

# VMware Tanzu Mission Control Concepts

VMware Tanzu Mission Control

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](http://www.vmware.com)

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

# Contents

	<i>About VMware Tanzu Mission Control Concepts</i>	4
<b>1</b>	<b>What is Tanzu Mission Control</b>	5
	Tanzu Mission Control Versions	7
<b>2</b>	<b>Access Control</b>	9
<b>3</b>	<b>Users and Groups</b>	12
<b>4</b>	<b>Cluster Lifecycle Management</b>	14
	Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control	17
<b>5</b>	<b>Observation and Analysis of Cluster Health and Resources</b>	19
	What Is a Healthy Cluster	19
<b>6</b>	<b>Local Image Registry</b>	23
<b>7</b>	<b>Events and Audit Logs</b>	25
<b>8</b>	<b>Policy-Driven Cluster Management</b>	28
<b>9</b>	<b>Cluster Inspections</b>	31
<b>10</b>	<b>Data Protection</b>	34
<b>11</b>	<b>What Happens When You Attach a Cluster</b>	39
	Memory and CPU Usage by Cluster Agent Extensions	40
<b>12</b>	<b>Pod Security Management</b>	42

# About *VMware Tanzu Mission Control Concepts*

The *VMware Tanzu Mission Control Concepts* documentation provides conceptual information describing VMware Tanzu Mission Control™.

To help you get started with Tanzu Mission Control, this information provides a conceptual walk-through of what it is, has, and does. This publication introduces the Tanzu Mission Control service and console, and the concepts surrounding how you can use it to manage your Kubernetes clusters.

## Intended Audience

This information is intended for administrators who want to use Tanzu Mission Control to create and manage Kubernetes clusters and their associated resources. This information is also intended for application administrators and developers who want to use Tanzu Mission Control to deploy and manage modern apps in a Kubernetes architecture. The information is written for developers who have a basic understanding of Kubernetes and are familiar with container deployment concepts. In-depth knowledge of Kubernetes is not required.

# What is Tanzu Mission Control

# 1

VMware Tanzu Mission Control is a platform for modern application management that provides a single control point for teams to more easily manage Kubernetes and operate modern containerized applications across multiple clouds and clusters.

As an API-driven service, Tanzu Mission Control enables you to declaratively manage all your clusters through its API, the CLI, or the web-based console.

Tanzu Mission Control provides instances of the service in regions around the world, including Australia, Canada, Ireland, Japan, USA, and most recently in Mumbai, India.

From the Tanzu Mission Control console, you can see your clusters and namespaces, and organize them into logical groups for easier management of resources, apps, users, and security. Some of the cluster management capabilities of Tanzu Mission Control include:

- Cluster Lifecycle Management

Using Tanzu Mission Control, you can connect to your own cloud provider account to create new clusters, resize and upgrade them, and delete clusters that are no longer needed. For more information, see [Chapter 4 Cluster Lifecycle Management](#).

- Cluster Observability and Diagnostics

See the health and resource usage for each of your clusters from a single console. View cluster details, namespaces, nodes, and workloads directly from the Tanzu Mission Control console. For more information, see [Chapter 5 Observation and Analysis of Cluster Health and Resources](#).

- Cluster Inspections

Run preconfigured inspections against your clusters using Sonobuoy to ensure consistency over your fleet of clusters. For more information, see [Chapter 9 Cluster Inspections](#).

- Data Protection

Back up and restore the data resources in your clusters using Velero to ensure the protection of the valuable data resources in your clusters. For more information, see [Chapter 10 Data Protection](#).

- Access Control

Tanzu Mission Control starts with a secure by default service, and allows you to use federated identity management and apply granular role-based access control to fine tune your security requirements. For more information, see [Chapter 2 Access Control](#).

- Policy Management

Rather than manually dealing with the many aspects of managing your Kubernetes resources and the apps that use them, you can create policies to consistently manage your clusters, namespaces, and workloads. For more information, see [Chapter 8 Policy-Driven Cluster Management](#).

Most of this feature set is available in Tanzu Mission Control regardless of the Tanzu edition that you use. Some features, like certain policy types and inspection types, are available only if you are using Tanzu Advanced Edition. For more information about the features available in Tanzu Mission Control for each edition, see the [VMware Tanzu Mission Control Feature Comparison Chart](#). For more information about Tanzu Advanced Edition, see [Overview of Tanzu Advanced Edition](#).

## Organizing Clusters and Namespaces

Tanzu Mission Control provides a hierarchy of objects to help you manage your resources. At the top of the structure is the organization, which typically correlates to a business or possibly a line of business within a large enterprise.

Under the organization are grouping objects that allow you to separate the concerns of the people that use your resources.

Through the Tanzu Mission Control console you can organize and view your Kubernetes resources in two different ways, enabling operations administrators to maintain control over clusters and infrastructure while allowing application teams self-serve access to namespaces.

- Cluster groups provide an infrastructure view.

Cluster groups allow you to organize your Kubernetes clusters into logical groupings, for example to align with business units. To get you started, Tanzu Mission Control provides a `default` cluster group, but you should create cluster groups to fit your business needs.

- Workspaces provide an application view

Workspaces allow you to organize your managed namespaces into logical groups across clusters, perhaps to align with development projects. To get you started, Tanzu Mission Control provides a `default` workspace, but you should create workspaces to fit your business needs.

By combining your resources into groups, you can simplify management by applying policies at the group level. For example, you can apply an access policy to an entire cluster group rather than creating separate policies for each individual cluster.

Also in the Tanzu Mission Control hierarchy, within the organization, is the credential object. A credential correlates to the role you use to connect to a particular cloud provider account for cluster lifecycle management. The credential object is not directly associated with any cluster group or workspace, so you can create clusters for multiple cluster groups through a single credential.

## Attached and Provisioned Clusters

Using Tanzu Mission Control, you can attach existing Kubernetes clusters from various cloud providers, organize them into logical groups, observe their health, and manage their security posture and configuration. For more information, see [Chapter 11 What Happens When You Attach a Cluster](#).

You can also provision clusters directly through Tanzu Mission Control, provisioned in your own cloud provider account using Cluster API, to leverage the built-in cluster lifecycle management best practices. For more information, see [Chapter 4 Cluster Lifecycle Management](#).

Clusters must belong to a cluster group. When you attach or provision a cluster in Tanzu Mission Control, you specify the cluster group to which the cluster belongs.

## Managed and Unmanaged Namespaces

In both attached and provisioned clusters, you can create namespaces that you can manage through Tanzu Mission Control using policies. When you create a namespace through Tanzu Mission Control, you specify the workspace to which the namespace belongs.

Your clusters can also have unmanaged namespaces that were created externally and don't need to be managed through Tanzu Mission Control. However, if you have unmanaged namespaces that you want to manage, you can attach them to your organization using Tanzu Mission Control to better monitor and secure them.

This chapter includes the following topics:

- [Tanzu Mission Control Versions](#)

## Tanzu Mission Control Versions

Learn about the feature sets available in each version of VMware Tanzu Mission Control.

Tanzu Mission Control is offered in multiple versions, tailoring the set of features to your cluster management needs.

Many of the features and capabilities are available in all versions. Some features, like certain policy types and inspection types, are available only with some versions. For more information about the features available in Tanzu Mission Control for each version, see the [VMware Tanzu Mission Control Feature Comparison Chart](#).

## Tanzu Mission Control Essentials

Crafted specifically for Tanzu services on VMware Cloud offerings like VMware Cloud on AWS, Tanzu Mission Control Essentials provides a set of essential capabilities to organize your Kubernetes clusters and namespaces for scalable operations, and secure them with access control policies.

