metric Time Range** drop-down menu in the upper-right corner.

Note HTTP response status codes indicate whether a specific HTTP request has been successful. These responses are divided into five categories:

- Informational responses (1XX)
- Successful responses (2XX)
- Redirection messages (3XX)
- Client error responses (4XX)
- Server error responses (5XX)

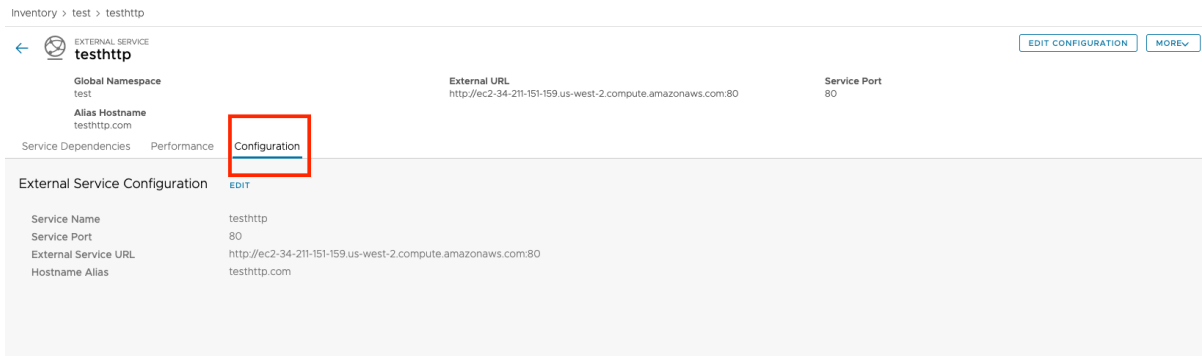
- a To view the metric values collected at a specific date and time, point to the appropriate data point on the chart, as shown the example below.



- b To display only the metrics that you want on the **Performance** tab, in the upper-right corner, click **Chart Settings** and select the check box next to each metric that you want to display.

You can select a maximum of four metrics to be displayed.

- 4 To view the configuration of the external service, click the **Configuration** tab. The **Configuration** tab displays the configuration details for each external service.



Add a Custom Certificate

Secure Sockets Layer (SSL) or TLS (Transport Layer Security) certificates are crucial to the confidentiality of internet browser connections and transactions.

Configuring a Certificate

- 1 Navigate to the **Home** page. From the navigation pane on the left, select **Keys & Certificates** from **Admin**.
- 2 Click **New Certificate**.
- 3 In the **New Certificate** dialog box, type the name of the certificate and select the file location.

The screenshot shows the 'New Certificate' dialog box. It has a close button (X) in the top right corner. Below the title, there's a note: '* indicates required information'. The form contains the following fields:

- Name ***: A text input field containing 'publiccert'. Below it, a small note says '2-1024 characters (a-z, A-Z, _ . ,)'.
- Description (optional)**: A text input field containing 'Optional'.
- Certificate Type ***: A radio button group with 'User-defined Certificate' selected.

Below these fields, there's a section titled 'Configuration for user-defined certificate'. It contains three rows, each with a label, a button, and a text field:

Certificate File *	SELECT .PEM FILE	public_cert.pem
Private Key (optional)	SELECT .PEM/KEY FILE	No file selected
Certificate Chain (optional)	SELECT .PEM FILE	No file selected

At the bottom right of the dialog box, there are two buttons: 'CANCEL' and 'SAVE'.

- 4 Click **Save**, a new certificate will be created.
- 5 On the **Keys & Certificates** page, you can view the list of certificates.

Create a Public Service

6

A public service is a service exposed for external access into the mesh. In Tanzu Service Mesh, you can configure public services to include integration to automate a supported global load balancer and public services without a global load balancer integration.

To create a public service in Tanzu Service Mesh, you must describe its configuration, including the URL at which it will be accessible. This configuration is registered in its global namespace and applied to all of the clusters where this service resides.

If you want to make the service accessible at an HTTPS URL, select a previously uploaded TLS certificate in the service configuration. Clients will use this certificate to make secure, encrypted connections to the service. The domain in the certificate must match the domain specified for the public service.

In Tanzu Service Mesh you can configure public services that integrate with global load balancing (GSLB), or *GSLB-enabled public services*, and public services without GSLB, or *non-GSLB public services*. For a discussion of the concept of global load balancing, see [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).

If you want to publish your application for external access and load balance user requests across multiple clusters where the application is deployed using a GSLB like AWS Route 53 or NSX Advanced Load Balancer, you need to configure a GSLB-enabled public service. GSLB helps achieve high availability of your application.

If GSLB is not used or if the public service is not accessible from outside, you need to configure a non-GSLB public service.

Based on the configuration provided for the public service, Tanzu Service Mesh creates an ingress gateway definition on each Kubernetes cluster where the public service is running. This allows access to the service instance from the cluster's ingress controller and Istio ingress gateways.

When instances of the public service are added or removed on Kubernetes clusters that are onboarded to Tanzu Service Mesh, the global namespace detects these changes and creates any necessary ingress configuration and/or GSLB configuration to make this service accessible from outside.

If a TLS certificate is selected for the service, the certificate will be added to the ingress controller on each cluster.

Important

- The URL configured to the public service restricts access to the public service that fronts the application to that URL only. Any attempts to access the application through other endpoint addresses, such as the cluster's IP address, will be rejected.
- If no GSLB is configured, manually add the URL for the public service into the DNS that clients will use to resolve the URL to the application. For testing, you can configure the local hosts file on your computer to resolve the URL to the public service. In a corporate environment, configure the corporate DNS to resolve the URL of the public service. For instructions on how to configure your DNS, see *Creating a Non-GSLB Public Service - Developer Environment* and *Creating a Non-GSLB Public Service - Corporate Environment* in the [Example section](#).
- If the GSLB that you use is not part of the supported list of providers in Tanzu Service Mesh, you must configure it manually because a GSLB is essentially a DNS service.
- Another option to connect to the application with anything other than the public URL (like the cluster IP), is to apply a custom * wildcard gateway definition directly to Istio on the Kubernetes cluster where the public service is running. Consider this example Istio manifest below:

```

apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: acme-gateway
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
    - port:
        number: 80
        name: http
        protocol: HTTP
      hosts:
        - "*"
---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: acme
spec:
  hosts:
    - "*"
  gateways:
    - acme-gateway
  http:
    - headers:
        request:
          add:
            x-allspark-request-header: gslb-cs1
      route:
        - destination:
            host: frontend.acme.com
  tcp: []

```

Note

- You can configure a public service in a global namespace before adding that service to the global namespace. When you add the service to the global namespace, Tanzu Service Mesh will read its configuration and make it public according to the settings that you specified.
- The gateway configuration above includes a virtual service that is used to control the routing of incoming traffic. For more information about virtual services, see the [Istio documentation](#).

The following procedure describes how to configure a public service in Tanzu Service Mesh, considering both GSLB and non-GSLB options.

Prerequisites

Before you begin configuring a public service, ensure that the following prerequisites are met:

- Verify that you are familiar with the concepts [global namespace](#) and [public service](#) in Tanzu Service Mesh.
- Choose a fully qualified domain name (FQDN) for the URL at which the public service will be accessible:
 - If this service is accessible from the Internet through a GSLB, it must be a valid FQDN that is resolvable on the Internet.
 - If no GSLB integration exists or if this public service is not accessible from the outside, you can define any FQDN that is configured in your DNS.
- (Only for GSLB-enabled public services) If a supported GSLB integration is used, configure the integration for it in Tanzu Service Mesh. Currently, Tanzu Service Mesh supports NSX Advanced Load Balancer and AWS Route53. For more information about creating these integrations, see [Create an Avi Integration Account](#) or [Create an AWS Integration Account](#).
- (Only for GSLB-enabled public services) When using a supported GSLB, create an external DNS account in Tanzu Service Mesh and select the domain provider that manages the public service's domain in the DNS account. For more information, see [Manage Domains](#).
- (Only for GSLB-enabled public services) Verify that you are familiar with the following GSLB concepts that are used in Tanzu Service Mesh. For more information about these concepts, see [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).
 - Global load balancing scheme (round robin, weighted, and active-passive)
 - Health checks
- (Only for GSLB-enabled public services) If you want to use the weighted or active-passive global load balancing scheme for a public service, add the required labels to the service configuration on each cluster where the service is deployed. For instructions on how to add these labels, see [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).
- To use an HTTPS URL for your public service, upload a Transport Layer Security (TLS) certificate for the service to Tanzu Service Mesh. For more information, see [Add Certificates](#).

Procedure

- 1 On the **Public Services** page of the **New Global Namespace** wizard or the **Edit Cluster** dialog box, click **Configure Public Services**.

You can configure a public service in the **New Global Namespace** wizard when you're creating a global namespace or in the **Edit Cluster** dialog box when you're editing the global namespace configuration. For information on accessing the **New Global Namespace** wizard, see [Connect Services Across Clusters with a Global Namespace](#). For information on accessing the **Edit Cluster** dialog box, see [Edit a Cluster](#).

- 2 Provide the following information about the public service.
 - **Service Name.** Select the service that you want to make public. The drop-down menu displays the names of the services in the global namespace. If you want to create a public service that isn't in the global namespace yet, type the service name in this field. When the service is added to the global namespace, Tanzu Service Mesh will make it public according to the configuration.
 - **Internal Service Port.** Specify the port on which the service will be accessible. If a single port is defined for the service in the service configuration, that port is selected by default. If more than one port is defined, select the port that you want to use. You can also specify a port number that is not on the list.

Note The public service will not work until the specified port is exposed by the service.

- **Public URL(s).** Specify the parts of the URL at which the service will be accessible: the protocol, subdomain, and domain. If you select **https**, in the drop-down menu below, select the name of the certificate that you want to use for the service. You can select from the certificates that your administrator has uploaded to Tanzu Service Mesh. The specified URL appears under **This service will be available at:**.
- **Global Load Balancing Scheme.** If you want a GSLB-enabled public service, select the appropriate scheme (**Round Robin**, **Weighted**, or **Active-Passive**). You must perform additional configuration steps based on your choice. If you do not want to use GSLB, select **Non GSLB**.

- 3 (For a GSLB-enabled public service), perform these steps based on the global load balancing scheme that you have selected.

Global load balancing scheme	Perform these steps
Round Robin	Under Health Checks & Failover , configure health checks. For more information, see step 4.
Weighted	<ol style="list-style-type: none"> Under Weighted GSLB, next to Service Label Weights, from the Service Label drop-down menu, select one of the predefined weigh labels. Under Weight, type the weight value that you want to associate with the service label. You can add a weight value between 0 and 20. To define an additional weight, click Add Service Label and repeat steps 2–3. Under Health Checks & Failover, configure health checks. For more information, see step 4. <p>Note Rather than using predefined weight labels, you can add custom weigh labels. To add a custom weight label:</p> <ol style="list-style-type: none"> Under Service Label, click Add Service Label. Type the label that you want. Click Add service label: <label name> that appears below. Under Weight, type the weight value to associate with this label. Assign this custom weight label to the public service instances you want. For each instance, edit the service configuration on its cluster and set <code>tsm_gslb_weight</code> under <code>labels</code> to the label name.
Active-Passive	<ol style="list-style-type: none"> To specify the service label for the active group of service instances, under Active Group, in the Active Group Label drop-down menu, select the predefined active group label. From the Group Load Balancing Scheme/Policy drop-down menu, select a group loading scheme for the active group instances. To specify the service label for the passive group of service instances, under Passive Group, in the Passive Group Label drop-down menu, select the predefined passive group label. From the Group Load Balancing Scheme/Policy drop-down menu, select a group loading scheme for the passive group instances. Under Health Checks & Failover, configure health checks. For more information, see step 4.

Global load balancing scheme	Perform these steps
	<p>If you selected Weighted from the Group Load Balancing Scheme drop-down menu for the active group or the passive group, next to Service Label Weights, define the service weights for the active group or passive group instances. For details, see the description of Weighted in this table.</p> <hr/> <p>Note Rather than using predefined active and passive group labels, you can add custom labels. To add a custom label:</p> <ol style="list-style-type: none"> 1 Next to Active Group Label or Passive Group Label, type the label that you want. 2 Click Add service label: <label name> that appears below. 3 Assign the active group label or the passive group label to the appropriate public service instances. For each instance, edit the service configuration on its cluster and set <code>tsm_gslb_group</code> under <code>labels</code> to the active group label or the passive group label.

4 Under **Health Checks & Failover**, configure health checks.

- To use the default health-check settings, click **Default TSM Health Checks**. To view the default settings, point to the information icon to the right of **Default TSM Health Checks**.
- To use custom health-check settings, click **Custom Health Checks**. Provide values for the following settings.

Setting	Description
Protocol	The protocol (HTTPS or HTTP) to use for health-check requests.
Relative Path	The URL path to which Tanzu Service Mesh sends health-check requests. This path can be different from the public URL that you specified for the public service.
Health Check Interval	The amount of time between health-check requests in seconds. You can select 10 seconds or 30 seconds. For example, if you select 30 seconds, Tanzu Service Mesh will send health-check requests to the service every 30 seconds.
Healthy & Unhealthy Threshold	The number of consecutive failed health checks for the service to be considered unhealthy, or the number of consecutive successful health checks for the service to be considered healthy. For example, with the default Healthy & Unhealthy Threshold value of 3, Tanzu Service Mesh determines that the service is unhealthy after three consecutive failed health-check requests and then considers the service healthy after the subsequent three successful requests.

- 5 To configure another GSLB-enabled or non-GSLB public service in the global namespace, at the bottom of the **Public Services** page, click **Add Public Service** and repeat steps 2–4.
- 6 Click **Next**.
The **Configuration Summary** page, under **Public Services**, displays a summary of each configured public service. If you want to make changes to the configuration, click **Edit**.
- 7 To save the public services that you have configured in the global namespace, click **Finish**.

Example

Creating a GSLB-Enabled Public Service Using a Supported GSLB Integration

An example of a public service with GSLB is a sample e-commerce application. Let's assume that the application has a frontend service that needs to be made public at `store.acme.com` to handle user requests.

To create this public service in Tanzu Service Mesh, you must describe its configuration, including the URL at which it will be accessible (`store.acme.com`). This configuration is registered in its global namespace and is applied to all of the clusters where this service resides and to the GSLB where clients connect.

Creating a Non-GSLB Public Service - Developer Environment

In a development environment where no DNS access exists, developers might need to create a public service without GSLB and configure their local host file for access.

Let's assume that there is a test service called *test-app* that needs to be accessible for local development.

Perform these steps to configure a non-GSLB public service and update the local hosts file:

- 1 Create the *test-app* public service in Tanzu Service Mesh, providing an FQDN that does not exist on the Internet, for example, `test-app.local.com`.
- 2 After the configuration is complete, Tanzu Service Mesh will generate an ingress gateway definition for each Kubernetes cluster where the service runs.
- 3 To access *test-app* from your computer, edit your local hosts file (usually located at `/etc/hosts` on Unix-based systems or `C:\Windows\System32\drivers\etc\hosts` on Windows) to resolve `test-app.local.com` to the IP of the appropriate ingress controller of your Kubernetes cluster, for example: `192.168.0.10 test-app.local.com`.

Creating a Non-GSLB Public Service - Corporate Environment

In a corporate environment, where access to a DNS exists and where that DNS is not a supported GSLB, the operator must manually configure the DNS to point to the ingress where the public service exists.

Let's assume that there is a corporate application named *corp-app* that needs to be made accessible within the corporate network.

- 1 Create the *corp-app* public service in Tanzu Service Mesh, specifying an FQDN that is recognized within the corporate DNS, for example, `corp-app.internal.com`.
- 2 After the configuration is complete, Tanzu Service Mesh will create an ingress gateway definition on each Kubernetes cluster where the service is running.
- 3 Manually configure the corporate DNS to resolve `corp-app.internal.com` to the IP of the corresponding ingress controller of your Kubernetes cluster.

When these steps are completed, users within your corporate network can access the *corp-app* service at the `corp-app.internal.com` URL.

What to do next

For non-GSLB public services, after the configuration is done, you must manually configure your external DNS to point to the ingress where the public service exists. Depending on your environment, this could be a local host file modification or a corporate DNS configuration. This step ensures that the public service is accessible from the intended network. For instructions on how to configure your DNS, see *Creating a Non-GSLB Public Service - Developer Environment* and *Creating a Non-GSLB Public Service - Corporate Environment* in the [Example section](#).

For information about how to monitor the health of a public service, view its details, and monitor the performance of the public service with metric charts, see [Monitor a Public Service](#).

For information about editing the configuration of a public service, including its public URL, see [Edit the Configuration of a Public Service](#).

Configure Global Load Balancing for Your Application in Tanzu Service Mesh

When publishing your application for external access, you can configure a supported global load balancing (GSLB) for it so that user requests are load balanced across multiple clusters in one or multiple zones or regions. With global load balancing, you can achieve high availability of your application.

A general practice for achieving high availability is to deploy an application in more than one cluster and position a global load balancer between them to direct north-south traffic to the frontend services deployed on these clusters.

In Tanzu Service Mesh, you can publish your application to external clients through a [public service](#) from within its [global namespace](#). A public service is an HTTP or HTTPS service that you expose externally from a global namespace. When configuring a public service, you specify one or more fully qualified domain names (FQDNs) for the public service to designate them as the public URLs. A public URL is an URL at which the public service is exposed to external clients.

If you want to publish your application for external access and have an integrated load balancer, such as AWS Route 53 or NSX Advanced Load Balancer, and want to load balance user requests across the clusters where the application is deployed, you need to add global load balancing (GSLB) configuration parameters to the configuration of the public service in its global namespace. In Tanzu Service Mesh, public services that have incoming requests load balanced according to their GSLB configuration are called *GSLB-enabled public services*.

Tanzu Service Mesh will distribute the traffic to the instances of a GSLB-enabled public service according to its global load balancing configuration. AWS Route 53 or NSX Advanced Load Balancer will provide domain name system (DNS) and GSLB capabilities for the service.

The GSLB configuration of a public service consists of the following elements.

- Global load balancing scheme

As part of the GSLB configuration, you must select a global load balancing scheme for each public URL of the public service. You can choose between a round robin, weighted, or active-passive global load balancing scheme.

The *round robin* algorithm distributes user requests equally among the healthy service instances by forwarding requests to each instance in turn.

The *weighted* load balancing algorithm splits traffic to the service into percentage-based portions according to relative service weights. The traffic percentages are calculated according to the formula: a service weight divided by the sum of all of the service weights. For example, if you assign a service instance a weight of 20 and give other two instances a weight of 10, the load balancing algorithm will compute that 50% of the traffic goes to the first instance, and 25% of the traffic is sent to each of the other two instances. To define weights for the public service, you must add a weight label to the service configuration on each cluster where the service is deployed and then associate these predefined weight labels with the appropriate weight values in the UI. A percentage of the traffic will be routed to a specific public service instance based on the weight value associated with the weight label that is defined for that instance in its service configuration.

With the *active-passive* scheme, you can configure a failover routing policy by defining active and passive groups of service instances. Traffic is routed to the active group until at least one instance in the active group is healthy. If health checks determine that all the active instances are unhealthy, the load balancing algorithm fails over to the instances in the passive group. To configure active and passive groups, you must add an active group label or a passive group label to the service configuration on each cluster where the service is deployed and then select these predefined labels for the active and passive groups respectively in the UI. A specific public service instance is assigned to the active group or passive group based on whether the active group label or passive group label is defined in its service configuration.

You additionally define a group load balancing scheme, round robin or weighted, for the service instances within the active group and the passive group. Round robin is selected by default. For example, if you specify a group load balancing scheme of round robin for the passive group and if all the active group instances are unhealthy, the traffic will be rotated through the passive group instances according to the round robin algorithm.

■ Health checks

You also configure health checks in the global load balancing configuration. Tanzu Service Mesh uses the specified health-check settings to periodically check whether the public service is reachable and functional. If health checks determine that a service instance is unhealthy, Tanzu Service Mesh routes traffic to other instances according to the configured global load balancing scheme. You can use the default health-check settings or define custom settings.

You can create a GSLB-enabled public service and provide its GSLB and health-check parameters in the **New Global Namespace** wizard when you're creating a global namespace or in the **Edit Cluster** dialog box when you're editing the global namespace configuration. For information about creating a GSLB service, see [Chapter 6 Create a Public Service](#).

As noted above, if you want to use the weighted or active-passive global load balancing scheme for a public service, you must add weight labels or active group and passive group labels to the service configuration on each cluster where the service is deployed. You must add these labels on the clusters as a prerequisite to adding the GSLB configuration for the public service.

The following procedure describes the steps that you must perform on each cluster where instances of the public service are deployed to add the weight labels or active and passive group labels for the service.

