Getting Started with VMware Integrated OpenStack with Kubernetes

VMware Integrated OpenStack 5.0

vmware®

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

https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

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

Contents

1 Getting Started with VMware Integrated OpenStack with Kubernetes 5

2 About VMware Integrated OpenStack with Kubernetes 6

- VMware Integrated OpenStack with Kubernetes Architecture 6
- Backend Networking 7

3 Installing VMware Integrated OpenStack with Kubernetes 11

- System Requirements 11
- Deploy the VMware Integrated OpenStack with Kubernetes OVA in the vSphere Web Client 12

4 Adding a Cloud Provider 14

- OpenStack Cloud Provider 14
- VMware SDDC Provider 17

5 Creating Your First Cluster 23

- Understanding Cluster Configuration Settings 23
- Add a New Kubernetes Cluster 24
- Configure User and Group Access for an Exclusive Cluster 26
- Create a Namespace for Users and Groups on a Shared Cluster 26

6 Managing Your Deployment 28

- Customizing Your Cluster 28
- Cluster Management 30
- Increasing Capacity 31
- Monitoring Your Kubernetes Cluster 32
- Certificate Management Using the CLI 34
- Collect Logs from SDDC Provider Services 35
- Add Helm to Your VMware Integrated OpenStack with Kubernetes Deployment 36
- Upgrade to VMware Integrated OpenStack with Kubernetes 5.0 37
- Upgrade Clusters to Support Current Kubernetes Version 37

7 Optimizing Kubernetes Cluster Performance 39

8 Troubleshooting Your VMware Integrated OpenStack with Kubernetes Infrastructure 42

- Troubleshoot Dashboard Load Failure 42
- Troubleshoot Cluster Update Failure 43

9 VMware Integrated OpenStack with Kubernetes CLI Command Reference 46

vkube cluster update Command 46

vkube cluster heal 46

vkube job backup Command 47

vkube job restore Command 47

vkube job recover Command 48

vkube job get Command 48

vkube job list Command 49

vkube nodegroup Command 49

Getting Started with VMware Integrated OpenStack with Kubernetes

1

Getting Started with VMware Integrated OpenStack with Kubernetes provides information about how to install, deploy, and use VMware Integrated OpenStack with Kubernetes.

Intended Audience

As a system administrator, you can use VMware Integrated OpenStack with Kubernetes to manage containers in your Kubernetes cluster.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

About VMware Integrated OpenStack with Kubernetes 2

With VMware Integrated OpenStack with Kubernetes you can deploy and maintain enterprise class Kubernetes clusters in an OpenStack environment.

The Kubernetes clusters are configured to use VMware Integrated OpenStack enterprise-grade services such as Keystone authentication for your cluster, Block Storage Cinder to provide persistent storage for your stateful applications, and Neutron Load Balancing as a Service (LBaaS) for your application services.

You deploy Kubernetes clusters through the VMware Integrated OpenStack with Kubernetes vApp in vCenter. The vApp provides a workflow that guides you through and completes the Kubernetes deployment process.

This chapter includes the following topics:

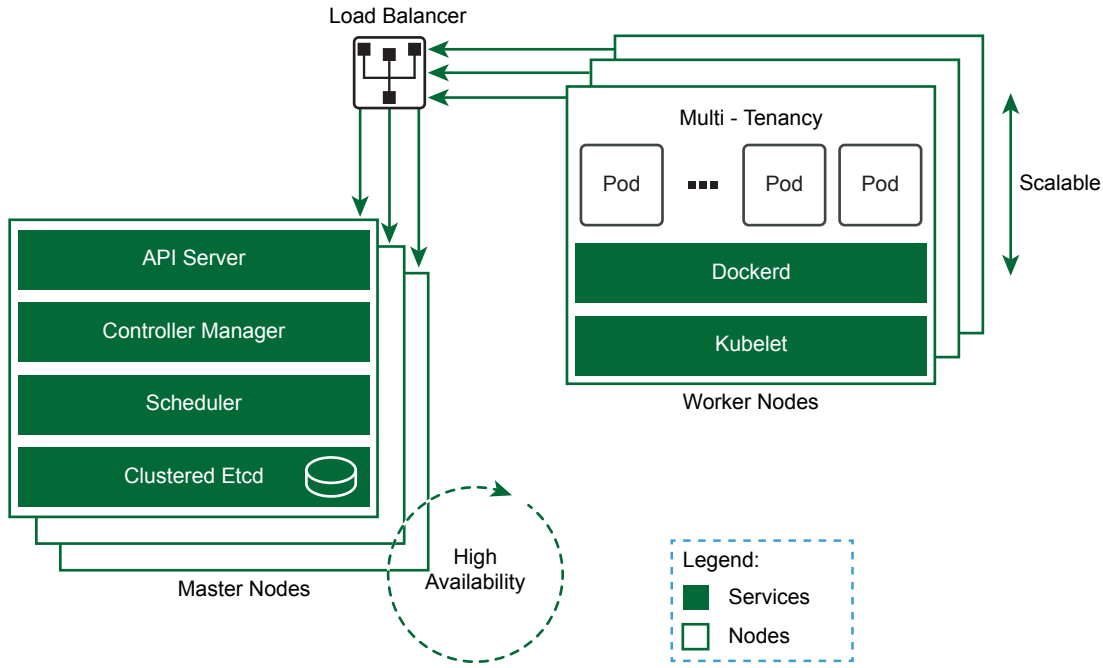
- VMware Integrated OpenStack with Kubernetes Architecture
- Backend Networking

VMware Integrated OpenStack with Kubernetes Architecture

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure. By combining Kubernetes with VMware Integrated OpenStack, you can use a common infrastructure management layer to provision both VMs and containers.

VMware Integrated OpenStack with Kubernetes builds high-availability Kubernetes clusters that support scalability and multi-tenancy.

Figure 2-1. VMware Integrated OpenStack with Kubernetes Built Cluster



The high-availability Kubernetes cluster consists of load-balanced master nodes, replicated API servers, and clustered etcd services. In addition, you can scale out or scale in the worker nodes in a Kubernetes cluster to meet changing demands for capacity.

Using the concept of a namespace, a single Kubernetes cluster can be shared among multiple users or groups, or partitioned into multiple virtual clusters. With the namespace management feature, you can configure multi-tenancy on a shared Kubernetes cluster. Or you can create a Kubernetes cluster in exclusive mode, where any authorized user or group has privileges to manage the namespace.

Backend Networking

VMware Integrated OpenStack with Kubernetes supports VDS, NSX-V, and NSX-T backend networking.

Networking Support

Container network and load balancer support for Kubernetes Services is dependent on the backend networking.

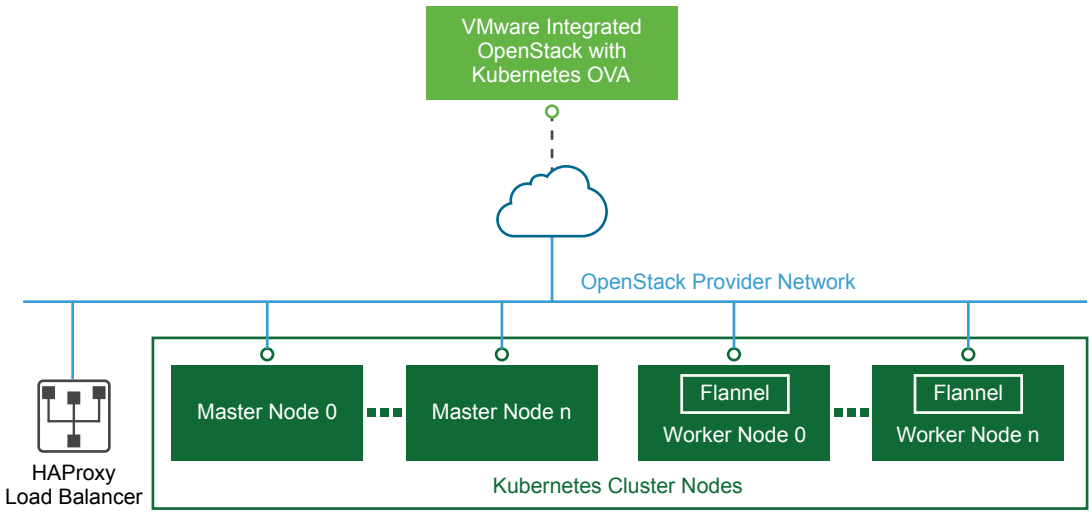
Backend Networking	Container Network	Load Balancer	Network Policy
VDS	Flannel	Kubernetes Nginx Ingress Controller	Unsupported
NSX-V	Flannel	NSX Edge	Unsupported
NSX-T	NSX-T Container Plugin	NSX-T Edge	Supported

