

Using VMware Tanzu Mission Control

VMware Tanzu Mission Control

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

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

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

Copyright © 2024 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to <https://www.broadcom.com>. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

	<i>About Using VMware Tanzu Mission Control</i>	10
1	Log In to the Tanzu Mission Control Console	11
2	Log In with the Tanzu CLI	12
3	Managing Your Tanzu Mission Control Subscription	13
4	Managing Account Credentials	15
	Connect an AWS Account for Data Protection	16
	Create a Stack for TMC-Managed Data Protection	17
	Create a Data Protection Credential for Self-Provisioned Storage	18
	Create a Tanzu Observability Credential Object	19
	Edit a Tanzu Observability Credential Object	20
	Remove an Account Credential	21
5	Connecting Through a Proxy	23
	Create a Proxy Configuration Object	24
	Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu	25
	Create a Proxy Configuration Object for AKS Clusters in Tanzu Mission Control	27
	Create a Proxy Configuration Object for EKS Clusters in Tanzu Mission Control	27
	Delete a Proxy Configuration Object	28
6	Managing Clusters	30
	Create a Cluster Group	30
	Delete a Cluster Group	31
	Attach a Cluster	32
	Attach a Cluster Running Behind a Proxy Server Using the CLI	34
	Reattach a Cluster	35
	Detach a Cluster	36
	Manage Detached Cluster Resources	37
	Remove a Cluster from Your Organization	38
	Move a Cluster Between Cluster Groups	41
	Connect to a Managed Cluster with kubectl	41
	Enable Access to Public Cloud Clusters Through Tanzu Mission Control	42
7	Managing Namespaces	45

- Create a Workspace 45
- Delete a Workspace 46
- Create a Managed Namespace 46
- Delete a Managed Namespace 47
- Attach a Namespace 48

8 Managing the Lifecycle of Azure AKS Clusters 49

- Create an Azure AKS Credential 49
- Edit an Azure AKS Credential 51
- Delete an Azure AKS Credential 51
- Create an Azure AKS Cluster 52
- Edit an Azure AKS Cluster 54
- Delete an Azure AKS Cluster 55
- Add an Existing Azure AKS Cluster into Tanzu Mission Control Management 56
- Upgrade an Azure AKS Cluster 57
- Modify AKS Cluster Node Pools 57
- AKS Credential Resources 58

9 Managing the Lifecycle of AWS EKS Clusters 59

- Creating and Managing New AWS EKS Clusters 59
 - Constraints for Lifecycle Management of EKS Clusters 60
 - Create a VPC with Subnets for EKS Cluster Lifecycle Management 61
 - Create an Account Credential for EKS Cluster Lifecycle Management 62
 - Update Your Credential for an AWS EKS Account 63
 - Clean Up Your AWS EKS Account After Deleting a Credential 64
 - Create an EKS Cluster 65
 - Assume an EKS Cluster Lifecycle Role in Tanzu Mission Control 66
 - Enable IAM Principal Access to Clusters Managed by Tanzu Mission Control 67
- Adding Existing AWS EKS Clusters into Tanzu Mission Control Management 68
 - Update aws-auth Configmap for Access to Existing EKS Clusters 68
 - Add an Existing EKS Cluster into Tanzu Mission Control Management 70
- Tanzu Mission AWS EKS Control Tags 71

10 Managing the Lifecycle of Tanzu Kubernetes Clusters 73

- Register a Management Cluster with Tanzu Mission Control 74
 - Complete the Registration of a Management Cluster 76
 - Complete the Registration of a Supervisor Cluster in vSphere with Tanzu 78
- Deregister a Management Cluster from Tanzu Mission Control 80
 - Manually Remove the Cluster Agent from a Supervisor Cluster in vSphere with Tanzu 81
- Add a Workload Cluster into Tanzu Mission Control Management 83
- Remove a Workload Cluster from Tanzu Mission Control Management 84

Provisioning Tanzu Kubernetes Grid Workload Clusters	85
Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster	85
Provision a Cluster using a Cluster Class	86
Provision a Cluster in vSphere with Tanzu using a Cluster Class	90
Provision a Cluster in vSphere with Tanzu	94
Provision a Workload Cluster in vSphere	96
Provision a Workload Cluster in Azure	99
Provision a Workload Cluster in AWS	101
Advanced Options During Cluster Creation	103
Create a Node Pool	104
Edit a Node Pool	106
Delete a Node Pool	107
Relaxing Pod Security in a Provisioned Cluster	108
Upgrade Kubernetes on Your Cluster	109
Delete a Provisioned Cluster	110
Manage Certificates	110
11 Managing a Local Image Registry	112
Add a Local Image Registry for Tanzu Mission Control	113
Add a Local Image Registry for Tanzu Mission Control Using the CLI	113
Add a Local Image Registry for Tanzu Mission Control Using the API	114
Attach a Kubernetes Cluster with a Local Image Registry	117
Attach a Kubernetes Cluster with a Local Image Registry Using the CLI	117
Attach a Kubernetes Cluster with a Local Image Registry Using the API	118
Configuring Inspections for use with a Local Image Registry	119
Delete a Local Image Registry	120
Sync Images to Your Local Image Registry	121
12 Managing Cluster Resources with Continuous Delivery	124
Enable Continuous Delivery for a Cluster or Cluster Group	126
Disable Continuous Delivery	127
Create a Repository Credential for a Cluster or Cluster Group	128
Generate an SSH Key for Authentication to a Git Repository	129
Edit a Repository Credential	130
Delete a Repository Credential	130
Add a Git Repository to a Cluster or Cluster Group	131
Edit a Git Repository	132
Remove a Git Repository	133
Add a Kustomization to a Cluster or Cluster Group	133
Edit a Kustomization	135
Delete a Kustomization	135

13 Managing Cluster Secrets 137

- View Cluster Secrets 138
- Create a Kubernetes Secret in a Cluster or Cluster Group 139
- Edit a Cluster Secret 140
- Delete a Cluster Secret 141

14 Managing Integrations 143

- Enable Observability for Your Organization 143
- Disable Observability for Your Organization 144
- Add a Cluster to Observability 145
- Add a Cluster Group to Observability 146
- Rotate Your Observability Credentials 147
- Edit the Configuration of Your Tanzu Observability Collector 148
- Remove a Cluster from Observability 149
- Enable Service Mesh for Your Organization 149
- Disable Service Mesh for Your Organization 150
- Add a Cluster to Service Mesh 151
- Remove a Cluster from Service Mesh 152
- Upgrade Service Mesh 153
- Roll Back Service Mesh 153

15 Manage Object Labels 155

16 Managing Event Logs 157

- Generate an Audit Report 157
- Download an Audit Report 158
- Delete or Cancel an Audit Report 159

17 Managing Packages and Releases in Your Cluster 160

- View Packages 161
- Enable Helm Service on Your Cluster or Cluster Group 162
- Disable Helm Service 163
- Install a Helm Chart from a Helm Repository (Create a Release) 164
- Install a Helm Chart from a Git Repository 165
- Access Image Pull Secrets for Private Helm Charts 167
- Example Helm Release Using Private Image 168
- Edit an Installed Helm Release 169
- Delete an Installed Helm Release 170
- Install a Package 170
- Edit an Installed Package 173
- Delete an Installed Package 174

- [Add a Package Repository to Your Cluster](#) 174
 - [Edit a Custom Repository URL](#) 175
 - [Remove a Package Repository from Your Cluster](#) 176
 - [Disable a Package Repository in Your Cluster](#) 177

- 18 Install Tanzu Application Platform on a Cluster** 178
 - [Install Tanzu Application Platform on Multiple Clusters](#) 182
 - [Edit a Tanzu Application Platform Configuration](#) 187
 - [Delete a Tanzu Application Platform Configuration](#) 188

- 19 Protecting Data** 190
 - [Create a Target Location for Data Protection](#) 191
 - [Account Setup for Azure Blob Target Location](#) 192
 - [Delete a Target Location](#) 195
 - [Enable Data Protection for a Cluster](#) 196
 - [Disable Data Protection for a Cluster](#) 197
 - [Enable Data Protection on a Cluster Group](#) 198
 - [Define the Backup Schedule for a Cluster Group](#) 198
 - [Requirements for CSI Volume Backup](#) 200
 - [Back Up the Data Resources in Your Cluster](#) 201
 - [View the Contents of a Backup](#) 203
 - [Restore a Backup](#) 204
 - [Restore a Backup from a Different Cluster](#) 206
 - [View the Contents of a Restore](#) 209
 - [About Backup and Restore Hooks](#) 209
 - [Backup and Restore Hook Examples](#) 211

- 20 Inspecting Clusters** 218
 - [Start a Cluster Inspection](#) 219
 - [Stop a Cluster Inspection](#) 220
 - [View and Download Inspection Results](#) 220

- 21 Viewing Your Policies** 222
 - [View the Policy Assignments for an Object](#) 223
 - [View Policy Insights](#) 223
 - [About Policy Insights](#) 224
 - [Export Policy Code](#) 225
 - [View Your Access Policies](#) 226
 - [View Identities and Roles](#) 227
 - [View Access Roles and Permissions for Tanzu Mission Control](#) 228

22 Managing Access to Your Resources 230

- Add a Role Binding 231
- Edit a Role Binding 232
- Remove a Role Binding 233
- Add a Role Binding on a Credential 233
- Edit the Role Binding for a Credential 234
- Remove the Role Binding from a Credential 235
- Create a Custom Access Role 235
- Edit a Custom Access Role 237
- Delete a Custom Access Role 238

23 Managing Pod Security 240

- Create a Security Policy 241
- Edit a Security Policy 242
- Delete a Security Policy 243

24 Mutating Kubernetes Resources 244

- Create a Mutation Policy 245
- Edit a Mutation Policy 246
- Delete a Mutation Policy 247

25 Managing Access to Image Registries 248

- Create an Image Registry Policy 248
- Edit an Image Registry Policy 250
- Delete an Image Registry Policy 250

26 Managing Network Communication for Your Clusters 252

- Create a Network Policy 252
 - Rules in Network Policies 253
- Edit a Network Policy 255
- Delete a Network Policy 256

27 Managing Resource Consumption in Your Clusters 257

- Create a Quota Policy 257
- Edit a Quota Policy 258
- Delete a Quota Policy 259

28 Creating Custom Policies 260

- Create a Policy Template 260
- Delete a Policy Template 262
- Add a Custom Policy 262

- Edit a Custom Policy 264
- Delete a Custom Policy 264

29 Managing Administrative Settings 266

- Configure Policy Settings 267

30 Object Missing 268

- How to Get Support 268
- Reported Subscription Usage Doesn't Seem Right 269
- Issues Attaching a Cluster 270
- Issues Registering a Tanzu Kubernetes Grid Management Cluster 274
- Issues Registering a Supervisor Cluster 277
- Workload Cluster Provision Failures 283
- Troubleshooting Issues with Tanzu Application Platform 287
 - Tanzu Application Platform Packages Are Not Installed 287
 - Failed Disk Space Check for Tanzu Kubernetes Grid Cluster 288
 - Tanzu Developer Portal does not show cluster resources in multi-cluster TAP 1.7.3 deployment 289
- Troubleshooting Issues with Data Protection 290
 - Troubleshooting Issues with FSB or CSI 290
 - Unable to download backup or restore logs 291
 - Unable to download failed backup logs 291
 - Backups Partially Fail Due to Existing Lock 292
 - Source File Not Found During Backup 294
 - Backups Partially Fail When Volumes Present on Control Plane Nodes 295
 - Velero namespace stuck Terminating 297
 - Restore operation is stuck in pending state 298
- Manage Issues with AWS EKS Credentials in Tanzu Mission Control 299
- Monitoring of AWS GuardDuty for Unauthorized Access 303

31 Links to Examples, Blogs, and More Information 304

About *Using VMware Tanzu Mission Control*

The *Using VMware Tanzu Mission Control* documentation provides information about using VMware Tanzu Mission Control™.

To help you get started with Tanzu Mission Control , this documentation provides procedures showing how to organize and manage your clusters and other Kubernetes resources.

Important On September 11, 2024, cloud services from VMware Tanzu transitioned away from VMware Cloud Services Console to the new Tanzu Platform cloud services console. You now access VMware Tanzu cloud services from <https://console.tanzu.broadcom.com>, using your existing credentials. See [KB 374361](#) for information about the required actions to take, including updating Kubernetes collector configurations. See also [Using VMware Tanzu Platform cloud services console](#).

Intended Audience

This information is intended for platform administrators and operators 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.

Log In to the Tanzu Mission Control Console

1

Access the Tanzu Mission Control console to start managing clusters.

Follow these steps to log in to the Tanzu Mission Control console.

Prerequisites

You must be a member of Tanzu Platform Cloud Services organization that has access to Tanzu Mission Control to log in to the console.

Procedure

- 1 Open a browser and log in to the Tanzu Platform cloud services console.

<https://console.tanzu.broadcom.com/>

- 2 If you belong to multiple organizations, make sure you have selected the appropriate one.
To change organizations, click your name in the title bar.

- 3 Click the **Tanzu Mission Control** tile to open the Tanzu Mission Control console.

After you have logged in, you can return directly to the Tanzu Mission Control console using the following URL, replacing *org-name* with the name of your organization.

```
https://org-name.tmc.cloud.vmware.com/
```

Log In with the Tanzu CLI

2

Access your organization's VMware Tanzu Mission Control resources using the `tanzu` command-line interface.

This topic describes how to install the `tanzu` CLI and the plug-ins for Tanzu Mission Control.

Note

Prerequisites

You must be a member of a Tanzu Platform Cloud Services organization that has access to Tanzu Mission Control to log in with the CLI.

Before starting this procedure, make sure you have already logged in to the Tanzu Mission Control console.

If you intend to run the Tanzu Mission Control CLI from a location protected by a proxy server, you must add the following URL to your proxy allowlist.

```
console.tanzu.broadcom.com
```

The following procedure outlines the high-level flow of logging in at the command line. For a detailed description, see [Install and Configure the Tanzu CLI and Tanzu Mission Control Plug-ins in VMware Tanzu CLI Reference - Tanzu Mission Control Plug-ins](#).

Procedure

- 1 Retrieve an API token through Tanzu Platform Cloud Services.

Note An API token is not required for TMC Self-Managed.

- 2 Download and install the binary.
- 3 Initialize the CLI and then log in.

For a detailed description of these steps, see [Install and Configure the Tanzu CLI and Tanzu Mission Control Plug-ins in VMware Tanzu CLI Reference - Tanzu Mission Control Plug-ins](#).

What to do next

After you log in, you can manage your clusters at the command line.

Managing Your Tanzu Mission Control Subscription

3

Learn more about how to calculate the usage of your Tanzu Mission Control subscription.

What counts as usage?

Tanzu Mission Control is licensed according to how many CPUs, vCPUs, or cores you place under management. That means Tanzu Mission Control tracks overall CPU Capacity of your clusters, including any space that's reserved for system daemons that power the OS and Kubernetes. On other pages in the Tanzu Mission Control console, like the cluster details page, you see allocatable CPU reported, which is what's left over for your nodes after system-reserved, kube-reserved, and eviction-threshold have been subtracted from your CPU capacity. There's not typically a large difference between allocatable and CPU capacity in most environments. Learn more about [CPU Capacity and Allocatable CPU from Kubernetes](#) in the *Kubernetes documentation*.

How is usage calculated?

Tanzu Mission Control collects Kubernetes CPU information and approximates 2 Kubernetes CPUs = 1 Core or 2 vCPUs. For CPU-based subscriptions in vSphere environments, you can have up to 32 cores or 64 vCPU per CPU. However, you need to license your entire host to purchase CPU-based vSphere subscriptions.

How many cores do I need?

When managing Tanzu Kubernetes clusters in a vSphere environment, for example, say you have a 200-core vSphere cluster with half of the cluster housing traditional VMs, and the other half of the VMs used as TKG nodes. In this scenario, you would need 100 cores of TMC ($200 \text{ cores} / 2 = 100 \text{ cores}$). This is the same basic principle as your Tanzu Kubernetes Grid subscription.

When managing Kubernetes clusters in a public cloud environment, the calculation is similar. Say you have 400 vCPUs of EKS cluster nodes, where 1 core = 2 vCPUs (typical for a public cloud). In this scenario, you would need 200 cores of TMC ($400 \text{ vCPUs} / 2 = 200 \text{ cores}$).

About Tanzu Mission Control in vSphere+

Tanzu Mission Control Essentials with vSphere+ is applied to the same Tanzu Platform Cloud Services organization that redeems vSphere+ if you have one or multiple organizations.

Each organization that has an entitlement to Tanzu Mission Control behaves at the highest tier of Tanzu Mission Control redeemed in that organization. That is, if you already have Tanzu Mission Control Advanced in a given organization, and you activate vSphere+ in that organization, your Tanzu Mission Control instance has all the capabilities of Tanzu Mission Control Advanced, not just those in Tanzu Mission Control Essentials.

Managing Account Credentials

4

As a platform operator or infrastructure operator, use VMware Tanzu Mission Control to connect your cloud provider account for data protection and complete cluster lifecycle management.

When you register your cloud provider account in Tanzu Mission Control, you can use that connection, or credential, to manage data protection or provision Tanzu Kubernetes clusters and leverage the built-in cluster lifecycle management best practices of Cluster API. For more information, see [Data Protection](#) and [Cluster Lifecycle Management](#) in *VMware Tanzu Mission Control Concepts*.

About Cloud Provider Accounts

There is a one-to-one relationship between your cloud provider account and the connection you create to it in Tanzu Mission Control.

- For a cloud provider account, you create only one CloudFormation stack for Tanzu Mission Control.
- When you generate a template in Tanzu Mission Control to create a stack, it can be used only once.
- You can use a cloud provider account for only one connection in one organization. You cannot use the same stack or cloud provider account for connections in multiple organizations.

Read the following topics next:

- [Connect an AWS Account for Data Protection](#)
- [Create a Data Protection Credential for Self-Provisioned Storage](#)
- [Create a Tanzu Observability Credential Object](#)
- [Edit a Tanzu Observability Credential Object](#)
- [Remove an Account Credential](#)

Connect an AWS Account for Data Protection

Set up a cloud provider account connection (or credential), to enable you to perform data protection backups and restores in your cloud provider account through VMware Tanzu Mission Control.

To use data protection features with a target location managed by Tanzu Mission Control, you must first connect a cloud provider account. For more information, see [Data Protection in VMware Tanzu Mission Control Concepts](#).

Note If you want to use a data protection target location that you create and manage externally, see [Create a Data Protection Credential for Self-Provisioned Storage](#).

Prerequisites

Before you can set up a connection to your cloud provider account, make sure you have access to the account.

Also make sure you have the appropriate permissions to create the credential.

- To create a data protection credential, you must be associated with the `organization.credential.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the Credentials tab of the Administration page, click **Create Credential**, and then select the type of credential to create.

To use the AWS S3 storage managed by Tanzu Mission Control in your cloud provider account, select **AWS S3** under TMC provisioned storage.

- 3 On the Create credential page, provide a name for the credential, click **Generate template**, and then click **Next**.

The name that you enter is the name that appears in the list of connected accounts on the Administration page.

When you click **Generate template**, Tanzu Mission Control generates the template and then downloads it.

Note Do not reuse a template from a previously created stack. Each time you create a cloud provider account connection, you must download the template and create a new stack, even if you use the same AWS account.

- 4 In the AWS console, create a CloudFormation stack using the downloaded template, and when it completes retrieve the ARN.

For more information, see [Create a Stack for TMC-Managed Data Protection](#).

- 5 In the Tanzu Mission Control console, still on the Create credential page, click **Next** and then paste the role ARN that you copied from the AWS console.

- 6 Click **Create Credential** to create the connection to your cloud provider account.

Results

After you complete this procedure, you have a credential that you can use to perform actions through Tanzu Mission Control that require access to your cloud provider account. You can see your new credential listed on the Administration page in the Tanzu Mission Control console, and can choose that credential when you initiate an action that is dependent on your cloud provider account.

Create a Stack for TMC-Managed Data Protection

Create a CloudFormation stack in your AWS account that VMware Tanzu Mission Control can use to back up cluster resources, and get the ARN for the stack to complete the credential creation.

Prerequisites

Make sure you have access to the account and that you have prepared the account to allow Tanzu Mission Control to create clusters.

- 1 Log in to the AWS console.
- 2 Use the EC2 service to create an SSH key pair for each region that you plan to use with Tanzu Mission Control.

Note The SSH key pair is not required to set up the cloud provider account connection. However, Tanzu Mission Control requires an SSH key pair to create clusters. This key pair must exist for every region in which you want to create clusters. If you create a cloud provider account connection and subsequently attempt to use Tanzu Mission Control to create a cluster in a region for which you have not defined this key pair, cluster creation fails. This failure occurs later in the cluster creation process, and appears as though creation is simply stalled or stuck. Therefore, it is best to create the key pair in each region at the time you create the cloud provider account connection.

Procedure

- 1 In the AWS console, go to the CloudFormation service, and make sure you are in the region where you want to create the CloudFormation stack.
- 2 If you have previously used the `clusterawsadm` tool to create a stack, remove the stack.
Search for the stack `cluster-api-provider-aws-sigs-k8s-io`. If it exists, select the stack and click **Delete**.
- 3 On the Stack details page, click **Create stack** (with new resources).
- 4 On the Create stack page, in the Specify template area, click **Upload a template file**.
- 5 Click **Choose file**, select the template file you downloaded through the Tanzu Mission Control console, and then click **Open**.
- 6 Click **Next**.

- 7 On the Specify stack details page, provide a name for the stack, and then click **Next**.
- 8 On the Configure stack options page, accept all of the default values and click **Next**.
- 9 On the Review page, scroll to the bottom and select the checkbox that acknowledges the creation of IAM resources, and then click **Create stack**.

After a couple minutes, the Stack details page shows your new stack with the status of `CREATE_COMPLETE`. You might need to click the refresh button to update the status.

What to do next

After the stack is created, you can retrieve the role ARN. You need the role ARN to connect this CloudFormation stack in your AWS account to Tanzu Mission Control.

- 1 After the stack creation is complete, click the **Outputs** tab.
- 2 On the outputs tab, find the message created by the template that shows the role ARN for the stack.
- 3 Copy the role ARN shown in the message (for example, `arn:aws:iam::01234567890:role/clusterlifecycle.tmc.cloud.vmware.com`), and then return to the Tanzu Mission Control console to finish creating the connection.

Create a Data Protection Credential for Self-Provisioned Storage

Set up a credential for data protection that allows you to create a target location that uses a self-provisioned storage location that you create and maintain (either in your cloud provider account or in an on-premises data center).

To create backups using Tanzu Mission Control and save them in your self-provisioned storage (AWS S3 or S3-compatible or Azure Blob), you must first create a credential object that stores access credentials for the storage location.

Prerequisites

Before you set up a data protection credential that provides access to self-provisioned storage, make sure you have access to the account and that you have the credentials to access it.

Also make sure you have the appropriate permissions to create the credential object.

- To create a data protection credential, you must be associated with the `organization.credential.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the Accounts tab of the Administration a page, click **Create Account Credential**, and then select the type of credential to create.
 - For S3-compatible storage, choose Self provisioned storage: **AWS S3 or S3-compatible**.

- For Azure Blob storage, choose Self provisioned storage: **Azure Blob**.
- 3 On the Create credential page, provide a name for the credential.
- The name that you enter is the name that appears in the list of credential objects on the Accounts tab of the Administration page.
- 4 Provide the credentials required to access the storage location.

For AWS S3 or S3-compatible storage:

- Enter the access key ID and secret access key for your S3-compatible storage.

For Azure Blob storage:

- a Enter the following identifiers for the Blob storage in your Azure account:

- subscription ID
- tenant ID
- resource group
- client ID

- b Enter your client secret key.

- c Select the name of your Azure cloud from the dropdown list.

For more information about setting up your Azure subscription and retrieving the values for these fields, see [Account Setup for Azure Blob Target Location](#) .

- 5 After you finish configuring the access details for your storage account, click **Create**.

Results

After you complete this procedure, you have a credential that you can use to create a target location for data protection that points to a self-provisioned storage location. You can see your new credential listed on the Administration page in the Tanzu Mission Control console, and can choose that credential when you create a target location.

Create a Tanzu Observability Credential Object

Set up a credential object to connect to your Tanzu Observability by Wavefront account, to enable you to add clusters to Tanzu Observability through VMware Tanzu Mission Control.

Prerequisites

Before you create a Tanzu Observability credential, make sure you have access to the account and that you have generated a Tanzu Observability API token. For more information, see [Generate a Token for Tanzu Mission Control](#) in the *Tanzu Observability by Wavefront* documentation.

Also make sure you have the appropriate permissions in Tanzu Mission Control.

- To create a Tanzu Observability credential, you must be associated with the `organization.credential.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the the Accounts tab of the Administration page, click **Create Account Credential**, and then select **Tanzu Observability by Wavefront credential**.
- 3 On the Create credential page, provide a name for the credential.
- 4 Enter the Wavefront instance URL and API token from your Wavefront account.

Note The value for these fields come from your Wavefront account, and not your Tanzu Mission Control account. For information about how to retrieve these values from your Wavefront account, see [Generate a Token for Tanzu Mission Control](#) and [Your Wavefront Account](#) in the *Tanzu Observability by Wavefront* documentation.

- 5 Click **Create**.

You can optionally click **Create and Add Role Binding** which directs you to the Access tab of the Administration page where you can apply a role binding to the credential object to secure it. For more information about access control, see [Chapter 22 Managing Access to Your Resources](#).

Results

After you complete this procedure, you have a credential that you can use to add managed clusters to Tanzu Observability by Wavefront through Tanzu Mission Control. You can see your new credential listed on the Administration page in the Tanzu Mission Control console.

Edit a Tanzu Observability Credential Object

Edit an existing Tanzu Observability credential to update the API token.

If your API token for Tanzu Observability expires, the clusters that use the credential stop sending data. This procedure describes how to update your API token.

Prerequisites

Before you edit a Tanzu Observability credential, make sure you have access to the account and that you have generated a Tanzu Observability API token. For more information, see [Generate a Token for Tanzu Mission Control](#) in the *Tanzu Observability by Wavefront* documentation.

Also make sure you have the appropriate permissions in Tanzu Mission Control.

- To create a Tanzu Observability credential, you must be associated with the `organization.credential.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the the Accounts tab of the Administration page, locate the credential that you want to update, and then click **Edit**.
- 3 On the Edit credential page, enter the new Wavefront API token from your Wavefront account.

Note The value for this field comes from your Wavefront account. For information about how to retrieve an API token from your Wavefront account, see [Generate a Token for Tanzu Mission Control](#) in the *Tanzu Observability by Wavefront* documentation.

- 4 Click **Save**.

Results

After you complete this procedure, you can continue to use this credential to add managed clusters to Tanzu Observability by Wavefront through Tanzu Mission Control.

Remove an Account Credential

As a platform operator or infrastructure operator, you can remove an account credential when it is no longer needed.

When you remove an account credential, Tanzu Mission Control releases its connection to the account, but does not remove any of the resources that you might have created in that account. If you have resources in the account, they continue to exist until you use your cloud provider's tools to remove them.

Prerequisites

Prior to removing an account credential, remove any dependencies on it. For example, before you delete a data protection credential, make sure there are no target location objects that are using it. If you have scheduled backups that target a deleted data protection credential, those backups will fail.

Log in to the Tanzu Mission Control console, and make sure you have the appropriate permissions to remove a connection.

- To remove an account credential, you must be associated with the `organization.credential.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the Credentials tab of the Administration page, locate the account credential that you want to remove.
- 3 Click the menu icon for the credential, and then choose **Remove**.

4 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, Tanzu Mission Control removes the account credential.

Note After deleting a credential for an AWS EKS account, you might need to manually remove resources from the account. For more information, see [Clean Up Your AWS EKS Account After Deleting a Credential](#).

Connecting Through a Proxy

5

You can create a proxy configuration that enables Tanzu Mission Control to connect with clusters that are protected by a proxy server.

The Kubernetes clusters that you manage using Tanzu Mission Control run cluster agent extensions that make connections to Tanzu Mission Control to allow bilateral communication. This applies to Tanzu Kubernetes clusters that you register with Tanzu Mission Control and their workload clusters, as well as clusters that were created elsewhere and subsequently attached.

If you have a proxy server that manages outbound traffic for your clusters, the cluster agent extensions must identify the proxy server and get authorization to communicate with Tanzu Mission Control. By creating a proxy configuration object, you can consistently reuse the proxy settings for multiple clusters.

Proxy Requirements for Managing Clusters with Tanzu Mission Control

Make sure that your proxy meets the following requirements.

- The proxy must support HTTP 2.0 traffic because the Tanzu Mission Control cluster agent extensions communicate using gRPC protocol over HTTP 2.0.
- If your proxy uses SSL inspection, make sure you enable the streaming of large objects in your proxy's settings.

Read the following topics next:

- [Create a Proxy Configuration Object](#)
- [Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu](#)
- [Create a Proxy Configuration Object for AKS Clusters in Tanzu Mission Control](#)
- [Create a Proxy Configuration Object for EKS Clusters in Tanzu Mission Control](#)
- [Delete a Proxy Configuration Object](#)

Create a Proxy Configuration Object

Create a proxy configuration in VMware Tanzu Mission Control that allows outbound traffic through the proxy that protects your managed clusters.

A proxy configuration identifies the proxy server for one or more clusters and the credentials required to authorize outbound traffic through it. When you create a proxy configuration object, you can use it when registering an Azure AKS, AWS EKS, or a Tanzu Kubernetes Grid management cluster, provisioning a workload cluster, or attaching a cluster.

Note If you have already set up a proxy configuration in your Tanzu Kubernetes cluster, use those same settings (including the no proxy list) in the proxy configuration you create in Tanzu Mission Control for that cluster.

For information about using this feature with Tanzu Kubernetes Grid Service clusters, see [Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu](#).

For information about proxy configuration for Azure AKS clusters, see [Create a Proxy Configuration Object for AKS Clusters in Tanzu Mission Control](#).

For information about proxy configuration for AWS EKS clusters, see [Create a Proxy Configuration Object for EKS Clusters in Tanzu Mission Control](#).

Note Tanzu Mission Control managed AKS and EKS clusters can be configured with transparent mode proxy configuration. In such cases, the Tanzu Mission Control agent and its extensions and components are able to connect via traffic proxy in transparent mode, but for nodes or pods outside the `system-vmware-tmc` namespace, you must manually set up for using the transparent proxy.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to create a proxy configuration object.

- To create a proxy configuration, you must be associated with the `organization.credential.admin` role.

Make sure you have the proxy address and credentials, and the appropriate list of non-proxied addresses for the cluster. If necessary, you can run the following command on your Tanzu Kubernetes Grid management cluster to retrieve the proxy address and the no-proxy CIDRs.

```
kubectl get kubeadmconfig -n tkg-system
```

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**.
- 2 On the Administration page, click the **Proxy Configuration** tab.

- 3 Click **Create Proxy Configuration**.
- 4 On the Create proxy page, enter a name for the proxy configuration.
- 5 You can optionally provide a description.
- 6 Select the proxy type, either **Explicit** or **Transparent**.

Option	Actions
Explicit	<ol style="list-style-type: none"> a Specify the URL or IP address of the proxy server, and the port on which outbound traffic is allowed. b Enter the credentials (username and password) that permit outbound traffic through the proxy server. c You can optionally enter an alternative server/port and username/password for HTTPS traffic. d If your explicit proxy uses a root certificate or CA certificate (for example, if your proxy uses SSL inspection), enter the certificate into the provided box. <hr/> <p>Note Custom CA certificates are not supported for the following operations:</p> <ul style="list-style-type: none"> ■ Registering a Supervisor in TMC for vSphere with Tanzu versions prior to vSphere 7.0.3. ■ Lifecycle management of Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu versions prior to vSphere 8. <p>Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu.</p>
Transparent	<p>Provide the custom root or CA certificate in CRT format.</p> <p>Transparent proxy is not supported for lifecycle management of Tanzu Kubernetes clusters.</p>

- 7 In the **No proxy list**, you can optionally specify a comma-separated list of outbound destinations that must bypass the proxy server.
- 8 Click **Create**.

What to do next

After you create a proxy configuration object, you can use it when registering a Tanzu Kubernetes Grid management cluster, provisioning a workload cluster, or attaching a conformant Kubernetes cluster.

Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu

Create a proxy configuration in VMware Tanzu Mission Control that allows outbound traffic through the proxy that protects your managed clusters, specifically for Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu (vSphere 7.0.3a or later).

A proxy configuration identifies the proxy server for one or more clusters and the credentials required to authorize outbound traffic through it. When you create a proxy configuration object, you can use it when registering a Tanzu Kubernetes Grid Service Supervisor Cluster cluster running in vSphere with Tanzu or provisioning a workload cluster.

When you create a proxy configuration object to use with this type of cluster, you must identify certain subnets in the workload network of the Supervisor Cluster to exclude from proxying, in addition to identifying the proxy server.

- In a proxy configuration object for workload clusters, make sure the **No proxy list** includes the CIDRs for pod, ingress, and egress from the workload network of the Supervisor Cluster, as well as *.svc and *.cluster.local.
- In a proxy configuration object for the Supervisor Cluster, there are no specific requirements for the **No proxy list**.
- You can optionally create a proxy configuration object to use with both a Supervisor Cluster and its workload clusters that includes the requirements for the workload clusters in the **No proxy list**.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to create a proxy configuration object.

- To create a proxy configuration, you must be associated with the `organization.credential.admin` role.

Make sure your vSphere version is 7.0.3a or later.

You must configure a content library for the provisioner namespace on the Supervisor Cluster. For more information, see [Configure a vSphere Namespace for Tanzu Kubernetes Releases](#) in the *vSphere with Tanzu Configuration and Management* documentation.

You also need to locate the CIDRs for pod, ingress, and egress in the workload network of your Supervisor Cluster that you need for the **No proxy list**. For information about how to find these settings in your Supervisor Cluster, see [Configuration Parameters for the Tanzu Kubernetes Grid Service v1alpha2 API](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**.
- 2 On the Administration page, click the **Proxy Configuration** tab.
- 3 Click **Create Proxy Configuration**.
- 4 On the Create proxy page, enter a name for the proxy configuration.
- 5 You can optionally provide a description.

- 6 Specify the URL or IP address of the proxy server, and the port on which outbound traffic is allowed.
- 7 Enter the credentials (username and password) that permit outbound traffic through the proxy server.
- 8 You can optionally enter an alternative server/port and username/password for HTTPS traffic.
- 9 In **No proxy list**, you can optionally specify a comma-separated list of outbound destinations that must bypass the proxy server.

For a Tanzu Kubernetes Grid Service workload cluster, make sure you add the requirements listed above to the **No proxy list**.

- 10 Click **Create**.

Create a Proxy Configuration Object for AKS Clusters in Tanzu Mission Control

Tanzu Mission Control supports creating new AKS (Azure Kubernetes Service) clusters with a proxy configuration of type `explicit` as part of create API/manifests and being applied to the complete cluster. If proxy configuration of type `transparent` is used, then it is used for the Tanzu Mission Control agent and extensions only.

Tanzu Mission Control managed AKS clusters can be configured with `transparent` mode proxy configuration. In such cases, the Tanzu Mission Control agent and its extensions and components are able to connect via traffic proxy in transparent mode. For nodes or pods outside the `system-vmware-tmc` namespace, you need to manually setup for using the transparent proxy.

Procedure

- 1 To configure an `explicit` or `transparent` proxy for an AKS cluster, see [Create a Proxy Configuration Object](#).
- 2 To configure a `transparent` proxy in AKS for other namespaces, you must manually configure them to accept the transparent proxy's custom certificate if you need them to connect via proxy to the internet over HTTPS. For more information, see: <https://learn.microsoft.com/en-us/azure/aks/custom-certificate-authority#custom-ca-installation-on-aks-node-pools>.

Create a Proxy Configuration Object for EKS Clusters in Tanzu Mission Control

Tanzu Mission Control supports creating new clusters with a proxy configuration of type `explicit` as part of create API/manifests and being applied to the complete EKS (Elastic Kubernetes Service) cluster. If proxy configuration of type `transparent` is used, then it is used for the Tanzu Mission Control agent and extensions only.

Tanzu Mission Control managed EKS clusters can be configured with `transparent` mode proxy configuration. In such cases, the Tanzu Mission Control agent and its extensions and components are able to connect via traffic proxy in transparent mode. For nodes or pods outside the `system-vmware-tmc` namespace, you need to manually setup for using the transparent proxy.

Procedure

- 1 To configure an `explicit` or `transparent` proxy for EKS in Tanzu Mission Control, see [Create a Proxy Configuration Object](#).
- 2 To configure a `transparent` proxy in EKS for other namespaces, you must manually configure them to accept the transparent traffic proxy's custom certificate if you need them to connect via proxy to internet over HTTPS. For more information, see [AWS private certificate issuer plugin](#).

Delete a Proxy Configuration Object

Delete a proxy configuration in Tanzu Mission Control.

A proxy configuration identifies the proxy server for your clusters and the credentials required to authorize outbound traffic through it. When you are done using a given proxy configuration, you can remove it from Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions.

- To delete a proxy configuration, you must be associated with the `organization.credential.admin` role.

Make sure you have removed all dependencies on the proxy configuration object you want to delete. Because you use the proxy configuration to facilitate communication between your cluster and Tanzu Mission Control, all of the clusters that use a proxy configuration must be disconnected before you can delete the proxy configuration.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**.
- 2 On the Administration page, click the **Proxy Configuration** tab.
- 3 In the table of proxy configurations, locate the one you want to delete.
- 4 Click the menu icon for the proxy configuration you want to delete, and then choose **Delete**.

If there are still dependencies on this proxy configuration, they are displayed in a warning dialog. Dependencies can include attached clusters as well as registered Tanzu Kubernetes

clusters and their managed workload clusters. Before you can delete this proxy configuration, you must remove these dependencies by detaching or deregistering the clusters. Take note of the dependencies that you still need to remove, click **OK** to close the dialog, and then return to this procedure after removing the remaining dependencies.

- 5 In the confirmation dialog, click **Delete**.

Managing Clusters

6

As a platform operator or infrastructure operator, use VMware Tanzu Mission Control to bring your Kubernetes clusters together, organize them, and manage them consistently.

Through Tanzu Mission Control, you can bring all of your clusters together and organize them into cluster groups to facilitate consistent management. You can also provision resources and create new clusters directly from Tanzu Mission Control. For more information about cluster management see the following topics in *VMware Tanzu Mission Control Concepts*:

- [What is Tanzu Mission Control](#)
- [Cluster Lifecycle Management](#)
- [What Happens When You Attach a Cluster](#)

Read the following topics next:

- [Create a Cluster Group](#)
- [Delete a Cluster Group](#)
- [Attach a Cluster](#)
- [Attach a Cluster Running Behind a Proxy Server Using the CLI](#)
- [Reattach a Cluster](#)
- [Detach a Cluster](#)
- [Remove a Cluster from Your Organization](#)
- [Move a Cluster Between Cluster Groups](#)
- [Connect to a Managed Cluster with kubectl](#)
- [Enable Access to Public Cloud Clusters Through Tanzu Mission Control](#)

Create a Cluster Group

Create a cluster group in VMware Tanzu Mission Control to organize your Kubernetes clusters.

The cluster group is an organizational tool that helps you monitor and manage your cluster resources. For more information, see [What Is Tanzu Mission Control](#).

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to create a cluster group.

- To create a cluster group, you must be associated with the `organization.edit` role.

Procedure

- 1 In the left navigation pane, click **Cluster Groups**.
- 2 In the top right corner, click **New Cluster Group**.
- 3 Provide a name for the cluster group.
- 4 You can optionally add a description and labels.
- 5 Click **Create**.

Results

Now you have a cluster group that you can populate with attached and provisioned clusters, and manage with policies.

Delete a Cluster Group

Remove a cluster group from your VMware Tanzu Mission Control organization.

Prerequisites

Before starting this procedure, perform the following tasks:

- Log in to the Tanzu Mission Control console.
- Make sure the cluster group contains no clusters (either attached or provisioned).
- Make sure you have the appropriate permissions. To delete a cluster group, you must be associated with the `organization.edit` role in your organization.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.
- 2 In the cluster group list, click the cluster group you want to delete.
- 3 Make sure the cluster group is empty.

If there are any clusters (either attached or provisioned), you can not delete the cluster group.

- 4 In the top right corner, click **Actions**, and then choose **Delete cluster group** from the dropdown.
- 5 In the confirmation dialog, click **Yes**.

Results

The cluster group is permanently removed from your Tanzu Mission Control organization.

Attach a Cluster

Attach an existing cluster to your organization using the Tanzu Mission Control console.

To attach a cluster you run a set of extensions on the cluster and associate it with a cluster group in Tanzu Mission Control.

Prerequisites

You use both the Tanzu Mission Control console and the Kubernetes CLI to attach a cluster.

- Open a browser and log in to the Tanzu Mission Control console.
- Open a command window and connect to your cluster with `kubectl`.

Make sure you have the appropriate permissions.

- To attach a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to attach the cluster.
- On the cluster, you must have `admin` permissions to install and run the cluster agent extensions.

If you plan to use a local image registry, you need to create credentials and download the image registries. For more information, see [Managing a Local Image Registry](#).

If you have a proxy server that manages outbound traffic for your clusters, you need to enable the cluster to communicate with Tanzu Mission Control through the proxy.

- You can create a proxy configuration object in Tanzu Mission Control and use it when registering, provisioning, or attaching the cluster. For more information, see [Chapter 5 Connecting Through a Proxy](#).
- You can enable all outbound traffic to Tanzu Mission Control for the proxy server by adding some URLs to the proxy server's allowlist, as described in [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Attach cluster** from the dropdown.
- 3 Select the cluster group in which you want to register the cluster, and enter a name for the cluster.

This name appears in the list of clusters on the Cluster page.

- 4 You can optionally provide a description and one or more labels.
- 5 Click **Next**.

- 6 You can toggle the **Local Image Registry Setting** to **Yes** to enable it for the cluster and then specify the local image registry in the **Set a local image registry for this cluster** field, or click **Add New Local Image Registry** to add a new registry.
- 7 You can optionally select a proxy configuration for the cluster.
 - a Click to toggle the **Set proxy** option to **Yes**.
 - b Select the proxy configuration you defined for this cluster.
- 8 Click **Next**.

When you click **Next**, Tanzu Mission Control generates a YAML manifest specifically for your cluster, and displays the command to run the script.

- 9 Copy the provided command, open a command window, and then run the command.

Make sure your current context is set appropriately for the workload cluster you want to attach.

 - If you attach using a proxy configuration, make sure you have the latest version of the Tanzu Mission Control CLI (`tmc`) installed, and then run the `tmc` command, replacing `<kubeconfig>` with the appropriate kubeconfig for the cluster.
 - If you attach without a proxy configuration, connect to the cluster with `kubectl`, and then run the `kubectl` command.

The YAML manifest runs a set of extensions in your cluster to connect it with the Tanzu Mission Control cluster agent service. Be aware of the following:

- The generated YAML manifest is specifically tailored to your cluster and cannot be used for other clusters.
 - The manifest contains credentials that permit the cluster to attach to the cluster agent service. These credentials are valid for 48 hours. If you attempt to install the agent extensions after the credentials lapse, the attachment fails and your cluster remains indefinitely in **Pending** status in the Tanzu Mission Control console.
- 10 After the extensions are up and running in your cluster, return to the Tanzu Mission Control console and click **Verify Connection**.

After you click **Verify Connection**, you can see the cluster on the Clusters page. The cluster shows a **Pending** status until the verification is complete, and then changes to **Ready**.

Results

After the connection is verified, the cluster is attached and you can manage it using Tanzu Mission Control.

Attach a Cluster Running Behind a Proxy Server Using the CLI

Use the Tanzu Mission Control CLI to attach a cluster running on a network that is protected by a proxy server.

To attach a cluster that is running behind a proxy server, you can use the Tanzu Mission Control CLI. The `tmc cluster attach` command in this procedure generates a YAML manifest and runs it on your cluster to attach it to your Tanzu Mission Control organization.

This procedure is only necessary if your cluster is running behind a proxy server. If your cluster is not behind a proxy server, you can attach it as described in [Attach a Cluster](#).

Note The Tanzu Mission Control CLI has been updated to facilitate the use of a proxy configuration object. As a result, the following flags for this command are deprecated:

- `--http-proxy-url`
 - `--http-proxy-username`
 - `--http-proxy-password`
-

Prerequisites

Make sure you have the appropriate permissions.

- To attach a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to attach the cluster.
- To use a proxy configuration, you must be associated with the `organization.credential.view` role in your organization.
- On the cluster, you must have `admin` permissions to install and run the cluster agent extensions.

Create a proxy configuration object, as described in [Chapter 5 Connecting Through a Proxy](#).

Download and install the Tanzu Mission Control CLI, as described in [Chapter 2 Log In with the Tanzu CLI](#).

Procedure

- ◆ Run the `tmc cluster attach` command with the following flags to register the cluster and generate the YAML.
 - `--name <cluster name as you want it to appear in the console>`
 - `--cluster-group <name of the cluster group in which you want to attach the cluster>`
 - `-k <kubeconfig for the cluster>`

- `--proxy <name of the proxy configuration you created for the cluster>`

```
tmc cluster attach --name myclustername --cluster-group myclustergroup -k mykubeconfig --
proxy myproxyconfig
```

When you run the command, Tanzu Mission Control registers the cluster with the cluster agent using the cluster name and cluster group that you provided, and then runs the YAML manifest to attach your cluster.

Reattach a Cluster

Attach a cluster to your VMware Tanzu Mission Control organization that was previously attached and subsequently fell into a disconnected state.

Kubernetes clusters that are managed by Tanzu Mission Control can fall into a `disconnected` state, which means there has been no communication from the cluster to the cluster agent for an extended period of time. Sometimes this situation can be caused by environmental factors like network connectivity that are beyond the control of Tanzu Mission Control. However, sometimes a disconnected state can indicate that something was altered on the cluster that is preventing the Tanzu Mission Control cluster agent extensions from communicating with the cluster agent service.

The procedure described in this topic shows how to verify whether the connection is recoverable, and if so, how to reattach the cluster to your Tanzu Mission Control organization.

This procedure is also applicable for clusters on which you might have started, but never completed, the attach process and the client credentials have expired.

Prerequisites

You use both the Tanzu Mission Control console and the Tanzu Mission Control CLI to reattach a cluster.

- Open a browser and log in to the Tanzu Mission Control console.
- Open a command window and log in with the Tanzu Mission Control CLI.

Make sure you have the appropriate permissions.

- To attach or reattach a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to attach the cluster.
- On the cluster, you must have `admin` permissions to install and run the cluster agent extensions.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster you want to reattach.

When an attached cluster is in a disconnected state, the cluster detail page displays a command that you can use to evaluate the condition of the cluster.

- 2 On the cluster detail page, copy the displayed command.

The command looks something like this:

```
tmc cluster validate -k <path_to_KUBECONFIG>
```

- 3 In the command window, run the command, replacing `<path_to_KUBECONFIG>` with the path and name of your kubeconfig file for this cluster.

The resulting output shows the tests that were run to determine the cause of the disconnect, and the `tmc` command that you can use to reattach the cluster. The command looks something like this:

```
tmc cluster reattach -n <CLUSTER_NAME> -k /path/to/kubeconfig
```

- 4 In the command window, run the displayed command, replacing `<CLUSTER_NAME>` with the name of the cluster as displayed on cluster detail page in the Tanzu Mission Control console.

Results

The `reattach` command performs the necessary updates to the cluster agent extensions on the cluster to re-establish the connection between the cluster and the Tanzu Mission Control cluster agent.

Detach a Cluster

Detach a cluster from your VMware Tanzu Mission Control organization.

When you attach a cluster, Tanzu Mission Control installs cluster agent extensions on the cluster and registers the cluster with the Tanzu Mission Control cluster agent service. For more information, see [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

Similarly, when you detach a cluster, Tanzu Mission Control tells the cluster agent extensions to uninstall themselves, and then removes the cluster from the cluster agent registry for your organization.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To detach a cluster, you must be associated with the `clustergroup.edit` role for the cluster group.
- You must be associated with the `cluster.admin` role for the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.

- 2 On the Cluster groups page, click the cluster group that contains the cluster you want to detach, and then click the cluster.
- 3 On the cluster detail page, click **Actions** and then choose **Detach** from the dropdown.
- 4 In the confirmation dialog, enter the name of the cluster and then click **Detach**.

Results

When you click **Detach**, Tanzu Mission Control tells the cluster agent extensions that are installed in the cluster to uninstall themselves, and then de-registers the cluster from your organization. This process takes a few minutes. While Tanzu Mission Control is waiting for notification that the cluster agent extensions have completed their uninstall, the cluster appears with the `Deleting` status in the Tanzu Mission Control console.

If for some reason Tanzu Mission Control cannot communicate with the cluster for an extended period of time, the cluster can remain in the `Deleting` status. If you have a cluster in this condition, you can optionally force the detach of the cluster, as described in [Remove a Cluster from Your Organization](#).

What to do next

See [Manage Detached Cluster Resources](#) for information about handling resources after detaching a cluster.

Manage Detached Cluster Resources

Managed resources and catalog-related Kubernetes resources are not removed when you detach a cluster.

Managed Resources

Managed resources are retained after a normal or forced cluster detach. These resources include managed namespaces, policies, and any workloads running within those namespaces. You can use the following CLI commands to list and choose which managed resources to maintain after a cluster detach.

Use the following command to list all the managed resources that were left behind by the detach operation.

```
tmc cluster managed-resources list
```

Use this command to retain specified managed resources:

```
tmc cluster managed-resources retain < resource-type > [ resource-name ]
```

Where:

resource-type is a mandatory argument which specifies the type of resource being retained such as pods, deployment.apps, StatefulSet, etc.

resource-name is an optional argument which specifies the name of the resource being retained. If *resource-name* is absent, all the k8s resources belonging to the specified *resource-type* will be retained.

Once all of the managed resources are retained, remaining resources can be cleaned up using the `kubectl` command.

CRDs and CRs should not be deleted before retaining the necessary k8s resources.

Catalog-related Kubernetes Resources

The detach process does not remove the catalog-related Kubernetes resources installed on your cluster, as that would impact any workloads you have deployed using the catalog. If you want to remove these resources, run the following `kubectl` commands:

```
kubectl delete crd packageinstalls.packaging.carvel.dev
kubectl delete crd packagerepositories.packaging.carvel.dev
kubectl delete crd internalpackagemetadatas.internal.packaging.carvel.dev
kubectl delete crd internalpackages.internal.packaging.carvel.dev
kubectl delete crd apps.kappctrl.k14s.io
kubectl delete APIService v1alpha1.data.packaging.carvel.dev
kubectl delete ns tanzu-system
kubectl delete ns tanzu-package-repo-global
kubectl delete tanzupackage-install-admin-role kapp-controller-cluster-role
kubectl delete clusterrolebinding kapp-controller-cluster-role-binding pkg-apiserver:system:auth-delegator
kubectl delete psp tanzu-system-kapp-ctrl-restricted
kubectl delete rolebinding pkgserver-auth-reader -n kube-system
kubectl delete crd secretexports.secretgen.carvel.dev
kubectl delete crd secretimports.secretgen.carvel.dev
kubectl delete psp tanzu-system-secretgen-ctrl-restricted
```

Additionally, if you have enabled continuous delivery on your cluster prior to detaching it, you can remove the Flux-related resources using the following `kubectl` commands:

```
kubectl delete packageinstalls.packaging.carvel.dev -n tanzu-fluxcd-packageinstalls kustomize-controller
kubectl delete packageinstalls.packaging.carvel.dev -n tanzu-fluxcd-packageinstalls source-controller
kubectl delete ns tanzu-fluxcd-packageinstalls
kubectl delete ns tanzu-continuousdelivery-resources
```

Remove a Cluster from Your Organization

Use the **Manually delete agent extensions** option to forcibly remove a cluster from your VMware Tanzu Mission Control organization.

When you provision or attach a cluster, Tanzu Mission Control installs cluster agent extensions on the cluster and registers the cluster with the Tanzu Mission Control cluster agent service. For more information, see [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

When you remove an attached cluster from your Tanzu Mission Control organization using the **Manually delete agent extensions** option, Tanzu Mission Control makes no attempt to remove the cluster agent extensions installed on the cluster. If you choose this option, you must manually remove the cluster agent extensions as described in this procedure.

Note If you use **Manually delete agent extensions** to release control of a provisioned cluster, Tanzu Mission Control attempts to delete the cluster but it is possible that the cluster and its resources might remain as they were prior to removing the cluster from your organization. These resources, that were provisioned through Tanzu Mission Control, might continue to run in the cloud provider account in which you created them. You must manually delete the cluster and its resources using your cloud provider's tools to avoid incurring costs associated with them.

In contrast, when you detach a cluster without the **Manually delete agent extensions** option, Tanzu Mission Control tells the cluster agent extensions to uninstall themselves, and then removes the cluster from the cluster agent registry for your organization. Similarly when you use the standard delete of a provisioned cluster, Tanzu Mission Control deletes the cluster from your cloud provider before removing it from the cluster agent registry for your organization.

