Tanzu for Kubernetes Operations on VxRail

Building Modern Applications across Clouds

Table of contents

Executive Summary

In this era of modern app development, the complexity of multi-layer design and distributed architectures, drive the need for the operations model to evolve. Kubernetes has emerged as the de facto container orchestration platform and the number of containerized applications running in production continues to grow. As such, customers are increasingly interested in a consistent runtime across their deployments regardless of where they reside. They are also looking for consistent operations and management with fine grain visibility into their Kubernetes and application frameworks. They want their environments to be secure, easy to deploy, manage and upgrade, while managing the application sprawl of microservices

Cloud native application development leverages DevOps and CI/DC (Continuous Integration/Continuous Delivery) practices that streamline the delivery of production ready microservice applications. Cloud native application model suits many workloads, and an increasing number of companies are “born in the cloud” or migrating to the cloud. When companies build and operate applications using a cloud native architecture, they bring new ideas to market and respond to customer demands faster.

This reference architecture presents a design and implementation methodology to integrate on-premises private cloud powered by Tanzu for Kubernetes Operations, running on Dell VxRail with Tanzu for Kubernetes Operations on VMware Cloud on AWS. This integration is further extended to other Kubernetes cloud offerings, such as Amazon EKS, that bring all Kubernetes clusters under a common management domain using Tanzu Mission Control.

On-demand, multi-cloud connectivity and management can be a challenge technically, as well as costly for most customers. First, connections to the cloud providers must be established and maintained separately, which adds complexity to an already complex paradigm. Second, when applications are spread over smaller clusters across clouds to provide separate fault domains, management of these clusters on an ongoing basis is not trivial. In this reference architecture we present a solution using a third-party cloud connectivity provider such as Equinix which can manage connection enumeration to most major cloud providers simultaneously. Such a solution would be beneficial for customers connecting to multiple cloud providers, however, is not a requirement for this reference architecture. With centralized policy and cluster management, and fine grain insight into cluster operations, Tanzu Mission Control and Tanzu Observability provide a common management methodology across all cloud instances.

For end-to-end connectivity, cross-site load balancing and ingress, NSX Advanced Load Balancer provides a robust networking stack that can support global DNS services as Kubernetes deployment instances grow from on-premises private cloud to multi-cloud environment. This reference architecture uses this NSX Advanced Load Balancer GSLB (Global Server Load Balancing) capability to load balance application instances across clouds. AKO (NSX Advanced Load Balancer Kubernetes Operator) and AMKO (Avi Multi-Cluster Operator) provides ingress services across Kubernetes deployments.

For application security, NSX Advanced Load Balancer features an Intelligent Web Application Firewall (iWAF) that covers OWASP CRS protection, support for compliance regulations such as PCI DSS, HIPAA, and GDPR, and signature-based detection. It deploys positive security model and application learning to prevent web application attacks. Additionally, built-in analytics provide actionable insights on performance, end-user interactions and security events in a single dashboard (Avi App Insights) with end-to-end visibility.

Dell VxRail delivers a turnkey experience and is fully integrated, pre-configured, and pre-tested solution. Tanzu for Kubernetes Operations on VxRail is a future proof solution that simplifies transformation journey to modern applications for most customers. Whether its move from legacy to cloud native applications, repatriating cloud native applications to on-premises private cloud, or architecting distributed application on multi-cloud environment, Tanzu Kubernetes on VxRail is the all-encompassing solution.

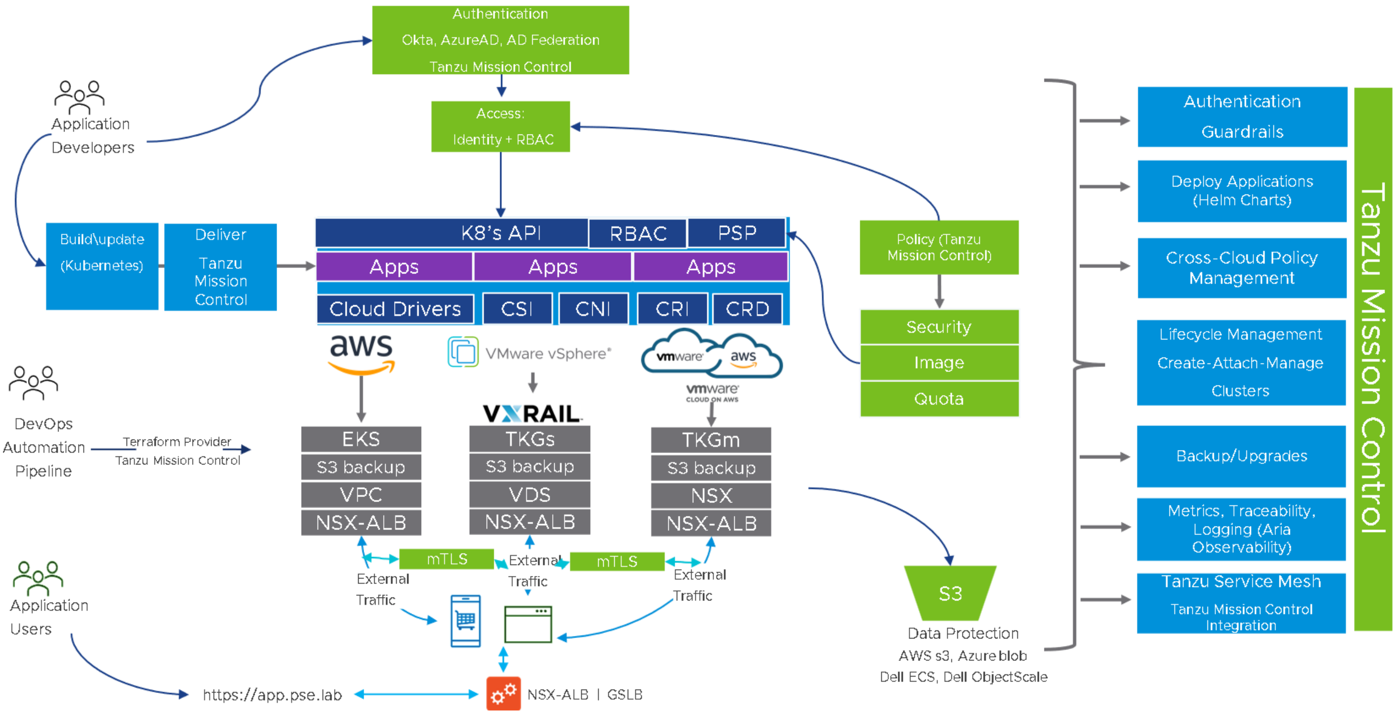
Audience

This white paper is intended for architects, engineers, consultants, and IT administrators who design, implement, and manage modern application environment on-premises or in the cloud. Readers with strong understanding of technologies such as VMware NSX Advanced Load Balancer, vSphere with Tanzu, VMware vSAN, and cloud native concepts will benefit from the content in this paper.

Architecture Overview

The solution is built on multi-cloud architecture, including On-premises private cloud, Amazon EKS and VMware Cloud on AWS, with the three site instances making up the global DNS namespace. Amazon EKS was included in the architecture to demonstrate public cloud management and integration capabilities of Tanzu portfolio of products. Another public cloud, such as Azure or Google (GCP) can also be integrated with relevant ease. NSX Advanced Load Balancer (formerly known as Avi) manages the global DNS zone. User queries for applications to the corporate DNS are directed to the appropriate site holding the application. Instances of the applications are installed on multiple sites for load balancing and high availability. Load balancing based on geo-location or priority, provides improved performance by directing the user to the nearest or highest priority site holding the desired application. In this architecture the primary site is the on-premises private-cloud infrastructure and software built on Tanzu Kubernetes Grid service. This primary site holds the Active directory domain infrastructure and corporate DNS services. This site also has the leader GSLB (Global Server Load Balancing) service instance. Avi multi-cluster Kubernetes operator (AMKO) is installed here, which manages and coordinates ingress and load balancing from Avi Kubernetes operators (AKO) from all sites. VMware Cloud on AWS (VMC) and Amazon EKS makeup the other two GSLB follower instances. These follower sites only require AKO (Avi Kubernetes operator) installation. Tanzu Service Mesh, a part of Tanzu for Kubernetes Operations provided end-to-end connectivity, security, and insights for microservices running across clouds making up the global application namespace.

Figure 1: Logical Architecture Overview



Tanzu Mission Control (Tanzu Mission Control) managed the complete lifecycle of Kubernetes clusters across sites including Amazon EKS clusters. Tanzu Mission Control also provides centralized policy management and developer self-service access to all three sites. Tanzu Observability now part of VMware Aria, provided fine grain insight and observability for VMware Tanzu and Amazon EKS clusters. Tanzu Mission Control also provided data-protection through Velero and Restic open-source software using S3 compatible storage, on-premises or in the cloud.

An intermediate site connecting the on-premises private cloud to multiple cloud providers is hosted at Equinix. On-demand connections to multiple cloud providers are enumerated from Equinix, per customer performance and cost requirements.

Note: An intermediate cloud services provider is not a requirement for this reference architecture. It is however an option for customers who want to take advantage of these services for their multi-cloud connectivity. Alternatively, customers can connect to their cloud providers individually via VPN or direct connections. For more information on how Equinix is configured to provide multi-cloud connectivity, please see Multi-Cloud Network Connectivity with Equinix Overview.

Key Components

This solution is built upon a solid foundation using Dell VxRail cluster made up of four V570 model HCI nodes. When configured with VMware vSAN and NSX Advanced Load Balancer, Dell VxRail provides an enterprise grade software defined datacenter architecture that is agile, easy to manage and secure. vSphere with Tanzu enhances these underlying qualities and delivers a developer-ready, modern application platform for upstream Kubernetes clusters. From a manageability perspective, Tanzu Mission Control and Tanzu Observability provides a solution that is future-proof and extensible from on-premises to the cloud. A description of the key components follows.

Dell VxRail

Whether accelerating data center modernization, deploying a hybrid cloud, or creating a developer-ready Kubernetes platform, VxRail delivers a turnkey experience that enables customers to continuously innovate. The only hyperconverged system jointly engineered by Dell Technologies and VMware, it is fully integrated, pre-configured, and pre-tested, automating lifecycle management and simplifying operations. Powered by VMware vSAN or VMware Cloud Foundation, VxRail transforms HCI networking and simplifies VMware cloud adoption, while meeting any HCI use case - including support for the most demanding workloads and applications. Learn more.

NSX Advanced Load Balancer

VMware NSX Advanced Load Balancer provides multi-cloud load balancing, web application firewall and application analytics across on-premises data centers and any cloud. The software-defined platform delivers applications consistently across bare metal servers, virtual machines, and containers to ensure a fast, scalable, and secure application experience. Learn more.

Tanzu Mission Control

VMware Tanzu Mission Control is a centralized management hub, with a robust policy engine, which simplifies multi-cloud and multi-cluster Kubernetes management. Whether you are new to Kubernetes, or quite experienced, Tanzu Mission Control helps platform operators reduce complexity, increase consistency, and offer a better developer experience. Learn more.

Tanzu Observability (Aria)

VMware Tanzu Observability by Wavefront now part of VMware Aria is an observability platform specifically designed for enterprises needing monitoring, observability, and analytics for their cloud-native applications and environments. DevOps, SRE and developer teams use Tanzu Observability to proactively alert on, rapidly troubleshoot and optimize performance of their modern applications running on the enterprise multi-cloud. Learn more.

Tanzu Kubernetes Grid

Tanzu Kubernetes Gid Standard has everything an enterprise needs to make best use of Kubernetes as part of its vSphere-based infrastructure. Kubernetes is embedded in the vSphere control plane, addressing the needs of both operators and developers. Operators can support virtual machines and containers side-by-side on a unified platform, group these elements into applications and simplify management. Developers gain self-service access to resources using Kubernetes APIs and speed development processes.

Tanzu Service Mesh

Tanzu Service Mesh provides advanced, end-to-end connectivity, security, and insights for modern applications—across application end-users, microservices, APIs, and data—enabling compliance with Service Level Objectives (SLOs) and data protection and privacy regulations. More information can be found at VMware Tanzu Service Mesh.

VMware Cloud on AWS

VMware Cloud on AWS is an integrated cloud offering jointly developed by Amazon Web Services (AWS) and VMware. You can deliver a highly scalable and secure service by migrating and extending your on-premises VMware vSphere-based environments to the AWS Cloud running on Amazon Elastic Compute Cloud (Amazon EC2). Learn more.

Amazon Elastic Kubernetes Service

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that you can use to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes.

Key Software Components

The following table shows key software component versions used in this reference architecture.

Table 1: Key software components

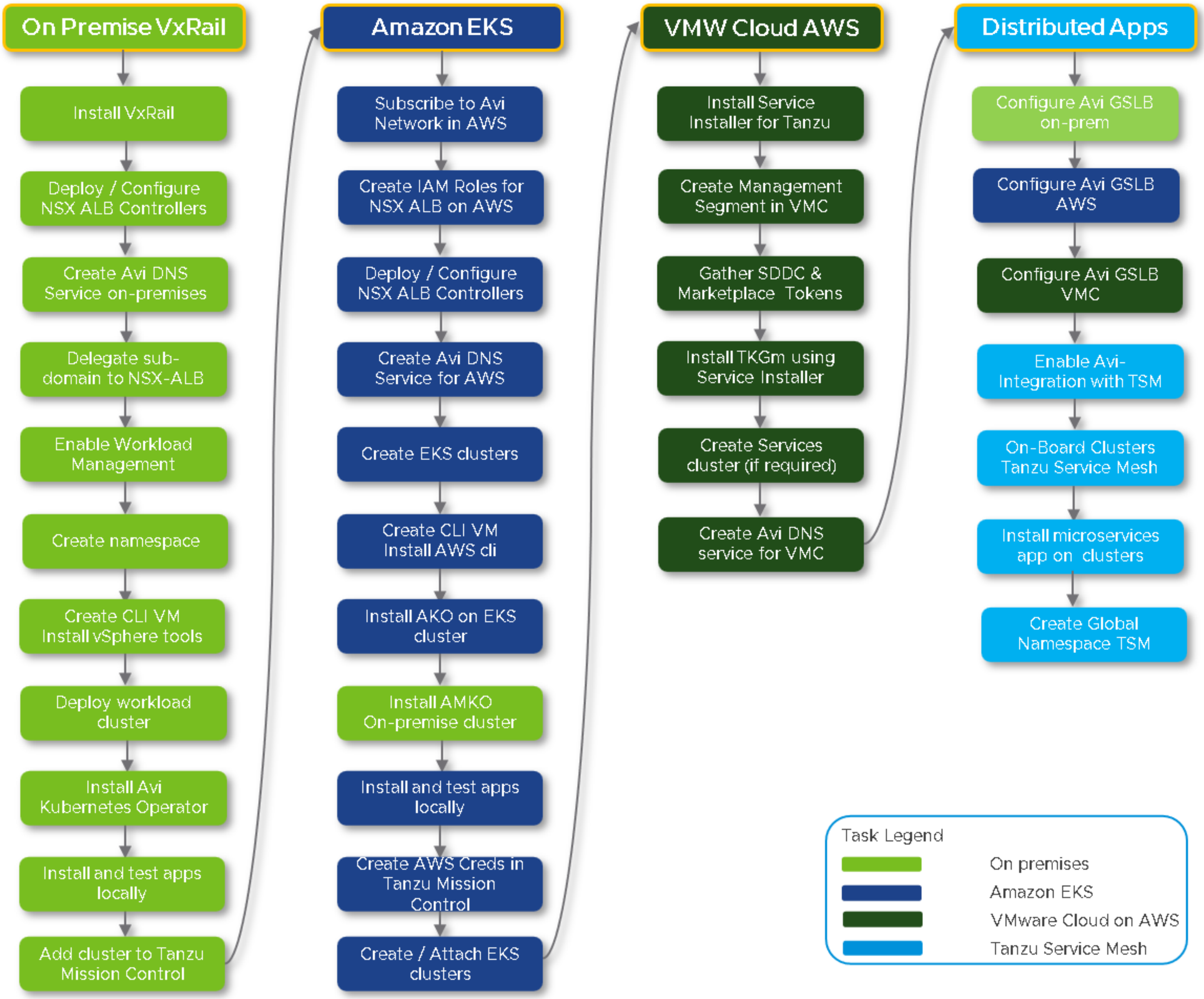
Software Specifications		
On-Premises Component	**Version**	**Notes**
– VxRail Manager	– 7.0.350-27409467	
– VMware ESXi	– VMware ESXi, 7.0.3, 19193900	4 x VxRail V570 nodes
– VMware vCenter Server	– vSphere Client version 7.0.3.00600	
– NSX Advanced Load Balancer	– 20.1.7 Enterprise	3 x Avi appliances for HA
– vSphere with Tanzu	– Tanzu Kubernetes Grid	K8s version v1.20.12+vmware.1
– Pod Networking (CNI)	– Antrea Advanced	1 Core
VMware Cloud on AWS	**Version**	**Notes**
– VMware ESXi	– VMware ESXi, 7.0.3, 19888012	3 x ESXi nodes
– VMware (SDDC) vCenter Server	– vSphere Client version 7.0.3.20000	
– NSX Advanced Load Balancer	– 21.1.4 Enterprise	No orchestrator mode
– Tanzu Kubernetes Grid	– Multi-Cloud (TKGm) 1.5.3->1.6.0 *	K8s version v1.22.9 ->1.23.8 *
– Service Installer for VMware Tanzu	– 1.3	
– Pod Networking (CNI)	– Antrea Advanced	1 Core
Amazon EKS	**Version**	**Notes**
– Kubernetes version	– v1.22.9	
– NSX Advanced Load Balancer	– 21.1.4	Enterprise license
– Pod Networking (CNI)	– Amazon Native VPC CNI	
SaaS Services	**Version**	**Notes**
– Tanzu Mission Control	– Advanced	1 Core
– Tanzu Observability		20 PPS
– Tanzu Service Mesh	Advanced	

* On VMware Cloud on AWS, Kubernetes version was upgraded from 1.5.3 to 1.6.0 to match latest available version at the time of the release of this second version of the RA. This also validated the upgrade process without issue in the reference environment. Instructions to upgrade the management and workload clusters can be found here.

Solution Configuration

Any solution, especially a multi-cloud solution as complex as presented in this document can be configurated in multiple ways depending on the requirements at hand. This document presents the solution configuration in a modular fashion where each site configuration is independent of each other except for GSLB and Avi DNS service configuration which depends on the number of sites configured. Distributed application functionality depends on these services across sites. The flow chart below shows high-level workflow used to configure the sites in this reference architecture.

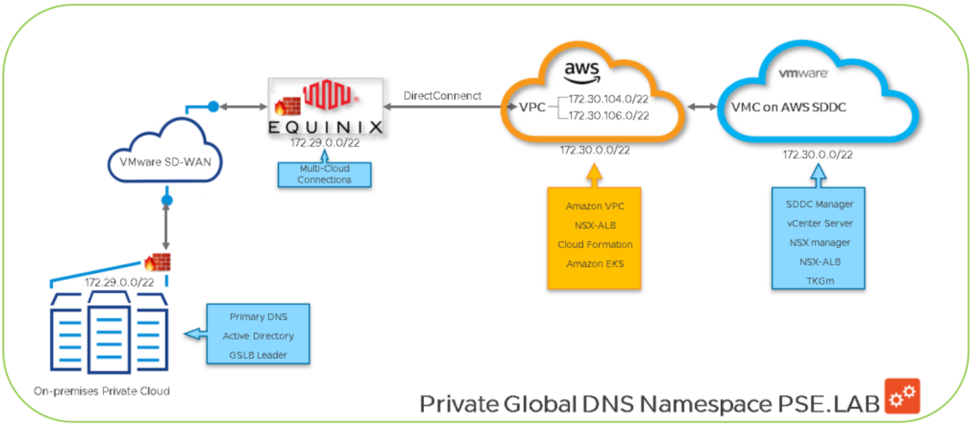
Figure 2: Configuration workflow


On Premise VxRail
Install VxRail
Deploy / Configure NSX ALB Controllers
Create Avi DNS Service on-premises
Delegate sub-domain to NSX-ALB
Enable Workload Management
Create namespace
Create CLI VM Install vSphere tools
Deploy workload cluster
Install Avi Kubernetes Operator
Install and test apps locally
Add cluster to Tanzu Mission Control
Amazon EKS
Subscribe to Avi Network in AWS
Create IAM Roles for NSX ALB on AWS
Deploy / Configure NSX ALB Controllers
Create Avi DNS Service for AWS
Create EKS clusters
Create CLI VM Install AWS cli
Install AKO on EKS cluster
Install AMKO On-premise cluster
Install and test apps locally
Create AWS Creds in Tanzu Mission Control
Create / Attach EKS clusters
VMW Cloud AWS
Install Service Installer for Tanzu
Create Management Segment in VMC
Gather SDDC & Marketplace Tokens
Install TKGm using Service Installer
Create Services cluster (if required)
Create Avi DNS service for VMC
Distributed Apps
Configure Avi GSLB on-prem
Configure Avi GSLB AWS
Configure Avi GSLB VMC
Enable Avi-Integration with TSM
On-Board Clusters Tanzu Service Mesh
Install microservices app on clusters
Create Global Namespace TSM
Task Legend
On premises
Amazon EKS
VMware Cloud on AWS
Tanzu Service Mesh

Solution Network Overview

As depicted in figure 2, the on-premises private cloud (datacenter) is connected to Equinix via VMware SD-WAN. As compared to MPLS circuits which are expensive, VMware SD-WAN (formerly known as VeloCloud) provides a cost effective, secure, and zero-touch deployment option for WAN (Wide-Area-Network). Connections to AWS VPC and VMware Cloud on AWS are configured via AWS DirectConnect. DirectConnect bandwidth can be configured per customer requirements from 50Mbps to 10 Gbps. For this reference architecture 500 Mbps was configured for these connections.

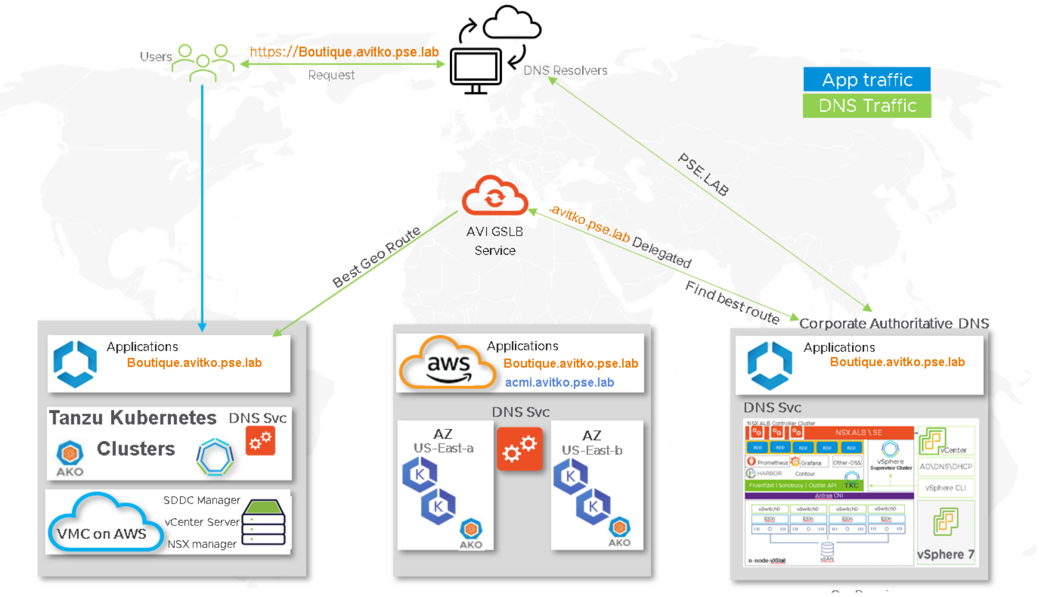
Figure 3: Logical Lab Network



Note: An intermediate cloud services provider is not a requirement for this reference architecture. It is however an option for customers who want to take advantage of these services for their multi-cloud connectivity. Alternatively, customers can connect to their cloud providers individually via VPN or direct connections. For more information on how Equinix is configured to provide multi-cloud connectivity, please see [Multi-Cloud Network Connectivity with Equinix Overview]().

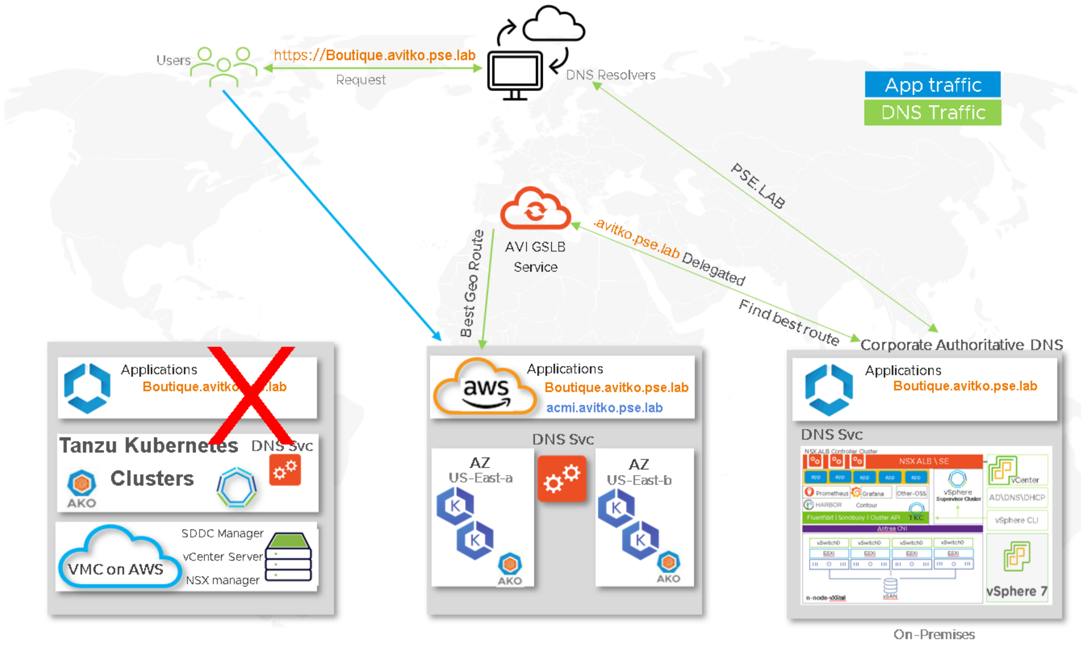
NSX Advanced Load Balancer provide GSLB functionality in this reference architecture. Global server loading balancing (GSLB) is the process of balancing an application's load across instances of the application that have been deployed to multiple locations. Load balancing can be performed based on user's geo-location or round-robin algorithms. With GSLB, when a Kubernetes application is installed, a virtual service is created with application's URL. Users access the application using its URL. The user is directed to the appropriate site based on algorithm and preference set by the GSLB administrator. Figure 4 below shows the GSLB workflow.

Figure 4: Global server load balancing



In instances when an application or site is unavailable the requests are serviced by the active sites. For this reference architecture the authoritative DNS sever was in the on-prem datacenter. For redundancy, secondary DNS servers were also installed on the Equinix site. If desired, the corporate DNS server or DNS zones can also be hosted on Amazon Route53. Figure 5 shows the workflow when a site is not available in GSLB environment.

Figure 5: Site failure in GSLB

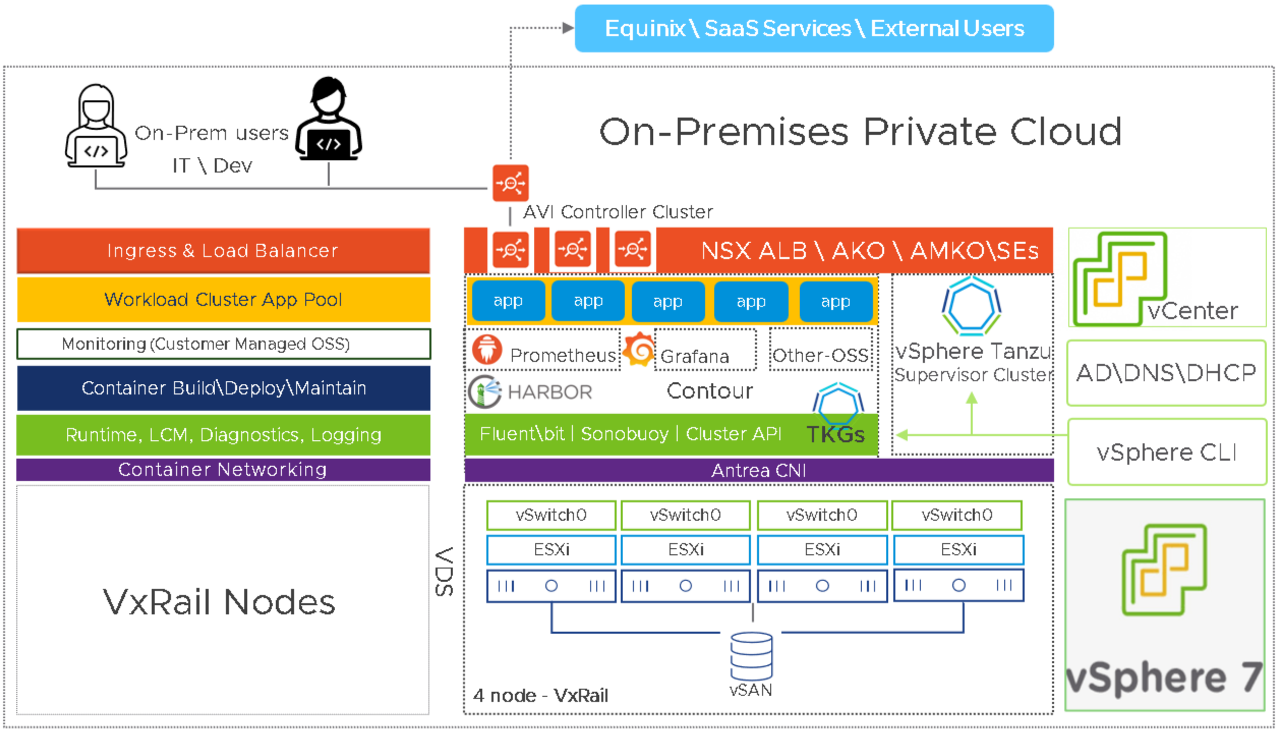


On-Premises Private Cloud

A four node Dell VxRail V570 cluster makes up the infrastructure foundation of this on-premises modern application solution. vSphere with Tanzu provides the capability to run Kubernetes workloads natively on the ESXi hypervisor and create upstream compliant Kubernetes clusters on demand. The NSX Advanced Load Balancer provides dynamically scaling load balancing endpoints for Tanzu Kubernetes clusters provisioned by the Tanzu Kubernetes Grid Service. Along with its Avi Kubernetes Operator and Avi Multi-Cluster Kubernetes Operator, NSX Advanced Load Balancer provides L4 and L7 ingress and load balancing to the deployed workloads. This site also serves as the “leader” GSLB site. VMware vCenter Server along with Dell VxRail Manager makes up the local infrastructure management domain. Harbor is used as the local registry and can be installed manually or via Tanzu Mission Control. In addition to Tanzu Observability, local monitoring and diagnostic, tools such as, Prometheus, Grafana, Fluent are also installed.

VMware vSAN provides enterprise class hyperconverged storage, which is consistent across deployments and integrates fully with VMware Tanzu. From a storage perspective vSAN future-proofs the solution with its integration with object storage types such as Dell ECS, Dell ObjectScale and others.

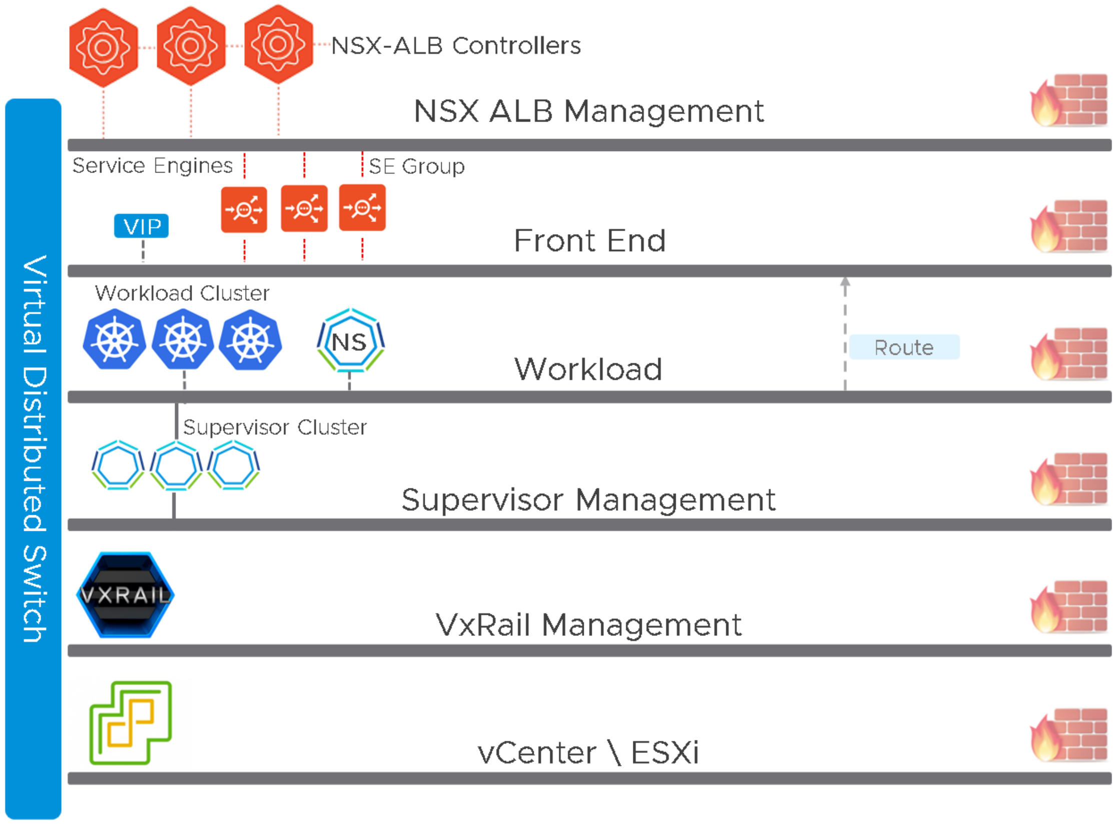
Figure 6: On-premises private cloud architecture



On-premises network overview

Dell VxRail deployment creates the Virtual Distributed Switch with minimum required port groups, such as vCenter and VxRail Management. Additional port groups for vSphere with Tanzu were created for NSX ALB and supervisor node management, front-end, and workload networks. Placing these networks on separate port groups provides isolation and enables application of granular security\firewall policies. Figure 7 depicts the high-level logical diagram of the network stack configured in the lab.

Figure 7: Logical Network Architecture



Note: The VIP and workload network should be routable. NSX Advanced Load Balancer management and vCenter Server management networks should also be able to communicate with each other.

In addition to networks provisioned with Dell VxRail deployment additional network were created for traffic segmentation. Table 3 lists these additional required network\port groups with a brief description.

Table 3: Tanzu Kubernetes networks

Network	Description
NSX ALB Management	NSX Advanced Load Balancer controllers and Service Engines connect to this network
Supervisor Management	TKGs Supervisor nodes are placed on this network
Front End	This is where the users connect to and holds the virtual services and VIPs
Workload	TKG workload cluster control plane and worker nodes connect here

Dell VxRail

Joint engineering between Dell and VMware leads to a curated and optimized VxRail hyperconverged experience. This deep integration combined with the simplicity of the VxRail HCI System Software enables seamless adoption of new technology and features, and provides an ideal platform across core, edge, and cloud.

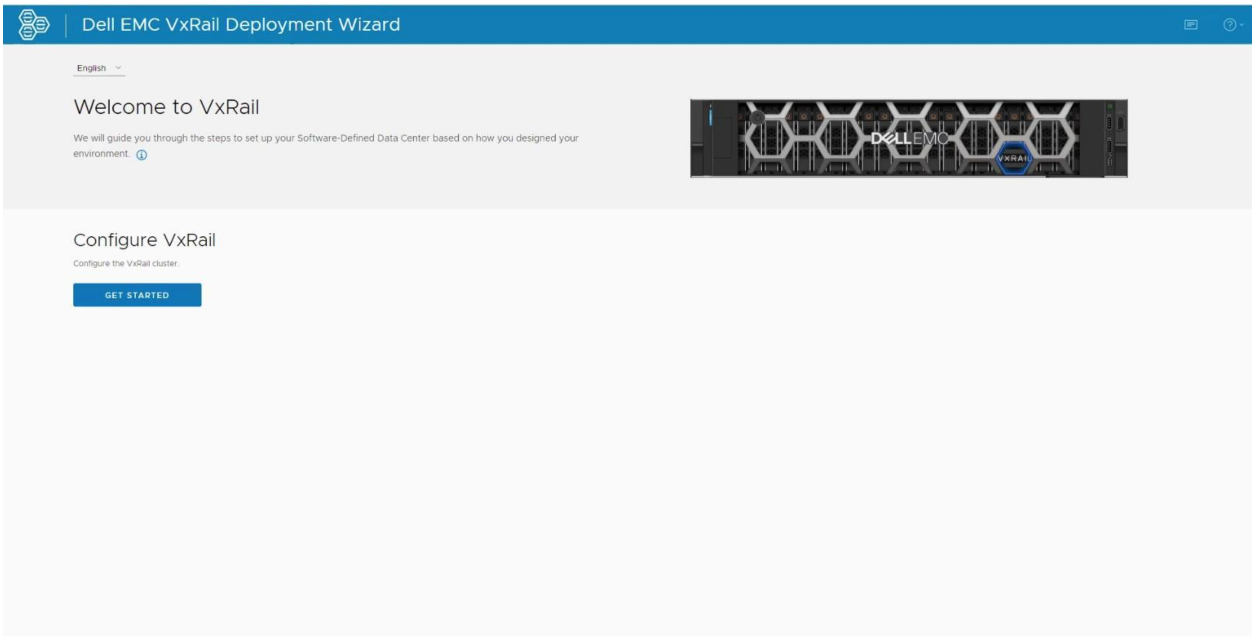
- ✓ Consistent ease of use with automated full stack lifecycle management
- ✓ Simplify with a consistent operational model across your infrastructure landscape.
- ✓ Single point of support with 97% of all cases resolved in house.

Installation

Note: Prior to starting VxRail cluster installation, ensure that hardware is setup properly including Top-Of-Rack switches, and that the nodes are imaged with the desired VxRail version specific image. This image includes ESXi, vSAN, hardware firmware/drivers, and VxRail HCI System Software that will be deployed and configured automatically based on the desired VxRail cluster JSON file configuration parameters. Please consult Dell VxRail support documentation for more details.

- Once the VxRail nodes are powered on, from your jump host, go to https://192.168.10.200 to connect to the VxRail Manager. The VxRail Deployment Wizard welcome screen displays, as shown in figure below. Click GET STARTED on the welcome screen. (NOTE: The following screenshot instructions may vary from the actual deployment configuration used in this reference architecture. These deployment images are shown for general awareness purposes only.)

Figure 8: VxRail Installation.



Welcome to VxRail

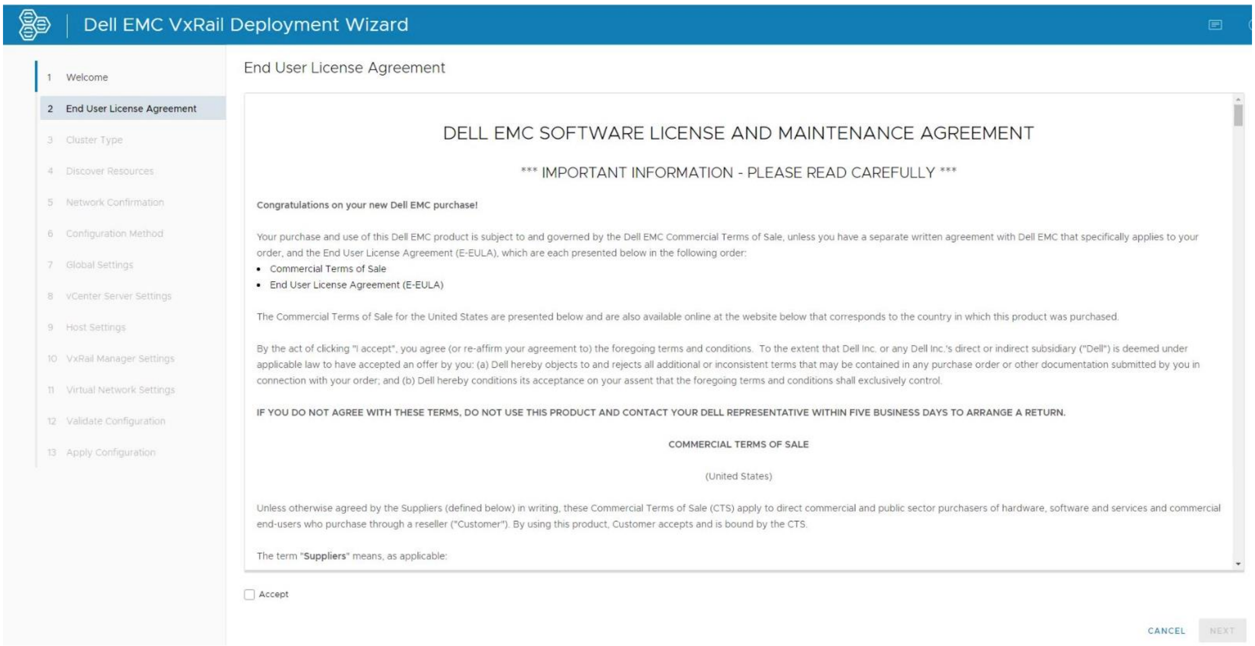
We will guide you through the steps to set up your Software-Defined Data Center based on how you designed your environment.

Configure VxRail

Configure the VxRail cluster.

- On the EULA page, review the terms provided, and if you agree, click ACCEPT.

Figure 9: Accept EULA.


Dell EMC VxRail Deployment Wizard
1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration
CANCEL
NEXT

End User License Agreement

DELL EMC SOFTWARE LICENSE AND MAINTENANCE AGREEMENT

*** IMPORTANT INFORMATION - PLEASE READ CAREFULLY ***

Congratulations on your new Dell EMC purchase!

Your purchase and use of this Dell EMC product is subject to and governed by the Dell EMC Commercial Terms of Sale, unless you have a separate written agreement with Dell EMC that specifically applies to your order, and the End User License Agreement (E-EULA), which are each presented below in the following order:

- Commercial Terms of Sale
- End User License Agreement (E-EULA)

The Commercial Terms of Sale for the United States are presented below and are also available online at the website below that corresponds to the country in which this product was purchased.

By the act of clicking "I accept", you agree (or re-affirm your agreement to) the foregoing terms and conditions. To the extent that Dell Inc. or any Dell Inc.'s direct or indirect subsidiary ("Dell") is deemed under applicable law to have accepted an offer by you: (a) Dell hereby objects to and rejects all additional or inconsistent terms that may be contained in any purchase order or other documentation submitted by you in connection with your order; and (b) Dell hereby conditions its acceptance on your assent that the foregoing terms and conditions shall exclusively control.

IF YOU DO NOT AGREE WITH THESE TERMS, DO NOT USE THIS PRODUCT AND CONTACT YOUR DELL REPRESENTATIVE WITHIN FIVE BUSINESS DAYS TO ARRANGE A RETURN.

COMMERCIAL TERMS OF SALE

(United States)

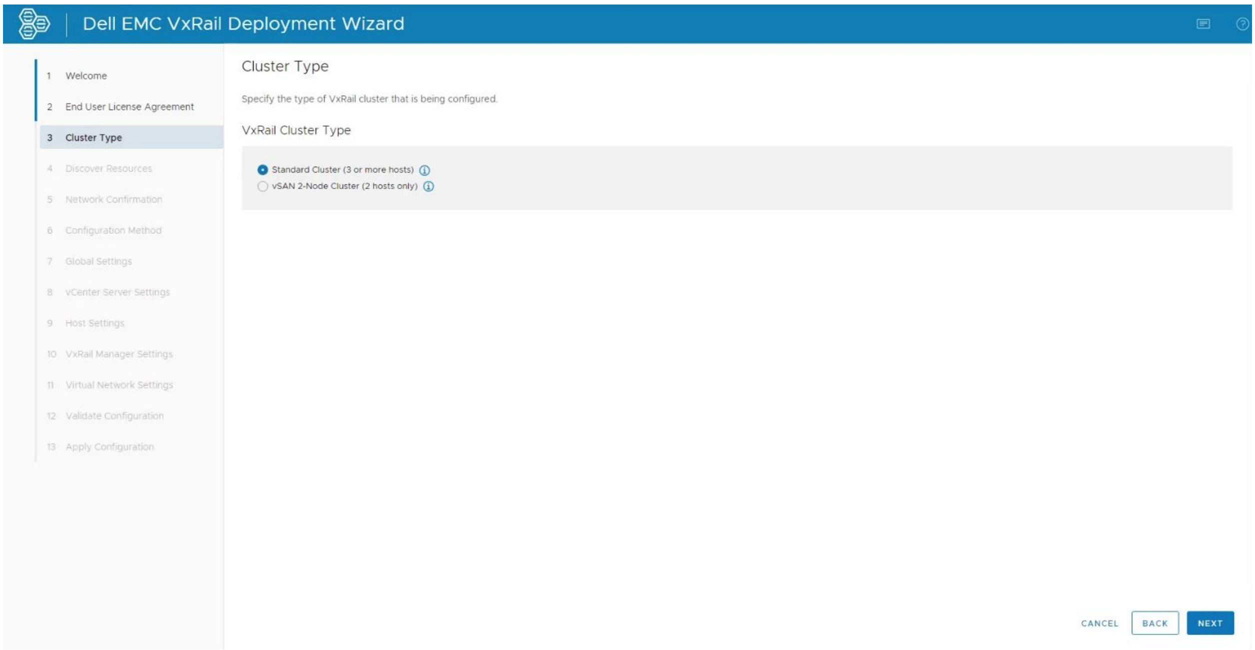
Unless otherwise agreed by the Suppliers (defined below) in writing, these Commercial Terms of Sale (CTS) apply to direct commercial and public sector purchasers of hardware, software and services and commercial end-users who purchase through a reseller ("Customer"). By using this product, Customer accepts and is bound by the CTS.

The term "**Suppliers**" means, as applicable:

□ Accept

- The VxRail cluster type page displays. Select Standard Cluster. This will deploy a VxRail with vSAN HCI cluster deployment type which is what we are using in this reference architecture.

Figure 10: Select cluster type.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Cluster Type

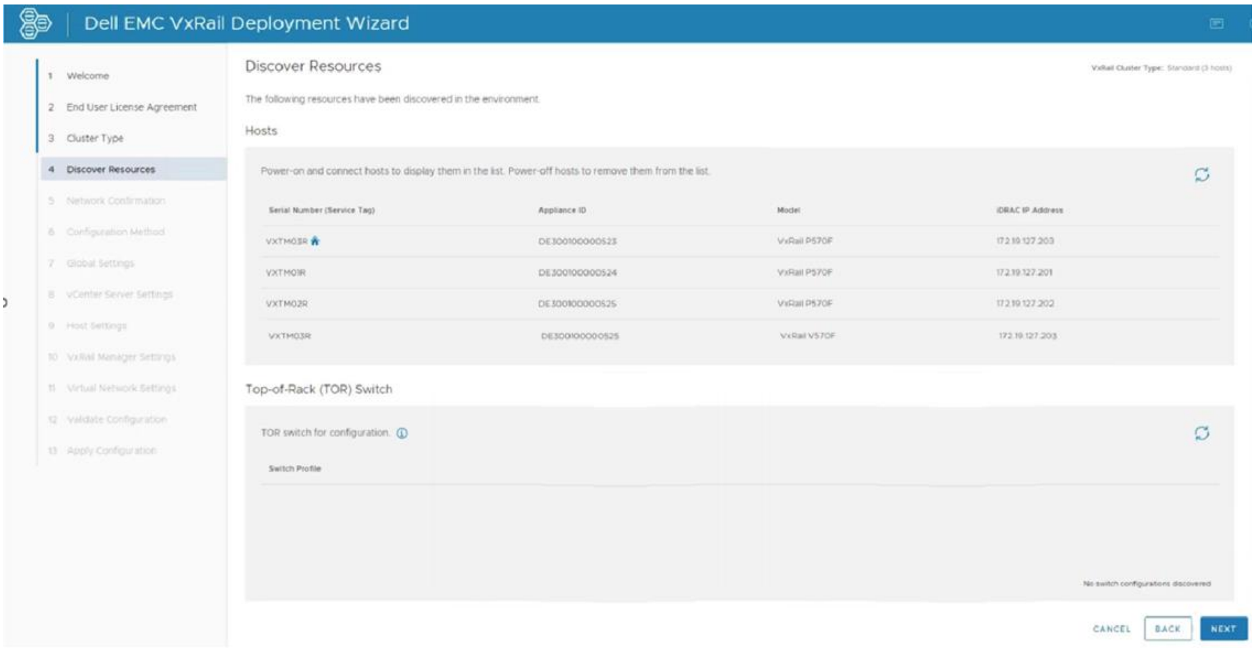
Specify the type of VxRail cluster that is being configured.

VxRail Cluster Type

● Standard Cluster (3 or more hosts)
○ vSAN 2-Node Cluster (2 hosts only)

- When all VxRail nodes are discovered click NEXT.

Figure 10: VxRail nodes discovered.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Discover Resources

VxRail Cluster Type: Standard (3 hosts)

The following resources have been discovered in the environment.

Hosts

Power-on and connect hosts to display them in the list. Power-off hosts to remove them from the list.

Serial Number (Service Tag)	Appliance ID	Model	iDRAC IP Address
VXTM03R	DE300100000523	VxRail P570F	172.19.127.203
VXTM01R	DE300100000524	VxRail P570F	172.19.127.201
VXTM02R	DE300100000525	VxRail P570F	172.19.127.202
VXTM03R	DE300100000525	VxRail V570F	172.19.127.203

Top-of-Rack (TOR) Switch

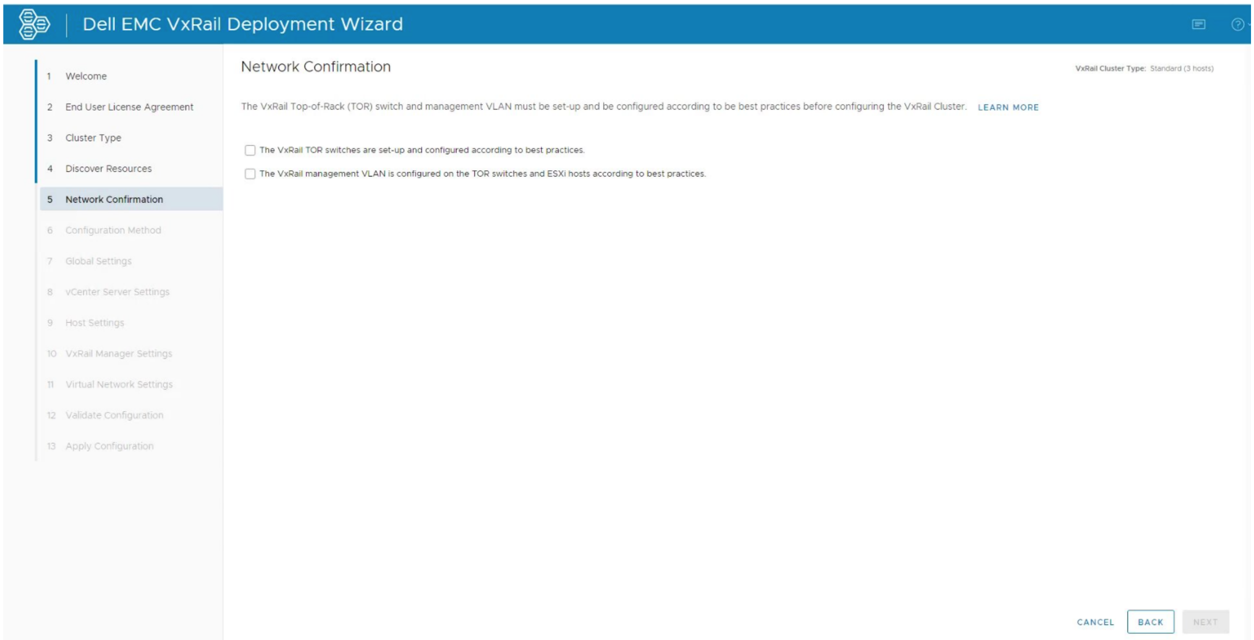
TOR switch for configuration.

Switch Profile

No switch configurations discovered

- Acknowledge that network is configured per best practices, by checking the two boxes.

Figure 11: VxRail network confirmation.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Network Confirmation

VxRail Cluster Type: Standard (3 hosts)

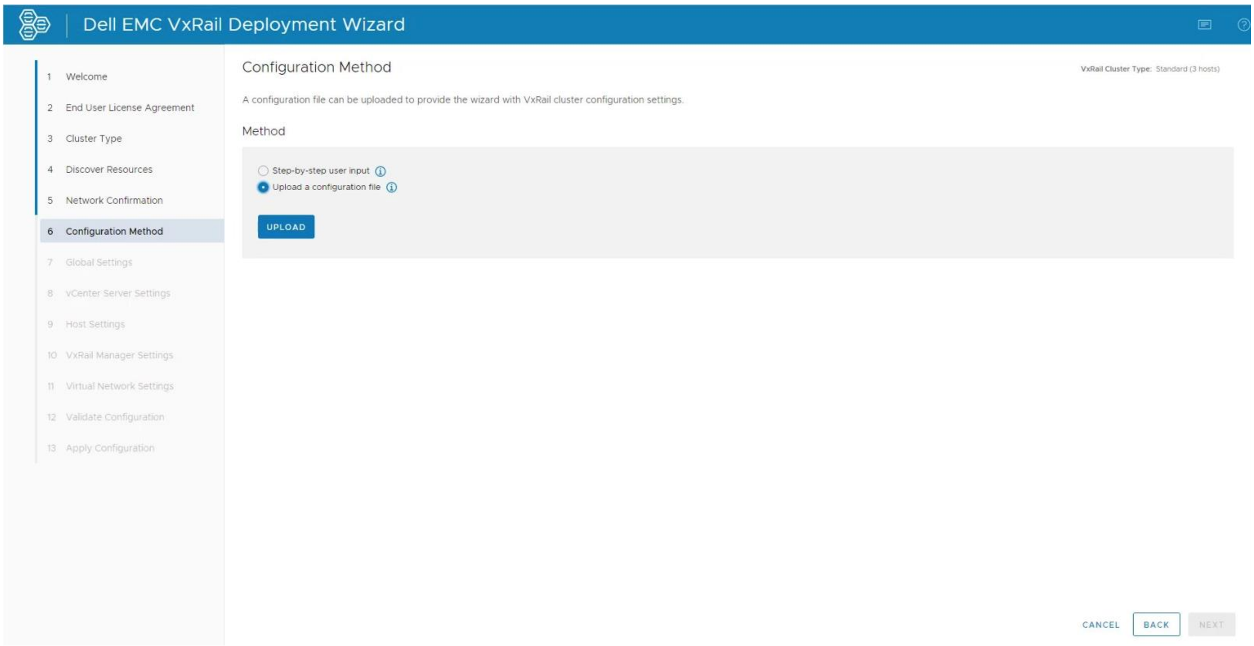
The VxRail Top-of-Rack (TOR) switch and management VLAN must be set-up and be configured according to be best practices before configuring the VxRail Cluster. LEARN MORE

☐ The VxRail TOR switches are set-up and configured according to best practices.

☐ The VxRail management VLAN is configured on the TOR switches and ESXi hosts according to best practices.

- There are two configuration methods. Users can choose to provide inputs for each step of the process or use a preconfigured JSON file. A JSON file with preconfigured cluster configuration parameters was used for this reference architecture. Click upload.

Figure 12: VxRail configuration methods.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Configuration Method

VxRail Cluster Type: Standard (3 hosts)

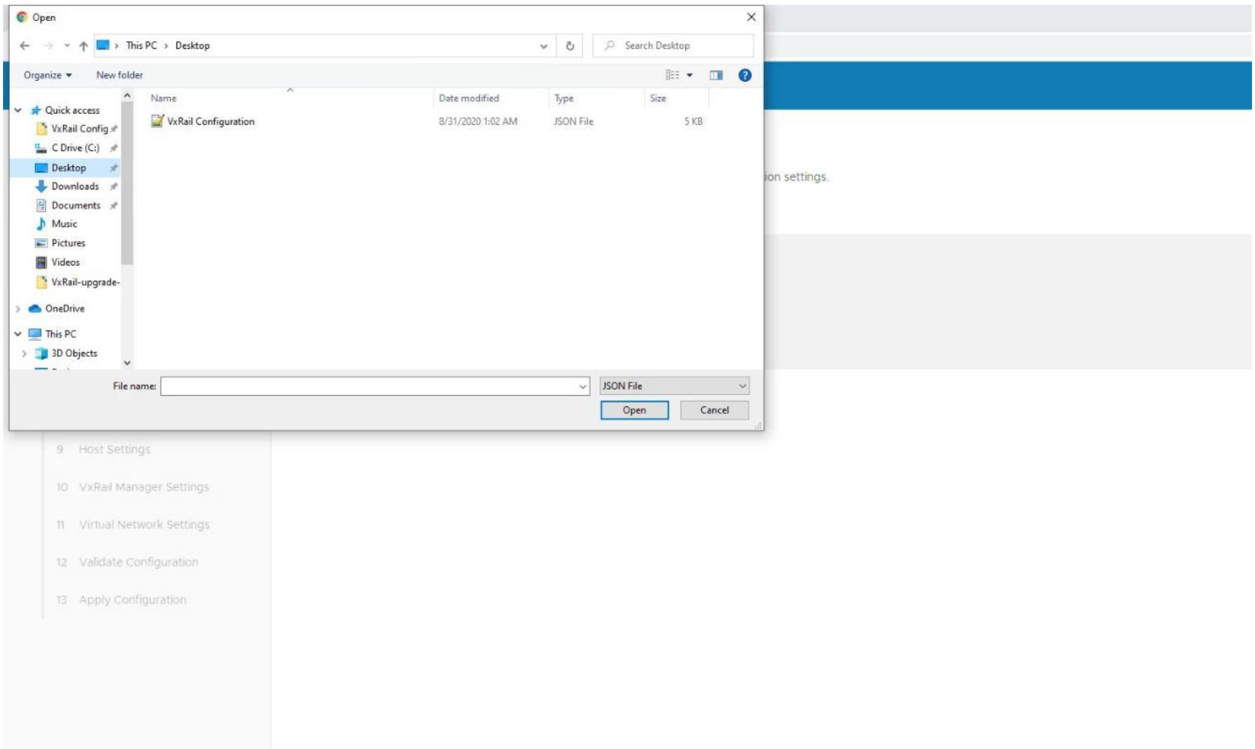
A configuration file can be uploaded to provide the wizard with VxRail cluster configuration settings.

Method

○ Step-by-step user input ⓘ
● Upload a configuration file ⓘ

- Browse and select the preconfigured JSON file. Click “Open” to upload VxRail configuration file.

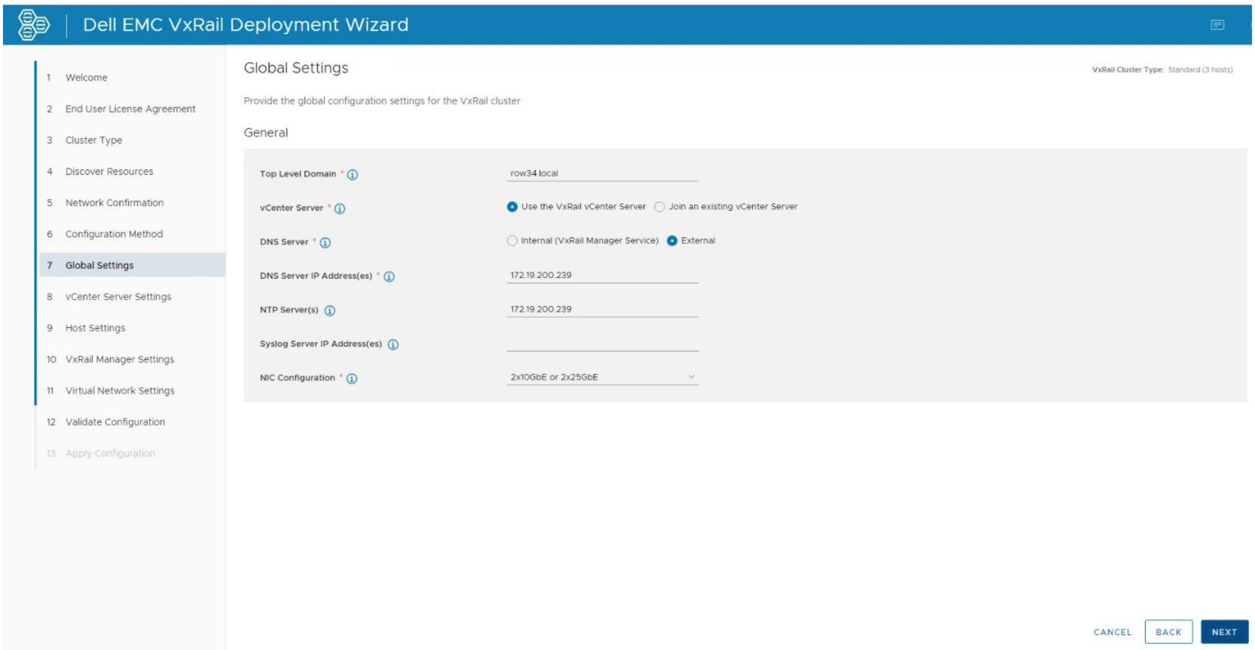
Figure 13: VxRail configuration file.



Name	Date modified	Type	Size
VxRail Configuration	8/31/2020 1:02 AM	JSON File	5 KB

- All required cluster configuration parameters have now automatically populated. Validate and confirm the global settings and click NEXT.

Figure 14: VxRail global settings.


Dell EMC VxRail Deployment Wizard
1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration
CANCEL
BACK
NEXT

Global Settings

VxRail Cluster Type: Standard (3 hosts)

Provide the global configuration settings for the VxRail cluster

General

Top Level Domain *: row34.local

vCenter Server *: ● Use the VxRail vCenter Server ○ Join an existing vCenter Server

DNS Server *: ○ Internal (VxRail Manager Service) ● External

DNS Server IP Address(es) *: 172.19.200.239

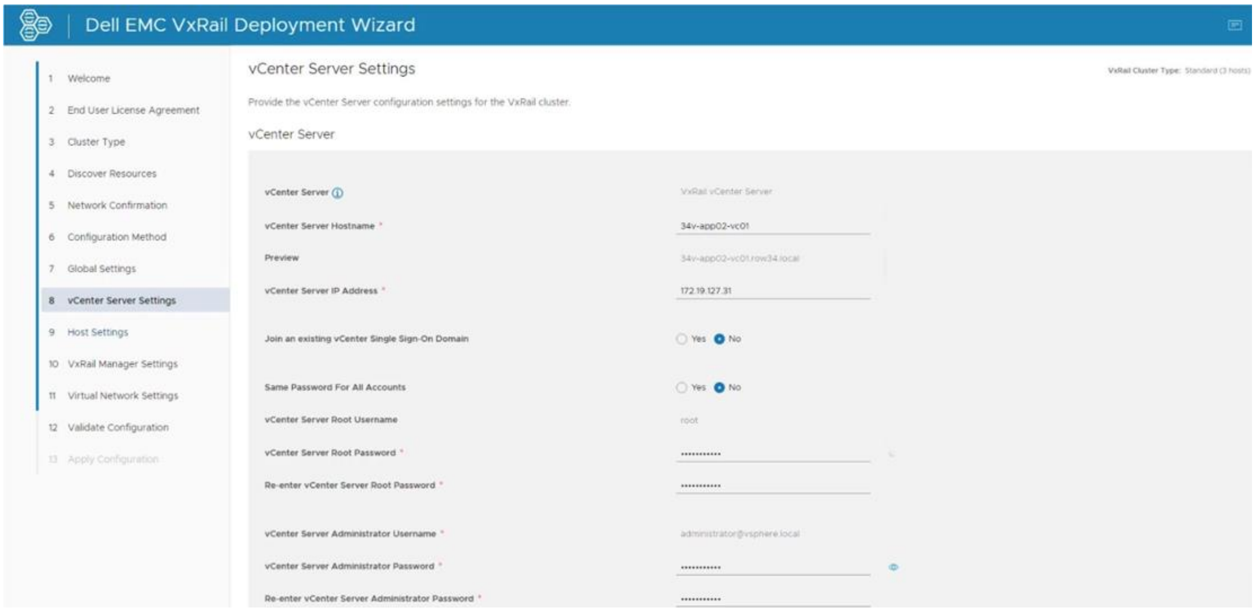
NTP Server(s): 172.19.200.239

Syslog Server IP Address(es):

NIC Configuration *: 2x10GbE or 2x25GbE

- Validate and confirm the vCenter settings and click NEXT.

Figure 15: VxRail vCenter settings.


Dell EMC VxRail Deployment Wizard
1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

vCenter Server Settings

VxRail Cluster Type: Standard (3 hosts)

Provide the vCenter Server configuration settings for the VxRail cluster.

vCenter Server

vCenter Server: VxRail vCenter Server

vCenter Server Hostname *: 34v-app02-vc01

Preview: 34v-app02-vc01.row34.local

vCenter Server IP Address *: 172.19.127.31

Join an existing vCenter Single Sign-On Domain: ○ Yes ● No

Same Password For All Accounts: ○ Yes ● No

vCenter Server Root Username: root

vCenter Server Root Password *: ••••••••••••

Re-enter vCenter Server Root Password *: ••••••••••••

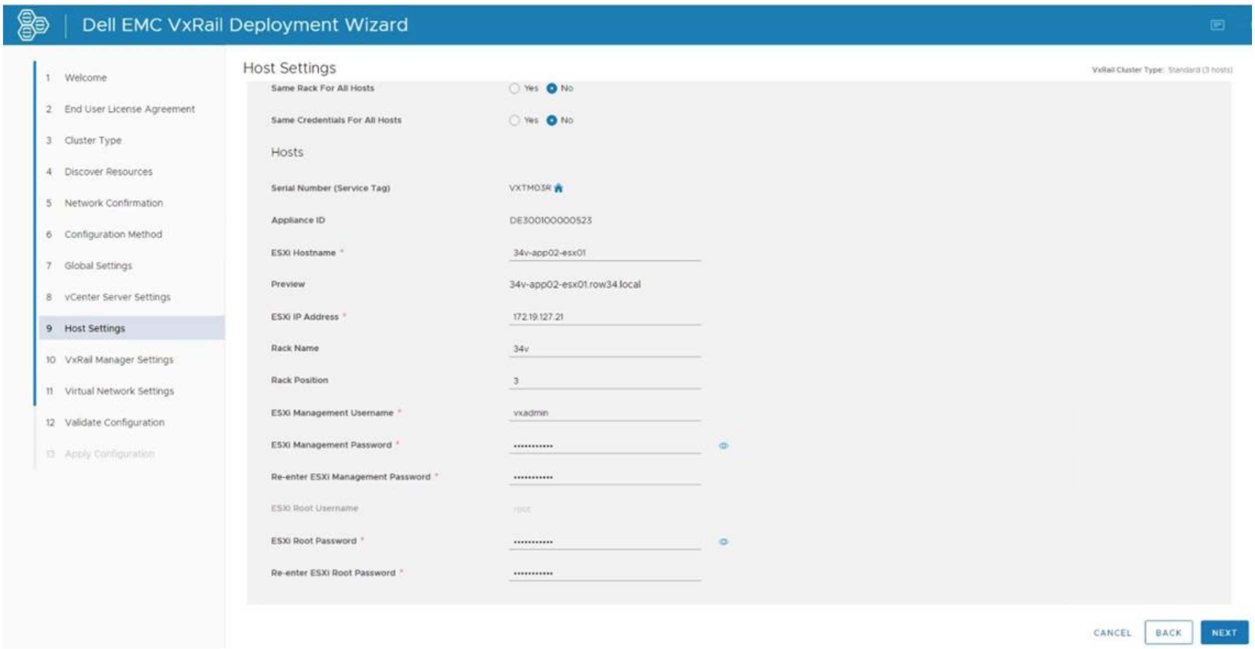
vCenter Server Administrator Username *: administrator@vsphere.local

vCenter Server Administrator Password *: ••••••••••••

Re-enter vCenter Server Administrator Password *: ••••••••••••

- Validate and confirm the individual ESXi hosts settings and click NEXT.

Figure 16: VxRail host settings.


Dell EMC VxRail Deployment Wizard
1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration
CANCEL
BACK
NEXT

Host Settings

VxRail Cluster Type: Standard (3 hosts)

Same Rack For All Hosts: ○ Yes ● No

Same Credentials For All Hosts: ○ Yes ● No

Hosts

Serial Number (Service Tag): VXTM03R

Appliance ID: DE300100000523

ESXi Hostname *: 34v-app02-esx01

Preview: 34v-app02-esx01.row34.local

ESXi IP Address *: 172.19.127.21

Rack Name: 34v

Rack Position: 3

ESXi Management Username *: vxadmin

ESXi Management Password *: ••••••••••••

Re-enter ESXi Management Password *: ••••••••••••

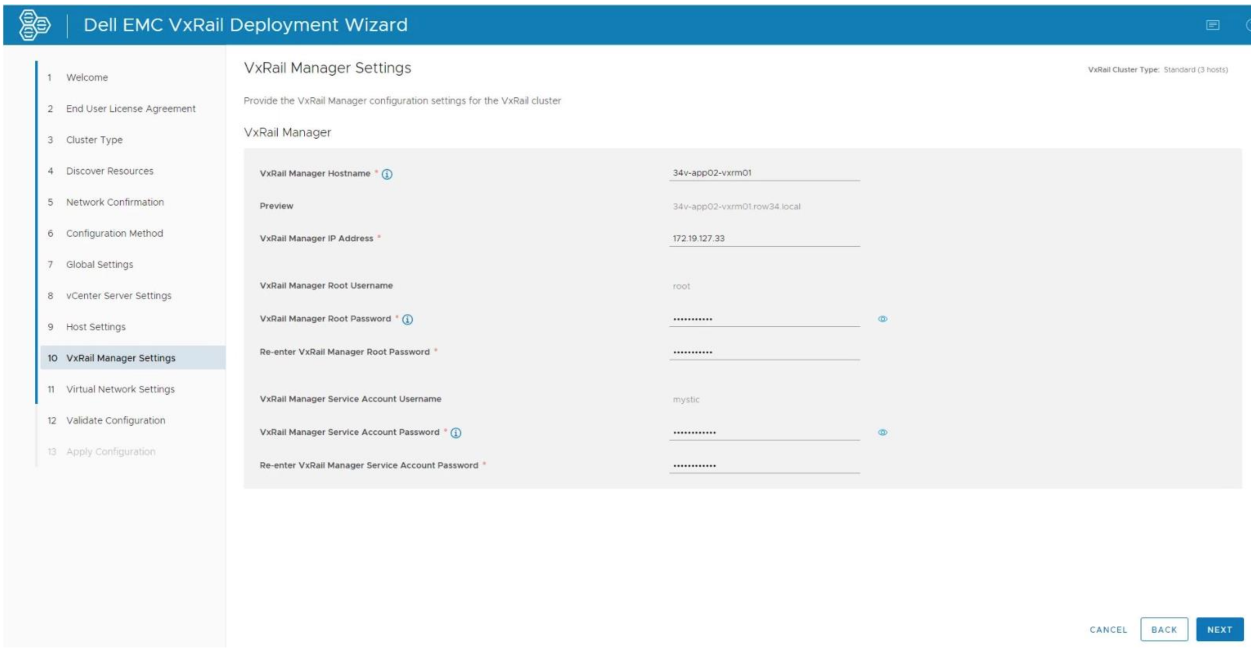
ESXi Root Username: root

ESXi Root Password *: ••••••••••••

Re-enter ESXi Root Password *: ••••••••••••

- Validate and confirm the VxRail Manager settings and click NEXT.

Figure 17: VxRail Manager settings.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

VxRail Manager Settings

VxRail Cluster Type: Standard (3 hosts)

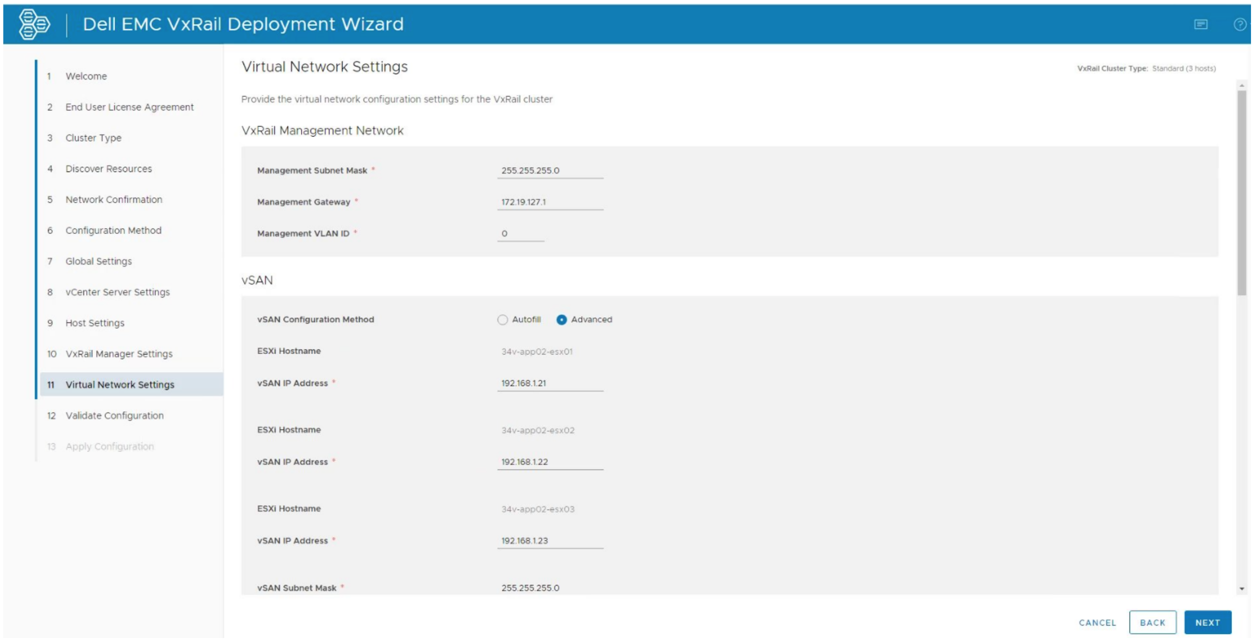
Provide the VxRail Manager configuration settings for the VxRail cluster

VxRail Manager

VxRail Manager Hostname *	34v-app02-vxrm01
Preview	34v-app02-vxrm01.row34.local
VxRail Manager IP Address *	172.19.127.33
VxRail Manager Root Username	root
VxRail Manager Root Password *	•••••••••••
Re-enter VxRail Manager Root Password *	•••••••••••
VxRail Manager Service Account Username	mystic
VxRail Manager Service Account Password *	••••••••••••
Re-enter VxRail Manager Service Account Password *	••••••••••••

- Validate and confirm the virtual network settings. Click NEXT.

Figure 18: VxRail virtual network settings.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Virtual Network Settings

VxRail Cluster Type: Standard (3 hosts)

Provide the virtual network configuration settings for the VxRail cluster

VxRail Management Network

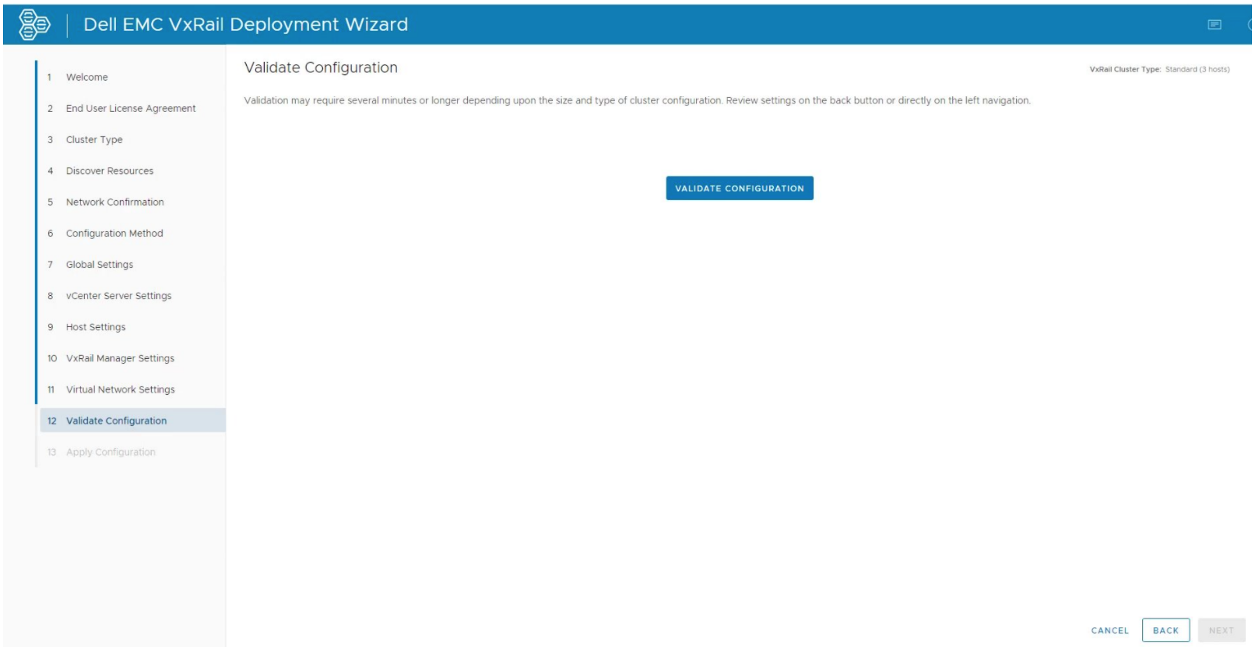
Management Subnet Mask *	255.255.255.0
Management Gateway *	172.19.127.1
Management VLAN ID *	0

vSAN

vSAN Configuration Method	○ Autofill ● Advanced
ESXi Hostname	34v-app02-esx01
vSAN IP Address *	192.168.1.21
ESXi Hostname	34v-app02-esx02
vSAN IP Address *	192.168.1.22
ESXi Hostname	34v-app02-esx03
vSAN IP Address *	192.168.1.23
vSAN Subnet Mask *	255.255.255.0

- Next, execute an automated VxRail Manager cluster configuration validation check to ensure all parameters have been entered correctly. Click on the Validate Configuration button.

Figure 19: VxRail validate settings.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

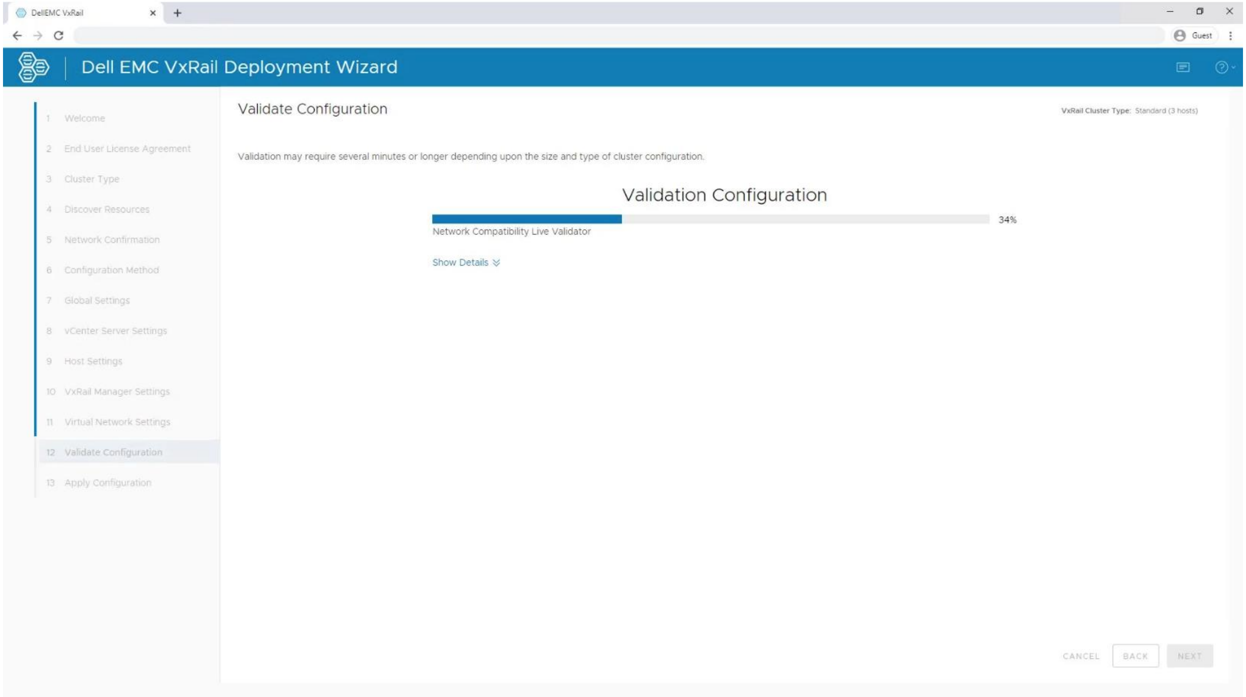
Validate Configuration

VxRail Cluster Type: Standard (3 hosts)

Validation may require several minutes or longer depending upon the size and type of cluster configuration. Review settings on the back button or directly on the left navigation.

- VxRail Manager will automatically validate the configuration input provided on previous screens.

Figure 20: VxRail validation process.



Dell EMC VxRail Deployment Wizard

1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration

Validate Configuration

VxRail Cluster Type: Standard (3 hosts)

Validation may require several minutes or longer depending upon the size and type of cluster configuration.

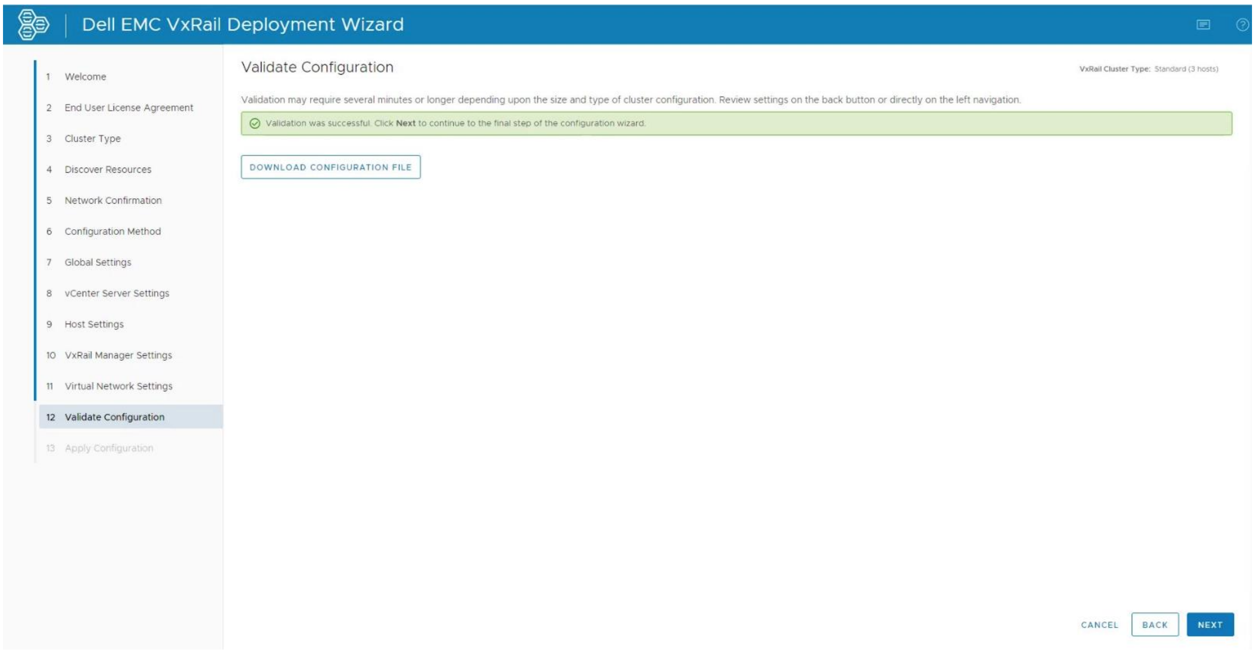
Validation Configuration

34%

Network Compatibility Live Validator

- The configuration JSON file can be downloaded once the validation has completed.

Figure 21: VxRail validation complete.


Dell EMC VxRail Deployment Wizard
1 Welcome
2 End User License Agreement
3 Cluster Type
4 Discover Resources
5 Network Confirmation
6 Configuration Method
7 Global Settings
8 vCenter Server Settings
9 Host Settings
10 VxRail Manager Settings
11 Virtual Network Settings
12 Validate Configuration
13 Apply Configuration
DOWNLOAD CONFIGURATION FILE
CANCEL
BACK
NEXT

Validate Configuration

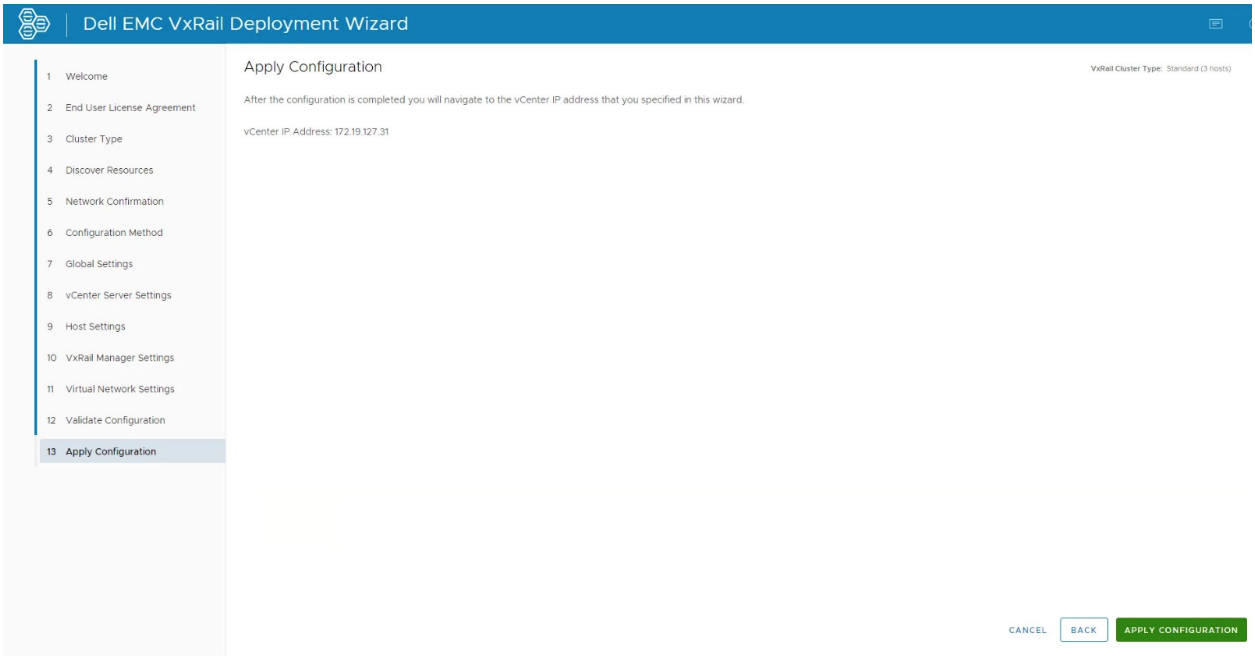
VxRail Cluster Type: Standard (3 hosts)

Validation may require several minutes or longer depending upon the size and type of cluster configuration. Review settings on the back button or directly on the left navigation.

Validation was successful. Click **Next** to continue to the final step of the configuration wizard.

- Once the configuration validation has completed successfully, the cluster configuration can be used to automatically create the cluster.

Figure 22: VxRail apply configuration.