Where:

- The container network is the Container Network Interface (CNI) plugin for container networking management. The plugin can allocate IP addresses, create a firewall, and create ingress for the pods.
  - Flannel is a network fabric for containers and is the default for VDS and NSX-V networking.
  - NSX-T Container Plug-in (NCP) is a software component that sits between NSX manager and the Kubernetes API server. It monitors changes on Kubernetes objects and creates networking constructs based on changes reported by the Kubernetes API. NCP includes native support for containers. It is optimized for NSX-T networking and is the default.
- The load balancer is a type of service that you can create in Kubernetes. To access the load balancer, you specify the external IP address defined for the service.
  - The Kubernetes Nginx Ingress Controller is deployed on VDS by default but can be deployed on any backend platform. The Ingress Controller is a daemon, deployed as a Kubernetes Pod. It watches the ingress endpoint of the API server for updates to the ingress resource so that it can satisfy requests for ingress. Because the load balancer service type is not supported in a VDS environment, the Nginx Ingress Controller is used to expose the service externally.
  - The NSX Edge load balancer distributes network traffic across multiple servers to achieve optimal resource use, provide redundancy, and distribute resource utilization. It is the default for NSX-V and is leveraged for Kubernetes services.
  - The NSX-T Edge load balancer functions the same as the NSX Edge load balancer and is the default for NSX-T. However, when using the NCP, you cannot specify the IP address of the static load balancer.
- By default, pods accept network traffic from any source. Network policies provide a way of restricting the communication between pods and with other network endpoints. See https://kubernetes.io/docs/concepts/services-networking/network-policies/.

VDS Backend

VDS or vSphere Distributed Switch supports virtual networking across multiple hosts in vSphere.

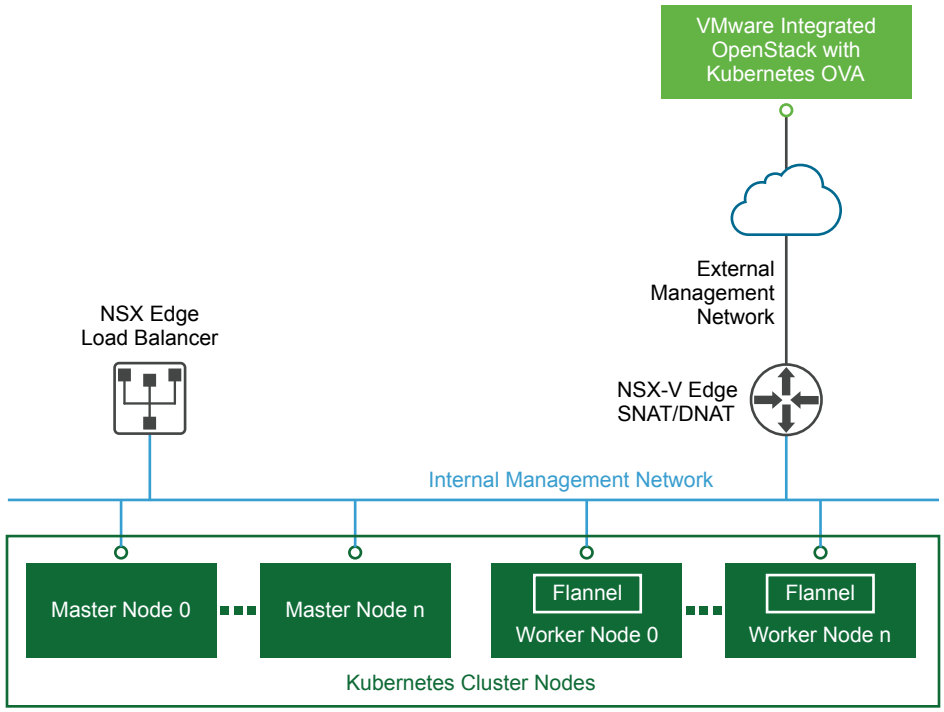


With the VDS backend, VMware Integrated OpenStack with Kubernetes deploys Kubernetes cluster nodes directly on the OpenStack provider network. The OpenStack cloud administrator must verify that the provider network is accessible from outside the vSphere environment. VDS networking does not include native load balancing functionality for the cluster nodes, so VMware Integrated OpenStack with Kubernetes deploys HAProxy nodes outside the Kubernetes cluster to provide load balancing.

NSX-V Backend

NSX-V is the VMware NSX network virtualization and security platform for vSphere.

Figure 2-2. NSX-V backend networking

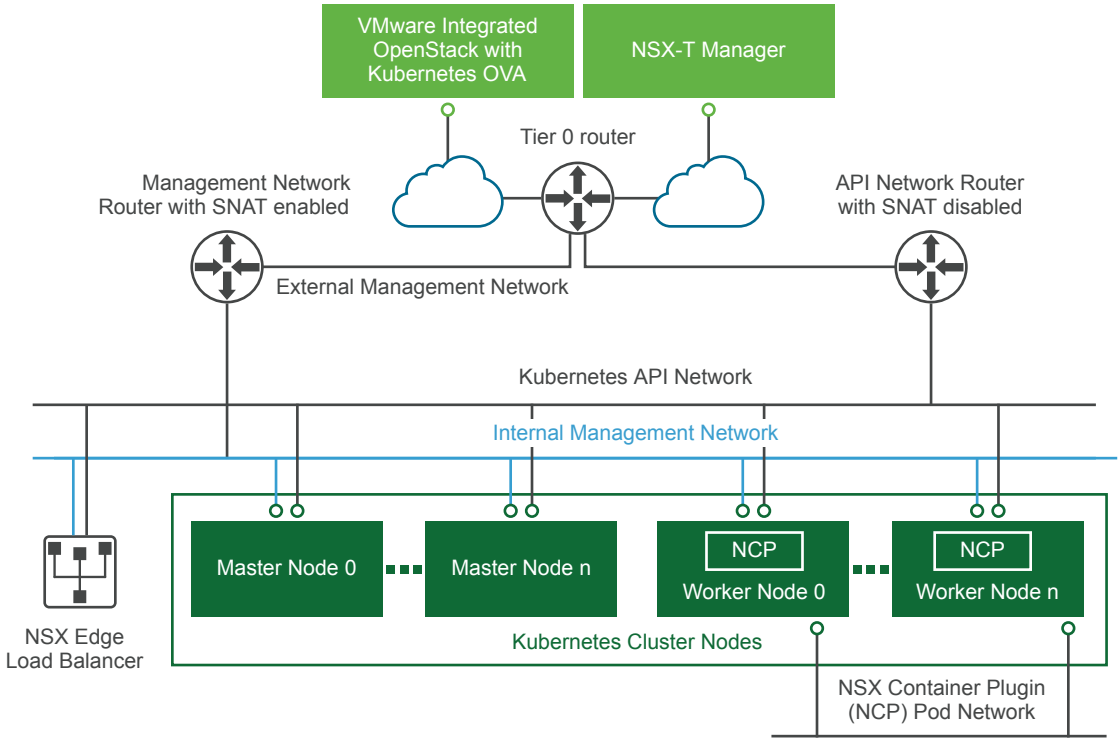


With the NSX-V backend, VMware Integrated OpenStack with Kubernetes deploys multiple nodes within a single cluster behind the native NSX Edge load balancer. The NSX Edge load balancer manages up to 32 worker nodes. Every node within the Kubernetes cluster is attached to an internal network and the internal network is attached to a router with a default gateway set to the external management network.