Unless you have reason to forcibly remove this cluster from your organization, use the standard detach or delete without the **Manually delete agent extensions** option.

Some reasons to use the **Manually delete agent extensions** option include the following:

- You have attempted to delete the cluster and it appears to be stuck in the Deleting state for an extended period of time.
- You want to detach the cluster, but for some reason Tanzu Mission Control cannot communicate with it.

In either case, make sure you remove the cluster agent extensions manually as described in this procedure, or delete the cluster if it is no longer needed.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To detach or delete a cluster, you must be associated with the `clustergroup.edit` role for the cluster group.
- You must also be associated with the `cluster.admin` role for the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.
- 2 On the Cluster groups page, click the cluster group that contains the cluster you want to remove from your organization, and then click the cluster.

- 3 On the cluster detail page, click **Actions** and then choose the appropriate action from the dropdown.
 - For an attached cluster, choose **Detach**.
 - For a provisioned cluster, choose **Delete**.
- 4 In the confirmation dialog, select **Manually delete agent extensions** and then enter the name of the cluster.
- 5 Click **Detach** or **Delete**.
Tanzu Mission Control de-registers the cluster from your organization.
- 6 If you no longer need the cluster, delete it using your cloud provider's tools.
- 7 If you want to keep the cluster, remove the cluster agent extensions.
 - a Open a command window and connect to your cluster with `kubectl`.
 - b Use the following `kubectl` commands to remove the namespace, CRDs, clusterroles/clusterrolebindings, and other resources installed by Tanzu Mission Control.

```
kubectl delete namespace vmware-system-tmc
kubectl delete crd extensions.clusters.tmc.cloud.vmware.com
kubectl delete crd agents.clusters.tmc.cloud.vmware.com
kubectl delete crd extensionresourceowners.clusters.tmc.cloud.vmware.com
kubectl delete crd extensionintegrations.clusters.tmc.cloud.vmware.com
kubectl delete crd extensionconfigs.intents.tmc.cloud.vmware.com
kubectl delete clusterrole extension-updater-clusterrole extension-manager-role agent-updater-role vmware-system-tmc-psp-agent-restricted
kubectl delete clusterrolebinding extension-updater-clusterrolebinding extension-manager-rolebinding agent-updater-rolebinding vmware-system-tmc-psp-agent-restricted
kubectl delete psp vmware-system-tmc-agent-restricted
```

- c Use the following `kubectl` commands to remove the catalog-related Kubernetes resources installed by Tanzu Mission Control.

Note Removing these resources impacts any workloads that you have deployed using the catalog.

```
kubectl delete crd packageinstalls.packaging.carvel.dev
kubectl delete crd packagerepositories.packaging.carvel.dev
kubectl delete crd internalpackagemetadatas.internal.packaging.carvel.dev
kubectl delete crd internalpackages.internal.packaging.carvel.dev
kubectl delete crd apps.kappctrl.k14s.io
kubectl delete APIService v1alpha1.data.packaging.carvel.dev
kubectl delete ns tanzu-system
kubectl delete ns tanzu-package-repo-global
kubectl delete tanzupackage-install-admin-role kapp-controller-cluster-role
kubectl delete clusterrolebinding kapp-controller-cluster-role-binding pkg-apiserver:system:auth-delegator
kubectl delete psp tanzu-system-kapp-ctrl-restricted
kubectl delete rolebinding pkgserver-auth-reader -n kube-system
```


These `kubectl` commands remove the Tanzu Mission Control cluster agent extensions from your cluster.

Move a Cluster Between Cluster Groups

As a platform operator, move a managed Kubernetes cluster from one cluster group to another in VMware Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To move a cluster between cluster groups, you must be associated with the `clustergroup.admin` role for both the source and the target cluster groups.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster you want to move.
- 2 On the cluster detail page, click **Actions**, and then choose **Move**.
- 3 In the Move cluster dialog, select the target cluster group to which you want to move the cluster, and then click **Move**.

Results

When you click **Move**, Tanzu Mission Control removes the cluster's association from the source cluster group and then associates the cluster with the target cluster group. During this process, policies on the cluster change. The inherited policies from the source cluster group are removed from the cluster, and the policies that are inherited from the target cluster group are applied.

What to do next

Connect to a Managed Cluster with `kubectl`

To use `kubectl` and the `tanzu` CLI with a managed cluster, download the generated configuration file and initialize the connection.

After you have attached or provisioned a new cluster using Tanzu Mission Control, you can connect to it with `kubectl` using the configuration file that Tanzu Mission Control generates for you.

In addition to provisioned and attached Tanzu Kubernetes Grid clusters, Tanzu Mission Control supports `kubeconfig` access to the following types of attached public cloud clusters:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)

- Google Kubernetes Engine (GKE)

For more information, see [Enable Access to Public Cloud Clusters Through Tanzu Mission Control](#).

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that you have already performed the following tasks:

- Install the Kubernetes command-line interface (`kubectl`).
- Open a browser window, and log in to the Tanzu Mission Control console.
- Open a command window, and log in with the `tanzu` CLI.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click the cluster to which you want to connect.
- 3 On the cluster detail page, in the upper right corner, click **Access this cluster**.
- 4 In the resulting popup modal, click **Download KUBECONFIG file**.
- 5 Save the downloaded YAML file in a location that is accessible to `kubectl` (for example, in `~/.kube/config` or in a location specified in the `KUBECONFIG` environment variable).
- 6 Run a `kubectl` command to initialize the configuration. For example, the following command retrieves a list of namespaces in your cluster.

```
kubectl get namespaces
```

You can optionally use the `--kubeconfig` flag to specify the location and name of your downloaded kubeconfig YAML file. The command looks something like this:

```
kubectl --kubeconfig=/path/to/kubeconfig-my-provisioned-cluster.yml get namespaces
```

Note The path/file value that you pass to the `--kubeconfig` flag must be an absolute path.

If you are not logged in with the `tanzu` CLI, you are prompted to provide an API token to log in. For information about retrieving an API token, see [Chapter 2 Log In with the Tanzu CLI](#). An API token is not required for Tanzu Mission Control Self-Managed.

Enable Access to Public Cloud Clusters Through Tanzu Mission Control

Modify the Pinniped load balancer on an attached Kubernetes cluster running in a public cloud provider to enable kubeconfig access from Tanzu Mission Control.

As a platform operator, you can configure an attached cluster to allow users in your organization to connect to the cluster with `kubectl`.

When you attach a cluster, Tanzu Mission Control deploys Pinniped (an open source community standard) to support authentication into the cluster. By default, the Pinniped ingress load balancer is configured as private for the following types of clusters:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)

To access a cluster using the kubeconfig that you can generate through Tanzu Mission Control, you must have direct access to the private network on which it is running, because the load balancer is private by default.

This procedure describes how to update the configuration of the Pinniped load balancer to allow operators in your organization access to the cluster using the kubeconfig generated through Tanzu Mission Control.

Note Use this approach judiciously. This procedure results in having a public cloud cluster with a public ingress load balancer. To minimize exposure, use this procedure only when other more secure approaches are not available.

This procedure is only necessary if all of the following are true:

- The cluster is running in one of the public clouds listed above.
- You have users in your organization that do not have direct access to the network on which the cluster is running.
- Those users need to access the cluster using the kubeconfig generated by Tanzu Mission Control.

If the users in your organization that need `kubectl` access to the cluster have direct access to the network on which the cluster is running, then this procedure is not necessary.

For more information about generating a kubeconfig for an attached cluster, see [Connect to a Managed Cluster with kubectl](#).

For more information about Pinniped, go to <https://pinniped.dev>.

Prerequisites

This procedure assumes that you have already attached your Kubernetes cluster to Tanzu Mission Control, and that you have direct access to the cluster.

Make sure you have the appropriate permissions to enable access.

- On the cluster, you must have `admin` permissions on the control plane to update the Pinniped configuration.

Procedure

- 1 Open a command window and set your `kubectl` context for the cluster.

2 Edit the Pinniped configuration.

- a Run the following `kubectl` command.

```
kubectl edit credentialissuer cluster-auth-pinniped-config
```

- b Locate the `annotations` section of `spec.impersonationProxy.service`.

For example:

```
spec:
  impersonationProxy:
    mode: auto
    service:
      annotations:
        networking.gke.io/load-balancer-type: Internal
        service.beta.kubernetes.io/aws-load-balancer-connection-idle-timeout: "4000"
        service.beta.kubernetes.io/aws-load-balancer-internal: "true"
        service.beta.kubernetes.io/azure-load-balancer-internal: "true"
```

- c Remove the annotation that corresponds to public cloud provider that the cluster is running on.

- GKE: `networking.gke.io/load-balancer-type: Internal`
- AKS: `service.beta.kubernetes.io/azure-load-balancer-internal: "true"`
- EKS: `service.beta.kubernetes.io/aws-load-balancer-internal: "true"`

- d Save your changes.

3 Delete the Pinniped service to force the cluster to restart it.

```
kubectl delete services -n vmware-system-tmc cluster-auth-pinniped-impersonation-proxy-load-balancer
```

4 Verify the new configuration

- a Run the following command to confirm that the load balancer is now public.

```
kubectl get services -n vmware-system-tmc
```

The resulting output from this command should contain a line showing the Pinniped load balancer, that looks something like this:

NAME	EXTERNAL-IP	PORT(S)	AGE	TYPE	CLUSTER-
cluster-auth-pinniped-impersonation-proxy-load-balancer	10.0.###.113	52.143.###.159	443:30520/TCP	2h	LoadBalancer

- b Verify that the `EXTERNAL-IP` column contains a value.
- c When you subsequently download the kubeconfig from Tanzu Mission Control, make sure the endpoint in the URL matches the one from the output of this command.

Managing Namespaces

7

As an application developer or operator, use VMware Tanzu Mission Control to organize your namespaces and create new ones.

Through Tanzu Mission Control, you can organize your namespaces into workspaces to facilitate consistent management. You can also create new namespaces directly from Tanzu Mission Control. For more information about namespace management, see [What is Tanzu Mission Control](#) in *VMware Tanzu Mission Control Concepts*.

Read the following topics next:

- [Create a Workspace](#)
- [Delete a Workspace](#)
- [Create a Managed Namespace](#)
- [Delete a Managed Namespace](#)
- [Attach a Namespace](#)

Create a Workspace

Create a workspace in VMware Tanzu Mission Control to organize your Kubernetes namespaces.

The workspace is an organizational tool that helps you monitor and manage your Kubernetes namespaces within and across clusters. For more information, see [What Is Tanzu Mission Control](#).

Prerequisites

Log in to the Tanzu Mission Control console.

You must also have the appropriate permissions. To create a workspace, you must be associated with the `organization.edit` role.

Procedure

- 1 In the left navigation pane, click **Workspaces**.
- 2 In the top right corner, click **New Workspace**.
- 3 Provide a name for the workspace.
- 4 You can optionally add a description and labels.

- 5 Click **Create**.

Results

Now you have a workspace that you can populate with managed namespaces from attached and provisioned clusters, and manage with policies.

Delete a Workspace

Remove a workspace from your VMware Tanzu Mission Control organization.

Prerequisites

Before starting this procedure, perform the following tasks:

- Log in to the Tanzu Mission Control console.
- Make sure the workspace contains no namespaces.
- Make sure you have the appropriate permissions. To delete a workspace, you must be associated with the `organization.edit` role in your organization.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Workspaces**.
- 2 In the workspace list, click the workspace you want to delete.
- 3 Make sure the workspace is empty.
If there are any namespaces, you can not delete the workspace.
- 4 In the top right corner, click **Actions**, and then choose **Delete workspace** from the dropdown.
- 5 In the confirmation dialog, click **Yes**.

Results

The workspace is permanently removed from your Tanzu Mission Control organization.

Create a Managed Namespace

Use VMware Tanzu Mission Control to create a namespace on an attached or provisioned cluster in your organization.

Prerequisites

- Log in to the Tanzu Mission Control console.
- Make sure you have the appropriate permissions. To create a namespace, you must be associated with the following roles:
 - In the workspace, you must be associated with the `workspace.edit` and `namespace.create` roles.

- In the cluster, you must be associated with the `cluster.edit` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Workspaces**.
- 2 On the Workspaces page, click the workspace in which you want to manage the new namespace.
- 3 On the workspace detail page, in the upper right corner, click **New Namespace**.
- 4 Select the cluster in which you want to create the namespace, and make sure the appropriate workspace is selected.
- 5 Provide a name for the cluster.
- 6 You can optionally provide a description, and create labels to apply to the namespace.
- 7 Click **Create**.

Results

When you click **Create**, Tanzu Mission Control adds the namespace to the specified cluster and registers it as a managed namespace in the specified workspace.

Delete a Managed Namespace

Remove a managed namespace from your VMware Tanzu Mission Control organization.

Your clusters can have both managed and unmanaged namespaces. You can manipulate only managed namespaces using Tanzu Mission Control. Unmanaged namespaces that were created outside of Tanzu Mission Control must be managed directly in the cluster (for example, using `kubectl`).

Prerequisites

Before starting this procedure, perform the following tasks:

- Log in to the Tanzu Mission Control console.
- Make sure you have the appropriate permissions. To delete a namespace, you must be associated with the `workspace.edit` role in the workspace that contains the namespace.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Workspaces**.
- 2 In the workspace list, click the workspace that contains the namespace you want to delete.
- 3 In the namespace list, click the namespace you want to delete.
The namespace detail page shows the workloads that are running in the namespace.
- 4 In the top right corner, click **Actions**, and then choose **Delete namespace** from the dropdown.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the namespace and all the workloads that are running there are permanently removed from your Tanzu Mission Control organization.

Attach a Namespace

Attach a namespace to your organization, so you can manage it with VMware Tanzu Mission Control.

If you have unmanaged namespaces in your organization's clusters (either provisioned or attached), you can use this procedure to bring them under the management of Tanzu Mission Control to monitor, secure, and apply policies to them.

Prerequisites

- Log in to the Tanzu Mission Control console.
- Make sure you have the appropriate permissions. To attach a namespace, you must be associated with the following roles:
 - In the workspace, you must be associated with the `workspace.edit` role.
 - In the cluster, you must be associated with the `cluster.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click the cluster that contains the namespaces you want to attach.
- 3 On the cluster detail page, click the **Namespaces** tab.
- 4 Select the namespaces you want to attach by clicking the appropriate checkboxes.
- 5 Directly above the list of namespaces, click **Attach Namespaces** when you have finished selecting all of the namespaces you want to attach.
- 6 In the confirmation dialog, select the workspace in which you want to manage the namespaces, and then click **Attach**.

Managing the Lifecycle of Azure AKS Clusters



You can create and manage Azure AKS credentials and perform cluster lifecycle management operations including create, update, upgrade, and delete directly from Tanzu Mission Control. Creating and managing Azure AKS clusters with Tanzu Mission Control includes multiple steps, which are summarized here.

Note Tanzu Mission Control is the source of truth for clusters created and managed by Tanzu Mission Control. Anything changed outside of Tanzu Mission Control is reported as drift and you will see an `Out of Sync` message.

Note For information about using the `tanzu` CLI to manage AKS clusters, see the topic for [aks-cluster](#) in *VMware Tanzu CLI Reference - Tanzu Mission Control Plug-ins*.

- 1 Create an Azure AKS credential.
- 2 Create an Azure AKS cluster.

Read the following topics next:

- [Create an Azure AKS Credential](#)
- [Edit an Azure AKS Credential](#)
- [Delete an Azure AKS Credential](#)
- [Create an Azure AKS Cluster](#)
- [Edit an Azure AKS Cluster](#)
- [Delete an Azure AKS Cluster](#)
- [Add an Existing Azure AKS Cluster into Tanzu Mission Control Management](#)
- [Upgrade an Azure AKS Cluster](#)
- [Modify AKS Cluster Node Pools](#)
- [AKS Credential Resources](#)

Create an Azure AKS Credential

Set up a credential that allows VMware Tanzu Mission Control to connect to your Azure subscription and manage resources in your Azure account.

An account credential is required for managing the lifecycle of Azure AKS clusters. A single credential can contain multiple subscriptions. You can create clusters in any of the subscriptions from that credential. You specify which subscription to use when creating AKS clusters.

Subscriptions are added at credential creation and can not be edited.

Note There is a five (5) minute sync period between the Azure account and Tanzu Mission Control, so resources created (such as a subnet) may not appear as available in Tanzu Mission Control immediately. The syncing must be complete before you try to create another resource.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to create Azure AKS credentials.

- To create a credential for Azure AKS, you must be associated with the `cluster.admin` role.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 Click the **Accounts** tab.
- 3 Click **Create Credential** and select **Azure AKS** from the dropdown list.
- 4 Enter a name for the credential, and optionally a description and one or more labels.
- 5 Click **Next**.
- 6 Select the type of service principal to use.

Tanzu Mission Control uses a service principal to connect to your Azure subscriptions.

- Select **Existing Service Principal** and enter the IDs and certificate.
- Select **New Service Principal** and create a service principal with contributor role on each Azure subscription it has access to. Select either **Azure CLI** or **Azure Portal UI** and follow the instructions for the selected method.

- 7 Enter one or more subscription IDs to associate them with the credential.
- 8 Click **Next**.
- 9 You can optionally change the region in which to place the Tanzu Mission Control management plane resources.
- 10 Click **Create**.

Results

Tanzu Mission Control creates the credential and makes it available for use. This process typically takes a few minutes. After about 15 minutes, the credential should be ready for cluster management.

What to do next

After the credential is created and available, you can use it to create Azure AKS clusters.

Edit an Azure AKS Credential

You can update an Azure AKS credential for use with Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to manage clusters.

- You must have the `cluster.admin` role on the cluster.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**.
- 2 On the Administration page, click the **Accounts** tab.
- 3 Click the menu icon for the credential you want to update, and then choose **Edit**.
- 4 Modify the credential as necessary.
 - a Edit the description.
 - b Add, update, or remove labels.
 - c Add subscriptions.
 - d Edit the private key and certificate.
- 5 Click **Save**.

Delete an Azure AKS Credential

Use VMware Tanzu Mission Control to delete an Azure credential that you no longer need.

A credential cannot be deleted if there are AKS clusters associated with it in Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to delete Azure AKS credentials.

- To delete an Azure AKS credential, you must be associated with the `cluster.admin` role.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 Click the **Credentials** tab.
- 3 On the the Credentials page, select the credential that you want to delete.
- 4 Click on the options menu (ellipses), select **Remove**, and then click **Yes, Remove** on the warning screen.

Results

The credential is removed.

Create an Azure AKS Cluster

Create an Azure AKS cluster in your connected Azure account using Tanzu Mission Control.

Prerequisites

- Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).
- To create Azure AKS clusters, you must be associated with the Tanzu Mission Control `cluster.admin` role.

Make sure you have already created an account credential that provides access to your Azure account, as described in [Create an Azure AKS Credential](#).

The Azure service principal that is attached to the Tanzu Mission Control credential must be added to the AAD (Azure Active Directory) admin group(s) before starting the cluster creation process. Otherwise, the Tanzu Mission Control agent deployment process fails with the following error:

```
error: You must be logged in to the server (Unauthorized)
```

When creating AKS clusters, Tanzu Mission Control uses the following naming convention:

```
aks.generated-id.resource-group.cluster-name
```

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 Click **Clusters** in the left navigation pane.
- 3 Click **Create cluster**, and then choose **Create AKS cluster** from the dropdown.
- 4 Enter a name for the cluster, and specify the cluster group.

The cluster name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a maximum length of 63 characters.

- 5 You can optionally add a description and label.
- 6 Click **Next**.
- 7 Configure the control plane.
 - a Select the account credential and subscription in which you want to create the cluster.
 - b Select the resource group.
 - c Optionally enter an AKS tag for the node pool.
 - d In the **Cluster Details** section, select the version of Kubernetes to use for the cluster, and its region.

Note Tanzu Mission Control provides a list of regions in which you can create the control plane. In Azure, the availability of geographical regions is determined by your subscription. It is possible to select a region from the list for which you do not have a subscription, in which case an error message appears after Tanzu Mission Control attempts to create the node plane.

- e Select the type of cluster identity you want to use.

You can accept the default system-assigned managed identity or specify a user-assigned managed identity. If you select user-assigned, you must provide a valid managed identity defined in your Azure subscription. The format of the identity looks something like this:

```
"/subscriptions/my-subscription-id/resourcegroups/my-resource-group/providers/
Microsoft.ManagedIdentity/userAssignedIdentities/my-managed-identity"
```

For more information about managed identities in AKS, see the following topics in the *Azure documentation*:

- [What are managed identities for Azure resources?](#)
- [Use a managed identity in Azure Kubernetes Service \(AKS\)](#)

- f In the **Network** section, select the network configuration, either Kubenet or Azure CNI, and specify the DNS name prefix. Kubenet and CNI differ in how they assign IP addresses to nodes and pods. For more information, see [Microsoft Azure Container Networking in the Azure documentation](#).
 - g Click **Next**.
- 8 Configure node pools.
 - a Specify the node name.
 - b Select the mode, either System or User.
 - c Select the availability zones.
 - d Optionally enter a description.
 - e Optionally enter an AKS tag for the node pool, AKS labels, and taints.
 - f Configure the compute parameters by selecting VM size.
 - g Configure the scaling parameters.
 - h Click **Disk encryption setting** to provide your own key for managing disk encryption.
- 9 Click **Create**.

Results

The cluster is created and under Tanzu Mission Control lifecycle management.

Edit an Azure AKS Cluster

You can edit an Azure AKS cluster under Tanzu Mission Control management to make changes to its configuration.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to manage clusters.

- To edit an AKS cluster under Tanzu Mission Control management, you must be associated with the `managementcluster.admin` role.
- You must also have the `cluster.admin` role on the cluster.

For more information about roles, see [Access Control](#).

Procedure

- 1 In the Tanzu Mission Control console, click **Clusters** in the left navigation pane.

- 2 Click on the cluster you want to edit.
- 3 On the cluster details page, click **Actions** and then choose **Edit** from the dropdown.
- 4 You can edit the cluster group information, including changing the group to which the cluster belongs and adding an optional label and tag.
- 5 You can modify the cluster identity selection.
- 6 The Network section provides information about current settings and lets you change certain settings, including traffic routing and security settings.
- 7 When you finish making modifications to the cluster, click **Save**.

Delete an Azure AKS Cluster

You can delete AKS clusters that you no longer want to manage using VMware Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to delete AKS clusters.

- To delete AKS clusters, you must be associated with the `cluster.admin` role.

For more information about Tanzu Mission Control roles and permissions, see [Access Control](#) and [Users and Groups](#).

Procedure

- 1 In the Tanzu Mission Control console, click **Cluster Groups** in the left navigation pane.
- 2 Select the cluster you want to delete.
- 3 Click **Delete**.
- 4 Select one of the deletion options in the Delete cluster screen.
- 5 Enter the name of the cluster that you are deleting.
- 6 Click **Delete**.

Results

The AKS cluster is deleted.

Add an Existing Azure AKS Cluster into Tanzu Mission Control Management

Bring a pre-existing AKS cluster running in a connected Azure account under the management of Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to manage clusters.

- To bring an AKS cluster under Tanzu Mission Control management, you must be associated with the `managementcluster.admin` role, as well as the `clustergroup.edit` role to add it to a cluster group.
- You must also have the `cluster.admin` role on the cluster.

For more information about roles, see [Access Control](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add cluster**, and then choose **Manage existing AKS cluster** from the dropdown.
- 3 Select the account credential you want to use.
- 4 Enter the resource ID for the cluster you want to add.
 - a Select the method you want to use to find the resource ID for an existing clusters, either Azure CLI or the Azure portal.
 - b Follow the provided instructions to retrieve the resource ID.

5 Click **Next**.

6 Specify the cluster group name.

Note The agent name is a concatenation of the cluster name, region, credential name, and credential type. Use the agent name to refer to the cluster when using the API or CLI.

7 You can optionally add a label.

8 You can optionally configure a proxy.

9 Click **Manage**.

Note Following is a list of reasons that prevent a cluster from being managed by Tanzu Mission Control:

- Kubernetes version is not supported by Tanzu Mission Control
 - Kubernetes versions of any node pool under the cluster are not aligned with the version of the control plane
 - Windows cluster
 - Dual stack cluster
 - Arm architecture cluster
 - Cluster with "AvailabilitySet"
 - A cluster without run command capability (disabled)
 - Cluster already managed by another credential
 - Cluster and Nodepool not in Healthy/Ready state
-

Results

The cluster is now under Tanzu Mission Control management.

Upgrade an Azure AKS Cluster

Upgrade the version of Kubernetes that is running on an Azure AKS cluster under Tanzu Mission Control management.

See [Upgrade Kubernetes on Your Cluster](#) for information about how to upgrade Kubernetes on your cluster.

Modify AKS Cluster Node Pools

You can modify Azure AKS cluster node pools under Tanzu Mission Control management to change settings such as compute and scaling.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure that you are logged in to your Azure account.

Make sure you have the appropriate permissions to modify cluster node pools.

- To modify AKS cluster node pools under Tanzu Mission Control management, you must be associated with the `managementcluster.admin` role, as well as the `clustergroup.edit` role.
- You must also have the `cluster.admin` role on the cluster.

For more information about roles, see [Access Control](#).

Procedure

- 1 In the Tanzu Mission Control console, click **Clusters** in the left navigation pane.
- 2 Select the cluster for which you want to edit node pools.
- 3 Click the **Node Pools** tab.
- 4 In the **Node pool** screen you can select the options that you want, including scaling the number of nodes up or down, and selecting the image types and instance types. Follow the links in the console to find more information about specific Azure AKS node settings.

AKS Credential Resources

Tanzu Mission Control creates a number of AKS resources for AKS lifecycle management when you create an AKS credential.

Resources and Naming Convention

When you create an AKS credential in Azure using Tanzu Mission Control, it creates a number of resources, such as storage, keyvault, function, etc., and these management plane resources get put in their own resource group with the naming convention `vmwmtmc-XXXXXXX-XXXXXX`. They are tagged with the Tanzu Mission Control Org, Credential Name, and Account ID.

Managing the Lifecycle of AWS EKS Clusters

9

You can manage new and existing AWS EKS clusters and perform cluster lifecycle management operations including create, update, upgrade, and delete directly from Tanzu Mission Control.

To create new AWS EKS clusters and manage them with Tanzu Mission Control, see [Creating and Managing AWS EKS Clusters](#).

To add existing AWS EKS clusters to Tanzu Mission Control, see [Managing Existing AWS EKS Clusters](#).

Read the following topics next:

- [Creating and Managing New AWS EKS Clusters](#)
- [Adding Existing AWS EKS Clusters into Tanzu Mission Control Management](#)
- [Tanzu Mission AWS EKS Control Tags](#)

Creating and Managing New AWS EKS Clusters

You can create and manage AWS EKS credentials and perform cluster lifecycle management operations including create, update, upgrade, and delete directly from Tanzu Mission Control. Creating and managing AWS EKS clusters with Tanzu Mission Control includes multiple steps, which are summarized here.

The steps to create and manage new AWS EKS clusters are given below in the order in which they should be done.

Note Tanzu Mission Control is the source of truth for clusters created and managed by Tanzu Mission Control. Anything changed outside of Tanzu Mission Control is reported as drift and you will see an `Out of Sync` message.

Note For information about using the `tanzu` CLI to manage EKS clusters, see the topic for `ekscluster` in *VMware Tanzu CLI Reference - Tanzu Mission Control Plug-ins*.

Procedure

- 1 [Create a VPC with Subnets for EKS Cluster Lifecycle Management](#).

- 2 [Create an Account Credential for EKS Cluster Lifecycle Management.](#)
 - a You may need to [Update Your Credential for an AWS EKS Account.](#)
 - b If you delete a credential, you need to [Clean Up Your AWS EKS Account After Deleting a Credential.](#)
- 3 [Create an EKS Cluster.](#)

Constraints for Lifecycle Management of EKS Clusters

Make sure your AWS EKS clusters are properly configured. To manage the lifecycle of EKS clusters with Tanzu Mission Control, your clusters must satisfy the constraints described in this topic.

Supported Kubernetes Versions

Only the following versions of Kubernetes are supported:

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25

Clusters and nodes must be ACTIVE

All clusters and node groups in the cluster must be in an ACTIVE state.

Add-on validation

EKS supports the following add-ons, which are supported by Tanzu Mission Control as part of a managed cluster. These add-ons need policies in the Node IAM role. Tanzu Mission Control creates a Node IAM role when you create a CloudFormation stack while registering the AWS account with Tanzu Mission Control. This Node IAM role will have the necessary policies to run these four add-ons.

- All supported add-ons must be in an Active state.
- Supported add-ons:
 - Amazon VPC CNI
 - CoreDNS
 - kube-proxy
 - Amazon EBS CSI Driver

The Amazon EBS CSI Driver is not required for bringing an existing EKS cluster under Tanzu Mission Control management. However, it is added when clusters are provisioned through Tanzu Mission Control.

Only the add-ons in the list above are supported.

If any of these add-ons are missing, Tanzu Mission Control can not manage the cluster.

Node validation

If a cluster has unmanaged nodes, users are responsible for managing such nodes even if the cluster is managed by Tanzu Mission Control.

Any cluster without at least one x86 node group is not supported.

AWS Fargate

Tanzu Mission Control does not support EKS clusters with Fargate profiles.

Unsupported AMI types

- Windows based AMIs
- ARM based AMIs

IPv6

Tanzu Mission Control does not support lifecycle management of clusters with IPv6 networking.

EKS-managed third party cluster

Tanzu Mission Control does not support lifecycle management of EKS clusters that are third party clusters registered with EKS.

Create a VPC with Subnets for EKS Cluster Lifecycle Management

Use AWS tools to create a virtual private cloud (VPC) with subnets.

A functional EKS-compliant VPC is a prerequisite for EKS cluster deployments. AWS provides a CloudFormation template that creates a configuration that supports EKS cluster deployments. Use the template to create a VPC with subnets. For more information about VPCs and subnets for EKS, see [Amazon EKS VPC and subnet requirements and considerations](#)

Prerequisites

Log in to your AWS account.

Procedure

- 1 To create a VPC for lifecycle management of your EKS clusters, follow the procedure at <https://docs.aws.amazon.com/eks/latest/userguide/creating-a-vpc.html> in the *Amazon EKS documentation*.
- 2 Use the procedure entitled *Public and private subnets*.

3 Use the URL for IPv4 subnets.

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml
```

Results

The template creates a VPC with two public subnets and two private subnets.

Note Clusters must have outbound access.

What to do next

After you have created the VPC, you can create an account credential and start provisioning EKS clusters.

Create an Account Credential for EKS Cluster Lifecycle Management

Set up a credential that allows Tanzu Mission Control to manage resources in your AWS account.

An account credential is required for managing the lifecycle of EKS clusters.

Note There is a five (5) minute sync period between the AWS account and Tanzu Mission Control, so resources created (such as a subnet) may not appear as available in Tanzu Mission Control immediately. The syncing must be complete before you try to create another resource.

Prerequisites

Log in to the Tanzu Mission Control console.

Log in to your AWS account.

Make sure you have the appropriate permissions to create credentials.

- To create EKS credentials, you must be associated with the Tanzu Mission Control role `cluster.admin` role.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the the Accounts tab of the Administration page, click **Create Credential** and choose **AWS EKS**.
- 3 On the Create credential page, provide a name for the credential.
- 4 You can optionally provide a description and labels.
- 5 Click **Next**.
- 6 Click **Generate Template**, and then after the template is generated, click **Next**.

- 7 Use the generated template in one of two ways to create the AWS CloudFormation stack, either via the AWS CLI or the AWS console UI.
- 8 Retrieve the Role ARN using the command as shown below, or using the console by navigating to **CloudFormation** > **Stacks** > *<your stack>* > **Outputs**.

```
aws iam get-role --role-name
clusterlifecyle.<GeneratedTemplateID>.eks.tmc.cloud.vmware.com --query 'Role.Arn' --
output text
```

- 9 Copy the Role ARN and paste it into the Role ARN field.
- 10 Click **Create**.

Results

When you click **Create**, Tanzu Mission Control creates the credential. The process of creating and validating the credential can take up to 15 minutes.

Note that as part of the AWS EKS credential, the template creates the following AWS IAM roles:

- control-plane.\${GeneratedTemplateID}.eks.tmc.cloud.vmware.com - this is for control plane communications
- worker.\${GeneratedTemplateID}.eks.tmc.cloud.vmware.com - this is for the worker nodes
- lambda.\${GeneratedTemplateID}.eks.tmc.cloud.vmware.com - this role allows Lambda to retrieve EKS cluster, VPC, AMI, Region, and Availability Zone information
- cloudwatch.\${GeneratedTemplateID}.eks.tmc.cloud.vmware.com - this allows CloudWatch to invoke Lambda functions
- clusterlifecyle.\${GeneratedTemplateID}.eks.tmc.cloud.vmware.com - this role is for managing EKS cluster lifecycles

Note Deleting Tanzu Mission Control credentials does not delete these roles. After you have deleted your credentials you need to delete the Cloud Formation template to remove all these roles. For more information, see [Clean Up Your AWS EKS Account After Deleting a Credential](#).

What to do next

After you have created the credential, you can use it when creating an EKS cluster in your AWS account.

Update Your Credential for an AWS EKS Account

An invalid AWS EKS credential must be updated for cluster actions to function properly.

When you select a cluster on the Clusters page it might show a message indicating the credential for this cluster is no longer valid.

Prerequisites

Log in to the Tanzu Mission Control console.

Log in to your AWS account.

Make sure you have the appropriate permissions to manage credentials.

- To manage account credentials, you must be associated with the `cluster.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**.
- 2 On the Administration page, click the **Credentials** tab.
- 3 Click the menu icon for the invalid credential, and then choose **Update**.
- 4 Click **Generate Template**.
- 5 After the CloudFormation template is created, click **Next**.
- 6 Follow the on-screen instructions to update the CloudFormation stack using either the CLI or console.
- 7 Click **Update**.

Clean Up Your AWS EKS Account After Deleting a Credential

You can manually delete the remnants of your account credential in your AWS account if the procedure to delete a credential through Tanzu Mission Control fails to do so.

There are three types of things that are deployed in the AWS account that you need to clean up to complete the deletion: the lambdas, the Cloud Watch Events, and an `SSM` parameter.

This manual clean-up method is required only if the force credential delete fails.

Prerequisites

Log in to your AWS account.

Procedure

- 1 Find and delete the lambdas. To find the lambdas, from the AWS console, navigate to **Lambda > Functions** and search by tag. Tanzu Mission Control adds two types of tags to the lambdas:
 - `account-uid.cloud.vmware.com` is the user account/org_id
 - `account.tmc.cloud.vmware.com` is the name of the credential
- 2 Find and delete the Cloud Watch events, which are listed under the Amazon Event bridge rules. These have the same tags applied as above, and there are two event rules that correspond to the two lambdas from the previous step.

3 Search for the `ssm` parameter in the Amazon Systems Manager parameter store. There is one parameter that Tanzu Mission Control uses called the `agent_token`, and it has the same two tags as above.

4 Locate and delete the CloudFormation template.

You can search CloudFormation by the credential name. Delete that CloudFormation template after the credential is deleted from Tanzu Mission Control. This deletes all of the following:

- `control-plane.${GeneratedTemplateID}.eks.tmc.cloud.vmware.com` - this is for control plane communications
- `worker.${GeneratedTemplateID}.eks.tmc.cloud.vmware.com` - this is for the worker nodes
- `lambda.${GeneratedTemplateID}.eks.tmc.cloud.vmware.com` - this role allows Lambda to retrieve EKS cluster, VPC, AMI, Region, and Availability Zone information
- `cloudwatch.${GeneratedTemplateID}.eks.tmc.cloud.vmware.com` - this allows CloudWatch to invoke Lambda functions
- `clusterlifecycle.${GeneratedTemplateID}.eks.tmc.cloud.vmware.com` - this role is for managing EKS cluster lifecycles

Create an EKS Cluster

Create an EKS cluster in your connected Amazon Web Services (AWS) account.

After you connect an AWS account, use Tanzu Mission Control to provision resources and create EKS clusters.

When creating EKS clusters, Tanzu Mission Control uses a naming scheme of `eks.<credentials>.<region>.<name>`.

Prerequisites

- You must have an account credential to provide access to your AWS account, as described in [Create an Account Credential for EKS Cluster Lifecycle Management](#).
- You must have a VPC and subnets configured, as described in [Create a VPC with Subnets for EKS Cluster Lifecycle Management](#).
- Log in to the Tanzu Mission Control console.
- To create EKS clusters, you must be associated with the Tanzu Mission Control `cluster.admin`, `cluster.view`, `cluster.update`, and `account.credential.get` roles.

For more information, see [Access Control](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 From the Tanzu Mission Control console, click **Clusters** in the left navigation pane.
- 2 Click **Create cluster**, and then choose **Create EKS cluster** from the dropdown.

- 3 Enter a name for the cluster, and specify the cluster group. The cluster name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a maximum length of 63 characters.
- 4 You can optionally add a description and label.
- 5 Click **Next**.
- 6 Configure the control plane.
 - a Select the account credential for the account in which you want to create the cluster.
 - b Select the version of Kubernetes to use for the cluster.
 - c In the **Network** section, select the region where your VPC is configured.
 - d Select your VPC.
 - e Click **Next**.
- 7 Configure node pools.
 - a Enter a name for the node pool and optionally a description of the node pool.
 - b Optionally, add an EKS tag, a Kubernetes tag, and a Kubernetes taint.
 - c Configure the Compute parameters by selecting an AMI and an instance type. If you are using a custom AMI, select it from the list.
 - d Configure the Scaling parameters.
- 8 Click **Create**.

Assume an EKS Cluster Lifecycle Role in Tanzu Mission Control

Administrators can assign roles that allow users to assume EKS lifecycle management permissions in Tanzu Mission Control.

There may be instances when a user can not access EKS clusters because the nodes didn't come up. An administrator can edit the configuration to allow an AWS IAM user to access the cluster. The role can be edited either via the AWS console or AWS CLI.

Prerequisites

Make sure that you are logged in to your AWS account.

Procedure

- 1 Use the key name, which is part of the Tanzu Mission Control label, to search for roles in the AWS console.

- If you do not have a kubeconfig, add your ARN as the trusted relationship to the `clusterlifecycle.<tmc.cloud.vmware.com/cred-cloudformation-key>.eks.tmc.cloud.vmware.com` role created for the credential:

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam:xxxxxxx:user/<username>"
    },
    "Action": "sts:AssumeRole"
},
```

The name of the cluster lifecycle role will be `clusterlifecycle.<tmc.cloud.vmware.com/cred-cloudformation-key>.tmc.cloud.vmware.com`. The `tmc.cloud.vmware.com/cred-cloudformation-key` can be found as an annotation on the Tanzu Mission Control credential.

- Use the AWS CLI to get the kubeconfig and access the cluster:

```
aws sts assume-role --role-arn "arn:aws:iam:xxxxxxx:role/
clusterlifecycle.<tmc.cloud.vmware.com/cred-cloudformation-key>
.eks.tmc.cloud.vmware.com" --role-session-name AWSCLI-Session

aws eks update-kubeconfig --region <region-name> --name <cluster-name> --kubeconfig file-
name
```

Enable IAM Principal Access to Clusters Managed by Tanzu Mission Control

You can assign a user role ARN to get access to AWS EKS clusters managed by Tanzu Mission Control

The steps below describe how to modify the configuration if you already have a kubeconfig. For more information, see the AWS documentation for [Enabling IAM principal access to your cluster](#)

Prerequisites

Make sure that you are logged in to your AWS account.

Procedure

- Log in to the AWS console and get your user ARN from the Identity and Access Management (IAM) page.
- Download the following YAML template for configuring the necessary ClusterRole and ClusterRoleBinding settings and apply it to your EKS cluster.

```
curl -o eks-console-full-access.yaml https://amazon-eks.s3.us-west-2.amazonaws.com/docs/
eks-console-full-access.yaml

kubectl apply -f eks-console-full-access.yaml
```

3 Edit the configmap:

```
kubectl edit configmap/aws-auth -n kube-system
```

4 Add the following under the mapRoles section:

```
- "groups":
  - "system:masters"
  "rolearn": <value here should be any rolearn that the admin wants to grant access to>
  "username": "<Kubernetes user name>"
```

For example:

```
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      rolearn: arn:aws:iam::1234197287370:role/adopted-nodegroup-1
      username: system:node:{{EC2PrivateDNSName}}
    ...
```

Adding Existing AWS EKS Clusters into Tanzu Mission Control Management

You can use Tanzu Mission Control to manage existing AWS EKS clusters.

Do the following to bring existing AWS EKS clusters under Tanzu Mission Control management.

- 1 [Update aws-auth Configmap for Access to Existing EKS Clusters](#)
- 2 [Add an Existing EKS Cluster into Tanzu Mission Control Management](#)

Update aws-auth Configmap for Access to Existing EKS Clusters

You must edit the `aws-auth` configmap to give the cluster lifecycle role access to the EKS cluster.

Prerequisites

- You must have an account credential to provide access to your AWS account, as described in [Create an Account Credential for EKS Cluster Lifecycle Management](#).

Procedure

- 1 Connect to your AWS account via AWS CLI.
- 2 Run the following command to add a new kubeconfig context for the cluster you want to bring under Tanzu Mission Control management:

```
aws eks update-kubeconfig --name mycluster-name --kubeconfig mykubeconfig-name --region myregion-name
```

For example:

```
aws eks update-kubeconfig --name myusername-cna-eksattached --kubeconfig myusername-cna-eksattached.yaml --region east
```

- 3 Run the following command to get the `aws-auth` configmap:

```
kubectl get configmap -n kube-system aws-auth -o yaml --kubeconfig=path-to/kubeconfig
```

For example:

```
kubectl get configmap -n kube-system aws-auth -o yaml --kubeconfig=myusername-cna-eksattached.yaml
```

Note The `aws-auth` configmap is not created until there is at least one node group associated with the cluster.

- 4 Run the following command to create a new group and cluster role binding:

```
kubectl create clusterrolebinding my-cluster-rolebinding-name \
--clusterrole=cluster-admin \
--group=mygroup-name \
--kubeconfig=mykubeconfig-name
```

For example:

```
kubectl create clusterrolebinding my-privileged-cluster-role-binding \
--clusterrole=cluster-admin \
--group=tmc-cluster-access \
--kubeconfig=myusername-cna-eksattached.yaml
```

- 5 Run the following command to edit the `aws-auth` configmap:

```
kubectl edit configmaps -n kube-system aws-auth -o yaml --kubeconfig=mykubeconfig-name
```

For example:

```
kubectl edit configmaps -n kube-system aws-auth -o yaml --kubeconfig=myusername-cna-eksattached.yaml
```

- 6 Add the following code block to the `mapRoles` section of the `aws-auth` configmap:

```
- groups:
  - mygroup-name
  rolearn: my-TMC-EKS-credential-ARN
```

For example:

```
- groups:
  - tmc-cluster-access
    rolearn: arn:aws:iam::1234567890:role/
clusterlifecycle.1234567890.eks.tmc.cloud.vmware.com
```

Note To gather the role created by the CloudFormation Stack ARN:

- a From the AWS Console, navigate to **CloudFormation > Stacks**.
- b Select the stack which corresponds to the appropriate credential.

Tanzu Mission Control adds a label key to help map your EKS credential to the appropriate CloudFormation stack (for example, *eks-tmc-cloud-vmware-com-1234567812345678165*).

- 7 (Optional) Run the following command to validate changes to and formatting of the `aws-auth` configmap:

```
kubectl get configmap aws-auth -n kube-system -o yaml --kubeconfig=mykubeconfig-name
```

For example:

```
kubectl get configmap aws-auth -n kube-system -o yaml --kubeconfig=myusername-cna-eksattached.yaml
```

Add an Existing EKS Cluster into Tanzu Mission Control Management

Use Tanzu Mission Control to manage an existing EKS cluster running in a connected AWS account.

When you connect an AWS account for cluster lifecycle management with Tanzu Mission Control, the EKS clusters are visible to Tanzu Mission Control, but they are not added into Tanzu Mission Control management by default. To add an AWS EKS cluster into Tanzu Mission Control management, you must identify the EKS cluster you want to add, and then specify the cluster group with which to associate the cluster.

Note The Tanzu Mission Control provider in Terraform does not support the ability to add an AWS EKS cluster into Tanzu Mission Control management.

Prerequisites

Make sure your EKS clusters satisfy the minimum requirements for lifecycle management through Tanzu Mission Control, as described in [Constraints for Lifecycle Management of EKS Clusters](#).

Make sure that you have edited the `aws-auth` configmap to provide lifecycle role access to the EKS cluster, as described in [Update aws-auth Configmap for Access to Existing EKS Clusters](#).

Make sure that you have defined an account credential to provide access to your AWS account, as described in [Create an Account Credential for EKS Cluster Lifecycle Management](#).

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to create clusters.

- To add an AWS EKS cluster into Tanzu Mission Control management, you must be associated with the Tanzu Mission Control `cluster.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Clusters** in the left navigation pane.
- 2 Click **Manage existing cluster**.
- 3 In section 1 of the **Manage existing EKS clusters** dialog, specify the AWS Account credential.
You can click **Refresh Account Information** to refetch the selected account information and populate the rest of the dialog.
- 4 Select the EKS cluster that you want to manage by providing its ARN. You can retrieve the ARN using either the AWS console or the AWS CLI. Paste the cluster ARN into the **Cluster ARN** field and click **Next**.
- 5 In section 2 of the **Manage existing EKS clusters** dialog, specify the name of the cluster group for the newly managed cluster.

When you click **Next**, Tanzu Mission Control searches for a cluster with that name. If the cluster name already exists, you are prompted to change the name to a Tanzu Mission Control compatible name. The cluster name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a maximum length of 63 characters.

For example, if the EKS cluster name is `ABCD`, that is an invalid Tanzu Mission Control name, and you are prompted to change the name to a Tanzu Mission Control compatible name. If you change it to `abcd`, for example, the agent name changes to `eks.cred_name.region.abcd`, while the EKS cluster API will still use `ABCD` as the cluster name.

- 6 In section 3 of the **Manage existing EKS clusters** dialog you can optionally configure the Proxy configuration.
- 7 Click **Manage**.

Results

The cluster is now under Tanzu Mission Control management. Note that it can take up to five minutes before you can perform lifecycle management actions on your newly managed EKS cluster.

Tanzu Mission AWS EKS Control Tags

Tanzu Mission Control uses the tags below to identify resources for AWS EKS lifecycle management.

Tags

Tanzu Mission Control uses the tags given below.

```
"Tags": [  
  {  
    "Key": "template-version.eks.tmc.cloud.vmware.com",  
    "Value": "1.0"  
  },  
  {  
    "Key": "org-id.tmc.cloud.vmware.com",  
    "Value": {  
      "Ref": "OrgID"  
    }  
  },  
  {  
    "Key": "cf-stack-name.eks.tmc.cloud.vmware.com",  
    "Value": {  
      "Ref": "AWS::StackName"  
    }  
  },  
  {  
    "Key": "account-name.eks.tmc.cloud.vmware.com",  
    "Value": {  
      "Ref": "CredentialName"  
    }  
  }  
]
```


Managing the Lifecycle of Tanzu Kubernetes Clusters

10

As a platform operator or infrastructure operator, you can use VMware Tanzu Mission Control to register Tanzu Kubernetes Grid management clusters, provision new workload clusters, and manage the lifecycle of these clusters.

When you register a management cluster (Tanzu Kubernetes Grid or Tanzu Kubernetes Grid Service), you can bring all of its workload clusters under the management of Tanzu Mission Control, which allows you to facilitate consistent management using all of the capabilities of Tanzu Mission Control, as well as provisioning resources and creating new clusters directly from Tanzu Mission Control.

Note Version 2.5 of Tanzu Kubernetes Grid does not support deployment of clusters to AWS or Azure. To manage the lifecycle of clusters on these platforms using Tanzu Mission Control, you can create a credential to connect to your account, and then deploy and manage native EKS/AKS clusters.

For more information about managing the lifecycle of Tanzu Kubernetes clusters, see [Cluster Lifecycle Management](#) in *VMware Tanzu Mission Control Concepts*.

Read the following topics next:

- [Register a Management Cluster with Tanzu Mission Control](#)
- [Deregister a Management Cluster from Tanzu Mission Control](#)
- [Add a Workload Cluster into Tanzu Mission Control Management](#)
- [Remove a Workload Cluster from Tanzu Mission Control Management](#)
- [Provisioning Tanzu Kubernetes Grid Workload Clusters](#)
- [Create a Node Pool](#)
- [Edit a Node Pool](#)
- [Delete a Node Pool](#)
- [Relaxing Pod Security in a Provisioned Cluster](#)
- [Upgrade Kubernetes on Your Cluster](#)
- [Delete a Provisioned Cluster](#)
- [Manage Certificates](#)

Register a Management Cluster with Tanzu Mission Control

As a platform operator, you can work with your Tanzu Kubernetes Grid administrator to register a management cluster with VMware Tanzu Mission Control to enable lifecycle management of its workload clusters.

When you register a management cluster, you create secure connection to Tanzu Mission Control that allows you to subsequently bring its workload clusters under management, assign them to cluster groups, and apply policies. You can also manage the entire lifecycle of your clusters (including create, update, and delete) from Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console.

To register a Tanzu Kubernetes Grid management cluster with Tanzu Mission Control requires the following permissions:

- You must have `admin` privileges on the management cluster.
For clusters running in vSphere or vSphere with Tanzu, this is typically the vSphere administrator.
- In Tanzu Mission Control, you must be associated with the `managementcluster.admin` role.

Make sure your management cluster satisfies the minimum requirements, as described in [Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control](#) in *VMware Tanzu Mission Control Concepts*.

- To register a Tanzu Kubernetes Grid management cluster running in Amazon Web Services (AWS) with Tanzu Mission Control requires permissions in addition to the minimum requirements for deploying the management cluster in your AWS account. For more information, see [Prepare to Deploy Management Clusters to Amazon EC2](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*.

If you have a proxy server that manages outbound traffic for your clusters, you need to enable the cluster to communicate with Tanzu Mission Control through the proxy.

- You can create a proxy configuration object in Tanzu Mission Control and use it when registering, provisioning, or attaching the cluster. For more information, see [Chapter 5 Connecting Through a Proxy](#).
- You can enable all outbound traffic to Tanzu Mission Control for the proxy server by adding some URLs to the proxy server's allowlist, as described in [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

Note Tanzu Mission Control does not support proxy configuration for Tanzu Kubernetes clusters running in AWS.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.

- 2 Click the **Management clusters** tab.
- 3 Click **Register Management Cluster**, and then choose the kind of cluster you are registering.
 - For management clusters running in Tanzu Kubernetes Grid, choose **Tanzu Kubernetes Grid**.
 - For Tanzu Kubernetes Grid Service Supervisor Clusters running in vSphere with Tanzu or VMware Cloud with Tanzu services, choose **vSphere with Tanzu**.
- 4 On the Register page, provide a name for the management cluster, and select a default cluster group for workload clusters.

When you add workload clusters, you can choose the cluster group into which to place them. This setting simply indicates the default choice.

- 5 You can optionally provide a description and labels for the management cluster.
- 6 Click **Next**.
- 7 You can optionally toggle on local image registry and select a proxy configuration for the cluster.
 - a Click to toggle the **Local Image Registry Setting** option to **Yes** to enable it, then select the image registry from the dropdown list.

You can optionally accept the option to set the default workload cluster local image registry to be the same as what you selected for the management cluster. If you deselect this option, you can select a different registry or leave it blank.
 - b Click to toggle the **Set proxy** option to **Yes**.
 - c Select the proxy configuration you defined for this cluster.
 - d You can optionally specify an alternative proxy configuration to use as the default selection for managed workload clusters.

When you add workload clusters, you can choose which proxy configuration to use. This setting simply indicates the default choice.

Note To use a proxy configuration to register a Tanzu Kubernetes Grid Service Supervisor Cluster, the cluster must be running on vSphere 7.0.3a or later.

- 8 Click **Next**.

When you click **Next**, Tanzu Mission Control generates a YAML file that defines how the management cluster connects to Tanzu Mission Control for registration. The credential provided in the YAML expires after 48 hours. You can optionally click View YAML to see the code.

- 9 Copy the URL provided on the Register page, and give it to the administrator of your Tanzu Kubernetes Grid deployment to install the cluster agent on your management cluster and complete the registration process.

For clusters running in vSphere or vSphere with Tanzu, this is typically the vSphere administrator.

- 10 Click **View Management Cluster**.

Results

When you click **View Management Cluster**, the management cluster detail page is displayed. Because the registration is not yet complete, you cannot yet view the contents of the management cluster.

What to do next

After you have started the registration process from Tanzu Mission Control, you must complete the registration on the Tanzu Kubernetes Grid side within 48 hours.

- To complete the registration for Tanzu Kubernetes Grid Service in vSphere with Tanzu, see [Complete the Registration of a Supervisor Cluster in vSphere with Tanzu](#).
- To complete the registration for a management cluster running in Azure or vSphere, see [Complete the Registration of a Management Cluster](#).