Apply Configuration

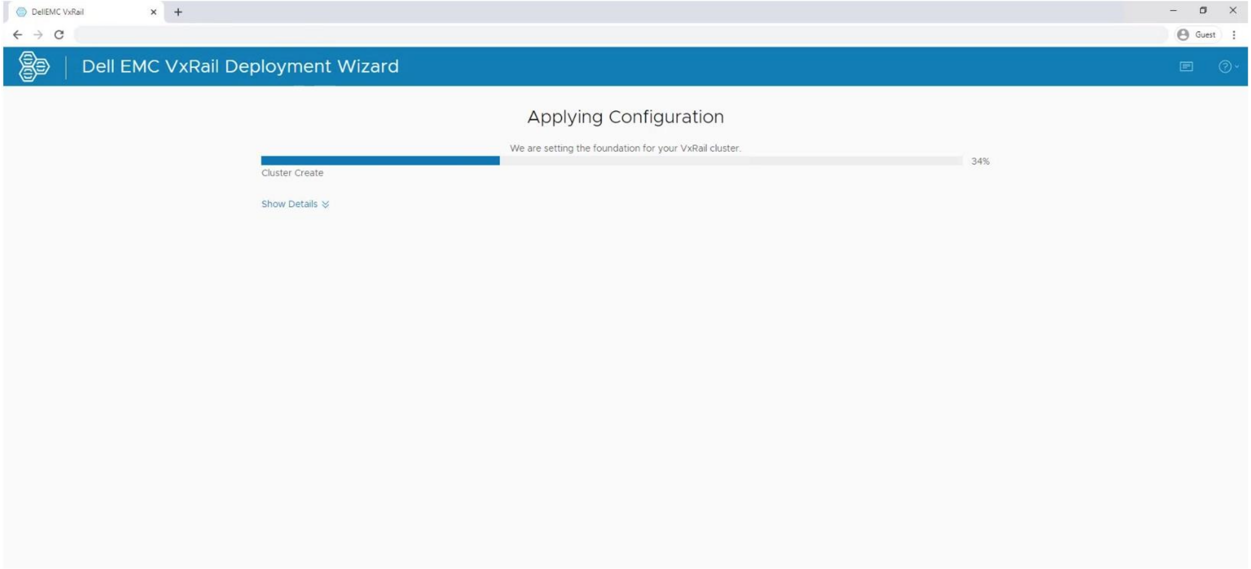
VxRail Cluster Type: Standard (3 hosts)

After the configuration is completed you will navigate to the vCenter IP address that you specified in this wizard.

vCenter IP Address: 172.19.127.31

- The process will take a few minutes to complete.

Figure 23: VxRail applying configuration.



Applying Configuration

We are setting the foundation for your VxRail cluster.

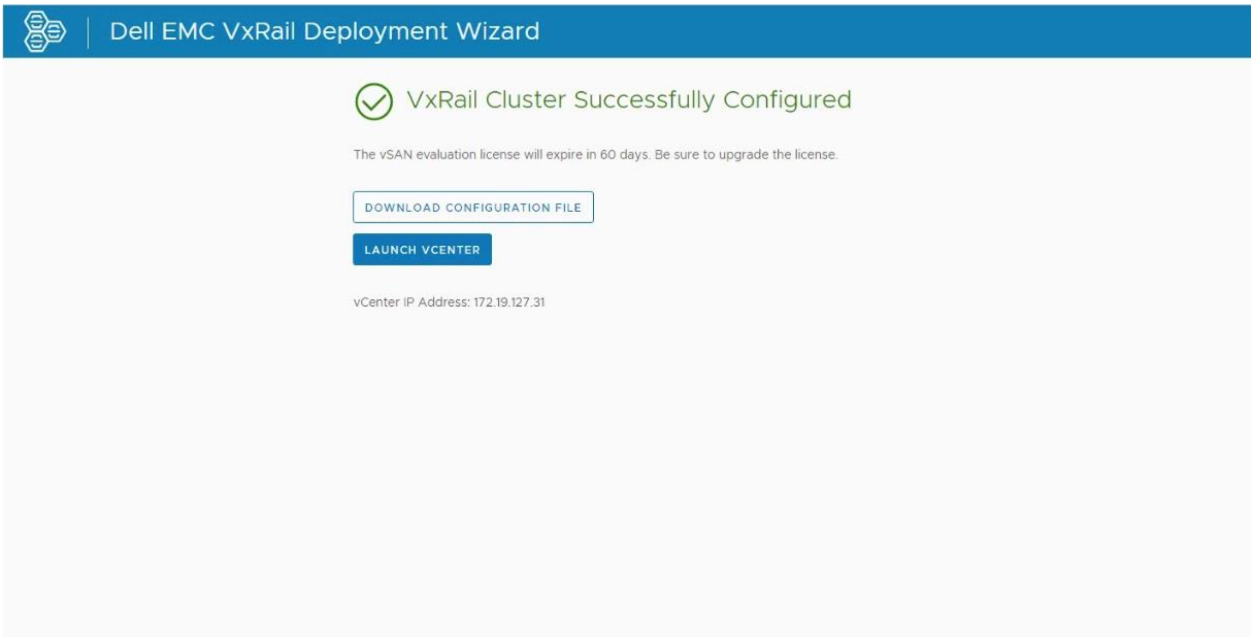
34%

Cluster Create

Show Details

- After a few minutes, the VxRail Installation completes and vCenter (configured with the VxRail Manager vCenter Plugin) can be accessed by clicking "LAUNCH VCENTER" button.

Figure 24: VxRail Installation complete.


Dell EMC VxRail Deployment Wizard

VxRail Cluster Successfully Configured

The vSAN evaluation license will expire in 60 days. Be sure to upgrade the license.

DOWNLOAD CONFIGURATION FILE
LAUNCH VCENTER

vCenter IP Address: 172.19.127.31

VMware NSX Advanced Load Balancer

NSX Advanced Load Balancer (formerly known as Avi) comes in two editions, Essentials, and Enterprise. To use L7 load balancing with NSX Advanced Load Balancer, the Enterprise edition is required and was used for this reference architecture. The NSX Advanced Load Balancer provides dynamically scaling load balancing endpoints for Tanzu Kubernetes clusters provisioned by the Tanzu Kubernetes Grid Service. Once you have configured the Controller, it automatically provisions load balancing endpoints for you. The Controller creates a virtual service and deploys Service Engine VMs to host that service. This virtual service provides load balancing for the Kubernetes control plane. NSX Advanced Load Balancer has some key components that are explained below.

NSX Advanced Load Balancer Controller: As the name suggests, NSX Advanced Load Balancer Controller controls and manages the provisioning of service engines, coordinating resources across service engines, and aggregating service engine metrics and logging. It interacts with vCenter Server to automate the load balancing for Kubernetes clusters. It is deployed as an OVA and provides a Web interface and CLI.

NSX Advanced Load Balancer Service Engine: Service Engine runs one or more virtual services and is a data plane component and runs as a virtual machine. Service engines are provisioned and controlled by the controller. The service engines have two interfaces. One connects to the NSX Advanced Load Balancer Controller management network and the second connects to the front-end network from where virtual services are accessed. For service engine sizing guidance, see Sizing Service Engines.

Avi Kubernetes Operator (AKO): Avi Kubernetes Operator runs as a Kubernetes POD in the Supervisor, management, and workload clusters to provide ingress and load balancing.

Avi Multi-Cluster Kubernetes Operator (AMKO): Avi Multi-Cluster Kubernetes Operator runs in a pod in the Tanzu GSLB leader cluster. In conjunction with Avi Kubernetes Operator, Avi Multi-Cluster Kubernetes Operator facilitates multi-cluster application deployment. It maps the same application deployed on multiple clusters to a single GSLB service, extending application ingresses across multi-region and multi-availability zone deployments.

Steps to configure NSX Advanced Load Balancer

NSX Advanced Load Balancer Controller is deployed as VM using the OVA that can be downloaded from https://customerconnect.vmware.com/ using an account that has access to downloading software packages. Once the OVA is downloaded import it to vCenter. For this reference architecture version 20.1.7 was used with Enterprise license. The process of deploying and configuring NSX Advanced Load Balancer follows. Screenshots are used where necessary to emphasize specific configurations.

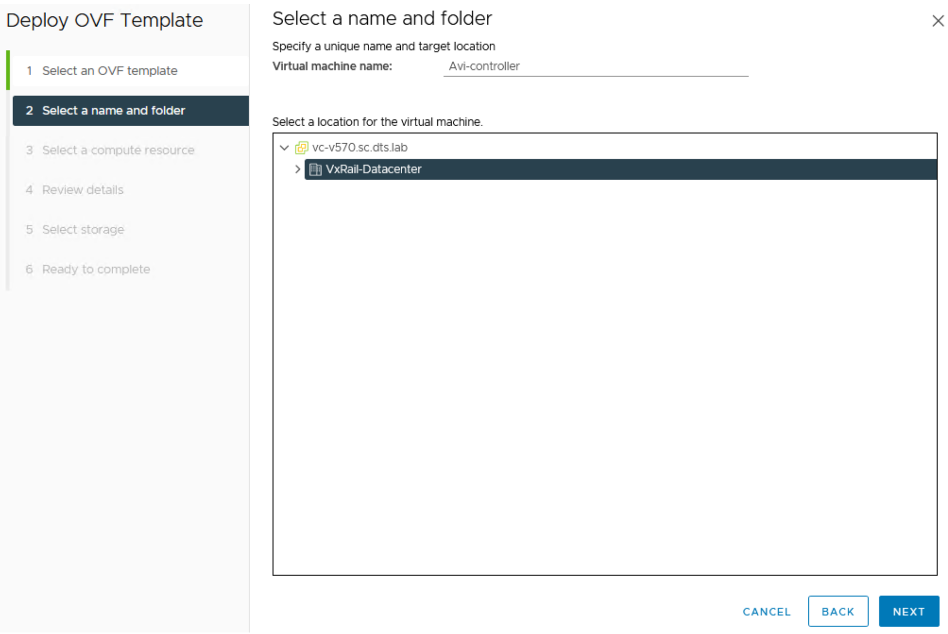
Prerequisites:

- VxRail cluster is already installed and configured.
- vSphere distributed switch port groups for required networks are created as described in network overview section previously.
- A resource pool is created in vCenter that will hold the NSX Advanced Load Balancer virtual machines.

Controller Deployment: On-Premises

1. Import controller OVA and provide a name for the controller.

Figure 25: Import OVA



Select a name and folder

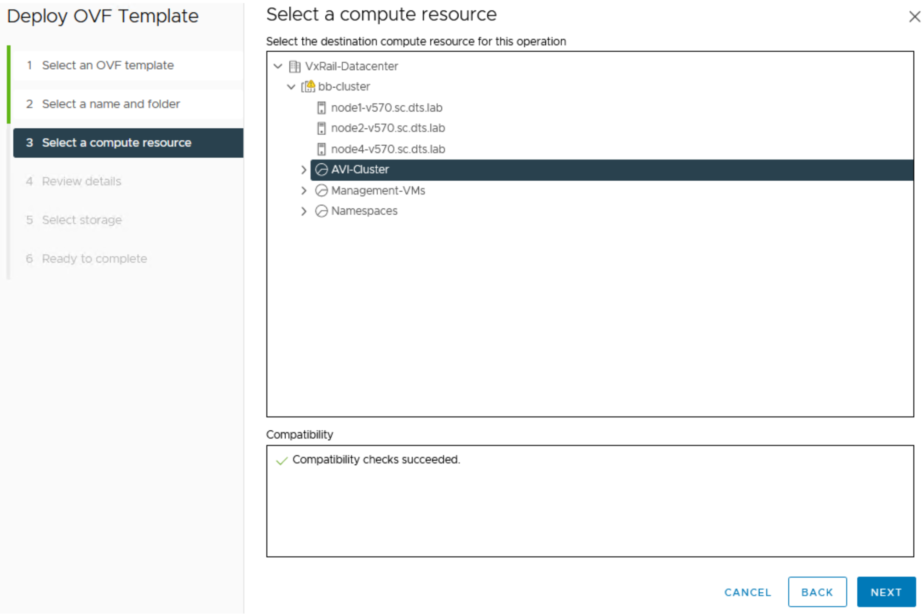
Specify a unique name and target location

Virtual machine name: Avi-controller

Select a location for the virtual machine.

2. Select the resource pool for the controllers.

Figure 26: Select resource pool



Deploy OVF Template

1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Ready to complete

Select a compute resource

Select the destination compute resource for this operation

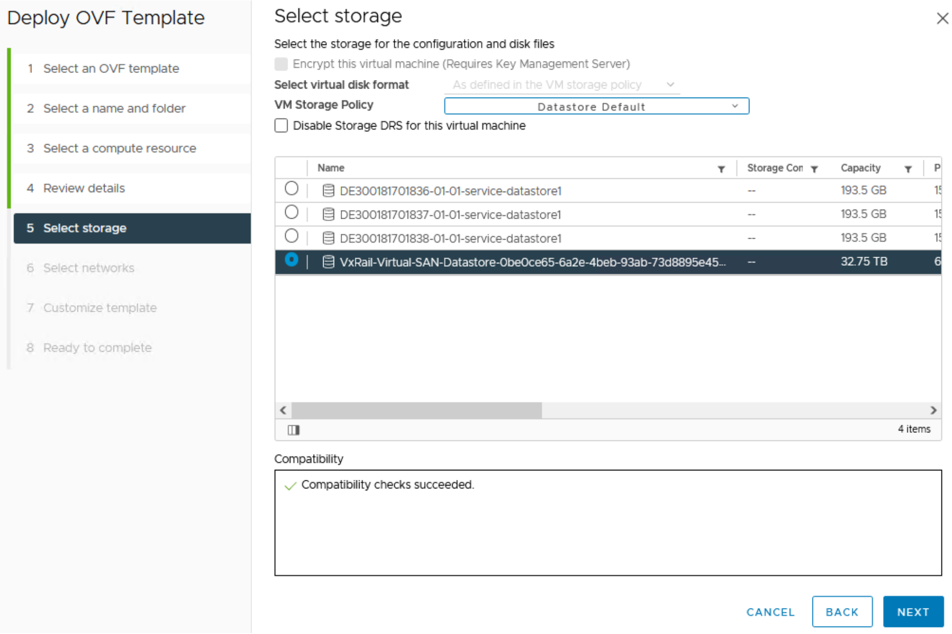
- VxRail-Datacenter
  - bb-cluster
    - node1-v570.sc.dts.lab
    - node2-v570.sc.dts.lab
    - node4-v570.sc.dts.lab
  - AVI-Cluster
  - Management-VMs
  - Namespaces

Compatibility

✓ Compatibility checks succeeded.

3. For storage, select vSAN datastore that was created during VxRail deployment.

Figure 27: Select storage



Deploy OVF Template

1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete

Select storage

Select the storage for the configuration and disk files

□ Encrypt this virtual machine (Requires Key Management Server)

Select virtual disk format: As defined in the VM storage policy

VM Storage Policy: Datastore Default

□ Disable Storage DRS for this virtual machine

	Name	Storage Con	Capacity	P
○	DE300181701836-01-01-service-datastore1	--	193.5 GB	15
○	DE300181701837-01-01-service-datastore1	--	193.5 GB	15
○	DE300181701838-01-01-service-datastore1	--	193.5 GB	15
●	VxRail-Virtual-SAN-Datastore-0be0ce65-6a2e-4beb-93ab-73d8895e45...	--	32.75 TB	6

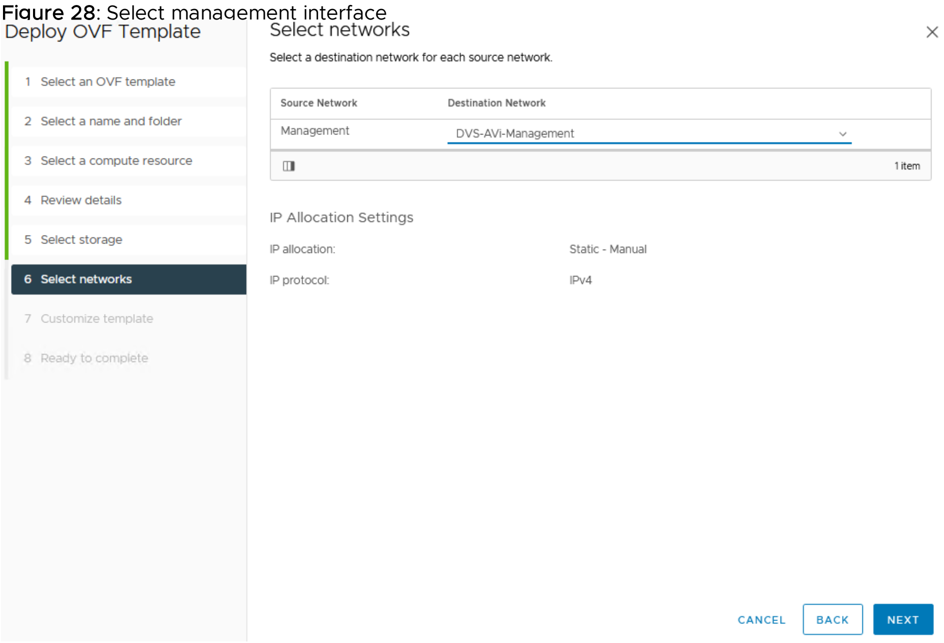
4 items

Compatibility

✓ Compatibility checks succeeded.

4. Select VDS port group that is designated for NSX Advanced Load Balancer management interface. Ensure that the port group to which NSX Advanced Load Balancer is attached, can communicate with port group to which vCenter Server management network resides.

Figure 28: Select management interface



Deploy OVF Template

1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete

Select networks

Select a destination network for each source network.

Source Network	Destination Network
Management	DVS-AVi-Management

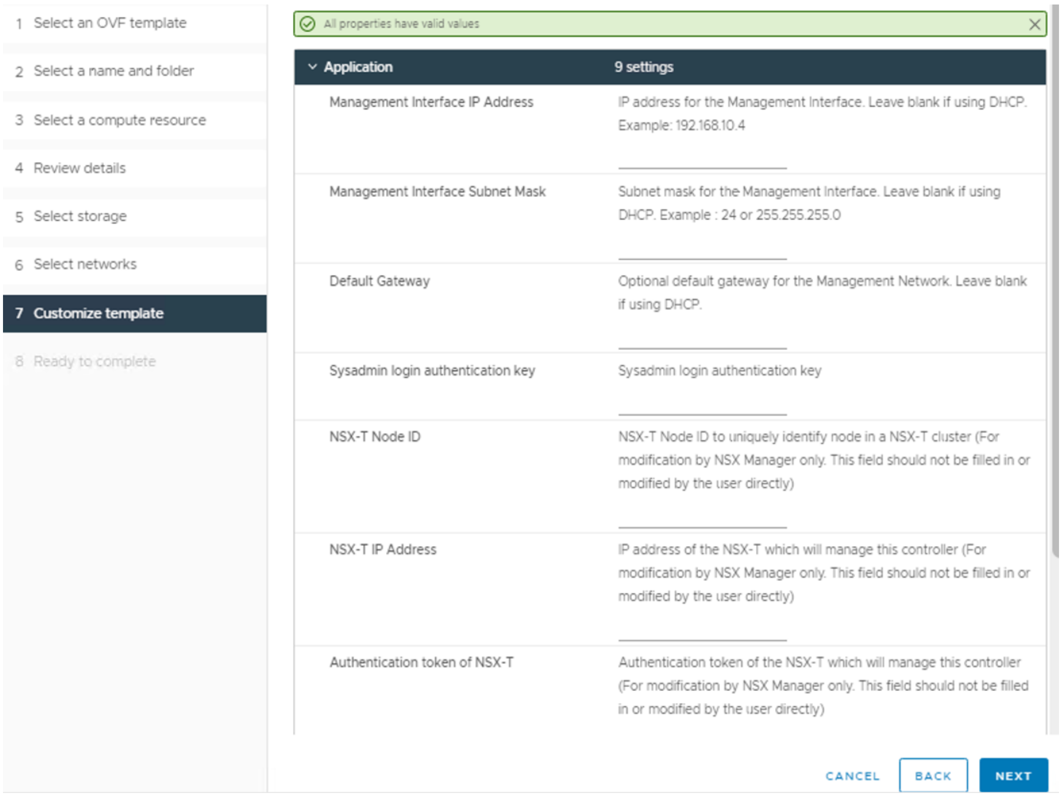
1 item

IP Allocation Settings

IP allocation:	Static - Manual
IP protocol:	IPv4

5. On the next screen enter the required information and proceed to finish on the next screen.

Figure 29: Required information



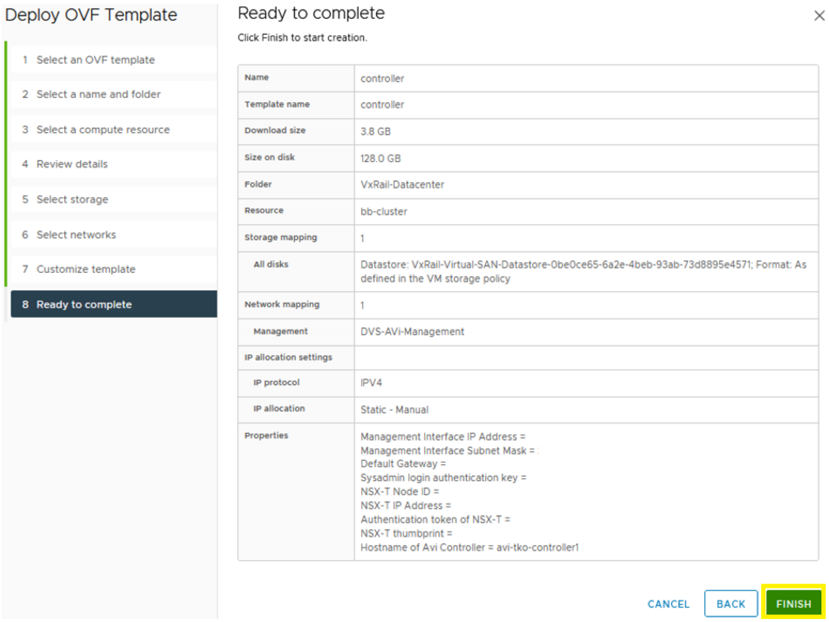
1 Select an OVF template
2 Select a name and folder
3 Select a compute resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete

All properties have valid values

Application — 9 settings

Management Interface IP Address	IP address for the Management Interface. Leave blank if using DHCP. Example: 192.168.10.4
Management Interface Subnet Mask	Subnet mask for the Management Interface. Leave blank if using DHCP. Example : 24 or 255.255.255.0
Default Gateway	Optional default gateway for the Management Network. Leave blank if using DHCP.
Sysadmin login authentication key	Sysadmin login authentication key
NSX-T Node ID	NSX-T Node ID to uniquely identify node in a NSX-T cluster (For modification by NSX Manager only. This field should not be filled in or modified by the user directly)
NSX-T IP Address	IP address of the NSX-T which will manage this controller (For modification by NSX Manager only. This field should not be filled in or modified by the user directly)
Authentication token of NSX-T	Authentication token of the NSX-T which will manage this controller (For modification by NSX Manager only. This field should not be filled in or modified by the user directly)

Figure 30: Complete OVF deployment



Ready to complete

Click Finish to start creation.

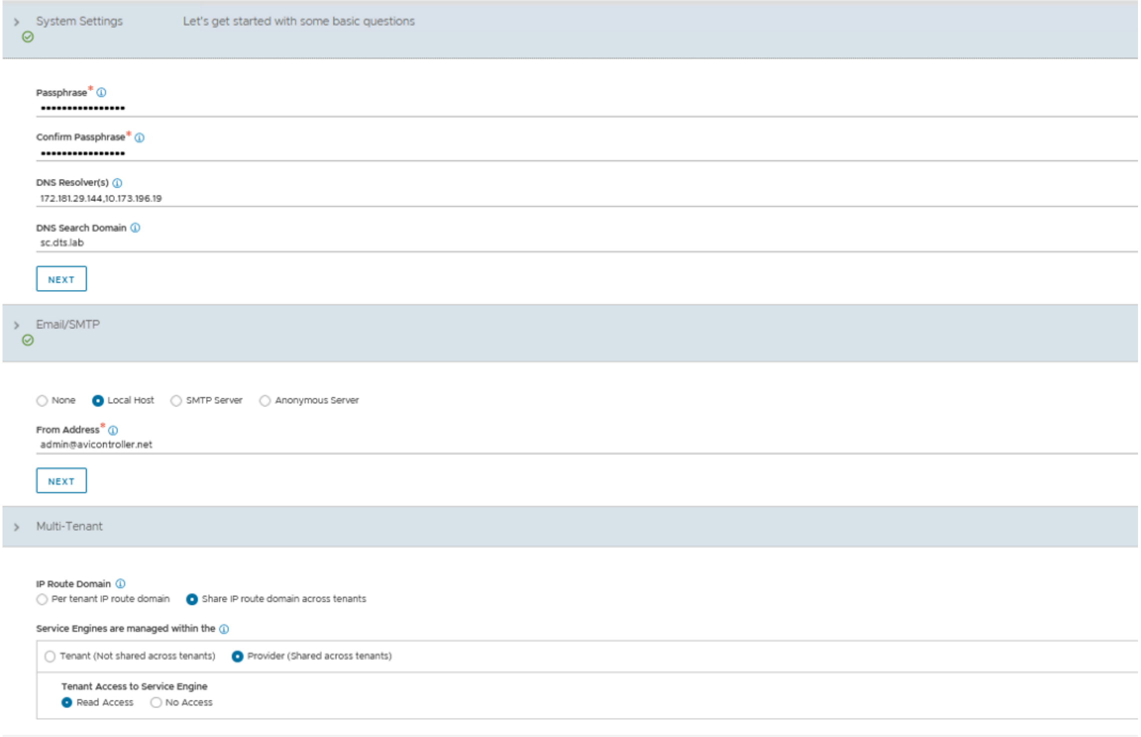
Name	controller
Template name	controller
Download size	3.8 GB
Size on disk	128.0 GB
Folder	VxRail-Datacenter
Resource	bb-cluster
Storage mapping	1
All disks	Datastore: VxRail-Virtual-SAN-Datastore-0be0ce65-6a2e-4beb-93ab-73d8895e4571; Format: As defined in the VM storage policy
Network mapping	1
Management	DVS-AVi-Management
IP allocation settings	
IP protocol	IPV4
IP allocation	Static - Manual
Properties	Management Interface IP Address = Management Interface Subnet Mask = : Default Gateway = Sysadmin login authentication key = NSX-T Node ID = NSX-T IP Address = Authentication token of NSX-T = NSX-T thumbprint = Hostname of Avi Controller = avi-tko-controller1

Login and Initial Configuration

Once the controller is deployed and ready, access the admin portal from a browser using the previously configured hostname or IP address. Please note that it takes a few minutes for the controller to be available for login.

1. On the login screen create a new password and create the admin account
2. On the next screen fill in the system settings including choosing a passphrase, DNS, domain, and SMTP information. Choose your multi-tenant settings and click save. Figure 9 depicts this screen.

Figure 31: Initial login setup



Passphrase* ⓘ
••••••••••••••••

Confirm Passphrase* ⓘ
••••••••••••••••

DNS Resolver(s) ⓘ
172.181.29.144,10.173.196.19

DNS Search Domain ⓘ
sc.dts.lab

○ None ● Local Host ○ SMTP Server ○ Anonymous Server

From Address* ⓘ
admin@avicontroller.net

IP Route Domain ⓘ
○ Per tenant IP route domain ● Share IP route domain across tenants

Service Engines are managed within the ⓘ
○ Tenant (Not shared across tenants) ● Provider (Shared across tenants)

Tenant Access to Service Engine
● Read Access ○ No Access

Controller Configuration

Once the controller is deployed, several tasks need to be performed for NSX Advanced Load Balancer to work with Tanzu Kubernetes Grid Service. These tasks are summarized below.

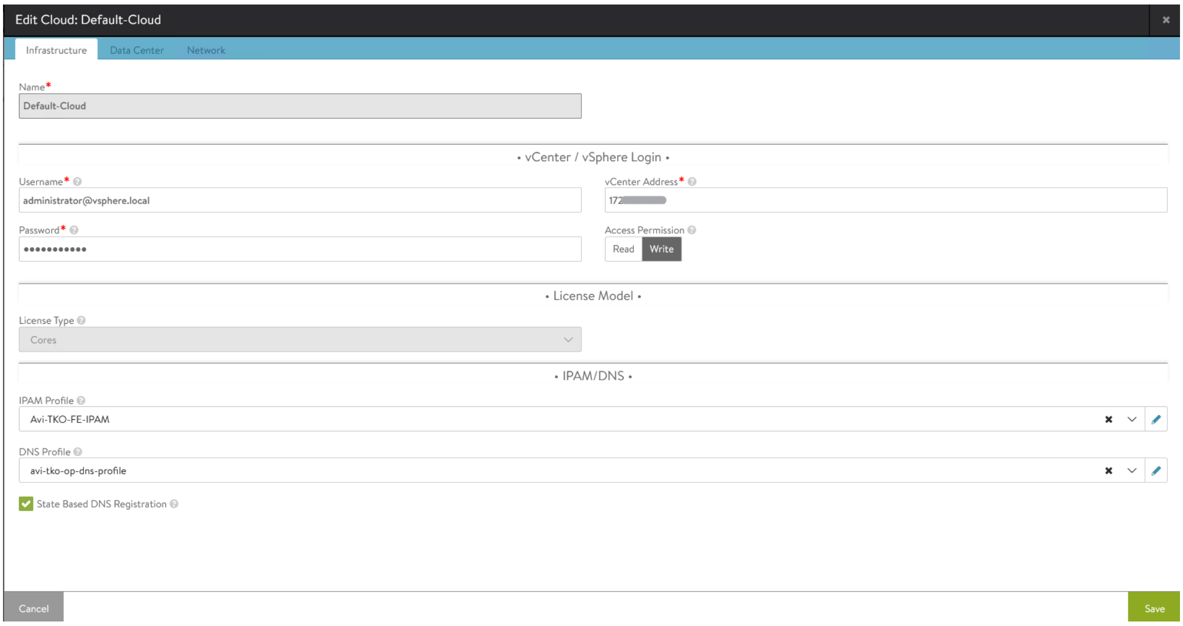
1. Configure the default cloud instance.
2. Configure settings for system access.
3. Configure Service Engine group.
4. Configure the VIP network.
5. Create IPAM Profile
6. Add IPAM to Default-Cloud instance.
7. Create DNS service.
8. Add IPAM and DNS Profile to the default cloud instance.
9. Export the SSL/TLS certificate.
10. Create route between workload and front-end networks.

Configure default cloud instance.

Currently only Default-Cloud instance is supported on vSphere with Tanzu. Access the Default-Cloud settings via Infrastructure ->Clouds and click the pencil icon to edit the cloud configuration.

1. On the Infrastructure tab, fill in the IP address, username, and password for vCenter Server. Ensure that under access permissions “Write” is selected. NSX Advanced Load Balancer requires write permissions to vCenter to creating, modifying, and removing Service Engines or other resources automatically as requirements change.

Figure 32: Add vCenter.



Edit Cloud: Default-Cloud

Name* Default-Cloud

• vCenter / vSphere Login •

Username* administrator@vsphere.local

vCenter Address* 172[redacted]

Password* •••••••••••

Access Permission: Read [✓ Write]

• License Model •

License Type: Cores

• IPAM/DNS •

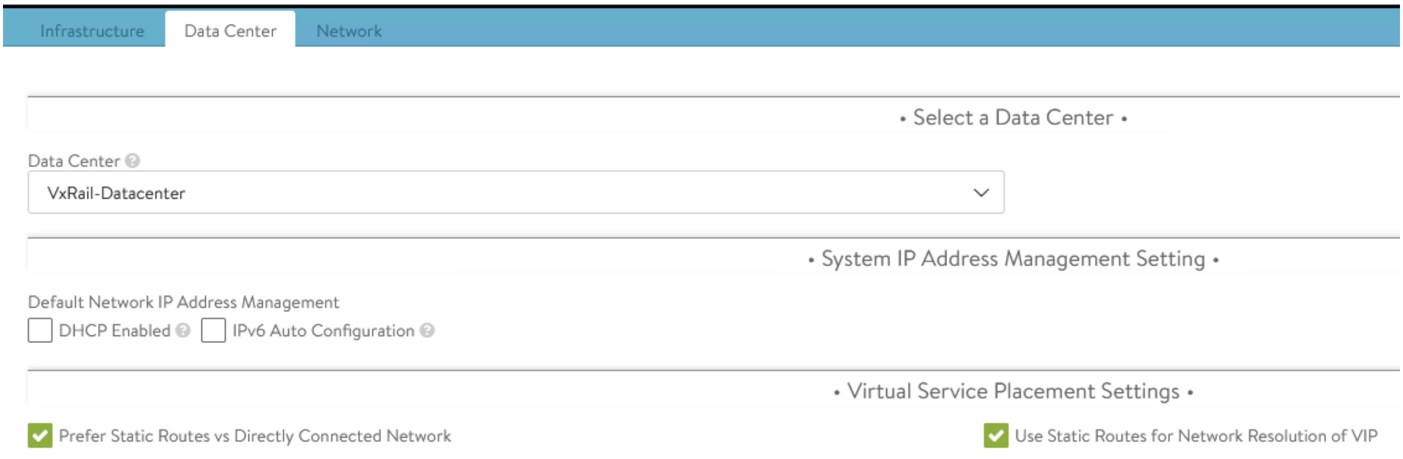
IPAM Profile: Avi-TKO-FE-IPAM

DNS Profile: avi-tko-op-dns-profile

☑ State Based DNS Registration

2. On the Data Center tab select your data center.

Figure 33: Select Data Center



• Select a Data Center •

Data Center: VxRail-Datacenter

• System IP Address Management Setting •

Default Network IP Address Management

□ DHCP Enabled □ IPv6 Auto Configuration

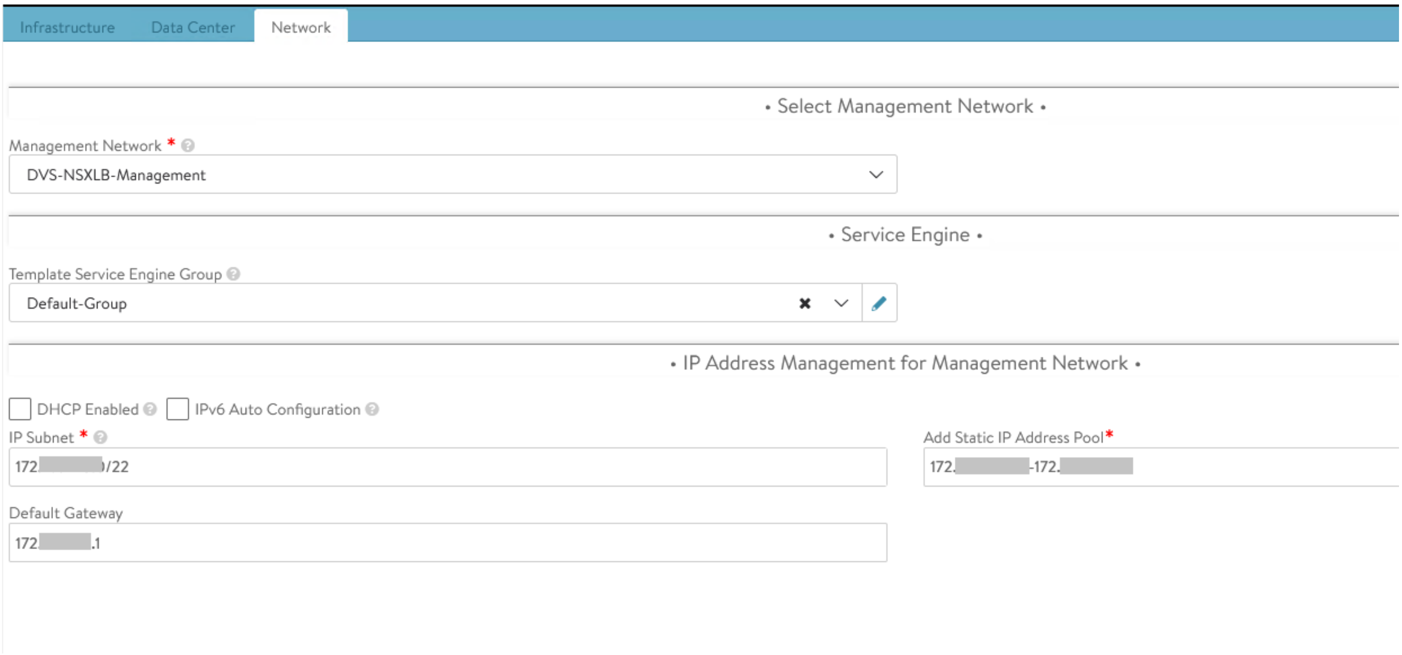
• Virtual Service Placement Settings •

☑ Prefer Static Routes vs Directly Connected Network

☑ Use Static Routes for Network Resolution of VIP

3. On the Network tab, select network designated for NSX Advanced Load Balancer management network, IP subnet, default gateway and an IP range for the static pool.

Figure 34: Select network.



• Select Management Network •

Management Network *: DVS-NSXLB-Management

• Service Engine •

Template Service Engine Group: Default-Group

• IP Address Management for Management Network •

□ DHCP Enabled □ IPv6 Auto Configuration

IP Subnet *: 172[redacted]/22

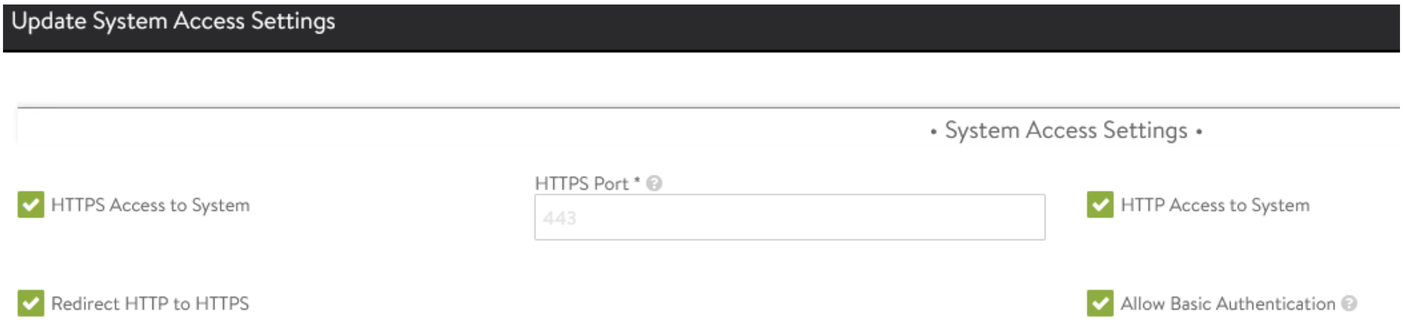
Add Static IP Address Pool*: 172.[redacted]-172.[redacted]

Default Gateway: 172[redacted].1

Configure settings for system access.

1. Basic authentication can be set using the following process. Navigate to Administrator > Settings > Access Settings and check “Allow Basic Authentication.”

Figure 35: Set basic authentication.



Update System Access Settings

• System Access Settings •

☑ HTTPS Access to System

HTTPS Port * 443

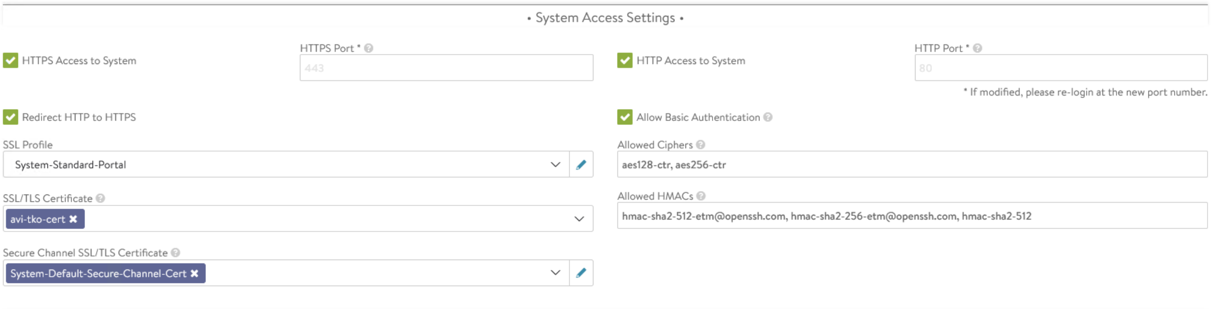
☑ HTTP Access to System

☑ Redirect HTTP to HTTPS

☑ Allow Basic Authentication

Staying on the same screen delete the existing SSL\TLS certificate and create a new one with your specific organization information. The Controller has a default self-signed certificate. But this certificate does not have the correct SAN (Subject Alternate Name). Certificate must be replaced with a valid external or self-signed certificate that has the correct SAN. Step-by-step instructions visit [NSX Advanced Load Balancer documentation]() page.

Figure 36: Create certificate.



• System Access Settings •

☑ HTTPS Access to System

HTTPS Port * 443

☑ HTTP Access to System

HTTP Port * 80

* If modified, please re-login at the new port number.

☑ Redirect HTTP to HTTPS

☑ Allow Basic Authentication

SSL Profile: System-Standard-Portal

Allowed Ciphers: aes128-ctr, aes256-ctr

SSL/TLS Certificate: avi-tko-cert

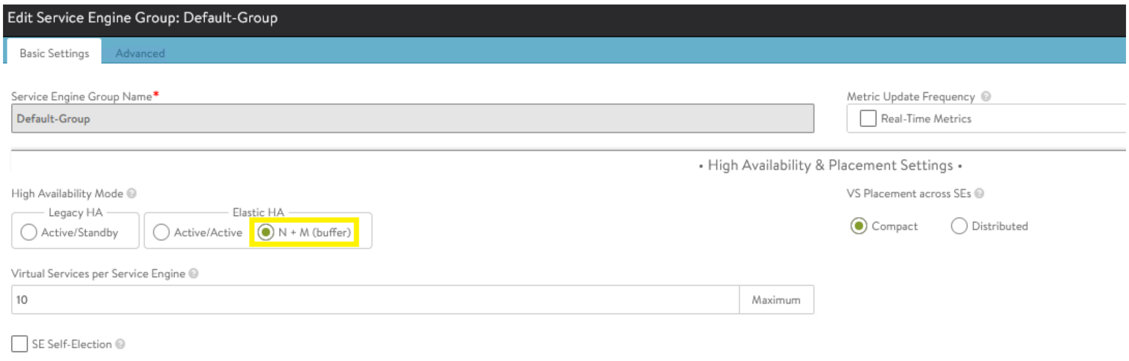
Allowed HMACs: hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512

Secure Channel SSL/TLS Certificate: System-Default-Secure-Channel-Cert

Configure Service Engine group.

1. From Infrastructure > Service Engine Group > Basic Settings, ensure that N+M (buffer) is selected under elastic HA. This is the default mode, where “N” is the minimum number of Service Engines required to place virtual services in a SE group and “M” is the additional Service Engines that the controller spins up to manage Service Engine failures without reducing the capacity of the group.

Figure 37: Create service engine group.



Edit Service Engine Group: Default-Group

Service Engine Group Name*: Default-Group

Metric Update Frequency: □ Real-Time Metrics

• High Availability & Placement Settings •

High Availability Mode

Legacy HA: ○ Active/Standby

Elastic HA: ○ Active/Active ● N + M (buffer)

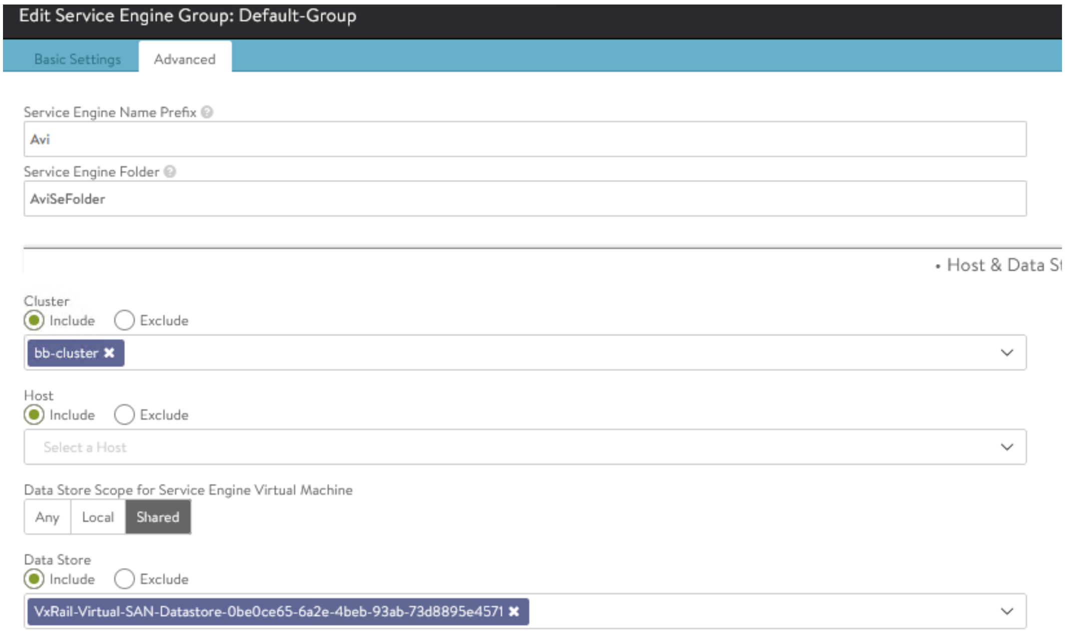
VS Placement across SEs: ● Compact ○ Distributed

Virtual Services per Service Engine: 10 Maximum

□ SE Self-Election

2. In the “Advanced” tab select your cluster and vSAN datastore

Figure 38: Select cluster and vSAN datastore.



Edit Service Engine Group: Default-Group

Service Engine Name Prefix: Avi

Service Engine Folder: AviSeFolder

Cluster: ● Include ○ Exclude

bb-cluster ✖

Host: ● Include ○ Exclude

Select a Host

Data Store Scope for Service Engine Virtual Machine: Any | Local | [✓ Shared]

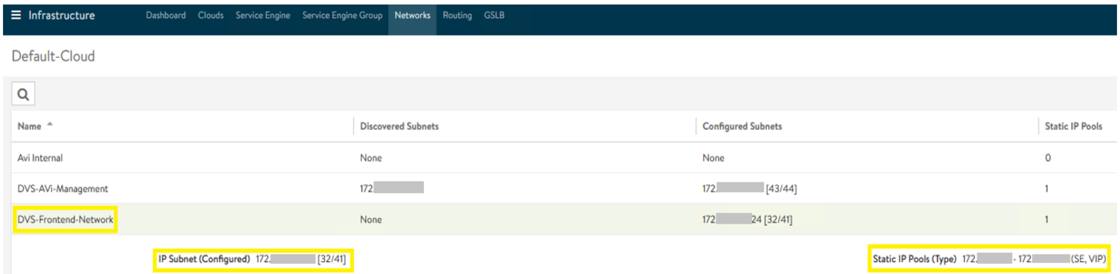
Data Store: ● Include ○ Exclude

VxRail-Virtual-SAN-Datastore-0be0ce65-6a2e-4beb-93ab-73d8895e4571 ✖

Configure the VIP network.

This network is where various Kubernetes control plane and Kubernetes applications require load balancing services. In this case the VIPs reside on the front-end network.

Figure 39: Configure VIP network.



Default-Cloud

Name	Discovered Subnets	Configured Subnets	Static IP Pools
Avi Internal	None	None	0
DVS-AVi-Management	172[redacted]	172.[redacted] [43/44]	1
DVS-Frontend-Network	None	172[redacted]24 [32/41]	1

IP Subnet (Configured) 172.[redacted] [32/41]

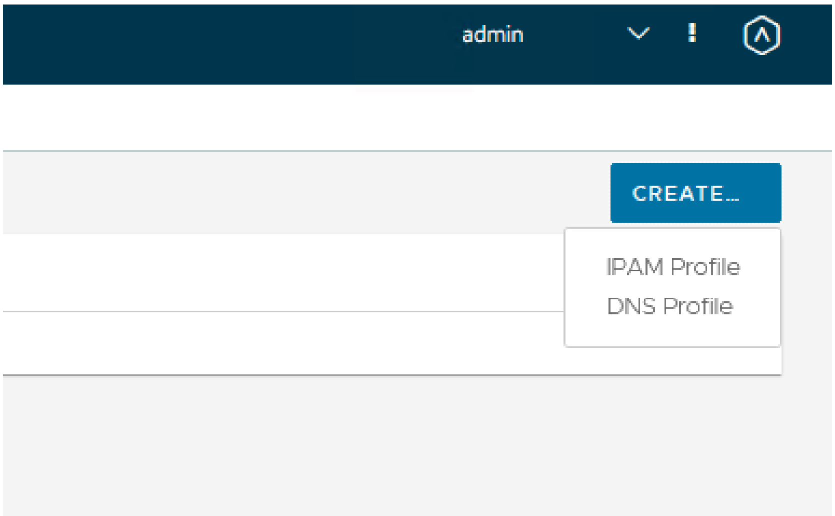
Static IP Pools (Type) 172.[redacted] - 172[redacted](SE, VIP)

Create IPAM Profile

Create IPAM for the VIP network created earlier to assign IPs to the virtual services. This can be accessed via Templates > Profiles > IPAM/DNS Profiles.

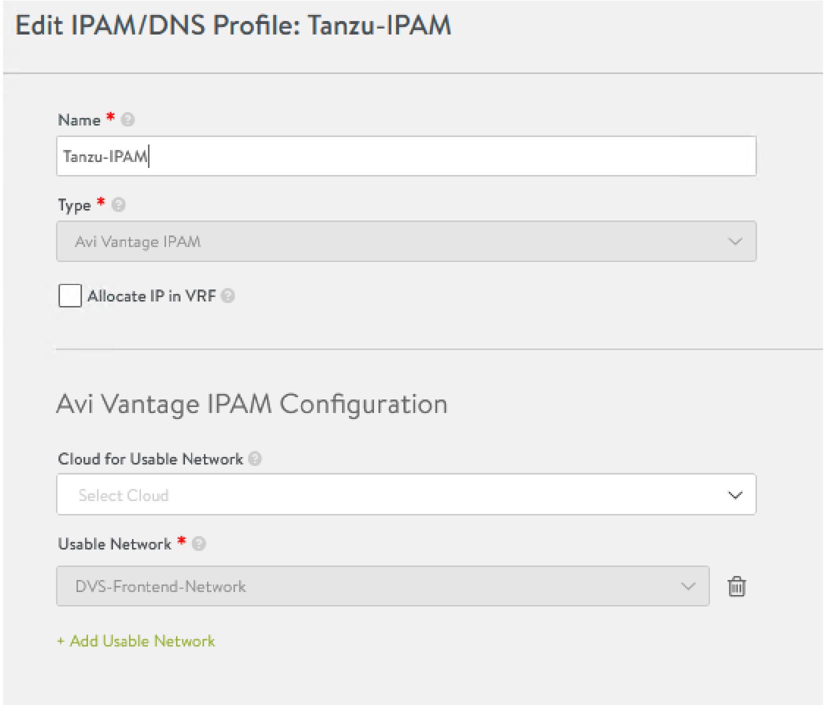
1. Create a new IPAM Profile by using the Create button on the top right-hand side of the screen.

Figure 40: Create IPAM



2. Enter a name. In the Type field, select Avi Vantage IPAM, and add a usable network which will be your VIP network.

Figure 41: Create IPAM


Edit IPAM/DNS Profile: Tanzu-IPAM

Name * Tanzu-IPAM

Type * Avi Vantage IPAM

□ Allocate IP in VRF

Avi Vantage IPAM Configuration

Cloud for Usable Network: Select Cloud

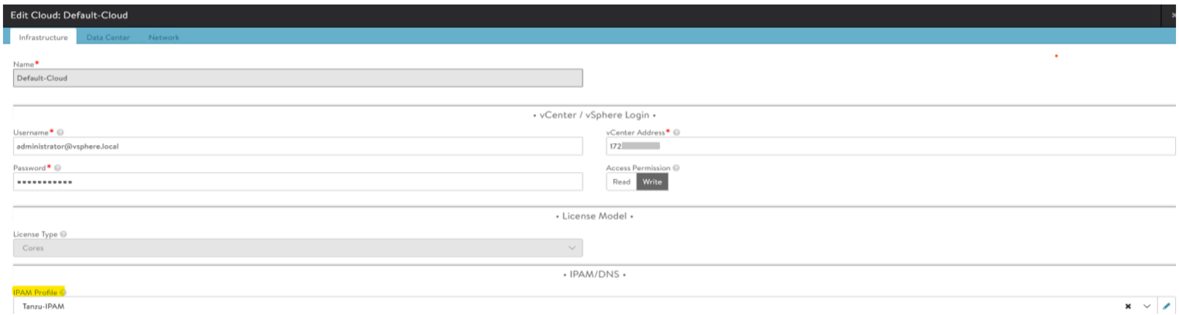
Usable Network * DVS-Frontend-Network

Add IPAM to Default-Cloud instance.

The new IPAM needs to be added to the default-cloud instance.

1. Navigate to Infrastructure > Default-Cloud and edit. Select the newly created IPAM profile from the dropdown list.

Figure 42: Add IPAM to Default-Cloud



Edit Cloud: Default-Cloud

Infrastructure | Data Center | Network

Name*: Default-Cloud

• vCenter / vSphere Login •

Username*: administrator@vsphere.local

vCenter Address*: 172[redacted]

Password*: ●●●●●●●●●●●●

Access Permission: Read | Write

• License Model •

License Type: Cores

• IPAM/DNS •

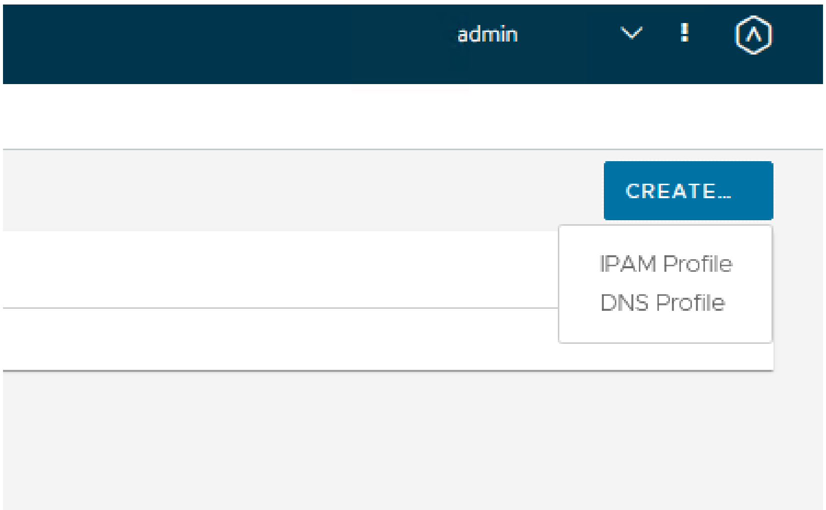
IPAM Profile: Tanzu-IPAM

Create DNS Profile

Create DNS profile via Templates > Profiles > IPAM/DNS Profiles.

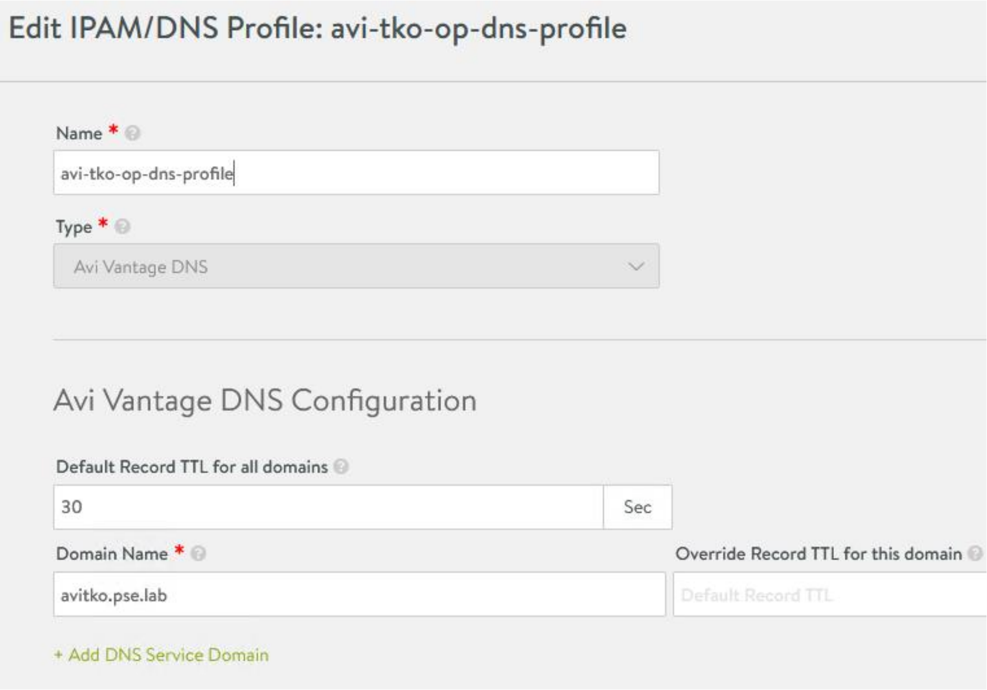
3. Create a new DNS Profile by using the Create button on the top right-hand side of the screen.

Figure 43: Create DNS profile.



4. Enter a name. In the “Type” field, select Avi Vantage DNS, and add the desired sub-domain. This subdomain will be delegated to NSX Advanced Load Balancer.

Figure 44: Create DNS profile.



Edit IPAM/DNS Profile: avi-tko-op-dns-profile

Name *: avi-tko-op-dns-profile

Type *: Avi Vantage DNS

Avi Vantage DNS Configuration

Default Record TTL for all domains: 30 Sec

Domain Name *: avitko.pse.lab

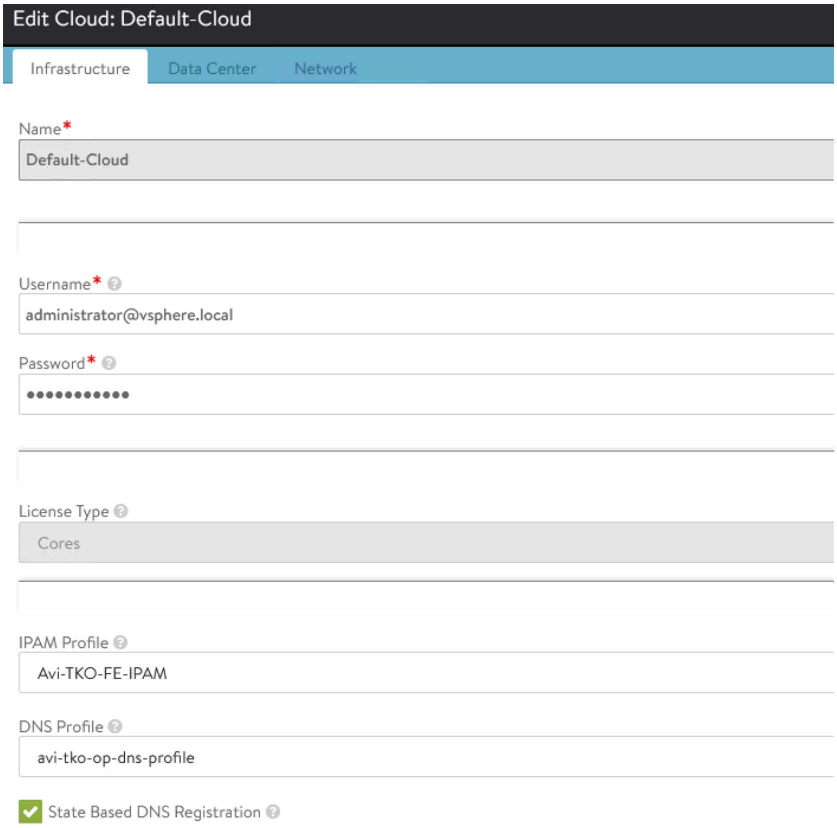
Override Record TTL for this domain: Default Record TTL

Add IPAM and DNS profile to Default-Cloud instance.

The new IPAM and DNS profile needs to be added to the default-cloud instance.

2. Navigate to Infrastructure > Default-Cloud and edit. Select the newly created IPAM and DNS profile from the dropdown list.

Figure 45: Add IPAM and DNS profiles to Default-Cloud



Edit Cloud: Default-Cloud

Name*: Default-Cloud

Username*: administrator@vsphere.local

Password*: ●●●●●●●●●●●

License Type: Cores

IPAM Profile: Avi-TKO-FE-IPAM

DNS Profile: avi-tko-op-dns-profile

☑ State Based DNS Registration

Create DNS service.

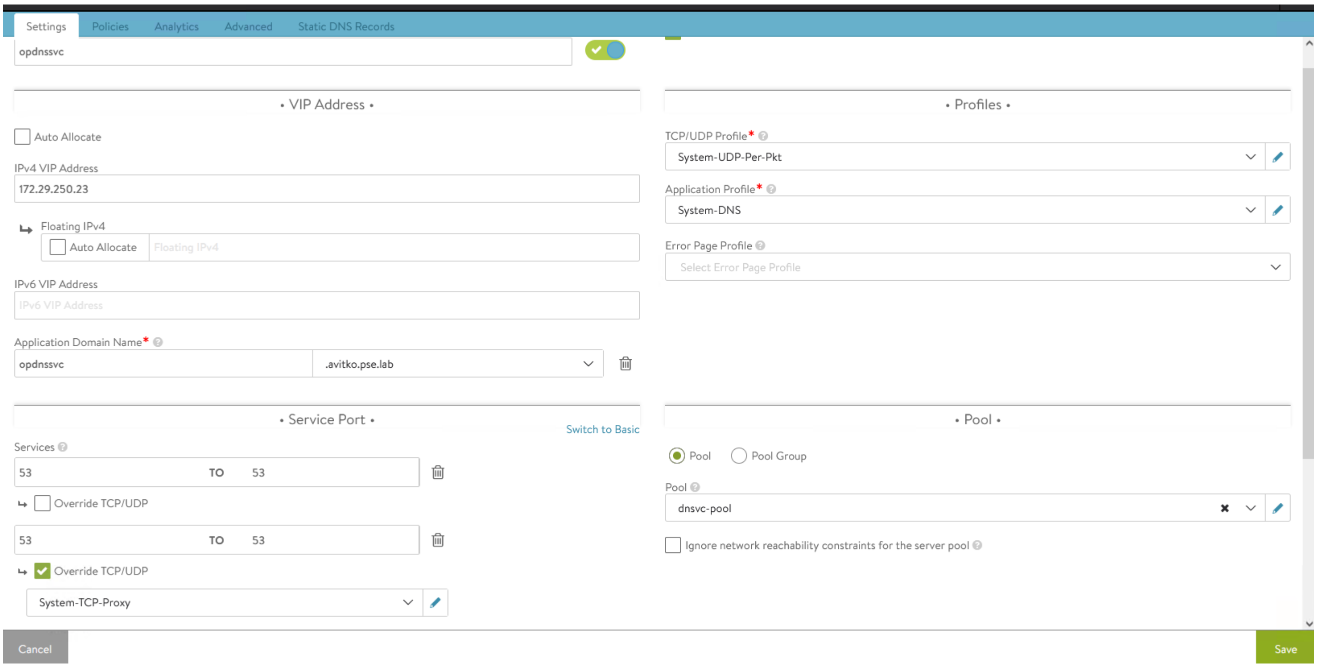
NSX Advanced Load Balancer provides generic DNS virtual services that can be implemented with various functionalities to meet different requirements. The DNS virtual service can be used to load balance DNS servers, hosting static DNS entries, virtual service IP address DNS hosting or hosting GSLB service DNS entries. For more information on NSX Advanced Load Balancer features please visit NSX Advanced Load Balancer features.

For this reference architecture a DNS virtual service was created that served as DNS sever for a subdomain of the primary Active Directory domain via DNS delegation. This generic process is outlined below. With this DNS configuration along with IPAM, a DNS entry will be created for services. Delegation of DNS domain will depend on your Active Directory architecture. The domain delegation process for this reference architecture is described later in this document under DNS Domain Delegation.

1. Create DNS virtual service.

   To create DNS virtual service, navigate to Applications > Virtual Services > Create Virtual Service. Give the service a name and select TCP/UDP and application profile. Application profile is of type "System DNS." Click save.

Figure 46: Create DNS service.



opdnssvc

• VIP Address •

□ Auto Allocate

IPv4 VIP Address: 172.29.250.23

Floating IPv4: □ Auto Allocate | Floating IPv4

IPv6 VIP Address: IPv6 VIP Address

Application Domain Name*: opdnssvc | .avitko.pse.lab

• Service Port •

Services

53 TO 53

□ Override TCP/UDP

53 TO 53

☑ Override TCP/UDP

System-TCP-Proxy

• Profiles •

TCP/UDP Profile*: System-UDP-Per-Pkt

Application Profile*: System-DNS

Error Page Profile: Select Error Page Profile

• Pool •

● Pool ○ Pool Group

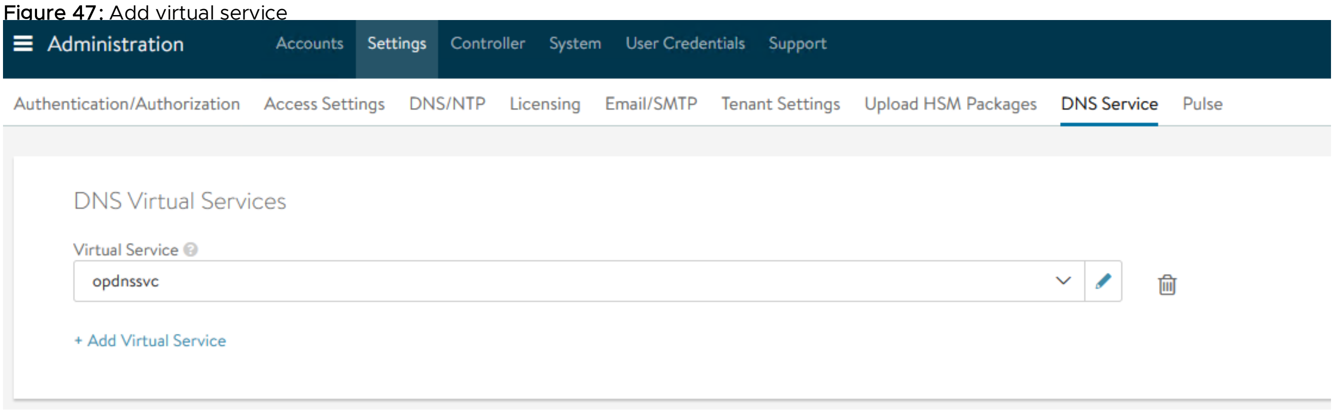
Pool: dnsvc-pool

□ Ignore network reachability constraints for the server pool

2. Add DNS virtual service to Default-Cloud instance.

   Navigate to Administrator > Settings > DNS Service and select virtual DNS service.

Figure 47: Add virtual service



DNS Virtual Services

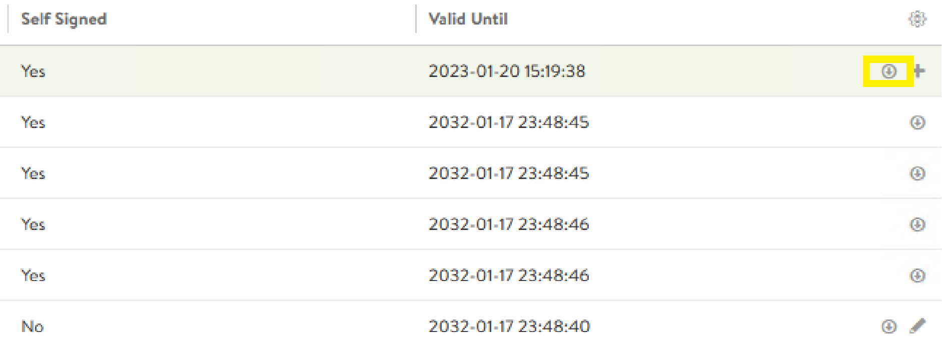
Virtual Service: opdnssvc

Export the SSL/TLS certificate.

The certificate created in the earlier steps will be needed during Tanzu Workload Management deployment. Following these steps, export the certificate to be used later.

1. Go to Templates > Security and select the certificate created earlier.
2. Click the down arrow on the right-hand side to export the certificate.

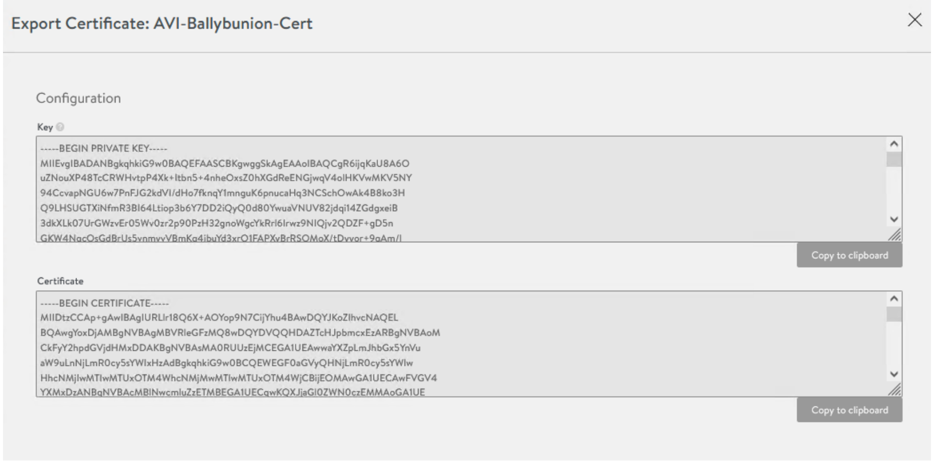
Figure 48: Export certificate



Self Signed	Valid Until
Yes	2023-01-20 15:19:38
Yes	2032-01-17 23:48:45
Yes	2032-01-17 23:48:45
Yes	2032-01-17 23:48:46
Yes	2032-01-17 23:48:46
No	2032-01-17 23:48:40

3. Copy the certificate to clipboard.

Figure 49: Copy certificate



Export Certificate: AVI-Ballybunion-Cert

Configuration

Key

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCgR6ijqKaU8A6O
uZNouXP48TcCRWHvtpP4Xk+ltbn5+4nheOxsZ0hXGdReENGjwqV4olHKVwMKV5NY
94CcvapNGU6w7PnFJG2kdVI/dHo7fknqY1mnguK6pnucaHq3NCSchOwAk4B8ko3H
Q9LHSUGTXiNfmR3BI64Ltiop3b6Y7DD2iQyQ0d80YwuaVNUV82jdqi14ZGdgxeiB
3dkXLk07UrGWzvEr05Wv0zr2p90PzH32gnoWgcYkRrl6lrwz9NIQjv2QDZF+gD5n
GKW4NqcOsGdBrUs5vnmvvVBmKq4ibuYd3xrO1FAPXvBrRSOMoX/tDvvor+9qAm/l
```

Certificate

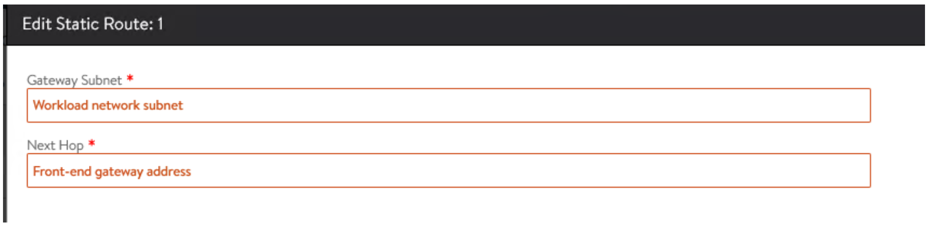
```
-----BEGIN CERTIFICATE-----
MIIDtzCCAp+gAwIBAgIURLlr18Q6X+AOYop9N7CijYhu4BAwDQYJKoZIhvcNAQEL
BQAwgYoxDjAMBgNVBAgMBVRleGFzMQ8wDQYDVQQHDAZTcHJpbmcxEzARBgNVBAoM
CkFyY2hpdGVjdHMxDDAKBgNVBAsMA0RUUzEjMCEGA1UEAwwaYXZpLmJhbGx5YnVu
aW9uLnNjLmR0cy5sYWIxHzAdBgkqhkiG9w0BCQEWEGF0aGVyQHNjLmR0cy5sYWIw
HhcNMjIwMTIwMTUxOTM4WhcNMjMwMTIwMTUxOTM4WjCBijEOMAwGA1UECAwFVGV4
YXMxDzANBgNVBAcMBlNwcmluZzETMBEGA1UECgwKQXJjaGl0ZWN0czEMMAoGA1UE
```

Create route between workload and front-end networks.

If the VIP and workloads are on separate networks, as in the case here, a route needs to be created between the front-end and workload networks.

1. Navigate to Infrastructure > Routing > Static Route tab.
2. Create a new static route.

Figure 50: Create route.



Edit Static Route: 1

Gateway Subnet *
Workload network subnet

Next Hop *
Front-end gateway address

Enable vSphere with Tanzu Workload Management

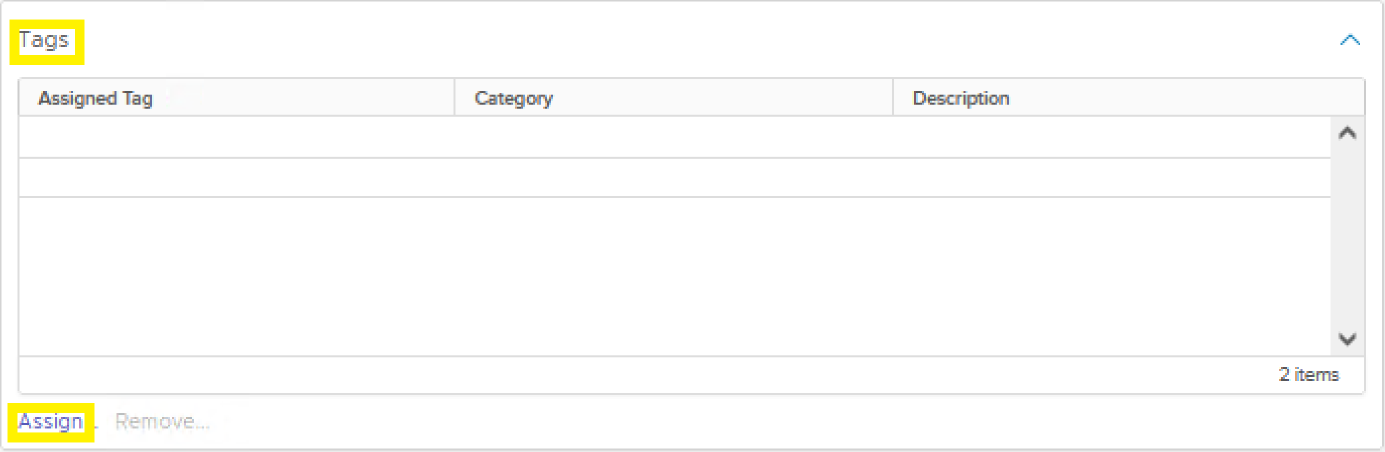
Once NSX Advanced Load Balancer has been successfully deployed, vSphere with Tanzu Workload Management can be enabled. As a best practice, a Tanzu specific storage policy needs to be defined and storage tagged prior to enabling Workload Management. This storage policy should be different than the default vSAN storage policy that is initially configured by the VxRail HCI System software during initial VxRail cluster creation. For vSphere with Tanzu use cases, storage policies and tags are used to assign storage to Kubernetes cluster nodes and persistent volumes. Using a separate and dedicated storage policy ensures that Tanzu workloads are placed on the desired storage pool, separate from other vSphere workloads. The process for doing this is described below.

Note: Dell VxRail provides options for cluster vCenter Server deployment configurations. VxRail can deploy a vCenter Server that is hosted on the deployed cluster and managed by VxRail HCI System Software as part of Its cluster lifecycle management capabilities. This Is referred to as a VxRail managed vCenter on cluster deployment configuration. The other option that is available is to use an existing, customer managed vCenter Server that can be used to manage the Dell VxRail cluster deployment. In this configuration, vCenter server lifecycle management would be the responsibility of the customer to manage. Please consult VxRail documentation for more info. For this reference architecture document, a VxRail managed on cluster vCenter Server deployment configuration was used.

Create storage tag and policy.

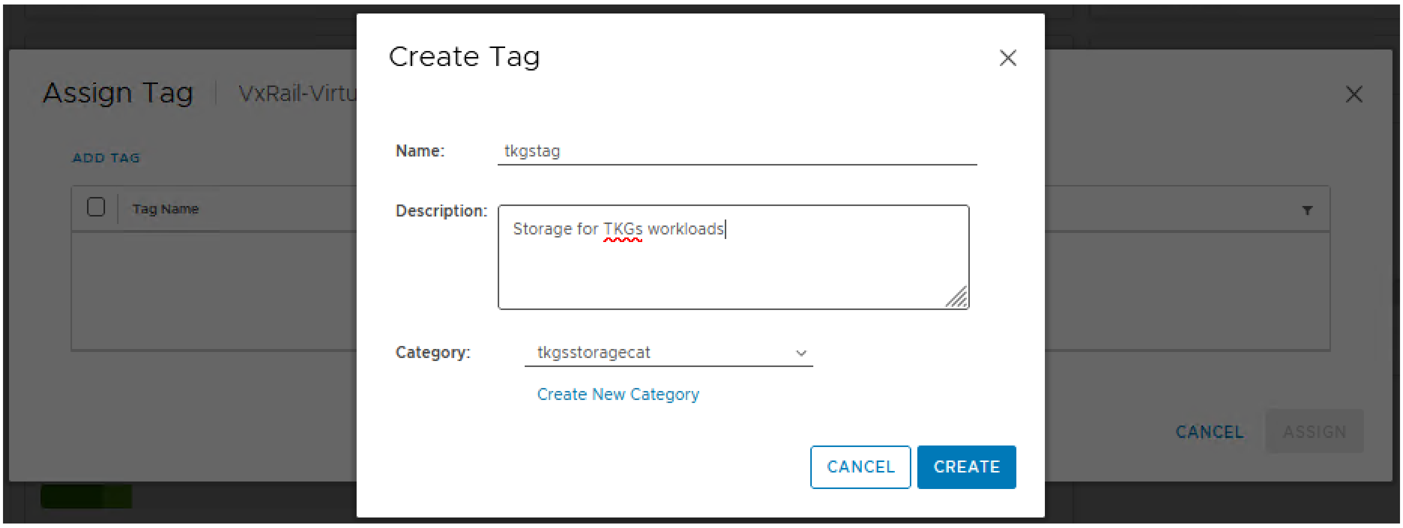
1. Select your cluster in vCenter Server and go to Datastores. Select the Datastore to be used for Workload Management.
2. Under "Tags" click "Assign."

Figure 51: Create tag



3. On the next screen click "ADD TAG" and Create Tag dialog opens. Enter a name for the tag and select a category. If desired a new category can be created. Click CREATE to create the tag.

Figure 52: Create tag.



Create Tag

Name: tkgstag

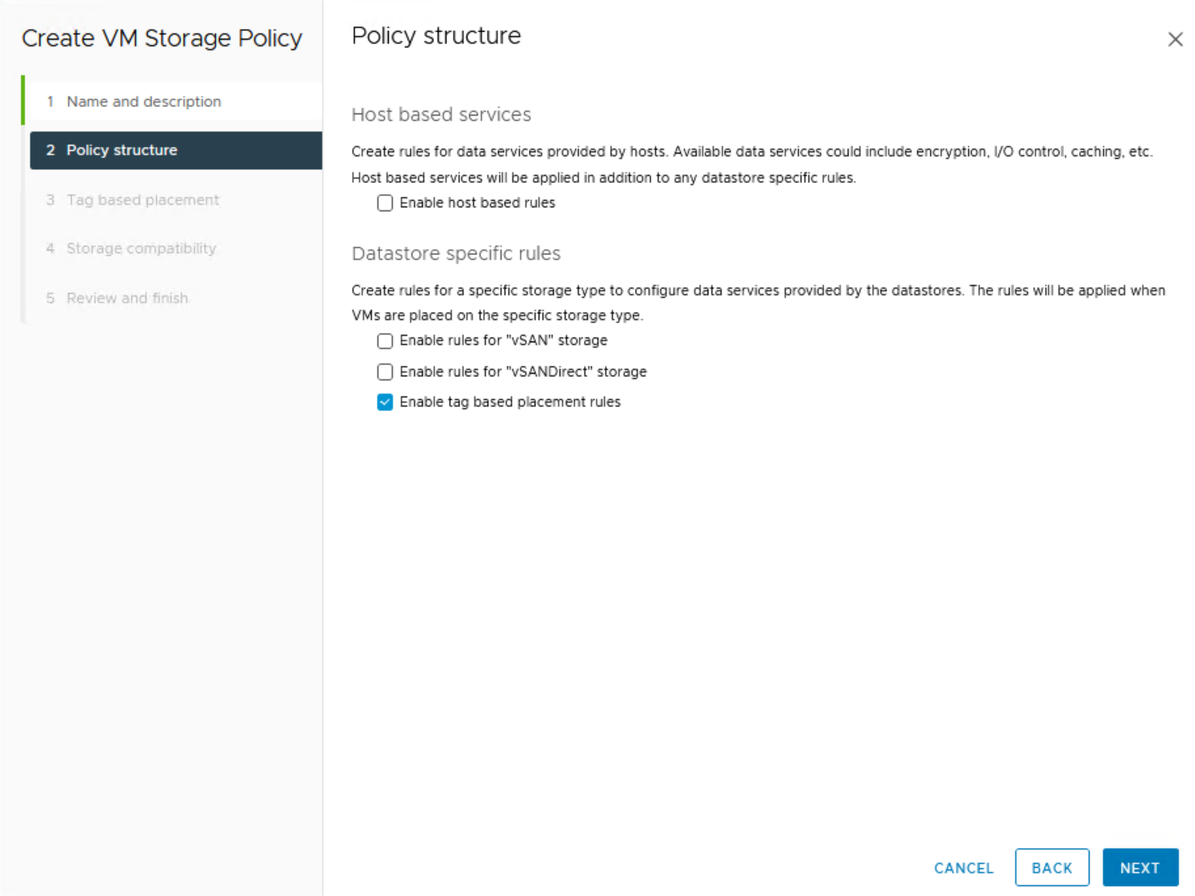
Description: Storage for TKGs workloads

Category: tkgsstoragecat

4. In vCenter Server navigate to Menu > Policies and Profiles > VM Storage Policies and select CREATE. On the next screen give the policy a name and click NEXT.

5. For Policy Structure, check the "Enable tag-based placement rules" and click NEXT.

Figure 53: Placement Rules



Create VM Storage Policy

1 Name and description
2 Policy structure
3 Tag based placement
4 Storage compatibility
5 Review and finish

Policy structure

Host based services

Create rules for data services provided by hosts. Available data services could include encryption, I/O control, caching, etc. Host based services will be applied in addition to any datastore specific rules.

□ Enable host based rules

Datastore specific rules

Create rules for a specific storage type to configure data services provided by the datastores. The rules will be applied when VMs are placed on the specific storage type.

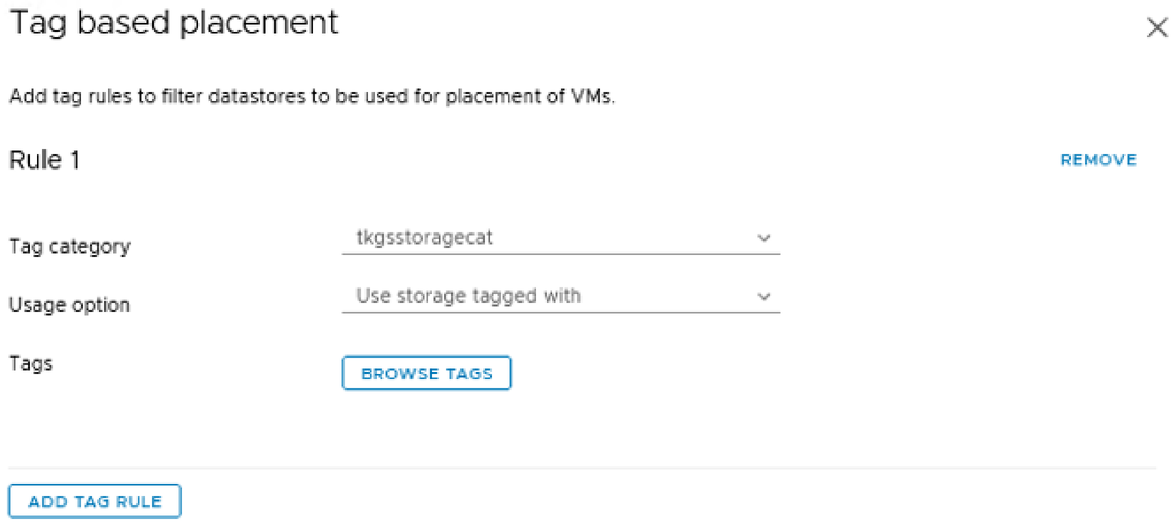
□ Enable rules for "vSAN" storage

□ Enable rules for "vSANDirect" storage

☑ Enable tag based placement rules

6. Create a rule by selecting the tag category, and "Use storage tagged with" as usage option. Browse and select the tag created earlier. Click NEXT.

Figure 54: Create rule



Tag based placement

Add tag rules to filter datastores to be used for placement of VMs.

Rule 1

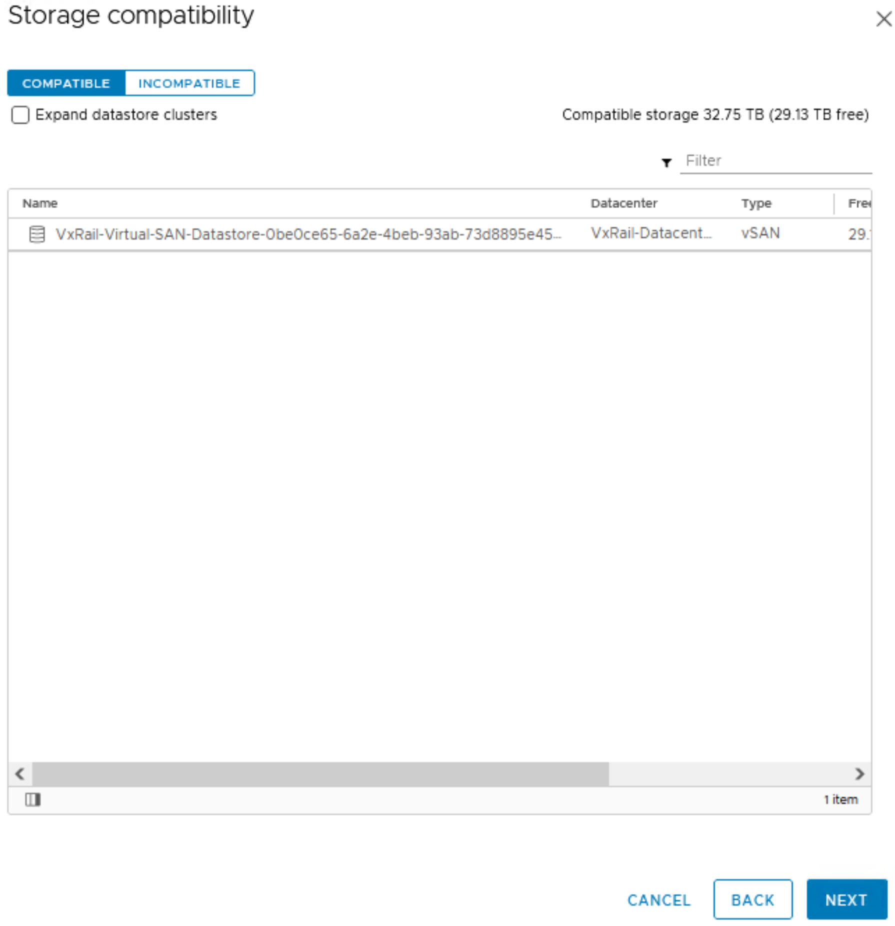
Tag category: tkgsstoragecat

Usage option: Use storage tagged with

Tags

7. Select storage and click next and finish the policy creation.

Figure 55: Select storage



Storage compatibility

☐ Expand datastore clusters

Compatible storage 32.75 TB (29.13 TB free)

Name	Datacenter	Type	Free
VxRail-Virtual-SAN-Datastore-0be0ce65-6a2e-4beb-93ab-73d8895e45...	VxRail-Datacent...	vSAN	29.

1 item

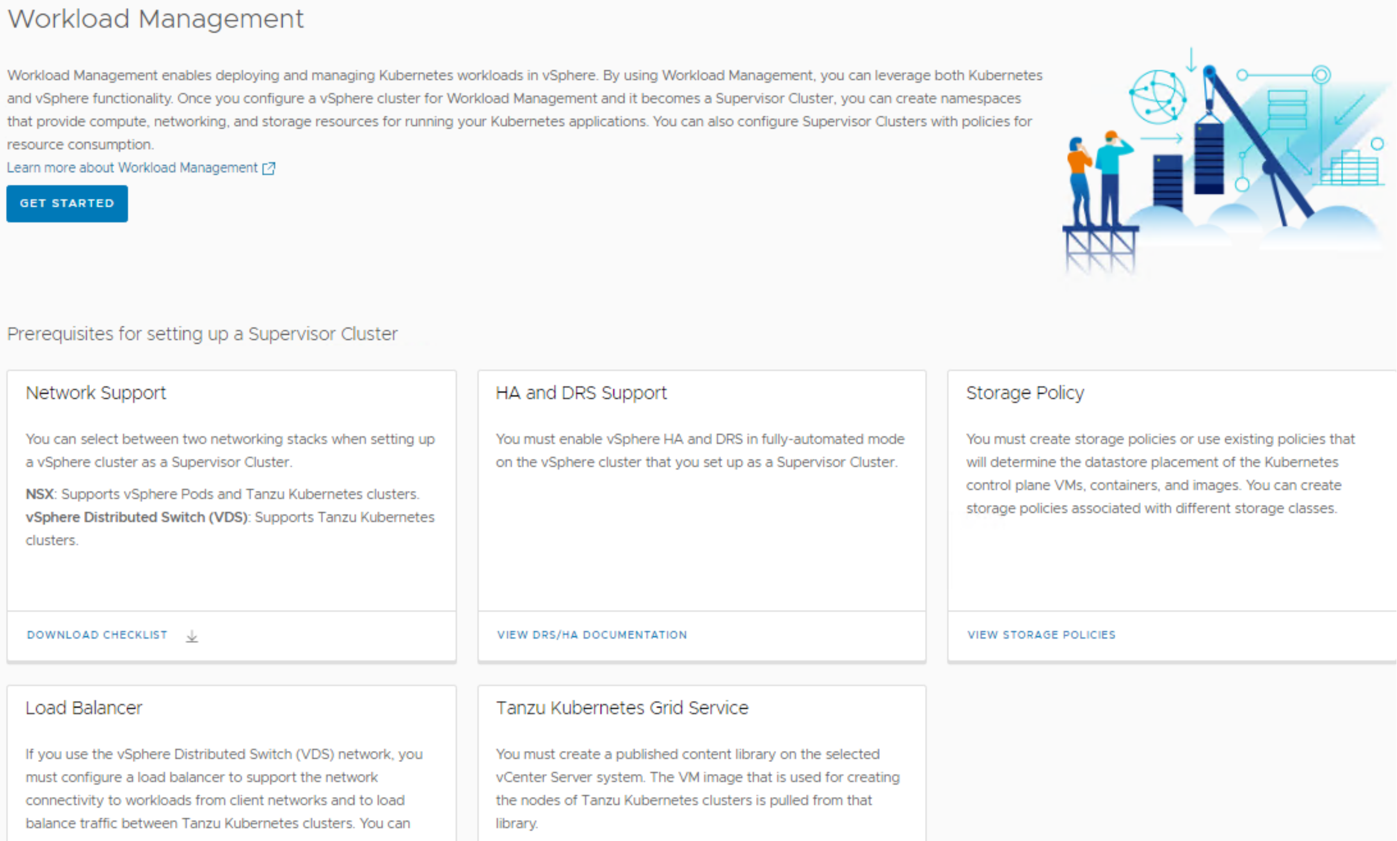
Create content library.

A content library is required by Tanzu Kubernetes Grid Service that will hold the images required by vSphere to deploy Supervisor and workload clusters. The subscription URL used for this content library is https://wp-content.vmware.com/v2/latest/lib.json. Create a content library with the given subscription URL. Please note that it will take some time before content is downloaded and available in the library.

Enable Workload Management

1. Navigate to Menu > Workload Management. Review the prerequisites for setting up Supervisor cluster and ensure that they are met before proceeding. Click "Get Started."

Figure 56: Enable Workload Management



Workload Management

Workload Management enables deploying and managing Kubernetes workloads in vSphere. By using Workload Management, you can leverage both Kubernetes and vSphere functionality. Once you configure a vSphere cluster for Workload Management and it becomes a Supervisor Cluster, you can create namespaces that provide compute, networking, and storage resources for running your Kubernetes applications. You can also configure Supervisor Clusters with policies for resource consumption.

Learn more about Workload Management

Prerequisites for setting up a Supervisor Cluster

Network Support

You can select between two networking stacks when setting up a vSphere cluster as a Supervisor Cluster.

NSX: Supports vSphere Pods and Tanzu Kubernetes clusters.

vSphere Distributed Switch (VDS): Supports Tanzu Kubernetes clusters.

HA and DRS Support

You must enable vSphere HA and DRS in fully-automated mode on the vSphere cluster that you set up as a Supervisor Cluster.

Storage Policy

You must create storage policies or use existing policies that will determine the datastore placement of the Kubernetes control plane VMs, containers, and images. You can create storage policies associated with different storage classes.

Load Balancer

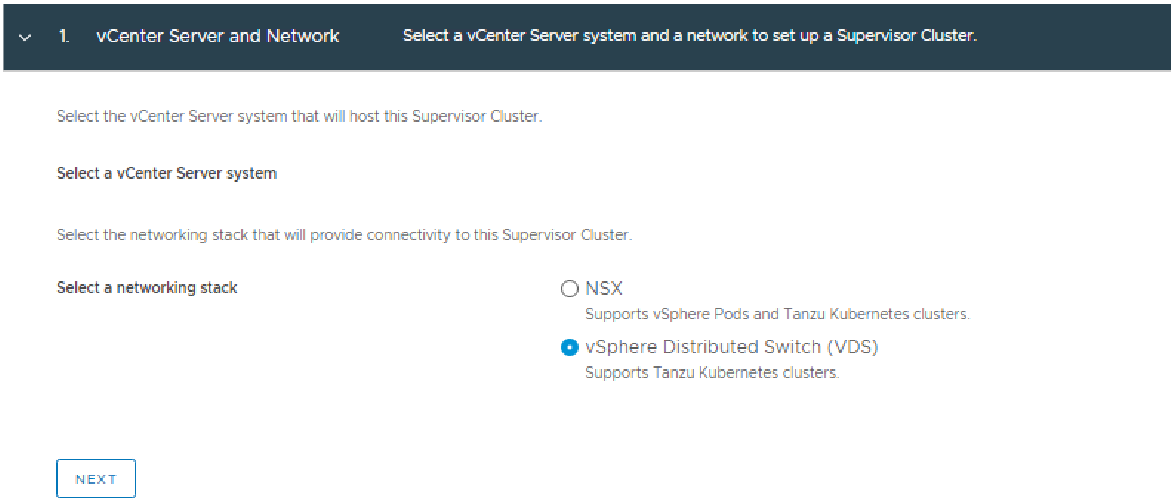
If you use the vSphere Distributed Switch (VDS) network, you must configure a load balancer to support the network connectivity to workloads from client networks and to load balance traffic between Tanzu Kubernetes clusters. You can

Tanzu Kubernetes Grid Service

You must create a published content library on the selected vCenter Server system. The VM image that is used for creating the nodes of Tanzu Kubernetes clusters is pulled from that library.

2. Select vSphere Distributed Switch and click next.

Figure 57: vCenter Server and Network



1. vCenter Server and Network — Select a vCenter Server system and a network to set up a Supervisor Cluster.

Select the vCenter Server system that will host this Supervisor Cluster.

Select a vCenter Server system

Select the networking stack that will provide connectivity to this Supervisor Cluster.

Select a networking stack

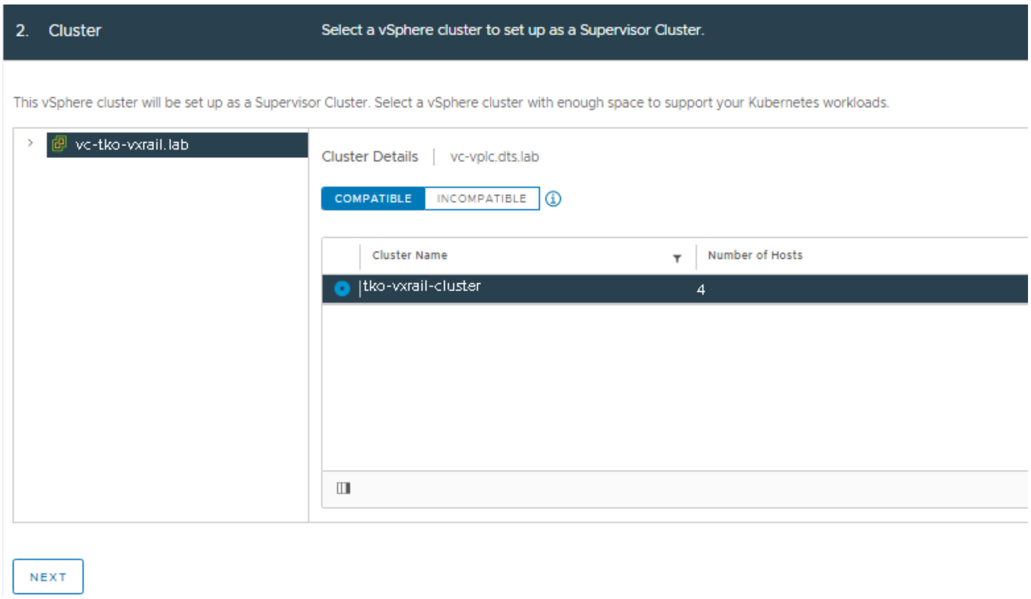
○ NSX
Supports vSphere Pods and Tanzu Kubernetes clusters.

● vSphere Distributed Switch (VDS)
Supports Tanzu Kubernetes clusters.

NEXT

3. Select the cluster.

Figure 58: Select cluster.



2. Cluster — Select a vSphere cluster to set up as a Supervisor Cluster.

This vSphere cluster will be set up as a Supervisor Cluster. Select a vSphere cluster with enough space to support your Kubernetes workloads.

vc-tko-vxrail.lab

Cluster Details | vc-vplc.dts.lab

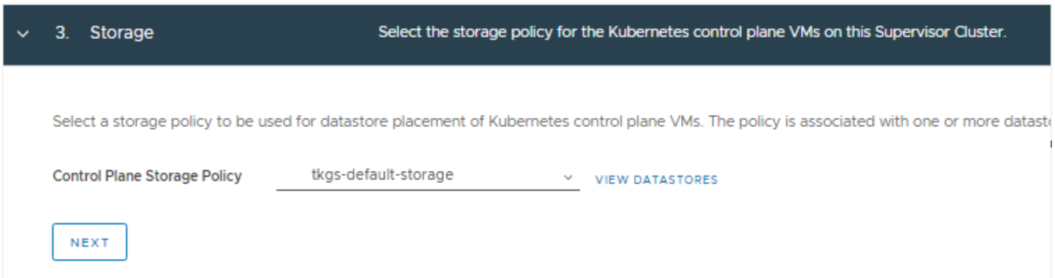
COMPATIBLE | INCOMPATIBLE

	Cluster Name	Number of Hosts
●	tko-vxrail-cluster	4

NEXT

4. Select the storage policy created previously.

Figure 59: Storage Policy



3. Storage — Select the storage policy for the Kubernetes control plane VMs on this Supervisor Cluster.

Select a storage policy to be used for datastore placement of Kubernetes control plane VMs. The policy is associated with one or more datast

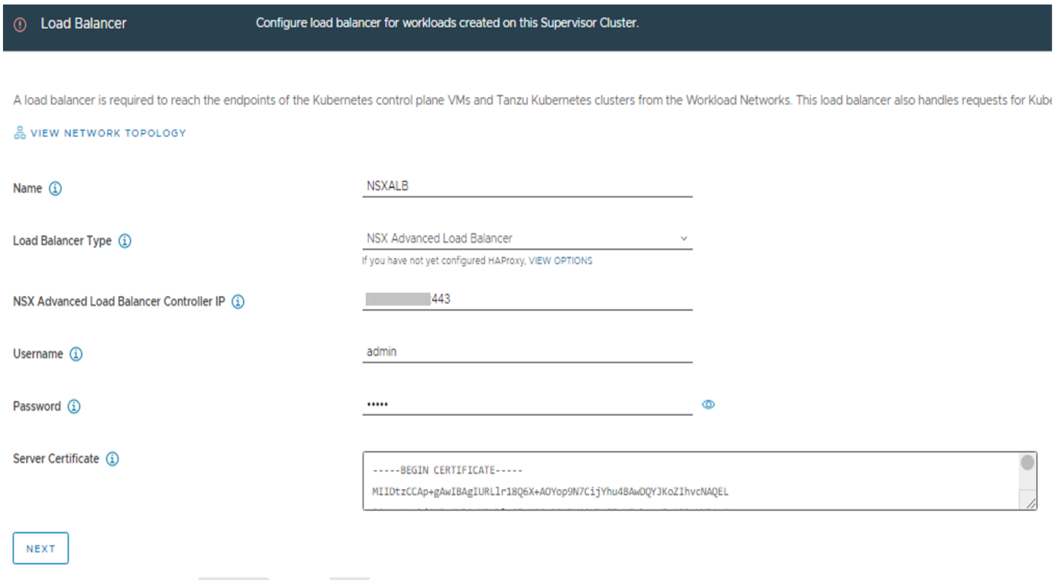
Control Plane Storage Policy: tkgs-default-storage

VIEW DATASTORES

NEXT

5. On the next screen fill out the NSX Advanced Load Balancer details and copy and paste the certificate exported earlier. Ensure to use "< IP address>:443 " format when entering the IP address for the controller. Click NEXT.

Figure 60: NSX Advanced Load Balancer details



Load Balancer — Configure load balancer for workloads created on this Supervisor Cluster.

A load balancer is required to reach the endpoints of the Kubernetes control plane VMs and Tanzu Kubernetes clusters from the Workload Networks. This load balancer also handles requests for Kub

VIEW NETWORK TOPOLOGY

Name: NSXALB

Load Balancer Type: NSX Advanced Load Balancer
If you have not yet configured HAProxy, VIEW OPTIONS

NSX Advanced Load Balancer Controller IP: [redacted]:443

Username: admin

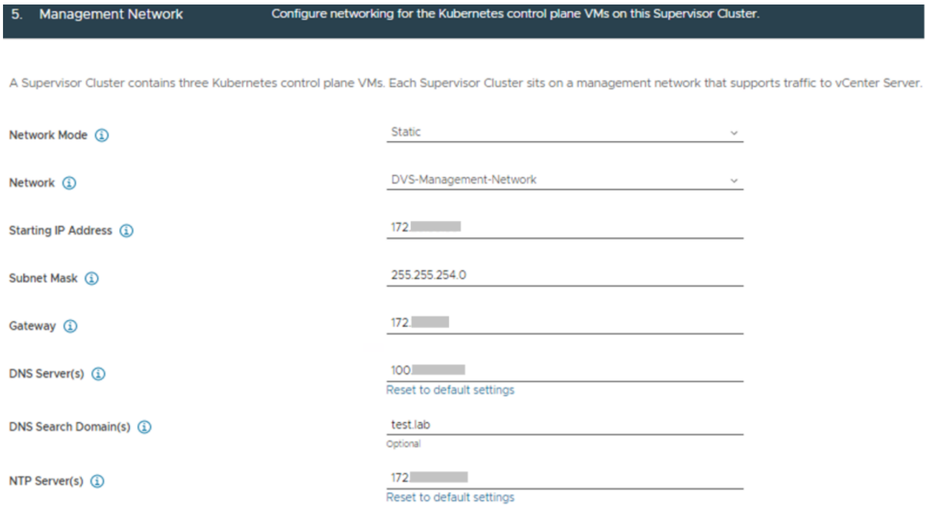
Password: •••••

Server Certificate:

```
-----BEGIN CERTIFICATE-----
MIIDtzCCAp+gAwIBAgIURLlr18Q6X+AOYop9N7CijYhu4BAwDQYJKoZIhvcNAQEL
```

6. Fill in the management network details. Either DHCP or Static assignment can be used. When using static IP address assignment, ensure to reserve a block of five IP addresses for control plane VMs in the Supervisor cluster. When using DHCP, ensure that the DHCP server in your environment supports client identifiers to provide IP addresses for Supervisor Cluster control plane VMs and floating IP. The DHCP server must also be configured with compatible DNS server(s), NTP server(s), and DNS search domain(s). Click NEXT.

Figure 61: Configure management network.



5. Management Network — Configure networking for the Kubernetes control plane VMs on this Supervisor Cluster.

A Supervisor Cluster contains three Kubernetes control plane VMs. Each Supervisor Cluster sits on a management network that supports traffic to vCenter Server.

Network Mode: Static

Network: DVS-Management-Network

Starting IP Address: 172[redacted]

Subnet Mask: 255.255.254.0

Gateway: 172[redacted]

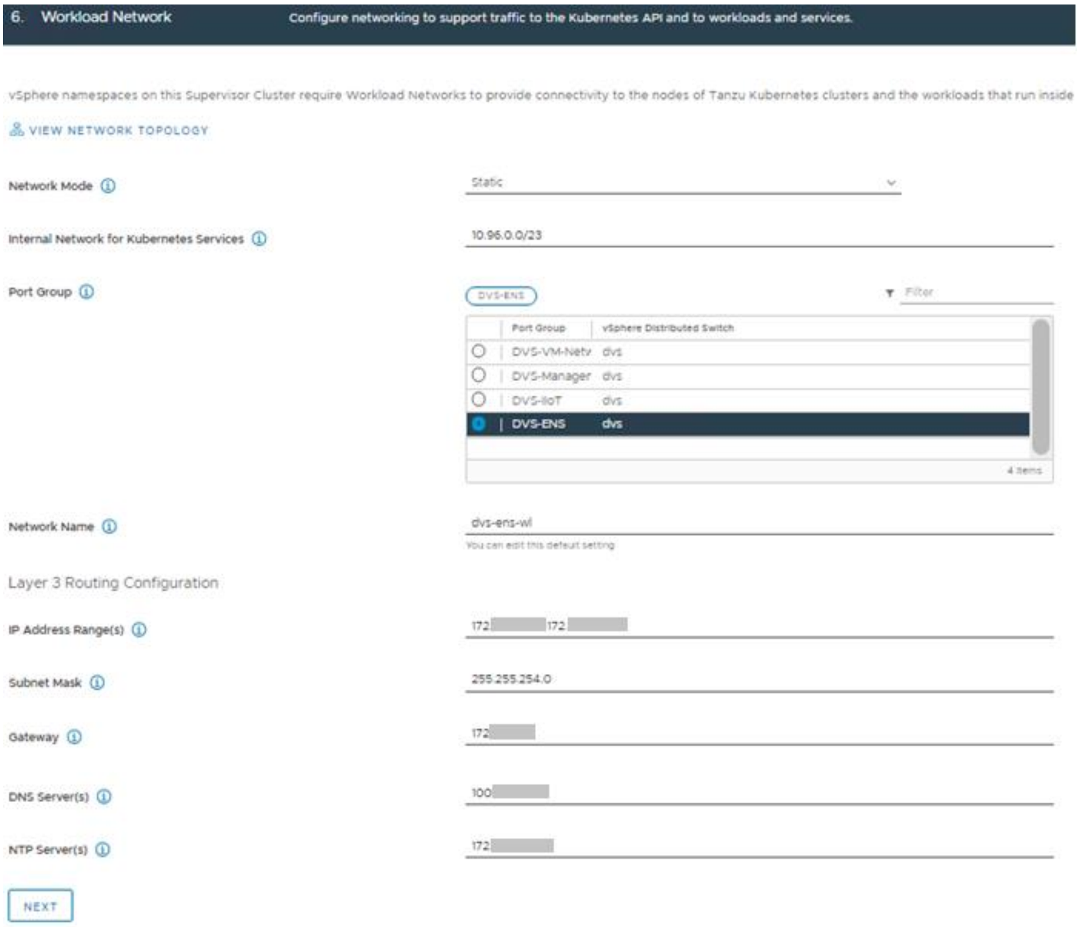
DNS Server(s): 100[redacted]
Reset to default settings

DNS Search Domain(s): test.lab
Optional

NTP Server(s): 172[redacted]
Reset to default settings

7. vSphere namespaces on this Supervisor Cluster require Workload Networks to provide connectivity to the nodes of Tanzu Kubernetes clusters and the workloads that run inside them. Internal IP addresses are used to allocate Kubernetes services of type ClusterIP. These IP addresses are internal to the cluster but should not conflict with any other IP range. Configure the workload network information page for your specific network. Click NEXT.

Figure 62: Configure workload network



6. Workload Network — Configure networking to support traffic to the Kubernetes API and to workloads and services.

vSphere namespaces on this Supervisor Cluster require Workload Networks to provide connectivity to the nodes of Tanzu Kubernetes clusters and the workloads that run inside

Network Mode: Static

Internal Network for Kubernetes Services: 10.96.0.0/23

Port Group: DVS-ENS

	Port Group	vSphere Distributed Switch
○	DVS-VM-Netv	dvs
○	DVS-Manager	dvs
○	DVS-IIoT	dvs
●	DVS-ENS	dvs

4 items

Network Name: dvs-ens-wl

You can edit this default setting

Layer 3 Routing Configuration

IP Address Range(s): 172[redacted]172[redacted]

Subnet Mask: 255.255.254.0

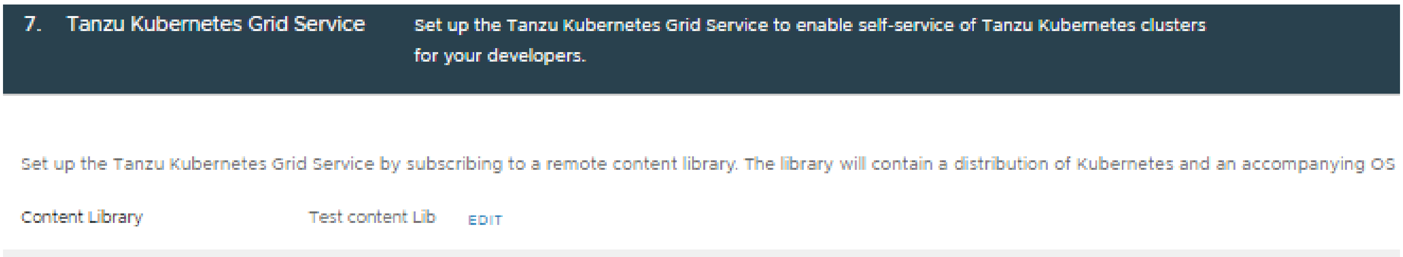
Gateway: 172[redacted]

DNS Server(s): 100[redacted]

NTP Server(s): 172[redacted]

8. Add content library.

Figure 63: Add content library



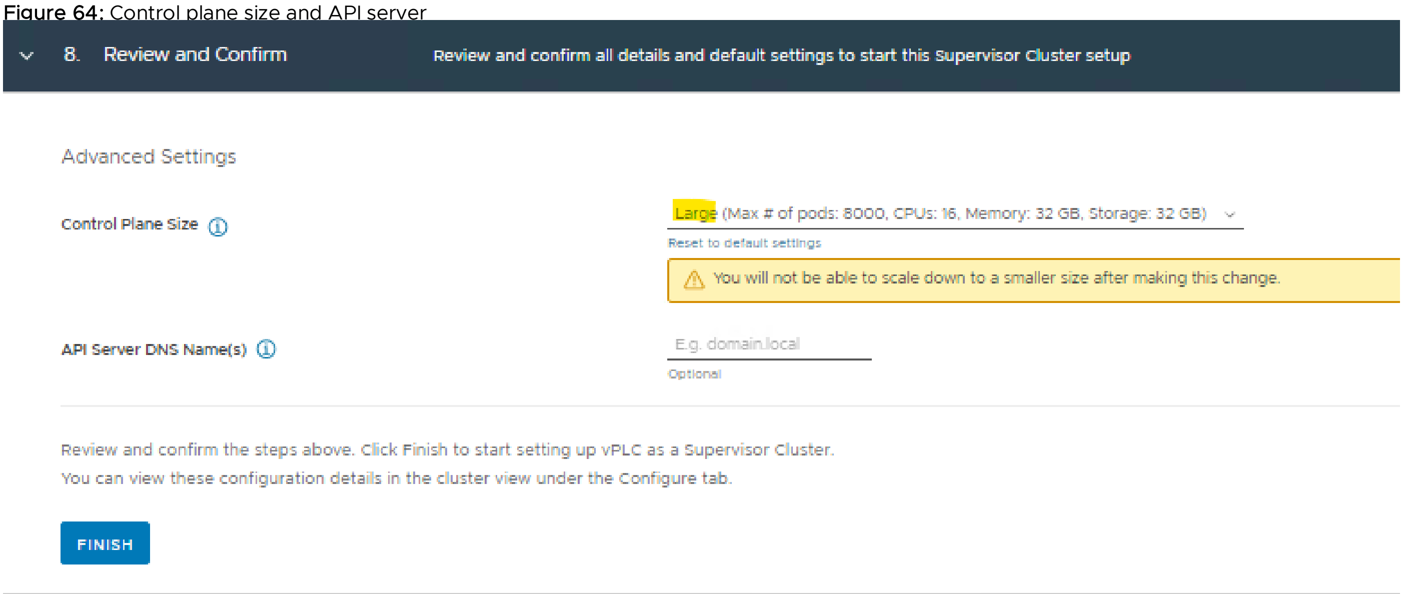
7. Tanzu Kubernetes Grid Service — Set up the Tanzu Kubernetes Grid Service to enable self-service of Tanzu Kubernetes clusters for your developers.

Set up the Tanzu Kubernetes Grid Service by subscribing to a remote content library. The library will contain a distribution of Kubernetes and an accompanying OS

Content Library: Test content Lib

9. Select the size of the control plane VM per your requirements and optionally enter DNS name designated for Kubernetes API server. For production deployments with Tanzu Mission Control integration, a large form factor is recommended for Supervisory control plane nodes. Click Finish to start the configuration process.

Figure 64: Control plane size and API server



8. Review and Confirm — Review and confirm all details and default settings to start this Supervisor Cluster setup

Advanced Settings

Control Plane Size: Large (Max # of pods: 8000, CPUs: 16, Memory: 32 GB, Storage: 32 GB)

Reset to default settings

You will not be able to scale down to a smaller size after making this change.

API Server DNS Name(s): E.g. domain.local

Optional

Review and confirm the steps above. Click Finish to start setting up vPLC as a Supervisor Cluster.

You can view these configuration details in the cluster view under the Configure tab.

Authentication and access

Authentication to Tanzu Kubernetes clusters can be accomplished in different ways depending on your architecture, user authentication and access requirements. In an on-premises environment the simple and reliable method is to use vCenter SSO to authenticate users or to add a local identity source such as Active Directory over LDAPS. For this reference architecture Active Directory authentication with LDAPS was used to authenticate users. The domain controller was configured with Certificate Authority and the controller certificate was exported to be used in the identity source configuration process. In a multi-cloud environment, an external identity source such as Azure Active Directory or Okta can be incorporated for user authentication. VMware cloud services also provide a way to authenticate via federated domains. For more information visit VMware Cloud Services Enterprise Federation page.

What follows are some steps that the vSphere administrator will perform to give access to the domain users in the namespace created for initial workload cluster deployment.

vSphere Administrator Tasks.

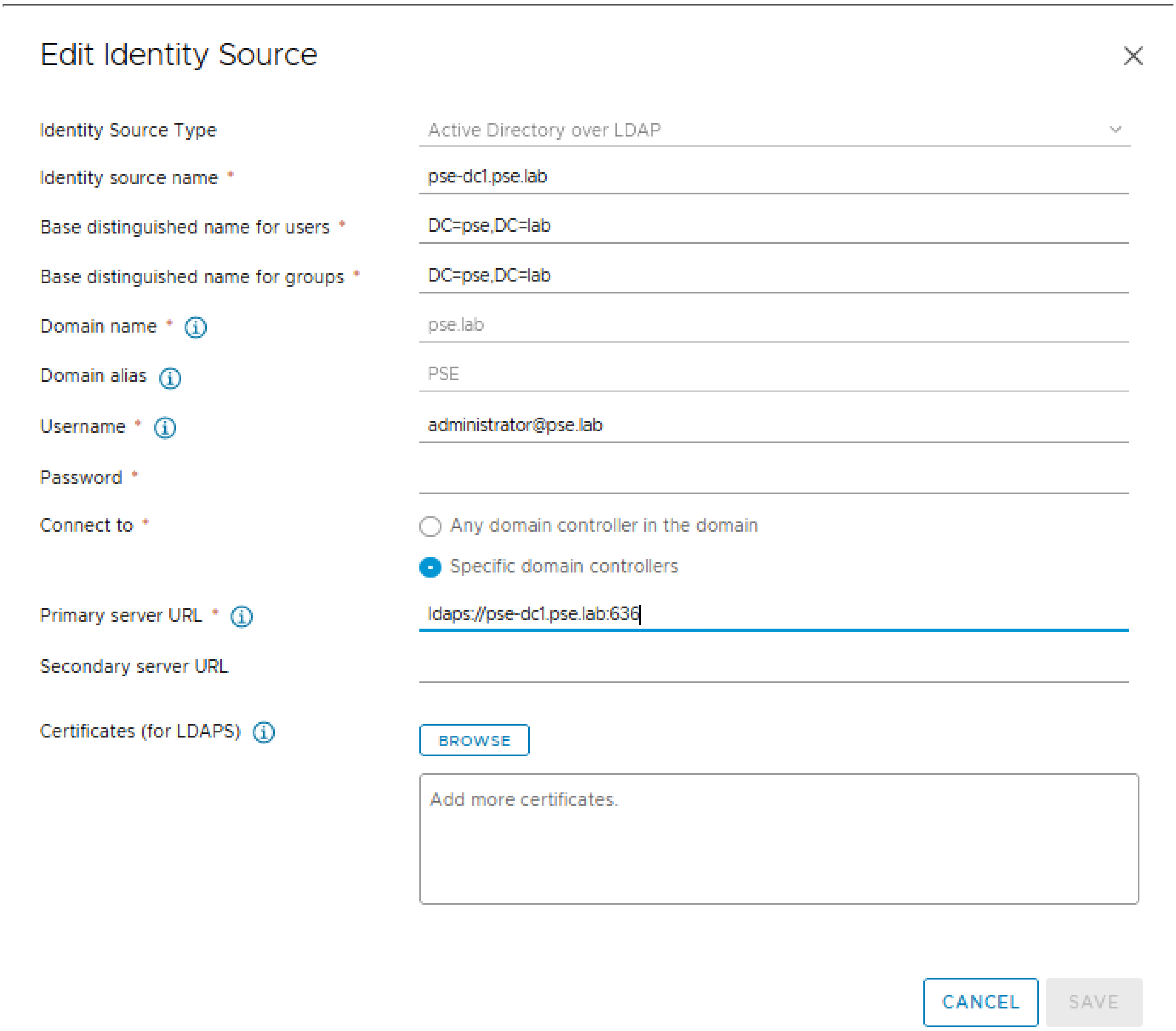
1. Add Active Directory as identity source to vCenter Server.
2. Create namespace for DevOps admins and developers to deploy clusters to.
3. Assign permissions to DevOps engineers in the namespace.
4. Assign storage policies, virtual machine classes and quotas to namespace.
5. Provide namespace access information to DevOps and/or Developers.

Add Active Directory as identity source to vCenter.

1. vSphere menu > Administration > Single Sign On > Configuration > Identity Provide > Identity Sources and click ADD.
2. Fill in the required information for the domain, upload the domain controller certificate and connect to the domain controller using port LDAPS port (636)

Note: Use of domain names with “. Local” is not supported. Please see KB article. For information on configuring LDAPS in your environment please see Microsoft Documentation.

Figure 65: Configure identity source.

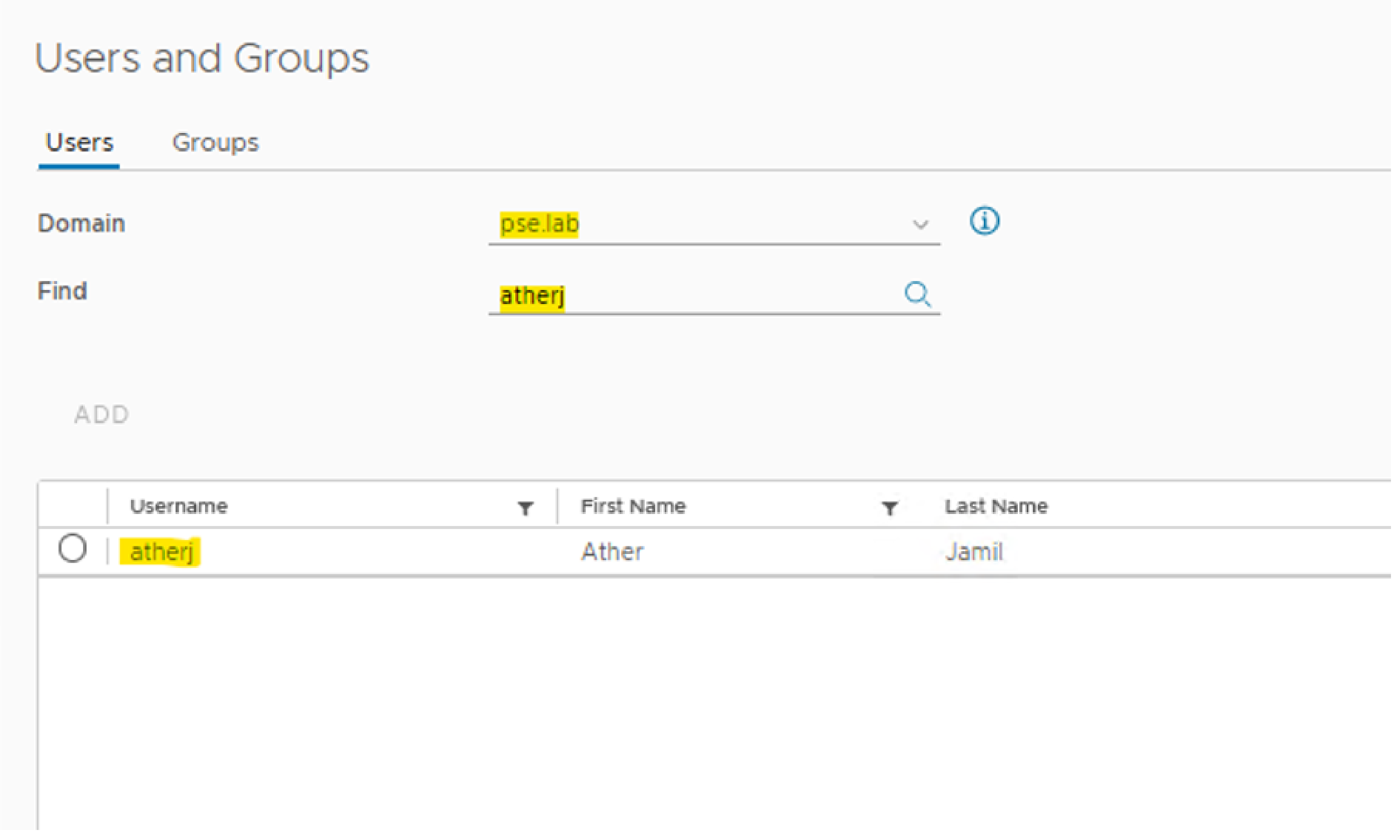


Edit Identity Source

Identity Source Type	Active Directory over LDAP
Identity source name *	pse-dc1.pse.lab
Base distinguished name for users *	DC=pse,DC=lab
Base distinguished name for groups *	DC=pse,DC=lab
Domain name *	pse.lab
Domain alias	PSE
Username *	administrator@pse.lab
Password *	
Connect to *	○ Any domain controller in the domain ● Specific domain controllers
Primary server URL *	ldaps://pse-dc1.pse.lab:636
Secondary server URL	
Certificates (for LDAPS)	BROWSE Add more certificates.

3. To verify the Active Directory integration was successful, navigate to Users and Groups. The domain now should be visible in the drop-down list and query to find a user should be successful.

Figure 66: Validate AD integration.



Users and Groups

Users | Groups

Domain: pse.lab

Find: atherj

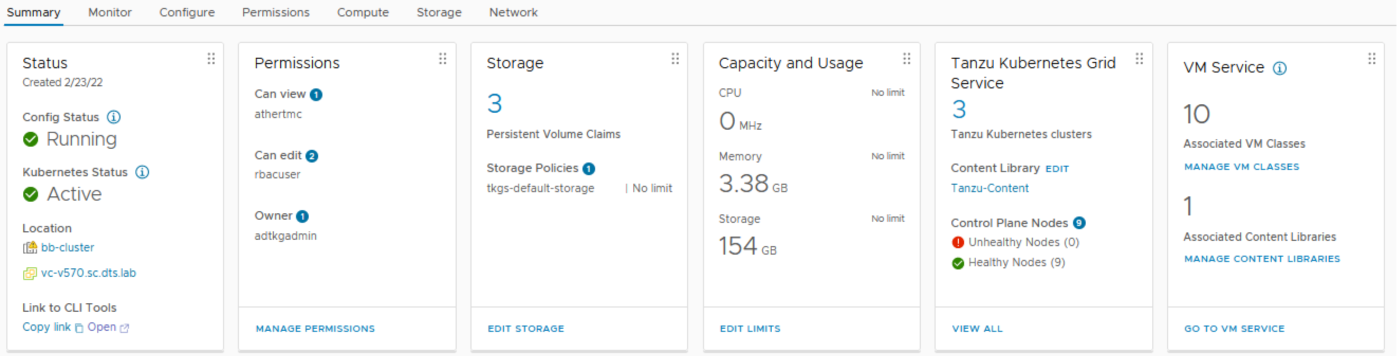
ADD

	Username	First Name	Last Name
○	atherj	Ather	Jamil

Create namespace.

Tanzu Kubernetes Grid Service uses namespaces to provide tenant separation and isolation. Namespaces are defined on the Supervisor cluster and can be configured with user permissions, resource quotas and storage policies. Depending on requirements you assign VM classes and content libraries to the namespaces to download latest Tanzu Kubernetes releases and VM images. The number of namespaces created depends on organizational requirements. For this reference architecture a single namespace was created.

Figure 67: Namespace configuration



Status
Created 2/23/22
Config Status: Running
Kubernetes Status: Active
Location: bb-cluster, vc-v570.sc.dts.lab
Link to CLI Tools: Copy link, Open

Permissions
Can view 1: athertmc
Can edit 2: rbacuser
Owner 1: adtkgadmin

Storage
3 Persistent Volume Claims
Storage Policies 1: tkgs-default-storage | No limit

Capacity and Usage
CPU: 0 MHz (No limit)
Memory: 3.38 GB (No limit)
Storage: 154 GB (No limit)

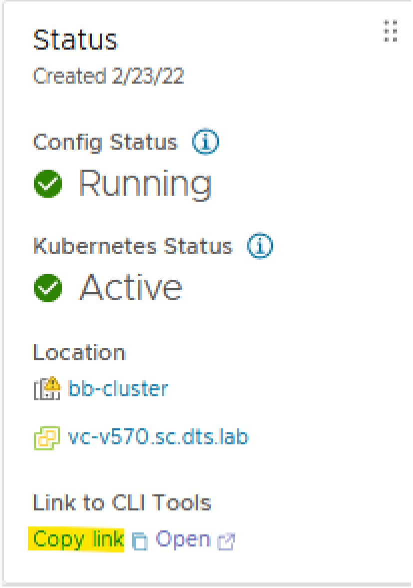
Tanzu Kubernetes Grid Service
3 Tanzu Kubernetes clusters
Content Library: Tanzu-Content
Control Plane Nodes 9
Unhealthy Nodes (0)
Healthy Nodes (9)

VM Service
10 Associated VM Classes
1 Associated Content Libraries

Provide namespace and login information to users.

Once the namespace is configured, the administrator needs to provide the DevOps team with relevant information such as username and password, vCenter Server certificate, as well as the namespace URL, so they can begin to create clusters and deploy workloads on them. The user will install the certificate on the access machine where he or she intend to run Kubernetes commands. The namespace URL can be obtained from the namespace configuration status page as show in figure 43.

Figure 68: Access URL


Status
Created 2/23/22

Config Status
✓ Running

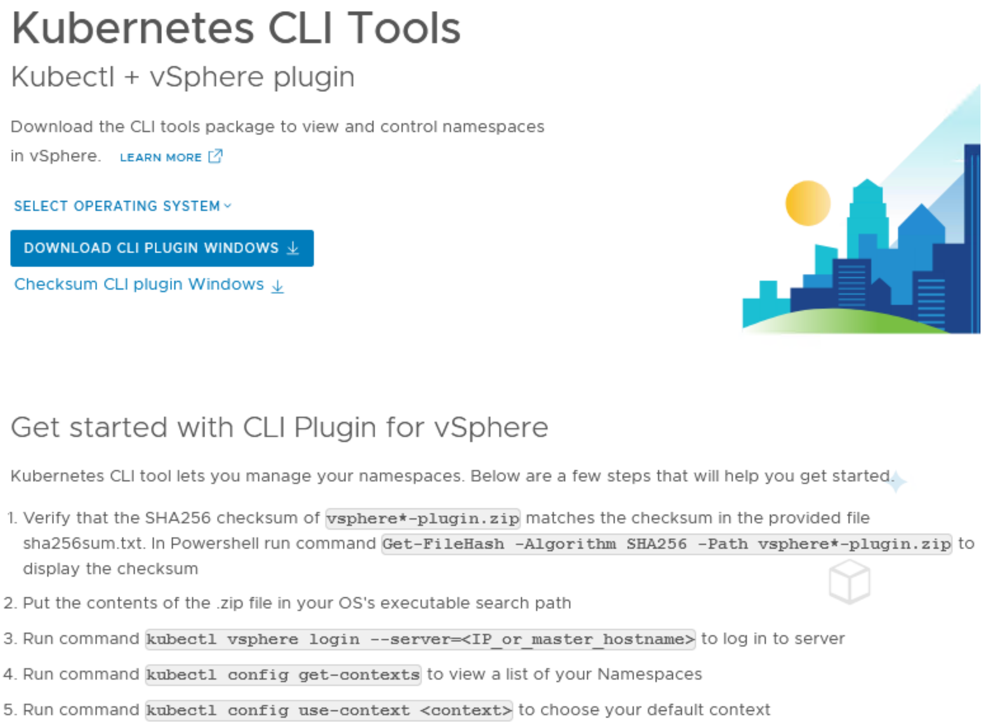
Kubernetes Status
✓ Active

Location
bb-cluster
vc-v570.sc.dts.lab

Link to CLI Tools
Copy link Open

The URL provides instructions to the user to download vSphere CLI tools to access the namespace.

Figure 69: Kubernetes CLI tools


Kubernetes CLI Tools

Kubectl + vSphere plugin

Download the CLI tools package to view and control namespaces in vSphere. LEARN MORE

SELECT OPERATING SYSTEM

DOWNLOAD CLI PLUGIN WINDOWS

Checksum CLI plugin Windows

Get started with CLI Plugin for vSphere

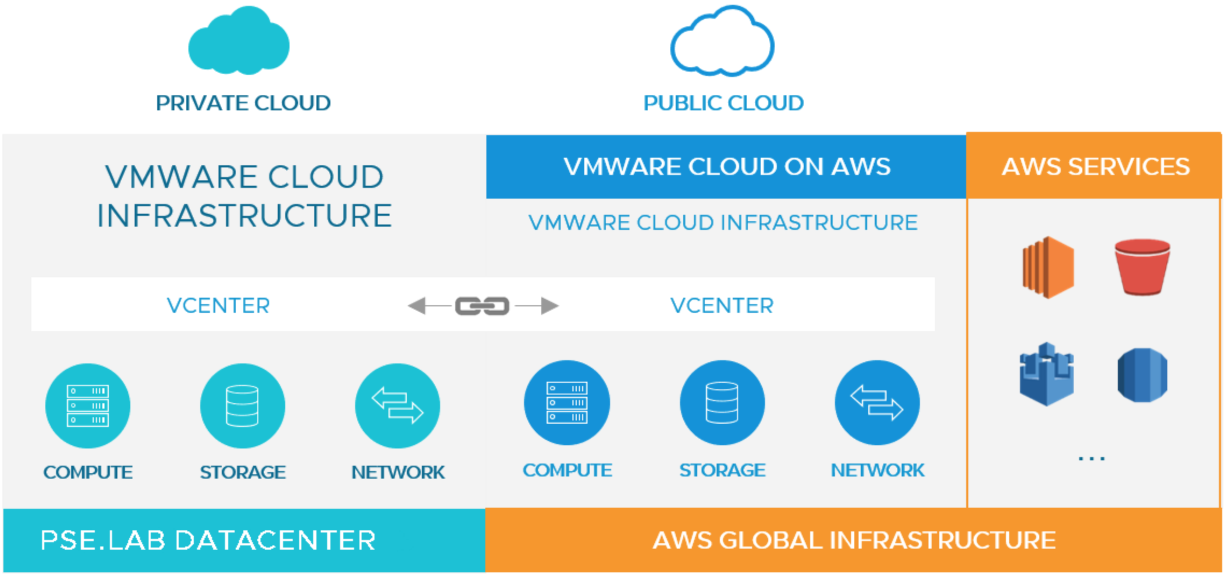
Kubernetes CLI tool lets you manage your namespaces. Below are a few steps that will help you get started.

1. Verify that the SHA256 checksum of `vsphere*-plugin.zip` matches the checksum in the provided file sha256sum.txt. In Powershell run command `Get-FileHash -Algorithm SHA256 -Path vsphere*-plugin.zip` to display the checksum
2. Put the contents of the .zip file in your OS's executable search path
3. Run command `kubectl vsphere login --server=<IP_or_master_hostname>` to log in to server
4. Run command `kubectl config get-contexts` to view a list of your Namespaces
5. Run command `kubectl config use-context <context>` to choose your default context

VMware Cloud (VMC) on AWS

VMware Cloud on AWS is an integrated cloud offering jointly developed by Amazon Web Services (AWS) and VMware. You can deliver a highly scalable and secure service by migrating and extending your on-premises VMware vSphere-based environments to the AWS Cloud running on Amazon Elastic Compute Cloud (Amazon EC2).

Figure 70: VMware cloud on AWS



Note: This document does not discuss provisioning VMware Cloud on AWS. For on-boarding SDDC clusters on AWS please refer to the VMware Cloud on AWS On-Boarding Guide

Note: At the time of writing of this paper, VMC only supports deployment of vSphere with Tanzu with NSX-T. Deployment with NSX Advanced load balancer only is not supported. It can however still be used as a load balancer along with NSX-T.

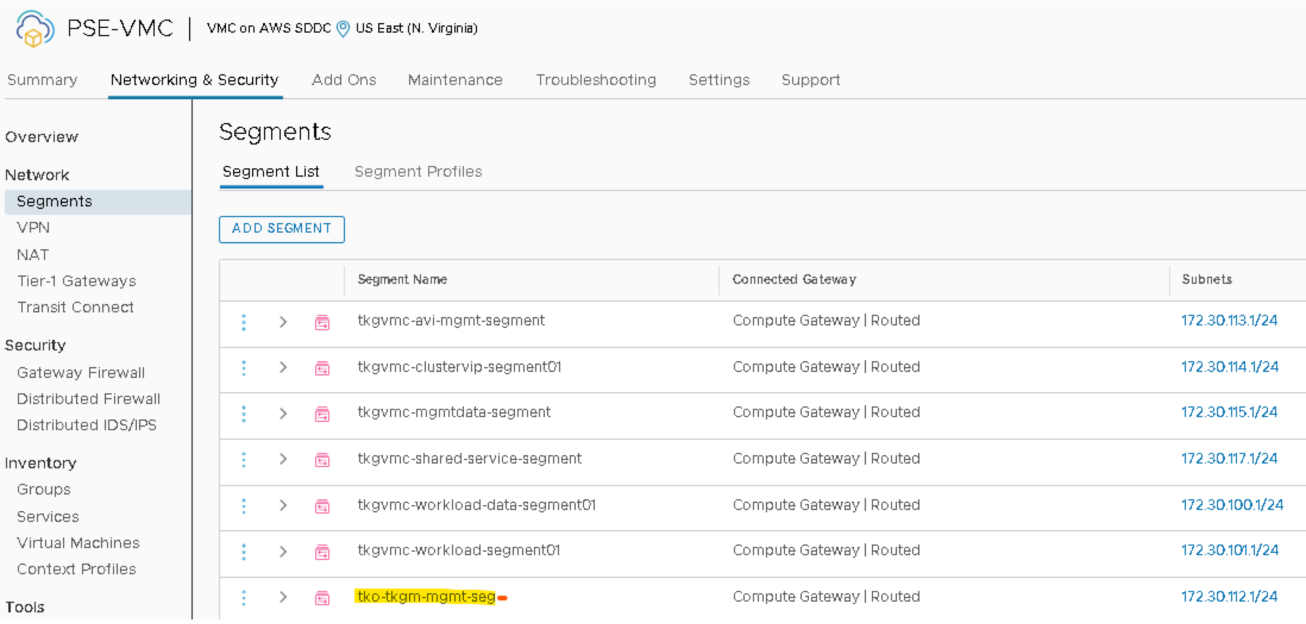
Installing Tanzu for Kubernetes Operations on VMC

Installation of Tanzu Kubernetes Grid clusters is now made easy by Service Installer for Tanzu. For this reference architecture, Service installer for Tanzu (SIVT) version 1.3 was used to deploy Tanzu Kubernetes Grid (multi-cloud) clusters on VMC. Service installer for Tanzu documentation has step by step deployments guides that can be used to deploy Tanzu Kubernetes Grid on vSphere as well as on VMC. For detailed step by step process, please refer to Service Installer for Tanzu deployment guide for VMC.

For this reference architecture Service Installer for Tanzu (SVIT) version 1.3 was installed on the VMC cluster. SVIT only requires a single segment to be created to be used for Tanzu Kubernetes Grid management. SVIT creates the remaining segments based on SVIT reference design for deploying Tanzu Kubernetes Grid on VMC.

For example, in the below snapshot, segment “tko-tkgm-mgmt-seg” was manually created for Tanzu for Kubernetes Operations (TKO) management traffic. The remaining segment are created by SVIT based on the input provided during the deployment process.

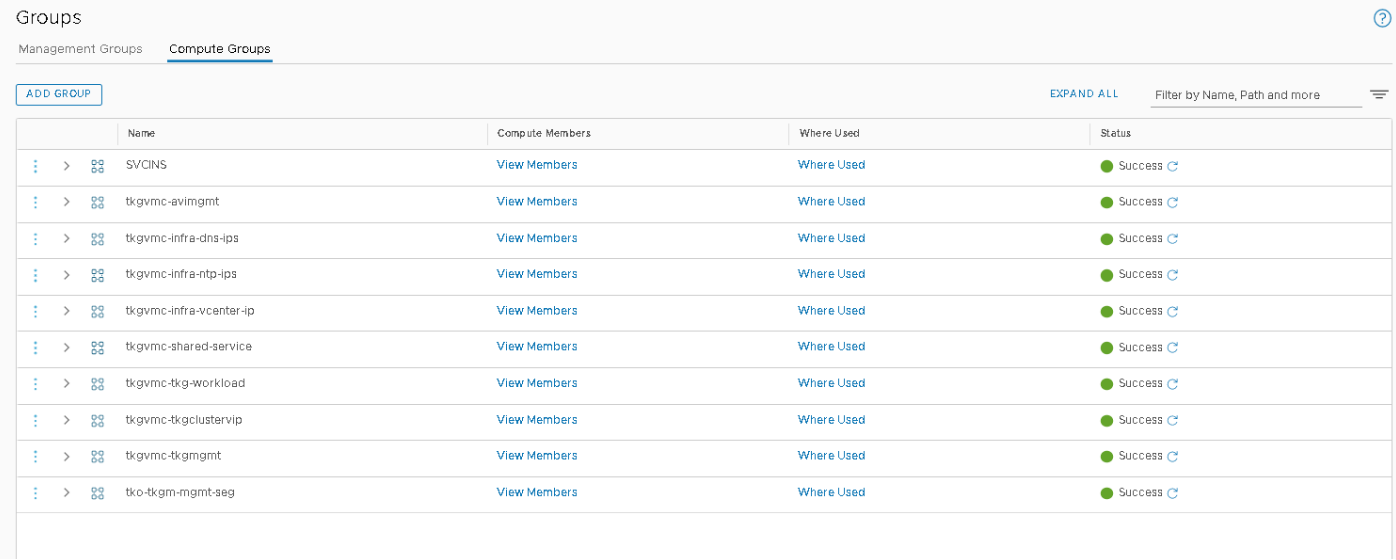
Figure 71: VMC network segment creation



Segment Name	Connected Gateway	Subnets
tkgvmc-avi-mgmt-segment	Compute Gateway | Routed	172.30.113.1/24
tkgvmc-clustervip-segment01	Compute Gateway | Routed	172.30.114.1/24
tkgvmc-mgmtdata-segment	Compute Gateway | Routed	172.30.115.1/24
tkgvmc-shared-service-segment	Compute Gateway | Routed	172.30.117.1/24
tkgvmc-workload-data-segment01	Compute Gateway | Routed	172.30.100.1/24
tkgvmc-workload-segment01	Compute Gateway | Routed	172.30.101.1/24
tko-tkgm-mgmt-seg	Compute Gateway | Routed	172.30.112.1/24

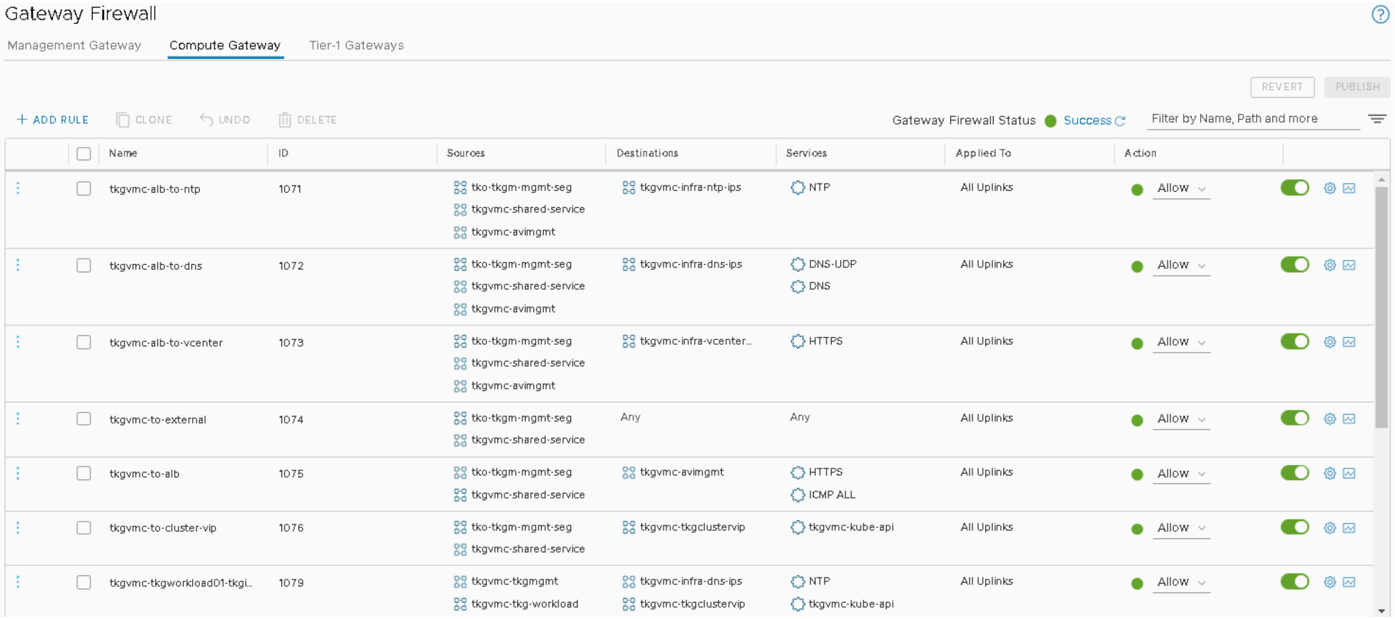
SVIT will also create groups, Gateway Firewall rules and assign them to the groups.

Figure 72: Groups



Name	Compute Members	Where Used	Status
SVCINS	View Members	Where Used	● Success
tkgvmc-avimgmt	View Members	Where Used	● Success
tkgvmc-infra-dns-ips	View Members	Where Used	● Success
tkgvmc-infra-ntp-ips	View Members	Where Used	● Success
tkgvmc-infra-vcenter-ip	View Members	Where Used	● Success
tkgvmc-shared-service	View Members	Where Used	● Success
tkgvmc-tkg-workload	View Members	Where Used	● Success
tkgvmc-tkgclustervip	View Members	Where Used	● Success
tkgvmc-tkgmgmt	View Members	Where Used	● Success
tko-tkgm-mgmt-seg	View Members	Where Used	● Success

Figure 73: Firewall rules

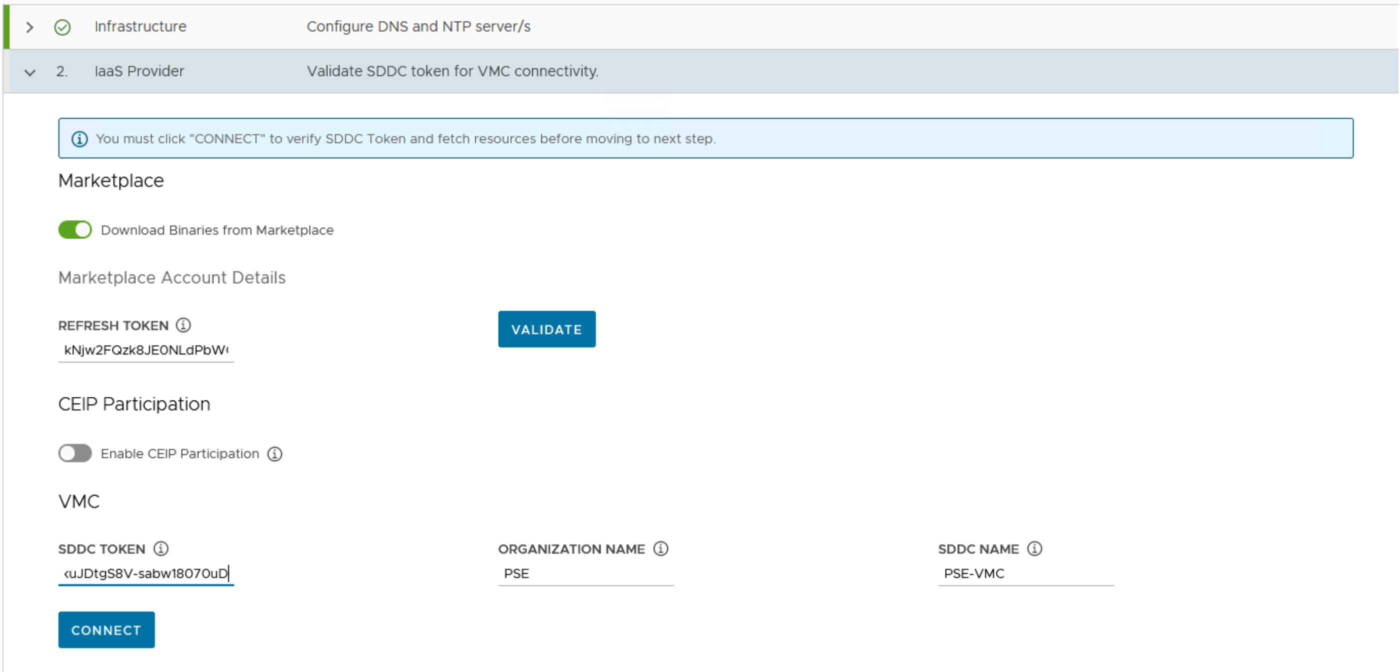


	Name	ID	Sources	Destinations	Services	Applied To	Action
□	tkgvmc-alb-to-ntp	1071	tko-tkgm-mgmt-seg tkgvmc-shared-service tkgvmc-avimgmt	tkgvmc-infra-ntp-ips	NTP	All Uplinks	● Allow
□	tkgvmc-alb-to-dns	1072	tko-tkgm-mgmt-seg tkgvmc-shared-service tkgvmc-avimgmt	tkgvmc-infra-dns-ips	DNS-UDP DNS	All Uplinks	● Allow
□	tkgvmc-alb-to-vcenter	1073	tko-tkgm-mgmt-seg tkgvmc-shared-service tkgvmc-avimgmt	tkgvmc-infra-vcenter...	HTTPS	All Uplinks	● Allow
□	tkgvmc-to-external	1074	tko-tkgm-mgmt-seg tkgvmc-shared-service	Any	Any	All Uplinks	● Allow
□	tkgvmc-to-alb	1075	tko-tkgm-mgmt-seg tkgvmc-shared-service	tkgvmc-avimgmt	HTTPS ICMP ALL	All Uplinks	● Allow
□	tkgvmc-to-cluster-vip	1076	tko-tkgm-mgmt-seg tkgvmc-shared-service	tkgvmc-tkgclustervip	tkgvmc-kube-api	All Uplinks	● Allow
□	tkgvmc-tkgworkload01-tkgi...	1079	tkgvmc-tkgmgmt tkgvmc-tkg-workload	tkgvmc-infra-dns-ips tkgvmc-tkgclustervip	NTP tkgvmc-kube-api	All Uplinks	● Allow

High-level steps to deploy Tanzu for Kubernetes Operations with NSX Advanced Load Balancer on VMC using SVIT as follows.

1. Download SVIT OVA (Open Virtual Appliance) from VMware Marketplace. A marketplace account is required.
2. Create a SVIT virtual machine using the OVA. Ensure that the virtual machine resides on the management network that you manually created.
3. Obtain your SDDC token as well as a marketplace token. These tokens will be needed for SVIT to configure SDDC and to download NSX Advanced Load Balancer OVA and other Kubernetes images from VMware Marketplace. Below is a snapshot of where these tokens are entered in SVIT user interface.

Figure 74: Tokens



Infrastructure — Configure DNS and NTP server/s

2. IaaS Provider — Validate SDDC token for VMC connectivity.

ⓘ You must click "CONNECT" to verify SDDC Token and fetch resources before moving to next step.

Marketplace

● Download Binaries from Marketplace

Marketplace Account Details

REFRESH TOKEN ⓘ
kNjw2FQzk8JEONLdPbW

VALIDATE

CEIP Participation

○ Enable CEIP Participation ⓘ

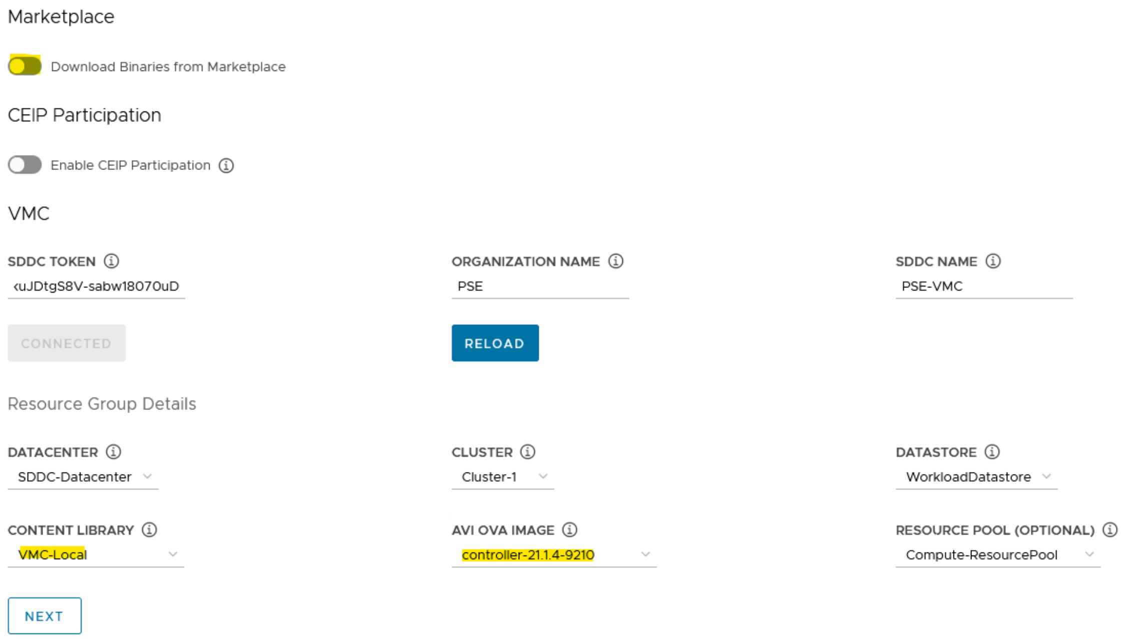
VMC

SDDC TOKEN ⓘ: uJDtgS8V-sabw18070uD
ORGANIZATION NAME ⓘ: PSE
SDDC NAME ⓘ: PSE-VMC

CONNECT

Note: In instances where SVIT cannot download NSX Advanced Load Balancer controller and Kubernetes images from the VMware Marketplace, a local content library needs to be created and images and controller. OVA uploaded. SVIT will install the controller from the local content library as shown in figure below.

Figure 75: local content library option



Marketplace

● Download Binaries from Marketplace

CEIP Participation

○ Enable CEIP Participation ⓘ

VMC

SDDC TOKEN ⓘ: ‹uJDtgS8V-sabw18070uD

ORGANIZATION NAME ⓘ: PSE

SDDC NAME ⓘ: PSE-VMC

CONNECTED | RELOAD

Resource Group Details

DATACENTER ⓘ: SDDC-Datacenter

CLUSTER ⓘ: Cluster-1

DATASTORE ⓘ: WorkloadDatastore

CONTENT LIBRARY ⓘ: VMC-Local

AVI OVA IMAGE ⓘ: controller-21.1.4-9210

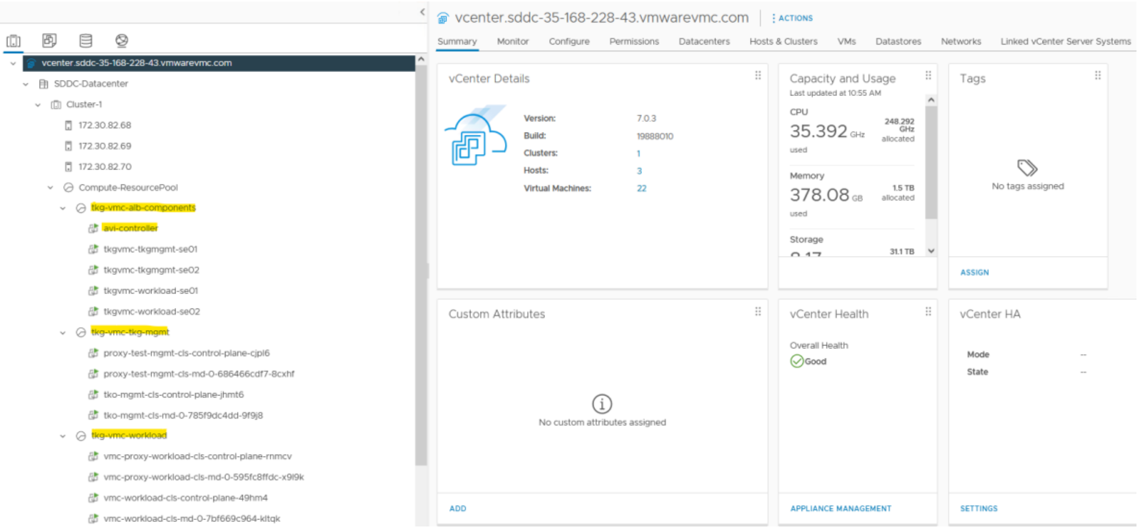
RESOURCE POOL (OPTIONAL) ⓘ: Compute-ResourcePool

NEXT

4. Enter your environment specific parameters on the remaining SVIT screen and review the configuration. On the review page you will have an option to view and safe the configuration .yaml file to your local disk or to save it to SVIT virtual machine. The default location for SVIT deployment yaml files is "/opt/vmware/arcas/src/"
5. Finally, run the command with desired parameters to deploy Tanzu for Kubernetes Operations. The command and parameters are in the SVIT deployment guide mentioned previously.

If deployment is successfully completed, appropriate resource groups and clusters will be created and viewable in vCenter.

Figure 76: VMC Tanzu Clusters



- vcenter.sddc-35-168-228-43.vmwarevmc.com
  - SDDC-Datacenter
    - Cluster-1
      - 172.30.82.68
      - 172.30.82.69
      - 172.30.82.70
      - Compute-ResourcePool
        - tkg-vmc-alb-components
          - avi-controller
          - tkgvmc-tkgmgmt-se01
          - tkgvmc-tkgmgmt-se02
          - tkgvmc-workload-se01
          - tkgvmc-workload-se02
        - tkg-vmc-tkg-mgmt
          - proxy-test-mgmt-cls-control-plane-cjpl6
          - proxy-test-mgmt-cls-md-0-686466cdf7-8cxhf
          - tko-mgmt-cls-control-plane-jhmt6
          - tko-mgmt-cls-md-0-785f9dc4dd-9f9j8
        - tkg-vmc-workload
          - vmc-proxy-workload-cls-control-plane-rnmcv
          - vmc-proxy-workload-cls-md-0-595fc8ffdc-x9l9k
          - vmc-workload-cls-control-plane-49hm4
          - vmc-workload-cls-md-0-7bf669c964-kltqk

vcenter.sddc-35-168-228-43.vmwarevmc.com | ACTIONS

Summary | Monitor | Configure | Permissions | Datacenters | Hosts & Clusters | VMs | Datastores | Networks | Linked vCenter Server Systems

vCenter Details

Version:	7.0.3
Build:	19888010
Clusters:	1
Hosts:	3
Virtual Machines:	22

Capacity and Usage
Last updated at 10:55 AM

CPU: 35.392 GHz used, 248.292 GHz allocated

Memory: 378.08 GB used, 1.5 TB allocated

Storage: 31.1 TB

Tags: No tags assigned — ASSIGN

Custom Attributes: No custom attributes assigned — ADD

vCenter Health: Overall Health ✓ Good — APPLIANCE MANAGEMENT

vCenter HA: Mode --, State -- — SETTINGS

Note: Tanzu CLI and accompanying tools will be required to manage clusters. Visit Install Tanzu CLI for installation instructions.

Amazon Elastic Kubernetes Service (EKS)

Amazon Elastic Kubernetes Service is a managed service that can be used to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes. Amazon EKS runs a single tenant control plane for each cluster. This means that the same control plane infrastructure cannot be shard across clusters. A customer has the option to create worker nodes as self-managed Amazon EC2 nodes or to deploy their applications workloads to AWS Fargate, which is a serverless compute engine. For this reference architecture Amazon EC2 nodes were used. For more information on Fargate visit AWS Fargate page.

Amazon EKS clusters can be deployed either via “eksctl” utility or via AWS EKS user interface via AWS console. When eksctl utility is used with default settings, a new Amazon Cloud Formation stack is created which also includes an associated VPC (Virtual Private Cloud) and subnets. This is a viable option when there is a need for a separate VPC for EKS clusters. For this reference architecture AWS console was used to create EKS clusters and nodes and integrated into an existing VPC. This process is outlined below.

Note: This process requires that an AWS VPC exists, and subnets configured per requirements. For this reference architecture a VPC was created with subnet connections and routing in place to the internal private network, which included connections to VMC on AWS as well as the on-premises environment through Equinix. This configuration is depicted in the previously mentioned guide Multi-Cloud Network Connectivity with Equinix Overview and figure 3. For more information on how to create AWS VPC please consult AWS documentation.

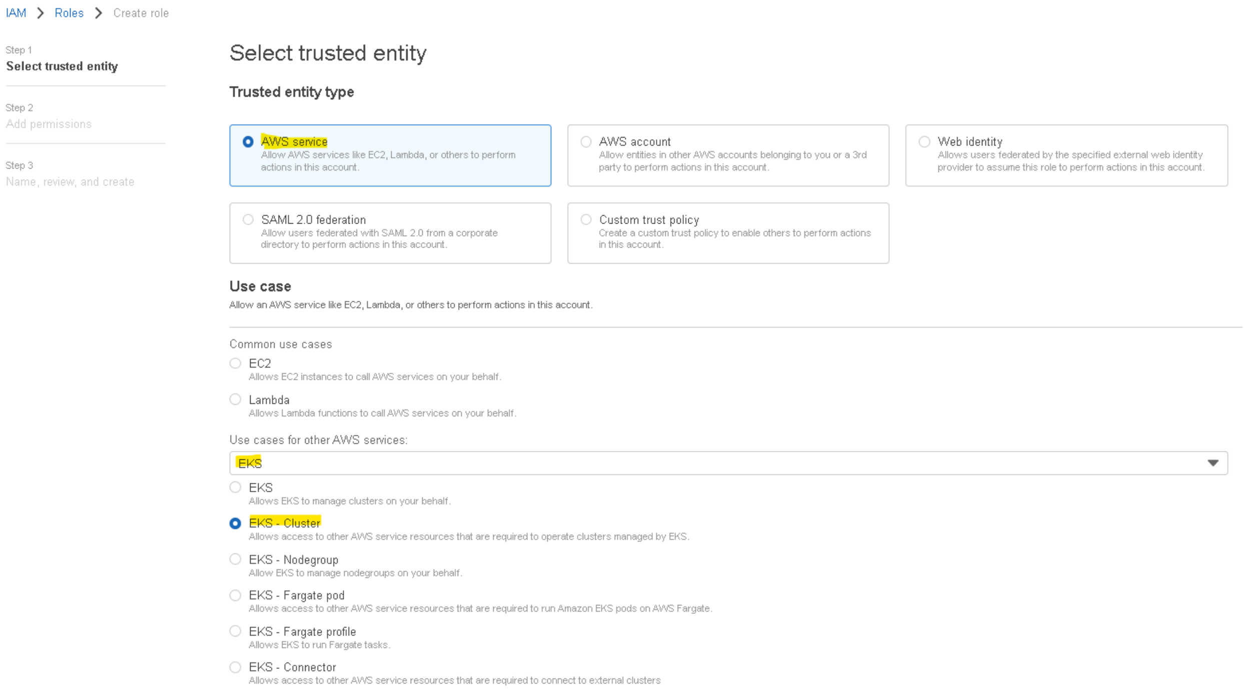
Creating EKS clusters

Amazon EKS make calls to other AWS services on your behalf to manage the resources that you use with the service. Prior to creating EKS clusters, an IAM (Identity Access Management) role needs to exist or be created for EKS service. Determine if your account already has a role named “eksClusterRole.” If it does not exist, then create one as follows. The example below shows the role created for this reference architecture.

Create IAM Role

1. From IAM > Roles menu create a role. Select “AWS Service” and search for “EKS” in “Use cases for other AWS services” search field. Select EKS Clusters and click next.

Figure 77: Create IAM Role



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

○ EC2
Allows EC2 instances to call AWS services on your behalf.

○ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

EKS

○ EKS
Allows EKS to manage clusters on your behalf.

● EKS - Cluster
Allows access to other AWS service resources that are required to operate clusters managed by EKS.

○ EKS - Nodegroup
Allow EKS to manage nodegroups on your behalf.

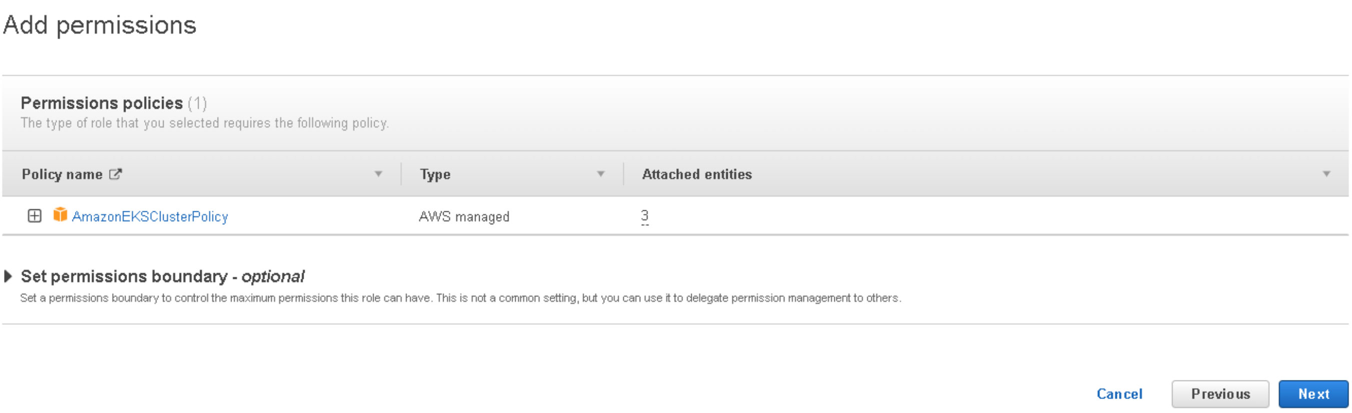
○ EKS - Fargate pod
Allows access to other AWS service resources that are required to run Amazon EKS pods on AWS Fargate.

○ EKS - Fargate profile
Allows EKS to run Fargate tasks.

○ EKS - Connector
Allows access to other AWS service resources that are required to connect to external clusters

2. On “Add Permissions screen, Amazon EKSClusterPolicy will be automatically added. Click Next.

Figure 78: Add permissions.



Add permissions

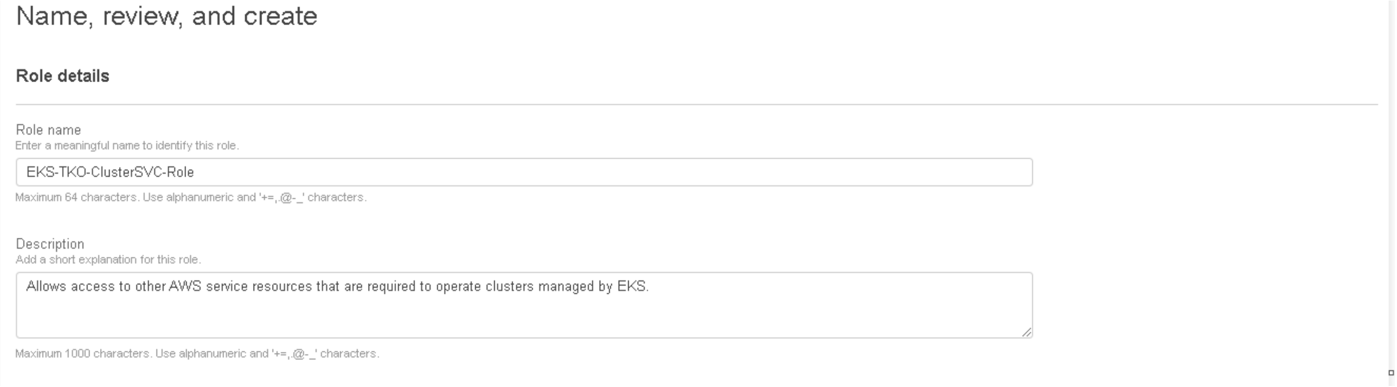
Permissions policies (1)
The type of role that you selected requires the following policy.

Policy name	Type	Attached entities
AmazonEKSClusterPolicy	AWS managed	3

▸ **Set permissions boundary - *optional***
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

3. Give the role a meaningful name and click Create Role.

Figure 79: Create role.



Name, review, and create

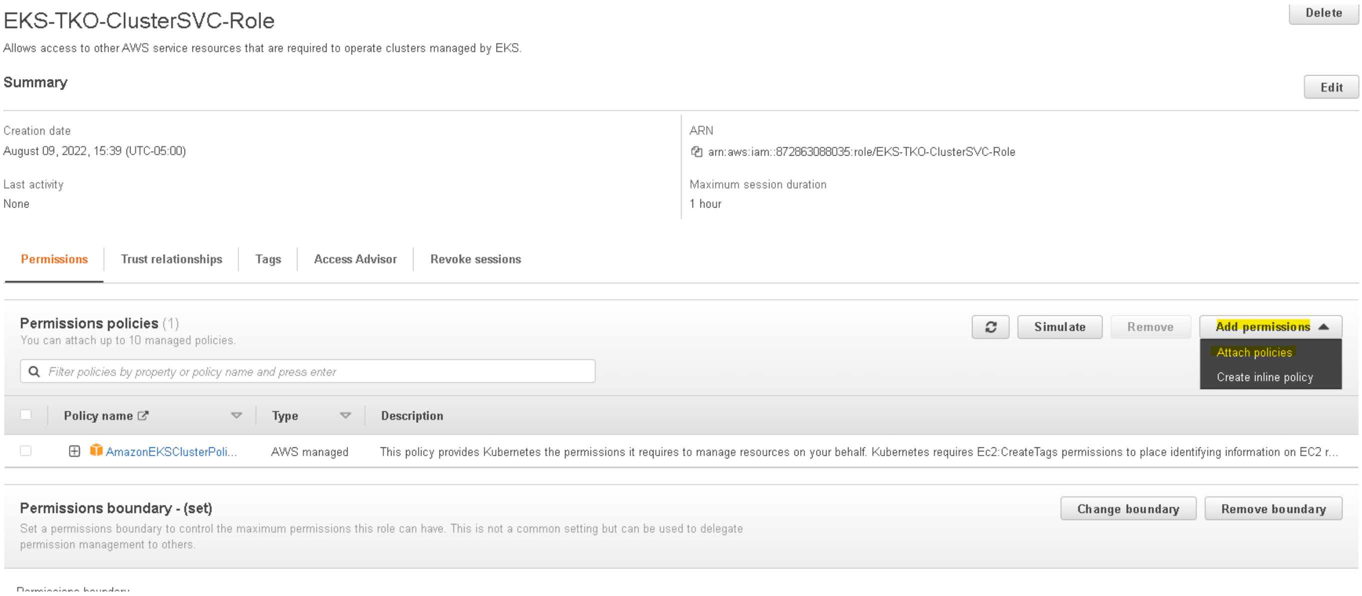
Role details

Role name
Enter a meaningful name to identify this role.
EKS-TKO-ClusterSVC-Role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows access to other AWS service resources that are required to operate clusters managed by EKS.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

4. Edit the role just created and select "Attach Policies."

Figure 80: Attach policies.



EKS-TKO-ClusterSVC-Role
Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Summary

Creation date
August 09, 2022, 15:39 (UTC-05:00)

Last activity
None

ARN
arn:aws:iam::872863088035:role/EKS-TKO-ClusterSVC-Role

Maximum session duration
1 hour

Permissions policies (1)
You can attach up to 10 managed policies.

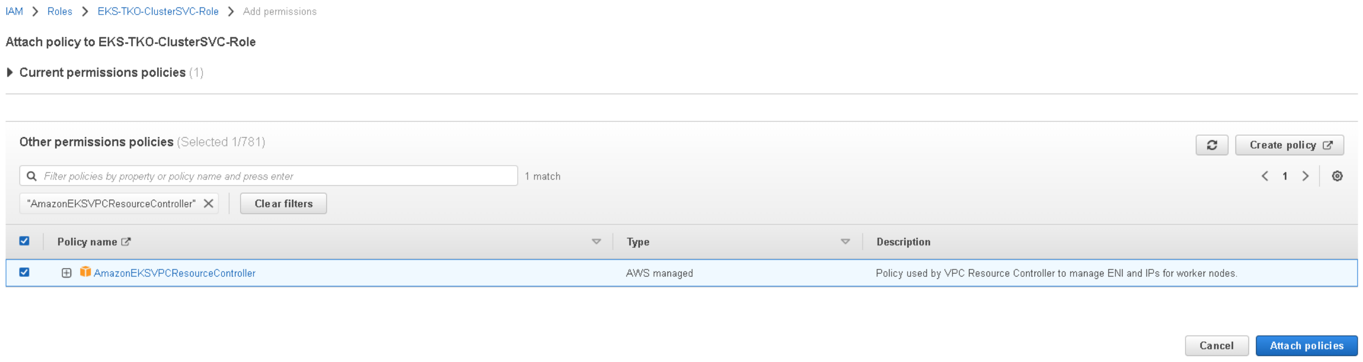
Filter policies by property or policy name and press enter

Policy name	Type	Description
AmazonEKSClusterPoli...	AWS managed	This policy provides Kubernetes the permissions it requires to manage resources on your behalf. Kubernetes requires Ec2:CreateTags permissions to place identifying information on EC2 r...

Permissions boundary - (set)
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

5. On the next screen search for "AmazonEKSVPCResourceController" and select "Attach Policy"

Figure 81: Attach polices.



IAM > Roles > EKS-TKO-ClusterSVC-Role > Add permissions

Attach policy to EKS-TKO-ClusterSVC-Role

▸ Current permissions policies (1)

Other permissions policies (Selected 1/781)

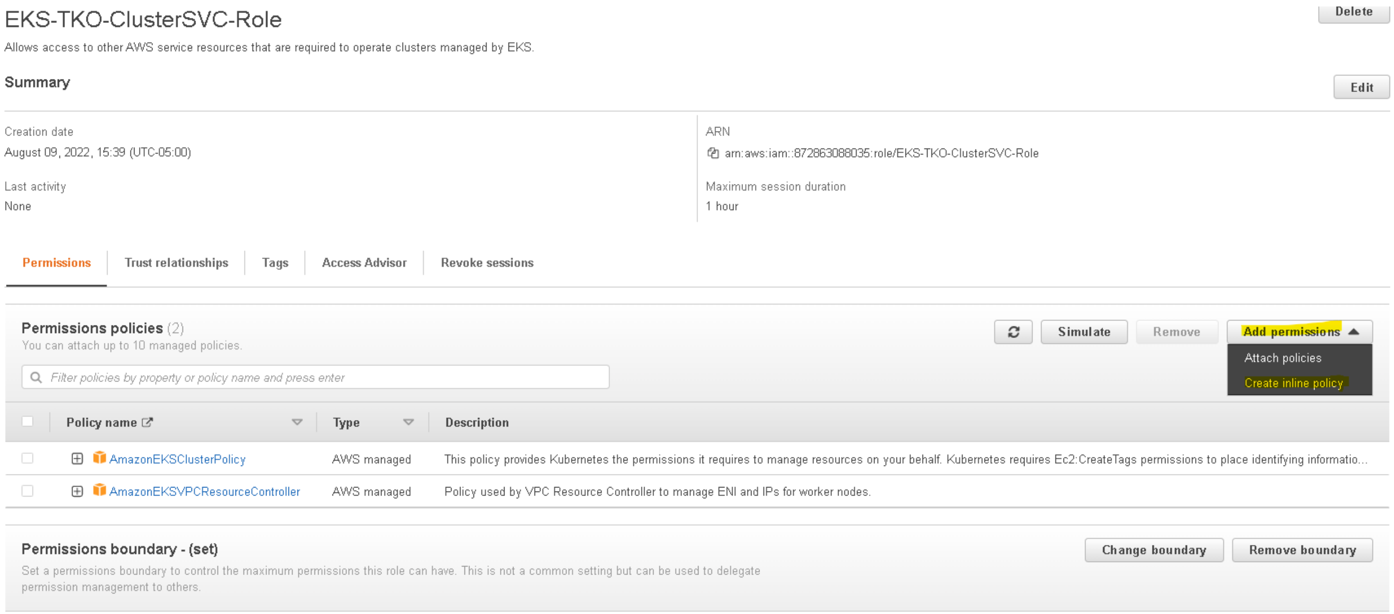
Filter policies by property or policy name and press enter — 1 match

"AmazonEKSVPCResourceController" ✕ — Clear filters

☑	Policy name	Type	Description
☑	⊞ AmazonEKSVPCResourceController	AWS managed	Policy used by VPC Resource Controller to manage ENI and IPs for worker nodes.

6. If Amazon CloudWatch monitoring is desired an inline policy needs to be created for CloudWatch to receive the clusters metrics. Create an incline policy by selecting “Create Inline Policy.”

Figure 82: Inline Policies



EKS-TKO-ClusterSVC-Role

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Summary

Creation date
August 09, 2022, 15:39 (UTC-05:00)

Last activity
None

ARN
arn:aws:iam::872863088035:role/EKS-TKO-ClusterSVC-Role

Maximum session duration
1 hour

Permissions policies (2)
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

□	Policy name	Type	Description
□	⊞ AmazonEKSClusterPolicy	AWS managed	This policy provides Kubernetes the permissions it requires to manage resources on your behalf. Kubernetes requires Ec2:CreateTags permissions to place identifying informatio...
□	⊞ AmazonEKSVPCResourceController	AWS managed	Policy used by VPC Resource Controller to manage ENI and IPs for worker nodes.

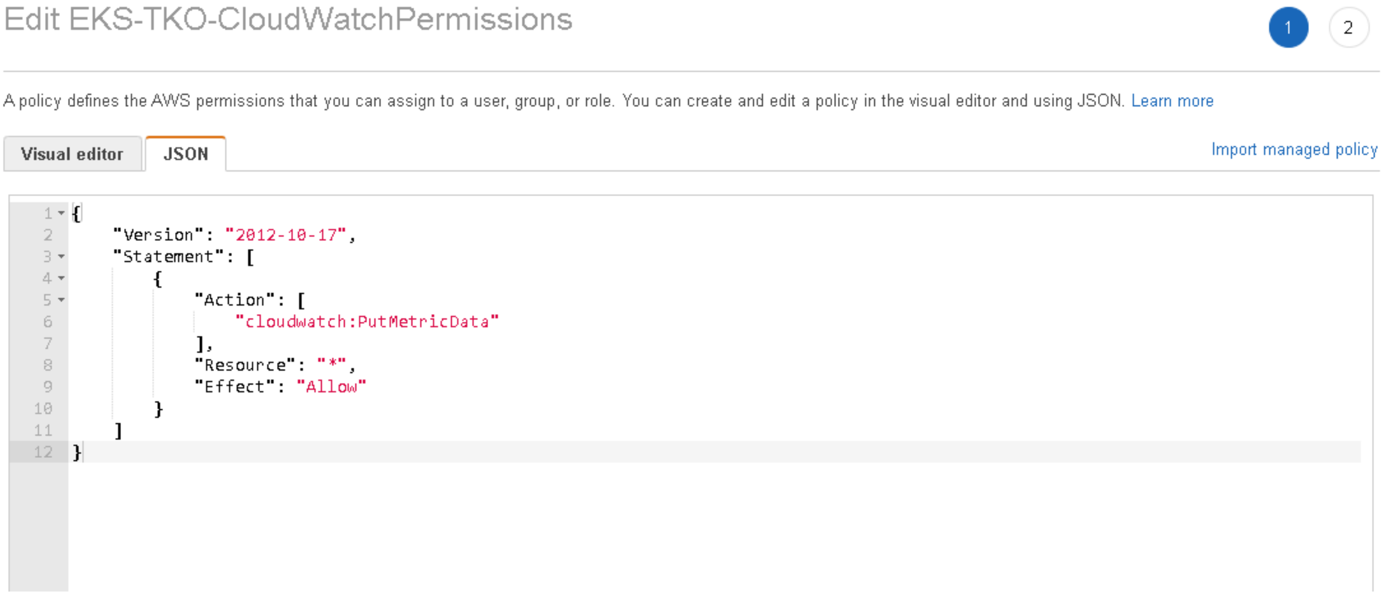
Permissions boundary - (set)
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

7. On the next screen select “JSON” and enter or paste the policy definition below and attach policy.

Inline cloud watch

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

Figure 83: Inline polices.

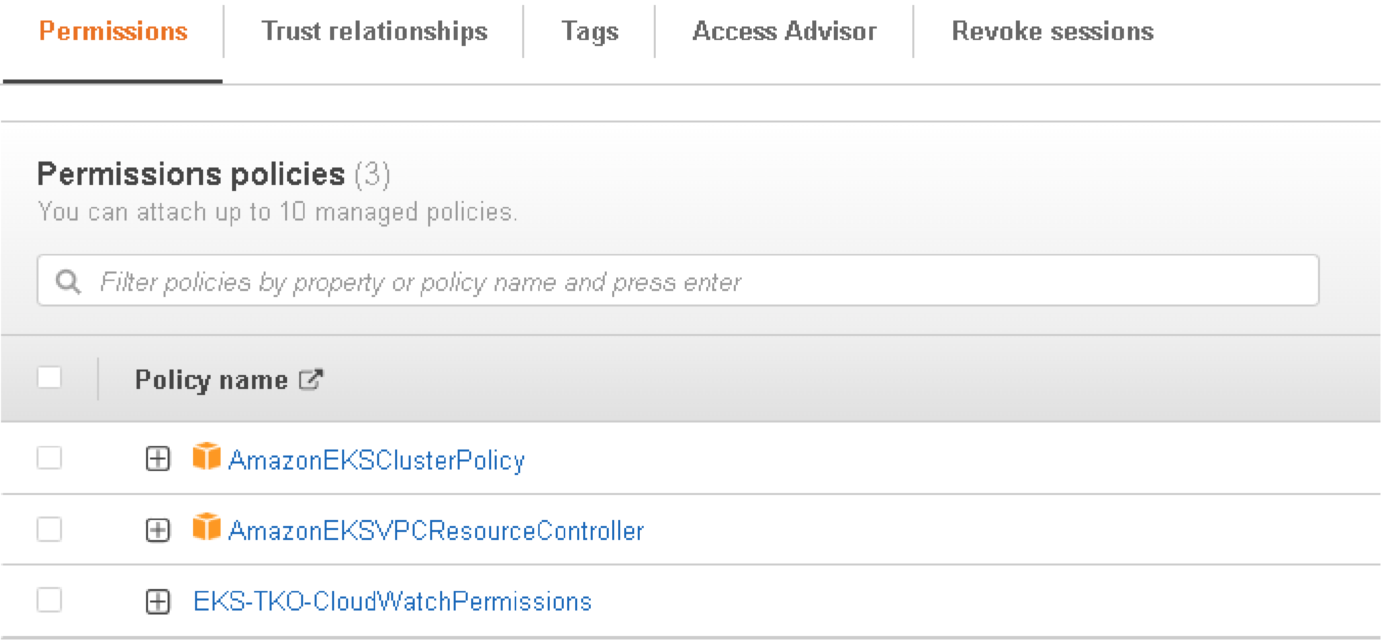


A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

8. Once created, the IAM role should have the following policies. EKS clusters can now be created.

Figure 84: IAM role created.



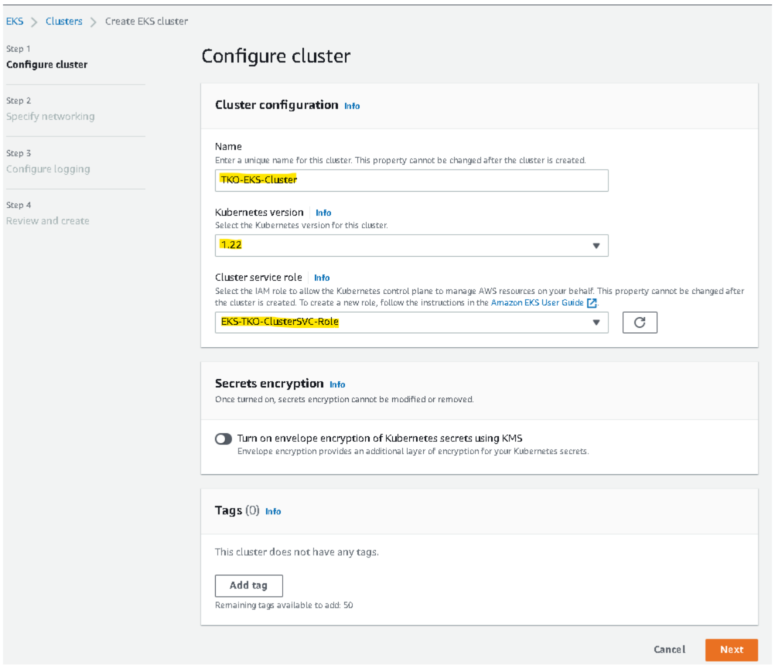
Permissions policies (3)
You can attach up to 10 managed policies.

	Policy name
□	AmazonEKSClusterPolicy
□	AmazonEKSVPCResourceController
□	EKS-TKO-CloudWatchPermissions

Create EKS Cluster

1. In Amazon console, navigate to EKS > Clusters > Create EKS cluster. Assign cluster a name, choose Kubernetes version and select cluster service role created earlier. Click Next.

Figure 85: Create EKS cluster.



EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Review and create

Configure cluster

Cluster configuration Info

Name
Enter a unique name for this cluster. This property cannot be changed after the cluster is created.
TKO-EKS-Cluster

Kubernetes version Info
Select the Kubernetes version for this cluster.
1.22

Cluster service role Info
Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This property cannot be changed after the cluster is created. To create a new role, follow the instructions in the Amazon EKS User Guide.
EKS-TKO-ClusterSVC-Role

Secrets encryption Info
Once turned on, secrets encryption cannot be modified or removed.

○ Turn on envelope encryption of Kubernetes secrets using KMS
Envelope encryption provides an additional layer of encryption for your Kubernetes secrets.

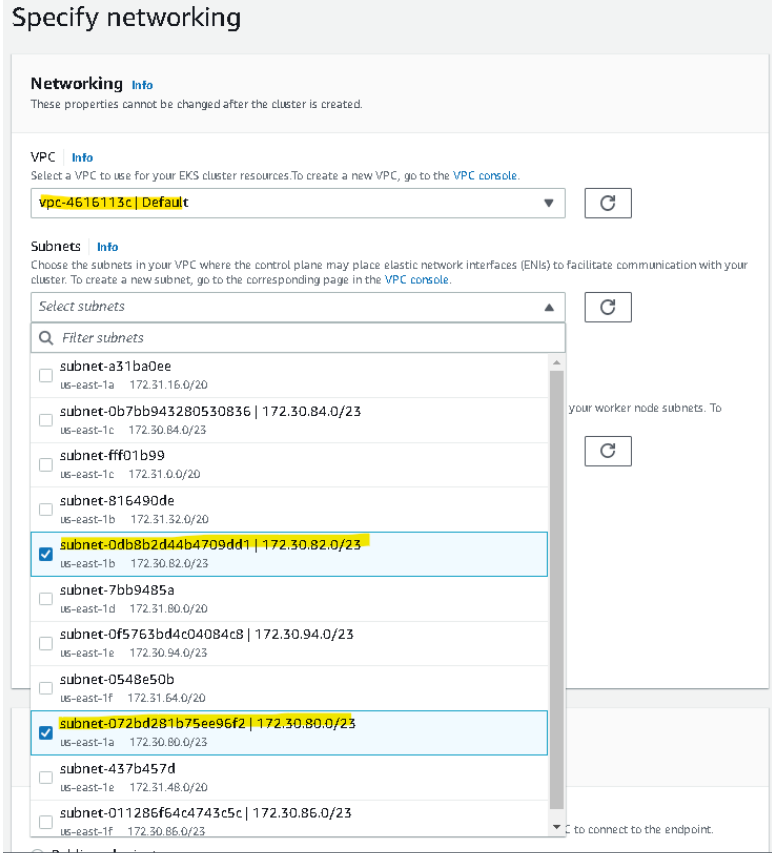
Tags (0) Info

This cluster does not have any tags.

Remaining tags available to add: 50

2. Specify VPC and subnets to be used. EKS cluster creation wizard adds all subnets available in the VPC by default. Remove any unwanted subnets. For this reference architecture only two subnets in two availability zones are required.

Figure 86: Set networking.



Specify networking

Networking Info
These properties cannot be changed after the cluster is created.

VPC Info
Select a VPC to use for your EKS cluster resources.To create a new VPC, go to the VPC console.
vpc-4616113c | Default

Subnets Info
Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the VPC console.
Select subnets

Filter subnets

□ subnet-a31ba0ee
us-east-1a 172.31.16.0/20

□ subnet-0b7bb943280530836 | 172.30.84.0/23
us-east-1c 172.30.84.0/23

□ subnet-fff01b99
us-east-1c 172.31.0.0/20

□ subnet-816490de
us-east-1b 172.31.32.0/20

☑ subnet-0db8b2d44b4709dd1 | 172.30.82.0/23
us-east-1b 172.30.82.0/23

□ subnet-7bb9485a
us-east-1d 172.31.80.0/20

□ subnet-0f5763bd4c04084c8 | 172.30.94.0/23
us-east-1e 172.30.94.0/23

□ subnet-0548e50b
us-east-1f 172.31.64.0/20

☑ subnet-072bd281b75ee96f2 | 172.30.80.0/23
us-east-1a 172.30.80.0/23

□ subnet-437b457d
us-east-1e 172.31.48.0/20

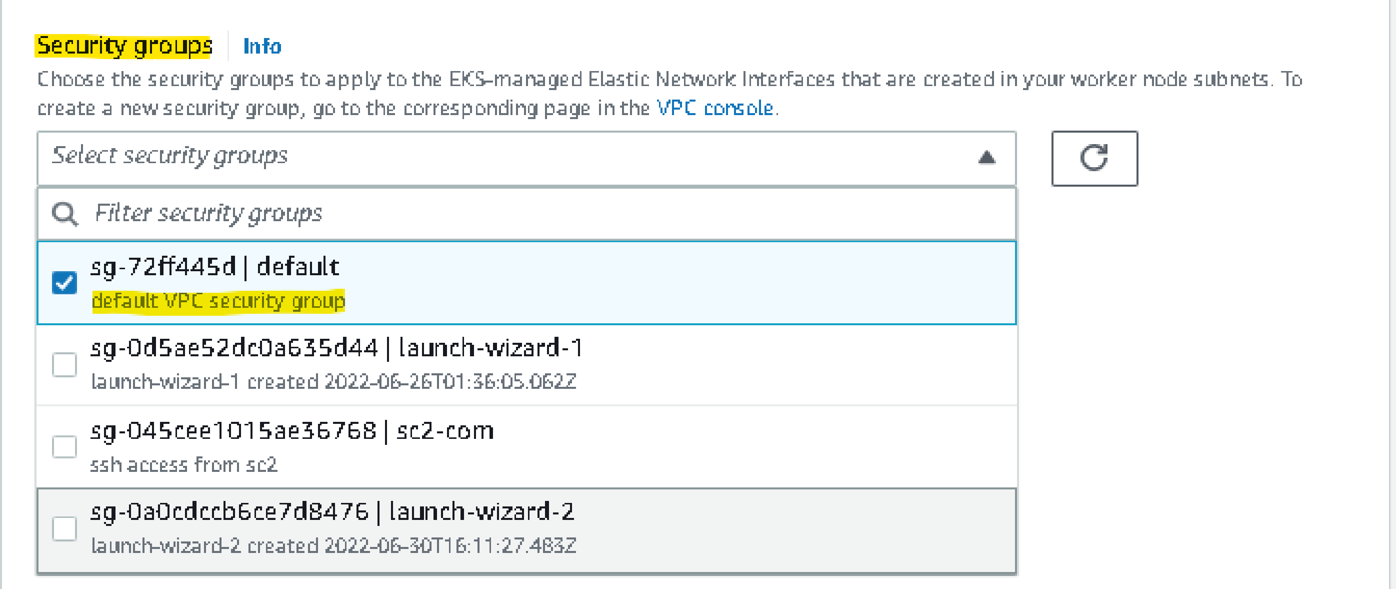
□ subnet-011286f64c4743c5c | 172.30.86.0/23
us-east-1f 172.30.86.0/23

your worker node subnets. To

to connect to the endpoint.

3. Select the desired security group. The default security group for the VPC was selected here.

Figure 87: Security groups



Security groups Info

Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker node subnets. To create a new security group, go to the corresponding page in the VPC console.

Select security groups

Filter security groups

☑ sg-72ff445d | default
default VPC security group

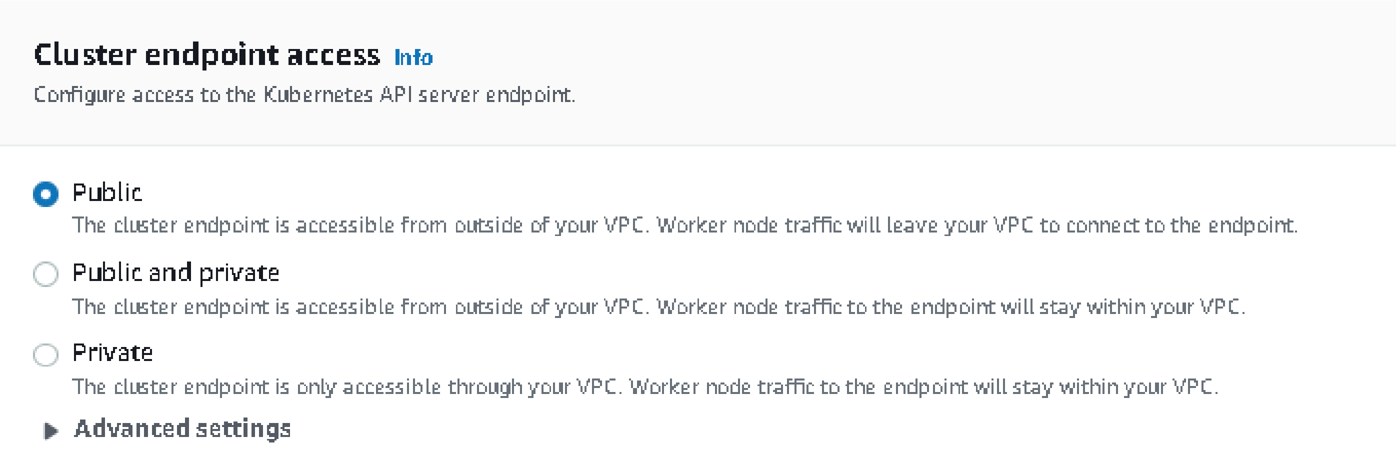
□ sg-0d5ae52dc0a635d44 | launch-wizard-1
launch-wizard-1 created 2022-06-26T01:36:05.062Z

□ sg-045cee1015ae36768 | sc2-com
ssh access from sc2

□ sg-0a0cdccb6ce7d8476 | launch-wizard-2
launch-wizard-2 created 2022-06-30T16:11:27.483Z

4. Select the Cluster endpoint access type. This depends on your environment and cluster requirements. Amazon EKS creates an endpoint for the managed Kubernetes API server that you use to communicate with your cluster (using Kubernetes management tools such as kubectl). By default, this API server endpoint is public to the internet, and access to the API server is secured using a combination of AWS Identity and Access Management (IAM) and native Kubernetes Role Based Access Control (RBAC).

Figure 88: Endpoint access



Cluster endpoint access Info

Configure access to the Kubernetes API server endpoint.

● Public
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

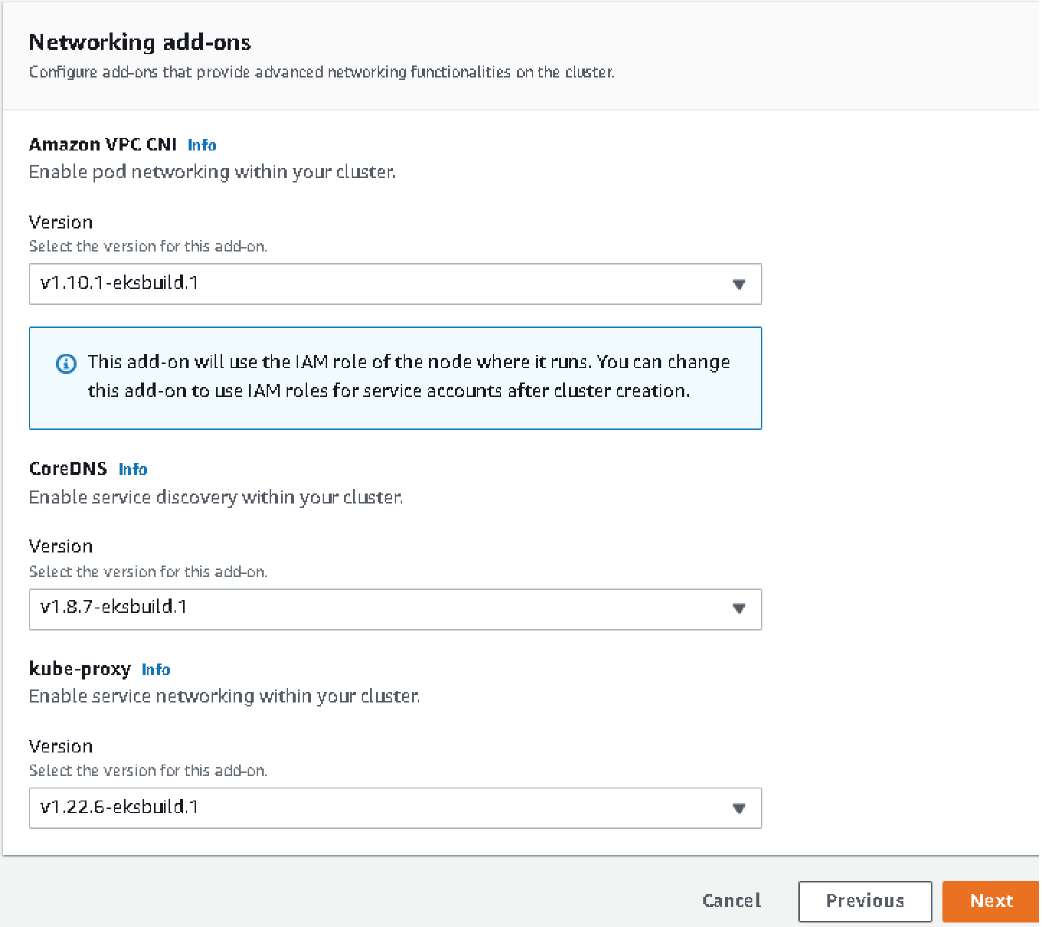
○ Public and private
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

○ Private
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

▶ Advanced settings

5. Select any specific versions of CNI, CoreDNS and Kube-proxy you require. Defaults were used as shown below.

Figure 89: Add-ons



Networking add-ons

Configure add-ons that provide advanced networking functionalities on the cluster.

Amazon VPC CNI Info

Enable pod networking within your cluster.

Version

Select the version for this add-on.

v1.10.1-eksbuild.1

This add-on will use the IAM role of the node where it runs. You can change this add-on to use IAM roles for service accounts after cluster creation.

CoreDNS Info

Enable service discovery within your cluster.

Version

Select the version for this add-on.

v1.8.7-eksbuild.1

kube-proxy Info

Enable service networking within your cluster.

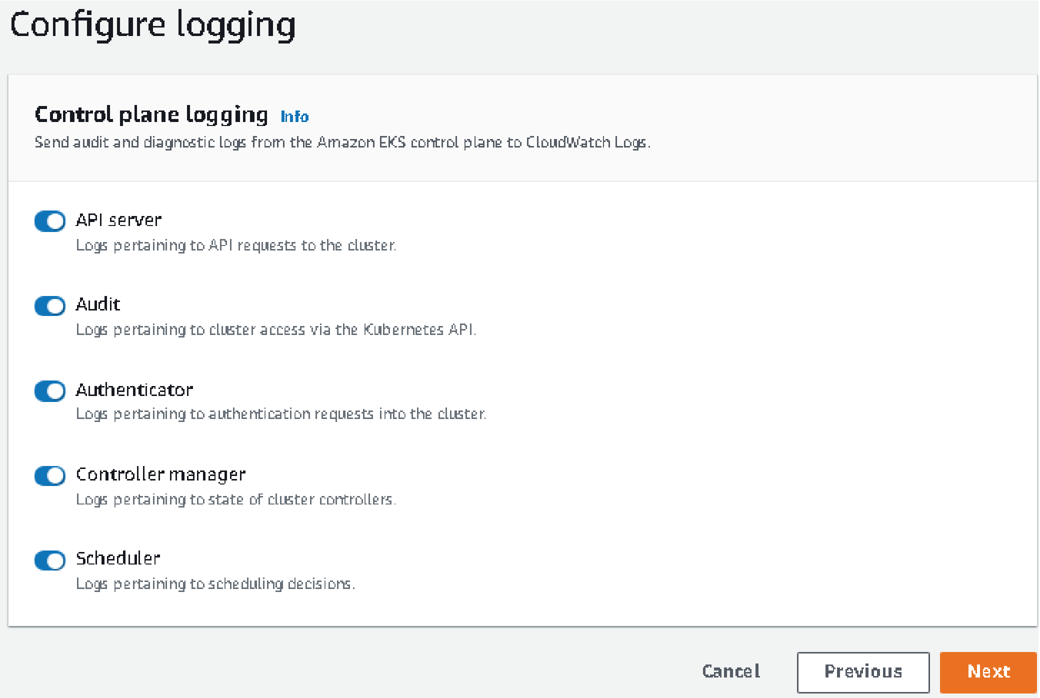
Version

Select the version for this add-on.

v1.22.6-eksbuild.1

6. Configure Logging options.

Figure 90: Logging



Configure logging

Control plane logging Info

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

● API server
Logs pertaining to API requests to the cluster.

● Audit
Logs pertaining to cluster access via the Kubernetes API.

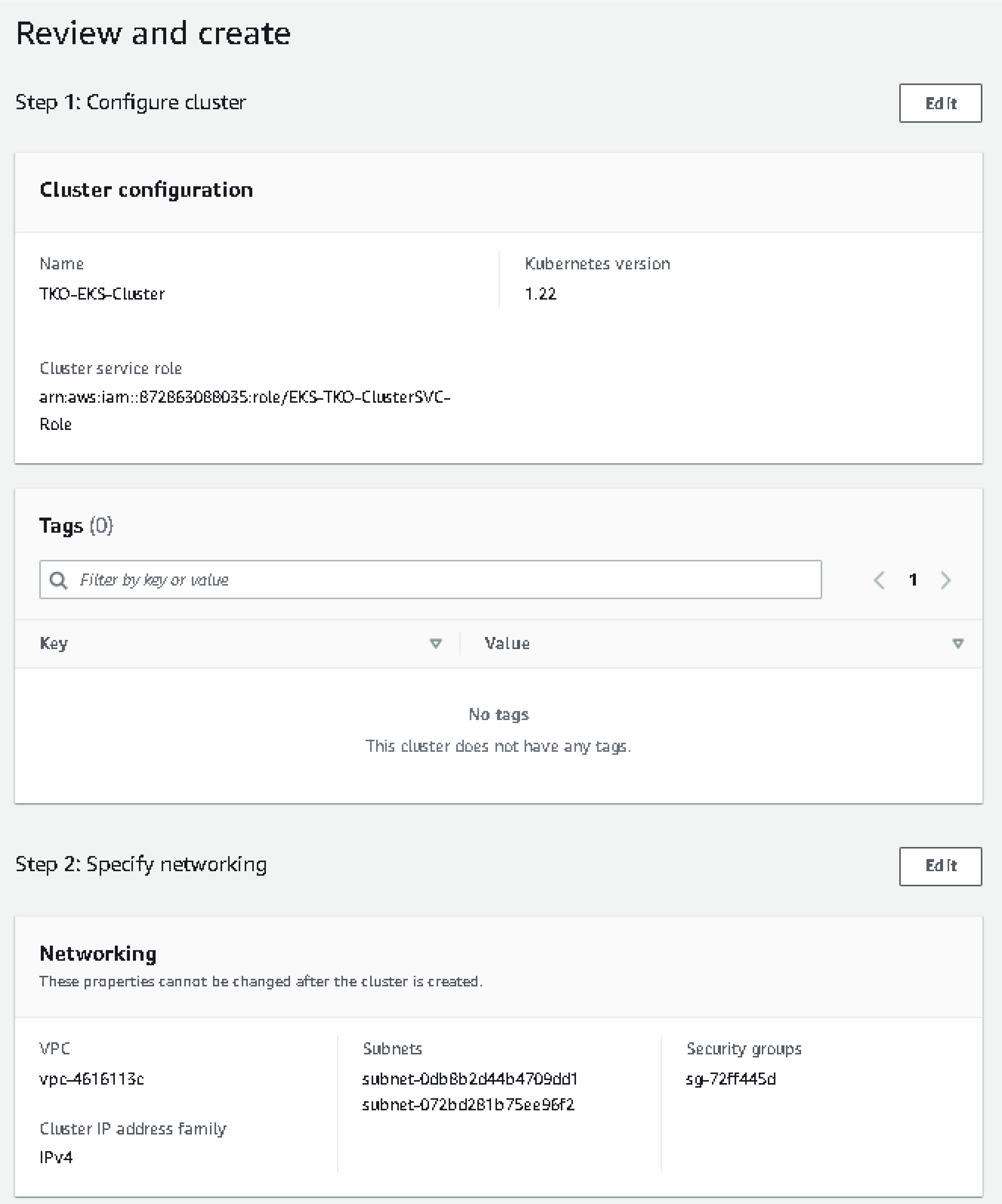
● Authenticator
Logs pertaining to authentication requests into the cluster.

● Controller manager
Logs pertaining to state of cluster controllers.

● Scheduler
Logs pertaining to scheduling decisions.

7. Finally review configuration and create the cluster.

Figure 91: Finish cluster creation



Review and create

Step 1: Configure cluster

Cluster configuration

Name
TKO-EKS-Cluster

Kubernetes version
1.22

Cluster service role
arn:aws:iam::872863088035:role/EKS-TKO-ClusterSVC-Role

Tags (0)

Filter by key or value

Key	Value

No tags
This cluster does not have any tags.

Step 2: Specify networking

Networking
These properties cannot be changed after the cluster is created.

VPC
vpc-4616113c

Cluster IP address family
IPv4

Subnets
subnet-0db8b2d44b4709dd1
subnet-072bd281b75ee96f2

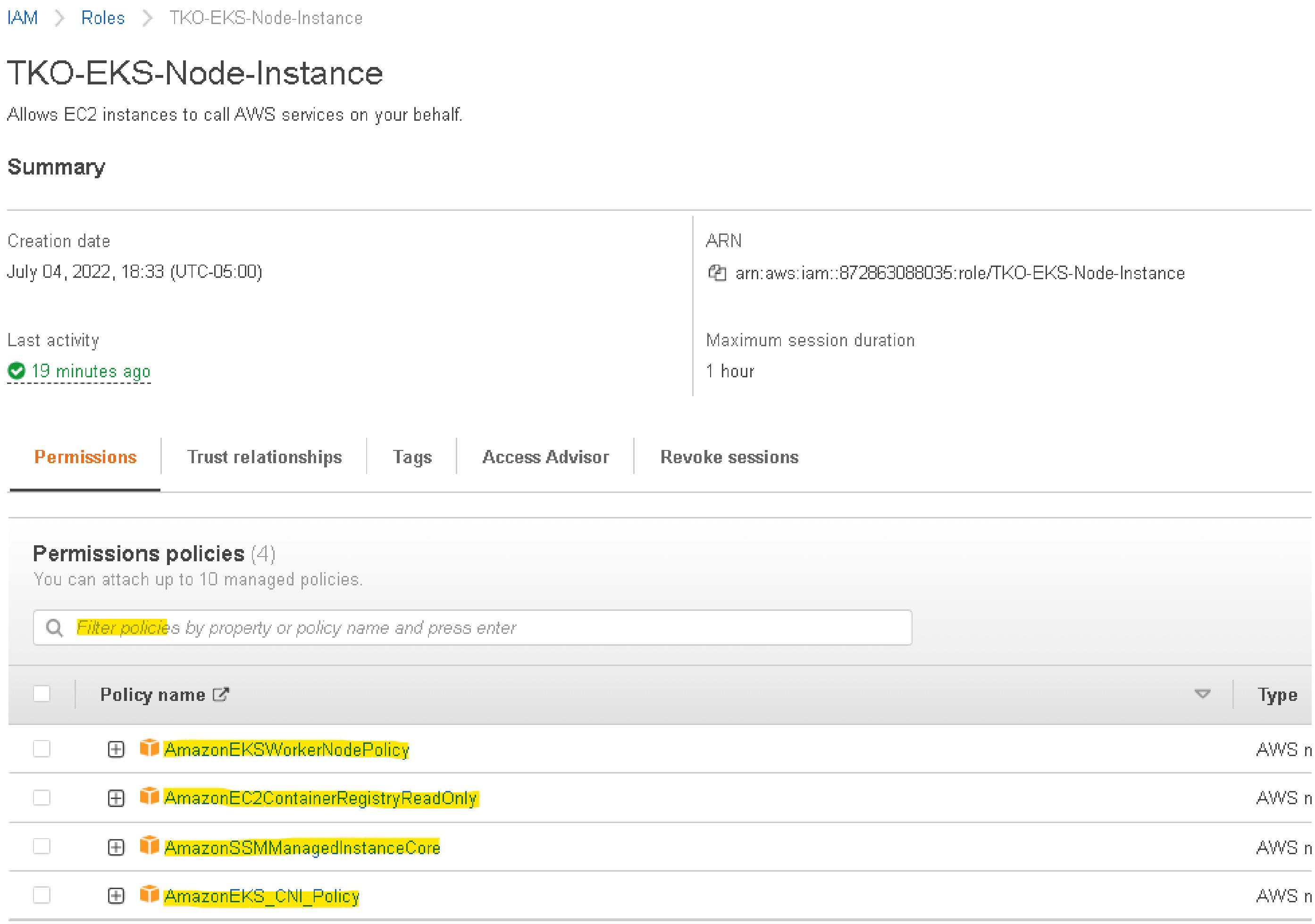
Security groups
sg-72ff445d

Create Node Group

A node group now needs to be created where your workloads will run. These will be Amazon EC2 AMI (Amazon Machine Image) instances. Prior to creating a node group, a "Node Instance Role" needs to be created via IAM > Roles > console.

1. Create node instance role and add the required policies for the EKS nodes. These policies will be used in the next step when creating node group. These are AWS managed policies and can be added by searching via the "Permissions policies" search bar.

Figure 92: Node instance policies



IAM > Roles > TKO-EKS-Node-Instance

TKO-EKS-Node-Instance

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
July 04, 2022, 18:33 (UTC-05:00)

ARN
arn:aws:iam::872863088035:role/TKO-EKS-Node-Instance

Last activity
19 minutes ago

Maximum session duration
1 hour

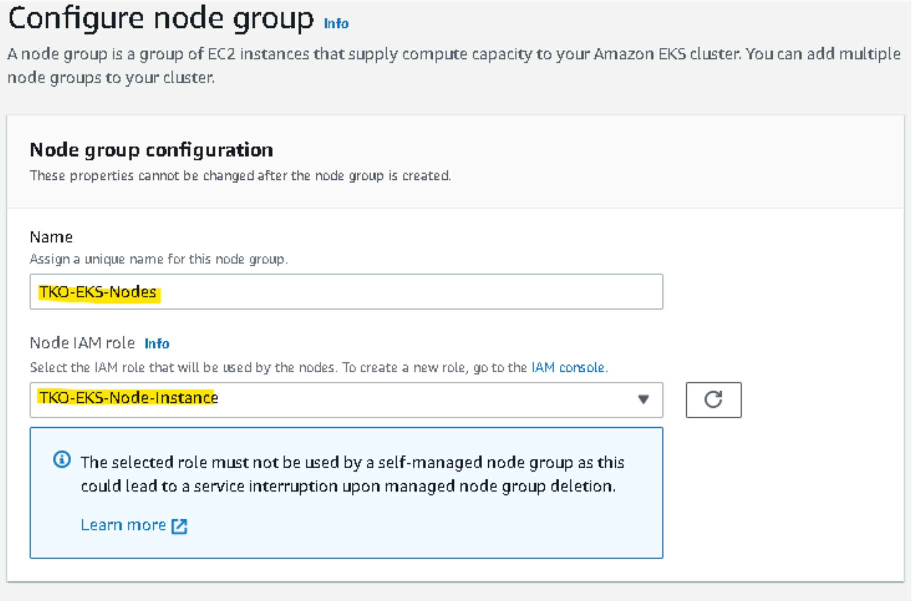
Permissions policies (4)
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

□	Policy name	Type
□	AmazonEKSWorkerNodePolicy	AWS n
□	AmazonEC2ContainerRegistryReadOnly	AWS n
□	AmazonSSMManagedInstanceCore	AWS n
□	AmazonEKS_CNI_Policy	AWS n

2. Navigate to your EKS cluster and click Add node group. Give the node group a name and select the IAM role created earlier. Leave everything else at default values and click next.

Figure 93: Create node group.



Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name
Assign a unique name for this node group.
TKO-EKS-Nodes

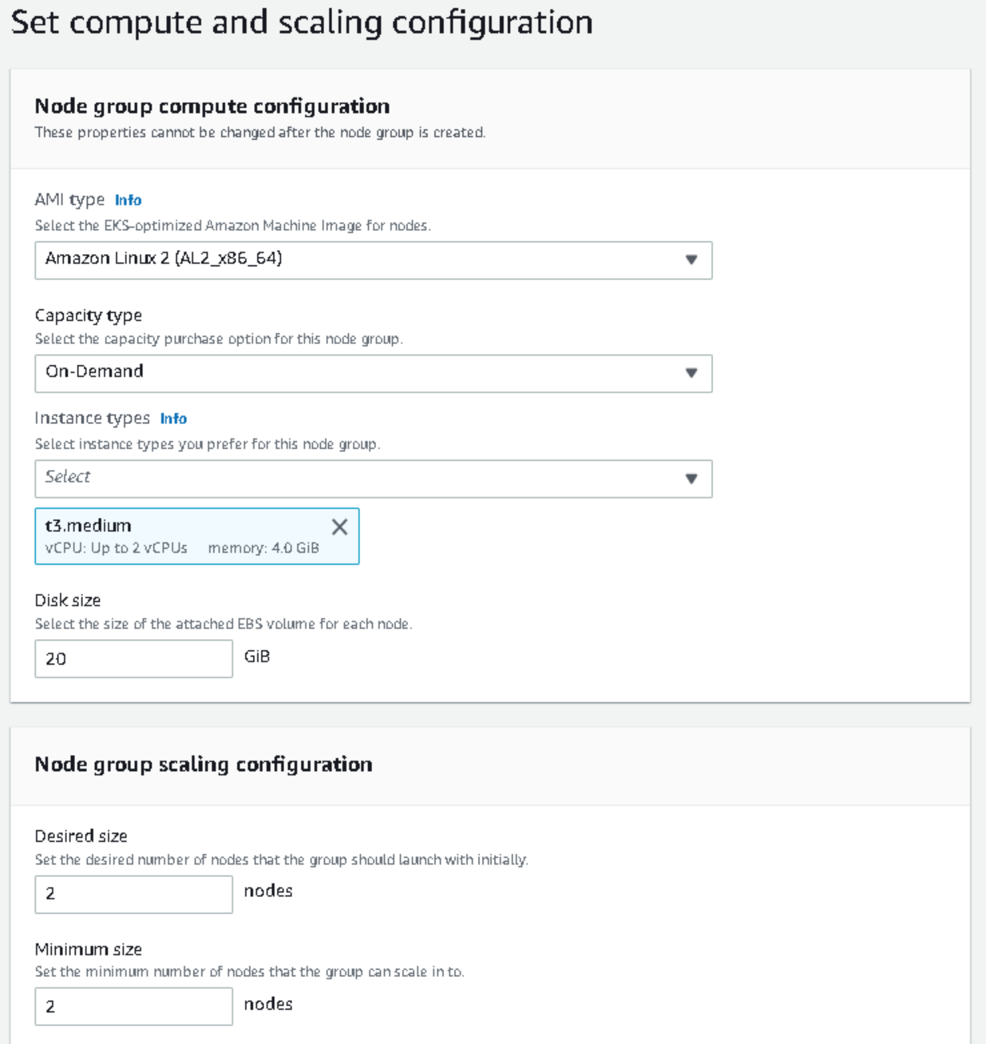
Node IAM role Info
Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.
TKO-EKS-Node-Instance

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

3. Select your desired node configurations and click next.

Figure 94: Set configurations.



Set compute and scaling configuration

Node group compute configuration

These properties cannot be changed after the node group is created.

AMI type Info
Select the EKS-optimized Amazon Machine Image for nodes.
Amazon Linux 2 (AL2_x86_64)

Capacity type
Select the capacity purchase option for this node group.
On-Demand

Instance types Info
Select instance types you prefer for this node group.
Select

t3.medium
vCPU: Up to 2 vCPUs memory: 4.0 GiB

Disk size
Select the size of the attached EBS volume for each node.
20 GiB

Node group scaling configuration

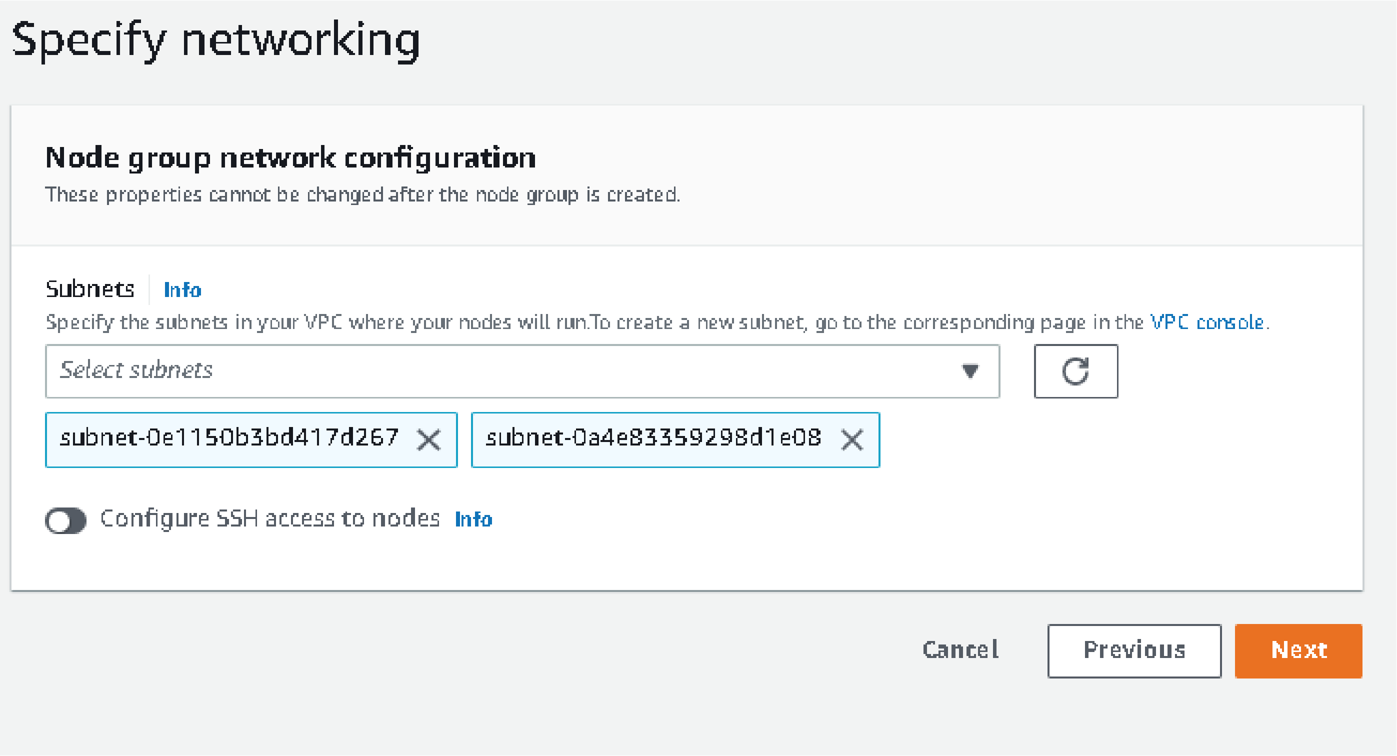
Desired size
Set the desired number of nodes that the group should launch with initially.
2 nodes

Minimum size
Set the minimum number of nodes that the group can scale in to.
2 nodes

4. On the next page the subnets selected during cluster creation will automatically be selected. If node access via SSH, enable the option. On the next screen review and create the node group.

Note: SSH option requires that SSH key pair is already defined. In addition, when enabling this option, managed node groups will create a security group with port 22 inbound access. If launching your worker in a public subnet, it is strongly recommended to restrict the source IP address ranges.

Figure 95: Cluster networking



Specify networking

Node group network configuration

These properties cannot be changed after the node group is created.

Subnets | Info

Specify the subnets in your VPC where your nodes will run.To create a new subnet, go to the corresponding page in the VPC console.

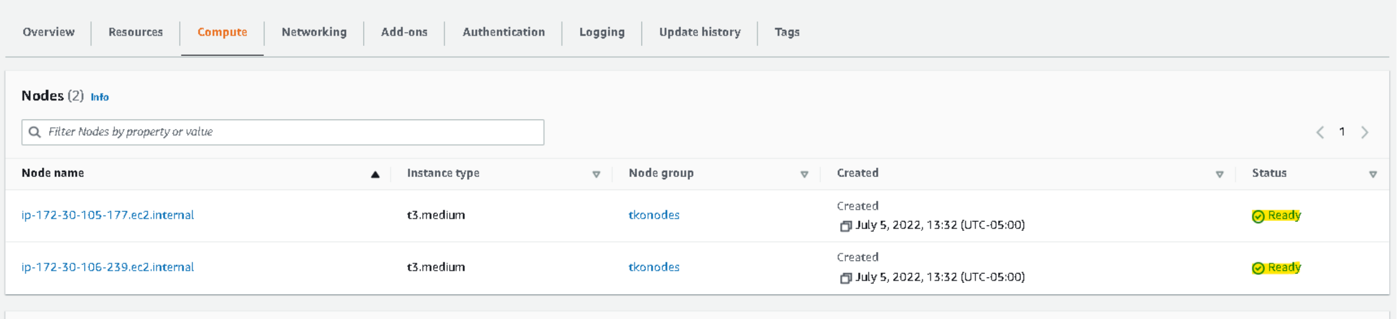
Select subnets ▼

subnet-0e1150b3bd417d267 ✕ subnet-0a4e83359298d1e08 ✕

○ Configure SSH access to nodes Info

5. Once the node group has been created, node status can be verified on the cluster's "Compute" tab.

Figure 96: Node status



Nodes (2) Info

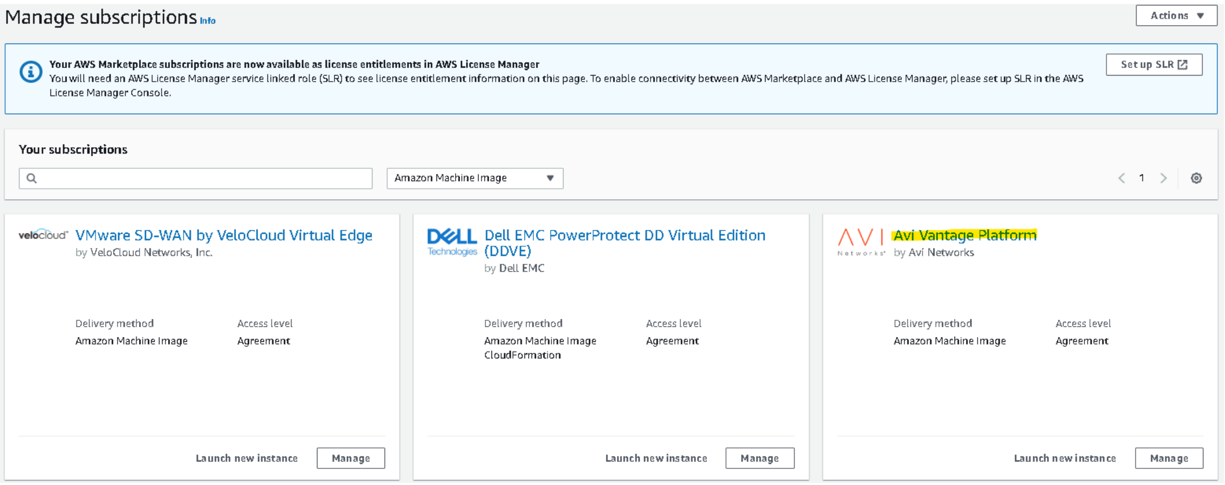
Filter Nodes by property or value

Node name	Instance type	Node group	Created	Status
ip-172-30-105-177.ec2.internal	t3.medium	tkonodes	Created July 5, 2022, 13:32 (UTC-05:00)	✓ Ready
ip-172-30-106-239.ec2.internal	t3.medium	tkonodes	Created July 5, 2022, 13:32 (UTC-05:00)	✓ Ready

Installing NSX Advanced Load Balancer on AWS

NSX Advanced Load Balancer (Avi) is available as a subscription from Amazon Marketplace.

Figure 97: NSX Advanced Load Balancer (Avi) subscription



Manage subscriptions Info

Your AWS Marketplace subscriptions are now available as license entitlements in AWS License Manager
You will need an AWS License Manager service linked role (SLR) to see license entitlement information on this page. To enable connectivity between AWS Marketplace and AWS License Manager, please set up SLR in the AWS License Manager Console.

Your subscriptions

Amazon Machine Image

VMware SD-WAN by VeloCloud Virtual Edge
by VeloCloud Networks, Inc.
Delivery method: Amazon Machine Image
Access level: Agreement

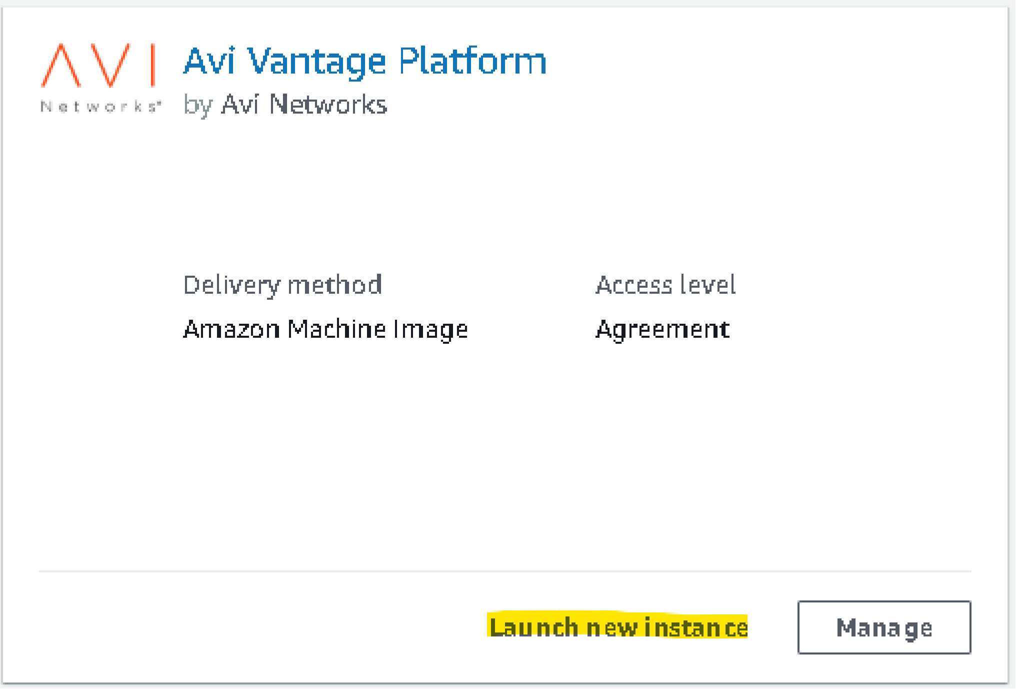
Dell EMC PowerProtect DD Virtual Edition (DDVE)
by Dell EMC
Delivery method: Amazon Machine Image, CloudFormation
Access level: Agreement

Avi Vantage Platform
by Avi Networks
Delivery method: Amazon Machine Image
Access level: Agreement

NSX Advanced load balancer runs as an Amazon AMI on the VPC and provides services comparable to a traditional datacenter installation. Deployment process is as follows. Subscribe to the service prior to launching the AMI installation.

1. With a valid subscription, launch a new instance.

Figure 98: Launch Avi

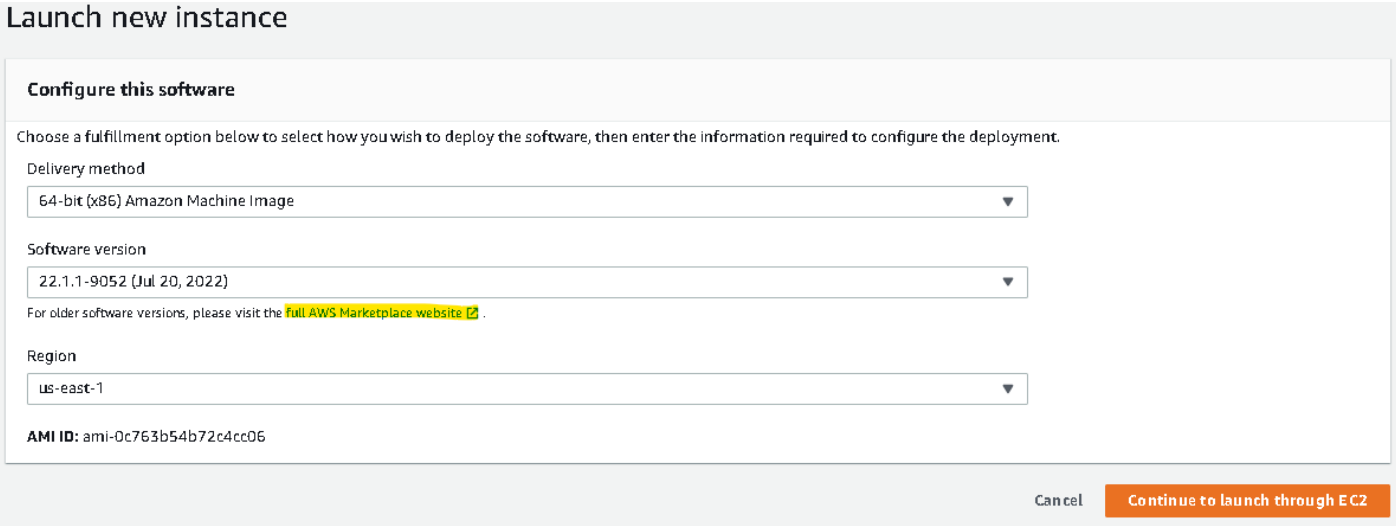


Avi Vantage Platform
by Avi Networks

Delivery method: Amazon Machine Image
Access level: Agreement

2. Select the region and version. By default, only the latest version will be displayed. To change the version, click on the “full AWS Marketplace website” link.

Figure 99: Launch Avi



Launch new instance

Configure this software

Choose a fulfillment option below to select how you wish to deploy the software, then enter the information required to configure the deployment.

Delivery method: 64-bit (x86) Amazon Machine Image

Software version: 22.1.1-9052 (Jul 20, 2022)

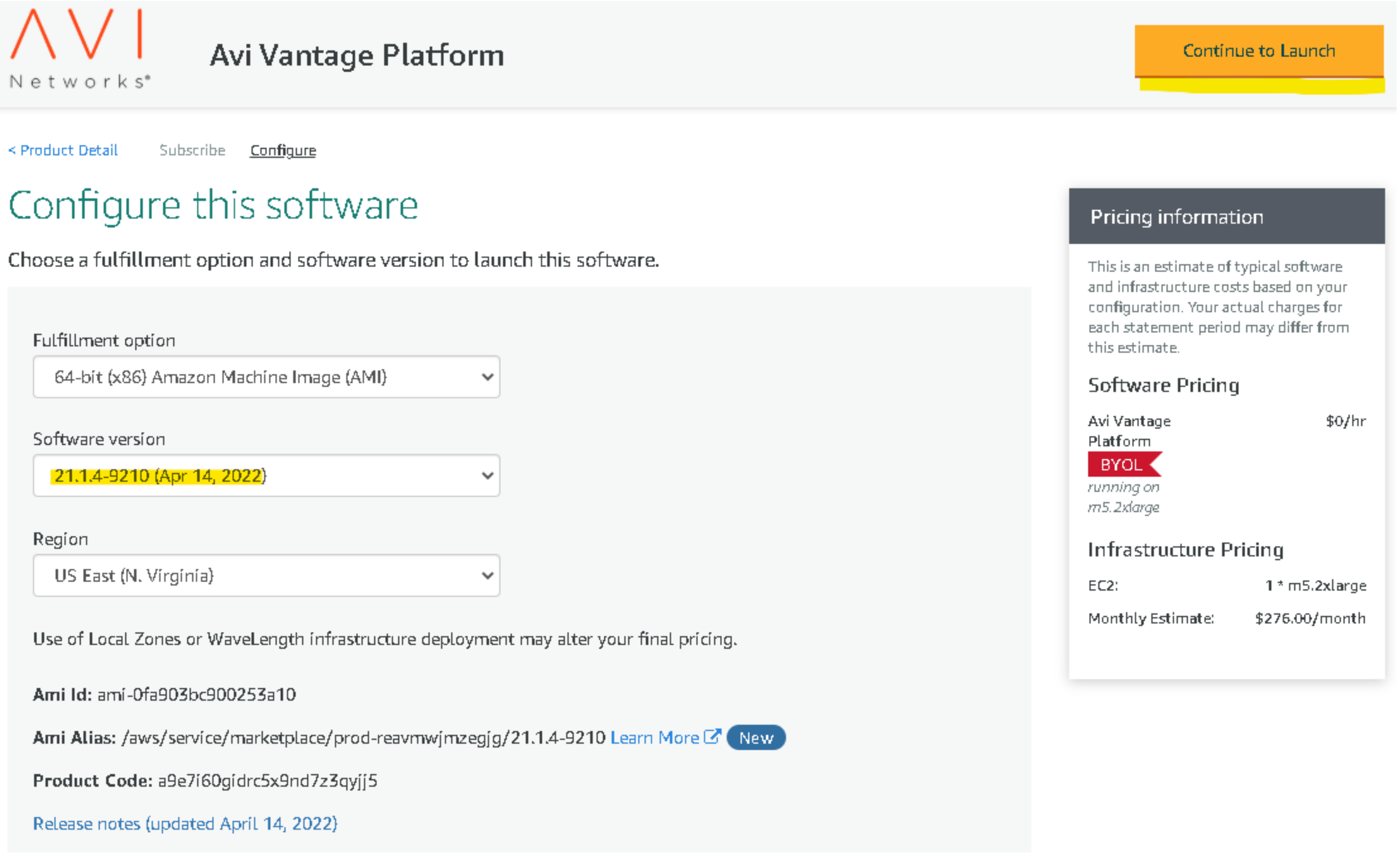
For older software versions, please visit the full AWS Marketplace website.

Region: us-east-1

AMI ID: ami-0c763b54b72c4cc06

3. For this reference architecture 21.1.4 was used. Click "Continue to launch."

Figure 100: Launch Avi



Avi Vantage Platform

< Product Detail Subscribe Configure

Configure this software

Choose a fulfillment option and software version to launch this software.

Fulfillment option: 64-bit (x86) Amazon Machine Image (AMI)

Software version: 21.1.4-9210 (Apr 14, 2022)

Region: US East (N. Virginia)

Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

Ami Id: ami-0fa903bc900253a10

Ami Alias: /aws/service/marketplace/prod-reavmwjmzegjg/21.1.4-9210 Learn More New

Product Code: a9e7i60gidrc5x9nd7z3qyjj5

Release notes (updated April 14, 2022)

Pricing information

This is an estimate of typical software and infrastructure costs based on your configuration. Your actual charges for each statement period may differ from this estimate.

Software Pricing

Avi Vantage Platform BYOL running on m5.2xlarge: $0/hr

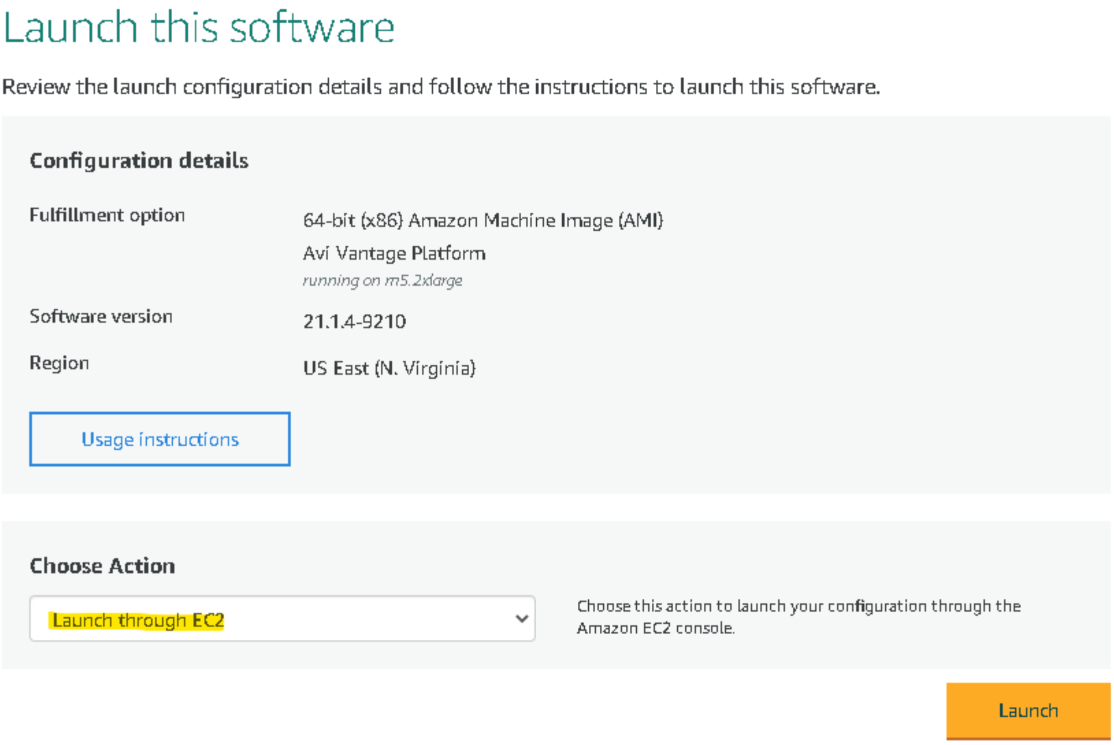
Infrastructure Pricing

EC2: 1 * m5.2xlarge

Monthly Estimate: $276.00/month

4. Select "Launch through EC2" and click "Launch."

Figure 101: Launch Avi



Launch this software

Review the launch configuration details and follow the instructions to launch this software.

Configuration details

Fulfillment option	64-bit (x86) Amazon Machine Image (AMI) Avi Vantage Platform *running on m5.2xlarge*
Software version	21.1.4-9210
Region	US East (N. Virginia)

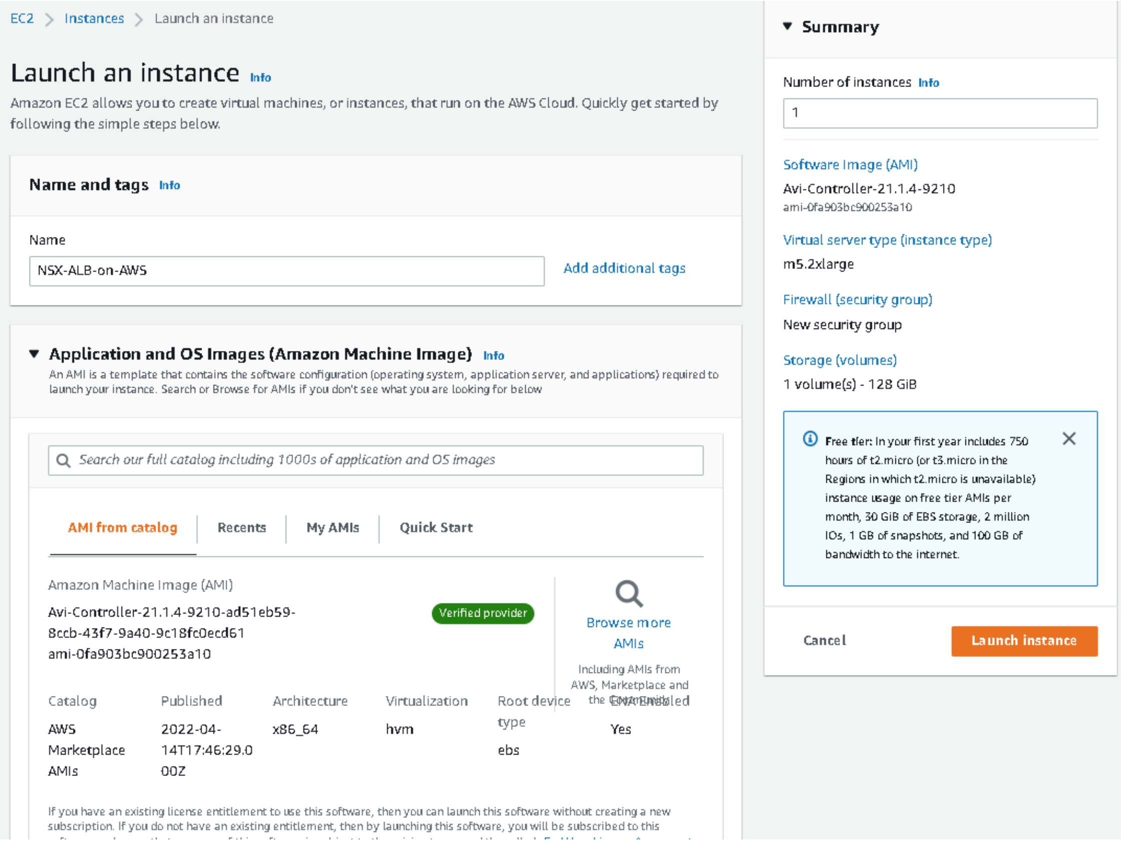
Choose Action

Launch through EC2

Choose this action to launch your configuration through the Amazon EC2 console.

5. On the next page, give the instance a name.

Figure 102: Launch Avi



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: NSX-ALB-on-AWS

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
AMI from catalog
Recents
My AMIs
Quick Start
Verified provider
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)
Avi-Controller-21.1.4-9210-ad51eb59-8ccb-43f7-9a40-9c18fc0ecd61
ami-0fa903bc900253a10

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
AWS Marketplace AMIs	2022-04-14T17:46:29.000Z	x86_64	hvm	ebs	Yes

If you have an existing license entitlement to use this software, then you can launch this software without creating a new subscription. If you do not have an existing entitlement, then by launching this software, you will be subscribed to this

Summary

Number of instances Info: 1

Software Image (AMI)
Avi-Controller-21.1.4-9210
ami-0fa903bc900253a10

Virtual server type (instance type)
m5.2xlarge

Firewall (security group)
New security group

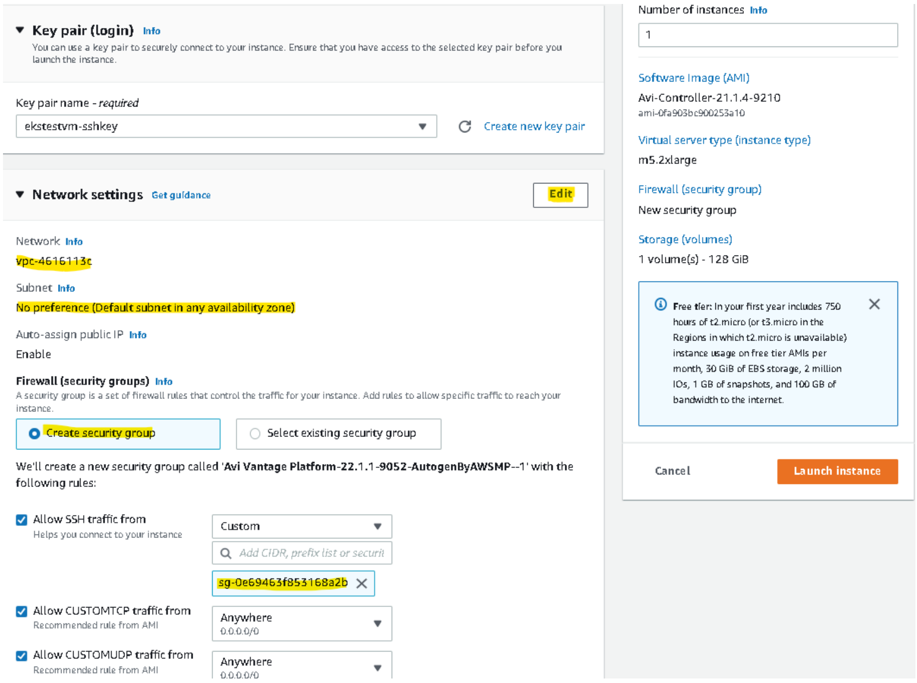
Storage (volumes)
1 volume(s) - 128 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

6. On the same page select a SSH key pair. Create one if one does not exist. This will be needed when accessing the controller via SSH. In "Network settings," verify the VPC, subnet options and traffic rules. Inbound SSH traffic

can be filtered based on security groups or CIDR or IP address bases. Default security group for the VPC is selected in this case. Select "Create security group" option. This option will create a security group for the controller based on recommended settings.

Figure 103: Launch Avi



▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

ekstestvm-sshkey ▼ Create new key pair

▼ Network settings Get guidance Edit

Network Info

vpc-4616113c

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**Avi Vantage Platform-22.1.1-9052-AutogenByAWSMP--1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance
Custom ▼
Add CIDR, prefix list or securit
sg-0e69463f853168a2b ✕

☑ Allow CUSTOMTCP traffic from
Recommended rule from AMI
Anywhere
0.0.0.0/0 ▼

☑ Allow CUSTOMUDP traffic from
Recommended rule from AMI
Anywhere
0.0.0.0/0 ▼

Number of instances Info

1

Software Image (AMI)

Avi-Controller-21.1.4-9210
ami-0fa903bc900253a10

Virtual server type (instance type)

m5.2xlarge

Firewall (security group)

New security group

Storage (volumes)

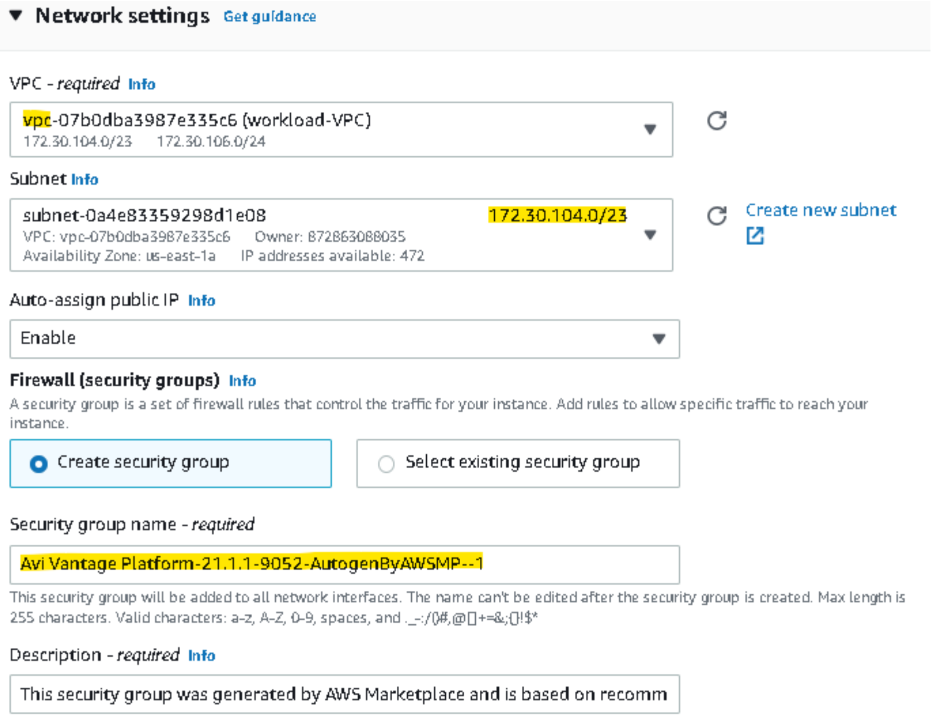
1 volume(s) - 128 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

7. Edit the network configuration to modify the default settings for VPC and subnets. This will also show you the name of the security group that will be created as well as VPC and subnet selection options will be available.

Figure 104: Launch Avi



▼ **Network settings** Get guidance

VPC - *required* Info

vpc-07b0dba3987e335c6 (workload-VPC)
172.30.104.0/23 172.30.106.0/24 ▼

Subnet Info

subnet-0a4e83359298d1e08 172.30.104.0/23
VPC: vpc-07b0dba3987e335c6 Owner: 872863088035
Availability Zone: us-east-1a IP addresses available: 472 ▼

Create new subnet

Auto-assign public IP Info

Enable ▼

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*

Avi Vantage Platform-21.1.1-9052-AutogenByAWSMP--1

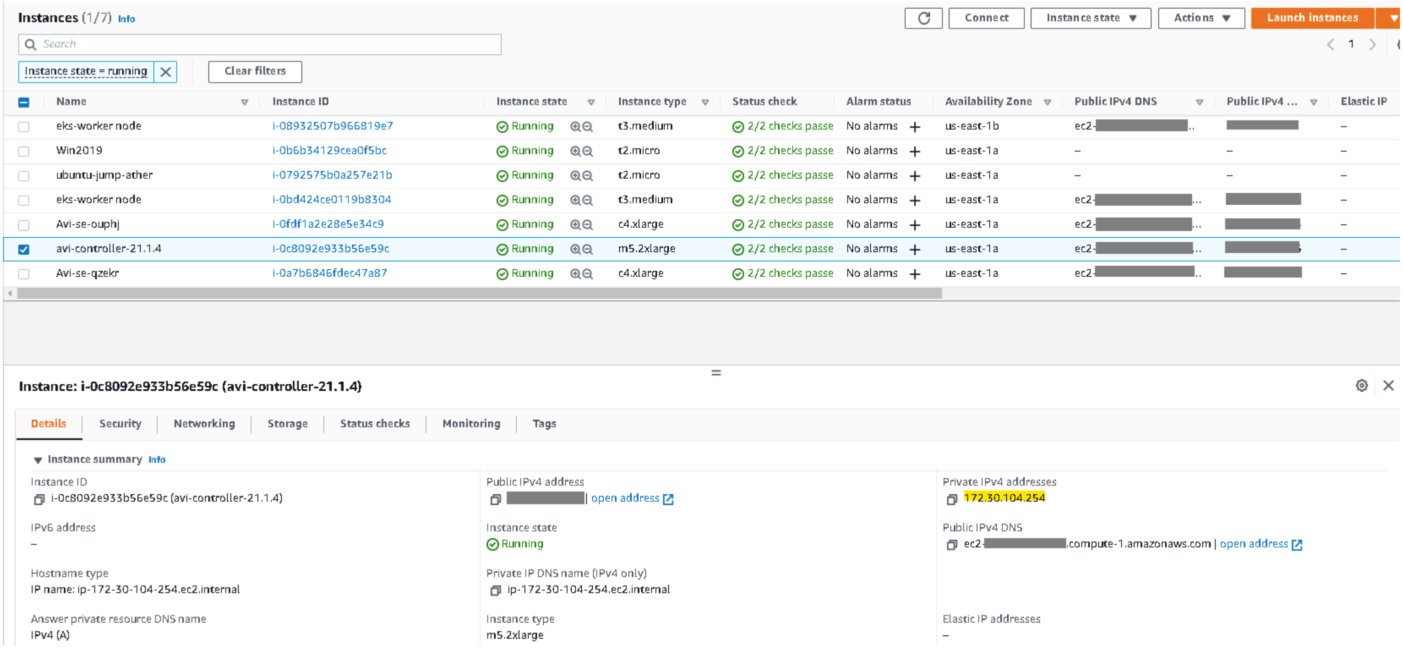
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

This security group was generated by AWS Marketplace and is based on recomm

8. Click Launch Instance and the deployment process will start. This will take a few minutes. Once the process is complete; the controller will be available in EC2 console and can be accessed from the public or private IPs assigned during provisioning.

Figure 105: Avi created.



Instances (1/7) Info

Instance state = running ✕

▬	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	eks-worker node	i-08932507b966819e7	✓ Running	t3.medium	✓ 2/2 checks passe	No alarms +	us-east-1b	ec2-[redacted]..	[redacted]	–
□	Win2019	i-0b6b34129cea0f5bc	✓ Running	t2.micro	✓ 2/2 checks passe	No alarms +	us-east-1a	–	–	–
□	ubuntu-jump-ather	i-0792575b0a257e21b	✓ Running	t2.micro	✓ 2/2 checks passe	No alarms +	us-east-1a	–	–	–
□	eks-worker node	i-0bd424ce0119b8304	✓ Running	t3.medium	✓ 2/2 checks passe	No alarms +	us-east-1a	ec2-[redacted]...	[redacted]	–
□	Avi-se-ouphj	i-0fdf1a2e28e5e34c9	✓ Running	c4.xlarge	✓ 2/2 checks passe	No alarms +	us-east-1a	ec2-[redacted]...	[redacted]	–
☑	avi-controller-21.1.4	i-0c8092e933b56e59c	✓ Running	m5.2xlarge	✓ 2/2 checks passe	No alarms +	us-east-1a	ec2-[redacted]...	[redacted]	–
□	Avi-se-qzekr	i-0a7b6846fdec47a87	✓ Running	c4.xlarge	✓ 2/2 checks passe	No alarms +	us-east-1a	ec2-[redacted]..	[redacted]	–

Instance: i-0c8092e933b56e59c (avi-controller-21.1.4)

▼ Instance summary Info

Instance ID
i-0c8092e933b56e59c (avi-controller-21.1.4)

IPv6 address
–

Hostname type
IP name: ip-172-30-104-254.ec2.internal

Answer private resource DNS name
IPv4 (A)

Public IPv4 address
[redacted] | open address

Instance state
✓ Running

Private IP DNS name (IPv4 only)
ip-172-30-104-254.ec2.internal

Instance type
m5.2xlarge

Private IPv4 addresses
172.30.104.254

Public IPv4 DNS
ec2-[redacted].compute-1.amazonaws.com | open address

Elastic IP addresses
–

Configuring NSX Advanced Load Balancer on AWS

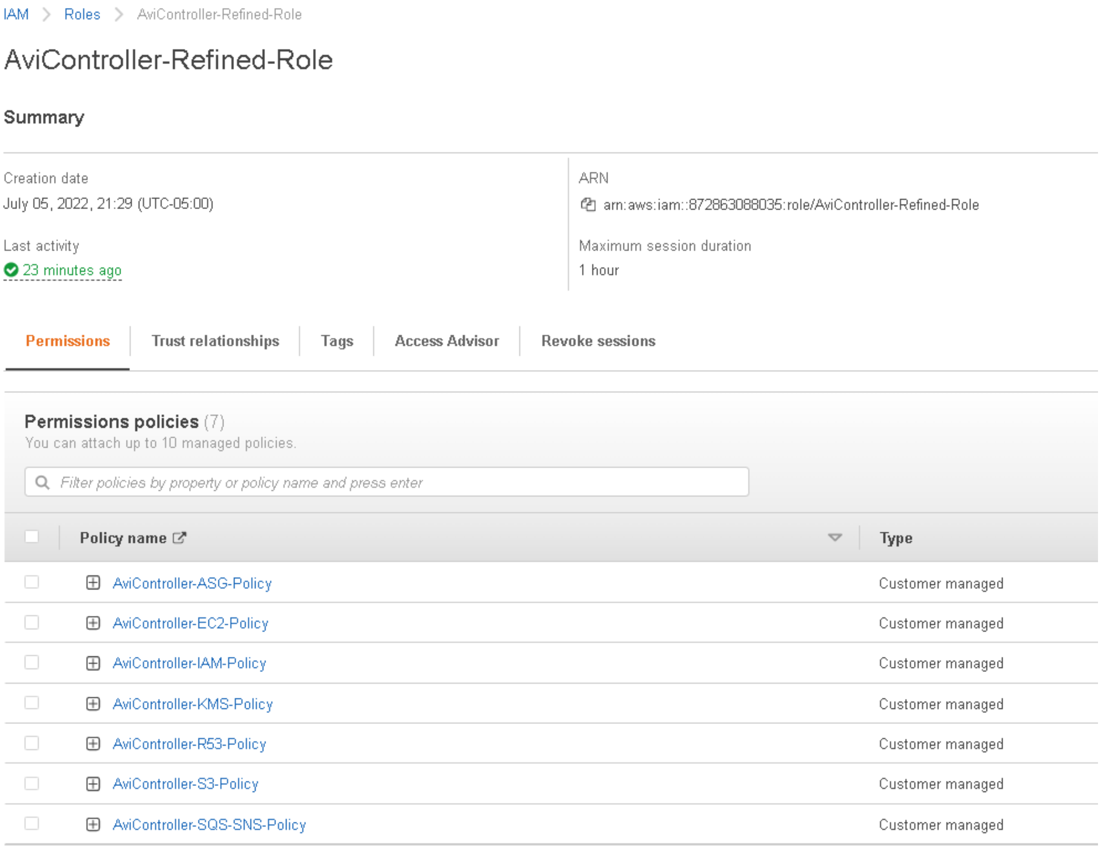
Prior to configuring NSX Advanced Load Balancer on AWS a credential method must be chosen. There are several methods AWS credential can be assigned to NSX Advanced Load Balancer components.

- **AWS customer account key:** A unique authentication key associated with the AWS account. Access credentials are needed by the NSX Advanced Load Balancer Controller to communicate with AWS APIs.

  Note: AWS cloud configuration with NSX Advanced Load Balancer SaaS only supports the Use Access/Secret Key credentials method.
- **Identity and Access Management (IAM) roles:** IAM roles are the set of policies that define access to resources within AWS. The roles and the policies that define their access are defined in JSON files. This method does not require an AWS account key. Instead, the role and policy files must be downloaded from NSX Advanced Load Balancer and installed using the AWS CLI. Use this method if you do not want to enter AWS credentials.
- **Use Cross-Account AssumeRole:** NSX Advanced Load Balancer can be deployed for Amazon Web Services (AWS) with multiple AWS accounts utilizing the IAM AssumeRole functionality that provides access across AWS accounts to the AWS resources/API from the respective accounts, instead of sharing user Access Key ID and Secret Access Key from different accounts.
- For the detailed information on Cross-Account AssumeRole, refer to AWS Cross-Account AssumeRole Support.

For this deployment **IAM roles** method was used. This method does not require a customer account key. Instructions on creating AWS required roles can be found here These roles are required for NSX Advanced Load Balancer to gain access to AWS objects including AMI images that will be used to deploy service engine. Once the roles are configured, they will appear in AWS IAM > Roles.

Figure 106: Avi IAM refined role.


IAM > Roles > AviController-Refined-Role

AviController-Refined-Role

Summary

Creation date
July 05, 2022, 21:29 (UTC-05:00)

Last activity
23 minutes ago

ARN
arn:aws:iam::872863088035:role/AviController-Refined-Role

Maximum session duration
1 hour

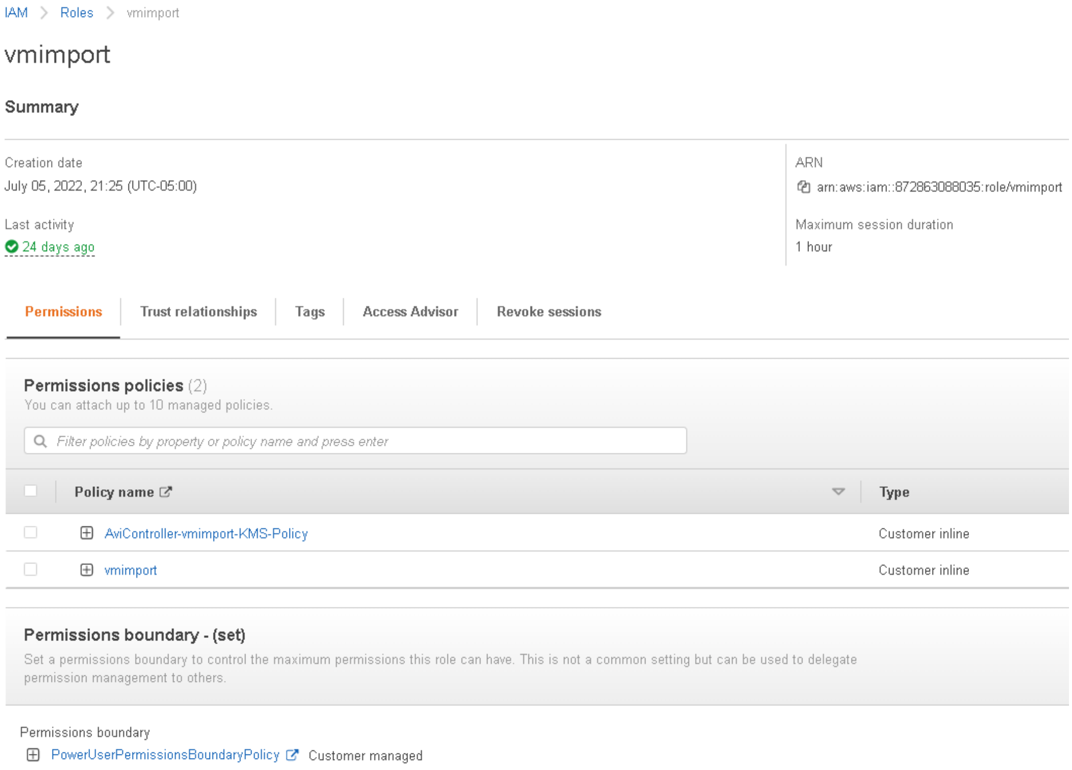
Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (7)
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

Policy name	Type
AviController-ASG-Policy	Customer managed
AviController-EC2-Policy	Customer managed
AviController-IAM-Policy	Customer managed
AviController-KMS-Policy	Customer managed
AviController-R53-Policy	Customer managed
AviController-S3-Policy	Customer managed
AviController-SQS-SNS-Policy	Customer managed

Figure 107: Avi vmimport role



vmimport

Summary

Creation date
July 05, 2022, 21:25 (UTC-05:00)

Last activity
24 days ago

ARN
arn:aws:iam::872863088035:role/vmimport

Maximum session duration
1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (2)
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

Policy name	Type
AviController-vmimport-KMS-Policy	Customer inline
vmimport	Customer inline

Permissions boundary - (set)
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

Permissions boundary
PowerUserPermissionsBoundaryPolicy Customer managed

Once the roles are configured. NSX Advanced Load Balancer can be configured using the public or private IP address. Complete the NSX Advanced Load Balancer configuration per environment requirements. Instructions to configure NSX Advanced Load Balancer after provisioning can be accessed here.

Global DNS Namespace and Ingress

NSX Advanced Load Balancer is the key component that facilities automated application and resource discovery by building a unified and global DNS namespace. When an application is created, a service URL is automatically created by NSX Advanced Load Balancer components through which users can access the applications. AKO and AMKO are needed for multi-cluster ingress for services.

AKO and AMKO installation

Both AKO and AMKO are required for multi-cluster ingress and load balancing. AKO is installed on all participating clusters, where AMKO is installed on the leader cluster. An optional second instance of AMKO can be installed on a second cluster for redundancy. For this reference architecture a single instance of AMKO was installed that coordinated with AKO instances on all clusters.

AKO and AMKO can be installed Helm Charts. Helm is a pre-requisite and installation instructions can be found here. A high-level process is as follows.

1. Install Helm.
2. Create avi-system namespace “*kubectl create ns avi-system*.”
3. Add AKO Helm chart “*helm repo add ako https://projects.registry.vmware.com/chartrepo/ako*”
4. Export AKO parameters to values.yaml file “helm show values ako/ako --version 1.7.1 > values.yaml”
5. Edit the value.yaml file per your environmental requirements. A sample yaml file used to deploy AKO on EKS clusters is in Appendix A for reference.
6. Install AKO: “*helm install ako/ako --generate-name --version <AKO version> -f values.yaml --set ControllerSettings.controllerHost=<IP of AVi controller> --set avicredentials.username=admin --set avicredentials.password=<controller password> --namespace=avi-system*”

   AMKO is installed on the GSLB leader cluster in similar manner. For more information on value file parameters visit Install and Configure AMKO page. This page also provides information on AMKO federation, if required by the environment. For this reference architecture non-federated AMKO architecture was used.

Note: Yaml files exported during the above process sometimes do not output the yaml in a correct format. Double check the correct yaml format of these yaml files before installation, by using a yaml verification tool such as yamllint or another similar application.

Note: AWS creates external load balancer for service type “loadbalancer” by default. To avoid the creation of AWS load balancer, annotate the service with “ *service.beta.kubernetes.io/aws-load-balancer-internal: "true"* “

Configuring GSLB with NSX Advanced Load Balancer

Configuring NSX Advanced Load balancer to serve as a global DNS namespace provider is a multi-step process and depends on the use case at hand. Below are some of the use cases for use of GSLB functionality.

GSLB Use Cases:

- Optimal application experience for geographically distributed users
  - Multiple applications are deployed in multiple data centers.
  - Avi GSLB can steer user traffic to the most optimal location.
- Application high availability across data center failures
  - Applications are deployed in multiple data centers.
  - In case of a data center failure, application instances running in the remaining data center(s) can take over the user traffic.
- Disaster recovery
  - Applications are deployed in two data centers.
  - While both are healthy, all traffic is directed to the primary DC.

  - If the primary DC fails, the global DNS directs all user traffic to the other.
- Hybrid cloud with "cloud bursting"
  - Applications are deployed across private and public clouds.
  - When/if an application experiences an unusually high request load, Avi GSLB "bursts" to the public cloud site to absorb the load.

The GSLB configuration discussed in this reference architecture addresses all the above use cases. "Cloud bursting" however was not tested but can be configured with the same architecture. What follows are the steps to configure GSLB on on-premises private cloud and Amazon EKS clusters. VMC or another public cloud instance can be added using the same steps.

Note: In the configuration example, pse.lab is the primary domain and avitko.pse.lab is the subdomain. In this architecture DNS authoritative server for **pse.lab** domain lives on the on-premises datacenter which is also the GSLB leader site. This is not a requirement. Authoritative DNS server can be on a follower site or integrated with external DNS service such as Route53. Subdomain **avitko.pse.lab** will be delegated to NSX Advanced Load Balancers which will serve requests to applications hosted on the sub-domain. For example, request to "app.avitko.pse.lab" will be resolved by the NSX Advanced Load Balancers.

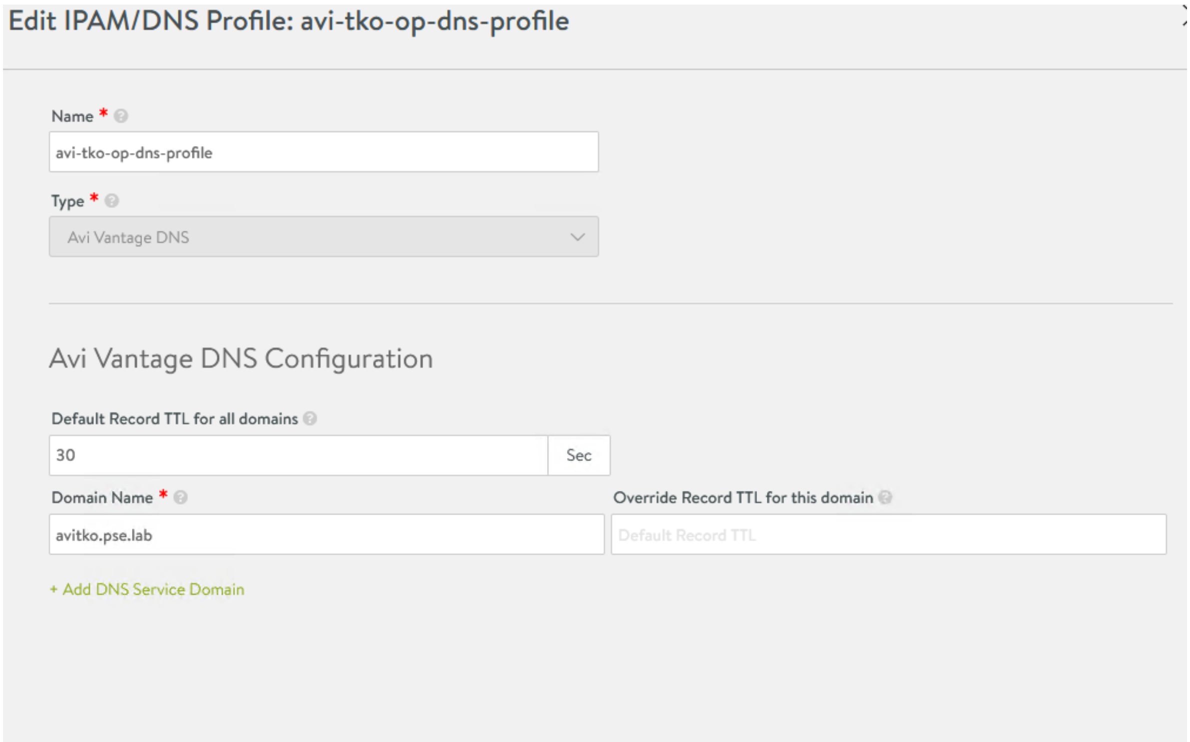
Configuring Sites for GSLB

Create DNS virtual service:

Following are the steps for GSLB configuration for avi controllers on on-premises and on AWS. The leader controller resides on-premises.

1. On the on-premises Avi controller, create a DNS profile specifying the sub-domain avitko.pse.lab.

Figure 108: DNS profile



Edit IPAM/DNS Profile: avi-tko-op-dns-profile

Name *: avi-tko-op-dns-profile

Type *: Avi Vantage DNS

Avi Vantage DNS Configuration

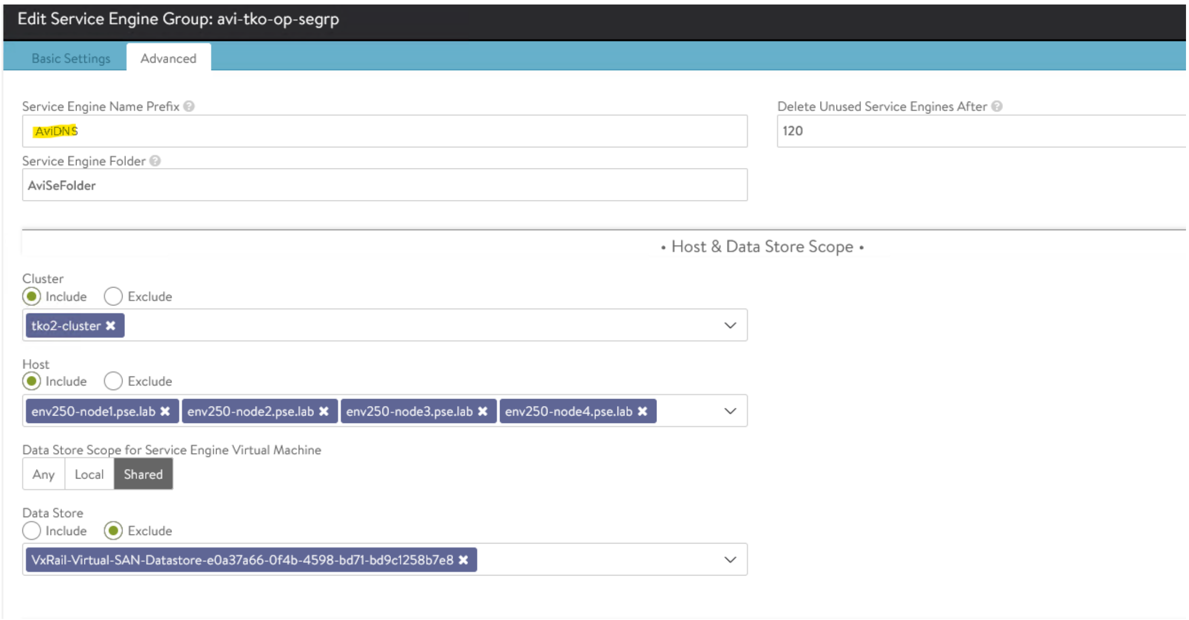
Default Record TTL for all domains: 30 Sec

Domain Name *: avitko.pse.lab

Override Record TTL for this domain: Default Record TTL

2. Create a service engine group for the DNS service. It is a recommended practice that separate service engine group be created for the DNS service. It is also good practice to pre-fix the service engines names with a distinguishable string. Example "Avidns- "

Figure 109: Service engine group



Edit Service Engine Group: avi-tko-op-segrp

Service Engine Name Prefix: AviDNS

Delete Unused Service Engines After: 120

Service Engine Folder: AviSeFolder

• Host & Data Store Scope •

Cluster
● Include ○ Exclude
tko2-cluster ✖

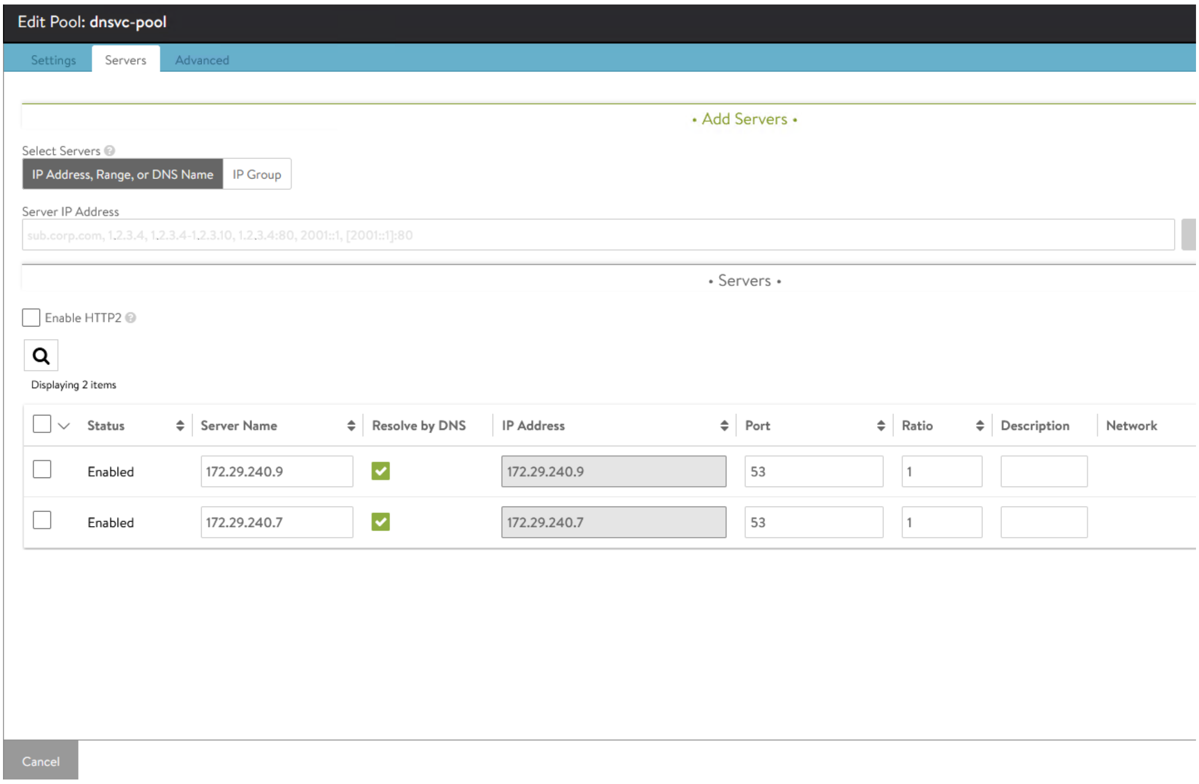
Host
● Include ○ Exclude
env250-node1.pse.lab ✖ env250-node2.pse.lab ✖ env250-node3.pse.lab ✖ env250-node4.pse.lab ✖

Data Store Scope for Service Engine Virtual Machine
Any | Local | [✓ Shared]

Data Store
○ Include ● Exclude
VxRail-Virtual-SAN-Datastore-e0a37a66-0f4b-4598-bd71-bd9c1258b7e8 ✖

3. Create a pool of DNS servers

Figure 110: DNS pool



Edit Pool: **dnsvc-pool**

• Add Servers •

Select Servers
[✓ IP Address, Range, or DNS Name] | IP Group

Server IP Address: sub.corp.com, 1.2.3.4, 1.2.3.4-1.2.3.10, 1.2.3.4:80, 2001::1, [2001::1]:80

• Servers •

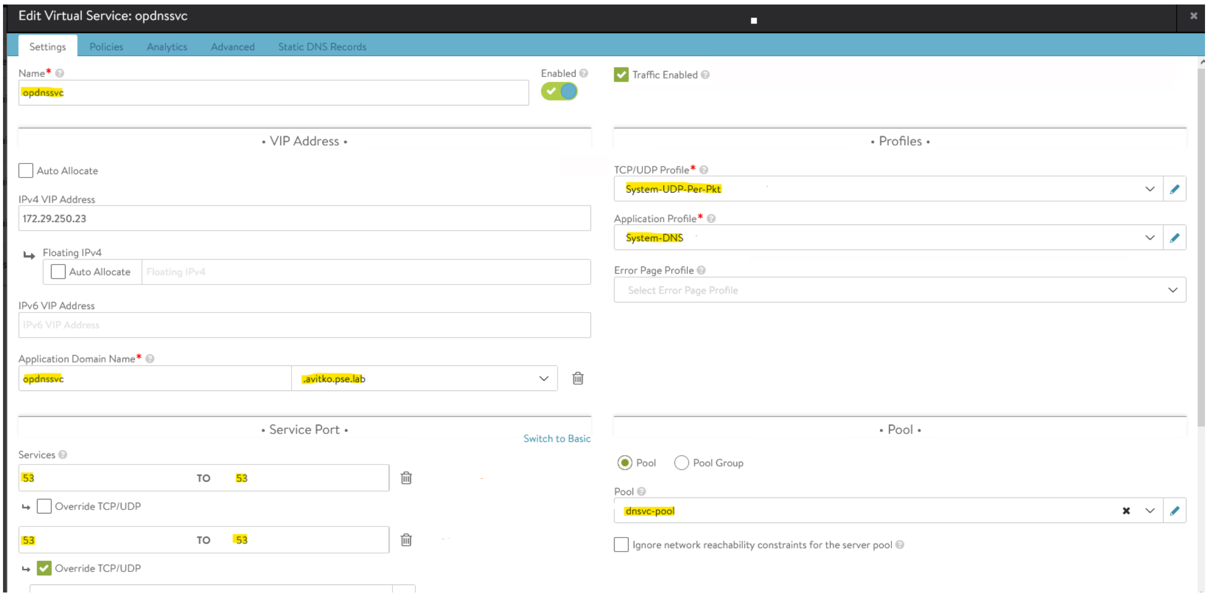
□ Enable HTTP2

Displaying 2 items

□	Status	Server Name	Resolve by DNS	IP Address	Port	Ratio	Description	Network
□	Enabled	172.29.240.9	☑	172.29.240.9	53	1		
□	Enabled	172.29.240.7	☑	172.29.240.7	53	1		

4. Create a virtual service for DNS

Figure 111: Create DNS service.



Edit Virtual Service: opdnssvc

Settings | Policies | Analytics | Advanced | Static DNS Records

Name* opdnssvc

Enabled ●

☑ Traffic Enabled

• VIP Address •

□ Auto Allocate

IPv4 VIP Address: 172.29.250.23

↳ Floating IPv4: □ Auto Allocate | Floating IPv4

IPv6 VIP Address: IPv6 VIP Address

Application Domain Name*: opdnssvc .avitko.pse.lab

• Profiles •

TCP/UDP Profile*: System-UDP-Per-Pkt

Application Profile*: System-DNS

Error Page Profile: Select Error Page Profile

• Service Port •

Switch to Basic

Services

53 TO 53

↳ □ Override TCP/UDP

53 TO 53

↳ ☑ Override TCP/UDP

• Pool •

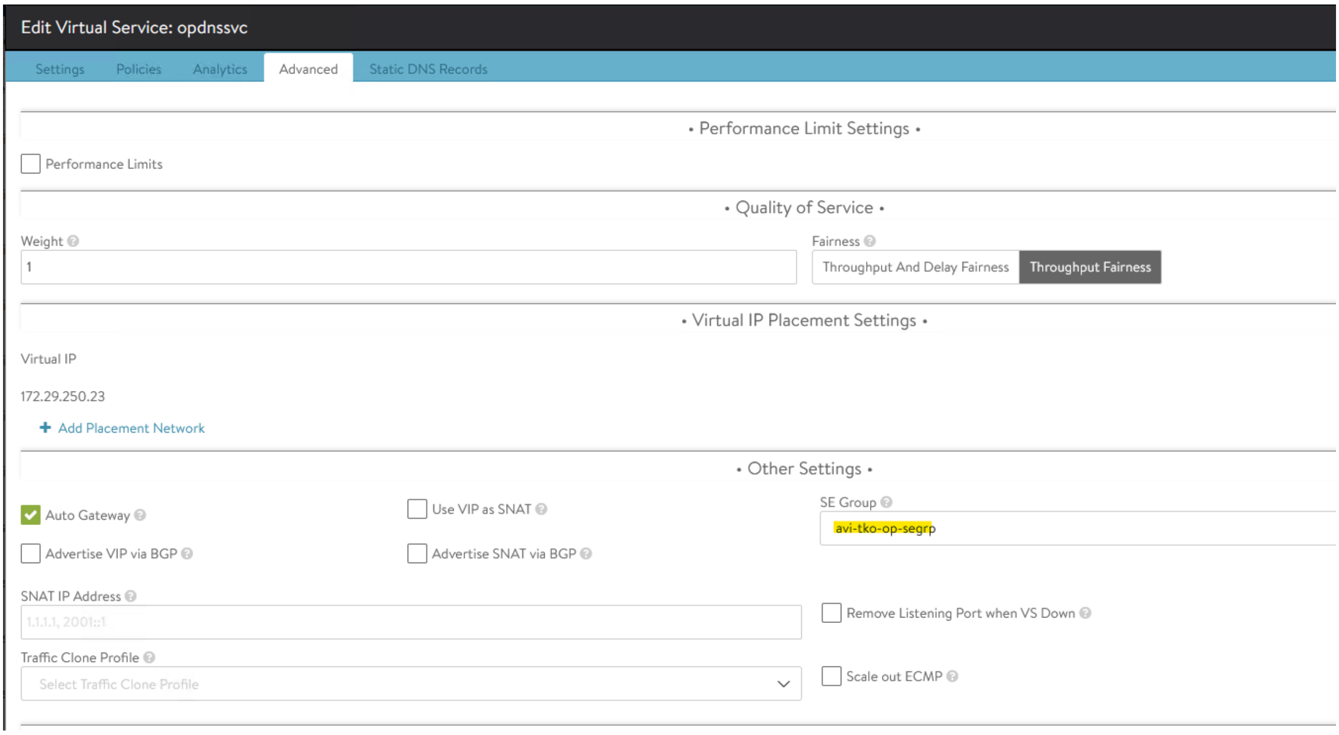
● Pool ○ Pool Group

Pool: dnsvc-pool

□ Ignore network reachability constraints for the server pool

5. Add the pool created earlier to the virtual service

Figure 112: Create DNS service.



Edit Virtual Service: opdnssvc

Settings | Policies | Analytics | Advanced | Static DNS Records

• Performance Limit Settings •

□ Performance Limits

• Quality of Service •

Weight: 1

Fairness: Throughput And Delay Fairness | [✓ Throughput Fairness]

• Virtual IP Placement Settings •

Virtual IP

172.29.250.23

+ Add Placement Network

• Other Settings •

☑ Auto Gateway

□ Use VIP as SNAT

SE Group: avi-tko-op-segrp

□ Advertise VIP via BGP

□ Advertise SNAT via BGP

SNAT IP Address: 1.1.1.1, 2001::1

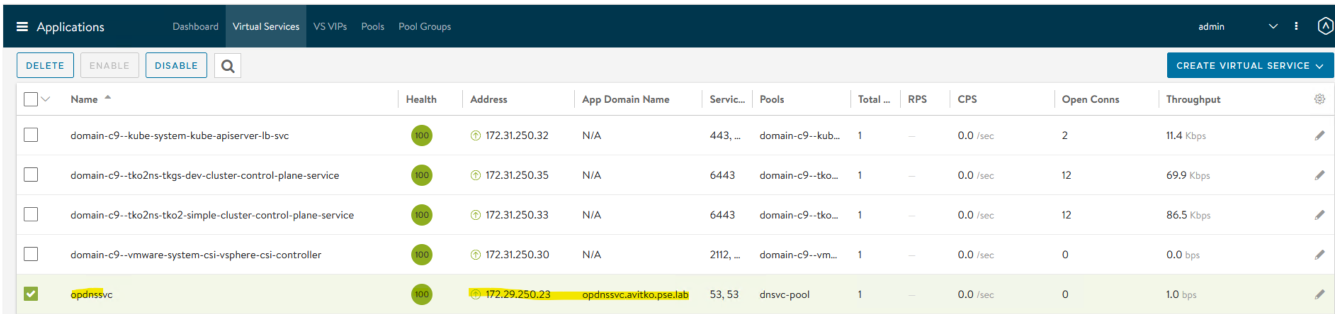
□ Remove Listening Port when VS Down

Traffic Clone Profile: Select Traffic Clone Profile

□ Scale out ECMP

6. Ensure that virtual service is up and running.

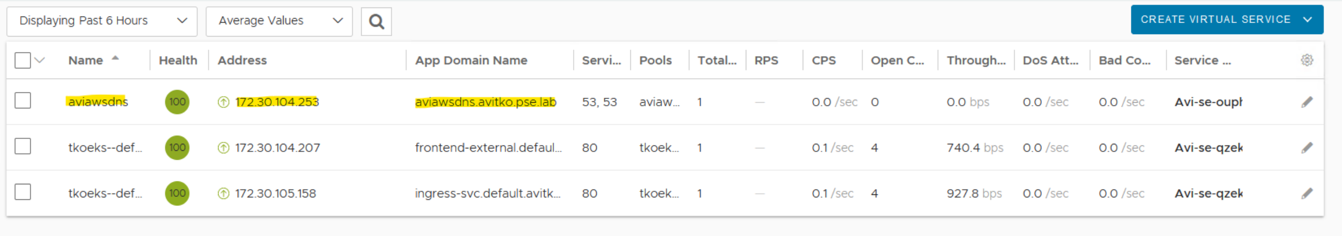
Figure 113: Verify DNS service.



	Name	Health	Address	App Domain Name	Servic...	Pools	Total ...	RPS	CPS	Open Conns	Throughput
□	domain-c9--kube-system-kube-apiserver-lb-svc	100	172.31.250.32	N/A	443, ...	domain-c9--kub...	1	–	0.0 /sec	2	11.4 Kbps
□	domain-c9--tko2ns-tkgs-dev-cluster-control-plane-service	100	172.31.250.35	N/A	6443	domain-c9--tko...	1	–	0.0 /sec	12	69.9 Kbps
□	domain-c9--tko2ns-tko2-simple-cluster-control-plane-service	100	172.31.250.33	N/A	6443	domain-c9--tko...	1	–	0.0 /sec	12	86.5 Kbps
□	domain-c9--vmware-system-csi-vsphere-csi-controller	100	172.31.250.30	N/A	2112, ...	domain-c9--vm...	1	–	0.0 /sec	0	0.0 bps
☑	opdnssvc	100	172.29.250.23	opdnssvc.avitko.pse.lab	53, 53	dnsvc-pool	1	–	0.0 /sec	0	1.0 bps

7. Follow the same steps on NSX Advanced Load Balancer on AWS and create a virtual service for DNS.

Figure 114: Create DNS service on AWS.



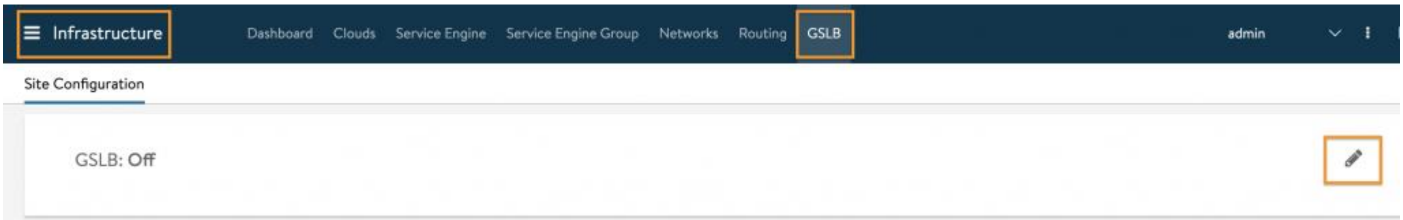
	Name	Health	Address	App Domain Name	Servi...	Pools	Total...	RPS	CPS	Open C...	Through...	DoS Att...	Bad Co...	Service ...
□	aviawsdns	100	172.30.104.253	aviawsdns.avitko.pse.lab	53, 53	aviaw...	1	–	0.0 /sec	0	0.0 bps	0.0 /sec	0.0 /sec	Avi-se-oupł
□	tkoeks--def...	100	172.30.104.207	frontend-external.defaul...	80	tkoek...	1	–	0.1 /sec	4	740.4 bps	0.0 /sec	0.0 /sec	Avi-se-qzek
□	tkoeks--def...	100	172.30.105.158	ingress-svc.default.avitk...	80	tkoek...	1	–	0.1 /sec	4	927.8 bps	0.0 /sec	0.0 /sec	Avi-se-qzek

8. Note the IP addresses of the services. These will be used when domain delegation is created.

Configure GSLB Sites

1. On the on-premises controller, which will be the leader, configure GSLB via Infrastructure > GSLB. Click the pencil icon.

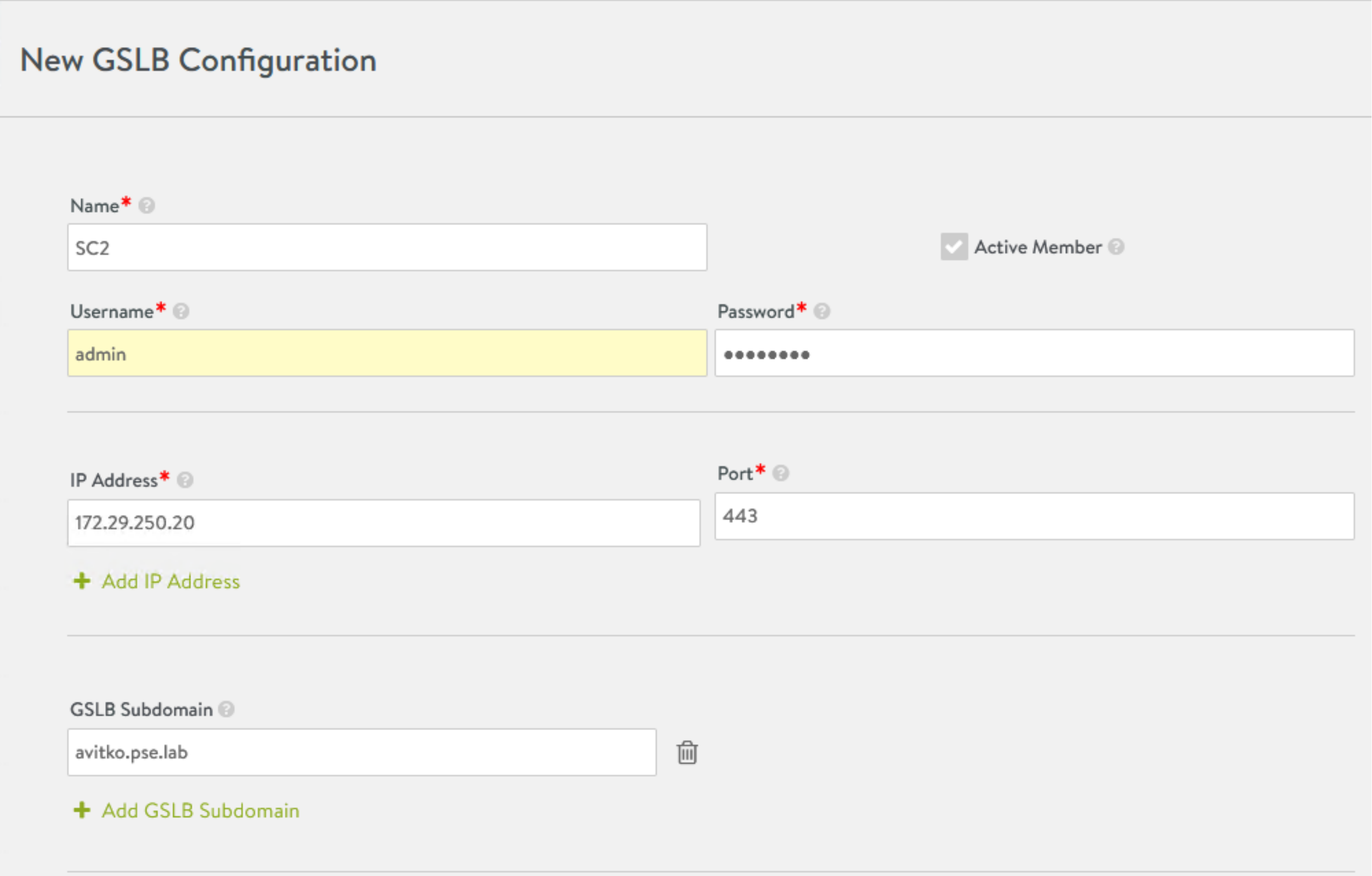
Figure 115: Enable GSLB



GSLB: Off

2. Give it a name. Enter the credentials for the controller and IP address. Enter the subdomain to be delegated to the load balancer.

Figure 116: Configure GSLB sites.



New GSLB Configuration

Name* SC2

☑ Active Member

Username* admin

Password* ••••••••

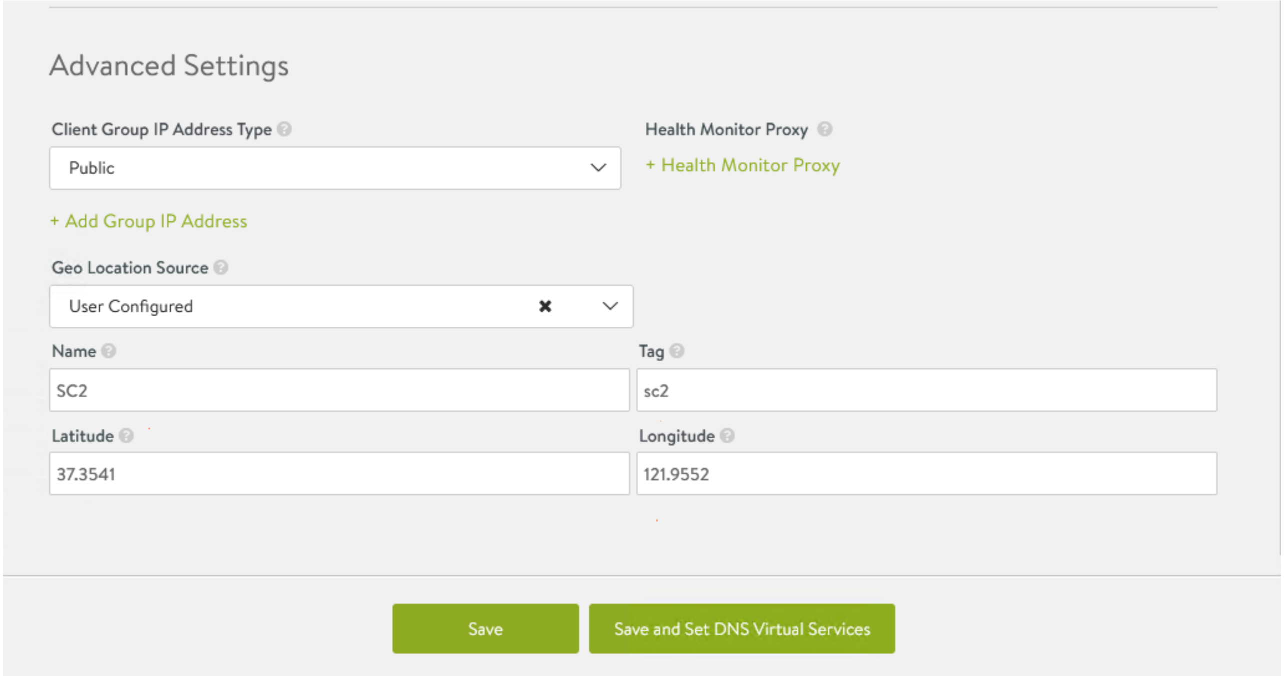
IP Address* 172.29.250.20

Port* 443

GSLB Subdomain avitko.pse.lab

3. Under advanced settings configure the geo location parameters if load balancing based on Geo location is required. Click “Save and Set DNS Virtual Services.”

Figure 117: Configure GSLB sites.



Advanced Settings

Client Group IP Address Type: Public

Health Monitor Proxy

Geo Location Source: User Configured

Name: SC2

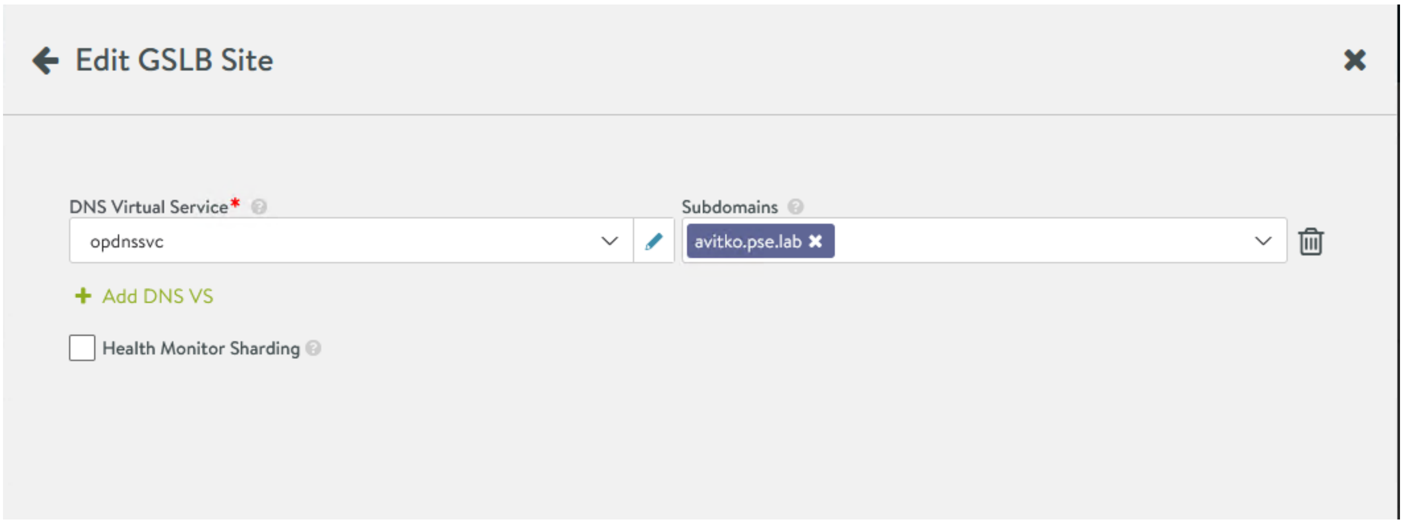
Tag: sc2

Latitude: 37.3541

Longitude: 121.9552

4. On the next screen, select the subdomain and virtual service created earlier for on-premises DNS. Click “Save.”

Figure 118: Configure GSLB sites.



Edit GSLB Site

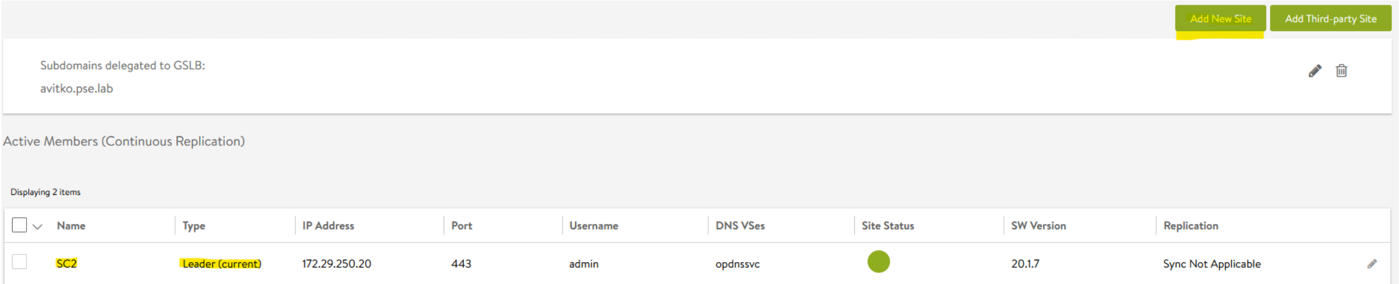
DNS Virtual Service*: opdnssvc

Subdomains: avitko.pse.lab

□ Health Monitor Sharding

5. GSLB for on-premises is created. While on the same screen click “Add new site.”

Figure 119: Configure GSLB sites.



Subdomains delegated to GSLB:
avitko.pse.lab

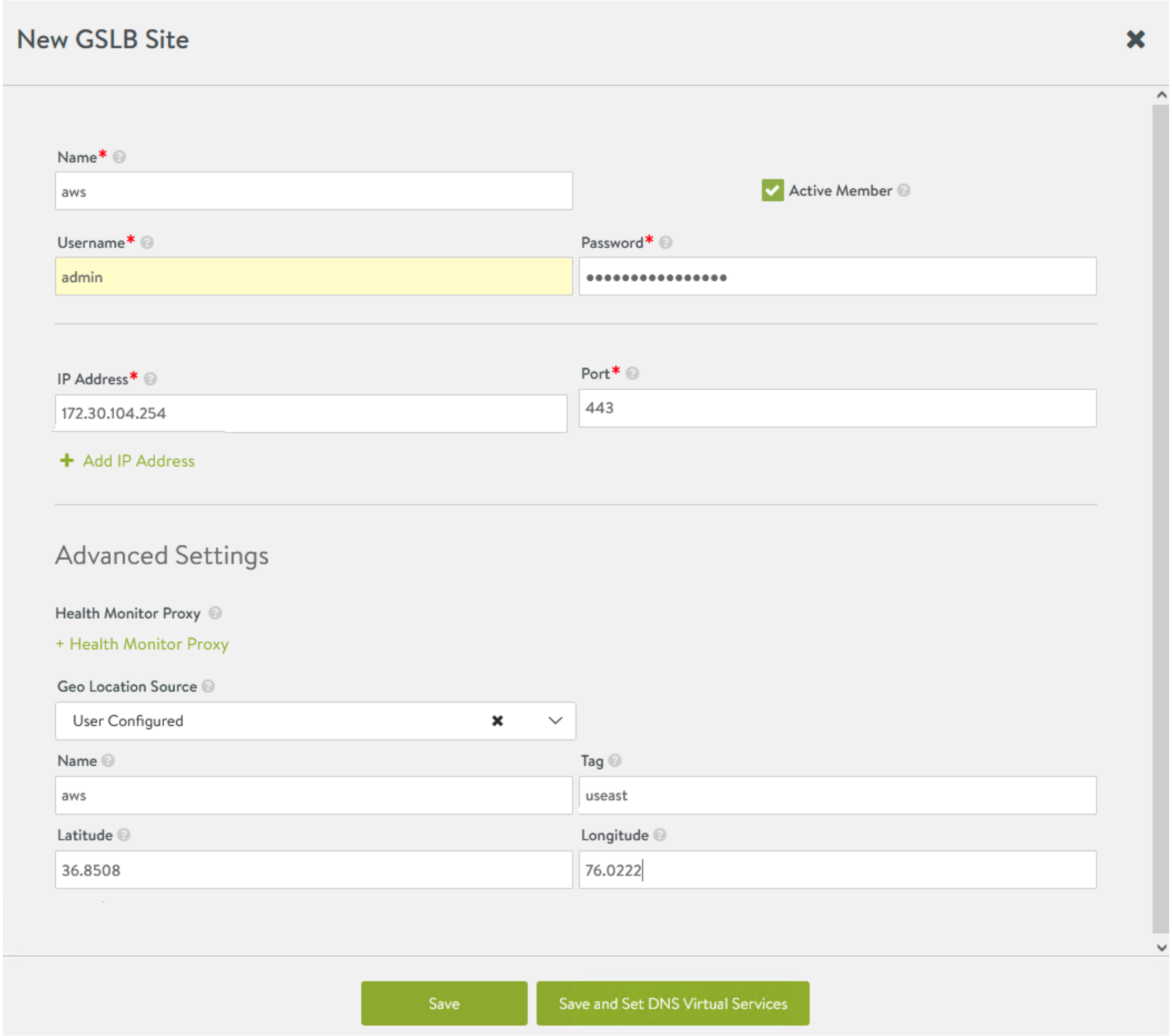
Active Members (Continuous Replication)

Displaying 2 items

Name	Type	IP Address	Port	Username	DNS VSes	Site Status	SW Version	Replication
SC2	Leader (current)	172.29.250.20	443	admin	opdnssvc	●	20.1.7	Sync Not Applicable

6. Enter the information for the NSX Advanced Load Balancer on AWS including geo location parameters if required. Ensure that “Active Member” is checked.

Figure 120: Configure GSLB sites.


New GSLB Site

Name* aws

☑ Active Member

Username* admin

Password* ••••••••••••••••

IP Address* 172.30.104.254

Port* 443

Advanced Settings

Health Monitor Proxy

Geo Location Source: User Configured

Name: aws

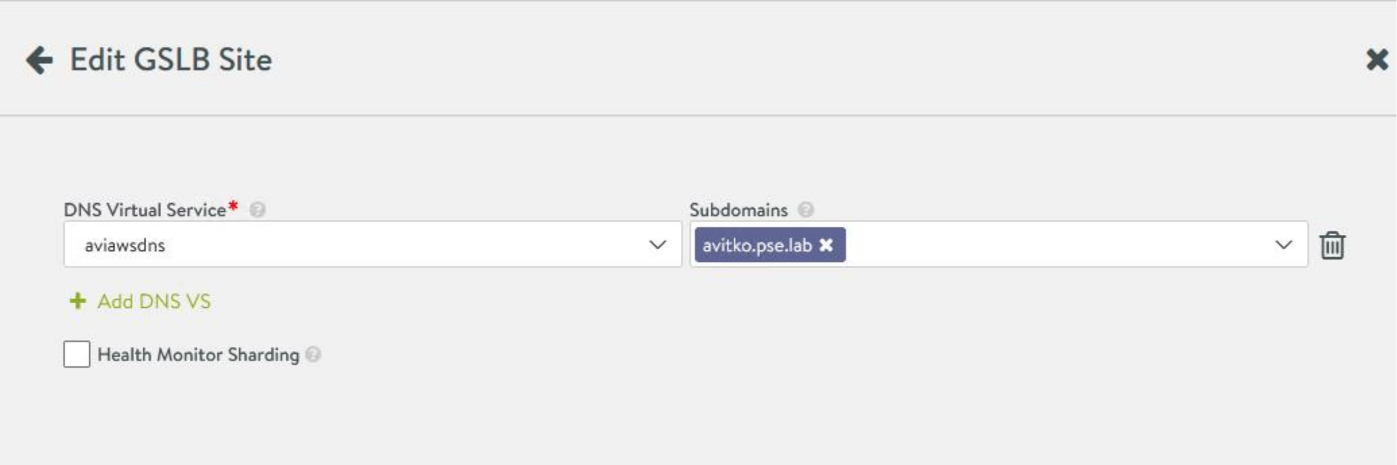
Tag: useast

Latitude: 36.8508

Longitude: 76.0222

7. Click “Save and Set DNS Virtual Services” and select the subdomain and virtual created on controller on AWS. Click Save.

Figure 121: Configure GSLB sites.


Edit GSLB Site

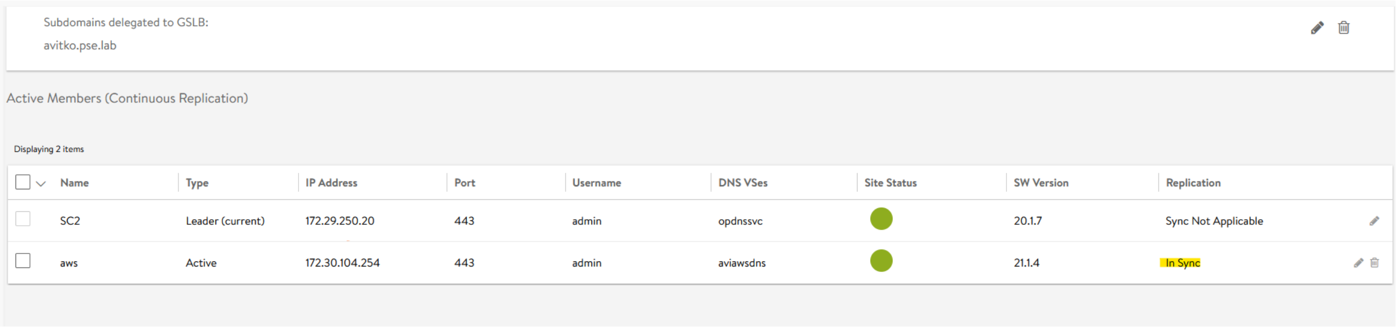
DNS Virtual Service*: aviawsdns

Subdomains: avitko.pse.lab

□ Health Monitor Sharding

8. GSLB for both on-premises and AWS are created and should be running. AWS service will show “In Synch” when synch is successful.

Figure 122: Configure GSLB sites.



Subdomains delegated to GSLB:
avitko.pse.lab

Active Members (Continuous Replication)

Displaying 2 items

☐	Name	Type	IP Address	Port	Username	DNS VSes	Site Status	SW Version	Replication
☐	SC2	Leader (current)	172.29.250.20	443	admin	opdnssvc	●	20.1.7	Sync Not Applicable
☐	aws	Active	172.30.104.254	443	admin	aviawsdns	●	21.1.4	In Sync

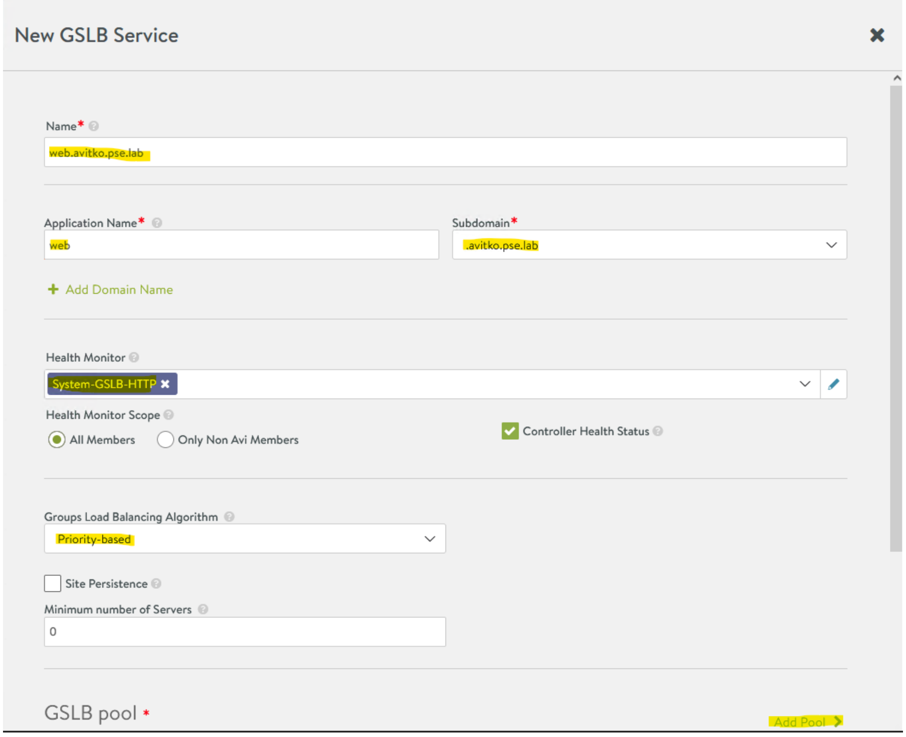
Configuring GSLB Services

Once the GSLB sites are configured and synchronized, GSLB service for desired applications or services can be created. Create a GSLB service to test and verify functionality as follows.

Note: When a Kubernetes application is installed on multiple sites, and the app selector label matches the label assigned during AMKO installation, AMKO will automatically create the GSLB service for the application. Following process creates the creation of GSLB service to illustrate the functionality.

1. On the on-premises controller navigate to Applications > GSLB Service. Give the service a name and select the sub-domain. Select a health monitoring option and a load balancing algorithm. Load balancing can be based on service priority or geo location. Click “Add Pool.”

Figure 123: Configure GSLB services.



New GSLB Service

Name*: web.avitko.pse.lab

Application Name*: web

Subdomain*: .avitko.pse.lab

+ Add Domain Name

Health Monitor: System-GSLB-HTTP ✖

Health Monitor Scope: ● All Members ○ Only Non Avi Members

☑ Controller Health Status

Groups Load Balancing Algorithm: Priority-based

☐ Site Persistence

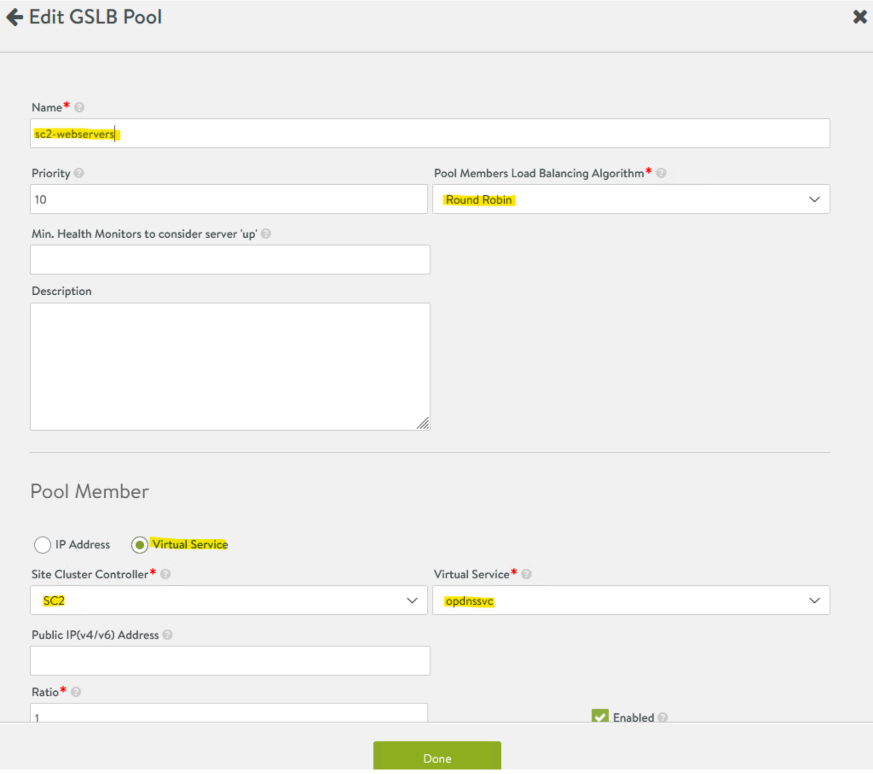
Minimum number of Servers: 0

GSLB pool *

Add Pool >

2. Give the pool a name and select load balancing algorithm. Under “Pool Member” site and virtual service created earlier. Click done.

Figure 124: Configure GSLB services.



Edit GSLB Pool

Name* sc2-webservers

Priority 10

Pool Members Load Balancing Algorithm* Round Robin

Min. Health Monitors to consider server 'up'

Description

Pool Member

○ IP Address ● Virtual Service

Site Cluster Controller* SC2

Virtual Service* opdnssvc

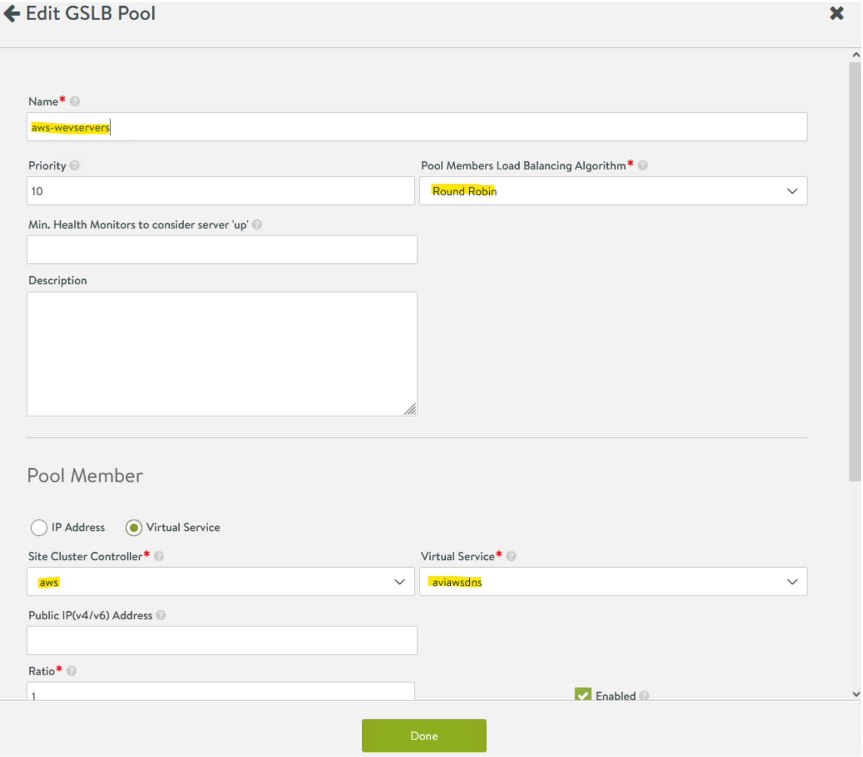
Public IP(v4/v6) Address

Ratio* 1

☑ Enabled

3. Create another pool for AWS in similar fashion.

Figure 125: Configure GSLB services.



Edit GSLB Pool

Name* aws-wevservers

Priority 10

Pool Members Load Balancing Algorithm* Round Robin

Min. Health Monitors to consider server 'up'

Description

Pool Member

○ IP Address ● Virtual Service

Site Cluster Controller* aws

Virtual Service* aviawsdns

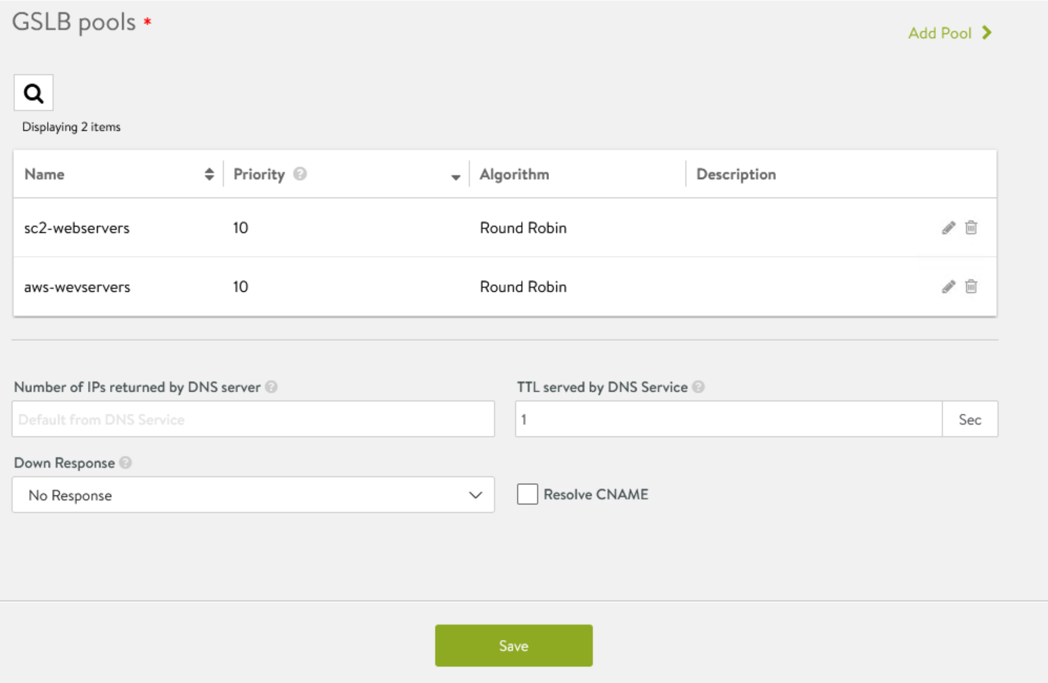
Public IP(v4/v6) Address

Ratio* 1

☑ Enabled

4. You should now have two pools showing in the virtual GSLB service. Click save.

Figure 126: Configure GSLB services.



GSLB pools *

Displaying 2 items

Name	Priority	Algorithm	Description
sc2-webservers	10	Round Robin	
aws-wevservers	10	Round Robin	

Number of IPs returned by DNS server: Default from DNS Service

TTL served by DNS Service: 1 Sec

Down Response: No Response

□ Resolve CNAME

5. The GSLB service is created. This service is on the “avitko.pse.lab” domain which will be delegated to NSX Advanced Load Balancer. Hence next step is to create a delegation for this sub-domain.

DNS Domain Delegation

The Avi DNS virtual service is a generic DNS infrastructure that can implement the following functionality.

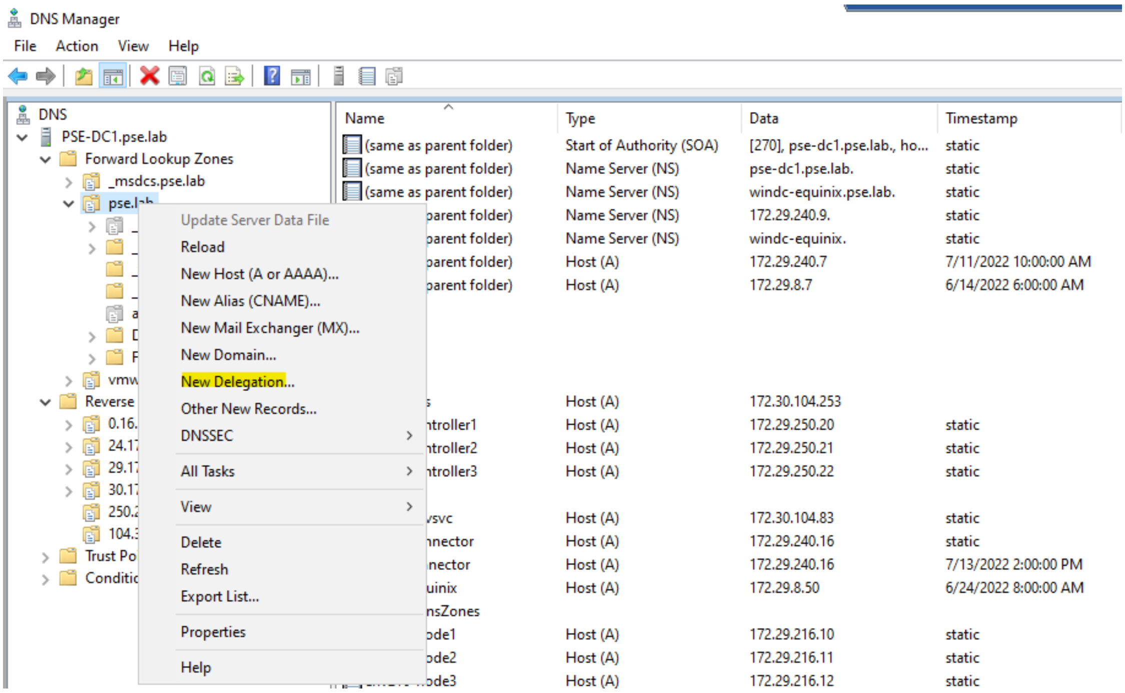
- DNS Load Balancing
- Hosting Manual or Static DNS Entries
- Virtual Service IP Address DNS Hosting
- Hosting GSLB Service DNS Entries

NSX Advanced Load Balancer DNS service can be deployed in a couple of ways. It can be deployed as an authoritative name server for a sub-domain delegated to it or as a primary DNS server for the domain. In the latter case, any requests that do not match DNS records in NSX Advanced Load Balancer are “proxied” to the corporate DNS server. For this reference architecture sub-domain delegation was used as described below. For more information on NSX Advanced Load Balancer DNS architecture please visit Avi DNS Architecture and Features page.

To create DNS domain delegation, the IP addresses of the DNS services created earlier for on-premises and AWS sites will be needed. This reference architecture uses Microsoft Active Directory Domain DNS.

1. In DNS manager, create an A Record of the two DNS services that were created.
2. On the Domain controller open the DNS manager and create a “New Delegation.”

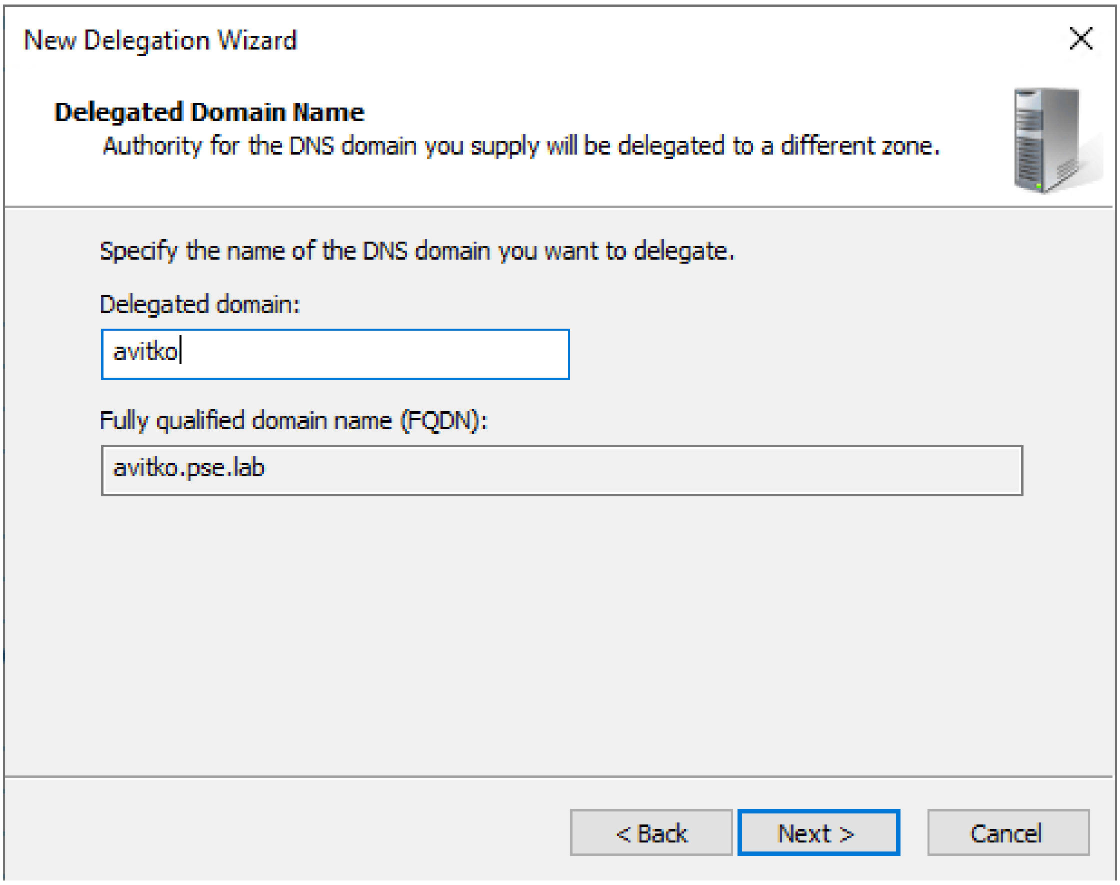
Figure 127: DNS domain delegation



Name	Type	Data	Timestamp
(same as parent folder)	Start of Authority (SOA)	[270], pse-dc1.pse.lab., ho...	static
(same as parent folder)	Name Server (NS)	pse-dc1.pse.lab.	static
(same as parent folder)	Name Server (NS)	windc-equinix.pse.lab.	static
parent folder)	Name Server (NS)	172.29.240.9.	static
parent folder)	Name Server (NS)	windc-equinix.	static
parent folder)	Host (A)	172.29.240.7	7/11/2022 10:00:00 AM
parent folder)	Host (A)	172.29.8.7	6/14/2022 6:00:00 AM
s	Host (A)	172.30.104.253	
ntroller1	Host (A)	172.29.250.20	static
ntroller2	Host (A)	172.29.250.21	static
ntroller3	Host (A)	172.29.250.22	static
vsvc	Host (A)	172.30.104.83	static
nnector	Host (A)	172.29.240.16	static
nector	Host (A)	172.29.240.16	7/13/2022 2:00:00 PM
uinix	Host (A)	172.29.8.50	6/24/2022 8:00:00 AM
nsZones			
ode1	Host (A)	172.29.216.10	static
ode2	Host (A)	172.29.216.11	static
ode3	Host (A)	172.29.216.12	static

3. Enter the name of the sub-domain. In this case its “avitko.” Click Next

Figure 128: DNS domain delegation



New Delegation Wizard

Delegated Domain Name

Authority for the DNS domain you supply will be delegated to a different zone.

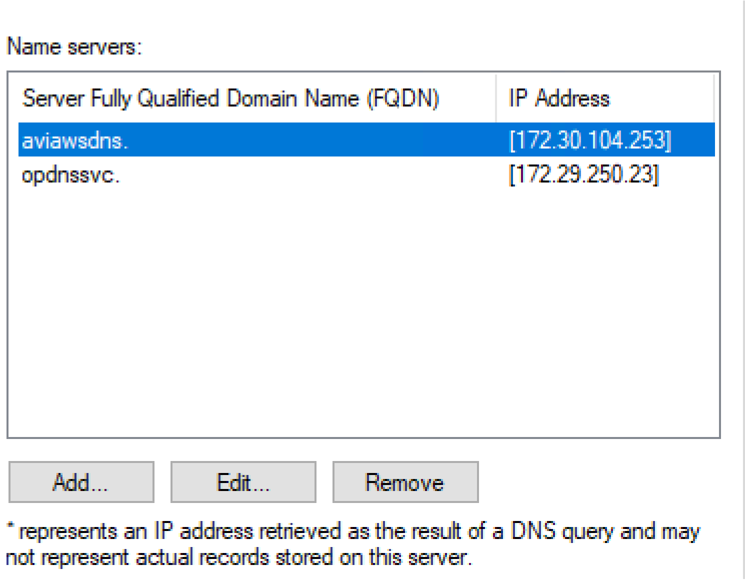
Specify the name of the DNS domain you want to delegate.

Delegated domain: avitko

Fully qualified domain name (FQDN): avitko.pse.lab

4. Add the two DNS services to the delegation and click Next and finish the process.

Figure 129: DNS domain delegation



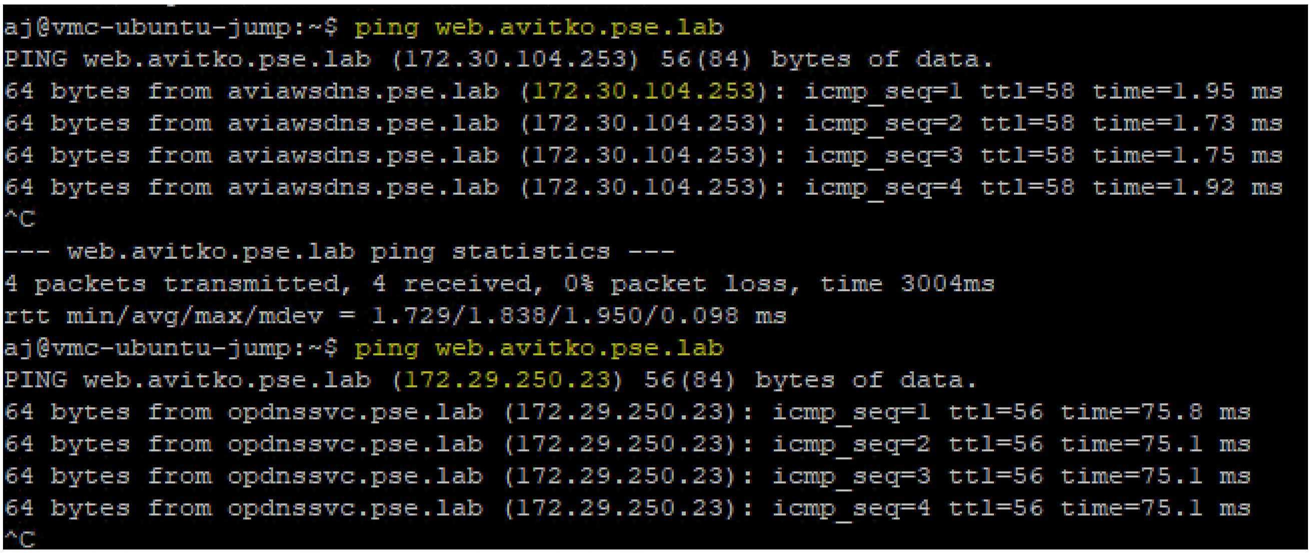
Name servers:

Server Fully Qualified Domain Name (FQDN)	IP Address
aviawsdns.	[172.30.104.253]
opdnssvc.	[172.29.250.23]

* represents an IP address retrieved as the result of a DNS query and may not represent actual records stored on this server.

5. The delegation is complete.
6. Test the delegation by pinging the service URL. You should get a response from one of the two DNS services. This indicates that domain delegation is functioning properly. Since we selected round-robin algorithm, subsequent ping should be answered by the service on the other site. This indicates that load balancing configuration is functional. Geo location option can be set when desired as primary response type with round-robin as second option. In this case, users will be directed to the nearest application instance based on their location.

Figure 130: Test DNS domain delegation



