# VMware Tanzu Kubernetes Grid: Install, Configure, Manage

Lecture Manual

Tanzu Kubernetes Grid 1.3

VMware® Education Services
VMware, Inc.
www.vmware.com/education

VMware Tanzu Kubernetes Grid: Install, Configure, Manage

Lecture Manual

Tanzu Kubernetes Grid 1.3

Part Number EDU-EN-TKGICM13-LECT (17-SEP-2021)

The training material is provided "as is," and all express or implied conditions, representations, and warranties, including any implied warranty of merchantability, fitness for a particular purpose or noninfringement, are disclaimed, even if VMware, Inc., has been advised of the possibility of such claims. This training material is designed to support an instructor-led training course and is intended to be used for reference purposes in conjunction with the instructor-led training course.

The training material is not a standalone training tool. Use of the training material for self-study without class attendance is not recommended. These materials and the computer programs to which it relates are the property of, and embody trade secrets and confidential information proprietary to, VMware, Inc., and may not be reproduced, copied, disclosed, transferred, adapted or modified without the express written approval of VMware, Inc.

www.vmware.com/education

# Contents

# Module 1

# Course Introduction

## 1-2 Course Introduction

## 1-3 Importance

Understanding Tanzu Kubernetes Grid is essential for organizations that build modern applications platforms based on Kubernetes.

## 1-4 Learner Objectives

- Describe Tanzu Kubernetes Grid
- Differentiate between the Kubernetes life cycle management options in the VMware Tanzu portfolio
- Explain how to prepare a vSphere environment to install Tanzu Kubernetes Grid
- Describe how to initialize a Tanzu Kubernetes Grid instance
- Detail how to create Tanzu Kubernetes clusters
- Explain how to deploy the Tanzu Kubernetes Grid extensions on a Tanzu Kubernetes cluster
- Describe how to troubleshoot a Tanzu Kubernetes Grid instance

## 1-5 Course Outline

1. Course Introduction
2. Introducing VMware Tanzu Kubernetes Grid
3. Management Clusters
4. Tanzu Kubernetes Clusters
5. Configuring and Managing Tanzu Kubernetes Grid Instances
6. Troubleshooting Tanzu Kubernetes Grid

# 1-6 Typographical Conventions

The following typographical conventions are used in this course.

| Conventions | Use and Examples |
|---|---|
| `Monospace` | Identifies command names, command options, parameters, code fragments, error messages, filenames, folder names, directory names, and path names:<br>• Run the `esxtop` command.<br>• ... found in the `/var/log/messages` file. |
| **`Monospace Bold`** | Identifies user inputs:<br>• Enter **`ipconfig /release`**. |
| **Boldface** | Identifies user interface controls:<br>• Click the **Configuration** tab. |
| *Italic* | Identifies book titles:<br>• *vSphere Virtual Machine Administration* |
| < > | Indicates placeholder variables:<br>• <ESXi_host_name><br>• ... the `Settings/<Your_Name>.txt` file |

## 1-7 References

| Title | Location |
|---|---|
| Cluster API | https://cluster-api.sigs.k8s.io |
| Kubernetes Cluster API Provider vSphere | https://github.com/kubernetes-sigs/cluster-api-provider-vsphere |
| Fluent Bit | https://docs.fluentbit.io/manual/ |
| Pinniped | https://pinniped.dev/ |
| Contour | https://projectcontour.io/docs/ |

## 1-8 VMware Online Resources

Documentation for VMware Tanzu Kubernetes Grid: https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/index.html

VMware Communities: http://communities.vmware.com

- Start a discussion.
- Access the VMware knowledge base.
- Access documentation, technical papers, and compatibility guides.
- Access communities.
- Access user groups.

VMware Support: http://www.vmware.com/support

VMware Hands-on Labs: http://hol.vmware.com

VMware Education: http://www.vmware.com/education

- Access course catalog and worldwide course schedule.

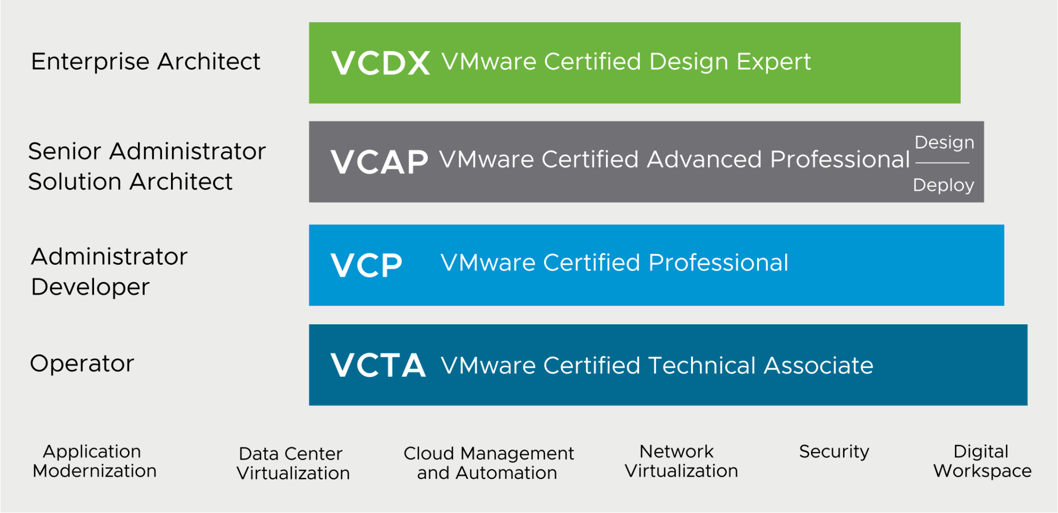
## 1-9 VMware Education Overview

Your instructor will introduce other Education Services offerings available to you:

- VMware Learning Paths:
  - Help you find the course that you need based on the product, your role, and your level of experience
  - Can be accessed at https://vmware.com/education
- VMware Learning Zone, which is the official source of digital training, includes the following options:
  - On-Demand Courses: Self-paced learning that combines lecture modules with hands-on practice labs
  - VMware Lab Connect: Self-paced, technical lab environment that lets you practice skills learned during instructor-led training
  - Certification Exam Prep: Comprehensive video-based reviews of exam topics and objectives to help you take your certification exam
- For more information, see https://vmwarelearningzone.vmware.com

# 1-10 VMware Certification Overview

VMware certifications validate your expertise and recognize your technical knowledge and skills with VMware technology.



Enterprise Architect — VCDX VMware Certified Design Expert

Senior Administrator
Solution Architect — VCAP VMware Certified Advanced Professional (Design / Deploy)

Administrator
Developer — VCP VMware Certified Professional

Operator — VCTA VMware Certified Technical Associate

Application Modernization · Data Center Virtualization · Cloud Management and Automation · Network Virtualization · Security · Digital Workspace

VMware certification sets the standards for IT professionals who work with VMware technology. Certifications are grouped into technology tracks. Each track offers one or more levels of certification (up to five levels).

For the complete list of certifications and details about how to attain these certifications, see https://vmware.com/certification.

# 1-11 VMware Badge Overview

VMware badges are digital emblems of skills and achievements.

vmware
PROFESSIONAL
DATA CENTER VIRTUALIZATION 2020
CERTIFICATION

vmware
IMPLEMENTATION EXPERT
DATA CENTER VIRTUALIZATION 2020
MILESTONE

vmware
SPECIALIST
vSAN 2020
SPECIALIST

vmware
IT ACADEMY
NETWORK VIRTUALIZATION CONCEPTS
KNOWLEDGE

vmware
HYPER-CONVERGED INFRASTRUCTURE
HCI TECHNICAL WHITEBOARD 2020
SKILL

vmware
SUBJECT MATTER EXPERT
CLOUD MANAGEMENT & AUTOMATION 2020
CONTRIBUTOR

Digital badges have the following features:

- Easy to share in social media (LinkedIn, Twitter, Facebook, blogs, and so on)
- Tethered to VMware to validate and verify achievement
- Contain metadata with skill tags and accomplishments
- Based on Mozilla's Open Badges standard

For the complete list of digital badges, see http://www.pearsonvue.com/vmware/badging.

# Module 2

# Introducing VMware Tanzu Kubernetes Grid

## 2-2 Importance

Kubernetes is a key technology in the VMware Tanzu portfolio and Tanzu Kubernetes Grid is the common implementation of Kubernetes across supported cloud platforms.

## 2-3 Module Lessons

1. Tanzu Kubernetes Grid in the VMware Tanzu Portfolio
2. Tanzu Kubernetes Grid Concepts
3. Tanzu CLI and Carvel Tools
4. Cluster API

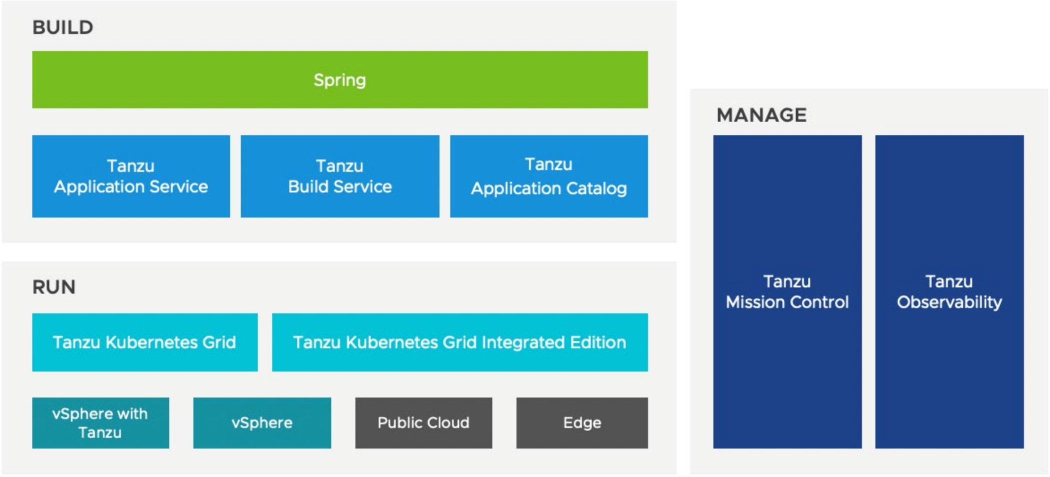
# 2-4 Lesson 1: Tanzu Kubernetes Grid in the VMware Tanzu Portfolio

## 2-5 Learner Objectives

- Describe the VMware Tanzu products responsible for Kubernetes life cycle management

## 2-6 VMware Tanzu Portfolio

The goal of the VMware Tanzu portfolio is to provide a modern applications platform.



Edge refers to environments located in remote locations outside of a typical data center environment. These environments have limited resources, for example, a vSphere cluster consisting of two to three ESXi hosts.

## 2-7 Kubernetes Life Cycle Management

The following products from the VMware Tanzu portfolio provide Kubernetes life cycle management (LCM):

- Run:
  - VMware Tanzu Kubernetes Grid
  - VMware vSphere with Tanzu
  - VMware Tanzu Kubernetes Grid Integrated Edition
- Manage:
  - VMware Tanzu Mission Control

# 2-8 Tanzu Kubernetes Grid

Tanzu Kubernetes Grid:

- Is a multicloud Kubernetes distribution that runs on:
  - vSphere, VMware Cloud on AWS, Azure VMware Solution
  - Amazon Web Services (AWS)
  - Microsoft Azure
- Automates the life cycle management of multiple Tanzu Kubernetes clusters using Cluster API
- Is an open source-aligned Kubernetes distribution
- Includes Kubernetes binaries that are tested, signed, and supported by VMware
- Includes signed and supported versions of open-source applications to provide the networking, authentication, ingress, logging, and monitoring services that a production Kubernetes environment requires

# 2-9 vSphere with Tanzu

vSphere with Tanzu:

- Provides a Kubernetes experience that is tightly integrated within vSphere 7
- Contains multiple services which provide access to infrastructure through a Kubernetes API
- Contains the Tanzu Kubernetes Grid Service:
  - Runs on supervisor clusters in vSphere with Tanzu
  - Creates Tanzu Kubernetes clusters that are optimized for vSphere

# 2-10 Tanzu Kubernetes Grid Integrated Edition

Tanzu Kubernetes Grid Integrated Edition:

- Is a multicloud Kubernetes distribution that you can run on:
  - vSphere
  - Google Cloud Platform
  - AWS
  - Microsoft Azure
- Was previously known as VMware Enterprise PKS
- Automates the life cycle management of multiple Kubernetes clusters using BOSH
- Includes Kubernetes binaries that are tested, signed, and supported by VMware
- Provides advanced networking with VMware NSX-T Data Center
- Provides integrations to vRealize Log Insight, vRealize Operations, and Tanzu Observability
- Supports Microsoft Windows workloads

# 2-11 VMware Tanzu Mission Control

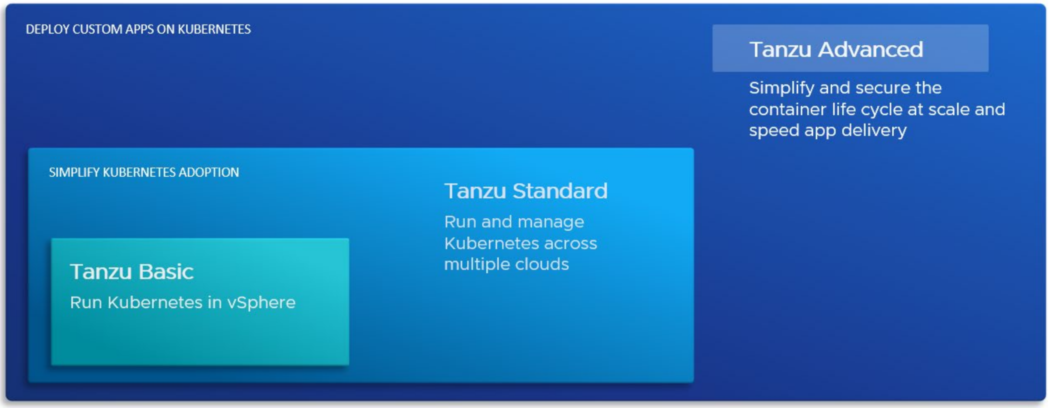
VMware Tanzu Mission Control:

- Provides a centralized management platform for consistently operating and securing multiple Kubernetes clusters and modern applications across multiple teams and clouds
- Is available through VMware Cloud services
- Provides a hosted Tanzu Kubernetes Grid implementation as a managed service

# 2-12 VMware Tanzu Editions

VMware Tanzu editions are groupings of the VMware Tanzu products that are designed for organizations at different stages of Kubernetes adoption.

Tanzu Kubernetes Grid is included in each VMware Tanzu edition.



DEPLOY CUSTOM APPS ON KUBERNETES

Tanzu Advanced

Simplify and secure the container life cycle at scale and speed app delivery

SIMPLIFY KUBERNETES ADOPTION

Tanzu Standard

Run and manage Kubernetes across multiple clouds

Tanzu Basic

Run Kubernetes in vSphere

VMware Tanzu editions:

- VMware Tanzu Basic:
  - Deploy Kubernetes using the Tanzu Kubernetes Grid Service in vSphere with Tanzu or Tanzu Kubernetes Grid.
- VMware Tanzu Standard:
  - Deploy Kubernetes across multiple public clouds using Tanzu Kubernetes Grid.
  - Manage Kubernetes using VMware Tanzu Mission Control.
- VMware Tanzu Advanced:
  - Developers use Spring Framework to reduce application development time.
  - Tanzu Build Service and Tanzu Application Catalog are used to build applications and maintain a secure and approved catalog of container images.

For more information about each VMware Tanzu edition, see *VMware Tanzu Overview* at https://tanzu.vmware.com/tanzu.

# 2-13 Review of Learner Objectives

- Describe the VMware Tanzu products responsible for Kubernetes life cycle management

# 2-14 **Lesson 2: Tanzu Kubernetes Grid Concepts**

## 2-15 Learner Objectives

- Describe the Tanzu Kubernetes Grid concepts
- Describe the components of a Tanzu Kubernetes Grid instance

# 2-16 Bootstrap Machines

A bootstrap machine is typically a VM on which you download and run the Tanzu CLI.

This machine is used to initialize a Tanzu Kubernetes Grid instance by bootstrapping a management cluster on the cloud infrastructure of choice.

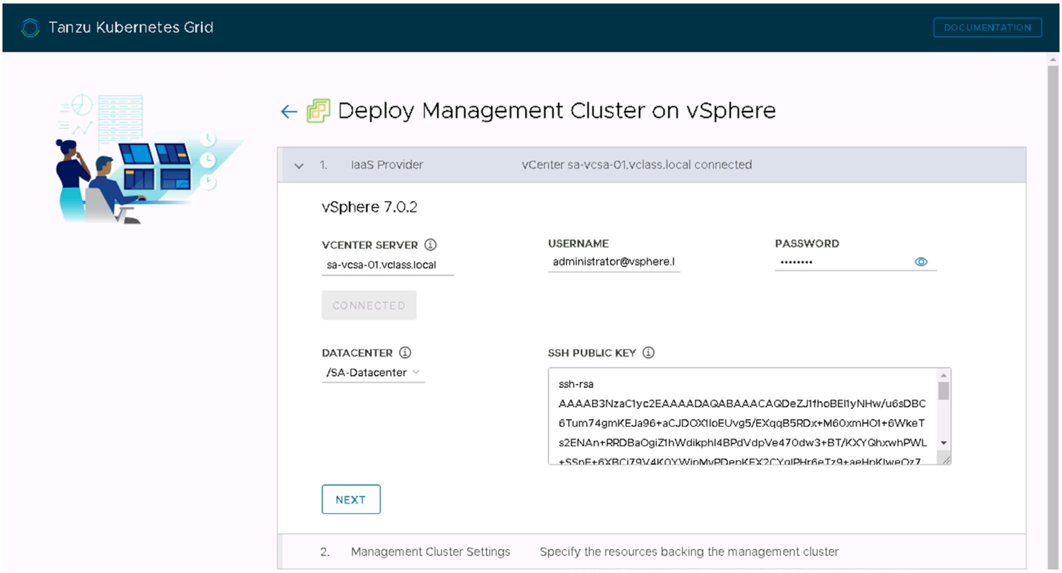
After bootstrapping the management cluster, the machine is used to manage the Tanzu Kubernetes Grid instance.

# 2-17 Tanzu CLI and Installer Interface

The Tanzu CLI is used to initialize a Tanzu Kubernetes Grid instance by creating management clusters.

After the management cluster is created, the Tanzu CLI communicates with it to create, scale, upgrade, and delete Tanzu Kubernetes clusters.

The installer interface is launched from the Tanzu CLI and is a graphical wizard which guides you through the configuration of a management cluster.