NSX-T Backend

NSX-T supports networking on a variety of compute platforms including KVM.

Figure 2-3. NSX-T backend networking



With the NSX-T backend, the VMware Integrated OpenStack with Kubernetes deploys worker nodes each with three NICs.

- One NIC connects to the NCP Pod network, an internal network with no routing outside the vSphere environment. When the NSX Container Plugin is enabled, this NIC is dedicated to Pod traffic. IP addresses used for translating Pod IPs using SNAT rules and for exposing ingress controllers using SNAT/DNAT rules are referred to as external IPs.
- One NIC connects to the Kubernetes API network which is attached to a router with SNAT disabled. Special Pods such as KubeDNS can access the API server using this network.
- One NIC connects to the internal management network. This NIC is accessible from outside the vSphere environment through a floating IP that is assigned by the external management network.

NSX-T networking does not include native load balancing functionality, so VMware Integrated OpenStack with Kubernetes creates two separate load balancer nodes. The nodes connect to the management network and API network.

Installing VMware Integrated OpenStack with Kubernetes

3

You install VMware Integrated OpenStack with Kubernetes as a virtual appliance in vSphere.

This chapter includes the following topics:

- System Requirements
- Deploy the VMware Integrated OpenStack with Kubernetes OVA in the vSphere Web Client

System Requirements

Before you begin the deployment tasks, your system must comply with all hardware, software, networking, and storage requirements.

System requirements specific to VMware Integrated OpenStack with Kubernetes are required in addition to system requirements for VMware Integrated OpenStack. See the *Installing and Configuring VMware Integrated OpenStack* guide.

Hardware Requirements for VMware Integrated OpenStack with Kubernetes

The hardware requirements are based on the number of VMs used for each component.

Core VMware Integrated OpenStack with Kubernetes Components

Component	VMs	vCPU	vRAM (GB)	vDisk Space (GB)
VMware Integrated OpenStack with Kubernetes	1	4	16	60

Software Requirements for VMware Integrated OpenStack with Kubernetes

Before you begin installing VMware Integrated OpenStack with Kubernetes, verify that the software components meet all of the version prerequisites for vSphere, ESXi hosts, and the NSX product.

Requirement	Description
VMware Integrated OpenStack	■ Deployment on an OpenStack provider requires VMware Integrated OpenStack version 3.1 or 4.0
vSphere version	■ vSphere 6.0 or later with Virtual Distributed Switch. ■ Deployment with NSX-T backend networking requires vSphere 6.5.
ESXi hosts	■ Deployment on an SDDC provider requires ESXi version 6.0 or later
NSX	■ Deployment with NSX backend networking requires NSX version 6.2 or 6.3.
NSX-T	■ Deployment with NSX-T backend networking requires NSX-T version 2.0

Deploy the VMware Integrated OpenStack with Kubernetes OVA in the vSphere Web Client

VMware Integrated OpenStack with Kubernetes is a vApp that you deploy using a wizard in the vSphere Web Client.

Prerequisites

- Verify that vSphere is installed and correctly configured. See System Requirements.
- Obtain the VMware Integrated OpenStack with Kubernetes OVA file. Go to downloads for VMware Integrated OpenStack from https://www.vmware.com/go/download-openstack.

Procedure

1. Log in to the vSphere Web Client.
2. Go to the vCenter **Hosts and Cluster** view.
3. Right click the cluster where you want to deploy VMware Integrated OpenStack with Kubernetes and select **Deploy OVF Template**.
4. Access the downloaded VMware Integrated OpenStack with Kubernetes OVA.
5. Specify the destination and configure the OVA deployment.
   a. Select the target datacenter created specifically for the VMware Integrated OpenStack with Kubernetes OVA, and click **Next**.
   b. Select a target host, cluster, resource pool, or vApp and click **Next**.
   c. Review the product, version, vendor, publisher, size and description details and click **Next**.
   d. Select your storage options and click **Next**.

e To set up your networks, select the destination network for your source and click **Next**.

The source is your management network and provides access to infrastructure components such as vCenter server, ESX, NSX-V, and NSX-T. The destination network you select must have access to the infrastructure components.

f Customize the deployment properties by configuring the networking properties and root user properties.

- Networking properties are required to configure a static IP for the VM. Leave the properties blank if you want to the DHCP server to provide the IP address.
- The root user uses the initial password to log in to VMware Integrated OpenStack with Kubernetes.

6 Click **Next**.

7 Review the deployment settings and click **Finish** to deploy VMware Integrated OpenStack with Kubernetes.

8 After the file finishes deploying into vCenter Server, power on the VMware Integrated OpenStack with Kubernetes appliance.

9 Wait for the machine to start, and right-click to obtain the IP address for the VM.

Adding a Cloud Provider

4

Before you deploy a Kubernetes cluster, you must configure a cloud provider. VMware Integrated OpenStack with Kubernetes uses the cloud provider to create the infrastructure required to deploy all your Kubernetes clusters. User management on the provider provides the basis for VMware Integrated OpenStack with Kubernetes user management on the cluster.

When choosing the type of provider to create, consider the following:

- With an existing VMware Integrated OpenStack deployment, you can create an OpenStack provider.
- Without an existing VMware Integrated OpenStack deployment, you can create an SDDC provider if you do not want to deploy a standalone VMware Integrated OpenStack instance.

This chapter includes the following topics:

- OpenStack Cloud Provider
- VMware SDDC Provider

OpenStack Cloud Provider

With an OpenStack provider, VMware Integrated OpenStack with Kubernetes deploys and configures Kubernetes clusters on an existing VMware Integrated OpenStack deployment.

An OpenStack provider supports NSX-V or NSX-T networking.

Network Requirements for OpenStack Provider

In addition to system requirements required for installation, your NSX-V or NSX-T network must satisfy requirements for an OpenStack provider.

NSX-V Requirements

The following requirements apply to NSX-V networking:

- Ubuntu 16.04 image installed in your VMware Integrated OpenStack cloud.
- An external network used to communicate with the VMware Integrated OpenStack with Kubernetes OVA. The external network must have at least one floating IP available. Additional floating IPs are required if you expose Kubernetes applications using Service with Load Balancer.
- An internal network for Kubernetes nodes. DNS must be available in your internal network.

- A centralized, exclusive router with a gateway configured for the external network.
- Security group for the ingress traffic. Configure Ingress with ports 22, 443, 11000, and 11001 open or cluster creation will fail.

NSX-T Requirements

In addition to the requirements for NSX-V networking, verify that NSX-T is version 2.0.

Input Parameters for an OpenStack Provider

This section describes the input parameters required to add an OpenStack provider. In addition, NSX-T backend networking requires specific configuration parameters.

An OpenStack provider requires the following information.

Table 4-1. OpenStack Authentication

Variable	Description
Keystone Public URL	Full Keystone public endpoint URL including protocol (http or https), port and API version. For example, https://openstack.cloud:5000/v3.
Username	OpenStack username
Password	OpenStack password
Project name	OpenStack project name
Region name (Default: nova)	OpenStack region name
Domain name	OpenStack domain name. Must be set for v3.
CA Certificate	Certificate for authentication with the OpenStack Keystone service that is located on the VMware Integrated OpenStack loadbalancer node at `/usr/local/share/ca-certificates/vio.crt`.