```
aj@vmc-ubuntu-jump:~$ ping web.avitko.pse.lab
PING web.avitko.pse.lab (172.30.104.253) 56(84) bytes of data.
64 bytes from aviawsdns.pse.lab (172.30.104.253): icmp_seq=1 ttl=58 time=1.95 ms
64 bytes from aviawsdns.pse.lab (172.30.104.253): icmp_seq=2 ttl=58 time=1.73 ms
64 bytes from aviawsdns.pse.lab (172.30.104.253): icmp_seq=3 ttl=58 time=1.75 ms
64 bytes from aviawsdns.pse.lab (172.30.104.253): icmp_seq=4 ttl=58 time=1.92 ms
^C
--- web.avitko.pse.lab ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.729/1.838/1.950/0.098 ms
aj@vmc-ubuntu-jump:~$ ping web.avitko.pse.lab
PING web.avitko.pse.lab (172.29.250.23) 56(84) bytes of data.
64 bytes from opdnssvc.pse.lab (172.29.250.23): icmp_seq=1 ttl=56 time=75.8 ms
64 bytes from opdnssvc.pse.lab (172.29.250.23): icmp_seq=2 ttl=56 time=75.1 ms
64 bytes from opdnssvc.pse.lab (172.29.250.23): icmp_seq=3 ttl=56 time=75.1 ms
64 bytes from opdnssvc.pse.lab (172.29.250.23): icmp_seq=4 ttl=56 time=75.1 ms
^C
```

Lifecyle Management via Tanzu Mission Control

As companies grow their cloud native environments to multiple cloud providers, platform consistency and manageability becomes a challenge. Each cloud provider has its own management portal and lifecycle management of such environment can become a nightmare. Enterprises need a solution to help platform operators efficiently expand control and provide Kubernetes environments with guardrails so DevOps teams can have consistency and developers can operate autonomously, in a self-service fashion. For user authentication, an identity source such as Microsoft Active Directory or another a 3rd party identity source needs to be federated with Tanzu Mission Control. Please see "Self-Service Federation Setup" in Tanzu Mission Control Documentation.

VMware Tanzu Mission Control is a centralized management hub with cluster lifecycle management and a unified policy engine that simplifies multi-cloud and multi-cluster Kubernetes management across teams in the enterprise.

Administrator can perform several tasks to manage their on-premises or multi-cloud environments. Some of the tasks that an administrator needs to perform to administer their environment is listed and explained below.

1. Create a cluster group.
2. Add management cluster to Tanzu Mission Control
3. Create Kubernetes workload clusters.
4. Attach existing Kubernetes clusters.
5. Install Tanzu toolkit packages and applications via helm charts.
6. Configure policies and policies templates.
7. DevOps access and automation via Tanzu Mission Control CLI
8. Enable continuous delivery (CD) via Git repository integration.
9. Create, attach, or delete Amazon EKS clusters.
10. Run conformance and security inspections on clusters.
11. Enable data protection for clusters.

Note: It is not the intent of this document to cover all aspects and features of Tanzu Mission Control. For more details please see Tanzu Mission Control documentation.

Create a cluster group

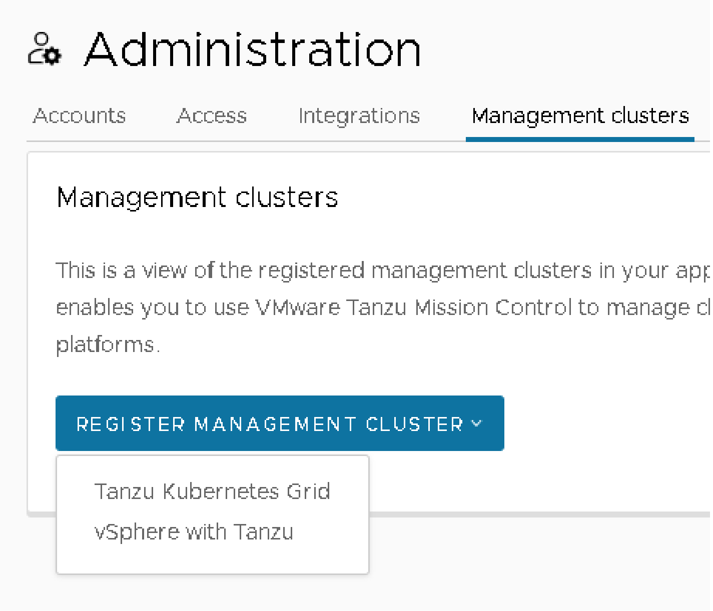
Creating a cluster group for different deployments or site is an optional step. The advantage is that it organizes different cluster types and policies can be applied to all cluster at the group level. Create cluster groups from the left menu pane in Tanzu Mission Control portal.

Add management cluster to Tanzu Mission Control

For Tanzu Mission Control to manage the Tanzu Kubernetes Grid environment, the management cluster need to be registered to it. The following steps depict the management cluster registration process.

1. In the Tanzu Mission Control portal, navigate to Administration > Management clusters and click on Register Management Cluster and select type of management cluster you are registering.

Figure 131: Register management cluster



Administration

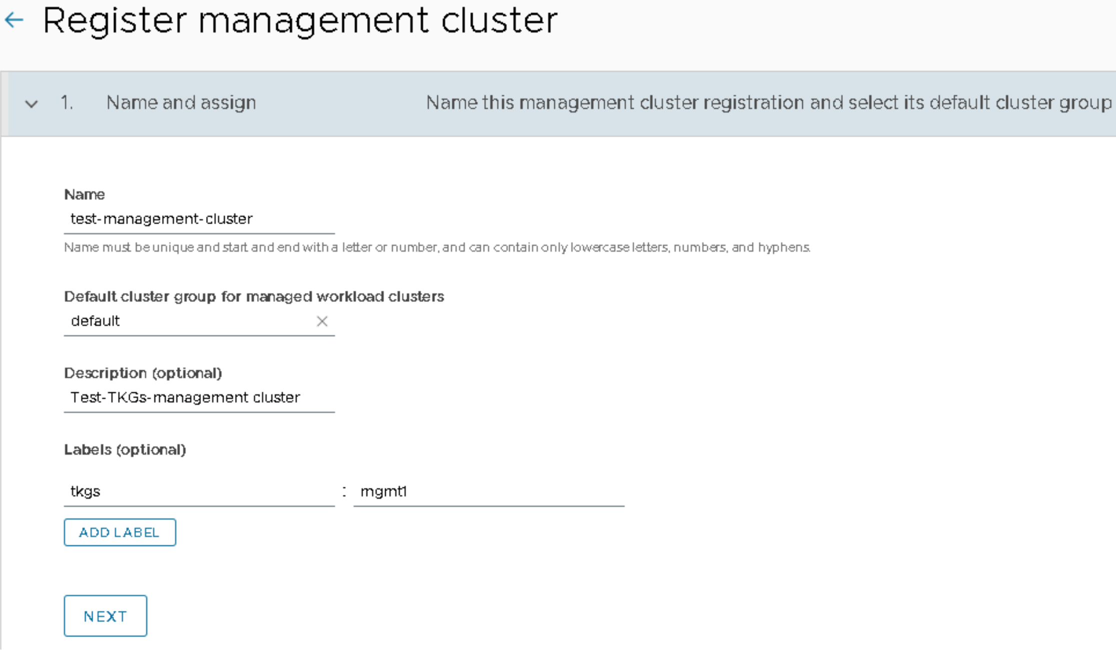
Accounts Access Integrations Management clusters

Management clusters

This is a view of the registered management clusters in your app enables you to use VMware Tanzu Mission Control to manage cl platforms.

2. Enter name, cluster group, description, and label information if desired. Labels help organize various Tanzu Mission Control objects and that can be sorted and displayed easily.

Figure 132: Register management cluster



Register management cluster

1. Name and assign — Name this management cluster registration and select its default cluster group

Name: test-management-cluster

Name must be unique and start and end with a letter or number, and can contain only lowercase letters, numbers, and hyphens.

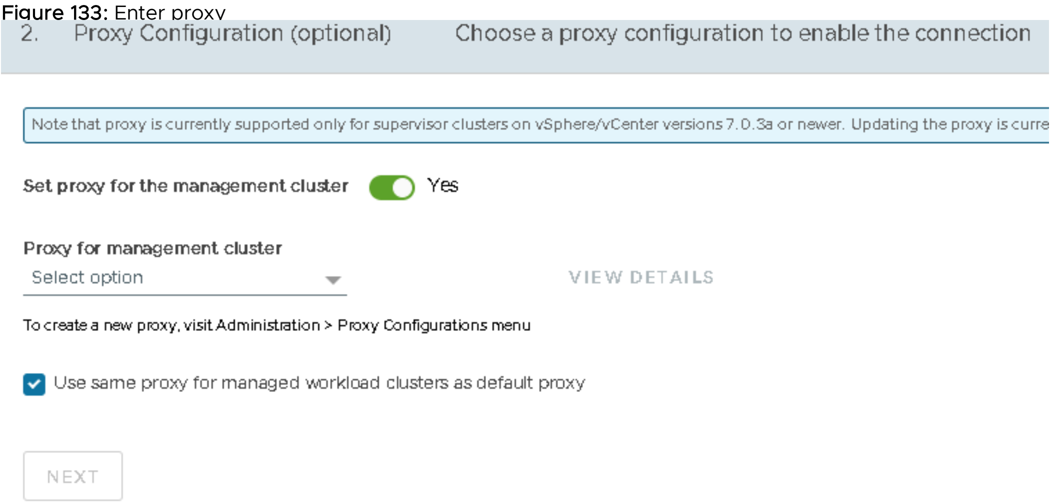
Default cluster group for managed workload clusters: default

Description (optional): Test-TKGs-management cluster

Labels (optional): tkgs : mgmt1

3. Enter proxy information if your management cluster is behind a proxy.

Figure 133: Enter proxy



2. Proxy Configuration (optional) Choose a proxy configuration to enable the connection

Note that proxy is currently supported only for supervisor clusters on vSphere/vCenter versions 7.0.3a or newer. Updating the proxy is curre

Set proxy for the management cluster: Yes

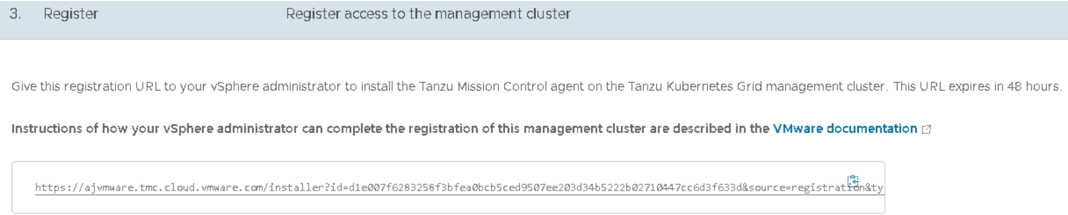
Proxy for management cluster: Select option

To create a new proxy, visit Administration > Proxy Configurations menu

☑ Use same proxy for managed workload clusters as default proxy

4. Copy and provide the registration URL that has the registration key to the vSphere administrator. The vSphere administrator will perform the next step in registering the management cluster to Tanzu Mission Control.

Figure 134: Copy registration URL



3. Register Register access to the management cluster

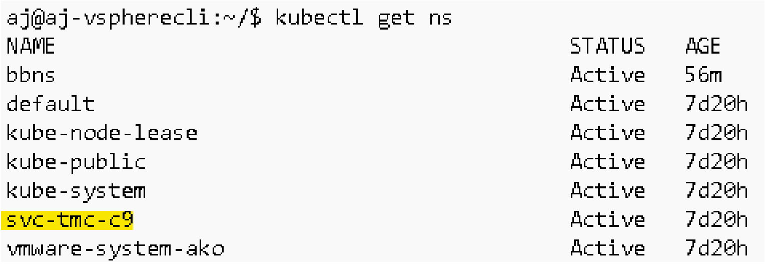
Give this registration URL to your vSphere administrator to install the Tanzu Mission Control agent on the Tanzu Kubernetes Grid management cluster. This URL expires in 48 hours.

Instructions of how your vSphere administrator can complete the registration of this management cluster are described in the VMware documentation

https://ajvmware.tmc.cloud.vmware.com/installer?id=d1e007f6283258f3bfea0bcb5ced9507ee203d34b5222b02710447cc6d3f633d&source=registration&ty

5. As a vSphere administrator, login to the Supervisor cluster and list namespaces. Take note of the Tanzu Mission Control service namespace.

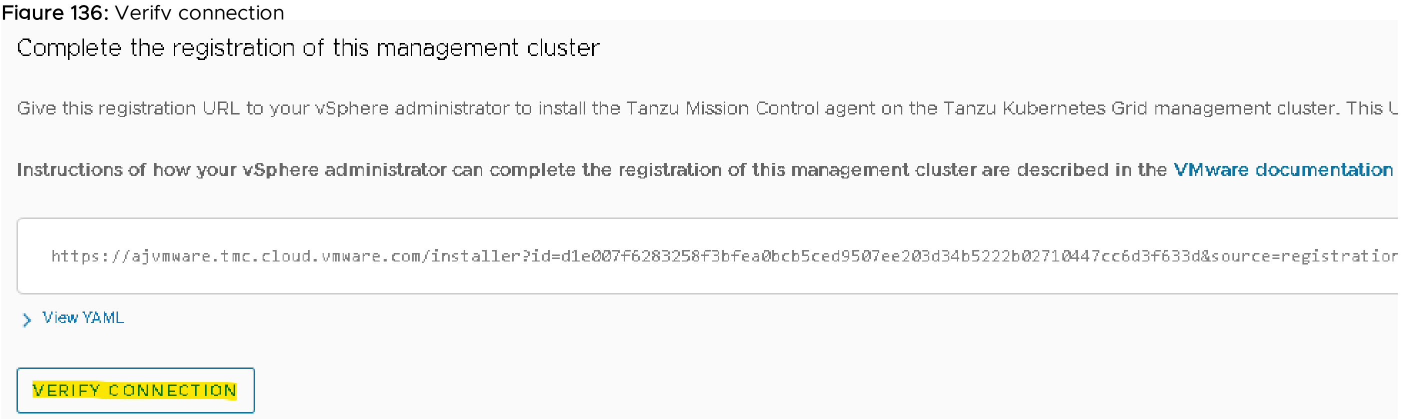
Figure 135: Tanzu Mission Control namespace