Complete the Registration of a Management Cluster

As a cluster administrator, you can register a Tanzu Kubernetes Grid or Tanzu Community Edition management cluster, that is running in Azure, AWS, or vSphere, with your Tanzu Mission Control organization.

To register a management cluster, you run the cluster agent registration script, provided by Tanzu Mission Control, on the cluster. The script creates a namespace and installs a set of cluster agent extensions and custom resource definitions into your cluster, which enables communication with Tanzu Mission Control.

Prerequisites

This procedure assumes that you have already started the registration process in Tanzu Mission Control as described in [Register a Management Cluster with Tanzu Mission Control](#), and that you have a Tanzu Kubernetes Grid management cluster that satisfies the minimum requirements, as described in [Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control](#) in *VMware Tanzu Mission Control Concepts*.

Make sure you have the appropriate permissions.

- To register a management cluster with Tanzu Mission Control, you must have `admin` permissions on the cluster.
- To use a proxy configuration defined in Tanzu Mission Control, you must be associated with the `organization.credential.view` role.

Procedure

- 1 If you are using a proxy configuration, use the generated `tmc` command to register the cluster.
 - a In a command window, log in with the Tanzu Mission Control CLI (`tmc`), making sure you have the latest version.
 - b Run the `tmc managementcluster register` command provided by Tanzu Mission Control, inserting the appropriate `kubeconfig`.

For example:

```
tmc managementcluster register my-mgmt-cluster --kubeconfig my-mgmt-cluster-kubeconfig
--continue-bootstrap
```

- 2 Use the generated YAML manifest in a `kubectl apply` command to register the cluster, if you are not using a proxy configuration.
 - a In a command window, connect to the management cluster with `kubectl`.

Make sure your current context is set appropriately for the management cluster you want to register, and not one of its workload clusters.
 - b Run a `kubectl apply` command like the following to start the installation.

Make sure you use the URL provided by Tanzu Mission Control, and that it is enclosed in quotes.

```
kubectl apply -f "https://my-org.tmc.cloud.vmware.com/installer?
id=verylonginstallerid&source=registration"
```

Results

When you run the appropriate command, a namespace called `vmware-system-tmc` is created, and then the Tanzu Mission Control cluster agent is installed on the management cluster. The installation process may take a few minutes.

When the installation is complete, your management cluster is registered with Tanzu Mission Control. You can return to the Tanzu Mission Control console and view the registered cluster on the Management clusters tab of the Administration page. It might take a few minutes for Tanzu Mission Control to start receiving health information from the management cluster.

Note Management clusters created using Tanzu Community Edition prior to version 0.12 are shown as Tanzu Kubernetes Grid clusters.

Complete the Registration of a Supervisor Cluster in vSphere with Tanzu

As a vSphere administrator, you can register a Tanzu Kubernetes Grid Service Supervisor Cluster running in vSphere with Tanzu with Tanzu Mission Control.

Note The procedure described in this topic explains how to complete the registration of a Supervisor Cluster running in vSphere with Tanzu using the command-line approach. You can also accomplish this task using the vSphere Client web interface, as described in [Integrate the Tanzu Kubernetes Grid on the Supervisor with Tanzu Mission Control](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Prerequisites

This procedure assumes that you have already started the registration process in Tanzu Mission Control as described in [Register a Management Cluster with Tanzu Mission Control](#), and that you have a Tanzu Kubernetes Grid Service Supervisor Cluster running in vSphere with Tanzu.

To register a Supervisor Cluster running in vSphere with Tanzu with Tanzu Mission Control so that you can manage the lifecycle of its workload clusters, your vSphere instance must be one of the following:

- vSphere 7.0 U3a or later
- vSphere 8.0 U1 or later

Procedure

- 1 In a command window, log in to the Supervisor Cluster with administrative credentials, as described in [Connect to the Supervisor Cluster as a vCenter Single Sign-On User](#) in the *vSphere with Tanzu Configuration and Management* documentation.
- 2 Make sure your current context is set appropriately for the Supervisor Cluster you want to register.

You might need to run a `kubectl config use-context` command to explicitly set the context, for example:

```
kubectl config use-context CONTEXT-NAME-IP
```

Use the context that is listed as an IP address.

- 3 Locate the Tanzu Mission Control namespace on the Supervisor Cluster.

The following `kubectl` command returns namespaces on the cluster.

```
kubectl get ns
```

The Tanzu Mission Control namespace begins with `svc-tmc-`, for example:

```
svc-tmc-c8           Active   14m
```

- 4 Create a YAML file to contain the `AgentInstall` resource information, for example `tmc-registration.yaml`.

- a Enter the following YAML code in the file.

```
apiVersion: installers.tmc.cloud.vmware.com/v1alpha1
kind: AgentInstall
metadata:
  name: tmc-agent-installer-config
  namespace: TMC-NAMESPACE
spec:
  operation: INSTALL
  registrationLink: TMC-REGISTRATION-URL
```

- b Replace `TMC-NAMESPACE` with the name of namespace that you retrieved from the cluster.
- c Replace `TMC-REGISTRATION-URL` with the URL provided by Tanzu Mission Control when you started the registration process.

The resulting YAML file should look something like this:

```
apiVersion: installers.tmc.cloud.vmware.com/v1alpha1
kind: AgentInstall
metadata:
  name: tmc-agent-installer-config
  namespace: svc-tmc-c8
spec:
  operation: INSTALL
  registrationLink: https://myorg.tmc.cloud.vmware.com/installer?id=121f2verylongstring23e&source=registration
```

- d Save and close the file.
- 5 In your command window, run a `kubectl create` command to apply the registration YAML to your Supervisor Cluster.

For example:

```
kubectl create -f tmc-registration.yaml
```

When you run this command the Tanzu Mission Control cluster agent is installed on the Supervisor Cluster. The resulting output looks something like this:

```
agentinstall.installers.tmc.cloud.vmware.com/tmc-agent-installer-config created
```

- 6 You can optionally run a `kubectl describe` command like the following to monitor the progress of the installation.

```
kubectl -n TMC-NAMESPACE describe agentinstall tmc-agent-installer-config
```

When the `status:` line at the bottom of the output changes from `INSTALLATION_IN_PROGRESS` to `INSTALLED`, the installation is complete.

Results

When the installation is complete, your Supervisor Cluster running in vSphere with Tanzu is registered with Tanzu Mission Control. You can return to the Tanzu Mission Control console and view the registered Supervisor Cluster on the Management clusters tab of the Administration page. It might take a few minutes for Tanzu Mission Control to start receiving health information from the Supervisor Cluster.

Deregister a Management Cluster from Tanzu Mission Control

As a platform operator, you can remove the integration between a management cluster and Tanzu Mission Control.

Prerequisites

This task assumes you have registered a management cluster with Tanzu Mission Control as described in [Register a Management Cluster with Tanzu Mission Control](#).

Before you can deregister a management cluster, you must first remove its workload clusters from Tanzu Mission Control management, as described in [Remove a Workload Cluster from Tanzu Mission Control Management](#).

To deregister a management cluster from Tanzu Mission Control requires the following permissions:

- You must have `admin` privileges on the management cluster.
For Tanzu Kubernetes Grid Service in vSphere with Tanzu, this is typically the vSphere administrator.
- In Tanzu Mission Control, you must be associated with the `organization.admin` role.

Log in to the Tanzu Mission Control console.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 Click the **Management clusters** tab, and then click the management cluster you want to deregister.
- 3 On the management cluster detail page, click **Deregister**.
- 4 In the confirmation dialog, select the deregistration option to use.
 - Select **Deregister and remove agent** to remove the cluster agent extensions from the management cluster and deregister it from Tanzu Mission Control.
 - Select **Deregister only** to release the management cluster from Tanzu Mission Control management without attempting to remove the cluster agent extensions.

This option might be useful if the management cluster has been deleted or Tanzu Mission Control no longer has connectivity to the management cluster.

- 5 Enter the name of the management cluster, and then click **Deregister**.

Results

When you click **Deregister**, Tanzu Mission Control releases the management cluster, and it is no longer listed on the Management clusters tab of the Administration page. If you selected **Deregister and remove agent**, Tanzu Mission Control attempts to remove the cluster agent extensions from the management cluster before deregistering it. This process might take a few minutes.

What to do next

If you selected **Deregister only**, Tanzu Mission Control releases the management cluster, but does not attempt to remove the cluster agent extensions. Your Tanzu Kubernetes Grid administrator must manually remove the cluster agent extensions from the management cluster.

If your management cluster is a Tanzu Kubernetes Grid Service supervisor cluster running in vSphere with Tanzu, your Tanzu Kubernetes Grid administrator is typically the vSphere administrator. For more information, see [Manually Remove the Cluster Agent from a Supervisor Cluster in vSphere with Tanzu](#).

Manually Remove the Cluster Agent from a Supervisor Cluster in vSphere with Tanzu

As a vSphere administrator, you can remove the Tanzu Mission Control cluster agent from a Tanzu Kubernetes Grid Service supervisor cluster running in vSphere with Tanzu.

When you register a Tanzu Kubernetes Grid Service Supervisor Cluster running in vSphere with Tanzu, Tanzu Mission Control installs a cluster agent to facilitate management of the cluster from Tanzu Mission Control. If you subsequently deregister the cluster without removing the cluster agent, you need to remove the agent manually.

Prerequisites

This procedure assumes that you have a Tanzu Kubernetes Grid Service Supervisor Cluster running in vSphere with Tanzu that has been registered with Tanzu Mission Control.

To remove the cluster agent from a Supervisor Cluster, you must have administrator access to the Supervisor Cluster.

Procedure

- 1 In a command window, log in to the Supervisor Cluster with administrative credentials, as described in [Connect to the Supervisor Cluster as a vCenter Single Sign-On User](#) in the *vSphere with Tanzu Configuration and Management* documentation.

- 2 Make sure your current context is set appropriately for the Supervisor Cluster from which you want to remove the cluster agent.

You might need to run a `kubectl config use-context` command to explicitly set the context, for example:

```
kubectl config use-context CONTEXT-NAME
```

- 3 Locate the Tanzu Mission Control namespace on the Supervisor Cluster.

The following `kubectl` command returns namespaces on the cluster.

```
kubectl get ns
```

The Tanzu Mission Control namespace begins with `svc-tmc-`, for example:

```
svc-tmc-c8                Active    14m
```

- 4 Delete the existing `AgentInstall` resource.

Run the following command, replacing `TMC-NAMESPACE` with the name of your Tanzu Mission Control namespace.

```
kubectl delete -n TMC-NAMESPACE agentinstall tmc-agent-installer-config
```

The output from this command looks something like this:

```
agentinstaller.installers.tmc.cloud.vmware.com "tmc-agent-installer-config" deleted
```

- 5 Create a YAML file to contain the `AgentInstall` resource information, for example `deregistration.yaml`.
 - a Enter the following YAML code in the file, replacing `TMC-NAMESPACE` with the name of your Tanzu Mission Control namespace.

```
apiVersion: installers.tmc.cloud.vmware.com/v1alpha1
kind: AgentInstall
metadata:
  name: tmc-agent-installer-config
  namespace: TMC-NAMESPACE
spec:
  operation: UNINSTALL
```

- b Save and close the file.

- 6 In your command window, run a `kubectl apply` command to apply the deregistration YAML to your Supervisor Cluster.

For example:

```
kubectl apply -f deregistration.yaml
```

When you run this command, the uninstall process begins. The resulting output looks something like this:

```
agentinstall.installers.tmc.cloud.vmware.com/tmc-agent-installer-config created
```

Results

After you issue the command to uninstall the agent, the Supervisor Cluster starts removing the agent. This process can take a few minutes. When the process completes, the Tanzu Mission Control cluster agent is uninstalled from the Supervisor Cluster.

Add a Workload Cluster into Tanzu Mission Control Management

Use Tanzu Mission Control to manage Tanzu Kubernetes clusters running in a registered management cluster.

When you register a Tanzu Kubernetes Grid management cluster with Tanzu Mission Control, pre-existing workload clusters running in the management cluster are exposed to Tanzu Mission Control, but they are not added into Tanzu Mission Control management by default. You must identify the workload clusters you want to add and specify the cluster group with which to associate the clusters.

Prerequisites

This procedure assumes that you have a Tanzu Kubernetes Grid cluster, and have already registered its management cluster with Tanzu Mission Control.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions:

- To bring a workload cluster under Tanzu Mission Control management, you must be associated with the `managementcluster.admin` role, as well as the `clustergroup.edit` role to add it to a cluster group.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**, and then click the **Management clusters** tab.
- 2 In the table of management clusters, click the management cluster that contains the workload cluster you want to add.

Note Management clusters created using Tanzu Community Edition prior to version 0.12 are shown as Tanzu Kubernetes Grid clusters.

- 3 On the management cluster detail page, click the **Workload clusters** tab.
- 4 In the list of workload clusters, select the clusters you want to add by clicking the checkbox next to the name, and then click **Manage # Cluster(s)**.

- 5 In the confirmation dialog, you can optionally change the default selections for the workload cluster.

The default selections for cluster group and proxy configuration are specified by the management cluster.

- a You can optionally specify an alternative cluster group for this workload cluster.
- b You can optionally specify an alternative proxy configuration for this workload cluster.
- c You can optionally specify a local image registry for this workload cluster.

- 6 Click **Manage**.

Results

When you click **Manage**, Tanzu Mission Control installs the cluster agent extensions on the workload cluster and adds it to the specified cluster group.

Remove a Workload Cluster from Tanzu Mission Control Management

Release a workload cluster that was previously added to Tanzu Mission Control management.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions:

- To remove a workload cluster from Tanzu Mission Control management, you must be associated with the `managementcluster.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Administration**, and then click the **Management clusters** tab.
- 2 In the table of management clusters, click the management cluster that contains the workload cluster you want to remove from management.
- 3 On the management cluster detail page, click the **Workload clusters** tab.
- 4 In the list of workload clusters, select the cluster you want to remove by clicking the checkbox next to the name, and then click **Unmanage Cluster**.

You cannot remove multiple workload clusters at the same time.

- 5 In the confirmation dialog, click **Unmanage**.

Results

When you click **Unmanage**, Tanzu Mission Control removes the cluster agent extensions from the workload cluster and removes it from the cluster group.

Provisioning Tanzu Kubernetes Grid Workload Clusters

If you've registered a management cluster or Supervisor Cluster with Tanzu Mission Control, you can create workload clusters directly from the Tanzu Mission Control console. You'll follow slightly different instructions depending on where your management cluster or Supervisor Cluster is running.

Note Version 2.5 of Tanzu Kubernetes Grid does not support deployment of clusters to AWS or Azure. To manage the lifecycle of clusters on these platforms using Tanzu Mission Control, you can create a credential to connect to your account, and then deploy and manage native EKS/AKS clusters.

About creating clusters using ClusterClass

You can leverage the power of `ClusterClass` to use a predefined baseline configuration to create workload clusters with a consistent size and shape. This functionality is available in the following versions of Tanzu Kubernetes Grid:

- Tanzu Kubernetes Grid version 2.1 or later
- Tanzu Kubernetes Grid Service running in vSphere with Tanzu on vSphere version 8.0 or later

Using Tanzu Mission Control, you can provision new workload clusters in management clusters running in these environments using only the `ClusterClass` workflow.

For more information, see the following topics:

- [Provision a Cluster using a Cluster Class](#)
- [Provision a Cluster in vSphere with Tanzu using a Cluster Class](#)

Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster

Use VMware Tanzu Mission Control to create a provisioner in your management cluster.

To create workload clusters in your Tanzu Kubernetes Grid management cluster, you must have a provisioner. For more information about provisioners and Tanzu Kubernetes clusters, see [Cluster Lifecycle Management](#) in *VMware Tanzu Mission Control Concepts*.

Prerequisites

Make sure you have the appropriate permissions to create a provisioner.

Procedure

- ◆ For a Tanzu Kubernetes cluster running in vSphere, AWS, or Azure, you add a provisioner by creating a namespace in the management cluster, which you can do using `kubectl`. For more information, see [Create Namespaces in the Management Cluster](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*.

- ◆ For a Tanzu Kubernetes cluster running in vSphere with Tanzu, you add a provisioner by creating a vSphere namespace in the Supervisor Cluster, which you can do in your vSphere environment. For more information, see [Configuring and Managing vSphere Namespaces](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Provision a Cluster using a Cluster Class

Use VMware Tanzu Mission Control to create a new Tanzu Kubernetes cluster using a cluster class.

Starting in version 2.1 of Tanzu Kubernetes Grid, you can leverage the power of `ClusterClass` to use a predefined baseline configuration to create workload clusters with a consistent size and shape.

For more information about the variables that you can use in class-based clusters in Tanzu Kubernetes Grid, see the following topics in the *VMware Tanzu Kubernetes Grid Product Documentation*:

- [Configuration File Variable Reference](#)
- [Configuration File to Cluster Class Variable Translations](#)

For more information about `ClusterClass` in the Cluster API, see [Introducing ClusterClass and Managed Topologies in Cluster API](#) in the *Kubernetes Blog*.

Note The data protection features of Tanzu Mission Control are not compatible with clusters created using the Tanzu Kubernetes release `v1.23.8---vmware.2-tkg.1-zshippable`. If you rely on Tanzu Mission Control for data protection, use a different Tanzu Kubernetes release when creating your clusters.

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid management cluster with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#)

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- To see and use a cloud provider account connection for creating a cluster, you must be associated with the `organization.credential.view` role.
- You must also have `admin` privileges on the management cluster to provision resources within it.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.

If the management cluster that you selected is running in version 2.1 (or later) of Tanzu Kubernetes Grid, you enter the workflow to create a cluster using a cluster class.

- 4 On the Create cluster page, select the provisioner in which you want to create the cluster, and then click **Next**.
- 5 Enter the name, group, class, and other details for the cluster.

- a Enter a name for the cluster.

Cluster names must be unique within a management cluster.

- b Select the cluster group to which you want to attach your cluster.

- c Select the cluster class that you want to use as the template for this cluster.

The list of cluster classes that you can choose from is taken from the provisioner namespace in your management cluster.

- d You can optionally enter a description and apply labels.

- e Click **Next**.

- 6 Configure your cluster class variables.

The cluster variables are retrieved from the management cluster, and are specific to the platform in which you are deploying the workload cluster. The required variables for each platform are listed below.

- Required cluster class variables for vSphere

- `vcenter`

- `identityRef`

- `user (sshAuthorizedKeys)`

- `vipNetworkInterface`

- `aviAPIServerHAProvider` (Set to `true` if the cluster uses NSX Advanced Load Balancer.)

- `worker`

- Required cluster class variables for AWS
 - `region`
 - `identityRef`
 - `sshKeyName`
 - `worker`
- Required cluster variables for Azure
 - `subscriptionID`
 - `location`
 - `sshPublicKey`
 - `network`
 - `clusterRole`
 - `worker`

For more information about the cluster class variables, see [Configuration File Variable Reference](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*.

7 Click **Next**.

8 Configure your network settings.

The network settings cannot be changed after the cluster is created.

- a You can optionally define an alternative CIDR for the pod and service.
- b You can optionally define a service domain for the cluster.
- c You can optionally use a local image registry.
- d You can optionally specify a proxy configuration to use for this cluster.

Note This proxy setting enables communication between your cluster and Tanzu Mission Control after the cluster is created. The proxy used during the provisioning process is defined in the `imageRepository` variable in your cluster class.

When provisioning a cluster that needs a proxy, make sure the proxy configuration object includes the non-proxied addresses in the **No proxy list**, as described in [Create a Proxy Configuration Object](#).

- e Select the CNI setting to use for the cluster, and then click **Edit setting** to configure your platform-specific network settings.
 - Antrea

For more information about Antrea settings, see [Antrea CNI Configuration](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*.
 - Calico

Skip CNI binaries allows you to not install the plugin binaries for Calico.

Veth MTU allows you to define the maximum transmission unit (MTU) for Calico. The default setting of 0 causes the MTU to be auto-detected.
 - None

To use a different cluster network interface, you can select **None** and then manually configure the CNI after the cluster is created.
 - f Click **Next**.
- 9 Configure your control plane.
- a Select the Kubernetes version and operating system to use for the cluster's control plane. The latest supported version is preselected for you.
 - b Select the type of cluster you want to create.

The primary difference between the two is that the highly available cluster is deployed with multiple control plane nodes.
 - c Specify the network resources for your control plane, based on the selected platform.
 - Network resource settings for vSphere
 - 1 You can optionally change the control plane specifications for `machine.diskGiB`, `machine.memoryMiB`, and `machine.numCPUs`.
 - 2 You can optionally add nameservers, search domains, and node labels for the control plane.
 - 3 You can optionally add labels and annotations for the control plane.
 - Network resource settings for AWS
 - 1 Select a VPC (virtual private cloud) to contain the cluster. For highly available clusters, select a VPC with at least three subnets in different availability zones.
 - 2 You can optionally provide `securityGroupOverrides`. For more information, see [Advanced Options During Cluster Creation](#).
 - Network resource settings for Azure
 - 1 Select or create control plane resources the cluster.
 - 2 You can optionally add node labels.

- d You can optionally modify the instance type and storage volume size for the control plane, and add labels and annotations for the control plane.
 - e Click **Next**.
- 10** You can optionally define the default node pool and create additional node pools for your cluster.

The node pool settings for your cluster are defined in the `worker` variable in your cluster class. You can override those settings here.

- a Specify the number of worker nodes to provision.
- b Select the class and operating system for worker nodes.
- c You can optionally specify a failure domain for the node pool.
- d To configure metadata labels for your node pool, click **Add Node Pool Label**.
- e To configure metadata annotation for your node pool, click **Add Node Pool Annotation**.
- f If you want to create another node pool, click **Add Node Pool**.
- g Click **Next**.

For more information about node pools, see [Create a Node Pool](#)

- 11** You can optionally click **Add a Variable** to provide values for additional variables to customize your cluster.

For more information about the cluster class variables, see [Configuration File Variable Reference](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*.

- 12** When you ready to provision the new cluster, click **Create Cluster**.

Results

When you click **Create Cluster**, you are directed the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for the new workload cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Provision a Cluster in vSphere with Tanzu using a Cluster Class

Use VMware Tanzu Mission Control to create a new Tanzu Kubernetes cluster using a cluster class.

Starting in version 8.0 of vSphere, you can leverage the power of `ClusterClass` to use a predefined baseline configuration to create clusters with a consistent size and shape in Tanzu Kubernetes Grid Service Supervisor Clusters.

For more information about `ClusterClass` in the Cluster API, see [Introducing ClusterClass and Managed Topologies in Cluster API](#) in the *Kubernetes Blog*.

Note The data protection features of Tanzu Mission Control are not compatible with clusters created using the Tanzu Kubernetes release `v1.23.8---vmware.2-tkg.1-zshippable`. If you rely on Tanzu Mission Control for data protection, use a different Tanzu Kubernetes release when creating your clusters.

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid Service Supervisor Cluster (vSphere version 8 or later) with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#)

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- To see and use a cloud provider account connection for creating a cluster, you must be associated with the `organization.credential.view` role.
- You must also have `admin` privileges on the management cluster to provision resources within it.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.

If the management cluster that you selected is a Supervisor Cluster running in version 8.0 (or later) of vSphere, you enter the workflow to create a cluster using a cluster class.

- 4 On the Create cluster page, select the provisioner in which you want to create the cluster, and then click **Next**.

5 Enter the name, group, and other details for the cluster.

a Enter a name for the cluster.

Cluster names must be unique within an organization.

b Select the cluster group to which you want to attach your cluster.

c You can optionally enter a description and apply labels.

d Click **Next**.

6 Select the cluster class that you want to use as the template for this cluster.

The list of cluster classes that you can choose from is taken from the provisioner namespace in your Supervisor Cluster.

7 You can optionally specify a proxy configuration to use for this cluster.

Note When provisioning a cluster in vSphere with Tanzu using a proxy, make sure the proxy configuration object includes the CIDRs for pod, ingress, and egress from the workload network of the Supervisor Cluster in the **No proxy list**, as described in [Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu](#).

8 Click **Next**.

9 Configure your control plane.

a Select the Kubernetes version and operating system to use for the cluster's control plane.

The latest supported version is preselected for you.

b Select the type of cluster you want to create.

The primary difference between the two is that the highly available cluster is deployed with multiple control plane nodes.

c Select the instance type and storage class for the control plane.

For more information about the instance types available in vSphere with Tanzu, see [Virtual Machine Class Types for Tanzu Kubernetes Clusters](#) in the *vSphere with Tanzu Configuration and Management* documentation.

d You can optionally configure additional storage volumes for your control plane.

To configure additional volumes, click **Add Volume** and then specify the name, mount path, and capacity for the volume. To add another, click **Add Volume** again.

e Click **Next**.

10 Configure your network settings.

a You can optionally define an alternative service domain.

b You can optionally define an alternative CIDR for the pod and service.

The network settings cannot be changed after the cluster is created.

- 11 You can optionally define the default node pool and create additional node pools for your cluster.
- a Specify the number of worker nodes to provision.
 - b Select the class and instance type for worker nodes.
 - c Select the storage class.
 - d To configure metadata labels for your node pool, click **Add Label**.
 - e To configure worker taints for your node pool, click **Add Taint**.
 - f To configure worker labels for your node pool, click **Add Label**.
 - g To configure additional storage volumes for your node pool, click **Add Volume**.
 - h If you want to create another node pool, click **Add Node Pool**.
 - i You can optionally click **Add Volume** to create one or more volumes that are available to all of the node pools in the cluster.
 - j Click **Next**.

For more information about node pools, see [Create a Node Pool](#)

- 12 You can optionally provide values for additional variables to customize your cluster.
- NTP server
 - user password secret key and secret name
 - user SSH authorized key
 - extension certificate name and key
 - cluster encryption config YAML
 - default registry secret key and secret name
 - default registry secret namespace
 - trust

To add a trust, click **Add Trust**, and then enter the name and data for a trust.

- 13 When you ready to provision the new cluster, click **Create Cluster**.

Results

When you click **Create Cluster**, you are directed the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for the new cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Provision a Cluster in vSphere with Tanzu

Use VMware Tanzu Mission Control to provision the necessary resources and create a new Tanzu Kubernetes cluster.

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid Service Supervisor Cluster with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#)

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- To see and use a cloud provider account connection for creating a cluster, you must be associated with the `organization.credential.view` role.
- You must also have `admin` privileges on the management cluster to provision resources within it.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.
- 4 On the Create cluster page, select the provisioner in which you want to create the cluster, and then click **Next**.
- 5 Enter the name, group, and other details for the cluster.
 - a Enter a name for the cluster.
Cluster names must be unique within an organization.
 - b Select the cluster group to which you want to attach your cluster.
 - c You can optionally enter a description and apply labels.
 - d Click **Next**.

6 Select your configuration options.

- a Select the Kubernetes version to use for the cluster.

The latest supported version is preselected for you.

- b You can optionally define an alternative CIDR for the pod and service.

The Pod CIDR and Service CIDR cannot be changed after the cluster is created.

- c You can optionally specify a proxy configuration to use for this cluster.

Note When provisioning a cluster in vSphere with Tanzu using a proxy, make sure the proxy configuration object includes the CIDRs for pod, ingress, and egress from the workload network of the Supervisor Cluster in the **No proxy list**, as described in [Create a Proxy Configuration Object for a Tanzu Kubernetes Grid Service Cluster Running in vSphere with Tanzu](#).

- d You can optionally select the default storage class for the cluster and allowed storage classes.

The list of storage classes that you can choose from is taken from your vSphere namespace.

- e Click **Next**.

7 Select the type of cluster you want to create.

The primary difference between the two is that the highly available cluster is deployed with multiple control plane nodes.

- a Choose the cluster type.

- b You can optionally select a different instance type for the cluster's control plane node and its storage class.

For more information about the instance types available in vSphere with Tanzu, see [Virtual Machine Class Types for Tanzu Kubernetes Clusters](#) in the *vSphere with Tanzu Configuration and Management* documentation.

- c You can optionally configure additional storage volumes for your control plane.

To configure additional volumes, click **Add Volume** and then specify the name, mount path, and capacity for the volume. To add another, click **Add Volume** again.

- d Click **Next**.

8 You can optionally define the default node pool and create additional node pools for your cluster.

- a Specify the number of worker nodes to provision.
- b Select the instance type for workload clusters.
- c Select the storage class.

- d To configure additional storage volumes for your node pool, click **Add Volume**.
- e To provide labels for the nodes and cloud, enter the label and then click **Add Label**.

For more information about node pools, see [Create a Node Pool](#)

- 9 When you ready to provision the new cluster, click **Create Cluster**.

Results

When you click **Create Cluster**, you are directed the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for your cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Provision a Workload Cluster in vSphere

Use VMware Tanzu Mission Control to provision the necessary resources and create a new Tanzu Kubernetes cluster in vSphere (not in vSphere with Tanzu).

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid management cluster with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#).

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- To see and use a cloud provider account connection for creating a cluster, you must be associated with the `organization.credential.view` role.
- You must also have `admin` privileges on the management cluster to provision resources within it.

If you have a proxy server that manages outbound traffic for your clusters, you need to enable the cluster to communicate with Tanzu Mission Control through the proxy.

- You can create a proxy configuration object in Tanzu Mission Control and use it when registering, provisioning, or attaching the cluster. For more information, see [Chapter 5 Connecting Through a Proxy](#).
- You can enable all outbound traffic to Tanzu Mission Control for the proxy server by adding some URLs to the proxy server's allowlist, as described in [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.
- 4 On the Create cluster page, select the provisioner in which you want to create the cluster.
- 5 Enter the name, group, and other details for the cluster.
 - a Enter a name for the cluster.

Cluster names must be unique within an organization.
 - b Select the cluster group to which you want to attach your cluster.
 - c You can optionally enter a description and apply labels.
 - d Click **Next**.
- 6 Select your configuration options.
 - a Select the data center in which you want to create the workload cluster.

The vSphere data center contains an inventory of resource objects that can be used by your cluster. The available data centers are those that exist in the vSphere instance where your management cluster is deployed.
 - b Select the Kubernetes version and operating system to use for the cluster.

The latest supported version is preselected for you.

Selecting a non-default operating system for your cluster is supported for management clusters running in Tanzu Kubernetes Grid version 1.5 or later. For more information about the available operating systems, see [Tanzu Kubernetes Releases](#) in the *VMware Tanzu Kubernetes Grid Documentation*.
 - c Enter the contents of the SSH public key created during deployment of the management cluster.
 - d Select a vSphere network to use as the Kubernetes service network.
 - e You can optionally define an alternative CIDR for the pod and service.

The Pod CIDR and Service CIDR cannot be changed after the cluster is created.
 - f You can optionally specify a proxy configuration to use for this cluster.
 - g Click **Next**.

7 Select your resource options.

The resources that are available to include in the cluster are defined by the vSphere data center that you selected in the previous step.

- a Select the resource pool to use for the workload cluster.
- b Select the VM folder in which to place the workload cluster VMs.
- c Select a vSphere datastore for the workload cluster to use.

8 Select the type of cluster you want to create.

The primary difference between the two is that the highly available cluster is deployed with multiple control plane nodes.

- ◆ You can optionally select a different instance type for the cluster's control plane node.

9 Specify a control plane endpoint for the workload cluster.

The control plane endpoint is the unique address for the control plane of the workload cluster.

If the Avi load balancer is enabled on your management cluster, this value is inherited from the management cluster and is not editable.

10 You can optionally specify an alternative port number for the API server.

If the Avi load balancer is enabled on your management cluster, this value is inherited from the management cluster and is not editable.

11 Click **Next**.**12** You can optionally define the default node pool for your cluster.

- a Select the instance type for workload clusters.
- b Specify the number of worker nodes to provision.
- c You can optionally provide labels for the nodes and cloud. Make sure you click **Add Label** after entering the key and value.

13 Depending on the kind of cluster you are creating, you can optionally specify advanced configuration settings.

For more information, see [Advanced Options During Cluster Creation](#).

14 When you ready to provision the new cluster, click **Create Cluster**.**Results**

When you click **Create Cluster**, you are directed to the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for your cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Provision a Workload Cluster in Azure

Use VMware Tanzu Mission Control to provision the necessary resources and create a new Tanzu Kubernetes cluster in Azure.

Note Version 2.5 of Tanzu Kubernetes Grid does not support deployment of clusters to Azure. To manage the lifecycle of clusters on this platform using Tanzu Mission Control, you can create a credential to connect to your account/subscription, and then deploy and manage native AKS clusters.

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid management cluster with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#).

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- You must also have `admin` privileges on the management cluster to provision resources within it.

If you have a proxy server that manages outbound traffic for your clusters, you need to enable the cluster to communicate with Tanzu Mission Control through the proxy.

- You can create a proxy configuration object in Tanzu Mission Control and use it when registering, provisioning, or attaching the cluster. For more information, see [Chapter 5 Connecting Through a Proxy](#).
- You can enable all outbound traffic to Tanzu Mission Control for the proxy server by adding some URLs to the proxy server's allowlist, as described in [What Happens When You Attach a Cluster](#) in *VMware Tanzu Mission Control Concepts*.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.

- 4 On the Create cluster page, select the provisioner and subscription in which you want to create the cluster, and then click **Next**.

A provisioner is a namespace in your management cluster. A subscription is a logical container in your Azure account that holds your resources.

- 5 Specify the name, group, and other details for the cluster.

- a Enter a name for the cluster.

Cluster names must be unique within an organization.

- b Select the cluster group to which you want to attach your cluster.

- c You can optionally enter a description and apply labels.

Labels that you add here are used as cloud labels to tag resources in Azure.

- d Click **Next**.

- 6 Select your configuration options.

- a Select the region and enter a name for the new resource group in which to create the workload cluster.

- b Select the Kubernetes version and operating system to use for the cluster.

The latest supported version is preselected for you.

Selecting a non-default operating system for your cluster is supported for management clusters running in Tanzu Kubernetes Grid version 1.5 or later. For more information about the available operating systems, see [Tanzu Kubernetes Releases](#) in the *VMware Tanzu Kubernetes Grid Documentation*.

- c Enter the contents of your SSH public key.

- d You can optionally click the **Private** toggle to provision a private cluster in Azure.

If you choose to create a private workload cluster, you must choose an existing VNET in the next step. The VNET must be either your management cluster VNET or a VNET peered with your management cluster VNET. If you specify an API server private IP address, it must be within the CIDR range of the control plane subnet.

- e Select or create a VNET (Azure virtual network) to use as the Kubernetes service network.

If you select an existing VNET, make sure it already has the following resources:

- 1 control plane subnet - that uses a network security group (NSG) that allows inbound connection on port 6443 which is used by kubeadm to bootstrap the control planes
- 1 node subnet - that has a default NSG attached
- a route table - that is associated with the node subnet

- f You can optionally define an alternative CIDR for the pod and service.

The Pod CIDR and Service CIDR cannot be change after the cluster is created.

g You can optionally specify a proxy configuration to use for this cluster.

If you are creating a private cluster, make sure `.capz.io` is in the `NoProxyList` for your proxy.

h Click **Next**.

7 Select the type of cluster you want to create, and the control plane instance type.

The primary difference between the two is that the highly available cluster is deployed with three control plane nodes.

8 You can optionally specify an alternative port number for the API server.

9 You can optionally define the default node pool for your cluster.

a Select the instance type for workload clusters.

b Specify the number of worker nodes to provision.

Highly available Tanzu Kubernetes clusters in Azure are created with three node pools with one worker node each by default. If you want more node pools for your cluster, you can add them later. For more information about node pools, see [Create a Node Pool](#).

10 When you ready to provision the new cluster, click **Create Cluster**.

Results

When you click **Create Cluster**, you are directed the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for your cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Provision a Workload Cluster in AWS

Use VMware Tanzu Mission Control to provision the necessary resources and create a new Tanzu Kubernetes cluster in your Amazon Web Services account.

Note Version 2.5 of Tanzu Kubernetes Grid does not support deployment of clusters to AWS. To manage the lifecycle of clusters on this platform using Tanzu Mission Control, you can create a credential to connect to your account, and then deploy and manage native EKS clusters.

Prerequisites

Before you can create new clusters using Tanzu Mission Control, you must first establish a connection with your management cluster.

- 1 Register your Tanzu Kubernetes Grid management cluster with Tanzu Mission Control, as described in [Register a Management Cluster with Tanzu Mission Control](#).
- 2 Create a provisioner namespace into which you will provision the cluster, as described in [Create a Provisioner in Your Tanzu Kubernetes Grid Management Cluster](#).

Make sure you have the appropriate permissions to create a Tanzu Kubernetes cluster.

- To provision a cluster, you must be associated with the `clustergroup.edit` role on the cluster group in which you want to put the new cluster.
- To see and use a cloud provider account connection for creating a cluster, you must be associated with the `organization.credential.view` role.
- You must also have `admin` privileges on the management cluster to provision resources within it.
- Creating a new Tanzu Kubernetes cluster in your AWS account requires additional permissions in your AWS account. For more information, see [Required AWS Permissions](#) in the *VMware Tanzu Kubernetes Grid Product Documentation*. See also the VMware knowledge base article at <http://kb.vmware.com/kb/87547>.

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Clusters**.
- 2 On the Clusters page, click **Add Cluster**, and then choose **Create Tanzu Kubernetes Grid cluster** from the dropdown.
- 3 Click to select the management cluster in which to create the new workload cluster, and then click **Continue to Create Cluster**.
- 4 On the Create cluster page, select the provisioner namespace in which you want to create the cluster, and then click **Next**.
- 5 Specify the name, group, and other details for the cluster.
 - a Enter a name for the cluster.
Cluster names must be unique within an organization.
 - b Select the cluster group to which you want to attach your cluster.
 - c You can optionally enter a description and apply labels.
 - d Click **Next**.
- 6 Select your configuration options.
 - a Select the region in which you want to create the workload cluster.
 - b Select the SSH public key created during deployment of the management cluster.

- c Select the Kubernetes version and operating system to use for the cluster.

The latest supported version is preselected for you.

Selecting a non-default operating system for your cluster is supported for management clusters running in Tanzu Kubernetes Grid version 1.5 or later. For more information about the available operating systems, see [Tanzu Kubernetes Releases](#) in the *VMware Tanzu Kubernetes Grid Documentation*.

- d Select or create a VPC (virtual private cloud) to contain the cluster.

If you create a new VPC, specify the CIDR for the VPC.

- e You can optionally define an alternative CIDR for the pod and service.

The Pod CIDR and Service CIDR cannot be changed after the cluster is created.

- f Click **Next**.

- 7 Select the type of cluster you want to create, and the control plane instance type.

The primary difference between the two is that the highly available cluster is deployed with multiple control plane nodes, in multiple availability zones across the region, with one public subnet and one private subnet for each availability zone.

- a Select the instance type to use for the control plane node.
- b Select the availability zone for the cluster.
- c Specify the public and private CIDRs for the availability zone.

- 8 Define the default node pool for your cluster.

- a Select the instance type for worker nodes.
- b Specify the number of worker nodes to provision.

- 9 Depending on the kind of cluster you are creating, you can optionally specify advanced configuration settings.

For more information, see [Advanced Options During Cluster Creation](#).

- 10 When you are ready to provision the new cluster, click **Create Cluster**.

Results

When you click **Create Cluster**, you are directed to the cluster detail page where you can see its status is `Unknown` while it is being created. Tanzu Mission Control provisions the resources necessary for your cluster in your management cluster. It then creates the workload cluster and attaches it to your organization in the cluster group that you specified. This process takes a few minutes.

Advanced Options During Cluster Creation

When provisioning a new cluster using VMware Tanzu Mission Control, you can provide additional options using the **Advanced settings** selector, such as security groups for Tanzu

Kubernetes clusters running in AWS EC2, and Avi configuration for Tanzu Kubernetes clusters running in vSphere.

Depending on the platform that your management cluster is running on, you might have additional configuration options available.

Security Groups for Tanzu Kubernetes clusters running in AWS EC2

This feature is available for Tanzu Kubernetes Grid clusters version 1.5 or later running in Amazon Web Services.

When you create a new workload cluster in a Tanzu Kubernetes Grid management cluster running in AWS EC2, you can specify values for the following variables to use your own security group, rather than using the default security group.

- `AWS_SECURITY_GROUP_BASTION` is the ID of a user-created security group used to control in-bound access to the bastion.
- `AWS_SECURITY_GROUP_CONTROLPLANE` is the ID of a user-created security group used to control in-bound access to the control plane nodes. This group must allow access to port 6443.
- `AWS_SECURITY_GROUP_NODE` is the ID of a user-created security group used to control in-bound access to all nodes.
- `AWS_SECURITY_GROUP_LB` is the ID of a user-created security group used by the Kubernetes AWS Cloud Provider for setting rules for elastic load balancers (ELBs).
- `AWS_SECURITY_GROUP_APISERVER_LB` is the ID of a user-created security group used for Kubernetes API Server ELB, and controls inbound access to the control plane endpoint.

Avi Kubernetes Operator Configuration for Clusters Running in vSphere

This feature is available for Tanzu Kubernetes Grid clusters version 1.6 or later running in vSphere.

When you create a new workload cluster in a Tanzu Kubernetes Grid management cluster running in vSphere, you can specify one or more label selectors to identify the Avi Kubernetes Operator (AKO) configuration that you want to use for the new cluster.

If you have a custom AKO configuration (`AKODeploymentConfig`, or ADC) defined in your management cluster, you can specify one or more values for `AVI_LABELS` that identify the ADC that you want to use in the new cluster.

Label selectors must be JSON-formatted, for example `{"mykey": "myvalue"}`. You can add multiple, comma-separated label selectors. If you add multiple label selectors, the ADC must have all specified labels to match. For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

Create a Node Pool

Create a node pool for your Kubernetes cluster provisioned through VMware Tanzu Mission Control.

For clusters that you create in Tanzu Mission Control, you can define a pool of worker nodes on which your workloads can run. Because Tanzu Mission Control cannot provision additional resources in a cluster that is created elsewhere and subsequently attached, you cannot create a node pool in an attached cluster.

This procedure describes how to create a node pool for an existing provisioned cluster. You can also define a node pool when creating a cluster, and the steps are essentially the same.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To define a node pool, you must be associated with the `cluster.edit` role for the cluster.
- To create a cluster, you must also be associated with the `clustergroup.edit` role for the cluster group.

For more information about adding permissions to custom roles, see [Create a Custom Access Role](#).

Procedure

- 1 Navigate to the cluster in which you want to create a node pool.
- 2 On the cluster detail page, click the **Node pools** tab, and then click **New node pool**.
- 3 Provide a name for the node pool.
The name must be unique within the cluster.
- 4 Specify the number of nodes to create for this node pool.
- 5 Select the instance type for the worker nodes.
- 6 For Tanzu Kubernetes clusters running in vSphere, you can specify the storage class for the worker nodes.
- 7 Select the availability zone in which to create the nodes or, when running on vSphere 8.0, select the failure domain.

In vSphere 8.0, vSphere with Tanzu allows you to provision guest cluster worker nodes across multiple failure domains (vSphere zones) for clusters deployed in high availability mode.

You can select the failure domain from the dropdown list, which lists the domains that are retrieved from the Supervisor Cluster.

Note

- The failure domain is required in highly available clusters, and optional in single node clusters.
 - The failure domain can not be modified after it the node pool is configured.
-

- 8 You can optionally create additional volumes for the node pool.
 - a To create a volume, click **Add Volume**.
 - b Specify the name, mount path, and capacity for the volume.
 - c To create additional volumes, click **Add Volume** again.
 - d If necessary, you can click the delete icon to remove a volume that you have added.

Make sure your volume definitions are correct before proceeding. You cannot edit the configured volumes after creating the node pool.

- 9 You can optionally provide a description, and labels for the node and cloud.
 - The node labels that you add here are applied to the Kubernetes worker nodes in this node pool. To see the node labels in your cluster, use the following command:

```
kubectl get nodes --show-labels
```

- The cloud labels that you add here are applied to resources that are provisioned for this node pool in your cloud provider account (for example, AWS tags on EC2 instances).

- 10 Click **Save**.

Results

When you click **Save**, Tanzu Mission Control provisions the necessary resources and then launches the requested nodes.

Edit a Node Pool

Resize a provisioned cluster by changing size of the node pool.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To edit a node pool, you must be associated with the `cluster.edit` role for the cluster.

For more information about adding permissions to custom roles, see [Create a Custom Access Role](#).

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster whose node pool you want to edit.
- 2 On the cluster detail page, click the **Node pools** tab.
- 3 Click to expand the node pool you want to edit.
- 4 Click **Edit**.

5 Make the desired changes.

You can modify the following settings in your node pool.

- number of worker nodes
- instance type
- storage class (for Tanzu Kubernetes clusters running in vSphere)

However, you cannot modify the other aspects of the node pool after it is created.

6 Click **Save**.

Results

If you decrease the number of nodes in the node pool, Tanzu Mission Control removes the additional nodes and allows the cluster to redistribute the workloads running on the nodes as necessary.

If you increase the number of nodes in the node pool, Tanzu Mission Control provisions the necessary resources and then launches the additional nodes.

Delete a Node Pool

Delete a node pool from a provisioned cluster.

Note Highly available Tanzu Kubernetes clusters in Azure are created with three node pools with one worker node each by default. You can add and delete node pools in this type of cluster, but three node pools must remain in the cluster.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To delete a node pool, you must be associated with the `cluster.edit` role for the cluster.

For more information about adding permissions to custom roles, see [Create a Custom Access Role](#).

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster whose node pool you want to delete.
- 2 On the cluster detail page, click the **Node pools** tab.
- 3 Click to expand the node pool you want to delete.
- 4 Click **Delete**.
- 5 In the confirmation dialog, click **Confirm**.

Results

When you delete a node pool, Tanzu Mission Control removes the nodes and allows the cluster to redistribute the workloads running on the nodes as necessary. If you delete your only node pool, your cluster is left with no worker nodes and its health status shows as `Disconnected`. You can add a new node pool to reconnect the cluster.

Relaxing Pod Security in a Provisioned Cluster

Apply a less restrictive pod security policy to a provisioned cluster to allow for privileged pods.

To keep your provisioned clusters secure by default, Tanzu Mission Control applies a restrictive pod security policy that prevents the use of privileged options in your containers, such as running a container as root or using privileged mode. However, you might want to enable your pods to use some of these privileged options in some of your clusters. So, Tanzu Mission Control provides a preconfigured cluster role (`vmware-system-tmc-ppsp-privileged`) that you can use to apply a less restrictive pod security policy for specified identities.

Prerequisites

Before you begin this procedure, you must have a cluster that you have provisioned through Tanzu Mission Control.

Open a command window and connect to your cluster with `kubectl`.

Make sure you have the appropriate permissions.

- To modify role bindings in the cluster, you must be associated with the `cluster.admin` role.

Procedure

- 1 To enable privileged mode for a specific pod, bind the privileged role to a service account.

Use the following `kubectl` command, replacing `my-namespace` and `my-service-account` with the appropriate values.

```
kubectl create rolebinding my-privileged-role-binding \
--clusterrole=vmware-system-tmc-ppsp-privileged \
--user=system:serviceaccount:my-namespace:my-service-account \
-n my-namespace
```

- 2 To enable privileged mode for an entire cluster, bind the privileged role to the group containing all authenticated users.

```
kubectl create clusterrolebinding my-privileged-cluster-role-binding \
--clusterrole=vmware-system-tmc-ppsp-privileged \
--group=system:authenticated
```

Upgrade Kubernetes on Your Cluster

Upgrade the version of Kubernetes that is running on a cluster that you manage through VMware Tanzu Mission Control.

When you upgrade your cluster, Tanzu Mission Control performs an in-place upgrade where each node is replaced and then deleted, but the cluster is not deleted. Tanzu Mission Control also upgrades the version of your cluster's CoreDNS and kube-proxy.

For AKS clusters, Tanzu Mission Control uses Azure tooling and the AKS API to perform the upgrade, starting with the control plane nodes before proceeding to the worker nodes. For more information, see <https://learn.microsoft.com/en-us/azure/aks/upgrade-cluster?tabs=azure-cli#upgrade-an-aks-cluster>.

For EKS clusters, Tanzu Mission Control uses AWS tooling to perform the upgrade. For more information, see [UpdateClusterVersion](#) in the *Amazon EKS API Reference*.

For Tanzu Kubernetes clusters, Tanzu Mission Control uses Cluster API to upgrade each node sequentially, starting with the control plane nodes before proceeding to the worker nodes. Be aware of the following constraints when upgrading your Tanzu Kubernetes cluster.

- You can upgrade a cluster up to one minor version of Kubernetes from its current version. If necessary, you can perform subsequent upgrades to move the version forward.
- Upgrading your version of Kubernetes is a one-way operation. You cannot subsequently downgrade the Kubernetes version, or undo an upgrade.
- To upgrade a workload cluster, the cluster must be using Antrea CNI for its container network interface. For more information, see [Requirements for Registering a Tanzu Kubernetes Cluster with Tanzu Mission Control](#).
- When you upgrade a class-based workload cluster (TKG 2.3 and later), the cluster is also updated to use the latest default cluster class from its management cluster.

Prerequisites

Before you begin this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To upgrade a cluster, you must be associated with the `cluster.admin` role on the cluster.

The option to upgrade is available only if a valid upgrade is available for the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.
- 2 Click the cluster group that contains the managed cluster you want to upgrade, and then click the cluster.
- 3 On the cluster detail page, click **Upgrade** in the top right corner.
- 4 In the confirmation dialog, select the version to which you want to upgrade and then click **Upgrade**.

Results

When you click **Upgrade**, Tanzu Mission Control sets the status of the cluster to `Upgrading`, and temporarily suspends cluster lifecycle management operations until the upgrade is complete.

If you return to the detail page of the cluster or cluster group while the upgrade is in progress, you see that the cluster is still listed with its status as `Upgrading`.

Delete a Provisioned Cluster

Remove a cluster that you created through VMware Tanzu Mission Control.

Prerequisites

Before you begin this procedure, log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To delete a provisioned cluster you must be associated with the `clustergroup.edit` role on the cluster group to which the cluster belongs.
- You must also be associated with the `cluster.admin` role on the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.
- 2 Click the cluster group that contains the provisioned cluster you want to delete, and then click the cluster.
- 3 On the cluster detail page, click **Actions** in the top right corner, and then choose **Delete** from the dropdown.
- 4 In the confirmation dialog, click **Delete**.

Results

When you confirm deletion of the cluster, Tanzu Mission Control stops all processes running on every node in the cluster, then proceeds to remove all resources that were provisioned for the cluster, and finally removes the cluster from the cluster agent registry. This process takes a little time. If you return to the detail page of the cluster or cluster group while the delete is in progress, you see that the cluster is still listed with its status as `Deleting`.

Manage Certificates

Clusters must be upgraded at least once a year in order for the certificates to be rotated before they expire.

Tanzu Mission Control does not rotate the certificates created during cluster creation by default. These certificates are valid for one year while the CAs are valid for 10 years. The certificates get regenerated for one year whenever the nodes are rolled. In the case of control plane nodes are rolled during cluster upgrade.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to update a cluster.

- To update a cluster, you must be associated with the `cluster.admin` role.

Procedure

- 1 In the left navigation pane, click **Clusters**.
- 2 Select the cluster to upgrade.
- 3 Click **Upgrade**.

Results

The certificate is renewed.

Managing a Local Image Registry

11

Use of a local image registry requires different considerations and configurations than when using the standard Tanzu Mission Control SaaS registry.

Toggle Local Image Registry

You can toggle support for local image registry and add local image registries when attaching a cluster to Tanzu Mission Control.

Supporting Self-Signed Registries on Tanzu Kubernetes Grid Clusters

For image registries using self-signed certificates, the cluster should be trusting the registry's certificate. A Tanzu Kubernetes Grid cluster can be provisioned by passing the configuration `TKG_CUSTOM_IMAGE_REPOSITORY_CA_CERTIFICATE` with the registry's certificate, which then gets injected into the cluster. Tanzu Kubernetes Grid limits management clusters and workload clusters to using the same registry for pulling Tanzu Kubernetes Grid related images.

Due to the known issue above in Tanzu Kubernetes Grid, to support the local image registry with CA certs scenarios in Tanzu Kubernetes Grid 1.6.1, you must make sure Tanzu Kubernetes Grid is pulling its own images from the same registry as the Tanzu Mission Control local image registry defined, not from the default registry.

You can bring up a management cluster using the local image registry by following the instructions given in [Prepare an Internet-Restricted Environment](#).

For information on how to pull and push images from a public registry to the local registry, see [Copy Images into an Airgapped Environment](#).

Register the management cluster as described in [Complete the Registration of a Management Cluster](#).

Read the following topics next:

- [Add a Local Image Registry for Tanzu Mission Control](#)
- [Attach a Kubernetes Cluster with a Local Image Registry](#)
- [Configuring Inspections for use with a Local Image Registry](#)

- [Delete a Local Image Registry](#)
- [Sync Images to Your Local Image Registry](#)

Add a Local Image Registry for Tanzu Mission Control

You can add a local image registry using the Tanzu Mission Control console.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to add a local image registry.