Table 4-2. Image and Flavor

Variable	Description
Image username	Used to establish SSH connection with cluster nodes. This user must be able run sudo without a password. For example, the default user for Ubuntu cloud images is **`ubuntu`**.
Image ID of the Ubuntu image	OpenStack image ID **Note** The image must have `'{"disk.EnableUUID":"TRUE"}'` in the vmware_extra_config property.
Flavor ID	OpenStack flavor ID

Table 4-3. Networking and Security

Variable	Description
NSX-T Networking	See Configuration Information for NSX-T Networking.
Security Group ID	Security group ID to be applied to all VMs
Internal Network ID of Kubernetes cluster network	Internal network ID used for nodes IPs
Internal network Subnet ID	Subnet ID of the internal network used for allocating the IPs
External Network ID used for floating IPs	External network ID used to assign floating IPs

Configuration Information for NSX-T Networking

NSX-T networking requires specific input parameters.

Variable	Description
Manager address	NSX-T manager FQDN or IP
Username	NSX-T manager username
Password	NSX-T manager password
Tier 0 Router	Tier 0 router ID configured for OpenStack
Transport zone	Transport zone ID configured for OpenStack

Add an OpenStack Provider

Use the deployment wizard to add an OpenStack provider.

Prerequisites

Verify that you have the data for provider configuration. See Input Parameters for an OpenStack Provider.

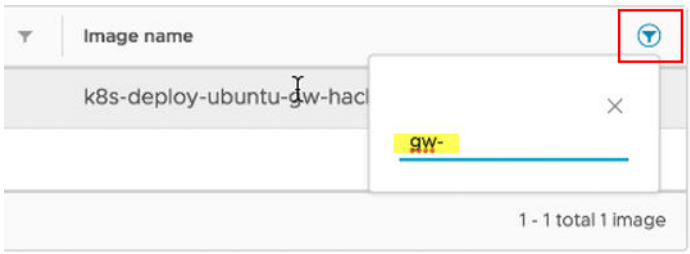
Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Cloud Providers** home page, click **Deploy a New Provider**.
3. On the Intro page, click **Next**.

   Alternatively, you can click **Choose File** to upload a JSON file containing provider information. The information automatically populates fields in the subsequent wizard screens.
4. Specify the provider name and select the **OpenStack** provider type. Click **Next**.
5. Configure OpenStack authentication.
   - Specify the Keystone Public URL.
   - Specify the username, the password, and the Project name.
   - The Region name and the Domain name are optional.

6 Specify the image username. VMware Integrated OpenStack with Kubernetes displays a list of image IDs. Select an ID and click **Next**.

To search for a particular image name, click the filter icon and type a few letters of the image name.



7 Select a flavor for the Kubernetes cluster nodes and click **Next**.

8 Configure the Neutron networking.

- NSX-V networking is the default. No special networking information is required.
- If NSX-T networking is being used, click the **NSX-T Networking** box and add NSX-T networking information.

9 Click **Next**.

10 Select the Security group and click **Next**.

11 Select the External network and click **Next**.

12 Select the Internal network and scroll down to select the Subnet ID. Click **Next**.

13 Review the Configuration Summary and click **Finish** to add the provider.

VMware SDDC Provider

With an SDDC provider, VMware Integrated OpenStack with Kubernetes creates an embedded VMware Integrated OpenStack deployment on an existing vSphere infrastructure. If you configure the embedded VMware Integrated OpenStack deployment for authentication with the local user database, you must add a cluster user on the provider.

An SDDC provider supports VDS, NSX-V, or NSX-T networking.

Network Requirements for SDDC Provider

In addition to system requirements required for installation, your VDS, NSX-V, or NSX-T network must satisfy requirements for an SDDC provider.

VDS Requirements

The following requirements apply to VDS networking:

- One or more vSphere clusters with at least one ESXi host configured.
- Uplinks from all hosts in the active clusters.

- Port group that can access the management network.
- Each cluster requires a port group with IP addresses for:
  - two load balancers and one virtual IP
  - each master node
  - each worker node

A network with multiple clusters requires IP addresses for all port groups.

NSX-V Requirements

The following requirements apply to NSX-V networking:

- vSphere 6.0 or later installed.
- NSX-V 6.2 or later installed and configured.
- One or more vSphere clusters with at least one ESXi host configured in an NSX transport zone.
- The Kubernetes cluster must be in a single NSX transport zone.
- Each cluster requires one VDS-based port group with at least five static IP addresses. A network with multiple clusters requires IP addresses for all port groups.
- Datastore connected to the vSphere cluster

NSX-T Requirements

In addition to the requirements for NSX-V networking, the following requirements apply to NSX-T networking:

- NSX edge installed and configured in NSX-T.
- A range of IP addresses available to avoid conflict in the datacenter.

Input Parameters for an SDDC Provider

This section describes the input parameters required to add an SDDC provider. In addition, NSX-V or NSX-T backend networking require specific configuration parameters. Authentication also requires specific configuration parameters.

An SDDC provider requires the following information.

Table 4-4. vSphere Authentication

Variable	Description
vSphere hostname	FQDN or IP of vCenter server
vSphere username	vCenter server username
vSphere password	vCenter server password
Ignore the vCenter Server certificate validation?	If checked, VMware Integrated OpenStack with Kubernetes does not verify the vCenter Server certificate when connecting to the vCenter.

Table 4-5. vSphere Cluster and Datastore Configuration

Variable	Description
Compute cluster	vSphere compute cluster used to deploy Kubernetes cluster nodes
Datastores	vSphere datastores used to store Kubernetes cluster nodes, images, and volumes

Table 4-6. Management Network Setting for Kubernetes Cluster Nodes

Variable	Description
Port Group	Distributed port group that Kubernetes cluster nodes connect to. Not applicable for NSX-T networking.
VLAN ID (optional)	VLAN ID of the management portgroup. Leave blank if not using VLAN.
Network CIDR	Management network address in CIDR format such as 192.168.0.0/24.
IP Range	Start and end IP addresses of the management network allocation IP range.
Gateway	Gateway IP for the management network
DNS (optional)	DNS servers to be used if DNS for the management network is unavailable. To specify multiple servers, use comma separated values.

Networking Parameters

NSX-V or NSX-T networking requires specific input parameters.

Table 4-7. Configuration Information for NSX-V Networking with SDDC provider

Variable	Description
Manager address	FQDN or IP of the NSX-V manager
Username	NSX-V manager username
Password	NSX-V manager password
Ignore the NSX-V SSL certificate validation?	If checked, VMware Integrated OpenStack with Kubernetes does not verify the NSX-V SSL certificate when connecting to the NSX-V server.
Transport zone	Transport zone configured for NSX-V networking
Edge resource pool	vSphere resource pool for the NSX Edge VMs
Edge datastore	vSphere datastore for NSX Edge VMs
Virtual Distributed Switch	vSphere Distributed Switch configured for NSX-V networking
External network	vSphere distributed port group on the distributed switch

Table 4-8. Configuration Information for NSX-T Networking with SDDC provider

Variable	Description
Manager address	FQDN or IP of the NSX-T manager
Username	NSX-T manager username
Password	NSX-T manager password
Ignore the NSX-T SSL certificate validation?	If checked, VMware Integrated OpenStack with Kubernetes does not verify the NSX-T SSL certificate when connecting to the NSX-T server.
Tier 0 Router	Tier 0 router pre-configured for NSX-T networking
Default overlay transport zone	Overlay transport zone pre-configured for NSX-T networking
Default VLAN transport zone	VLAN transport zone pre-configured for NSX-T networking

Authentication Source Parameters

If you create a standalone user database, VMware Integrated OpenStack with Kubernetes creates a Kubernetes cluster admin user in the database to start. VMware Integrated OpenStack with Kubernetes also supports both Active Directory as an LDAP server on Windows and LDAP server for Unix and Linux.