Procedure

- 1 Set the current context to the cluster and the namespace where the public service instances are deployed.

```
kubectl config set-context {context} --cluster={cluster_name} --namespace={namespace_name}
```

- 2 To edit the service configuration on the cluster, run this command.

```
kubectl -n {namespace_name} edit svc/{service_name}
```

3 In the service configuration, under `labels`, add the following labels.

Label	Description
<code>tsm_gslb_group</code>	<p>The active group or passive group label. Set the label to a meaningful value to specify whether this is the active group label or passive group label. See examples below.</p> <hr/> <p>Important In the service configuration of a specific public service instance, you can use <code>tsm_gslb_group</code> as the active group label or the passive group label, but not for both groups.</p> <hr/>
<code>tsm_gslb_weight</code>	The weight label. Set the label to a meaningful value to help associate the label with the appropriate weight in the UI. See examples below.

Note Make sure that you add the label names exactly as are given in the Label column.

For example, a public service named *cart* is deployed in three clusters: cluster A, cluster B, and cluster C. You want the public service instances on clusters A and B to be in the active group, and the public service instance on cluster C in the passive group.

You add the following labels to the service configuration on each cluster.

Cluster A

Active group label

```
tsm_gslb_group: ActiveGroup
```

Weight label

```
tsm_gslb_weight: primaryload
```

Cluster B

Active group label

```
tsm_gslb_group: ActiveGroup
```

Weight label

```
tsm_gslb_weight: secondaryload
```

Cluster C

Passive group label

```
tsm_gslb_group: PassiveGroup
```

What to do next

You can view the global load balancing configuration of the public service on the **Configuration** tab of the public service details page. For more information, see [Monitor a Public Service](#).

You can edit the global load balancing configuration from the public service details page. For more information, see [Edit the Configuration of a Public Service](#).

You can view the information about how traffic is routed to the different public URLs of a public service, including the service-level metrics like requests per second (RPS), on the **GSLB Routing** tab of the public service details page. For more information, see [Monitor a Public Service](#).

Monitor a Public Service

Tanzu Service Mesh provides detailed information to help you monitor the health and performance of a public service. This information includes the overall health status, global server load balancing (GSLB) routing information for the service's public URLs, and performance metrics. You can also view the different details about the public service, including its configuration.

You can monitor a public service and view its details and configuration from its details page in the Tanzu Service Mesh Console UI.

The details include the health status of the public service. Tanzu Service Mesh computes an overall health status based on the health status of each public URL of the public service. Tanzu Service Mesh periodically sends the configured number of connection attempts, or health check probes, to each URL to evaluate the health status of the URL. If the service responds successfully to the probes sent to a particular URL on all the clusters, that URL is considered healthy. If the service fails to respond to the probes sent to a URL on all or some of the clusters, that URL is considered unhealthy. The overall status of the public service is healthy if all its public URLs are healthy. If at least one of the public URLs is unhealthy, the overall state of the public service is considered unhealthy.

A public service can have one of the following color-coded statuses:

- Healthy (in green) - The public service is healthy and reachable at all its public URLs.

- Syncing (in blue) - A temporary state that is displayed when you create a public service or edit the public service configuration. This status is displayed for a few minutes while Tanzu Service Mesh is applying the configuration of the public service to AWS, NSX Advanced Load Balancer, and the client clusters, and while Tanzu Service Mesh is sending health check probes to the service on the different clusters to see if it's healthy. When the public service configuration is applied, and the health checks are complete, this status changes to Healthy, Warning, or Error.

Note The **Health Check Interval** and **Healthy & Unhealthy Threshold** specified in the health check settings of the public service affect the length of time that the public service is in Syncing state.

- Warning (in yellow) - The public service is unreachable at some of its public URLs because of a problem with the service on some of the clusters.
- Error (in red) - The public service is down or unreachable on all the clusters because of a problem.

Prerequisites

- [Access the Tanzu Service Mesh Console.](#)
- [Chapter 6 Create a Public Service.](#)

Procedure

- 1 Open the public service details page.
 - a On the Home page, on the **GNS Overview** tab, in the card for the global namespace that contains the public service, click the name of the global namespace.
 - b On the global namespace details page, click the **Public Services** tab.

- c In the **Public Service** column, click the name of the public service.

The top of the public service details page displays the following summary information:

- The public URLs of the public service, that is, the URLs at which users and external clients can access the service.
- Overall health status of the public service. The status is displayed in a color-coded rectangle. Tanzu Service Mesh computes the status based on whether the service is reachable at its public URLs. To see the health status of the service at each URL with a breakdown by cluster in the **Public Service Status** window, click the status rectangle. If the status is **Warning** or **Error**, the **Public Service Status** window displays the details of the error, and the clusters where the service is unhealthy is indicated with a red circle with an exclamation mark.

Note If the public service is in the **Warning** or **Error** state, running this kubectl command on a cluster can help determine the cause of the problem:

```
kubectl --context {cluster} get pods -A
```

If the service is not running correctly, it has a status other than Running in the output of the command. Take the appropriate corrective action according to the error status.

If the service has a status of Running, and other services on the cluster can access the service, there might be a problem with the ingress gateway on the cluster or with the global server load balancing (GSLB) configuration of the public service. In that case, to resolve the problem, open a support request with VMware.

-
- The name and domain of the global namespace that contains the public service, and the local address of the public service within the global namespace. To view the details page for the global namespace, click its name.
 - The names of the clusters that host instances of the public service. To view the details page for a cluster, click its name.

The tabs of the public service details page display the different details and configuration information for the public service. For more information about the details that are displayed on each tab, see the following steps.

- d To view the GSLB routing information for the public service, click the **GSLB Routing** tab.

The top of the **GSLB Routing** tab displays a selector for switching between the different public URLs of the public service and the aggregated requests per second (RPS), 99th percentile latency, and error rate metrics for the selected URL. If the public service has more than two public URLs, you can access the other URLs by clicking the box with three ellipses. The legend that identifies the color codes used for the clusters appears to the right of the metrics.

The tab displays a GSLB routing data card for the selected public URL. A separate color-coded rectangle appears inside the card for each cluster. The boxes inside a cluster rectangle represent the number of public service instances deployed in that cluster. To display the metrics and details for a service instance in a hover card, point to the box for that instance inside its cluster rectangle.

The global load balancing scheme (Round Robin, Weighted, or Failover) specified for the URL appears in the upper-right corner. The RPS and error rate metrics are broken down by cluster. If a global load balancing scheme of Failover (Active/Passive) is specified, the metrics are also broken down for the active group and passive group.

The lines to the left of the GSLB routing data card represent the incoming connections of the public service to other services, and the lines to the right of the card represent the outgoing connections. The names of the connected services appear above the boxes at the end of the lines.

- 2 To view the global load balancing (GLSB) routing information for the public service, click the **GSLB Routing** tab.

The top of the **GSLB Routing** tab displays a selector for switching between the different public URLs of the public service and the aggregated requests per second (RPS), 99th percentile latency, and error rate metrics for the selected URL. If the public service has more than two public URLs, you can access the other URLs by clicking the box with three ellipses. The legend that identifies the color codes used for the clusters appears to the right of the metrics.

The tab displays a GSLB routing data card for the selected public URL. A separate color-coded rectangle appears inside the card for each cluster. The boxes inside a cluster rectangle represent the number of public service instances deployed in that cluster. To display the metrics and details for a service instance in a hover card, point to the box for that instance inside its cluster rectangle.

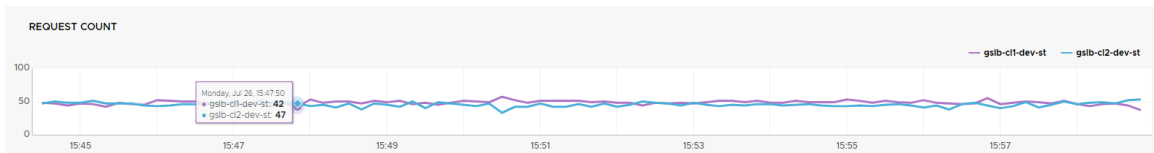
The global load balancing scheme (Round Robin, Weighted, or Failover) specified for the URL appears in the upper-right corner. The RPS and error rate metrics are broken down by cluster. If a global load balancing scheme of Failover (Active/Passive) is specified, the metrics are also broken down for the active group and passive group.

The lines to the left of the GSLB routing data card represent the incoming connections of the public service to other services, and the lines to the right of the card represent the outgoing connections. The names of the connected services appear above the boxes at the end of the lines.

- 3 To monitor the performance of the public service on the different clusters by using metric charts, click the **Performance** tab.

The performance charts on the **Performance** tab displays metrics collected for the public service in the time range selected in the **Metric Time Range** drop-down menu in the upper-right corner. Each chart displays a separate colored line for each cluster where the public service is deployed. The cluster color-code legend appears at the upper right above each chart.

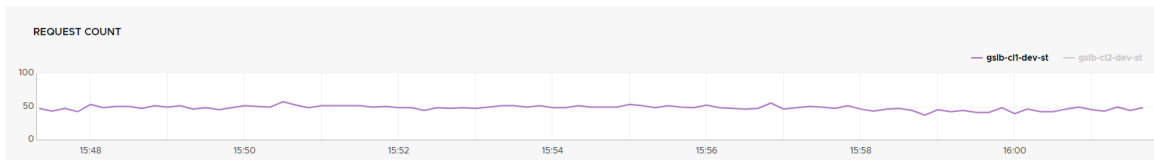
- a To view the metric values collected at a specific date and time, point to the appropriate data point on the chart, as shown the example below.



- b To display the metric lines only for some of the clusters and hide the lines for the other clusters on a chart, in the cluster color-code legend above the chart, click the name of each cluster that you want to hide.

The chart displays the metric lines only for the selected clusters. The names of the clusters that you chose to hide appear gray in the cluster color-code legend.

This example shows that the Requests Count chart displays the metric line only for a cluster named *gsib-cl1-dev-st* and hides the line for the *gsib-cl2-dev-st* cluster.



- c To display only the metrics that you want on the **Performance** tab, in the upper-right corner, click **Chart Settings** and select the check box next to each metric that you want to display.

You can select a maximum of four metrics to be displayed.

- 4 To view the details and the infrastructure metrics for the nodes that contain the public service instances, click the **Nodes** tab.

To view the details of a node on the node details page, in the **Node Name** column, click the name of the node.

- 5 To view the configuration of the public service, click the **Configuration** tab.

The **Configuration** tab displays the configuration details for each public URL of the public service. For information about the different configuration details and settings, see [Chapter 6 Create a Public Service](#) and [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).

Note You cannot edit the configuration details on the **Configuration** tab. To edit the configuration of a public service, in the upper-right corner of the public service details page, click **Edit Configuration** and then make the changes that you want in the **Public Service Configuration** window. For more information, see [Edit the Configuration of a Public Service](#).

Edit the Configuration of a Public Service

You can edit the configuration details of a public service. For example, you can change the name of the public service or the port on which it is accessible. You can also add or remove public URLs, that is, the URLs at which the public service is exposed to external clients.

You can edit the configuration details of a GSLB-enabled public service or non-GSLB public service from the public service details page. For an explanation of GSLB-enabled and non-GSLB public services, see [Create a Public Service](#).

Prerequisites

- Verify that you are familiar with the concepts of [global namespace](#) and [public service](#) in Tanzu Service Mesh.
- [Access the Tanzu Service Mesh Console](#).
- [Chapter 6 Create a Public Service](#).

Procedure

- 1 Open the public service details page.
 - a On the Home page, on the **GNS Overview** tab, in the card for the global namespace that contains the public service, click the name of the global namespace.
 - b On the global namespace details page, click the **Public Services** tab.
 - c In the **Public Service** column, click the name of the public service whose configuration you want to edit.
- 2 In the upper-right corner of the public service details page, click **Edit Configuration**.
- 3 In the **Edit Global Namespace** wizard, go to the **Public Services** page.
- 4 Under **GSLB Public Services Configuration**, in the configuration section of the public service, make the changes that you want.

For a description of the configuration details of a GSLB-enabled public service or a non-GSLB public service, see [Chapter 6 Create a Public Service](#).

- 5 (Optional) To remove the public service from the global namespace, in the upper-right corner of its configuration section, click **Remove**.
- 6 (Optional) To configure a new public service within the global namespace, click **Add New Public Service** at the bottom of the page.

For detailed instructions on configuring a public service, see [Chapter 6 Create a Public Service](#).

- 7 On the **Configuration Summary** page of the wizard, under **Public Services**, review the changes you have made and click **Finish**.

To make additional changes to the configuration of the public service, next to **Public Services**, click **Edit** to return to the **Public Services** pages and make the changes you want.

Results

The **Configuration** tab of the public service details page reflects the changes in the configuration.

Note You can also edit the configuration of a public service from the details page of its global namespace.

- 1 On the **GNS Overview** tab of the Home page, in the card for the global namespace, click its name.
 - 2 In the upper-right corner of the global namespace details page, click **Edit Configuration**.
 - 3 In the **Edit Global Namespace** wizard, on the **Public Services** page, make the changes that you want to the configuration of the public service.
-

Headless Service with StatefulSet

7

StatefulSets are used to manage stateful applications, such as databases or other applications, that keep track of their state. By using StatefulSets, a set of pods can be deployed and scaled within a global namespace, ensuring that they are ordered and unique. Headless service is a regular Kubernetes service where the **spec.clusterIP** is explicitly set to "None" and **spec.type** is set to "ClusterIP". Instead, SRV records are created for all the named ports of service's endpoints.

Prerequisite

- [Onboard the clusters where your services are deployed to Tanzu Service Mesh](#). For a global namespace, it can be a single cluster with multiple namespaces.
- Create a global namespace to which you can add the headless service. For information about creating a global namespace and adding services to it, see [Connect Services Across Clusters with a Global Namespace](#).
- Create a Headless service with the label and set the clusterIP field to None.

Context

Deploying and replicating stateful applications poses the following challenges:

- Pod replicas cannot be created or deleted at the same time. StatefulSet will not create the next pod until the previous pod is up and running.
- Pods are not interchangeable. Pod replicas have a persistent identifier across any rescheduling.
- Continuous data synchronization between pods is necessary to maintain same states.
- If all pods die or the clusters running these pods crash, data will be lost. Each pod should have its own persistent storage with replicable data and pod state, stored in the pod's own storage, so that when a pod dies, the Persistent Pod Identifiers will reattach the volume to the replaced pod. Reattaching persistent volumes requires remote storage.
- StatefulSet pods get fixed ordered names (\$statefulsetname-\$ordinal). For example, if you create a StatefulSet with a name of *mongo* with three replicas, the replicated pods get names **mongo-0**, **mongo-1**, and **mongo-2**.

- Most importantly, the stateful workloads often has to reach a specific pod directly (for example, during database write operations) or have pod-pod communication, **without load balancing**.

Headless Services are the best solution for the above issues. Tanzu Service Mesh implements Headless service for stateful applications that allows clients to directly access pods without using Kubernetes' load balancing. If you deploy an application within a global namespace as StatefulSet with Headless services, Tanzu Service Mesh supports sending traffic to those services from other pods within the cluster or from remote clusters using the Headless service name. Each pod from StatefulSet gets its own DNS name of the form: **`${podname}.{governing service name}`**. When a pod restarts, IP address changes but the name and endpoint remains the same.

- When the StatefulSet is in the same global namespace as the source traffic, then the client will use the Kubernetes service associated with the statefulset to resolve the endpoint addresses. Hostnames resolve to all statefulset pod endpoint addresses.
- When a client is in a remote cluster but within the same global namespace, it resolves to a number of virtual IPs corresponding to the StatefulSet replica count. Routing is handled by existing global namespace service entries and virtual services.

Creating a Headless Service

- 1 A headless service defined in one of the namespaces of a global namespace having the following definition can be extended by creating a selectorless headless service in the other namespaces of the global namespace.

```
apiVersion: v1
kind: Service
metadata:
  name: foo
spec:
  clusterIP: None
  selector:
    key1: val1
    key2: val2
  ports:
    - port: 8080
      protocol: TCP
  type: ClusterIP
```

- 2 Create a headless service in the other global namespace namespaces. The selectors which would have been defined in the **spec.selector** section will be defined in a custom annotation:

```
apiVersion: v1
kind: Service
metadata:
  name: foo
  annotations: tsm.tanzu.vmware.com/endpoints.statefulset: '{"key1": "val1", "key2": "val2"}'
spec:
  clusterIP: None
```

```
ports:
  - port: 8080
    protocol: TCP
type: ClusterIP
```

- 3 Tanzu Service Mesh will auto manage the endpoints for this headless service in the namespaces where the selectorless headless service is defined.

Use case: Kafka Installation with Headless Services for access within a global namespace

- 1 Create two namespaces, 'nsOne' and 'nsTwo' on the cluster, 'clusterOne'.

```
k --context clusterOne create ns nsOne
k --context clusterOne create ns nsTwo
```

- 2 Create a namespace 'nsOne' on the cluster, 'clusterTwo'.

```
k --context clusterTwo create ns nsOne
```

- 3 Create a global namespace called 'kafka-gns' having the following members: 'clusterOne' / 'nsOne', 'clusterOne' / 'nsTwo', and 'clusterTwo' / 'nsOne'.
- 4 Install Kafka on 'clusterOne' / 'nsOne'.

```
#add helm repo
helm repo add bitnami https://charts.bitnami.com/bitnami
#install kafka on clusterOne/nsOne
helm install kafka --kube-context=clusterOne -n nsOne --set replicaCount=3 bitnami/kafka
```

- 5 Create the following headless service with no selectors in the other namespaces of the global namespaces- 'clusterOne' / 'nsTwo' and 'clusterTwo' / 'nsOne'.

```
apiVersion: v1
kind: Service
metadata:
  annotations: tsm.tanzu.vmware.com/endpoints.statefulset: '{"app.kubernetes.io/
component":"kafka","app.kubernetes.io/instance":"kafka","app.kubernetes.io/name":"kafka"}'

  name: kafka-headless
spec:
  clusterIP: None
  ports:
    - name: tcp-client
      port: 9092
      protocol: TCP
      targetPort: kafka-client
    - name: tcp-internal
      port: 9093
      protocol: TCP
      targetPort: kafka-internal
```

- 6 Test the setup by running a producer and consumer from any of the global namespace members. The following command would start and open a prompt within a temporary kafka-client container.

```
kubect1 --context=${cluster} --namespace ${ns} run kafka-client --image docker.io/bitnami/kafka:3.1.0-debian-10-r89 --rm -it --command -- sh
```

- 7 At the prompt within the temporary kafka-client container, create a producer and a consumer using the . service name.

```
# create topic and populate it with messages
kafka-producer-perf-test.sh --topic topic-foo --num-records 10000 --throughput -1 --record-size 1000 --producer-props bootstrap.servers=kafka:9092
# consume messages from the previously created topic
kafka-consumer-perf-test.sh --bootstrap-server kafka:9092 --topic topic-foo --messages 10000 --timeout 10000
```

Advantages of Headless Services

- Direct access to each pod.
- Easy Pod discovery in the StatefulSet.
- Pods can be addressed more generally by using their DNS names.
- Utilizes each pod's sticky identity in a stateful service (i.e. you can address a specific pod by name).
- Write operations are synchronized.

Manage the Tanzu Service Mesh Components with Tanzu Service Mesh Lifecycle Manager Operator



Tanzu Service Mesh Lifecycle Manager Operator automatically manages the Tanzu Service Mesh components on onboarded clusters. You can use Tanzu Service Mesh Lifecycle Manager Operator to determine the health of the components on a cluster.

Tanzu Service Mesh Lifecycle Manager Operator is a component of Tanzu Service Mesh that is installed on a cluster during onboarding. When installed, Tanzu Service Mesh Lifecycle Manager Operator automatically performs the following operations on the cluster:

- Installs the components of a Tanzu Service Mesh agent on the cluster. A Tanzu Service Mesh agent is an instance of the Tanzu Service Mesh software that runs on each onboarded cluster.
- Redeploys missing agent components on the cluster. If an agent component was accidentally removed, Tanzu Service Mesh Lifecycle Manager Operator automatically recovers and redeploys the component.
- Monitors the health of the agent components by periodically performing health checks and reporting on the health state of each component.
- Removes the Tanzu Service Mesh components when the cluster is removed from Tanzu Service Mesh.

This topic contains instructions on how you can use Tanzu Service Mesh Lifecycle Manager Operator to determine the current health state of the agent components installed on your cluster. You can use the health state information provided by Tanzu Service Mesh Lifecycle Manager Operator for debugging and troubleshooting. You can also provide this information to the Tanzu Service Mesh team to help you with debugging.