- To add a local image registry you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click Administration in the left navigation pane.
- 2 Click the **Local image registries** tab.
- 3 Click **Add Local Image Repository**.
- 4 Provide a name for the image registry, and optionally a description.
- 5 Enter the URL for the image registry, without the HTTP prefix.
- 6 Enter credentials for the image registry. For an unauthenticated registry, skip to the next step. To create an Authenticated registry, fill in the Access ID and Access Secret fields.

Results

After adding the local registry, you are redirected to the **Image registries** tab and your new registry appears in the table.

Add a Local Image Registry for Tanzu Mission Control Using the CLI

You can add a local image registry to Tanzu Mission Control using the command line interface. It can be added in three different ways: (1) credentials in the command line arguments, (2) credentials with Docker config file, and (3) encoded credentials in dockerconfigjson format

Prerequisites

Make sure you have the appropriate permissions to add local image registries.

- To add local image registries you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Note You can use the TMC CLI commands as shown below, or you can use the Tanzu CLI command `tanzu mission-control account credential create --input-template image-registry --data-values-file <data_file>`.

Procedure

- 1 Use the following command to create the local image registry with credentials:

```
tmc account credential create --template image-registry --access-id demouser
--access-secret demopassword --name test-registry --registry-url myregistry.io
--registry-namespace test-ns
```

- 2 Use the following command to create the local image registry with credentials and CA cert:

```
tmc account credential create --template image-registry --access-id demouser
--access-secret demopassword --name test-registry --registry-url myregistry.io
--registry-namespace testns --ca-cert="$(cat /tmp/ca.crt)"
```

- 3 Use the following command to create the local image registry with the credentials in dockerconfigjson format:

```
tmc account credential create --template image-registry --name test-registry
--registry-url myregistry.io --registry-namespace testns --dockerconfig-file
dockerconfig.json
```

The `dockerconfig.json` can contain values like below:

```
{
  "auths": {
    "myregistry.io": {
      "auth": "ZGVtb3VzZXI6ZGVtb3Bhc3N3b3Jk",
      "password": "demopassword",
      "username": "demouser"
    }
  }
}
```

- 4 Use the following command to create the unauthenticated local image registry:

```
tmc account credential create --template image-registry --name test-registry
--registry-url myregistry.io --registry-namespace testns
```

- 5 Use the following command to list image-registry credentials:

```
tmc account credential list --capability IMAGE_REGISTRY
```

- 6 Use the following to get specific image-registry credentials:

```
tmc account credential get test-registry
```

Add a Local Image Registry for Tanzu Mission Control Using the API

You can create a local image registry with the Tanzu Mission Control [CredentialResource](#) API.

Prerequisites

Make sure you have the appropriate permissions to add local image registries.

- To add local image registries you must be associated with the Tanzu Mission Control role `cluster.admin` role.

The following are the input parameters required for creating the local image registry:

- Registry configuration
 - The registry configuration `RegistryURL`, `AccessID` and `AccessSecret` is encoded and stored as `dockerconfigjson` in the resource spec.
- Registry Namespace

Procedure

- 1 The following is an example payload to create an unauthenticated local image registry:

```
{
  "credential": {
    "fullName": {
      "name": "test-registry"
    },
    "meta": {
      "annotations": {
        "registry-namespace": "test-ns"
      }
    },
    "spec": {
      "capability": "IMAGE_REGISTRY",
      "data": {
        "keyValue": {
          "data": {
            "registry-url": "aHR0cDovL215cmVnaXN0cnkuaW8=",
            "ca-cert":
"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURFekNDQWZlZ0F3SUJBZ01RT3l2YmhoaUFiVDhYRG
ZVSjJ2aWp5VEFOQmdrcWhraUc5dzBCQVF....."
          }
        }
      },
      "meta": {
        "provider": "GENERIC_KEY_VALUE"
      }
    },
    "type": {
      "kind": "Credential",
      "package": "vmware.tanzu.manage.v1alpha1.account.credential",
      "version": "v1alpha1"
    }
  }
}
```

2 The following is an example payload to create an authenticated local image registry:

```
{
  "credential": {
    "fullName": {
      "name": "test-registry"
    },
    "meta": {
      "annotations": {
        "registry-namespace": "test-ns"
      }
    },
    "spec": {
      "capability": "IMAGE_REGISTRY",
      "data": {
        "keyValue": {
          "data": {
            ".dockerconfigjson":
"eyJhdXRocyI6eyJodHRwOi8vbXlyZWdpc3RyeS5pbyI6eyJlc2VybmFtZSI6ImRlbW91c2VyIiwicGFzc3
dvcmQiOiJkZWlvcGFzc3dvcmQiLCJhdXRoIjoiWkdWdGIZVnpaWEk2WkdWdGIZQmhjM04zYjNKayJ9fX0="
, // base64 encode value of the dockerconfigjson structb value or converting its
json output to []bytes
            "ca-cert":
"LS0tLS1CRUdJTjBDRVJUSUZJQ0FURSU0tLS0tCk1JSURFekNDQWZlZ0F3SUJBZ01RT3l2YmhoaUFiVDhYRG
ZVSjJ2aWp5VEFOQmdrcWhraUc5dzBCQVF....."
          },
          "type": "DOCKERCONFIGJSON_SECRET_TYPE"
        }
      },
      "meta": {
        "provider": "GENERIC_KEY_VALUE"
      }
    },
    "type": {
      "kind": "Credential",
      "package": "vmware.tanzu.manage.v1alpha1.account.credential",

```

```

    "version": "v1alpha1"
  }
}

```

Note Note that the `dockerconfigjson` contains the registry configuration in the following format:

```

{
  "auths": {
    "myregistry.io": {
      "auth": "ZGVtb3VzZXI6ZGVtb3Bhc3N3b3Jk",
      "password": "demopassword",
      "username": "demouser"
    }
  }
}

```

Attach a Kubernetes Cluster with a Local Image Registry

You can attach existing Kubernetes clusters to Tanzu Mission Control with a local image registry.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to attach a cluster using a local image registry.

- To attach a cluster using a local image registry you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- 1 Attach the cluster as described in [Attach a Cluster](#).
- 2 In step 6 of the instructions, in addition to selecting a proxy, you can toggle the **Local Image Registry Setting** to **Yes** to enable it for the cluster and then specify the local image registry in the **Set a local image registry for this cluster** field or click **Add New Local Image Registry** to add a new registry.

Results

The cluster is attached and using the specified local image registry.

Attach a Kubernetes Cluster with a Local Image Registry Using the CLI

You can attach existing Kubernetes clusters to Tanzu Mission Control with a local image registry using the `tmc` CLI.

Prerequisites

Make sure you have the appropriate permissions to add a cluster using the local image registry with the CLI.

- To add a cluster using the local image registry you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- ◆ The local image registry name must be passed to the CLI as shown below:

```
tmc cluster create --template attached --cluster-group default --name demo-cluster
--image-registry test-registry
tmc cluster attach --name demo-cluster --continue-bootstrap --kubeconfig
/path/to/kubeconfig
```

Or use the following:

```
tmc cluster attach --name=demo-cluster -m=attached -p=attached
--image-registry=test-registry --kubeconfig=/path/to/kubeconfig
```

Attach a Kubernetes Cluster with a Local Image Registry Using the API

You can attach existing Kubernetes clusters to Tanzu Mission Control with a local image registry using the API.

Prerequisites

Make sure you have the appropriate permissions to add a cluster using the local image registry.

- To add a local image registry you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- ◆ Use the API for creating clusters in Tanzu Mission Control, [ClusterResourceService](#), to attach an existing Kubernetes cluster using a local image registry, as shown in the following example:

```
{
  "fullName": {
    "managementClusterName": "attached",
    "name": "demo-cluster",
    "provisionerName": "attached"
  },
  "meta": {
    "description": "An attached cluster."
  },
  "spec": {
    "clusterGroupName": "default",
    "imageRegistry": "test-registry"
  }
}
```

```

    },
    "type": {
      "kind": "Cluster",
      "package": "vmware.tanzu.manage.v1alpha1.cluster",
      "version": "v1alpha1"
    }
  }
}

```

Configuring Inspections for use with a Local Image Registry

You can configure inspections for use with a local image registry in Tanzu Mission Control.

Tanzu Mission Control supports three types of inspection scans with local image registry:

- CIS Scan
- Lite Scan
- Conformance Scan

Instructions for running the Conformance and Lite scans are given in the steps below. You can directly run the CIS scan using the CLI as shown here:

```

tmc cluster inspection scan create --cluster-name <cluster-name> --inspection-type
CIS --management-cluster-name <mc-name> --provisioner-name <provisioner-name>

```

The Lite and Conformance scans use third-party images to run the scans. These third-party images are not included as part of the Tanzu Mission Control distributed images. You must copy these images to run these scans. The list of the images needed for Kubernetes clusters versions 1.21 - 1.23 is included **NEED EXTERNAL REFERENCE TO LIST**.

There are two ways to copy these images to the registry: (1) manually push the images to the registry, or (2) install Sonabuoy and use it to push the images.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to configure inspections.

- To configure inspections you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- 1 Use the image list here to manually tag and push the images to the custom registry. For example:

```

docker pull k8s.gcr.io/e2e-test-images/agnhost:2.36
docker tag k8s.gcr.io/e2e-test-images/agnhost:2.36
<customer-registry>/extensions/inspection-images/agnhost:2.36
docker push <customer-registry>/extensions/inspection-images/agnhost:2.36

```

- 2 Install Sonobuoy and let it take care of the image pushing. Follow the instructions at XREF to install Sonobuoy. Then use the following commands to push the images to the repository for all three versions:

```
sonobuoy images push --kubernetes-version v1.23.3
--e2e-repo <customer-registry>/extensions/inspection-images --custom-registry
<customer-registry>/extensions/inspection-images

sonobuoy images push --kubernetes-version v1.22.3
--e2e-repo <customer-registry>/extensions/inspection-images --custom-registry
<customer-registry>/extensions/inspection-images

sonobuoy images push --kubernetes-version v1.21.3
--e2e-repo <customer-registry>/tmc-unstable/extensions/inspection-images
--custom-registry <customer-registry>/extensions/inspection-images
```

- a In addition to the Sonobuoy commands, manually push the below image as it is not pushed as part of the Sonobuoy list:

```
docker pull k8s.gcr.io/e2e-test-images/agnhost:2.31
docker tag k8s.gcr.io/e2e-test-images/agnhost:2.31
<customer-registry>/extensions/inspection-images/agnhost:2.31
docker push <customer-registry>/extensions/inspection-images/agnhost:2.31
```

More details on using these commands can be found XREF <https://sonobuoy.io/docs/v0.56.13/airgap/#test-images>.

Note You might encounter some errors while pushing some images as they require authentication. This is expected and should be ignored. Those images are not used as part of Conformance. Refer to the Sonobuoy <https://sonobuoy.io/docs/v0.56.13/airgap/#test-images> Notes section for more details.

- 3 Once the images are in place, you can run the Lite and Conformance scans.

```
tmc cluster inspection scan create --cluster-name <cluster-name> --inspection-type
LITE --management-cluster-name <mc-name> --provisioner-name <provisioner-name>

tmc cluster inspection scan create --cluster-name <cluster-name> --inspection-type
CONFORMANCE --management-cluster-name <mc-name> --provisioner-name
<provisioner-name>
```

Note The inspection type must be in all capitals as shown above.

Results

Delete a Local Image Registry

You can delete Local Image Registries from Tanzu Mission Control when they are no longer required for your environment.

Prerequisites

Make sure you have the appropriate permissions to add local image registries.

- To delete local image registries you must be associated with the Tanzu Mission Control role `cluster.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 Click the **Local image registries** tab.
- 3 In the table of registries, click the menu icon for the registry you want to delete, and then choose **Delete**.

If the registry is being used in association with a cluster, you will get a message notifying you that you cannot delete it until you remove those clusters from this registry. Click Close, remove the associated clusters, and then repeat this procedure.

- 4 In the confirmation dialog, click **Delete**.

Sync Images to Your Local Image Registry

Synchronize the images in your local image registry with their external source.

To avoid having stale images in your local repository, sync images on a daily basis.

Tanzu Mission Control publishes the list of artifacts without registry address. To pull images from Tanzu Mission Control, use the registry address that has been shared for your organization. Note that mirrored images can have a different base path, but the repository name and the image's `SHA256` checksum should be the same.

Prerequisites

Log in with the `tanzu` CLI.

Make sure you have the appropriate permissions to sync images.

- To sync images you must be associated with the `cluster.admin` role in TMC.

Procedure

- 1 List the images using the `tanzu` CLI.

```
tanzu mission-control agentartifacts list
```

2 Copy the Tanzu Mission Control agent and dependent images to the local registry.

a Using Docker:

```
> docker pull ${TMC_REGISTRY}/extensions/agent-updater/agent-
updater@sha256:1a95482a28666fa859a5d09c7a067289d6dff194ee42ab27217fcfa87f15180c
> docker tag ${TMC_REGISTRY}/extensions/agent-updater/agent-
updater@sha256:1a95482a28666fa859a5d09c7a067289d6dff194ee42ab27217fcfa87f15180c $
{LOCAL_REGISTRY}/extensions/agent-updater/agent-updater
> docker push ${LOCAL_REGISTRY}/extensions/agent-updater/agent-updater latest
```

b Using Skopeo:

```
> skopeo copy \
docker://${TMC_REGISTRY}/extensions/agent-updater/agent-
updater@sha256:1a95482a28666fa859a5d09c7a067289d6dff194ee42ab27217fcfa87f15180c \
docker://${LOCAL_REGISTRY}/extensions/agent-updater/agent-
updater@sha256:1a95482a28666fa859a5d09c7a067289d6dff194ee42ab27217fcfa87f15180c
```

Here is a sample bash script to copy the artifacts listed in the file to local registry using Skopeo:

```
#!/bin/bash
IMAGE_LIST_FILE=images
TMC_REGISTRY=""
LOCAL_REGISTRY=""
LOCAL_REGISTRY_CREDS=""
while IFS= read -r line
do
echo "Copying $line"
eval "skopeo copy \
--dest-tls-verify=false \
--dest-creds $LOCAL_REGISTRY_CREDS \
docker://$TMC_REGISTRY/$line \
docker://$LOCAL_REGISTRY/$line"
done < "$IMAGE_LIST_FILE"
```

3 Copy the Tanzu Standard package repository.

For the Carvel catalog feature, the Tanzu Standard package repository image must be copied to the local registry. You can use the `imgpkg` tool from Carvel to copy the image and its dependencies. The Tanzu Standard package repository image is available here:

```
${TMC_IMAGE_REGISTRY}/packages/standard/repo:v2.2.0_update.1
```

For example

```
imgpkg copy -b ${TMC_REGISTRY}/packages/standard/repo:v2.2.0-update.1 --to-repo $
{LOCAL_REGISTRY}/packages/standard/repo
```

If your local image registry needs authentication, you can provide that using the following variables:

- IMGPKG_REGISTRY_HOSTNAME_0
- IMGPKG_REGISTRY_USERNAME_0
- IMGPKG_REGISTRY_PASSWORD_0

Managing Cluster Resources with Continuous Delivery

12

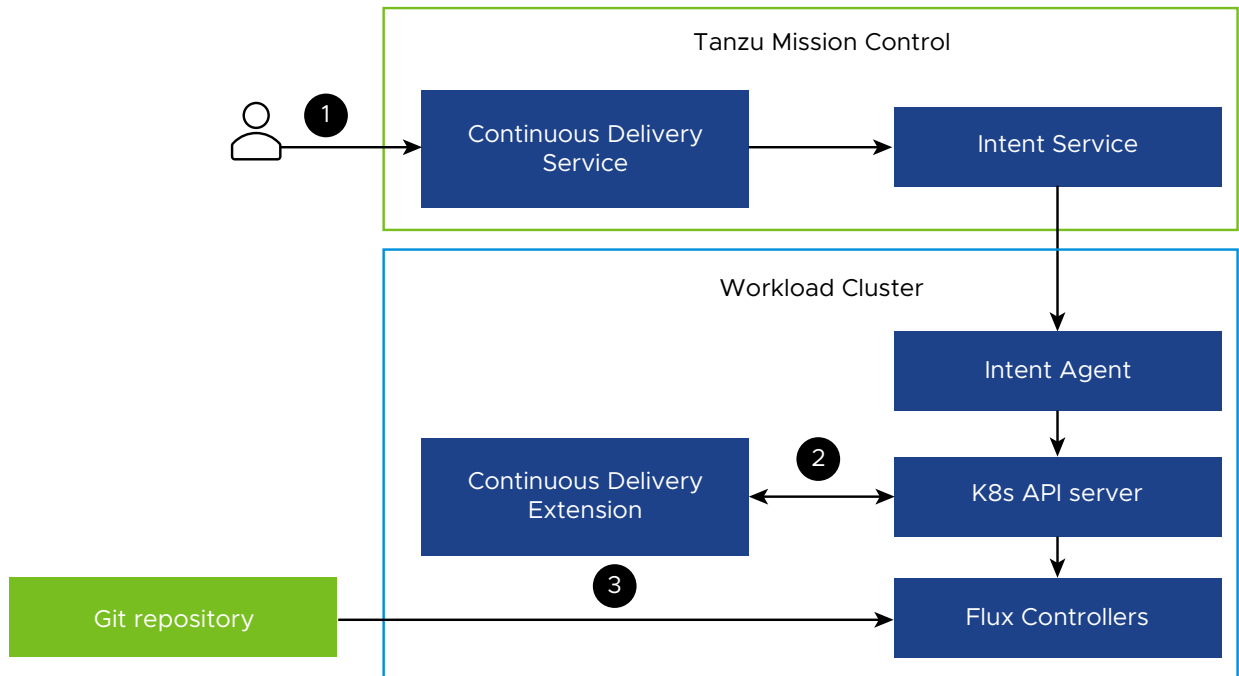
Use Tanzu Mission Control to connect your Kubernetes clusters and cluster groups to a Git repository, and then manage the resources declaratively from the repository.

As a cluster administrator, you can use Tanzu Mission Control to set up continuous delivery for your clusters and cluster groups. You can define the configuration of your cluster or cluster group (as well as other resources like Helm packages) declaratively using YAML in a Git repository, connect your cluster or cluster group to the repository, and then sync the repository. After you set up continuous delivery for a cluster or cluster group, Tanzu Mission Control drives the continuous delivery of repository objects to the cluster(s). You can enable continuous delivery with or without authentication, depending on your requirements.

Tanzu Mission Control uses Flux (an open source community standard) for continuous delivery. Flux uses Kustomize to sync YAML to your cluster. Although it is commonly used to apply overlay YAML to existing resources, Kustomize can also be used to create and manage new resources. Flux CD runs in your cluster, connects to your repositories, and periodically syncs your defined kustomization files to your cluster.

The following diagram depicts the basic flow for configuring continuous delivery through Tanzu Mission Control.

- 1 The cluster administrator uses Tanzu Mission Control to configure the Git repository and kustomization.
- 2 The continuous delivery extension in the cluster creates CRDs for the Git repository and kustomization.
- 3 The Flux controllers sync the Git repository to the cluster.



For more information about continuous delivery using Flux, visit <https://fluxcd.io/docs/>.

For more information about Kustomize, see [Declarative Management of Kubernetes Objects Using Kustomize](#) in the Kubernetes documentation.

Note Continuous delivery is not supported for OpenShift clusters.

Read the following topics next:

- [Enable Continuous Delivery for a Cluster or Cluster Group](#)
- [Disable Continuous Delivery](#)
- [Create a Repository Credential for a Cluster or Cluster Group](#)
- [Edit a Repository Credential](#)
- [Delete a Repository Credential](#)
- [Add a Git Repository to a Cluster or Cluster Group](#)
- [Edit a Git Repository](#)
- [Remove a Git Repository](#)
- [Add a Kustomization to a Cluster or Cluster Group](#)
- [Edit a Kustomization](#)
- [Delete a Kustomization](#)

Enable Continuous Delivery for a Cluster or Cluster Group

Use Tanzu Mission Control to set up your Kubernetes cluster or cluster groups for continuous delivery.

When you enable continuous delivery on a cluster or cluster group, you can associate a Git repository and sync folders in the repository to the cluster.

Note that if a resource is created at cluster group level, it gets fanned out to clusters under that cluster group. These are read-only resources at the cluster level and you can't edit or delete them from the cluster page.

Clusters can be moved from one cluster group to another. When that's done, resources from the old cluster group get deleted and resources from the new cluster group are added.

Note Continuous delivery is not supported for Openshift clusters.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to enable continuous delivery.

- To enable continuous delivery, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- 1 Click Cluster or Cluster groups in the left navigation pane.
- 2 Select the cluster or cluster group for which you want to enable continuous delivery.
- 3 Click on the **Add-ons** tab.
- 4 Click **Enable Continuous Delivery**.

Results

When you click **Enable Continuous Delivery**, Tanzu Mission Control installs Flux on your clusters to sync kustomizations from your Git repository. The installation adds the following namespaces in your clusters:

- `tanzu-fluxcd-packageinstalls` - namespace for Flux package install resources (source controller and Kustomization controller)
- `tanzu-source-controller` - namespace for Flux source controller deployments
- `tanzu-kustomize-controller` - namespace for Flux Kustomize controller deployments
- `tanzu-continuousdelivery-resources` - namespace for user-created `GitRepository` and `Kustomization` resources, and `intents` resources

Prior to installing Flux, Tanzu Mission Control searches for Flux CRDs in your cluster. If Flux CRDs are present, Tanzu Mission Control uses the currently installed instance rather than installing a new one. If the CRDs are not present, Tanzu Mission Control installs the Flux source controller and Kustomize controller and subsequently manages their lifecycles.

What to do next

After you enable continuous delivery on your cluster or cluster group, you can add Git repositories and kustomizations for the cluster or cluster group.

If you intend to use continuous delivery to install applications from Helm charts stored in a Helm repository, you must first install the Flux Helm controller from the Tanzu standard repository. For more information, see [Chapter 17 Managing Packages and Releases in Your Cluster](#).

Disable Continuous Delivery

As a cluster administrator, you can disable continuous delivery for a cluster or cluster group.

When you disable continuous delivery on a cluster group, it gets disabled on all clusters in the group except on the ones where atomic resources (cluster level) are created.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

- To disable continuous delivery, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to disable continuous delivery.
- 2 Click the **Add-ons** tab.
- 3 In the top right corner, click **Actions**, and then choose **Disable continuous delivery**.

Results

When you disable continuous delivery for a cluster or cluster group, Tanzu Mission Control removes the continuous delivery extension and CRD, and the `GitRepository` and `Kustomization` intents. However, the package install resources in the `tanzu-fluxcd-packageinstalls` namespace are not removed from the clusters.

What to do next

To remove the Flux CD package install resources (`source-controller` and `kustomize-controller`), use the following commands. Make sure you replace `cluster-name` with the name of your cluster.

```
tmc cluster tanzupackage install delete source-controller --cluster-name cluster-name --
namespace-name tanzu-fluxcd-packageinstalls
```

```
tmc cluster tanzupackage install delete kustomize-controller --cluster-name cluster-name --
namespace-name tanzu-fluxcd-packageinstalls
```

Create a Repository Credential for a Cluster or Cluster Group

Use Tanzu Mission Control to create repository credentials in your cluster or cluster group for repositories that require authentication.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to create a repository credential.

- To create a repository credential, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

This procedure assumes that you have already enabled continuous delivery for your cluster, as described in [Enable Continuous Delivery for a Cluster](#).

If you want to use an SSH key to authenticate, then make sure you have already generated the key as described in [Generate an SSH Key for Authentication to a Git Repository](#).

This procedure is necessary only if you intend to connect to a repository that requires authentication. If your repository does not require authentication, you can skip this procedure.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to create a repository credential.
- 2 On the cluster or cluster group detail page, click the **Add-ons** tab.
- 3 In the navigation menu below **Secrets**, click **Repository credentials**.
- 4 Click **Create Repository Credential**.
- 5 Enter a name for the repository credential.
- 6 You can optionally provide a description for the credential.
- 7 Select the type of credential to create, and then enter the required information.
 - For a `Username/Password` credential, enter the user name and password required to access the repository.

If your repository is hosted in GitHub, use a personal access token because GitHub prevents cloning with username/password authentication. Put your personal access token in the password field.
 - For an `SSH Key` credential, enter the SSH key and known hosts through which to access the repository. For more information, see [Generate an SSH Key for Authentication to a Git Repository](#).

If you are using a Git repository that requires authentication for access, then you must use SSH key authentication.

- For a CA Certificate credential, enter the CA certificate and optionally username and password to access the repository, depending on how your Git repository is configured.

8 After you finish configuring the credential, click **Create Repository Credential**.

Results

When you click **Create Repository Credential**, Tanzu Mission Control creates a credential in your cluster. You can use this repository credential when you add a Git repository to your cluster. If you subsequently disable continuous delivery on your cluster, Tanzu Mission Control removes the credential.

Generate an SSH Key for Authentication to a Git Repository

Generate a key that you can use when adding a Git repository that requires authentication.

This approach requires that you generate an SSH key for Git access, and then use that key when you create a repository credential. You also need to identify your repository using the SSH URL when adding it to your cluster.

Prerequisites

Make sure that you have access to your Git repository.

Procedure

1 Generate an SSH key pair using `ssh-keygen`.

```
ssh-keygen -t key-type
```

For example:

```
ssh-keygen -t ecdsa-sha2-nistp256
```

2 Keep the private key, you'll use it when creating a repository credential .

3 Copy the public key to your Git repository server.

4 Retrieve the host key from Git using `ssh-keyscan`.

```
ssh-keyscan -t type hostname
```

For example:

```
ssh-keyscan -t ecdsa-sha2-nistp256 github.com
```

5 Copy the host key from this output, you'll use it when creating a repository credential.

Results

You now have the following keys that you can use when creating a repository credential to authenticate your cluster for continuous delivery from a Git repository.

- a private SSH key
- a host key

What to do next

Use your new SSH key when you create the repository credential, as described in [Create a Repository Credential for a Cluster or Cluster Group](#).

When you add the repository to your cluster, as described in [Add a Git Repository to a Cluster or Cluster Group](#), use the SSH URL.

Edit a Repository Credential

As a cluster administrator, you can modify the configuration of a repository credential.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to edit a repository credential.

- To edit a repository credential, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- 1 Navigate to the cluster or cluster group whose repository credential you want to edit.
- 2 On the cluster or cluster group detail page, click the **Add-ons** tab.
- 3 In the Secrets section of the navigation menu, click **Repository credentials**.
- 4 Click the repository credential that you want to edit, and then click **Edit**.
- 5 Modify the credential as necessary, and then click **Save**.

You cannot modify the name of the credential after you create it.

Delete a Repository Credential

As a cluster administrator, you can delete a repository credential that is no longer needed.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Remove a Git Repository](#).

Make sure you have the appropriate permissions to delete a repository credential.

- To delete a repository credential, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Before you can delete a repository credential, you must first remove the credential from any repositories that use it.

Procedure

- 1 Navigate to the cluster or cluster group whose repository credential you want to delete.
- 2 On the cluster or cluster group detail page, click the **Add-ons** tab.
- 3 In the Secrets section of the navigation menu, click **Repository credentials**.
- 4 Click the menu icon for the repository credential that you want to remove, and then choose **Delete**.

Add a Git Repository to a Cluster or Cluster Group

Use Tanzu Mission Control to add a Git repository to your cluster or cluster group to use for a kustomization.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to add a repository.

- To add a repository, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Make sure you have already enabled continuous delivery for the cluster, as described in [Enable Continuous Delivery for a Cluster or Cluster Group](#).

If your Git repository requires a secret, make sure that you have already added a repository credential, as described in [Create a Repository Credential for a Cluster or Cluster Group](#).

Procedure

- 1 Navigate to the cluster or cluster group to which you want to add a Git repository.
- 2 Click the **Add-ons** tab.
- 3 In the navigation menu on the Add-ons tab, under **Sources** click **Git repositories**.
- 4 Click **Add Git Repository**.
- 5 Enter a name and (optionally) a description for the repository.

- Specify the URL to the repository, beginning with HTTP, HTTPS, or SSH.

Note For clusters running behind a proxy that requires a self-signed CA certificate, you must provide the final redirected URL when you add the Git repository to the cluster.

The URL format for SSH is slightly different from the HTTP format. For example, if the Git repository looks like this:

```
https://github.com/my-integrations/my-private-configmap
```

then the SSH URL looks like this:

```
ssh://github.com/my-integrations/my-private-configmap.git
```

- Specify the credential setting for the repository.
 - You can optionally accept the default of `No credentials needed`.
 - To select a previously-defined credential, click the delete icon (x) and then select the credential from the dropdown.
- You can optionally expand **Advanced settings** to specify additional options, like Branch and Tag.
- After you have defined the configuration for the Git repository, click **Add Git Repository**.

Edit a Git Repository

You can change repository settings as needed.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to edit a repository.

- To edit a repository, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- Navigate to the cluster or cluster group for which you want to edit a git repository.
- Click the **Add-ons** tab.
- In the Add-ons menu in the Sources section click **Git repositories**.
- Click the menu icon for the Git repository that you want to edit, and then choose **Edit**.
- On the Edit Git repository page, make the necessary modifications, and then click **Save**.

You can edit the description, URL, and credential. However, you cannot modify the name of the repository after it has been created.

Results

When you click **Save**, the repository settings are updated. The new settings are used the next time a sync is performed.

Remove a Git Repository

As a cluster administrator, you can use Tanzu Mission Control to remove your cluster's or cluster group's connection to a repository.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Remove a Git Repository](#).

Make sure you have the appropriate permissions to delete a repository.

- To delete a repository, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Before you can remove a Git repository, you must first delete the kustomizations that use the repository.

Procedure

- 1 Navigate to the cluster or cluster group whose Git repository you want to delete.
- 2 Click the **Add-ons** tab.
- 3 In the navigation menu on the Add-ons tab, click **Git repositories** below Sources.
- 4 Click the menu icon for the Git repository that you want to remove, and then choose **Remove**.
- 5 In the confirmation dialog, click **Delete**.

Add a Kustomization to a Cluster or Cluster Group

As a cluster administrator, you can use Tanzu Mission Control to add a kustomization to a cluster or cluster group that has continuous delivery enabled on it.

A kustomization is a file or set of files that declaratively describes a set of Kubernetes configurations. For more information, see [Declarative Management of Kubernetes Objects Using Kustomize](#) in the Kubernetes documentation.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to add a kustomization.

- To create a kustomization, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Make sure you have already enabled continuous delivery for the cluster, as described in [Enable Continuous Delivery for a Cluster or Cluster Group](#), and defined a Git repository, as described in [Add a Git Repository to a Cluster or Cluster Group](#).

For Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu that have pod security policies, make sure you have the appropriate role bindings on the workload cluster. Without the appropriate permissions, kustomizations can fail to be deployed. For information about creating role bindings, see [Example Role Bindings for Pod Security Policy](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Procedure

- 1 Navigate to the detail page for the cluster or cluster group to which you want to add a kustomization.
- 2 Click the **Add-ons** tab.
- 3 In the Add-ons menu click **Kustomizations** in the Continuous Delivery section.
- 4 Click **Add kustomization**.
- 5 On the Add kustomization page, enter a name for the kustomization.
- 6 You can optionally provide a description for the kustomization.
- 7 Select the Git repository you want to use for the kustomization.
- 8 Specify the path of the kustomization from the root of the Git repository to the folder containing the kustomization file.
- 9 You can optionally expand the Advanced settings section to specify more options.

- a Specify a target namespace.

The target namespace indicates the namespace for objects that are created as part of the kustomization.

If you specify a target namespace, that namespace must already exist on the cluster or it must be defined in a manifest included in the kustomization. If it does not exist, applying the kustomization fails.

- If you defined a target namespace in a manifest and also specify one in TMC, then the value defined in the manifest takes precedence.
- If you did not define a target namespace in a manifest and you do not specify one in TMC, then the kustomization is added to the default namespace.

- b Click to toggle **Prune** on or off.

If you configure the kustomization to prune (toggled on), the resources on the cluster that were created through this kustomization are deleted when the kustomization is removed from the cluster. Resources created before prune is enabled are not deleted when the kustomization is removed.

- 10 After you finish configuring the kustomization, click **Add kustomization**.

Edit a Kustomization

As a cluster administrator, you can edit the description, path, and prune setting for a kustomization.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to edit a kustomization.

- To edit a kustomization, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- 1 Navigate to the detail page for the cluster or cluster group whose kustomization you want to delete.
- 2 Click the **Add-ons** tab.
- 3 In the navigation menu in the Continuous Delivery section, click **Kustomizations**.
- 4 Click the menu icon for the kustomization that you want to edit, and then choose **Edit**.
- 5 On the Edit kustomization page, make the necessary modifications, and then click **Save**.

You can edit the description, path, and prune setting. However, you cannot modify the name of the kustomization or its repository after it has been created.

Delete a Kustomization

Use Tanzu Mission Control to remove a kustomization from your cluster or cluster group.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Chapter 1 Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to delete a kustomization.

- To delete a kustomization, you must be associated with the `cluster.admin` or `clustergroup.admin` role.

Procedure

- 1 Navigate to the detail page for the cluster or cluster group whose kustomization you want to delete.
- 2 Click the **Add-ons** tab.
- 3 In the navigation menu, click **Kustomizations** below Continuous Delivery.

- 4 Click the menu icon for the kustomization that you want to remove, and then choose **Remove**.

Results

When you remove a kustomization from your cluster or cluster group, that kustomization is no longer applied when Flux performs a sync. Additionally, if the kustomization is configured to prune, the Kubernetes objects that have been applied by the kustomization are removed from the clusters.

Managing Cluster Secrets

13

As a platform operator, you can use Tanzu Mission Control to create and manage secrets for use in your clusters.

Tanzu Mission Control allows you to create image pull secrets in a namespace that allow namespaces in managed clusters to authenticate to private registries, including registries that store Tanzu Application Platform components.

You can also make a secret created in one namespace available for use by other namespaces in your cluster, providing a single place to manage secrets for the cluster. The secret is a Tanzu Mission Control resource.

About secret generation and exporting

Tanzu Mission Control leverages `secretgen-controller` to manage the export of secrets to all other namespaces in the cluster. This does not copy the secret into other namespaces, but allows other namespaces to request access to, and use that secret.

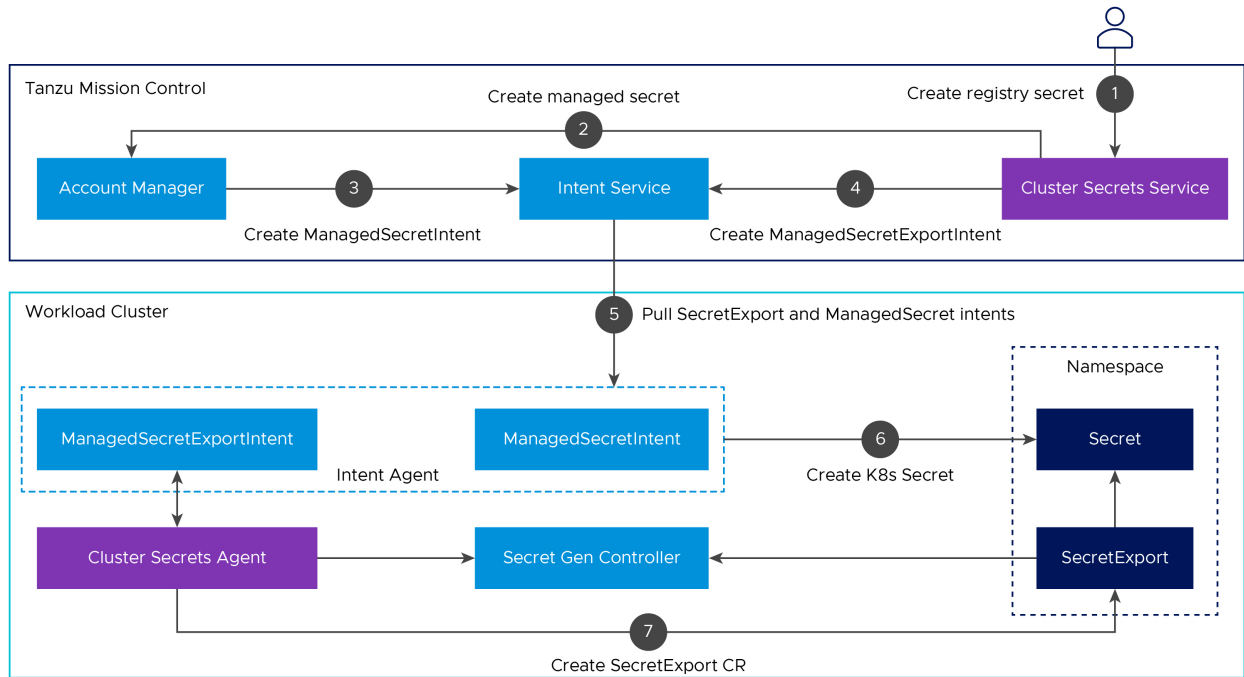
As shown in the architecture diagram below, Tanzu Mission Control creates and exports secrets using the cluster secret service. When a secret is created and exported to all namespaces through this service, Tanzu Mission Control stores the secret using the account manager service and uses the secure channel with the in-cluster intent agent to provision the secret to the cluster and to send the secret export intent to the cluster secret agent, a new in-cluster agent.

The cluster secret agent is responsible for deploying `secretgen-controller`, and for creating secret export CRs that are picked up by `secretgen-controller`.

As a best practice, export only credentials that allow read-only access to the registry.

For more information about `secretgen-controller`, see [secretgen-controller](#) in the Carvel documentation on GitHub.

Figure 13-1. Cluster Secrets Architecture



Read the following topics next:

- [View Cluster Secrets](#)
- [Create a Kubernetes Secret in a Cluster or Cluster Group](#)
- [Edit a Cluster Secret](#)
- [Delete a Cluster Secret](#)

View Cluster Secrets

View the secrets that have been created in your clusters to see their status and the namespaces to which they apply.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to view cluster secrets.

- To view cluster secrets, you must be associated with the `cluster.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster or cluster group whose secrets you want to see.
- 2 On the detail page, click the **Add-ons** tab.

- 3 Click **Kubernetes secrets** in the Secrets section of the Add-ons menu.

The table shows the secrets that have been created for this cluster, along with their status and other information about the secret.

- 4 To view the details for a secret, click the name of the secret.

Results

The secret details page shows the current information and status of the secret.

Table 13-1. Secret Status

Status	Meaning
Reconciling	Secret has been created in Tanzu Mission Control and is reconciling with the cluster.
ReconcileFailed	Secret has been created in Tanzu Mission Control and tried to reconcile with the cluster but failed. It will re-try until it succeeds.
ReconcileSucceeded	Secret has been created in Tanzu Mission Control and reconciled with the cluster successfully.
Deleting	Secret has been deleted by the user and Tanzu Mission Control is in the process of deleting.

Table 13-2. Export Status

Status	Meaning
Reconciling	User exported the secret to all namespaces and Tanzu Mission Control is in the process of executing the required operations via <code>secretgen-controller</code> .
ReconcileFailed	User exported the secret to all namespaces but <code>secretgen-controller</code> has failed to reconcile the secret export.
ReconcileSucceeded	The secret was successfully exported to specified namespaces.
Deleting	User removed the export from all namespaces and Tanzu Mission Control is in the process of removing the export via <code>secretgen-controller</code> .
FeatureDisabled	Feature is not available, typically because the cluster is running a version of Tanzu Kubernetes Grid that is not compatible with this feature.

Create a Kubernetes Secret in a Cluster or Cluster Group

Create a secret to use with cluster repositories.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to create a cluster secret.

- To create a cluster secret, you must be associated with the `cluster.edit` role.
- To create a Kubernetes secret for a cluster group, you must be associated with the `clustergroup.edit` role.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster or cluster group for which you want to create a secret.
- 2 On the detail page, click the **Add-ons** tab.
- 3 Click **Kubernetes secrets** in the Secrets section of the Add-ons menu.
- 4 Click **Create Kubernetes Secret**.
- 5 Enter a name for the secret.
- 6 Select the namespace in which to create the secret from the dropdown list.
- 7 Select the type of secret to create, and then enter the details for the secret.
 - For an opaque secret, enter one or more key value pairs.
 - For a registry secret:
 - a Enter the URL for the registry.
 - b Enter a valid username and password for the registry URL.
- 8 Click **Create**.

Tanzu Mission Control generates the secret.
- 9 Click **Next**.
- 10 You can optionally click the toggle to make the secret available to all namespaces in the cluster.
- 11 To see the contents of the secret, click **View YAML**.
- 12 Click **Finish**.

When you click **Finish**, Tanzu Mission Control applies the secret to your cluster in the specified namespace, and directs you to the Secrets tab of the detail page where you can see the new secret in the table.

Edit a Cluster Secret

Edit an existing cluster secret to change the username, password, and availability.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to edit a cluster secret.

- To edit a cluster secret, you must be associated with the `cluster.edit` role.
- To edit a Kubernetes secret for a cluster group, you must be associated with the `clustergroup.edit` role.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster or cluster group that contains the secret you want to edit.
- 2 On the detail page, click the **Add-ons** tab.
- 3 Click **Kubernetes secrets** in the Secrets section of the Add-ons menu.
- 4 Click the name of the secret that you want to edit.
- 5 Click the **Action** button and select **Edit**.
- 6 You can optionally change the username and password.
- 7 You can optionally click the toggle to make this secret available to all namespaces in the cluster.
- 8 You can optionally click **View YAML** to see the contents of the secret.
- 9 When you have made the necessary changes, click **Save**.

Delete a Cluster Secret

Delete a secret that is no longer needed from your cluster.

Prerequisites

Log in to the Tanzu Mission Control console, as described in [Log In to the Tanzu Mission Control Console](#).

Make sure you have the appropriate permissions to delete a cluster secret.

- To delete a cluster secret, you must be associated with the `cluster.edit` role.
- To delete a Kubernetes secret for a cluster group, you must be associated with the `clustergroup.edit` role.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster or cluster group that contains the secret you want to delete.
- 2 On the detail page, click the **Add-ons** tab.

- 3 Click **Kubernetes secrets** in the Secrets section of the Add-ons menu.
- 4 Click the name of the secret that you want to delete.
- 5 Click the **Action** button and select **Delete**.
- 6 In the confirmation dialog, click **Delete**.

Results

When you click **Delete**, Tanzu Mission Control removes the secret from your cluster. It is no longer available in any of the namespaces with which it was shared.

As a platform operator, you can integrate external services to add functionality to VMware Tanzu Mission Control.

Through Tanzu Mission Control, you can enable connections to external services that allow you to extend how you manage your Kubernetes clusters.

For example, you can enable integration with Tanzu Observability by Wavefront, and then add clusters to capture metrics for analysis in Wavefront. For more information, see [Observation and Analysis of Cluster Health and Resources](#) in *VMware Tanzu Mission Control Concepts*.

Read the following topics next:

- [Enable Observability for Your Organization](#)
- [Disable Observability for Your Organization](#)
- [Add a Cluster to Observability](#)
- [Add a Cluster Group to Observability](#)
- [Rotate Your Observability Credentials](#)
- [Edit the Configuration of Your Tanzu Observability Collector](#)
- [Remove a Cluster from Observability](#)
- [Enable Service Mesh for Your Organization](#)
- [Disable Service Mesh for Your Organization](#)
- [Add a Cluster to Service Mesh](#)
- [Remove a Cluster from Service Mesh](#)
- [Upgrade Service Mesh](#)
- [Roll Back Service Mesh](#)

Enable Observability for Your Organization

As a platform operator, you can enable Tanzu Observability by Wavefront for your organization, which allows you to connect your managed clusters to your Wavefront account to collect and analyze cluster metrics.

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that you have set up a Wavefront account.

Note Although you don't specify your Wavefront account until you add clusters, you should have the account set up and ready before enabling this functionality at the organizational level.

For more information about Tanzu Observability by Wavefront, see [Observation and Analysis of Cluster Health and Resources](#) in *VMware Tanzu Mission Control Concepts*.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To enable observability for your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane, click **Administration**.
- 2 On the Administration page, click the **Integrations** tab.
- 3 In the Tanzu Observability by Wavefront box, click **Enable**.
- 4 In the confirmation dialog, click **Confirm**.

Results

When you click **Confirm**, the Tanzu Observability by Wavefront integration is enabled for your organization, and you can start adding clusters to collect data for analysis through your Wavefront account.

What to do next

After you have enabled observability for your organization, you can [Create a Tanzu Observability Credential Object](#) and then [Add a Cluster to Observability](#).

Disable Observability for Your Organization

As a platform operator, you can disable Tanzu Observability by Wavefront for your organization.

Prerequisites

Log in to the Tanzu Mission Control console.

Before you attempt to disable integration with Tanzu Observability by Wavefront, make sure that you remove Tanzu Observability from all clusters.

Make sure you have the appropriate permissions.

- To disable observability for your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane, click **Administration**.
- 2 On the Administration page, click the **Integrations** tab.
- 3 In the Tanzu Observability by Wavefront box, click **Disable**.
- 4 In the confirmation dialog, click **Confirm**.

Results

When you click **Confirm**, the Tanzu Observability by Wavefront integration is disabled for your organization, and you can no longer collect data for analysis through your Wavefront account.

Add a Cluster to Observability

As an infrastructure operator in VMware Tanzu Mission Control, you can install the collector on a cluster to start sending data to your Wavefront account.

After observability has been enabled on your organization, you can then add the clusters that you want to monitor in Wavefront. When you add a cluster to Tanzu Observability, an extension is installed on the cluster to collect and send metrics data in regular intervals (by default, every 60 seconds).

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To add your cluster to Tanzu Observability by Wavefront, you must be associated with the `cluster.admin` role on the cluster.
- To use an existing Tanzu Observability credential, you must be associated with the `credential.view` role on the credential.

For more information about Tanzu Observability credentials, see [Create a Tanzu Observability Credential Object](#).

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster you want to add to Tanzu Observability.
- 2 On the cluster detail page, click the **Actions** dropdown menu and then choose **Tanzu Observability by Wavefront > Add**.
- 3 In the Add dialog, specify how to attach to your Wavefront account.
 - If you have access to an existing Tanzu Observability credential, you can select it in the **Tanzu Observability credential** field.

- If you want to enter your authorization credentials directly, click **Setup New Credential**, and then enter the Wavefront instance URL and API token from your Wavefront account.

Note The value for these fields come from your Wavefront account, and not your Tanzu Mission Control account. For information about how to retrieve these values from your Wavefront account, see [Your Wavefront Account](#) in the *Tanzu Observability by Wavefront* documentation.

4 Click **Confirm**.

Results

When you click **Confirm**, Tanzu Mission Control installs an extension on your cluster to collect data from the cluster and send it to your Wavefront account in one minute intervals.

Add a Cluster Group to Observability

As an infrastructure operator in VMware Tanzu Mission Control, you can implement observability on a cluster group, which installs the collector on a set of clusters to enable them to start sending data to your Wavefront account.

After observability has been enabled for your organization, you can then add the clusters that you want to monitor in Wavefront. When you implement observability on a cluster group, Tanzu Mission Control pushes that out to each cluster in the group that isn't already added to Tanzu Observability. On each cluster that you add to Tanzu Observability, an extension is installed to collect and send metrics data in regular intervals (by default, every 60 seconds). Any clusters that have individually been added to Tanzu Observability are marked as overrides, and are not included in the cluster group integration.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To edit the configuration of a cluster group, you must be associated with the `clustergroup.edit` role on the cluster group.
- To add the clusters in a cluster group to Tanzu Observability by Wavefront, you must be associated with the `cluster.admin` role on each cluster.
- To use an existing Tanzu Observability credential, you must be associated with the `credential.view` role on the credential.

For more information about Tanzu Observability credentials, see [Create a Tanzu Observability Credential Object](#).

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster group that you want to add to Tanzu Observability.

- 2 On the cluster group detail page, click the **Integrations** tab.
- 3 Click the **Actions** dropdown menu, and then choose **Tanzu Observability by Wavefront > Add**.
- 4 In the Add dialog, select the Tanzu Observability credential to use to attach to your Wavefront account.
- 5 Click **Confirm**.

Results

When you click **Confirm**, Tanzu Mission Control installs an extension on each cluster in the cluster group to collect data from the cluster and send it to your Wavefront account in regular intervals. It might take several minutes before the collector is installed and running on each of the clusters in the cluster group.

Note If a cluster in the cluster group has already been added to Tanzu Observability, that cluster is marked as an override and is not included in the group integration.

Rotate Your Observability Credentials

As an infrastructure operator in Tanzu Mission Control, you can update the API token required to send data to your Wavefront account.

An API token from Tanzu Observability by Wavefront is required for your clusters to send data to your Wavefront account. As a best practice, you should use a service account for this access and regularly update the token to promote a secure environment.

If you use a Tanzu Observability credential object, you can update the credential as described in [Edit a Tanzu Observability Credential Object](#). If you entered the authorization credentials directly when you added the cluster to observability, this procedure explains how to update your API token.

For more information about Tanzu Observability credentials, see [Create a Tanzu Observability Credential Object](#).

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To edit the API token entered directly for your cluster, you must be associated with the `cluster.admin` role on the cluster.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster for which you want to update the API token.

- 2 On the cluster detail page, click the menu icon for the Tanzu Observability, and then choose **Rotate Credential**.
- 3 In the dialog, enter the new API token.
- 4 Click **Confirm**.

Edit the Configuration of Your Tanzu Observability Collector

Use Tanzu Mission Control to modify the collector configuration that controls how data is sent to Tanzu Observability.

The collector configuration for Tanzu Observability specifies a number of details about the data that is sent to your Tanzu Observability account, like what data is collected and how often it is sent. Using Tanzu Mission Control, you can change the configuration to tune data collection for each of your clusters.

For more information about collector configuration for Tanzu Observability, go to <https://github.com/wavefrontHQ/wavefront-collector-for-kubernetes/blob/main/docs/configuration.md#configuration-file>.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To modify the Tanzu Observability collector configuration for your cluster, you must be associated with the `cluster.admin` role on the cluster.

This procedure assumes that you have a managed Kubernetes cluster that you have already added to Tanzu Observability.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster whose Tanzu Observability collector you want to configure.
- 2 On the cluster details page, in the top right corner, click **Actions** and then choose **Tanzu Observability > Configure collector**.

The Configure page shows the YAML code for the collector as it is currently configured.

- 3 Edit the YAML code to configure the collector.
 - You can edit the code directly in the provided code box.
 - If you have already edited the collector, you can click **Reset to Default Configuration** to pull the default configuration from your Tanzu Observability account.
 - To load a new configuration from a file, click **Import** and then select the file to import.

- 4 When you have finished making changes to the collector configuration YAML code, click **Save**.

Make sure you click **Save** to apply your changes. Regardless of the method that you use to update the collector configuration YAML code, your changes are not applied to the cluster until you save.

Remove a Cluster from Observability

As an infrastructure operator in VMware Tanzu Mission Control, you can remove a cluster from Tanzu Observability by Wavefront to stop sending data to your Wavefront account.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To remove your cluster from Tanzu Observability by Wavefront, you must be associated with the `cluster.admin` role on the cluster.

Procedure

- 1 Navigate to the cluster you want to remove in the Tanzu Mission Control console.
- 2 On the cluster detail page, click the **Actions** dropdown menu and then choose **Tanzu Observability by Wavefront > Remove**.
- 3 In the Remove dialog, click **Confirm**.

Results

When you click **Confirm**, Tanzu Mission Control removes the extension from your cluster.

Enable Service Mesh for Your Organization

As a platform operator, you can enable Tanzu Service Mesh for your organization, which allows you to connect your managed clusters to your Tanzu Service Mesh account.

Note To integrate Tanzu Service Mesh with Tanzu Mission Control, you must be running under a Tanzu Service Mesh Advanced license. Tanzu Mission Control does not support integration with Tanzu Service Mesh Enterprise. Although you can enable the integration, you cannot add clusters to Tanzu Service Mesh Enterprise using Tanzu Mission Control.

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that you have set up a Tanzu Service Mesh account.

Note To enable Tanzu Service Mesh and then add clusters, your Tanzu Service Mesh account must be in the same organization as your Tanzu Mission Control account.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To enable service mesh for your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane, click **Administration**.
- 2 On the Administration page, click the **Integrations** tab.
- 3 In the Tanzu Service Mesh box, click **Enable**.
- 4 In the confirmation dialog, click **Confirm**.

Results

When you click **Confirm**, the Tanzu Service Mesh integration is enabled for your organization, and you can start adding clusters to your Tanzu Service Mesh account.

Disable Service Mesh for Your Organization

As a platform operator, you can disable Tanzu Service Mesh for your organization.

Prerequisites

Log in to the Tanzu Mission Control console.

Before you attempt to disable integration with Tanzu Service Mesh, make sure that you remove Tanzu Service Mesh from all clusters.

Make sure you have the appropriate permissions.

- To disable service mesh for your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane, click **Administration**.
- 2 On the Administration page, click the **Integrations** tab.
- 3 In the Tanzu Service Mesh box, click **Disable**.
- 4 In the confirmation dialog, click **Confirm**.