```
aj@aj-vspherecli:~/$ kubectl get ns
NAME                                         STATUS   AGE
bbns                                         Active   56m
default                                      Active   7d20h
kube-node-lease                              Active   7d20h
kube-public                                  Active   7d20h
kube-system                                  Active   7d20h
svc-tmc-c9                                   Active   7d20h
vmware-system-ako                            Active   7d20h
```

6. Create and apply .yaml file using the registration URL and svc-tmc-xx namespace as shown below.
   Sample yaml:
   apiVersion: installers.tmc.cloud.vmware.com/v1alpha1
   kind: AgentInstall
   metadata:
     name: tmc-agent-installer-config
     namespace: svc-tmc-c9
   spec:
     operation: INSTALL
     registrationLink: https://org.tmc.cloud.vmware.com/installer?id= 17e139c2ba3551axxxxxxxxx
7. Apply the yaml via kubectl create -f <filename.yaml> to complete the registration process.
8. In Tanzu Mission Control console, verify that connection to the Supervisor cluster is successful and cluster is added and functional.

Figure 136: Verify connection



Complete the registration of this management cluster

Give this registration URL to your vSphere administrator to install the Tanzu Mission Control agent on the Tanzu Kubernetes Grid management cluster. This U

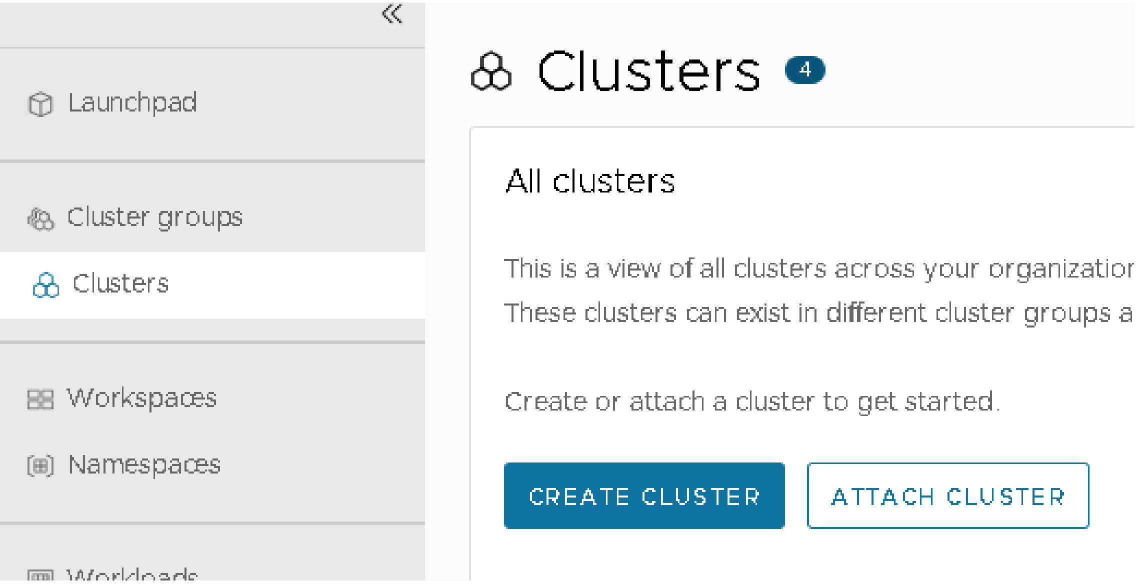
Instructions of how your vSphere administrator can complete the registration of this management cluster are described in the VMware documentation