Table 4-9. Local Admin User Authentication Source

Variable	Description
Kubernetes cluster admin user	Admin user for authentication with the local user database
Kubernetes cluster admin password	Password for authentication with the local user database

Table 4-10. Active Directory as LDAP Backend Authentication Source

Variable	Description	Default
Encryption	SSL or None	None
Hostname	FQDN or IP of the LDAP or AD server	None
Port	Port	636 for SSL 389 for non-SSL
Bind user	LDAP bind user.. Same as Kubernetes cluster admin user.	None
Bind Password	Password for LDAP bind user. Same as Kubernetes cluster admin user.	None
User Tree DN	Search base for users	None
Group Tree DN	Search base for groups	None
User object/class	LDAP objectclass for users	organizationalPerson
User ID attribute	LDAP attribute mapped to user ID. This must not be a multivalued attribute.	cn

Table 4-10. Active Directory as LDAP Backend Authentication Source (Continued)

Variable	Description	Default
User name attribute	LDAP attribute mapped to user name.	userPrincipalName
User mail attribute	LDAP attribute mapped to user e-mail	mail
User password attribute	LDAP attribute mapped to password	userPassword
User enabled attribute	LDAP attribute mapped to user enabled flag	userAccountControl
Group object/class	LDAP objectclass for groups	group
Group ID attribute	LDAP attribute mapped to group ID	cn
Group name attribute	LDAP attribute mapped to group name	sAMAccountName
Group member attribute	LDAP attribute mapped to group member	memberOf
Group description attribute	LDAP attribute mapped to group description	description

Add a VMware SDDC Cloud Provider

Use the deployment wizard to add an SDDC cloud provider.

Prerequisites

Verify that you have the data for provider configuration. See Input Parameters for an SDDC Provider.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Cloud Providers** home page, click **Deploy a New Provider**.
3. On the Intro page, click **Next**.

   Alternatively, you can click **Choose File** to upload a JSON file containing provider information. The information automatically populates fields in the subsequent wizard screens.
4. Specify the provider name and select the **SDDC** provider type. Click **Next**.
5. Specify the vSphere vCenter hostname, username, and password. Leave the **Ignore the vCenter Server certificate validation** option checked. Click **Next**.
6. With the vSphere vCenter information provided, VMware Integrated OpenStack with Kubernetes displays a list of available vSphere clusters. Select a cluster and click **Next**.
7. Select at least one vSphere datastore and click **Next**.

8 Configure networking.

Select VDS, NSX-V, or NSX-T networking and select the network.

9 Configure the management network.

- Select a Port Group.
- VLAN ID is optional.
- Provide the Network Address.
- Provide the IP range.
- Provide the Gateway.
- DNS is optional.

10 Click **Next**

11 Configure the authentication source.

Select **Local Admin User** or **Active Directory as LDAP Backend**.

- For Local Admin User, specify the Kubernetes cluster admin username and password.
- For Active Directory as LDAP backend, see Authentication Source Parameters.

12 Click **Next**

13 Review the Configuration Summary and click **Finish** to add the provider.

Manage Users and Groups on Your SDDC Provider from the UI

If you selected Local Admin User as the authentication source for your SDDC provider, you must add a user for the provider. Later when you create a Kubernetes cluster, you will select this user for the cluster.

Procedure

1 Login to VMware Integrated OpenStack with Kubernetes.

2 On the **Cloud Providers** home page, select an SDDC cloud provider.

3 To manage or add users, click **Users**.

A list of the existing users appears.

- Click **+New** to add a new user with username and password.
- Click the three dots to the right of a user name to delete or edit an existing user.

4 To manage or add groups, click Groups.

A list of the existing groups appears.

- Click **+New** to add a new group and select the users to add to the group.
- Click the three dots to the right of a group name to delete or edit an existing group.

Creating Your First Cluster

5

You create a Kubernetes cluster based on a provider and populate it with master and worker nodes. VMware Integrated OpenStack with Kubernetes supports exclusive and shared cluster types.

This chapter includes the following topics:

- Understanding Cluster Configuration Settings
- Add a New Kubernetes Cluster
- Configure User and Group Access for an Exclusive Cluster
- Create a Namespace for Users and Groups on a Shared Cluster

Understanding Cluster Configuration Settings

After creating a provider, you create a Kubernetes cluster. To configure your cluster correctly, review the cluster configuration settings.

Node Types

A Kubernetes cluster is comprised of two types of nodes. Each node in the VMware Integrated OpenStack with Kubernetes is a VM. Node settings can be changed after cluster deployment.

Master Nodes	A master node provides the Kubernetes API service, scheduler, replicator, and so on. It manages the worker nodes. A cluster with a single master node is valid but has no redundancy.
Worker Nodes	A worker node hosts your containers. A cluster with a single worker node is valid but has no redundancy.

Cluster Types

VMware Integrated OpenStack with Kubernetes supports two types of clusters. The cluster type cannot be changed after cluster deployment.

Exclusive Cluster	In an exclusive cluster, multi-tenancy is not supported. Any authorized Kubernetes user using the Kubernetes CLI of APIs has namespace management privileges.

	The exclusive cluster provides a familiar environment for developers who deploy Kubernetes themselves.
Shared Cluster	In a shared cluster, multi-tenancy is supported and enforced by the Kubernetes namespace. Only a VMware Integrated OpenStack with Kubernetes administrator using the VMware Integrated OpenStack with Kubernetes interface or CLI has namespace management privileges. The shared cluster is an environment where the administrator can manage resource isolation among users.

Add a New Kubernetes Cluster

You deploy a Kubernetes cluster on an OpenStack or SDDC provider. VMware Integrated OpenStack with Kubernetes supports exclusive and shared cluster types.

Prerequisites

- Verify that the cloud provider you deployed is active. See Chapter 4 Adding a Cloud Provider.
- Determine the type of cluster you want to add.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Clusters** home page, click **Deploy a New Cluster** or click **+NEW** to add a cluster to a set of existing clusters.
3. On the Intro page, click **Next**.

   Alternatively, you can click **Choose File** to upload a JSON file containing cluster configuration information. The information automatically populates fields in the subsequent wizard screens.
4. Highlight the infrastructure provider and click **Next**.

5 On the Configure node group page, configure the master and worker node groups.

a Specify the number of master nodes in the cluster. The default node profile is selected. To select a different node profile, uncheck the box.

Note Because etcd servers coexist on master nodes, you must specify an odd number of nodes to support high availability and fault tolerance.

A table appears listing a selection of node profiles.

b Specify the number of worker nodes in the cluster. The default node profile is selected. To select a different node profile, uncheck the box.

A table appears listing a selection of node profiles.

Note If the group includes multiple nodes, and you want to use a worker node configured with an extra network in the ENS transport zone, select the node profile labeled `node_profile=high_performance`. To use this network in a pod deployment with multiple NICs, you must assign your pod to this node by adding the following field setting to your pod configuration:

```
nodeSelector:
    node_profile: high_performance
```

c Click **Next**.

For more information about node profiles, see Customizing Your Cluster.

6 Configure the cluster.

a Provide a cluster name.

b (optional) Specify the DNS servers.

c (optional) Specify the IP address or FQDN for the Log Insight server configured to receive logs from the Kubernetes cluster.

d Select the **Shared Cluster** or **Exclusive Cluster** type.

- For a shared cluster, specify a namespace.
- For an exclusive cluster, specify a user and group.

e Provide information to create an NSX-T external network IP pool with addresses for the cluster.

Note If the infrastructure provider is configured with NSX-T networking, every Kubernetes cluster should use its own pool. NSX-T Backend.

- For the external IP pool, specify a CIDR value with a network address and not a host address. For example, 4.3.0.0/16.
- Specify the range of addresses to be allocated for the IP pool.
- (optional) Specify the gateway for the IP pool.
- (optional) Specify the DNS server IP.

7 Click **Next**.

8 Review the Configuration Summary and click **Finish** to add the cluster

Configure User and Group Access for an Exclusive Cluster

Once a Kubernetes cluster is created, you can authorize users or groups for the cluster. The users and groups belong to the SDDC or OpenStack provider where the cluster was created.