Deploy Management Cluster on vSphere

1. IaaS Provider — vCenter sa-vcsa-01.vclass.local connected

vSphere 7.0.2

VCENTER SERVER: sa-vcsa-01.vclass.local

USERNAME: administrator@vsphere.l

PASSWORD: ••••••••

DATACENTER: /SA-Datacenter

SSH PUBLIC KEY:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDeZJ1fhoBEl1yNHw/u6sDBC
6Tum74gmKEJa96+aCJDOX1IoEUvg5/EXqqB5RDx+M60xmHO1+6WkeT
s2ENAn+RRDBaOgiZ1hWdikphI4BPdVdpVe470dw3+BT/KXYQhxwhPWL
+SSnF+6XBCi79V4K0YWipMvPDenKFX2CYqIPHr6eTz9+aeHpKIweOz7
```

2. Management Cluster Settings — Specify the resources backing the management cluster

# 2-18 Tanzu Kubernetes Cluster Plans

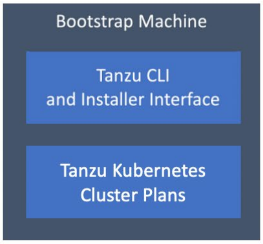
A cluster plan is the blueprint that describes the configuration with which to deploy a Tanzu Kubernetes cluster.

A cluster plan provides a set of configurable values that describe settings, for example, the amount of control plane machines, worker machines, virtual CPUs, memory, and other parameters.

The following cluster plans exist by default:

- `dev`
- `prod`

Existing cluster plans can be customized and new cluster plans can be created.


Bootstrap Machine
Tanzu CLI and Installer Interface
Tanzu Kubernetes Cluster Plans

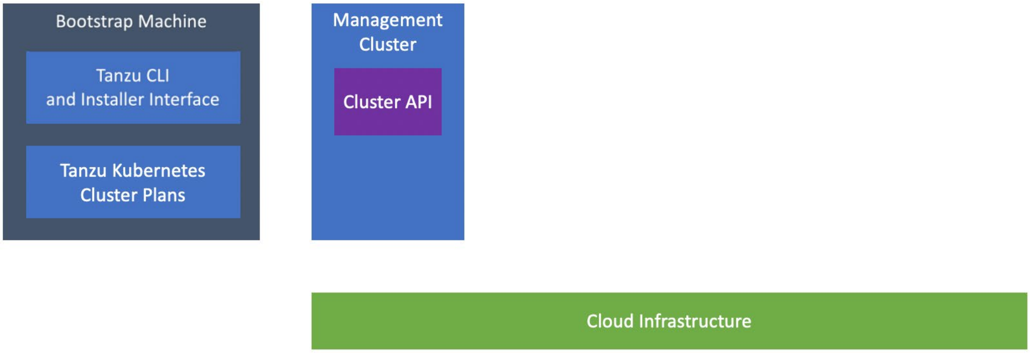
# 2-19 Management Clusters

The management cluster is the first element that you deploy when you create a Tanzu Kubernetes Grid instance.

This cluster is a Tanzu Kubernetes cluster dedicated to running Cluster API which provides life cycle management for Tanzu Kubernetes workload clusters.

The cluster is composed of one or more control plane nodes and one or more worker nodes.

In vSphere with Tanzu, the supervisor cluster performs the role of the management cluster.



# 2-20 Tanzu Kubernetes Clusters

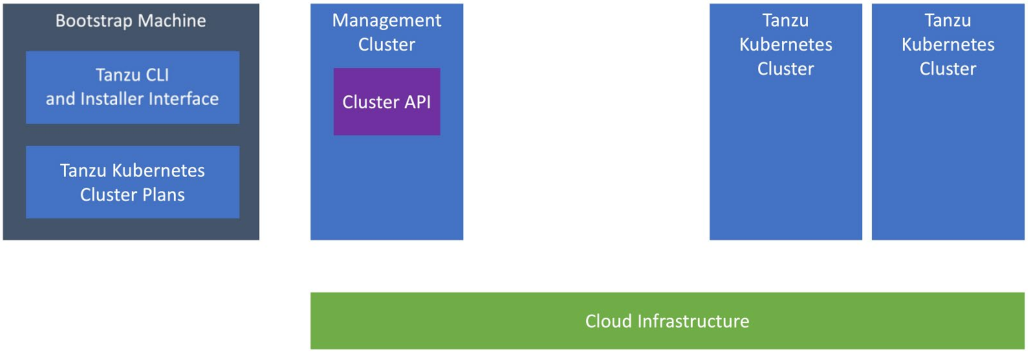
Tanzu Kubernetes clusters are Kubernetes clusters that you deploy from the management cluster by using the Tanzu CLI.

These clusters are composed of one or more control plane nodes and one or more worker nodes.

The terms Tanzu Kubernetes cluster and workload cluster are used interchangeably.

Tanzu Kubernetes clusters are created using a cluster plan.

You can manage the entire life cycle of Tanzu Kubernetes clusters by using the Tanzu CLI.



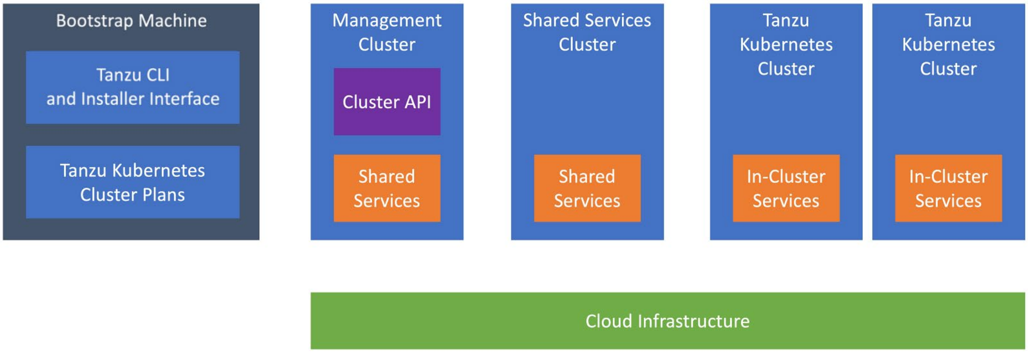
# 2-21 Shared and In-Cluster Services

Shared and in-cluster services are services that run in a Tanzu Kubernetes Grid instance, providing authentication, ingress, logging, and service discovery.

Shared services run on the management cluster or a dedicated shared-services cluster and are used by multiple Tanzu Kubernetes clusters.

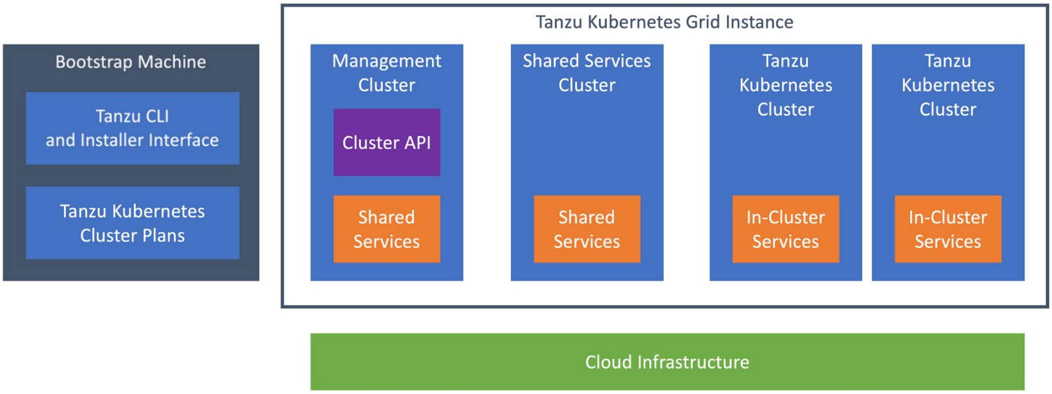
The Tanzu Kubernetes Grid extensions bundle is downloaded from the VMware website and provides the YAML configurations for supported services.

In-cluster services are deployed to specific Tanzu Kubernetes clusters.



# 2-22 Tanzu Kubernetes Grid Instances

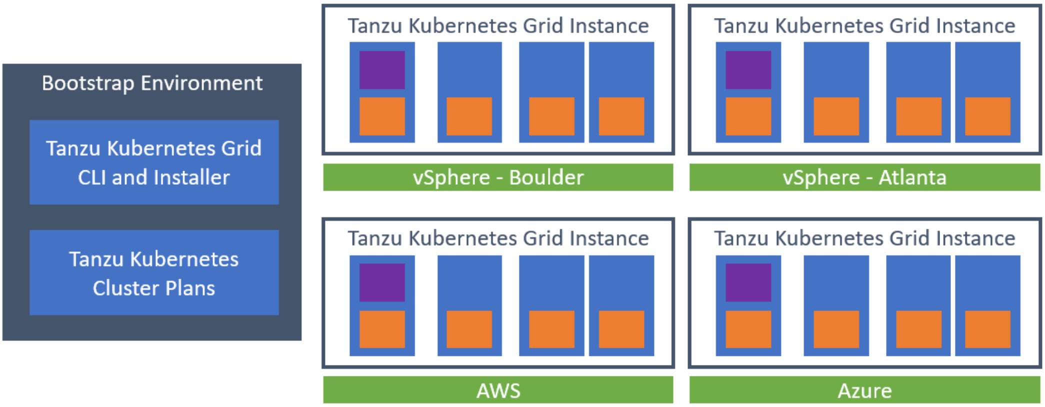
A Tanzu Kubernetes Grid instance is a full deployment of Tanzu Kubernetes Grid, including the management cluster, the deployed Tanzu Kubernetes clusters, and the shared and in-cluster services that you configure.



# 2-23 Bootstrapping Many Instances

A single bootstrap machine can bootstrap many instances of Tanzu Kubernetes Grid for:

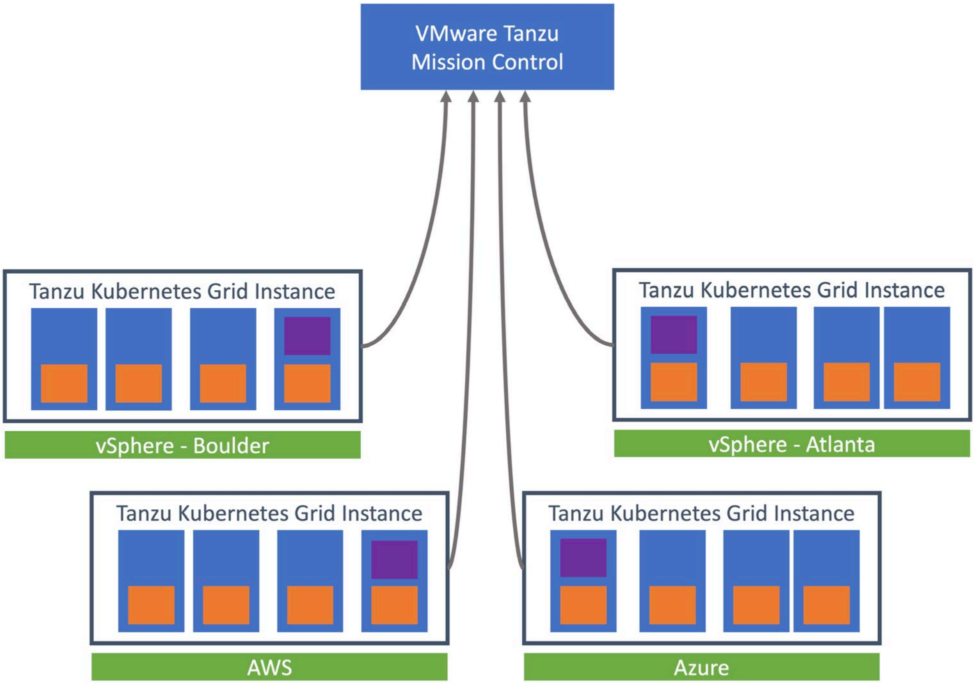
- Environments, such as production, staging, and test
- IaaS providers, such as vSphere and AWS
- Failure domains, for example, Datacenter-1, AWS us-east-2, or AWS us-west-2


Bootstrap Environment
Tanzu Kubernetes Grid CLI and Installer
Tanzu Kubernetes Cluster Plans
Tanzu Kubernetes Grid Instance
vSphere - Boulder
Tanzu Kubernetes Grid Instance
vSphere - Atlanta
Tanzu Kubernetes Grid Instance
AWS
Tanzu Kubernetes Grid Instance
Azure

# 2-24 Registering with VMware Tanzu Mission Control

Registering Tanzu Kubernetes Grid management clusters and workload clusters with VMware Tanzu Mission Control provides a central control plane for:

- Applying consistent Kubernetes policies across clusters
- Self-service provisioning and life cycle management of clusters
- Data protection of Kubernetes cluster workloads


VMware Tanzu Mission Control
Tanzu Kubernetes Grid Instance
vSphere - Boulder
Tanzu Kubernetes Grid Instance
vSphere - Atlanta
Tanzu Kubernetes Grid Instance
AWS
Tanzu Kubernetes Grid Instance
Azure

# 2-25 Review of Learner Objectives

- Describe the Tanzu Kubernetes Grid concepts
- Describe the components of a Tanzu Kubernetes Grid instance

# 2-26 **Lesson 3: Tanzu CLI and Carvel Tools**

# 2-27 Learner Objectives

- Describe the Tanzu CLI
- List the requirements for a bootstrap machine
- Describe the Carvel Tool Set

# 2-28 About the Tanzu CLI

The Tanzu CLI is:

- Available for Windows, MacOS, and Linux
- Used to interact with different products in the VMware Tanzu portfolio
- Used with Tanzu Kubernetes Grid to:
    - Initialize a Tanzu Kubernetes Grid instance
    - Create, scale, upgrade, and delete Tanzu Kubernetes clusters

# 2-29 Tanzu CLI Plug-Ins

The Tanzu CLI provides additional functionality through the use of plug-ins.

| Plug-Ins | Purpose |
|---|---|
| management-cluster | Initializes a Tanzu Kubernetes Grid instance and manages management clusters. |
| cluster | Manages Tanzu Kubernetes workload clusters. |
| kubernetes-release | Kubernetes release operations. |
| login | Sets the management cluster context. |
| pinniped-auth | Manages the authentication flow when logging in to a Tanzu Kubernetes Grid instance using OIDC or LDAP. |

# 2-30 Installing Tanzu CLI Plug-Ins

Ways to install Tanzu CLI plug-ins:

- From the Internet using `tanzu plugin install <PLUGIN-NAME>`
- From a local directory using `tanzu plugin install <PLUGIN-NAME> --local <DIR>`
- All available plug-ins using `tanzu plugin install all [--local <DIR>]`

# 2-31 Tanzu CLI Command Syntax

The Tanzu CLI uses the following syntax:

- `tanzu <RESOURCE> [<SUB-RESOURCE>] <VERB> --<OPTIONS>`

Tanzu CLI command examples:

- `tanzu management-cluster get`
- `tanzu cluster create --file tkc-01.yaml`
- `tanzu cluster list`
- `tanzu cluster get tkc-01`
- `tanzu cluster kubeconfig get --admin`
- `tanzu kubernetes-release get`

# 2-32 CLI Configuration Files

The Tanzu CLI creates the following configuration files.

| File | Description |
|---|---|
| ~/.kube-tkg/config | Stores the Kubernetes contexts for the management clusters that the Tanzu CLI manages. This file is comparable to the ~/.kube/config file used by kubectl. |
| ~/.tanzu/config.yaml | Stores a list of management clusters that the Tanzu CLI is managing. |
| ~/.tanzu/tkg/providers | Stores the YAML files that are templated and merged when a cluster configuration is generated. |
| ~/.tanzu/tkg/clusterconfigs | Management cluster configuration files created by the installer UI are saved here. |
| ~/.tanzu/tkg/bom | Stores the Bill of Materials (BOM) files which list the container image versions provided by the installed Tanzu Kubernetes Grid version. |
| ~/.tanzu/pinniped/sessions.yaml | Used by the pinniped-auth plug-in to store authenticated session details when accessing clusters using OIDC or LDAP. |

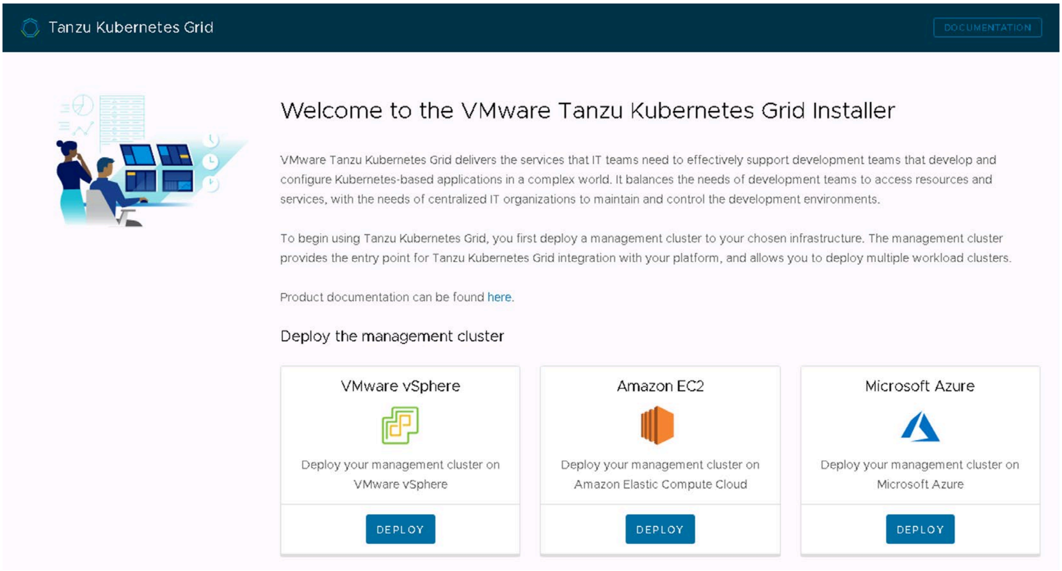
# 2-33 Bootstrap Machine Requirements

The bootstrap machine on which you initialize a Tanzu Kubernetes Grid instance using tanzu `management-cluster create` must meet the following requirements:

- kubectl is installed.
- Docker is installed, if you are installing Tanzu Kubernetes Grid on Linux.
- Docker Desktop is installed, if you are installing Tanzu Kubernetes Grid on MacOS or Windows.
- A minimum of 6 GB of memory is available for the containers used during the bootstrapping process.
- IP network connectivity exists to the vCenter Server instance and to the network to which the management cluster is deployed.

# 2-34 About the Tanzu Kubernetes Grid Installer Interface

The Tanzu Kubernetes Grid installer interface is a graphical wizard that guides you through the configuration of a management cluster.



Welcome to the VMware Tanzu Kubernetes Grid Installer

VMware Tanzu Kubernetes Grid delivers the services that IT teams need to effectively support development teams that develop and configure Kubernetes-based applications in a complex world. It balances the needs of development teams to access resources and services, with the needs of centralized IT organizations to maintain and control the development environments.

To begin using Tanzu Kubernetes Grid, you first deploy a management cluster to your chosen infrastructure. The management cluster provides the entry point for Tanzu Kubernetes Grid integration with your platform, and allows you to deploy multiple workload clusters.

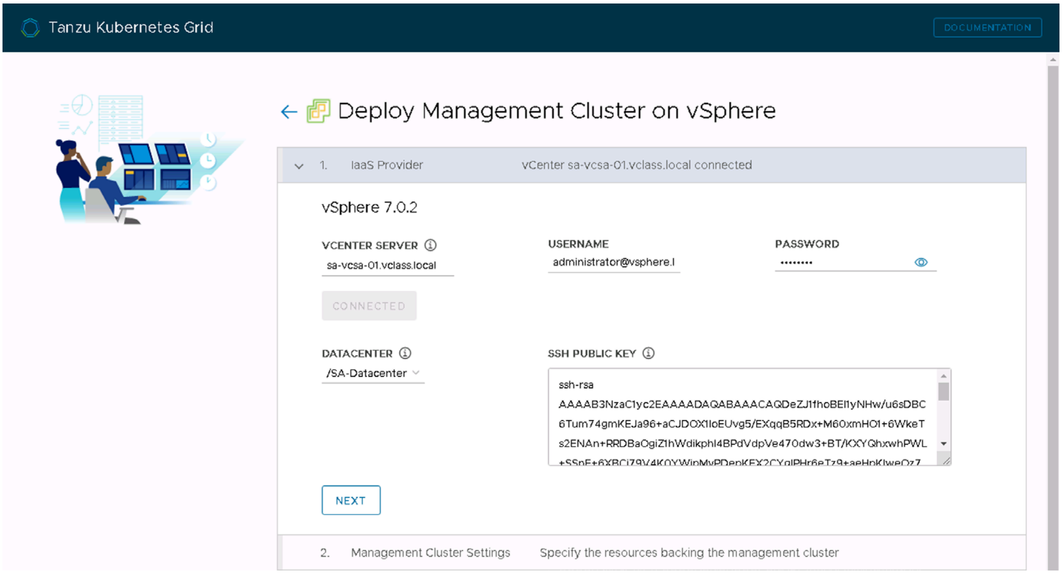
Product documentation can be found here.

Deploy the management cluster

VMware vSphere
Deploy your management cluster on VMware vSphere

Amazon EC2
Deploy your management cluster on Amazon Elastic Compute Cloud

Microsoft Azure
Deploy your management cluster on Microsoft Azure



Deploy Management Cluster on vSphere

1. IaaS Provider — vCenter sa-vcsa-01.vclass.local connected

vSphere 7.0.2

VCENTER SERVER: sa-vcsa-01.vclass.local

USERNAME: administrator@vsphere.l

PASSWORD: ••••••••

DATACENTER: /SA-Datacenter

SSH PUBLIC KEY:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDeZJ1fhoBEl1yNHw/u6sDBC
6Tum74gmKEJa96+aCJDOX1IoEUvg5/EXqqB5RDx+M60xmHO1+6WkeT
s2ENAn+RRDBaOgiZ1hWdikphl4BPdVdpVe470dw3+BT/KXYQhxwhPWL
+SSnE+6XBCi79V4K0YWipMyPDepKEX2CYqlPHr6eTz9+aeHpKlweOz7
```

2. Management Cluster Settings — Specify the resources backing the management cluster

# 2-35 Carvel Tools

Carvel tools are a collection of command-line tools that help in the building, configuration, and deployment of Kubernetes and Kubernetes-based workloads.

| Tool Name | Description |
|---|---|
| ytt | Templates and overlays Kubernetes configuration using YAML structures. |
| kapp | Installs, upgrades, and deletes multiple Kubernetes resources as a single application. |
| kapp-controller | Performs the same functionality as kapp but as a Kubernetes controller within a Kubernetes cluster. |
| kbld | Automates the building and pushing of container images. |
| imgpkg | Packages and distributes Kubernetes configuration and container images into a single container image, referred to as a bundle. |
| vendir | Syncs a directory structure from various data sources to ensure that a platform operations or development team are working on the same files. |

Tanzu Kubernetes Grid uses each of the Carvel tools internally to perform different tasks. The tools do not require installation but can be used for managing and troubleshooting operations.

Tanzu Kubernetes Grid extensions are deployed using kapp-controller and are inspected using the kapp CLI.

When preparing for an Internet-restricted installation, you can use imgpkg to query the image registry where the Tanzu Kubernetes Grid container images are hosted and to generate a list of Docker pull, tag, and push commands.

For more information about Carvel tools, see the Carvel website at https://carvel.dev/.

# 2-36 Lab 1: Setting Up a Bootstrap Machine

Verify that Docker is running and install the command-line tools:

1. Verify that Docker Is Running
2. Install the Kubernetes CLI
3. Enable Kubernetes CLI Autocompletion
4. Install the Tanzu CLI
5. Install the Tanzu CLI Plug-Ins
6. Enable Tanzu CLI Autocompletion
7. Install the Carvel Tools

## 2-37 Review of Learner Objectives

- Describe the Tanzu CLI
- List the requirements for a bootstrap machine
- Describe the Carvel Tool Set

# 2-38 **Lesson 4: Cluster API**

## 2-39 Learner Objectives

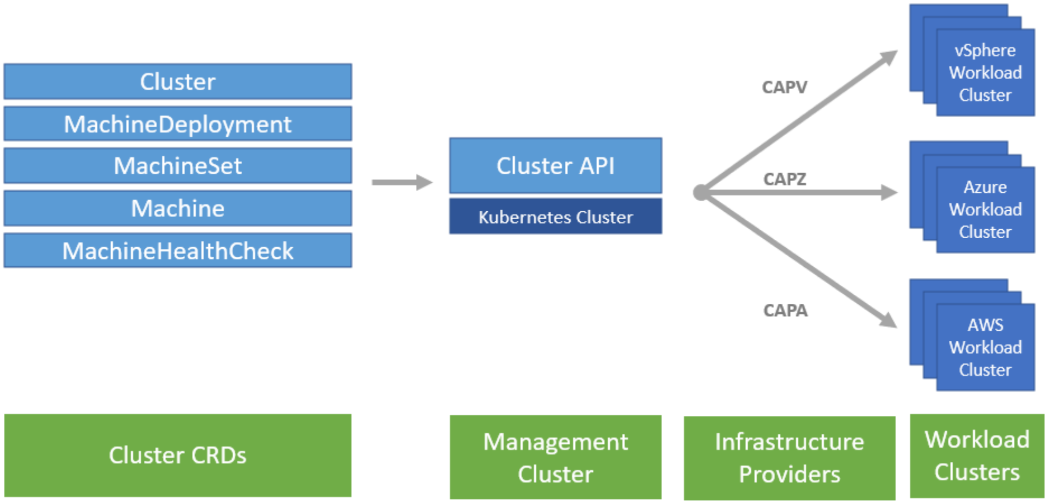
- Describe Cluster API
- List the infrastructure providers
- Detail the Cluster API controllers
- List the Cluster API Custom Resource Definitions

# 2-40 Introducing Cluster API

Cluster API is a Kubernetes project to provide declarative Kubernetes-style APIs to cluster provisioning, configuration, and management.

Cluster API controllers running on a Kubernetes cluster receive Cluster API definitions that specify the desired state of the new cluster.

They also request from the cloud provider to create the new cluster.



# 2-41 Cluster API CRDs

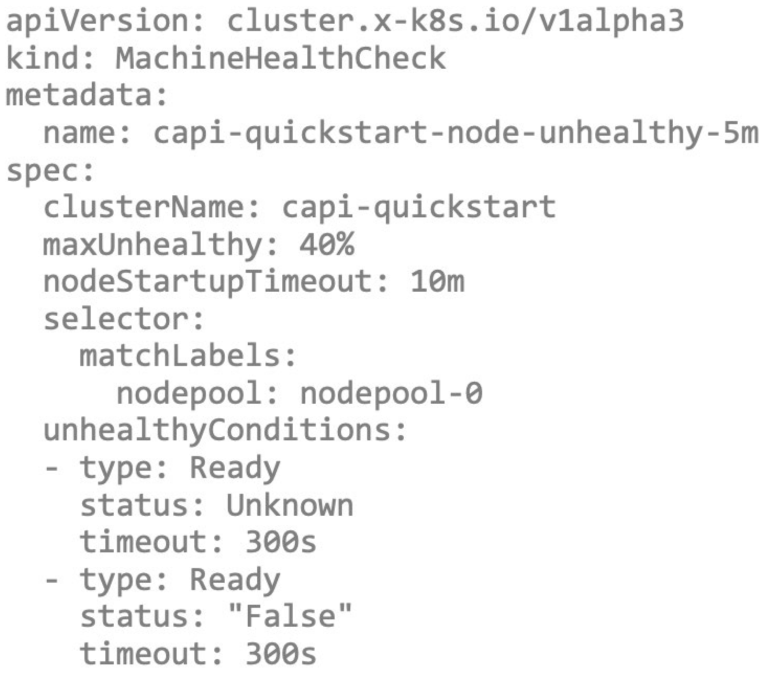
Cluster API provides the following Custom Resource Definitions (CRDs) that are used to describe a Kubernetes cluster.

| CRD | Description |
|---|---|
| Cluster | Describes a cluster. |
| KubeadmControlPlane | Describes a control plane (etcd, kube-apiserver). |
| MachineDeployment | Manages rolling out changes to a machine spec using MachineSets. Similar to a Kubernetes deployment resource. |
| MachineSet | Ensures that the required machine objects with the correct machine spec exist. |
| Machine | Describes a VM using a machine spec. |
| MachineHealthCheck | Defines the conditions when a machine should be considered unhealthy. Only supported for worker nodes. |

# 2-42 Machine Health Checks

Machine health checks allow Cluster API to monitor the health of cluster nodes:

- A timeout is specified for each of the conditions that can be checked.
- If any of these conditions are met for the duration of the timeout, the Machine is remediated.
- The action of remediating a Machine triggers a new Machine to be created to replace the failed one.

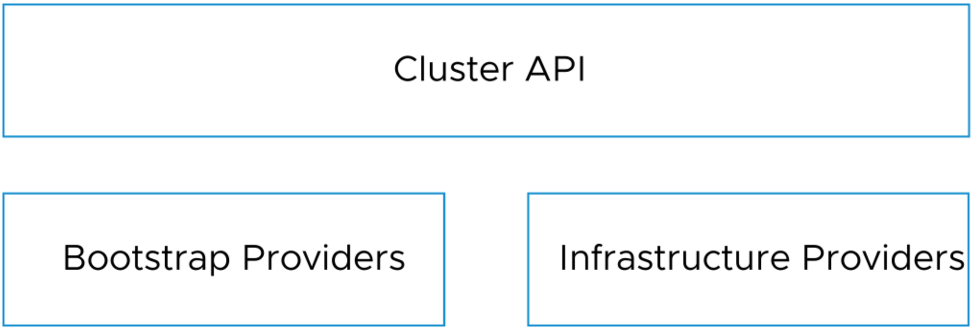


```
apiVersion: cluster.x-k8s.io/v1alpha3
kind: MachineHealthCheck
metadata:
  name: capi-quickstart-node-unhealthy-5m
spec:
  clusterName: capi-quickstart
  maxUnhealthy: 40%
  nodeStartupTimeout: 10m
  selector:
    matchLabels:
      nodepool: nodepool-0
  unhealthyConditions:
  - type: Ready
    status: Unknown
    timeout: 300s
  - type: Ready
    status: "False"
    timeout: 300s
```

# 2-43 Cluster API Components

The logic in Cluster API is divided per component:

- Bootstrap providers perform the steps to configure and start the Kubernetes processes that make up a cluster.
- Infrastructure providers perform the steps to create cloud resources, such as VMs.



Cluster API

Bootstrap Providers

Infrastructure Providers

## 2-44 Bootstrap Providers

Cluster API bootstrap provider is a component of Cluster API that is responsible for turning a Machine into a Kubernetes node.

Cluster API bootstrap provider kubeadm is a bootstrap provider implementation that uses kubeadm to perform Machine configuration.

## 2-45 Infrastructure Providers

Cluster API infrastructure providers exist for various cloud providers.

| Cluster API Infrastructure Provider | Cloud Provider | Supported by Tanzu Kubernetes Grid 1.3 |
|---|---|---|
| CAPV | vSphere | Yes |
| CAPA | Amazon AWS | Yes |
| CAPZ | Microsoft Azure | Yes |
| CAPG | Google Cloud | No |
| CAPD | Docker | No |
| CAPO | OpenStack | No |

## 2-46 Cluster API Controllers

The following controllers watch the Kubernetes API for Cluster API (CAPI) resources and perform the required steps to provision clusters.

| Controller | Description |
|---|---|
| capi-controller-manager | Provides the overall CAPI functionality. |
| capi-kubeadm-control-plane-controller-manager | Provides the functionality to configure the control plane nodes to run `etcd`, `kube-apiserver`, and other control plane components. |
| capi-kubeadm-bootstrap-controller-manager | Provides the functionality to configure the worker nodes and join the worker nodes to the cluster. |

# 2-47 About the Cluster API Provider for vSphere

The Cluster API provider for vSphere (CAPV) provides the following controller and CRDs.

| Controller or CRD | Description |
|---|---|
| capv-controller-manager | The controller watches for CAPI and CAPV resources and acts upon them by communicating with vSphere to provision and monitor resources. |
| VSphereCluster<br>VSphereMachine<br>VSphereMachineTemplate<br>VSphereVM | The CRDs describe how CAPV creates the vSphere specific resources for CAPI.<br>Most generic CAPI resources have a corresponding CAPV resource, for example, each CAPI Cluster resource has a corresponding VSphereCluster resource. |

# 2-48 Review of Learner Objectives

- Describe Cluster API
- List the infrastructure providers
- Detail the Cluster API controllers
- List the Cluster API Custom Resource Definitions

# 2-49 Key Points

- The VMware Tanzu portfolio is the VMware solution for the development and deployment of modern applications.
- Tanzu Kubernetes Grid is a common distribution of Kubernetes across multiple VMware products.
- The Tanzu CLI is used to deploy and manage Tanzu Kubernetes Grid instances.

# Module 3

# Management Clusters

## 3-2 Importance

Tanzu Kubernetes Grid management clusters provide the Kubernetes life cycle management functionality of Tanzu Kubernetes Grid on supported platforms.