```
https://ajvmware.tmc.cloud.vmware.com/installer?id=d1e007f6283258f3bfea0bcb5ced9507ee203d34b5222b02710447cc6d3f633d&source=registratior
```

Create Kubernetes workload clusters.

1. Navigate to Clusters and click create cluster.

Figure 137: Create cluster



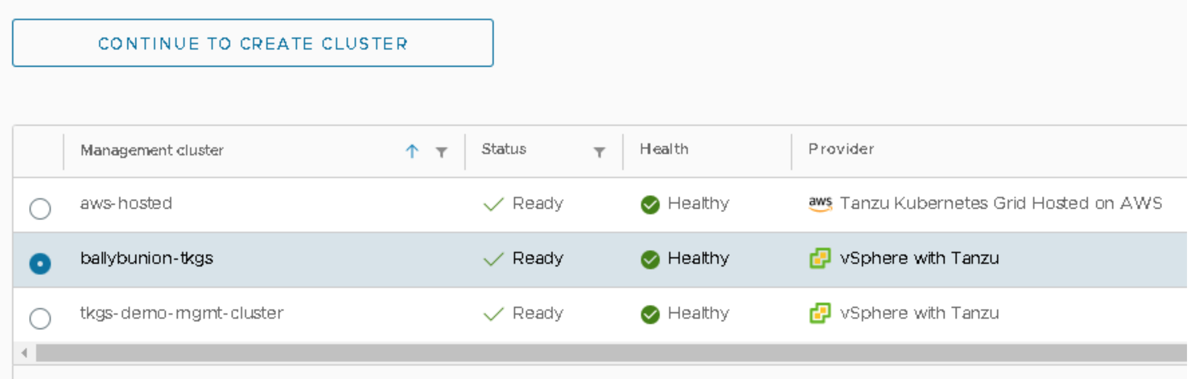
All clusters

This is a view of all clusters across your organizatior
These clusters can exist in different cluster groups a

Create or attach a cluster to get started.

2. Select the management cluster and click continue to create cluster.

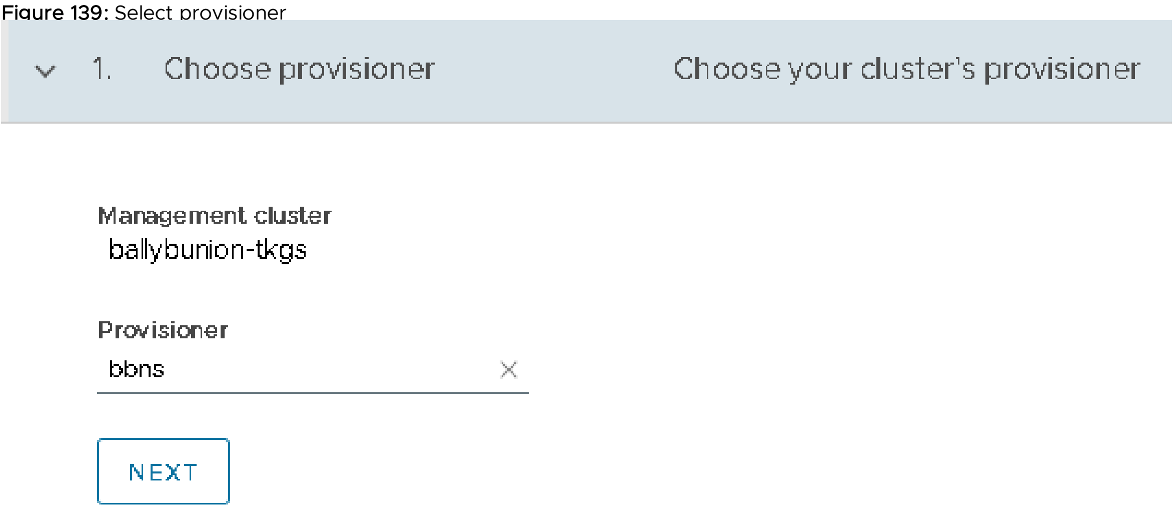
Figure 138: Select management cluster



	Management cluster	Status	Health	Provider
○	aws-hosted	✓ Ready	✓ Healthy	aws Tanzu Kubernetes Grid Hosted on AWS
●	ballybunion-tkgs	✓ Ready	✓ Healthy	vSphere with Tanzu
○	tkgs-demo-mgmt-cluster	✓ Ready	✓ Healthy	vSphere with Tanzu

3. Select provisioner which in this case is the namespace you created in workload management.

Figure 139: Select provisioner



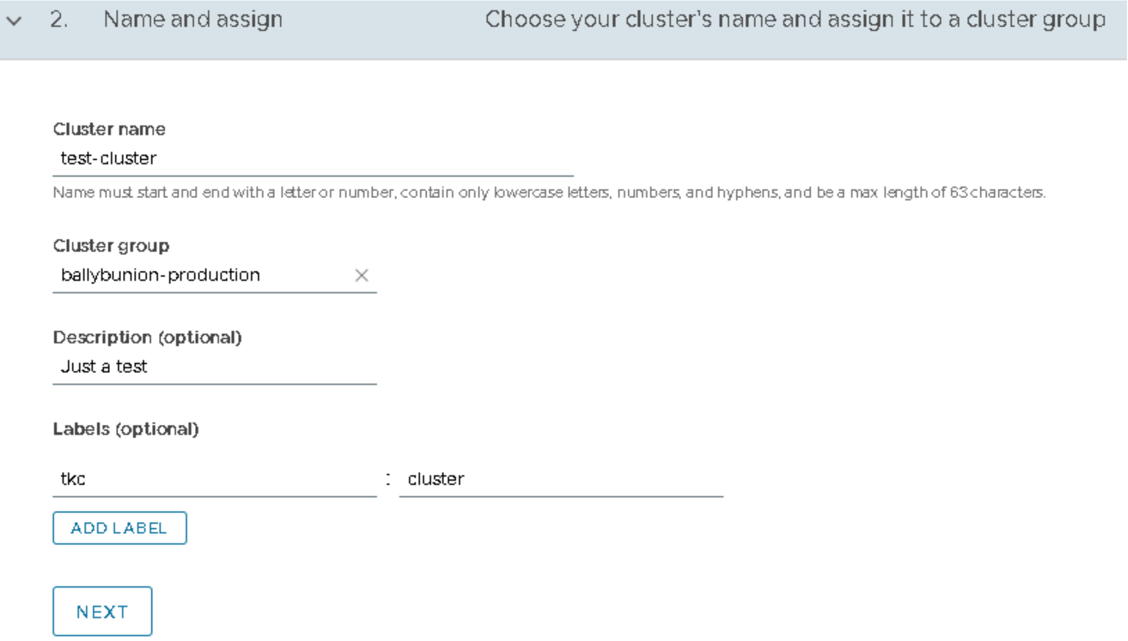
1. Choose provisioner — Choose your cluster's provisioner

Management cluster
ballybunion-tkgs

Provisioner
bbns

4. On the next screen give cluster a name and select a group.

Figure 140: Cluster name and group



2. Name and assign — Choose your cluster's name and assign it to a cluster group

Cluster name
test-cluster

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

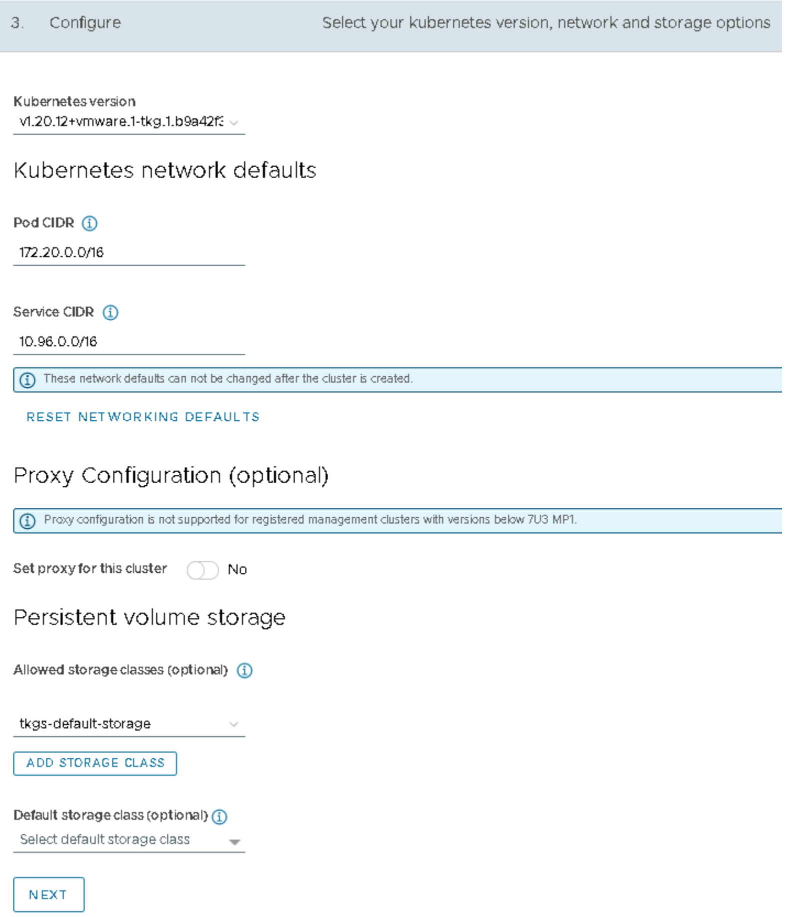
Cluster group
ballybunion-production

Description (optional)
Just a test

Labels (optional)
tkc : cluster

5. Select a Kubernetes version and assign network CIDR and storage class.

Figure 141: Configure parameters.



3. Configure — Select your kubernetes version, network and storage options

Kubernetes version: v1.20.12+vmware.1-tkg.1.b9a42f

Kubernetes network defaults

Pod CIDR: 172.20.0.0/16

Service CIDR: 10.96.0.0/16

These network defaults can not be changed after the cluster is created.

Proxy Configuration (optional)

Proxy configuration is not supported for registered management clusters with versions below 7U3 MP1.

Set proxy for this cluster: No

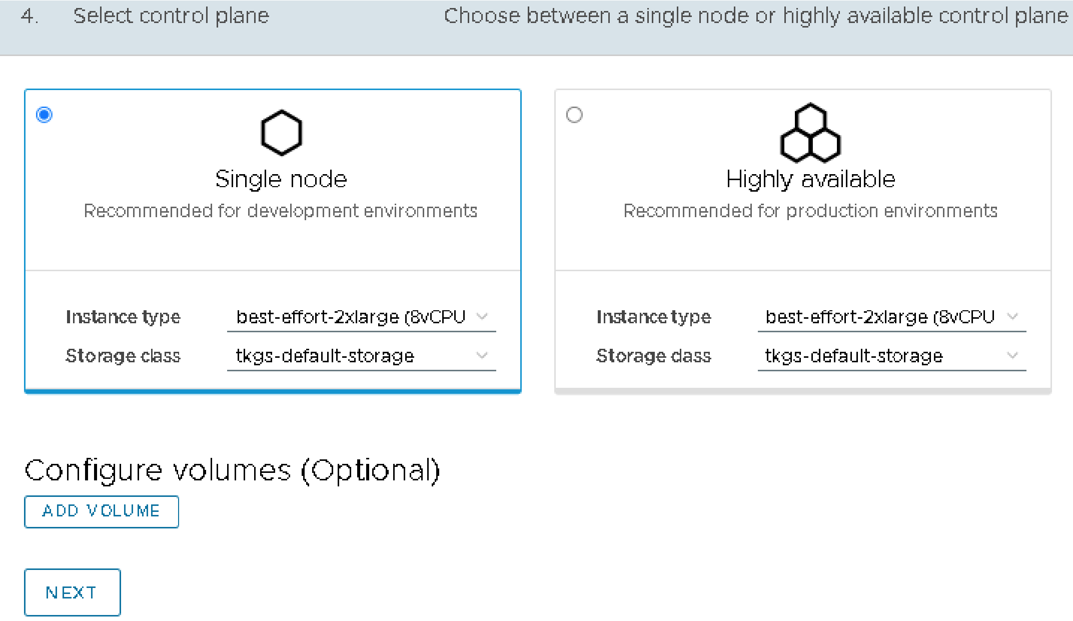
Persistent volume storage

Allowed storage classes (optional): tkgs-default-storage

Default storage class (optional): Select default storage class

6. On the next screen select a deployment model for your control plane nodes, select a VM class and storage policy. You can also create a volume at this point.

Figure 142: Configure control plane specifications.



4. Select control plane — Choose between a single node or highly available control plane

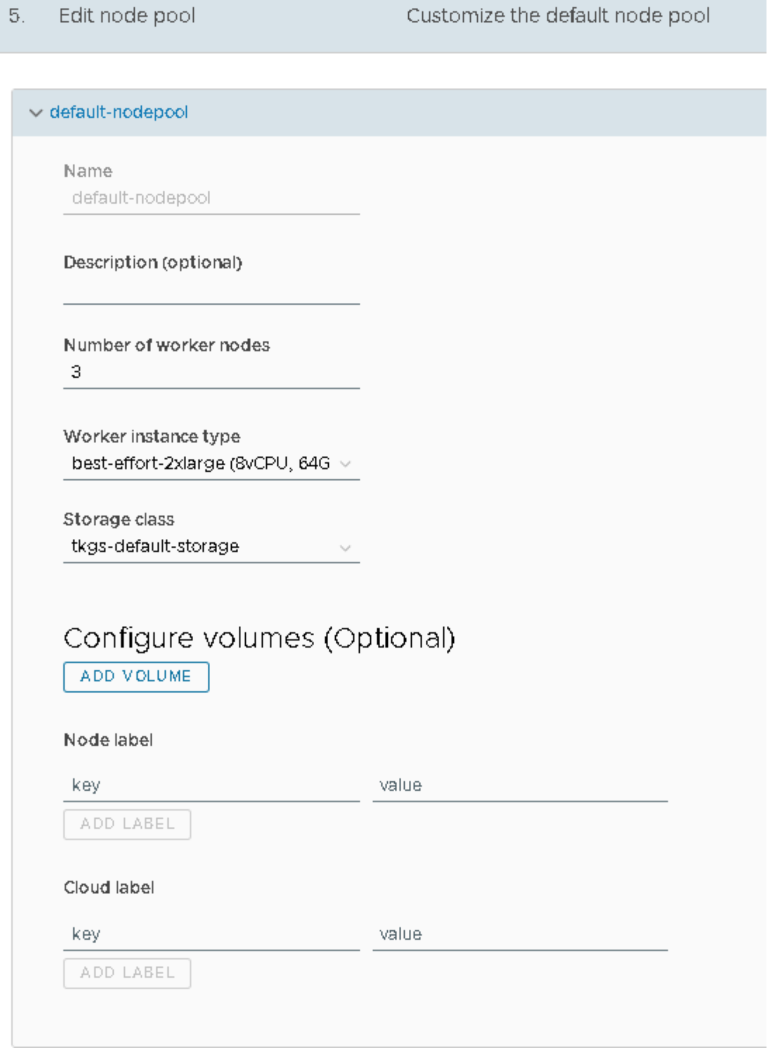
● Single node — Recommended for development environments
Instance type: best-effort-2xlarge (8vCPU
Storage class: tkgs-default-storage

○ Highly available — Recommended for production environments
Instance type: best-effort-2xlarge (8vCPU
Storage class: tkgs-default-storage

Configure volumes (Optional)

7. Modify the default pool configuration which has one node, to the desired number of worker nodes. Set the VM class and storage policy. Click Create Cluster to start the cluster creation process.

Figure 143: Modify default node pool



5. Edit node pool — Customize the default node pool

default-nodepool

Name: default-nodepool

Description (optional):

Number of worker nodes: 3

Worker instance type: best-effort-2xlarge (8vCPU, 64G

Storage class: tkgs-default-storage

Configure volumes (Optional)

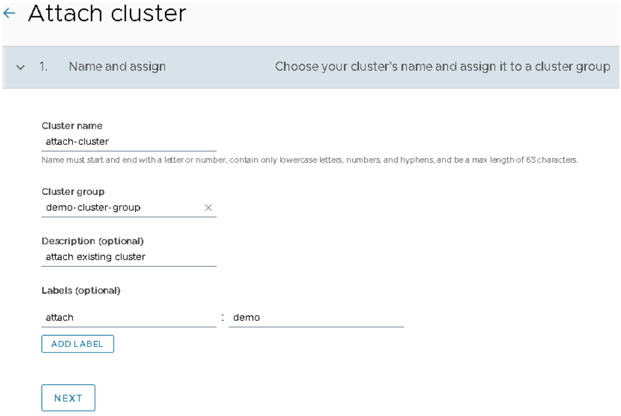
Node label: key : value

Cloud label: key : value

Attach existing Kubernetes clusters.

1. In Tanzu Mission Control navigate to Clusters > Attach Cluster and enter the desired information.

Figure 144: Attached Cluster



Attach cluster

1. Name and assign — Choose your cluster's name and assign it to a cluster group

Cluster name: attach-cluster

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

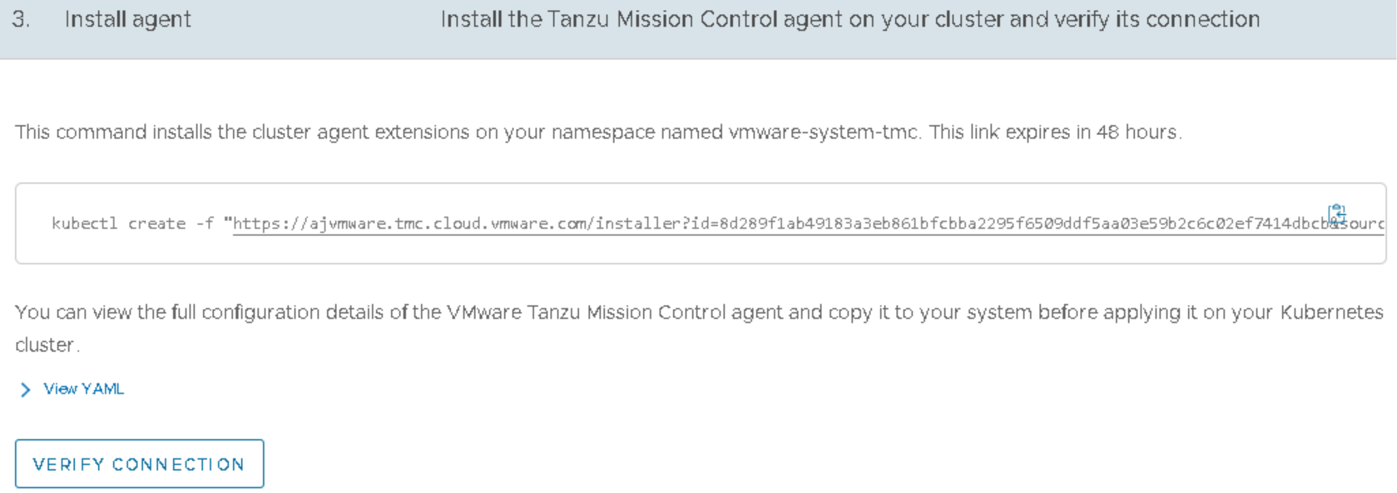
Cluster group: demo-cluster-group

Description (optional): attach existing cluster

Labels (optional): attach : demo

2. On the next screen enter proxy information if you cluster is behind a proxy.
3. On the "Install Agent" step copy the kubectl command.

Figure 145: Install agent.



3. Install agent Install the Tanzu Mission Control agent on your cluster and verify its connection

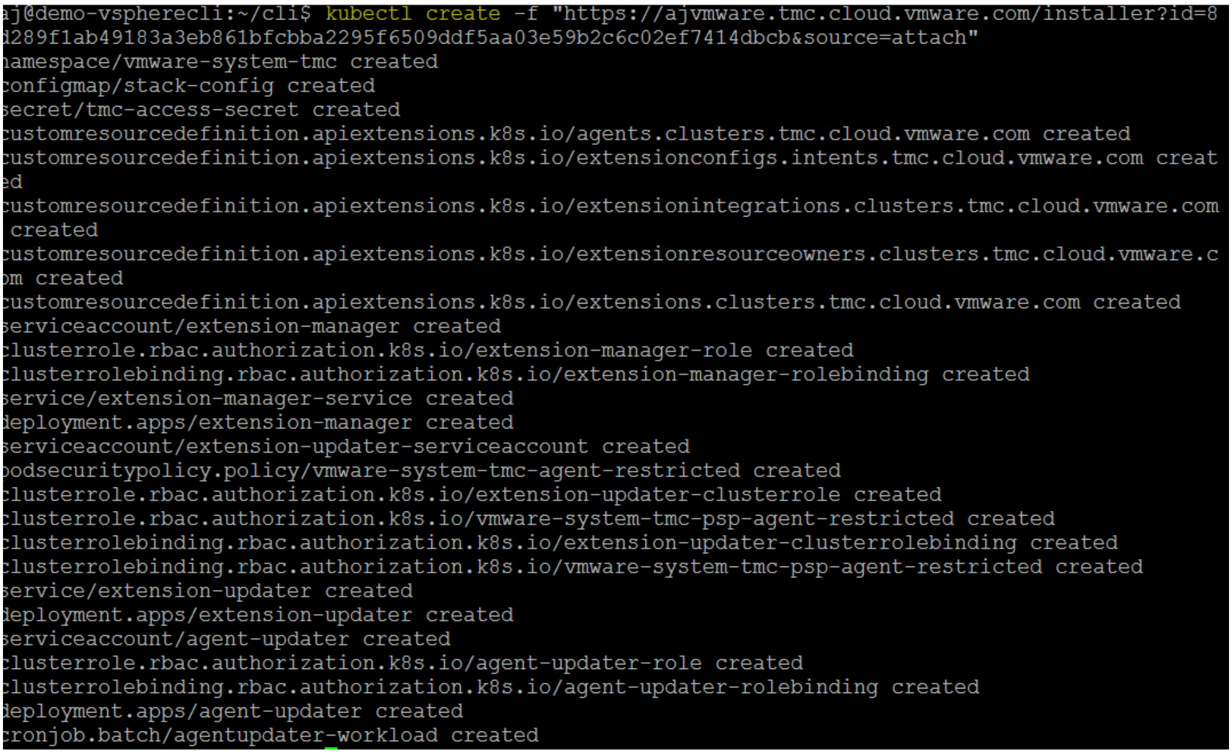
This command installs the cluster agent extensions on your namespace named vmware-system-tmc. This link expires in 48 hours.

```
kubectl create -f "https://ajvmware.tmc.cloud.vmware.com/installer?id=8d289f1ab49183a3eb861bfcbba2295f6509ddf5aa03e59b2c6c02ef7414dbcb&sourc
```

You can view the full configuration details of the VMware Tanzu Mission Control agent and copy it to your system before applying it on your Kubernetes cluster.

4. Login to the cluster and run the command. Cluster should be added, and policies created.

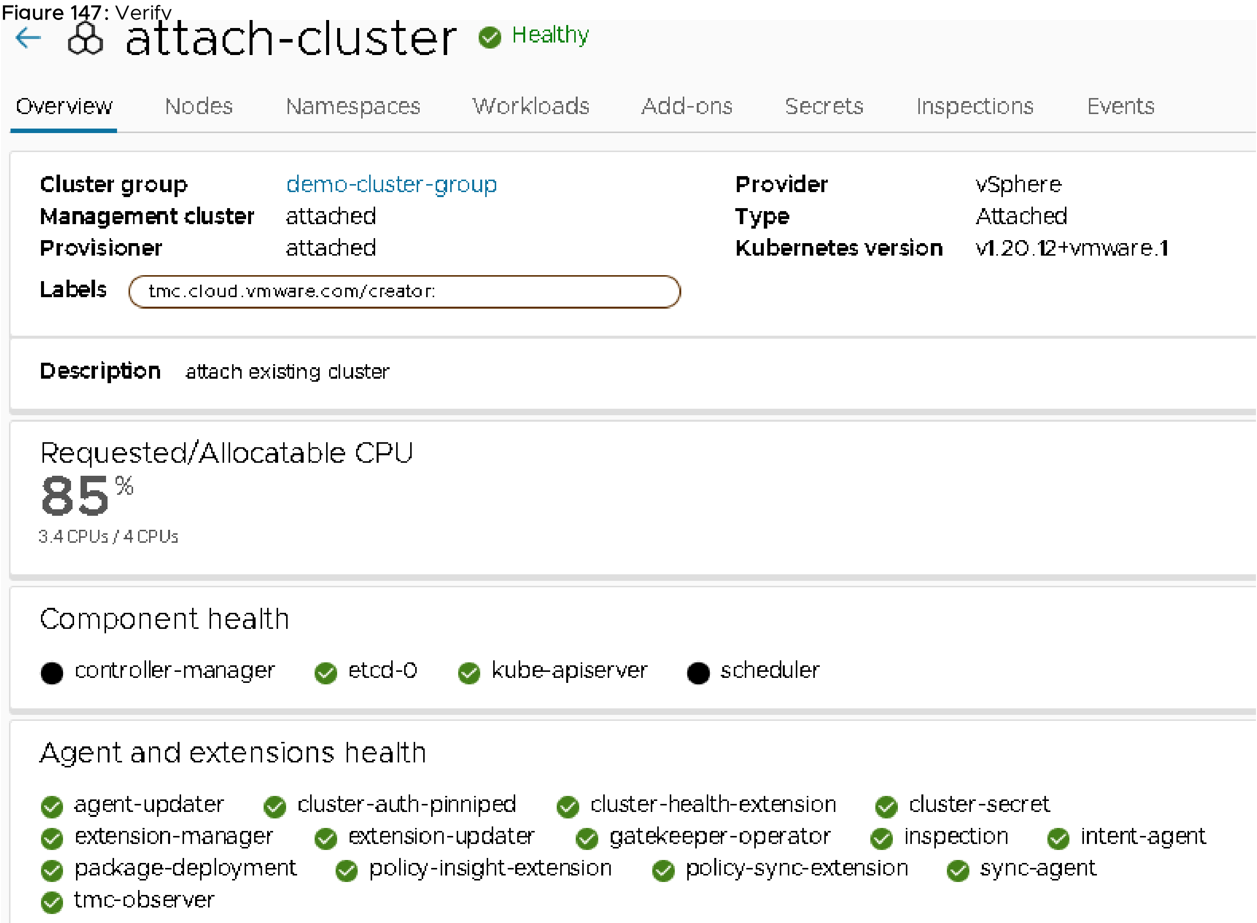
Figure 146: Run cli command



```
aj@demo-vspherecli:~/cli$ kubectl create -f "https://ajvmware.tmc.cloud.vmware.com/installer?id=8
d289f1ab49183a3eb861bfcbba2295f6509ddf5aa03e59b2c6c02ef7414dbcb&source=attach"
namespace/vmware-system-tmc created
configmap/stack-config created
secret/tmc-access-secret created
customresourcedefinition.apiextensions.k8s.io/agents.clusters.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/extensionconfigs.intents.tmc.cloud.vmware.com creat
ed
customresourcedefinition.apiextensions.k8s.io/extensionintegrations.clusters.tmc.cloud.vmware.com
 created
customresourcedefinition.apiextensions.k8s.io/extensionresourceowners.clusters.tmc.cloud.vmware.c
om created
customresourcedefinition.apiextensions.k8s.io/extensions.clusters.tmc.cloud.vmware.com created
serviceaccount/extension-manager created
clusterrole.rbac.authorization.k8s.io/extension-manager-role created
clusterrolebinding.rbac.authorization.k8s.io/extension-manager-rolebinding created
service/extension-manager-service created
deployment.apps/extension-manager created
serviceaccount/extension-updater-serviceaccount created
podsecuritypolicy.policy/vmware-system-tmc-agent-restricted created
clusterrole.rbac.authorization.k8s.io/extension-updater-clusterrole created
clusterrole.rbac.authorization.k8s.io/vmware-system-tmc-psp-agent-restricted created
clusterrolebinding.rbac.authorization.k8s.io/extension-updater-clusterrolebinding created
clusterrolebinding.rbac.authorization.k8s.io/vmware-system-tmc-psp-agent-restricted created
service/extension-updater created
deployment.apps/extension-updater created
serviceaccount/agent-updater created
clusterrole.rbac.authorization.k8s.io/agent-updater-role created
clusterrolebinding.rbac.authorization.k8s.io/agent-updater-rolebinding created
deployment.apps/agent-updater created
cronjob.batch/agentupdater-workload created
```

5. In the Tanzu Mission Control console verify that cluster has been attached

Figure 147: Verify



attach-cluster ✓ Healthy

Cluster group	demo-cluster-group	**Provider**	vSphere
Management cluster	attached	**Type**	Attached
Provisioner	attached	**Kubernetes version**	v1.20.12+vmware.1
Labels	tmc.cloud.vmware.com/creator:		

Description attach existing cluster

Requested/Allocatable CPU

85 %

3.4 CPUs / 4 CPUs

Component health

● controller-manager ✓ etcd-0 ✓ kube-apiserver ● scheduler

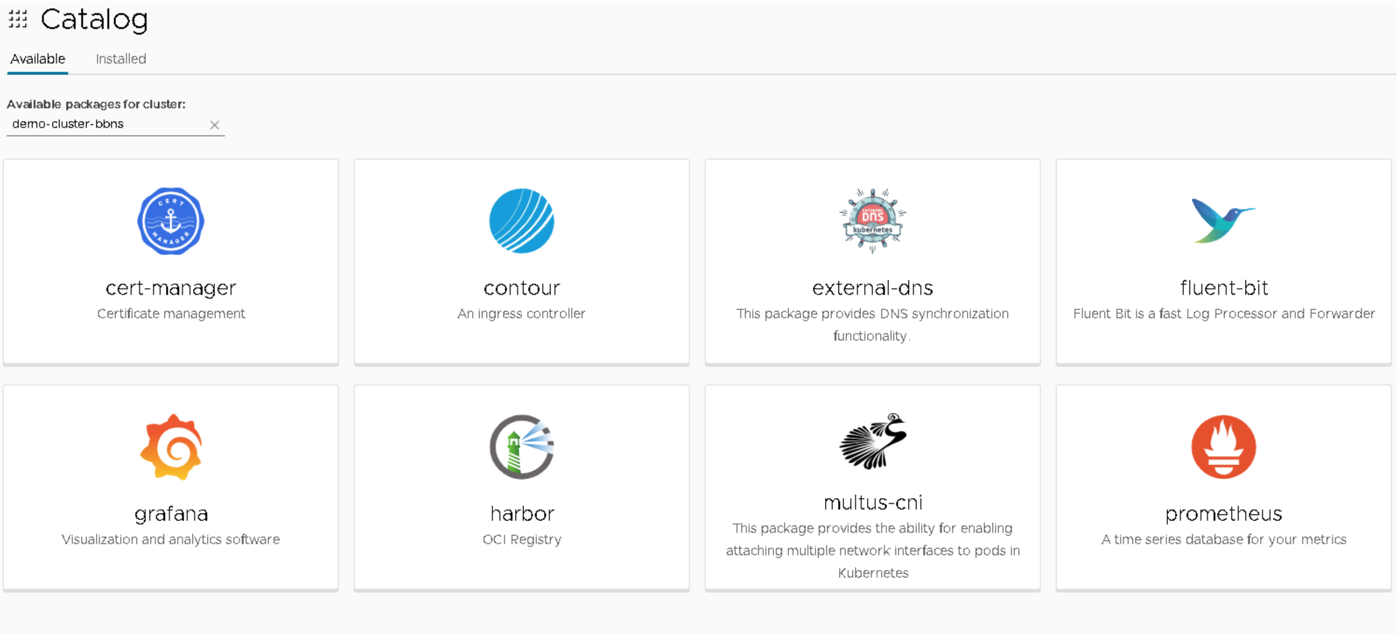
Agent and extensions health

✓ agent-updater ✓ cluster-auth-pinniped ✓ cluster-health-extension ✓ cluster-secret
✓ extension-manager ✓ extension-updater ✓ gatekeeper-operator ✓ inspection ✓ intent-agent
✓ package-deployment ✓ policy-insight-extension ✓ policy-sync-extension ✓ sync-agent
✓ tmc-observer

Tanzu toolkit packages and helm charts

Tanzu Mission Control operators can install, delete, and manage packages on Kubernetes clusters. Tanzu Mission Control uses Carvel for package management. The “Catalog” page shows the packages available to be installed on Kubernetes clusters.

Figure 148: Packages



Catalog

Available packages for cluster: demo-cluster-bbns

cert-manager
Certificate management

contour
An ingress controller

external-dns
This package provides DNS synchronization functionality.

fluent-bit
Fluent Bit is a fast Log Processor and Forwarder

grafana
Visualization and analytics software

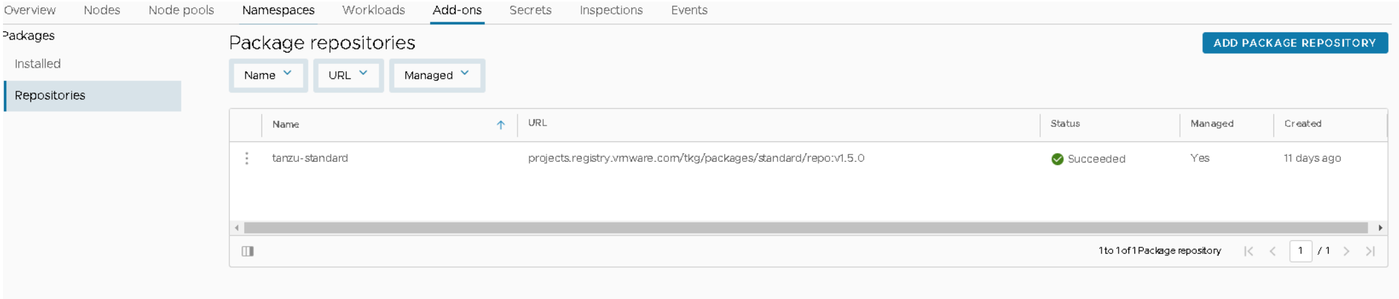
harbor
OCI Registry

multus-cni
This package provides the ability for enabling attaching multiple network interfaces to pods in Kubernetes

prometheus
A time series database for your metrics

Package repositories available for each cluster can be viewed, enabled, or disabled via Cluster > Add-on tab. Custom package repositories can be added via the “Add Package Repository” button.

Figure 149: Repositories



Package repositories

Name	URL	Status	Managed	Created
tanzu-standard	projects.registry.vmware.com/tkg/packages/standard/repo:v1.5.0	✓ Succeeded	Yes	11 days ago

1 to 1 of 1 Package repository 1 / 1

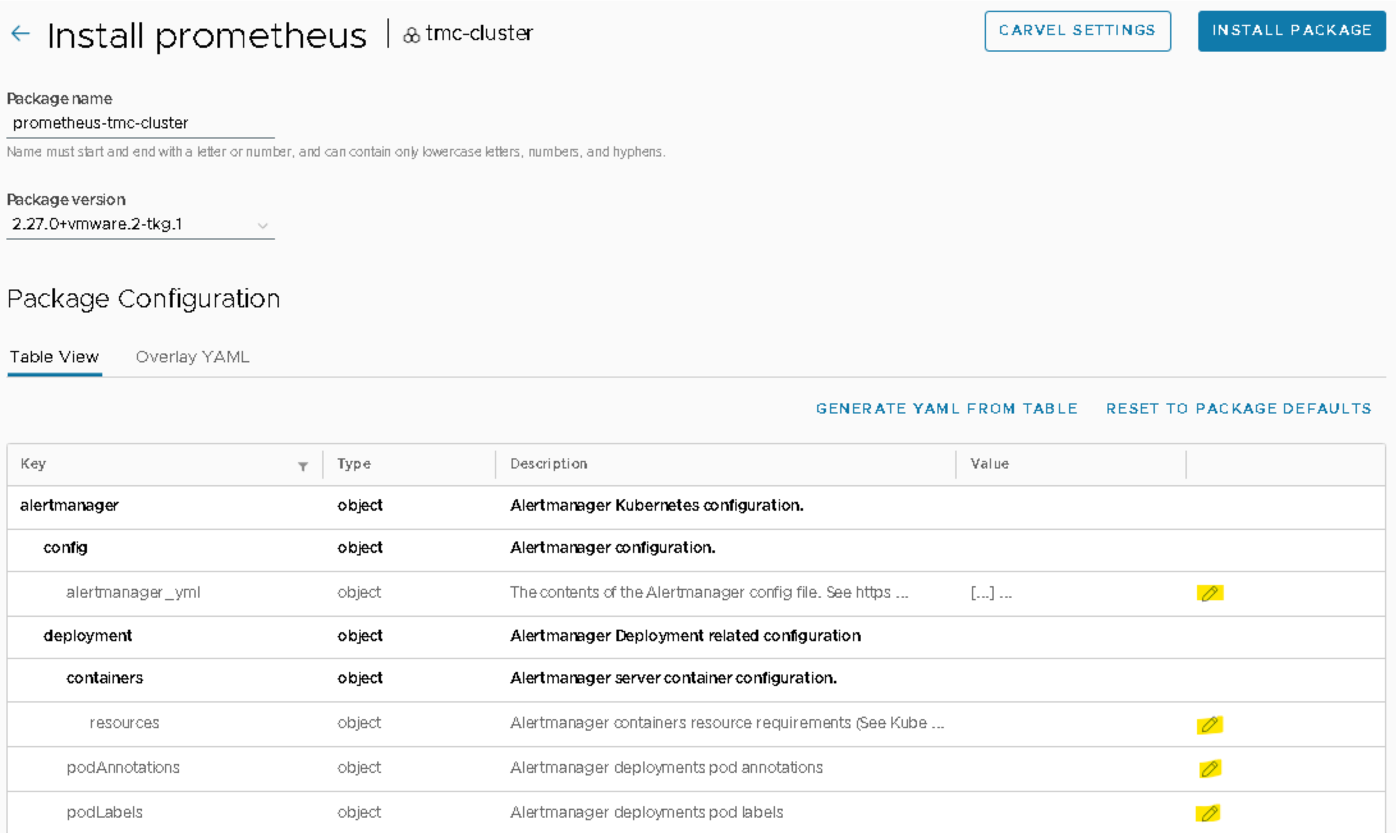
Install Packages

Figure 63 shows the packages available with Tanzu standard repository. Method of deployment is the same for all packages. Some packages however have more customizable fields in Tanzu Mission Control during installation. Below is an example of how to install Prometheus and Grafana using Tanzu Mission Control.

Install Prometheus and Grafana

1. Navigate to Catalog select a cluster and click on Prometheus and select install package.
2. Give the package a name and select a version to be installed from the drop-down list. Under package configuration, fields that have a pencil icon can be modified and configured per your configuration requirement.

Figure 150: Install Prometheus



Install prometheus | tmc-cluster

Package name
prometheus-tmc-cluster
Name must start and end with a letter or number, and can contain only lowercase letters, numbers, and hyphens.

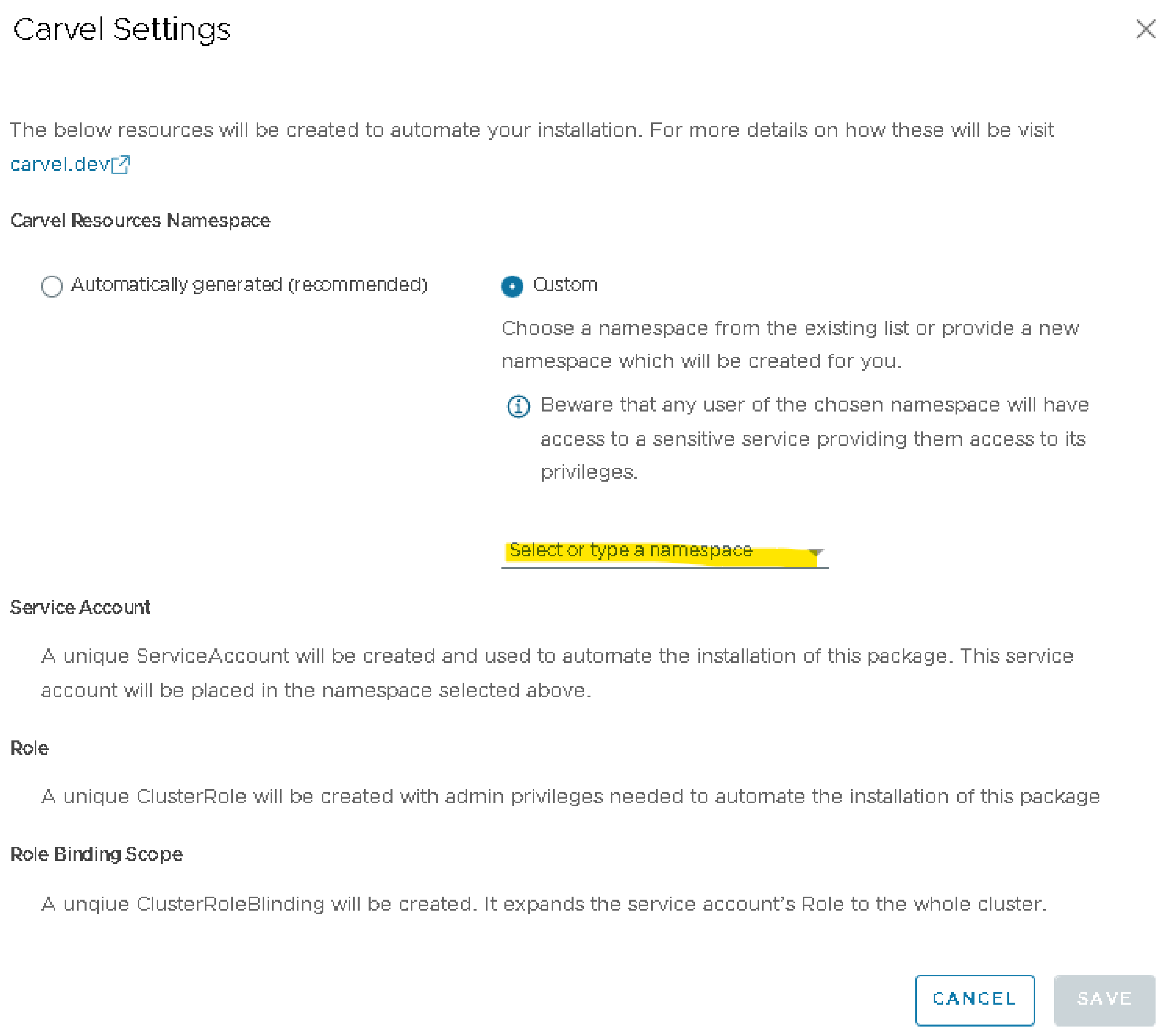
Package version
2.27.0+vmware.2-tkg.1

Package Configuration

Key	Type	Description	Value	
alertmanager	object	Alertmanager Kubernetes configuration.		
config	object	Alertmanager configuration.		
alertmanager_yml	object	The contents of the Alertmanager config file. See https ...	[...] ...	✎
deployment	object	Alertmanager Deployment related configuration		
containers	object	Alertmanager server container configuration.		
resources	object	Alertmanager containers resource requirements (See Kube ...		✎
podAnnotations	object	Alertmanager deployments pod annotations		✎
podLabels	object	Alertmanager deployments pod labels		✎

3. Some Carvel Package settings can be modified such as Carvel Resources namespace via the “Carvel Settings” button.

Figure 151: Carvel settings



Carvel Settings

The below resources will be created to automate your installation. For more details on how these will be visit carvel.dev

Carvel Resources Namespace

○ Automatically generated (recommended) ● Custom

Choose a namespace from the existing list or provide a new namespace which will be created for you.

Beware that any user of the chosen namespace will have access to a sensitive service providing them access to its privileges.

Select or type a namespace

Service Account

A unique ServiceAccount will be created and used to automate the installation of this package. This service account will be placed in the namespace selected above.

Role

A unique ClusterRole will be created with admin privileges needed to automate the installation of this package

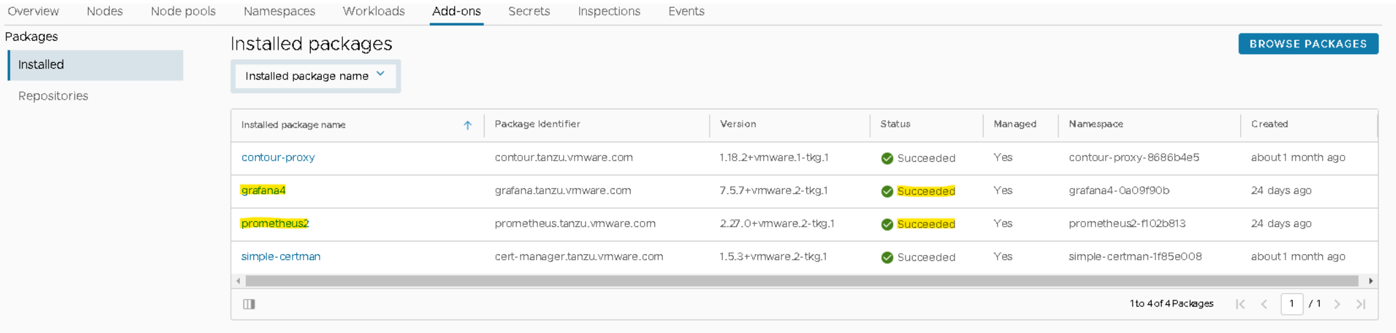
Role Binding Scope

A unqiue ClusterRoleBlinding will be created. It expands the service account’s Role to the whole cluster.

4. Click install package either leaving the settings at default or modify as needed.

5. Once Prometheus is installed successfully, install Grafana similarly.

Figure 152: Prometheus and Grafana



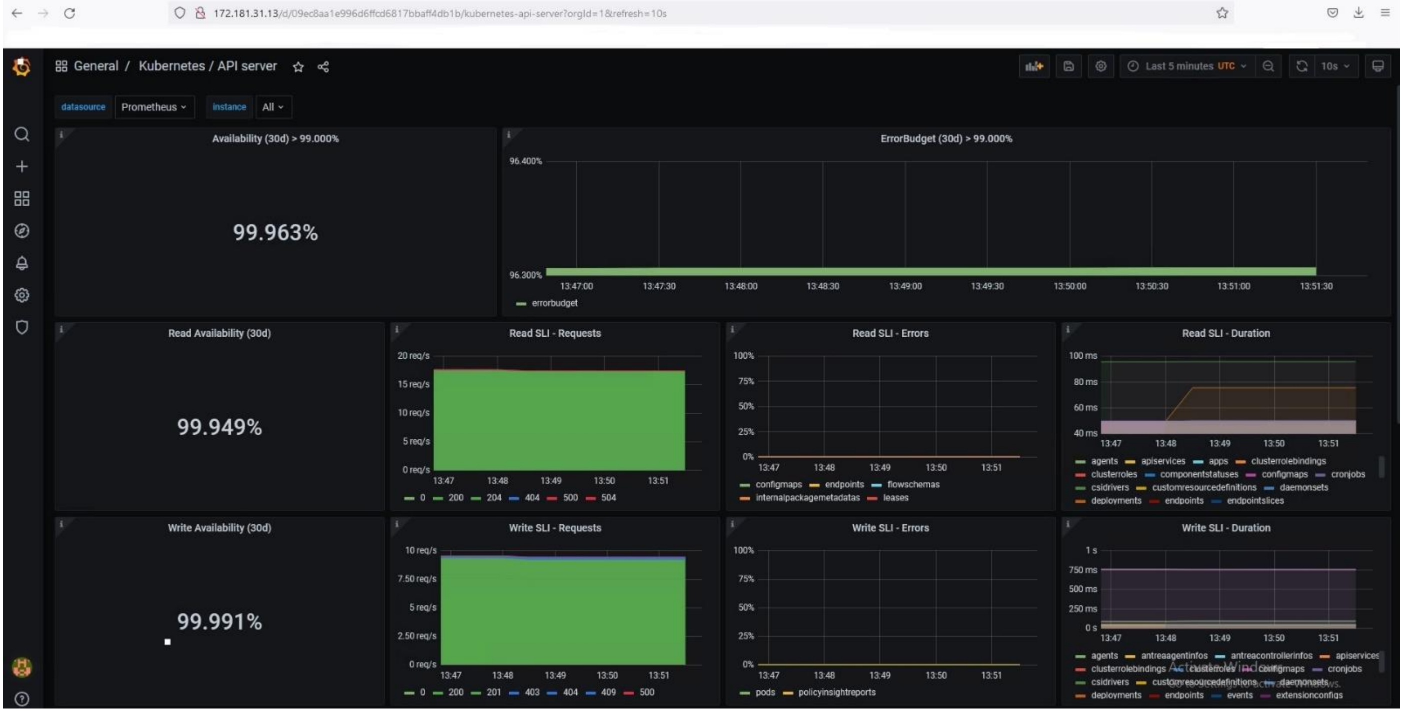
Installed packages

Installed package name	Package Identifier	Version	Status	Managed	Namespace	Created
contour-proxy	contour.tanzu.vmware.com	1.18.2+vmware.1-tkg.1	✓ Succeeded	Yes	contour-proxy-8686b4e5	about 1 month ago
grafana4	grafana.tanzu.vmware.com	7.5.7+vmware.2-tkg.1	✓ Succeeded	Yes	grafana4-0a09f90b	24 days ago
prometheus2	prometheus.tanzu.vmware.com	2.27.0+vmware.2-tkg.1	✓ Succeeded	Yes	prometheus2-f102b813	24 days ago
simple-certman	cert-manager.tanzu.vmware.com	1.5.3+vmware.2-tkg.1	✓ Succeeded	Yes	simple-certman-1f85e008	about 1 month ago

1 to 4 of 4 Packages 1 / 1

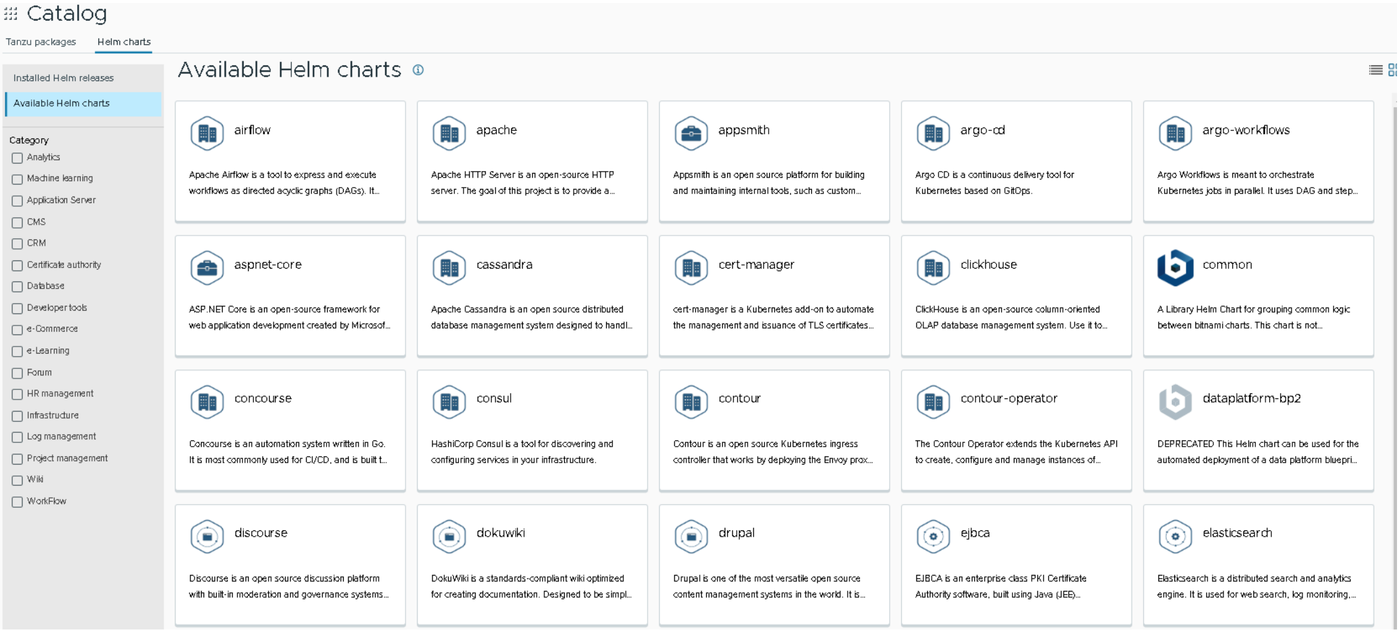
6. Verify that you can access Grafana via its external IP address. Grafana is installed in the “tanzu-system-dashboards” namespace. Use “*kubectl get svc -n tanzu-system-dashboards”* command to get the external IP address Grafana is running on.

Figure 153: Grafana



In addition to Tanzu packages and extensions, Tanzu Mission Control can deploy variety of applications from the “Catalog” using helm charts.

Figure 154: Helm Charts



Catalog

Tanzu packages | Helm charts

Installed Helm releases

Available Helm charts

Category
☐ Analytics
☐ Machine learning
☐ Application Server
☐ CMS
☐ CRM
☐ Certificate authority
☐ Database
☐ Developer tools
☐ e-Commerce
☐ e-Learning
☐ Forum
☐ HR management
☐ Infrastructure
☐ Log management
☐ Project management
☐ Wiki
☐ WorkFlow

Available Helm charts

airflow
Apache Airflow is a tool to express and execute workflows as directed acyclic graphs (DAGs). It...

apache
Apache HTTP Server is an open-source HTTP server. The goal of this project is to provide a...

appsmith
Appsmith is an open source platform for building and maintaining internal tools, such as custom...

argo-cd
Argo CD is a continuous delivery tool for Kubernetes based on GitOps.

argo-workflows
Argo Workflows is meant to orchestrate Kubernetes jobs in parallel. It uses DAG and step...

aspnet-core
ASP.NET Core is an open-source framework for web application development created by Microsof...

cassandra
Apache Cassandra is an open source distributed database management system designed to handl...

cert-manager
cert-manager is a Kubernetes add-on to automate the management and issuance of TLS certificates...

clickhouse
ClickHouse is an open-source column-oriented OLAP database management system. Use it to...

common
A Library Helm Chart for grouping common logic between bitnami charts. This chart is not...

concourse
Concourse is an automation system written in Go. It is most commonly used for CI/CD, and is built t...

consul
HashiCorp Consul is a tool for discovering and configuring services in your infrastructure.

contour
Contour is an open source Kubernetes ingress controller that works by deploying the Envoy prox...

contour-operator
The Contour Operator extends the Kubernetes API to create, configure and manage instances of...

dataplatform-bp2
DEPRECATED This Helm chart can be used for the automated deployment of a data platform bluepri...

discourse
Discourse is an open source discussion platform with built-in moderation and governance systems...

dokuwiki
DokuWiki is a standards-compliant wiki optimized for creating documentation. Designed to be simpl...

drupal
Drupal is one of the most versatile open source content management systems in the world. It is...

ejbca
EJBCA is an enterprise class PKI Certificate Authority software, built using Java (JEE)...

elasticsearch
Elasticsearch is a distributed search and analytics engine. It is used for web search, log monitoring,...

Configure Policies

Various types of policies can be created by the platform administrator to manage operations of Kubernetes environments or other organizational objects. The two policies most relevant to Kubernetes operations are Role

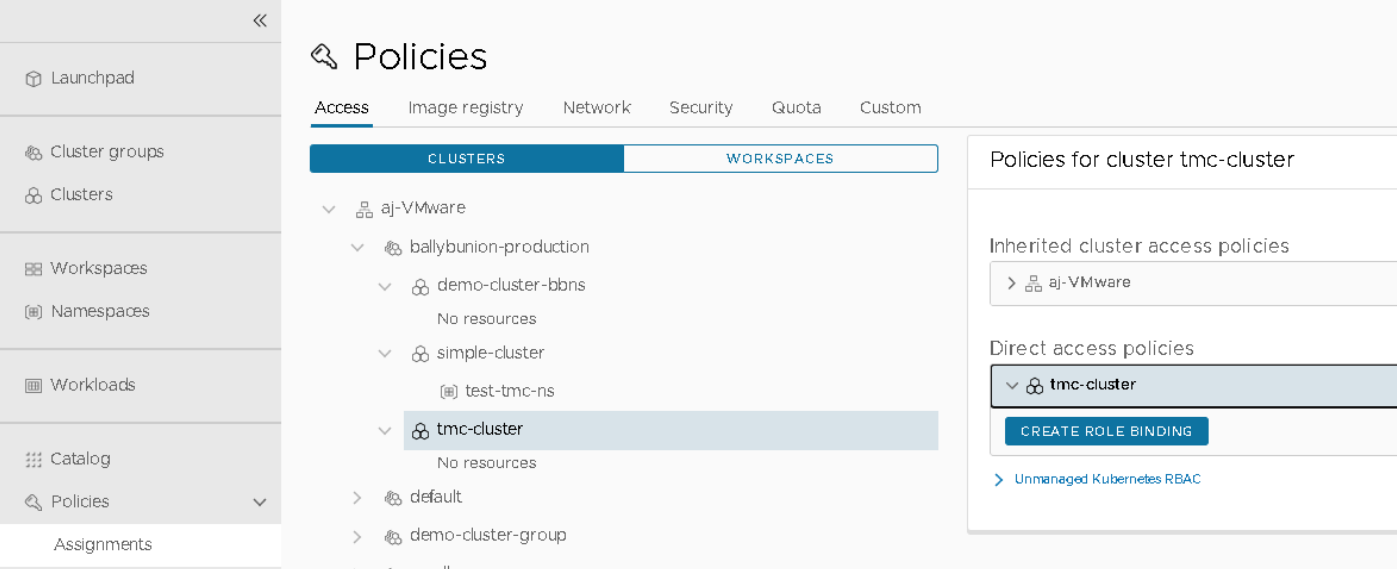
Based Access Control (RBAC) and Security Policies. Please note that security policies are supported on Kubernetes version 1.16 or higher. The application of these policies is discussed in the following section. For more information on policies, roles and role-bindings, please see Policy-Driven Cluster Management.

RBAC and Role binding

Access policies control how users and groups access and manage resources, such as clusters via Tanzu Mission Control. Organizations have predefined roles that govern access to an object based on granted permissions, whereas role binding defines the scope of the access policy to which the role applies. Roles are bound to a given user or group effectively granting permissions to the user or group of users to the desired object. The following example binds a user identity to a cluster via Tanzu Mission Control policy management engine.

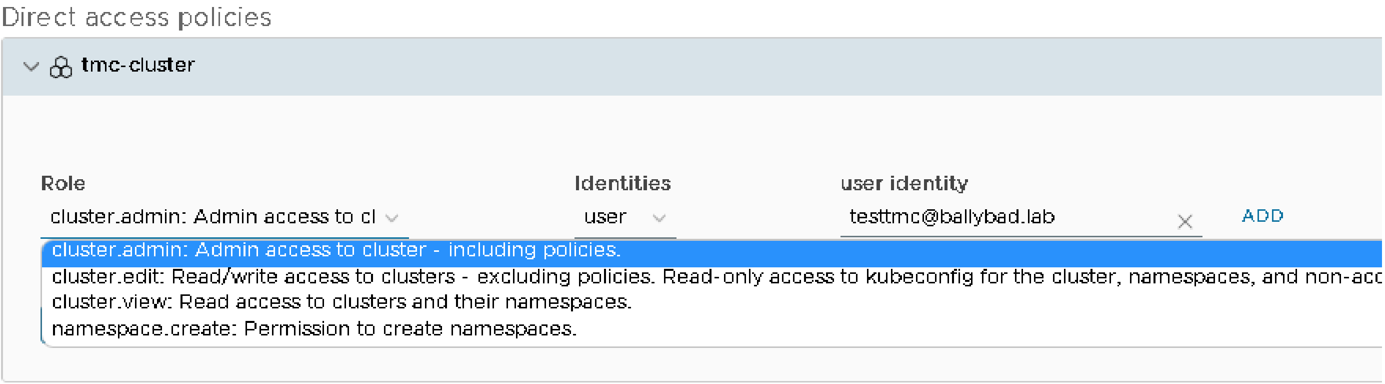
1. From left pane in Tanzu Mission Control navigate to Policies > Assignments > Access tab > Clusters and select the cluster or a group of clusters you want to apply the policy to. Expand the cluster name under “Direct access policies.”

Figure 155: Apply role binding.



2. Create role binding for a user and assign a cluster level role.

Figure 156: Create role binding.



Role: cluster.admin: Admin access to cl

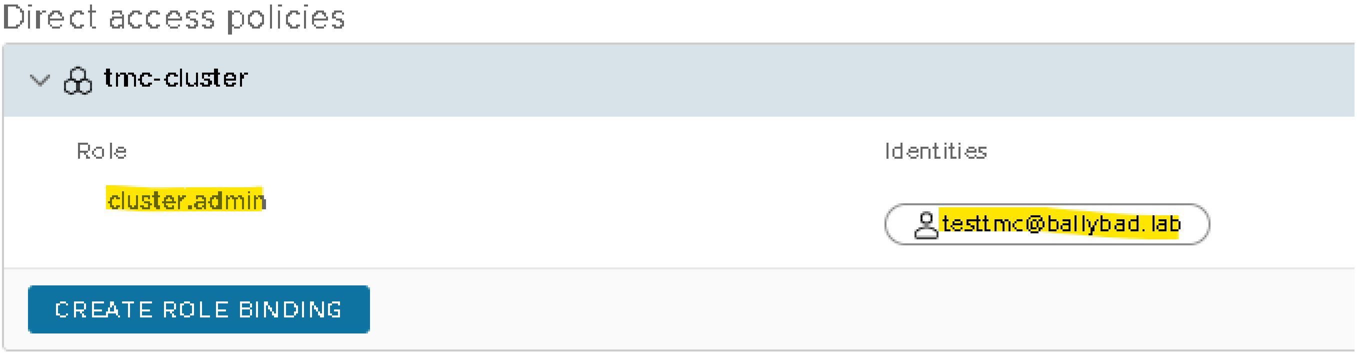
Identities: user

user identity: testtmc@ballybad.lab

cluster.admin: Admin access to cluster - including policies.
cluster.edit: Read/write access to clusters - excluding policies. Read-only access to kubeconfig for the cluster, namespaces, and non-acc
cluster.view: Read access to clusters and their namespaces.
namespace.create: Permission to create namespaces.

3. Click ADD and SAVE. Role binding will be created.

Figure 157: Role binding created



Direct access policies

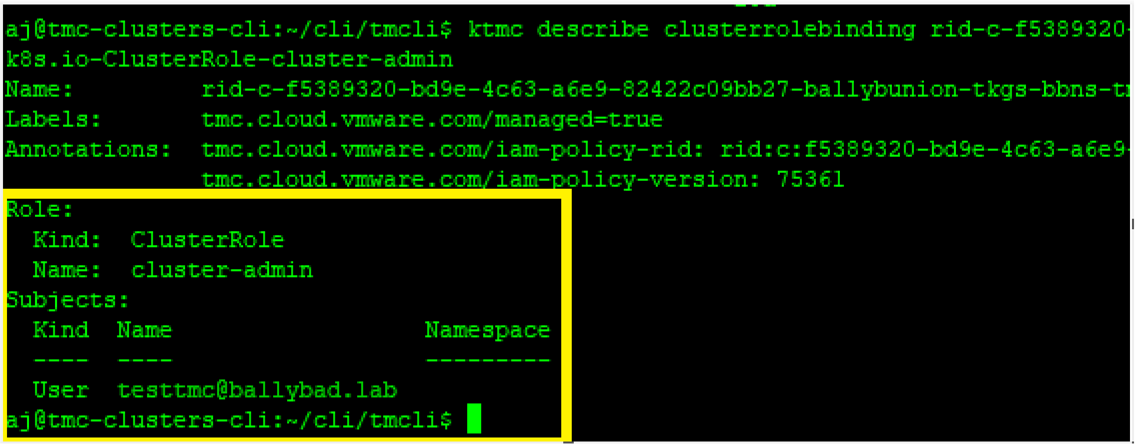
tmc-cluster

Role: cluster.admin

Identities: testtmc@ballybad.lab

4. Verify that role binding is created on the cluster correctly. Use "*Kubectl describe*" command to view role binding configured.

Figure 158: Verify



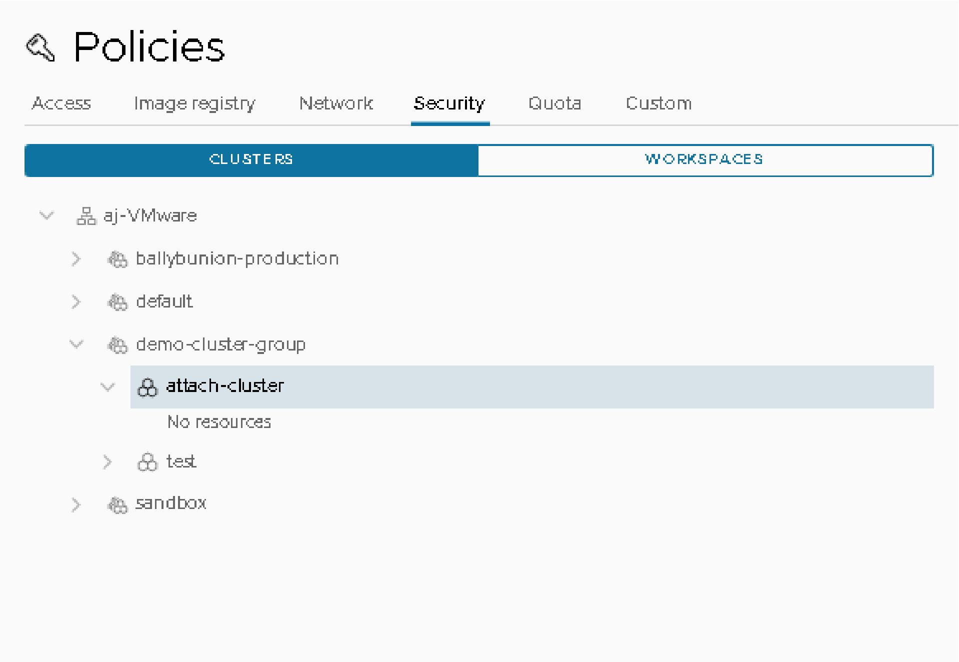
```
aj@tmc-clusters-cli:~/cli/tmcli$ ktmc describe clusterrolebinding rid-c-f5389320-
k8s.io-ClusterRole-cluster-admin
Name:         rid-c-f5389320-bd9e-4c63-a6e9-82422c09bb27-ballybunion-tkgs-bbns-t
Labels:       tmc.cloud.vmware.com/managed=true
Annotations:  tmc.cloud.vmware.com/iam-policy-rid: rid:c:f5389320-bd9e-4c63-a6e9-
              tmc.cloud.vmware.com/iam-policy-version: 75361
Role:
  Kind:  ClusterRole
  Name:  cluster-admin
Subjects:
  Kind  Name                 Namespace
  ----  ----                 ---------
  User  testtmc@ballybad.lab
aj@tmc-clusters-cli:~/cli/tmcli$
```

Security Policies

Security policies allow you to manage the security context in which deployed pods operate in your clusters by imposing constraints on your clusters that define what pods can do and which resources they have access to. Tanzu Mission Control security policies are not implemented using the Kubernetes native "PodSecurityPolicy" object. Tanzu Mission Control uses Gatekeeper project from Open Policy Agent (OPA Gatekeeper). The security-sensitive aspects of the pod specification that they control are, however, the same. For more information, see the OPA Gatekeeper documentation. Tanzu Mission Control with Tanzu Standard only supports pre-defined, "Basic" and "Strict" policies. For custom policy implementation Tanzu Advanced is required. Security Policies can be assigned via Policies > Assignments > Security Tab. Below is an example of how to configure and verify security policies.

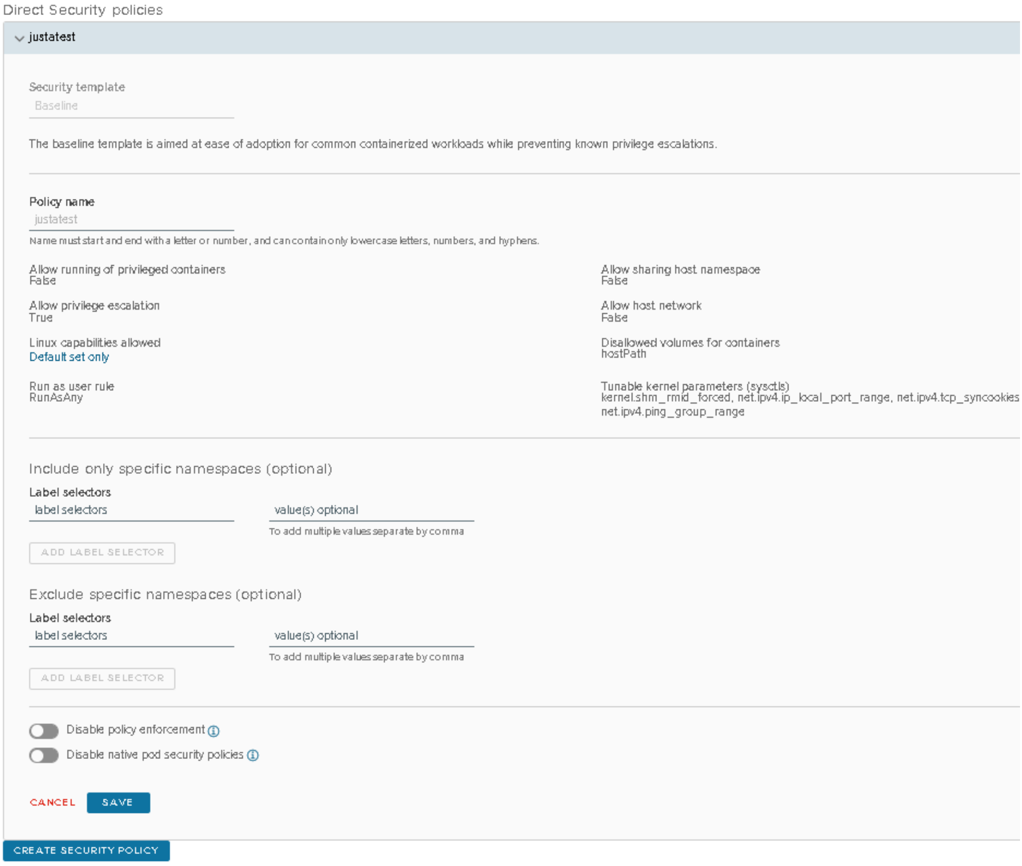
1. Select the cluster or group of clusters the policy will be applied to.

Figure 159: Select cluster.



2. Under “Direct Security Policies” click “create Security Policy.” Select either Basic or Strict security template per your requirements. Give policy a name and enter label selector information if required.

Figure 160: Create policy.


Direct Security policies

justatest

Security template
Baseline

The baseline template is aimed at ease of adoption for common containerized workloads while preventing known privilege escalations.

Policy name
justatest

Name must start and end with a letter or number, and can contain only lowercase letters, numbers, and hyphens.

Allow running of privileged containers
False

Allow privilege escalation
True

Linux capabilities allowed
Default set only

Run as user rule
RunAsAny

Allow sharing host namespace
False

Allow host network
False

Disallowed volumes for containers
hostPath

Tunable kernel parameters (sysctls)
kernel.shm_rmid_forced, net.ipv4.ip_local_port_range, net.ipv4.tcp_syncookies
net.ipv4.ping_group_range

Include only specific namespaces (optional)

Label selectors
label selectors

value(s) optional
To add multiple values separate by comma

Exclude specific namespaces (optional)

Label selectors
label selectors

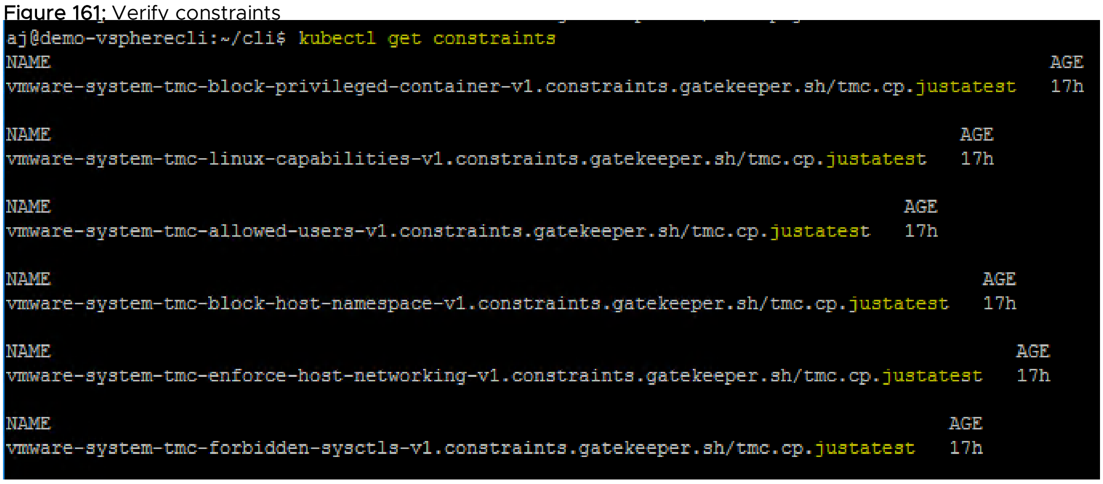
value(s) optional
To add multiple values separate by comma

Disable policy enforcement

Disable native pod security policies

3. Verify that policy is applied to the cluster. Since policies are applied via Gatekeeper constraints and not Kubernetes native POD security policy, you will run command “*kubectl get constraints*” to display applied policies to the cluster. Each constraint that has been applied will be appended by the policy name.

Figure 161: Verify constraints



```
aj@demo-vspherecli:~/cli$ kubectl get constraints
NAME                                                                                   AGE
vmware-system-tmc-block-privileged-container-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h

NAME                                                                           AGE
vmware-system-tmc-linux-capabilities-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h

NAME                                                                      AGE
vmware-system-tmc-allowed-users-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h

NAME                                                                             AGE
vmware-system-tmc-block-host-namespace-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h

NAME                                                                                AGE
vmware-system-tmc-enforce-host-networking-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h

NAME                                                                          AGE
vmware-system-tmc-forbidden-sysctls-v1.constraints.gatekeeper.sh/tmc.cp.justatest   17h
```

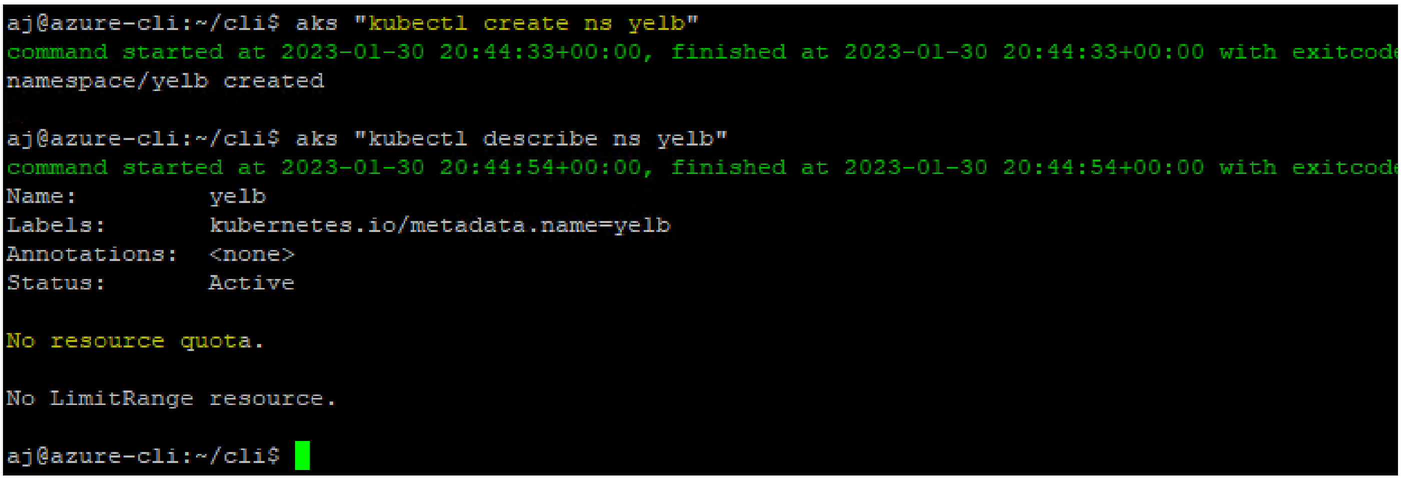
Quota Policies

Quota policies restricts or set boundaries on usage of cluster resource usage. In Tanzu Mission Control there are three preconfigured templates (small, medium, large) that define common limits on CPU and memory requests. There is also a custom template that allows you specify CPU, memory, and storage limits, as well as limits on a variety of object types, including those listed under Object Count Quota in the Kubernetes documentation.

Below example illustrates the use of quotas for a particular namespace (yelb).

1. A sample namespace is created in a cluster with no quotas attached.

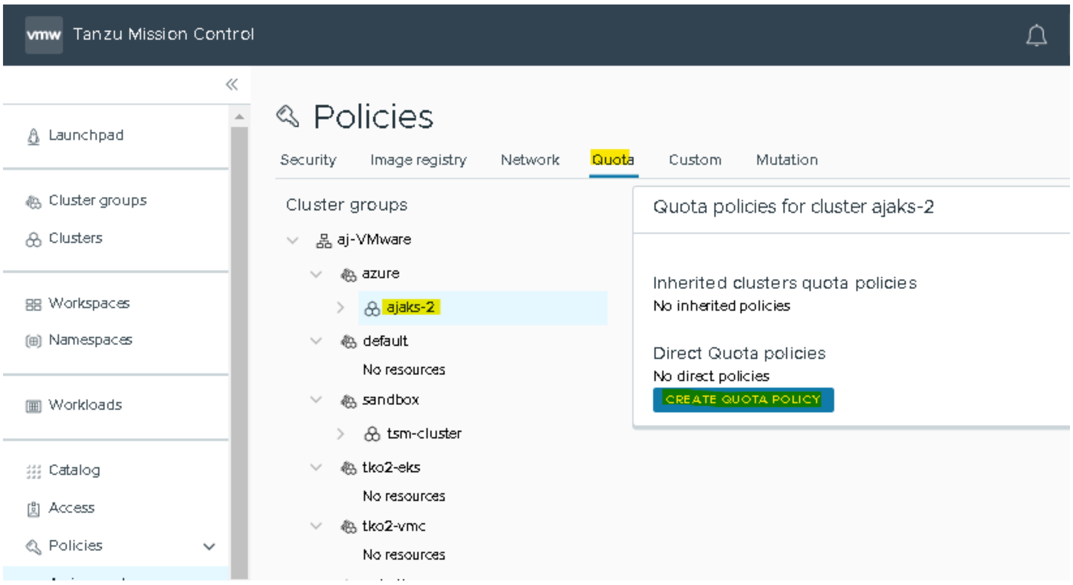
Figure 162: Create namespace.



```
aj@azure-cli:~/cli$ aks "kubectl create ns yelb"
command started at 2023-01-30 20:44:33+00:00, finished at 2023-01-30 20:44:33+00:00 with exitcode
namespace/yelb created

aj@azure-cli:~/cli$ aks "kubectl describe ns yelb"
command started at 2023-01-30 20:44:54+00:00, finished at 2023-01-30 20:44:54+00:00 with exitcode
Name:         yelb
Labels:       kubernetes.io/metadata.name=yelb
Annotations:  <none>
Status:       Active

No resource quota.

No LimitRange resource.

aj@azure-cli:~/cli$ 
```

2. On Tanzu Mission Control the Policies page, click the Quota tab and use the tree control to navigate to and select the cluster or group object for the quota policy needs to be created.

Figure 163: Quota Policy



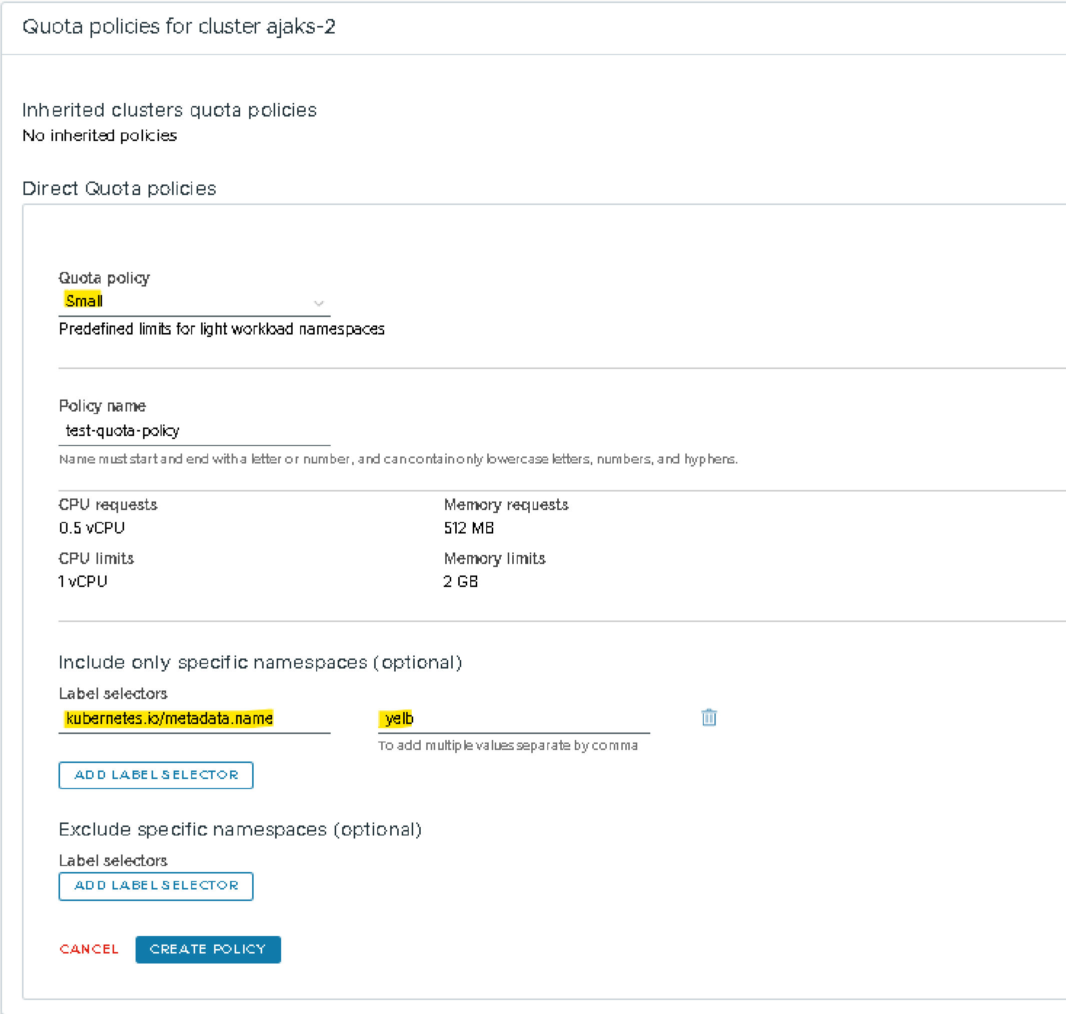
Quota policies for cluster ajaks-2

Inherited clusters quota policies
No inherited policies

Direct Quota policies
No direct policies

3. Select the policy template to use either from the predefined list or create a custom policy. In this example "small" predefined policy is used.

Figure 164: Quota Policy



Quota policies for cluster ajaks-2

Inherited clusters quota policies
No inherited policies

Direct Quota policies

Quota policy
Small
Predefined limits for light workload namespaces

Policy name
test-quota-policy
Name must start and end with a letter or number, and can contain only lowercase letters, numbers, and hyphens.

CPU requests 0.5 vCPU	Memory requests 512 MB
CPU limits 1 vCPU	Memory limits 2 GB

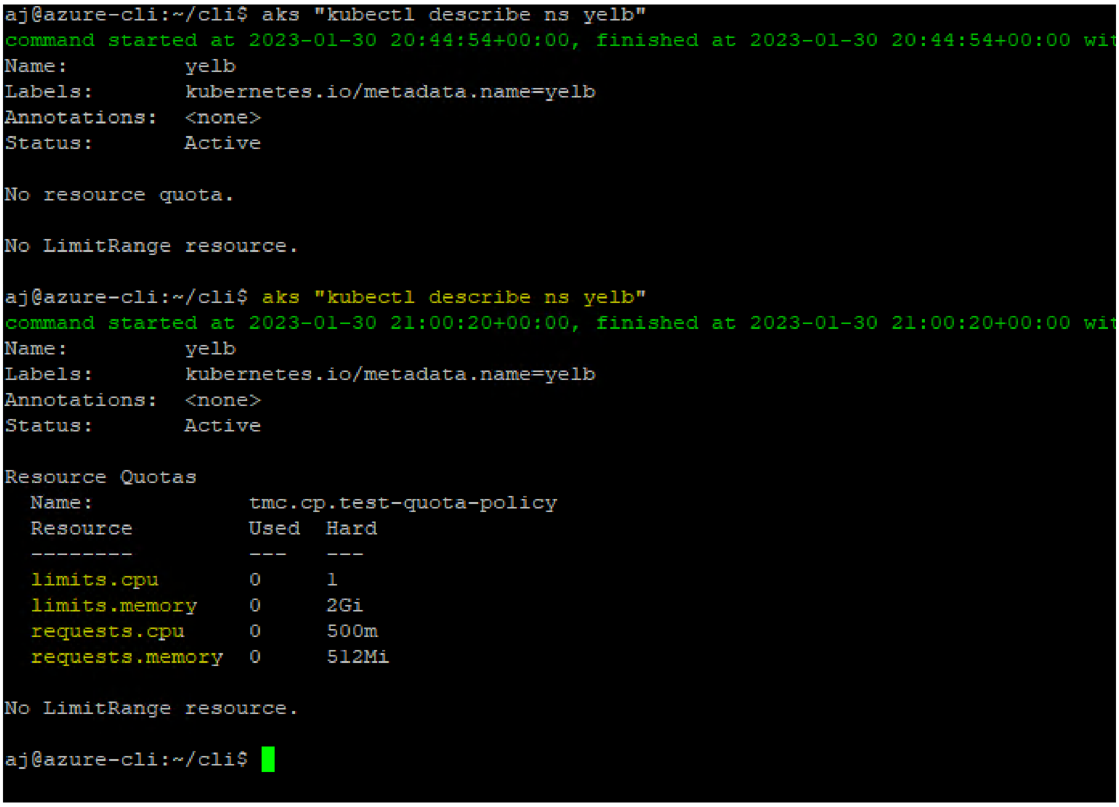
Include only specific namespaces (optional)
Label selectors
kubernetes.io/metadata.name yelb
To add multiple values separate by comma

Exclude specific namespaces (optional)
Label selectors

4. Optionally add label selectors to include or exclude in the calculation of aggregate resource usage. In this example the namespace is used to define the aggregate limit is assigned.
5. Optionally repeat this step to add more label selectors for this policy. Click Create Policy.

Screen shot below shows that the quota was applied to the namespace.

Figure 165: Quota Applied

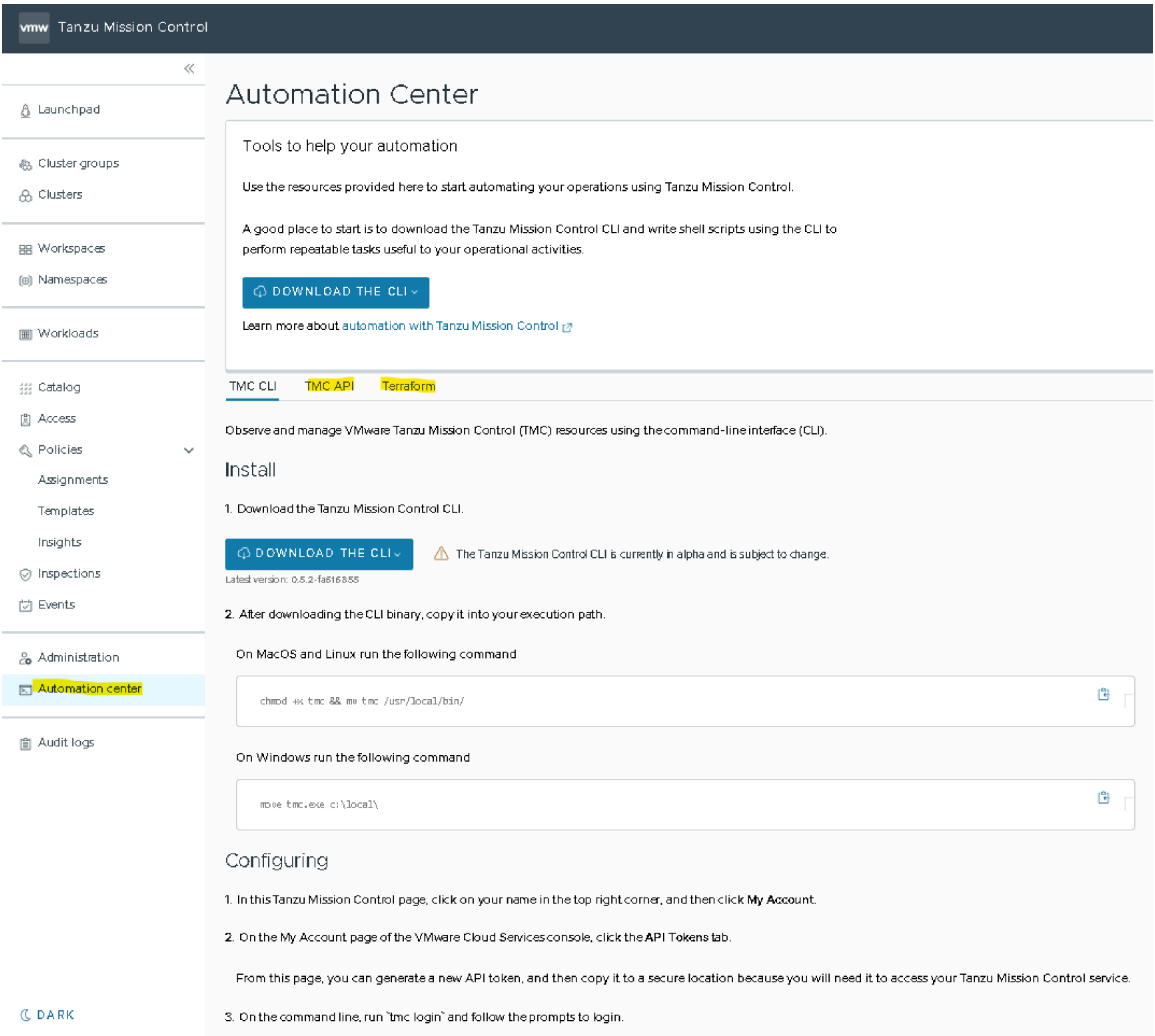


```
aj@azure-cli:~/cli$ aks "kubectl describe ns yelb"
command started at 2023-01-30 20:44:54+00:00, finished at 2023-01-30 20:44:54+00:00 wit
Name:         yelb
Labels:       kubernetes.io/metadata.name=yelb
Annotations:  <none>
Status:       Active

No resource quota.