## Tanzu Mission Control Standard

The standard version of Tanzu Mission Control comes with Tanzu Standard Edition, and provides an expanded set of capabilities for enterprises to manage Kubernetes across clusters, clouds, and teams. In addition to the basic multi-cloud, multi-cluster management capabilities included in Tanzu Mission Control Essentials, the standard version also offers capabilities like data protection and conformance inspection.

## Tanzu Mission Control Advanced

The advanced version of Tanzu Mission Control is available standalone and contains all currently available functionality. This includes custom roles for access policies, additional policy types like networking, quota, and image registry policies, as well as custom policies for more granular control over your Kubernetes deployments.

# Access Control

# 2

Use access policies to implement role-based access control (RBAC) for the users and resources in your organization.

For a manageable security posture, VMware Tanzu Mission Control allows you to secure the resources in your organization with access policies that govern the users and groups that can see and edit them. This section discusses access policies; see [Chapter 3 Users and Groups](#) for more information about combining users into manageable groups.

## Managing Service Security Posture with Access Policies

Access policies allow you to control the permissions granted to the users of Tanzu Mission Control in your organization. Each object in your organizational hierarchy has an access policy where you can specify permissions using role bindings that associate a role with an identity. For more information about the organizational hierarchy, see [Chapter 1 What is Tanzu Mission Control](#).

In addition to the direct policy defined for a given object, each object also has inherited policies that are defined in the parent objects. For more information about policy inheritance, see [Chapter 8 Policy-Driven Cluster Management](#).

There are predefined roles for each type of object in your organization. Each role defines a set of permissions that apply to a given type of object. By contrast, the access policy where you define the role binding defines the scope to which the role applies. To learn how to create role bindings in Tanzu Mission Control, see [Managing Access to Your Resources](#) in *Using VMware Tanzu Mission Control*.

For example, the `cluster.edit` role grants permission to make edits to a cluster. If that role is bound to a given identity (group or individual) in the access policy for a cluster, this set of permissions applies only to the cluster. But if you bind the `cluster.edit` role to an identity in a cluster group access policy, members included in that identity are granted permissions to edit all clusters included in that cluster group.

The following table shows the roles you can use in access policies for each type of object in the hierarchy.

Table 2-1.

Role	organization	cluster group / cluster / workspace	namespace	Notes
<code>.admin</code>	x	x	x	Grants full root-level access to the object, including permission to see and edit access policies.
<code>.edit</code>	x	x	x	Grants permission to view the object, and create and delete child objects.
<code>.view</code>	x	x	x	Grants permission to see the object and its resources and child objects.
<code>.create</code>			x	Applicable only on the workspace, grants permission to create a namespace. Identities with this role are assigned the <code>.admin</code> role on the namespace that they create.
<code>.credential.admin</code>	x			Grants permission to create and edit credentials for data protection, connections to cloud provider accounts, and other features.
<code>.credential.view</code>	x			Grants permission to see and use credentials for data protection, connections to cloud provider accounts, and other features.

## Best Practices for Assigning Roles

When assigning roles, consider the following best practice guidelines.

- Use groups in role bindings rather than individual identities.
- Use a Kubernetes service account identity in role bindings for permissions that are not tied to individuals.
- Assign only the roles that grant the permissions necessary for an entity to perform its function within the organization.
- Use the `.admin` role judiciously and sparingly. The `.admin` role allows full root access to all of the resources and policies of an object, and recursively for its child objects, from within Tanzu Mission Control and also directly in the cluster.

## About Roles in VMware Cloud Services

For services in the VMware Cloud Services platform, the organization provides two roles, owner and member. As an organization owner, you can specify the roles of members in your organization, both at the organizational level and at the service level. For the Tanzu Mission Control service, there are two service roles:

- Service Member

This role provides typical service usage permissions for most members in your organization.

- Service Admin

This role provides additional permissions for administrators of the service in your organization.

As an organization owner, you can also invite additional members to your organization, and specify the organization and service roles in the invitations that you send out. For information about assigning roles in VMware Cloud Services and inviting users to join your organization, see [Identity & Access Management](#) in the *Using VMware Cloud* documentation.

## Initial Security Posture for Default Resource Groups

The initial setup for your organization in Tanzu Mission Control contains a cluster group named `default` and a workspace named `default`. To help you get started, these `default` resource groups have relaxed permissions that allow all authenticated users associated with the service member role to create and manipulate clusters and other resources.

As a best practice, create new cluster groups and workspaces for both development and production activities, and apply appropriate access control to them. Use the `default` cluster group and workspace only to initially familiarize your users with the service.

# Users and Groups

# 3

Manage the users in your organization with user groups.

For a manageable security posture, VMware Tanzu Mission Control allows you to combine the members of your organization into logical user groups and secure those groups with access policies. This section addresses user groups; see [Chapter 2 Access Control](#) for more information about implementing access policies that include those user groups.

As a service provided through VMware Cloud Services, the top-level group of users in Tanzu Mission Control is the organization, which is a construct of the VMware Cloud Services platform. Access to services is provided through the organization and individuals are included as members of the organization.

## Managing Users and Groups

You use VMware Cloud Services tools to invite users to your organization and organize them into user groups. By combining your users into groups, you can simplify access control by creating access policies that bind roles to groups rather than individuals. For more information about creating user groups in VMware Cloud Services, see [Working with Groups](#) in the *Using VMware Cloud* documentation.

You can also set up federation with your corporate domain that allows you to use your organization's single sign-on and identity source. For more information about federating identity management, see [What is enterprise federation and how does it work](#) in the *Using VMware Cloud Services Console* documentation.

## Best Practice for Creating Groups

The VMware Cloud Services tools for user and group management allow you to create user groups in two ways:

- Add users and then combine them into groups.
- Create groups and then add users to them.

As a best practice, add users through the group to which they initially belong. Use the **Groups** tab under Identity and Access Management in the VMware Cloud Services console, rather than the **Active Users** tab. In this way, the new user is added to a group to which you have already assigned roles through an access policy. If you use the Active Users tab, the new user is added to the organization and service, but because they are not yet added to a group, they will likely have only minimal access to the service until you take the additional step of adding them to a group.

## About Roles in VMware Cloud Services

For services in the VMware Cloud Services platform, the organization provides two roles, owner and member. While these roles provide a base set of permissions for each individual, they do not have an impact on the groups to which an individual can belong, or the service-level roles to which an individual or group can be bound.

For more context around these roles, see [Chapter 2 Access Control](#).

# Cluster Lifecycle Management

# 4

VMware Tanzu Mission Control allows you to have complete control over the entire lifecycle of provisioned Tanzu Kubernetes clusters, from create to delete and everything in-between.

When you create a cluster in Tanzu Mission Control, you have control over its entire lifecycle. In addition to scaling node pools up and down, creating and deleting namespaces, and other capabilities that are available in attached clusters, you can also create and delete clusters as necessary.

When you register a Tanzu Kubernetes Grid management cluster or create a cloud provider account connection in Tanzu Mission Control, you can provision clusters and leverage the built-in cluster lifecycle management best practices of Cluster API and Tanzu Kubernetes Grid. Tanzu Mission Control uses the Cluster API declarative pattern of lifecycle management for continuous monitoring and reconciliation of your clusters. By contrast, in clusters that have been created elsewhere and subsequently attached, you can have read/write privileges that allow you to control many aspects of the cluster with some limitations.

---

**Note** You use Tanzu Mission Control to monitor and manage your Kubernetes clusters, both attached and provisioned. However, while VMware monitors clusters for aggregate health trends to proactively investigate widespread issues, Tanzu Mission Control does not actively diagnose and cannot repair many kinds of issues with individual clusters, such as failures in Kubernetes or the underlying cloud provider. For example, if the API server in your cluster is not responsive to API calls, Tanzu Mission Control might report that it is unhealthy or disconnected, but does not attempt to proactively fix the issue. If you have an issue with a provisioned cluster that you think might be caused by the Tanzu Mission Control service or the Tanzu Kubernetes Grid implementation, refer to your support contract for how to address this kind of issue.