Prerequisites

Verify that the cluster where you want to determine the health of the installed Tanzu Service Mesh components is onboarded to Tanzu Service Mesh. For more information about onboarding a cluster, see [Onboard a Cluster to Tanzu Service Mesh](#) in *Getting Started with VMware Tanzu Service Mesh*.

Procedure

- 1 In a terminal window, run the following command.

```
kubectl --context {cluster_name} -n vmware-system-tsm get tsmclusters.tsm.vmware.com tsm-client-cluster -o yaml
```

Where `{cluster_name}` is the name of your cluster.

In the output of the command, under `components`, the `healthState` field contains the health state determined by Tanzu Service Mesh Lifecycle Manager Operator for each component. The `name` field contains the name of the component.

Note

- A `healthState` of `Unknown` is shown if Tanzu Service Mesh Lifecycle Manager Operator hasn't determined the health state yet. `Unknown` is also shown for the agent components that represent jobs, not long-running tasks, such as `ecr-read-only--renew-token`.
 - The `state` field for each component indicates the operational status of the component. If the `state` is other than `OK`, the health state may not be available. For example, if the `state` is `Installing` (the component is being installed on the cluster) or `Upgrading` (the component is being upgraded to a new version), Tanzu Service Mesh Lifecycle Manager Operator cannot determine the health state of the component. Wait until the component is installed or upgraded and then rerun the command above to retrieve the health state.
-

- 2 If the `healthState` of a component is `Unhealthy`, provide this health state to the Tanzu Service Mesh team to help diagnose and resolve the problem with the component.

Manage Tanzu Service Mesh Updates

To take advantage of the latest features, enhancements, and fixes in new versions of Tanzu Service Mesh, you can install upgrades on your clusters.

When a new version of Tanzu Service Mesh is released, the version number is displayed at the top of the **Software Upgrades** page. You can upgrade your clusters to this version. The [VMware Tanzu Service Mesh Data Plane Release Notes](#) contains information about the released version. You can access the release notes by following a link on the **Software Upgrades** page.

When an upgrade is installed on a cluster, the data plane components that run on the cluster are upgraded to the versions released within the selected Tanzu Service Mesh version. You can see the versions of the data plane components released within a specific version in the *VMware Tanzu Service Mesh Data Plane Release Notes*.

If an upgrade fails for some reason, Tanzu Service Mesh automatically rolls back to the previous version installed on the cluster. You can also choose to roll back a cluster to the previous version manually.

Consider the following restrictions that apply to upgrades and rollbacks:

- You can only upgrade from a major version to a minor or patch version released within that major version or to the next major version. An upgrade cannot skip major versions. For example, if your clusters are on version 2.0.0, you can upgrade to 2.0.1, 2.1.0, or 3.0.0, but not to 4.0.0.
- You can roll back only to the minor version, the patch version, or the major version that was last installed on the clusters. For example, if your clusters are on version 4.0.1, you can roll back to version 4.0.0 or 3.0.0, but not to 2.0.0.
- The latest minor version and the latest major version can be both available to upgrade to. For example, if your clusters are on version 2.0.0, you can choose to upgrade to 2.0.1 or to 3.0.0.

The following procedure describes how to install a Tanzu Service Mesh upgrade and roll back a cluster to the previously installed version.

Prerequisites

Verify that you are in the Tanzu Service Mesh Console. For information about accessing the Tanzu Service Mesh Console, see [Access the Tanzu Service Mesh Console](#).

Procedure

- 1 In the navigation pane on the left, click **Admin > Software Upgrades**.

The latest available version is displayed at the top of the **Software Upgrades** page. You can click the **Release Notes** link to open the release notes for this version and the past released versions.

One of the following general statuses appears at the top of the page:

- **All clusters are up to date.** All your clusters use the latest available version of Tanzu Service Mesh and don't need an upgrade.
- **{Number} clusters have out of date software** – The indicated number of clusters use an old version of Tanzu Service Mesh. You can upgrade the clusters to the latest version.
- **Updating {number} clusters** – This status appears if an upgrade or a rollback is in progress for one or more clusters.

The table below displays the names of your clusters and displays one of the following statuses for each cluster:

- **Up to Date – v{version}** . The cluster has the latest version of Tanzu Service Mesh and does not need to be upgraded.
- **Out of Date – v{version}**. The cluster uses an old version of Tanzu Service Mesh. You can upgrade it to the latest version.
- **Updating to v{version}** - The cluster is being upgraded or rolled back to a specified version.

2 To install an upgrade on a cluster, perform these steps:

- a In the table, at the right end of the cluster row, click **Update to v{version}**.
- b If both a minor version and a major version are available to upgrade to, click the three dots to the right of **Update to v{version}**, select the version to upgrade to, and then click **Update to v{version}**.

The status of the cluster changes to **Updating to v{version}**, and a progress bar appears to the right of the status. The message `Cluster {cluster_name} - Upgrading mesh dependencies` with a progress bar appears in the lower-right corner.

Note You can upgrade more than one cluster simultaneously by clicking **Update to v{version}** at the end of each cluster row.

When the upgrade is complete, **Up to Date – v{version} Tanzu Service Mesh upgraded** appears to the right of the cluster name in the table, and the message `Cluster {cluster_name} - Tanzu Service Mesh upgraded` appears in the lower-right corner.

Note

- If you chose to upgrade the cluster to a version that is not the latest version available, a status of **Out of Date – v{version} Tanzu Service Mesh upgraded** appears to the right of the cluster name after the upgrade. You can upgrade to the latest version by clicking **Update to v{version}**.
 - If the upgrade fails for some reason, Tanzu Service Mesh displays an error message and automatically rolls the cluster back to the previously installed version.
-

3 To roll back a cluster to the previously installed version of Tanzu Service Mesh, at the right end of the cluster row, click **Rollback To v{version}**.

Note The rollback option is available only for clusters with a status of **Out of Date**.

The status of the cluster changes to **Updating to v{version}**, the message `Upgrading mesh dependencies` with a progress bar appears to the right of the status. The message `Cluster {cluster_name} - Upgrading mesh dependencies` with a progress bar also appears in the lower-right corner.

Note When Tanzu Service Mesh is automatically performing a rollback, the message `Rolling back mesh dependencies` appears to the right of the status. The message `Cluster {cluster_name} - Rolling back mesh dependencies` also appears in the lower-right corner.

When a rollback is complete, **Out of Date – v{version} Tanzu Service Mesh upgraded** appears to the right of the cluster name in the table, and the message `Cluster {cluster_name} - Tanzu Service Mesh upgraded` appears in the lower-right corner.

Note When an automatic rollback is complete, **Out of Date – v{version} Tanzu Service Mesh rolled back** appears to the right of the cluster name in the table, and the message `Cluster {cluster_name} - Tanzu Service Mesh rolled back` appears in the lower-right corner.

View Details of a Tanzu Mission Control Managed Cluster

9

If you have clusters managed by VMware Tanzu Mission Control and you have added them to Tanzu Service Mesh, you can view details about the clusters in the Tanzu Mission Control console by using links in the Tanzu Service Mesh Console UI.

You can use links in the Tanzu Service Mesh Console UI to go to the details of your Tanzu Mission Control cluster on the cluster detail page in the Tanzu Mission Control console. For a description of the details displayed on the cluster detail page, see the [VMware Tanzu Mission Control product documentation](#).

Note The links to a Tanzu Mission Control cluster in the Tanzu Service Mesh Console UI are only available if you have enabled Tanzu Service Mesh in Tanzu Mission Control. For more information, see the *Using VMware Tanzu Mission Control* documentation.

Prerequisites

Verify that:

- You have enabled Tanzu Service Mesh for your organization in VMware Tanzu™ Mission Control™. For more information, see the [Using VMware Tanzu Mission Control documentation](#).
- You have added your Tanzu Mission Control clusters to Tanzu Service Mesh from the Tanzu Mission Control console. For more information, see the [Using VMware Tanzu Mission Control documentation](#).
- You are in the Tanzu Service Mesh Console. For information about accessing the Tanzu Service Mesh Console, see [Access the Tanzu Service Mesh Console](#).

Procedure

- ◆ To go to the cluster details in Tanzu Mission Control by using one of the links in the Tanzu Service Mesh Console UI perform the steps described in the following table.

To access this link	Perform these steps
The Tanzu Mission Control link on the cluster card	<ol style="list-style-type: none"> 1 In the navigation panel on the left, click Home. 2 On the GNS Overview tab, in the upper-right corner of the card for the Tanzu Mission Control cluster, click the three vertical dots and click Open with Tanzu Mission Control.
The Tanzu Mission Control link in the clusters table	<ol style="list-style-type: none"> 1 In the navigation panel on the left, click Inventory and then Infrastructure. 2 Click the Clusters tab. 3 In the Management column, click Tanzu Mission Control. <hr/> <p>Note</p> <ul style="list-style-type: none"> ■ If the Management column is not visible, click Column Settings at the upper-right, above the cluster table, and select the Management check box. ■ You can also click the three vertical dots to the left of the cluster name in the table and then click Open with Tanzu Mission Control on the menu.
The Tanzu Mission Control link on the Cluster Details page	<ol style="list-style-type: none"> 1 In the navigation panel on the left, click Inventory and then Infrastructure. 2 Click the Clusters tab. 3 In the clusters table, click the name of the Tanzu Mission Control cluster. 4 In the upper-right corner of the cluster details page, click Tanzu Mission Control.

View the Topology of Services in a Global Namespace or a Cluster

10

You can view the topology of the services in a global namespace or a cluster as a chart. You can use the topology graph to understand the service dependencies and determine the health of the services.

Tanzu Service Mesh generates a topology graph view dynamically by observing the traffic that is flowing between the services in a global namespace or a cluster. The topology graph visualizes the communication patterns between the services and helps you gain insights into the service dependencies. The topology graph also shows the key metrics for the services, thus helping you determine the health of the services. The following key health metrics are displayed:

- The service's incoming requests per second (RPS) metric
- The error rate, that is, the percentage of failed requests to the service
- The 99th percentile latency of requests processed by the service

Prerequisites

- [Access the Tanzu Service Mesh Console.](#)
- To view the topology of services in a cluster, [onboard the cluster to Tanzu Service Mesh.](#)
- To view the topology of services in a global namespace, [Chapter 3 Connect Services Across Clusters with a Global Namespace](#) and map the services to the global namespace.

Procedure

- 1 On the Home page, perform one of the following steps.

Option	Description
To view the service topology in a global namespace	<ol style="list-style-type: none"> a Click the GNS Overview tab. b On the card for the global namespace that you want, click the global namespace name or the topology thumbnail.
To view the service topology in a cluster	<ol style="list-style-type: none"> a Click the Cluster Overview tab. b On the card for the cluster that you want, click the cluster name or the topology thumbnail.

Note

- You can also click the three vertical dots in the upper-right corner of the card and then click **Browse Global Namespace Topology** or **Browse Cluster Topology**.
- You can also view the topology graph from the **Global Namespaces** or **Clusters** page. To the left of the global namespace or cluster name, click the three vertical dots and then click **Browse Global Namespace Topology** or **Browse Cluster Topology**. To access the **Global Namespaces** page, in the navigation pane on the left, select **Inventory > Global Namespaces**. To access the **Clusters** page, select **Inventory > Infrastructure** and then click the **Clusters** tab.

The Topology Browser window displays the topology graph for the services in the global namespace or the cluster. The nodes in the graph represent the services, and the lines represent the incoming and outgoing connections between the services based on observed traffic.

For a global namespace that is mapped to services from multiple clusters, the service topology in each cluster is displayed inside a colored rectangle, and the cross-cluster service connections are shown. The color legend at the top of the window identifies the clusters by name.

Tip

- To focus on the incoming and outgoing connections of a specific service and hide the other connections, click the service node. To return to the full graph view, in the upper-left corner of the window, click **Back**.
- To view the incoming and outgoing connections of a service on the **Service Dependencies** tab of the service details page, click the service name under the service node.
- You can also reposition the service nodes and entire cluster rectangles in the graph. To reposition a service node, drag it to a new position. To reposition an entire cluster rectangle, click anywhere in the box and drag it to a new position. Tanzu Service Mesh automatically saves the new positions to your user preferences and retrieves them on any computer and device that you use.

The key health metrics are displayed for each service under its node in the graph. The RPS metric between two services is overlaid on the connection between the services. The total metric values for each cluster are displayed at the top of the cluster rectangle. To show only the metrics that you want in the graph, click **Graph Settings** and select the metrics to show under the service nodes and at the top of the cluster rectangles. You can select a maximum of four metrics to be displayed.

To show or hide the RPS metric on the service connections, under **Service Connections**, select or deselect **Show RPS**.

Note When you point to a service name in the graph, the key health metrics and details are displayed for that service in a card. The metric values in the card are displayed for the last 5 minutes.

The chart reflects the state of communication between the services for the time frame selected in the **Metric Time Range** drop-down menu in the upper-right corner of the window (for example, **Last 30 minutes** or **Last 7 days**). To display the topology chart for a different time frame, select the time frame from the **Metric Time Range** drop-down menu.


Note





- Tanzu Service Mesh collects and retains metrics data from the last 7 days. You can select an option from the **Metric Time Range** drop-down menu to see data from the last 5 minutes to up to the last 7 days. This is a global setting and applies not only to the data on the Home page but also to the data, graphs, and charts on the other pages in the Tanzu Service Mesh Console UI.
- If two services don't communicate during the time frame selected in the **Metric Time Range** drop-down menu, no line is shown between those services in the graph.
- To view the services that don't have connections with other services in the global namespace or cluster based on observed traffic, click **Unconnected Services** at the bottom of the window.


- 2 (Optional) To perform the following actions from the Topology Browser window, click these options and buttons in the window.

Option or Button	Action
Open GNS Details Page or Open Cluster Details Page	Open the details page for the global namespace or cluster.
Show drop-down menu in the upper-left corner	View all the services in the cluster or the services in a specific namespace by selecting All Services or the name of the namespace.
Show Service Versions or Hide Service Versions	Show or hide the versions of each service inside the service node.
Edit Configuration	Edit the global namespace configuration in the Edit Global Namespace window or edit the cluster details in the Edit Cluster window.

Option or Button	Action
More Actions	Click this button and then Delete Global Namespace or Remove Cluster to delete the global namespace or remove the cluster from Tanzu Service Mesh .

Option or Button	Action
	<p>Download the information about the service connections shown in the topology graph to a comma-separated values (CSV) file to make this information accessible to users with visual impairments.</p> <p>The header of the CSV file contains the name of your Tanzu Service Mesh organization, the name of the global namespace or of the cluster, and the time range applied to the topology graph. The file also contains the following columns:</p> <ul style="list-style-type: none"> ■ Services. Lists the services in the global namespace or the cluster. If you selected Show Service Versions in the Topology Browser window, this column has a heading of Service Versions and lists the services in the format <code>service, service version, cluster</code>. ■ Source Service Versions (incoming Requests). Contains the source services, that is, the services from which each service in the Services column receives traffic. The source services are listed in the format <code>(service A, cluster), (service B, cluster)</code>. If you selected Show Service Versions in the Topology Browser window, the source services are listed in the format <code>(service A, service version A version, cluster), (service B, service B version, cluster)</code>. ■ Destination Service Versions (Outgoing Requests). Contains the destination services, that is, the services to which each service in the Services column sends traffic. The destination services are listed in the format <code>(service A, cluster), (service B, cluster)</code>. If you selected Show Service Versions in the Topology Browser

Option or Button	Action
	<p>window, the destination services are listed in the format (service A, service version A version, cluster), (service B, service B version, cluster).</p> <hr/> <p>Note</p> <ul style="list-style-type: none"> ■ If you downloaded the service topology information for a cluster, the columns in the CSV file do not contain the cluster name as the last value. ■ You can also download the service topology information to a CSV file from a global namespace card or a cluster card on the GNS View tab or the Cluster View tab of the Home page. In the upper-right corner of the card, click the three vertical dots and then click Download Topology as CSV. ■ If not traffic is observed between the services in a global namespace or a cluster during a selected time frame, the service topology information is unavailable for download. In this case, the Download Topology button () in the Topology Browser window and the Download Topology as CSV option on the global namespace or cluster card are unavailable.
	Resize the topology graph so that it fits in the window.
	Zoom out on the graph.
	Zoom in on the graph.

- 3 To close the Topology Browser window, in the upper-left corner, click  or press Escape.

View the Proxy Configuration Settings

11

In the Tanzu Service Mesh Console UI, you can view the proxy configuration settings for a cluster that connects to Tanzu Service Mesh through a proxy server.

If, when onboarding a cluster, you specified that the cluster connects to Tanzu Service Mesh through a proxy server, you can view the proxy configuration settings that you provided in the Tanzu Service Mesh Console UI.

The configuration settings are displayed in these locations in the UI:

- The cluster details page. The **Proxy Connection** field at the top of this page displays the host name or IP address of the proxy server.
- The **Proxy Connection** column in the clusters table. The **Proxy Connection** column displays the host name or IP address of the proxy server.

This topic contains instructions on how to access each of these locations.

Prerequisites

- [Access the Tanzu Service Mesh Console.](#)

Procedure

- 1 To access the cluster details page:
 - a In the navigation panel on the left side, click **Home**.
 - b Click the **Cluster Overview** tab.
 - c At the top of the card of the cluster for which you want to view the proxy settings, click the cluster name.

The top of the cluster details page displays the **Proxy Connection** field.

2 To access the **Proxy Connection** column in the clusters table:

- a In the navigation panel on the left side, click **Inventory > Infrastructure**.
- b On the **Infrastructure** page, click the **Clusters** tab.
- c Locate the **Proxy Connection** column in the table.

Note If the **Proxy Connection** column is not visible, in the upper-right corner above the table, click **Column Settings**. Under **Show Columns**, select the check box next to **Proxy Connection**.

Configure the Export of API Audit Logs to Splunk

12

You can export API audit logs collected in Tanzu Service Mesh to Splunk Cloud Platform. To export the logs, you need to provide the information about the HTTP Event Collector (HEC) on Splunk Cloud Platform through the Tanzu Service Mesh API.

All calls that users make to the Tanzu Service Mesh APIs are logged for audit purposes. You can send the API audit logs to your Splunk Cloud Platform instance for analysis and visualization of the API events and for retrieving relevant data through search.

Note Currently, you can export API audit logs only to Splunk Cloud Platform using an HTTP Event Collector (HEC) input.

Tanzu Service Mesh sends the logs to a specified HTTP Event Collector (HEC) input on Splunk Cloud Platform. For Tanzu Service Mesh to connect to the HEC input, you must configure an external Splunk account through the Tanzu Service Mesh API and provide the following information about the HEC in the account configuration:

- HEC host - The URL that hosts the HEC.
- HEC port - The port configured for the HEC endpoint.
- HEC endpoint - The HEC endpoint to use. Typically, you use the `/services/collector/event` endpoint for JavaScript Object Notation (JSON)-formatted events or the `services/collector/raw` endpoint for raw events.
- HEC token - The token for Tanzu Service Mesh to use to authenticate connections to HEC. For more information about HEC tokens, see the [Splunk Cloud Platform documentation](#).

You must also create an external audit storage configuration for the Splunk account through the TSM API.

The following procedure describes how to provide the required HEC information to Tanzu Service Mesh through the API.

Prerequisites

- Deploy a Splunk Cloud Platform instance.

Note Export of API audit logs from Tanzu Service Mesh has been tested with Splunk Cloud version 9.0.2209.3.

- Secure your Splunk Cloud Platform instance with certificates. For more information, see the [Splunk documentation](#). You must use certificates signed by a trusted third-party certificate authority (CA).
-

Note

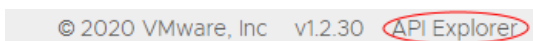
- Tanzu Service Mesh does not support Splunk configured with self-signed certificates.
-
- Update the Splunk configuration to use your certificates. The procedure below includes steps on how to update the Splunk configuration.
 - Configure an HEC input on Splunk Cloud Platform. The setup includes enabling HEC on Splunk Cloud Platform to allow use of HEC inputs. During the setup, ensure that you select **Enable SSL** for your HEC input because Tanzu Service Mesh supports export of logs over HTTPS only. Configure the remaining optional settings as necessary. Be sure to create an HEC token. For more information about configuring HEC on Splunk Cloud Platform, see the [Splunk documentation](#).
 - Verify the setup of HEC by sending data to HEC. For information about sending data to HEC, see the [Splunk documentation](#). Also see [Example of sending data to HEC with an HTTP request](#).
-

Note When testing sending data to HEC, use `https` and ensure that you do not allow insecure connections with the `-k` or `--insecure` argument when using a `curl` command.

- Know the URL of your Splunk Cloud Platform instance and the port on which it is accessible, and have the HEC token string ready.
- Verify that you are familiar with Splunk concepts and terminology. To become familiar with Splunk concepts and terminology, see the [Splunk Quick Reference Guide](#).