No LimitRange resource.

aj@azure-cli:~/cli$ aks "kubectl describe ns yelb"
command started at 2023-01-30 21:00:20+00:00, finished at 2023-01-30 21:00:20+00:00 wit
Name:         yelb
Labels:       kubernetes.io/metadata.name=yelb
Annotations:  <none>
Status:       Active

Resource Quotas
  Name:            tmc.cp.test-quota-policy
  Resource         Used  Hard
  --------         ---   ---
  limits.cpu       0     1
  limits.memory    0     2Gi
  requests.cpu     0     500m
  requests.memory  0     512Mi

No LimitRange resource.

aj@azure-cli:~/cli$
```

DevOps access and automation via Tanzu Mission Control CLI

Tanzu Mission Control provides resource management, including clusters via Tanzu Mission Control CLI that can be downloaded via the Tanzu Mission Control portal. In addition, Tanzu Mission Control provides Tanzu Mission Control API and Terraform to manage Tanzu Kubernetes Grid clusters.

Figure 166: Tanzu Mission Control CLI



Automation Center

Tools to help your automation

Use the resources provided here to start automating your operations using Tanzu Mission Control.

A good place to start is to download the Tanzu Mission Control CLI and write shell scripts using the CLI to perform repeatable tasks useful to your operational activities.

Learn more about automation with Tanzu Mission Control

Observe and manage VMware Tanzu Mission Control (TMC) resources using the command-line interface (CLI).

Install

1. Download the Tanzu Mission Control CLI.

⚠ The Tanzu Mission Control CLI is currently in alpha and is subject to change.

Latest version: 0.5.2-fa616855

2. After downloading the CLI binary, copy it into your execution path.

On MacOS and Linux run the following command

```
chmod +x tmc && mv tmc /usr/local/bin/
```

On Windows run the following command

```
move tmc.exe c:\local\
```

Configuring

1. In this Tanzu Mission Control page, click on your name in the top right corner, and then click **My Account**.

2. On the My Account page of the VMware Cloud Services console, click the **API Tokens** tab.

From this page, you can generate a new API token, and then copy it to a secure location because you will need it to access your Tanzu Mission Control service.

3. On the command line, run `tmc login` and follow the prompts to login.

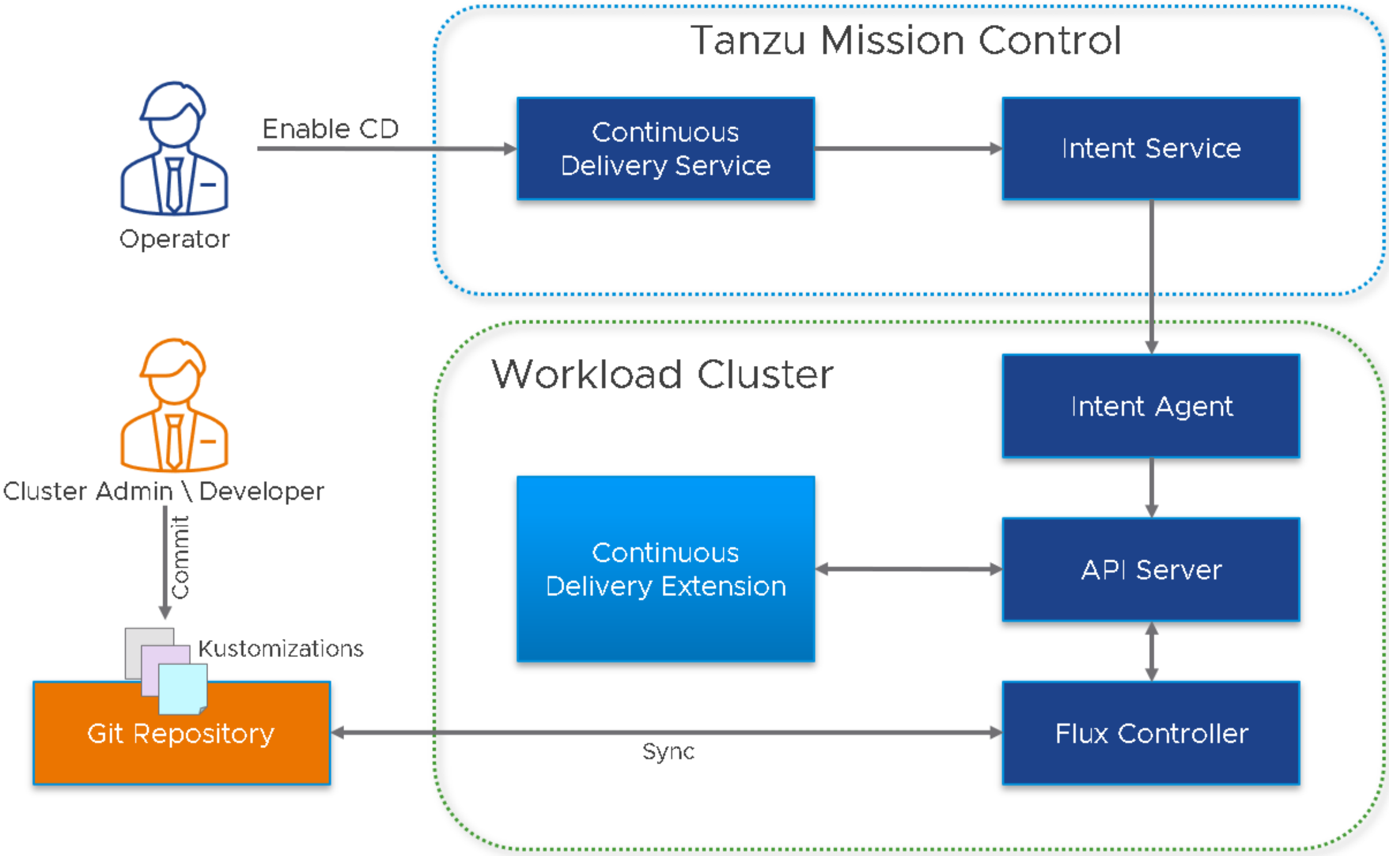
In addition to Tanzu Kubernetes clusters, Tanzu Mission Control can manage complete lifecycle of Amazon EKS clusters as well as existing Azure AKS and Google GKE or any other supported Kubernetes cloud deployment.

Enable Continuous Deliver (CD)

Tanzu Mission Control can now be used to connect Kubernetes clusters to a Git repository, and then manage the cluster's resources declaratively from the repository. Cluster administrators can use Tanzu Mission Control to set up continuous delivery for your clusters. Administrators define the configuration of a cluster (as well as other resources like Helm packages) declaratively using YAML in a Git repository, connect the cluster to the repository, and then synchronize the repository to the cluster. After continuous delivery configuration of a cluster, Tanzu Mission Control drives the continuous delivery of repository objects to the cluster. Continuous delivery can be enabled with or without authentication, depending on requirements.

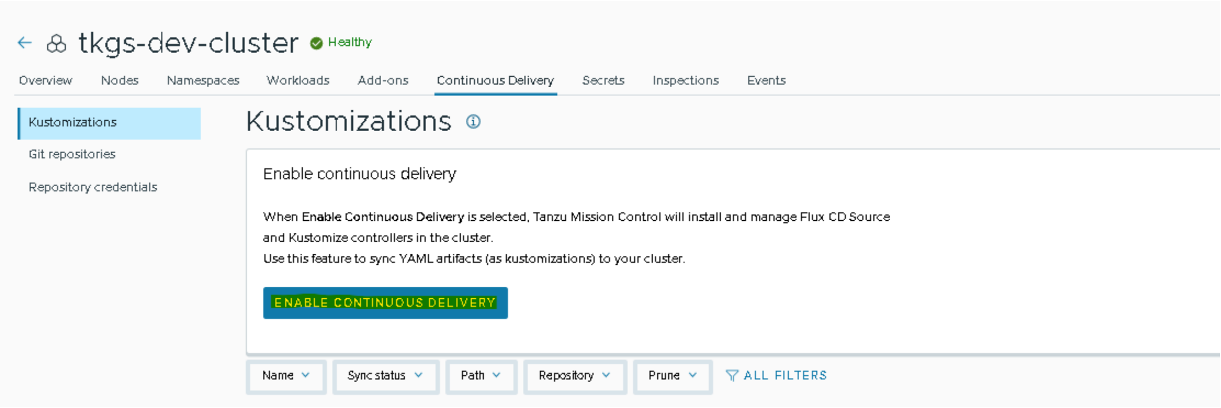
Tanzu Mission Control uses Flux (an open-source community standard) for continuous delivery. Flux uses "Kustomize" to synchronize YAML to your cluster. Kustomize is a standalone tool used to customize Kubernetes objects. Although it is commonly used to apply overlay YAML to existing resources, Kustomize can also be used to create and manage new resources. Flux CD runs in your cluster, connects to your repositories, and periodically synchronizes your defined Kustomization files to your cluster.

Figure 167: Configuration Workflow



Continuous delivery can be enabled on a cluster from Cluster > “Continuous Delivery” tab.

Figure 168: Continuous delivery



Kustomizations

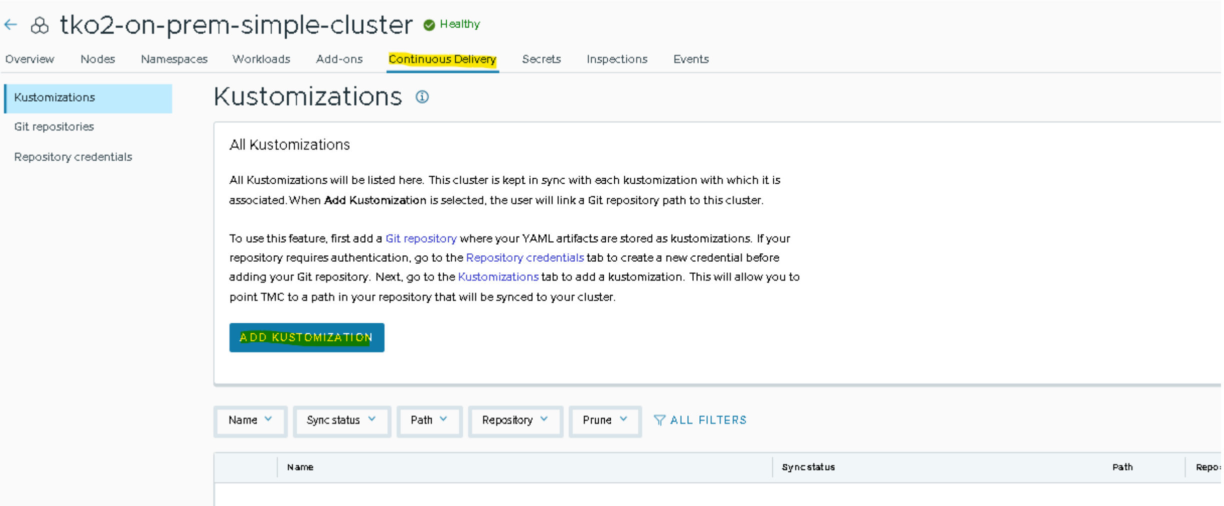
Enable continuous delivery

When Enable Continuous Delivery is selected, Tanzu Mission Control will install and manage Flux CD Source and Kustomize controllers in the cluster.
Use this feature to sync YAML artifacts (as kustomizations) to your cluster.

ENABLE CONTINUOUS DELIVERY

Customizations can now be added if have the repository credentials and Git repositories configured beforehand.

Figure 169: Continuous delivery



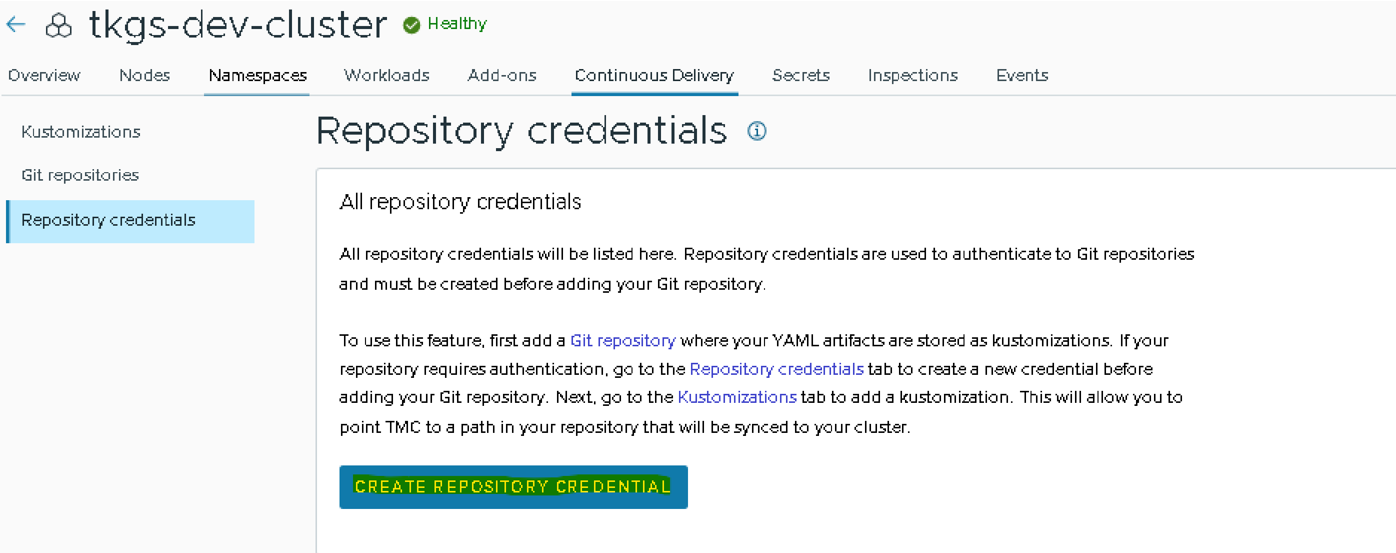
All Kustomizations

All Kustomizations will be listed here. This cluster is kept in sync with each kustomization with which it is associated. When **Add Kustomization** is selected, the user will link a Git repository path to this cluster.

To use this feature, first add a Git repository where your YAML artifacts are stored as kustomizations. If your repository requires authentication, go to the Repository credentials tab to create a new credential before adding your Git repository. Next, go to the Kustomizations tab to add a kustomization. This will allow you to point TMC to a path in your repository that will be synced to your cluster.

To configure credentials, click on repository credentials.

Figure 170: Continuous delivery



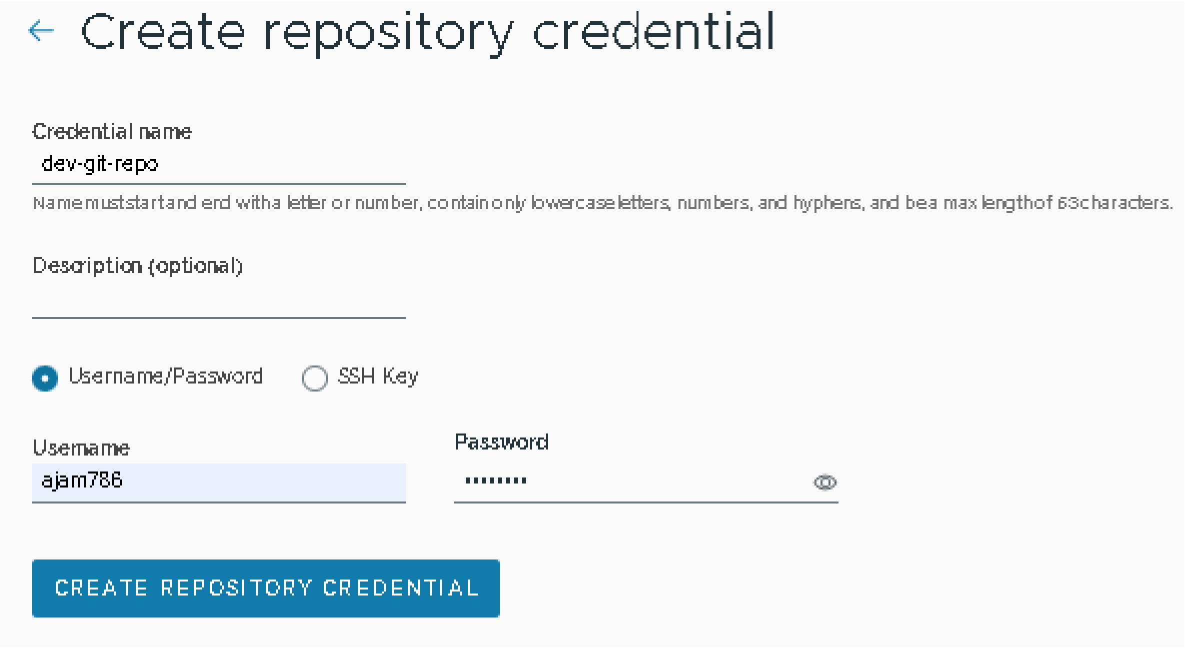
All repository credentials

All repository credentials will be listed here. Repository credentials are used to authenticate to Git repositories and must be created before adding your Git repository.

To use this feature, first add a Git repository where your YAML artifacts are stored as kustomizations. If your repository requires authentication, go to the Repository credentials tab to create a new credential before adding your Git repository. Next, go to the Kustomizations tab to add a kustomization. This will allow you to point TMC to a path in your repository that will be synced to your cluster.

Create credentials either via Gitlab usename/password or ssh key.

Figure 171: Continuous delivery



Create repository credential

Credential name: dev-git-repo

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

Description (optional)

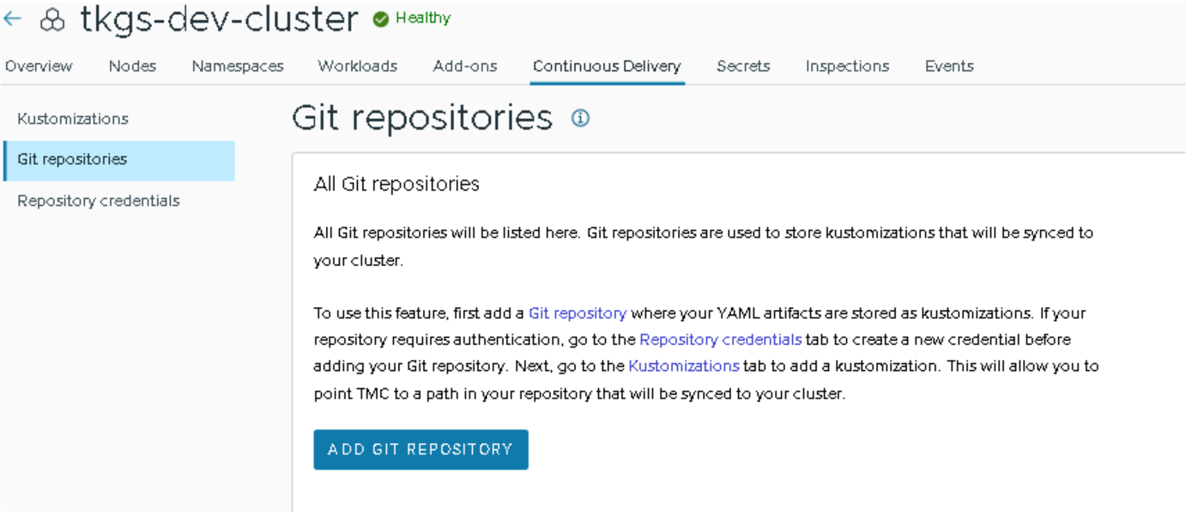
● Username/Password ○ SSH Key

Username: ajam786

Password: ••••••••

Git repository can now be added by clicking the “Add Git Repository” button and entering the Git repo information.

Figure 172: Continuous delivery



tkgs-dev-cluster ✓ Healthy

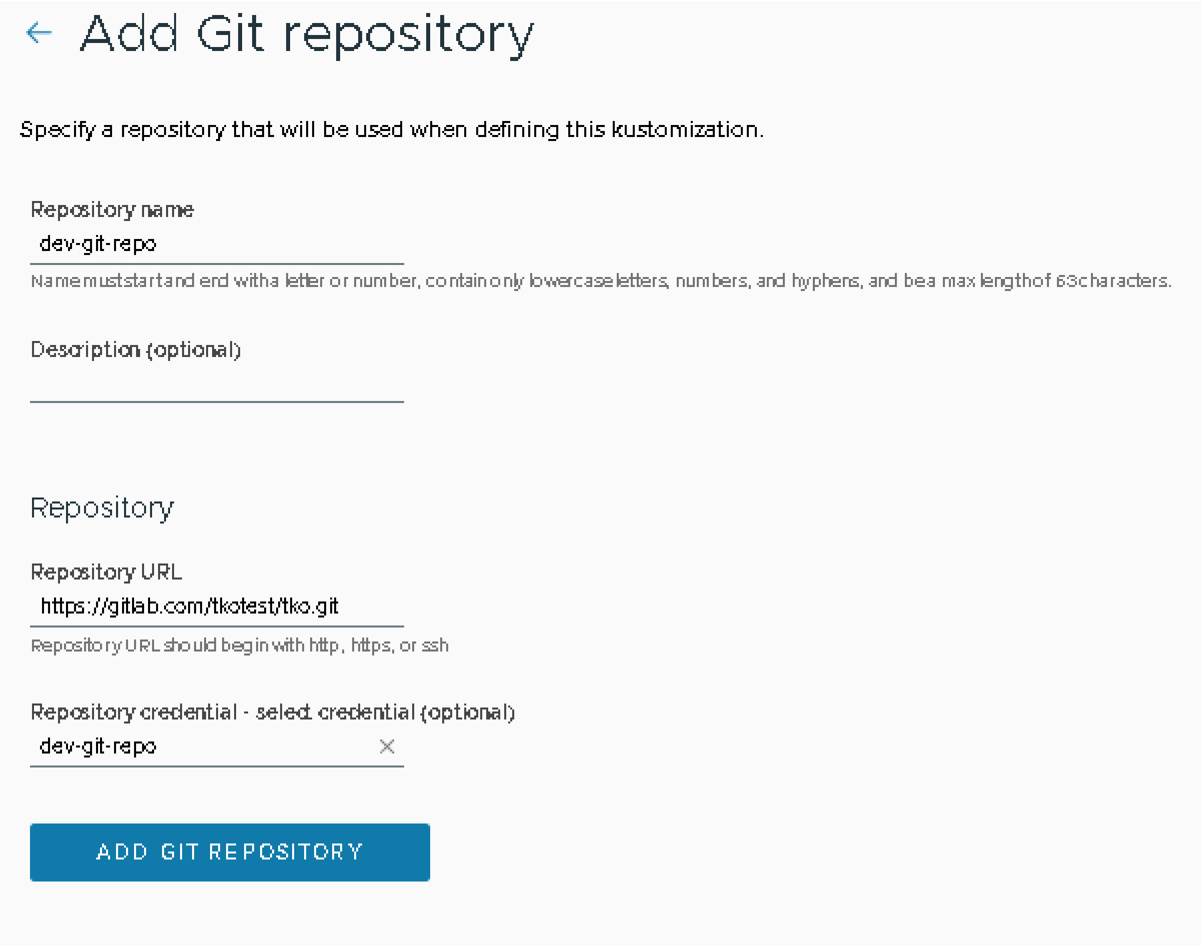
Git repositories

All Git repositories

All Git repositories will be listed here. Git repositories are used to store kustomizations that will be synced to your cluster.

To use this feature, first add a Git repository where your YAML artifacts are stored as kustomizations. If your repository requires authentication, go to the Repository credentials tab to create a new credential before adding your Git repository. Next, go to the Kustomizations tab to add a kustomization. This will allow you to point TMC to a path in your repository that will be synced to your cluster.

Figure 173: Continuous delivery



Add Git repository

Specify a repository that will be used when defining this kustomization.

Repository name
dev-git-repo
Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

Description (optional)

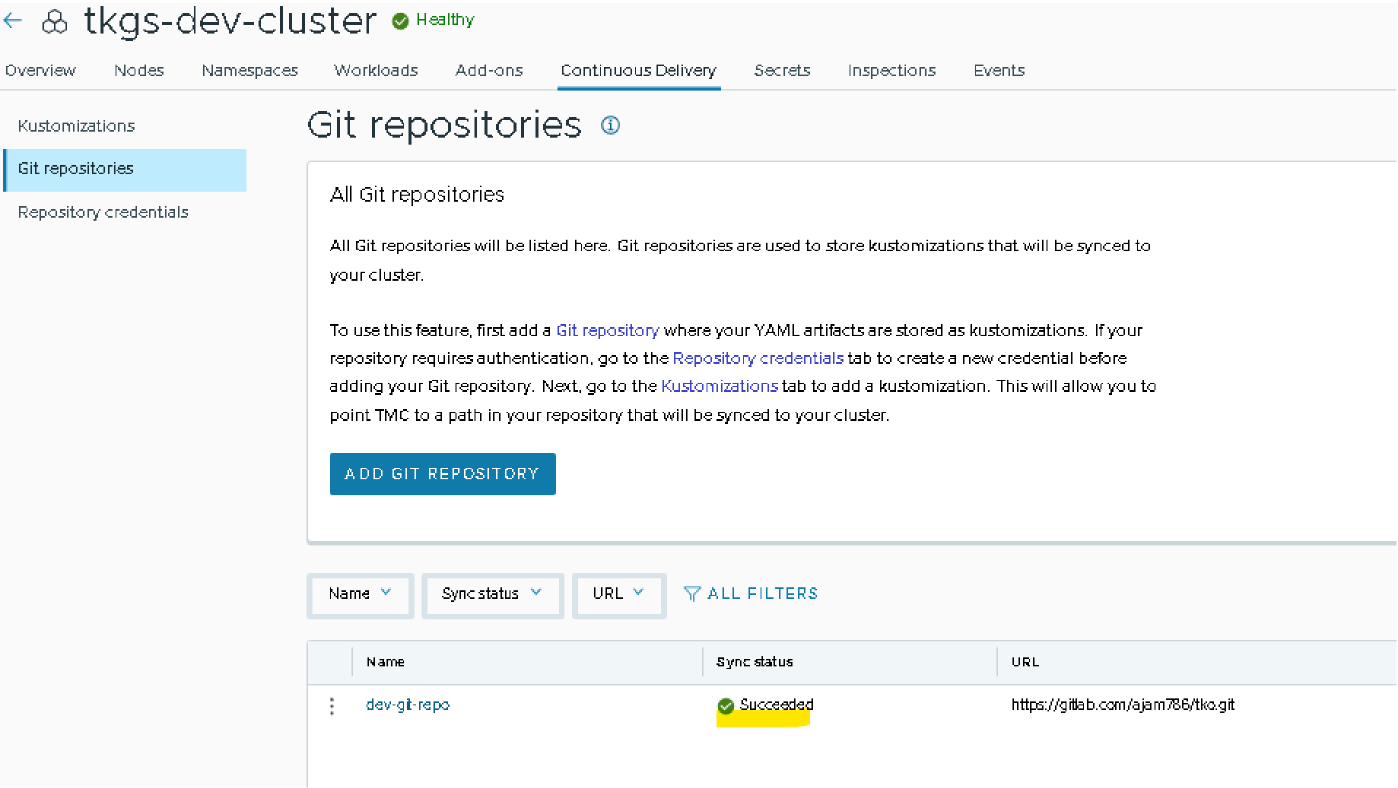
Repository

Repository URL
https://gitlab.com/tkotest/tko.git
Repository URL should begin with http, https, or ssh

Repository credential - select credential (optional)
dev-git-repo

Once the credential verification process is complete, the repository will be ready to be used.

Figure 174: Continuous delivery



All Git repositories

All Git repositories will be listed here. Git repositories are used to store kustomizations that will be synced to your cluster.

To use this feature, first add a Git repository where your YAML artifacts are stored as kustomizations. If your repository requires authentication, go to the Repository credentials tab to create a new credential before adding your Git repository. Next, go to the Kustomizations tab to add a kustomization. This will allow you to point TMC to a path in your repository that will be synced to your cluster.

Name	Sync status	URL
dev-git-repo	Succeeded	https://gitlab.com/ajam786/tko.git

Note: For more information about continuous delivery using Flux, visit https://fluxcd.io/docs/.
For more information about Kustomize, see Declarative Management of Kubernetes Objects Using Kustomize in the Kubernetes documentation.
Please consult Tanzu Mission Control documentation for further details on Continuous Delivery feature and use cases.

Lifecycle management of Amazon EKS clusters

Tanzu Mission Control provides capability to manage complete lifecycle of Amazon EKS clusters. With lifecycle management for Amazon EKS clusters, operations teams will be able to offer more choice to their developers. By centralizing management of multiple Kubernetes cluster types with Tanzu Mission Control, operations teams will be able to efficiently manage their Kubernetes estate through consistent deployment patterns and granular access control and other policies. This capability in preview and intended for general availability soon.

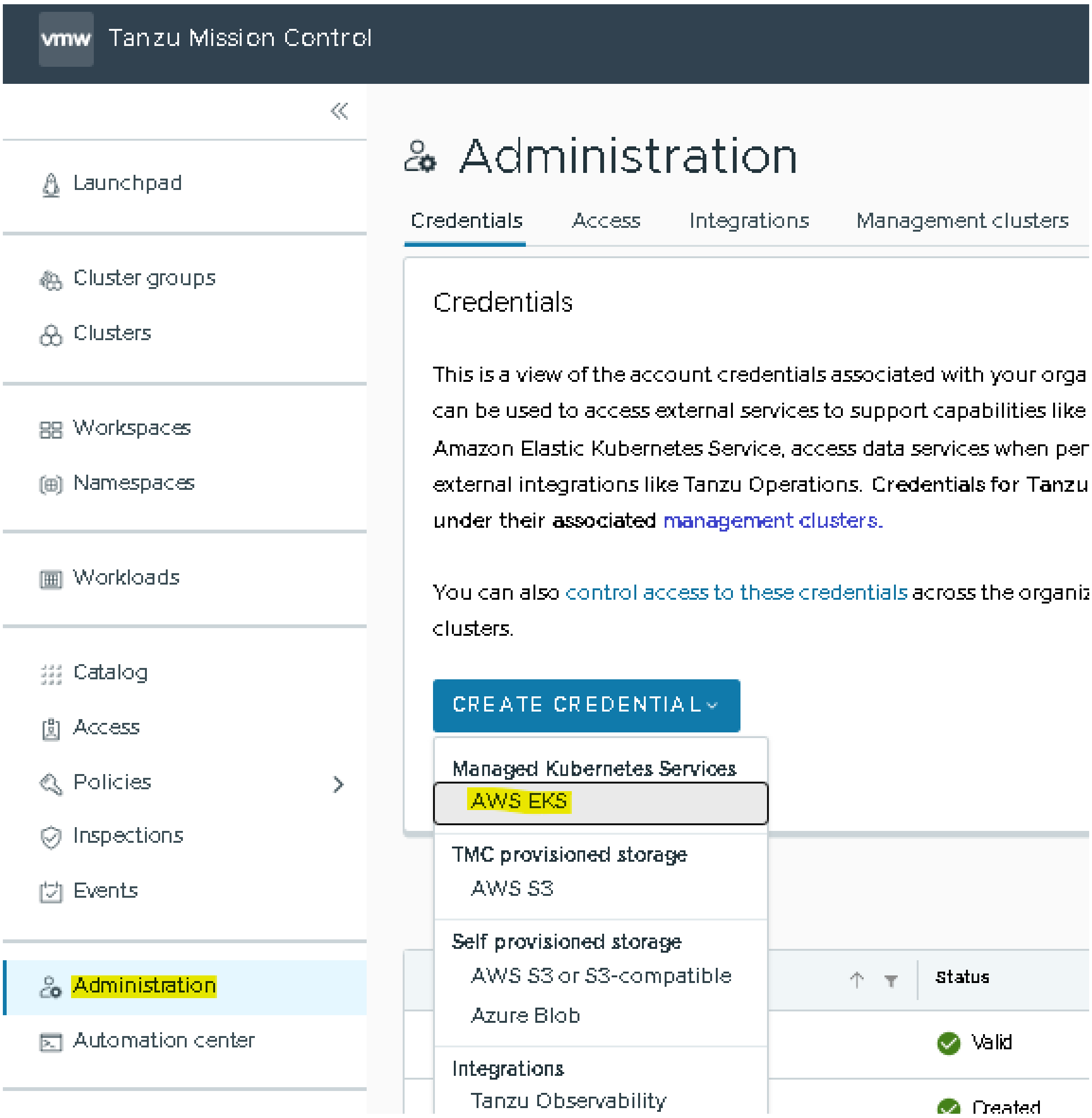
Pre-requisites

- A VPC created with public and private networks.
- User has access to Tanzu Mission Control role cluster.admin role, needed to create credentials.

Create credentials for Amazon EKS lifecycle management.

1. In the Tanzu Mission Control console, click Administration in the left navigation pane.
2. On the Credentials tab of the Administration page, click Create Credential and choose AWS EKS.

Figure 175: Credentials



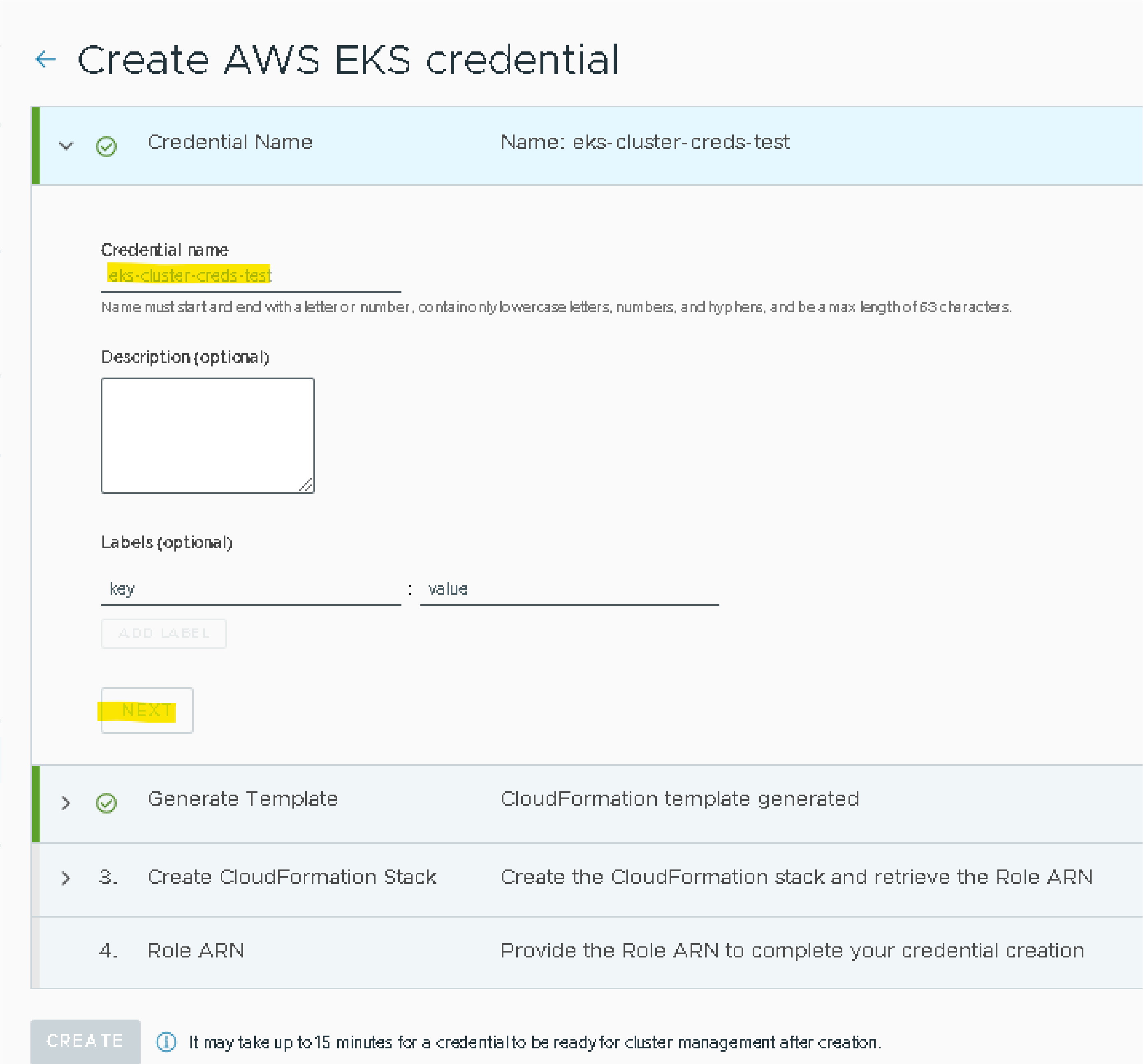
Credentials

This is a view of the account credentials associated with your orga can be used to access external services to support capabilities like Amazon Elastic Kubernetes Service, access data services when per external integrations like Tanzu Operations. Credentials for Tanzu under their associated management clusters.

You can also control access to these credentials across the organiz clusters.

3. On the Create credential page, provide a name for the credential.
4. You can optionally provide a description and labels.
5. Click Next.

Figure 176: EKS Credentials



Create AWS EKS credential

Credential Name — Name: eks-cluster-creds-test

Credential name: eks-cluster-creds-test

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

Description (optional)

Labels (optional)

key : value

Generate Template — CloudFormation template generated

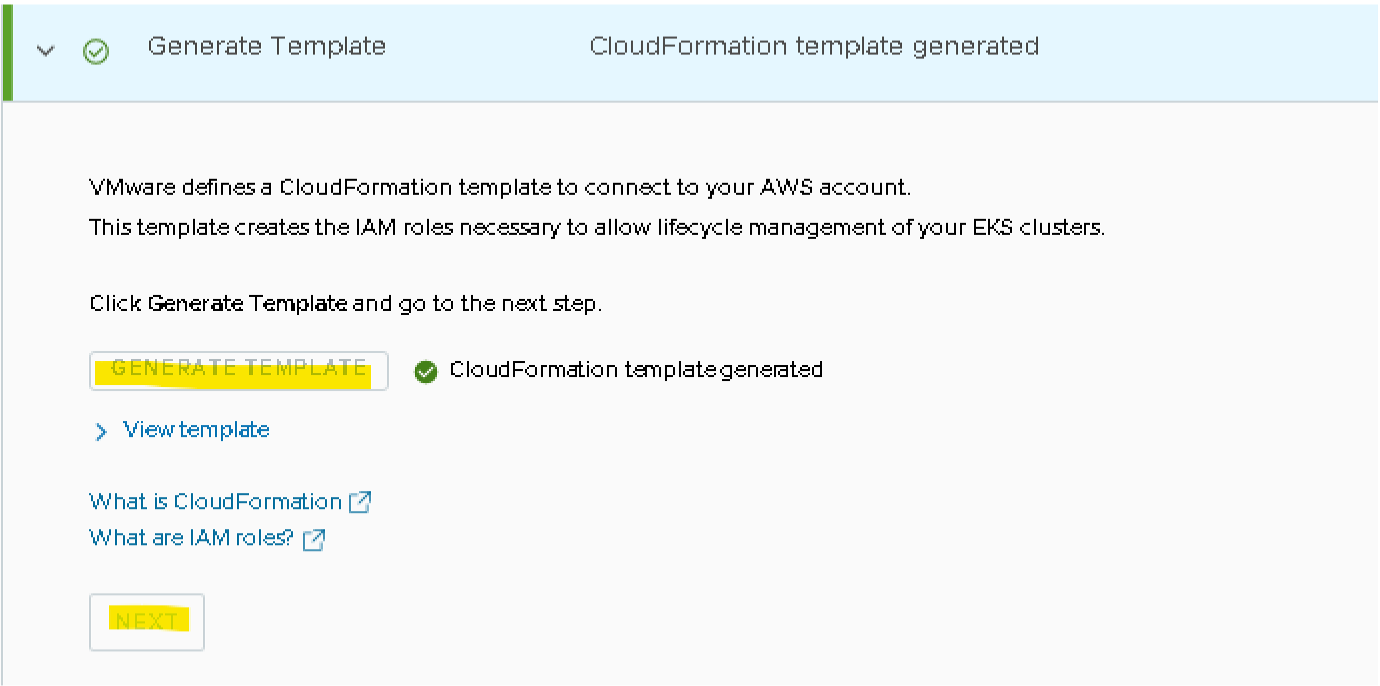
3. Create CloudFormation Stack — Create the CloudFormation stack and retrieve the Role ARN

4. Role ARN — Provide the Role ARN to complete your credential creation

It may take up to 15 minutes for a credential to be ready for cluster management after creation.

6. Click Generate Template, and then after the template is generated, click Next.

Figure 177: Generate template.



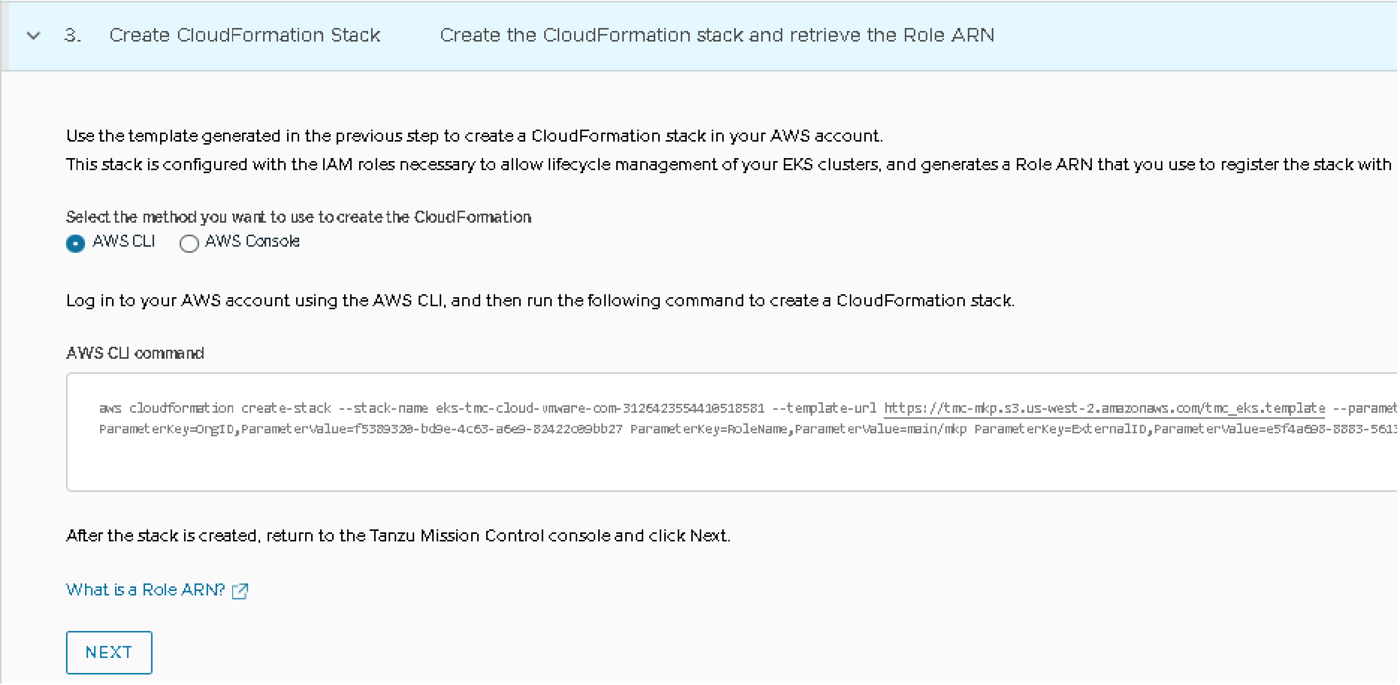
Generate Template   CloudFormation template generated

VMware defines a CloudFormation template to connect to your AWS account.
This template creates the IAM roles necessary to allow lifecycle management of your EKS clusters.

Click Generate Template and go to the next step.

7. Use the generated template in one of two ways to create the AWS CloudFormation stack, either via the AWS CLI or the AWS console UI.

Figure 178: Create Cloud Formation stack.



3. Create CloudFormation Stack   Create the CloudFormation stack and retrieve the Role ARN

Use the template generated in the previous step to create a CloudFormation stack in your AWS account.
This stack is configured with the IAM roles necessary to allow lifecycle management of your EKS clusters, and generates a Role ARN that you use to register the stack with

Select the method you want to use to create the CloudFormation
● AWS CLI ○ AWS Console

Log in to your AWS account using the AWS CLI, and then run the following command to create a CloudFormation stack.

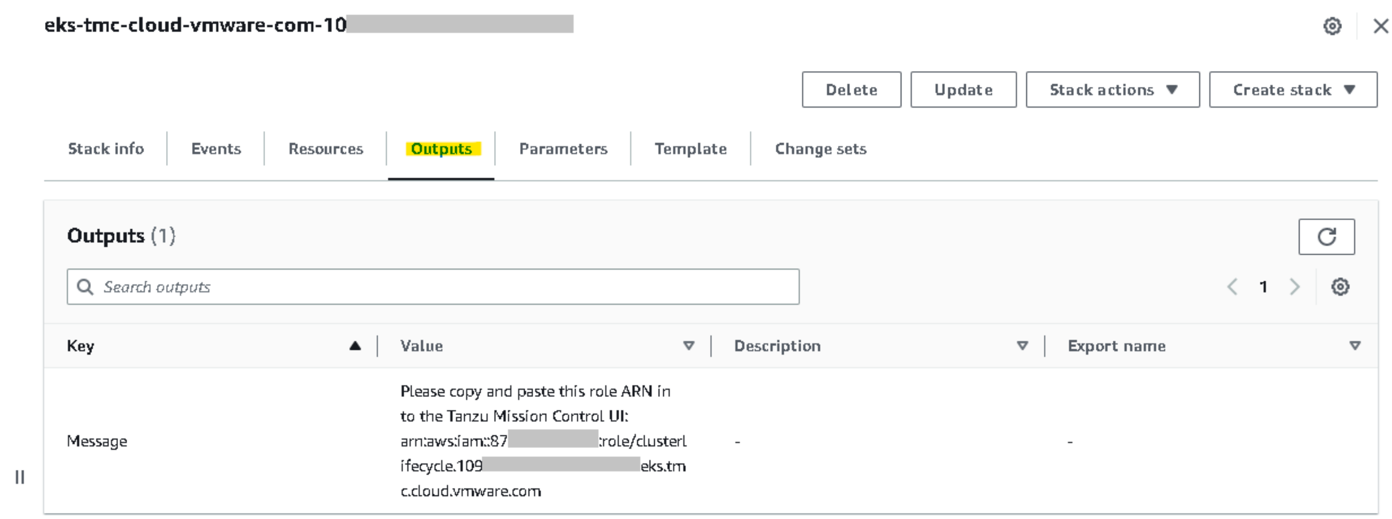
AWS CLI command

```
aws cloudformation create-stack --stack-name eks-tmc-cloud-vmware-com-3126423554410518581 --template-url https://tmc-mkp.s3.us-west-2.amazonaws.com/tmc_eks.template --paramet
ParameterKey=OrgID,ParameterValue=f5389320-bd9e-4c63-a6e9-82422c09bb27 ParameterKey=RoleName,ParameterValue=main/mkp ParameterKey=ExternalID,ParameterValue=e5f4a698-8883-561
```

After the stack is created, return to the Tanzu Mission Control console and click Next.

8. Retrieve the Role ARN using the CLI or AWS console UI by navigating to CloudFormation > Stacks > <your stack> > Outputs.

Figure 179: Retrieve credentials via UI.



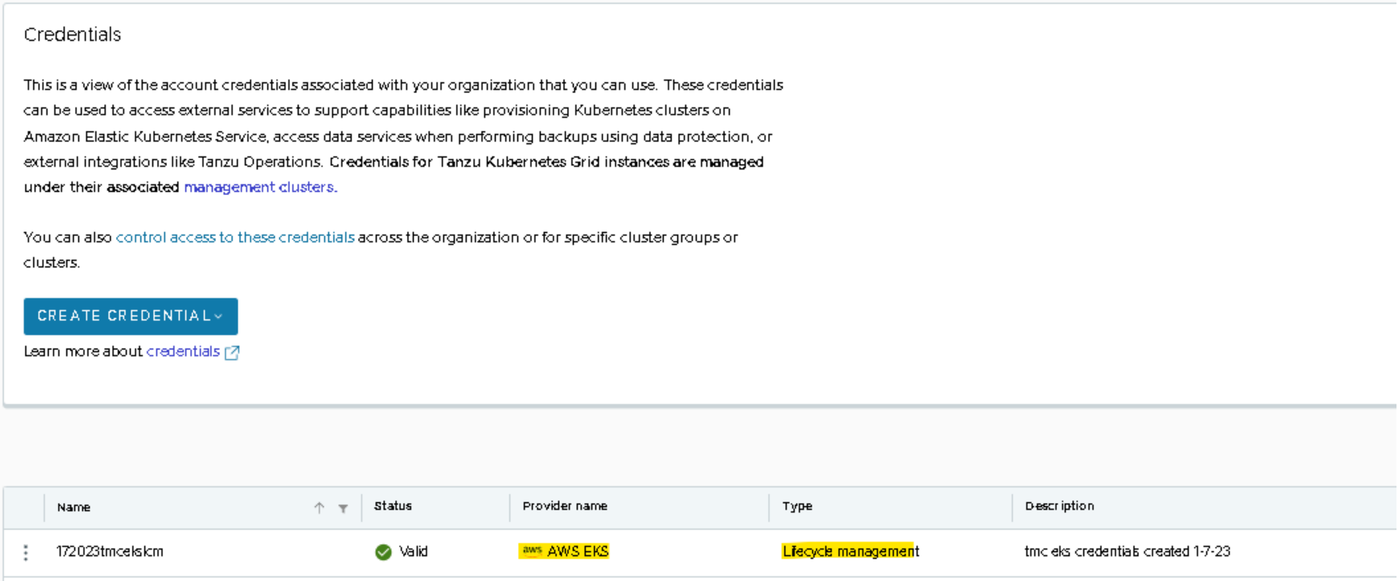
eks-tmc-cloud-vmware-com-10[redacted]

Outputs (1)

Key	Value	Description	Export name
Message	Please copy and paste this role ARN in to the Tanzu Mission Control UI: arn:aws:iam::87[redacted]:role/clusterlifecycle.109[redacted]eks.tmc.cloud.vmware.com	-	-

9. Provide Tanzu Mission Control with the ARN role in the last step. Credentials will be created in a few minutes.

Figure 180: Credentials created.



Credentials

This is a view of the account credentials associated with your organization that you can use. These credentials can be used to access external services to support capabilities like provisioning Kubernetes clusters on Amazon Elastic Kubernetes Service, access data services when performing backups using data protection, or external integrations like Tanzu Operations. Credentials for Tanzu Kubernetes Grid instances are managed under their associated management clusters.

You can also control access to these credentials across the organization or for specific cluster groups or clusters.

Learn more about credentials

Name	Status	Provider name	Type	Description
172023tmceksкm	✓ Valid	aws AWS EKS	Lifecycle management	tmc eks credentials created 1-7-23

Enable data protection for clusters.

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

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

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

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

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

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

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

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

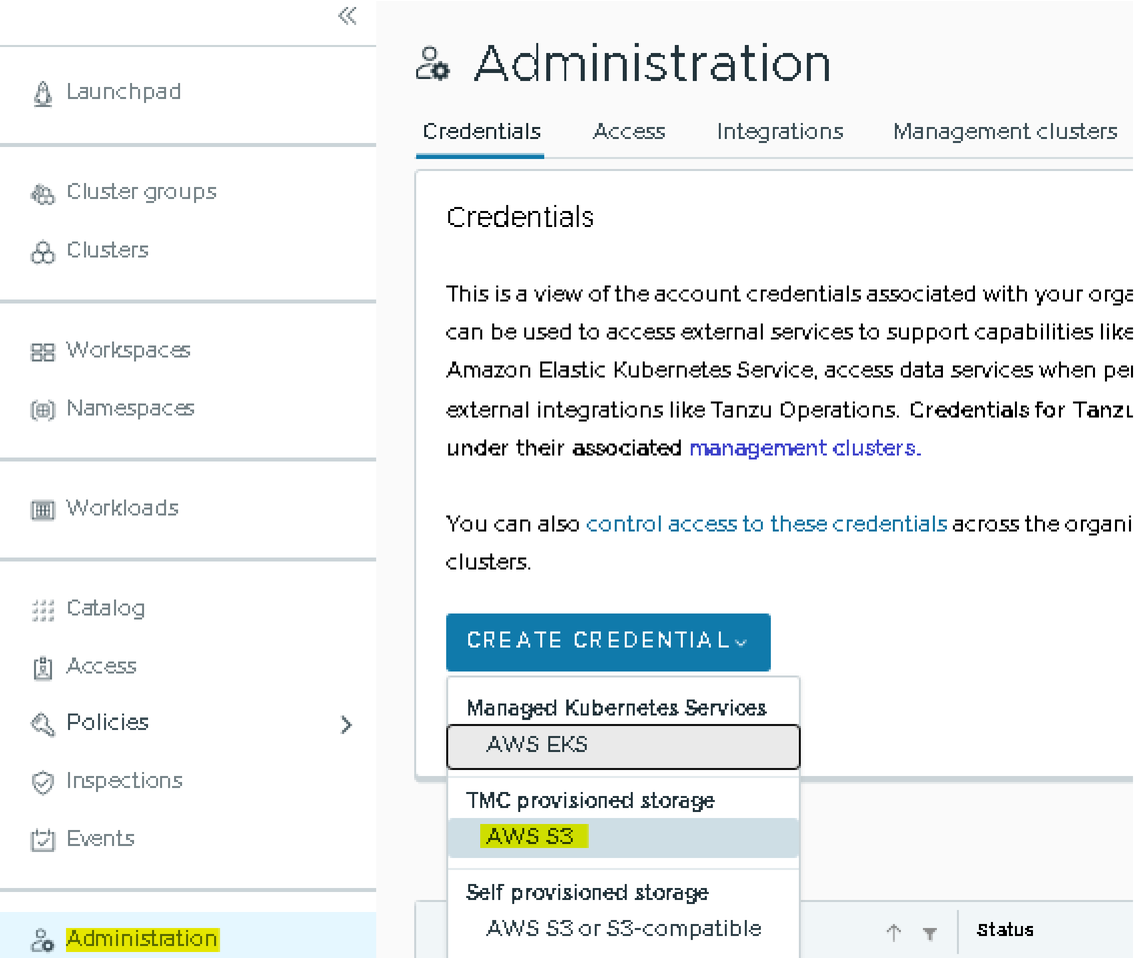
Before you define a backup for a cluster, you must create a target location and credential that you will use to perform the backup.

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

High-level process of creating backup on Amazon S3 follows. Similar process can be followed to backup resources on another S3 compatible storage or Azure Blob storage. The following example uses Amazon S3 storage and assumes that a S3 bucket already exists.

1. Create Amazon AWS credentials.

Figure 181: Create Credentials.



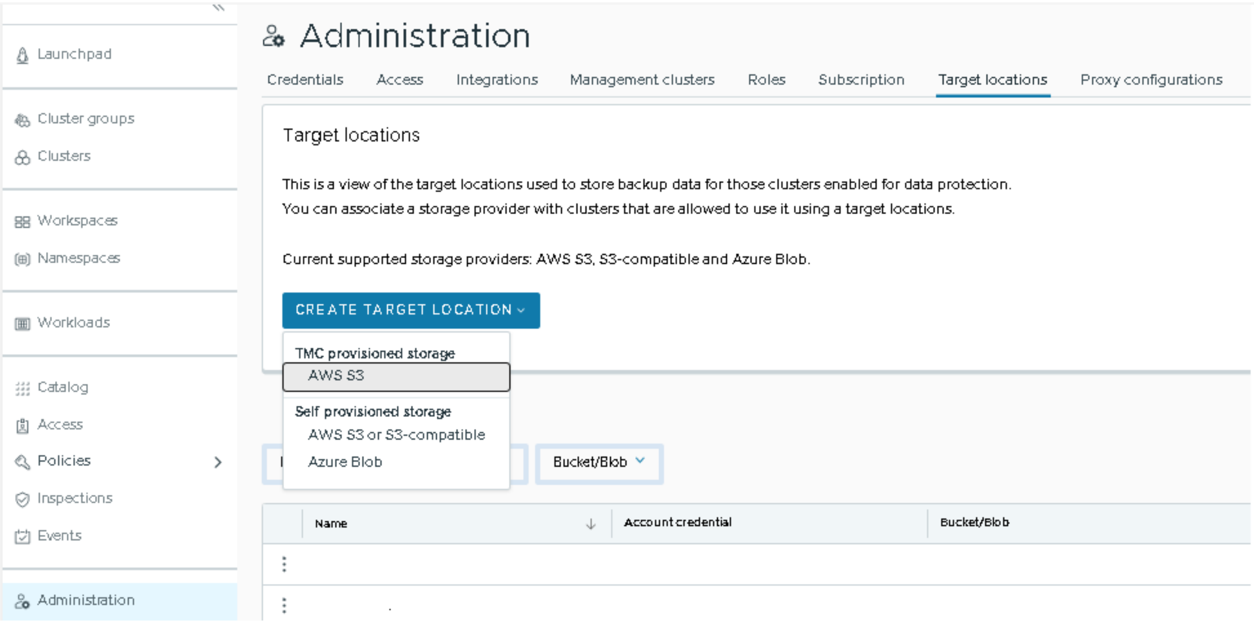
Credentials

This is a view of the account credentials associated with your orga can be used to access external services to support capabilities like Amazon Elastic Kubernetes Service, access data services when pe external integrations like Tanzu Operations. Credentials for Tanzu under their associated management clusters.

You can also control access to these credentials across the organi clusters.

2. In Tanzu Mission Control create a backup target location
   - In the Tanzu Mission Control console, click Administration in the left navigation pane. On the Administration a page, click the Target Locations tab.
   - Click Create Target Location, and then choose the type of storage for the new target location.
   - Select Tanzu Mission Control provisioned storage: AWS S3

Figure 182: Target Location



Administration

Credentials | Access | Integrations | Management clusters | Roles | Subscription | Target locations | Proxy configurations

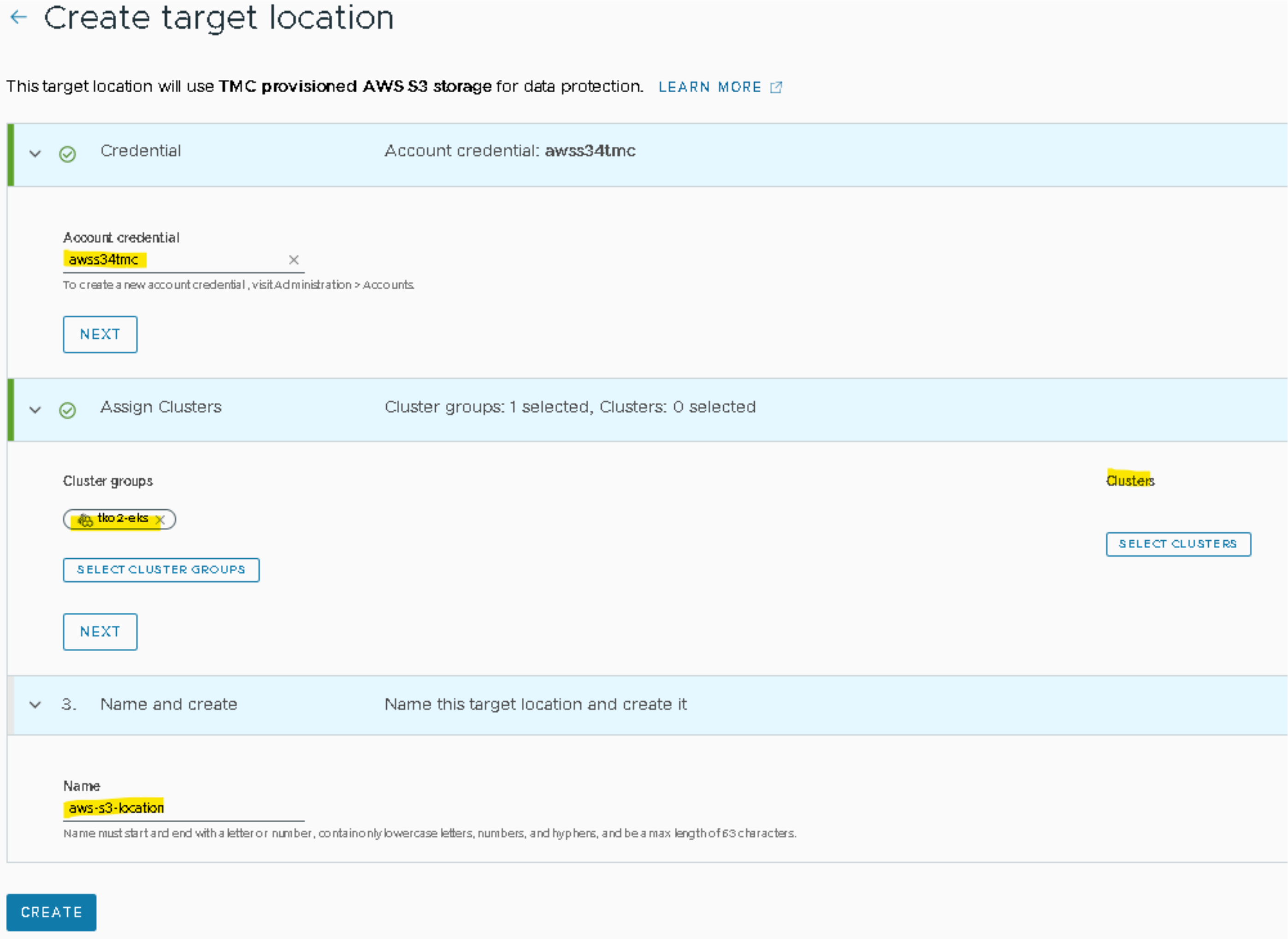
Target locations

This is a view of the target locations used to store backup data for those clusters enabled for data protection.
You can associate a storage provider with clusters that are allowed to use it using a target locations.

Current supported storage providers: AWS S3, S3-compatible and Azure Blob.

3. Fill-in the required information and create backup location.

Figure 183: Create location.



Create target location

This target location will use **TMC provisioned AWS S3 storage** for data protection. LEARN MORE

Credential — Account credential: **awss34tmc**

Account credential: awss34tmc

To create a new account credential, visit Administration > Accounts

Assign Clusters — Cluster groups: 1 selected, Clusters: 0 selected

Cluster groups: tko2-eks

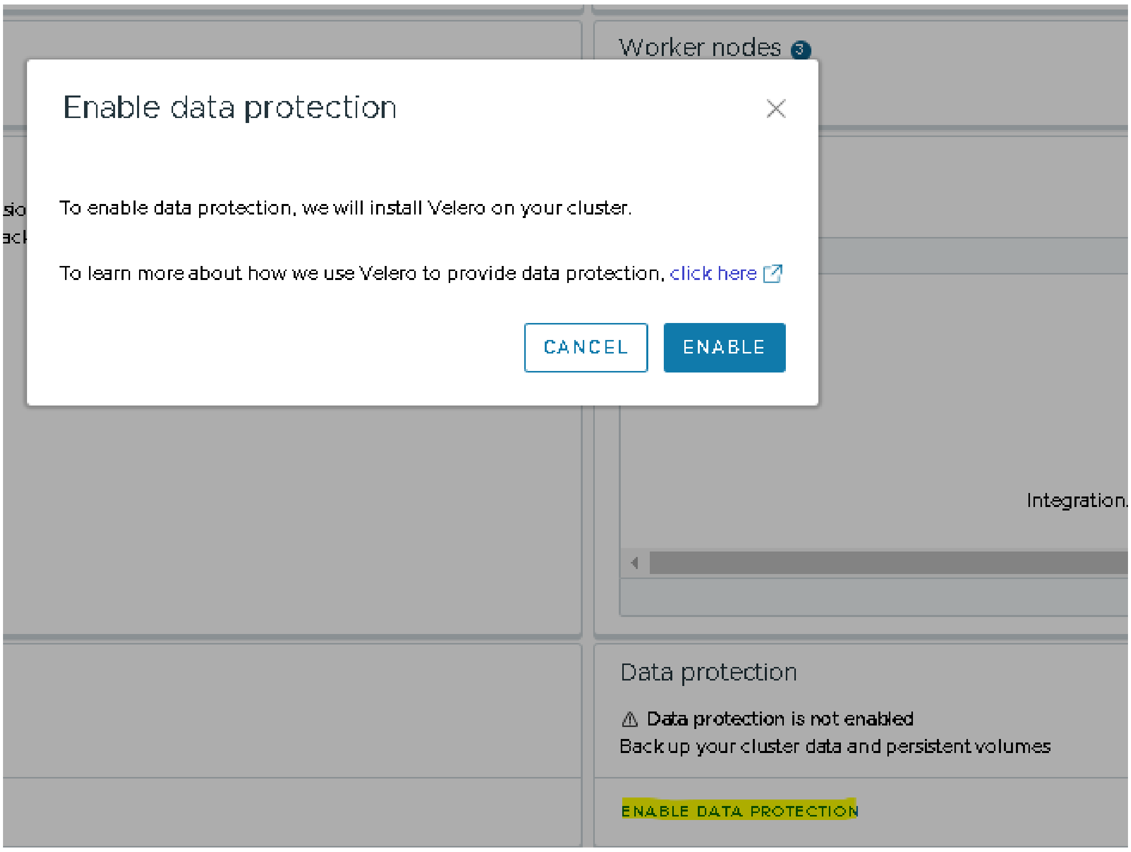
3. Name and create — Name this target location and create it

Name: aws-s3-location

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

4. Enable data protection on the cluster. Creating required pods takes a few minutes to complete.

Figure 184: Enable Data protection on cluster.



Worker nodes 3

Enable data protection

To enable data protection, we will install Velero on your cluster.

To learn more about how we use Velero to provide data protection, click here

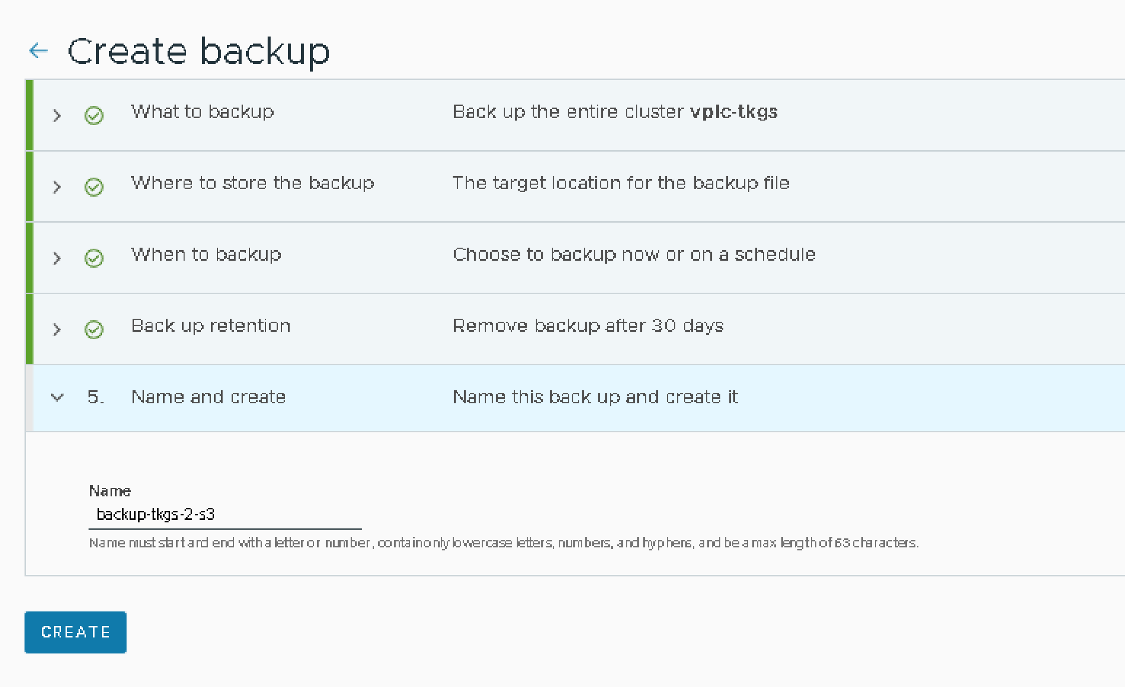
Data protection

⚠ Data protection is not enabled

Back up your cluster data and persistent volumes

5. After data protection is enabled, create or schedule backup using the backup location created earlier.

Figure 185: Create backup.



← Create backup

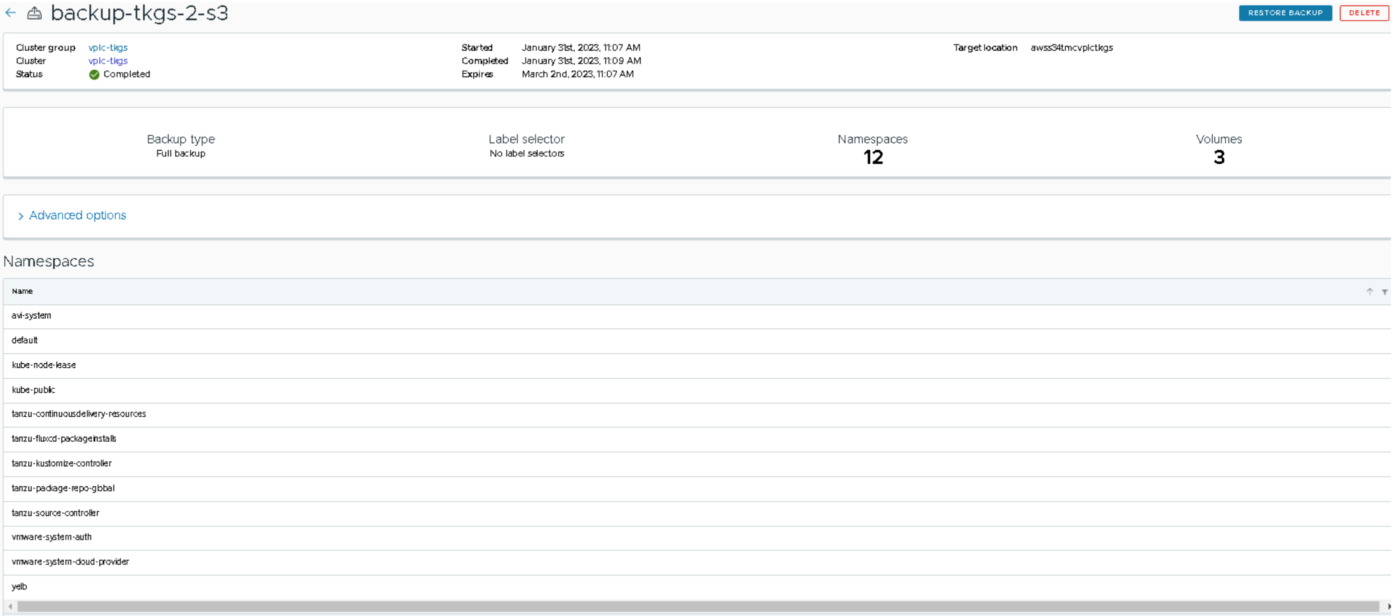
› ✓	What to backup	Back up the entire cluster **vplc-tkgs**
› ✓	Where to store the backup	The target location for the backup file
› ✓	When to backup	Choose to backup now or on a schedule
› ✓	Back up retention	Remove backup after 30 days
⌄ 5.	Name and create	Name this back up and create it

Name: backup-tkgs-2-s3

Name must start and end with a letter or number, contain only lowercase letters, numbers, and hyphens, and be a max length of 63 characters.

Once the backup is processed the status of the backup can be verified in Tanzu Mission Control user interface or viewing the S3 storage bucket.

Figure 186: Backup complete



backup-tkgs-2-s3

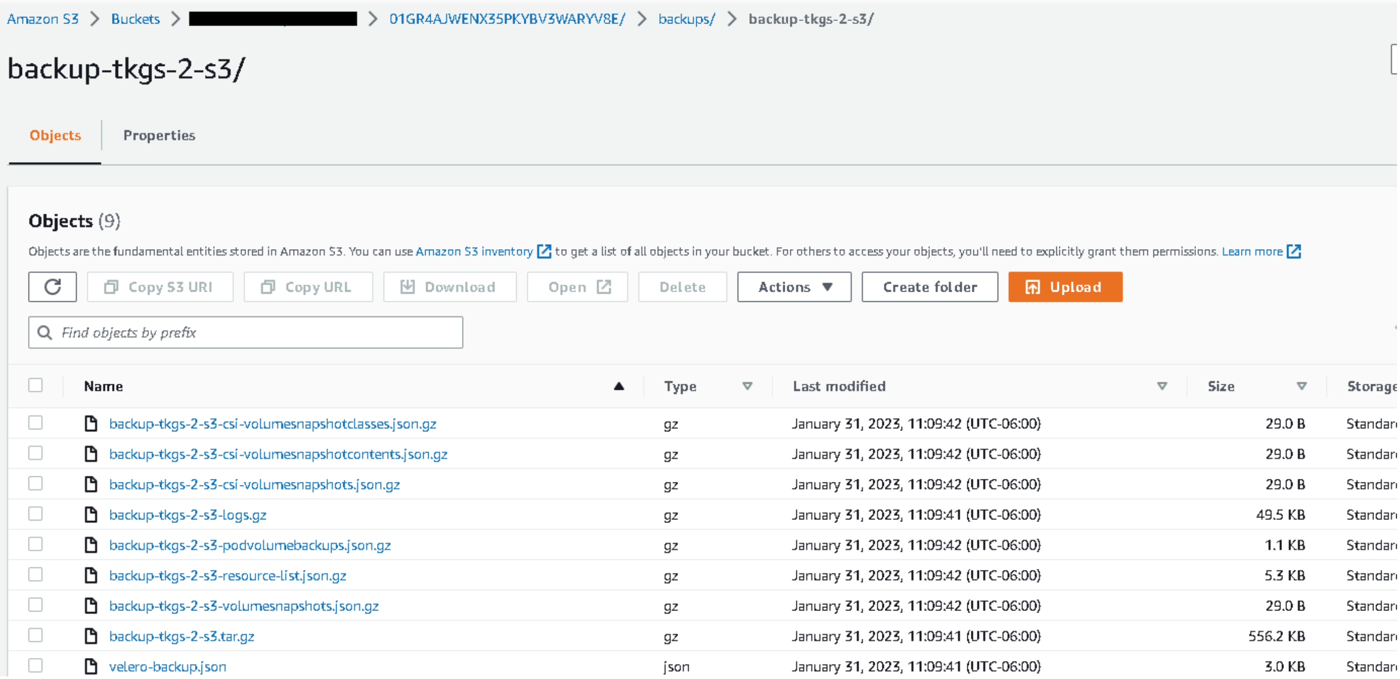
Cluster group	vplc-tkgs	Started	January 31st, 2023, 11:07 AM	Target location	awss34tmcvplctkgs
Cluster	vplc-tkgs	Completed	January 31st, 2023, 11:09 AM		
Status	Completed	Expires	March 2nd, 2023, 11:07 AM		

Backup type	Label selector	Namespaces	Volumes
Full backup	No label selectors	12	3

Namespaces

Name
avi-system
default
kube-node-lease
kube-public
tanzu-continuousdelivery-resources
tanzu-fluxcd-packageinstalls
tanzu-kustomize-controller
tanzu-package-repo-global
tanzu-source-controller
vmware-system-auth
vmware-system-cloud-provider
yelb

Figure 187: Amazon AWS bucket



Amazon S3 > Buckets > [redacted] > 01GR4AJWENX35PKYBV3WARYV8E/ > backups/ > backup-tkgs-2-s3/

backup-tkgs-2-s3/

Objects (9)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage
□	backup-tkgs-2-s3-csi-volumesnapshotclasses.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	29.0 B	Standar
□	backup-tkgs-2-s3-csi-volumesnapshotcontents.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	29.0 B	Standar
□	backup-tkgs-2-s3-csi-volumesnapshots.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	29.0 B	Standar
□	backup-tkgs-2-s3-logs.gz	gz	January 31, 2023, 11:09:41 (UTC-06:00)	49.5 KB	Standar
□	backup-tkgs-2-s3-podvolumebackups.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	1.1 KB	Standar
□	backup-tkgs-2-s3-resource-list.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	5.3 KB	Standar
□	backup-tkgs-2-s3-volumesnapshots.json.gz	gz	January 31, 2023, 11:09:42 (UTC-06:00)	29.0 B	Standar
□	backup-tkgs-2-s3.tar.gz	gz	January 31, 2023, 11:09:41 (UTC-06:00)	556.2 KB	Standar
□	velero-backup.json	json	January 31, 2023, 11:09:41 (UTC-06:00)	3.0 KB	Standar

Cluster Inspections

Tanzu Mission Control Advanced edition provides preconfigured cluster inspections using Sonobuoy, an open-source community standard.

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

The Conformance inspection validates the binaries running on your cluster and ensures that your cluster is properly installed, configured, and working. Reports can be generated from within Tanzu Mission Control to assess and address any issues that arise. For more information, see the Kubernetes Conformance documentation at Kubernetes conformance.

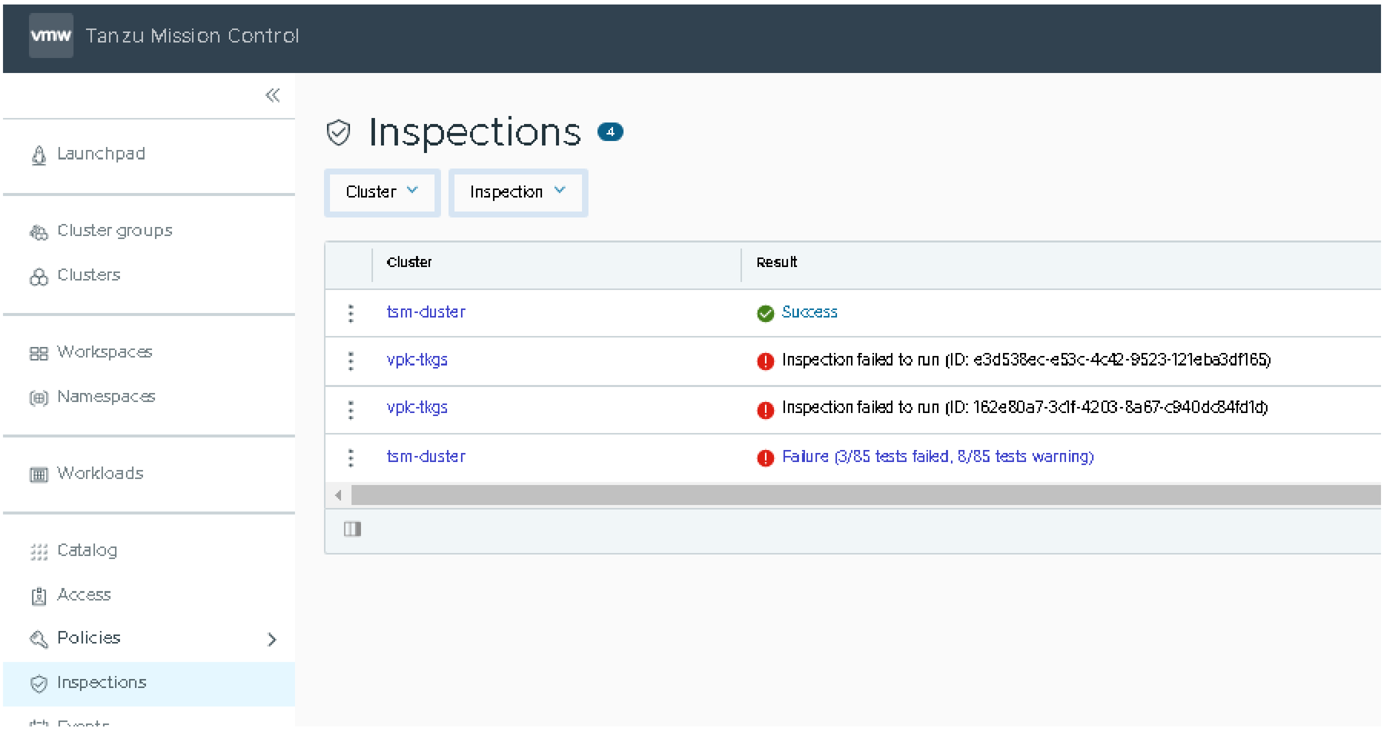
The CIS benchmark inspection evaluates your cluster against the CIS Benchmark for Kubernetes published by the Center for Internet Security. This inspection type is available in the advanced version of Tanzu Mission Control.

The Lite inspection is a node conformance test that validates whether nodes meet requirements for Kubernetes. For more information, see Validate node setup in the Kubernetes documentation.

Because the cluster inspections provide a point-in-time report of the condition of the cluster, run them periodically (to avoid drifting out of conformance) and any time significant alterations are made, such as after patching or upgrading a cluster.

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

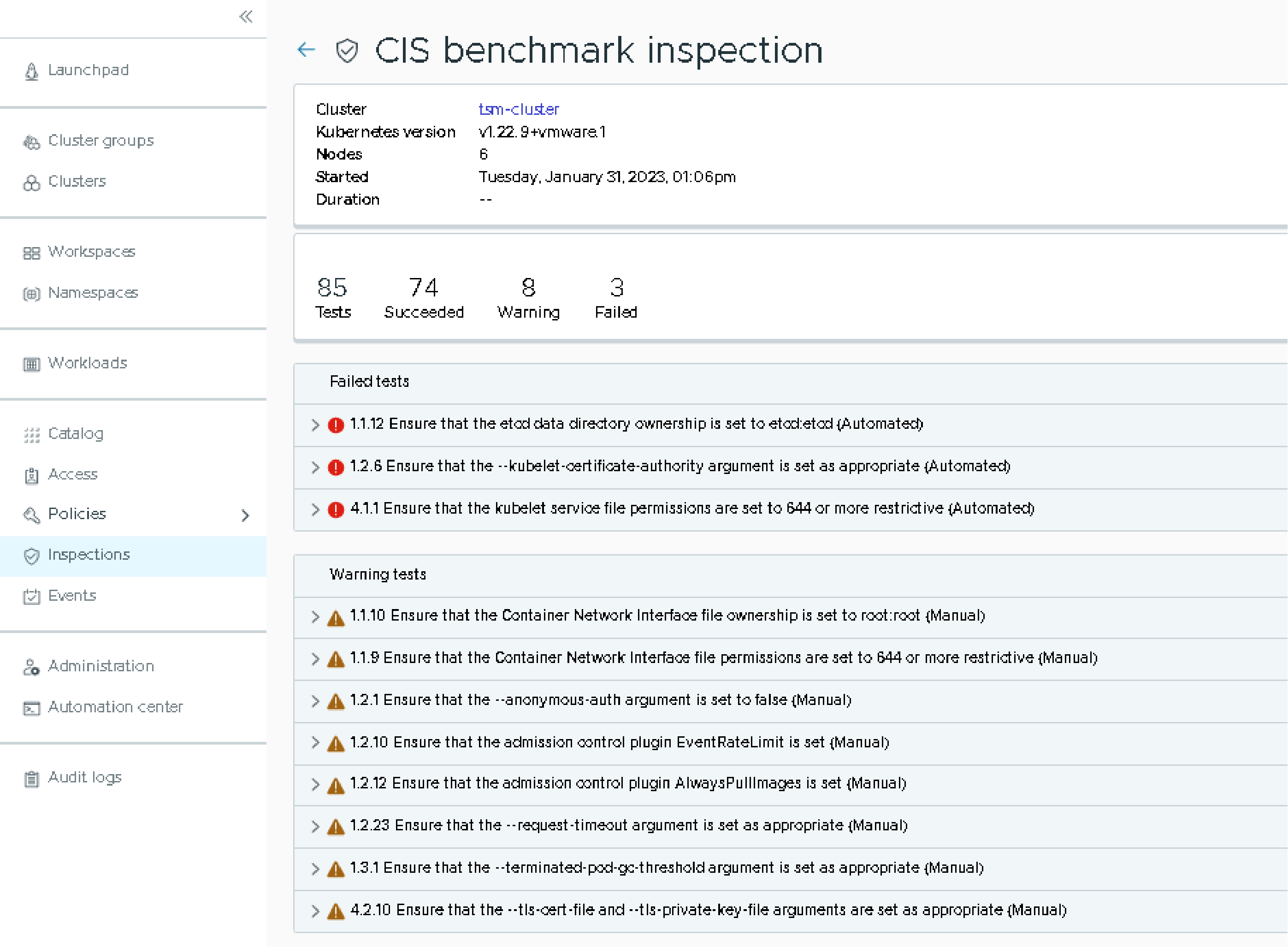
Figure 188: Inspections



Cluster	Result
tsm-duster	Success
vpk-tkgs	Inspection failed to run (ID: e3d538ec-e53c-4c42-9523-121eba3df165)
vpk-tkgs	Inspection failed to run (ID: 162e80a7-3d1f-4203-8a67-c940dc84fd1d)
tsm-duster	Failure (3/85 tests failed, 8/85 tests warning)

A list of inspections tests and results can then be viewed at the cluster level for details of the test runs.

Figure 189: Cluster inspection details.



CIS benchmark inspection

Cluster	tsm-cluster
Kubernetes version	v1.22.9+vmware.1
Nodes	6
Started	Tuesday, January 31, 2023, 01:06pm
Duration	--

85	74	8	3
Tests	Succeeded	Warning	Failed

Failed tests

- 1.1.12 Ensure that the etcd data directory ownership is set to etcd:etcd (Automated)
- 1.2.6 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Automated)
- 4.1.1 Ensure that the kubelet service file permissions are set to 644 or more restrictive (Automated)

Warning tests

- 1.1.10 Ensure that the Container Network Interface file ownership is set to root:root (Manual)
- 1.1.9 Ensure that the Container Network Interface file permissions are set to 644 or more restrictive (Manual)
- 1.2.1 Ensure that the --anonymous-auth argument is set to false (Manual)
- 1.2.10 Ensure that the admission control plugin EventRateLimit is set (Manual)
- 1.2.12 Ensure that the admission control plugin AlwaysPullImages is set (Manual)
- 1.2.23 Ensure that the --request-timeout argument is set as appropriate (Manual)
- 1.3.1 Ensure that the --terminated-pod-gc-threshold argument is set as appropriate (Manual)
- 4.2.10 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Manual)

Cloud Native Applications

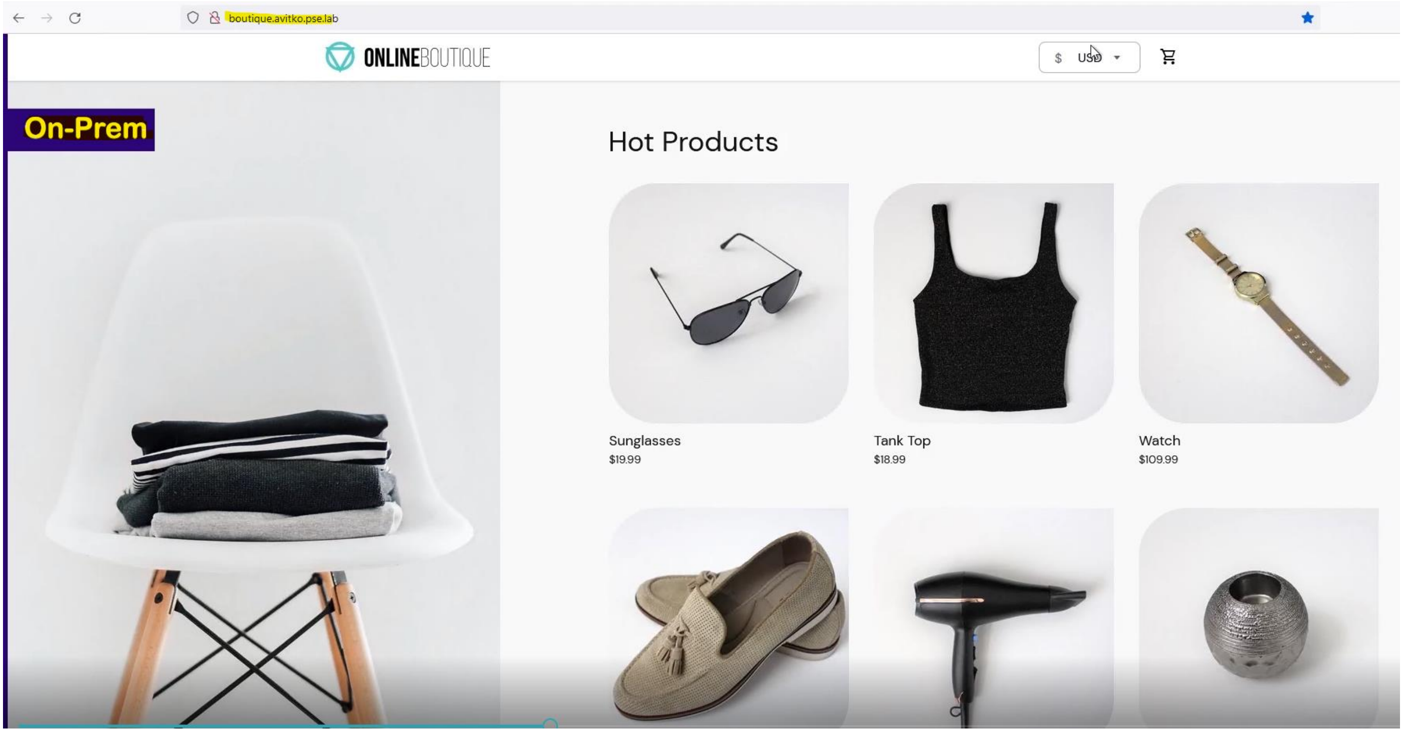
Cloud native is an approach to building and running applications that exploits the advantages of the cloud computing delivery model. When companies build and operate applications using a cloud native architecture, they bring new ideas to market faster and respond sooner to customer demands. While public cloud has affected the thinking about infrastructure investment in virtually every industry, cloud-like delivery is not exclusive to public environments. Cloud native development is appropriate for both public and private clouds; it is about how applications are created and deployed, not where.

To assess the workings of this multi-cloud reference architecture, a cloud-native application (Online Boutique) developed by Google was installed and scenarios depicted in figure 4 and figure 5 were tested. Online Boutique is a cloud-native microservices demo application. Online Boutique consists of an 11-tier microservices application. The application is a web-based e-commerce app where users can browse items, add them to the cart, and purchase them. Below is an example of a scenario that was evaluated based on Avi GSLB functionality discussed in this document.

Scenario: Application unavailability

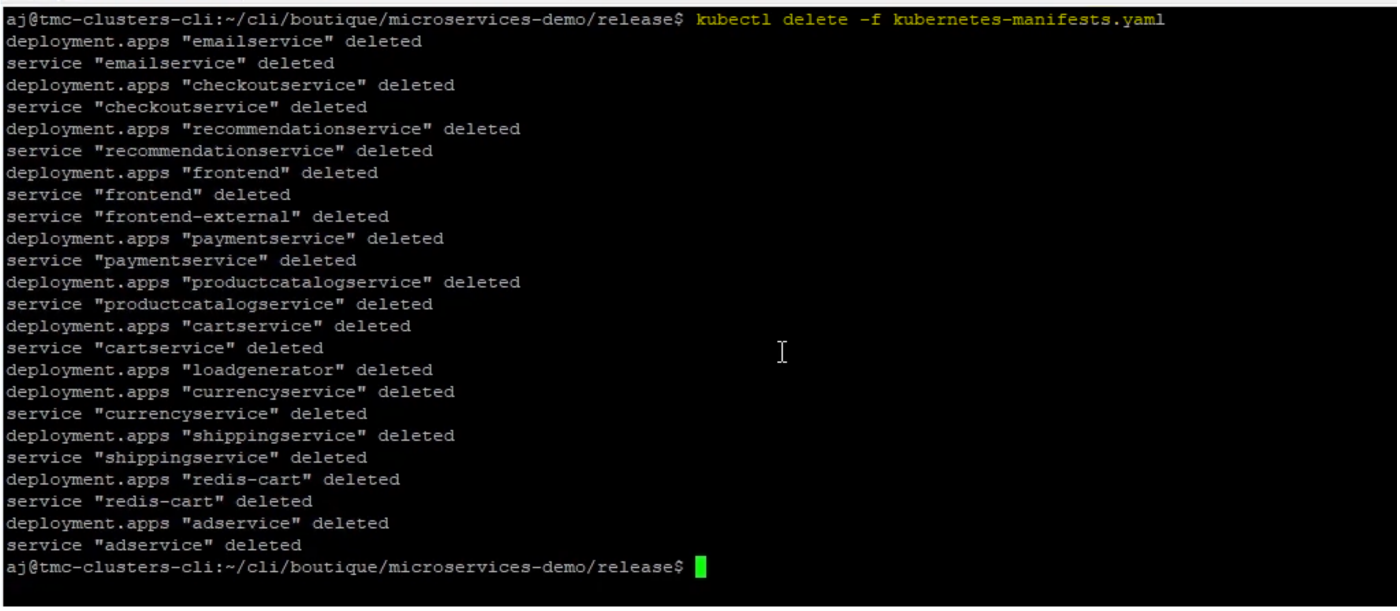
With the architecture represented in the reference architecture, the application is installed on multiple sites and can be accessed via a single URL “boutique.avitko.pse.lab.” Which site the user connects to, depends on the load balancing algorithm selected. With Geo location option, the users will be directed to the nearest instance of the application, improving performance and user experience. In the example below, the application instances reside on on-premises private cloud and Amazon AWS EKS clusters. When a user initially connects to the application, he/she are directed to the on-premises instance of the application since this is the closest Geo location to the user. This is indicated by a label “On-Prem” on the application frontend user interface.

Figure 190: User connection



To simulate application unavailability or site failure, the application is uninstalled.

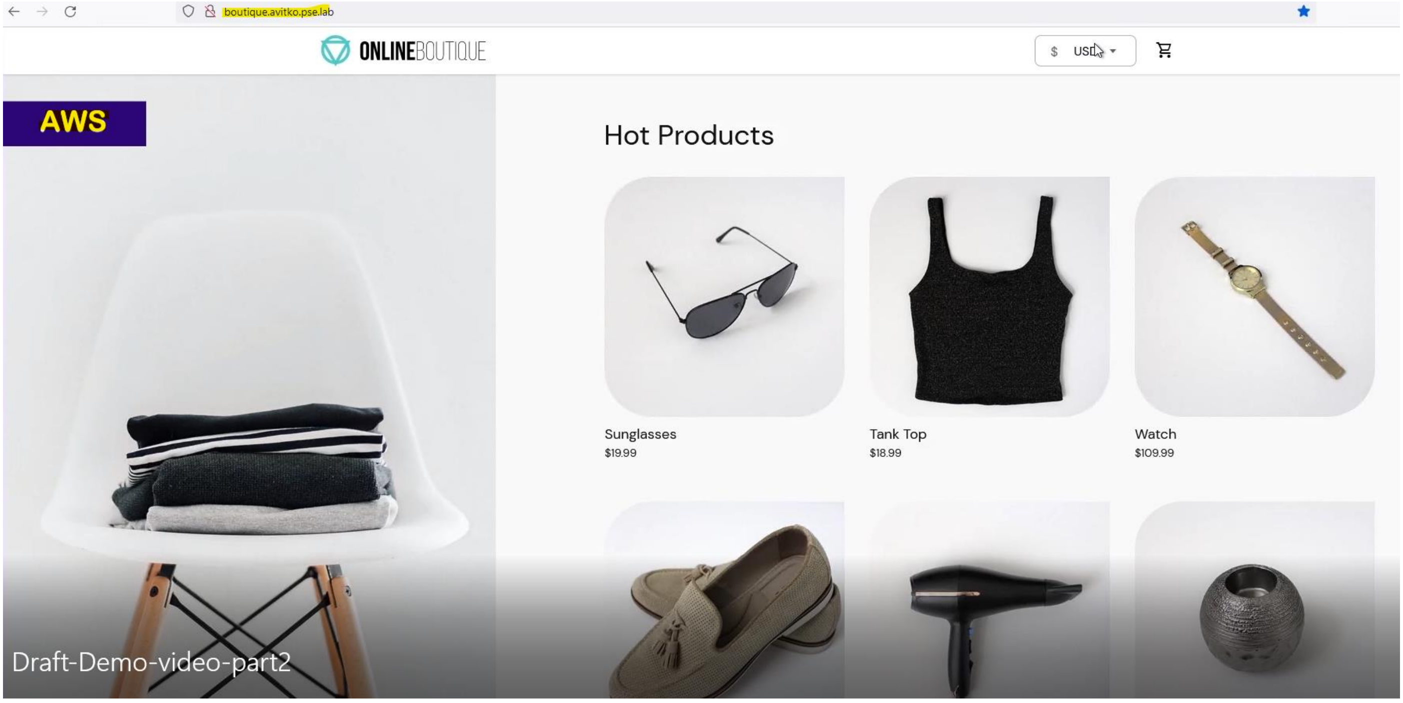
Figure 191: Application unavailable.