---

## What is a Tanzu Kubernetes Cluster?

A Tanzu Kubernetes cluster is an opinionated installation of Kubernetes open-source software that is built and supported by VMware. It is part of a Tanzu Kubernetes Grid instance that includes the following components:

- management cluster - a Kubernetes cluster that performs the role of the primary management and operational center for the Tanzu Kubernetes Grid instance
- provisioner - a namespace on the management cluster that contains one or more workload clusters

- workload cluster - a Tanzu Kubernetes cluster that runs your application workloads

Tanzu Kubernetes clusters include those created using the following technologies:

- Tanzu Kubernetes Grid
- Tanzu Kubernetes Grid Service
- Tanzu Community Edition

To manage the lifecycle of your Tanzu Kubernetes clusters in Tanzu Mission Control, you must register the management cluster. After you register the management cluster, you can identify the existing workload clusters in that Tanzu Kubernetes Grid instance that you want to manage through Tanzu Mission Control. You can also create new workload clusters in a registered management cluster.

For information about the minimum requirements for registering a management cluster, see [Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control](#).

In vSphere with Tanzu, the functionality of the management cluster is provided through the vSphere Supervisor Cluster, and a provisioner is called a vSphere namespace. For more information about Tanzu Kubernetes Grid Service in vSphere with Tanzu, see [vSphere with Tanzu Configuration and Management](#).

For more information about Tanzu Kubernetes Grid and Tanzu Kubernetes clusters, see [VMware Tanzu Kubernetes Grid](#).

For more information about Tanzu Community Edition managed clusters, see [Getting Started with Managed Clusters](#).

## What Happens When You Create a Cluster using Tanzu Mission Control

When you create a cluster, Tanzu Mission Control performs the following actions:

- provisions the necessary resources in your specified cloud account
- creates a Tanzu Kubernetes cluster according to your specifications
- attaches the cluster to your organization

## Resource Usage in Your Cloud Provider Account

For each cluster you create, Tanzu Mission Control provisions a set of resources in your connected cloud provider account.

For development clusters that are not configured for high availability, Tanzu Mission Control provisions the following resources:

- 3 VMs

The VMs include a control plane node, a worker node (to run the cluster agent extensions), and a bastion host. If you specify additional VMs in your node pool, those are provisioned as well.

- 4 security groups (one for the load balancer and one for each of the initial VMs)
- 1 private subnet and 1 public subnet in the specified availability zone
- 1 public and 1 private route table in the specified availability zone
- 1 classic load balancer
- 1 internet gateway
- 1 NAT gateway in the specified availability zone
- 1 VPC elastic IP

For production clusters that are configured for high availability, Tanzu Mission Control provisions the resource listed above and the following additional resources to support replication in two additional availability zones:

- 2 additional control plane VMs
- 2 additional private and public subnets
- 2 additional private and public route tables
- 2 additional NAT gateways
- 2 additional VPC elastic IPs

Your cloud provider implements a set of default limits or quotas on these types of resources, and allows you to modify the limits. Typically the default limits are sufficient to get started creating clusters from Tanzu Mission Control. However, as you increase the number of clusters you are running or the workloads on your clusters, you will encroach on these limits. When you reach the limits imposed by your cloud provider, any attempts to provision that type of resource fail. As a result, Tanzu Mission Control will be unable to create a new cluster, or you might be unable to create additional deployments on your existing clusters.

For example, if your quota on internet gateways is set to 5 and you already have five in use, then Tanzu Mission Control is unable to provision the necessary resources when you attempt to create a new cluster.

Therefore regularly assess the limits you have specified in your cloud provider account, and adjust them as necessary to fit your business needs.

## ConfigMap Usage for Custom Configurations

Tanzu Mission Control includes a ConfigMap that you use to customize certain elements of a cluster.

There may be not enough default pod resources to create more than five clusters concurrently when creating the clusters using Tanzu Mission Control. Use the ConfigMap to specify the parameter values for creating the lcm-tkg-extension pods. You can specify the CPU, memory, and concurrency resources for lcm-tkg-extension pods.

The default lcm-tkg-extension-concurrency is 5, the default lcm-tkg-extension-memory-request is 64Mi, and the default lcm-tkg-extension-memory-limit is 512Mi.