Procedure

- 1 To create an external account for Splunk through the API, perform these steps.
 - a On the bottom bar of the Tanzu Service Mesh Console UI, click **API Explorer**.



For information about how to access the Tanzu Service Mesh Console UI, see [Access the Tanzu Service Mesh Console](#).

- b On the **API Explorer** page, click the **Rest HTTP** tab.

- c Go to the **External Accounts** API, click the `PUT /v1alpha1/external-accounts/{id}` endpoint, and click **Try It Out**.
- d Under **Parameters**, enter an ID for the external Splunk account, for example, *splunkaccount*.

- e Under **Request Body**, provide the following parameters.

Parameter	Description
name	Enter the ID of the external Splunk account that you entered under Parameters , for example, <i>splunkaccount</i> .
description	Enter a description of the account.
provider	Set to <code>SPLUNK_ENT</code> .
provider_url	<p>Provide the URL of your Splunk Cloud Platform instance and the port on which it is accessible in the format <code>https://{host}:{port}/{hec-endpoint}</code>.</p> <ul style="list-style-type: none"> ■ <code>{host}</code> is the name of the Splunk Cloud Platform instance that runs HEC. ■ <code>{port}</code> is the HEC port number. ■ <code>{hec-endpoint}</code> is the HEC endpoint to use. The <code>services/collector/event</code> endpoint is typically used to collect JSON-formatted events. <p>Tanzu Service Mesh will use this parameter as the HEC host and port to connect to. An example of <code>provider_url</code>:</p> <pre>https://http-inputs.my-splunk-inst.splunkcloud.com:8088/services/collector/event</pre> <hr/> <p>Important Make sure that the URL uses the HTTPS protocol.</p>
authentication_type	Set to <code>TOKEN</code> .
auth_password (inside authentication)	This parameter is not needed. Delete it.
auth_token (inside authentication)	<p>Set <code>access_key</code> inside <code>auth_token</code> to the HEC token string.</p> <hr/> <p>Note</p> <ul style="list-style-type: none"> ■ <code>secret_access_key</code> is not needed, so delete it. ■ Make sure that <code>access_key</code> contains only the HEC token string, such as B5A79AAD-D822-46CC-80D1-819F80D7BFB0. Do not add Splunk or any spaces before the token string.
certificate_id	Set to an empty string (<code>""</code>).

- f Click **Execute**.

The API returns a 201 status code to indicate that the external Splunk account was created.

- 2 To create an external audit storage configuration for the Splunk account through the API, perform these steps.
 - a Go to the **External Audit Storage** API, click the `PUT /v1alpha1/external-audit/storage` endpoint, and click **Try It Out**.
 - b Under **Request Body**, provide the following parameters.

Parameter	Description
<code>infrastructure_account_id</code>	Set this parameter to the ID of the external Splunk account that you created in step 2.
<code>storage_type</code>	Set to <code>SPLUNK_ENT</code> .
<code>storage_config</code>	This parameter is not needed. Delete it from the request body.
<code>certificate_id</code>	Set to an empty string.

- c Click **Execute**.

The API returns a 200 status code to indicate that the external audit storage configuration was created and saved.

Results

Tanzu Service Mesh will send API audit logs to the specified HEC input on Splunk Cloud Platform.

Note

- If you no longer want to export API audit logs to your Splunk instance, delete the external Splunk account and the external audit storage configuration for Splunk through the API.

To delete the external Splunk account:

- 1 In the API Explorer, go to the **External Accounts** API.
- 2 Click the `DELETE /v1alpha1/external-accounts/{id}` endpoint and click **Try It Out**.
- 3 Provide the ID of the Splunk account and click **Execute**.

To delete the external audit storage configuration for Splunk:

- 1 Go to the **External Audit Storage** API.
- 2 Click the `DELETE /v1alpha1/external-audit/storage` endpoint and click **Try It Out**.
- 3 Click **Execute**.

- If you want to modify the external Splunk account (for example, you need to provide a different HEC token in the account), delete the existing Splunk account and its external audit storage configuration and create a new account with the parameters you want. For instructions on deleting a Splunk account and its external audit storage configuration, see the first note above. For instructions on creating an external Splunk account, see steps 1 and 2 in the procedure.

GitOps Workflow with Tanzu Service Mesh

13

GitOps is an operational framework that uses Git repositories as a single source of truth. Under GitOps, you describe the desired state of a Tanzu Service Mesh configuration using a declarative specification and place it in a Git repository.

In Git, you can define role-based access control (RBAC), version control, governance, audit trail, and any other operations that are required. A Git audit trail is totally acceptable for the organization's auditing purposes.

After changes to the configuration are made, approved, and merged, CI/CD pipelines are commonly triggered to apply them to the infrastructure.

GitOps automation overwrites any configuration drift caused by manual local changes and errors. As a result, the environment always uses the desired state defined in Git.

You can make Tanzu Service Mesh an integral part of your organization's GitOps workflow to deliver Git-central changes and updates to the Tanzu Service Mesh configuration and ensure the convergence of configuration between the Git repository and the clusters, by placing our declarative specifications in Git and applying them using the Tanzu Service Mesh CLI.

Read the following topics next:

- [Integrating Tanzu Service Mesh into a GitOps workflow](#)
- [Keeping Git as the Source of Truth for Configuration](#)
- [Managing Declarative Manifests](#)

Integrating Tanzu Service Mesh into a GitOps workflow

The following high-level steps provide general guidance on how to integrate Tanzu Service Mesh into the GitOps workflow that your organization uses. Adapt this guidance to fit your organization's business needs.

Tanzu Service Mesh provides a CLI that your organization can use to integrate into the GitOps workflow. You can use the Tanzu Service Mesh CLI to apply Tanzu Service Mesh feature and policy configurations (such as global namespace and access control policy configurations) in a declarative way to Tanzu Service Mesh SaaS by using YAML manifests.

Procedure

- 1 Set up a dedicated Git repository to use as a single source of truth for your infrastructure configuration.

- 2 To describe the appropriate configurations, create declarative manifest files.

For information about creating declarative manifest files, see [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#) and [Use an Existing Object or Policy Configuration to Create a Manifest](#).

- 3 Place the manifest files in the Git repository. To control access to the files, consider setting up role-based access control (RBAC) on the repository.

Also consider setting up an appropriate approval workflow and appropriate audits in the repository.

- 4 Clone the Git repository that contains the manifest files to a local folder.

- 5 [Install the CLI](#).

- 6 To apply feature and policy configurations from the manifest files to Tanzu Service Mesh SaaS, run the CLI.

The Tanzu Service Mesh CLI interacts with the Tanzu Service Mesh API gateway in the Tanzu Service Mesh SaaS. The CLI sends the manifest files to API Gateway, a single entry point into the Tanzu Service Mesh API.

The API gateway receives the manifest files and applies them to the customer's tenant in Tanzu Service Mesh. A manifest file includes criteria that specify clusters and namespaces to which the configuration applies. As a result, objects (such as global namespaces) or policies (such as access control policies) get created or updated in the clusters based on the configuration defined in the manifest.

For more information about using the Tanzu Service Mesh CLI, see [Common CLI Tasks](#).

Note If you use a CI/CD pipeline, make sure that you supply the Tanzu Service Mesh CLI to the pipeline runner to automatically apply configurations to or delete configurations from Tanzu Service Mesh SaaS.

Keeping Git as the Source of Truth for Configuration

The configuration in your Tanzu Service Mesh tenant can drift from the desired configuration state defined in the Git repository if someone goes directly to the Tanzu Service Mesh tenant and makes changes. This can cause the Git configuration and the tenant configuration to diverge. You need to implement a mechanism to ensure that the Git repository remains the source of truth for the configuration state.

One option is to create a cron job that runs a script at regular intervals (for example, every 10 minutes). The script will re-apply the manifest files from the Git repository and apply the configurations from the files to Tanzu Service Mesh SaaS at the configured interval. This approach will help overwrite the divergences and automatically drive the configuration state in Git.

Note Changes made to the configuration in Tanzu Service Mesh SaaS do not affect the manifest files stored in the Git repository. Because Git is the source of truth, the described convergence mechanism will be continuously applying changes from Git to the tenant in SaaS to keep them in sync with the configuration state in the Git repository.

Make sure that the cron job is running on a machine that can access your repository and Tanzu Service Mesh SaaS.

Managing Declarative Manifests

To describe Tanzu Service Mesh object and policy configurations, such as global namespaces and access control policies, you need to create appropriate declarative manifest files and maintain the files in your Git repository. You can then use the Tanzu Service Mesh to apply manifest files to the configuration in your Tanzu Service Mesh tenant.

For more information about creating declarative manifest files, see [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#) and [Use an Existing Object or Policy Configuration to Create a Manifest](#).

You can use the Tanzu Service Mesh CLI to apply changes to the configuration in your Tanzu Service Mesh tenant by using declarative manifest files.

The Tanzu Service Mesh CLI sends manifest files to Tanzu Service Mesh SaaS, and Tanzu Service Mesh SaaS then applies the configurations from the manifest files to your tenant based on the cluster and namespace selection criteria in the manifests.

The Tanzu Service Mesh CLI is integrated into the VMware Tanzu command-line interface (Tanzu CLI) as a plugin. With the Tanzu CLI, you can perform different tasks for different Tanzu products, including Tanzu Service Mesh. To use the Tanzu Service Mesh CLI, you must install the Tanzu CLI and then download and install the Tanzu CLI plugin for Tanzu Service Mesh. For more information about the Tanzu CLI, see the [VMware Tanzu CLI Documentation](#). For instructions on installing the Tanzu CLI plugin for Tanzu Service Mesh, see [Install the CLI](#).

Using Tanzu Service Mesh CLI, you can do the following things:

- Create new objects (for example, new global namespaces) and policies (for example, access control policies) in Tanzu Service Mesh.
- Update the configurations of existing objects and policies in Tanzu Service Mesh.
- Remove objects and policies from Tanzu Service Mesh.
- Query existing objects in Tanzu Service Mesh SaaS for their configuration.
- Get the specification of an API as a reference for a manifest.

You can run the CLI manually. You can also configure a CI/CD pipeline to run the CLI automatically. Whenever changes made to the declarative manifest files are merged in the Git repository, a pipeline can trigger the CLI to apply the configuration changes to Tanzu Service Mesh.

For overview information about how the Tanzu Service Mesh CLI fits in an organization's GitOps workflow, see [Integrating Tanzu Service Mesh into a GitOps workflow](#).

For information about how to apply a configuration from a manifest file to your tenant, install the CLI, and how to perform other tasks with the CLI, see [Common CLI Tasks](#).

Prerequisites

Before you start using the CLI, verify that the following prerequisites are met:

- [Install the Tanzu CLI.](#)
- [Install the CLI.](#)
- Create declarative manifests. For more information, see [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#) and [Use an Existing Object or Policy Configuration to Create a Manifest](#).
- Make sure that the machine on which the CLI is installed can access the local folder that contains the manifest (YAML) files and can access Tanzu Service Mesh SaaS.
- If you want a CI/CD pipeline to run the CLI automatically on changes to the manifest files, make sure that you supply the Tanzu Service Mesh CLI to the pipeline runner.

Read the following topics next:

- [Common CLI Tasks](#)

Common CLI Tasks

Find instructions on how to perform common tasks using the Tanzu Service Mesh CLI.

Install the CLI

The Tanzu Service Mesh CLI is integrated with the VMware Tanzu command-line interface (Tanzu CLI) as a plugin. You use the Tanzu Service Mesh plugin to perform the Tanzu Service Mesh CLI commands from the Tanzu CLI.

You can use the Tanzu CLI to perform tasks to create and manage your VMware Tanzu infrastructure. With the Tanzu CLI, you can download plugins that provide commands that perform different functions for different Tanzu products, including Tanzu Service Mesh. A Tanzu CLI *plugin* is an executable binary that packages a group of CLI commands. For more information about the Tanzu CLI, see the [VMware Tanzu CLI Documentation](#).

To be able to use the Tanzu Service Mesh CLI commands from the Tanzu CLI, you must install the Tanzu Service Mesh plugin.

Prerequisites

- [Install the Tanzu CLI.](#)

Procedure

- 1 Run the following command.

```
tanzu plugin install service-mesh
```

This command installs the latest version of the plugin. To install a specific version of the plugin, specify the version with the `version` flag. For example, to install version 1.0.0 of the plugin, run:

```
tanzu plugin install service-mesh --version 1.0.0
```

The Tanzu Service Mesh plugin is downloaded from the Tanzu CLI plugins repository and is installed.

- 2 To verify that you installed the Tanzu Service Mesh plugin successfully, run the following command.

```
tanzu plugin list
```

The list of plugins in the output includes the Tanzu Service Mesh plugin and shows the version of the plugin.

What to do next

To start using the Tanzu Service Mesh CLI, you need to [Log in to the Tanzu Service Mesh CLI](#).

To get Help on all the available CLI commands, run `tanzu sm -h`.

Log in to the Tanzu Service Mesh CLI

To start using the Tanzu Service Mesh CLI, you must log in to the SaaS endpoint on which you want to perform actions.

Prerequisites

- [Install the CLI](#).
- The Tanzu Service Mesh CLI interacts with Tanzu Service Mesh SaaS. Make sure that you know the URL of the Tanzu Service Mesh SaaS server.
- In VMware Cloud Services, generate an API token to authenticate with the Tanzu Service Mesh SaaS server. For instructions on generating an API token, see step 1 in [Authentication with the Tanzu Service Mesh REST API](#).

Procedure

- ◆ On the machine where the Tanzu Service Mesh is installed, open a terminal window and run the following CLI command.

```
tanzu sm login -s {SaaS server host} -t {API token}
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{SaaS server host}` is the host name in the URL of the Tanzu Service Mesh SaaS server, and `{API token}` is the generated API token.

For example, if the URL of the Tanzu Service Mesh SaaS server is `https://my-tsm-prod.servicemesh.biz`, provide the host name `my-tsm-prod.servicemesh.biz` in the command.

Results

You can now run other CLI commands. To get a list of the available commands, run `tanzu sm -h`.

Get a List of CLI Commands and Get Help on a Command

You can get a list of the commands in the Tanzu Service Mesh CLI and get Help on a specific command.

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)

Procedure

- 1 To get a list of the commands in the CLI and the flags available for all the commands, run the following CLI command.

```
tanzu sm -h
```

Where `sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. You can replace the `-h` flag with `--help`.

- 2 To get Help on a command, run the following command.

```
tanzu sm {command} -h
```

To get Help on the `apply` command, run the following command.

```
tanzu sm apply -h
```

Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification

You can run a CLI command to retrieve the corresponding API specification from Tanzu Service Mesh SaaS. Based on the API specification, you can describe a configuration that creates a Tanzu Service Mesh object, such as a global namespace, or a policy, such as an access control policy, in a declarative manifest file.

The API specifications that you retrieve with the CLI are in the Kubernetes YAML manifest format. The CLI command that you use to retrieve an API specification from Tanzu Service Mesh SaaS requires that you specify the short name or the long name of the API. The following table identifies the available APIs and lists the long and short names of each API.

Table 14-1. Available APIs

API long name	API short name
accesscontrolpolicies.gns.tsm.vmware.com	accesscontrolpolicies
apidiscoveries.gns.tsm.vmware.com	apidiscoveries
attackdiscoveries.gns.tsm.vmware.com	attackdiscoveries
autoscalingpolicies.autoscaling.tsm.vmware.com	autoscalingconfigs
certificateauthorities.ca.tsm.vmware.com	certificateauthorities
certificates.certificates.tsm.vmware.com	certificates
certificatestatuses.certificates.tsm.vmware.com	certificatestatuses
clusterappses.clusters.tsm.vmware.com	clusterapps
clusteronboardingmanifests.clusters.tsm.vmware.com	clusteronboardingmanifests
clusteronboardingurls.clusters.tsm.vmware.com	clusteronboardingurls
clusteronboardurls.clusters.tsm.vmware.com	clusteronboardurls
clusters.clusters.tsm.vmware.com	clusters
externalaccounts.externalaccounts.tsm.vmware.com	externalaccount
externalaccountstatuses.externalaccounts.tsm.vmware.com	externalaccountstatuses
externaldnsdelegateddomains.externaldns.tsm.vmware.com	
externaldnsdomainses.externaldns.tsm.vmware.com	externaldnsdelegateddomains
externaldnss.externaldns.tsm.vmware.com	externaldnss
externaldnsstatuses.externaldns.tsm.vmware.com	externaldnsstatuses
externalservices.gns.tsm.vmware.com	externalservices
geodiscoveries.gns.tsm.vmware.com	geodiscovery
globalnamespacecapabilities.gns.tsm.vmware.com	globalnamespacecapabilities
globalnamespacememberses.gns.tsm.vmware.com	globalnamespacememberses
globalnamespaces.gns.tsm.vmware.com	globalnamespace
gnsroutingpolicies.gns.tsm.vmware.com	
gnsservicegroupmembers.gns.tsm.vmware.com	gnsservicegroupmembers
gnsservicegroups.gns.tsm.vmware.com	gnsservicegroup
gnsservicelevelobjectives.gns.tsm.vmware.com	gnsservicelevelobjective

Table 14-1. Available APIs (continued)

API long name	API short name
healthchecks.templates.tsm.vmware.com	healthchecks
jobdownloads.jobs.tsm.vmware.com	jobdownloads
jobs.jobs.tsm.vmware.com	jobs
jwkses.certificates.tsm.vmware.com	jwkses
piidiscoveries.gns.tsm.vmware.com	piidiscoveries
policyconfigs.acp.tsm.vmware.com	policyconfigs
projects.project.tsm.vmware.com	projects
publicservicedomainses.gns.tsm.vmware.com	publicservicedomainses
publicserviceroutes.gns.tsm.vmware.com	publicserviceroutes
publicserviceroutestatuses.gns.tsm.vmware.com	publicservicestatuses
publicservices.gns.tsm.vmware.com	publicservices
resourcegroupdetaileddlists.resourcegroups.tsm.vmware.com	resourcegroupdetaileddlists
resourcegroupmembers.resourcegroups.tsm.vmware.com	resourcegroupmembers
resourcegroups.resourcegroups.tsm.vmware.com	resourcegroup
schemaviolationdiscoveries.gns.tsm.vmware.com	schemaviolationdiscoveries
segmentationpolicies.gns.tsm.vmware.com	segmentationpolicies
servicelevelobjectives.slo.tsm.vmware.com	servicelevelobjectives
sharedservices.gns.tsm.vmware.com	sharedservice
supportedexternalaccounts.externalaccounts.tsm.vmware.com	supported-external-accounts
trafficroutingpolicies.gns.tsm.vmware.com	trafficroutingpolicy
userdiscoveries.gns.tsm.vmware.com	userdiscoveries

Note Some of the APIs don't have a short name.

To complete the configuration, you must save the specification to a YAML manifest file and then provide values for the fields in the manifest file.

Important To define a configuration for a global namespace or for an access control policy, you must combine more than one API specification in a declarative manifest file. For more information about creating a global namespace manifest file and an access control policy manifest file, see [Create a Global Namespace](#) and [Create an Access Control Policy](#).

Note You can also create a manifest based on the configuration of an existing object or an existing policy. For more information, see [Use an Existing Object or Policy Configuration to Create a Manifest](#).

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)
- Be familiar with the Kubernetes YAML manifest format.

Procedure

- 1 Run the following CLI command.

```
tanzu sm get spec {API}
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{API}` is the API short name or long name. Find the short or long name of the API you want in the Available APIs table above.

To retrieve the API specification for a global namespace, you can run the following command. In this example, the command includes the short name of the API.

```
tanzu sm get spec globalnamespaces
```

The CLI returns the API specification in the Kubernetes YAML manifest format. For the example above, the CLI returns the following output.

```
apiVersion: gns.tsm.vmware.com/v1
kind: GlobalNamespace
metadata:
  labels:
    projectId: string
    name: string
spec:
  api_discovery_enabled: true
  ca: string
  ca_type: PreExistingCA
  color: string
  description: string
  display_name: string
  domain_name: string
```

```

match_conditions:
  - cluster:
      match: string
      type: string
    namespace:
      match: string
      type: string
mtls_enforced: true
name: string
use_shared_gateway: true
version: string

```

- 2 Copy the API specification to a YAML file.
- 3 In the file, enter appropriate values for the fields in the configuration.

Note For information about the meaning of the fields in the configuration and what value to provide for each field, see the schema of the appropriate API in API Explorer in Tanzu Service Mesh. Perform the following steps:

- a In Tanzu Service Mesh SaaS UI, click **API Explorer** on the bottom bar.
 - b Go to the appropriate API in API Explorer.
 - c In the API section, under **Request Body**, click **Schema**.
-

- 4 Save the changes in the file.