Results

When you click **Confirm**, the Tanzu Service Mesh integration is disabled for your organization, and you can no longer connect clusters to your Tanzu Service Mesh account.

Add a Cluster to Service Mesh

As an infrastructure operator in VMware Tanzu Mission Control, you can install the Tanzu Service Mesh extensions on a cluster and connect it to your Tanzu Service Mesh account.

After Tanzu Service Mesh (TSM) has been enabled on your organization, you can then add the clusters that you want to manage with TSM. When you add a cluster to TSM, an extension is installed on the cluster to communicate with the TSM service.

Note To integrate Tanzu Service Mesh with Tanzu Mission Control, you must be running under a Tanzu Service Mesh Advanced license. Tanzu Mission Control does not support integration with Tanzu Service Mesh Enterprise. Although you can enable the integration, you cannot add clusters to Tanzu Service Mesh Enterprise using Tanzu Mission Control.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To add your cluster to Tanzu Service Mesh, you must be associated with the `cluster.admin` role on the cluster.

Make sure your cluster satisfies the requirements for Tanzu Service Mesh, as described in [Tanzu Service Mesh Environment Requirements and Supported Platforms](#) in the *VMware Tanzu Service Mesh* documentation.

Procedure

- 1 Navigate to the cluster you want to add in the Tanzu Mission Control console.
- 2 On the cluster page, click **Add Integration** in the **Integrations** pane and select Tanzu Service Mesh.
- 3 In the Add dialog, select the options that you want for handling namespaces.

Note System namespaces are always excluded from Tanzu Service Mesh.

- You can select **cluster admin delegation**. This option allows a platform administrator to delegate injection decision-making to the cluster operator. When this option is enabled, TSM does not specify an inclusion or exclusion model on the cluster namespaces, which effectively delegates the task of labeling namespaces to the cluster operator. You can choose to define no namespaces for injection during onboarding, and enable this option to delegate the operation entirely. When this option is disabled, you choose the namespaces to exclude when adding the cluster to TSM.

If cluster admin delegation is selected, then you need to add the following label to the namespaces that you want to include in TSM:

```
Istio-injection=enabled --overwrite
```

- You can choose to enable Tanzu Service Mesh on all namespaces in the cluster (excluding system namespaces).
 - To exclude namespaces from service mesh, select **Exclude namespaces**.
 - a To exclude a specific namespace, choose **Is exactly**, select the namespace, and then click **Add Exclusion**.
 - b To exclude namespaces based on a starting pattern, choose **Starts with**, enter the first few characters of the namespaces you want to exclude, and then click **Add Exclusion**.
- 4 You can optionally choose an alternative certificate authority.
- The dropdown list shows the CAs that have been configured in TSM. If you don't choose an alternative, the cluster uses the default CA defined by TSM.
- 5 Click **Confirm**.

Results

When you click **Confirm**, Tanzu Mission Control installs an extension on your cluster to enable Tanzu Service Mesh. After the extension is installed, you can access the Tanzu Service Mesh console through the Integrations box on the Overview tab of the cluster details page in the Tanzu Mission Control console.

Remove a Cluster from Service Mesh

As an infrastructure operator in VMware Tanzu Mission Control, you can remove a cluster from Tanzu Service Mesh.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To remove your cluster from Tanzu Service Mesh, you must be associated with the `cluster.admin` role on the cluster.

Procedure

- 1 Navigate to the cluster you want to remove in the Tanzu Mission Control console.
- 2 On the cluster detail page, click the **Actions** dropdown menu and then choose **Tanzu Service Mesh > Remove**.
- 3 In the Remove dialog, click **Confirm**.

Results

When you click **Confirm**, Tanzu Mission Control removes the extension from your cluster.

Upgrade Service Mesh

As a platform operator, you can upgrade Tanzu Service Mesh to take advantage of the latest features, enhancements, and fixes in new versions for managed clusters in your Tanzu Service Mesh account.

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that you have set up a Tanzu Service Mesh account.

Note To enable Tanzu Service Mesh and then add clusters, your Tanzu Service Mesh account must be in the same organization as your Tanzu Mission Control account.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To upgrade service mesh for your organization, you must be associated with the `organization.admin` role.

For more information about Tanzu Service Mesh upgrades, see [Manage Tanzu Service Mesh Updates](#)

Procedure

- 1 In the left navigation pane, click **Clusters**.
- 2 Select the cluster for which you want to upgrade the Tanzu Service Mesh version.
- 3 The Integrations section of the cluster screen shows Tanzu Service Mesh with the information icon next to it indicating that a newer version is available. Click the Tanzu Service Mesh menu and select **Upgrade to <version-number>**, where *<version-number>* is the desired version.
- 4 Click **Upgrade** in the confirmation dialog.

The upgrade process starts and progress is shown in the Integrations section of the cluster screen. Note that it can take approximately a minute to complete the process.

Results

Tanzu Service Mesh is upgraded to the selected version.

Roll Back Service Mesh

You can roll back the Tanzu Service Mesh version manually for managed clusters in your Tanzu Service Mesh account.

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that you have set up a Tanzu Service Mesh account.

Note To enable Tanzu Service Mesh and then add clusters, your Tanzu Service Mesh account must be in the same organization as your Tanzu Mission Control account.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To upgrade service mesh for your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane, click **Clusters**.
- 2 Select the cluster for which you want to roll back the Tanzu Service Mesh version.
- 3 Click the Tanzu Service Mesh menu and select **Rollback to <version-number>**, where *<version-number>* is the last version that was installed.
- 4 Click **Roll Back** in the confirmation dialog.

The rollback process starts and progress is shown in the Integrations section of the cluster screen. Note that it can take approximately a minute to complete the process.

Results

Tanzu Service Mesh is rolled back to the previous version.

Manage Object Labels

15

Use VMware Tanzu Mission Control to create labels to identify and organize your Kubernetes resources.

You can apply labels to the following objects:

- cluster groups
- clusters
- workspaces
- namespaces
- workloads

The labels you apply to an object are displayed in object lists and on the detail page for the object. For example, the Cluster groups page in the Tanzu Mission Control console shows a list of your cluster groups along with the labels that are applied to them. When you click on a cluster group, its labels are displayed in the summary panel at the top, and the list of clusters in the cluster group has a column that shows the labels defined for each cluster. You can filter or sort on the Labels column to find clusters with a given key or value.

In addition to the user-defined labels that you have applied to an object, there can also be system-defined labels that have been applied by Tanzu Mission Control. System-defined labels are outlined in blue, while user-defined labels are outlined in orange. You cannot modify system-defined labels.

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To manage the labels for an object, you must be associated with the `.edit` role for the object.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the object on which you want to apply or remove a label.
- 2 In the top right corner, click **Actions** and then choose **Edit labels** from the dropdown.

- 3 To add a label, enter a key and value, and then click **Add**.

You can optionally repeat this step to create additional labels.

- 4 To remove a label, click the delete icon (x) for the label.

You cannot modify a label you've already created, but you can delete it and then create a new one.

- 5 When you have finished editing the labels, click **Save** to apply the labels to the object.

Managing Event Logs

16

As a platform administrator, use VMware Tanzu Mission Control to generate a report of the audit events that occur in your organization.

Tanzu Mission Control continually collects and stores logs of audit events that describe activities that occur in your organization. You can use Tanzu Mission Control to generate and download reports of these events. For more information about audit event logging in Tanzu Mission Control, see [Event Logs](#) in *VMware Tanzu Mission Control Concepts*.

Viewing the Event Log

The Events page in the Tanzu Mission Control console shows the events that have occurred in your organization. By default, this view includes both cluster management events and audit events that have occurred in the last three days. When viewing your events, you have the following display options:

- You can use the events time filter to specify the time range of displayed events from the last 10 minutes to the last three days.
- You can filter or sort by name or type to narrow your view of events.
- You can click the expand icon for an event to see its payload.

Note You can also use the `events/stream` API (in `v1alpha1`) to set up auto-ingestion. For more information, see the code sample on VMware Developer at <https://developer.vmware.com/web/dp/samples?id=7995>.

Read the following topics next:

- [Generate an Audit Report](#)
- [Download an Audit Report](#)
- [Delete or Cancel an Audit Report](#)

Generate an Audit Report

Use VMware Tanzu Mission Control to create a report of audit events.

Tanzu Mission Control continually collects and stores logs of audit events that describe activities that occur through Tanzu Mission Control. The log entry for each event is retained for 60 days, after which it is deleted. For more information, see [Event Logs](#) in *VMware Tanzu Mission Control Concepts*.

Prerequisites

Before starting this procedure, log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To generate an audit report, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Logs**.
- 2 Specify the start date and end date for the report.
The maximum range for an audit report is seven days.
- 3 Click **Prepare logs for download**.

Results

When you click **Prepare logs for download**, Tanzu Mission Control starts retrieving the events that match the criteria that you specified. The process of compiling the log entries into a downloadable report takes some time. Depending on the extent of the date range you specified and the number of audit events that have occurred, the report can take up to several hours to generate. During this time, you cannot initiate another report.

What to do next

After the report generation is completed, you can download and examine it. An audit report is available for 60 days after the report is generated. After 60 days, the report is permanently deleted.

Download an Audit Report

Download a report of audit events, that you generated using VMware Tanzu Mission Control.

Prerequisites

Before starting this procedure, log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To generate or download an audit report, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Logs**.
The audit reports that you have generated are listed on the Logs page.
- 2 Locate the audit report you want to download in the list.
- 3 In the status column, click the **Download** link.

Results

When you click **Download**, the report is downloaded as a compressed file called `audit.log.gz`.

Delete or Cancel an Audit Report

Delete a previously generated report of audit events, or cancel a report that is currently running.

Prerequisites

Before starting this procedure, log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To delete or cancel audit reports, you must be associated with the `organization.admin` role.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Logs**.
The audit reports that you have generated are listed on the Logs page. A report that shows `Preparing` in the Status column is currently running.
- 2 Select the audit report you want to delete.
- 3 Click the **Delete** button.
- 4 In the confirmation dialog, click **Delete**.

Results

When you click **Delete**, Tanzu Mission Control deletes the selected report. If the report was in the process of running, the report generation is canceled. This is a permanent deletion, you cannot recover a deleted report.

Managing Packages and Releases in Your Cluster

17

As a platform operator, you can use VMware Tanzu Mission Control to manage the packages and releases installed in your clusters.

The Catalog page in the Tanzu Mission Control console allows you to view and install packages available from the Tanzu Standard package repository, Helm charts from the Bitnami repository, and packages from your own custom Carvel package repositories.

Note The catalog in Tanzu Mission Control restricts the installation of packages from the Tanzu Standard package repository to only workload clusters running in Tanzu Kubernetes Grid. You cannot use the catalog to install packages from the Tanzu Standard package repository to non-TKG clusters. For non-TKG clusters, you can explore packages in the Helm charts tab. Helm charts are open source from Bitnami, and are not supported by VMware.

Tanzu Mission Control uses Carvel (an open source community standard) for package management. For more information about packages and package management using Carvel, visit <https://carvel.dev/kapp-controller/>.

Set a Firewall Rule for kapp-controller

Tanzu Mission Control requires ingress access to workload clusters to communicate with `kapp-controller`.

Tanzu Mission Control uses Carvel's `kapp-controller` service running in the cluster to install packages and communicate with installed packages. Tanzu Mission Control uses port `32767` for the `kapp-controller` service on non-Tanzu Kubernetes Grid workload clusters, and port `10100` on Tanzu Kubernetes Grid workload clusters.

Add a firewall rule allowing ingress from the control plane to the workload clusters on the relevant ports to ensure the `kapp-controller` service can operate correctly.

Read the following topics next:

- [View Packages](#)
- [Enable Helm Service on Your Cluster or Cluster Group](#)
- [Disable Helm Service](#)
- [Install a Helm Chart from a Helm Repository \(Create a Release\)](#)

- [Install a Helm Chart from a Git Repository](#)
- [Access Image Pull Secrets for Private Helm Charts](#)
- [Example Helm Release Using Private Image](#)
- [Edit an Installed Helm Release](#)
- [Delete an Installed Helm Release](#)
- [Install a Package](#)
- [Edit an Installed Package](#)
- [Delete an Installed Package](#)
- [Add a Package Repository to Your Cluster](#)
- [Edit a Custom Repository URL](#)
- [Remove a Package Repository from Your Cluster](#)
- [Disable a Package Repository in Your Cluster](#)

View Packages

See the available and installed catalog deployments for the Kubernetes clusters in your organization on the Catalog page of the Tanzu Mission Control console.

There are two tabs on the Catalog page in the Tanzu Mission Control console, the **Tanzu packages** and the **Helm charts**. The Tanzu Packages tab shows the currently installed packages and the packages that are available to be installed for each cluster in your organization. These packages include those that are included in the Tanzu Standard package repository as well as any other repositories that you have associated with a cluster. The Installed table shows a list of packages that have been installed on your clusters. The Helm charts tab shows the Helm releases that are installed and the available Helm charts.

For more information about associating additional package repositories, see [Add a Package Repository to Your Cluster](#).

Prerequisites

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- For Bitnami Helm charts, all org members can browse the charts.
- To view the available and installed packages for a cluster, you must be associated with the `cluster.view` role on that cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Catalog**.

- 2 Select the type of deployments you want to see.
 - Click the **Helm charts** tab to see available Helm charts and installed Helm releases.
 - Click the **Tanzu packages** tab to see available and installed Tanzu packages.
- 3 For Helm charts and releases:
 - Click **Available Helm charts** to see the charts that are available to deploy to your clusters.

You can optionally filter the list of charts by category or package name.

The list shows all charts that are available from the Bitnami repository. These charts are not supported by VMware.
 - Click **Installed Helm releases** to see the releases that have been deployed to your clusters.

You can optionally filter the list to locate the release you are looking for.
- 4 For Tanzu packages:
 - To see the packages that are available to be installed for each cluster, click **Available Tanzu packages**, and then select the cluster for which you want to see available packages.
 - Click **Installed Tanzu packages** to see the packages that have been deployed to your clusters.

You can optionally filter the list by cluster name or package name to locate the package you are looking for.
- 5 Alternatively, you can navigate to the detail page for a given cluster, and then click the **Add-ons** tab to see the packages and releases installed on that cluster.

Enable Helm Service on Your Cluster or Cluster Group

Enable the Helm service when you want to install Helm charts on a cluster or cluster group.

When you enable the Helm service on a cluster or cluster group, you can then create releases in your cluster from Helm charts stored in the Bitnami repository.

Note

- Tanzu Mission Control supports Helm chart deployments only from the Bitnami public repository. Adding a Helm repository to your cluster is not supported.
 - Deploying Helm charts is not supported on OpenShift clusters.
-

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To enable Helm service for a cluster or cluster group, you must be associated with the `cluster.edit` and the `clustergroup.edit` role in the cluster.

For Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu that have pod security policies, make sure you have the appropriate role bindings on the workload cluster. Without the appropriate permissions, Helm releases fail to be deployed. For information about creating role bindings, see [Example Role Bindings for Pod Security Policy](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to enable Helm service.
You can enable Helm service from the cluster detail page, if it is not already enabled, by deploying Helm charts from the catalog.
- 2 Click the **Add Ons** tab.
- 3 Click **Helm repositories** in the Add-ons menu.
- 4 Click **Enable Helm**.

What to do next

After you have enabled the Helm service on your cluster or cluster group, you can start deploying releases from Helm charts.

Disable Helm Service

You can disable the Helm service on a cluster or cluster group when you no longer need to deploy Helm charts to that cluster.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To disable Helm service, you must be associated with the `cluster.edit` and the `clustergroup.edit` roles in the cluster.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to disable Helm.
- 2 Click the **Add-ons** tab.
- 3 Click **Disable Helm** on the Actions menu.

Results

The resources are not deleted from the cluster or cluster group but are deleted from VMware Tanzu Mission Control.

Install a Helm Chart from a Helm Repository (Create a Release)

You can select from available Helm charts to install on a cluster.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To create a Helm release, you must be associated with the `cluster.edit` role.

The Helm service must already be enabled to be able to install Helm releases on a cluster. For more information, see [Enable Helm Service on Your Cluster](#).

Note Clusters running Kubernetes version 1.25 and later with pod security admission (PSA) enabled require the labels listed below in the target namespaces where your Helm release deploys pods. Tanzu Mission Control adds these labels to the default namespace (`tanzu-helm-resources`) but you must add them to any non-default namespaces where your Helm release deploys pods.

- `pod-security.kubernetes.io/audit: privileged`
- `pod-security.kubernetes.io/audit-version: latest`
- `pod-security.kubernetes.io/enforce: privileged`
- `pod-security.kubernetes.io/enforce-version: latest`
- `pod-security.kubernetes.io/warn: privileged`
- `pod-security.kubernetes.io/warn-version: latest`

For more information about pod security admission, see [Pod Security Admission](#) in the *Kubernetes documentation*.

For Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu that have pod security policies, make sure you have the appropriate role bindings on the workload cluster. Without the appropriate permissions, Helm releases fail to be deployed. For information about creating role bindings, see [Example Role Bindings for Pod Security Policy](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Procedure

- 1 Click **Catalog** in the left navigation pane.
- 2 Click the **Helm charts** tab.

3 Click **Available Helm charts**.

You can optionally filter the view of available Helm charts by selecting from the categories.

4 Click on the Helm chart you want to install.

Tanzu Mission Control displays the list of releases created for the selected Helm chart.

5 Click **Create Helm Release**.

6 Enter the required information, including the package version, a description, and the cluster.

7 Click **Create Helm Release**.

Results

The release is created and Tanzu Mission Control displays the installed Helm releases, showing the details.

Install a Helm Chart from a Git Repository

Use Tanzu Mission Control to install a Helm chart on a cluster or cluster group from a Git repository.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To install a Helm chart on a cluster, you must be associated with the `cluster.edit` role.
- To install a Helm chart on a cluster group, you must be associated with the `clustergroup.edit` role.

The Helm service must already be enabled on a cluster to be able to install a Helm chart. For more information, see [Enable Helm Service on Your Cluster or Cluster Group](#).

Note Clusters running Kubernetes version 1.25 and later with pod security admission (PSA) enabled require the labels listed below in the target namespaces where your Helm release deploys pods. Tanzu Mission Control adds these labels to the default namespace (`tanzu-helm-resources`) but you must add them to any non-default namespaces where your Helm release deploys pods.

- `pod-security.kubernetes.io/audit: privileged`
- `pod-security.kubernetes.io/audit-version: latest`
- `pod-security.kubernetes.io/enforce: privileged`
- `pod-security.kubernetes.io/enforce-version: latest`
- `pod-security.kubernetes.io/warn: privileged`
- `pod-security.kubernetes.io/warn-version: latest`

For more information about pod security admission, see [Pod Security Admission](#) in the *Kubernetes documentation*.

For Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu that have pod security policies, make sure you have the appropriate role bindings on the workload cluster. Without the appropriate permissions, Helm releases fail to be deployed. For information about creating role bindings, see [Example Role Bindings for Pod Security Policy](#) in the *vSphere with Tanzu Configuration and Management* documentation.

Procedure

- 1 Navigate to the cluster or cluster group on which you want to install a Helm chart.
- 2 Click the **Add-Ons** tab.
- 3 Click **Helm releases**.
- 4 Click **Create Helm Release**.
- 5 Select the Git repository where your Helm chart is stored.

For information on how to add a Git repository, see [Add a Git Repository to a Cluster or Cluster Group](#).

- 6 Specify the location and name of the Helm chart.
- 7 Enter a descriptive name for the release and optionally a description.
- 8 Click **Create Helm Release**.

Access Image Pull Secrets for Private Helm Charts

Accessing images stored in a private registry (authenticated image registry) from a cluster requires additional steps for the Helm release workloads to run as expected.

Prerequisites

You must have the credentials (registry URL, username and password) to access the image registry.

You have a Helm chart using a private image.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To access an image stored in a private registry, you must be associated with the `cluster.edit` role in the cluster.

Procedure

- 1 Create a `placeholder-secret.yaml` file inside the templates folder with the following content. The `.dockerconfigjson` field will be populated by the secretgen controller automatically because of the annotation we have provided.

```
apiVersion: v1
kind: Secret
metadata:
  name: {{ .Values.placeholderSecret.name }}
  namespace: {{ .Values.placeholderSecret.namespace }}
  annotations:
    secretgen.carvel.dev/image-pull-secret: ""
type: kubernetes.io/dockerconfigjson
data:
  .dockerconfigjson: e30K
```

- 2 In the `values.yaml` file add fields for the placeholder secret so that it can be populated by Helm in the above file.

```
placeholderSecret:
  name: ""
  namespace: ""
```

- 3 Also, the `values.yaml` file should contain a field for adding `imagePullSecrets` and the same should be referenced in all the deployments using images from the private registry.

In `values.yaml`:

```
imagePullSecrets: []
```

In deployments, `.spec.template.spec` should have the following:

```
spec:
  template:
    spec:
      {{- with .Values.imagePullSecrets }}
      imagePullSecrets:
        {{- toYaml . | nindent 8 }}
      {{- end }}
```

- 4 While creating helm release you need to provide the placeholder secret and `imagePullSecrets` field in `.spec.values` in the HelmRelease definition.

```
spec:
  values:
    placeholderSecret:
      name: "myplaceholdersecretname"
      namespace: "myplaceholdersecretns"
    imagePullSecrets: [{ name: myplaceholdersecretname, name: anyothersecretname }]
```

- 5 To add a registry secret on a cluster with Tanzu Mission Control, see [Managing Cluster Secrets](#)

Example Helm Release Using Private Image

This example uses a private image stored at `harbor-repo.vmware.com/tmcbuildintegrations/private-helm-poc:0.1.0`.

The Helm repository and Helm chart which are referred to below are pushed in [this](#) github repository. It already has a `placeholder-secret.yaml` file in the `templates` directory and the `values.yaml` file with `placeholderSecret` and `imagePullSecrets` field.

Prerequisites

You must have the credentials (registry URL, username and password) to access the image registry.

You have a Helm chart using a private image.

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To access an image stored in a private registry, you must be associated with the `cluster.edit` role in the cluster.

Procedure

- 1 Attach a cluster to Tanzu Mission Control.
- 2 Add a secret with harbor registry credentials and export it to all namespaces.
- 3 Enable Helm service.

- 4 Create a `flux-system` namespace.
- 5 Manually add the following resources to the cluster using `kubectl`.

helm-repo.yaml

```
apiVersion: source.toolkit.fluxcd.io/v1beta1
kind: HelmRepository
metadata:
  name: helm-chart-poc
  namespace: flux-system
spec:
  interval: 1m
  url: https://hemakshis.github.io/helm-chart-poc
```

helm-release.yaml

```
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: new-chart
  namespace: flux-system
spec:
  interval: 5m
  chart:
    spec:
      chart: new-chart
      version: '0.1.4'
      sourceRef:
        kind: HelmRepository
        name: helm-chart-poc
        namespace: flux-system
      interval: 1m
  values:
    placeholderSecret:
      name: harbor-secret-placeholder
      namespace: flux-system
    imagePullSecrets: [{ name: harbor-secret-placeholder }]
```

Edit an Installed Helm Release

You can edit installed Helm releases as needed.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To edit a Helm release, you must be associated with the `cluster.edit` role.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to edit a Helm release.
- 2 On the cluster details page, click the **Add Ons** tab.
- 3 Click **Helm releases** in the Helm section of the Add-ons menu.
- 4 Click the **Edit** button on the Helm release that you want to edit.

When editing a Helm release you can select a new package version, change the chart path, and change the description.

- 5 Click **Save**.
- 6 Optional: Change the sync interval timing for Flux CD.

Delete an Installed Helm Release

You can delete Helm releases that you no longer need for a cluster or cluster group.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To delete a Helm release, you must be associated with the `cluster.edit` and `clustergroup.edit` roles in the cluster.

Procedure

- 1 Navigate to the cluster or cluster group for which you want to delete a Helm release.
- 2 On the cluster details page, click the **Add Ons** tab.
- 3 Click **Helm releases** in the Helm section of the Add-ons menu.
- 4 Select the Helm release that you want to delete.
- 5 Click **Delete** in the Actions menu.

Install a Package

Use the Catalog page of the Tanzu Mission Control console to install a package from your repository to your Kubernetes cluster.

The Available tab on the Catalog page in the Tanzu Mission Control console shows the packages that are available to be installed, including those that are in the Tanzu Standard package repository and other repositories that you have associated with a cluster.

For more information about associating additional repositories, see [Add a Package Repository to Your Cluster](#).

Note The catalog in Tanzu Mission Control restricts the installation of packages from the Tanzu Standard package repository to only workload clusters running in Tanzu Kubernetes Grid. You cannot use the catalog to install packages from the Tanzu Standard package repository to non-TKG clusters. For non-TKG clusters, you can explore packages in the Helm charts tab. Helm charts are open source from Bitnami, and are not supported by VMware.

Note Istio packages require that you add the Istio package repository to your cluster, and must be installed in the following order:

- 1 base
 - 2 Istiod
 - 3 gateway
-

Prerequisites

This procedure assumes that you already have a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To install an available package on a cluster, you must be associated with the `.admin` role on that cluster.

Note For clusters running Kubernetes version 1.25 and later with pod security admission (PSA) enabled, you must add the labels listed below in the target namespaces where your installed package deploys pods.

- `pod-security.kubernetes.io/audit: privileged`
- `pod-security.kubernetes.io/audit-version: latest`
- `pod-security.kubernetes.io/enforce: privileged`
- `pod-security.kubernetes.io/enforce-version: latest`
- `pod-security.kubernetes.io/warn: privileged`
- `pod-security.kubernetes.io/warn-version: latest`

For more information about pod security admission, see [Pod Security Admission](#) in the *Kubernetes documentation*.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Catalog**.
- 2 On the Catalog page, click the **Tanzu packages** tab.

- 3 Click **Available Tanzu Packages**, and then select the cluster in which you want to install a package.

After you select the cluster, you see the packages that are available to install on that cluster.

- 4 Click the package you want to install.

The package detail page shows the metadata provided by the package author.

- 5 On the package detail page, click **Install Package**.

- 6 On the Install page, provide a name for the installed instance of the package and select the version to install,

- 7 You can optionally click **Carvel Settings** to specify the resources namespace for Carvel and see other configuration settings.

- **Carvel Resources Namespace**

This namespace holds the resources required to automate the installation of this package. Users with access to this namespace can access the service account which has sensitive privileges granted by the generated `ClusterRole`. You can let Tanzu Mission Control generate the name of the namespace, or optionally provide your own.

- **Service Account**

Tanzu Mission Control generates a unique service account to automate the installation of the package.

- **Role**

Tanzu Mission Control generates a unique `ClusterRole` with `admin` privileges to automate the installation of the package.

- **Role binding scope**

Tanzu Mission Control generates a unique role binding that limits the scope of the generated `ClusterRole`.

The resources that are generated for this package install are unique to this package install, and are removed when the package install is deleted.

- 8 You can optionally configure values for the keys that the package has defined as configurable.

- a To edit a configurable value in the table view, click the edit icon in the last column of the table for the configurable key.
- b Enter the new value in the box.
- c Click **Save**.

- 9 Click **Install Package**.

Edit an Installed Package

Use the Catalog page of the Tanzu Mission Control console to edit a package that you previously installed.

The Installed tab on the Catalog page in the Tanzu Mission Control console, shows the packages that have been installed on your clusters. From here, you can locate and edit a previously installed package.

Prerequisites

This procedure assumes that you have already installed a package on a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To edit an installed package on a cluster, you must be associated with the `.admin` role on that cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Catalog**.
- 2 On the Catalog page, click the **Tanzu packages** tab.
- 3 Click **Installed**, and then locate the package you want to edit.
You can sort and filter the table to help find your package.
- 4 In the table, click the Installed package name of the package you want to edit.
The package detail page shows some metadata and configured values for the package.
- 5 You can optionally reconfigure values for the keys that the package has defined as configurable.
 - a To edit a configurable value in the table view, click the edit icon for the configurable key.
 - b Enter the new value in the box.
 - c Click **Save**.
- 6 You can optionally click **Reset to package defaults** to restore the default values defined by the package.
- 7 To change the version of the package, click **Actions**, and then choose **Change Desired Version**.
Then in the confirmation dialog, choose the version and click **Save**.
- 8 You can optionally click **Carvel Settings** to see the resources namespace for Carvel and other configuration settings.
The configuration setting for Carvel cannot be edited after the package is installed.
- 9 After you have finished making modifications, click **Apply Changes**.

Delete an Installed Package

Use the Tanzu Mission Control console to delete a package that you previously installed.

From the Installed tab on the Catalog page in the Tanzu Mission Control console, you can locate and delete a previously installed package.

Prerequisites

This procedure assumes that you have already installed a package on a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To delete an installed package on a cluster, you must be associated with the `.admin` role on that cluster.

Procedure

1 In the left navigation pane of the Tanzu Mission Control console, click **Catalog**.

2 On the Catalog page, click the **Tanzu packages** tab.

3 Click **Installed**, and then locate the package you want to delete.

You can sort and filter the table to help find your package.

4 In the table, click the Installed package name of the package you want to delete.

The package detail page shows some metadata and configured values for the package.

5 On the package detail page, click **Actions**, and then choose **Delete**.

6 In the confirmation dialog, type in the name of the package, and then click **Delete**.

Results

When you click , Tanzu Mission Control removes the installed package from your cluster, along with the following resources that were created during installation of the package:

- namespace
If you entered a custom namespace during installation, the namespace is not deleted.
- service account
- role
- role binding

Add a Package Repository to Your Cluster

Use Tanzu Mission Control to add a repository to a managed cluster to enable management of package installations.

When you attach a cluster in Tanzu Mission Control, the Tanzu Standard package repository is available to the cluster by default. If you have additional repositories from which you pull packages, use this procedure to make those repositories available to your cluster. When you add a repository to a cluster, the packages in the repository can be installed in that cluster using the Catalog page in the Tanzu Mission Control console.

Note To deploy the Tanzu-verified Istio packages to your clusters, use the following repository URL:

```
extensions.aws-usw2.tmc-dev.cloud.vmware.com/packages/istio-oss-packages:1.22.0-tanzu.1
```

Prerequisites

This procedure assumes that you have a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To add a repository to your cluster, you must be associated with the `.admin` role on that cluster.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster detail page of the cluster to which you want to add the repository.

- 2 Click the **Add-ons** tab, and then click **Repositories**.

The table on this page shows the package repositories that have been added to the cluster.

- 3 Click **Add Package Repository**.

- 4 Provide a name for the repository and the URL at which it resides, and then click **Add Repository**.

Edit a Custom Repository URL

Use Tanzu Mission Control to modify the URL of a package repository that you have added to a cluster.

When you add a repository to a cluster, the packages in the repository can be installed in that cluster using the Catalog page in the Tanzu Mission Control console. If the URL for that repository changes, use this procedure to update it.

Note The Tanzu standard package repository cannot be edited, but you can disable it if necessary. For more information, see [Disable a Package Repository in Your Cluster](#).

Prerequisites

This procedure assumes that you have already added a package repository to a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To edit a package repository in your cluster, you must be associated with the `.admin` role on that cluster.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster detail page of the cluster to which you added the repository.
- 2 Click the **Add-ons** tab, and then click **Repositories**.
The table on this page shows the package repositories that have been added to the cluster.
- 3 Click the menu icon for the repository that you want to update, and then choose **Edit**.
- 4 Provide the new URL for the repository, and then click **Save**.

Remove a Package Repository from Your Cluster

Use Tanzu Mission Control to remove a package repository that you have added to a cluster.

When you add a repository to a cluster, the packages in the repository can be installed in that cluster using the Catalog page in the Tanzu Mission Control console. When a package repository is no longer needed for a cluster, use this procedure to remove it.

When you remove a package repository from a cluster, you can no longer install packages or update existing installations from the repository.

Note The Tanzu standard package repository cannot be removed, but you can disable it if necessary. For more information, see [Disable a Package Repository in Your Cluster](#).

Prerequisites

This procedure assumes that you have already added a package repository to a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To remove a package repository from your cluster, you must be associated with the `.admin` role on that cluster.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster detail page of the cluster to which you added the repository.

- 2 Click the **Add-ons** tab, and then click **Repositories**.

The table on this page shows the package repositories that have been added to the cluster.

- 3 Click the menu icon for the repository that you want to remove, and then choose **Remove**.
- 4 In the confirmation dialog, click **Remove**.

Disable a Package Repository in Your Cluster

Use Tanzu Mission Control to disable a package repository in your cluster.

When you add a repository to a cluster, the packages in the repository can be installed in that cluster using the Catalog page in the Tanzu Mission Control console. When a package repository is no longer needed for a cluster, you can use this procedure to disable it.

When you disable a package repository in a cluster, you can no longer install packages or update existing installations from the repository. The repository and its contents are removed from the cluster, but the name and URL for the repository remain in the cluster in case you want to enable it later.

Prerequisites

This procedure assumes that you have already added a package repository to a Kubernetes cluster that is managed (either attached or provisioned) by Tanzu Mission Control.

Before you start this procedure, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To disable a package repository in your cluster, you must be associated with the `.admin` role on that cluster.

Procedure

- 1 In the Tanzu Mission Control console, navigate to the cluster detail page of the cluster to that contains the repository you want to disable.
- 2 Click the **Add-ons** tab, and then click **Repositories**.
- 3 Click the menu icon for the repository that you want to disable, and then choose **Disable**.
- 4 In the confirmation dialog, click **Disable**.

Install Tanzu Application Platform on a Cluster

18

Use Tanzu Mission Control to install Tanzu Application Platform (commonly called TAP) on a managed cluster.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To install Tanzu Application Platform (TAP) on a cluster, you must be associated with the `cluster.edit` role.

Make sure your cluster's configuration meets the TAP system requirements specified in [Prerequisites for installing Tanzu Application Platform](#) in the *VMware Tanzu Application Platform Documentation*.

Note

- If you have already enabled continuous delivery (CD) or Helm for a cluster, the latest verified version of FluxCD is deployed to the cluster. Because TAP uses an earlier version of FluxCD, and is therefore incompatible with the deployed version, you must first disable CD/Helm on the cluster before attempting to install TAP. If you attempt to install TAP on a cluster where CD/Helm is already enabled, you see an error that looks something like this:

```
Solution cannot be installed due to incompatible Continuous Delivery feature components on cluster(s)
```

This applies to individual clusters on which you enable CD or Helm, as well as clusters that belong to a cluster group on which you enable CD or Helm. Therefore, if you have enabled CD/Helm for a cluster group that contains one or more clusters on which you want to install TAP, then you must disable CD/Helm for the cluster group before attempting to install TAP on clusters in that cluster group.

In addition to disabling CD/Helm, you must make sure that the FluxCD package install resources are removed from all clusters, as described in [Disable Continuous Delivery](#).

If you install TAP prior to enabling CD/Helm, TMC uses the version of FluxCD installed by TAP.

- Tanzu Mission Control does not deploy TAP on the cluster if TAP packages are already present on the cluster. If you deploy TAP on a cluster through Tanzu Mission Control and then subsequently detach and re-attach the cluster, you must manually remove TAP packages on the cluster before installing TAP again from the Tanzu Mission Control console.
- There is a known issue related to Tanzu Application Platform TAP-GUI version 1.7.7/1.7.6 where installation fails on Openshift as part of the View profile.
- There is a known issue that when installing Tanzu Application Platform version 1.7.1 using Tanzu Mission Control on a multi-cluster environment, TAP does not receive ingress location values (secrets with authentication and CA data, and the ingress URL).

Use the Tanzu Mission Control console to update the build cluster configuration to point to the synced secrets and endpoint location found in the namespace `metadata-store-secrets`.

```
grype:
  metadataStore:
    authSecret:
      importFromNamespace: metadata-store-secrets
      name: store-auth-token
    caSecret:
      importFromNamespace: metadata-store-secrets
      name: store-ca-cert
    url: https://metadata-store.<user-supplied-domain-for-view-cluster>
```

Procedure

- 1 In the Tanzu Mission Control console, click **Catalog** in the left navigation pane.
- 2 Click on the **Solutions** tab, and then click on **Install Tanzu Application Platform** selector.
- 3 Select the **Single Cluster** tile, and then click the **Install Tanzu Application Platform** button.
- 4 Enter a name for your TAP deployment.
You can optionally provide a description.
- 5 Click **Next**.
- 6 Configure the profile parameters.

For a single-cluster deployment, you can install only the `Full` profile. Use the multi-cluster deployment to use a different profile, as described in [Install Tanzu Application Platform on Multiple Clusters](#).

- a Select the version of TAP that you want to install on the cluster.
- b Click **Select Cluster** to select the cluster on which to install TAP.
- c Specify the ingress domain. To access the Tanzu Developer Portal and deployed workloads using a public URL, you must specify an ingress domain. You should create the DNS record after completing the TAP installation.
- d Specify the name of the supply chain.

For more information about supply chains, see [Overview of Supply Chain Choreographer for Tanzu](#) in the *VMware Tanzu Application Platform Documentation*.

- e Specify the parameters for the image registry provider, including the secret name, the project path, and the secret namespace. These credentials are for pushing and pulling application images.

You can optionally click **Create Secret** to configure a new registry secret or opaque secret.

- f Select the namespace provisioner, either Controller or GitOps, and then configure the parameters for the selected provisioner.

For more information about the namespace provisioner, see [Overview of Namespace Provisioner](#) in the *VMware Tanzu Application Platform Documentation*.

- g Enter your entitlement account number.

The entitlement account number (EAN) is a unique identifier that VMware assigns to its customers.

- h By default, guest login is enabled.

By default, unauthenticated guest user access is enabled to the Tanzu Developer Portal. You can optionally disable guest access and configure an alternative authentication provider. For more information, see [Set up authentication for Tanzu Developer Portal](#) in the *VMware Tanzu Application Platform Documentation*.

- 7 Optionally, you can expand the **Advanced Configuration** section to configure parameters in the configuration form or YAML.

Note To install the Full profile with all build dependencies, add the following section to the YAML configuration.

```
buildservice:
  exclude_dependencies: true
```

- 8 Click **View License Agreement** to read the agreement, then click **Close and Accept** to agree to the license terms.

You must accept the license agreement to continue with the installation. You must also enable the option for inclusion in the Customer Experience Improvement Program for the TAP installation to proceed.

- 9 Click **Install TAP**.

Results

When you click **Install TAP**, Tanzu Mission Control installs Tanzu Application Platform on the cluster. Because the TAP installation also installs packages on the target cluster, it takes some time to complete the installation.

Note When installing or upgrading a Tanzu Application Platform (TAP) deployment on your cluster, you might encounter an intermediate state of installation failure with one or more packages. This situation can be safely ignored, as it is only temporary. After a short period of time, these issues are reconciled to a state of installation success.

What to do next

To view information about the installation, go to the Solutions tab on the Catalog page, and then click **Installed Tanzu Application Platforms**. From this page, you can do the following:

- You can access the developer portal. Click **Catalog**, then click **Installed Tanzu Application Platforms**, select a TAP installation, then click the link to the developer portal.
- You can verify the profile details and status of all packages installed. Click **Catalog**, then click **Installed Tanzu Application Platforms**, then select a TAP installation. To get help for issues with installing packages, see [Tanzu Application Platform Packages Are Not Installed](#) in the Troubleshooting section.
- You can create developer namespaces. Select **Catalog**, then click **Installed Tanzu Application Platforms**, select a TAP installation, and then click **Create Developer Namespace**. For more information about namespaces, see [Set up namespaces](#) in the *VMware Tanzu Application Platform Documentation*.
- You can [Edit a Tanzu Application Platform Configuration](#).
- You can [Delete a Tanzu Application Platform Configuration](#).

To learn more about Tanzu Application Platform, see the latest version of the [VMware Tanzu Application Platform Documentation](#).

Install Tanzu Application Platform on Multiple Clusters

Use Tanzu Mission Control to install Tanzu Application Platform (commonly called TAP) on multiple managed clusters.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To install Tanzu Application Platform (TAP) on a cluster, you must be associated with the `cluster.edit` role.

Make sure your configuration meets the TAP system requirements specified in [Prerequisites for installing Tanzu Application Platform](#) in the *VMware Tanzu Application Platform Documentation*.

Note

- If you have already enabled continuous delivery (CD) or Helm for a cluster, the latest verified version of FluxCD is deployed to the cluster. Because TAP uses an earlier version of FluxCD, and is therefore incompatible with the deployed version, you must first disable CD/Helm on the cluster before attempting to install TAP. If you attempt to install TAP on a cluster where CD/Helm is already enabled, you see an error that looks something like this:

```
Solution cannot be installed due to incompatible Continuous Delivery feature components on cluster(s)
```

This applies to individual clusters on which you enable CD or Helm, as well as clusters that belong to a cluster group on which you enable CD or Helm. Therefore, if you have enabled CD/Helm for a cluster group that contains one or more clusters on which you want to install TAP, then you must disable CD/Helm for the cluster group before attempting to install TAP on clusters in that cluster group.

In addition to disabling CD/Helm, you must make sure that the FluxCD package install resources are removed from all clusters, as described in [Disable Continuous Delivery](#).

If you install TAP prior to enabling CD/Helm, TMC uses the version of FluxCD installed by TAP.

- Tanzu Mission Control does not deploy TAP on the cluster if TAP packages are already present on the cluster. If you deploy TAP on a cluster through Tanzu Mission Control and then subsequently detach and re-attach the cluster, you must manually remove TAP packages on the cluster before installing TAP again from the Tanzu Mission Control console.
- There is a known issue related to Tanzu Application Platform TAP-GUI version 1.7.7/1.7.6 where installation fails on Openshift as part of the View profile.
- There is a known issue that when installing Tanzu Application Platform version 1.7.1 using Tanzu Mission Control on a multi-cluster environment, TAP does not receive ingress location values (secrets with authentication and CA data, and the ingress URL).

Use the Tanzu Mission Control console to update the build cluster configuration to point to the synced secrets and endpoint location found in the namespace ``metadata-store-secrets``.

```
grype:
  metadataStore:
    authSecret:
      importFromNamespace: metadata-store-secrets
      name: store-auth-token
    caSecret:
      importFromNamespace: metadata-store-secrets
      name: store-ca-cert
    url: https://metadata-store.<user-supplied-domain-for-view-cluster>
```

Procedure

- 1 In the Tanzu Mission Control console, click **Catalog** in the left navigation pane.
- 2 Click on the **Solutions** tab, and then click on **Install Tanzu Application Platform** selector.
- 3 Click the **Multi-Cluster** tile, and then click the **Install Tanzu Application Platform** button.
- 4 Enter a name for your TAP deployment.
You can optionally provide a description.
- 5 Click **Next**.
- 6 Select the version of TAP that you want to install on the clusters.
- 7 Select and configure the View profile, click on the **Profile View**.
 - a Click to expand the View profile configuration options.
 - b Click **Select Cluster** to select the cluster on which you want to install TAP with the View profile.
 - c Specify the Ingress domain.
To access the Tanzu Developer Portal and deployed workloads using a public URL, you must specify an ingress domain. You should create the DNS record after completing the TAP installation.
 - d Enter your entitlement account number.
The entitlement account number (EAN) is a unique identifier that VMware assigns to customers.
 - e You can optionally enable guest login.
- 8 Select and configure the Run profile.
 - a Click **Select Cluster** to select the cluster on which to install TAP with the Run profile.
 - b Specify the Ingress domain. To access the Tanzu Developer Portal and deployed workloads using a public URL, you must specify an ingress domain. You should create the DNS record after completing the TAP installation.
 - c Optionally you can expand the Configure the Git repository section and change the default settings.
 - d Enter your entitlement account number.
The entitlement account number (EAN) is a unique identifier that VMware assigns to customers.
 - e The Cluster URL is populated when you select the cluster. You can change it if needed.
 - f The CA certificate is populated when you select the cluster. You can change it if needed.

9 Select and configure the Build profile.

- a Click **Select Cluster** to select the cluster on which to install TAP with the Build profile.
- b Specify the name of the Supply chain.

For more information about supply chains, see [Overview of Supply Chain Choreographer for Tanzu](#) in the *VMware Tanzu Application Platform Documentation*.

- c Specify the parameters for the Image registry provider, including the secret name, the project path, and the secret namespace. These credentials are for pushing and pulling application images.

You can click **Create Secret** to configure a new password and new registry configuration.

- d Select the Namespace provisioner, either Controller or GitOps, and then configure the parameters for the selected provisioner.

For more information about Namespace provisioner, see [Overview of Namespace Provisioner](#) in the *VMware Tanzu Application Platform Documentation*.

- e Enter your entitlement account number.

The entitlement account number (EAN) is a unique identifier that VMware assigns to customers.

- f You can optionally expand the Configure the Git repository section and change the default settings.
- g The Cluster URL is populated when you select the cluster. You can change it if needed.
- h The CA certificate is populated when you select the cluster. You can change it if needed.
- i You can optionally expand the Advanced Configuration section of the page to configure parameters in the configuration form or YAML.

Note To install the Build profile with all build dependencies, add the following section to the YAML configuration.

```
buildservice:
  exclude_dependencies: true
```

10 Select and configure the Iterate profile.

- a Click **Select Cluster** to select the cluster on which to install TAP with the Iterate profile.
- b Specify the ingress domain.

To access the Tanzu Developer Portal and deployed workloads using a public URL, you must specify an ingress domain. Create the DNS record after completing the TAP installation.

- c Specify the name of the supply chain.

For more information about supply chains, see [Overview of Supply Chain Choreographer for Tanzu](#) in the *VMware Tanzu Application Platform Documentation*.

- d Specify the parameters for the Image registry provider, including the secret name, the project path, and the secret namespace. These credentials are for pushing and pulling application images.

You can optionally click **Create Secret** to configure a new registry secret or opaque secret.

- e Select the Namespace provisioner, either Controller or GitOps, and then configure the parameters for the selected provisioner.

For more information about Namespace provisioner, see [Overview of Namespace Provisioner](#) in the *VMware Tanzu Application Platform Documentation*.

- f You can optionally expand the Configure the Git repository section and change the default settings.

- g The Cluster URL is populated when you select the cluster. You can change it if needed.

- h The CA certificate is populated when you select the cluster. You can change it if needed.

- i Optionally, you can expand the Advanced Configuration section of the page to configure parameters in the configuration form or YAML.

Note To install the Iterate profile with all build dependencies, add the following section to the YAML configuration.

```
buildservice:
  exclude_dependencies: true
```

- 11 Click **Add Profile** to install a TAP profile on another cluster.
- 12 Click **Done** when finished configuring profiles on clusters.
- 13 Click **View License Agreement** to read the agreement, then click **Close and Accept** to agree to the license terms.

You must accept the VMware license agreement to continue with the installation. The option for inclusion in the Customer Experience Improvement Program must also be enabled for the TAP installation to proceed.

- 14 Click **Install TAP**.

Results

When you click **Install TAP**, Tanzu Mission Control installs Tanzu Application Platform and the configured profiles on the clusters that you specified.

Note When installing or upgrading a Tanzu Application Platform (TAP) deployment on your cluster, you might encounter an intermediate state of installation failure with one or more packages. This situation can be safely ignored, as it is only temporary. After a short period of time, these issues are reconciled to a state of installation success.

What to do next

To view information about the installations, go to the Solutions tab on the Catalog page, and then click **Installed Tanzu Application Platforms**. From this page, you can do the following:

- Verify the profile details and status of all packages installed.
- Edit the configuration of a TAP installation.
- Delete a TAP installation.

To learn more about Tanzu Application Platform, see the latest version of the [VMware Tanzu Application Platform Documentation](#).

Edit a Tanzu Application Platform Configuration

You can edit a Tanzu Application Platform (commonly called TAP) configuration on a Tanzu Mission Control managed cluster.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To edit Tanzu Application Platform (TAP) on a cluster, you must be associated with the `cluster.edit`.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Catalog**.
- 2 On the Solutions tab of the Catalog page, click **Installed Tanzu Application Platforms** to view currently installed TAP instances.

- 3 Click the menu icon next to the installation that you want to edit, and then choose **Edit**.

Note You can click on a specific installed TAP name to view details about that installation. On the TAP installation details page, you can click **Create Developer Namespace** to create a new developer namespace. The Developer namespaces section of the page shows existing ones.

When you click **Edit**, the Edit Tanzu Application Platform page lets you edit many aspects of the installation, similar to the installation page.

Note

- You cannot change the cluster on which it is installed.
 - If you update the TAP version, you can skip only one minor version between the currently installed version and the new version. For example, you can upgrade from version 1.7.x to 1.9.x, but you cannot upgrade directly from version 1.7.x to 1.10.x.
-

- 4 After you have finished editing your TAP deployment, click **Update**.

Results

When you click **Update**, Tanzu Mission Control pushes your edits to the TAP deployment on your cluster.

Note When installing or upgrading a Tanzu Application Platform (TAP) deployment on your cluster, you might encounter an intermediate state of installation failure with one or more packages. This situation can be safely ignored, as it is only temporary. After a short period of time, these issues are reconciled to a state of installation success.

Delete a Tanzu Application Platform Configuration

You can delete Tanzu Application Platform (commonly called TAP) from a Tanzu Mission Control managed cluster.

You can delete a TAP installation from a cluster when you no longer need it.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions.

- To delete Tanzu Application Platform (TAP) from a cluster, you must be associated with the `cluster.edit`.

For more information about roles and permissions in Tanzu Mission Control, see [Access Control](#) and [Users and Groups](#) in *VMware Tanzu Mission Control Concepts*.

Procedure

- 1 Select **Catalog**.
- 2 Select the **Solutions** tab.
- 3 Click **Installed Tanzu Application Platforms** to view currently installed TAP instances.
- 4 Click the menu next to the installation that you want to delete and select **Delete**.

Results

The selected TAP installation is removed from the cluster.

As a platform operator or infrastructure operator, use VMware Tanzu Mission Control to set up and manage data protection for the resources in your clusters.

Through Tanzu Mission Control, you can run backup and restore operations to protect your data. For more information about protecting the data resources in your Kubernetes clusters, see [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

Note

- The data protection features of Tanzu Mission Control are not compatible with clusters created using the Tanzu Kubernetes release `v1.23.8---vmware.2-tkg.1-zshippable`. If you rely on Tanzu Mission Control for data protection, use a different Tanzu Kubernetes release when creating your clusters.
- The data protection features of Tanzu Mission Control are not available in Tanzu Mission Control Essentials.

Read the following topics next:

- [Create a Target Location for Data Protection](#)
- [Delete a Target Location](#)
- [Enable Data Protection for a Cluster](#)
- [Disable Data Protection for a Cluster](#)
- [Enable Data Protection on a Cluster Group](#)
- [Define the Backup Schedule for a Cluster Group](#)
- [Requirements for CSI Volume Backup](#)
- [Back Up the Data Resources in Your Cluster](#)
- [View the Contents of a Backup](#)
- [Restore a Backup](#)
- [Restore a Backup from a Different Cluster](#)
- [View the Contents of a Restore](#)
- [About Backup and Restore Hooks](#)

Create a Target Location for Data Protection

As platform operator, you can create a data protection target location that you can use for storage of backups that you generate using VMware Tanzu Mission Control.

When you run a backup using Tanzu Mission Control, the resources that you specify to be backed up are written to a storage location that you identify. This location can be the AWS S3 storage that is managed by Tanzu Mission Control in your cloud provider account, or a storage location that you create and maintain in your cloud provider account (AWS S3 or S3-compatible storage or Azure Blob storage). This procedure shows how to create a target location that you can use for backups.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure that you have already created a data protection credential that provides the connection to your cloud provider account.

- For AWS S3 storage that is managed by Tanzu Mission Control, see [Connect an AWS Account for Data Protection](#).
- For Azure Blob storage or AWS S3 or S3-compatible storage, see [Create a Data Protection Credential for Self-Provisioned Storage](#).

Make sure that you have the appropriate permissions in your Tanzu Mission Control organization.

- To create a target location, you must be associated with the `organization.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the Administration page, click the **Target Locations** tab.
- 3 Click **Create Target Location**, and then choose the type of storage for the new target location.
 - TMC provisioned storage: **AWS S3**
 - Self provisioned storage: **AWS S3 or S3-compatible**
 - Self provisioned storage: **Azure Blob**
- 4 Select an account credential, and then click **Next**.
- 5 If you choose self-provisioned storage, specify the configuration of the storage.

If you choose the storage managed by Tanzu Mission Control, you can skip this step.

For AWS S3 or S3-compatible:

- a Enter the URL that identifies the AWS S3 or S3-compatible storage location.

Note Since Tanzu Mission Control supports actual S3 as well as any custom backend that is S3-compatible, the URLs can be in one of the following formats:

- <https://s3.us-west-2.amazonaws.com>
 - <https://111.222.333.444:9000>
-

- b Enter a name for the bucket in which to store backups.
- c Specify the region in which to store backups.
- d If your storage location uses a custom root certificate or CA certificate, then enter the certificate in the text box.

For Azure Blob:

- a Enter the account ID for the account where the Azure Blob resides.
- b Enter the container name for the Azure Blob.

For more information about setting up your Azure account for use with Tanzu Mission Control data protection, see [Account Setup for Azure Blob Target Location](#) .