## 3-3 Module Lessons

1. Preparing the vSphere Environment
2. NSX Advanced Load Balancer
3. Cluster Authentication
4. Creating Management Clusters
5. Managing Management Clusters

# 3-4 **Lesson 1: Preparing the vSphere Environment**

# 3-5 Learner Objectives

- Describe the vSphere requirements for deploying a management cluster
- List the differences between deploying on vSphere 6.7 Update 3 and vSphere 7

## 3-6 vSphere Requirements

Tanzu Kubernetes Grid has the following vSphere requirements:

- vSphere 6.7 Update 3, vSphere 7, VMware Cloud on AWS, or Azure VMware Solution
- vSphere cluster with a minimum of 2 ESXi hosts and vSphere DRS enabled
- A DHCP server to provide IP addresses to Tanzu Kubernetes Grid management and workload clusters
- Traffic to vCenter Server allowed from the network on which clusters run
- A dedicated vCenter Single Sign-On account with the required permissions for Tanzu Kubernetes Grid

# 3-7 vSphere Port Requirements

The following ports are required for components to communicate in a Tanzu Kubernetes Grid instance.

| Source Networks | Destination IP | Protocol:Port | Description |
|---|---|---|---|
| Workload cluster | Management cluster VIP | TCP:6443 | Allows workload clusters to register with the management cluster. |
| Management cluster | Workload cluster VIP | TCP:6443,5556 | Allows the management cluster to configure the workload cluster. |
| Management and workload cluster | Container registry | TCP:443 | Allows components to retrieve container images. |
| Management and workload cluster | vCenter Server | TCP:443 | Allows components to create VMs and storage volumes. |
| Management and workload cluster | DNS servers | UDP:53 | Allows communication to DNS servers. |
| Management and workload cluster | NTP servers | UDP:123 | Allows communication to NTP servers. |

# 3-8 Dedicated vSphere SSO User

When deploying Tanzu Kubernetes Grid using a dedicated vSphere SSO user, you assign the Tanzu Kubernetes Grid user a role that allows access to the following objects:

- Data centers or data center folders
- Datastores or datastore folders
- Hosts, clusters, or resource pools
- Networks to which clusters are assigned
- VM and template folders

# 3-9 Tanzu Kubernetes Grid User Permissions

The user used to deploy Tanzu Kubernetes Grid must have a role with the following permissions.

| vSphere Object | Required Permission |
|---|---|
| Datastore | Allocate space, datastore consumer |
| Global | Cloud Admin (vSphere with Tanzu only) |
| Network | Assign network |
| Resource | Assign VM to resource pool |
| Sessions | Message, validate session |
| vApp | Import |
| VM | Configuration > Add new disk, add existing disk<br>Configuration > Advanced configuration<br>Configuration > Change CPU count, memory, settings<br>Configuration > Configure Raw device<br>Interaction > Power on, off<br>Inventory > Create from existing, remove<br>Provisioning > Deploy template |

# 3-10 SSH Key Pairs

An optional SSH public key is added to all VMs that are created by Tanzu Kubernetes Grid.

The SSH key is used to remotely connect to a VM using SSH to troubleshoot or perform other administrative tasks.

If an SSH key is not provided, connecting to a control plane or worker node using SSH is not possible.

# 3-11 OVA Templates

Tanzu Kubernetes Grid provides base OS image templates in OVA format for you to import to vSphere:

- Ubuntu v20.04 Kubernetes v1.xx.yy OVA
- Photon v3 Kubernetes v1.xx.yy OVA

Ubuntu is the default OS.

Tanzu Kubernetes Grid creates the management cluster and Tanzu Kubernetes cluster node VMs from the templates.

After importing the OVA files, you must convert them to VM templates.

The base OS image template includes the version of Kubernetes that Tanzu Kubernetes Grid uses to create clusters.

# 3-12 Deploying in Internet-Restricted Environments

Deploying in an Internet-restricted environment is supported by loading all container images into a private Docker registry:

1. Within your firewall, install and configure a private Docker registry.

   ```
   harbor.vclass.local
   ```

2. Run the `docker pull` command for all required images from `projects.registry.vmware.com/tkg`.
3. Run the `docker tag` and `docker push` commands to tag and push the images to the private Docker registry `harbor.vclass.local/tkg`.
4. Set the environment variable.

   ```
   TKG_CUSTOM_IMAGE_REPOSITORY="harbor.vclass.local/tkg"
   ```

5. Proceed with the normal installation steps.

Steps 2 and 3 can be automated using the `gen-publish-images.sh` script, available from the Tanzu Kubernetes Grid documentation at https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-mgmt-clusters-airgapped-environments.html.

## 3-13 Lab 2: Preparing for an Internet-Restricted Installation

Prepare for an Internet-restricted installation by loading Tanzu Kubernetes Grid container images in an internal Harbor registry:

1. Create a Library in Harbor
2. Push Tanzu Kubernetes Grid Container Images to Harbor

## 3-14 Lab 3: Configuring vSphere

Create the vSphere resources required by Tanzu Kubernetes Grid:

1. Log in to the vSphere Client
2. Create Resource Pools
3. Create a VM Folder
4. Import the Base OS Template OVA Files
5. Convert the VMs to Templates

## 3-15 Review of Learner Objectives

- Describe the vSphere requirements for deploying a management cluster
- List the differences between deploying on vSphere 6.7 Update 3 and vSphere 7

# 3-16 **Lesson 2: NSX Advanced Load Balancer**

## 3-17 Learner Objectives

- Describe the NSX Advanced Load Balancer components
- Explain how Tanzu Kubernetes Grid integrates with NSX Advanced Load Balancer

# 3-18 Load Balancing Kubernetes Services

Tanzu Kubernetes Grid supports optional integration with NSX Advanced Load Balancer.

NSX Advanced Load Balancer does not require any licenses to be used for Tanzu Kubernetes Grid.

The integration allows NSX Advanced Load Balancer to provide layer 4 load balancing services when a load balancer service type is created on a Tanzu Kubernetes workload cluster.

NSX Advanced Load Balancer does not perform load balancing for the Kubernetes API on control plane nodes. That functionality is provided by kube-vip.

# 3-19 NSX Advanced Load Balancer Components

NSX Advanced Load Balancer has the following components.

| Component | Description |
|---|---|
| Controller | The controller is the single point of management and control. For high availability, it is typically deployed as a three-node cluster. As its name implies, the controller implements the control plane. |
| Service Engine | Service engines (SEs) manage all data plane operations by receiving and executing instructions from the controller. SEs perform load balancing by using virtual services. They collect real-time application telemetry from application traffic flows. |
| Admin Console | The admin console is a modern web-based UI that provides role-based access to control, manage, and monitor applications. The admin console runs as a web service on the controller. |

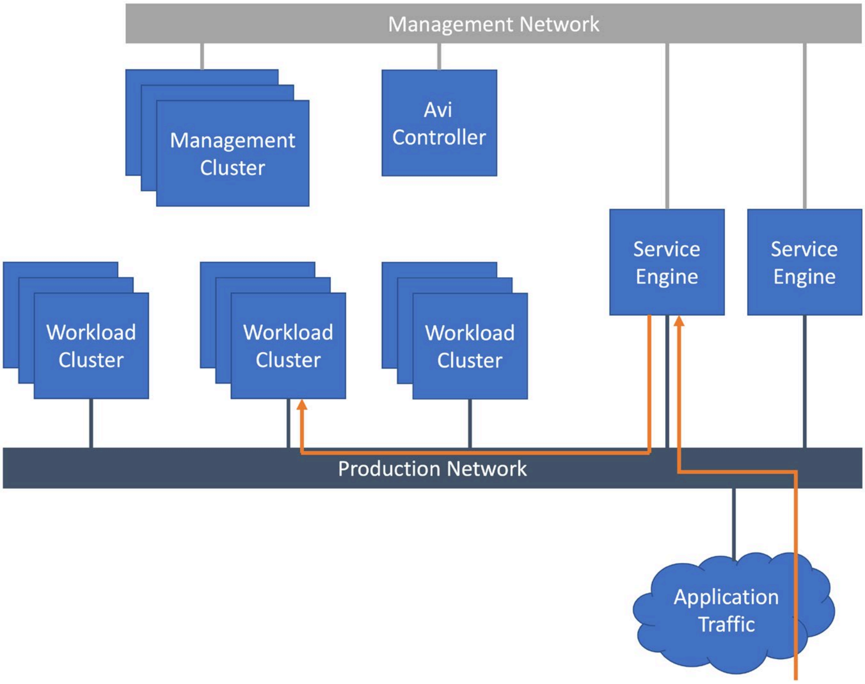
A best practice in a demo or evaluation environment is to use only one controller node. However, a production environment requires three controller nodes in a cluster.

Service engines might be configured with as little as 1 vCPU core and 1 GB RAM, or up to 36 vCPU cores and 128 GB RAM. These numbers are abstract because network and application traffic can vary significantly.

# 3-20 NSX Advanced Load Balancer Architecture

The example shows an NSX Advanced Load Balancer network topology that is integrated with Tanzu Kubernetes Grid.

When applications running on a workload cluster are exposed using a load balancer service type, traffic flowing inbound to the applications is routed to the SEs and, from there, routed to a NodePort IP address on the workload cluster.

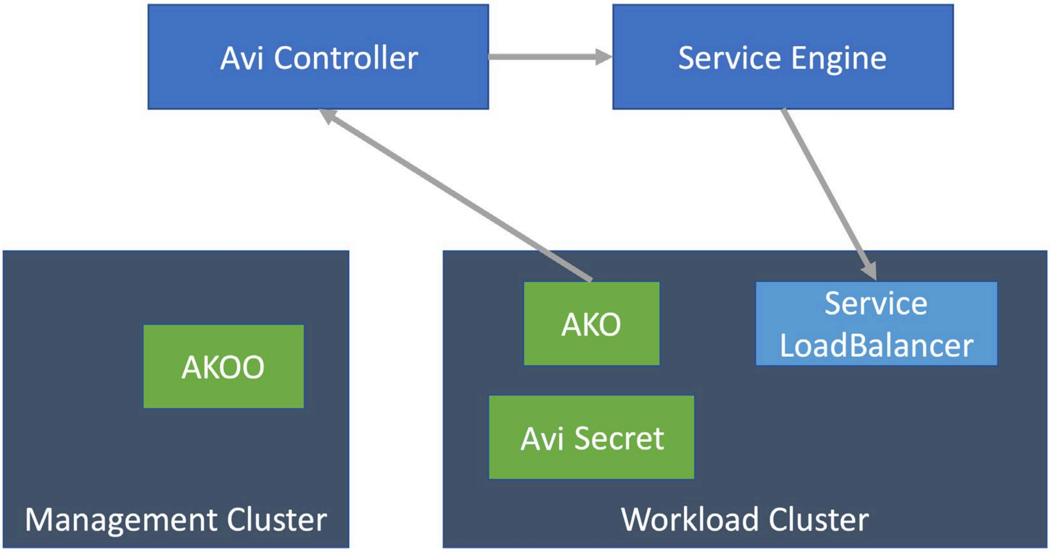


# 3-21 AKO Operator Functionality

The AKO Operator (AKOO) runs on a management cluster and is responsible for deploying the AKO on workload clusters.

When a new workload cluster is deployed, the AKOO performs the following actions:

- Creates a user dedicated to the workload cluster in the NSX Advanced Load Balancer controller.
- Creates a Kubernetes secret in the workload cluster, containing the new user credentials.
- Deploys the AKO controller to the workload cluster.

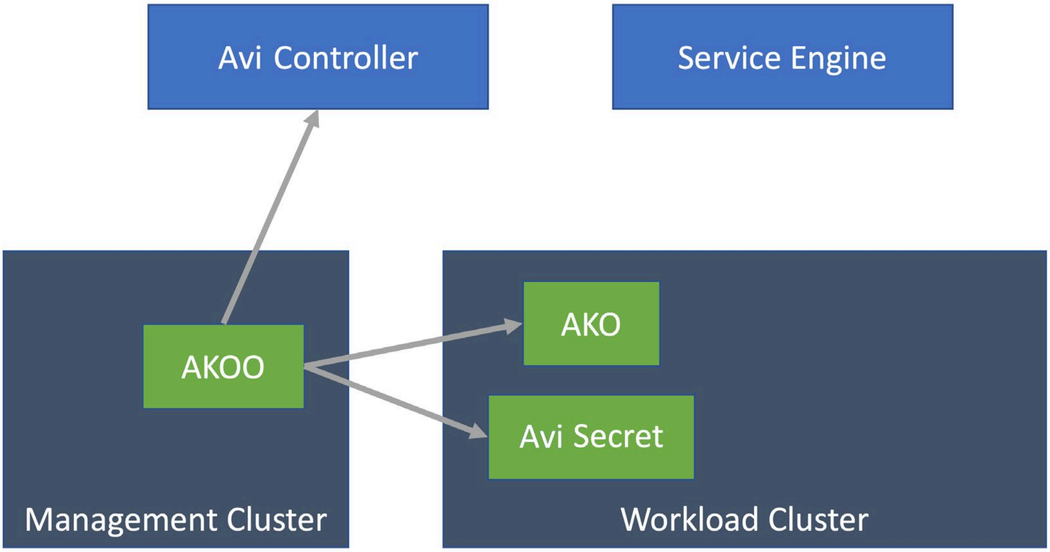


# 3-22 Avi Kubernetes Operator Functionality

The Avi Kubernetes Operator (AKO) runs on a workload cluster.

When a service of type load balancer is created on the workload cluster:

- The AKO operator sends a request to the NSX Advanced Load Balancer controller to create a virtual service.
- The NSX Advanced Load Balancer controller sends a request to the SE to configure a virtual service.
- The SE load balances traffic to pods running in the cluster.



# 3-23 Review of Learner Objectives

- Describe the NSX Advanced Load Balancer components
- Explain how Tanzu Kubernetes Grid integrates with NSX Advanced Load Balancer

## 3-24 **Lesson 3: Cluster Authentication**

## 3-25 Learner Objectives

- Describe how Kubernetes manages authentication
- Explain what Pinniped is
- Explain what Dex is
- Describe the Pinniped authentication workflow

# 3-26 External Authentication

Users are not directly managed or authenticated by Kubernetes.

An external identity service must be used to generate trusted tokens or certificates that a user can use to communicate with the Kubernetes API server.

Kubernetes supports OIDC tokens as a way to identify users who access the cluster.

## 3-27 User Authentication Using Pinniped

User authentication in Tanzu Kubernetes Grid is provided by Pinniped.

Pinniped allows you to integrate external OIDC or LDAP identity providers (IDP) into Tanzu Kubernetes clusters, so that you can control user access to those clusters.

## 3-28 About Dex

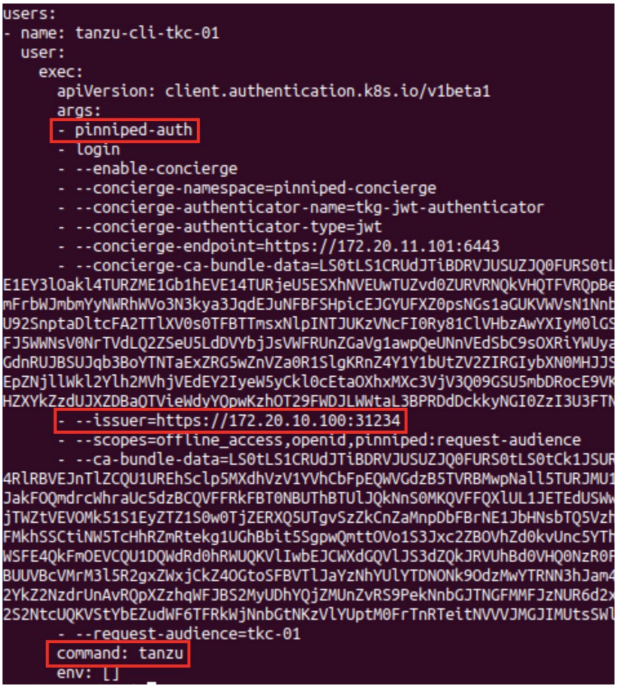
Dex is an identity service that supports LDAP authentication.

Pinniped uses Dex if LDAP authentication is required.

# 3-29 Kubeconfig Files Configured with Pinniped Authentication

When Pinniped authentication is enabled, the `tanzu cluster kubeconfig get` command returns a kubeconfig file that is configured to run the Tanzu pinniped-auth plug-in.

The pinniped-auth plug-in initiates the Pinniped authentication workflow which redirects the user to the Pinniped supervisor endpoint on the management cluster.

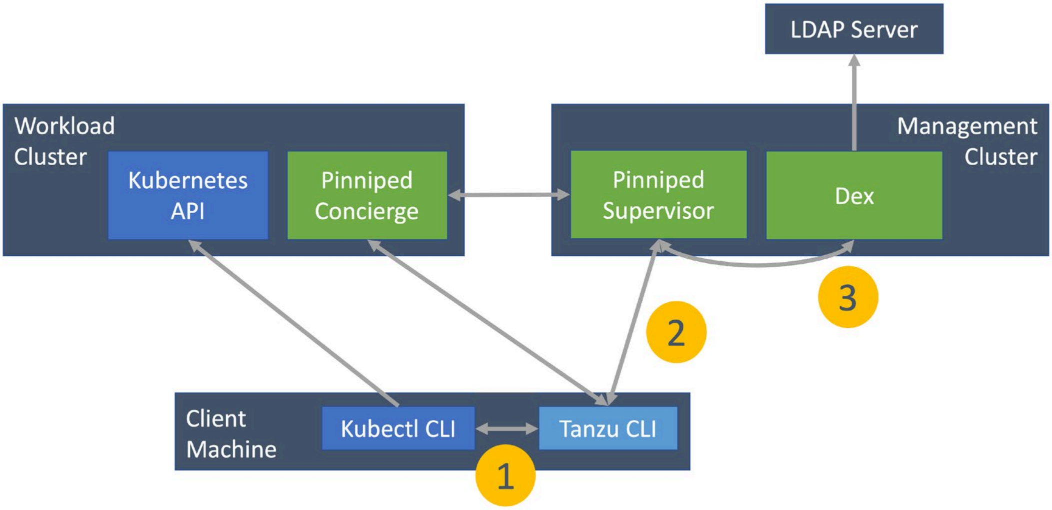


```
users:
- name: tanzu-cli-tkc-01
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - pinniped-auth
      - login
      - --enable-concierge
      - --concierge-namespace=pinniped-concierge
      - --concierge-authenticator-name=tkg-jwt-authenticator
      - --concierge-authenticator-type=jwt
      - --concierge-endpoint=https://172.20.11.101:6443
      - --concierge-ca-bundle-data=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tL
E1EY3lOakl4TURZME1Gb1hEVE14TURjeU5ESXhNVEUwTUZvd0ZURVRNQkVHQTFVRQpBe
mFrbWJmbmYyNWRhWVo3N3kya3JqdEJuNFBFSHpicEJGYUFXZ0psNGs1aGUKVWVsN1Nnb
U92SnptaDltcFA2TTlXV0s0TFBTTmsxNlpINTJUKzVNcFI0Ry81ClVHbzAwYXIyM0lGS
FJ5WWNsV0NrTVdLQ2ZSeU5LdDVYbjJsVWFRUnZGaVg1awpQeUNnVEdSbC9sOXRiYWUya
GdnRUJBSUJqb3BoYTNTaExZRG5wZnVZa0R1SlgKRnZ4Y1Y1bUtZV2ZIRGIybXN0MHJJS
EpZNjllWkl2Ylh2MVhjVEdEY2IyeW5yCkl0cEtaOXhxMXc3VjV3Q09GSU5mbDRocE9VK
HZXYkZzdUJXZDBaQTVieWdyYQpwKzhOT29FWDJLWWtaL3BPRDdDckkyNGI0ZzI3U3FTN
      - --issuer=https://172.20.10.100:31234
      - --scopes=offline_access,openid,pinniped:request-audience
      - --ca-bundle-data=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR
4RlRBVEJnTlZCQU1UREhSclp5MXdhVzV1YVhCbFpEQWVGdzB5TVRBMwpNall5TURJMU1
JakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0MKQVFFQXlUL1JETEdUSWw
jTWZtVEVOMk51S1EyZTZ1S0w0TjZERXQ5UTgvSzZkCnZaMnpDbFBrNE1JbHNsbTQ5Vzh
FMkhSSCtiNW5TcHhRZmRtekg1UGhBbit5SgpwQmttOVo1S3Jxc2ZBOVhZd0kvUnc5YTh
WSFE4QkFmOEVCQU1DQWdRd0hRWUQKVlIwbEJCWXdGQVlJS3dZQkJRVUhBd0VHQ0NzR0F
BUUVBcVMrM3l5R2gxZWxjCkZ4OGtoSFBVTlJaYzNhYUlYTDNONk9OdzMwYTRNN3hJam4
2YkZ2NzdrUnAvRQpXZzhqWFJBS2MyUDhYQjZMUnZvRS9PekNnbGJTNGFMMFJzNUR6d2x
2S2NtcUQKVStYbEZudWF6TFRkWjNnbGtNKzVlYUptM0FrTnRTeitNVVVJMGJIMUtsSWl
      - --request-audience=tkc-01
      command: tanzu
      env: []
```

# 3-30 Pinniped Authentication Workflow (1)

The following workflow is initiated when a kubectl command is run and Pinniped authentication is enabled:

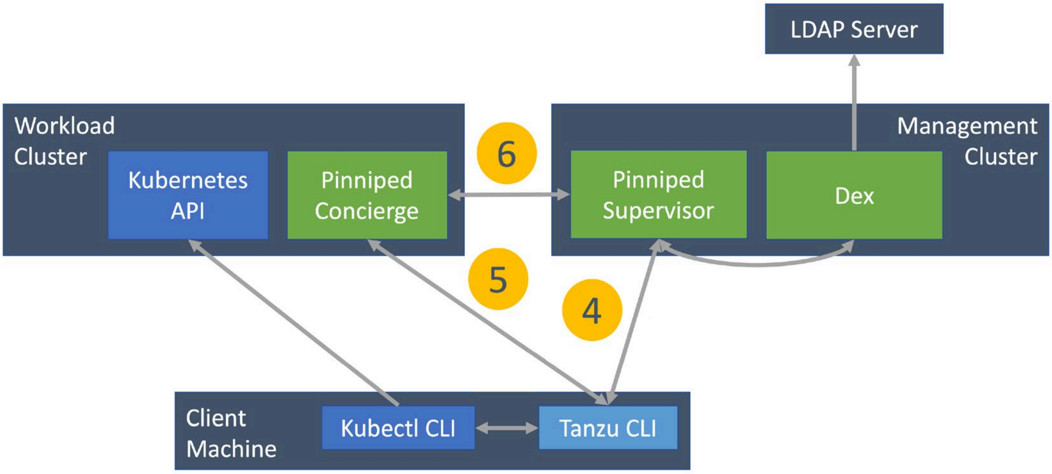
1. kubectl calls the Tanzu CLI.
2. The Tanzu CLI opens a web browser to Pinniped Supervisor, which redirects to the Dex login page.
3. The user provides their LDAP credentials and Dex authenticates with the LDAP server and redirects to Pinniped.



# 3-31 Pinniped Authentication Workflow (2)

The following workflow is initiated when a kubectl command is run and Pinniped authentication is enabled:

4. Pinniped generates an ID token and passes it to the Tanzu CLI.
5. The Tanzu CLI sends the ID token to Pinniped Concierge on the workload cluster.
6. Pinniped Concierge swaps the ID token for a client certificate.



# 3-32 Pinniped Authentication Workflow (3)

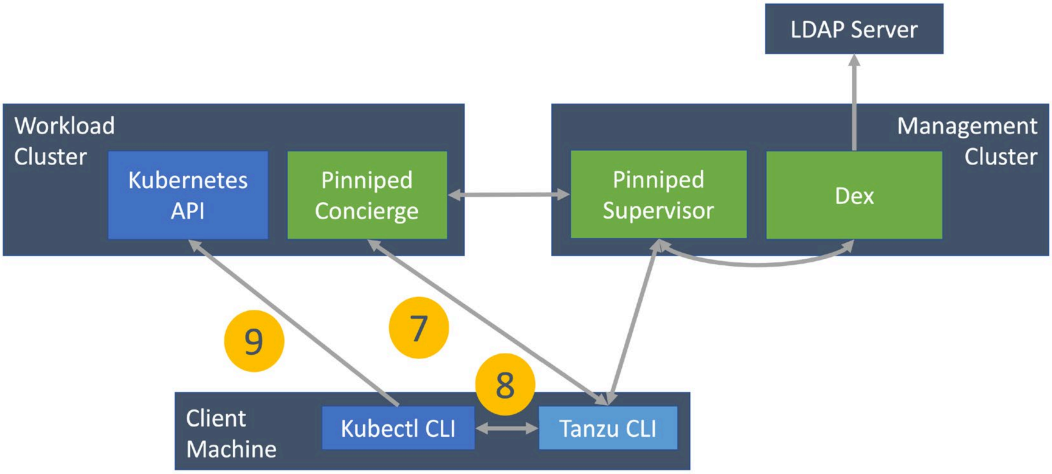
The following workflow is initiated when a kubectl command is run and Pinniped authentication is enabled:

7. The client certificate is passed to the Tanzu CLI.

8. The client certificate is passed to kubectl.

9. kubectl sends the client certificate with its request.

   The client certificate is trusted by the cluster and the user is authenticated.



## 3-33 Pinniped Session Cache

On a successful login, the Tanzu CLI stores the client certificate to `~/.tanzu/pinniped/session.yaml`.

Deleting the file results in the use having to re-authenticate using the Pinniped authentication workflow.