What to do next

Add the manifest file to the Git repository where you maintain all your manifest files.

Use an Existing Object or Policy Configuration to Create a Manifest

Using the Tanzu Service Mesh CLI, you can retrieve the configuration of an existing object or a policy and create another manifest based on the existing configuration.

For example, you can use the Tanzu Service Mesh CLI to retrieve the configuration of an existing global namespace and then create a manifest describing the configuration of another global namespace, based on that existing global namespace configuration.

Note You can also create a manifest based on an appropriate Tanzu Service Mesh API specification. For more information, see [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#).

Prerequisites

- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).
- Be familiar with the Kubernetes YAML manifest format.

Procedure

1 Run the following CLI command.

```
tanzu sm get {API} {object or policy name}
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{API}` is the short or long name of the appropriate API. For example, if you want to create a global namespace, provide the name of the global namespace API. For a list of the availables API and their names, see the Available APIs table in [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#). `{object or policy name}` is the name of the existing object or policy whose configuration you want to reuse to create a manifest.

For example, suppose you want to create a manifest of a global namespace based on the configuration of an existing global namespace called *acme*. To retrieve the configuration of the *acme* global namespace, you run the following command.

```
tanzu sm get globalnamespaces acme
```

Where `globalnamespaces` is the short name of the global namespace API, and *acme* is the name of the existing global namespace whose configuration we want to reuse.

The CLI returns the configuration of the specified object or policy in YAML format. In the example above, the CLI returns the configuration of the *acme* global namespace.

```
api_discovery_enabled: true
ca: default
ca_type: PreExistingCA
color: ""
description: ""
display_name: acme
domain_name: acme.lab
match_conditions:
- cluster:
    match: tsm
    type: START_WITH
  namespace:
    match: acme
    type: EXACT
mtls_enforced: false
name: acme
use_shared_gateway: true
version: "2.0"
```

2 Copy the configuration to a YAML file.

- 3 In the YAML file, enter appropriate values for the fields in the configuration.

For information about the meaning of the fields in the configuration and what value to provide for each field, see the schema of the appropriate API in API Explorer in Tanzu Service Mesh. Perform the following steps:

- a In Tanzu Service Mesh SaaS UI, click **API Explorer** on the bottom bar.
- b Go to the appropriate API in API Explorer.
- c In the API section, under **Request Body**, click **Schema**.

- 4 Save the changes in the file.

What to do next

Add the manifest file to the Git repository where you maintain all your manifest files.

Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI

Using the Tanzu Service Mesh CLI, you can apply the configuration from a declarative manifest file to your Tanzu Service Mesh tenant. After the configuration is applied, a defined object, such as a global namespace, or a defined policy, such as an access control policy, is created or updated on the clusters that match the selection criteria in the configuration.

To apply a manifest file, the Tanzu Service Mesh CLI sends it to API Gateway, a single entry point into the Tanzu Service Mesh API in Tanzu Service Mesh SaaS.

API Gateway receives the manifest file and applies the configuration to those of your clusters that match the cluster and namespace selection criteria in the manifest file. If an object (such as a global namespace) or a policy (such as an access control policy) with the name defined in the configuration does not exist in a cluster, that object or policy gets created. If an object or a policy with this name exists in a cluster, the object or policy gets updated.

Note

- This procedure provides instructions on how to apply a manifest manually. If you want the CLI to run automatically whenever changes to the manifest files are merged in the Git repo, set up a CI/CD pipeline. Make sure that you supply the Tanzu Service Mesh CLI to the pipeline runner. On changes to the manifest files, the pipeline will automatically trigger the CLI to apply the configuration changes to your Tanzu Service Mesh tenant.
 - To recreate your entire Tanzu Service Mesh environment, apply all the manifest files (global namespace and policy configurations) at once to Tanzu Service Mesh SaaS.
-

Prerequisites

- [Install the CLI.](#)
[Log in to the Tanzu Service Mesh CLI.](#)
- Create appropriate manifest YAML files. For more information about creating manifest files, see [Create a Declarative Manifest Based on a Tanzu Service Mesh API Specification](#).

- Make sure that the Tanzu Service Mesh CLI can access your declarative manifest files in the local folder where you cloned them from the Git repository.
- Make sure that the machine where the CLI is installed can access Tanzu Service Mesh SaaS.

Procedure

- 1 In a terminal window, change to the local folder where you cloned the manifest files from the Git repository.
- 2 To apply the manifest YAML file that you want, run the following CLI command.

```
tanzu sm apply -f {manifest YAML file}
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{manifest YAML file}` is the name of the manifest YAML file.

To apply a manifest file with a name of `my-global-namespace-configuration`, run the following command.

```
tanzu sm apply -f my-global-namespace-configuration.yaml
```

Results

When the object or policy is created as a result of applying the configuration, the command returns `{object} created` or `{policy} created` in the output.

When a global namespace configuration is applied, the command returns several `created` lines for the different configurations that make up the global namespace.

Note If the configuration from the manifest file was not failed because of an error, the command returns failure. To determine the exact error, run the `tanzu sm apply` command with the `debug` flag. For more information, see [Debug Problems with a Manifest File](#).

Onboard a Cluster Using the CLI

In addition to onboarding a cluster from the Tanzu Service Mesh UI and through the API, you can onboard a cluster using the Tanzu CLI. Onboarding involves registering a cluster with Tanzu Service Mesh and installing the necessary software components on the cluster.

Onboarding a cluster involves registering the cluster with Tanzu Service Mesh. Before you begin onboarding the cluster, you must prepare a registration YAML file to describe the cluster configuration required for the registration. During onboarding, to register the cluster with Tanzu Service Mesh, you must apply the configuration from the registration YAML file to the cluster.

Using the registration YAML file, you can specify a cluster identifier and a cluster name and configure different onboarding options. For example, if you want the cluster to connect to Tanzu Service Mesh through your organization's web proxy server, you can describe the proxy settings in the YAML file. If you want to enable automatic Istio sidecar injection for some of the namespaces in the cluster, you can define appropriate namespace inclusion rules in the file. For more information about the onboarding process and these and other onboarding options, see [Onboard a Cluster](#) in the *Getting Started with VMware Tanzu Service Mesh Guide*.

To create a registration YAML file, you must retrieve the cluster registration specification from the Tanzu Service Mesh API and then provide values for the fields in the specification. The following procedure includes instructions on how to create a registration YAML file.

Prerequisites

- Verify that your environment meets the requirements listed in [Tanzu Service Mesh Environment Requirements and Supported Platforms](#).
- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).
- [Install jq](#), a command-line JSON processor, and [install yq](#), a command-line YAML, JSON and XML processor. Some of the commands included in the following procedure use jq and yq.

Procedure

- 1 To create a registration YAML file, perform the following steps.
 - a To retrieve the cluster registration specification from the Tanzu Service Mesh API, run the following command.

```
tanzu sm get spec clusters
```

The output contains the following specification.

```
apiVersion: clusters.tsm.vmware.com/v1
kind: Cluster
metadata:
  labels:
    projectId: string
    name: string
spec:
  autoInstallServiceMesh: true
  autoInstallServiceMeshConfig:
    restrictDefaultExternalAccess: true
  caLabels:
  - key: string
    value: string
  description: string
  displayName: string
  enableInternalGateway: true
  enableNamespaceExclusions: true
  enableNamespaceInclusions: true
  labels:
  - key: string
    value: string
  namespaceExclusions:
  - match: string
    type: string
  namespaceInclusions:
  - match: string
    type: string
  proxyConfig:
    certificate: string
    host: string
    password: string
    port: 1.2
    protocol: HTTP
    proxy: Explicit
```

```
username: string
registryAccount: string
tags:
- string
```

b Copy the specification into a YAML file and provide values for the fields.

- Set `name` to the ID of the cluster. The cluster ID is a required parameter in making certain calls to the Tanzu Service Mesh API. In the Tanzu Service Mesh Console UI, the cluster ID appears on the cluster details page. After you onboard the cluster, you cannot change the cluster ID. To help identify the cluster in Tanzu Service Mesh, use a friendly cluster ID.
- Set `displayName` to the display name for the cluster, that is, the name that you want your cluster to have in the Tanzu Service Mesh Console. The cluster display name can be the same as the cluster ID or can be different. The cluster display name can include only lowercase letters and cannot contain special characters, such as a number sign (#), at sign (@), apostrophe ('), underscore (_), and uppercase letters. To help identify the cluster in Tanzu Service Mesh, use a friendly display name.
- Set `projectId` to **default**.
- You don't need to include all the fields from the specification in your registration YAML file. Some of the fields are optional. For information about the meaning of the fields in the specification, which fields are mandatory and which are optional, and what value to provide for each field, see the schema of the `/v1alpha2/{projectId}/default/clusters/{clusterId}` API in API Explorer in Tanzu Service Mesh. Perform the following steps.

a [Access the Tanzu Service Mesh Console](#).

b In Tanzu Service Mesh Console, click **API Explorer** on the bottom bar.

c In the API Explorer, expand `PUT /v1alpha2/{projectId}/default/clusters/{clusterId}`.

d Under **Request Body**, click **Schema**.

The mandatory fields are marked with a red asterisk (*).

- You can onboard more than one cluster, using a registration YAML file. Make sure that you put `---` at the end of the configuration for each cluster in the file.
- See an example of a registration YAML file below.

This example of a registration YAML file shows a cluster registration configuration with the minimum required fields.

```
apiVersion: clusters.tsm.vmware.com/v1
kind: Cluster
metadata:
  labels:
    projectId: default
```

```

name: my-sample-cluster
spec:
  displayName: my-sample-cluster
  description: ''
  tags: []
  labels: []
  caLabels:
    - key: CertificateAuthority
      value: TSM
  namespaceExclusions: []
  autoInstallServiceMesh: true
  enableNamespaceExclusions: true
  enableInternalGateway: false

```

c Save the changes in the file.

- 2 To generate a security token for use during onboarding, run this command.

```
token=`tanzu sm apply -f {cluster-registration-file-name}.yaml | jq -r .token`
```

This security token is used to establish a secure connection between Tanzu Service Mesh and your cluster during onboarding.

`{cluster-registration-file-name}.yaml` is the name of the registration YAML file that you created in step 1.

The command applies the registration YAML file, generates a security token, extracts the token from the output, and stores it in a `token` variable.

- 3 To generate the URL of the Tanzu Service Mesh SaaS server where the cluster will be onboarded (the onboarding URL), run the following command.

```
url=`tanzu sm get clusteronboardurls | yq .url`
```

The command generates the onboarding URL for the cluster, extracts the URL from the output, and stores the URL in a `url` variable.

- 4 To apply the configuration from the registration YAML file to the cluster, run the following command.

```
kubect1 apply -f "$url"
```

- 5 To create a Kubernetes secret to store the security token and then use the security token from the secret to establish a secure connection between the cluster and Tanzu Service Mesh during onboarding, run the following command.

```
kubect1 -n vmware-system-tsm create secret generic cluster-token --from-literal=token=$token
```

Results

The cluster is registered with the specified Tanzu Service Mesh SaaS server based on the onboarding URL, and the Tanzu Service Mesh software components are installed on the cluster. The onboarding takes a few minutes to complete.

To verify that the cluster has been successfully onboarded, run the following command.

```
tanzu sm get cluster {cluster-id} | yq .status
```

If the `message` field in the output is `Tanzu Service Mesh installed` and the `state` field is `Ready`, the cluster has been successfully onboarded.

Note As the onboarding progresses, the output of the command shows the current onboarding state. Repeat the command until you see the `Ready` state in the output.

Remove a Cluster from Tanzu Service Mesh Using the CLI

Removing a cluster from Tanzu Service Mesh involves unregistering the cluster with its Tanzu Service Mesh SaaS server and uninstalling the Tanzu Service Mesh software components on the cluster. You can remove a cluster from Tanzu Service Mesh using the Tanzu Service Mesh CLI.

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)
- [Install `yq`](#), a command-line YAML, JSON and XML processor. One of the commands included in the following procedure uses `yq`.

Procedure

- 1 To delete the registration YAML applied to the cluster during onboarding and thus unregister the cluster from Tanzu Service Mesh, run the following command.

```
tanzu sm delete -f {cluster-registration-file-name}.yaml
```

`{cluster-registration-file-name}.yaml` is the name of the registration YAML file.

The output contains a message that the cluster was deleted.

- 2 To verify that the cluster was successfully unregistered from Tanzu Service Mesh, run the following command.

```
tanzu sm get cluster {cluster-id} | yq .status
```

`null` in the output of the command indicates the cluster was successfully unregistered.

Note Because unregistering the cluster takes a few minutes, run the command until you see `null` in the output.

- 3 To remove the Tanzu Service Mesh components from the cluster, run the following command.

```
kubectl delete --ignore-not-found=true -f https://{SaaS-server-host-name}/cluster-
registration/k8s/client-cluster-uninstall.yaml
```

{SaaS-server-host-name} is the host name in the URL of the Tanzu Service Mesh server with which the cluster was registered during onboarding.

For example, if the cluster was registered with a server at `https://my-tsm-prod.servicemesh.biz`, run:

```
kubectl delete --ignore-not-found=true -f https://my-tsm-prod.servicemesh.biz/cluster-
registration/k8s/client-cluster-uninstall.yaml
```

Create a Global Namespace

To create a global namespace with the Tanzu Service Mesh CLI, you must describe the configuration of the global namespace in a YAML file and then apply the configuration file.

You must describe the different configuration details of a global namespace in the following manifests:

- Global namespace manifest. Contains the general details and namespace selection criteria about the global namespace.
- Public service manifest. Contains general configuration details about a public service in the global namespace. For a GSLB-enabled public service, the public service manifest also contains the GSLB configuration parameters.
- Public service route manifest. Defines the name of the internal service associated with the public service and the port of the internal service.
- API discovery manifest. Contains the configuration details that are required to observe and monitor API traffic between the services in the GNS and configure API security policies.
- PII discovery manifest. Contains the configuration details that are required to observe and monitor data in flight between services in the GNS and configure data security policies.

You must retrieve the templates of these manifests from the Tanzu Service Mesh API, combine them in a single YAML manifest file called *global namespace configuration file*, and provide values for the fields in the manifests.

Although the simplest approach is to have all of these manifests combined in a single file, it is not necessary. For example, if a different team manages the GSLB interaction, you can separate the public service GSLB configuration into a separate manifest file. However, all of these YAML manifest files must exist for a global namespace to function properly.

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)

- Be familiar with the concept of [global namespace](#) in Tanzu Service Mesh.
- Be familiar with the concept of [public service](#) in Tanzu Service Mesh.
- Be familiar with the concept of [API discovery](#) in Tanzu Service Mesh.
- Be familiar with the concept of [personal identifiable information \(PII\) discovery](#) in Tanzu Service Mesh.
- Be familiar with the Kubernetes YAML manifest format.
- The current public service configuration requires that a health check ID and an external DNS ID exist and be referred to in the PublicService manifest, which is part of the GlobalNamespace manifest. Before applying the public service, you must add the health check ID and the DNS ID to the PublicService manifest. For information on how to get a health check ID and an external DNS ID, see [Get a Health Check ID for a Public Namespace Configuration](#) and [Get an External DNS ID for a Public Service Configuration](#).

Procedure

- 1 Create a YAML manifest file to contain the global namespace configuration.

The steps below refer to this file as the *global namespace configuration file*.

- 2 To retrieve the template for each manifest that is included in the global namespace configuration, run the `tanzu sm get spec` CLI command.

- a To retrieve the global namespace manifest template, run the following command:

```
tanzu sm get spec globalnamespaces
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`.

- b To retrieve the public service manifest template, run the following command.

```
tanzu sm get spec publicservices
```

- c To retrieve the template for a public service route manifest, run the following command.

```
tanzu sm get spec publicserviceroutes
```

- d To retrieve the API discovery manifest template, run the following command:

```
tanzu sm get spec apidiscoveries
```

- e To retrieve the PII discovery manifest template, run the following command:

```
tanzu sm get spec piidiscoveries
```

- 3 Paste each returned manifest template into the global namespace configuration file that you created.

Important

- Arrange the manifests in the following order in the file:
 - Global namespace manifest
 - Public service manifest
 - Public service route manifest
 - API discovery manifest
 - PII discovery manifest
 - Make sure that you put `---` at the end of each manifest in the file.
-

- 4 Provide values for the fields in the resultant global namespace configuration file.
 - Add the health check ID to the public service manifest under `healthcheck_ids`.
 - Add the external DNS ID to the public service manifest as the value of `external_dns_id` under `public_domain`.
-

Note For information about the meaning of the fields in the configuration and what value to provide for each field, see the schema of the appropriate API in API Explorer in Tanzu Service Mesh. Perform the following steps:

- a In Tanzu Service Mesh SaaS UI, click **API Explorer** on the bottom bar.
 - b Go to the appropriate API in API Explorer.
 - c In the API section, under **Request Body**, click **Schema**.
-

- 5 Save the changes in the file.

Results

For information about how to apply the global namespace configuration YAML file to your Tanzu Service Mesh tenant, see [Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI](#).

What to do next

Add the global namespace configuration file to the Git repository where you maintain all your manifest files.

Get a Health Check ID for a Public Namespace Configuration

The public service API specification in a global namespace manifest file requires the ID of a health check.

To get a health-check ID, you can retrieve the API specification that corresponds to the health-check configuration from Tanzu Service Mesh SaaS, add the configuration to a manifest file, then provide values for the fields in the manifest file, and finally apply the manifest using the Tanzu Service Mesh.

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)
- [Access the Tanzu Service Mesh Console.](#)
- Be familiar with the concept of [health checks](#) in Tanzu Service Mesh.
- Be familiar with the Kubernetes YAML manifest format.

Procedure

- 1 To describe the health-check configuration, create a YAML manifest file.
- 2 To retrieve the API specification that corresponds to a health-check configuration, run the following command.

```
tanzu sm get spec healthchecks
```

- 3 Copy and paste the returned API specification into the manifest file.
- 4 Open the file for editing and then provide values for the fields.

For information about the meaning of the fields in the manifest and what value to provide for each field, see the schema of the `health-checks` API in API Explorer in Tanzu Service Mesh.

Perform the following steps:

- a In Tanzu Service Mesh SaaS UI, click **API Explorer** on the bottom bar.
- b In API Explorer, click `POST /v1alpha1/templates/health-checks`.
- c In the API, under **Request Body**, click **Schema**.

An example of a health check manifest with values.

```
metadata:
  project: default
spec:
  name: my-tsm.servicemesh.biz
  protocol: HTTP
  domain: my-tsm.servicemesh.biz
  port: 3000
  path: /
  healthThreshold: 3
  certificate_id: ''
  external_port: 80
  interval: 10
```

- 5 Save the changes in the file.

6 Run the following CLI command.

```
tanzu sm apply -f {manifest file}
```

Where `{manifest file}` is the name of the manifest file that describes the health-check configuration.

The command returns the health check ID.

What to do next

Add the health check ID to the public service API specification in the global namespace manifest file. For more information, see [Create a Global Namespace](#).

Create an Access Control Policy

Using the Tanzu Service Mesh CLI, you can retrieve all the API specifications that make up an access control policy configuration from Tanzu Service Mesh SaaS. You can combine these specifications in a declarative manifest file to describe a complete access control policy configuration.

A complete access control policy configuration consists of the following configurations:

- Service group configurations. You must provide a configuration for each service group that you want to define as a source service group or as a destination service group in the access control policy. For more information about the service group configuration, see [Create a Service Group](#).
- Access control policy configuration. You must provide the general policy details (its name, scope, and optionally labels) and the definitions of the source and destination service groups in the policy. For more information about the access control policy configuration, see [Access Control Policy: UI Configuration](#).

You must retrieve the API specifications that correspond to these configurations from Tanzu Service Mesh SaaS and combine them in the access control policy manifest file.

Prerequisites

- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).
- Be familiar with the concept of [service group](#) in Tanzu Service Mesh.
- Be familiar with the concept of [access control policy](#) in Tanzu Service Mesh.
- Be familiar with the Kubernetes YAML manifest format.

Procedure

- 1 Create a YAML manifest file to describe the access control policy configuration.

- 2 To retrieve the API specification for each configuration that is included in the access control policy configuration, run the `tanzu sm get spec` CLI command.

- a To retrieve the API specification for the service groups configuration, run the following command.

```
tanzu sm get spec gnsservicegroup
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`.

- a To retrieve the API specification for the access control policy configuration, run the following command.

```
tanzu sm get spec accesscontrolpolicies
```

- 3 Copy and paste each returned API specification into the manifest file.