After you have specified the configuration for the storage provider, click **Next**.

- 6 Specify the clusters that can use this target location for backup.

You can specify cluster groups as well as individual clusters for the target location.

- To specify cluster groups, click **Select Cluster Groups**.
- To specify individual clusters, click **Select Clusters**.

- 7 Click **Next**.
- 8 Provide a name for the target location.
- 9 Click **Create**.

Results

When you click **Create**, Tanzu Mission Control generates a backup location that can be used by the specified clusters.

Account Setup for Azure Blob Target Location

Learn how to set up your subscription in Microsoft Azure so that you can use it as a target location for data protection in Tanzu Mission Control.

To use Tanzu Mission Control for storing and retrieving backups in Azure Blob storage, you need the following items:

- An Azure storage account and a blob container in your Azure subscription.

- An Azure service principal identity that allows Velero to authenticate to the subscription where you want to store backups. You must create the service principal with Azure's built-in `Contributor` role. The service principal requires the following details:
 - Azure Subscription ID
 - Azure Tenant ID
 - Azure Client ID
 - Azure Client Secret

The sections in this topic show the Azure CLI (`az`) commands that you can use to create an Azure storage account and blob container, as well as the service principal. For information on how to install the latest Azure CLI, see [How to install the Azure CLI](#) in the Azure documentation.

Log in to your Azure subscription and set the context

- 1 Log in to your Azure subscription.

```
az login
```

- 2 Set variables.

Replace `<name-of-target-subscription>` with the name of your subscription.

```
AZURE_BACKUP_SUBSCRIPTION_NAME=<name-of-target-subscription>
AZURE_BACKUP_SUBSCRIPTION_ID=$(az account list --query="[?
name=='$AZURE_BACKUP_SUBSCRIPTION_NAME'].id | [0]" -o tsv)
```

- 3 Set your account.

```
az account set -s $AZURE_BACKUP_SUBSCRIPTION_ID
```

Create an Azure storage account and a blob container

If you already have an Azure storage account and blob container in your Azure subscription where you want store backups, you can use that and skip this section.

The following steps guide you through the creation of an Azure storage account and blob container in your Azure subscription, using Azure CLI (`az`) commands.

- 1 Log in to your Azure subscription, as described above.
- 2 Create a resource group and storage account.

The storage account can be created in an existing resource group or separated into its own resource group.

The sample script below shows how to generate a random name using `uuidgen`. You can name the account as you want, provided it is globally unique and follows the [Azure naming rules for storage accounts](#).

- a Create a resource group for the backup storage account.

The following commands create a resource group named *Velero_Backups*.

```
AZURE_BACKUP_RESOURCE_GROUP=Velero_Backups
az group create -n $AZURE_BACKUP_RESOURCE_GROUP --location WestUS
```

- b Create the storage account in the resource group.

The storage account must be created with a globally unique ID because it is used for DNS. The following commands generate an account ID prepended with *mybackups* and then create a storage account with that ID. Refer to [Azure Storage account](#) for more details.

```
AZURE_STORAGE_ACCOUNT_ID="mybackups$(uuidgen | cut -d '-' -f5 | tr 'A-Z' 'a-z')"
az storage account create --name $AZURE_STORAGE_ACCOUNT_ID --resource-group
$AZURE_BACKUP_RESOURCE_GROUP --sku Standard_GRS --encryption-services blob --https-
only true --kind BlobStorage --access-tier Hot
```

- c Create the blob container.

The commands below create a blob container named *myblobcontainer* in the storage account you created in the previous step. You can use a different name, preferably unique to a single Kubernetes cluster. Refer to [Azure Storage container](#) for more details.

```
BLOB_CONTAINER= myblobcontainer
az storage container create -n $BLOB_CONTAINER --public-access off --account-name
$AZURE_STORAGE_ACCOUNT_ID
```

Create a service principal

When you create and restore backups through Tanzu Mission Control, Velero requires a service principal identity to authenticate to the Azure subscription where backups are stored. If you already have an existing service principal with the appropriate permissions, you can use that and skip this section.

The following steps guide you through the process of creating a service principal using the built-in *Contributor* role, which has the permissions required by Velero to access the blob container in your subscription. For more information about the service principal identity and how to create it, refer to the following topics in the Azure documentation.

- [Application and service principal objects in Azure Active Directory](#)
- [Use the portal to create an Azure AD application and service principal that can access resources](#)

Use the following procedure to create a service principal with the *Contributor* role.

- 1 Log in to your Azure subscription, as described above.
- 2 Obtain your Azure account tenant ID.

```
AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv`
```

- 3 Create the service principal and let the CLI generate a password for you. Make sure to capture the generated password.

This example uses *velero* for the user name. You can create a unique user name per cluster rather than *velero*. Make sure that the user name value does not conflict with other service principals or app registrations.

```
AZURE_ROLE=Contributor
AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" --role $AZURE_ROLE --query 'password' -o tsv --scopes /subscriptions/$AZURE_BACKUP_SUBSCRIPTION_ID`
```

- 4 After creating the service principal, obtain the client ID.

```
AZURE_CLIENT_ID=`az ad sp list --display-name "velero" --query '[0].appId' -o tsv`
```

After you have completed these procedures, you have a storage account with a blob container and a service principal, as well as the information necessary to use your Azure subscription for a target location in Tanzu Mission Control.

For more information about using Velero with Microsoft Azure, go to <https://github.com/vmware-tanzu/velero-plugin-for-microsoft-azure#velero-plugins-for-microsoft-azure>.

Delete a Target Location

As platform operator, you can remove a data protection target location that you previously created.

During the course of operations, you might find it necessary to remove a target location. Because the target location is simply a reference to a storage resource, deleting the target location does not remove the data protection credential or the underlying storage resource. However, if you delete a target location, it is no longer accessible through Tanzu Mission Control to perform backups or restores.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure that you have the appropriate permissions in your Tanzu Mission Control organization.

- To delete a target location, you must be associated with the `organization.admin` role.

Procedure

- 1 In the Tanzu Mission Control console, click **Administration** in the left navigation pane.
- 2 On the Administration a page, click the **Target Locations** tab.
- 3 In the list of target locations, locate the target location that you want to delete.
- 4 Click the menu icon next to the target location that you want to delete, and then choose **Delete**.
- 5 In the confirmation dialog, click **Delete**.

Enable Data Protection for a Cluster

Use VMware Tanzu Mission Control to enable data protection capabilities, like backup and restore, on your attached or provisioned clusters.

This procedure describes how to install the data protection extension (and Velero) on your cluster so that you can use Tanzu Mission Control to perform data protection actions.

Note This procedure does not create a schedule or start a backup. It installs the components on your cluster that allow you to perform those actions. You can subsequently define a backup schedule and run a backup after you have enabled data protection on your cluster.

Prerequisites

This procedure assumes that you have a managed cluster, either attached or provisioned, and that it belongs to a cluster group that is associated with a target location, as described in [Create a Target Location for Data Protection](#).

Make sure you have the appropriate permissions.

- To enable data protection, you must be associated with the `cluster.admin` role in the cluster.
- To see and use a cloud provider account connection for data protection, you must be associated with the `organization.credential.view` role.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster for which you want to enable data protection.
- 2 In the Data Protection box, click **Enable Data Protection**.
- 3 In the Enable Data Protection dialog, select the volume backup type that you want to use.

File system backup (FSB) is selected by default. You can select FSB or CSI (container storage interface) or both.

- If both methods are selected, you can choose either method when creating a backup schedule for the cluster.
- If neither method is selected, you cannot backup or restore volume snapshots on the cluster.

You can also choose to enable data protection without a volume backup method, and then enable it later if necessary. For more information about volume backup methods, see [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

- 4 Click **Enable**.

Results

When you click **Enable**, Tanzu Mission Control installs the data protection extension, the selected plug-ins, and Velero on your cluster. After it is installed, the `data-protection` extension is listed on the cluster detail page along with its status.

What to do next

After you have enabled data protection for a cluster, you can perform actions like backing up and restoring data resources in your cluster.

Disable Data Protection for a Cluster

Use VMware Tanzu Mission Control to disable data protection and remove Velero from your cluster.

If you decide that you no longer need to back up data resources in a given cluster using Tanzu Mission Control, you can remove the data protection extension and Velero from your cluster.

Although this procedure disables your ability to perform backup and restore actions, disabling data protection does not, by default, remove the backups that you have already created. Existing backup files remain in the storage location identified by the data protection credential unless you explicitly choose to delete them.

Prerequisites

Make sure you have the appropriate permissions.

- To disable data protection, you must be associated with the `cluster.admin` role in the cluster.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster for which you want to disable data protection.
- 2 At the top of the cluster detail page, click **Actions**, and then choose **Disable data protection**.
- 3 In the confirmation dialog, you can optionally choose to permanently delete the cluster's existing backup files.

If you select the **Destroy all backup files associated with this cluster** option, Tanzu Mission Control permanently deletes the existing backup files. If this option is not selected, the existing backup files remain in the storage location identified by the data protection credential.

- 4 Click **Disable**.

Results

When you click **Disable**, Tanzu Mission Control removes the data protection extension and Velero from your cluster. After you disable data protection for a cluster, you can no longer perform actions like backing up and restoring data resources in your cluster.

Enable Data Protection on a Cluster Group

You can use the Tanzu Mission Control data protection features to create backups of clusters and cluster groups.

Prerequisites

Make sure you have the appropriate permissions.

- To configure data protection for cluster groups, you must be associated with the `clustergroup.admin` role in the cluster.

Procedure

- 1 Log in to the Tanzu Mission Control console and click Cluster Groups.
- 2 Click on the cluster group for which you want to enable data protection.
- 3 Click **Enable Data Protection**.

The Enable data protection dialog appears. The table at the top of the dialog lists any clusters that have data protection already enabled on an individual basis and shows the current status of the settings for each enabled cluster.
- 4 Select the group clusters and volume backup settings for the cluster group data protection configuration.
 - a Select the clusters for which you want to enable data protection. You can select all clusters in the group, clusters based on names, or groups based on Tanzu Mission Control labels.
 - b Select the volume backup plugins settings, FSB or CSI. For more information on volume settings, see [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.
- 5 Review your settings in the Preview changes section at the bottom of the dialog.
- 6 Click **Enable**.

Results

The selected clusters are configured for data protection and the **Data Protection** tab is added to the Cluster Group details page. From there you can view settings and configure backup schedules.

What to do next

Create backup schedules for clusters and cluster groups.

Define the Backup Schedule for a Cluster Group

You can create schedules for the backup of cluster groups using Tanzu Mission Control.

Prerequisites

Make sure you have the appropriate permissions.

- To schedule data protection backups for cluster groups, you must be associated with the `clustergroup.admin` role in the cluster.

Note You can have a maximum of 720 backups per backup schedule at any given point in time. This capacity is influenced by both the frequency and the retention period of the backup schedule. Therefore, if your schedule requires a high frequency then you might need to reduce the retention period.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the Cluster Group details page to create the backup schedule.

- 2 Click the **Data Protection** tab.

- 3 Click **Create Backup Schedule** in the Scheduling section of the page.

The Create Backup Schedule page appears. This page shows the current backup configuration status of clusters in the group and lets you configure the schedule to include or exclude clusters.

- 4 In section one of the page select the clusters you want to schedule. You can select all clusters in the group, clusters based on names, or clusters based on Tanzu Mission Control labels.

- 5 Click **Refresh** to preview the settings, then click **Next**.

- 6 In section two of the page, configure the Kubernetes resources to back up.

You can choose to back up the entire cluster, selected namespaces, or specify by label.

Note For more information about this and the following steps, see [Back Up the Data Resources in Your Cluster](#).

- 7 Click **Next**.

- 8 Specify the volume backup method and click **Next**.

- 9 Specify where to store the backup and click **Next**.

- 10 Configure the backup schedule and click **Next**.

- 11 Specify the backup retention period.

- 12 Enter a name for the backup schedule and click **Create**

Results

The cluster group backup schedule is created and ready to run as configured. You can navigate to the cluster group data protection tab or to an individual cluster data protection tab to verify the status of backup schedules and a list of backups.

Requirements for CSI Volume Backup

Be sure to meet these requirements when configuring data protection in VMware Tanzu Mission Control.

Requirements for CSI Volume Backup

- Your cluster must be Kubernetes version 1.20 or greater.
- Your cluster must be running a CSI driver capable of supporting volume snapshots at the [v1 API level](#).
- Your cluster must be installed with the `snapshot.storage.k8s.io/v1` API group. `volumesnapshots.snapshot.storage.k8s.io`, `volumesnapshotcontents.snapshot.storage.k8s.io`, and `volumesnapshotclass.snapshot.storage.k8s.io` CRDs should be present on the cluster. You should see output similar to that below:

```
kubectl api-resources -o name | grep "snapshot.storage.k8s.io"
volumesnapshotclasses.snapshot.storage.k8s.io
volumesnapshotcontents.snapshot.storage.k8s.io
volumesnapshots.snapshot.storage.k8s.io
```

In addition to the CRDs, there is the expectation that the volume snapshot controller and the CSI snapshot sidecar container have also been installed. Read more about them in the Kubernetes documentation at [external_snapshotter](#). These are typically installed by your cluster provider along with the above CRDs.

- For every unique CSI provisioner in the cluster, there should be a matching volume snapshot class to exist, with a label (`velero.io/csi-volumesnapshot-class`). Here is an example manifest:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-azuredisk-vsc
  labels:
    velero.io/csi-volumesnapshot-class: "true"
driver: disk.csi.azure.com
deletionPolicy: Delete
```

The name of the driver must correspond to that specified in the persistent volume or the persistent volume's storage class.

```
kubectl get pv pvc-8835819c-4946-4b7d-8b13-58f827adb5ff -o jsonpath='{.spec.csi.driver}'
disk.csi.azure.com
kubectl get storageclass managed -o jsonpath='{.provisioner}'
disk.csi.azure.com
```


- When restoring CSI volume snapshots across clusters, the name of the CSI driver in the destination cluster must be the same as that on the source cluster to ensure cross cluster portability of CSI volume snapshots. Additionally, the target cluster must have appropriate permissions to access the snapshots in the source cluster.

About TKG and the Container Storage Plug-in on vSphere

The vSphere Container Storage Plug-in is a volume plug-in that runs in a native Kubernetes cluster deployed in vSphere and is responsible for provisioning persistent volumes on vSphere storage. Before you can include CSI volume snapshots in a backup of a Tanzu Kubernetes Grid cluster running in vSphere, using the data protection features of Tanzu Mission Control, you must first deploy this plug-in on the cluster.

For Tanzu Kubernetes Grid Service clusters running in vSphere with Tanzu on vSphere, version 8.0u2 and later, this plug-in is installed by default.

For other Tanzu Kubernetes clusters, the procedure for deploying the Container Storage Plug-in on vSphere is explained in [Volume Snapshot and Restore](#) in the *VMware vSphere Container Storage Plug-in Documentation*.

Back Up the Data Resources in Your Cluster

Use Tanzu Mission Control to create backups of specified data resources in your cluster.

Using Tanzu Mission Control, you can back up an entire cluster or a subset of the data resources it contains. For more information, see [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

Prerequisites

To perform a backup, you must first create a target location, as described in [Create a Target Location for Data Protection](#) and associate it with the cluster or cluster group that contains your cluster.

Prepare the cluster, as described in [Enable Data Protection for a Cluster](#).

If you want to back up persistent volumes using CSI snapshot, make sure your cluster satisfies the CSI snapshot prerequisites described in [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

Make sure you have the appropriate permissions.

- To perform a backup, you must be associated with the `cluster.admin` role in the cluster.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster that you want to back up.
- 2 Click the **Data Protection** tab.

- 3 On the Data Protection tab, under Backups, click the **Create Backup** button.

You can also initiate a backup by clicking the **Create Backup** link in the Data Protection box on the Overview tab of the cluster detail page.

- 4 Specify the scope of the backup, and then click **Next**.

- the entire cluster
- selected namespaces
- resources identified by a label selector

- 5 You can optionally choose to include or exclude specific resources and exclude namespaces.

- a Click **Advanced options**.

- b **Exclude Cluster -scoped resources:** Specify cluster-scoped resources that you want to exclude from the backup.

- c **Exclude Namespace-scoped resources:** Specify a list of resource types that will not be included in the backup.

- d **Excluded namespaces:** Specify a list of namespaces that will not be included in the backup. As a consequence, any namespaces in the cluster not in this list will be backed up. This option is not available for selected namespaces backups.

- e **Include cluster-scoped resources:** Select specific cluster-scoped resources that you want to include in the backup. By default, cluster-scoped resources are backed up if you opt to backup the entire cluster or to backup resources identified by a label selector.

- 6 Select the volume backup method, either FSB or CSI.

By default all volumes are backed up using FSB. You can optionally select between opt-in and opt-out. For more information about these options, see [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

Select **Use CSI snapshot backup for any excluded and remaining volumes** if you want to take CSI snapshots of remaining volumes. For information about CSI requirements, see [Requirements for CSI Volume Backup](#).

For more information about how FSB and CSI work, see the section on FSB and CSI usage in [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

- 7 Choose a target location, and then click **Next**.

When **All locations** is selected, the dropdown list shows all target locations that have been defined for your organization. Select **Currently available locations** to show only the target locations that are immediately accessible from Tanzu Mission Control, and not those that might be temporarily inaccessible.

- 8 Optionally, you can define backup pre- and post-hooks by expanding the Backup Hooks section and defining the hooks. For more information about hooks, see [About Backup and Restore Hooks](#) and [Backup and Restore Hook Examples](#).
 - a Click **Add a Backup hook** to add a hook.
 - b Define the hook by naming it, and then specifying namespaces and/or label selector by defining the key, operator, and value. The hook will be applied to the pods based on specified namespace and/or label selector.
 - c Next, select Add a pre-hook and/or Add a post-hook. You must provide at least one hook.
 - d Enter a container name, optionally.
 - e Provide the command in the command text box.
 - f Select the **Command time-out** and **On Error** settings.
- 9 Specify the schedule for the backup, and then click **Next**.

You can choose **Now** to queue the backup immediately upon completion of this workflow, or choose one of the recurring schedule formats and provide a targeted start time. You can also choose **Custom**, and provide a cron expression to specify the schedule for your backup.

Note You can have a maximum of 720 backups per backup schedule at any given point in time. This capacity is influenced by both the frequency and the retention period of the backup schedule. Therefore, if your schedule requires a high frequency then you might need to reduce the retention period.

Note The **Now** option is not available when scheduling cluster group backups.

- 10 Specify how long to retain the backup, and then click **Next**.

You can specify any number of days, up to 365.

- 11 Provide a name for the backup.

The name that you provide can consist of lowercase letters, numbers, and hyphens.

- 12 Click **Backup**.

Results

When you click **Backup**, Tanzu Mission Control schedules the process of backing up the resources you have specified, and then returns you to the cluster detail page, which shows you the status of the backup operation.

View the Contents of a Backup

Use Tanzu Mission Control to view the contents of a backup of data resources from your cluster.

Using Tanzu Mission Control, you can view the results of a backup operation that might include an entire cluster or a subset of the data resources it contains.

Prerequisites

This procedure assumes that you have a cluster on which you have already performed at least one backup operation.

Make sure you have the appropriate permissions.

- To view the contents of a backup, you must be associated with the `cluster.admin` role in the cluster.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster.
- 2 Click the **Data Protection** tab.
- 3 On the Data Protection tab, in the Backups list, click the backup that you want to view.
The backup detail page shows the type of backup that was performed along with what was backed up, including namespaces, resources, and volumes.
- 4 On the backup detail page, click **Actions** and then choose **Download Logs** to get the logs that were generated during the backup.
- 5 Click **Download resource list** (above the Namespaces list) to get a list of the resources that were backed up.

Restore a Backup

Use Tanzu Mission Control to restore the contents of a backup of data resources from your cluster.

Using Tanzu Mission Control, you can restore the results of a backup operation that might include an entire cluster or a subset of the data resources it contains.

Prerequisites

This procedure assumes that you have a cluster on which you have already performed at least one backup operation.

Make sure you have the appropriate permissions.

- To restore a backup, you must be associated with the `cluster.admin` role in the cluster.

Procedure

- 1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster.
- 2 Click the **Data Protection** tab.

- 3 On the Data Protection tab, in the Backups list, click the menu icon for the backup that you want to restore, and then choose **Restore**.
- 4 Specify what you want to restore from the selected backup.
 - the entire backup
 - selected namespaces
 - resources identified by a label selector
- 5 If you choose to restore selected namespaces, you can optionally modify the target namespace to which you want to restore.
 - a Select the namespace you want to restore.
 - b Click the name of the target namespace.
 - c In the Select target namespace dialog, select the namespace to which you want to restore. You can optionally create a new namespace by entering a name for it.
The name you provide can consist of lowercase letters, numbers, and hyphens.
 - d Click **Confirm**.
- 6 You can optionally click **Advanced options** to specify which resources and namespaces to include or exclude.
 - a To exclude resources, enter the name of the resource you want to exclude, and then click **Add**.
Repeat this step to exclude multiple resources.

Note TKG clusters come with Antrea installed, which includes a webhook that prevents certain Antrea-related resources from being updated. This may cause the overall restore to complete with a partially failed status. This can be avoided by configuring the restore to exclude the problematic resources `tiers.crd.antrea.io` and `tiers.security.antrea.tanzu.vmware.com`.

- b To exclude namespaces, enter the name of the namespace you want to exclude, and then click **Add**.
Repeat this step to exclude multiple namespaces. This option is not available when restoring selected namespaces.

- c Click to toggle **Include cluster-scoped resources** to true if you want to include resources that are cluster-scoped (rather than namespace-scoped), such as `CustomResourceDefinition` and `ClusterRoles` in the restore.

If these resources were included in the backup, they will be restored. By default, cluster-scoped resources are restored if you restore the entire backup or resources identified by a label selector option.

- d You can optionally click to toggle **Update existing resource** to true.

If you select this option, resources that are present in the backup overwrite the corresponding resources on the cluster using a patch. By default, an existing resource in the cluster is not modified even if it differs from its counterpart in the backup.

- 7 Review and select the persistent volumes available to restore.

FSB backups are included in the restore. You can optionally choose to restore CSI volume snapshots.

- 8 Optionally, you can define restore hooks by expanding the Restore Hooks section and defining restore hooks.

- a Click **Add a Restore hook** to add a hook.
- b Define the hook by naming it, and then specifying namespaces and/or label selector by defining the key, operator, and value. The hook will be applied to the pods based on specified namespace and/or label selector.
- c Next, define the hook key, operator, and value.
- d Next, define one or more InitContainer restore hooks and/or Exec restore hooks.

- 9 Click **Next**.

- 10 Provide a name for the restored backup.

The name you provide can consist of lowercase letters, numbers, and hyphens.

- 11 Click **Restore**.

Restore a Backup from a Different Cluster

Use Tanzu Mission Control to restore the results of a backup operation from one cluster into another cluster.

Prerequisites

Before restoring a backup from one cluster into a different cluster, read about considerations for backup restoration between different clusters in [Data Protection](#) in *VMware Tanzu Mission Control Concepts*.

This procedure makes the following assumptions:

- You have a (source) cluster on which you have already performed at least one backup operation.
- You have another (target) cluster into which you want to restore the backup.
- Both clusters are managed by Tanzu Mission Control.
- You have enabled data protection on both clusters.
- The target location of the backup in the source cluster is accessible to the target cluster.
- The target cluster does not already contain a backup with the same name as the backup you want to restore from the source cluster.

Make sure you have the appropriate permissions.

- To restore a backup across different clusters, you must be associated with the `cluster.admin` role in both the source cluster and the target cluster.

Procedure

1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the target cluster in which to restore the backup.

2 Click the **Data Protection** tab.

The Backups list on the Data Protection tab shows the backups that have been created in this cluster.

3 In the Backups section, above the list, click **Restore another cluster backup**.

4 Select the source cluster that contains the backup you want to restore.

The dropdown list shows the clusters to which you have access that contain backups. The target location of the backup in the source cluster must be accessible to the target cluster.

5 Click to select the backup you want to restore, and then click **Next**.

6 Specify what you want to restore from the selected backup.

- the entire backup
- selected namespaces
- resources identified by a label selector

7 If you choose to restore selected namespaces, you can optionally modify the target namespace to which you want to restore.

- a Select the namespace you want to restore.
- b Click the name of the target namespace.

- c In the Specify target namespace dialog, select the namespace to which you want to restore. You can optionally create a new namespace by entering a name for it.
The name you provide can consist of lowercase letters, numbers, and hyphens.

- d Click **Confirm**.

- 8 You can optionally choose to include or exclude specific resources and to exclude namespaces.

- a Click **Advanced options**.

- b Enter the name of the resource you want to exclude, and then click **Add**. Repeat this step to exclude multiple resources.

Note TKG clusters come with Antrea installed, which includes a webhook that prevents certain Antrea-related resources from being updated. This may cause the overall restore to complete with a partially failed status. This can be avoided by configuring the restore to exclude the problematic resources `tiers.crd.antrea.io` and `tiers.security.antrea.tanzu.vmware.com`.

- c You can opt to select **Include cluster-scoped resources**. If set, cluster-scoped (i.e., non-namespaced) resources, such as `CustomResourceDefinition` and `ClusterRoles`, will be restored if the backup has cluster-scoped resources captured. By default, cluster-scoped resources are restored if you opt to restore the entire backup or resources identified by a label selector option.

- d You can also select **Update existing resource**. If this is set, resources on the cluster will be overwritten (via patch) by the corresponding resources present in the backup (if they exist). By default, an existing resource in the cluster will not be modified even if it differs from its counterpart in the backup.

- e You can optionally **Exclude namespaces**. Specify a list of namespaces that will not be included in the restore. This option is not available when restoring selected namespaces.

- 9 You can optionally choose to restore CSI volumes that were backed up.

In the Persistent volumes to restore section, all FSB backed up volumes are listed and will be restored.

Note Restoring CSI volumes across different cloud providers is supported if you moved the CSI snapshots to the backup target location during the backup operation.

- 10 You can optionally define restore hooks by expanding the Restore Hooks section and defining restore hooks.

- a Click **Add a Restore hook** to add a hook.

- b Define the hook by naming it, and then specifying namespaces and/or label selector by defining the key, operator, and value. The hook will be applied to the pods based on specified namespace and/or label selector.

- c Next, define the hook key, operator, and value.
- d Next, define one or more InitContainer restore hooks and/or Exec restore hooks.

11 Click **Next**.

12 Provide a name for the restored backup.

The name you provide can consist of lowercase letters, numbers, and hyphens.

13 Click **Restore**.

View the Contents of a Restore

Use Tanzu Mission Control to view the contents of a restore from a backup on your cluster.

Using Tanzu Mission Control, you can view the results of a restore operation that might include an entire cluster or a subset of the data resources it contains.

Prerequisites

This procedure assumes that you have a managed cluster on which you have already performed at least one backup operation and one restore.

Make sure you have the appropriate permissions.

- To view the contents of a restore, you must be associated with the `cluster.admin` role in the cluster.

Procedure

1 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster.

2 Click the **Data Protection** tab.

3 On the Data Protection tab, in the Restores list, click the restore that you want to view.

The restore detail page shows the type of restore that was performed along with what was restored, including namespaces and volumes.

4 On the restore detail page, you can also download information about the restore.

- Click **Actions** and then choose **Download Logs** to get the logs that were generated during the restore.
- Click **Actions** and then choose **Download Restore Result List** to get a list of restores that were generated from this restore. The downloaded file contains errors and warnings encountered during restore.

About Backup and Restore Hooks

You can customize backups and restores in Tanzu Mission Control using backup and restore hooks.

Backup Hooks

Application workloads may require custom scripts or commands to be run before or after the backup operation. Examples of using hooks while creating backups are:

- freezing the file system to ensure that all pending disk I/O operations have completed prior to backup
- deleting files that should not be backed up
- exporting database prior to backup

Velero supports executing commands in containers in pods during a backup. When performing a backup, you can specify one or more commands to execute in a container in a pod when that pod is being backed up. Pre hooks run before the pod is backed up. Post hooks run after the backup.

There are two ways to specify hooks, annotations on the pod itself, and in the BackupSpec. Tanzu Mission Control allows you to define backup hooks while scheduling backup and specifies hooks in the Backup spec.

For example, to freeze a file system, you may want to run:

- pre hook command “/sbin/fsfreeze --freeze /var/log/nginx”
- post hook command “/sbin/fsfreeze --unfreeze /var/log/nginx”

For more information about Velero and backup hooks, see <https://velero.io/docs/main/backup-hooks>.

Restore Hooks

Application workloads may require custom scripts or commands to be run during or after the restore process. Examples of using hooks while creating backups are:

- create a TEMP file prior to restore
- run a command into restored pod before the application containers can start

You can create restore hooks to run commands in init containers, before the application container starts, or in the application container itself. Velero supports two kinds of restore hooks:

- 1 `InitContainer` restore hooks: These will add init containers into restored pods to perform any necessary setup before the application containers of the restored pod can start. You can use `InitContainerhooks` to run any setup needed for the pod to resume running from its backed-up state. There are two ways to specify `InitContainerhooks`:
 - a specifying hooks in annotations
 - b specifying hooks in the `RestoreSpec`

Tanzu Mission control allows you to define `InitContainer` hooks while restoring a backup and specifies hooks in the `Restore spec`.

- 2 `Exec` restore hooks: These will execute custom commands or scripts in containers of a restored Kubernetes pod. You can use an `ExecRestore` hook to execute commands in a restored pod's containers after they start. There are two ways to specify `Exec` hooks:
 - a specifying hooks in annotations
 - b specifying hooks in the `RestoreSpec`

Tanzu Mission control allows you to define `Exec` hooks while restoring a backup and specifies hooks in the `Restore spec`.

For more information about Velero and restore hooks, see <https://velero.io/docs/main/restore-hooks/>.

Backup and Restore Hook Examples

These examples demonstrate how to use backup and restore hooks in Tanzu Mission Control.

Example 1: MongoDB Backup Hook

Note To run backups and restores, you must be associated with the `cluster.admin` role in the cluster.

- 1 Install `mongodb` via the Bitnami Helm chart (version 13.15.4 is used in this example).

```
$ kubectl create ns my-mongodb-app
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-mongodb-app bitnami/mongodb --namespace=my-mongodb-app --set
replicaSet.enabled=true
```

- 2 `Exec` into the created pod and enter some data into the database.

```
% kubectl -n my-mongodb-app exec -it my-mongodb-app-  
$ mongosh --authenticationDatabase admin -u root -p "${MONGODB_ROOT_PASSWORD}"
examples> db.movies.insertMany([
...   {
...     title: 'Titanic',
...     year: 1997,
...     genres: [ 'Drama', 'Romance' ],
...     rated: 'PG-13',
...     languages: [ 'English', 'French', 'German', 'Swedish', 'Italian', 'Russian' ],
...     released: ISODate("1997-12-19T00:00:00.000Z"),
...     awards: {
...       wins: 127,
...       nominations: 63,
...       text: 'Won 11 Oscars. Another 116 wins & 63 nominations.'
...     },
...     cast: [ 'Leonardo DiCaprio', 'Kate Winslet', 'Billy Zane', 'Kathy Bates' ],
...     directors: [ 'James Cameron' ]
...   },
...   ...
... ])
```

- 3 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster that you want to back up.
- 4 Click the **Data Protection** tab.
- 5 Create a namespace backup for MongoDB and create a backup pre-hook to flush all pending write operations to disk and lock the entire MongoDB instance to prevent additional writes. The post-hook then releases the lock once the backup has been completed.

Use the following settings to define the hooks in the Tanzu Mission Control console.

- For the hook name, enter `mongodb-hook`.
- Click **Included namespaces** and select `my-mongodb-app`.
- Click **Use label selector** to enable label selection.
- In the key field, enter `app.kubernetes.io/name`.
- For the operator, specify `=` (equal sign).
- Enter `mongodb` for the value.
- Configure the backup pre-hook 1, specifying the container name as `mongodb`, a timeout value of 30 seconds, and set to Fail on error. Configure the command as follows:

```
/bin/bash
-o
errexit
-o
pipefail
-c
mongosh --authenticationDatabase admin -u root -p "${MONGODB_ROOT_PASSWORD}" --
eval="db.fsyncLock()"
```

- Configure the backup post-hook 1, specifying the container name as `mongodb`, with a timeout value of 30 seconds and set to Fail on error. Configure the command as follows:

```
/bin/bash
-o
errexit
-o
pipefail
-c
mongosh --authenticationDatabase admin -u root -p "${MONGODB_ROOT_PASSWORD}" --
eval="db.fsyncUnlock()"
```

- 6 Delete the namespace and all the pods and persistent volumes in order to test the restore.

```
% kubectl delete ns my-mongodb-app
```

- Restore the created backup (no restore hooks required). After the restore completes, exec into the restore pod to verify that the data is as expected.

```
% kubectl -n my-mongodb-app exec -it my-mongodb-app-<random suffix> -- bash
$ mongosh --authenticationDatabase admin -u root -p "${MONGODB_ROOT_PASSWORD}"
test> use examples
switched to db examples
examples> db.movies.find( { } )
[
  {
    _id: ObjectId("64abe5d779ad9a6eaa4f456f"),
    title: 'Titanic',
    year: 1997,
    genres: [ 'Drama', 'Romance' ],
    rated: 'PG-13',
    languages: [ 'English', 'French', 'German', 'Swedish', 'Italian', 'Russian' ],
    released: ISODate("1997-12-19T00:00:00.000Z"),
    awards: {
      wins: 127,
      nominations: 63,
      text: 'Won 11 Oscars. Another 116 wins & 63 nominations.'
    },
    cast: [
      'Leonardo DiCaprio',
      'Kate Winslet',
      'Billy Zane',
      'Kathy Bates'
    ],
    directors: [ 'James Cameron' ]
  },
  ...
]
```

Example 2: Wordpress/MariaDB Backup and Restore Exec Hook

In the following example we use hooks to take a consistent backup of the mariadb instance used by Wordpress, but not the wordpress volume itself, since that should be mostly static data.

- Install Wordpress via the Bitnami Helm chart (version 16.1.25 in this example).

```
$ kubectl create ns wordpress
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install --namespace wordpress wordpress bitnami/wordpress
```

- Exec into the pod and populate some data. This example adds data to the database directly but you may alternatively populate data using the wordpress frontend.

```
$ kubectl -n wordpress exec -it wordpress-mariadb-0 -- bash
$ mariadb -u 'root' -p"${MARIADB_ROOT_PASSWORD}"
MariaDB [(none)]> CREATE DATABASE my_db_name;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> USE my_db_name;
Database changed
```

```

MariaDB [my_db_name]> CREATE TABLE users (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     name VARCHAR(50),
->     email VARCHAR(50)
-> );
Query OK, 0 rows affected (0.014 sec)

MariaDB [my_db_name]> INSERT INTO users (name, email) VALUES
->     ('John Doe', 'john@example.com'),
->     ('Jane Smith', 'jane@example.com');
Query OK, 2 rows affected (0.007 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [my_db_name]> SELECT * FROM users;
+----+-----+-----+
| id | name      | email                |
+----+-----+-----+
|  1 | John Doe  | john@example.com    |
|  2 | Jane Smith| jane@example.com    |
+----+-----+-----+
2 rows in set (0.000 sec)

```

- 3 Log in to the Tanzu Mission Control console, and then navigate to the cluster detail page for the cluster that you want to back up.
- 4 Click the **Data Protection** tab.
- 5 Take a namespace backup. Specify a pre hook to use the mariadb-dump tool to take a dump of the db. Optionally, specify a post hook to get rid of the dump taken after it has been backed up.
 - For the hook name, enter wordpress-hook.
 - Click Included namespaces and select wordpress.
 - Click **Use label selector** to enable label selection.
 - In the key field, enter `app.kubernetes.io/name`.
 - For the operator, specify = (equal sign).
 - Enter `mariadb` for the value.
 - Configure the backup pre-hook 1, specifying the container name as `mariadb`, a timeout value of 5 minutes, and set to Fail on error. Configure the command as follows:

```

/bin/bash
-o
errexit
-o
pipefail
-c
mariadb-dump -u 'root' -p"$MARIADB_ROOT_PASSWORD" --single-transaction --all-databases
> /bitnami/mariadb/mybackup

```

- Configure the backup post-hook 1, specifying the container name as mariadb, with a timeout value of 30 seconds and set to Fail on error. Configure the command as follows:

```
/bin/bash
-o
errexit
-o
pipefail
-c
rm -f /bitnami/mariadb/mybackup
```

- 6 Delete the namespace and all its resources to test the restore.

```
% kubectl delete ns wordpress
```

- 7 Perform a namespace restore. Specify an Exec restore hook that restores the database contents using the dump taken using the backup hook.

- For the hook name, enter wordpress-hook.
- Click Included namespaces and select wordpress.
- Click **Use label selector** to enable label selection.
- In the key field, enter `app.kubernetes.io/name`.
- For the operator, specify = (equal sign).
- Enter `mariadb` for the value.
- Configure the restore Exec hook 1, specifying the container name as mariadb, a timeout value of 10 minutes, and set to Fail on error. Configure the command as follows:

```
/bin/bash
-o
errexit
-o
pipefail
-c
sleep 1m && mariadb -u 'root' -p"$MARIADB_ROOT_PASSWORD" < /bitnami/mariadb/mybackup
&& rm -f /bitnami/mariadb/mybackup
```

- 8 Exec into the restored pod to verify that the data was restored.

```
$ kubectl -n wordpress exec -it wordpress-mariadb-0 -- bash
$ mariadb -u 'root' -p"$MARIADB_ROOT_PASSWORD"
MariaDB [(none)]> USE my_db_name;
Database changed
MariaDB [my_db_name]> SELECT * FROM users;
+----+-----+-----+
| id | name      | email      |
+----+-----+-----+
```

```

| 1 | John Doe | john@example.com |
| 2 | Jane Smith | jane@example.com |
+---+-----+-----+
2 rows in set (0.000 sec)

```

Note The above uses `mariadb-dump` to make a logical backup of the database. You may instead use `mariabackup` to take a physical backup, but this requires stopping the `mariadb` server while performing the restore.

Example 3: Init Container Restore Hooks

You can define init container hooks to perform necessary setup before the actual application containers start. The init container definition must be provided in YAML (as you would in a pod manifest). At a minimum, name, image and command must be populated. If you wish to define multiple init containers, place each YAML definition in a separate init container hook.

- For the hook name, enter `my-init-hook`.
- Click Included namespaces and select `my-namespace`.
- Click **Use label selector** to enable label selection.
- In the key field, enter `app.kubernetes.io/name`.
- For the operator, specify `=` (equal sign).
- Enter `myapp` for the value.
- Define the init container hook 1:

```

name: restore-hook-init1
image: alpine:latest
volumeMounts:
  - mountPath: /restores/pvc1-vm
    name: pvc1-vm
command:
  - /bin/ash
  - -c
  - echo -n "FOOBARBAZ" >> /restores/pvc1-vm/foobarbaz

```

- Define init container hook 2:

```

- name: restore-hook-init2
  image: alpine:latest
  volumeMounts:
    - mountPath: /restores/pvc2-vm
      name: pvc2-vm

```



```
command:  
- /bin/ash  
- -c  
- echo -n "DEADFEED" >> /restores/pvc2-vm/deadfeed
```

Note The above example assumes volumes pvc1-vm and pvc2-vm are defined on the pods for which this hook will be applied.

Inspecting Clusters

20

As a platform operator or infrastructure operator, use the cluster inspection features of VMware Tanzu Mission Control to ensure that your Kubernetes clusters conform to a standard.

Through Tanzu Mission Control, you can run preconfigured cluster inspections using Sonobuoy to make sure your clusters conform to community standards.

For more information about testing for cluster conformance, see [Cluster Inspections](#) in *VMware Tanzu Mission Control Concepts*.

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

About Conformance Inspections and Sonobuoy Permission Requirements

To run a `Conformance` inspection, Sonobuoy requires particular privileged permissions. When these permissions are restricted through native pod security policies or through a security policy implemented in Tanzu Mission Control, some tests in the `Conformance` inspection fail.

This behavior is seen in Tanzu Kubernetes clusters running in vSphere with Tanzu.

To prevent this kind of failure for the `Conformance` inspection, you can create a security policy on the cluster through Tanzu Mission Control that uses the `strict` template and disables native pod security policies.

For more information about creating security policies in Tanzu Mission Control, see [Create a Security Policy](#).

For more information about pod security policies in Tanzu Kubernetes clusters running in vSphere with Tanzu, see [Using Pod Security Policies with Tanzu Kubernetes Clusters and Example Role Bindings for Pod Security Policy](#) in *vSphere with Tanzu Configuration and Management*.

Read the following topics next:

- [Start a Cluster Inspection](#)
- [Stop a Cluster Inspection](#)

- [View and Download Inspection Results](#)

Start a Cluster Inspection

Intiate an inspection to verify the conformance of a cluster.

Cluster inspections in VMware Tanzu Mission Control are preconfigured inspections that allow you to test the conformance of your clusters using Sonobuoy. For more information, see [Cluster Inspections](#) in *VMware Tanzu Mission Control Concepts*.

Prerequisites

This procedure assumes that you already have a cluster (either provisioned or attached) that you want to inspect for conformance.

Make sure you have the appropriate permissions.

- To run a cluster inspection, you must be associated with the `cluster.edit` role in the cluster.

Log in to the Tanzu Mission Control console.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Cluster groups**.
- 2 Click the cluster group that contains the cluster you want to inspect, and then click the cluster.
- 3 On the Overview tab of the cluster detail page, in the Inspection box, click **Run Inspection**.
- 4 Make sure the correct cluster group and cluster are selected.
- 5 Choose the type of inspection you want to run, and then click **Run Inspection**.

For TKG Service clusters running Kubernetes version 1.26 or later, there is an additional requirement described below.

Results

When you click **Run Inspection**, the inspection starts and you are directed back to the cluster detail page that shows your inspection is running. When the inspection is complete, the result of the inspection is displayed in the Inspection box.

What to do next

For Tanzu Kubernetes Grid Service clusters with Kubernetes version 1.26 or later, running in vSphere with Tanzu, the Pod Security Admission (PSA) is set to enforce by default. As a result, inspections are unable to run until you add the required label to the namespace created during the inspection run.

To allow the inspection to run:

- 1 After the inspection starts, retrieve the name of the `image-pull-####` namespace that was created by the inspection.

```
kubectl get ns | grep "image-pull"
```

- 2 Apply the PSA label to the namespace.