Edit the ConfigMap to specify the parameter values as shown in the example below.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: tmc-lcm-config
  namespace: vmware-system-tmc
data:
  lcm-tkg-extension-concurrency: "6"
  lcm-tkg-extension-memory-request: 128Mi
  lcm-tkg-extension-memory-limit: 1Gi
```

You can also enable FIPS using the ConfigMap, as well as private registries. For more information, refer to the Tanzu Kubernetes Grid documentation.

## Audit Logging in Your Provisioned Cluster

When you provision a new Tanzu Kubernetes cluster through Tanzu Mission Control, preconfigured audit logging is enabled for the cluster. For more information, see [Chapter 7 Events and Audit Logs](#).

This chapter includes the following topics:

- [Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control](#)

## Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control

Learn about the minimum requirements for Tanzu Kubernetes clusters that you manage through VMware Tanzu Mission Control and the platforms that are supported for this integration.

### Configuration Requirements for Registering Tanzu Kubernetes Clusters

To effectively use Tanzu Mission Control to manage your Tanzu Kubernetes clusters, make sure that your clusters abide by the following configuration guidelines.

- Your Tanzu Kubernetes Grid management cluster must be a production cluster with multiple control plane nodes.
- Tanzu Kubernetes Grid workload clusters need at least 4 CPUs and 8 GB of memory.

- To add a Tanzu Kubernetes Grid workload cluster to Tanzu Observability, the cluster must have a minimum of two worker nodes.

## Supported Environments for Registering Tanzu Kubernetes Clusters

Tanzu Mission Control supports the registration of management clusters running in the following environments:

- Tanzu Kubernetes Grid Service Supervisor Clusters running in vSphere with Tanzu or running in VMware Cloud with Tanzu services.
- Tanzu Kubernetes Grid management clusters (version 1.3.1 or 1.4.1 or later) running in Microsoft Azure.
- Tanzu Kubernetes Grid management clusters (version 1.4.1 or later) running in Amazon Web Services (AWS).
- Tanzu Kubernetes Grid management clusters (version 1.3 or later) running in vSphere on Azure VMware Solution (AVS).
- Tanzu Kubernetes Grid management clusters (version 1.2 or later) running in vSphere, including vSphere on VMware Cloud on AWS (version 1.12 or 1.14 or 1.16 or 1.18 or 1.20).
- Tanzu Community Edition management clusters (version 0.11 or later) running in Microsoft Azure, Amazon Web Services, or vSphere.

Do not attempt to register any other kind of management cluster with Tanzu Mission Control.

- Tanzu Mission Control does not support the registration of Tanzu Kubernetes Grid management clusters prior to version 1.2.
- Tanzu Mission Control does not support the registration of Tanzu Community Edition management clusters running in Docker, or releases prior to version 0.11.

## CNI requirement for managing the lifecycle of a workload cluster

After you register a management cluster with Tanzu Mission Control, you can manage the lifecycle of workload clusters (create, upgrade, and delete) in that management cluster.

The Antrea CNI (container network interface) is required for cluster lifecycle management through Tanzu Mission Control. Although you can manage clusters that use an alternative CNI, such as Calico, you will not be able to upgrade the Kubernetes version on that cluster. If you intend to manage the lifecycle of a workload cluster through Tanzu Mission Control, make sure the cluster is created using the Antrea CNI.

# Observation and Analysis of Cluster Health and Resources

# 5

Use VMware Tanzu Mission Control to see what's happening in your clusters.

The Clusters page in the Tanzu Mission Control console displays a list of all the clusters attached to your organization that you have permissions to see. Similarly, the Namespaces page shows your namespaces (both managed and unmanaged), and the Workloads page shows the workloads running on your clusters. You can filter and sort the lists of objects to see only those you want to see.

From these top-level pages, you can drill down into the detail views of the objects and their parent and child objects to see a fine-grained description of what's happening in your clusters, nodes, namespaces, and workloads.

For example, the cluster detail page shows the health of the cluster's nodes and the components and cluster agent extensions running on the cluster. For a description of how Tanzu Mission Control determines the health of a cluster, see [What Is a Healthy Cluster](#).

## Tanzu Observability by Wavefront

In addition to the basic observability provided directly through Tanzu Mission Control, you can connect to your Wavefront account to capture metrics from your clusters. When you connect your cluster to an existing Wavefront account, Tanzu Mission Control installs an extension on your cluster to collect data and send it to your Wavefront account in one minute intervals. You can then log in to your Wavefront account to configure data collection and analyze the captured metrics. For more information about Tanzu Observability by Wavefront, see [What is Wavefront](#) in the *Tanzu Observability by Wavefront* documentation.

This chapter includes the following topics:

- [What Is a Healthy Cluster](#)

## What Is a Healthy Cluster

Learn how Tanzu Mission Control determines the health of attached and provisioned clusters.

The cluster detail page for each cluster in the Tanzu Mission Control console shows current overall health of the cluster at the top of the page. This health status is also displayed in the list of all clusters on the Clusters page. Additionally, further down the cluster detail page, more health information is broken out into detailed aspects of the overall health. Tanzu Mission Control continuously monitors each cluster and updates the console with changes.

The cluster agent extensions that are deployed on your cluster (both provisioned and attached) send change events from nodes and ports as they occur, and a regularly occurring component status event for each component to Tanzu Mission Control. These events are regarded collectively as the heartbeat, which Tanzu Mission Control uses to determine the health of the cluster.

## Cluster Health

The overall health of the cluster is an aggregation of health of the components and nodes in the cluster. The health status of the cluster can be one of the following values.

- **HEALTHY**

A cluster is healthy when all nodes and components are healthy, and a heartbeat for the cluster is received every minute.

- **UNHEALTHY**

A cluster is unhealthy if either of the following are reported as unhealthy:

- one or more of the cluster's control plane nodes
- one or more of the cluster's components

- **WARNING**

A cluster can have a warning status if any of its worker nodes are in an unhealthy or unknown state.

A cluster can also have a warning status if any nodes (worker or control plane) are in a warning state.

- **UNKNOWN**

The health status of a cluster is unknown if either of the following are reported as unknown:

- one or more of the cluster's control plane nodes
- one or more of the cluster's components

- **DISCONNECTED**

A cluster is considered disconnected if no heartbeat is received from the cluster for more than 3 minutes.

## Node Health

The title of the Worker nodes section shows you how many worker nodes you have in the cluster, and below that the number of worker nodes that are healthy. To see all the nodes (including the control plane), click the **Nodes** tab, which shows the health of each individual node in the Status column.

The information received in the change event for a node consists of the `NodeReady` condition, and a number of other conditions like `MemoryPressure`, `DiskPressure`, and `OutOfDisk`. Tanzu Mission Control uses the value of these conditions to determine the health of the node. The status of each condition is assumed to be unchanged until the node reports a change. Based on the reported conditions, the health status of a node can be one of the following:

- HEALTHY

If `NodeReady` is `True`, and all other conditions are healthy, then the node is healthy.

- UNHEALTHY

The node is unhealthy if `NodeReady` is `False`. The node is also unhealthy if `NodeReady` is `True` and more than half of the other conditions are in an unhealthy state.

- WARNING

The warning status indicates that `NodeReady` is `True`, but some (less than half) of the other conditions are in an unhealthy state.

- UNKNOWN

If `NodeReady` has any value other than `True` or `False`, the health status of the node is unknown. The node can also have an unknown status if no heartbeat has been received from the cluster for more than three minutes.

## Component Health

Tanzu Mission Control monitors the health of the following components running in the cluster:

- `kube-apiserver`
- `scheduler`
- `controller-manager`
- one or more `etcd` components (`etcd-0`, `etcd-1`, `etcd-2`, and so on)

The component status event reports the `Healthy` condition for each of these components every 45 seconds. The health status of each component can be one of the following:

- HEALTHY

If the last reported value of the `Healthy` condition of the component is `True`, then the component is healthy.

- UNHEALTHY

If the last reported value of the `Healthy` condition of the component is `False`, then the component is unhealthy.

- UNKNOWN

If the last reported value of the `Healthy` condition of the component is `Unknown`, or it is something other than `True` or `False`, then the health status of the component is unknown. The component can also be in this state if no heartbeat has been received from the cluster for more than three minutes.

# Local Image Registry

# 6

You can configure Tanzu Mission Control to pull its agent and extension images from a private, local OCI registry instead of from the Tanzu Mission Control SaaS registry.

A local image registry enables you to configure Tanzu Mission Control to pull its agent and extension images from your private registry (Harbor, for example). You can also perform security and vulnerability scans on all images on your private registry before deploying workloads to your production clusters.