# 3-34 Review of Learner Objectives

- Describe how Kubernetes manages authentication
- Explain what Pinniped is
- Explain what Dex is
- Describe the Pinniped authentication workflow

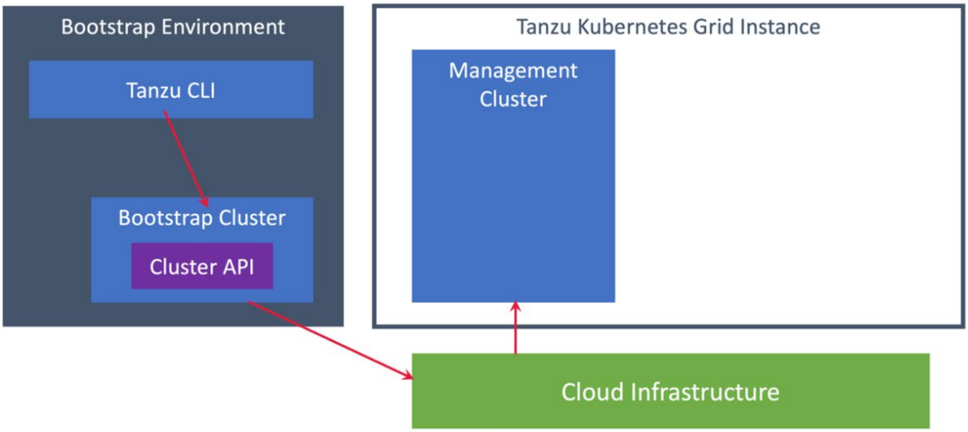
# 3-35 **Lesson 4: Creating Management Clusters**

# 3-36 Learner Objectives

- List the steps to install a Tanzu Kubernetes Grid management cluster
- Describe what happens when a management cluster is created

# 3-37 Initializing Tanzu Kubernetes Grid Instances

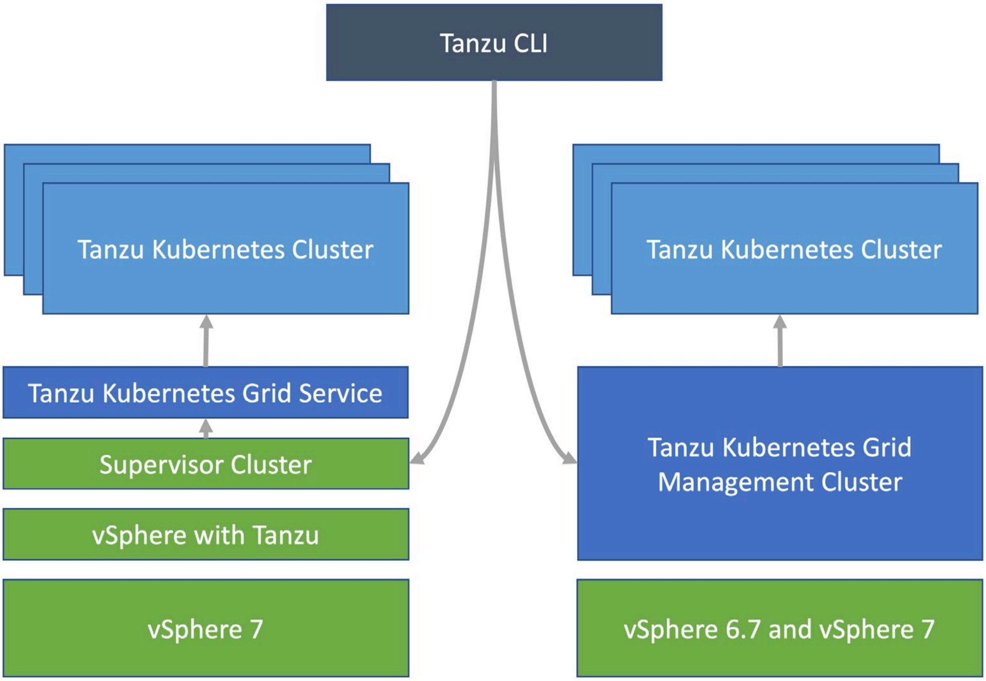
The Tanzu CLI creates a temporary bootstrap cluster which is used to create the management cluster.



# 3-38 Deploying Management Clusters: vSphere 7

Two options exist when deploying management clusters on vSphere 7:

- vSphere with Tanzu enabled:
  - Deploying a management cluster is not supported.
  - The supervisor cluster performs the role of the management cluster.
  - The Tanzu CLI connects to the supervisor cluster.
- vSphere with Tanzu disabled:
  - Deploy using the same process as with vSphere 6.7.



# 3-39 Deploying Management Clusters: Installer UI or CLI

Two methods exist to deploy management clusters:

- Run the Tanzu Kubernetes Grid installer, a wizard that guides you through the deployment process:
    - This is the recommended method, especially for platform operators who are new to Tanzu Kubernetes Grid.
    - The configuration generated by the installer UI is saved to a YAML configuration file.
- Create and edit YAML configuration files to use with CLI commands:
    - This method is useful when using YAML configuration files generated by the installer UI and adding more advanced configuration options.

# 3-40 Tanzu CLI Management Cluster Plug-In

The following flags are available when running the `tanzu management-cluster` command.

| Flag | Description |
|---|---|
| `create` | Creates a Tanzu Kubernetes Grid management cluster. |
| `upgrade` | Upgrades a management cluster. |
| `delete` | Deletes a management cluster |
| `kubeconfig` | Gets the kubeconfig file of a management cluster. |
| `register` | Registers a management cluster with VMware Tanzu Mission Control. |
| `credentials` | Updates the vSphere credentials for a management cluster. |
| `ceip-participation` | Enables or disables CEIP participation. |

# 3-41 Tanzu CLI Management Cluster Create Command

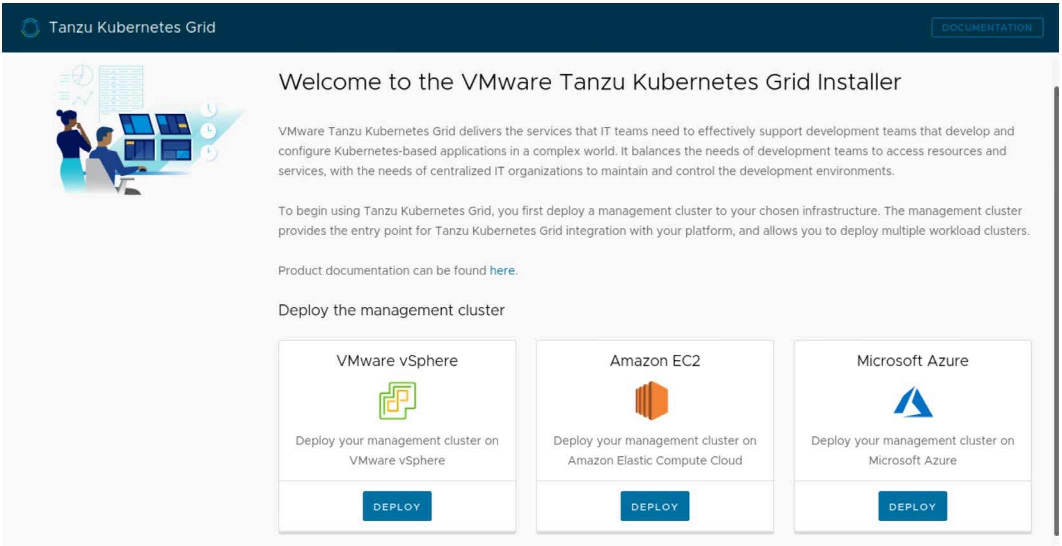
The following flags are available when running the `tanzu management-cluster create` command.

| Flag | Description |
|---|---|
| `-u, --ui` | Launches the management cluster installer UI. |
| `-f, --file` | Configuration file from which to create a management cluster. |
| `-t --timeout` | Time duration to wait for the management cluster deployment to complete. |
| `-e, --use-existing-bootstrap-cluster` | To use an existing bootstrap cluster. |
| `--browser` | To specify the browser in which to open the UI. |
| `-b, --bind` | To bind the UI to a specified IP address and port |

# 3-42 Tanzu Kubernetes Grid Installer (1)

Perform the following steps to install a Tanzu Kubernetes Grid management cluster:

1. Access the installer UI by running `tanzu management-cluster create --ui`.



Welcome to the VMware Tanzu Kubernetes Grid Installer

VMware Tanzu Kubernetes Grid delivers the services that IT teams need to effectively support development teams that develop and configure Kubernetes-based applications in a complex world. It balances the needs of development teams to access resources and services, with the needs of centralized IT organizations to maintain and control the development environments.

To begin using Tanzu Kubernetes Grid, you first deploy a management cluster to your chosen infrastructure. The management cluster provides the entry point for Tanzu Kubernetes Grid integration with your platform, and allows you to deploy multiple workload clusters.

Product documentation can be found here.

Deploy the management cluster

VMware vSphere

Deploy your management cluster on VMware vSphere

DEPLOY

Amazon EC2

Deploy your management cluster on Amazon Elastic Compute Cloud

DEPLOY

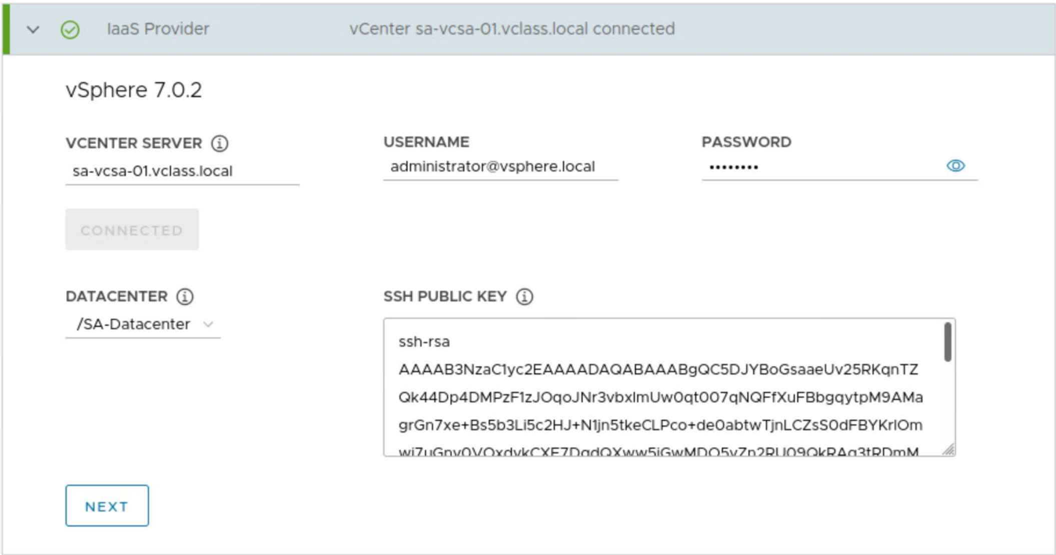
Microsoft Azure

Deploy your management cluster on Microsoft Azure

DEPLOY

# 3-43 Tanzu Kubernetes Grid Installer (2)

2. Provide vCenter Server credentials and SSH public key.



IaaS Provider    vCenter sa-vcsa-01.vclass.local connected

vSphere 7.0.2

VCENTER SERVER: sa-vcsa-01.vclass.local

USERNAME: administrator@vsphere.local

PASSWORD: •••••••••

CONNECTED

DATACENTER: /SA-Datacenter

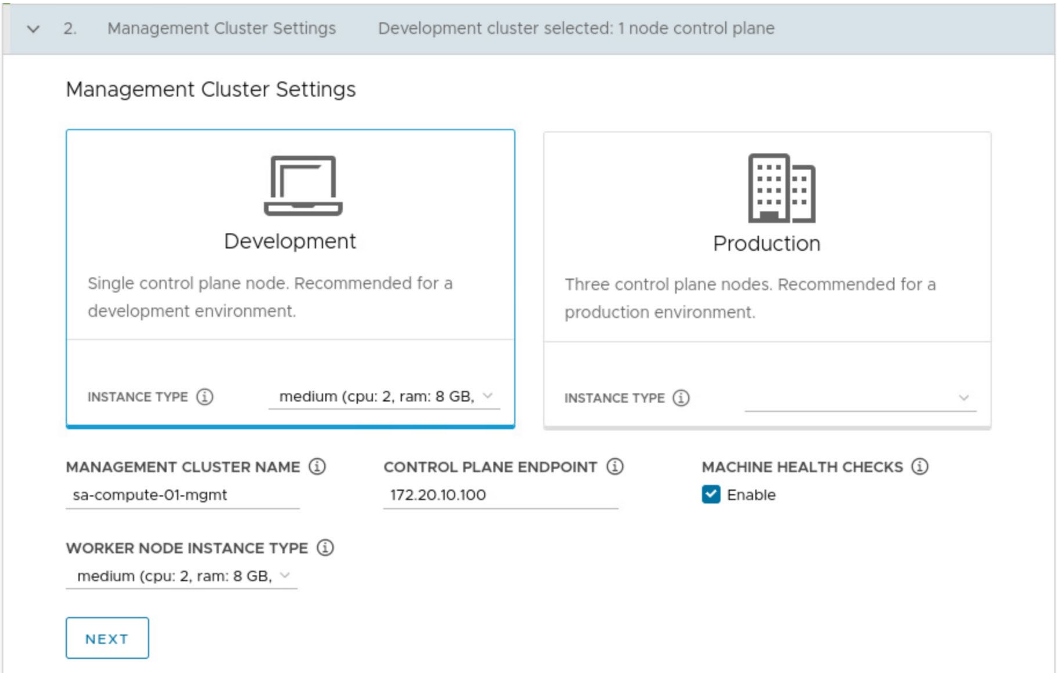
SSH PUBLIC KEY:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC5DJYBoGsaaeUv25RKqnTZ
Qk44Dp4DMPzF1zJOqoJNr3vbxlmUw0qt007qNQFfXuFBbgqytpM9AMa
grGn7xe+Bs5b3Li5c2HJ+N1jn5tkeCLPco+de0abtwTjnLCZsS0dFBYKrlOm
wi7uGnv0VOxdvkCXE7DqdQXww5iGwMDO5vZn2RU09QkRAg3tRDmM
```

NEXT

# 3-44 Tanzu Kubernetes Grid Installer (3)

3. Select the management cluster size and instance types.



Development cluster selected: 1 node control plane

Management Cluster Settings

Single control plane node. Recommended for a development environment.

INSTANCE TYPE: medium (cpu: 2, ram: 8 GB,

Three control plane nodes. Recommended for a production environment.

INSTANCE TYPE:

MANAGEMENT CLUSTER NAME: sa-compute-01-mgmt

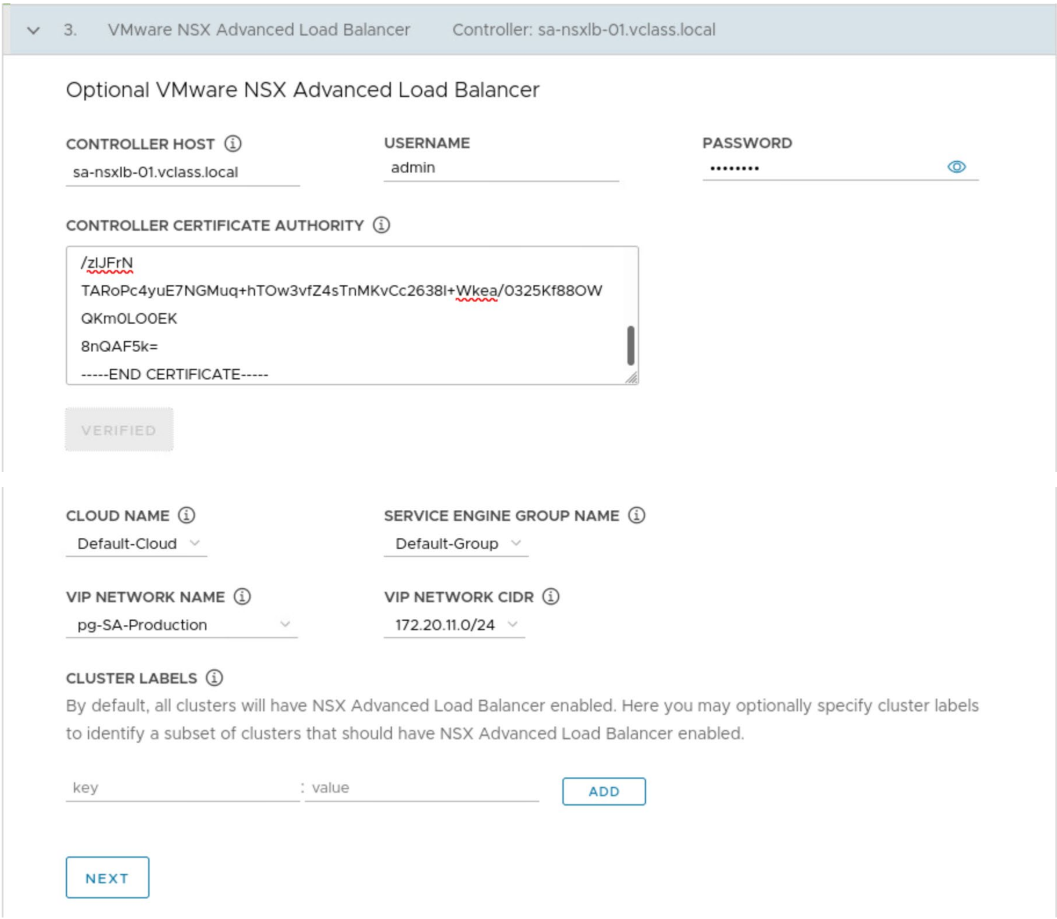
CONTROL PLANE ENDPOINT: 172.20.10.100

MACHINE HEALTH CHECKS: ☑ Enable

WORKER NODE INSTANCE TYPE: medium (cpu: 2, ram: 8 GB,

# 3-45 Tanzu Kubernetes Grid Installer (4)

4. (Optional) Configure NSX Advanced Load Balancer.



3. VMware NSX Advanced Load Balancer   Controller: sa-nsxlb-01.vclass.local

Optional VMware NSX Advanced Load Balancer

CONTROLLER HOST: sa-nsxlb-01.vclass.local

USERNAME: admin

PASSWORD: ••••••••

CONTROLLER CERTIFICATE AUTHORITY:

```
/zIJFrN
TARoPc4yuE7NGMuq+hTOw3vfZ4sTnMKvCc2638I+Wkea/0325Kf88OW
QKm0LO0EK
8nQAF5k=
-----END CERTIFICATE-----
```

VERIFIED

CLOUD NAME: Default-Cloud

SERVICE ENGINE GROUP NAME: Default-Group

VIP NETWORK NAME: pg-SA-Production

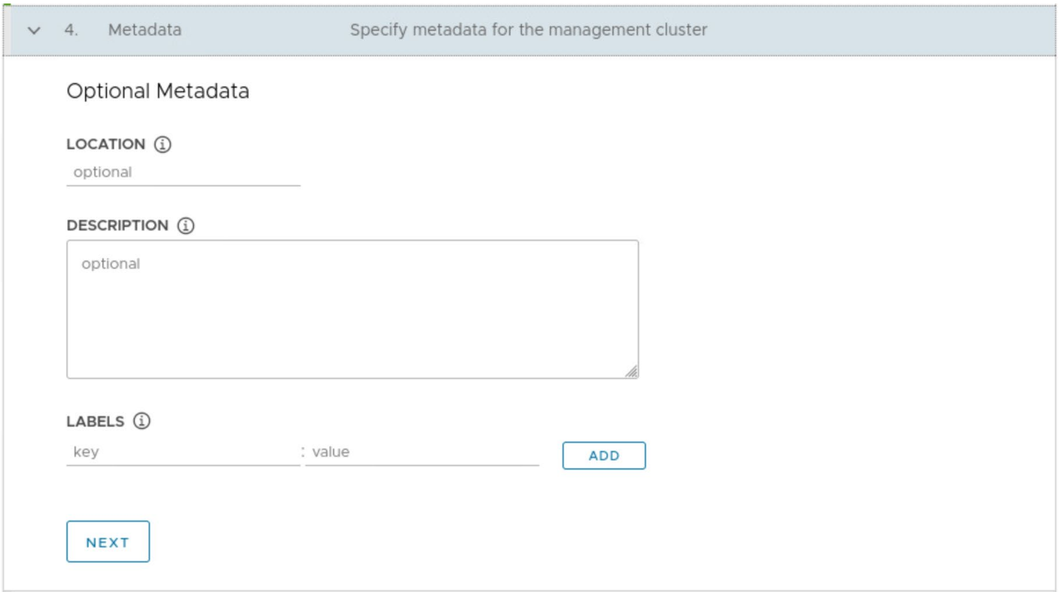
VIP NETWORK CIDR: 172.20.11.0/24

CLUSTER LABELS

By default, all clusters will have NSX Advanced Load Balancer enabled. Here you may optionally specify cluster labels to identify a subset of clusters that should have NSX Advanced Load Balancer enabled.

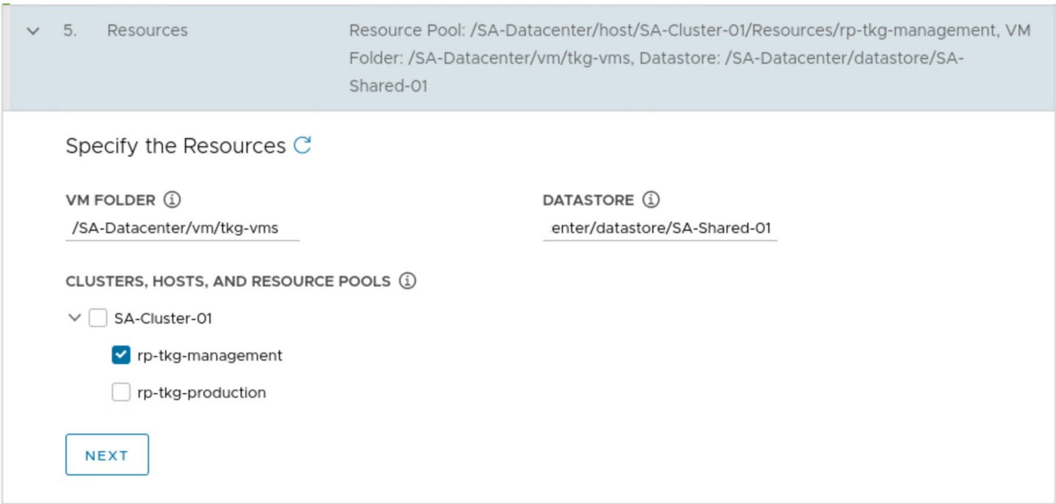
# 3-46 Tanzu Kubernetes Grid Installer (5)

5. Specify metadata and labels to be applied to the cluster.



# 3-47 Tanzu Kubernetes Grid Installer (6)

6. Specify VM folder, datastore, and compute resources.



5. Resources — Resource Pool: /SA-Datacenter/host/SA-Cluster-01/Resources/rp-tkg-management, VM Folder: /SA-Datacenter/vm/tkg-vms, Datastore: /SA-Datacenter/datastore/SA-Shared-01

Specify the Resources

VM FOLDER: /SA-Datacenter/vm/tkg-vms

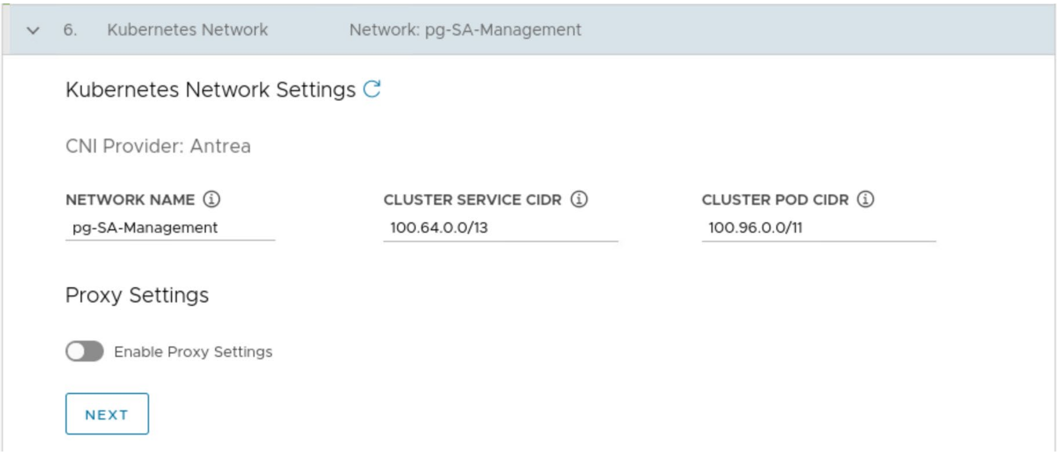
DATASTORE: enter/datastore/SA-Shared-01

CLUSTERS, HOSTS, AND RESOURCE POOLS

- □ SA-Cluster-01
  - ☑ rp-tkg-management
  - □ rp-tkg-production

# 3-48 Tanzu Kubernetes Grid Installer (7)

7. Specify vSphere network, Kubernetes service, and pod CIDR ranges.



6. Kubernetes Network Network: pg-SA-Management

Kubernetes Network Settings

CNI Provider: Antrea

NETWORK NAME: pg-SA-Management

CLUSTER SERVICE CIDR: 100.64.0.0/13

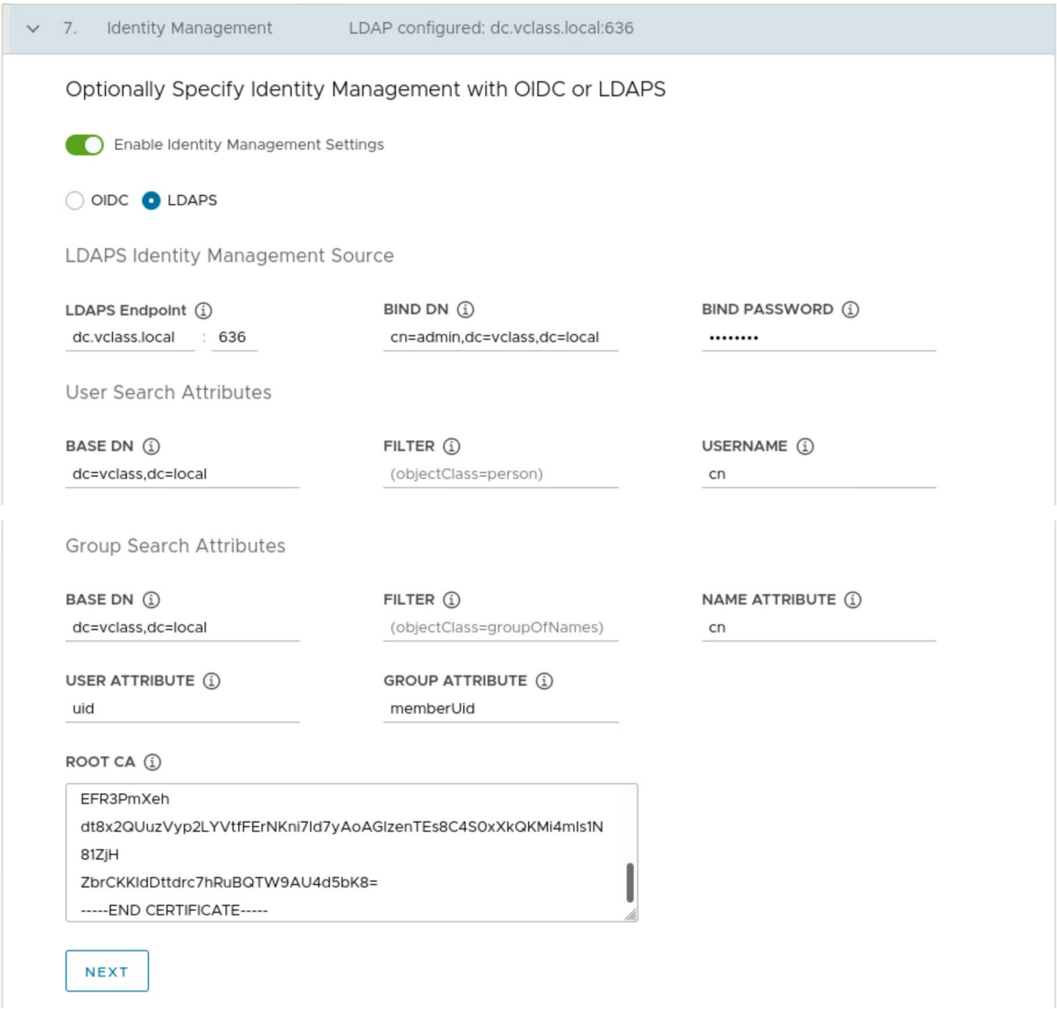
CLUSTER POD CIDR: 100.96.0.0/11

Proxy Settings

Enable Proxy Settings

# 3-49 Tanzu Kubernetes Grid Installer (8)

8. Configure identity management.



7. Identity Management LDAP configured: dc.vclass.local:636

Optionally Specify Identity Management with OIDC or LDAPS

Enable Identity Management Settings

○ OIDC ● LDAPS

LDAPS Identity Management Source

LDAPS Endpoint: dc.vclass.local : 636

BIND DN: cn=admin,dc=vclass,dc=local

BIND PASSWORD: ••••••••

User Search Attributes

BASE DN: dc=vclass,dc=local

FILTER: (objectClass=person)

USERNAME: cn

Group Search Attributes

BASE DN: dc=vclass,dc=local

FILTER: (objectClass=groupOfNames)

NAME ATTRIBUTE: cn

USER ATTRIBUTE: uid

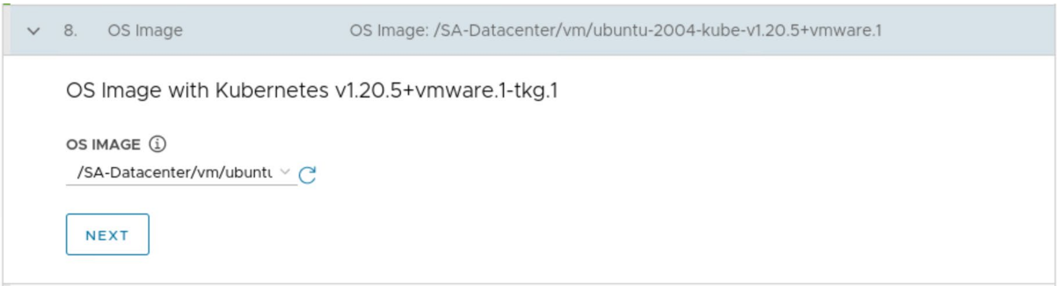
GROUP ATTRIBUTE: memberUid

ROOT CA:

```
EFR3PmXeh
dt8x2QUuzVyp2LYVtfFErNKni7Id7yAoAGIzenTEs8C4S0xXkQKMi4mIs1N
81ZjH
ZbrCKKIdDttdrc7hRuBQTW9AU4d5bK8=
-----END CERTIFICATE-----
```

# 3-50 Tanzu Kubernetes Grid Installer (9)

9. Specify the Kubernetes OS template that was previously uploaded to vCenter Server.



8. OS Image    OS Image: /SA-Datacenter/vm/ubuntu-2004-kube-v1.20.5+vmware.1

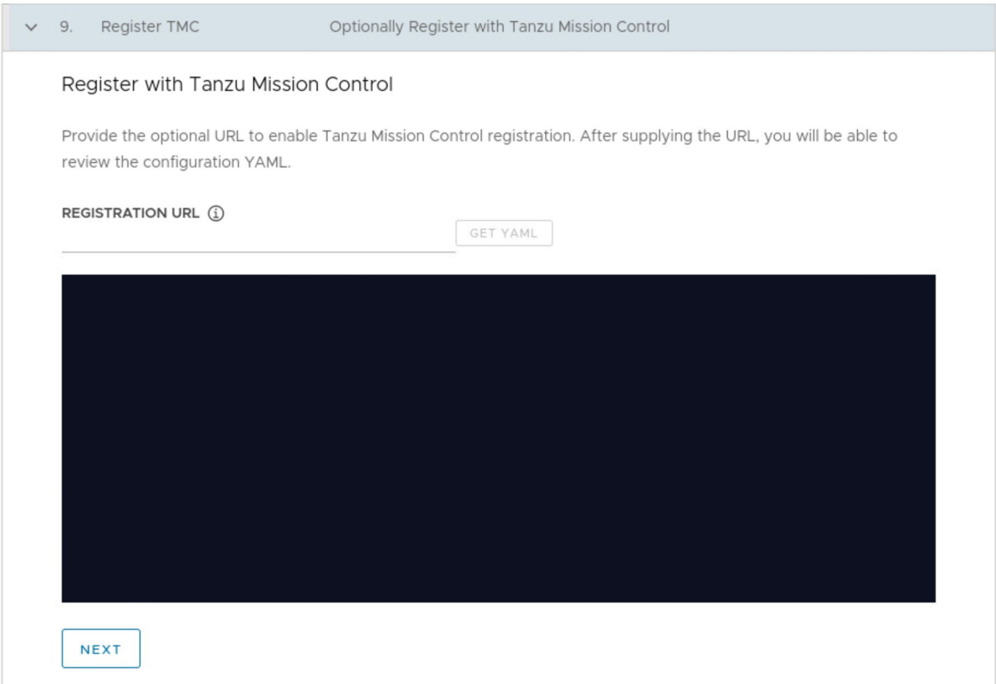
OS Image with Kubernetes v1.20.5+vmware.1-tkg.1

OS IMAGE

/SA-Datacenter/vm/ubuntu

# 3-51 Tanzu Kubernetes Grid Installer (10)

10. Register the management cluster with Tanzu Mission Control.



9. Register TMC — Optionally Register with Tanzu Mission Control

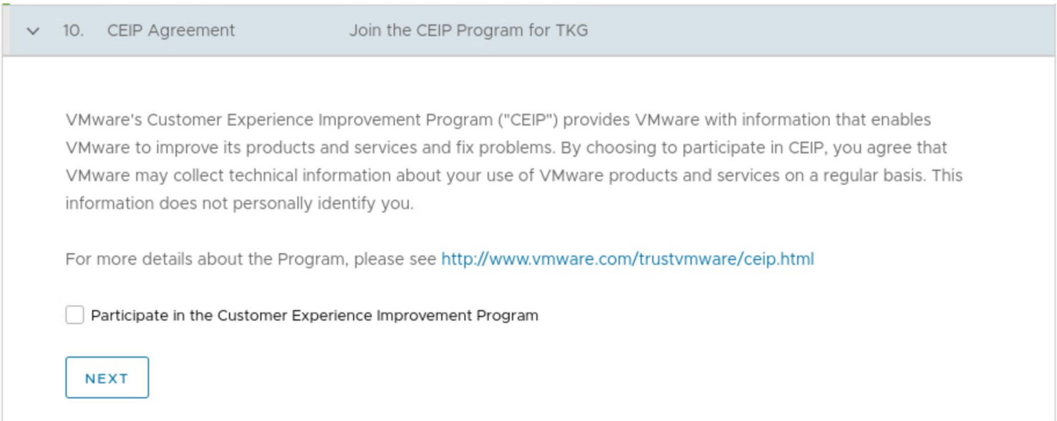
Register with Tanzu Mission Control

Provide the optional URL to enable Tanzu Mission Control registration. After supplying the URL, you will be able to review the configuration YAML.

REGISTRATION URL

# 3-52 Tanzu Kubernetes Grid Installer (11)

11. Register the management cluster with Tanzu Mission Control.



10. CEIP Agreement Join the CEIP Program for TKG

VMware's Customer Experience Improvement Program ("CEIP") provides VMware with information that enables VMware to improve its products and services and fix problems. By choosing to participate in CEIP, you agree that VMware may collect technical information about your use of VMware products and services on a regular basis. This information does not personally identify you.

For more details about the Program, please see http://www.vmware.com/trustvmware/ceip.html

□ Participate in the Customer Experience Improvement Program

For Customer Experience Improvement Program participants, the technical information collected consists of all or any of the following data:

- Configuration data: Technical data about how a customer's organization has configured VMware products and related environment information.

  Examples include version information, configuration settings, and technical data relating to the devices accessing those products or third-party applications or systems used in connection with the VMware product.

- Feature usage data: Data about how a customer organization uses the VMware product.

  Examples include details about which features the customer organization uses and metrics of UI activity.

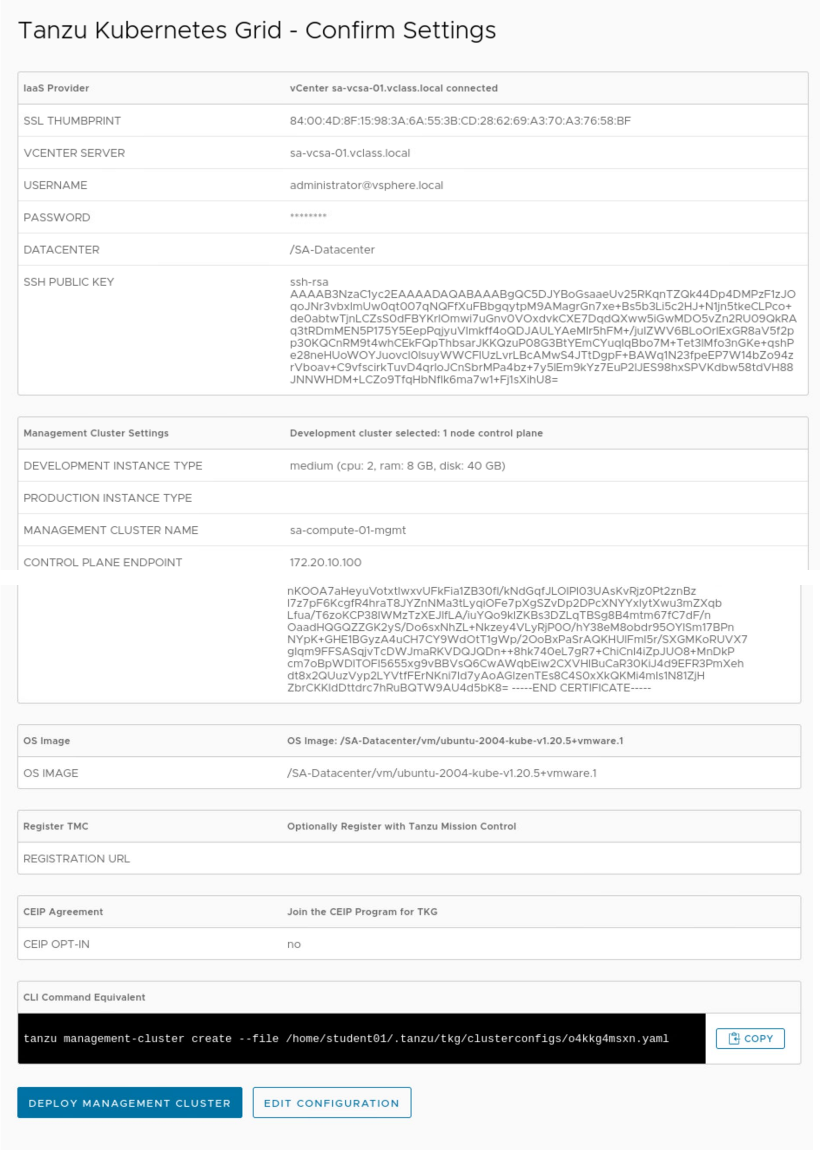
- Performance data: Data about the performance of VMware products.

  Examples include metrics of the performance and scale of VMware products, UI response times, and details about customer API calls.

For more information, see *Customer Experience Improvement Program* at https://www.vmware.com/solutions/trustvmware/ceip.html.

# 3-53 Tanzu Kubernetes Grid Installer (12)

12. Review the configuration and click **DEPLOY MANAGEMENT CLUSTER**.



## Tanzu Kubernetes Grid - Confirm Settings

| IaaS Provider | vCenter sa-vcsa-01.vclass.local connected |
|---|---|
| SSL THUMBPRINT | 84:00:4D:8F:15:98:3A:6A:55:3B:CD:28:62:69:A3:70:A3:76:58:BF |
| VCENTER SERVER | sa-vcsa-01.vclass.local |
| USERNAME | administrator@vsphere.local |
| PASSWORD | ******** |
| DATACENTER | /SA-Datacenter |
| SSH PUBLIC KEY | ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC5DJYBoGsaaeUv25RKqnTZQk44Dp4DMPzF1zJOqoJNr3vbxlmUw0qt007qNQFfXuFBbgqytpM9AMagrGn7xe+Bs5b3Li5c2HJ+N1jn5tkeCLPco+de0abtwTjnLCZsS0dFBYKrlOmwi7uGnv0VOxdvkCXE7DqdQXww5iGwMDO5vZn2RU09QkRAq3tRDmMEN5P175Y5EepPqjyuVlmkff4oQDJAULYAeMlr5hFM+/julZWV6BLoOrlExGR8aV5f2pp30KQCnRM9t4whCEkFQpThbsarJKKQzuP08G3BtYEmCYuqlqBbo7M+Tet3lMfo3nGKe+qshPe28neHUoWOYJuovcl0lsuyWWCFlUzLvrLBcAMwS4JTtDgpF+BAWq1N23fpeEP7W14bZo94zrVboav+C9vfscirkTuvD4qrloJCnSbrMPa4bz+7y5lEm9kYz7EuP2lJES98hxSPVKdbw58tdVH88JNNWHDM+LCZo9TfqHbNflk6ma7w1+Fj1sXihU8= |

| Management Cluster Settings | Development cluster selected: 1 node control plane |
|---|---|
| DEVELOPMENT INSTANCE TYPE | medium (cpu: 2, ram: 8 GB, disk: 40 GB) |
| PRODUCTION INSTANCE TYPE | |
| MANAGEMENT CLUSTER NAME | sa-compute-01-mgmt |
| CONTROL PLANE ENDPOINT | 172.20.10.100 |

| | |
|---|---|
| | nKOOA7aHeyuVotxtlwxvUFkFia1ZB30fl/kNdGqfJLOlPl03UAsKvRjz0Pt2znBzl7z7pF6KcgfR4hraT8JYZnNMa3tLyqiOFe7pXgSZvDp2DPcXNYYxlytXwu3mZXqbLfua/T6zoKCP38lWMzTzXEJlfLA/iuYQo9klZKBs3DZLqTBSg8B4mtm67fC7dF/nOaadHQGQZZGK2yS/Do6sxNhZL+Nkzey4VLyRjP0O/hY38eM8obdr95OYlSm17BPnNYpK+GHE1BGyzA4uCH7CY9WdOtT1gWp/2OoBxPaSrAQKHUlFml5r/SXGMKoRUVX7glqm9FFSASqjvTcDWJmaRKVDQJQDn++8hk740eL7gR7+ChiCnl4iZpJUO8+MnDkPcm7oBpWDlTOFl5655xg9vBBVsQ6CwAWqbEiw2CXVHlBuCaR30KiJ4d9EFR3PmXehdt8x2QUuzVyp2LYVtfFErNKni7Id7yAoAGlzenTEs8C4S0xXkQKMi4mls1N81ZjHZbrCKKldDttdrc7hRuBQTW9AU4d5bK8= -----END CERTIFICATE----- |

| OS Image | OS Image: /SA-Datacenter/vm/ubuntu-2004-kube-v1.20.5+vmware.1 |
|---|---|
| OS IMAGE | /SA-Datacenter/vm/ubuntu-2004-kube-v1.20.5+vmware.1 |

| Register TMC | Optionally Register with Tanzu Mission Control |
|---|---|
| REGISTRATION URL | |

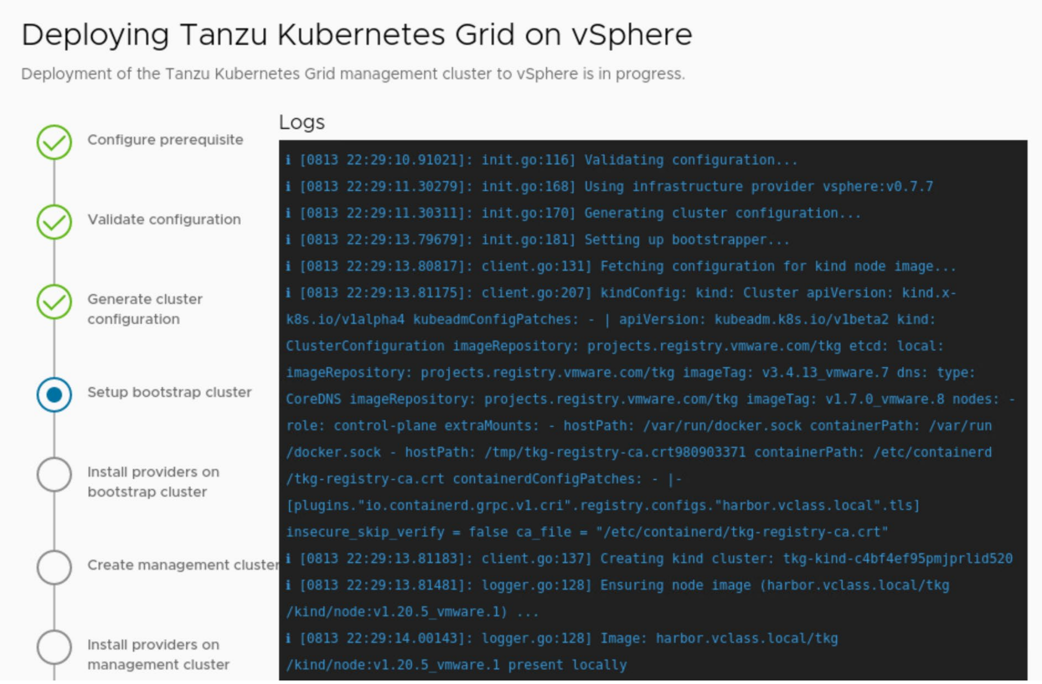
| CEIP Agreement | Join the CEIP Program for TKG |
|---|---|
| CEIP OPT-IN | no |

CLI Command Equivalent

```
tanzu management-cluster create --file /home/student01/.tanzu/tkg/clusterconfigs/o4kkg4msxn.yaml
```

# 3-54 Tanzu Kubernetes Grid Installer (13)

The installer saves the configuration to disk and begins the installation.



Deploying Tanzu Kubernetes Grid on vSphere

Deployment of the Tanzu Kubernetes Grid management cluster to vSphere is in progress.

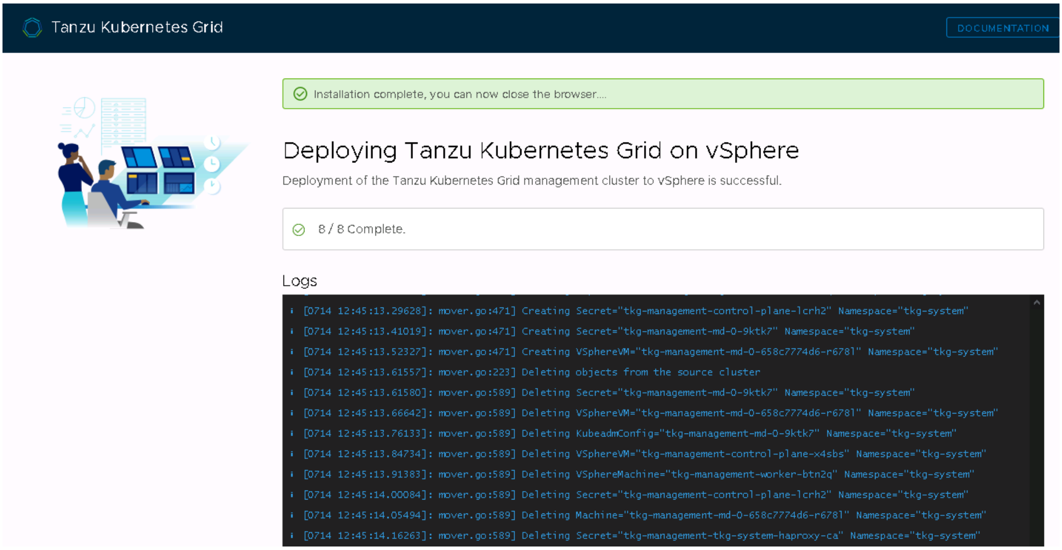
- ✓ Configure prerequisite
- ✓ Validate configuration
- ✓ Generate cluster configuration
- ● Setup bootstrap cluster
- ○ Install providers on bootstrap cluster
- ○ Create management cluster
- ○ Install providers on management cluster

Logs

```
i [0813 22:29:10.91021]: init.go:116] Validating configuration...
i [0813 22:29:11.30279]: init.go:168] Using infrastructure provider vsphere:v0.7.7
i [0813 22:29:11.30311]: init.go:170] Generating cluster configuration...
i [0813 22:29:13.79679]: init.go:181] Setting up bootstrapper...
i [0813 22:29:13.80817]: client.go:131] Fetching configuration for kind node image...
i [0813 22:29:13.81175]: client.go:207] kindConfig: kind: Cluster apiVersion: kind.x-
k8s.io/v1alpha4 kubeadmConfigPatches: - | apiVersion: kubeadm.k8s.io/v1beta2 kind:
ClusterConfiguration imageRepository: projects.registry.vmware.com/tkg etcd: local:
imageRepository: projects.registry.vmware.com/tkg imageTag: v3.4.13_vmware.7 dns: type:
CoreDNS imageRepository: projects.registry.vmware.com/tkg imageTag: v1.7.0_vmware.8 nodes: -
role: control-plane extraMounts: - hostPath: /var/run/docker.sock containerPath: /var/run
/docker.sock - hostPath: /tmp/tkg-registry-ca.crt980903371 containerPath: /etc/containerd
/tkg-registry-ca.crt containerdConfigPatches: - |-
[plugins."io.containerd.grpc.v1.cri".registry.configs."harbor.vclass.local".tls]
insecure_skip_verify = false ca_file = "/etc/containerd/tkg-registry-ca.crt"
i [0813 22:29:13.81183]: client.go:137] Creating kind cluster: tkg-kind-c4bf4ef95pmjprlid520
i [0813 22:29:13.81481]: logger.go:128] Ensuring node image (harbor.vclass.local/tkg
/kind/node:v1.20.5_vmware.1) ...
i [0813 22:29:14.00143]: logger.go:128] Image: harbor.vclass.local/tkg
/kind/node:v1.20.5_vmware.1 present locally
```

# 3-55 Tanzu Kubernetes Grid Installer (14)

When the installation is complete, you can close the browser.



Installation complete, you can now close the browser....

Deploying Tanzu Kubernetes Grid on vSphere

Deployment of the Tanzu Kubernetes Grid management cluster to vSphere is successful.

8 / 8 Complete.

Logs

```
i [0714 12:45:13.29628]: mover.go:471] Creating Secret="tkg-management-control-plane-lcrh2" Namespace="tkg-system"
i [0714 12:45:13.41019]: mover.go:471] Creating Secret="tkg-management-md-0-9ktk7" Namespace="tkg-system"
i [0714 12:45:13.52327]: mover.go:471] Creating VSphereVM="tkg-management-md-0-658c7774d6-r678l" Namespace="tkg-system"
i [0714 12:45:13.61557]: mover.go:223] Deleting objects from the source cluster
i [0714 12:45:13.61580]: mover.go:589] Deleting Secret="tkg-management-md-0-9ktk7" Namespace="tkg-system"
i [0714 12:45:13.66642]: mover.go:589] Deleting VSphereVM="tkg-management-md-0-658c7774d6-r678l" Namespace="tkg-system"
i [0714 12:45:13.76133]: mover.go:589] Deleting KubeadmConfig="tkg-management-md-0-9ktk7" Namespace="tkg-system"
i [0714 12:45:13.84734]: mover.go:589] Deleting VSphereVM="tkg-management-control-plane-x4sbs" Namespace="tkg-system"
i [0714 12:45:13.91383]: mover.go:589] Deleting VSphereMachine="tkg-management-worker-btn2q" Namespace="tkg-system"
i [0714 12:45:14.00084]: mover.go:589] Deleting Secret="tkg-management-control-plane-lcrh2" Namespace="tkg-system"
i [0714 12:45:14.05494]: mover.go:589] Deleting Machine="tkg-management-md-0-658c7774d6-r678l" Namespace="tkg-system"
i [0714 12:45:14.16263]: mover.go:589] Deleting Secret="tkg-management-tkg-system-haproxy-ca" Namespace="tkg-system"
```

# 3-56 Cluster Configuration Options (1)

The management cluster configuration file generated by the installer UI is saved to the `.tanzu/tkg/clusterconfigs` directory.

In the file are configuration parameters that were entered in the wizard and used in the deployment.

vSphere configuration:

```
VSPHERE_SERVER: sa-vcsa-01.vclass.local

VSPHERE_USERNAME: administrator@vsphere.local

VSPHERE_PASSWORD: <encoded:Vk13YXJlMSE=>

VSPHERE_DATASTORE: /SA-Datacenter/datastore/SA-Shared-01

VSPHERE_FOLDER: /SA-Datacenter/vm/tkg-vms

VSPHERE_DATACENTER: /SA-Datacenter

VSPHERE_NETWORK: pg-SA-Management

VSPHERE_RESOURCE_POOL: /SA-Datacenter/host/SA-Cluster-01/Resources/rp-tkg-management
```

# 3-57 Cluster Configuration Options (2)

Kubernetes network CIDRs:

```
SERVICE_CIDR: 100.64.0.0/13
CLUSTER_CIDR: 100.96.0.0/11
```

VM characteristics:

```
VSPHERE_WORKER_MEM_MIB: "8192"
VSPHERE_WORKER_NUM_CPUS: "2"
VSPHERE_WORKER_DISK_GIB: "40"
VSPHERE_CONTROL_PLANE_NUM_CPUS: "2"
VSPHERE_CONTROL_PLANE_MEM_MIB: "8192"
VSPHERE_CONTROL_PLANE_DISK_GIB: "40"
```

## 3-58 Cluster API Configuration

The following Cluster API components from folder `.tanzu/tkg/providers/` are deployed on the management cluster:

- `cluster-api/v0.3.14/core-components.yaml`
- `control-plane-kubeadm/v0.3.14/control-plane-components.yaml`
- `bootstrap-kubeadm/v0.3.14/bootstrap-components.yaml`

## 3-59 Infrastructure Configuration

The Cluster API for vSphere controllers and CRDs listed in the following YML file are deployed on the management cluster when running on vSphere 6.7 Update 3:

```
.tanzu/tkg/providers/infrastructure-
vsphere/v0.7.7/infrastructure-components.yaml
```

For vSphere 7, no configuration is required because the supervisor cluster is preconfigured with the Cluster API for vSphere controllers and CRDs.

For other IaaS providers, the following directories contain the deployment YML files:

- Amazon Web Services: `.tanzu/tkg/providers/infrastructure-aws`
- Microsoft Azure: `.tanzu/tkg/providers/infrastructure-azure`

# 3-60 Management Cluster Deployment Workflow

The `tanzu management-cluster create` command performs the following steps:

1. It validates the provided configuration.
2. Using `kind`, it creates a local bootstrap cluster on the bootstrap machine.
3. It deploys the Cluster API pods to the bootstrap cluster.
4. It deploys the requested management cluster configuration, using Cluster API CRDs, to the bootstrap cluster.
5. The Cluster API provider for vSphere on the bootstrap cluster communicates with vSphere to clone and configure the cluster VMs in vSphere.
6. When the cluster is available, the Cluster API pods and configuration are moved to the vSphere management cluster.
7. The local bootstrap cluster is deleted.

When using Amazon Web Services or Microsoft Azure, the matching Cluster API provider (CAPZ or CAPA) communicates with the IaaS providers to create and configure the cluster VMs.

## 3-61 Deployment Logs (1)

The following logs are an example of a successful deployment of a management cluster:

```
Validating the pre-requisites...