```
kubectl label ns image-pull-#### pod-security.kubernetes.io/enforce: privileged
```

Stop a Cluster Inspection

Stop a running inspection.

Prerequisites

Before starting this procedure, perform the following tasks:

- Log in to the Tanzu Mission Control console.
- Make sure you have the appropriate permissions.
 - To view the inspections for a cluster, you must be associated with the `cluster.view` role in the cluster.
 - To stop a running cluster inspection, you must be associated with the `cluster.edit` role in the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Inspections**.
The Inspections page lists the clusters that have been inspected along with their inspection results, as well as the clusters that have inspections that are currently running.
- 2 Locate your cluster in the list of inspections, and then click the cluster.
- 3 On the Overview tab of the cluster detail page, in the Inspection box, click **Cancel Inspection**.

View and Download Inspection Results

Use VMware Tanzu Mission Control to view the results of cluster inspections that you have run.

Prerequisites

Before starting this procedure, perform the following tasks:

- Log in to the Tanzu Mission Control console.
- Make sure you have the appropriate permissions.
 - To view the inspections for a cluster, you must be associated with the `cluster.view` role in the cluster.

Procedure

- 1 In the left navigation pane of the Tanzu Mission Control console, click **Inspections**.

The Inspections page lists the clusters that have been inspected along with their inspection results, as well as the clusters that have inspections that are currently running. All of the clusters in your organization that you have permission to see are listed.

You can alternatively navigate to a cluster and click the **Inspections** tab on the cluster detail page to see just the inspections that have been run for that cluster.

- 2 Locate your cluster in the list of inspections.

You can sort or filter the list to quickly find your cluster and its inspections. The Inspection field shows the type of inspection that was run, while the Result field shows the status of the inspection (success, failure, or in progress).

- 3 To see the inspection report, click the link in the Result field.

The detail page for the inspection shows the type of inspection that was run with a summary of the number of tests that succeeded and failed. The individual tests are listed below the summary.

- 4 You can optionally download a compressed TAR file of the inspection result by clicking the **Actions** dropdown at the top of the inspection detail page, and then choosing **Download**.
- 5 From the **Actions** dropdown, you can also delete the inspection report that is stored by Tanzu Mission Control.

Viewing Your Policies

21

As a platform administrator, use VMware Tanzu Mission Control to see and create policies for the Kubernetes objects in your organization, and evaluate any issues that occur with those policies.

Policies allow you to define rules to govern the objects and members in your organization. For more information about policies in Tanzu Mission Control, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

The Policies page in the Tanzu Mission Control console has subpages that enable you to view your policies in different ways.

- The Assignments page has an objects tree that shows the hierarchy of Kubernetes objects in your organization, which you can view through the Clusters hierarchy (infrastructure view) or the Workspaces hierarchy (application view). This page also has tabs that show the different types of policies you can create.
- The Insights page has a table that displays issues that have occurred in policies that you have implemented in your organization. This page has tabs that allow you to filter the contents of the table by policy type.

This feature is only available in the advanced version of Tanzu Mission Control.

Viewing Your Access Policies

You can view and manage the access policies for workload clusters and namespaces through the Access page in the Tanzu Mission Control console. However, the access policies for some objects are managed on other pages.

- Access policies for credentials are managed on the Access tab of the Administration page. Click **Administration** in the left navigation pane, and then click **Access**.
- Access policies for management clusters and their provisioners are managed on the Access tab of the management cluster detail page. Click **Administration** in the left navigation pane, and then click **Management clusters**. On the Management clusters tab, click the management cluster to go to its detail page, and then click **Access**.

Read the following topics next:

- [View the Policy Assignments for an Object](#)
- [View Policy Insights](#)

- [Export Policy Code](#)
- [View Your Access Policies](#)
- [View Identities and Roles](#)
- [View Access Roles and Permissions for Tanzu Mission Control](#)

View the Policy Assignments for an Object

Display the policy assignments for an object in the Tanzu Mission Control console.

Most policy types are managed through the Policies page in the Tanzu Mission Control console, as described below. However, access policies are managed separately. For more information about access policies, see [Chapter 22 Managing Access to Your Resources](#).

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to view policies.

- To view the policies for an object, you must be associated with the `.admin` role on that object.

Procedure

- 1 Click **Policies** in the left navigation pane of the Tanzu Mission Control console to show the subpages, and then click **Assignments**.
- 2 Click the tab for the type of policy you want to see.
For example, to see image registry policies click the **Image registry** tab.
- 3 Use the tree control to navigate to the object for which you want to see policies, and then click the object whose policies you want to see.

The Policies page shows the direct policy for the object and the object's inherited policies.

View Policy Insights

Display policy issues for an object in the Tanzu Mission Control console.

Policy insights belong to two general types.

- A violation issue indicates that an object in your organization has violated the constraints of a policy that governs it.
- A sync issue indicates that Tanzu Mission Control is unable to apply the policy to an object.

Prerequisites

Make sure you have the appropriate permissions to view policies.

- To view the policies for an object, you must be associated with the `.admin` role on that object.

Note This feature is only available in the advanced version of Tanzu Mission Control.

Procedure

- 1 Click **Policies** in the left navigation pane of the Tanzu Mission Control console to show the subpages, and then click **Insights**.

The table on the Policy insights page shows all of the issues that have occurred in your organization.

- 2 Click a tab to see only the issues for a single policy type.
For example, to see issues with security policies click the **Security** tab.
- 3 You can also filter and sort the table using the column headers.
- 4 To evaluate the issue, you can click the provided links to navigate to the policy and objects involved in a policy issue.

About Policy Insights

Learn about the policy insights that Tanzu Mission Control tracks.

Tanzu Mission Control (TMC) monitors the policies you implement and reports any potential issues, or insights.

The Policy insights page in the Tanzu Mission Control console shows the detected insights and provides links to the cluster, cluster group, and policy where the insight was found. The types of insights that you might see on this page include the following:

- `Violation` - indicates that Kubernetes resources are not in compliance with the policy. (This insight applies only to Gatekeeper-based policies.)
- `Sync` - indicates that the policy failed to be created on the cluster.
- `Threshold` - indicates that a quota policy for a Kubernetes resource is approaching (80%) or has exceeded (100%) the quota specified in a quota policy.
- `Health` - indicates that a policy is not enforced due to policy operator health issues. (This insight applies only to Gatekeeper-based policies.)
- `Incompatibility` - indicates that a component used by Tanzu Mission Control is installed on the cluster, but was not installed by TMC. This situation might potentially cause issues with the functionality of policies applied to the cluster. For example, if Gatekeeper is already installed on the cluster.

Tanzu Mission Control policies and Gatekeeper

Some of the policy types in Tanzu Mission Control (TMC) are implemented using OPA Gatekeeper. These policy types include security policy, image registry policy, and mutation policy. When you implement one of these policies, Tanzu Mission Control installs Gatekeeper on the cluster and maintains that installation while you have TMC policies applied to that cluster.

If there is an installation of Gatekeeper on the cluster that was not installed by TMC, then TMC does not alter or maintain that installation. In many cases, this situation does not impact the functionality of the policy. However, because Gatekeeper was not installed and configured by TMC, Gatekeeper-based policies implemented through TMC might not function as expected. This situation raises an `Incompatibility` insight.

For best results, if you are using TMC policies on a cluster, remove external installations of Gatekeeper and let TMC manage Gatekeeper for you.

This potential issue does not impact access, quota, and network policies.

Export Policy Code

Download the code for a defined policy in YAML or JSON format through the Tanzu Mission Control console.

For policy types that are managed through the Policies page in the Tanzu Mission Control console, excluding access policies which are managed separately, you can download the JSON or YAML code for the policy. This allows you to treat policies as code so that you can perform versioning, tracking, back up, and other functions.

For example, you can define a policy in Tanzu Mission Control and then download the policy code. Then you can use that file to check into source control, or make changes to the policy and apply it to the cluster or other clusters.

Note This capability is not available for access policies.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to download policies.

- To view and download the policies for an object, you must be associated with the `.admin` role on that object.

Procedure

- 1 Click **Policies** in the left navigation pane of the Tanzu Mission Control console to show the subpages, and then click **Assignments**.
- 2 On the Policies Assignments page, click the tab for the type of policy you want to see or create.

For example, to see image registry policies click the **Image registry** tab.

- 3 Use the tree control to navigate to the object for which you want to see policies, and then click the object whose policies you want to see.

The Policies assignments page shows the direct policy for the object and the object's inherited policies.

- 4 Download your policy code.

The procedure differs slightly, depending on which type of policy you want to download.

- For mutation and image registry policy types:
 - a In the list of defined policies for an object, click the menu icon for the policy and then choose the format you want to export.
- For other policy types:
 - a In the list of defined policies for an object, click to expand the options for the policy.
 - b Click **Download**, and then choose the format you want to export.

Results

When you choose your download format (**Download JSON** or **Download YAML**), Tanzu Mission Control generates the code for the policy and downloads the file through your browser.

View Your Access Policies

Display the access policy and role bindings for an object in the Tanzu Mission Control console.

Unlike other policy types, that are managed through the Policies page in the Tanzu Mission Control console, access policies are managed separately.

Before you can view your access policies, log in to the Tanzu Mission Control console and make sure you have the appropriate permissions.

- To view or edit the policies for an object, you must be associated with the `.admin` role on that object.

View the Access Policies for Objects in the Organizational Hierarchy

Access policies for the organization, cluster groups and workload clusters, and workspaces and namespaces are managed on the Access page.

- 1 Click **Access management** in the left navigation pane of the Tanzu Mission Control console, and then click **Access policies**.
- 2 Click to select the organizational view (Clusters or Workspaces).
- 3 Use the tree control to navigate to the object for which you want to see policies, and then click the object whose policies you want to see.

View the Access Policies for Management Clusters and Provisioners

Access policies for management clusters and their provisioners are managed on the Access tab of the management cluster detail page.

- 1 Click **Administration** in the left navigation pane.
- 2 On the Administration page, click **Management clusters**.
- 3 On the Management clusters tab, click the management cluster to go to its detail page, and then click the **Access** tab.
- 4 Use the tree control to navigate to the object for which you want to see policies, and then click the object whose policies you want to see.

View the Access Policies for Credentials

Access policies for credentials are managed on the Access tab of the Administration page.

- 1 Click **Administration** in the left navigation pane.
- 2 On the Administration page, click **Access**.
- 3 Select a credential from the list.

View Identities and Roles

Use the Tanzu Mission Control console to see the identities (users, groups, and service accounts) in your organization and click to see mapped roles, resources, and other details.

When you create a role binding in an access policy for an resource, Tanzu Mission Control creates a mapping of the identity (user, group, or service account), the role, and the resource. This mapping defines an access policy. The User permissions page in the Tanzu Mission Control console shows these access policy details from the perspective of the identity, so you can see the roles with which they are associated and the resources (such as clusters, workspaces, and organization) to which they have access.

Prerequisites

Make sure you have the appropriate permissions to view identities and roles.

- To view the identities and roles in your organization, you must be associated with the `organization.admin` role.

Procedure

- 1 Click **Access management** in the left navigation pane of the Tanzu Mission Control console, and then click **User permissions**.

The table on the User permissions page lists the identities in your organization. You can filter the table to more quickly find the identity you're looking for.

- 2 To view more details for a given identity, click on a link in the row for the identity.
 - Click an identity name to see the resources that the identity has access to and the roles they are associated with for that resource. From here, you can drill down further to see the details for each role. You can also access the role binding editor for a resource from the context menu.
 - Click the resource type or the role to see the role bindings that the identity has for each resource.

View Access Roles and Permissions for Tanzu Mission Control

As an organization administrator, you can use VMware Tanzu Mission Control to view the built-in and custom roles that can be used in your access policies, as well as the individual permissions that are included in these roles.

Tanzu Mission Control provides a set of roles that you can use in your organization's access policies, and you can create your own access roles using a set of focused permissions. You can view these roles and permissions through the Roles tab on the Administration page of the Tanzu Mission Control console.

Prerequisites

Make sure you have the appropriate permissions.

- To view the roles in your organization and their included permissions, you must be associated with a `.admin` role.
- To view the complete set of individual permission primitives that are available to be included in custom roles, you must be associated with the `organization.admin` role for the organization.

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Roles** tab.

The Roles tab shows a table that contains the built-in roles and the custom roles that have been defined for your organization.

 - To filter what appears in the table, you can click the filter icon in the name column.
 - To see only the custom roles, you can click the **Hide Tanzu built-in roles** toggle.
- 2 To see the definition for a role, click the name of the role in the table.

The role detail page shows the permissions that are included in the role, any Kubernetes RBAC rules that are defined for it, and the visibility of the role.

- 3 To see a complete list of the permissions that you can use to define a role, click **Create Custom Role**.

The Create page shows a table of permissions, which you can sort and filter if necessary.

Managing Access to Your Resources

22

Define who has access to each resource in your organization using role-based access control.

VMware Tanzu Mission Control uses secure-by-default, role-based access control (RBAC) to manage user permissions at each level of the hierarchical structure for your organization. Each object is protected by an access policy that defines who has access to that resource, and these policies are inherited down through the organizational hierarchy. For more information, see [Access Control](#) in *VMware Tanzu Mission Control Concepts*.

In addition to the built-in roles provided by Tanzu Mission Control, you can create custom roles from a set of built-in permissions to suit the needs of your organization.

You can view and manage the access policies for workload clusters and namespaces through the Access page in the Tanzu Mission Control console. However, the access policies for some objects are managed on other pages.

- Access policies for credentials are managed on the Access tab of the Administration page.
- Access policies for management clusters and their provisioners are managed on the Access tab of the management cluster detail page.

For more information, see [View Your Access Policies](#).

Read the following topics next:

- [Add a Role Binding](#)
- [Edit a Role Binding](#)
- [Remove a Role Binding](#)
- [Add a Role Binding on a Credential](#)
- [Edit the Role Binding for a Credential](#)
- [Remove the Role Binding from a Credential](#)
- [Create a Custom Access Role](#)
- [Edit a Custom Access Role](#)
- [Delete a Custom Access Role](#)

Add a Role Binding

Create a role binding in the access policy for an object to specify permissions for a member or group.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Access page for the type of object for which you want to add a role binding, as described in [View Your Access Policies](#).

Make sure you have the appropriate permissions.

- To edit the access policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 Navigate to the object whose access policy you want to add a role binding to, as described in [View Your Access Policies](#).
- 2 In the organizational view, select the object.
- 3 Click the arrow next to the object name under **Direct access policies**.
- 4 Click **Create Role Binding**.
- 5 Select the role that you want to bind to an identity.
- 6 Select the identity type that you want to bind.
 - **user**
 - **group** can be any group you have defined for your organization in Tanzu Platform Cloud Services.
 - **Kubernetes service account** identifies a service account, and the namespace in which it is defined.
- 7 Enter one or more identities, clicking **Add** after each identity.
- 8 Click **Save**.

Results

When you click **Save**, the new role binding is applied to the policy and is displayed on the Access page.

Example: Grant yourself access to your first cluster

The first time you create or attach a cluster might happen before your organization has a robust hierarchy of guardrail policies to manage access to clusters and other organizational objects. If you're an administrator for your Tanzu Mission Control organization, you'll already have access by default. But if you aren't, or if you want to share this cluster with a colleague, you'll need to set a direct access policy. Here is an example of how to do that.

- 1 In the organizational view on the Access page, select your cluster.
- 2 Click the arrow next to the cluster name under **Direct access policies**, and then click **Create Role Binding**.
- 3 Select the `cluster.admin` role to grant administrative access to this cluster.
- 4 Select the `user` type to grant access to individuals.
- 5 Enter user IDs for yourself or your colleagues in the **user identity** field, clicking **Add** after each identity.
- 6 Click **Save**.

Edit a Role Binding

Edit an existing role binding in the policy for an object to modify permissions for a member or group.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Access page for the type of object for which you want to edit a role binding, as described in [View Your Access Policies](#).

Make sure you have the appropriate permissions.

- To edit the access policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 Navigate to the object whose access policy you want to edit, as described in [View Your Access Policies](#).
- 2 In the organizational view, select the object.
- 3 Click the arrow next to the object name under **Direct Access Policies**.
- 4 Click **Edit** for the role binding you want to modify.
- 5 To remove an identity from the binding, click the delete (x) icon of the identity.
- 6 To bind additional identities to the existing role, select the identity type, enter the name of the identity, and then click **Add**.
- 7 When you are finished making changes, click **Save**.

Results

When you click **Save**, the new role binding is applied to the access policy and is displayed on the appropriate Access page.

Remove a Role Binding

Remove an existing role binding from the policy for an object.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Access page for the type of object in which you want to delete a role binding, as described in [View Your Access Policies](#).

Make sure you have the appropriate permissions.

- To edit the access policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 Navigate to the object whose access policy you want to edit, as described in [View Your Access Policies](#).
- 2 In the organizational view, select the object.
- 3 Click the arrow next to the object name under **Direct Access Policies**.
- 4 Click **Delete** for the role binding you want to remove.
- 5 In the confirmation dialog, click **Yes, Delete Role Binding**.

Results

When you confirm the deletion, the role binding is removed from the access policy.

Add a Role Binding on a Credential

Create a role binding in the access policy for a credential to specify permissions for a member or group.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Make sure you have the appropriate permissions.

- To view and edit the access policy for a credential, you must be associated with the `credential.admin` role on the credential, or the `organization.credential.admin` role for the organization.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Access** tab.
- 2 On the Access tab of the Administration page, select the credential whose access policy you want to edit.
- 3 Click the arrow next to the credential name under **Direct access policies**.
- 4 Click **Create Role Binding**.
- 5 Select the role that you want to bind to an identity.
- 6 Select the identity type that you want to bind.
- 7 Enter the identity to bind, and click **Add**.
- 8 After adding the role bindings you want to create, click **Save**.

Results

When you click **Save**, the new role bindings are applied to the policy and displayed on the Access tab.

Edit the Role Binding for a Credential

Edit an existing role binding in the policy for a credential to modify permissions for a member or group.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Settings page.

Make sure you have the appropriate permissions.

- To view and edit the access policy for a credential, you must be associated with the `credential.admin` role on the credential, or the `organization.credential.admin` role for the organization.

Procedure

- 1 On the Settings page in the Tanzu Mission Control console, click **Access**.
- 2 On the Access tab of the Settings page, select the credential whose access policy you want to edit.
- 3 Click the arrow next to the credential name under **Direct access policies**.
- 4 Click **Edit** for the role binding you want to modify.
- 5 To remove an identity from the binding, click the delete (x) icon of the identity.
- 6 To bind additional identities to the existing role, select the identity type, enter the name of the identity, and then click **Add**.
- 7 When you are finished making changes, click **Save**.

Results

When you click **Save**, the new role binding is applied to the access policy and is displayed on the Access tab.

Remove the Role Binding from a Credential

Remove an existing role binding from the policy for a credential.

Prerequisites

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Make sure you have the appropriate permissions.

- To delete the access policy for a credential, you must be associated with the `credential.admin` role on the credential, or the `organization.credential.admin` role for the organization.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Access** tab.
- 2 On the Access tab of the Administration page, select the credential whose access policy you want to delete.
- 3 Click the arrow next to the credential name under **Direct access policies**.
- 4 Click **Delete** for the role binding you want to remove.
- 5 In the confirmation dialog, click **Yes, Delete Role Binding**.

Results

When you confirm the deletion, the role binding is removed from the access policy.

Create a Custom Access Role

As an administrator, you can create a custom role that you can use in access policies in VMware Tanzu Mission Control.

Tanzu Mission Control provides a set of standard roles that you can use to establish a baseline security posture for your organization. To suit the needs of your organization, you can also create custom roles that have a more specific focus. For example, say your organization has a small group that manages the data protection aspects of all your clusters. In such a case, you can create a custom role that contains all of the necessary permissions for this set of tasks without adding other administrative permissions that the members of this group don't need.

When creating a custom role, consider the following facets of a custom role:

- Custom roles can be used only in role bindings in access policies for the type of object for which you specify the role's visibility. You cannot use custom roles in access policies for management clusters.

- Almost all of the aspects of the custom role are optional. However, to have functional value, you must include at least one of Tanzu permissions, Kubernetes RBAC rules, or an aggregation.
- When you use aggregation, only Kubernetes RBAC rules are included. Tanzu permissions are not included. To use aggregation, create the custom roles with the base rules and label them, and then create the custom role with the aggregation using the label selector to identify the custom roles with the base rules. For more information about role aggregation, see [Aggregated ClusterRoles](#) in the Kubernetes documentation.

Prerequisites

Make sure you have the appropriate permissions.

- To create a custom role, you must be associated with the `organization.admin` role for the organization.

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Roles** tab.
- 2 Click **Create Custom Role**.

Note You can optionally use an existing role, either built-in or custom, as a pattern for the new role. Click the menu icon for the existing role in the table on the Roles tab, and then choose **Create role from selected**.

- 3 Provide a name for the custom role.
- 4 You can optionally provide a description so other team members understand the purpose of the custom role.
- 5 You can optionally specify one or more labels to apply to the custom role.
You can use labels to identify multiple roles that you want to aggregate into a single combined role.
- 6 Select the visibility for the custom role.
 - a Click **Cluster** or **Workspace**.
 - b Select the hierarchy levels to which the role can be applied.
- 7 You can optionally specify Tanzu permissions to include in the custom role.
You can sort and filter the permissions displayed in the table to locate the individual permissions you want to add.
 - To select a permission, click its checkbox.

- To select all permissions, click the select all checkbox at the top of the table. Be aware that the select all checkbox selects all of the available permissions, not just those that are currently displayed. If you have applied a filter, you do not see all of the permissions you have selected by clicking the select all checkbox.
- 8 You can optionally add one or more Kubernetes RBAC rules for the custom role.
 - a Select one or more verbs (Kubernetes permission type).
You can optionally type in your own custom verb.
 - b Select the type.
 - **Resource** allows you to choose from Kubernetes resources.
 - **Non-resource URL** allows you to specify a URL that is not a Kubernetes resource, for example `/healthz`.
 - c Select one or more Kubernetes resource type, or enter the URL for a custom non-resource.
 - d You can optionally enter a resource name for easier identification.
 - e You can optionally specify an API group.
 - f You can optionally click **Add Another** and repeat these steps to include additional Kubernetes RBAC rules.
- 9 To aggregate with other roles, enter one or more label selectors to identify the roles to aggregate.

For more information about role aggregation, see [Aggregated ClusterRoles](#) in the Kubernetes documentation.
- 10 You can optionally click **Deprecate** to toggle the prevention of new role bindings from using the custom role.
- 11 Click **Create**.

Results

When you click **Create**, Tanzu Mission Control creates the custom role. It is now available for creating access policies for the objects you specified for its visibility.

Edit a Custom Access Role

Use Tanzu Mission Control to edit the custom roles created for your organization.

As an administrator, you can create custom roles to meet the needs of your organization. After a role has been created, you cannot modify the name or visibility of the role, but you can edit the permissions and rules that it contains.

Prerequisites

Make sure you have the appropriate permissions.

- To edit or create a custom role, you must be associated with the `organization.admin` role for the organization.

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Roles** tab.
- 2 Click on the custom role you want to modify.
You cannot modify the Tanzu built-in roles.
- 3 On the custom role detail page, click **Actions** (on the top right of the page), and then choose **Edit** from the dropdown.
- 4 You can optionally edit the description.
- 5 Edit the permissions you want to include by selecting or deselecting the appropriate checkboxes.
To view permissions that are not already included, click the **Hide unselected permissions** toggle.
- 6 Edit the Kubernetes RBAC rules as necessary.
 - To remove a rule, click its delete icon.
 - To add a new rule, enter the parameters and then click **Add Rule**.
 - To change an existing rule, click to adjust the parameters.
- 7 You can optionally click **Deprecate** to toggle the prevention of new role bindings from using the custom role.
This action is helpful when you are preparing to remove a role from your organization.
- 8 Click **Save**.

Delete a Custom Access Role

Use Tanzu Mission Control to remove the custom roles created for your organization.

As an administrator, you can delete custom roles from your organization. If a role is currently in use in an access policy, you cannot delete the role, but you can tag the role as deprecated so that it can't be added to access policies.

Prerequisites

Make sure you have the appropriate permissions.

- To delete a custom role, you must be associated with the `organization.admin` role for the organization.

Log in to the Tanzu Mission Control console, and then go to the Administration page.

Before you can delete a custom role, you must remove the role from any access policy that currently uses it.

Procedure

- 1 On the Administration page in the Tanzu Mission Control console, click the **Roles** tab.
- 2 Click on the custom role you want to delete.
You cannot delete the Tanzu built-in roles.
- 3 On the custom role detail page, click **Actions** (on the top right of the page), and then choose **Delete** from the dropdown.
- 4 In the confirmation dialog, enter the name of the role, and then click **Delete**.
If the role is currently used by an access policy, it cannot be deleted. If the policy you are trying to delete is in use, Tanzu Mission Control displays a list of the policies that use it. Click **View Policy** to go to the policy and remove all role bindings that use the custom role before deleting it.
- 5 Click **Delete**.

Use security policies and mutation policies to manage the security context in which deployed pods operate in your clusters.

Using VMware Tanzu Mission Control, you can make the deployments to your clusters more secure by implementing constraints that govern what deployed pods can do. Security policies and mutation policies, implemented using OPA Gatekeeper, can help prevent the deployment of pods that don't conform to your specifications.

- A security policy allows you to restrict certain aspects of pod execution in your clusters, such as privilege escalation, Linux capabilities, and allowed volume types. When a pod is deployed to a cluster with a security policy, and it does not conform to the constraints specified in the policy, the deployment is disallowed.
- A mutation policy is similar to security policy, except that it can alter (mutate) the pod specification (podspec) to enforce conformance to the policy. When a pod is deployed to the cluster, the policy mutates the podspec as defined in the policy before admitting the request to deploy the pod. For more information, see [Chapter 24 Mutating Kubernetes Resources](#).

For more details about how security policies work in Tanzu Mission Control, see [Pod Security Management](#) in *VMware Tanzu Mission Control Concepts*.

Note The Custom template for security policies is available in Tanzu Mission Control only if you are using the advanced version of Tanzu Mission Control.

Evaluation and Precedence of Security and Mutation Policies

Both security policies and mutation policies can be applied in the clusters hierarchy (infrastructure view) in Tanzu Mission Control, and they are inherited down through the hierarchy. You can read more about that in [Pod Security Management](#) in *VMware Tanzu Mission Control Concepts*.

When a workload is deployed to a cluster that has both mutation policies and security policies, first the mutation policies are evaluated and applied, mutating the podspec as specified. Then, after all mutations have been applied, the security policies are evaluated to determine if the podspec is conformant to allow deployment.

When multiple mutation policies are applied to a cluster, through either direct or inherited policies, they are evaluated as follows:

- Mutation policies are run against the podspec starting with those applied to the cluster, then those inherited from the cluster group, and finally those inherited from the organization.
- If there are multiple mutation policies at any level of the hierarchy, they are run in alphanumeric order.

Because a mutation policy can overwrite changes applied by a previously run mutation policy, the order in which they are run is significant.

Read the following topics next:

- [Create a Security Policy](#)
- [Edit a Security Policy](#)
- [Delete a Security Policy](#)

Create a Security Policy

Add a security policy that governs how pods can run in your clusters.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the security policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To create a security policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Security** tab.
- 2 Use the tree control to navigate to and select the object for which you want to create a security policy.
- 3 Click **Create Security Policy**.
- 4 Select the security template that you want to use.
 - The **Strict** template is a preconfigured set of constraints that define a tight security context for pods in your clusters. The detailed options defined in this template are displayed on the form in the Tanzu Mission Control console.
 - The **Baseline** template is a preconfigured set of constraints that prevents known privilege escalations, but is less stringent than the `strict` template to ease adoption of the security policy for common containerized workloads. The detailed options defined in this template are displayed on the form in the Tanzu Mission Control console.

- The **Custom** template allows you to specify how to handle the various aspects of pod security for your clusters.

Note The Custom template for security policies is available only if you are using the advanced edition of Tanzu Mission Control.

- 5 Provide a policy name.
- 6 If you choose the **Custom** policy template, specify the detailed aspects for the policy.
- 7 You can optionally provide label selectors to specify particular namespaces that you want to include or exclude for this policy.

For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

- 8 Select the **Enforcement action** you want the policy to use.
 - **Deny** (default) indicates the policy is fully enforced, denying admission requests with any violation.
 - **Dry run** indicates the policy is not enforced, but allows you to see the impact of the policy for testing. If this option is selected, the policy does not prevent containers from being scheduled on the cluster, but you do receive alerts for policy violations. You can later edit this policy to re-enable policy enforcement.
 - **Warn** works like dry run, except that it provides immediate feedback when a potential denial occurs.
- 9 You can optionally select **Disable native pod security policies** to restrict the enforcement of native Kubernetes pod security policies that are implemented on the cluster.

If this option is selected for any policy (direct or inherited) that impacts a cluster, then native Kubernetes pod security policies are disabled for that cluster. If this option is not selected, then any native pod security policies defined on the cluster take precedence over the policy you define here.

You can later edit this policy to re-enable native pod security policies.

- 10 Click **Create policy**.

Edit a Security Policy

Edit an existing security policy that constrains pod execution in your clusters.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the security policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the security policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Security** tab.
- 2 Use the tree control to navigate to and select the object whose security policy you want to edit.
- 3 Click the policy name, and then click **Edit**.
- 4 Make the desired changes to the policy.
- 5 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object and is displayed on the Policies page.

Delete a Security Policy

Delete an existing security policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the security policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete the security policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Security** tab.
- 2 Use the tree control to navigate to and select the object whose security policy you want to delete.
- 3 Click the policy name.
- 4 Click **Delete**.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the policy is removed from the object.

Use a mutation policy to modify Kubernetes resources when pods are deployed in your clusters.

Using VMware Tanzu Mission Control, you can make the deployments to your clusters more consistent by implementing policies that add or modify resources in those deployments before they are admitted. Mutation policies, implemented using OPA Gatekeeper, can help prevent the deployment of pods that don't conform to your specifications.

You can use a mutation policy to alter the following types of Kubernetes resource properties:

- annotations
- labels
- pod security properties

A mutation policy can alter (mutate) the pod specification (podspec) to enforce conformance to the policy. When a pod is deployed to the cluster, the policy mutates the podspec as defined in the policy before admitting the request to deploy the pod.

Mutating Labels and Annotations

You can create a mutation policy to specify a label to be applied to a Kubernetes resource before it is admitted to a cluster. When a deployment is evaluated for admission, the policy creates the label (key/value pair) only if it does not already exist. If there is a pre-existing label on the resource, the policy cannot change it. Also, because the policy evaluates deployments only during admission, mutations implemented by the policy can be altered manually after admission.

For more information about how OPA Gatekeeper handles label and annotation mutations, see [Mutation: AssignMetadata](#) in the *OPA Gatekeeper documentation*.

Evaluation and Precedence of Mutation Policies

Mutation policies can be applied in the clusters hierarchy (infrastructure view) in Tanzu Mission Control, and they are inherited down through the hierarchy. You can read more about that in [Pod Security Management](#) in *VMware Tanzu Mission Control Concepts*.

When multiple mutation policies are applied to a cluster, through either direct or inherited policies, they are evaluated as follows:

- Mutation policies are run against the podspec starting with those applied to the cluster, then those inherited from the cluster group, and finally those inherited from the organization.
- If there are multiple mutation policies at any level of the hierarchy, they are run in alphanumeric order.

Because a mutation policy can overwrite changes applied by a previously run mutation policy, the order in which they are run is significant.

When a workload is deployed to a cluster that has both mutation policies and security policies, first the mutation policies are evaluated and applied, mutating the podspec as specified. Then, after all mutations have been applied, the security policies are evaluated to determine if the podspec is conformant to allow deployment.

Read the following topics next:

- [Create a Mutation Policy](#)
- [Edit a Mutation Policy](#)
- [Delete a Mutation Policy](#)

Create a Mutation Policy

Add a mutation policy that can alter a podspec to specify how pods can run in your clusters.

A mutation policy, implemented using OPA Gatekeeper, allows you to enforce the conformance of pods running in your clusters by changing certain properties of the pod specification (podspec) prior to allowing deployment.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the mutation policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To create a mutation policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Mutation** tab.
- 2 Use the tree control to navigate to and select the object for which you want to create a mutation policy.
- 3 Click **Create Mutation Policy**.
- 4 Select the mutation template that you want to use.
- 5 Provide a policy name.

- 6 In the grid, locate the property for which you want to provide a mutation, and then click the edit icon.

You can optionally filter the displayed properties by **Property** or **Description**, and use the toggle to **Show only defined mutations**.

- 7 Provide the details about how you want to mutate the property.

The contents of the property mutation dialog shows the options available for the property, and varies according to which property you choose to mutate.

- 8 Click **Save**.

- 9 You can optionally provide label selectors to specify particular namespaces that you want to include or exclude for this policy.

For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

- 10 Click **Create policy**.

Results

When you click **Create policy**, the policy is created and applied to the object and is displayed on the Policies page. Because a mutation policy assesses and mutates pods only at the time of admission, the policy impacts only new, incoming requests for creating or updating pods. The mutation policy does not impact pods that have already been admitted into the cluster.

Edit a Mutation Policy

Edit an existing mutation policy that constrains pod execution in your clusters.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the mutation policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the mutation policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Mutations** tab.
- 2 Use the tree control to navigate to and select the object whose mutation policy you want to edit.
- 3 Click the arrow to expand the list of direct policies applied to the object.
- 4 Click menu icon for the policy you want to edit, and then choose **Edit**.

- 5 Make the desired changes to the policy.

You can add, modify, and remove property mutations and label selectors. But you cannot change the name of the policy or its template.

- 6 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object and is displayed on the Policies page. Because a mutation policy assesses and mutates pods only at the time of admission, the policy impacts only new, incoming requests for creating or updating pods. The mutation policy does not impact pods that have already been admitted into the cluster.

Delete a Mutation Policy

Delete an existing mutation policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the mutation policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete a mutation policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Mutation** tab.
- 2 Use the tree control to navigate to and select the object whose mutation policy you want to delete.
- 3 Click the arrow to expand the list of direct policies applied to the object.
- 4 Click the menu icon for the policy you want to delete.
- 5 Choose **Delete**.
- 6 In the confirmation dialog, click **Delete**.

Results

When you click **Delete**, the policy is removed from the object. However, the mutations that have already been applied by the policy are not backed out of pods that were deployed while the policy was active.

Managing Access to Image Registries

25

Define the registries from which images can be pulled for deployment in your managed namespaces.

Using VMware Tanzu Mission Control, you can make the deployments to namespaces in your clusters more secure by restricting the image registries from which images can be pulled, as well as the images that can be pulled from a registry. By default, Tanzu Mission Control does not impose any such restriction, and allows you to manage image registry restrictions at the organizational level and at the workspace level.

Each namespace and workspace can be protected by an image registry policy that defines the registries from which an image can be pulled, and these policies are inherited down through the organizational hierarchy. For more information, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

Note This feature is only available in the advanced version of Tanzu Mission Control.

Read the following topics next:

- [Create an Image Registry Policy](#)
- [Edit an Image Registry Policy](#)
- [Delete an Image Registry Policy](#)

Create an Image Registry Policy

Add an image registry policy that restricts the images that can be pulled for deployment in your managed namespaces.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the image registry policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To create an image registry policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Image registry** tab.
- 2 Use the tree control to navigate to and select the object for which you want to create an image registry policy.
- 3 Click **Create Image Registry Policy**.
- 4 Select the recipe you want to use.
 - The `allowed-name-tag` recipe allows you to create rules using an image name or tag name or both.
 - The `block-latest-tag` recipe prevents the use of images that are tagged `latest`.
 - The `custom` recipe allows you to create rules using multiple factors.
 - The `require-digest` recipe prevents the use of images that do not have a digest.
 - The `Allow Registry` recipe is deprecated. You can replace existing policies that use this recipe with a new policy using the `custom` recipe with a `hostname` rule.
- 5 Provide a policy name.
- 6 Specify the details for the selected recipe (if required).

The `allowed-name-tag` and `custom` recipes allow you to create multiple rules using a combination of options. Only the options that you specify are restricted by the rule. You can create multiple rules.

Make sure you click **Add Rule** for each rule that you define.

The `block-latest-tag` and `require-digest` recipes do not require any further specification.
- 7 You can optionally provide label selectors to specify particular namespaces that you want to include or exclude for this policy.

For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.
- 8 Select the **Enforcement action** you want the policy to use.
 - **Deny** (default) indicates the policy is fully enforced, denying admission requests with any violation.
 - **Dry run** indicates the policy is not enforced, but allows you to see the impact of the policy for testing. If this option is selected, the policy does not prevent containers from being scheduled on the cluster, but you do receive alerts for policy violations. You can later edit this policy to re-enable policy enforcement.
 - **Warn** works like dry run, except that it provides immediate feedback when a potential denial occurs.
- 9 Click **Create Policy**.

Results

When you click **Create Policy**, the new image registry policy is applied to the object and is displayed on the Policies page.

Edit an Image Registry Policy

Edit an existing image registry policy that restricts the images that can be pulled for deployment in your managed namespaces.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the image registry policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the image registry policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Image registry** tab.
- 2 Use the tree control to navigate to and select the object whose image registry policy you want to edit.
- 3 Click the policy name to display the policy.
- 4 Click the **Actions** drop-down, and then choose **Edit**.
- 5 Edit the policy.

After an image registry policy is created, you cannot change the name or the recipe for the policy. But you can edit the details of the policy. For more information about image registry policy details, see [Create an Image Registry Policy](#).

- 6 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object and is displayed on the Policies page.

Delete an Image Registry Policy

Delete an existing image registry policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the image registry policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete the image registry policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Image registry** tab.
- 2 Use the tree control to navigate to and select the object whose image registry policy you want to delete.
- 3 Click the policy name to display the policy.
- 4 Click the **Actions** drop-down, and then choose **Delete**.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the policy is removed from the object, and any existing image registry policies in child objects are enabled.

Managing Network Communication for Your Clusters

26

Define how pods communicate using network policies.

Using VMware Tanzu Mission Control, you can create a network policy that defines how pods communicate with each other and other network endpoints, using preconfigured templates called recipes. By default, Tanzu Mission Control does not impose any such restriction, and allows you to manage network restrictions at the organizational level and at the workspace level.

Tanzu Mission Control implements network policies using Kubernetes native network policies. Each namespace and workspace can be governed by a network policy, and these policies are inherited down through the organizational hierarchy. Network policies are additive, both inherited and direct network policies are applied and are effective on your namespaces according to Kubernetes rules.

For more information about Kubernetes native network policies, see [Network Policies](#) in the Kubernetes documentation. For more information about policy inheritance in Tanzu Mission Control, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

Note This feature is only available in the advanced version of Tanzu Mission Control.

Read the following topics next:

- [Create a Network Policy](#)
- [Edit a Network Policy](#)
- [Delete a Network Policy](#)

Create a Network Policy

Add a network policy that governs how your pods communicate with each other.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the network policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To create a network policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Network** tab.
- 2 Use the tree control to navigate to and select the object for which you want to create a network policy.
- 3 Click **Create Network Policy**.
- 4 Select the network policy recipe to use.
- 5 Provide a policy name.
- 6 If you select a pod-specific recipe, you must specify the labels to identify the pods to which the policy applies.
 - a Under **Labels**, enter the key and value on which to filter pods.
 - b Click **Add Label**.

You can optionally repeat this step to add multiple labels. Each label that you add increases the potential group of pods that are impacted by the policy. The policy impacts the pods that have any of the labels that you include.

- 7 If you select the `custom-egress` or `custom-ingress` recipe, you can add a rule to define criteria to restrict network traffic. For more information, see [Rules in Network Policies](#).
- 8 You can optionally provide label selectors to specify particular namespaces that you want to include or exclude for this policy.

Note The label selector for network policies in Tanzu Mission Control matches only the labels that you have applied to resources through Tanzu Mission Control. Labels that are created outside of Tanzu Mission Control are not evaluated.

For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.

- 9 Click **Create Policy**.

Results

When you click **Create Policy**, the new network policy is applied to the object and is displayed on the Policies page.

Rules in Network Policies

Learn how to define a rule in a network policy.

Some of the network policy recipes allow you to provide a rule that uses a set of criteria to identify the target locations with which to permit or restrict communication, and the port on which they can communicate. The criteria used in these rules can include the following types:

- IP range (allow and exclude)
- label selector (pods and namespaces)

- port and protocol

You can define multiple criteria of a given type in a single rule, and use these criteria in combination with each other. The location criteria (IP range and label selector) that you define are specific to the template that you are using. For the `custom-ingress` template, you identify sources from which to allow traffic; and for the `custom-egress` template, you identify destinations to which to allow traffic.

If you do not specify any location criteria, the policy does not restrict traffic by location. All sources or destinations are allowed. Likewise, if you do not specify any ports, all ports are allowed.

IP Range Criteria

Within a rule, you can click **IP Block** to define location criteria based on IP range (expressed in CIDR notation).

When you specify a range for allowed IP addresses, traffic is permitted on all IP addresses in that range. You can also optionally exclude a range of IP addresses within the allowed range.

If you specify multiple IP ranges for a given location, the location must match any one of the criteria. For example, if you define three allowed IP ranges, traffic is allowed to (or from) locations within any one of the three ranges.

Label Selector Criteria

Within a rule, you can click **Selector** to define location criteria based on label selectors for pods and namespaces.

If you specify multiple label selectors for a given type, the location must match any one of the criteria to allow traffic. For example, if you define three pod selectors, traffic is allowed to (or from) pods that have a label matching any one of the three selectors.

Note The label selector for network policies in Tanzu Mission Control matches only the labels that you have applied to resources through Tanzu Mission Control. Labels that are created outside of Tanzu Mission Control are not evaluated.

If you specify a location using both the pod selector and the namespace selector in a single location definition, then both must be satisfied.

Port Criteria

The port and protocol fields allow you to specify a port on which to allow traffic, and the protocol that the traffic must use. You can specify multiple ports, and each one must have a corresponding protocol. The port can be either a numerical or named port.

If you specify multiple ports, the channel must match any one of the criteria to allow traffic. For example, if you define three ports, traffic is allowed through any one of the three ports.

Rule Enforcement

When you define a rule with multiple criteria, network traffic is not permitted unless it satisfies at least one location criteria and at least one port criteria.

For example, say you create a `custom-ingress` policy with a rule that contains the following criteria.

- Pod Selector = `this:that` and Namespace Selector = `release:production`
- Allow IP Range = `192.168.1.1/16`
- Port = `1232` and Protocol = `TCP`

When you apply this policy, ingress traffic is not allowed unless it comes in on port `1232` using the `TCP` protocol AND it comes from either one of the following locations:

- an IP address in the `192.168.1.1/16` range
- a pod with a `this:that` label in a namespace with a `release:production` label

Edit a Network Policy

Edit an existing network policy that governs how your pods communicate with each other.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the network policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the network policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Network** tab.
- 2 Use the tree control to navigate to and select the object whose network policy you want to edit.
- 3 Click the policy name to display the recipe type used in the policy.
- 4 Click **Edit**.
- 5 Edit the policy.

You cannot change the recipe type for an existing network policy.

- To remove an existing label, click the delete icon.
- To add another label, enter the key and value, and then click **Add Label**.
- To remove a port specification, delete the value in the **Port** field.
- To remove an IP block location specification, delete the value in the **Allow IP Range** field.

6 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object and is displayed on the Policies page.

Delete a Network Policy

Delete an existing network policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the network policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete the network policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Network** tab.
- 2 Use the tree control to navigate to and select the object whose network policy you want to delete.
- 3 Click the policy name to display the recipe type used in the policy.
- 4 Click **Delete**.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the policy is removed from the object.

Managing Resource Consumption in Your Clusters

27

Use a namespace quota policy to restrict the aggregate quantity of resources used in your clusters.

Using VMware Tanzu Mission Control, you can create a quota policy that allows you to set limits on the resources that can be consumed by the namespaces in your clusters. When you create a quota policy, you can choose from three preconfigured sets of resource limits (small, medium, large), or configure your own with a custom template that allows you to specify the quantity limits of various resource types.

After you apply a quota policy at the organization, cluster group, or cluster level, Tanzu Mission Control monitors the resource requests in the namespaces in your managed clusters. When the resource requests approach the limit you set, you see a warning in the table on the Policy insights page. For more information about policy insights, see [View Policy Insights](#).

Note This feature is only available in the advanced version of Tanzu Mission Control.

Read the following topics next:

- [Create a Quota Policy](#)
- [Edit a Quota Policy](#)
- [Delete a Quota Policy](#)

Create a Quota Policy

Add a quota policy that restricts the usage of resources in your clusters.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the quota policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To create a quota policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Quota** tab.

2 Use the tree control to navigate to and select the object for which you want to create a quota policy.

3 Click **Create Quota Policy**.

4 Select the policy template to use.

There are three preconfigured templates (small, medium, large) that define common limits on CPU and memory requests. There is also a custom template that allows you specify CPU, memory, and storage limits, as well as limits on a variety of object types, including those listed under [Object Count Quota](#) in the Kubernetes documentation.

5 Provide a policy name.

6 If you choose the custom template, you can specify the aggregate resource limits to use.

7 You can optionally add label selectors to include or exclude in the calculation of aggregate resource usage.

- To enter a label selector, enter the key and value, and then click **Add label selector**.

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 field on the form.

You can optionally repeat this step to add more label selectors for this policy.

8 Click **Create Policy**.

Results

When you click **Create Policy**, the new quota policy is applied to the object and is displayed on the Policies page. You can optionally repeat this procedure to create additional quota policies.

Edit a Quota Policy

Modify the set of namespaces that are included in your quota policy.

You can edit a quota policy to modify the label selectors, and if the policy uses the custom template you can adjust the resource limits. However, when you create a quota policy, the template for that policy is set and you cannot change it. To change the template, delete the existing policy, and then create a new one with the desired template.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the quota policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the quota policy for an object, you must be associated with the `.admin` role for that object.

Procedure

1 On the Policies page, click the **Quota** tab.

- 2 Use the tree control to navigate to and select the object whose quota policy you want to edit.
- 3 Click the policy name to display the policy.
- 4 Click **Edit**.
- 5 Edit the policy.
 - You can edit the namespace label selectors of the quota policy. To remove an existing label, click the delete icon next to the label. To add another label selector, enter the key and value, and then click **Add label selector**.
 - If the policy uses the custom template, you can also adjust the specified limits.
- 6 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object and is displayed on the Policies page.

Delete a Quota Policy

Delete an existing quota policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the quota policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete the quota policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Quota** tab.
- 2 Use the tree control to navigate to and select the object whose quota policy you want to delete.
- 3 Click the policy name to display the policy.
- 4 Click **Delete**.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the policy is removed from the object.

Define and implement specialized policies that govern your Kubernetes clusters.

For many aspects of cluster management, VMware Tanzu Mission Control provides specific policies that you can use to enforce rules on your fleet of Kubernetes clusters, such as access policies, image registry policies, and namespace quota policies. However, not all of the aspects that you might want to control are covered by these baseline policies. Custom policies are somewhat open-ended and provide the opportunity to address aspects of cluster management that specifically suit the needs of your organization.

To implement a custom policy, you must first have a template that declaratively defines the structure of the policy. The custom policy template can then be used to create and apply custom policies to your clusters.

Custom policies in Tanzu Mission Control are implemented using the Gatekeeper project from Open Policy Agent (OPA). To create a custom policy template, you use Rego, the policy language of OPA. For more information about Rego, see the OPA [Policy Language](#) documentation.

Note This feature is only available in the advanced version of Tanzu Mission Control.

Read the following topics next:

- [Create a Policy Template](#)
- [Delete a Policy Template](#)
- [Add a Custom Policy](#)
- [Edit a Custom Policy](#)
- [Delete a Custom Policy](#)

Create a Policy Template

Create a template in the Tanzu Mission Control console that you can use to apply custom policies.

The template provides a declarative definition of a policy, which you can use to apply custom constraints on your managed Kubernetes resources. This template represents a `ConstraintTemplate` object, which contains the schema of the constraint and the Rego code that defines how it is enforced.

Tanzu Mission Control provides some sample preconfigured policy templates that you can use as a starting place.

- You can use the `tmc-require-labels` template to enforce labels with a key and optional value on specified Kubernetes resources (for example, ensuring that all pods and namespaces in a cluster have a label with the key `owner`).
- You can use `tmc-https-ingress` the template to enforce that all ingress objects created on a cluster have `tls` configuration and that the `allow-http` annotation set to `false`.

For more information about defining a `ConstraintTemplate` object, see [Constraint Templates](#) in the OPA Gatekeeper documentation on GitHub.

Prerequisites

Make sure you have the appropriate permissions to create policy templates.

- To create a policy template, you must be associated with the `.admin` role or the `organization.policytemplate.edit` role on the organization.

Procedure

- 1 Click **Policies** in the left navigation pane of the Tanzu Mission Control console to show the subpages, and then click **Templates**.
- 2 On the Custom policy templates page, click **Create Template**.
- 3 On the Create page, you can optionally provide a description for the template.
- 4 Define the template.

You can write the template definition directly in the code box provided, or click **Import** to use a YAML file that you have already written.

- 5 You can optionally define Kubernetes resources to be cached for the policy.

To enforce a custom policy, OPA might need access to more state than just the object under test. For example, if the policy calculates the number of pods running in the namespace, OPA needs access to all those pods. If your template requires such data replicated in OPA cache, specify the group, version, and kind of each Kubernetes resource that needs to be cached. For more information about cached resources, see [Replicating Data](#) in the OPA Gatekeeper documentation.

- 6 Click **Create**.

Results

When you click **Create**, Tanzu Mission Control creates the template and enters it in the table on the Custom policy templates page. You can now use this template to create a policy and apply it to your clusters.

Delete a Policy Template

Delete a custom policy template that you have previously created.

Because the structure of custom policies are dependent on a template for their definition, you cannot edit a policy template after it is created. However, you can delete a template along with the policies that use it, and then create a new template and policies. This procedure describes how to delete a policy template.

You cannot delete the preconfigured policy templates that are provided by Tanzu Mission Control. The preconfigured templates are denoted by a `tmc-` prefix.

Prerequisites

Make sure you have the appropriate permissions to delete policy templates.

- To delete a policy template, you must be associated with the `.admin` role or the `organization.policytemplate.edit` role on the organization.

Procedure

- 1 Click **Policies** in the left navigation pane of the Tanzu Mission Control console to show the subpages, and then click **Templates**.
- 2 In the table on the Custom policy templates page, click the menu icon for the template you want to delete, and then choose **Delete**.

You can optionally save a local copy of the template definition by clicking **Download YAML** from the popup menu.

- 3 In the confirmation dialog, click **Delete**.

If there are custom policies that use this template, an additional dialog is displayed.

- 4 To delete the template and all of the custom policies that use it, select **Force delete template**, and then click **Delete**.

Results

When you click **Delete**, Tanzu Mission Control removes the template, as well as the associated policies if you selected that option.

Add a Custom Policy

Use VMware Tanzu Mission Control to create and apply a custom policy from an existing policy template.

Using a policy template that provides a declarative definition of a policy, you can provide parameters and apply a custom policy to manage your Kubernetes resources.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the custom policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions to add custom policies.

- To add a custom policy, you must be associated with the `.admin` role on that object.

Procedure

- 1 On the Policies page, click the **Custom** tab.
- 2 Use the tree control to navigate to and select the object to which you want to apply a custom policy.
- 3 Click **Create Custom Policy**.
- 4 On the custom policy create form, select the policy template you want to use, and then provide a name for the policy.
- 5 Specify the target resources on which to enforce the policy, and then click **Add Resource**.
A target resource, identified by a kind and an API group, specifies the Kubernetes API resource on which the policy is enforced.
- 6 Specify parameters for your policy, if defined by the schema of the selected template.
Not all custom policies require parameters. If the selected template does not accept parameters, the Parameters section is not displayed on the form.
- 7 You can optionally provide label selectors to specify particular namespaces that you want to include or exclude for this policy.
For more information about how label selectors work, see [Policy-Driven Cluster Management](#) in *VMware Tanzu Mission Control Concepts*.
- 8 Select the **Enforcement action** you want the policy to use.
 - **Deny** (default) indicates the policy is fully enforced, denying admission requests with any violation.
 - **Dry run** indicates the policy is not enforced, but allows you to see the impact of the policy for testing. If this option is selected, the policy does not prevent containers from being scheduled on the cluster, but you do receive alerts for policy violations. You can later edit this policy to re-enable policy enforcement.
 - **Warn** works like dry run, except that it provides immediate feedback when a potential denial occurs.
- 9 Click **Create Policy**.

Results

When you click **Create Policy**, Tanzu Mission Control installs the Gatekeeper admission webhook on your cluster, synchronizes the policy template to your cluster, and then creates the policy and applies it to your cluster.

Edit a Custom Policy

Edit an existing custom policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the custom policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To edit the custom policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Custom** tab.
- 2 Use the tree control to navigate to and select the object whose custom policy you want to edit.
- 3 Click the policy name, and then click **Edit**.
- 4 Make the desired changes to the policy.
You cannot edit the template or name of a custom policy.
- 5 After you finish editing the policy, click **Save**.

Results

When you click **Save**, the updated policy is applied to the object.

Delete a Custom Policy

Delete an existing custom policy.

Prerequisites

Log in to the Tanzu Mission Control console, go to the Policies page and view the custom policies for the object, as described in [View the Policy Assignments for an Object](#).

Make sure you have the appropriate permissions.

- To delete a custom policy for an object, you must be associated with the `.admin` role for that object.

Procedure

- 1 On the Policies page, click the **Custom** tab.
- 2 Use the tree control to navigate to and select the object whose custom policy you want to delete.
- 3 Click the policy name.
- 4 Click **Delete**.
- 5 In the confirmation dialog, click **Yes**.

Results

When you click **Yes**, the custom policy and its template are removed from the cluster.

Managing Administrative Settings

29

Use Tanzu Mission Control to specify the default settings for policies.

Tanzu Mission Control allows you set guardrails on various types of policies. Some of these policies are enforced using the OPA Gatekeeper open source package. These policies include security, image registry, mutation, and custom policies. When any one of these policies is created and applied to a cluster, Tanzu Mission Control installs OPA Gatekeeper on your cluster.

The Gatekeeper package contains various resources which Tanzu Mission Control configures with default values. The Settings tab of the Administration page in the Tanzu Mission Control console allows you to specify some of these default configuration values to suit the needs of your organization.

The following are some of the Gatekeeper configurations that can be changed:

- Configurations for Gatekeeper `controller-manager` and `audit` deployments:
 - You can update the number of replicas and CPU/Memory limits/requests.These configurations can help customize the deployments as required for the cluster or organization.
- Validating webhook configuration installed by Gatekeeper:
 - You can update the `timeout` value, `rules` value, and the `failure policy` that define which incoming requests should be validated.

This is an admission webhook installed by Gatekeeper to validate incoming requests against the defined policies. For more information about admission webhook, see [Customizing Admission Behavior](#) in the *OPA Gatekeeper* documentation. For more information about timeout, rules in matching requests, and failure policy, see the respective sections under [Webhook configuration](#) in the Kubernetes documentation.

Settings are applied in a hierarchical manner with inheritance. The settings of the organization cascade down through cluster groups and clusters. When these are set at the higher level, such as the organization level, they can be overridden by editing at specific lower levels like clusters. As an organization administrator, you can change the settings at any level. To modify settings at the cluster group level, you must have `clustergroup.admin` permissions, and to modify settings at the cluster level, you must have `cluster.admin` permissions on the cluster.

- Note that these are sensitive settings. If set incorrectly, they can impact policies applied on the cluster and some policies could stop working as expected. Use this feature with caution.

Read the following topics next:

- [Configure Policy Settings](#)

Configure Policy Settings

You can set specific default values for policies in Tanzu Mission Control to customize the policy configurations for organizations, clusters, and cluster groups.

Prerequisites

Log in to the Tanzu Mission Control console.

Make sure you have the appropriate permissions to configure policy settings.

- To configure policy settings at the organization level, you must be associated with the `organization.admin` role.
- To configure policy settings at the cluster group level, you must be associated with the `clustergroup.admin` role for the cluster group.
- To configure policy settings at the cluster level, you must be associated with the `cluster.admin` role for the cluster.

Procedure

- 1 Click **Administration** in the left navigation pane.
- 2 Click the **Settings** tab.
- 3 Selection the organization, cluster group, or cluster for which you want to configure settings, and then click **Create Setting**.

In the Create Setting dialog, the **Select setting** dropdown lists the policy settings you can modify.

- 4 In the Gatekeeper settings section, click the edit icon for the setting that you want to change.

Object Missing

30

This object is not available in the repository.

Read the following topics next:


- [How to Get Support](#)
- [Reported Subscription Usage Doesn't Seem Right](#)
- [Issues Attaching a Cluster](#)
- [Issues Registering a Tanzu Kubernetes Grid Management Cluster](#)
- [Issues Registering a Supervisor Cluster](#)
- [Workload Cluster Provision Failures](#)
- [Troubleshooting Issues with Tanzu Application Platform](#)
- [Troubleshooting Issues with Data Protection](#)
- [Manage Issues with AWS EKS Credentials in Tanzu Mission Control](#)
- [Monitoring of AWS GuardDuty for Unauthorized Access](#)

How to Get Support

Depending on which edition of Tanzu Mission Control you're using and your troubleshooting needs, you can get support through our support portal, from directly within the product, or you can reach out to the community for support.

How to Contact Paid Support

If you pay for Tanzu Mission Control, you are entitled to support. There are three easy ways to access support.

- 1 Access support directly from <https://customerconnect.vmware.com/support>.
- 2 You can file a support request from directly within Tanzu Mission Control. Click the help icon  to open the help panel and select **Create a Support Request**.
- 3 You can access support from within the **Tanzu Platform cloud services console** by navigating to **Support Requests** and selecting **Create a Support Request**.

Reported Subscription Usage Doesn't Seem Right

If you see a discrepancy between the usage reported in the Subscription information page and what you think you're actually consuming, there are a few things to look for.

Problem

Sometimes the usage that appears in the Subscription information page might differ from what you expect to see. There are a few reasons for this and some solutions.

Cause

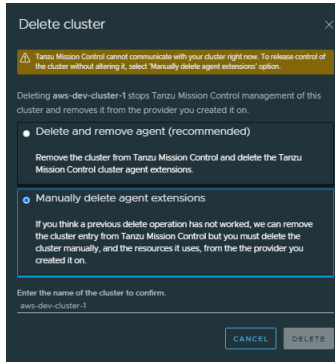
- Do you have disconnected clusters? These clusters appear on the Clusters page with a health status of "disconnected." Usually this happens because you delete a cluster directly from the infrastructure.
- Are you counting CPU Capacity or Allocatable CPU? Tanzu Mission Control is licensed according to how many CPUs, vCPUs, or cores you place under management. So the Subscription information page tracks overall CPU Capacity of your clusters, including any space that's reserved for system daemons that power the OS and Kubernetes. On other pages in Tanzu Mission Control, like the cluster details page, you see Allocatable CPU reported, which is what's left over for your nodes after system-reserved, kube-reserved, and eviction-threshold have been subtracted from your CPU Capacity. Learn more about [CPU Capacity and Allocatable CPU from Kubernetes](#).
- How are you calculating usage? Tanzu Mission Control collects Kubernetes CPU information and approximates 2 Kubernetes CPUs = 1 Core or 2 vCPUs. For CPU-based subscriptions in vSphere environments, you can have up to 32 cores or 64 vCPU per CPU. However, you need to license your entire host to purchase CPU-based vSphere subscriptions.

If the cause of the problem is that you have disconnected clusters, follow the solution steps to delete the disconnected clusters.

Solution

- 1 Navigate to the Clusters page.
- 2 Click the menu icon (☰) to the left of the disconnected cluster and choose **delete**.
The Delete cluster dialog appears.

Figure 30-1. Delete Cluster Dialog



- 3 Select **Manually delete agent extensions** to remove this disconnected cluster from Tanzu Mission Control management.

Note If the cluster still exists in your infrastructure, this operation doesn't delete it or remove the cluster agent extensions, because the cluster is disconnected from Tanzu Mission Control. You'll have to delete the cluster or remove the agent manually.


- 4 Enter the name of the cluster to confirm the delete operation and click **Delete**.

Issues Attaching a Cluster

If you're having trouble bringing an existing cluster under Tanzu Mission Control management, there are a number of steps you can take to identify and fix the problem.

Problem

You might suspect you're having trouble attaching a cluster if:

- When attaching the cluster, you see the verify connection fail.
- On the cluster detail page, your cluster is stuck in the pending status .
- In the CLI, you see the attach operation timeout and receive a message telling you the cluster didn't achieve the desired state.
- If you're attaching a cluster using the API, you see the status stuck in pending for longer than 7 minutes.

Cause

There are a few common reasons for cluster attach issues. Follow the solution described to identify which of these causes is the issue and help you correct it.

- You might need to rerun the cluster agent installation if it failed.
- You might have a resource scheduling issue preventing the cluster agent from being installed.
- You might have a problem with the cluster connectivity or proxy configuration.

- You might be attaching a windows-only cluster, and Tanzu Mission Control only supports clusters that have at least some Linux-based nodes.
- You might be attaching an ARM cluster, which is currently unsupported.
- You might be attaching a cluster on an unsupported version of kubernetes.
- You might be attaching a cluster that isn't CNCF conformant, and Tanzu Mission Control only supports CNCF conformant clusters.

Solution

- 1 Check if the problem is your access rights or network connection to the cluster by running a simple kubectl command like get namespaces.

```
kubectl get namespaces
```

If you cannot access the cluster, contact your kubernetes administrator.

- 2 Check if the problem is a cluster agent installation failure.
 - a Verify that the cluster agent installed successfully by checking if the vmware-system-tmc namespace exists.

```
kubectl get namespace vmware-system-tmc
```

NAME	STATUS	AGE
vmware-system-tmc	Active	1m

- a If the namespace doesn't exist, try reinstalling the agent.

3. Install agent Install the Tanzu Mission Control agent on your cluster and verify its connection

This command installs the cluster agent extensions on your namespace named vmware-system-tmc. This link expires in 48 hours.

```
kubectl create -f "https://starteruat12122.stable.tmc-dev.cloud.vmware.com/installer?id=..."
```

You can view the full configuration details of the VMware Tanzu Mission Control agent and copy it to your system before applying it on your Kubernetes cluster.

[View YAML](#)

[VERIFY CONNECTION](#)

For example:

```
kubectl create -f "https://myorgname.tmc.cloud.vmware.com/installer?id=4...9&source=attach"
```

3 Investigate the agent pods to locate the source of the problem.

a Check the status of the agent pods.

```
kubectl get pods -n vmware-system-tmc -l "tmc-extension-name in (extension-
manager,extension-updater,agent-updater)"
```

NAME	READY	STATUS	RESTARTS	AGE
agent-updater-cc9c67b4d-sp6q9	1/1	Running	0	3m
agentupdater-workload-27478576-8rxvk	0/1	Completed	0	60s
extension-manager-85f9cf5497-rk8ms	1/1	Failed	0	3m
extension-updater-6c4c95c5bf-9wl6p	1/1	Pending	0	3m

b If any of the pods are in a failed state, fetch the logs for the failed pod to learn more information about the problem. For example:

```
kubectl logs -l "tmc-extension-name=extension-updater" -n vmware-system-tmc
```

For help solving pod failures, see [Common Log Errors and Solutions for Pod Failure](#) in the [Examples](#) section of this page.

c If the pods have not been scheduled (they appear as “pending” or not at all), inspect the deployments of extension-manager, extension-updater and agent-updater to find any deployments that are not in a ready state. For example:

```
kubectl get deployments -n vmware-system-tmc -l "tmc-extension-name in (extension-
manager,extension-updater,agent-updater)"
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
agent-updater	1/11	1	3m	
extension-manager	1/11	1	3m	
extension-updater	0/11	1	3m	

d If any of the deployments show as 0/1 , describe the deployment to understand how to correct the issue.