Procedure

1 Login to VMware Integrated OpenStack with Kubernetes.

2 On the **Clusters** home page, click the three dots to the right of an existing exclusive cluster and select **+Configure user & group**.

3 In the **Configure user and group for cluster** dialogue box, check the boxes for users or groups that you want to authorize for access to the cluster. Or check off the boxes for users or groups that you no longer want to authorize for access to the cluster.

4 Click **OK**.

Create a Namespace for Users and Groups on a Shared Cluster

The shared cluster has restricted access with multi-tenancy support based on the Kubernetes namespace. An administrator can create a namespace and authorize access to users or groups specified in a namespace policy.

Procedure

1 Login to VMware Integrated OpenStack with Kubernetes.

2 On the **Clusters** home page, click the three dots to the right of an existing shared cluster and select **+Add namespace**.

3 In the **Add a new namespace** dialogue box, type a name for the namespace and check the boxes for users or groups that you want to authorize for access to the namespace.

4 Click **OK**.

Managing Your Deployment

6

Once you create your provider and cluster, you can use VMware Integrated OpenStack with Kubernetes to manage your deployment.

This chapter includes the following topics:

- Customizing Your Cluster
- Cluster Management
- Increasing Capacity
- Monitoring Your Kubernetes Cluster
- Certificate Management Using the CLI
- Collect Logs from SDDC Provider Services
- Add Helm to Your VMware Integrated OpenStack with Kubernetes Deployment
- Upgrade to VMware Integrated OpenStack with Kubernetes 5.0
- Upgrade Clusters to Support Current Kubernetes Version

Customizing Your Cluster

After adding a cloud provider, you may want to create a new cluster by customizing the node's OS image, the VM hardware resources, or the network settings. A node profile is an optional resource that you can create on a cloud provider to customize the hardware and infrastructure settings used by the nodes of a Kubernetes cluster.

For example, if you deploy a cluster on an SDDC provider, the nodes of the cluster may be deployed with a default Operating System such as Ubuntu 16.04 and the following hardware configuration:

- 4 vcpu's
- 8 GB memory
- 40 GB hard disk

Using a node profile allows you to customize these settings for your new clusters.

You can create multiple node profiles. If no node profile is created, default settings from the cloud provider are applied when creating the cluster.

Add a Node Profile For Your OpenStack Provider

Use the interactive wizard to add a new node profile for your OpenStack cloud provider.

Prerequisites

Verify that the OpenStack cloud provider you deployed is active. See Add an OpenStack Provider.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Cloud Providers** home page, select an OpenStack cloud provider.
3. On the **Node Profiles** tab, click **+New**.
4. To customize the node profile, enter infrastructure settings.

   If you skip any setting, default values from the OpenStack provider are used. See Input Parameters for an OpenStack Provider.

Variable	Description
Image username	Used to establish SSH connection with cluster nodes. This user must be able run `sudo` without a password. For example, the default user for Ubuntu cloud operating systems is **`ubuntu`**.
Flavor ID	OpenStack flavor ID.
Internal Network ID	Internal network ID used for nodes IPs
Internal Subnet ID	Subnet ID of the internal network used for allocating the IPs
External Network ID	External network ID used to assign floating IPs
Configure Extra Network	If the provider is configured for NSX-T networking, this option appears. Click to display a list of available extra networks and extra network subnets. ■ Select **Regular** or **High-Performance**. ■ From the list of extra networks found, select the network. Extra networks for enhanced networking services are configured with VMware Integrated OpenStack. For more information, see Expose Multiple Availability Zones for Enhanced Networking Services in the *Administering VMware Integrated OpenStack* guide.

5. Click **OK**.

Add a Node Profile For Your SDDC Provider

Use the interactive wizard to add a new node profile for your SDDC cloud provider.

Prerequisites

Verify that the SDDC cloud provider you deployed is active. See Add a VMware SDDC Cloud Provider.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Cloud Providers** home page, select an SDDC cloud provider.

3 On the **Node Profiles** tab, click **+New**.

4 To customize the node profile, enter infrastructure settings.

Settings are grouped into three categories. In each category, you must specify all or none of the values.

a Specify the hardware settings.

If left unspecified, default values are applied.

Variable	Description
vCPU	The number of vCPUs to be allocated to each cluster node VM. Default value is 2.
Memory (MB)	The memory to be allocated to each cluster node VM. Default value is 4096.
Disk (GB)	The hard disk size to be allocated to each cluster node VM. Default value is 40.

b Specify the operating system settings.

If left unspecified, a default Operating System running Ubuntu 16.04 is used to create the nodes of the cluster.

Variable	Description
VM Template	The VM Template used for the cluster nodes. The VM template must be for a supported cloud-ready operating system.
VM Template User	Used to establish SSH connection with cluster nodes. This user must be able run `sudo` without a password. For example, the default user for Ubuntu cloud operating systems is **`ubuntu`**.

c Specify the management network settings.

If left unspecified, parameters used when creating the VMware SDDC provider are applied. See Input Parameters for an SDDC Provider.

Variable	Description
Network CIDR	Management network address in CIDR format such as 192.168.0.0/24.
Management network gateway	Gateway IP for the management network
Management IP Range start	Start IP address of the management network allocation IP range.
Management IP Range end	End IP address of the management network allocation IP range.

5 Click **OK**.

Cluster Management

After creating your Kubernetes cluster, you may want to expand the cluster.

Scale Your Cluster

If the cluster you added is not large enough, you can increase the number of Kubernetes worker nodes to increase capacity. You can also decrease the number of worker nodes.

Prerequisites

Verify that the state of the Kubernetes cluster you want to scale is ACTIVE.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Clusters** home page, click the three dots to the right of a cluster name and select **Scale Cluster**.
3. In the **Scale cluster** dialogue box, enter the desired number of worker nodes for the cluster.
4. Click **OK**

Delete Your Cluster

If a cluster is no longer needed, you can delete it to free resources for another use.

Procedure

1. Login to VMware Integrated OpenStack with Kubernetes.
2. On the **Clusters** home page, click the three dots to the right of a cluster name and select **Delete Cluster**.
3. Click **OK** to confirm.

Increasing Capacity

To increase the capacity of your Kubernetes cluster, you can add cluster storage or add a new vSphere cluster.

Add Capacity with an OpenStack Provider

With an OpenStack provider, contact your vSphere administrator to add capacity.

Add Capacity with an SDDC Provider

With an SDDC provider, you use VMware Integrated OpenStack with Kubernetes to add capacity to your Kubernetes cluster.

If you have storage space on a vSphere cluster, you can add datastores to your Kubernetes cluster. If you do not have space on an existing vSphere cluster, you can add a vSphere cluster.

Procedure

1. Log in to VMware Integrated OpenStack with Kubernetes.
2. On the **Cloud Providers** home page, select a provider name.
   - To add storage, click **Datastores**.
   - To add a vSphere cluster, click **vSphere Clusters**.

Monitoring Your Kubernetes Cluster

Kubernetes master nodes and worker nodes run as VMs on vSphere. To use the general vSphere monitoring mechanism to monitor activity on these VMs, you must know how to identify the Kubernetes nodes.

- Names of VMs for Kubernetes master nodes start with

  `k8s-master-<sequence number>-<Cluster ID>`

- Names of VMs for Kubernetes worker nodes start with

  `k8s-node-<sequence number>-<Cluster ID>`

To find the Cluster ID, you use the vkube CLI or by clicking the URL under the cluster name in the UI.

A Kubernetes dashboard installed with VMware Integrated OpenStack with Kubernetes displays resources available to the cluster and namespaces. In addition, you can configure external monitoring tools to collect more data from the Kubernetes cluster than what the dashboard provides.