```
aj@tmc-clusters-cli:~/cli/boutique/microservices-demo/release$ kubectl delete -f kubernetes-manifests.yaml
deployment.apps "emailservice" deleted
service "emailservice" deleted
deployment.apps "checkoutservice" deleted
service "checkoutservice" deleted
deployment.apps "recommendationservice" deleted
service "recommendationservice" deleted
deployment.apps "frontend" deleted
service "frontend" deleted
service "frontend-external" deleted
deployment.apps "paymentservice" deleted
service "paymentservice" deleted
deployment.apps "productcatalogservice" deleted
service "productcatalogservice" deleted
deployment.apps "cartservice" deleted
service "cartservice" deleted
deployment.apps "loadgenerator" deleted
deployment.apps "currencyservice" deleted
service "currencyservice" deleted
deployment.apps "shippingservice" deleted
service "shippingservice" deleted
deployment.apps "redis-cart" deleted
service "redis-cart" deleted
deployment.apps "adservice" deleted
service "adservice" deleted
aj@tmc-clusters-cli:~/cli/boutique/microservices-demo/release$
```

At this point connectivity is lost to the application. When the user’s browser or client application retirees the connection, the user is redirected to the “AWS” site since the “On-Prem” instance is unavailable indicated by the “AWS” label on the frontend user interface.

Figure 192: Application available



Distributed Microservices via Tanzu Service Mesh

VMware Tanzu Service Mesh is an enterprise-class service mesh solution that provides reliable control and security for microservices, end users, and data across all your clusters and clouds in the most demanding multi-cluster and multi-cloud environments.

To control application traffic, Tanzu Service Mesh provides fine-grained, traffic management policies that give you complete control and visibility into how traffic and API calls flow between your services and across clusters and clouds. To secure communication between services and protect sensitive data, you can use Tanzu Service Mesh to implement a zero-trust security model for cloud-based applications.

Note: Zero-Trust security, PII Data/user security and API security is only available with Enterprise Edition of Tanzu Service Mesh.

You can measure application performance with a configurable service level objective (SLO) definition. For more information, see the Service Level Objectives with Tanzu Service Mesh documentation. As application demands change, you can auto-scale services to maintain SLOs using Tanzu Service Mesh Service Auto-scaler. For more information, see the Service Autoscaling in Tanzu Service Mesh documentation .

Tanzu Service Mesh supports cross-cluster and cross-cloud use cases with global namespaces. With global namespaces, you can securely deploy applications across clusters and clouds and have consistent traffic management policies, application continuity, and security policies across cloud silos and boundaries, regardless of where the applications are running. Global namespaces can be each considered to mark an application boundary and as such provide strongly isolated environments for application teams and business units managing different applications and data.

What follows is a discussion of the major configuration areas of Tanzu Service Mesh as they relate to this reference architecture. The flow of configuration is as follows.

- Onboard Clusters
- Integrate NSX Advanced Load Balancer with Tanzu Service Mesh
- Add DNS and Domains
- Install Applications
- Create Global Namespace for applications.
- Create public service.
- Service Autoscaling
- Service Level Policies

Onboard Clusters

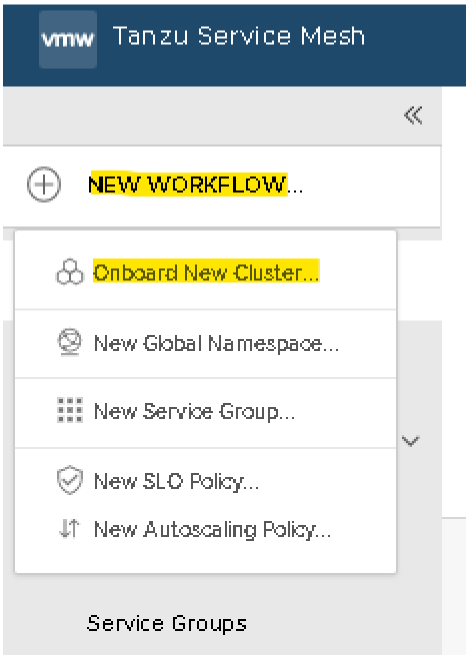
The first step is to onboard clusters that need to be part of Tanzu Service Mesh. This onboarding can be done either through Tanzu Service Mesh or if the cluster is already part of Tanzu Mission Control, the cluster can directly be onboarded from Tanzu Mission Control user interface.

If a proxy server is configured in your corporate environment, when onboarding your cluster, specify that it will connect to Tanzu Service Mesh through a proxy server. All traffic between the cluster and Tanzu Service Mesh will be routed through the proxy server and will be encrypted using Transport Layer Security (TLS). All requests that are sent from the cluster to Tanzu Service Mesh will be authorized using access tokens.

If the Avi controller is behind a private network and cannot be reached directly by the Tanzu Service Mesh global controller, a proxy connection is needed through one of the Kubernetes clusters available on Tanzu Service Mesh. A WebSockets proxy is implemented on all the client Kubernetes clusters onboarded into Tanzu Service Mesh for this purpose. In this way, Tanzu Service Mesh can connect to the Avi controller through the client cluster, which should have connectivity to the Avi controller as well. A **cluster label** must be assigned to the cluster to use the proxy.

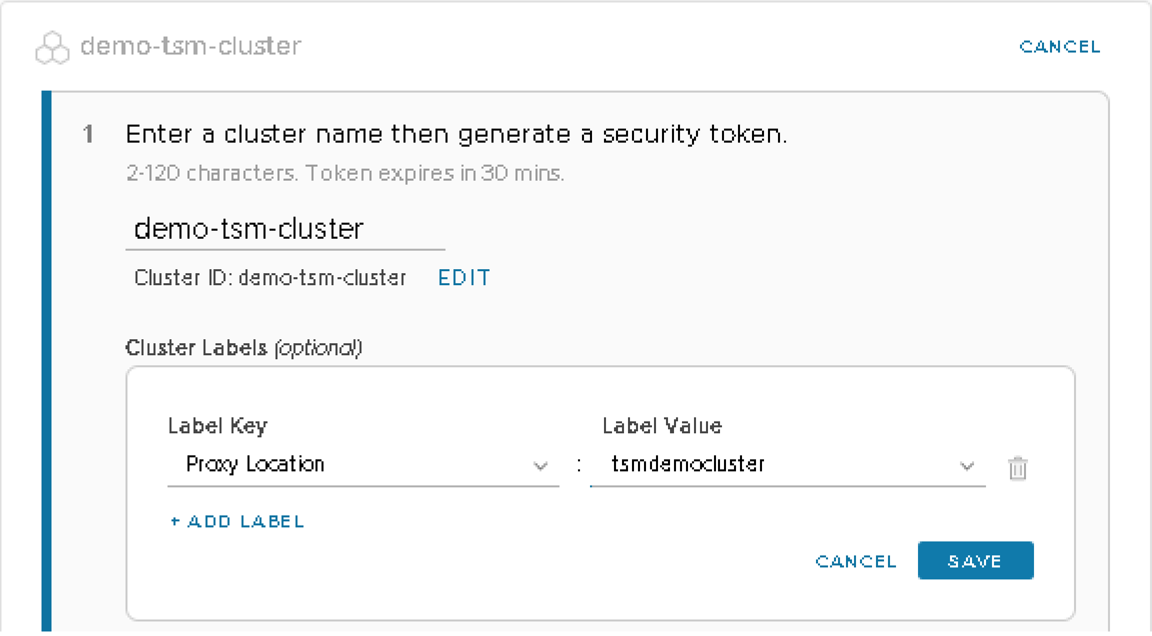
1. To onboard a cluster from Tanzu Service Mesh user interface Click “New Workflow” and select “On-board Clusters

Figure 193: Onboard Cluster



2. Give the cluster a name and create a “proxy location” cluster label and save.

Figure 194: Onboard cluster



Onboard Clusters

demo-tsm-cluster

1 Enter a cluster name then generate a security token.

2-120 characters. Token expires in 30 mins.

demo-tsm-cluster

Cluster ID: demo-tsm-cluster

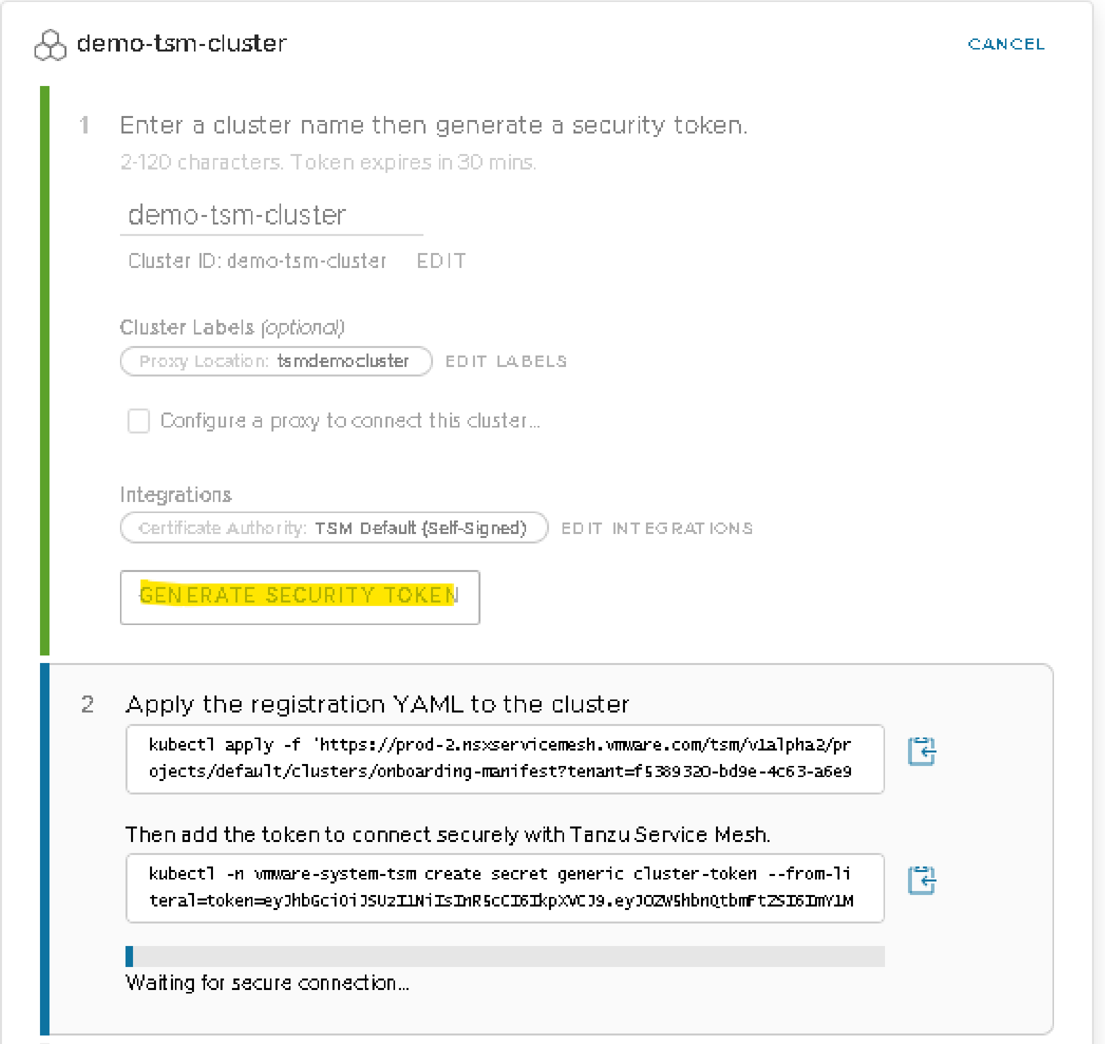
Cluster Labels *(optional)*

Label Key: Proxy Location

Label Value: tsmdemocluster

3. Click “Generate Security Token and copy the two commands to be run on the cluster.

Figure 195: Generate Tokens



demo-tsm-cluster

CANCEL

1 Enter a cluster name then generate a security token.

2-120 characters. Token expires in 30 mins.

demo-tsm-cluster

Cluster ID: demo-tsm-cluster EDIT

Cluster Labels *(optional)*

Proxy Location: tsmdemocluster EDIT LABELS

□ Configure a proxy to connect this cluster...

Integrations

Certificate Authority: TSM Default (Self-Signed) EDIT INTEGRATIONS

GENERATE SECURITY TOKEN

2 Apply the registration YAML to the cluster

```
kubectl apply -f 'https://prod-2.nsxservicemesh.vmware.com/tsm/v1alpha2/pr
ojects/default/clusters/onboarding-manifest?tenant=f5389320-bd9e-4c63-a6e9
```

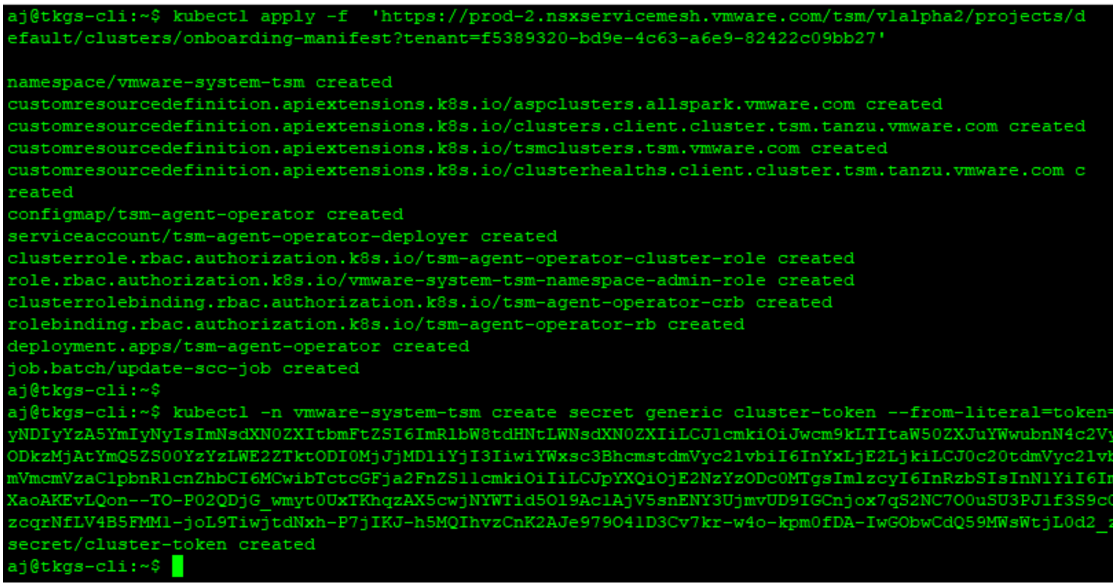
Then add the token to connect securely with Tanzu Service Mesh.

```
kubectl -n vmware-system-tsm create secret generic cluster-token --from-li
teral=token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJOZW5hbnQtbmFtZSI6ImY1M
```

Waiting for secure connection...

4. Login to the cluster and run the two copied commands.

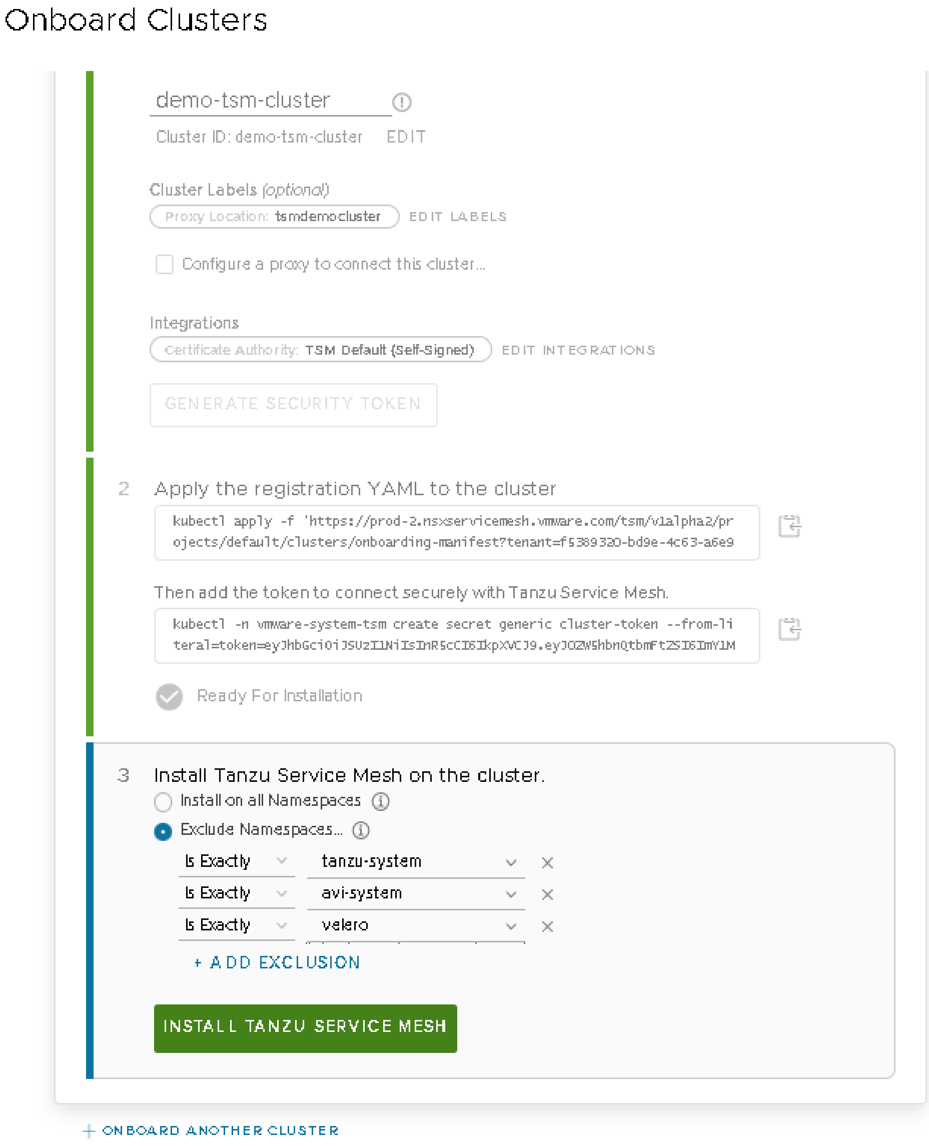
Figure 196: Run onboarding commands.



```
aj@tkgs-cli:~$ kubectl apply -f  'https://prod-2.nsxservicemesh.vmware.com/tsm/v1alpha2/projects/d
efault/clusters/onboarding-manifest?tenant=f5389320-bd9e-4c63-a6e9-82422c09bb27'

namespace/vmware-system-tsm created
customresourcedefinition.apiextensions.k8s.io/aspclusters.allspark.vmware.com created
customresourcedefinition.apiextensions.k8s.io/clusters.client.cluster.tsm.tanzu.vmware.com created
customresourcedefinition.apiextensions.k8s.io/tsmclusters.tsm.vmware.com created
customresourcedefinition.apiextensions.k8s.io/clusterhealths.client.cluster.tsm.tanzu.vmware.com c
reated
configmap/tsm-agent-operator created
serviceaccount/tsm-agent-operator-deployer created
clusterrole.rbac.authorization.k8s.io/tsm-agent-operator-cluster-role created
role.rbac.authorization.k8s.io/vmware-system-tsm-namespace-admin-role created
clusterrolebinding.rbac.authorization.k8s.io/tsm-agent-operator-crb created
rolebinding.rbac.authorization.k8s.io/tsm-agent-operator-rb created
deployment.apps/tsm-agent-operator created
job.batch/update-scc-job created
aj@tkgs-cli:~$
aj@tkgs-cli:~$ kubectl -n vmware-system-tsm create secret generic cluster-token --from-literal=token=
yNDIyYzA5YmIyNyIsImNsdXN0ZXItbmFtZSI6ImRlbW8tdHNtLWNsdXN0ZXIiLCJlcmkiOiJwcm9kLTItaW50ZXJuYWwubnN4c2Vy
ODkzMjAtYmQ5ZS00YzYzLWE2ZTktODI0MjJjMDliYjI3IiwiYWxsc3BhcmstdmVyc2lvbiI6InYxLjE2LjkiLCJ0c20tdmVyc2lvb
mVmcmVzaClpbnRlcnZhbCI6MCwibTctcGFja2FnZS1lcmkiOiIiLCJpYXQiOjE2NzYzODc0MTgsImlzcyI6InRzbSIsInN1YiI6In
XaoAKEvLQon--TO-P02QDjG_wmyt0UxTKhqzAX5cwjNYWTid5O19AclAjV5snENY3UjmvUD9IGCnjox7qS2NC7O0uSU3PJ1f3S9c0
zcqrNfLV4B5FMMl-joL9TiwjtdNxh-P7jIKJ-h5MQIhvzCnK2AJe979041D3Cv7kr-w4o-kpm0fDA-IwGObwCdQ59MWsWtjL0d2_z
secret/cluster-token created
aj@tkgs-cli:~$
```

5. Choose to install Tanzu Service Mesh on all namespaces or select namespaces that should be excluded. Click “Install Tanzu Service Mesh.”

Figure 197: Install Tanzu Service Mesh



Onboard Clusters

demo-tsm-cluster

Cluster ID: demo-tsm-cluster EDIT

Cluster Labels *(optional)*

Proxy Location: tsmdemocluster EDIT LABELS

□ Configure a proxy to connect this cluster...

Integrations

Certificate Authority: TSM Default (Self-Signed) EDIT INTEGRATIONS

GENERATE SECURITY TOKEN

2 Apply the registration YAML to the cluster

```
kubectl apply -f 'https://prod-2.nsxservicemesh.vmware.com/tsm/v1alpha2/pr
ojects/default/clusters/onboarding-manifest?tenant=f5389320-bd9e-4c63-a6e9
```

Then add the token to connect securely with Tanzu Service Mesh.

```
kubectl -n vmware-system-tsm create secret generic cluster-token --from-li
teral=token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJOZW5hbnQtbmFtZSI6ImY1M
```

✓ Ready For Installation

3 Install Tanzu Service Mesh on the cluster.

○ Install on all Namespaces

● Exclude Namespaces...

Is Exactly	tanzu-system
Is Exactly	avi-system
Is Exactly	velero

+ ADD EXCLUSION

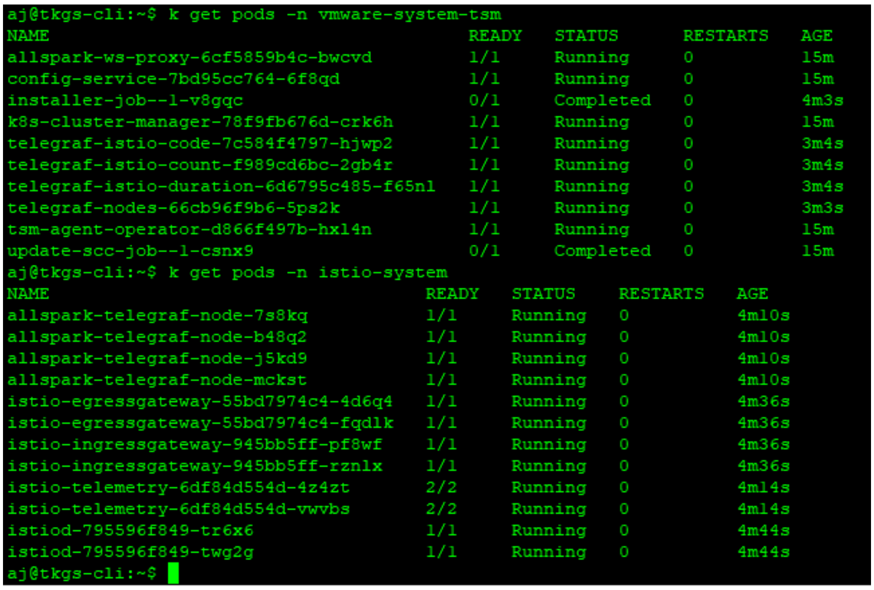
INSTALL TANZU SERVICE MESH

+ ONBOARD ANOTHER CLUSTER

Note: The system namespaces on the cluster, such as kube-system, kube-public, and istio-system, are excluded from Tanzu Service Mesh by default.

6. Verify that all pods under namespaces *“vmware-system-tsm”* and *“istio-system”* are running.

Figure 198: Verify installation.



```
aj@tkgs-cli:~$ k get pods -n vmware-system-tsm
NAME                                         READY   STATUS      RESTARTS   AGE
allspark-ws-proxy-6cf5859b4c-bwcvd           1/1     Running     0          15m
config-service-7bd95cc764-6f8qd              1/1     Running     0          15m
installer-job--1-v8gqc                       0/1     Completed   0          4m3s
k8s-cluster-manager-78f9fb676d-crk6h         1/1     Running     0          15m
telegraf-istio-code-7c584f4797-hjwp2         1/1     Running     0          3m4s
telegraf-istio-count-f989cd6bc-2gb4r         1/1     Running     0          3m4s
telegraf-istio-duration-6d6795c485-f65nl     1/1     Running     0          3m4s
telegraf-nodes-66cb96f9b6-5ps2k              1/1     Running     0          3m3s
tsm-agent-operator-d866f497b-hxl4n           1/1     Running     0          15m
update-scc-job--1-csnx9                      0/1     Completed   0          15m
aj@tkgs-cli:~$ k get pods -n istio-system
NAME                                     READY   STATUS    RESTARTS   AGE
allspark-telegraf-node-7s8kq             1/1     Running   0          4m10s
allspark-telegraf-node-b48q2             1/1     Running   0          4m10s
allspark-telegraf-node-j5kd9             1/1     Running   0          4m10s
allspark-telegraf-node-mckst             1/1     Running   0          4m10s
istio-egressgateway-55bd7974c4-4d6q4     1/1     Running   0          4m36s
istio-egressgateway-55bd7974c4-fqdlk     1/1     Running   0          4m36s
istio-ingressgateway-945bb5ff-pf8wf      1/1     Running   0          4m36s
istio-ingressgateway-945bb5ff-rznlx      1/1     Running   0          4m36s
istio-telemetry-6df84d554d-4z4zt         2/2     Running   0          4m14s
istio-telemetry-6df84d554d-vwvbs         2/2     Running   0          4m14s
istiod-795596f849-tr6x6                  1/1     Running   0          4m44s
istiod-795596f849-twg2g                  1/1     Running   0          4m44s
aj@tkgs-cli:~$
```

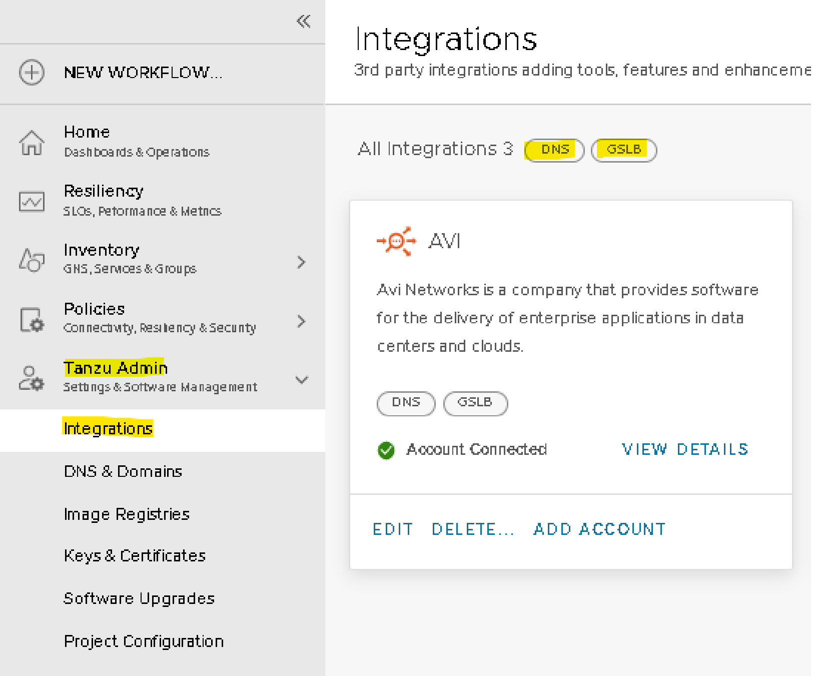
Integrate NSX Advanced Load Balancer with Tanzu Service Mesh

In this reference architecture, NSX Advanced Load Balancer plays a central role in connecting sites, clusters and services and providing load balancing and ingress. For Tanzu Service Mesh to provide central management and GSLB, NSX Advanced Load Balancer leader controller needs to be integrated with Tanzu Service Mesh. Applications are exposed to users through a public service configured in a global namespace. NSX Advanced Load Balancer routes user requests to optimal application instances by using the global load balancing configuration specified for the public service. Following procedure outlines the integration process.

1. From Tanzu Service Mesh left navigation pane click Tanzu Admin > Integration

2. On the **Integrations** page, under **All Integrations**, find the Avi card with the **DNS** and **GSLB** labels.

Figure 199: Integration



Integrations

3rd party integrations adding tools, features and enhanceme

All Integrations 3

AVI

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

Account Connected

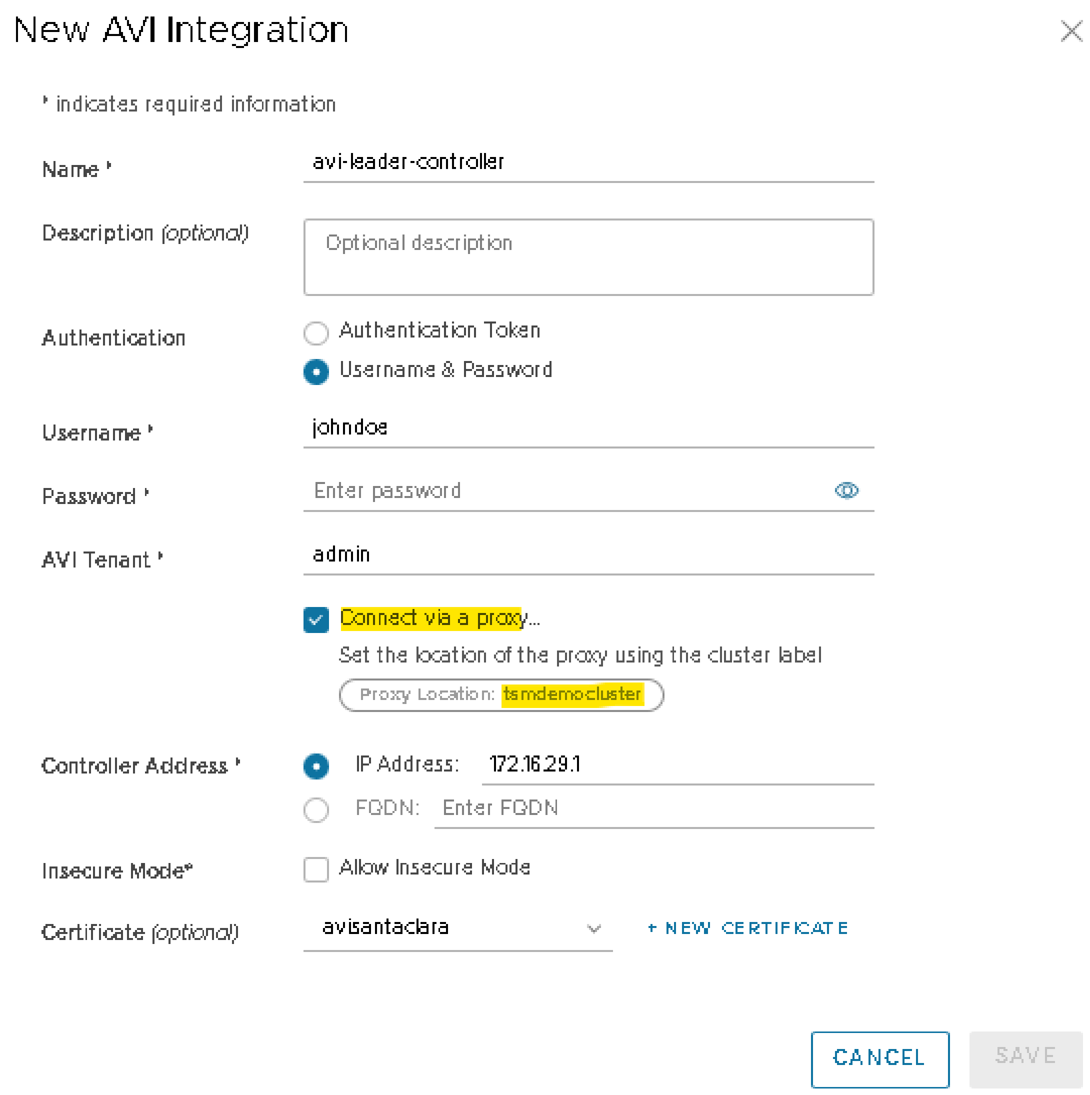
3. Select one of the following options.

   a. If you are creating the first Avi integration account, at the bottom of the card, click **Configure**.

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

4. Enter required information for the environment including the proxy location created while onboarding the cluster.

Figure 200: NSX Advanced Load Balancer Integration



New AVI Integration

* indicates required information

Name * avi-leader-controller

Description *(optional)* Optional description

Authentication ○ Authentication Token ● Username & Password

Username * johndoe

Password * Enter password

AVI Tenant * admin

☑ Connect via a proxy...
Set the location of the proxy using the cluster label
Proxy Location: tsmdemocluster

Controller Address * ● IP Address: 172.16.29.1 ○ FQDN: Enter FQDN

Insecure Mode* □ Allow Insecure Mode

Certificate *(optional)* avisantaclara

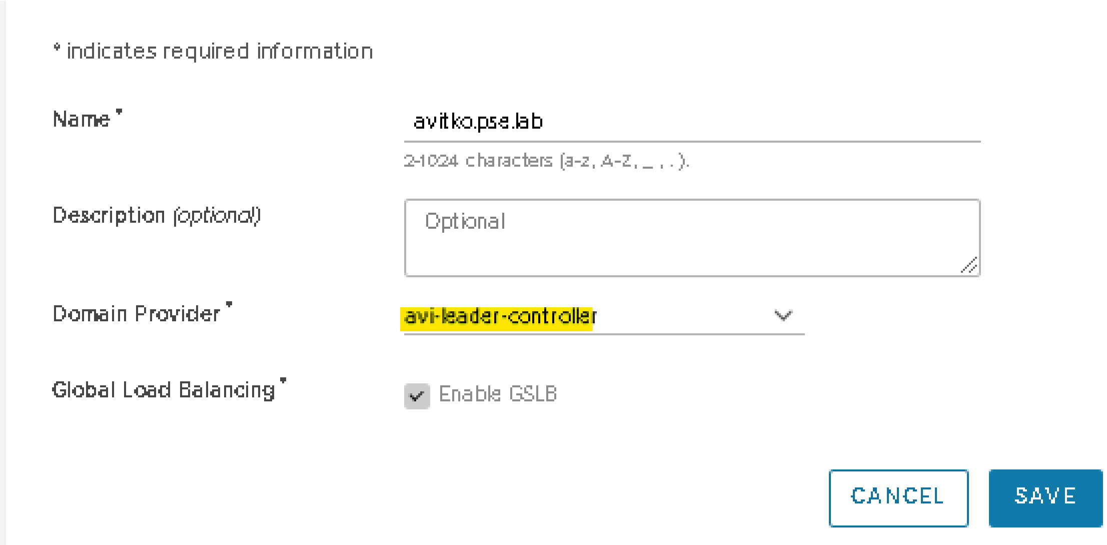
Note: It is recommended to use certificate from NSX Advanced Load Balancer leader controller instead on "*insecure mode*." Insecure mode allows to still use TLS, but do not require globally trusted certificates.

Add DNS and Domains

Once the integration has been created, add a DNS domain the activate it. NSX Advanced Load Balancer account status will still show disconnected (Red) until the DNS and Domain is linked to the integration.

1. Click Tanzu Admin and select DNS and Domains. In the New DNS Account dialog box, select the name of the Avi integration account created in previous step as Domain Provider.

Figure 201: Add domain.



* indicates required information

Name *: avitko.pse.lab

2-1024 characters (a-z, A-Z, _ , .).

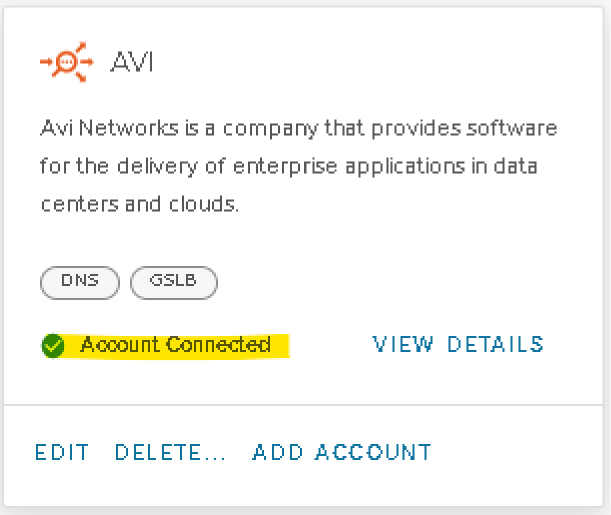
Description (optional): Optional

Domain Provider *: avi-leader-controller

Global Load Balancing *: ☑ Enable GSLB

2. Once done, go back to Integrations page and verify that the Avi integration is green.

Figure 202: Integration complete



AVI

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

✓ Account Connected

Installing Applications

To enable end-user access to frontend services outside the global namespace, Tanzu Service Mesh exposes the service via “public service” construct. A similar construct named “External service” is also implemented. External services are services that exist outside the VMware Tanzu Service Mesh (for example, third-party database services) but are made accessible by services within a global namespace of the VMware Tanzu Service Mesh. Services can run on virtual machines, external Kubernetes clusters, Tanzu Application Service environments, lambda functions or even on bare metal, and can be accessed over TCP, TLS, HTTP, or HTTPS.

Note: When mapping services to the global namespace where you want to configure a public service for GSLB, verify that all the services reside in namespaces with the same name. This is required by the current version of Tanzu Service Mesh and will change in the future.

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

To enable the cross-cluster communication between the services, application manifest for Kubernetes deployment may have to be edited for the appropriate service on one cluster, to specify the domain name of the global namespace and prefixing the domain name with the name of the service on the other cluster.

For example, if a service called ‘frontend’ on one cluster needs to communicate to another service called ‘cart’ on another cluster, the deployment manifests of frontend service on the first cluster needs to be edited to set appropriate variable to “cart.avitko.pse.lab,” for example. The “cart” prefix is required for “frontend” service to communicate with “cart” service. In addition, service protocol type and port need to be added, if it does not exist in the application manifests. Example below.

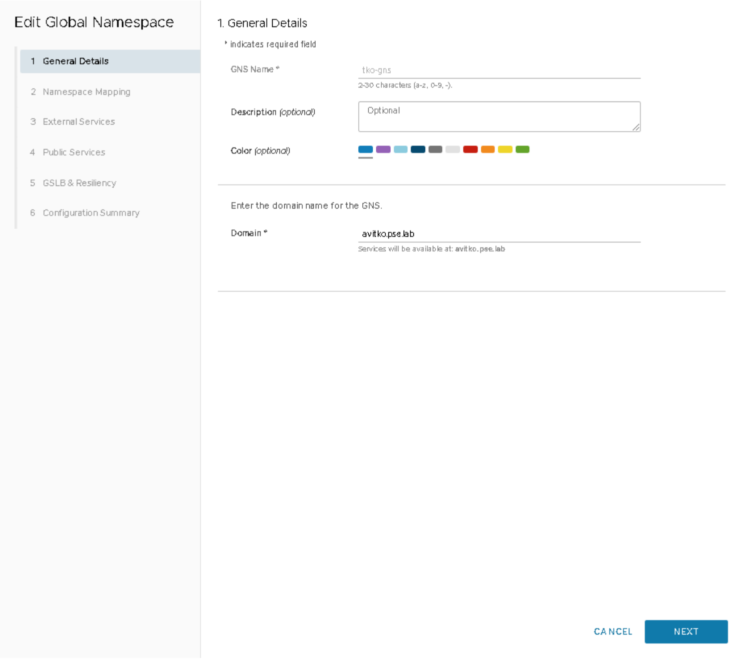
```
apiVersion: v1
kind: Service
metadata:
  name: order-svc
spec:
  ports:
    - appProtocol: http
      number: 3001
```

Create Global Namespace for applications.

With global namespaces in Tanzu Service Mesh, can connect and secure the services across clusters. A global namespace map, discover and connect services automatically across clusters. A global namespace can be shared across a single cluster, multiple clusters, or even clusters in different clouds. Below is a high-level process of creating a global namespace.

1. In the navigation panel on the left, click New Workflow and then click New Global Namespace. On the General Details page of the New Global Namespace wizard, enter a unique name and a domain name for the global namespace.

Figure 203: Global namespace



1. General Details

* indicates required field

GNS Name * tko-gns

2-30 characters (a-z, 0-9, -).

Description *(optional)* Optional

Color *(optional)*

Enter the domain name for the GNS.

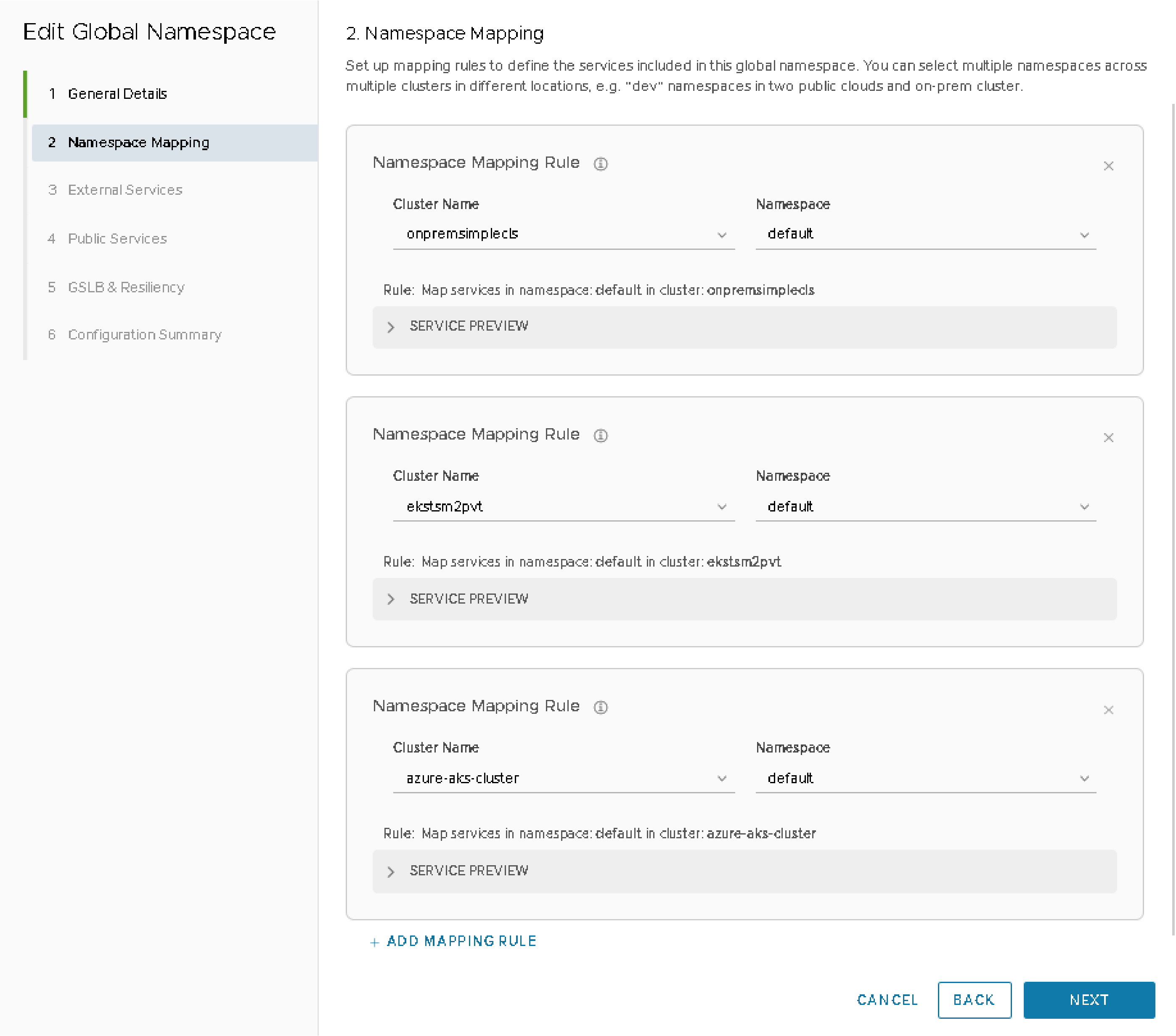
Domain * avitko.pse.lab

Services will be available at: avitko.pse.lab

2. On the Namespace Mapping page, to add the services in your application to the global namespace, specify their Kubernetes namespace-cluster pairs. Under Namespace Mapping Rule, in the left drop-down menu, select the namespace on one of your clusters that holds some of the services and in the right drop-down menu, select the

name of the cluster. Click Add Mapping Rule to create multiple namespace mapping rules for the same or different clusters.

Figure 204: Namespace mappings



Edit Global Namespace

1 General Details
2 Namespace Mapping
3 External Services
4 Public Services
5 GSLB & Resiliency
6 Configuration Summary

2. Namespace Mapping

Set up mapping rules to define the services included in this global namespace. You can select multiple namespaces across multiple clusters in different locations, e.g. "dev" namespaces in two public clouds and on-prem cluster.

Namespace Mapping Rule

Cluster Name: onpremsimplecls
Namespace: default

Rule: Map services in namespace: default in cluster: onpremsimplecls

SERVICE PREVIEW

Namespace Mapping Rule

Cluster Name: ekstsm2pvt
Namespace: default

Rule: Map services in namespace: default in cluster: ekstsm2pvt

SERVICE PREVIEW

Namespace Mapping Rule

Cluster Name: azure-aks-cluster
Namespace: default

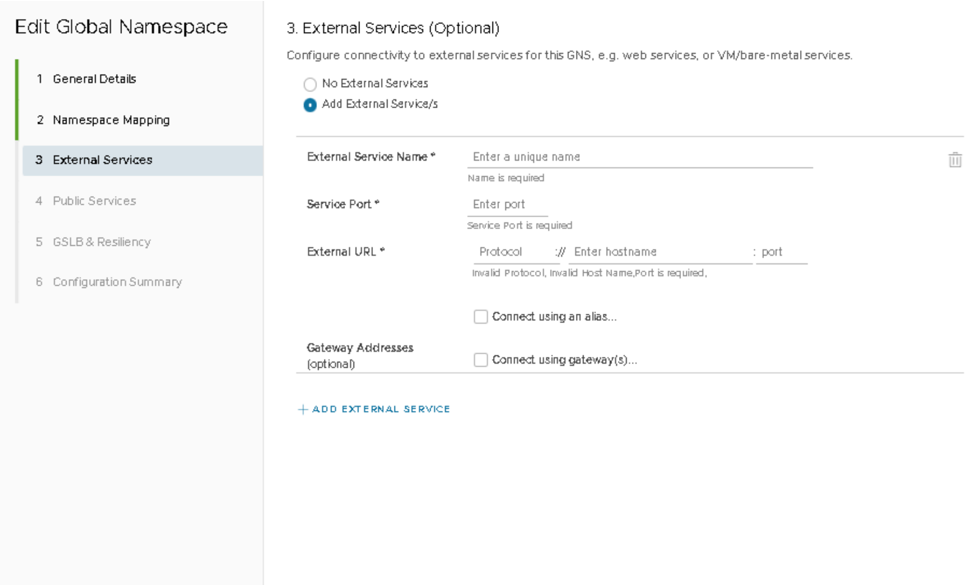
Rule: Map services in namespace: default in cluster: azure-aks-cluster

SERVICE PREVIEW

3. To configure external services in the global namespace, click Add Public Service(s), provide the configuration for each public service, and click Next.

Note: No external service was configured for this reference architecture.

Figure 205: External Service



Edit Global Namespace

1 General Details
2 Namespace Mapping
3 External Services
4 Public Services
5 GSLB & Resiliency
6 Configuration Summary

3. External Services (Optional)

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

○ No External Services
● Add External Service/s

External Service Name * Enter a unique name
Name is required

Service Port * Enter port
Service Port is required

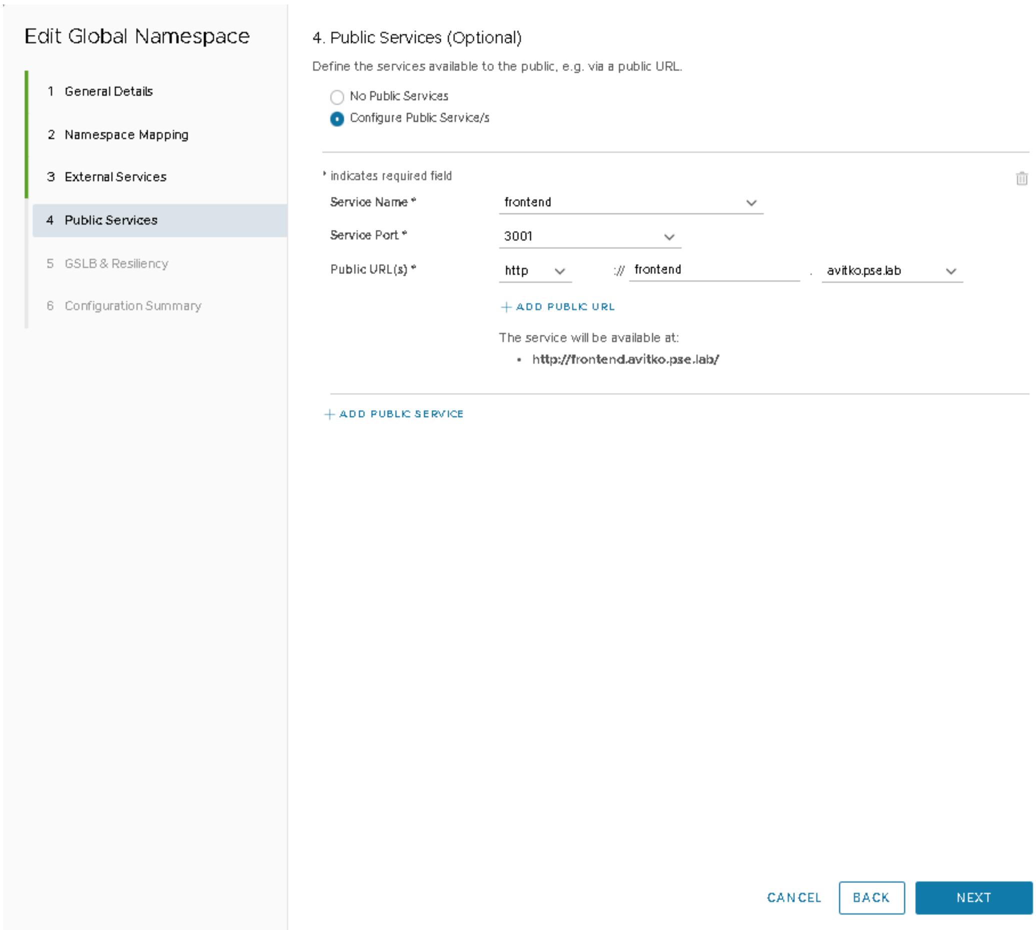
External URL * Protocol :// Enter hostname : port
Invalid Protocol, Invalid Host Name,Port is required.

□ Connect using an alias...

Gateway Addresses (optional) □ Connect using gateway(s)...

4. To configure public services in the global namespace, click Configure Public Service(s), provide the configuration for each public service, and click Next.

Figure 206: Public service



Edit Global Namespace

1 General Details
2 Namespace Mapping
3 External Services
4 Public Services
5 GSLB & Resiliency
6 Configuration Summary

4. Public Services (Optional)

Define the services available to the public, e.g. via a public URL.

○ No Public Services
● Configure Public Service/s

* indicates required field

Service Name * frontend

Service Port * 3001

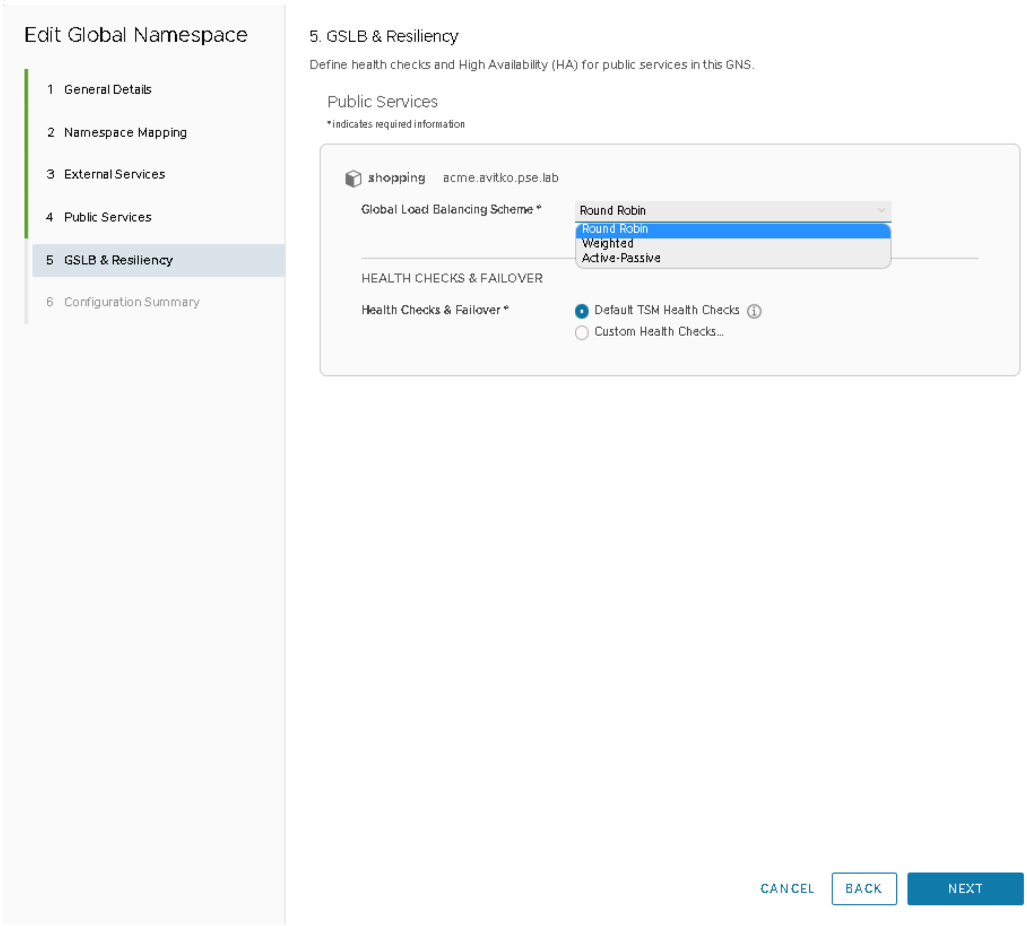
Public URL(s) * http :// frontend . avitko.pse.lab

The service will be available at:

- http://frontend.avitko.pse.lab/

5. On the GLSB & Resiliency page, configure global load balancing scheme for the public services and click Next.

Figure 207: load balancing scheme



Edit Global Namespace

1 General Details
2 Namespace Mapping
3 External Services
4 Public Services
5 GSLB & Resiliency
6 Configuration Summary

5. GSLB & Resiliency

Define health checks and High Availability (HA) for public services in this GNS.

Public Services

*indicates required information

shopping acme.avitko.pse.lab

Global Load Balancing Scheme *: Round Robin

Round Robin
Weighted
Active-Passive

HEALTH CHECKS & FAILOVER

Health Checks & Failover *

● Default TSM Health Checks

○ Custom Health Checks...

6. On the Configuration Summary page, review the configuration of the global namespace and click Finish.

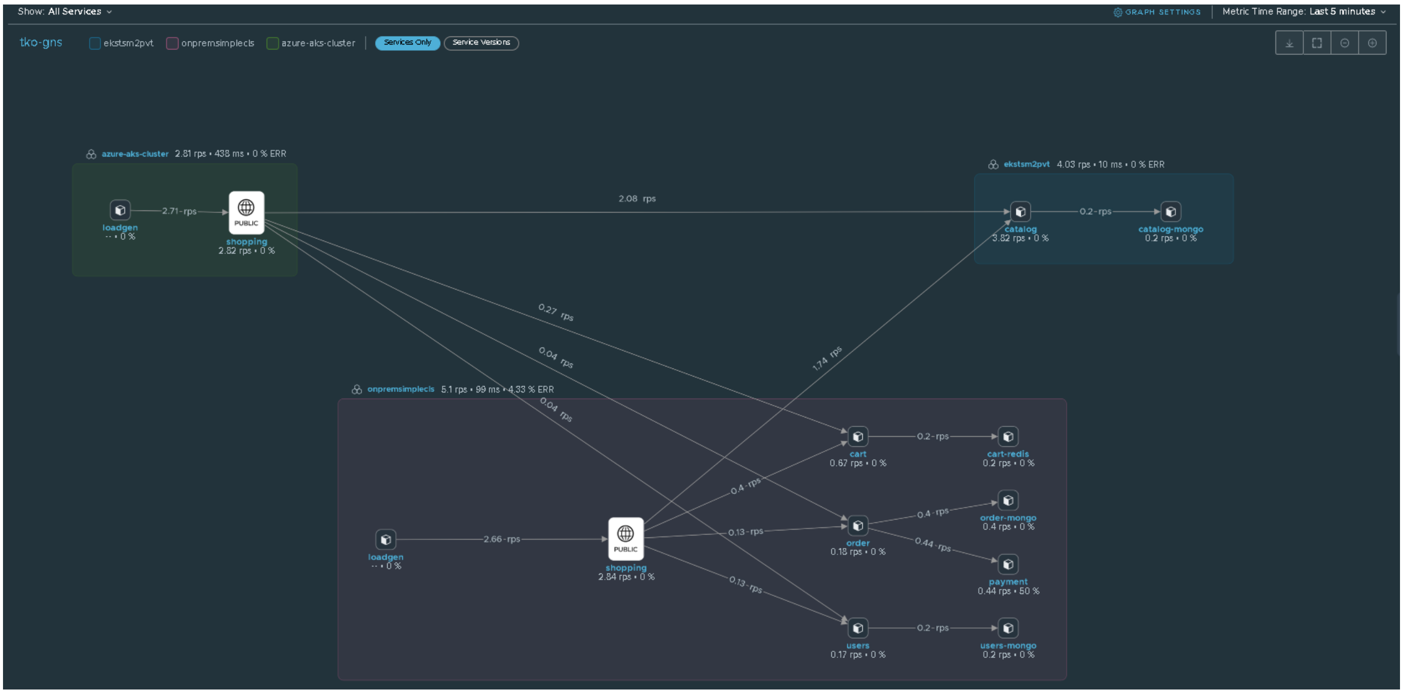
Service autoscaling policies

Autoscaling represents the ability of a service to automatically scale up or down to efficiently handle changes of the service demand. With Tanzu Service Mesh Service Autoscaler, developers and operators can have automatic scaling of microservices that meet changing levels of demand based on metrics, such as CPU or memory usage. These metrics are available to Tanzu Service Mesh without needing additional code changes or metrics plugins.

Tanzu Service Mesh Autoscaler supports configuring an autoscaling policy for services inside a global namespace through the UI as well as API. For more information, see Configure GNS-Scoped Autoscaling Policy Using Tanzu Service Mesh UI. Tanzu Service Mesh Autoscaler also provides a Kubernetes Custom Resource Definition to configure autoscaling for services directly in cluster namespaces. For more information, see Deploying the Tanzu Service Mesh Service Autoscaler Through CRD. This approach for configuring autoscaling with CRD is available only for org-scoped autoscaling policies. Once an autoscaling policy is configured, Tanzu Service Mesh starts to monitor the Autoscaler metric for the service and scales the service accordingly.

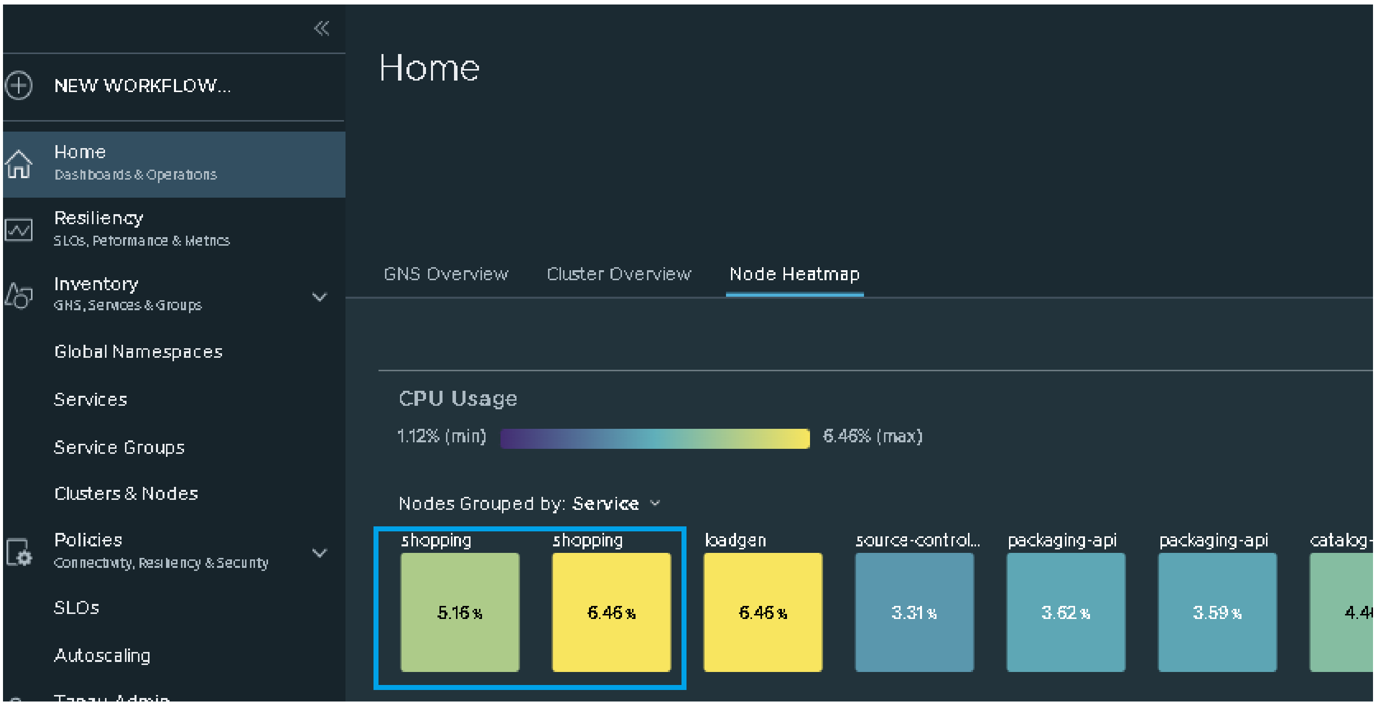
Below is an example of configuring autoscaling a front-end service called “shopping” based on service CPU usage. Figure 208 shows three clusters as part of the global namespace with application services distributed across the three cloud instances. The catalog service is on Amazon EKS cluster. There are two instances of the frontend “shopping” service, one on Azure AKS and one on on-premises private cloud. On-premises private cloud also holds the rest of the application services.

Figure 208: Distributed application



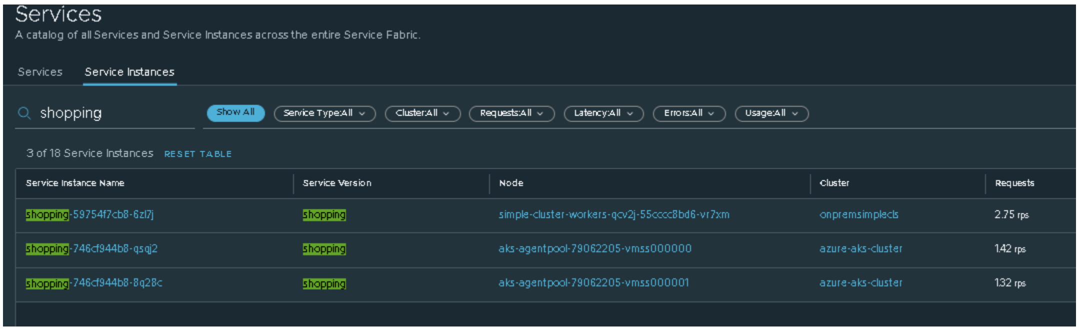
To ensure that the frontend service where the users connect to the applications, performs efficiently, an autoscaling policy will be created to scale shopping service up or down depending on CPU usage of the microservice. A load generator will be used to simulated user traffic by generating load on the frontend service. Figure 209 shows the current CPU usage for both services prior to load generation.

Figure 209: CPU usage



There are three instances of the application currently running as shown below.

Figure 210: Service instances



Services

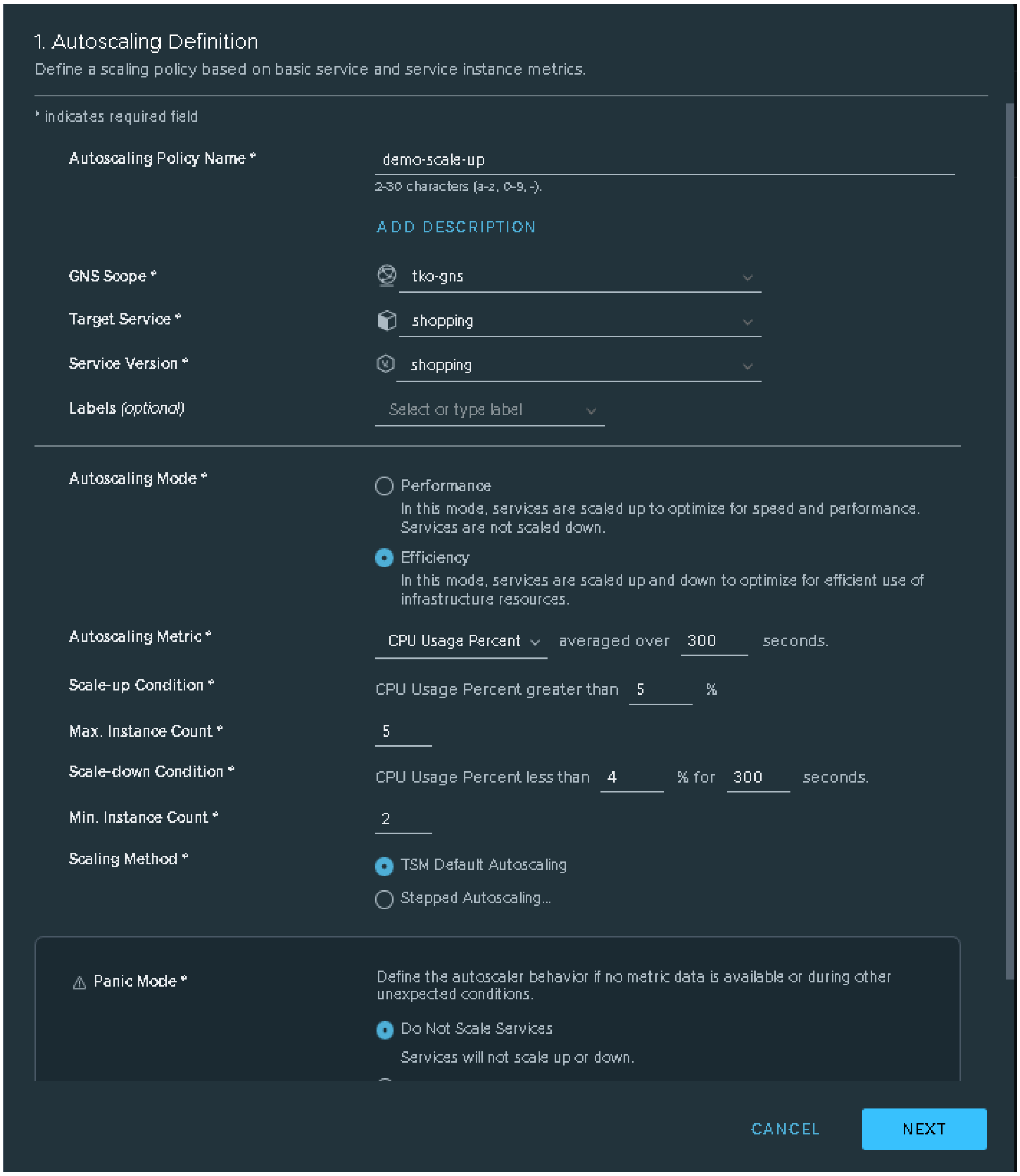
A catalog of all Services and Service Instances across the entire Service Fabric.

3 of 18 Service Instances

Service Instance Name	Service Version	Node	Cluster	Requests
shopping-59754f7cb8-6zl7j	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-vr7xm	onpremsimplecls	2.75 rps
shopping-746cf944b8-qsqj2	shopping	aks-agentpool-79062205-vmss000000	azure-aks-cluster	1.42 rps
shopping-746cf944b8-8q28c	shopping	aks-agentpool-79062205-vmss000001	azure-aks-cluster	1.32 rps

To simulate service scale-up, an Autoscaling policy is created show below in figure 211.

Figure 211: Simulate service scale-up.



1. Autoscaling Definition

Define a scaling policy based on basic service and service instance metrics.

* indicates required field

Autoscaling Policy Name *: demo-scale-up

2-30 characters (a-z, 0-9, -).

GNS Scope *: tko-gns

Target Service *: shopping

Service Version *: shopping

Labels *(optional)*: Select or type label

Autoscaling Mode *:

○ Performance
In this mode, services are scaled up to optimize for speed and performance. Services are not scaled down.

● Efficiency
In this mode, services are scaled up and down to optimize for efficient use of infrastructure resources.

Autoscaling Metric *: CPU Usage Percent averaged over 300 seconds.

Scale-up Condition *: CPU Usage Percent greater than 5 %

Max. Instance Count *: 5

Scale-down Condition *: CPU Usage Percent less than 4 % for 300 seconds.

Min. Instance Count *: 2

Scaling Method *:

● TSM Default Autoscaling

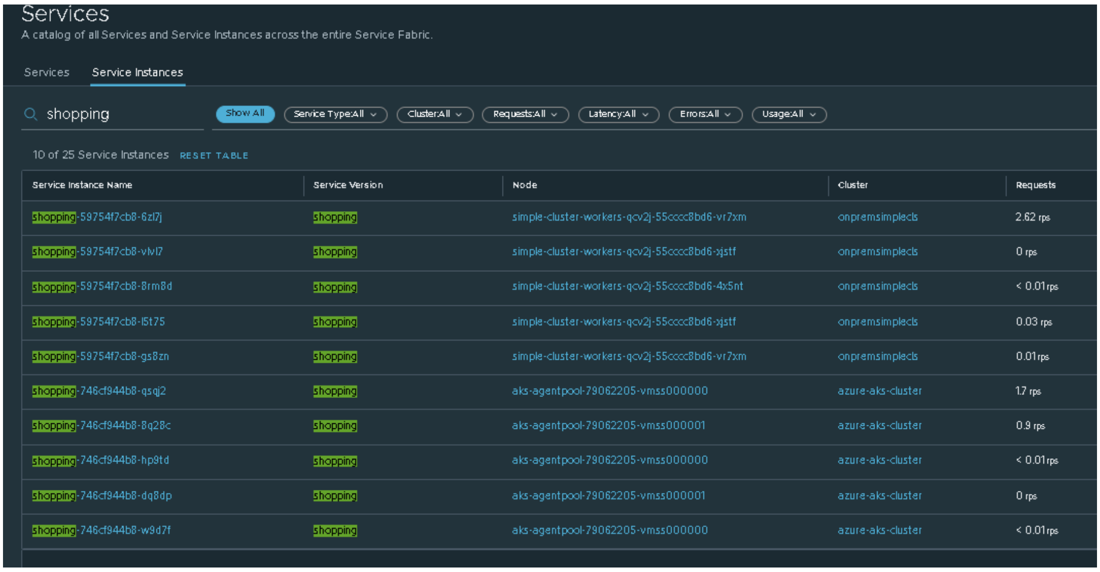
○ Stepped Autoscaling...

Panic Mode *: Define the autoscaler behavior if no metric data is available or during other unexpected conditions.

● Do Not Scale Services
Services will not scale up or down.

After 5 minutes interval set in the policy, where the CPU usage was above the 5% threshold the on-premises and Azure AKS shopping services are scaled up as shown in figure 212 below.

Figure 212: Services scaled up.



Services

A catalog of all Services and Service Instances across the entire Service Fabric.

10 of 25 Service Instances

Service Instance Name	Service Version	Node	Cluster	Requests
shopping-59754f7cb8-6zl7j	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-vr7xm	onpremsimplecls	2.62 rps
shopping-59754f7cb8-vlvl7	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-xjstf	onpremsimplecls	0 rps
shopping-59754f7cb8-8rm8d	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-4x5nt	onpremsimplecls	< 0.01 rps
shopping-59754f7cb8-l5t75	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-xjstf	onpremsimplecls	0.03 rps
shopping-59754f7cb8-gs8zn	shopping	simple-cluster-workers-qcv2j-55cccc8bd6-vr7xm	onpremsimplecls	0.01 rps
shopping-746cf944b8-qsqj2	shopping	aks-agentpool-79062205-vmss000000	azure-aks-cluster	1.7 rps
shopping-746cf944b8-8q28c	shopping	aks-agentpool-79062205-vmss000001	azure-aks-cluster	0.9 rps
shopping-746cf944b8-hp9td	shopping	aks-agentpool-79062205-vmss000000	azure-aks-cluster	< 0.01 rps
shopping-746cf944b8-dq8dp	shopping	aks-agentpool-79062205-vmss000001	azure-aks-cluster	0 rps
shopping-746cf944b8-w9d7f	shopping	aks-agentpool-79062205-vmss000000	azure-aks-cluster	< 0.01 rps

Note: The above is only an example of CPU usage and scaling policy setting accordingly. CPU usage and corresponding policy settings will vary as requirements changes.

Service Level Object Policy

Service level objectives (SLOs) provide a formalized way to describe, measure, and monitor the performance, quality, and reliability of microservice applications. SLOs provide a shared quality benchmark for application and platform teams to reference for gauging service level agreement (SLA) compliance and continuous improvement.

An SLO describes the high-level objective for acceptable operation and health of one or more services over a length of time (for example, a week or a month). Operators can specify, for example, that a service or application should be healthy 99 percent of the time. An SLO of 99 percent permits a service to have an Error Budget of 1 percent of the time which means to be “unhealthy” 1 percent of the time, which allows for realistic downtime, error cases, planned maintenance windows, and service upgrades. Teams can specify which performance characteristics and thresholds are key to the health of their applications. Multiple SLOs can be defined for a single service, reflecting the reality of Quality of Service (QoS) contracts between different classes of end users.

An SLO consists of one or more service level indicators (SLIs). SLOs defined using a combination of SLIs allow teams to describe service health in a more precise and relevant way. SLIs capture important low-level performance characteristics for a particular service. Tanzu Service Mesh collects SLI metrics on ten second intervals for every service instance that is part of the mesh. An example of an SLI would be 99 percent of successful requests respond with latencies faster than 350 ms (99th percentile latency < 350 ms). Another example is an SLI set for a service that responds with error codes for fewer than 0.1 percent of requests (error rate < 0.1%). Tanzu Service Mesh incorporates SLO and SLI measurements by displaying them in real time through its user interface.

Actionable SLOs can also be used in combination to, or drive autoscaling for services. An example of this configuration is found here.

Appendix A: Sample .yaml files

For reference, the yaml files used for AKO and AMKO are provided below. These files are for reference only. Configuration is provided as a sample only and will vary based on the environment.

AKO yaml

```
---
AKOSettings:
  apiServerPort: 8080
  clusterName: tko2-simple-cluster
  cniPlugin: antrea
  deleteConfig: "false"
  disableStaticRouteSync: "false"
  enableEVH: false
  enableEvents: "true"
  fullSyncFrequency: "1800"
  layer7Only: false
  logLevel: WARN
  namespaceSelector:
    labelKey: ""
    labelValue: ""
  primaryInstance: true
  servicesAPI: true
  vipPerNamespace: "false"
ControllerSettings:
  cloudName: Default-Cloud
  controllerHost: "172.29.250.20"
  controllerVersion: "20.1.7"
  serviceEngineGroupName: Default-Group
  tenantName: admin
L4Settings:
  autoFQDN: default # Set to “disabled” if using Tanzu Service Mesh
  defaultDomain: pse.lab
L7Settings:
  defaultIngController: "true"
  enableMCI: "false"
  noPGForSNI: false
  passthroughShardSize: SMALL
  serviceType: ClusterIP
  shardVSSize: LARGE
NetworkSettings:
  bgpPeerLabels: []
  enableRHI: false
  nodeNetworkList: []
  nsxtT1LR: ""
  vipNetworkList:
    - cidr: 172.31.248.0/22
      networkName: DVS-Frontend-Network
avicredentials:
#  authtoken: ~
#  certificateAuthorityData: ~
  password: password
  username: admin
image:
  pullPolicy: IfNotPresent
  repository: projects.registry.vmware.com/ako/ako
logFile: avi.log
mountPath: /log
nodePortSelector:
  key: ""
  value: ""
persistentVolumeClaim: ""
podSecurityContext: {}
rbac:
  pspEnable: false
replicaCount: 1
resources:
  limits:
    cpu: 350m
    memory: 400Mi
  requests:
    cpu: 200m
    memory: 300Mi
```

AMKO Yaml

```
---
configs:
  controllerVersion: "20.1.7"
  gslbLeaderController: "172.29.250.20"
  logLevel: INFO
  memberClusters:
    -      clusterContext: "arn:aws:eks:us-east-1:8728XXXXXXX:cluster/tkoeks"
    -      clusterContext: tko2-simple-cluster
  refreshInterval: 1800
  useCustomGlobalFqdn: false
globalDeploymentPolicy:
  appSelector:
    label:
      amko: "gslb   <example label key-value for an ingress/service type LB>"
  matchClusters:
    -      cluster: "arn:aws:eks:us-east-1:8728XXXXXXX:cluster/tkoeks"
    -      cluster: tko2-simple-cluster
gslbLeaderCredentials:
  password: password
  username: admin
image:
  pullPolicy: IfNotPresent
  repository: projects.registry.vmware.com/ako/amko
logFile: amko.log
mountPath: /
multiClusterIngress:
  enable: false
persistentVolumeClaim: ""
rbac:
  pspEnable: false
replicaCount: 1
resources:
  limits:
    cpu: 250m
    memory: 300Mi
  requests:
    cpu: 100m
    memory: 200Mi
service:
  port: 80
  type: ClusterIP
serviceAccount:
  annotations: {}
  create: true
  name: ~
serviceDiscovery: ~
```

About the Author

Ather Jamil is a member of the technical staff in the Office of the CTO at VMware. Ather brings 25 plus years of experience in the Information Technology industry. Starting his career with Compaq Computers in the early 90' while still in college, Ather was soon leading efforts in building several "industry first" technologies including PC Blades, virtual storage, and architectures and composable infrastructure. Ather also spent part of his career building video analytics and intelligence solution for several organizations in the public sector and has authored several papers in the information technology field. Building innovative and cutting-edge solutions that solve customer problems is Ather's passion and part of his responsibilities at VMware.