Important

- Make sure that you put `---` at the end of every API specification in the file.
 - You must add a configuration for each service group that you want to define as a source service group or a destination service group to the manifest file.
-

- 4 Provide values for the fields in the resultant configuration.

For information about the meaning of the fields in the configuration and what value to provide for each field, see the schema of the appropriate API in API Explorer in Tanzu Service Mesh. Perform the following steps:

- a In Tanzu Service Mesh SaaS UI, click **API Explorer** on the bottom bar.
 - b Go to the appropriate API in API Explorer.
 - c In the API section, under **Request Body**, click **Schema**.

- 5 Save the changes in the file.

Results

For information about how to apply the access control policy configuration in a manifest file to your Tanzu Service Mesh tenant, see [Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI](#).

What to do next

Add the access control policy manifest file to the Git repository where you maintain all your manifest files.

Create a Public Service Using the CLI

You create a public service with the Tanzu CLI by describing its configuration in a YAML file and then applying the configuration file.

You must create the following manifests to describe the configuration of a public service:

- A health check manifest. Defines the health check settings for the public service. Based on the health-check settings, Tanzu Service Mesh periodically sends requests to the service to check whether it is reachable and functional.
- A public service manifest. Contains the general configuration details about the public service, including its name, the fully qualified domain name (FQDN) of the service, and its external port. For a GSLB-enabled public service, the public service manifest also contains the GSLB parameters.
- A public service route manifest. Defines the name of the internal service associated with the public service and the internal port of the internal service. When a user makes a request to the public service, Tanzu Service routes the request to the internal service in the cluster for processing.

You must retrieve the template for each manifest from the Tanzu Service Mesh API, provide values for the fields in the manifest, and then combine the manifests in a single YAML file, called *public service configuration file*. Arrange the manifests in the public service configuration file in the following order:

- 1 Health check manifest
- 2 Public service manifest
- 3 Public service route manifest

You must then apply the resulting public service configuration file to your Tanzu Service Mesh tenant to create the public service based on the configuration.

Prerequisites

- Become familiar with the concepts [global namespace](#) and [public service](#) in Tanzu Service Mesh. Also become familiar with the concepts of a [GSLB-enabled public service](#) and a [non-GSLB public service](#).
- Choose a fully qualified domain name (FQDN) at which the public service will be exposed. If this service is accessible from the Internet through a GSLB, it must be a valid FQDN that is resolvable on the Internet. If no GSLB integration exists or if the public service is not accessible from the outside, you can define any FQDN that is configured in your DNS.
- (Only for a GSLB-enabled public service) In the configuration for a GSLB-enabled public service, you must include the ID of an external DNS account. For information on how to get an external DNS account ID, see [Get an External DNS ID for a Public Service Configuration](#).
- (Only for a GSLB-enabled public service) Become familiar with the following GSLB concepts that are used in Tanzu Service Mesh. For more information about these concepts, see [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).
 - Global load balancing scheme (round robin, weighted, and active-passive)
 - Health checks

- (Only for a GSLB-enabled public service) If you want to use the weighted or active-passive (failover) global load balancing scheme for the public service, add the required labels to the service configuration on each cluster where the service is deployed. For instructions on how to add these labels, see [Configure Global Load Balancing for Your Application in Tanzu Service Mesh](#).
- If you want to expose the public service at an HTTPS URL, add the Transport Layer Security (TLS) certificate that you want to use for the service to Tanzu Service Mesh. You must provide the name of the certificate in the public service configuration. For information about adding a certificate to Tanzu Service Mesh, see [Add a Certificate to Tanzu Service Mesh Using the CLI](#).

Important Make sure that the Common Name (CN) on the certificate is the same FQDN that you specify for the public service in the public service configuration YAML file.

- Be familiar with the Kubernetes YAML manifest format.
- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).

Procedure

- 1 Create a YAML file to contain the configuration of the public service.

The steps below refer to this file as the *public service configuration file*.

- 2 Retrieve the template for a health check manifest:

```
tanzu sm get spec healthchecks
```

- 3 Paste the health check manifest template into the public service configuration file that you created and provide values for the following health check settings.

Setting	Description
projectId	Name of the associated Tanzu Service Mesh project. Set to default .
name (under name and spec)	Provide a name to identify these health-check settings.
protocol	The protocol (HTTP or HTTPS) to use for health-check requests.
domain	The FQDN of the endpoint to which Tanzu Service Mesh will send health-check requests. By default, the domain is the FQDN of the public service in the public service manifest.
port	The internal port of the internal service. This port must be the same as the target_port in the public service route manifest.

Setting	Description
<code>path</code>	The path to the endpoint for health-check requests. If you specify a path, Tanzu Service Mesh will send health-check requests to the endpoint that is determined as the FQDN (the value of <code>domain</code>) and the value of <code>path</code> . If you don't want to specify a special path for the endpoint, set the default value <code>"/</code> .
<code>healthThreshold</code>	The number of consecutive failed health checks for the service to be considered unhealthy, or the number of consecutive successful health checks for the service to be considered healthy. For example, set <code>healthThreshold</code> to 3 .
<code>certificate_id</code>	If <code>protocol</code> is HTTPS , provide the id of the certificate to use for HTTPS health checks. If <code>protocol</code> is HTTP , set this field to an empty string (<code>""</code>). Tip To get a list of the IDs of all the certificates that exist in Tanzu Service Mesh, run: <pre>tanzu sm get certificate</pre>
<code>external_port</code>	The external port of the endpoint for health checks. If <code>protocol</code> is HTTP , set this field to 80 . If <code>protocol</code> is HTTPS , set this field to 443 .
<code>interval</code>	The amount of time between health-check requests in seconds. You can set <code>interval</code> to 10 seconds or 30 seconds. For example, if you specify 30 seconds, Tanzu Service Mesh will send health-check requests to the service every 30 seconds.

See an example of a health check manifest in the example public service configurations in the **Example: Public Service Configuration Files** section.

- Retrieve the template for a public service manifest:

```
tanzu sm get spec publicservices
```

- Paste the public service manifest template into the public service configuration file that you created and provide values for the following fields.

Field	Description
<code>gnsId</code>	The name of the global namespace that contains the public service configuration.
<code>projectId</code>	The associated Tanzu Service Mesh project. Set this field to default .
<code>name</code> (in metadata)	Set this field to the FQDN of the public service, that is, the value of the <code>fqdn</code> field in <code>spec</code> .

Field	Description
<code>external_port</code>	The external port at which the public service is exposed depending on the value of the <code>external_protocol</code> field in the public service manifest. If <code>external_protocol</code> is HTTP , set this field to 80 . If <code>external_protocol</code> is HTTPS , set this field to 443 .
<code>external_protocol</code>	The protocol at which the public service is exposed. Set to HTTP or HTTPS .
<code>fqdn</code>	The FQDN at which the public service is exposed. Make sure that this FQDN matches the values of the <code>primary_domain</code> and <code>sub_domain</code> under <code>public_domain</code> . See an example of an FQDN in the example public service configurations in the Example section.
<code>gslb</code>	This field contains the fields that describe the global load balancing (GSLB) configuration. For a non-GSLB public service, leave the <code>gslb</code> field blank. For a GSLB-enabled public service, provide values for the nested fields within the <code>gslb</code> field. See the descriptions below.
<code>type</code> (only a GSLB-enabled public service)	Specify the global load balancing (GSLB) scheme for the public service: ROUND_ROBIN , WEIGHTED , or FAILOVER . For a description of these GSLB schemes, see Configure Global Load Balancing for Your Application in Tanzu Service Mesh . On the destination documentation page, the FAILOVER GSLB scheme is referred to as the active-passive GSLB scheme.
<code>active_group</code> (only a GSLB-enabled public service)	If you set <code>type</code> to FAILOVER , under <code>label_values</code> , specify the service label for the active group of service instances and under <code>type</code> , specify the group loading balancing scheme (ROUND_ROBIN or WEIGHTED) for the active group instances. For detailed information about the parameters of the Failover (Active-Passive) GSLB scheme, see the description of Active-Passive in the table under step 3 in Create a Public Service . If you set <code>type</code> to ROUND_ROBIN or WEIGHTED , remove the <code>active_group</code> field.
<code>passive_group</code> (only a GSLB-enabled public service)	If you set <code>type</code> to FAILOVER , under <code>label_values</code> , specify the service label for the passive group of service instances and under <code>type</code> , specify the group loading balancing scheme (ROUND_ROBIN or WEIGHTED) for the passive group instances. For detailed information about the parameters of the Failover (Active-Passive) GSLB scheme, see the description of Active-Passive in the table under step 3 in Create a Public Service . If you set <code>type</code> to ROUND_ROBIN or WEIGHTED , remove the <code>passive_group</code> field.

Field	Description
<code>weighted_policy</code>	<p>If you set type to WEIGHTED, under <code>label_to_weight</code>, for each service label in <code>label_value</code>, specify a weigh value in <code>weight</code>. For detailed information about the parameters of the WEIGHTED GSLB scheme, see the description of Weighted in the table under step 3 in Create a Public Service.</p> <p>If you set type to ROUND_ROBIN or FAILOVER, remove the <code>weighted_policy</code> field.</p>
<code>ha_policy</code>	Deprecated field. Set this field to an empty string ("").
<code>healthcheck_ids</code>	Set this field to the name of the health check in the health check manifest (the value of the <code>name</code> field).
<code>ingress_on_internal_gateways</code>	This field is currently not used. Remove it.
<code>name</code> (in spec)	Optionally provide a description for the public service or set this field to an empty string ("").
<code>certificate_id</code>	<p>If you set <code>external_protocol</code> for the public service to HTTPS, provide the name of the certificate that you want to use to secure connections to the public service.</p> <p>Tip To get a list of the names of all the certificates that exist in Tanzu Service Mesh, run:</p> <pre>tanzu sm get certificate</pre>
<code>external_dns_id</code> (only a GSLB-enabled public service)	<p>Provide the name of the external DNS account to use for the public service.</p> <p>For a non-GSLB public service, remove this field.</p> <p>Tip To get a list of the IDs of all the external DNS accounts in Tanzu Service Mesh, run:</p> <pre>tanzu sm get externaldns</pre>
<code>primary_domain</code>	The primary domain in the FQDN of the public service.
<code>sub_domain</code>	The subdomain in the FQDN of the public service.
<code>ttl</code>	Specify in seconds how long the DNS resolver caches a query before requesting a new one. The default value is 300 .
<code>wildcard_certificate_id</code>	This field is currently not used. Set it to an empty string ("").

See an example of a public service manifest in the example public service configurations in the **Example: Public Service Configuration Files** section.

6 Retrieve the template for a public service route manifest:

```
tanzu sm get spec publicserviceroute
```

7 Paste the public service route manifest template into the public service configuration file that you created and provide values for the following fields:

Field	Description
fqdn	The FQDN of the public service. This FQDN must be the same as the value of the <code>fqdn</code> field in the public service manifest.
gnsId	The name of the global namespace that contains the public service. This name must be the same as the value of the <code>gnsId</code> field in the public service manifest.
name	The name of the public service route configuration.
paths	This field is currently not used. Set it to the default value <code>"/"</code> .
target	The name of the internal service that processes requests sent to the public service.
target_port	The internal port at which the internal service is accessible in the cluster.

See an example of a public service route manifest in the example public service configurations in the **Example: Public Service Configuration Files** section.

Note Make sure that you put `---` after the health check manifest and the public service manifest in the file.

8 Apply the resulting public service configuration YAML file:

```
tanzu sm apply -f {file-name.yaml}
```

The output contains messages that the specified health check, public service, and public service route were created.

Example: Public Service Configuration Files

Configuration for a non-GSLB public service

```
apiVersion: templates.tsm.vmware.com/v1
kind: HealthCheck
metadata:
  labels:
    projectId: default
  name: sample-healthcheck
spec:
  name: sample-healthcheck
  protocol: HTTP
  domain: my-subdomain.shopping.com
```



```

port: 3000
path: "/"
healthThreshold: 3
certificate_id: ""
external_port: 80
interval: 10
---
apiVersion: gns.tsm.vmware.com/v1
kind: PublicService
metadata:
  labels:
    gnsId: my-gns
    projectId: default
    name: my-subdomain.shopping.com
spec:
  fqdn: my-subdomain.shopping.com
  name: ""
  external_port: 80
  external_protocol: HTTP
  ttl: 300
  public_domain:
    primary_domain: shopping.com
    sub_domain: my-subdomain
    certificate_id: ""
  gslb:
    ha_policy: ""
    wildcard_certificate_id: ""
    healthcheck_ids:
      - sample-healthcheck
---
apiVersion: gns.tsm.vmware.com/v1
kind: PublicServiceRoute
metadata:
  labels:
    fqdn: my-subdomain.shopping.com
    gnsId: my-gns
    projectId: default
    name: my-sample-pub-svc.3000
spec:
  paths:
    - "/"
  target: shopping
  target_port: 3000

```

Configuration for a GSLB-enabled public service

```

apiVersion: templates.tsm.vmware.com/v1
kind: HealthCheck
metadata:
  labels:
    projectId: default
    name: sample-healthcheck
spec:
  name: sample-healthcheck
  protocol: HTTP

```

```

    domain: my-subdomain.shopping.com
    port: 3000
    path: "/"
    healthThreshold: 3
    certificate_id: ""
    external_port: 80
    interval: 10
---
apiVersion: gns.tsm.vmware.com/v1
kind: PublicService
metadata:
  labels:
    gnsId: my-gns
    projectId: default
  name: my-subdomain.shopping.com
spec:
  fqdn: my-subdomain.shopping.com
  name: ""
  external_port: 443
  external_protocol: HTTPS
  ttl: 300
  public_domain:
    external_dns_id: 188aacb9-4503-486a-9818-653a9240ef7d
    primary_domain: shopping.com
    sub_domain: my-subdomain
    certificate_id: myHttpsCert
  gslb:
    type: ROUND_ROBIN
  ha_policy: ""
  wildcard_certificate_id: ""
  healthcheck_ids:
    - sample-healthcheck
---
apiVersion: gns.tsm.vmware.com/v1
kind: PublicServiceRoute
metadata:
  labels:
    fqdn: my-subdomain.shopping.com
    gnsId: my-gns
    projectId: default
  name: my-sample-pub-svc.3000
spec:
  paths:
    - "/"
  target: shopping
  target_port: 3000

```

What to do next

To verify that the new public service was added to its global namespace in the Tanzu Service Mesh Console, perform these steps:

- 1 [Access the Tanzu Service Mesh Console.](#)
- 2 In the navigation pane on the left, click **Home**.

- 3 On the **GNS Overview** tab, click the name of the global namespace that contains the public service.
- 4 On the global namespace details page, click the **Public Services** tab.
- 5 To view the details of the public service, including its configuration, click the name of the public service.

To edit the configuration of a public service, perform these steps:

- 1 [Delete an Object or a Policy from Tanzu Service Mesh SaaS Using the CLI](#), passing the name of the public service configuration YAML file in the `tanzu sm delete` command.
- 2 Recreate the configuration of the public service with the values you want in a YAML file by following the instructions in the procedure above.
- 3 [Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI](#).

Note At the top of the public service details page, a health status of **Syncing** or **Error** is initially displayed for a new public service for a few minutes while the public service is starting, and Tanzu Service Mesh determines its health by sending health check requests to the endpoint specified in the health check manifest. If the public service responds successfully to the health check requests according to the specified health check settings, the health status **Syncing** or **Error** is no longer displayed.

Get an External DNS ID for a Public Service Configuration

Creating a public service with the Tanzu Service Mesh CLI requires providing the ID of external DNS account among other configuration details.

You can get an external DNS account ID by retrieving a list of the IDs of all DNS accounts from Tanzu Service Mesh and then selecting the ID that you want.

Prerequisites

- Become familiar with the concept of [external DNS name](#) in Tanzu Service Mesh.
- [Create an DNS provider integration account](#) (AVI or AWS) in the Tanzu Service Mesh Console.
- [Create an external DNS account](#) in the Tanzu Service Mesh Console.
- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).

Procedure

- 1 Retrieve a list of the IDs of all external DNS accounts from Tanzu Service Mesh:

```
tanzu sm get externaldnss
```

- 2 Copy the DNS account ID that you want from the list in the output.

Tip To verify that this is the ID of the DNS account that you want to use, run this command:

```
tanzu sm get externaldnss {DNS-account-ID}
```

The output contains the details of the external DNS account. The name of the account and the other details should help you determine if this is the right DNS account.

Add a Certificate to Tanzu Service Mesh Using the CLI

Using the Tanzu Service Mesh CLI, you can add transport layer security (TLS) certificates to Tanzu Service Mesh for use in different situations where secure, TLS-encrypted connections are needed.

You must use a TLS certificate for a public service that is exposed at an HTTPS URL. In this case, you must include the name of the certificate in the public service configuration to ensure secure HTTPS connections to the service.

To add a certificate to Tanzu Service Mesh, you must retrieve the template for a certificate manifest from the Tanzu Service Mesh API and then provide values for the fields in the manifest, including the public certificate and private key strings. As a final step, you must apply the resulting certificate manifest to your tenant in Tanzu Service Mesh.

Prerequisites

- Verify that you are familiar with public-key infrastructure (PKI) concepts *certificate*, *private key*, *certificate authority (CA)*, and *certificate chain*.
- You have a public certificate and a private key from a trusted certificate authority (CA) and know the location of the certificate and private key files. The certificate file must be in PEM (.pem) format. The private key file must be in PEM or KEY (.key) format.

Note You can also add a self-signed certificate to Tanzu Service Mesh, such as one that you can generate by using the OpenSSL toolkit. Instructions on generating a self-signed certificate are out of scope of this documentation.

- To ensure that the certificate works correctly, verify that the common name (CN) on the certificate is the same as the domain specified in the configuration of a [public service](#). For more information about creating public services with the Tanzu Service Mesh CLI, see [Create a Public Service Using the CLI](#).
- [Install the CLI](#).
- [Log in to the Tanzu Service Mesh CLI](#).
- Be familiar with the Kubernetes YAML manifest format.

Procedure

1 Retrieve the certificate manifest template:

```
tanzu sm get spec certificates
```

2 Provide values for the following fields in the manifest:

Field	Description
projectId	The associated Tanzu Service Mesh project. Set this field to default .
name (in metadata and spec)	<p>The name of the certificate. Provide a friendly name to help identify the certificate in the Tanzu Service Mesh Console.</p> <p>The name can contain only alphanumeric characters and underscores (_) and cannot contain numbers and special characters, such as, hyphens (-), ampersands (&), and pound signs (#). The name must contain a minimum of 2 characters and a maximum of 1,024 characters.</p>
certificateType	Set to UnmanagedCertificate .
description	An optional description of the certificate. If you don't want to provide a description, set this field to an empty string ("").
managedCertificate	Configurations of internally managed certificates. Remove this field and the nested fields.
unManagedCertificate	Configurations of externally managed certificates.
cert_chain	<p>Optional field. Insert the contents of the CA certificate chain file.</p> <p>If you don't have a certificate chain file from your CA, remove this field.</p>
private_key	Insert the private key string from the private key file, including the -----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY----- statements.
signed_certificate	Insert the public certificate string from the public certificate file, including ---- BEGIN CERTIFICATE---- and ----END CERTIFICATE---- statements.

3 Apply the certificate manifest YAML file:

```
tanzu sm apply -f {file-name.yaml}
```

The output contains a message that the specified certificate was created.

What to do next

The new certificate is added to the **Keys & Certificates** page in the Tanzu Service Mesh Console. To view the certificate, perform the following steps:

- 1 [Access the Tanzu Service Mesh Console.](#)
- 2 In the navigation pane on the left, click **Admin > Keys & Certificates**.
- 3 On the **Keys & Certificates** page, in the table on the **Keys & Certificates** tab, view the details of the certificate.

Debug Problems with a Manifest File

If a declarative manifest file is not applied because of an error, you can rerun the manifest application CLI command with a debugging option to see the error in the output, correct the error, and then re-apply the manifest.

Prerequisites

- [Install the CLI.](#)
- [Log in to the Tanzu Service Mesh CLI.](#)
- Make sure that the Tanzu Service Mesh CLI can access your declarative manifest files in the local folder where you cloned them from the Git repository.