Setting up management cluster...
Validating configuration...
Using infrastructure provider vsphere:v0.6.4
Generating cluster configuration...
```

## 3-62 Deployment Logs (2)

The following logs are an example of a successful deployment of a management cluster:

```
Setting up bootstrapper...
Installing providers on bootstrapper...
Installing cert-manager
Waiting for cert-manager to be available...
Installing Provider="cluster-api" Version="v0.3.5"
TargetNamespace="capi-system"
Installing Provider="bootstrap-kubeadm" Version="v0.3.5"
TargetNamespace="capi-kubeadm-bootstrap-system"
Installing Provider="control-plane-kubeadm" Version="v0.3.5"
TargetNamespace="capi-kubeadm-control-plane-system"
Installing Provider="infrastructure-vsphere" Version="v0.6.4"
TargetNamespace="capv-system"
```

# 3-63 Deployment Logs (3)

The following logs are an example of a successful deployment of a management cluster:

```
Start creating management cluster...

Saving management cluster kuebconfig into
/home/vmware/.kube/config

Installing providers on management cluster...

Installing cert-manager

Waiting for cert-manager to be available...

Installing Provider="cluster-api" Version="v0.3.5"
TargetNamespace="capi-system"

Installing Provider="bootstrap-kubeadm" Version="v0.3.5"
TargetNamespace="capi-kubeadm-bootstrap-system"

Installing Provider="control-plane-kubeadm" Version="v0.3.5"
TargetNamespace="capi-kubeadm-control-plane-system"

Installing Provider="infrastructure-vsphere" Version="v0.6.4"
TargetNamespace="capv-system"
```

# 3-64 Deployment Logs (4)

The following logs are an example of a successful deployment of a management cluster:

```
Waiting for the management cluster to get ready for move...

Moving all Cluster API objects from bootstrap cluster to management cluster...

Performing move...

Discovering Cluster API objects

Moving Cluster API objects Clusters=1

Creating objects in the target cluster

Deleting objects from the source cluster

Context set for management cluster sa-compute-01-mgmt as 'sa-compute-01-mgmt-admin@sa-compute-01-mgmt'.

Management cluster created!
```

# 3-65 Management Cluster VMs

The following management cluster VMs are created by Tanzu Kubernetes Grid.

| Name | Description |
|---|---|
| `<CLUSTER_NAME>-control-plane-xxxxx` | Control plane VMs. |
| `<CLUSTER_NAME>-md-0-xxxxxxxxxx-xxxxx` | Worker node VMs.<br>*md* is an abbreviation for MachineDeployment, which is a Cluster API custom resource. |

## 3-66 Lab 4: Deploying a Management Cluster

Create a Tanzu Kubernetes Grid management cluster by using the Tanzu Kubernetes Grid installer:

1. Create a Management Cluster
2. Rename the Management Cluster Configuration File
3. Examine the Management Cluster

## 3-67 Review of Learner Objectives

- List the steps to install a Tanzu Kubernetes Grid management cluster
- Describe what happens when a management cluster is created

# 3-68 Lesson 5: Managing Management Clusters

## 3-69 Learner Objectives

- Describe the commands available for working with management clusters

# 3-70 Working with Multiple Management Clusters

You can list, view, and switch contexts between multiple management clusters by using the following commands.

| Command | Description |
|---|---|
| `tanzu login` | Displays the list of management clusters that you deployed and enables changes to the `.kube-tkg/config` context of the Tanzu Kubernetes Grid CLI to a different management cluster. |
| `tanzu management-cluster get` | Displays the details of a specific management cluster. |

# 3-71 Adding Existing Management Clusters to the Tanzu CLI

To add a management cluster that was created in another environment:

1. Run the `tanzu login` command.
2. Select **new server**.
3. Select **Server endpoint** and provide the login URL of a vSphere with Tanzu supervisor cluster.
4. Select **Local kubeconfig** and provide the kubeconfig file of another management cluster.

You can also add to a management cluster by using the following command.

```
tanzu login --kubeconfig <KUBE-CONFIG-PATH> --context
<CONTEXT-NAME> --name <MGMT-CLUSTER-NAME>
```

# 3-72 Review of Learner Objectives

- Describe the commands available for working with management clusters

# 3-73 Key Points

- OVA templates with the required Kubernetes version are deployed to vSphere.
- The Tanzu Kubernetes Grid installer provides a UI to enter the initial configuration.
- The bootstrap cluster is created locally and is used to create the management cluster on vSphere.

# Module 4

# Tanzu Kubernetes Clusters

## 4-2 Importance

Tanzu Kubernetes clusters are where application workloads run. Provisioning and configuring Tanzu Kubernetes clusters is the key functionality that Tanzu Kubernetes Grid provides.

## 4-3 Module Lessons

1. Building Custom Cluster API Machine Images
2. Deploying Tanzu Kubernetes Clusters
3. Managing Tanzu Kubernetes Clusters
4. Tanzu Kubernetes Cluster Architecture

## 4-4 **Lesson 1: Building Custom Cluster API Machine Images**

## 4-5 Learner Objectives

- Describe the steps to build a custom image
- Describe the available customizations

# 4-6 Deploying Clusters Using Custom Machine Images

You can build custom machine images to use as a VM template for the Tanzu Kubernetes workload cluster nodes.

Each custom machine image packages a base OS version and a Kubernetes version, as well as any additional customizations, into an image that runs on vSphere, Amazon Web Services (AWS), or Microsoft Azure infrastructure.

The base OS can be an OS that VMware supports but does not distribute, such as Red Hat Enterprise Linux (RHEL) v7.

# 4-7 Custom Machine Image Customizations

Several variables can be used to customize the image build.

| Variable | Description |
|---|---|
| `disable_public_repos` | If set to "true", disable all existing package repositories defined in the OS before installing any packages. |
| `extra_debs` | Names of additional deb packages to install. |
| `extra_repos` | Files to add to the image, containing repository definitions. |
| `extra_rpms` | Names of additional RPM packages to install. |
| `containerd_additional_settings` | Contains additional configuration for containerd. |
| `additional_executables` | If set to "true", load additional executables from a url. |
| `ansible_user_vars` | String that the user can pass to use in the ansible roles. |

For more information, see *The Image Builder Book* on the Kubernetes website at https://image-builder.sigs.k8s.io/capi/capi.html#customization.

# 4-8 Building Custom Machine Images: Requirements

To create a custom machine image, you must meet the following requirements:

- An account on your target infrastructure, such as vSphere, AWS, or Microsoft Azure.
- Docker is installed on a macOS or Linux workstation.
- For AWS, the `aws` CLI is installed.
- For Microsoft Azure, the `az` CLI is installed.
- For vSphere, a copy of the `OVFTool` Linux installer.

# 4-9 Building Custom Machine Images: Steps

To create a custom machine image, the following steps are required:

1 Determine and download the Image Builder configuration version that you want to build from.

  Each version corresponds to the Kubernetes version that Image Builder uses. For example, `TKG-Image-Builder-for-Kubernetes-v1.20.5-master.zip` builds a Kubernetes v1.20.5 image.

2 For vSphere, load OVF Tool in the Image Builder Container Image.

3 Prepare the configuration.

4 Run Image Builder.

# 4-10 Using Custom Machine Images

After you create a custom image, you enable the Tanzu CLI to use the image by creating a custom Tanzu Kubernetes release based on the image:

1. Open the Bill of Materials (BOM) file corresponding to the Kubernetes version of your custom image.
2. Edit or add configuration.
3. Save the BOM file.
4. Base64-encode the file contents.
5. Create a ConfigMap YAML file in the tkr-system namespace.
6. Save the ConfigMap file, set the kubectl context to a management cluster that you want to add Tanzu Kubernetes release to, and apply the file to the cluster.

   Example:

   ```
   kubectl apply -f my-custom-tkr-bom-v1.20.5---vmware.2-
   tkg.1.yaml
   ```
7. Verify that the custom Tanzu Kubernetes release was added by running `tanzu kubernetes-release get` or `kubectl get tkr`, and look for the `CUSTOM-TKR` value set, listed above, in the output.

## 4-11 Review of Learner Objectives

- Describe the steps to build a custom image
- Describe the available customizations

# 4-12 Lesson 2: Deploying Tanzu Kubernetes Clusters

## 4-13 Learner Objectives

- Describe the options for deploying Tanzu Kubernetes clusters
- Describe how Tanzu Kubernetes clusters are created

# 4-14 Tanzu CLI Cluster Plug-In

The Tanzu CLI core executable must be installed, followed by the CLI plug-ins related to Tanzu Kubernetes cluster management and feature operations.

After the plug-ins are installed, the following commands are available when running the `tanzu cluster` command.

| Command | Description |
|---|---|
| `create` | Creates a Tanzu Kubernetes Grid cluster. |
| `credentials` | Updates the vSphere credentials for a cluster. |
| `delete` | Deletes a cluster. |
| `get` | Get details from a cluster. |
| `kubeconfig` | Gets the kubeconfig file of a cluster. |
| `list` | Lists clusters. |
| `machinehealthcheck` | Configures machine health checks. |
| `scale` | Scales a cluster. |
| `upgrade` | Upgrades a cluster. |

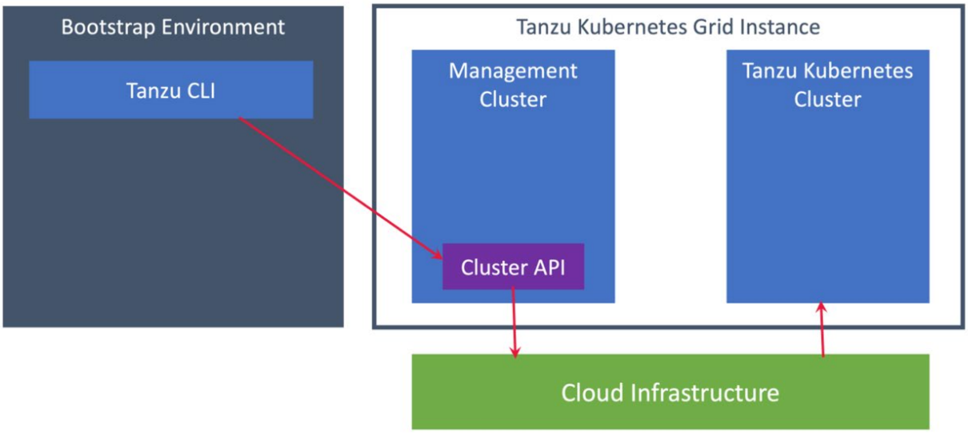
# 4-15 Tanzu CLI Cluster Create Command

The following flags are available when running `tanzu cluster create`.

| Flag | Description |
|---|---|
| `-d, --dry-run` | Displays only the deployment YAML, instead of creating the cluster. |
| `-f, --file` | Specifies a configuration file from which to create a cluster. |
| `--tkr` | Creates a cluster using a specific Tanzu Kubernetes release version. |
| `--log-file` | Provides a path to a file for saving the cluster create operation logs. |

# 4-16 Creating Tanzu Kubernetes Clusters

When you run `tanzu cluster create -f tkc-01.yaml`, the Tanzu CLI communicates with the management cluster to create Tanzu Kubernetes clusters.



The `tanzu cluster create` command performs the following steps:

1. It validates the provided configuration by querying vSphere for the defined resources.
2. It deploys the requested cluster configuration to the management cluster by using Cluster API CRDs.
3. The Cluster API provider for vSphere on the management cluster communicates with vSphere to clone and configure the cluster VMs in vSphere.

# 4-17 Cluster Configuration Files

Cluster configuration files contain the configuration parameters that get passed to the Tanzu CLI and combined with the cluster plan template files.

Example configuration parameters:

```
VSPHERE_WORKER_MEM_MIB: "8192"

VSPHERE_WORKER_NUM_CPUS: "2"

VSPHERE_WORKER_DISK_GIB: "40"

VSPHERE_CONTROL_PLANE_NUM_CPUS: "2"

VSPHERE_CONTROL_PLANE_MEM_MIB: "8192"

VSPHERE_CONTROL_PLANE_DISK_GIB: "40"

SERVICE_CIDR: 100.64.0.0/13

CLUSTER_CIDR: 100.96.0.0/11
```

For a complete list of configuration parameters, see Tanzu CLI Configuration File Variable Reference at https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-tanzu-config-reference.html.

## 4-18 Source of Cluster Configuration Values

When the Tanzu CLI creates a cluster, it combines configuration values from the following locations:

- Cluster plan YAML configuration files in `~/.tanzu/tkg/providers`
- `~/.tanzu/tkg/cluster-config.yaml` or other file passed to the CLI `-f` option
- Environment variables
- Command-line input

# 4-19 Deploying Clusters with a Highly Available Control Plane

The `CLUSTER_PLAN` parameter specifies the number of control plane nodes:

- Setting `CLUSTER_PLAN` to `dev` deploys a single control plane node.
- Setting `CLUSTER_PLAN` to `prod` deploys a highly available cluster with three control plane nodes.

The following VMs are created by Cluster API if the `prod` plan is selected:

- Three control plane VMs, with names similar to `tkc-01-control-plane-nj4z6`
- Three worker node VMs, with names similar to `tkc-01-md-0-6ff9f5cffb-jhcrh`

# 4-20 Previewing the YAML File for Tanzu Kubernetes Clusters

You can use the `--dry-run` flag to preview the YAML file that Tanzu Kubernetes Grid will create when it deploys a Tanzu Kubernetes cluster.

Example command:

```
tanzu cluster create -f tkc-01.yaml --dry-run
```

This flag is useful if you need to inspect the generated YAML configuration to ensure the correct values were entered.

# 4-21 Overriding Default Configuration Parameters

The Tanzu CLI `management-cluster` and `cluster` commands read the configuration parameters from the specified file by using `-f` or `--file`.

The parameters can be overridden by setting environment variables before running the CLI:

- `export VSPHERE_NETWORK="pg-SA-Production"`
- `export VSPHERE_RESOURCE_POOL="/SA-Datacenter/host/SA-Cluster-01/Resources/rp-tkg-production"`

The environment variables take precedence over the configuration file parameters.

# 4-22 Deploying Clusters that Run a Specific Kubernetes Version

Each release of Tanzu Kubernetes Grid:

- Provides a default Kubernetes version
- Supports a defined set of Kubernetes versions

Example command to deploy a cluster using a specific Kubernetes version:

-
  ```
  tanzu cluster create -f tkc-01.yaml --tkr v1.19.9---
  vmware.2-tkg.1
  ```

## 4-23 Lab 5: Deploying a Tanzu Kubernetes Cluster

Deploy a Tanzu Kubernetes cluster that will be given to a development team:

1. Create a Tanzu Kubernetes Cluster Configuration File
2. Create a Tanzu Kubernetes Cluster

## 4-24 Review of Learner Objectives

- Describe the options for deploying Tanzu Kubernetes clusters
- Describe how Tanzu Kubernetes clusters are created

# 4-25 **Lesson 3: Managing Tanzu Kubernetes Clusters**

## 4-26 Learner Objectives

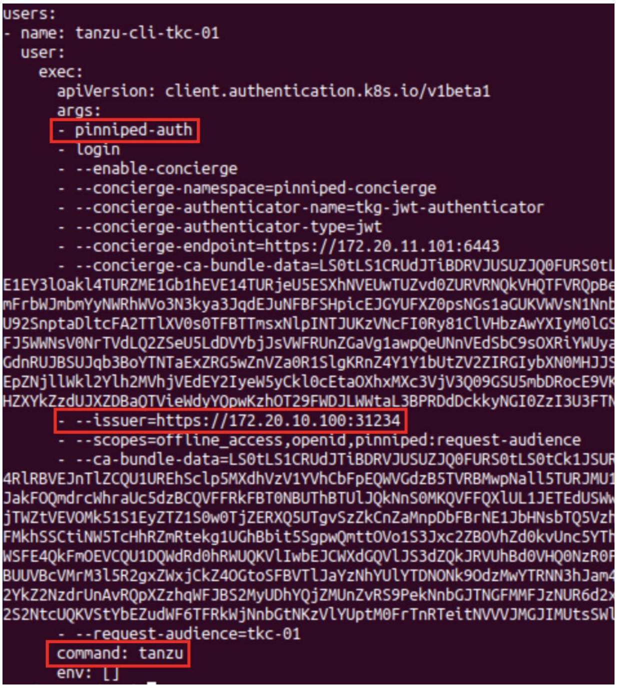
- Describe the commands available for working with Tanzu Kubernetes clusters

# 4-27 Granting Authenticated Access to a Cluster

Running the `tanzu cluster kubeconfig get <CLUSTER-NAME>` command generates a kubeconfig file that can be shared with users who require access to a cluster.

The kubeconfig file is configured with instructions that run the Tanzu CLI and trigger the Pinniped authentication workflow.

Though this process authenticates a user to the cluster, the user still requires permissions to be granted on the cluster to interact with Kubernetes API resources.



```
users:
- name: tanzu-cli-tkc-01
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1beta1
      args:
      - pinniped-auth
      - login
      - --enable-concierge
      - --concierge-namespace=pinniped-concierge
      - --concierge-authenticator-name=tkg-jwt-authenticator
      - --concierge-authenticator-type=jwt
      - --concierge-endpoint=https://172.20.11.101:6443
      - --concierge-ca-bundle-data=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tL
E1EY3lOakl4TURZME1Gb1hEVE14TURjeU5ESXhNVEUwTUZvd0ZURVRNQkVHQTFVRQpBe
mFrbWJmbmYyNWRhWVo3N3kya3JqdEJuNFBFSHpicEJGYUFXZ0psNGs1aGUKVWVsN1Nnb
U92SnptaDltcFA2TTlXV0s0TFBTTmsxNlpINTJUKzVNcFI0Ry81ClVHbzAwYXIyM0lGS
FJ5WWNsV0NrTVdLQ2ZSeU5LdDVYbjJsVWFRUnZGaVg1awpQeUNnVEdSbC9sOXRiYWUya
GdnRUJBSUJqb3BoYTNTaExxZRG5wZnVZa0R1SlgKRnZ4Y1Y1bUtZV2ZIRGIybXN0MHJJS
EpZNjllWkl2Ylh2MVhjVEdEY2IyeW5yCkl0cEtaOXhxMXc3VjV3Q09GSU5mbDRocE9VK
HZXYkZzdUJXZDBaQTVieWdyYQpwKzhOT29FWDJLWWtaL3BPRDdDckkyNGI0ZzI3U3FTN
      - --issuer=https://172.20.10.100:31234
      - --scopes=offline_access,openid,pinniped:request-audience
      - --ca-bundle-data=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR
4RlRBVEJnTlZCQU1UREhSclp5MXdhVzV1YVhCbFpEQWVGdzB5TVRBMwpNall5TURJMU1
JakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0MKQVFFQXlUL1JETEdUSWw
jTWZtVEVOMk51S1EyZTZ1S0w0TjZERXQ5UTgvSzZkCnZaMnpDbFBrNE1JbHNsbTQ5Vzh
FMkhSSCtiNW5TcHhRZmRtekg1UGhBbit5SgpwQmttOVo1S3Jxc2ZBOVhZd0kvUnc5YTh
WSFE4QkFmOEVCQU1DQWdRd0hRWUQKVlIwbEJCWXdGQVlJS3dZQkJRVUhBd0VHQ0NzR0F
BUUVBcVMrM3l5R2gxZWxjCkZ4OGtoSFBVTlJaYzNhYUlYTDNONk9OdzMwYTRNN3hJam4
2YkZ2NzdrUnAvRQpXZzhqWFJBS2MyUDhYQjZMUnZvRS9PekNnbGJTNGFMMFJzNUR6d2x
2S2NtcUQKVStYbEZudWF6TFRkWjNnbGtNKzVlYUptM0FrTnRTeitNVVVJMGJIMUtsSWl
      - --request-audience=tkc-01
      command: tanzu
      env: []
```

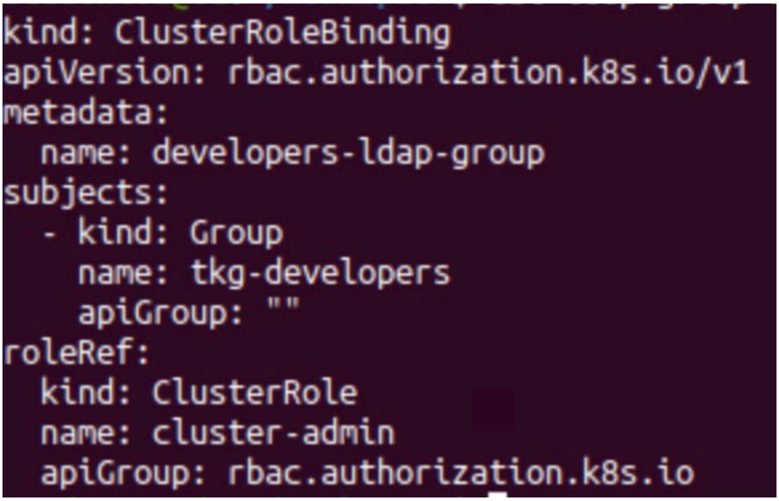
# 4-28 Granting Access to Cluster Resources

An authenticated user needs permissions to perform actions on the cluster.

RoleBindings or ClusterRoleBindings are required to grant access to specific cluster resources.

In this example, ClusterRoleBinding grants users in the `tkg-developers` group full `cluster-admin` permissions.

Typically, permissions would be more strict and detailed.



```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: developers-ldap-group
subjects:
  - kind: Group
    name: tkg-developers
    apiGroup: ""
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

# 4-29 Scaling Clusters

To scale clusters, use the `scale` command and include the required `controlplane-machine-count` and `worker-machine-count` values.

```
tanzu cluster scale tkc-01 --controlplane-machine-count 3 --
worker-machine-count 3
```

The Tanzu CLI modifies the Cluster API spec of the cluster, which triggers the Cluster API controllers to create new control plane or worker nodes.

Control plane and worker nodes can be scaled up and down.

Control plane nodes can only be scaled to an odd number of nodes. This is to prevent quorum issues during etcd leader elections.

## 4-30 Scaling Management Clusters

Because the management clusters run in the `tkg-system` namespace, the `--namespace` option must be specified when scaling a management cluster.

```
tanzu cluster scale <mgmt-cluster-name> --controlplane-
machine-count 3 --worker-machine-count 3 --namespace tkg-
system
```

Scaling management clusters is not typically needed because the management workloads do not consume much resources.

# 4-31 Cluster Autoscaler

To enable Cluster Autoscaler for the workload cluster, you set the `AUTOSCALER_` options in the configuration file that you use to deploy the cluster.

| | |
|---|---|
| `ENABLE_AUTOSCALER` | Set to `true` or `false` to enable or disable autoscaling. |
| `AUTOSCALER_MAX_NODES_TOTAL` | Limits the count of all nodes, both the worker node and control plane. |
| `AUTOSCALER_SCALE_DOWN_DELAY_AFTER_ADD` | Amount of time that Cluster Autoscaler waits after a scale-up operation to resume scale-down scans. |
| `AUTOSCALER_SCALE_DOWN_DELAY_AFTER_DELETE` | Amount of time that Cluster Autoscaler waits after deleting to resume scale-down scans. |
| `AUTOSCALER_MIN_SIZE_0,`<br>`AUTOSCALER_MAX_SIZE_0` | Cluster Autoscaler does not attempt to scale the nodes beyond or under this limit. |

- For clusters with a single machine deployment such as dev clusters on vSphere, Amazon EC2, or Azure and prod clusters on vSphere or Azure, set:
  `AUTOSCALER_MIN_SIZE_0 and AUTOSCALER_MAX_SIZE_0`
- For clusters with multiple machine deployments such as prod clusters on Amazon EC2, set:
  `AUTOSCALER_MIN_SIZE_0 and AUTOSCALER_MAX_SIZE_0`
  `AUTOSCALER_MIN_SIZE_1 and AUTOSCALER_MAX_SIZE_1`
  `AUTOSCALER_MIN_SIZE_2 and AUTOSCALER_MAX_SIZE_2`

# 4-32 Machine Health Checks

MachineHealthCheck is a controller that provides node health monitoring and node auto-repair for Tanzu Kubernetes clusters.