Using your own local image registry makes Tanzu Mission Control SaaS suitable for users that need to perform security and vulnerability scans on all images before deploying to their production clusters. Edge users have more control over when Tanzu Mission Control images are pulled/replicated across the thin network pipes that connect many edge sites to central data centers.

Support for local image registry lets you:

- Use your own local/private OCI image registry.
- Configure Tanzu Mission Control to deploy its agents and extensions from your private registry.
- Run security and vulnerability scans on Tanzu Mission Control cluster components before deploying to your production clusters.
- Control when Tanzu Mission Control agent and extension images are replicated from the Tanzu Mission Control SaaS registry to your private registry (e.g., off-peak for edge locations).

Supported flows include:

- Attach a cluster
- Register a Tanzu Kubernetes Grid management cluster
- Provision/Manage a Tanzu Kubernetes Grid workload cluster
- Configure Policy/IAM
- Configure Data Protection

Supported registry configurations include:

- HTTP, HTTPS

- Unauthenticated
- Authenticated
- Internal/Trusted CA Cert (for clusters preconfigured with CA certs by the user)

**Table 6-1. Supported Versions**

Functionality	Supported Version
Inspection	Kubernetes Versions 1.23 and later
LCM of Tanzu Kubernetes Grid clusters	Tanzu Kubernetes Grid on vSphere v1.6.1 and later

---

**Note** Tanzu Observability and Tanzu Service Mesh are not supported for use with local image registry.

---

# Events and Audit Logs

# 7

Learn how VMware Tanzu Mission Control logs events that occur in your organization.

To help you monitor the activities that occur in your organization, Tanzu Mission Control provides logging of audit events.

This includes audit logging of service-level actions that are performed through Tanzu Mission Control, as well as cluster-level actions that take place on Tanzu Kubernetes clusters that you provisioned through Tanzu Mission Control.

Tanzu Mission Control also logs the prior three days of cluster-level interactions and system state changes.

Those changes may be the creation, deletion, or modification of a cluster, cluster group, or namespace. It also covers actions such as running inspections, applying policies, and more.

## Cluster Management Events

Tanzu Mission Control generates events for user activity and system state change, including cluster-level interactions that occur between Tanzu Mission Control and your managed clusters. The Events page in the Tanzu Mission Control console shows the last 72 hours of events across your entire fleet of Kubernetes clusters.

The kinds of events that you'll see on this page include cluster health and lifecycle events, inspections, as well as policy insights. In addition to the event summaries listed on the Events page, you can expand each event to view the body of the event or click the link to go to the object where it occurred. This makes it easier to access audit log events on the Tanzu Mission Control console along with regular cluster events to provide a comprehensive view.

You can filter the list of events by name and type to see only the events you want to see using the rich filter capability in Tanzu Mission Control. In addition, specific audit events are available from the Tanzu Mission Control console, without the need to request the download of an audit log file and subsequent offline processing in a log viewer to see the content.

## Tanzu Mission Control Service Audit Logging

As an organization administrator, you can monitor the activities that are initiated through Tanzu Mission Control, using the service audit logging capability.

To enable you to track and understand the activities that impact the clusters in your organization, Tanzu Mission Control collects and stores logs of audit events that describe activities that occur through Tanzu Mission Control. Each log entry describes the following:

- what was done
- who performed the action
- when and where the action occurred

Because some actions result in subsequent actions, Tanzu Mission Control logs all pertinent events from the original action performed through Tanzu Mission Control to the resulting actions that are initiated on your clusters.

Be aware of the following constraints on audit reports:

- Tanzu Mission Control event log entries are retained in the service for 60 days, so you can generate an audit report going back up to 60 days.
- The maximum date range for an audit report is seven days.
- Generated audit reports are retained for 60 days after generation, after which they are deleted.

## Audit Logging in Your Provisioned Cluster

When you provision a new Tanzu Kubernetes cluster through Tanzu Mission Control, preconfigured audit logging is enabled for the cluster. Although the audit logging policy is not configurable, logging is set to commonly useful levels for certain operations to capture important information without inflating the size of the logs. The logging levels are listed below with the types of operations that use each one.

**None** - No logs are collected for the following types of operations:

- kube-proxy watching endpoint resource
- kubelet getting nodes and node status
- Controller manager, scheduler and endpoint controller reading the endpoints
- API server getting namespaces and namespace finalizers
- Controller manager getting metrics
- Read-only APIs like `/swagger`, `/version`, and `/healthz`
- All getting events
- Garbage collector getting resources
- Tanzu Mission Control getting resources

**Metadata** - Only metadata about the operation (and not any data that was passed in or out) is collected for the following types of operations:

- Config maps, secrets, and token reviews - because they can contain sensitive data

`Request` - The information that came in the request is collected for the following types of operations:

- kubelet updating nodes and node status
- All read-only operations - log the request, but not the data returned back

`RequestResponse` - Both the request and the returned data are collected for the following types of operations:

- All other operations, including write operations like create, update, and delete

In addition, the location, size, and duration of audit log files are configured as described below.

- Audit log entries are written to `audit.log` files in the `/var/log/kubernetes/` directory on the control plane nodes for your cluster. You can aggregate the logs, if desired, using a log collector and a log aggregation service.
- The maximum size for a log file is 100 MB. When this limit is reached, a new log file is started. The rotated files are also stored in the `/var/log/kubernetes/` directory.
- The maximum number of log files stored on a node is 20 (for a total max size of 2 GB for all stored logs). When this limit is reached, a new log file replaces the oldest log file.
- A log file is stored for a maximum of 7 days, after which it is deleted. This limit is not often hit on a typical cluster, as clusters running workloads tend to churn through the maximum number of log files more quickly.

## Import Logs into Your Preferred Logging System

You can ingest Tanzu Mission Control events, including audit logs, via an event stream API, a secure and scalable solution. For instructions on how to set up automated ingestion of events using the streaming API, [visit this VMware developer resource](#).

## Best Practices for Naming Resource Fields

Users are advised to not enter any Personally Identifying Information (PII) into any of the resource name fields such as cluster name, cluster group name, workspace name, or namespace name. These fields can be customized as text fields by users for easy reference but are not meant to include any PII information such as user name, email address, or any such identifying information.

It is strongly recommended as a best practice to avoid any accidental exposure of any PII entered into these non-PII fields by the users who have access to Tanzu Mission Control in their organization. These fields are handled as per rules applicable to non-PII fields and any PII info entered into these non-PII fields is at the sole discretion of the user understanding the potential risk involved.

# Policy-Driven Cluster Management



In VMware Tanzu Mission Control you can create policies of various types to manage the operation and security posture of your Kubernetes clusters and other organizational objects.

Policies allow you to provide a set of rules that govern your organization and all the objects it contains. The policy types available in Tanzu Mission Control include the following:

- access policy

Access policies allow you to use predefined roles to specify which identities (individuals and groups) have what level of access to a given resource. For more information, see [Chapter 2 Access Control](#).

- image registry policy

Image registry policies allow you to specify the source registries from which an image can be pulled.

- network policy

Network policies allow you to use preconfigured templates to define how pods communicate with each other and other network endpoints.

- quota policy

Quota policies allow you to constrain the resources used in your clusters, as aggregate quantities across specified namespaces, using preconfigured and custom templates. For more information, see [Managing Resource Consumption in Your Clusters](#) in *Using VMware Tanzu Mission Control*.

- security policy

Security policies allow you to manage the security context in which deployed pods operate in your clusters by imposing constraints on your clusters that define what pods can do and which resources they have access to. For more information, see [Chapter 12 Pod Security Management](#).

- custom policy

Custom policies allow you to implement additional business rules, using templates that you define, to enforce policies that are not already addressed using the other built-in policy types. For more information, see [Creating Customized Policies](#) in *Using VMware Tanzu Mission Control*.

---

**Note** Some policy types are available only in the advanced version of Tanzu Mission Control. For more information, see the [VMware Tanzu Mission Control Feature Comparison Chart](#).

---

## Policy Inheritance

In the Tanzu Mission Control resource hierarchy, there are three levels at which you can specify policies.

- organization
- object groups (cluster groups and workspaces)
- Kubernetes objects (clusters and namespaces)