Use the Kubernetes Dashboard to Monitor Your Cluster

When you create a Kubernetes cluster, VMware Integrated OpenStack with Kubernetes installs a Kubernetes dashboard and Heapster

To access the Kubernetes dashboard for your Kubernetes cluster, find the endpoint for your cluster. In your browser, type **<endpoint_IP>/ui**, for example `https://192.168.112.194:443/ui`.

You can also access the Kubernetes dashboard directly from the VMware Integrated OpenStack with Kubernetes interface.

Procedure

1. On the cluster list view, click the URL under cluster name.
2. On the cluster detail view, click the Dashboard icon on top right.
3. When prompted for authentication, enter the username and password.
   - For an Exclusive Cluster, any user can use the Kubernetes dashboard.
   - For a Shared Cluster, only the admin user can use the Kubernetes dashboard.

The Kubernetes dashboard displays resources available to the whole cluster such as namespaces and nodes. It also displays resources available to a namespace such as deployments and pods. If you specify app details or upload a YAML or JSON file, you can deploy a Containerized App.

Deploying Grafana for Heapster

Heapster is an external tool that monitors container clusters and provides detailed resource usage information about applications running on a Kubernetes cluster.

For general information about Heapster, see https://github.com/kubernetes/heapster.

Heapster and Influx DB as a backend are deployed by default during cluster creation. You can use the following YAML file to deploy the Grafana UI as a visualization dashboard.

```
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: monitoring-grafana
  namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: grafana
    spec:
      containers:
      - name: grafana
        image: gcr.io/google_containers/heapster-grafana-amd64:v4.0.2
        ports:
          - containerPort: 3000
            protocol: TCP
        volumeMounts:
        - mountPath: /var
          name: grafana-storage
        env:
        - name: INFLUXDB_HOST
          value: monitoring-influxdb
        - name: GRAFANA_PORT
          value: "3000"
          # The following env variables are required to make Grafana accessible via
          # the kubernetes api-server proxy. On production clusters, we recommend
          # removing these env variables, setup auth for grafana, and expose the grafana
          # service using a LoadBalancer or a public IP.
        - name: GF_AUTH_BASIC_ENABLED
          value: "false"
        - name: GF_AUTH_ANONYMOUS_ENABLED
          value: "true"
        - name: GF_AUTH_ANONYMOUS_ORG_ROLE
          value: Admin
        - name: GF_SERVER_ROOT_URL
          # If you're only using the API Server proxy, set this value instead:
          # value: /api/v1/proxy/namespaces/kube-system/services/monitoring-grafana/
          value: /
      volumes:
      - name: grafana-storage
        emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  labels:
    # For use as a Cluster add-on (https://github.com/kubernetes/kubernetes/tree/master/cluster/addons)
```

```
    # If you are NOT using this as an addon, you should comment out this line.
    kubernetes.io/cluster-service: 'true'
    kubernetes.io/name: monitoring-grafana
  name: monitoring-grafana
  namespace: kube-system
spec:
  # In a production setup, we recommend accessing Grafana through an external Loadbalancer
  # or through a public IP.
  # type: LoadBalancer
  # You could also use NodePort to expose the service at a randomly-generated port
  # type: NodePort
  ports:
  - port: 80
    targetPort: 3000
  selector:
    k8s-app: grafana
```

In a production environment, uncomment `type: LoadBalancer` or `type: NodePort` to expose the service externally.

Deploying Prometheus

Prometheus is a general purpose monitoring tool that can also collect and analyze metrics from applications running on a Kubernetes cluster.

- Prometheus is available from https://prometheus.io/
- The Prometheus Operator makes the Prometheus configuration Kubernetes native and manages Prometheus instances on top of Kubernetes. For information about the Prometheus Operator project, see https://github.com/coreos/prometheus-operator.
- kube-prometheus is a repository in the Prometheus Operator project. It includes manifests, Grafana dashboards, and Prometheus rules, documentation, and scripts, to provide single-command deployments of end-to-end Kubernetes cluster monitoring with Prometheus. For information about kube-prometheus, see https://github.com/coreos/prometheus-operator/tree/master/contrib/kube-prometheus.

To deploy Prometheus instances that are managed by the Prometheus Operator, use the assets in kube-prometheus repository.

Certificate Management Using the CLI

You use self-signed certificates in VMware Integrated OpenStack with Kubernetes for authentication. In some cases, certificates may need to be modified and regenerated.

Manage SSL Certificates for Kubernetes API Servers

If Kubernetes API servers are accessed over a public internet, you may want to use a certificate signed by a trusted certificate authority (CA) to further secure your Kubernetes deployment.

You can use VMware Integrated OpenStack with Kubernetes CLI to prepare a certificate signing request. After the CA generates a signed certificate, you can use the CLI to upload it to a target Kubernetes cluster.

Prerequisites

If application pods are deployed already and these pods use the Kubernetes secret tokens, back up application data and remove the pods before updating the API server certificate. The certificate update invalidates the secret tokens, so you must re-create the pods following the update.

Procedure

1 Login as root to the VMware Integrated OpenStack with Kubernetes VM. Provide the root password set during OVA deployment.

```
vkube login --insecure
```

2 Generate a certificate signing request.

```
vkube cluster list --insecure
vkube cluster csr <cluster_id>
   --country-name <value1>
   --locality-name <value2>
   --organization name <value3>
   --organizatio-unit-name <value4>
   --state-name <value5>
   --insecure
```

3 Copy the existing Kubernetes `/etc/kubernetes/openssl.conf` file from the Kubernetes Master0 node and send it with the Certificate Signing Request (CSR) file to your company's CA administrator.

4 Using the CSR file and the extfile from openssl.conf, the CA administrator generates a signed certificate. Upload the API server's certificate and corresponding CA certificate to the Kubernetes cluster.

```
vkube cluster crt <cluster-id> --insecure --ca-file-name ca.pem --crt-file-name apiserver.pem
```

5 Login to the Master0 node and type:

```
kubectl get pod --namespace=kube-system
```

When all the pods change to status running, the cluster is ready to use.

Collect Logs from SDDC Provider Services

If you encounter system errors, you can send logs collected from the SDDC provider to the product support team.

Use the following procedure to collect container logs from VMware Integrated OpenStack with Kubernetes.

Procedure

1 Log in to VMware Integrated OpenStack with Kubernetes as root.

2 Run the command to collect logs.

```
log_collect_VM [–S] [–U] [–n] [–f] [–h]
```

Option	Description
–S	Collect log file data since date YYYY-MM-DD
–U	Collect log file data until date YYYY-MM-DD
–n	Name of the container from which to collect the log
–f	File name in tar format to which logs are saved. Default is: `blueshift_.tag.gz`.
–h	Show the use and arguments for this command and exit.

Add Helm to Your VMware Integrated OpenStack with Kubernetes Deployment

Helm is a tool that streamlines the installation and management of Kubernetes applications. When you deploy Kubernetes applications with Helm initialized in your cluster, you can use Helm to version, upgrade, and rollback those deployments.

Helm operates on packages of configured Kubernetes resources. Tiller is the Helm server that runs in Kubernetes and handles the Helm packages. To add Helm to your VMware Integrated OpenStack with Kubernetes deployment, you initialize Helm and install the Tiller server in the Kubernetes cluster.

Prerequisites

Verify that VMware Integrated OpenStack with Kubernetes 4.1 or later is installed.

Procedure

1 Log in to VMware Integrated OpenStack with Kubernetes.

2 On the **Clusters** home page, identify the target cluster where you want to add Helm. Click the three dots to the right of the cluster name and select **Download kubectl config file**.

3 Using SSH, log in to VMware Integrated OpenStack with Kubernetes VM.

   a Copy the downloaded config file to `/root/.kube/config`.

   b In the config file, edit the `username` and `password` to match the username and password for the target cluster.

```
username: "<CLUSTER_USERNAME>"
password: "<CLUSTER_PASSWORD>"
```

   c Type the command to initialize Helm.

```
helm init
```

Upgrade to VMware Integrated OpenStack with Kubernetes 5.0

VMware Integrated OpenStack with Kubernetes 5.0 supports Kubernetes version 1.9.8. You run a script to upgrade your VMware Integrated OpenStack with Kubernetes 4.1 deployment.