The controller monitors for the following node conditions:

- Ready
- MemoryPressure
- DiskPressure
- PIDPressure
- NetworkUnavailable

If a node's status reports the condition for the specified amount of time, the node is considered unhealthy and is recreated by Cluster API.

## 4-33 Configuring Machine Health Checks

All created clusters must have **ENABLE_MHC** set to **true** or **false** to enable or disable **MachineHealthCheck**. This setting can be modified later by using the tanzu CLI.

You use the `tanzu cluster machinehealthcheck set` command to configure or modify machine health checks.

Example: `tanzu cluster machinehealthcheck set tkc-01 --unhealthy-conditions "Ready:False:5m,Ready:Unknown:5m"`

If the node's Ready status reports as `False` or `Unknown` for five minutes, the node is considered unhealthy and is recreated.

## 4-34 Lab 6: Managing a Tanzu Kubernetes Cluster

Examine the default workloads running on the Tanzu Kubernetes cluster and scale the cluster to add an additional worker node:

1. Examine the Tanzu Kubernetes Cluster
2. Scale the Tanzu Kubernetes Cluster

## 4-35 Lab 7: Providing Access to Developers

Provide developer access to the Tanzu Kubernetes cluster:

1. Generate a kubeconfig File for the Cluster
2. Create Role Binding to Grant Developer Access

## 4-36 Review of Learner Objectives

- Describe the commands available for working with Tanzu Kubernetes clusters

# 4-37 Lesson 4: Tanzu Kubernetes Cluster Architecture

## 4-38 Learner Objectives

- Describe the VMs that make up a Tanzu Kubernetes cluster
- Describe the pods that run on a Tanzu Kubernetes cluster
- Describe the Tanzu Kubernetes Grid core add-ons that are installed on a cluster

# 4-39 Tanzu Kubernetes Cluster VMs

Depending on the plan used, the following VMs are created:

- One or more control plane VMs, with names similar to `tkc-01-control-plane-nj4z6`
- One or more worker node VMs, with names similar to `tkc-01-md-0-6ff9f5cffb-jhcrh`

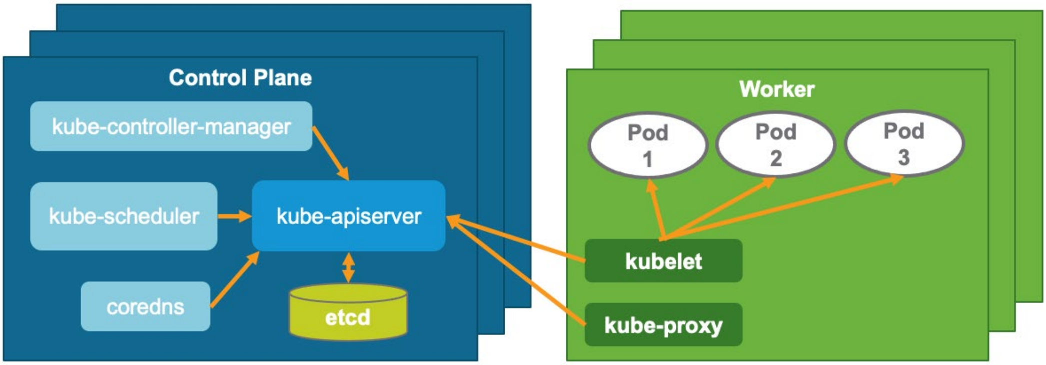
# 4-40 Upstream Kubernetes Components (1)

The following components are standard upstream Kubernetes components.

| Component | Description |
|---|---|
| etcd | Consistent and highly available key value store used as Kubernetes' backing store for all cluster data. |
| kube-apiserver | A component of the Kubernetes control plane that exposes the Kubernetes API. |
| kube-controller-manager | A daemon that watches the state of the cluster through the kube-apiserver and makes changes to move the current state toward the desired state. |
| kube-scheduler | Watches for newly created pods with no assigned node and selects a node for them to run on. |
| kubelet | Watches for requests to run pods. Starts the pods on the worker node. |
| kube-proxy | Maintains network rules which allow network communication to your pods from network sessions inside or outside of your cluster. |
| coredns | An extensible DNS server that serves as the Kubernetes cluster DNS. |

# 4-41 Upstream Kubernetes Components (2)

The following components are standard upstream Kubernetes components.



# 4-42 Kubernetes API High-Availability

`kube-vip` provides a high-availability load balancing solution for the Kubernetes control plane.

| Component | Description |
|---|---|
| kube-vip | The kube-vip pod runs on each control plane node. One node is elected as the leader and advertises the virtual IP address using Address Resolution Protocol (ARP).<br>If the node fails or becomes unhealthy, leader election happens again and another control plane node advertises the virtual IP address using ARP. |

## 4-43 Authentication Core Add-Ons

Pinniped is used for cluster authentication.

| Component | Description |
|---|---|
| pinniped-supervisor | The Pinniped supervisor runs on the management cluster. It is an OIDC server which allows users to authenticate with an external identity provider (IDP) and then issues an ID token which a user passes to the workload cluster. |
| pinniped-concierge | The Pinniped concierge runs on management and workload clusters. It exchanges the ID tokens for certificates which are used to authenticate Kubernetes API requests to the cluster. |
| dex | Dex runs on the management cluster. It is used by Pinniped to perform LDAP authentication. |

# 4-44 Networking Core Add-Ons

Tanzu Kubernetes Grid supports either Antrea or Calico as the container network interface (CNI) for in-cluster networking.

| Component | Description |
|---|---|
| antrea-controller | The Antrea controller watches NetworkPolicy, Pod, and Namespace resources from the Kubernetes API, computes NetworkPolicies, and distributes the computed policies to all antrea-agent pods. |
| antrea-agent | The Antrea agent manages the Open vSwitch (OVS) bridge and Pod interfaces, and implements Pod networking with OVS on every Kubernetes node. |
| calico-kube-controllers | Calico controllers watch the Kubernetes API for Namespace, Pod, NetworkPolicy and other resources. When changes are detected, the state is passed to Calico-node Pods. |
| calico-node | The Calico-node Pod is deployed on every Node and provides the overlay networking and routing between pods on different nodes. It watches the calico-kube-controllers for state changes and updates Pod networking, as needed. |

# 4-45 vSphere Core Add-Ons

The following vSphere components are deployed on Tanzu Kubernetes Grid clusters running on vSphere.

| Component | Description |
|---|---|
| vsphere-cloud-controller-manager | The vSphere cloud provider interface (CPI) implements Kubernetes functionality specific to vSphere. The vSphere CPI connects to vCenter Server and maps information about infrastructure, such as VMs, disks, and so on, back to the Kubernetes API. |
| vsphere-csi-controller | The vSphere CSI controller watches Kubernetes events related to PVC or PV objects, such as creation or deletion. It invokes the cloud-native storage (CNS) component on vCenter Server for volume operations such as as create, delete, update, attach, and detach. PVC or PV object metadata is synced with the vSphere CNS database. |
| vsphere-csi-node | The vSphere CSI node pod interacts with the local kubelet on the worker node. It performs operations related to pod volume access, such as format, mount, and unmount. |

# 4-46 Metrics Core Add-Ons

Tanzu Kubernetes Grid deploys a metrics server to management and workload clusters.

| Component | Description |
|---|---|
| metrics-server | Metrics Server collects resource metrics from kubelets and exposes them in the Kubernetes API for use by Horizontal Pod Autoscaler and Vertical Pod Autoscaler. |

## 4-47 Review of Learner Objectives

- Describe the VMs that make up a Tanzu Kubernetes cluster
- Describe the pods that run on a Tanzu Kubernetes cluster
- Describe the Tanzu Kubernetes Grid core add-ons that are installed on a cluster

## 4-48 Key Points

- Cluster API provides the main functionality of Tanzu Kubernetes Grid.
- The Tanzu Kubernetes Grid CLI provides options for creating Tanzu Kubernetes clusters of various sizes.
- Tanzu Kubernetes clusters run upstream versions of Kubernetes.

# Module 5

# Configuring and Managing Tanzu Kubernetes Grid Instances

## 5-2 Importance

A standard Kubernetes installation lacks some of the functionality that is required for a production-ready cluster. Tanzu Kubernetes Grid extensions provide the required logging, ingress, service discovery, and monitoring functionality to make a Tanzu Kubernetes cluster production-ready.

## 5-3 Module Lessons

1. Tanzu Kubernetes Grid Extensions
2. Image Registry
3. Logging
4. Ingress
5. Service Discovery
6. Cluster Monitoring

# 5-4 **Lesson 1: Tanzu Kubernetes Grid Extensions**

# 5-5 Learner Objectives

- Describe the Tanzu Kubernetes Grid extensions

# 5-6 About the Tanzu Kubernetes Grid Extensions Bundle

Tanzu Kubernetes Grid extensions bundle includes binaries for tools that help you to provide in-cluster and shared services to your Tanzu Kubernetes Grid instance.

VMware builds all the provided binaries and container images.

# 5-7 Extensions Included in the Bundle

The Tanzu Kubernetes Grid extensions bundle includes the following extensions.

| Function | Extension |
|---|---|
| Container Registry | Harbor |
| Logging | Fluent Bit |
| Ingress | Contour |
| Service Discovery | External DNS |
| Monitoring | Prometheus |
|  | Grafana |

# 5-8 Container Images

VMware builds all the container images used by Tanzu Kubernetes Grid and the Tanzu Kubernetes Grid extensions.

All container images are hosted on projects.registry.vmware.com, an instance of the Harbor image registry that is managed by VMware.

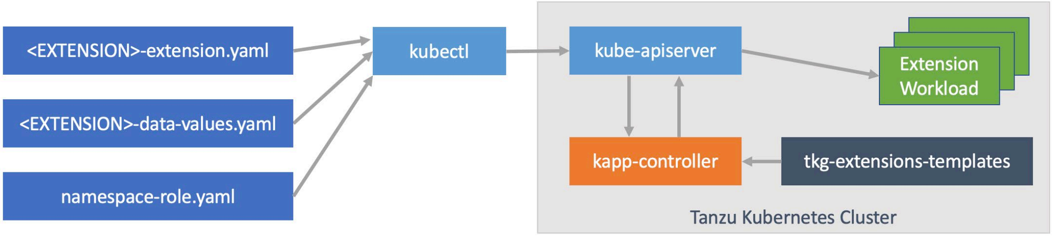
For Internet-restricted environments, container images can be copied to an accessible image registry. The extension manifest files must be updated with the new image registry path.

# 5-9 Deploying Extensions with kapp-controller

When extensions are deployed:

- The namespace, configuration, and extension YAML files for the specific extension are applied.
- kapp-controller combines the configuration with the YTT templates from the tkg-extensions-templates and generates the Deployment, ConfigMaps, Secrets, and any other resource definitions required for the extension.
- The extension workload is started by using standard Kubernetes processes.

The kapp CLI is used to inspect the state of workloads that are deployed by kapp-controller.



# 5-10 Configuring Extensions with Kubernetes Secrets

Extensions are configured by creating a Kubernetes Secret containing extension-specific configuration data.

The configuration file for each extension has the name `<extension>-data-values.yaml`. and is used to create a Secret with the name `<extension>-data-values`.

When kapp-controller deploys an extension, it reads the Secret and deploys the extension with the specified configuration.

Changes to the Secret resource cause kapp-controller to reconcile the changes in the deployed resources.

# 5-11 About cert-manager

cert-manager is a native Kubernetes certificate management controller that:

- Adds certificates and certificate issuers as resource types in Kubernetes clusters
- Simplifies the process of obtaining, renewing and using certificates
- Can generate certificates internally or connect to external services, such as Let’s Encrypt, to request certificates

While not considered a Tanzu Kubernetes Grid extension, cert-manager is provided in the extensions bundle because Contour, Grafana, Prometheus, and Harbor depend on it to provide certificates.

## 5-12 Deleting Extensions

For troubleshooting or if no longer required, extensions can be uninstalled by running the following commands:

- `kubectl delete -f <EXTENSION>-extension.yaml`
- `kubectl delete -f namespace-role.yaml`

For more information about deleting Tanzu Kubernetes Grid extensions, see *Delete Tanzu Kubernetes Grid Extensions* at https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-extensions-delete-extensions.html.

# 5-13 Lab 8: Unzipping the Tanzu Kubernetes Grid Extensions Bundle

Unzip the Tanzu Kubernetes Grid extensions files in preparation for deploying the extensions:

1. Unzip the Tanzu Kubernetes Grid Extensions
2. Change the Image Registry Location

# 5-14 Lab 9: Deploying cert-manager

Deploy the cert-manager component:

1. Deploy cert-manager

# 5-15 Review of Learner Objectives

- Describe the Tanzu Kubernetes Grid extensions

## 5-16 **Lesson 2: Image Registry**

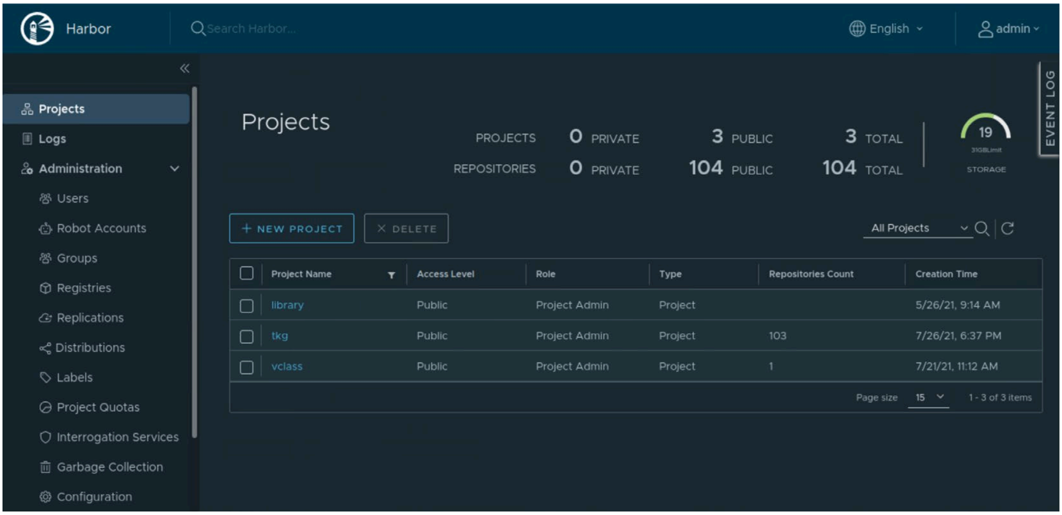
## 5-17 Learner Objectives

- Describe the Harbor Image Registry

# 5-18 About Harbor

Harbor is a container image registry that supports:

- Role-based access control (RBAC)
- Identity integration with LDAP or OIDC
- Scanning images for vulnerabilities
- Signed images
- Image replication between registries



Projects

PROJECTS 0 PRIVATE 3 PUBLIC 3 TOTAL

REPOSITORIES 0 PRIVATE 104 PUBLIC 104 TOTAL

| □ | Project Name | Access Level | Role | Type | Repositories Count | Creation Time |
|---|---|---|---|---|---|---|
| □ | library | Public | Project Admin | Project |  | 5/26/21, 9:14 AM |
| □ | tkg | Public | Project Admin | Project | 103 | 7/26/21, 6:37 PM |
| □ | vclass | Public | Project Admin | Project | 1 | 7/21/21, 11:12 AM |

Page size 15 1 - 3 of 3 items

# 5-19 Harbor Components

| Component | Description |
|---|---|
| Core | Acts as a front-end API for various functionality. |
| Portal | The Harbor UI. |
| Registry | Provides the Docker registry functionality. |
| Database | A PostgreSQL database for storing configuration and metadata. |
| Redis | Used by various Harbor services for caching to improve performance. |
| Jobservice | Running background jobs such as image replication. |
| Trivy | Vulnerability scanner for container images stored in Harbor. |
| Clair | Vulnerability scanner for container images stored in Harbor. |
| Notary | Provides image signing functionality. |

# 5-20 Vulnerability Scanning

Trivy provides the following vulnerability scanning functionality in Harbor:

- Scans container images stored in Harbor
- Scans OS packages in container images, such as Photon OS, Ubuntu, Debian, Red Hat, and others
- Scans language-specific packages in container images, such as Java, Python, Go, and others

Trivy does not scan running containers.

For more information about the scans performed by Trivy and the vulnerability databases used by Trivy, see *aquasecurity/trivy* on the GitHub website at https://github.com.

# 5-21 Harbor Deployment Options

Harbor is deployed on a shared-services cluster so that all workload clusters deployed in the Tanzu Kubernetes Grid instance can access it.

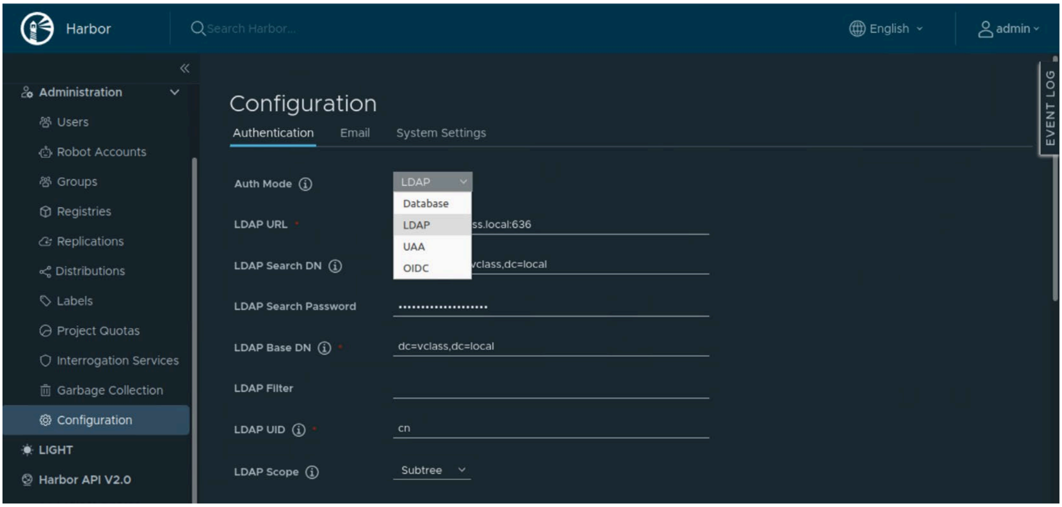
Common Harbor configuration options are listed.

| Key | Value |
|---|---|
| hostname | The Harbor host name. |
| tls.crt<br>tls.key<br>ca.crt | Provide custom TLS certificates. |
| harborAdminPassword | Sets the password for the admin user. |
| persistence.persistentVolumeClaim.registry.size | The size of the persistent volume used to store container images. |

If Contour and External DNS are configured, the Harbor host name will be registered with DNS and externally accessible.

# 5-22 Harbor Authentication

Select **Configuration** > **Authentication** to configure LDAP and OIDC authentication.



Configuration

Authentication | Email | System Settings

Auth Mode: LDAP (Database, LDAP, UAA, OIDC)

LDAP URL *: ss.local:636

LDAP Search DN: vclass,dc=local

LDAP Search Password: ••••••••••••••••••••

LDAP Base DN *: dc=vclass,dc=local

LDAP Filter:

LDAP UID *: cn

LDAP Scope: Subtree

# 5-23 Review of Learner Objectives

- Describe the Harbor Image Registry

# 5-24 **Lesson 3: Logging**

## 5-25 Learner Objectives

- Describe Fluent Bit
- Detail the logs that Fluent Bit collects
- Describe basic Fluent Bit configuration

# 5-26 About Fluent Bit

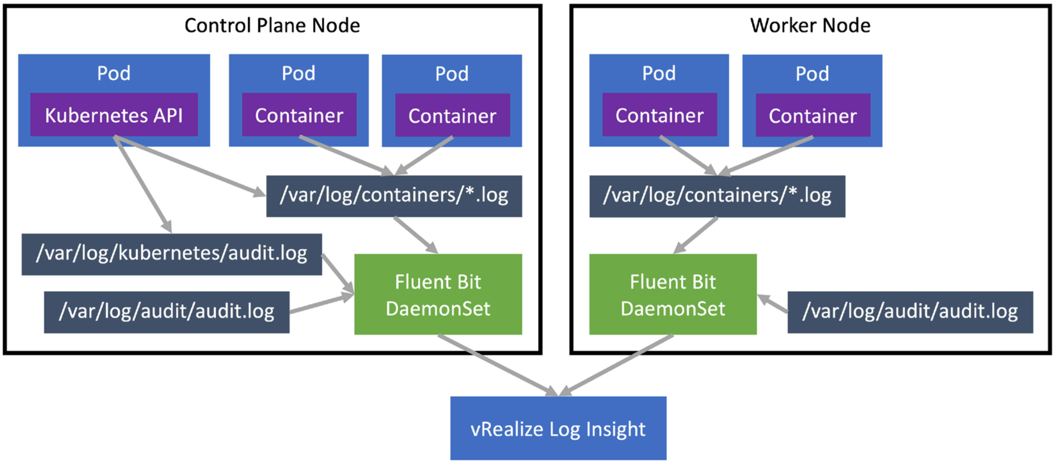
Fluent Bit is a lightweight logging framework that:

- Processes Kubernetes containers logs
- Enriches logs with Kubernetes metadata
- Outputs logs to services such as:
  - vRealize Log Insight
  - Syslog
  - Elasticsearch
  - Splunk
  - HTTP endpoint

# 5-27 How Fluent Bit Works

Fluent Bit is deployed as a DaemonSet and collects the following logs.

| Logs | Description |
|---|---|
| /var/log/containers/*.log | Container logs from all nodes. |
| /var/log/kubernetes/audit.log | Kubernetes API server audit logs on control plane nodes. |
| /var/log/audit/audit.log | OS audit logs on all nodes. |



# 5-28 Metadata Added By Fluent Bit

Fluent Bit adds the following metadata to logs:

- Pod ID
- Pod name
- Container ID
- Container name
- Labels
- Annotations
- Cluster name
- Instance name

# 5-29 Fluent Bit Syslog Configuration

The following parameters are required when configuring output to Syslog.

| Paramater | Description |
|---|---|
| host | The syslog host name. |
| port | The Syslog port. |
| mode | Sends logs using TCP or UDP. |
| format | Typically set to rfc5424. |
| instance_name | The instance name is arbitrary but using the management cluster name is practical. |
| cluster_name | The cluster name is arbitrary but should be the workload cluster name. |

## 5-30 Lab 10: Deploying Fluent Bit

Configure and deploy Fluent Bit to forward cluster logs to vRealize Log Insight:

1. Configure Fluent Bit
2. Deploy Fluent Bit
3. Access vRealize Log Insight to View Logs

## 5-31 Review of Learner Objectives

- Describe Fluent Bit
- Detail the logs that Fluent Bit collects
- Describe basic Fluent Bit configuration

# 5-32 **Lesson 4: Ingress**

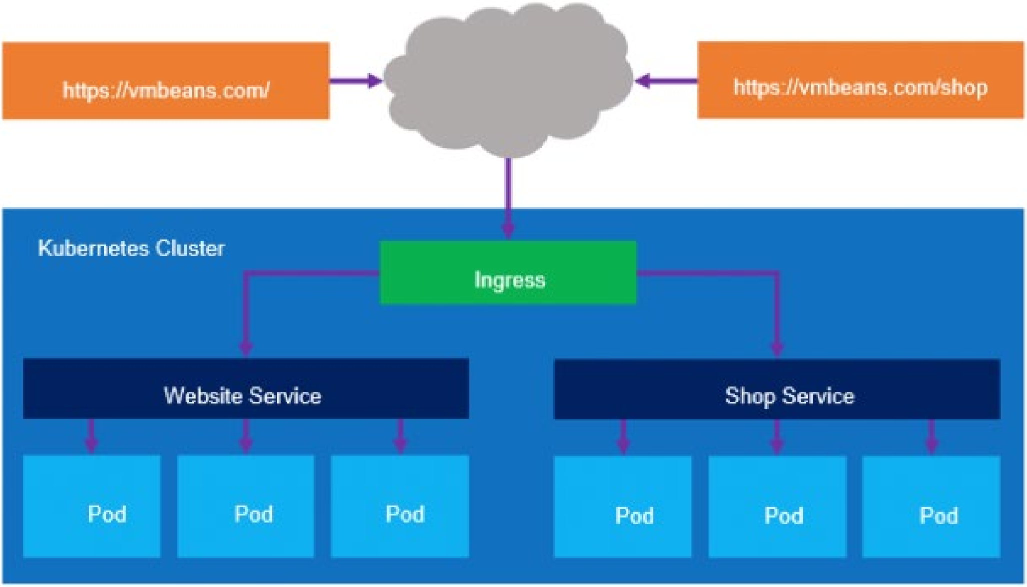
# 5-33 Learner Objectives

- Describe the Contour ingress controller
- Explain how to install Contour on a Tanzu Kubernetes cluster

# 5-34 About Ingress

Ingress in Kubernetes:

- Is a method for routing external traffic to pods
- Operates on layer 7 traffic
- Routes requests based on HTTP headers
- Can be implemented by various ingress controllers:
  - Contour
  - NSX-T
  - NSX Advanced Load Balancer
  - NGINX
  - Amazon ALB