In addition to the direct policy defined for a given object, each object also has inherited policies that are defined in the parent objects. For example, a cluster has a direct policy and also has inherited policies from the cluster group and organization to which it is attached.

## Labels and Selectors

A label is a key/value pair attached to a Kubernetes object (such as a namespace, node, or pod) that allows you to specify identifying attributes for that object. A selector provides the means to identify the objects that have a given label.

Some types of policies allow you to specify a label selector, which identifies the objects that you want to include or exclude from the policy.

When you specify a label selector to include (or exclude) in a policy, the objects of the given type within the scope of the policy that have the specified label are included (or excluded) in the set that are impacted by the policy. If no label selectors are specified, the policy applies to all such objects within the scope of the policy.

For example, say you create a quota policy with the following details:

- created at the cluster group level on the `mycompany-project1` cluster group
- namespace label selector to include: `project1-phase:phase02`
- namespace label selector to exclude: `basic:compliance`

When this quota policy is saved, it impacts all clusters in the `mycompany-project1` cluster group, constrains the namespaces with the `project1-phase:phase02` label, and does not impact the namespaces with the `basic:compliance` label.

When adding label selectors, make sure you click the **Add label selector** button after entering the key and values. This action applies the label selector to the policy and opens a new entry row on the form. Each row (in both include and exclude) represents a rule that must match to identify the objects for which the policy is effective. These selector rules are applied as follows:

- A rule with a key and one value is a match if a label with the specified key exists on the object and it equals the specified value.
- A rule with a key and no value is a match if a label with the specified key exists on the object, regardless of the value.
- A rule with a key and multiple, comma-separated values is a match if a label with the specified key exists on the object and it equals at least one of the specified values.
- When multiple label selector rules are specified, all of the rules must be satisfied to identify the object.

For more information about label selectors, see [Labels and Selectors](#) in the Kubernetes documentation.

# Cluster Inspections

# 9

Using VMware Tanzu Mission Control, you can run preconfigured cluster inspections using Sonobuoy, an open source community standard.

---

**Note** The cluster inspection features of Tanzu Mission Control are not available in Tanzu Mission Control Essentials. Also, the CIS Benchmark inspection type is only available in Tanzu Mission Control Advanced.

---

The following cluster inspections are available from the Overview and Inspection tabs of the cluster detail page in the Tanzu Mission Control console.

- The **Conformance** inspection validates the binaries running on your cluster and ensures that your cluster is properly installed, configured, and working. You can view the generated report from within Tanzu Mission Control to assess and address any issues that arise. For more information, see the Kubernetes Conformance documentation at <https://github.com/cncf/k8s-conformance/tree/master/docs>.
- The **CIS benchmark** inspection evaluates your cluster against the CIS Benchmark for Kubernetes published by the [Center for Internet Security](#). This inspection type is available in the advanced version of Tanzu Mission Control.
- The **Lite** inspection is a node conformance test that validates whether nodes meet requirements for Kubernetes. For more information, see [Validate node setup](#) in the Kubernetes documentation.

Because the cluster inspections provide a point-in-time report of the condition of the cluster, you might want to run them periodically (to avoid drifting out of conformance) and any time you make significant alterations, such as after you patch or upgrade a cluster.

From the Inspections page in the Tanzu Mission Control console, you can view a list of the most recent inspections that have been run against all the clusters in your organization, along with the results of those inspections. This page also allows you to start a new inspection.

## About the CIS Benchmark Inspection and Provisioned Tanzu Kubernetes Clusters

The CIS Benchmark for Kubernetes is a set of opinionated and generalized tests that assess vulnerabilities in a Kubernetes implementation. In many implementations, including Tanzu Kubernetes clusters provisioned through Tanzu Mission Control, the vulnerabilities tested for in the benchmark can be mitigated in ways that are not detected by the inspection and result in a failure of some tests. For this reason, you can expect to see a failed status for a particular set of tests when you run the CIS benchmark inspection on a provisioned cluster. The following table shows the CIS benchmark test failures that occur in clusters provisioned through Tanzu Mission Control.

**Table 9-1. Expected Test Failures for CIS Benchmark Inspection on Provisioned Tanzu Kubernetes Clusters**

Section	Test Description	Explanation
1.1.12	Ensure that the etcd data directory ownership is set to etcd:etcd (Scored)	<p>The data directory (<code>/var/lib/etcd</code>) is owned by <code>root:root</code>.</p> <p>This is not an issue on clusters provisioned through Tanzu Mission Control.</p> <p>To provision clusters, Tanzu Mission Control uses Cluster API which, in turn, uses the <code>kubeadm</code> tool to provision Kubernetes.</p> <p><code>kubeadm</code> makes <code>etcd</code> run containerized as a static pod, therefore the directory does not need to be set to a particular user.</p> <p><code>kubeadm</code> configures the directory to not be readable by <code>non-root</code> users.</p>
1.2.6	Ensure that the <code>--kubelet-certificate-authority</code> argument is set as appropriate (Scored)	<p>This flag is not set.</p> <p>The <code>kubelet</code> server certificate is used only when running <code>kubect1 exec</code> or <code>kubect1 logs</code> or gathering metrics. Tanzu Mission Control configures worker nodes to exist in private subnets in AWS, and therefore "man in the middle" attacks require host networking to be compromised.</p> <p>Though unlikely, VMware views this as a potential security vulnerability and is tracking the issue in the Cluster API community.</p>
1.2.33	Ensure that the <code>--encryption-provider-config</code> argument is set as appropriate (Scored)	<p>This flag is not set.</p> <p>Tanzu Mission Control provides isolation for each cluster using AWS VPCs and security groups. This level of isolation significantly reduces the likelihood of a security breach involving access to an unencrypted volume.</p>
1.3.6	Ensure that the <code>RotateKubeletServerCertificate</code> argument is set to true (Scored)	<p>This argument is set to true in clusters running Kubernetes version 1.12 and later. To avoid this error, upgrade your cluster to a newer version of Kubernetes.</p>

**Table 9-1. Expected Test Failures for CIS Benchmark Inspection on Provisioned Tanzu Kubernetes Clusters (continued)**

Section	Test Description	Explanation
4.1.3	Ensure that the proxy kubeconfig file permissions are set to 644 or more restrictive (Scored)	In clusters provisioned through Tanzu Mission Control, this issue is mitigated by running the <code>kube-proxy</code> as a daemonset, which does not use an on-disk kubeconfig.  Therefore, the audit cannot be performed, as the file is mounted as a config map in the <code>kube-proxy</code> pod and is not accessible to the CIS Inspection pod.
4.1.4	Ensure that the proxy kubeconfig file ownership is set to root:root (Scored)	In clusters provisioned through Tanzu Mission Control, this issue is mitigated by running the <code>kube-proxy</code> as a daemonset, which does not use an on-disk kubeconfig.  Therefore, the audit cannot be performed, as the file is mounted as a config map in the <code>kube-proxy</code> pod and is not accessible to the CIS Inspection pod.
4.2.4	Ensure that the <code>--read-only-port</code> argument is set to 0 (Scored)	In clusters provisioned through Tanzu Mission Control, this issue is mitigated by disabling the read-only port.
4.2.6	Ensure that the <code>--protect-kernel-defaults</code> argument is set to true (Scored)	This flag is not set.  The CIS is concerned that without kernel default protection set, a pod might be run in the cluster that is a mismatch for the security posture of the cluster as a whole. This is a low-likelihood occurrence.
4.2.10	Ensure that the <code>--tls-cert-file</code> and <code>--tls-private-key-file</code> arguments are set as appropriate (Scored)	In clusters provisioned through Tanzu Mission Control, this issue is mitigated by disabling the read-only port.  By default, the TLS certificate and private key files are readable only by root.
4.2.12	Ensure that the <code>RotateKubeletServerCertificate</code> argument is set to true (Scored)	This argument is set to true in clusters running Kubernetes 1.12 and later. To avoid this error, upgrade your cluster to a newer version of Kubernetes.

## About the CIS Benchmark Inspection and Attached Clusters

Some cloud providers do not allow access to the control plane node of clusters running in their platform (for example, GKE and AKS). Because of this limitation, the CIS benchmark inspection cannot run all tests on clusters running in these environments. Therefore there are fewer results from running a CIS benchmark inspection on an attached cluster running in such an environment.

# Data Protection

# 10

Using VMware Tanzu Mission Control, you can protect the valuable data resources in your Kubernetes clusters using the backup and restore functionality provided by Velero, an open source community standard.

The data protection features of Tanzu Mission Control allow you to create the following types of backups for managed clusters (both attached and provisioned):

- all resources in a cluster
- selected or excluded namespaces in a cluster
- specific or excluded resources in a cluster identified by a given label

You can selectively restore the backups you have created, by specifying the following:

- the entire backup
- selected or excluded namespaces from the backup
- specific or excluded resources from the backup identified by a given label