For example:

```
kubectl describe deployment extension-updater -n vmware-system-tmc
```

These kubernetes errors are usually resource related. For example, you might get `<output>OutOfDisk`.

Ask your kubernetes administrator to solve these types of errors by expanding resources or relaxing policies.

- 4 If the deployments are all showing 1/1 READY, look for `Events` messages in the replicaset to understand how to correct the issue.
- a Look for replicaset that are not in the ready status. For example:

```
kubectl get replicaset -n vmware-system-tmc -l "tmc-extension-name in (extension-manager,extension-updater,agent-updater)"
```

- b Inspect any replicaset that are not in ready status and check the events messages in the replicaset to understand how to correct the issue. For example:

```
kubectl describe replicaset -n vmware-system-tmc extension-updater-6c4c95c5bf
```

The `Events` messages should contain enough information for you to solve the issue.

Example

Table 30-1. Common Log Errors and Solutions for Pod Failure

Application	Log	Solutions
Extension-Updater	failed to discover cluster metadata	Make sure your cluster is CNCF conformant. You can check the CNCF site for a list of Platform Certified Kubernetes - Distribution Providers .
Extension-Updater	failed to set up manager:get Tanzu Mission Control connection	<ol style="list-style-type: none"> 1 Make sure the cluster has internet connectivity. 2 If you're behind a firewall, set up or validate the configuration of your proxy. See Connecting through a Proxy for more information.
Extension-Updater	pre-flight checks failed	Make sure your cluster is using a version of kubernetes that is supported by Tanzu Mission Control.

What to do next

If you can't solve the issue on your own, you can create a log bundle to share with support. Run the following commands against the cluster you're trying to attach:

- 1 Print a report of components present in the cluster. `tmc cluster validate -h`
- 2 Archive the logs of all components in this cluster. `tmc cluster logs -h`

See [How to Get Support](#) if you need more information about contacting them.

Issues Registering a Tanzu Kubernetes Grid Management Cluster

If you're having trouble registering a Tanzu Kubernetes Grid Management cluster with Tanzu Mission Control, there are a number of steps you can take to identify the root cause and fix the problem.

Problem

You might suspect you're having trouble registering your Tanzu Kubernetes Grid management cluster if:

- When registering your management cluster, the verification step fails.
- If you're registering a management cluster through the CLI, the operation times out.
- If you're registering a management cluster through the API, the status of the cluster remains pending for longer than 7 minutes.

Cause

There are a few common reasons people have trouble registering their management clusters to Tanzu Mission Control. You can follow the solution described to identify which of these causes is the issue and help you correct it.

- You might need to rerun the registration step if the agent installation failed. Common reasons for failure:
 - You might have tried to attach instead of register the management cluster.
 - You might have tried to register a Tanzu Kubernetes Grid management cluster using the vSphere with Tanzu Supervisor cluster procedure and seen a message like “no such resource found.”
- You might have a resource scheduling issue or permission issue preventing the agent from installing successfully.
- You might have a problem with connectivity or proxy configuration.

Solution

- 1 Check the format of the registration URL to verify you're using the procedure to register a Tanzu Kubernetes Grid management cluster. You should see `source=registration` and `type=tkgm`. For example:

```
https://myorgname.tmc.cloud.vmware.com/installer?id=1234&source=registration&type=tkgm.
```

If you see `attach` or `tkgs` in the URL, you need to restart the registration process. Make sure you follow the procedure described in [Register a Management Cluster with Tanzu Mission Control](#).

- 2 Check if the problem is your access rights or network connection to the management cluster by running a simple kubectl command like `get namespaces`.

```
kubectl get namespaces
```

If you cannot access the cluster, contact your kubernetes administrator.

- 3 Check if the problem is the agent installation failure.
 - a Verify that the cluster agent installed successfully by checking if the `vmware-system-tmc` namespace exists.

```
kubectl get namespace vmware-system-tmc
```

NAME	STATUS	AGE
vmware-system-tmc	Active	1m

- b If the namespace doesn't exist, try rerunning the command to register your management cluster. Remember to run this command on your management cluster, using the URL from the Tanzu Mission Control registration UI. For example:

```
kubectl apply -f 'https://myorgname.tmc.cloud.vmware.com/installer?id=8...7&source=registration&type=tkgm' --kubeconfig="C:\path-to\my-management-clusters\kubeconfig"
```

- 4 Investigate the agent pods to locate the source of the problem.

- a Check the status of the agent pods.

```
kubectl get pods -n vmware-system-tmc -l "tmc-extension-name in (extension-manager,extension-updater,agent-updater)"
```

NAME	READY	STATUS	RESTARTS	AGE
agent-updater-cc9c67b4d-sp6q9	0/1	CLBO	0	3m
agentupdater-workload-27478576-8rxvk	0/1	CLBO	0	60s
extension-manager-85f9cf5497-rk8ms	0/1	CLBO	0	3m
extension-updater-6c4c95c5bf-9wl6p	0/1	CLBO	0	3m

- b If the pods are in a CLBO (crash loop backoff), retry the registration procedure and verify you're using the **Tanzu Kubernetes Grid** registration procedure instead of the **vSphere with Tanzu** registration procedure.

- c If any of the pods are in a failed state, fetch the logs for the failed pod to learn more information about the problem. For example:

```
kubectl logs -l "tmc-extension-name=extension-updater" -n vmware-system-tmc
```

For help solving pod failures, see [Common Log Errors and Solutions for Pod Failure](#) in the [Examples](#) section of this page.

- d If the pods have not been scheduled (they appear as “pending” or not at all), inspect the deployments of extension-manager, extension-updater and agent-updater to find any deployments that are not in a ready state.

```
kubectl get deployments -n vmware-system-tmc -l "tmc-extension-name in (extension-manager,extension-updater,agent-updater) "
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
agent-updater	1/11	1	3m	
extension-manager	1/11	1	3m	
extension-updater	1/11	1	3m	

- e If any of the deployments show as 0/1 , describe the deployment to understand how to correct the issue. For example:

```
kubectl describe deployment extension-updater -n vmware-system-tmc
```

These kubernetes errors are usually resource related. For example, you might get `<output>OutOfDisk`.

Ask your kubernetes administrator to solve these types of errors by expanding resources or relaxing policies.

- 5 If the deployments are all showing 1/1 READY, look for `Events` messages in the replicaset to understand how to correct the issue.

- a Look for replicaset that are not in the ready status. For example:

```
kubectl get replicaset -n vmware-system-tmc -l "tmc-extension-name in (extension-manager,extension-updater,agent-updater) "
```

- b Inspect any replicaset that are not in ready status and check the events messages in the replicaset to understand how to correct the issue. For example:

```
kubectl describe replicaset -n vmware-system-tmc extension-updater-6c4c95c5bf
```

The `Events` messages should contain enough information for you to solve the issue.

Example

Table 30-2. Common Log Errors and Solutions for Pod Failure

Application	Log	Solutions
Extension-Updater	Failed to discover cluster metadata	Make sure your cluster is CNCF conformant. You can check the CNCF site for a list of Platform Certified Kubernetes - Distribution Providers .
Extension-Updater	Failed to set up manager:get Tanzu Mission Control connection	<ol style="list-style-type: none"> 1 Make sure the cluster has internet connectivity. 2 If you're behind a firewall, set up or validate the configuration of your proxy. See Connecting through a Proxy for more information.
Extension-Updater	Pre-flight checks failed Or something like tmc.extension- flag.register.tkg.version.bloc k.aws: 1.3.0,1.3.1,1.4.0	Make sure your cluster is using a version of kubernetes that is supported by Tanzu Mission Control.

What to do next

If you can't solve the issue on your own, you can create a log bundle to share with support. Run the following commands against your management cluster:

- 1 Get the logs of the extension-updater `kubectl logs -n vmware-system-tmc -l "tmc-extension-name=extension-updater"`
- 2 Get the logs of the extension-manager `kubectl logs -n vmware-system-tmc -l "tmc-extension-name=extension-manager"`
- 3 Save the logs and create a bundle to share with support.

See [How to Get Support](#) if you need more information about contacting them.

Issues Registering a Supervisor Cluster

If you're having trouble registering a Supervisor Cluster running in vSphere with Tanzu with Tanzu Mission Control, there are a number of steps you can take to identify the root cause and fix the problem.

Problem

You might suspect you're having trouble registering your Supervisor Cluster if:

- When registering your Supervisor Cluster, the verification step fails.
- If you're registering a Supervisor Cluster through the CLI, the operation times out.

- If you're registering a Supervisor Cluster through the API, the status of the cluster remains pending for longer than 7 minutes.

Cause

Cause

There are a few common reasons you might have trouble registering your Supervisor Cluster with Tanzu Mission Control. You can follow the solution described to identify which of these causes is the issue and help you correct it.

- You might need to rerun the registration step if the agent installation failed. Common reasons for failure:
 - You might have tried to attach instead of register the Supervisor Cluster.
 - You might have tried to register a Supervisor Cluster using the Tanzu Kubernetes Grid procedure and seen a message like "no such resource found."
- You might have a resource scheduling issue or permission issue preventing the Tanzu Mission Control agent from installing successfully.
- You might have a problem with connectivity or proxy configuration.

Solution

- 1 Check the format of the registration URL to verify you're using the procedure to register a vSphere with Tanzu. You should see `source=registration` and `type=tkgs`. For example:

```
https://myorgname.tmc.cloud.vmware.com/installer?id=1234&source=registration&type=tkgs.
```

If you see `attach` or `tkgm` in the URL, you need to restart the registration process. Make sure you follow the procedure described in [Register a Management Cluster with Tanzu Mission Control](#).

- 2 Check if the problem is your access rights or network connection to the by running a simple `kubectl` command like `get namespaces`.

```
kubectl get namespaces
```

If you cannot access the cluster, contact your kubernetes administrator.

- 3 Verify that the `tmc` namespace exists. The `tmc` namespace is in the form of `svc-tmc-N`.

```
kubectl get namespaces | grep svc-tmc
```

```
svc-tmc-c8           Active   30h
```

If a `tmc` namespace doesn't exist, you might be registering an unsupported version of Tanzu Kubernetes Grid.

4 Check if the problem is the agent installation failure. Reinstall the agent if the installation failed.

- a Verify if the agent install succeeded by checking if AgentInstall exists.

```
kubectl get agentinstall -n svc-tmc-aN
```

NAME	AGE
tmc-agent-installer-config	4s

If the AgentInstall doesn't exist, try again to [register your supervisor cluster](#).

- b Check the AgentInstall status field to understand if there were any issues with the installation and how to correct them.

```
kubectl describe agentinstall -n svc-tmc-aN tmc-agent-installer-config
```

```
Name:          tmc-agent-installer-config
Namespace:     svc-tmc-aN
...
Status:
Message:       Installation inprogress
Retry Count:   5
Status:        INSTALLATION_IN_PROGRESS
Events:        <none>
```

5 Check if there was an issue applying the agent configuration.

```
kubectl get configmap -n svc-tmc-aN stack-config tmc-agent-installer-config
```

NAME	DATA	AGE
stack-config	9	9d

If the agent configuration doesn't exist:

- Delete the agent `kubectl delete AgentInstall -n svc-tmc-aN tmc-agent-installer-config`
- Try again to [register your supervisor cluster](#).

6 Investigate the agent pods to locate the source of the problem.

- a Check the status of the agent pods.

```
kubectl get pods -n svc-tmc-aN -l "tmc-extension-name in (extension-manager,extension-updater,agent-updater) "
```

NAME	READY	STATUS	RESTARTS	AGE
agent-updater-cc9c67b4d-sp6q9	0/1	CLBO	0	3m
agentupdater-workload-27478576-8rxvk	0/1	CLBO	0	60s
extension-manager-85f9cf5497-rk8ms	0/1	CLBO	0	3m
extension-updater-6c4c95c5bf-9wl6p	0/1	CLBO	0	3m

- b If the pods are in a CLBO (crash loop backoff), retry the [registration procedure](#) and verify you're using the **vSphere with Tanzu** registration procedure instead of the **Tanzu Kubernetes Grid** registration procedure.
- c If any of the pods are in a failed state, fetch the logs for the failed pod to learn more information about the problem. For example:

```
kubectl logs -l "tmc-extension-name=extension-updater" -n svc-tmc-aN
```

For help solving pod failures, see [Common Log Errors and Solutions for Pod Failure](#) in the [Examples](#) section of this page.

- d If the pods have not been scheduled (they appear as “pending” or not at all), inspect the deployments of extension-manager, extension-updater and agent-updater to find any deployments that are not in a ready state.

```
kubectl get deployments -n svc-tmc-aN -l "tmc-extension-name in (extension-
manager,extension-updater,agent-updater) "
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
agent-updater	1/11	1	3m	
extension-manager	1/11	1	3m	
extension-updater	1/11	1	3m	

- e If any of the deployments show as 0/1 , describe the deployment to understand how to correct the issue. For example:

```
kubectl describe deployment extension-updater -n svc-tmc-aN
```

These kubernetes errors are usually resource related. For example, you might get <output>OutOfDisk.

Ask your kubernetes administrator to solve these types of errors by expanding resources or relaxing policies.

- 7 If the deployments are all showing 1/1 READY, look for Events messages in the replicaset to understand how to correct the issue.

- a Look for replicaset that are not in the ready status. For example:

```
kubectl get replicaset -n svc-tmc-aN -l "tmc-extension-name in (extension-
manager,extension-updater,agent-updater) "
```

- b Inspect any replicaset that are not in ready status and check the events messages in the replicaset to understand how to correct the issue. For example:

```
kubectl describe replicaset -n vmware-system-tmc extension-updater-6c4c95c5bf
```

The Events messages should contain enough information for you to solve the issue.

Example

The following tables contain examples of common errors and how to solve them.

Table 30-3. How to Solve Common AgentInstall Status Messages

Status	Message	Solutions
INSTALLATION_IN_PROGRESS	Installation in progress	Wait until the installation completes.
INSTALL_FAILED	Registration link domain name verification failed	Check the registration link. Make sure you see something like, <code>https://myorgname.tmc.cloud.vmware.com/installer?id=1234&source=registration&type=tkgs</code> .
INSTALL_FAILED	Registration link is not an HTTPS link	Check the registration link. Make sure you see something like, <code>https://myorgname.tmc.cloud.vmware.com/installer?id=1234&source=registration&type=tkgs</code> .
INSTALL_FAILED	Get proxy options	Verify the proxy configured in the cluster.
INSTALL_FAILED	Download manifest from link	Make sure you downloaded and prepared the YAML from the Tanzu Mission Control Console as described in Complete the Registration of a Supervisor Cluster in vSphere with Tanzu . You might need to retry your registration procedure from the beginning.
INSTALL_FAILED	Parse manifest	Make sure you downloaded and prepared the YAML from the Tanzu Mission Control Console as described in Complete the Registration of a Supervisor Cluster in vSphere with Tanzu . You might need to retry your registration procedure from the beginning.
INSTALL_FAILED	Apply manifest	It's likely there aren't enough resources available for the agent to install successfully. Contact your vSphere administrator for help.

Table 30-4. Common Log Errors and Solutions for Pod Failure

Application	Log	Solutions
Extension-Updater	Failed to discover cluster metadata	Make sure your cluster is CNCF conformant. You can check the CNCF site for a list of Platform Certified Kubernetes - Distribution Providers .
Extension-Updater	Failed to set up manager:get Tanzu Mission Control connection	<ol style="list-style-type: none"> 1 Make sure the cluster has internet connectivity. 2 If you're behind a firewall, set up or validate the configuration of your proxy. See Connecting through a Proxy for more information.
Extension-Updater	Pre-flight checks failed Or something like tmc.extension- flag.register.tkg.version.block: 1.4.0	Make sure your cluster is using a version of kubernetes that is supported by Tanzu Mission Control.

What to do next

If you can't solve the issue on your own, you can create a log bundle to share with support. Run the following commands against your management cluster:

- 1 Get the logs of the extension-updater `kubectl logs -n svc-tmc-aN -l "tmc-extension-name=extension-updater"`
- 2 Get the logs of the extension-manager `<userinput> kubectl logs -n svc-tmc-aN -l "tmc-extension-name=extension-manager"`
- 3 Save the logs and create a bundle to share with support.

See [How to Get Support](#) if you need more information about contacting them.

Workload Cluster Provision Failures

Use these steps to troubleshoot failures of workload clusters.

Problem

Workload clusters are not being provisioned.

Cause

- Credentials may not be available.
- Extensions are not healthy.
- Status has not been accurately updated.
- Provisioning is not completed, but still ongoing.

- There could be issues with memory and CPU usage for the extensions.
- There may be issues with the Kubernetes pods.

Note The debug commands provided below are to be run in the management cluster.

Note The supervisor registration process differs from vSphere 7.X and vSphere 7.Xu3.

Solution

- 1 To view the status of a cluster, select **Clusters**.

Figure 30-2. Cluster Health

Name	Health	Status	Provider	Version	Requested/Allocatable memory	Requested/Allocatable CPU	Cluster group	Region
airnz-aws-ga	Disconnected	Unknown	AWS	1.18.3-1-amazon2	--	--	sudhakarf-grp	us-west-2
alicja-gke-cluster-1	Healthy	Ready	Google Cloud	v1.21.11-gke.1100	55% (4.34 GB/7.87 GB)	94% (2.66 CPUs/2.82 CPUs)	alicja-eks-group	us-central1
asdfasdfs	Unknown	Pending	Unknown	--	--	--	default	Unknown
asdfasdfs	Unknown	Pending	Unknown	--	--	--	default	Unknown
ash-am-crazy	Unknown	Pending	Unknown	--	--	--	default	Unknown
ashthyne-kind	Unknown	Detaching	Unknown	v1.17.0	--	--	default	Unknown
ashthyne-ikgs-test	Unknown	Creating	Unknown	v1.20.2-vmware.1-...	--	--	default	Unknown
bare-metal	Unknown	Unknown	Unknown	v1.15.1	--	--	finnerand	Unknown
bbbrundert-kind-local	Unknown	Pending	Unknown	--	--	--	bbbrundert-group	Unknown
bbbrundert-kg-cluster-1	Disconnected	Unknown	vSphere	v1.17.11-vmware.1	--	--	bbbrundert-group	Unknown
bitnami-cicd3	Warning	Ready	AWS	1.18.18-1-amazon2	5% (2.50 GB/45.82 GB)	30% (3.62 CPUs/12 CPUs)	bitnami-cluster-tmc	us-east-1
bitnami-cicd-carvel	Healthy	Ready	AWS	1.18.18-1-amazon2	7% (2.50 GB/38.24 GB)	34% (3.37 CPUs/10 CPUs)	bitnami-cluster-tmc	us-east-1
bitnami-manual-test	Healthy	Ready	AWS	1.19.8-2-amazon2	5% (2.52 GB/46.93 GB)	25% (2.97 CPUs/12 CPUs)	bitnami-cluster-tmc	us-east-1
bjung-eks	Disconnected	Unknown	AWS	v1.19.15-eks-9c63c4	--	--	default	us-west-2
bitnami-manual-test	Unknown	Unknown	Unknown	--	--	--	default	Unknown

The status of the cluster is shown next to its name at the top of the screen. In addition to an Unhealthy status, if it says Unknown, that may indicate an issue as well because processes may still be running and may time out.

You can also see the health of the components, agents, and extensions on this screen. See Base Extensions below for a complete list of extensions.

Also, the component health for controller manager and scheduler are known to be blank for provisioned clusters. You can double check this on the **Cluster** details page.

Maybe the cluster came up, but the extensions might be unhealthy.

- 2 If the extensions are not showing as healthy, you can log in to the management cluster and check the status of the pods directly:
 - a Check the Pods are not in an Error State. Are any pods failing to come up? Check the pod statuses with this command:

```
kubectl get pods -n vmware-system-tmc
```

Example failure statuses include: "CrashLoopBackOff," "CreateContainerError," and "Error."

- b If you see a pod that is failing to come up, check the logs for errors. First, export the name of the failing pod as an environment variable:

```
export PODNAME=<pod-name>
```

- c Next, check the logs for error messages:

```
kubectl logs $PODNAME -n vmware-system-tmc | grep -i error
```

- d Get the full log output and redirect it to a file locally in case you need to open a support ticket later

```
kubectl logs $PODNAME -n vmware-system-tmc > $PODNAME-logs.txt
```

Note If the pod is in "CrashLoopBackOff" it means it is continually restarting, and you may need to wait for the next restart to capture the logs.

Base Extensions

- agent-updater
- cluster-auth-pinniped
- cluster-health-extension
- cluster-secret
- extension-manager
- extension-updater
- gatekeeper-operator
- inspection
- intent-agent
- package-deployment
- policy-insight-extension
- policy-sync-extension
- sync-agent

- `tmc-observer`

3 Restart deployments where pods have not come up successfully.

- a First, get the name of the deployment associated with the pod you want to restart.
- b Check all deployments in the namespace by running the following command and then do a safe restart of all pods in that deployment:

```
kubectl get deploy -n vmware-system-tmc
```

- c Export the deployment name as an environment variable.

```
export DEPLOYNAME=<deployment-name>
```

- d Restart all pods in the deployment:

```
kubectl rollout restart deploy/$DEPLOYNAME
```

At this stage, if the issue is not resolved, open a support ticket, making sure to include the full pod logs that you saved to a file earlier in step 2.d.

- 4 When creating a management cluster, you may see issues in the create screen. If they are errors that Tanzu Mission Control is aware of, for example, “AWS credentials not available” or “API Error: Failed to create cluster: management cluster or intent agent is not healthy (failed precondition),” these errors should have enough information to solve the issues. If not, collect the logs as described in step 2 above for including in a support ticket.
- 5 When creating a cluster and working in the **Configure** step, there might not be any Kubernetes versions listed, and it comes up as an empty drop down. This is because Tanzu Mission Control filters out the unsupported versions of Kubernetes, or because there are no images available to use in vSphere. Another example is Region, which might not support certain instance types. In this case you need to check your management cluster for issues.
 - a In Chrome or Firefox, open the **Network** view in **Inspect**, and select **Preserve Log** to ensure that you capture responses as new requests come through, then, go through the steps in the create cluster wizard to see the full API responses returned.
 - b Another option to using Network Inspect through the browser is to run the command `kubectl get vsphereoptions options -n vmware-system-tmc -o yaml`. The results of this command are what’s used to populate the **Create Cluster** page.

Note The example command given above is for vSphere specifically, and you must adjust it for other infrastructure providers as necessary.

To see a full list of all options CRDS for the various infrastructures run `kubectl get crds -n vmware-system-tmc | grep -I option`.

- c You can then run the command:

```
kubectl logs deploy/resource-retriever -n vmware-system-tmc
```

Once you have completed the **Create** page and click **Create** you get a message indicating that the cluster is being created, with a status bar.

- Sometimes it may seem to be taking a long time to create or is failing to create. It does take some time, but you can view logs to see what is occurring using the following commands.

```
kubectl logs deploy/lcm-tkg-extension -n vmware-system-tmc
```

- Sometimes during cluster creation there may be no obvious API error, and it may be reported as Ready when it is not. You can check the management cluster to see the actual status. To check the status of the cluster directly in the management cluster run:

```
kubectl get clusters -A
```

For further details on a particular cluster you can run:

```
kubectl describe cluster <cluster-name>
```

If the cluster is Ready in the management cluster but not showing as Ready in the administration console, this is an issue with the status being reported back to the cluster. You can try restarting the sync-agent and the cluster-agent pods and checking configurations to see where the failure to sync the status back is happening.

- Different extensions might be installed based on the cloud service provider. You can check to confirm that the correct extensions are being installed for your platform.
- Sometimes there is nothing wrong with the cluster, but status is not being sent back to Tanzu Mission Control and is not showing up or being updated. It could be a network communications issue. Check the cluster health extension, sync agent, and intent agent.
- There might be issues with memory and CPU usage for the extensions. For information about memory and CPU usage, see [Memory and CPU Usage by Cluster Agent Extensions in VMware Tanzu Mission Control Concepts](#).

Troubleshooting Issues with Tanzu Application Platform

Learn how to troubleshoot some common issues associated with installing Tanzu Application Platform (commonly called TAP) in Tanzu Mission Control.

You might encounter some of the following issues while installing Tanzu Application Platform on a cluster through Tanzu Mission Control.

Tanzu Application Platform Packages Are Not Installed

Under certain conditions you may have issues with installing Tanzu Application Platform packages.

Problem

Tanzu Application Platform packages are not getting installed.

Cause

There can be certain issues with the configuration provided which can result in failure for some packages.

Solution

- 1 Check package status.

From the below command, you can get the namespace created by the package where the resources could be failing and query for all the resources under that namespace:

```
kubectl describe pkgi <failed-package> -n tap-install
```

```
kubectl describe apps <failed-package> -n tap-install
```

- 2 Check tap resources.

Check for all the failed resources or get the logs in case of pods.

```
kubectl get all -n <ns-specific-to-a-package>
```

Failed Disk Space Check for Tanzu Kubernetes Grid Cluster

You might encounter issues about insufficient disk space when using Tanzu Mission Control to install Tanzu Application Platform (commonly called TAP) on a managed Tanzu Kubernetes Grid (TKG) cluster.

Problem

While installing TAP, a prerequisite check for sufficient ephemeral storage is failing on Tanzu Kubernetes Grid (TKG) cluster nodes.

Cause

Insufficient allocatable ephemeral storage on cluster nodes.

Solution

- 1 Check resources on your target cluster; set `kubeconfig` for your target cluster, and then run the following command to get the status of your cluster's allocatable resources.

```
kubectl get nodes -o=custom-  
columns=NAME:.metadata.name,ALLOCATABLE_CPU:.status.allocatable.cpu,ALLOCATABLE_MEMORY:.sta  
tus.allocatable.memory,ALLOCATABLE_DISK:.status.allocatable.ephemeral-storage
```

- 2 Verify that your cluster nodes have the resources required to install TAP full profile. For more information, see [Prerequisites for installing Tanzu Application Platform](#) in the *VMware Tanzu Application Platform Documentation*.

- 3 If allocatable disk space is lesser than the minimum required disk space of 100 Gi per node, verify your node-pools configurations. Note that disk space allocated to `/var/lib/kubelet` mount point is referred to as ephemeral storage and reported by Kubernetes as allocatable disk space.
 - a It is recommended to allocate disk space higher than 100Gi, to accommodate other non-TAP payloads, as other workloads might consume disk space as well.
 - b Following is an example of node pool volume configuration for two supported mount points, `/var/lib/kubelet` and `/var/lib/containerd`; 125Gi is allocated to `/var/lib/kubelet` (higher than 100 Gi is required; scale the size per your payload requirements). Also, reference your `$STORAGE_CLASS` as per environment.

```

- name: nodePoolVolumes
  value:
  - capacity:
      storage: "125Gi"
      mountPath: "/var/lib/kubelet"
      name: kubelet
      storageClass: $STORAGE_CLASS
  - capacity:
      storage: "100Gi"
      mountPath: "/var/lib/containerd"
      name: containerd
      storageClass: $STORAGE_CLASS
- name: controlPlaneVolumes
  value:
  - capacity:
      storage: "125Gi"
      mountPath: "/var/lib/kubelet"
      name: kubelet
      storageClass: $STORAGE_CLASS
  - capacity:
      storage: "100Gi"
      mountPath: "/var/lib/containerd"
      name: containerd

```

Tanzu Developer Portal does not show cluster resources in multi-cluster TAP 1.7.3 deployment

When the Tanzu Application Platform (TAP) v1.7.3 multi-cluster solution is installed through Tanzu Mission Control, resources on the Build, Iterate, and Run clusters are not visible in the Tanzu Developer Portal.

Problem

When you perform a multi-cluster deployment of TAP version 1.7.3 through Tanzu Mission Control, the Tanzu Developer Portal is unable to view resources from the Build, Run, and Iterate clusters.

Cause

The configuration of the View cluster does not allow it to access resources in the other clusters.

Solution

Follow the instructions in [Update Tanzu Developer Portal to view resources on multiple clusters](#) (in the *VMware Tanzu Application Platform Documentation*) to update the configuration of the View cluster to enable visibility into the the Build, Run, and Iterate clusters.

Troubleshooting Issues with Data Protection

Learn how to troubleshoot some common issues associated with data protection features in Tanzu Mission Control.

You might encounter some of the following issues while using the data protection features of Tanzu Mission Control.

Troubleshooting Issues with FSB or CSI

Use this information to troubleshoot issues you might encounter when using FSB or CSI with Tanzu Mission Control data protection features.

Problem

Enabling or disabling of an FSB or CSI volume plugin remains in the processing state for an extended period of time (more than 15 minutes).

Cause

Possible causes include:

- Cluster is disconnected.
- Backup or restore operation is in progress.
- The Tanzu Mission Control sync agent pod is not running.

Solution

- Plugin configuration changes cannot be applied to a disconnected cluster. Make sure the cluster is in a `Healthy` state.
- Wait until in-progress backups or restores are complete. To avoid interrupting ongoing backup/restore operations, the plugin configuration change is applied only after their completion.

- Make sure the Tanzu Mission Control sync agent pod is up and running.

```
kubectl -n vmware-system-tmc get pod | grep sync-agent
```

```
sync-agent-5dd847f6f5-52nfr      1/1      Running
0              70m
```

Try manually bouncing the sync agent pods:

```
kubectl -n vmware-system-tmc rollout restart deploy sync-agent
```

Unable to download backup or restore logs

Problem

When attempting to download logs or results from backup or restore operations, you might see the following error:

```
Your request header section exceeds the maximum allowed size
```

Cause

This error typically results from a browser cache issue.

Solution

- 1 Try using a different browser or open a new browser window in private mode.
- 2 Clean up the browser cookies associated with VMware.
- 3 Try downloading your logs or results again.

Unable to download failed backup logs

Problem

For a failed backup operation, the options to download logs and download resource list are disabled.

Cause

This situation is typically the result of nothing being uploaded into the bucket. Because logs and other details are fetched from the bucket you designated for the backup, these actions are disabled when the bucket has no contents and therefore there is nothing to download.

Solution

- 1 Connect to your cluster with kubectl.

- Retrieve the details about the backup operation, for example:

```
kubectl exec -it deploy/velero -n velero -- /velero describe backup my-backup-name --
details
```

- Check the velero pod logs, for example:

```
kubectl -n velero logs velero-pod-name | grep my-backup-name
```

Backups Partially Fail Due to Existing Lock

You can take one of these steps when a backup partially fails because of an existing lock.

Problem

When using filesystem volume backups (FSB), you observe something similar to below in the backup logs:

```
time="2023-03-27T04:01:04Z" level=error msg="Error backing up item" backup=velero/bk-entirecluster-daily-20230327040009 error="pod volume backup failed: running Restic backup, stderr=unable to create lock in backend: repository is already locked exclusively by PID 12576 on velero-7fdc5bff66-z88k8 by nonroot (UID 65532, GID 65532)\nlock was created at 2023-03-27 04:01:01 (2.490799836s ago)\nstorage ID 811e7acc\nthe `unlock` command can be used to remove stale locks\n: exit status 1" error.file="/go/src/github.com/vmware-tanzu/velero/pkg/restic/backupper.go:199" error.function="github.com/vmware-tanzu/velero/pkg/restic.(*backupper).BackupPodVolumes" logSource="pkg/backup/backup.go:417" name=airflow-db-migrations-56c64bd87d-mk9rc
```

You can confirm that the volume backups failed by using the Velero CLI to describe the backup with the details flag and checking under the "Restic Backups" heading.

```
$ velero backup describe <backup name> --details
```

Restic Backups:

Completed:

```
tanzu-system-dashboards/grafana-658c5dbc77-xphdz: sc-dashboard-volume, sc-datasources-volume, storage
```

```
tanzu-system-ingress/envoy-l8f5f: envoy-admin, envoy-config
```

```
tanzu-system-ingress/envoy-lx6jc: envoy-admin, envoy-config
```

```
tanzu-system-ingress/envoy-vb24x: envoy-admin, envoy-config
```

```
tanzu-system-ingress/envoy-x5vwk: envoy-admin, envoy-config
```

```
tanzu-system-monitoring/alertmanager-6546bb6b6d-s1967: storage-volume
```

```
tanzu-system-monitoring/prometheus-server-746cc78b85-vm6pn: storage-volume
```

Failed:

```
airflow/airflow-db-migrations-56c64bd87d-mk9rc: dags-data, logs-data
```

```
airflow/airflow-flower-76dfc68945-f9x4b: dags-data, logs-data
```

Finally, you should see evidence of the `prune` command being killed via signal. (This log will not be present in the backup logs but only in the velero pod logs.)

```
time="2023-03-27T16:49:05Z" level=warning msg="error pruning repository" error="error running command=restic prune --repo=s3:http://111.222.333.111/example-bkt/01G2Q0XXBKJ0GM7E7V34Q85PT/restic/airflow --password-file=/tmp/credentials/velero/velero-restic-credentials-repository-
```

```
password --cache-dir=/scratch/.cache/restic, stdout=loading indexes...\nloading all
snapshots...\nfinding data that is still in use for 355 snapshots\n[0:22] 100.00%
355 / 355 snapshots\n\nsearching used packs...\ncollecting packs for deletion and
repacking\n[0:00] 100.00% 529 / 529 packs processed\n\n\nnto repack:          106126 blobs /
98.205 MiB\nthis removes:          1540 blobs / 54.810 MiB\nnto delete:          908753 blobs /
639.015 MiB\ntotal prune:          910293 blobs / 693.825 MiB\nremaining:          311670 blobs /
2.882 GiB\nunused size after prune: 439.412 MiB (14.89% of remaining size)\n\nrepacking
packs\n, stderr=: signal: killed" error.file="/go/src/github.com/vmware-tanzu/velero/pkg/
restic/repository_manager.go:296" error.function="github.com/vmware-tanzu/velero/pkg/restic.
(*repositoryManager).exec" logSource="pkg/controller/restic_repository_controller.go:198"
resticRepo=velero/airflow-msk-pure-s3-bkt-ls6pk
```

Cause

Velero periodically "prunes" each restic repository to compact disk space. Under the hood, Velero cleans up stale locks and spawns a child process to run the `prune` command. The `prune` command acquires locks but then is subsequently OOM killed most likely due to insufficient resources given to the Velero container. This cycle continues so the repo is continuously locked. And when the backup is created, Restic fails to take a snapshot due to the existing lock.

There is a [velero issue](#) tracking the fix to not repeatedly attempt the `prune` command and thus prevent the Restic repository from being perpetually locked.

Solution

- ◆ You can use one of the following options to fix the issue:

Option	Description
Use CSI volume snapshots.	You may choose to back up your volumes using CSI volume snapshots instead of filesystem level volume backups. When creating a backup, select the option to perform CSI volume snapshots and use the opt-in approach for filesystem backups.
Increase the memory limit for the velero deployment.	<p>Velero maintains a restic "repository" for each namespace. The amount of memory required to perform the <code>prune</code> command is proportional to the size of the restic repository's index. So the solution is to increase the memory limits of the Velero container in the Velero deployment until the <code>prune</code> command has sufficient memory to run. (Note that the <code>prune</code> command is run by the Velero pod and not the <code>node-agent</code> pods.)</p> <p>Use the following command to edit the deployment (you don't need to update the resources under the <code>init</code> containers):</p> <pre>\$ kubectl -n velero edit deployment velero ... resources: limits: cpu: "1" memory: 512Mi <--- change this requests: cpu: 500m memory: 128Mi ...</pre> <p>Wait for a few minutes and check if the <code>prune</code> errors are still seen:</p> <pre>\$ kubectl -n velero logs <velero pod name> grep prune grep killed</pre> <p>If they're still present, try again by increasing the memory limits to a higher value.</p>

Source File Not Found During Backup

You can use these options to deal with missing source files during backups.

Problem

When using filesystem level volume backups (FSB), you observe something similar in the backup logs as shown below, especially for applications like ElasticSearch.

```
time="2023-07-07T02:11:30Z" level=error msg="Error backing up item" backup=velero/
sharedservice-full-cls-backup-daily-20230707000048 error="pod volume backup failed: running
backup, stderr=error running restic backup command restic backup --repo=s3:https://s3-ap-
southeast-1.amazonaws.com/tmc-cluster-backup-prod/01GB1KBTPXVZACM1NKZPXWD2YR/restic/prod-o1ly
--password-file=/tmp/credentials/velero/velero-repo-credentials-repository-password --cache-
dir=/scratch/.cache/restic . --tag=backup=sharedservice-full-cls-backup-daily-20230707000048
--tag=backup-uid=ab09902b-9d65-49b9-bc9b-3153a30d01ba --tag=ns=prod-o1ly --
tag=pod=elasticsearch-data-1 --tag=pod-uid=8a1bd1f5-ea5c-41ce-a170-28cf4ad18a8b --tag=pvc-
uid=4a8ba939-8bd0-443b-91aa-cb0a36066586 --tag=volume=data --host=velero --json --
```

```
parent=a9f71e50 with error: exit status 3 stderr: {"message_type": "error", "error": {"Op": "lstat", "Path": "indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1.fnm", "Err": 2}, "during": "archival", "item": "/host_pods/8a1bdlf5-ea5c-41ce-a170-28cf4ad18a8b/volumes/kubernetes.io~csi/pvc-4a8ba939-8bd0-443b-91aa-cb0a36066586/mount/indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1.fnm"}\n{"message_type": "error", "error": {"Op": "lstat", "Path": "indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1_Lucene90_0.dvd", "Err": 2}, "during": "archival", "item": "/host_pods/8a1bdlf5-ea5c-41ce-a170-28cf4ad18a8b/volumes/kubernetes.io~csi/pvc-4a8ba939-8bd0-443b-91aa-cb0a36066586/mount/indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1_Lucene90_0.dvd"}\n{"message_type": "error", "error": {"Op": "lstat", "Path": "indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1_Lucene90_0.dvm", "Err": 2}, "during": "archival", "item": "/host_pods/8a1bdlf5-ea5c-41ce-a170-28cf4ad18a8b/volumes/kubernetes.io~csi/pvc-4a8ba939-8bd0-443b-91aa-cb0a36066586/mount/indices/IF-3K2QkQLSG3-ZyKka1Nw/0/index/_2x79j_1_Lucene90_0.dvm"}\nWarning: at least one source file could not be read\n: error running restic
```

Cause

This happens when there is churn in the filesystem and the file is no longer present while performing pod volume backup.

Solution

- ◆ Use one of the following options:

Option	Description
Use CSI volume snapshots.	You may choose to back up your volumes using CSI volume snapshots instead of filesystem level volume backups. When creating a backup, select the option to perform CSI volume snapshots and use the opt-in approach for filesystem backups.
Use the opt-out approach.	If CSI volume snapshot is not supported in your cluster, then exclude the volume using the opt-out approach for filesystem backups.
Use an application specific backup method.	Some applications provide their own method to perform a backup. For example, backup of ElasticSearch using filesystem backup is not recommended. For information about backing up ElasticSearch, see https://www.elastic.co/guide/en/elasticsearch/reference/8.2/snapshot-restore.html#other-backup-methods .

Backups Partially Fail When Volumes Present on Control Plane Nodes

You can take these steps when a backup partially fails when volumes are present on the control plane nodes.

Problem

When using filesystem level volume backups (FSB), the overall backup ends up partially failed and a log similar to the following is present:

```
time="2023-07-18T07:48:39Z" level=error msg="Error backing up item" backup=velero/bk-2 error="daemonset pod not found in running state in node interop-
```

```
fresh-kind124-control-plane" error.file="/go/src/github.com/vmware-tanzu/velero/pkg/nodeagent/node_agent.go:74" error.function=github.com/vmware-tanzu/velero/pkg/nodeagent.IsRunningInNode logSource="pkg/backup/backup.go:425" name=app-0
```

Cause

Velero's filesystem backup integration works by creating a daemonset called node-agent and each of its pods is responsible for backing up volumes on its own node. By default, daemonset pods are only scheduled on worker nodes. This is expected behavior on most clusters since the control plane nodes are tainted by default.

In the unlikely scenario that there are pods with volumes present on the control plane nodes, there is no daemonset pod present to complete the volume backup and so these volumes are skipped and the overall backup partially fails. Note that this includes even empty Dir volumes.

Solution

- ◆ Use one of the following options to address the issue:

Note The following assumes that you have configured filesystem backups with the `opt-out` option.

Option	Description
Use CSI volume snapshots.	CSI volume snapshots do not have any such node level limitation as they do not rely on the node-agent daemonset. While creating a backup, select the option to perform CSI volume snapshots and use the opt-in approach for filesystem backups.
Exclude the namespace containing the problematic volumes.	In the Create Backup page, under Advanced options, add the namespace under Excluded namespaces. Note that this prevents all resources in the namespace from being backed up and not just volumes.
Exclude the problematic pod volumes.	Annotate the pods containing the problematic volumes with <code>backup.velero.io/backup-volumes-excludes=vol1,vol2,vol3</code> . Note that the volume names in the annotation would be those in the pod manifest under <code>.spec.volumes</code> and not the PVC or PV names. Also, in case the pods were created via another resource such as a deployment or daemonset, it may be a good idea to configure the parent resource such that spawned pods receive the annotation (<code>.spec.template.metadata.annotations</code> in case of deployments/daemonsets). This ensures that the annotation persists even if the pod is recreated.

Option	Description
Exclude the problematic pods.	Similar to the above option, you could exclude the pod plus all its volumes from being backed up by labeling it with <code>velero.io/exclude-from-backup=true</code> .
Tolerate the taint for the node-agent daemonset.	<p>Edit the daemonset with:</p> <pre>kubectl -n velero edit ds node-agent</pre> <p>Add the following under <code>.spec.template.spec</code>.</p> <pre>tolerations: - key: node-role.kubernetes.io/control-plane operator: Exists effect: NoSchedule - key: node-role.kubernetes.io/master operator: Exists effect: NoSchedule</pre> <p>Note that the exact taint key used could differ across clusters. In the end, verify that the daemonset pods were spawned on every node in the cluster.</p>

Velero namespace stuck Terminating

You may encounter this issue when Data Protection is disabled on a cluster or when a cluster is detached from Tanzu Mission Control.

Problem

The Velero namespace is stuck in the Terminating phase either when Data Protection is disabled on a cluster or when a cluster is detached from Tanzu Mission Control.

Cause

Starting with Velero 1.12, certain resources like restores now make use of finalizers. In order for the Velero namespace to finish deleting, all finalizers for resources within the namespace must be removed. During disablement of Data Protection on a cluster or during the detach of a cluster from TMC, in rare situations, the Velero pods as well as the TMC data protection pods may be cleaned up before getting a chance to remove all the finalizers.

Solution

- ◆ Use one of the following solutions:
 - Remove finalizers from each resource. Look for resources still present in the Terminating namespace.:

```
kubectl -n velero get restores
kubectl -n velero get datauploads
kubectl -n velero get datadownloads
```

Remove the finalizers from each resource using `kubectl edit` or this one line command:

```
kubectl -n velero patch restore <restore name> -p '{"metadata":{"finalizers":null}}'
--type=merge
```

- Force delete the namespace:

```
kubectl get ns velero -o json | jq '.spec.finalizers = []' | kubectl replace --raw
"/api/v1/namespaces/velero/finalize" -f -
```

Restore operation is stuck in pending state

You may encounter this issue when a pod in the cluster expects a `schedulerName`.

Prerequisites

To fix this issue, you can create a resource modifier configmap in the `velero` namespace on the target cluster, and then reference it when performing the restore.

A resource modifier is a set of rules for applying patches to resources. The following example shows a resource modifier that removes the `schedulerName` field from pod specs during restore.

```
version: v1
resourceModifierRules:
- conditions:
  groupResource: pods
  patches:
  - operation: remove
    path: "/spec/schedulerName"
```

Problem

A restore procedure appears to be stuck in `pending state`.

Cause

This situation can arise when a pod in the backup has a `schedulerName` defined in the pod spec that is not present in the cluster to which it is being restored.

Solution

- 1 Create a YAML file (for example, `my-resource-modifier.yaml`) that defines the resource modifier as shown above.
- 2 Create the resource modifier configmap in the `velero` namespace on the target cluster. For example:

```
kubectl -n velero create cm my-res-mod-configmap --from-file my-resource-modifier.yaml
```

- 3 Add the name of the resource modifier configmap to your data values file. For example:

```
ClusterName: target-cluster
ManagementClusterName: attached
```

```

ProvisionerName: attached
BackupName: example-backup
BackupSourceClusterName: source-cluster
BackupSourceManagementClusterName: attached
BackupSourceProvisionerName: attached
ResourceModifierConfigMap: my-res-mod-configmap

```

4 Then retry the restore operation.

```
tanzu mission-control data-protection restore create -v ./data-values-file
```

Manage Issues with AWS EKS Credentials in Tanzu Mission Control

This topic provides information and guidance for understanding AWS credential states and how to troubleshoot credential issues related to management of EKS clusters and Tanzu Mission Control accounts required for use of the EKS lifecycle management features.

Credential Phases and AWS Resources

Credential phases can be one of the following:

Figure 30-3. EKS Credential Phases

```

const (
    // Unspecified phase.
    Status_PHASE_UNSPECIFIED Status_Phase = 0
    // The credential is created and can be used.
    Status_CREATED Status_Phase = 1
    // The credential's capabilities are being validated by the intended service.
    //
    // Credentials can be used even if they have not been validated- this phase
    // is set by the intended service if it validates credentials.
    Status_VALIDATING Status_Phase = 2
    // The credential satisfies the intended service's requirements.
    Status_VALID Status_Phase = 3
    // The credential does not satisfy the intended service's requirements.
    //
    // Invalid credentials might require user action to fix their permissions- this information
    // is provided by the intended service.
    Status_INVALID Status_Phase = 4
    // An error occurred while the credential was being created or validated.
    Status_ERROR Status_Phase = 5
    // The credential clean up has begun.
    Status_DELETING Status_Phase = 6
    // The credential clean up has completed and will be removed from TMC.
    Status_DELETED Status_Phase = 7
)

```

For more information about credential phases, see `vmware.tanzu.manage.v1alpha1.account.credential.Status` in [Tanzu Mission Control API Reference](#).

Note that in the Tanzu Mission Control console, the DELETING phase appears as REMOVING.

Once a credential has been created, Tanzu Mission Control creates the following in the region where the Cloud Formation stack was initially created.

- 1 Data sync lambda (responsible for syncing data about EKS clusters in the AWS account back to Tanzu Mission Control).
- 2 Resource retriever lambda (responsible for syncing data about AWS account from enabled regions back to Tanzu Mission Control - e.g., region, VPCs, subnets, etc.).
- 3 A CloudWatch event (for logging)
- 4 An AWS Systems Manager or SSM Parameter (containing an authentication token for communicating with Tanzu Mission Control).

Once all of these have been deployed successfully and the lambdas have been able to run and sync back to Tanzu Mission Control, only then is a credential marked as VALID. This is the desired state of a healthy and ready to use credential.

Credentials in CREATED State

If a credential is in a CREATED state this means that the create request has been successful, however, it does not mean that the credential is useable. It is going through the pre-validating phase and will be marked either VALIDATING or INVALID shortly. If a credential is in a CREATED state for 20 minutes or more, you will need to reach out to customer support for further assistance.

Credentials in VALIDATING State

If a credential is stuck in a VALIDATING state, this means that the lambdas deployed to the account have not been able to communicate with Tanzu Mission Control.

Check that each of the resources listed in the section above was created successfully.

- To check the lambdas, go to the AWS console and in the Services menu search for lambda in the region where the Cloud Formation stack was initially created:
 - Search by the tag `account.tmc.cloud.vmware.com=credential-name` to see the lambdas associated with your credential.
 - Check that both the data-sync and resource-retriever lambda for the credential have been created and have run successfully.
- Once you confirm the lambdas are there, you can check the CloudWatch logs and metrics for each of them:
 - Click into each lambda to open up the full view of each one.
 - Scroll down and open the Monitor tab, then selection Metrics, Logs or Trace as appropriate.

- You can also opt to select "View the Logs in CloudWatch" if you are more familiar with that view.
- To check via the AWS console that the SSM parameter is present in the region where the Cloud Formation stack was created initially:
 - Search for Systems Manager in the Services menu.
 - Go to **Application Management > Parameter Store**.
 - Search by the tag `account.tmc.cloud.vmware.com=credential-name` to see the parameter associated with your credential.
 - You won't be able to view the actual authentication token, but you can verify that it is present and when it was last updated by looking at the History tab.

Credentials in INVALID State

The table below provides information about error messages you may receive if a credential is in an INVALID state, along with suggested fixes for the errors.

Error Message	What does it mean?	Who can fix it?	Fix
Credential is not associated with Tanzu Mission Control	This means that Tanzu Mission Control was not set for the credential.	User (if using the API or CLI), or Customer Support if using the console	Ensure that the LCM capability is included in the specification of the credential request. For example: ... "capability": "MANAGED_K8S_PROVIDER" } ...
Failed to get credentials for account <i><account-name></i> with error <i><error></i>	The service failed to get the credential from Tanzu Mission Control	User	To workaround: Try deleting and recreating a new credential (with a different name, in case the bad state of the original credential causes issues with the deletion).
received empty account credentials	The meta section of the spec in the credential request is empty	User (if using the API or CLI), or Customer Support if using the console	Make sure that there is a provider specified in the meta section of the credential request. For example: ... "meta": { "provider": 5 }, ...

My credential was VALID but now it's INVALID - What happened?

Tanzu Mission Control continually reconciles the accounts (a.k.a credentials) associated with it. There are some use cases where a credential that was previously marked as VALID can be later marked as INVALID.

For example:

- If the cluster options have not been updated in a long time, then we know the lambdas in the AWS account have lost connectivity with Tanzu Mission Control. This could happen in the case of an AWS outage even, in which case when connectivity resumes, the credential will be marked VALID again. However, if there is no such outage, you need to check that the lambdas are set up correctly and working and that the Tanzu Mission Control authentication token in SSM is still valid.

Credentials in DELETING/REMOVING State

If a credential is stuck in a DELETING state (or in the case of console users, in a REMOVING state) then there may have been a problem with the credential deletion that caused it to fail.

When a credential is deleted, the account-controller deletes the four items that it created in the AWS account:

- Data sync lambda
- Resource retriever lambda
- CloudWatch event rule
- SSM parameter

If you encounter issues trying to delete these, contact Customer Support.

Getting more Information about the State of a Credential

While the console only provides the high level state of the credential (for example, CREATING, ERROR, REMOVING, etc.), the Tanzu Mission Control API provides more information by including conditions.

You can use the `List Credentials` API to see what the conditions are for your credential. (For more information about, Tanzu Mission Control APIs, see [Tanzu Mission Control API Reference](#).)

Example:

```
# Using the GET Credential API endpoint
# {{hostname}}/v1alpha1/account/credentials/{{credential_name}}

"status": {
  "phase": "VALID",
  "conditions": {
    "Ready": {
      "type": "Ready",
      "status": "TRUE",
      "severity": "INFO",
      "lastTransitionTime": "2022-10-05T20:58:06.131349Z",
      "reason": "CredentialCreated"
    },
    "Scheduled": {
      "type": "Scheduled",
      "status": "TRUE",
```

```
    "severity": "INFO",  
    "lastTransitionTime": "2022-10-05T20:58:06.131349Z",  
    "reason": "CreatingCredential"  
  }  
}  
}
```

Monitoring of AWS GuardDuty for Unauthorized Access

VMware Information Security Operations has recently started the monitoring of AWS GuardDuty for AWS.

VMware Information Security Operations has recently started the monitoring of AWS GuardDuty for AWS AccountID-888520879783 and opened SOC#194362 to track the issue.

- Details: The anonymous user system:anonymous was granted API permissions on the EKS cluster tags-test. This enables unauthenticated access to the permitted APIs. If this behavior is not expected, it may indicate a configuration mistake or that your credentials are compromised.

Links to Examples, Blogs, and More Information

31

A lot of information, including code examples and detailed articles about various features, is available to help you with using Tanzu Mission Control.

Use the following links to get more information about Tanzu Mission Control and other Tanzu products.

- Modern Apps and Cloud Management Tech Zone: <https://apps-cloudmgmt.techzone.vmware.com/>
- Tanzu Developer Center: <https://developer.vmware.com/products/tanzu>
- Tanzu Labs: <https://tanzu.vmware.com/labs>
- Tanzu Blog: <https://tanzu.vmware.com/blog>
- Tanzu Application Platform: <https://tanzu.vmware.com/application-platform>
- Tanzu Service Mesh: <https://tanzu.vmware.com/service-mesh>
- Tanzu Kubernetes Grid: <https://tanzu.vmware.com/kubernetes-grid>
- Events: <https://tanzu.vmware.com/events>