# 5-35 About Contour

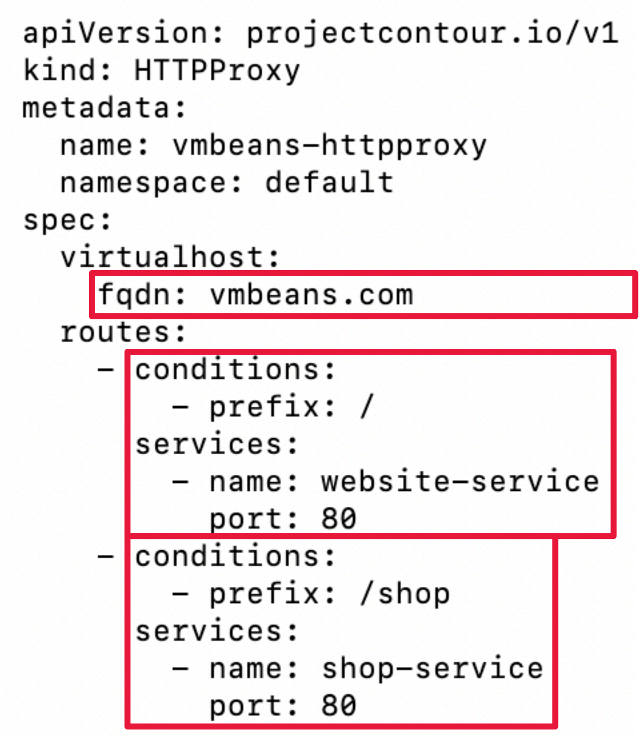
Contour:

- Is an open-source Kubernetes ingress controller
- Is used as the ingress controller for Tanzu Kubernetes Grid
- Supports Ingress and HTTPProxy resources
- Consists of Contour controller deployment and Envoy proxy DaemonSet
- Uses Envoy to perform traffic routing
- Supports dynamic configuration updates

# 5-36 About HTTPProxy Resources

HTTPProxy resources:

- Match requests based on the configured virtualhost FQDN field
- Route requests based on one or more routes that match the URL path
- Route requests to the defined service and port
- Support conditional routing based on URL prefixes and HTTP headers
- Support inclusion of one HTTPProxy resource within another to create tree-like routing structures

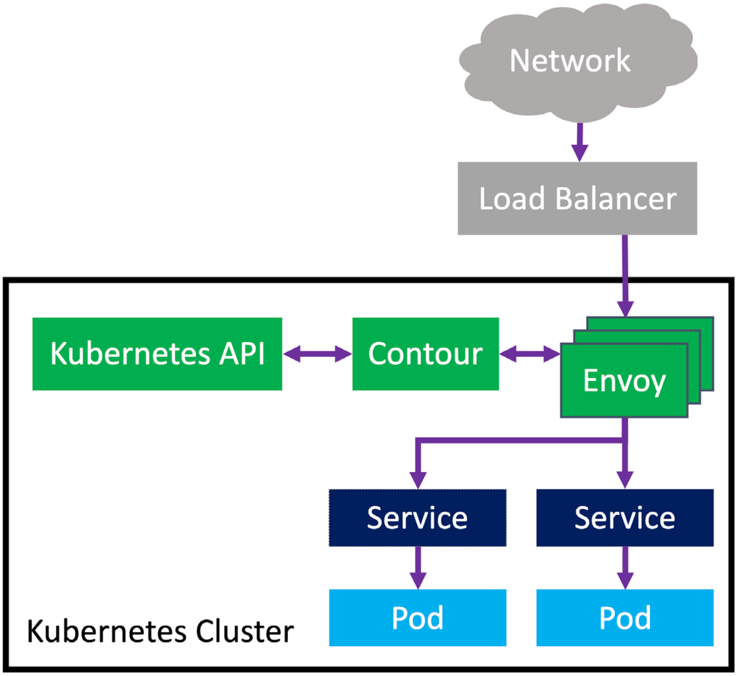


```
apiVersion: projectcontour.io/v1
kind: HTTPProxy
metadata:
  name: vmbeans-httpproxy
  namespace: default
spec:
  virtualhost:
    fqdn: vmbeans.com
  routes:
    - conditions:
      - prefix: /
      services:
        - name: website-service
          port: 80
    - conditions:
      - prefix: /shop
      services:
        - name: shop-service
          port: 80
```

# 5-37 How Contour Works

A Contour deployment dynamically configures an Envoy proxy DaemonSet by using the following steps:

1. Contour watches the Kubernetes API for ingress or HTTPProxy resources.
2. Contour generates a cache of ingress and HTTPProxy resources.
3. Envoy periodically polls Contour for configuration changes.
4. Envoy dynamically updates its routing configuration.
5. Requests are received from a load balancer or directly from the external network and routed to the correct pods.



# 5-38 Contour Configuration Options

The following manifests are provided in the `extensions/ingress/contour/vsphere` directory when deploying Contour on vSphere.

| Contour Configuration | Description |
|---|---|
| `contour-data-values-lb.yaml.example` | Deploys Contour and a load balancer service that routes traffic to Envoy. |
| `contour-data-values.yaml.example` | Deploys Contour only and manages routing traffic to Envoy separately. |

The default Contour configuration is suitable for most deployments.

More advanced options, such as `requestTimeout`, `connectionIdleTimeout` and `minimumProtocolVersion` (TLS), can be adjusted if required.

See the Tanzu Kubernetes Grid documentation for a full list of configuration options.

# 5-39 Lab 11: Deploying Contour

Configure and deploy Contour to enable HTTP routing to application workloads:

1. Configure Contour
2. Deploy Contour

# 5-40 Review of Learner Objectives

- Describe the Contour ingress controller
- Explain how to install Contour on a Tanzu Kubernetes cluster

# 5-41 **Lesson 5: Service Discovery**

## 5-42 Learner Objectives

- Describe Service Discovery
- Describe External DNS
- Detail the configuration options for BIND servers

# 5-43 About Service Discovery

Kubernetes provides a declarative API for deploying applications, services, ingress, and load balancer resources.

Services, ingress, and load balancer resources provide methods to route external IP network traffic into a cluster and to an application. However, IP addresses can change as workloads are created and destroyed.

Service discovery solves this problem by allowing applications to be registered in a service registry.

The service registry can be queried by applications or end users that need to access a specific application.

# 5-44 About External DNS

External DNS is a service discovery solution that:

- Uses DNS as a service registry
- Uses FQDNs defined in ingress, HTTPproxy, and load balancer resources
- Creates DNS entries for applications in the configured DNS server
- Allows users and applications outside a cluster to use DNS to access workloads inside a cluster

Internal DNS within a cluster still functions normally.

# 5-45 DNS Servers Supported by External DNS

External DNS supports the following DNS servers:

- RFC2136 (BIND) server
- Microsoft DNS using RFC2136 (Insecure) or RFC3645 (Secure)
- AWS (Route53)
- Microsoft Azure

For a full list of supported DNS servers, see *kubernetes-sigs/external-dns* on the GitHub website at https://github.com.

# 5-46 How External DNS Works

External DNS works by:

- Watching for host names in Kubernetes API resources:
    - Ingress and HTTPProxy resources are watched for the host field on each of the defined `rules`.
    - Load balancer resources are watched for the `external-dns.alpha.kubernetes.io/hostname` annotation.
- Creating a DNS entry in the configured DNS server that points to the ingress or load balancer IP address
- Deleting the DNS entry when the ingress, HTTPproxy, or load balancer resources are deleted

# 5-47 External DNS Configuration for BIND Servers

The example configuration file is `external-dns-data-values-rfc2136.yaml.example`. and it watches load balancer resources for External DNS annotations.

To have External DNS watch for Contour ingress and HTTPProxy resources, use the `external-dns-data-values-rfc2136-with-contour.yaml.example` file.

The following parameters are configured when using a BIND server.

| Key | Description |
|---|---|
| --rfc2136-host<br>--rfc2136-port | The IP address and port for the DNS server. |
| --rfc2136-zone | Which DNS zone to update. |
| --rfc2136-tsig-secret<br>--rfc2136-tsig-keyname | Authentication details for performing DNS updates. |
| --source | Which resources are watched, for example, Service, Ingress, HTTPproxy. |
| --domain-filter | Only creates DNS entries for specific domains or subdomains. |

## 5-48 Lab 12: Deploying External DNS

Configure and deploy External DNS to enable dynamic DNS updates for ingress and load balancer resources:

1. Retrieve the Secret Key for DNS Updates
2. Configure External DNS
3. Deploy External DNS

## 5-49 Review of Learner Objectives

- Describe Service Discovery
- Describe External DNS
- Detail the configuration options for BIND servers

# 5-50 Lesson 6: Cluster Monitoring

## 5-51 Learner Objectives

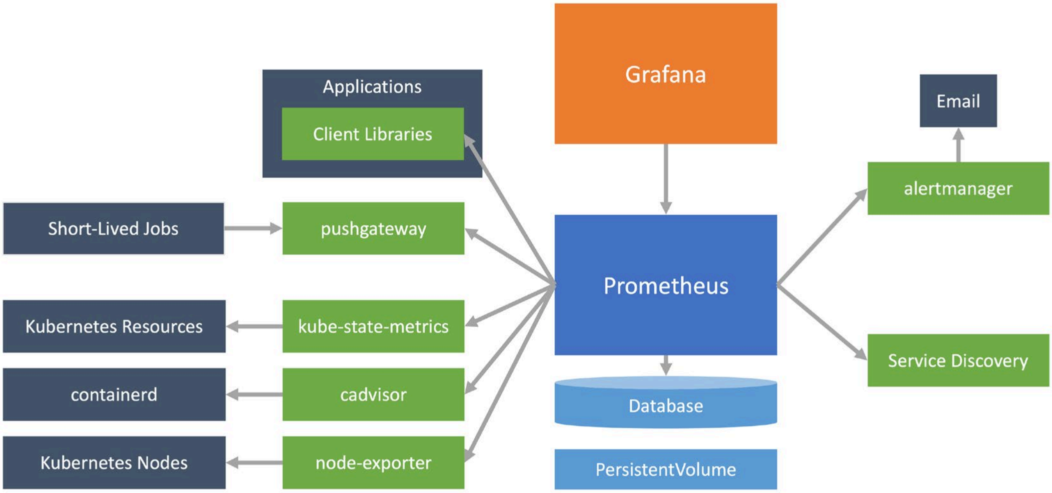
- Describe Prometheus
- Describe Grafana

# 5-52 Monitoring in Tanzu Kubernetes Grid

Tanzu Kubernetes Grid provides cluster monitoring services using the following open-source projects:

- Grafana: Visualization and analytics software. It enables you to query, visualize, alert on, and explore your metrics wherever they are stored.
- Prometheus: A systems monitoring and alerting toolkit. It collects metrics from clusters and applications at specified intervals and triggers alerts if certain conditions arise.

# 5-53 Grafana and Prometheus Architecture



# 5-54 Prometheus Components

Prometheus consists of the following components.

| Name | Description |
|---|---|
| server | Performs the job of scraping and storing metrics in the Prometheus database. |
| alertmanager | Provides alerting to external services, such as email. |
| cAdvisor | Provides container-level metrics. |
| Node Exporter | Provides OS-level metrics for control plane and worker nodes. |
| kube-state-metrics | Provides metrics for Kubernetes API resources, for example, pod status, pod resource limits, and many more. |
| Client Libraries | Applications can be instrumented with the Prometheus client libraries, providing Prometheus the ability to capture application-level metrics. |
| Pushgateway | Provides an endpoint for one-time jobs, such as batch jobs, to send metrics. |

# 5-55 Prometheus Configuration Options

The default Prometheus configuration values provide a working Prometheus deployment.

Some example options that can be modified in the `prometheus-data-values.yaml` file are listed.

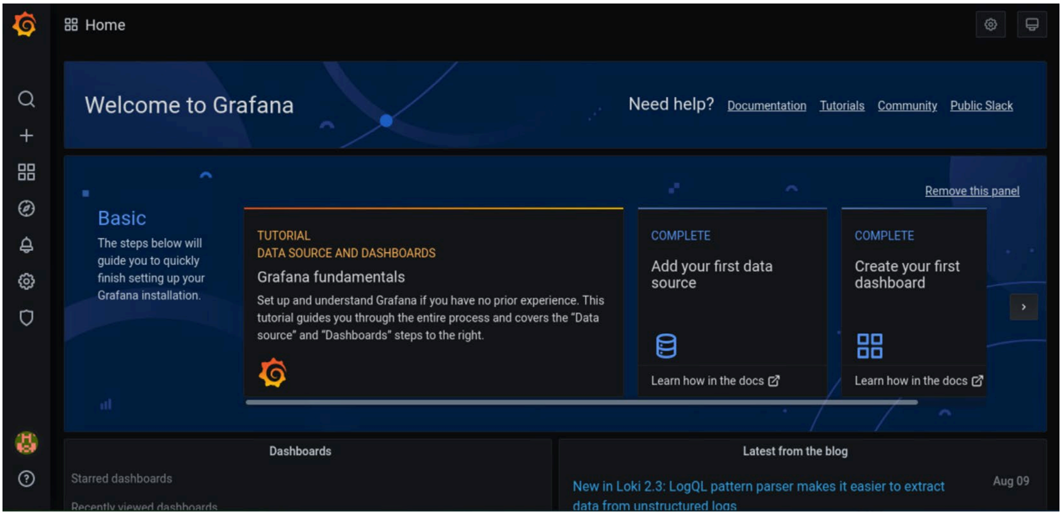
| Option | Description |
|---|---|
| monitoring.prometheus_server.config.alerting_rules_yaml | Configures alerting. |
| monitoring.alertmanager.config.email_receiver | Emails server configuration for alerting. |
| monitoring.ingress.enabled | Enables ingress access to the Prometheus dashboard. |

For more information about configuration options, see https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-extensions-prometheus.html.

# 5-56 About Grafana

Grafana is an open-source monitoring and observability platform that:

- Can connect to and pull data from many data sources
- Provides dynamic dashboards, graphs and visualization functionality for Prometheus



Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source
Learn how in the docs

COMPLETE
Create your first dashboard
Learn how in the docs

Dashboards
Starred dashboards
Recently viewed dashboards

Latest from the blog
New in Loki 2.3: LogQL pattern parser makes it easier to extract data from unstructured logs
Aug 09

# 5-57 Grafana Configuration Options

The following values must be configured in the `grafana-data-values.yaml` file before deploying Grafana.

| Options | Description |
|---|---|
| monitoring.grafana.secret.admin_password | The password for the admin account. |
| monitoring.grafana.ingress.virtual_host_fqdn | The host name to use for ingress. |

For more information about configuration options, see "Deploy Grafana on Tanzu Kubernetes Clusters" at https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-extensions-grafana.html.

## 5-58 Lab 13: Deploying Prometheus and Grafana

Configure and deploy Prometheus and Grafana to collect and visualize cluster metrics:

1. Deploy Prometheus
2. Configure Grafana
3. Deploy Grafana
4. Access the Grafana Web Interface

## 5-59 Lab 14: Deploying a Workload

Deploy a workload as a developer and inspect the workload with various tools:

1. Access the Cluster as a Developer
2. Deploy a Workload
3. Inspect the NSX Advanced Load Balancer UI
4. Inspect the Logs in vRealize Log Insight

## 5-60 Review of Learner Objectives

- Describe Prometheus
- Describe Grafana

# 5-61 Key Points

- The Tanzu Kubernetes Grid extensions provide functionality that is required for a production cluster.
- Harbor provides an image registry.
- Fluent Bit provides logging functionality.
- Contour provides ingress functionality.
- External DNS provides service discovery functionality.
- Prometheus and Grafana provide monitoring functionality.

# Module 6

# Troubleshooting Tanzu Kubernetes Grid

## 6-2 Importance

Understanding how to identify and correct issues is vital to the efficient operation of Tanzu Kubernetes Grid.

## 6-3 Module Lessons

1. Tanzu Kubernetes Grid Logs
2. Crash Diagnostics
3. Troubleshooting Commands

## 6-4 **Lesson 1: Tanzu Kubernetes Grid Logs**

## 6-5 Learner Objectives

- Describe the various Tanzu Kubernetes Grid logs
- Identify the location of Tanzu Kubernetes Grid logs

# 6-6 Monitoring Tanzu Kubernetes Cluster Deployments in Cluster API Logs

If a cluster deployment is failing, viewing the logs for the Cluster API pods can indicate what is failing.

During a management cluster deployment, these logs are accessible on the bootstrap cluster.

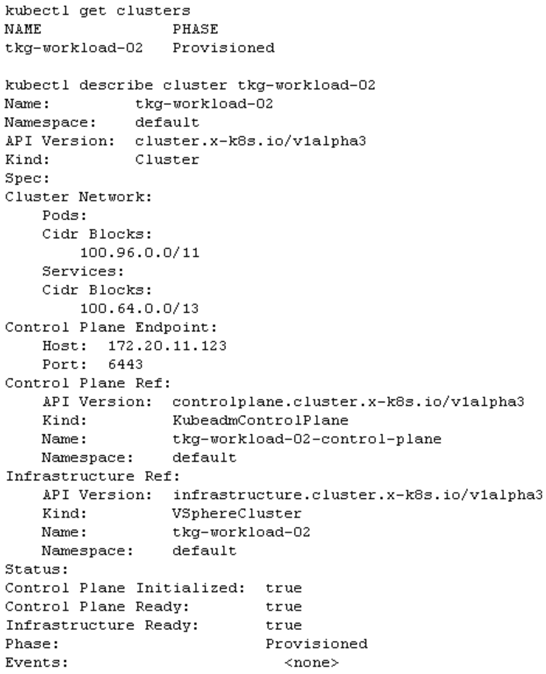
During a workload cluster deployment, these logs are accessible on the management cluster.

| Namespace | Pod |
|---|---|
| capi-kubeadm-bootstrap-system | capi-kubeadm-bootstrap-controller-manager |
| capi-kubeadm-control-plane-system | capi-kubeadm-control-plane-controller-manager |
| capi-system | capi-controller-manager |
| capv-system | capv-controller-manager |

# 6-7 Viewing Cluster API Resources

Each cluster that is deployed by Tanzu Kubernetes Grid has corresponding Cluster API resources which can be viewed using the `kubectl describe` command.

```
kubectl describe
  clusters
  kubeadmconfigs
  kubeadmcontrolplanes
  machinedeployments
  machinesets
  machines
  vsphereclusters
  haproxyloadbalancers
  vspheremachines
  vspherevms
```



```
kubectl get clusters
NAME              PHASE
tkg-workload-02   Provisioned

kubectl describe cluster tkg-workload-02
Name:         tkg-workload-02
Namespace:    default
API Version:  cluster.x-k8s.io/v1alpha3
Kind:         Cluster
Spec:
Cluster Network:
    Pods:
    Cidr Blocks:
        100.96.0.0/11
    Services:
    Cidr Blocks:
        100.64.0.0/13
Control Plane Endpoint:
    Host:  172.20.11.123
    Port:  6443
Control Plane Ref:
    API Version:  controlplane.cluster.x-k8s.io/v1alpha3
    Kind:         KubeadmControlPlane
    Name:         tkg-workload-02-control-plane
    Namespace:    default
Infrastructure Ref:
    API Version:  infrastructure.cluster.x-k8s.io/v1alpha3
    Kind:         VSphereCluster
    Name:         tkg-workload-02
    Namespace:    default
Status:
Control Plane Initialized:  true
Control Plane Ready:        true
Infrastructure Ready:       true
Phase:                      Provisioned
Events:                       <none>
```

## 6-8 About Kubernetes Audit Logging

Kubernetes audit logging:

- Is enabled by setting `ENABLE_AUDIT_LOGGING=true` when deploying a cluster
- Logs metadata about all requests made to the Kubernetes API server
- Logs audit events to the `/var/log/kubernetes/audit.log` file on each control plane node
- Is included in the logs sent to the logging server by Fluent Bit

# 6-9 About System Audit Logging

System audit logging:

- Is enabled by default on Tanzu Kubernetes clusters
- Uses the Linux auditd system to track audit events
- Logs audit events to the /var/log/audit/audit.log file on each node
- Is included in the logs sent to the logging server by Fluent Bit

# 6-10 Review of Learner Objectives

- Describe the various Tanzu Kubernetes Grid logs
- Identify the location of Tanzu Kubernetes Grid logs

## 6-11 **Lesson 2: Crash Diagnostics**

## 6-12 Learner Objectives

- Describe the purpose of Crash Diagnostics

# 6-13 About Crash Diagnostics

Crash Diagnostics or Crashd:

- Is a command-line tool to help platform operators investigate and troubleshoot unhealthy or unresponsive Kubernetes clusters
- Is available for Linux and MacOS
- Runs scripts that collect Kubernetes API output, node logs, and node command-line output through SSH

# 6-14 About Diagnostic Scripts

The diagnostic scripts that Crashd uses are written in the Starlark, which is a dialect of the Python programming language.

Crashd exposes commonly used operations as Starlark functions which are used to build diagnostics scripts.
For example:

- Run kubectl and collect the specified resources:

  ```
  kube_capture(what="objects", kinds=["deployments",
  "services"], namespaces=["default"])
  ```

- Run a command through SSH on the specific nodes:

  ```
  capture(cmd="df -h /", resources=nodes)
  ```

For examples of Crashd diagnostics scripts, see *vmware-tanzu/crash-diagnostics* on the GitHub website at https://github.com.

# 6-15 About the Default Diagnostics Script

Crashd provides a default diagnostics script, covering common use cases, which:

- Accepts arguments to specify what to collect
- Collects bootstrap, management, and workload cluster information
- Outputs an archive file containing all requested information

# 6-16 Passing Arguments

Ways to pass arguments to the diagnostics script:

- By entering the parameters in a configuration file and passing the file name:

  ```
  --args-file <ARGS_FILENAME>
  ```

- By entering the parameters as a comma-separated list of key-value pairs:

  ```
  --args key1=value1,key2=value2
  ```

Arguments passed using `--args` take precedence over arguments passed using `--args-file`.

# 6-17 Default Diagnostic Script Arguments

The default diagnostics script accepts the following arguments.

| Argument | Description |
|---|---|
| target | Where to collect the logs from, either bootstrap, mgmt, or workload. |
| infra | What infrastructure the cluster is running on, either vsphere, aws, or azure. |
| workdir | Where to store collected logs. |
| ssh_user | The username to use for SSH. |
| ssh_pk_file | The private key to use for SSH. |
| mgmt_cluster_ns | The namespace in the management cluster where the management cluster's own Cluster API resources are created. |
| mgmt_cluster_config | The kubeconfig file for the management cluster. |
| workload_clusters | A list of comma-separated workload cluster names. |
| workload_cluster_ns | The namespace in the management cluster where the workload clusters Cluster API resources are created. |

# 6-18 Running Crash Diagnostics

You use the Crash Diagnostics CLI `run` command to run a diagnostics script.

Examples:

- Pass the arguments from the args file and run diagnostics.crsh.

  ```
  crashd run --args-file args diagnostics.crsh
  ```

- Same as the previous command, but override the target and workload_cluster parameters.

  ```
  crashd run --args-file args --args
  target=workload,workload_cluster=tkc-01 diagnostics.crsh
  ```

## 6-19 Lab 15: Configuring and Running Crash Diagnostics

Install and configure Crash Diagnostics and run a log collection on the Tanzu Kubernetes cluster:

1. Install the Crash Diagnostics CLI
2. Configure the Crash Diagnostics Arguments File
3. Run the Crash Diagnostics CLI

## 6-20 Review of Learner Objectives

- Describe the purpose of Crash Diagnostics

# 6-21 **Lesson 3: Troubleshooting Commands**

## 6-22 Learner Objectives

- Describe how to use SSH to connect to a Tanzu Kubernetes VM
- Detail the steps to troubleshoot a failed cluster deployment

# 6-23 Connecting to Cluster Nodes with SSH

You can use SSH to connect to individual nodes of management clusters or Tanzu Kubernetes clusters.

To do so, the SSH key pair that you created when you deployed the management cluster must be available on the machine on which you run the SSH command.

The SSH key that you entered in the installer is associated with the capv user account:

- `ssh capv@VM_IP_ADDRESS`
- `sudo -i`

Because the SSH key is present on the system on which you are running the SSH command, no password is required.

Run the `sudo -i` command to change to root and launch a Bash shell.

## 6-24 Failure of the Management Cluster Create Command

When a cluster fails to create and the `Waiting for cert-manager to be available` error message displays, perform the following steps:

1. Verify whether the cluster nodes are all in the ready state.

   ```
   kubectl get nodes
   ```

2. Verify if any pods are failing.

   ```
   kubectl get pods -A
   ```

3. For any pods that are failing, verify the events and logs.

   ```
   kubectl describe pod -n <POD_NAMESPACE> <POD_NAME>
   ```

   ```
   kubectl logs -n <POD_NAMESPACE> <POD_NAME>
   ```

4. Resolve any issues with configuration or connectivity.

5. Delete the management cluster.

   ```
   tanzu management-cluster delete
   ```

6. Run the create command again to ensure a clean deployment.

   ```
   tanzu management-cluster create
   ```

# 6-25 Recovering Management Cluster Credentials

If you lose the credentials for a management cluster, you can recover them.

To recover the credentials from the management cluster control plane node:

1. Run `tanzu management-cluster create` to recreate the .kube-tkg/config file.
2. Obtain the IP address of the management cluster control plane node from vSphere.
3. Use SSH to log in to the management cluster control plane node.

   ```
   ssh capv@node_IP_adress
   ```

4. Access the admin.conf file for the management cluster.

   ```
   sudo cat /etc/kubernetes/admin.conf
   ```

5. Copy the cluster name, the cluster user name, the cluster context, and the client certificate data into the .kube-tkg/config file.

You can lose the credentials for a management cluster, for example, by inadvertently deleting the `.kube-tkg/config` file on the system on which you run `tanzu` commands.

## 6-26 Lab 16: Performing Basic Troubleshooting

Perform basic troubleshooting:

1. Connect to a Node by Using SSH

## 6-27 Lab 17: (Optional) Building Custom Images

Build a custom image using Image Builder:

1. Unzip the Tanzu Kubernetes Grid Image Builder Files
2. Load the OVF Tool into the Image Builder Container Image
3. Prepare the Image Builder Configuration
4. Run Image Builder

## 6-28 Review of Learner Objectives

- Describe how to use SSH to connect to a Tanzu Kubernetes VM
- Detail the steps to troubleshoot a failed cluster deployment

# 6-29 Key Points

- Crash Diagnostics is used to collect logs from a Tanzu Kubernetes cluster.
- kubectl is used to query Cluster API logs and resources to view the status of deployed clusters.