Procedure

- 1 In a terminal window, change to the local folder where you cloned the manifest files from the Git repository.
- 2 Run the following CLI command.

```
tanzu sm apply -f {manifest YAML file} --debug
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{manifest YAML file}` is the name of the manifest YAML file that was not applied.

- 3 Locate the error in the output of the command.
- 4 Correct the error and [Apply a Configuration to Tanzu Service Mesh SaaS Using the CLI.](#)

Delete an Object or a Policy from Tanzu Service Mesh SaaS Using the CLI

Using the Tanzu Service Mesh CLI, you can remove an object, such as a global namespace, or a policy, such as an access control policy, from the configuration in your Tanzu Service Mesh tenant.

To remove an object or a policy, you must run the appropriate CLI command. The CLI command includes the name of the manifest file that contains the configuration of the object or policy.

Prerequisites

- [Install the CLI.](#)

[Log in to the Tanzu Service Mesh CLI.](#)

- Make sure that you know the name of the manifest file with the configuration of the object or policy that you want to delete.
- Make sure that the Tanzu Service Mesh CLI can access the local folder that contains the manifest file.
- Make sure that the machine where the CLI is installed can access Tanzu Service Mesh SaaS.

Procedure

- 1 In a terminal window, change to the local folder that contains the manifest file.
- 2 Run the following CLI command.

```
tanzu sm delete -f {manifest YAML file}
```

`sm` is the alias for the Tanzu Service Mesh plugin in the Tanzu CLI. You can replace `sm` with `service-mesh`. `{manifest YAML file}` is the name of the manifest YAML file that contains the configuration of the object or policy that you want to delete from Tanzu Service Mesh.

Results

When the object or policy is deleted, the command returns `{object} deleted` or `{policy} deleted` in the output.

When a global namespace is deleted, the command returns several `deleted` lines for the different configurations that make up the global namespace.

Tanzu Service Mesh Administration

15

You can use the Admin area in the Tanzu Service Mesh Console user interface to perform the following administrative tasks.

Read the following topics next:

- [Manage Integrations](#)
- [Manage Domains](#)
- [Manage Keys and Certificates](#)

Manage Integrations

You can integrate with external services to add capabilities and features to Tanzu Service Mesh, such as domain name system (DNS) capability. You can use the Integrations page to manage external integrations in Tanzu Service Mesh.

You can enable connections to external services that extend how you manage your services with Tanzu Service Mesh. For example, you can configure an integration with Amazon Web Services (AWS) to use it as a DNS provider for your services.

To integrate Tanzu Service Mesh with an external service, you must create an integration account with that service. Tanzu Service Mesh will use the credentials that you provided in the account to connect to the service. After you create an integration account, you can reference it in appropriate places within Tanzu Service Mesh to enable the functionality provided by the service. You can create one or more integration accounts for an external service.

Create an AWS Integration Account

You can integrate with Amazon Web Services (AWS) to add domain name system (DNS) and global server load balancing (GSLB) functions in AWS to Tanzu Service Mesh.

This procedure describes how to create an integration account with AWS to add DNS and GSLB capabilities to Tanzu Service Mesh. After you create an AWS integration account, to make your organization's domains managed by AWS available in Tanzu Service Mesh, you must reference the account in an appropriate DNS account.

Prerequisites

- Know the access key ID and the associated secret access key of your AWS account.
- Make sure that your AWS account has the following permissions:
 - List access for hosted zones associated with the AWS account
 - Write and list access for records within a hosted zone
 - Read, write, and list access for health checks
 - Tagging and list access for tags for health checks and hosted zones
- **Note** The AWS Identity & Access Management (IAM) policy example contains least permissions required for your AWS account. The resource, "arn:aws:route53::hostedzone/*", can be further restricted by hosted zone ID as required: "arn:aws:route53::hostedzone/\${Id}".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:GetHealthCheckStatus",
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:ListResourceRecordSets",
        "route53:DeleteHealthCheck",
        "route53:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:route53::hostedzone/*",
        "arn:aws:route53::healthcheck/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListHealthChecks",
        "route53:CreateHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHostedZonesByName"
      ],
      "Resource": "*"
    }
  ]
}
```

- Access the Tanzu Service Mesh Console. For information about accessing the Tanzu Service Mesh Console, see [Access the Tanzu Service Mesh Console](#).

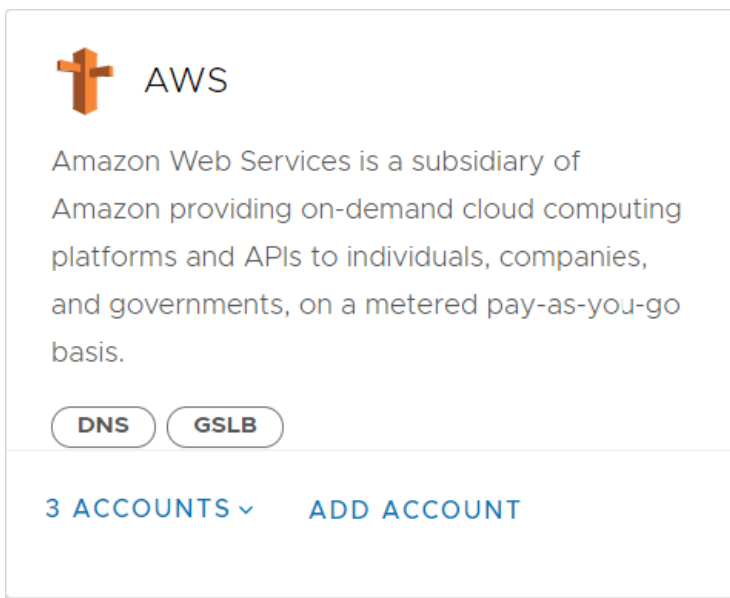
Procedure

- 1 In the navigation pane on the left, click **Admin > Integrations**.
- 2 On the **Integrations** page, under **All Integrations**, find the AWS card with a DNS label toward the bottom of the card.

Note To filter the external services on the page to only those services that provide DNS functionality, click the **DNS** label to the right of **All Integrations**.

If one or more AWS integration accounts exist in Tanzu Service Mesh, the number of accounts is displayed in the lower-left corner of the card.

The following image shows the AWS integration card. The card indicates that three AWS integration accounts exist in Tanzu Service Mesh.



- 3 Select one of the following options.
 - If you are creating the first AWS integration account, at the bottom of the card, click **Configure**.
 - If one or more AWS integration accounts exist and you are creating another account, at the bottom of the card, click **Add Account**.
- 4 In the **New AWS Integration** dialog box, provide the following information.
 - **Name.** The name for the account to help distinguish it from other accounts.
 - (Optional) **Description.** An optional description or details about the account.
 - **Access Key ID.** Your AWS access key ID.

- **Secret Access Key.** The secret access key associated with your access key ID.

Note

- For information about how to obtain an AWS access key ID and a secret access key, see the AWS documentation.
 - The credentials, such as an access key ID and a secret access key, that you provide in the **New AWS Integration** dialog box are encrypted and securely stored in Tanzu Service Mesh.
-

5 Click **Save**.

Results

The new account is added to the AWS integration card on the **Integrations** page.

Warning If you use Route 53 as a DNS service, do not edit the health check configuration created by Tanzu Service Mesh in AWS and do not edit or delete the records and tags in the hosted zones that Tanzu Service Mesh created for your domains. Editing or deleting these Tanzu Service Mesh-created data can break DNS resolution and global load balancing for your application.

What to do next

To edit or delete the account, click **Edit** or **Delete** in the AWS card. If you have more than one AWS account, in the lower-left corner of the card, click ***number* Accounts**, click the name of the account, and then click **Edit** or **Delete**.

To make your organization's domains managed by AWS available in Tanzu Service Mesh, [Manage Domains](#) selecting the name of the AWS integration account as the domain provider in the DNS account. For Route 53 health checks to function properly, the Ingress Gateway IP address should be public.

Create an Avi Integration Account

To make global server load balancing (GSLB) capabilities of NSX Advanced Load Balancer (formerly Avi Networks) available for your application in Tanzu Service Mesh, you need to create an Avi integration account.

Creating an Avi integration account is a required step if you need to configure global load balancing for your application using NSX Advanced Load Balancer. Your application will be exposed to users through a public service configured in a global namespace, and NSX Advanced Load Balancer will route user requests to optimal application instances by using the global load balancing configuration specified for the public service.

Prerequisites

- Configure GSLB sites, including a GSLB leader site, in Avi. Delegate the domains that you own to Avi GSLB. For more information about creating GSLB sites, see the [Avi documentation](#).

- Know the name of the Avi tenant that your Avi user account is associated with. For information about tenants, see the [Avi documentation](#).
- Know the user name and password of your Avi user account on the leader site.
- Know the IP address or the fully qualified domain name (FQDN) of the Controller cluster on the GSLB leader site.
- Access to the Tanzu Service Mesh Console. For information about accessing the Tanzu Service Mesh Console, see [Access the Tanzu Service Mesh Console](#).


Note If Avi Kubernetes Operator (AKO) is installed on the onboarded clusters where instances of the public service will be deployed, deactivate the `L4Settings.autoFQDN` configuration setting during installation. This setting is available starting with AKO version 1.3.3. If this setting is not deactivated, Tanzu Service Mesh will try to resolve the ingress gateway using the local FQDN rather than the external IP address, which will only work if the resolvers on the nodes point to Avi DNS. For information about the `L4Settings.autoFQDN` setting, see the AKO documentation on GitHub.

Procedure

- 1 Access the Tanzu Service Mesh Console.
- 2 In the navigation pane on the left, click **Admin > Integration**.
- 3 On the **Integrations** page, under **All Integrations**, find the Avi card with the **DNS** and **GSLB** labels.

If one or more Avi integration accounts exist in Tanzu Service Mesh, the number of accounts is displayed in the lower-left corner of the card.

The following image shows the Avi integration card.


AVI

Avi Networks is a company that provides software for the delivery of enterprise applications in data centers and clouds.

[EDIT](#)
[DELETE...](#)
[ADD ACCOUNT](#)

4 Select one of the following options.

- If you are creating the first Avi integration account, at the bottom of the card, click **Configure**.
- If one or more Avi integration accounts exist and you are creating another account, at the bottom of the card, click **Add Account**.

5 In the **New Avi Integration** dialog box, provide the following information.

- **Name.** Enter a friendly name for the account.
- **Description.** (Optional) enter a description of the account.
- **Authentication.** Select **Username & Password** and enter the user name and password of your Avi user account on the GSLB leader site. Tanzu Service Mesh will use these credentials to connect to the leader site on Avi.

Note The **Authentication Token** option is currently not supported.

- **Avi Tenant.** Enter the name of the Avi tenant with which your Avi user account is associated.
- **Controller Address.** Specify the IP address or the fully qualified domain name (FQDN) of the Controller cluster on the leader site.

New AVI Integration



* indicates required information

Name *	My Avi integration account
Description (optional)	<input type="text" value="Optional description"/>
Authentication	<input type="radio"/> Authentication Token <input checked="" type="radio"/> Username & Password
Username *	My user
Password *	<input type="password" value="....."/> <input type="button" value="eye icon"/>
AVI Tenant *	My Avi tenant
Controller Address *	<input checked="" type="radio"/> IP Address: 34.208.166.215 <input type="radio"/> FQDN: <input type="text" value="Enter FQDN"/>

CANCEL

SAVE

6 Click **Save**.

Results

The new account is added to the Avi integration card on the **Integrations** page.

What to do next

To edit or delete the account, click **Edit** or **Delete** in the Avi card. If you have more than one Avi integration account, in the lower-left corner of the card, click **number Accounts**, click the name of the account, and then click **Edit** or **Delete**.

To make the subdomains you delegated to Avi GSLB available for inclusion in the URL of a public service, you must also [Manage Domains](#) in Tanzu Service Mesh, selecting the name of the Avi integration account as the domain provider in the DNS account.

Create a Venafi Integration Account

You can set up Venafi accounts in Tanzu Service Mesh to integrate with the Venafi Trust Protection Platform as one of the external CA providers for automatic management of workload

certificates enabling mTLS. All certificates will be protected and controlled for workloads, and this process will be transparent to the Tanzu Service Mesh Controller. This topic describes how to integrate Venafi Trust Protection Platform (TPP) as a Root of Trust for workload encryption in addition to the self-signed CA that Tanzu Service Mesh already supports.

In Tanzu Service Mesh, create a Venafi integration account using the following steps.

Prerequisites

- You must have an access token for TPP Admin obtained from the TPP Admin URL.
- A global namespace can only be created when the onboarded clusters are all under the same CA. The creation of the global namespace creation will fail or will not proceed if services from clusters with different CAs (for example, self-signed on one cluster and Venafi on the other) are selected. For more information, see step 7 in [Onboard a Cluster to Tanzu Service Mesh](#)

Procedure

- 1 In the navigation pane on the left, click **Admin > Integration**.
- 2 On the **Integrations** page, locate the **Venafi** card.
- 3 Select one of the following options:
 - If you are creating the first Venafi integration account, click **Configure** at the bottom of the card.
 - If one or more Venafi integration accounts exist and you are creating another account, click **Add Account** at the bottom of the card.
- 4 In the **New Venafi Integration** dialog box, provide the following information.
 - **Name.** Enter a name for the account.
 - **Description.** (Optional) Give a brief description of the account.
 - **Label(s).** Label is created automatically with the syntax **Certificate Authority: Name of the Venafi account**. Label is assigned to the cluster to use the Venafi service during its onboarding. Refer to step 7 in [Onboard a Cluster to Tanzu Service Mesh](#) for more information.
 - **URL.** Enter the Venafi TPP admin URL.
 - **Access Token.** Enter the Access Token for authentication. Tanzu Service Mesh will use this token to access the Venafi site.
 - **Zone.** Enter the zone of the Venafi region.

New VENAFI Integration

×

* indicates required information

Name *	Venafi2
Description (optional)	Optional description
Label(s) *	Certificate Authority: Venafi2
Enter the details and access token for the Venafi account.	
URL *	https://tanzu-tpp.se.venafi.com/vedsdk
Access Token *
Zone *	Venafi Partners\VMware\Certificates

CANCEL

SAVE

5 Click **Save**. The **Trust Domain** modal window appears.

6 In the Trust Domain modal window, enter the following information and click **Save**.

Trust Domain. Enter the trust domain name. The trust domain corresponds to the trust root of a system and is part of a workload identity. A trust domain is required to onboard new clusters and create global namespaces.

Trust Domain

×

A Trust Domain is required to onboard Clusters and create Global Namespaces

* indicates required field

Trust Domain *	cluster.local
Eg. domain.internal. Required for Global Namespace creation.	

CANCEL

SAVE

You can view the Trust Domain list by clicking **Admin> Project Configuration** on the left navigation pane. The trust domains created for each project are listed here.

Important It is essential for clusters to have a common root certificate and a shared trust domain to communicate with one another.

Do not edit trust domains, even though the UI doesn't restrict them. You need to off-board the cluster and then on-board it back with the new trust domain in order for this to work.

Results

The new account is added to the Venafi integration card on the **Integrations** page. To check the health status of the connected clusters, select:

- On the **Global Namespaces** page, click the name of the global namespace that you created. The global namespace details page displays the summary information about the global namespace, including its overall health state. The status is **Healthy** if the configuration of the global namespace is synced and applied to the clusters that make up the global namespace. Verify that the **CA connected** status is displayed on each cluster within the global namespace by clicking the **Healthy** drop-down.

Note To access the **Global Namespaces** page, in the navigation pane on the left, select **Inventory > Global Namespaces**.

- On the **Clusters** tab, click the name of the desired cluster from the displayed table list. The cluster details page displays the summary information about the cluster, including its overall health state. Verify that the **CA connected** status is displayed on the selected cluster by clicking the **Healthy** drop-down.

Note To access the **Clusters** page, in the navigation pane on the left, select **Inventory > Clusters & Nodes** and select the **Clusters** tab.

What to do next

To edit or delete the account, click **Edit** or **Delete** in the **Venafi** card. If you have more than one Venafi account, in the lower-left corner of the card, click <<number>> **Accounts**, click the name of the account, and then click **Edit** or **Delete**. The message "This integration will be deleted" must be confirmed if you are deleting an account.

Create a Vault CA Integration Account

If your organization uses Vault as an internal certificate authority (CA) and you want to use certificates signed by Vault CA for secure mTLS communication between services within your service mesh, create an integration with Vault CA in Tanzu Service Mesh.

When your organization uses Vault as an internal CA, it can have one or multiple Vault servers that store a trusted root certificate and one or more intermediate certificates. Vault uses these CA certificates to generate and sign mTLS session certificates for authentication and encryption of services within the service mesh.

Every time a service needs an mTLS session certificate for secure communication with other services in the service mesh, the service sends a request to a Vault server from its Vault-connected cluster. Vault generates a certificate, signs it, and sends it back to the cluster. The service then uses the certificate to make mTLS authenticated and encrypted connections to other services.

If you want to use Vault CA for your service mesh, you must create a Vault integration in Tanzu Service Mesh. The integration defines a configuration that points to a Vault server. The Vault CA configuration contains such details as the URL of the Vault server, the endpoint on Vault to use for signing certificates, and the reference to the Vault server's bundle that contains the root certificate and the intermediate certificates.

After you create a Vault CA integration, you need to apply the CA label from the integration to a cluster when you onboard it. During onboarding, the Vault CA configuration is pushed to the services in the cluster, and the root and intermediate certificates are uploaded into the cluster's trust store.

Once the Vault CA configuration is applied to a cluster, the cluster is considered to be connected to the Vault server, and the services can send certificate generation and signing requests to the server. Because the cluster has the Vault server's root and intermediate certificates in its trust store, it trusts mTLS session certificates received from Vault CA.

You can view the CA health status for each connected cluster in the Tanzu Service Mesh UI.

Note You can also create a Vault CA integration by using the Tanzu Service Mesh API. For more information, see [Create a Vault CA Integration](#) in the *Tanzu Service Mesh API Programming Guide*.

Prerequisites

- [Access the Tanzu Service Mesh Console](#).
- Verify that you are familiar with Vault concepts, such as *PKI secrets engine*, *mount path*, *namespace*, *role*, and *policy*. For information about these Vault concepts, see the [HashiCorp Vault documentation](#).
- Verify that you are familiar with the Kubernetes concept *service account*. For more information about service accounts, see the [Kubernetes documentation](#).
- Configure Vault to trust connections from your Kubernetes clusters. See step 1 in the following procedure.

Procedure

- 1 To configure Vault to trust connections from your clusters and accept certificate generation and signing requests from them, perform the following steps for each cluster on Vault.

- a In your terminal, run the following commands to set environment variables for the Vault server, the Vault namespace, and the Vault authentication token to access the Vault API.

```
export VAULT_ADDR="vault_server_URL:8200"; export VAULT_NAMESPACE="namespace"export
VAULT_TOKEN=TOKEN
```

- b Set up a PKI secrets engine. For example, run these commands to set up a PKI secrets engine.

```
vault secrets enable pki

# Create the root certificate
vault write -field=certificate pki/root/generate/internal common_name="svc" ttl=8760h
> CA_cert.crt

vault write pki/config/urls issuing_certificates="$VAULT_ADDR/v1/pki/ca" \
  crl_distribution_points="$VAULT_ADDR/v1/pki/crl"

vault secrets enable -path=pki_int pki

vault write -format=json pki_int/intermediate/generate/internal common_name="svc
Intermediate Authority" | jq -r '.data.csr' > pki_intermediate.csr

# This is the intermediate certificate
vault write -format=json pki/root/sign-intermediate \
  csr=@pki_intermediate.csr \
  format=pem_bundle ttl=4380h | jq -r '.data.certificate' > intermediate.cert.pem

vault write pki_int/intermediate/set-signed certificate=@intermediate.cert.pem

vault policy write policy_name - <<EOF
path "policy_name*" { capabilities = ["read", "list"] }
path "policy_name/sign/example-dot-com" { capabilities = ["create", "update"] }
path "policy_name/issue/example-dot-com" { capabilities = ["create"] }
EOF

vault write policy_name/roles/example-dot-com \
  allowed_domains="svc" allow_subdomains=true \
  max_ttl="720h" require_cn=false \
  allowed_uri_sans="spiffe://domain.name/*"
```

The last command is used to configure a signing role on Vault. Provide the following values:

- `allowed_domains`. Specifies the domains that this role is allowed to issue certificates for. Set allowed domains to **svc**.

- `allow_subdomains`. Specifies if clients can request certificates with common names (CNs) that are subdomains of the CNs allowed by the other role options. Set `allow_subdomains` to **true**.
- `require_cn`. Set to **false**. This makes the `common_name` field optional during generation of a certificate.
- `allowed_uri_sans`. Defines allowed Subject Alternative Name (SAN) URIs. Set `allowed_uri_sans` to the trust domain configured for the project that you use in Tanzu Service Mesh. For example, if you use the default project in Tanzu Service Mesh, set `allowed_uri_sans` to **cluster.local**.