Additionally, you can schedule regular backups and manage the storage of backups and volume snapshots you create by specifying a retention period for each backup and deleting backups that are no longer needed.

When you perform a backup for a cluster, Tanzu Mission Control uses Velero to create a backup of the specified Kubernetes resources with snapshots of persistent volume data, and then stores the backup in the location that you specify.

---

**Note** The namespaces `kube-system`, `velero`, `tkg-system`, and `vmware-system-tmc` are not included in backups.

---

For more information about Velero, visit <https://velero.io/docs>.

For information on how to use the data protection features in Tanzu Mission Control, see [Protecting Data](#) in *Using VMware Tanzu Mission Control*.

---

**Note** The data protection features of Tanzu Mission Control are not available in Tanzu Mission Control Essentials.

---

## About Backup Storage

For the storage of your backups, you can specify a target location that allows Tanzu Mission Control to manage the storage of backups, provisioning resources as necessary according to your specifications. However, if you prefer to manage your own storage for backups, you can also specify a target location that points to a storage location that you create and maintain in your cloud provider account, such as an AWS S3 or S3-compatible storage location or an Azure Blob storage location. With self-provisioned storage, you can leverage existing storage investments for backups, reducing network and cloud storage costs, and apply existing storage policies, quotas, and encryption. For a list of supported S3-compatible providers, see [S3-Compatible object store providers](#) in the Velero documentation.

Before you define a backup for a cluster, you must create a target location and credential that you will use to perform the backup.

- The data protection credential specifies the access credentials for the account where your backup is stored. This account can be either your AWS account where Tanzu Mission Control manages backup storage, or an account where you manage backups (the account that contains your AWS S3 or S3-compatible storage or the subscription that contains your Azure Blob storage).
- The data protection target location identifies the place where you want the backup stored, and references the associated data protection credential. You can share the target location across multiple cluster groups and clusters.

## About Volume Backup

Tanzu Mission Control leverages Velero and supports backing up and restoring Kubernetes volumes. Velero Supports backing up and restoring volumes using the File System Backup (FSB) method and the Container Storage Interface (CSI) snapshot method.

## File System Backup

Kubernetes volumes attached to pods can be backed up from the file system of the volumes. This approach is called File System Backup (FSB) or Pod Volume Backup. Tanzu Mission Control uses Velero with restic to achieve this.

FSB can be enabled while enabling data protection on the cluster. You can also enable or disable FSB from the data protection page on the cluster. On enabling FSB, restic gets installed on the cluster and Velero backs up all pod volumes using restic. Data on the volumes backed up using FSB, will be copied to the backup storage location using restic.

FSB offers two approaches of discovering pod volumes to be backup::

- Opt-in approach: Every pod containing a volume to be backed up using FSB must be annotated with the volume's name using the `backup.velero.io/backup-volumes` annotation.

- Opt-out approach: All pod volumes are backed up using FSB, with the ability to opt-out any volumes that should not be backed up using the `backup.velero.io/backup-volumes-excludes` annotation on the pod.

For more information about Velero, see [Velero Backup](#).

If FSB is enabled on your cluster, FSB with Opt-out is the default setting for all backup operations. If FSB is enabled, during backup operation all volumes are evaluated based on the specified Opt-out or Opt-in approach to exclude or include volumes to be backed up using FSB.

Benefits of using FSB include:

- It is capable of backing up and restoring almost any type of Kubernetes volume. Therefore, if you need a volume type that doesn't have the concept of a native snapshot or CSI volume snapshot, FSB might be the best choice.
- It is not tied to a specific storage platform, so you could save the backup data to a different storage platform from the one backing Kubernetes volumes, for example, a durable storage media.

However, you should also be aware that:

- FSB backs up data from the live file system, so the backup data is less consistent than the CSI volume snapshot approach.
- It accesses the file system from the mounted hostpath directory, so the pods need to run as root user and even under privileged mode in some environments.

---

**Note** If both FSB and CSI volume snapshot are enabled, Velero will first backup volumes using FSB based on the specified approach (opt-in or opt-out). Volumes not included in FSB backup will be backed up by CSI volume snapshot if they meet prerequisites given in [Requirements for CSI Volume Backup](#) in *Using Tanzu Mission Control*. It is recommended to enable both FSB and CSI volume snapshots in case the cluster has volume types which do not support CSI volume snapshots. Such volumes can be backed up using FSB.

---

## Container Storage Interface (CSI) Volume Snapshot

Velero supports backup and restore of CSI driver backed volumes using the [Requirements for CSI Volume Backup](#) in *Using Tanzu Mission Control*. To create a CSI snapshot, it requires a volume snapshot class that tells the Kubernetes engine which driver file to use when creating snapshots.

CSI snapshot is available only for persistent volumes created using CSI drivers supporting volume snapshot. Data on the CSI snapshots is not copied to the backup storage location. You should check with your cloud provider about the snapshot durability.

---

**Note** CSI Snapshot is not available for TKG clusters.

---

For more information about CSI prerequisites, see [Requirements for CSI Volume Backup](#) in *Using Tanzu Mission Control*.

## FSB and CSI Usage

TMC allows you to enable and disable FSB and CSI snapshot independently.

- 1 If only FSB is enabled, Velero evaluates volumes to be backed up using FSB based on the specified approach (Opt-in or Opt-out). During restore, FSB backed up volumes will always be restored.
- 2 If both FSB and CSI volume snapshot are enabled and backup configured with CSI snapshot, Velero first evaluates volumes to be backed up using FSB based on the specified approach (Opt-in or Opt-out). CSI-driver based volumes not included in FSB backup are backed up by CSI volume snapshot if they meet the prerequisites stated in [Requirements for CSI Volume Backup](#) in *Using Tanzu Mission Control*.
- 3 If both FSB and CSI volume snapshot are enabled but backup is not configured with CSI snapshot, Velero will evaluate volumes to be backed up using FSB based on the specified approach (Opt-in or Opt-out). Volumes not included in FSB will not be backed up.
- 4 If only CSI Snapshot is enabled and backup configured with CSI snapshot, Velero will backup only CSI-driver based volumes if they meet prerequisites stated in [Requirements for CSI Volume Backup](#). None of the non-CSI driver based volumes will be backed up.
- 5 If neither FSB nor CSI snapshot is enabled, no volumes will be backed up.

It is recommended to enable both FSB and CSI volume snapshot, otherwise if the cluster has non-CSI driver based volume types those volumes will not be backed up. In addition, you may use the Opt-in or Opt-out approach to exclude the CSI-based volumes from FSB so they can be backed up using CSI snapshot.

## About Backup Restoration Between Different Clusters

When you create a backup using Tanzu Mission Control, that backup can be available for restoration to other clusters in your organization. This feature allows you to create a backup in one cluster and restore it to a different cluster, even clusters running on different platforms.

When migrating workloads between clusters running different versions of Kubernetes, consider the availability of resources in each version and the compatibility of API groups for each custom resource. If the source and target clusters are running different versions of Kubernetes, keep the following in mind:

- A Kubernetes version downgrade (restoring to a cluster running a lower version of Kubernetes) can cause incompatibility of core API groups and other issues associated with feature availability. Use this approach judiciously.
- If a Kubernetes version upgrade (restoring to a cluster running a higher version of Kubernetes) causes incompatibility of core API groups, you must update the impacted custom resources in the source cluster prior to creating the backup.

For example, `IngressClass` in `networking.k8s.io/v1beta1` API is no longer supported as of Kubernetes version 1.22.

For more information, see <https://velero.io/docs/main/migration-case/> in the Velero documentation.

You cannot restore a backup that contains restic volumes on cluster without restic. Additionally, you can restore a backup that contains volume snapshots to another cluster only if both clusters share the same cloud provider account.

When migrating workloads between clusters running on different cloud providers, consider the following items:

- By default, persistent volume claims (PVCs) might fail to bind to volumes because the appropriate storage class from the source cluster doesn't exist in the target cluster. To make sure your volumes bind, use a storage class map as described in <https://velero.io/docs/v1.8/restore-reference/#changing-pvpvc-storage-classes> in the Velero documentation.

For example, the following configmap maps the `default` and `managed-premium` storage classes from an AKS cluster to the `gp2` storage class in an EKS cluster.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: change-storage-class-config
  namespace: velero
  labels:
    velero.io/plugin-config: ""
    velero.io/change-storage-class: RestoreItemAction
data:
  # Map the "default" and "managed-premium" storage classes backed by AzureDisk on
  # the source cluster to "gp2", a storage class backed by AWS EBS on the current
  # (destination) cluster.
  default: gp2
  managed-premium: gp2
```