- c Set the `SA_CA_CERT` environment variable to the cluster's client certificate.

```
export SA_CA_CERT=$(kubectl get secret -n default service-account-name -o
jsonpath="{.data['ca\.crt']}" | base64 --decode; echo)
```

- d Set the `K8S_HOST` environment variable to the URL of the Kubernetes API server.

```
export K8S_HOST=$(kubectl config view --minify | grep server | cut -f 2- -d ":" | tr
-d " ")
```

- e Set the `CLUSTER_NAME` environment variable to the name of the cluster and enable the Kubernetes authentication method.

```
export CLUSTER_NAME=cluster_name
vault auth enable --path=$CLUSTER_NAME kubernetes
```

Make sure that the cluster name is unique.

- f Use the `/config` endpoint to configure Vault to communicate with the Kubernetes cluster.

```

vault write auth/$CLUSTER_NAME/config \
  kubernetes_host="$K8S_HOST" \
  kubernetes_ca_cert="$SA_CA_CRT"

```

This command imports the cluster's client certificate to Vault so that connections from the cluster can be trusted by Vault.

- g Create a named role.

```

vault write auth/$CLUSTER_NAME/role/role_name \
  bound_service_account_names=service_account_name \
  bound_service_account_namespaces=istio-system \
  policies=policy_name \
  ttl=1h

```

`bound_service_account_names` is the name of the Kubernetes service account on the cluster that has access to the role. The role will be used to associate the service account with the set of Vault policies specified in `policies` within the role configuration.

`bound_service_account_namespaces` is the namespace that is allowed to access the role. Set `bound_service_account_namespaces` to **istio-system**.

Tanzu Service Mesh will create the specified service account in the `istio-system` namespace on the cluster during onboarding. When sending requests to Vault, a service will authenticate to Vault as the specified service account from the `istio-system` namespace.

`policies` is the policy that gives read access to the paths of the PKI secrets engine paths. This references the name of the policy created in step b.

Here is an example of a role.

```

vault write auth/$CLUSTER_NAME/role/pki_int \
  bound_service_account_names=vault-issuer \
  bound_service_account_namespaces=istio-system \
  policies=pki_int \
  ttl=1h

```

- 2 In the navigation pane on the left, click **Admin > Integration**.
- 3 On the **Integrations** page, locate the **Vault** card.
- 4 Select one of the following options:
 - If you are creating the first Vault integration account, click **Configure** at the bottom of the card.
 - If one or more Vault integration accounts exist and you are creating another account, click **Add Account** at the bottom of the card.

5 In the **New Vault Integration** dialog box, provide the following information.

- **Name.** Enter a name for the account.
- **Description.** (Optional) Give a brief description of the account.
- **Label.** The label associated with this Vault CA integration account. A Vault CA label is created automatically with the syntax Certificate Authority: Name of the Vault account. You can use this generated label or provide your own label. Click the automatically generated label, click **New Value**, type your label, click **Add new** below, and click **Save**.
- **URL.** The URL of the Vault server accessible from the Kubernetes cluster, for example: `https://vault-cluster-public-vault.z1.hashicorp.cloud:8200`.
- **Service Account Name.** The name of the Kubernetes service account that services will use to authenticate to the Vault server when sending requests from the cluster. An example of the service account name is **vault-issuer**. This value must be the service account name that you provided in step 1 g.
- **Role.** The name of the Vault role, for example, **pki_int**. This value must be the name of the role that you created in step 1 g.
- **PKI Path.** The Vault path that will be used for signing certificates. This path must use the `sign` endpoint and must match the path that you specified in the `vault secrets enable` command when you were creating a PKI secrets engine (see step 1 b). An example of PKI Path is `pki-int/sign/example-dot-com`.
- **Mount Path.** (Optional) The Vault mount path to use when authenticating with Vault. If you specified the cluster name as the value of `path` when enabling the Kubernetes authentication method in step 1 e, use the format `/v1/auth/my_cluster_name` to provide the mount path. If you leave the **Mount Path** box blank, the default mount path of `/v1/auth/kubernetes` will be used.
- **Namespace.** (Optional) The namespace to use in Vault. Provide this value only if you use Vault Enterprise.
- **Certificate Authority Chain.** (Optional) The CA bundle file that contains the root certificate and intermediate certificates that Vault uses to sign certificates. Your clusters will validate certificates issued by Vault CA against the root and intermediate certificates in the CA bundle. Provide the CA bundle file only if your organization uses a self-signed root certificate issued by an internal CA and the Vault server uses an HTTPS URL. If you use a root certificate issued by a publicly trusted CA, you don't need to provide a CA bundle.

Select the name associated with the previously uploaded CA bundle. If you have not uploaded the CA bundle to Tanzu Service Mesh yet, perform the following steps.

Note The CA bundle file can contain a maximum of three certificates, including only one root certificate and a maximum of two intermediate certificates.

- a Click **New Certificate Chain**.

- b In the **New Certificate Authority Chain** dialog box, provide a name and optionally a description for the CA bundle.
- c Next to **Certificate Chain**, click **Select .PEM File** and browse to the CA bundle file that you want to upload.

The root and intermediate certificates in the CA bundle will be imported into the cluster's trust store during onboarding.

6 Click **Save**.

7 In the **Trust Domain** window, enter the trust domain name.

The trust domain corresponds to the trust root of a system and is part of a workload identity. A trust domain is required to onboard new clusters and create global namespaces.

Important The trust domain must exactly match the `allowed_uri_sans` value that you provided when you were creating a PKI secrets engine in step 1 b.

Results

The new account is added to the Vault integration card on the **Integrations** page. To view the details of the new Vault integration account and of the other existing integration accounts on the integration accounts details page, click **View Details** on the Vault integration card.

You can view the CA health status for the clusters connected to the Vault server specified in the integration account. A CA health status of **CA: Connected** means that the services in the cluster can successfully have certificates generated and signed by Vault CA. A CA health status of **CA: Unknown** means that the cluster is disconnected from the Vault server, and the services in the cluster cannot send certificate generation and signing requests to the server.

To view the CA health status of the connected clusters:

- On the **Global Namespaces** page, click the name of the global namespace that you created. The global namespace details page displays the summary information about the global namespace, including its overall health state. The overall status of **Healthy** means that the configuration of the global namespace is synced and applied to the clusters that make up the global namespace. Verify that the **CA: Connected** status is displayed for each cluster within the global namespace by clicking the **Healthy** drop-down menu at the top of the page.

To access the **Global Namespaces** page, in the navigation pane on the left, select **Inventory > Global Namespaces**.

- On the **Clusters** tab, click the name of the cluster from the displayed table. The cluster details page displays the summary information about the cluster, including its overall health state. Verify that the **CA: Connected** status is displayed for the selected cluster by clicking the **Healthy** drop-down menu at the top of the page.

To access the **Clusters** tab, in the navigation pane on the left, select **Inventory > Clusters & Nodes** and select the **Clusters** tab.

What to do next

- To apply the Vault CA configuration from the integration account to a cluster so that the services deployed in the cluster can send certificate generation and signing requests to the Vault server, select the CA label from the integration account when you [onboard the cluster](#) (see step 7).
- To edit a Vault integration account, click **Edit** in the Vault card. If you have more than one Vault account, in the lower-left corner of the card, click **<number> Accounts**, click the name of the account, and then click **Edit**. You can edit the following details:
 - **Description**. The description of the integration account in Tanzu Service Mesh.
 - **Service Account Name**. The name of the Kubernetes service account that services will use to authenticate to the Vault server when sending requests from the cluster.

Important Before changing the service account name in the integration account, change the service account name in the role configuration in Vault. See step 1 g.

- To delete a Vault integration, click **Delete**. To confirm the deletion, click **Confirm**. If any clusters use the Vault CA configuration from the account, a warning is displayed. Select **I understand the consequences of this action** and click **Confirm**.

Warning If you delete a Vault CA integration, and the associated CA issued certificates for the service pods on a cluster, the pods will continue to have the certificates until they expire. When the certificates expire, mTLS connectivity between the services will fail.

Also, if a pod gets deleted, it won't be automatically recreated because the CA was removed, and no new certificates can be created.

In this case, you will need to connect another CA to the cluster and restart all the pods so that they can start using the new CA's certificates.

Note You can also edit or delete an integration account from the integration account details page.

- 1 On the Vault card, click **View Details**.
 - 2 On the integration details page, in the section for the account, click **Edit Configuration** or click **More Actions** and then click **Delete Account**.
-

- You can [change the Vault CA integration for a cluster by using the Tanzu Service Mesh API](#). We recommend changing the CA integration for a cluster only if the change will not cause that cluster and the other clusters in the global namespace to use different roots of trust. The use of different roots of trust in the global namespace will cause cross-cluster connectivity problems.

Create a GitHub Integration Account

When creating a configured API, you must select the GitHub repository that contains the API documents from which Tanzu Service Mesh will create an API specification for the API. For Tanzu

Service Mesh to connect to the repository, you must create a GitHub integration to point Tanzu Service Mesh to the organization account on GitHub that owns the repository.

- Verify that you are familiar with the concept of [configured API](#) in Tanzu Service Mesh.
- Verify that you have created an organization account on GitHub.
- Verify that you have created a repository within the organization to store the API documents (YAML files) that Tanzu Service Mesh will use to create an API specification for the configured API. Verify that you know the name of the repository.
- Have your GitHub personal access token ready.
- [Access the Tanzu Service Mesh Console.](#)

Procedure

- 1 In the navigation pane on the left, click **Admin > Integration**.
- 2 On the **Integrations** page, under **All Integrations**, find the GitHub card.

If one or more GitHub integration accounts exist in Tanzu Service Mesh, the number of accounts is displayed in the lower-left corner of the card.
- 3 Select one of the following options.
 - If you are creating the first GitHub integration account, at the bottom of the card, click **Configure**.
 - If one or more GitHub integration accounts exist and you are creating another account, at the bottom of the card, click **Add Account**.
- 4 In the **New GITHUB Integration** dialog box, provide the following information.
 - **Name**. Enter a name to help identify the account in Tanzu Service Mesh.
 - **Description** (Optional). Enter an optional description of the account.
 - **Access Token**. Enter your personal access token on GitHub.
- 5 Click **Save**.

Results

The new account is added to the GitHub integration card on the **Integrations** page.

Tanzu Service Mesh will connect to GitHub to retrieve a list of the repositories owned by the associated organization account. Now when you create a configured API, you need to first select the appropriate Git integration account-organization pair and then the repository. For more information about creating a configured API, see [Configure an API](#).

What to do next

To view the details of the new GitHub integration account and of the other existing accounts on the integration accounts details page, click **View Details** on the GitHub card.

To edit the GitHub integration account, click **Edit** in the GitHub card and then make the changes you want in the **Edit GITHUB Integration** dialog box. If you have more than one GitHub integration accounts, in the lower-left corner of the card, click ***number* Accounts**, click the name of the account, and then click **Edit**.

To delete the GitHub integration account, click **Delete** in the GitHub card and then click **Confirm**. If you have more than one GitHub integration accounts, in the lower-left corner of the card, click ***number* Accounts**, click the name of the account, and then click **Delete**.

Note You can also edit or delete a GitHub integration account from the integration accounts details page. Click **View Details** on the GitHub card. To edit the account, next to the name of the account, click **Edit Configuration**. To delete the account, click **More Actions** and then click **Delete Account**.

Manage Domains

As an application operator, you can make your organization's domains available for different DNS use cases in Tanzu Service Mesh, such as public services. To enable domains, you must create DNS accounts on the **DNS & Domains** page.

When you create a DNS account, you specify the name of the external DNS provider that manages your organization's domains. The DNS provider name matches the name of the integration account that you have created for the provider. Tanzu Service Mesh uses the configuration in the integration account to connect to the DNS provider, get a list of the domains, and make them available for use in Tanzu Service Mesh. For more information about integration accounts, see [Create an AWS Integration Account](#).

The DNS accounts that you create determine the domains that are available for selection when users configure public services in Tanzu Service Mesh. For more information about public services, see [Chapter 6 Create a Public Service](#).

Prerequisites

Verify that:

- You have created an integration account for your external DNS provider (for example, AWS). For more information about integration accounts, see [Create an AWS Integration Account](#).
- You are in the Tanzu Service Mesh Console. For information about accessing the Tanzu Service Mesh Console, see [Access the Tanzu Service Mesh Console](#).

Procedure

- 1 On the navigation pane on the left, click **Admin > DNS & Domains**.
- 2 On the **DNS & Domains** page, above the table, click **New DNS Account**.

3 In the **New DNS Account** dialog box, provide the following information.

- **Name.** The name for the account to help distinguish it from other accounts. The name can contain only alphanumeric characters and underscores (_) and cannot contain numbers and special characters, such as ampersands (&) and pound signs (#). It must contain a minimum of 2 characters and a maximum of 1,024 characters.
- (Optional) **Description.** An optional description of the account.
- **Domain Provider.** The external domain provider. Select the name of the integration account for the external DNS service (for example, **AWS - my_dns_integration_account**).

4 Click **Save**.

Results

The message `DNS account created successfully` appears in the lower-left corner of the Tanzu Service Mesh Console. The new DNS account is added to the table on the **DNS & Domains** page. To edit or delete the account, click the three vertical dots to the left of the account name in the table and click **Edit** or **Delete** on the menu.

Warning If you use Route 53 as a DNS service, do not edit the health check configuration created by Tanzu Service Mesh in AWS and do not edit or delete the records and tags in the hosted zones that Tanzu Service Mesh created for your domains. Editing or deleting these Tanzu Service Mesh-created data can break DNS resolution and global load balancing for your application.

Manage Keys and Certificates

You can upload different keys, transport layer security (TLS) certificates, and certificate chains to Tanzu Service Mesh for different scenarios where secure, encrypted communication needs to be established for services in your service mesh.

You need to upload certificates in different situations where secure, encrypted connections need to be made to services in your service mesh, such as when [public services](#) are used.

You need to upload certificate chains in scenarios where you want Tanzu Service Mesh to integrate with an external service that uses self-signed certificates, such as when integration with an external certificate authority (CA) service is needed.

For instructions on adding TLS certificates and certificate chains to Tanzu Service Mesh, click the following links.

Add Certificates

As an application operator, you can upload transport layer security (TLS) certificates to Tanzu Service Mesh for use in different situations where secure, TLS-encrypted connections need to be established to services in your service mesh, such as when public services are used.

This procedure describes how to add a new certificate to Tanzu Service Mesh.

Prerequisites

Verify the following prerequisites:

- Verify that you are familiar with public-key infrastructure (PKI) concepts *certificate*, *private key*, *certificate authority (CA)*, and *certificate chain*.
- You have a certificate and a private key from a trusted certificate authority (CA) and know the location of the certificate and private key files. The certificate file must be in PEM (.pem) format. The private key file must be in PEM or KEY (.key) format.
- Users in your organization can configure a public service to be accessible at an HTTPS URL and select a certificate in the public service configuration to encrypt HTTPS traffic to the service. To ensure that the certificate works correctly, verify that it matches the domain specified for the public service. For more information about public services, see [Chapter 6 Create a Public Service](#).
- [Access the Tanzu Service Mesh Console](#).

Procedure

- 1 In the navigation pane on the left, click **Admin > Keys & Certificates**.
- 2 On the **Keys & Certificates** page, on the **Keys & Certificates** tab, click **New Certificate**.
- 3 In the **New Certificate** dialog box, provide the following information.
 - **Name**. The name of the certificate to help distinguish it from other certificates in Tanzu Service Mesh. The name can contain only alphanumeric characters and underscores (_) and cannot contain numbers and special characters, such as ampersands (&) and pound signs (#). It must contain a minimum of 2 characters and a maximum of 1,024 characters.
 - (Optional) **Description**. An optional description of the certificate.
 - **Certificate File**. Click **Select .PEM File** and browse to the certificate file that you want to upload.
 - **Private Key**. Click **Select .PEM /.KEY File** and browse to the private key file.
 - (Optional) **Certificate Chain**. Optionally upload the CA certificate chain file.

Note **Certificate Type** specifies the type of certificate. Currently, only user-defined certificates are available.

- 4 Click **Save**.

Results

The new certificate is added to the table on the **Keys & Certificates** page. To edit or delete the certificate, click the three vertical dots to the left of the certificate name in the table and click **Edit** or **Delete** on the menu. The table on the **Keys & Certificates** page displays details about the certificate, including the following details:

- The name of the certificate
- The date and time when the certificate becomes valid
- The date and time when the certificate expires
- Details of the certificate issuer (common name, organization, and organizational unit if specified)
- The organization to which the certificate was issued
- The certificate authority (CA) that issued the certificate
- The certificate serial number

Note If some of the details about the certificate are not visible in the table, in the upper-right of the table, click **Column Settings** and select the check box next to each column that you want to show in the table.

The certificate is available for selection in public service configurations that specify HTTPS. If a user selects the certificate for a public service, Tanzu Service Mesh will attach the certificate to the domain of the public service to encrypt traffic to the service.

Add Certificate Chains

If you want Tanzu Service Mesh to integrate with an external service that uses self-signed certificates, you must upload the associated certificate chain or certificate authority (CA) bundle to Tanzu Service Mesh so that Tanzu Service Mesh can trust that external service's certificates.

A common scenario for uploading a certificate chain is when you need to integrate with an external CA service, and that CA service issues certificates signed not by a publicly trusted CA, but by an internal CA. In that case, you must upload the CA's certificate chain to Tanzu Service Mesh and then reference the certificate chain in the CA integration account. This way, Tanzu Service Mesh will trust the certificates signed by the CA.

For example, you may want to integrate with Vault CA so that the services in your service mesh can make authenticated and encrypted connections to each other by using mutual TLS (mTLS) certificates generated by Vault CA. Since certificates issued by a Vault server are signed by an internal CA, you must upload the CA bundle in use by the server to Tanzu Service Mesh and then reference the CA bundle in the appropriate Vault CA integration account. During onboarding, if you associate a cluster with the Vault CA integration, the root and intermediate certificates

from the CA bundle will be imported into the cluster's trust store, and the cluster will use these CA certificates to make trusted HTTPS connections to the Vault server and trust the mTLS certificates provided by the server. For more information about Vault integration, see [Create a Vault CA Integration Account](#).

Note

- Certificate chains and CA bundles that you upload to Tanzu Service Mesh are different from transport layer security (TLS) certificates that you upload to Tanzu Service Mesh. Those certificates are used in different situations where secure inbound connections need to be established to services in your service mesh. For example, certificates are used to establish secure communications between clients and [public services](#) in Tanzu Service Mesh. For more information about uploading certificates to Tanzu Service Mesh, see [Add Certificates](#).
 - For simplicity, the following procedure refers only to certificate chains, but the information applies to both certificate chains and CA bundles.
-

Prerequisites

- Verify that you are familiar with public-key infrastructure (PKI) concepts *certificate chain*, *certificate authority (CA)*, *self-signed certificate*, *CA bundle*, *root certificate*, and *intermediate certificate*.
- Know the location of the certificate chain file that you want to add. The file must be in PEM (.pem) format.
- Verify that the certificate chain file contains a maximum of three certificates, including only one root certificate and a maximum of two intermediate certificates. Certificate chain files containing a larger number of certificates cannot be uploaded.
- [Access the Tanzu Service Mesh Console](#).

Procedure

- 1 In the navigation pane on the left, click **Admin > Keys & Certificates**.
- 2 On the **Keys & Certificates** page, click the **Certificate Chains** tab.
- 3 Click **New Certificate Chain**.
- 4 In the **New Certificate Authority Chain** dialog box, provide the following information about the certificate chain.
 - **Name.** Enter a unique name for the certificate chain in Tanzu Service Mesh. Use a name that will help identify the file when you reference it in a CA integration account.
 - **Description.** Enter an optional description.
 - **Certificate Chain.** Click **Select .PEM file** and browse to the certificate chain file that you want to upload.
- 5 Click **Save**.

Results

A row for the new certificate chain is added to the table on the **Certificate Chains** tab.

What to do next

To view the fields for each certificate in an uploaded certificate chain, click the certificate chain name in the table. The fields of the root certificate appear at the top of the certificate chain details page, followed by the fields of each subsequent intermediate certificate in the chain. The fields of the server certificate appear last.

As an example, the following fields are displayed for the root certificate:

- Certificate Info
- Not Valid Before
- Not Valid After
- Common Name
- Organization
- Organizational Unit
- Serial Number
- SHA-1 Fingerprint
- SHA-256 Fingerprint
- Signature Algorithm
- Issuer
- Subject
- Public Key Algorithm
- Public Key Length

Note

- If no value appears next to a field, that field was not found in the certificate chain file.
- The Certificate Info field indicates the type of certificate in the certificate chain, for example, **Root CA Certificate**.

You can edit a certificate chain record to upload a different certificate chain file into the record or to edit the description of the certificate chain. To edit:

- 1 On the **Certificate Chains** tab, click the certificate chain name in the table.
- 2 On the certificate chain details page, click **Edit Certificate Chain**.
- 3 In the **Edit Certificate Authority Chain** dialog box, make the changes you want and click **Save**.