- Custom resources from the source cluster might not exist in the target cluster.

For example, `tiers.crd.antrea.io` and `tiers.security.antrea.tanzu.vmware.com` from a Tanzu Kubernetes cluster are not found in an AKS cluster.

You can exclude resources during restore to help avoid this issue.

- Resource differences between the source and target cluster might impact functionality.

Some packages install webhooks that can cause issues when the source and target are not the same cluster.

For example, `mutatingwebhookconfiguration.admissionregistration.k8s.io` and `validatingwebhookconfiguration.admissionregistration.k8s.io` from an AKS will impact the functionality of an EKS cluster.

You can exclude resources during restore to help avoid this issue.

# What Happens When You Attach a Cluster

# 11

When you attach a cluster to your organization in VMware Tanzu Mission Control, the cluster agent service creates a namespace and installs a set of cluster agent extensions and custom resource definitions into your cluster. The cluster agent service and its extensions enable Tanzu Mission Control to communicate with your cluster (for example, to capture health information) and manage your cluster (for example, to enforce policies).

This attachment is required for all clusters that you manage through Tanzu Mission Control, including management clusters that you register with Tanzu Mission Control and their workload clusters (both pre-existing and provisioned), as well as the clusters that you create elsewhere and subsequently attach to Tanzu Mission Control.

The extension manager is one of the extensions installed into your cluster when you attach it. The extension manager oversees any other extensions that the cluster agent installs into your cluster as a result of using the capabilities of Tanzu Mission Control. For example, when you run a conformance inspection on your cluster, the cluster agent installs an extension that runs Sonobuoy. For information about the resource consumption of the cluster agent extensions, see [Memory and CPU Usage by Cluster Agent Extensions](#).

The process of attaching a cluster involves three steps:

- 1 Register the cluster with the Tanzu Mission Control service.
- 2 Install the cluster agent extensions on the cluster.
- 3 Verify the connection to confirm bilateral communication.

You must have `admin` permissions on the cluster to install the cluster agent extensions, and the extension manager retains `admin` permissions to make modifications to the cluster as necessary. The extensions that are subsequently installed by the extension manager have their own individual security profiles granting them only the permissions necessary to perform their function.

When you attach a cluster, be aware of the following:

- On the cluster, you must have `cluster.admin` permissions to install the cluster agent extensions.
- In Tanzu Mission Control, you must be associated with the `clustergroup.edit` role in a cluster group to attach a cluster.

- Cluster names must be unique within an organization.
- An attached cluster must belong to exactly one cluster group. For more information about cluster groups, see [Chapter 1 What is Tanzu Mission Control](#).

## Outbound Connections Made by the Cluster Agent Extensions

The cluster agent extensions running on the cluster make connections to Tanzu Mission Control URLs for outbound communications. This applies to Tanzu Kubernetes clusters that you register with Tanzu Mission Control as well as clusters that were created elsewhere and subsequently attached.

Outbound connections made by the cluster agent extensions communicate with Tanzu Mission Control through port 443. If you have a proxy server that manages outbound traffic for your clusters, you need to add the following URLs to your proxy allowlist to enable the cluster to communicate with Tanzu Mission Control.

- `*.tmc.cloud.vmware.com`

URLs in this domain include the Tanzu Mission Control service for your organization, as well as authentication, authorization, the Tanzu Standard package repository, and other services.

- `console.cloud.vmware.com`

This URL is required for logging in with the Tanzu Mission Control CLI. This is necessary only if you use the command-line interface from behind your proxy server.

- `registry.k8s.io`

This URL is required for running inspections in Tanzu Mission Control.

- `wavefront.com`

This URL is required for communication with Tanzu Observability.

You must also make sure that the proxy-related environment variables (`HTTP_PROXY`, `HTTPS_PROXY`, and `NO_PROXY`) are defined for the cluster's environment.

This chapter includes the following topics:

- [Memory and CPU Usage by Cluster Agent Extensions](#)

## Memory and CPU Usage by Cluster Agent Extensions

Learn how much memory and CPU usage is consumed by the cluster agent extensions that are installed by VMware Tanzu Mission Control on managed Kubernetes clusters, both attached and provisioned.

When you attach a cluster, register a management cluster, or create a workload cluster in a registered management cluster in Tanzu Mission Control, the cluster agent service installs a set of cluster agent extensions and custom resource definitions into your cluster to enable Tanzu Mission Control to communicate with your cluster.

In the following table, memory is expressed in mebibytes/gibibytes and CPU is expressed in cores/millicores.

**Table 11-1. Memory and CPU Reservations and Limits for Cluster Agent Extensions**

Extension	Node	Memory Reservation	Memory Limit	CPU Reservation	CPU Limit
agent-updater	Master	100Mi	150Mi	100m	100m
agentupdater-workload	Worker	100Mi	150Mi	100m	100m
cluster-health-extension	Worker	128Mi	2Gi	100m	1000m
manager (data-protection)	Worker	128Mi	512Mi	50m	100m
extension-manager	Master	100Mi	150Mi	100m	100m
extension-updater	Master	128Mi	512Mi	50m	100m
manager (inspection-extension)	Worker	128Mi	256Mi	10m	500m
intent-agent	Worker	150Mi	150Mi	100m	100m
manager (policy-sync-extension)	Worker	128Mi	256Mi	10m	500m
manager (policy-webhook)	Worker	128Mi	256Mi	100m	100m
sync-agent	Worker	128Mi	2Gi	100m	2000m
tmc-observer	Master	100Mi	150Mi	50m	100m

Note that information about all the extensions is shown in the agent and extensions health section of the cluster details page in the Tanzu Mission Control console. Some extensions consist of more than one pod, for example, the policy-webhook pod is part of the policy-sync-extension. You can check number of extensions running using this command:

```
kubectl get extensions
```

## Kubernetes Resource Units

For detailed information about kubernetes resource units and measurements, see <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/#resource-units-in-kubernetes>.

# Pod Security Management

# 12

Use security policies to impose constraints on your clusters that define what pods can do and which resources they have access to.

For a manageable security posture, VMware Tanzu Mission Control allows you to exercise control over activity in the clusters in your organization with security policies that govern certain aspects of pod execution in the cluster. These aspects, which are described in [Pod Security Standards](#) in the Kubernetes documentation, include privileged containers, volume types, privilege escalation, and Linux capabilities. Although security policies in Tanzu Mission Control are not implemented using the Kubernetes native PodSecurityPolicy object, the security-sensitive aspects of the pod specification that they control is the same.

Security policies in Tanzu Mission Control are implemented using the Gatekeeper project from Open Policy Agent (OPA Gatekeeper). For more information, see the [OPA Gatekeeper documentation](#).

---

**Note** Current Kubernetes security policy standards (k8s v1.23+) can be found here: <https://kubernetes.io/docs/concepts/security/pod-security-standards/>.

As per these standards, in restricted/strict policy mode, `runAsNonRoot` should always be true and `runAsUser` can be undefined or non-zero.

However, in strict security-policy mode, Tanzu Mission Control does not enforce `runAsNonRoot` and enforces `runAsUser` to be non-zero (undefined is not allowed).

---

**Note** This feature is available in Tanzu Mission Control only if you are using Tanzu Advanced Edition.

---

## Inheritance and Precedence

Because security policies control how pods are deployed on a cluster, they apply to the clusters hierarchy (infrastructure view) rather than the namespace hierarchy (application view). You can implement security policies on a single cluster, on a cluster group, or at the organizational level, and they are inherited down through the hierarchy.

In contrast to native Kubernetes pod security policies, where the least restrictive policy takes precedence, security policies in Tanzu Mission Control enforce all aspects of all applied policies, each to the most restrictive extent defined. You cannot relax the constraints of an inherited security policy by implementing a less restrictive policy on a child object.

## What Happens When You Add a Security Policy

When you add a security policy, Tanzu Mission Control applies the policy to each cluster impacted by the policy. If this is the first security policy for a cluster, Tanzu Mission Control installs an extension in the cluster, and then applies the policy.