Prerequisites

- Verify that VMware Integrated OpenStack with Kubernetes 4.1 is installed and running.
- Obtain the VMware Integrated OpenStack with Kubernetes upgrade tar file. Go to downloads for VMware Integrated OpenStack 5.0 from https://www.vmware.com/go/download-openstack.

Procedure

1. Log in to the VMware Integrated OpenStack with Kubernetes VM.
2. Untar the downloaded tar.gz file.

   ```
   tar –xzf upgrade–<build_number>.tar.gz
   ```

   The folder `upgrade–xxxxx` is created.
3. In the `upgrade–xxxxx` folder, run the script.

   ```
   cd upgrade–xxxxx
   ./install.sh
   ```

The script loads a new set of Docker images on the VM and updates the VMware Integrated OpenStack with Kubernetes services.

Upgrade Clusters to Support Current Kubernetes Version

VMware Integrated OpenStack with Kubernetes 4.1 supports Kubernetes version 1.8.1. You can upgrade clusters created using VMware Integrated OpenStack with Kubernetes 4.1 to be compatible with Kubernetes version 1.9.8.

The cluster upgrade is optional. For example, you do not need to upgrade a cluster to perform scale or delete operations using VMware Integrated OpenStack with Kubernetes 5.0.

Note VMware Integrated OpenStack with Kubernetes 4.1 clusters that have been upgraded to be compatible with Kubernetes 1.9.8 do not support new features in VMware Integrated OpenStack with Kubernetes 5.0.

Prerequisites

- Verify that VMware Integrated OpenStack with Kubernetes has been upgraded to 5.0.
- Verify that the cluster has more than one worker node.

Procedure

1. Log in to the VMware Integrated OpenStack with Kubernetes VM.
2. Run the CLI command.

   ```
   vkube cluster upgrade <ID>
   ```

   Where *ID* is the cluster ID.

After upgrading, the cluster appears with Kubernetes version 1.9.8.

Optimizing Kubernetes Cluster Performance

7

To ensure optimal Kubernetes cluster performance, you should follow certain best practices. This section highlights some of the key best practices.

Setting Adequate Quotas in OpenStack

For an OpenStack provider, set quotas that are large enough to accommodate a large cluster.

Table 7-1. Sample Commands

Command	Description
`nova quota-update --key-pairs 500 --instances 500 --cores 4000 --ram 12288000 <tenant_ID>`	Set quotas for a 500-node cluster, where each node has 8 vCPUs and 24G RAM
`neutron quota-update --tenant-id <tenant_ID> --pool 300 --port 1000 --loadbalancer 300 --floatingip 150`	Neutron command to allocate quota according to your network. Port number should be greater than instance plus load balancer number.
`cinder quota-update --volumes 500 --gigabytes 5000 <tenant_ID>`	Cinder command to allocate quota according to the number of persistent volumes that you want to create.

Best Practices for Creating Large Clusters

To create a large cluster, a best practice is to first create a small cluster, then scale it out. For example, to create a stable 500-node cluster, start by creating a 30-node cluster, then scale it out with a maximum of 30 nodes at a time until you reach 500 nodes.

Tips:

- If your cluster is larger than 200 nodes, you might see RPC timeouts in the OpenStack service logs. If that occurs, increase the RPC timeout setting for those services. For example for a Nova service, increase the value of the rpc_response_timeout configuration option in the nova.conf file.

- It may take time to refresh the status of created resources when scaling out a cluster. Add the --skip-refresh option to the `vkube cluster scaleout` command to decrease the deployment time. With this option, the scale out operation does not check the state of existing resources such as VMs or load balancers, and assumes that the resources are successfully created.

Managing High CPU Usage with an OpenStack Provider

If you are using VMware Integrated OpenStack deployed in compact mode as your OpenStack provider, you may notice high CPU usage on the controller or compute service VM's. If so, increase the number of vCPU's to 16 per VM.

Alternatives to Load Balancing with NSX-V Backing

When you create services in Kubernetes and you specify the type as LoadBalancer, NSX Edge load balancers are deployed for every service. The load balancer distributes the traffic to all Kubernetes worker nodes up to 32 members. If your Kubernetes cluster includes more that 32 worker nodes, use the Kubernetes Ingress resource instead.

Persistent Volume Claim Management

If you create many persistent volume claims and associated pods in parallel, you should use VMware Integrated OpenStack in HA mode. VMware Integrated OpenStack in compact mode may not provide enough capacity to handle the large number of incoming API requests.

If dynamic provisioning of persistent volumes fails even with VMware Integrated OpenStack in HA mode, check the OpenStack service logs to see if the failures are due to RPC timeouts and increase the RPC timeout setting for those services. For example for a Nova service, you can increase the value of the rpc_response_timeout configuration option in the nova.conf file.

Best Practice for Configuring an SDDC Provider with LDAP

When configuring an SDDC provider with LDAP, a best practice is to set the filters for the LDAP user and group ensure that each filter is returning fewer than 1000 users or groups. If the limit is exceeded, the query returns no result and reports an error.

Best Practices for Backup

If a cluster fails or the VMware Integrated OpenStack with Kubernetes dashboard become unresponsive, the following best practices will help to ensure that you can always restore your configuration.

- After deploying VMware Integrated OpenStack with Kubernetes, create a snapshot of your VMware Integrated OpenStack with Kubernetes VM on the vCenter server.
- When a provider or a cluster is added or deleted, create a backup.

- After changing the configuration of an application running on a cluster, backup the configuration of the application.
- Move backups from the VMware Integrated OpenStack with Kubernetes VM to external storage.
- Regularly delete unneeded backup files to ensure adequate storage space on the VMware Integrated OpenStack with Kubernetes VM.

Troubleshooting Your VMware Integrated OpenStack with Kubernetes Infrastructure

8

If you encounter problems with your VMware Integrated OpenStack with Kubernetes deployment, you can use CLI commands to restore your backups.

This chapter includes the following topics:

- Troubleshoot Dashboard Load Failure
- Troubleshoot Cluster Update Failure

Troubleshoot Dashboard Load Failure

The VMware Integrated OpenStack with Kubernetes dashboard is continually loading or the user interface becomes unresponsive.

Cause

The VMware Integrated OpenStack with Kubernetes VM is not reachable.

Solution

Perform each subsequent step in the procedure only if the previous step fails to reestablish a connection with the VMware Integrated OpenStack with Kubernetes VM.

Procedure

1. Reboot the VMware Integrated OpenStack with Kubernetes VM.
2. If the VM is still not reachable, restore the VM using a snapshot stored on the vCenter server.
   a. Restore the VM with a backup.
   b. If using an SDDC provider, refresh the provider.
3. If the VM is still not reachable, redeploy VMware Integrated OpenStack with Kubernetes.
   a. Power off the VM.
   b. Redeploy VMware Integrated OpenStack with Kubernetes using the same IP address.
   c. Restore the VMware Integrated OpenStack with Kubernetes backup.

Troubleshoot Cluster Update Failure

After running `vkube cluster update`, the cluster fails to update and appears to be in an ERROR state.

Cause

Master nodes or worker nodes in the cluster have failed.

Solution

To troubleshoot a cluster failure, refresh the cluster infrastructure. The same troubleshooting procedure applies when:

- Cluster dashboard is unresponsive or not reachable.
- kubectl fails to connect to the cluster.

Perform each subsequent step in the procedure only if the previous step fails to bring the cluster out of its error state.

Procedure

1. Refresh the cluster infrastructure.

   ```
   vkube cluster heal <cluster_id>
   ```

2. If the cluster remains in an ERROR state with the error `Recreation of 2 master node(s) exceeded the maximum of 1 out of 3 master node(s) allowed`, verify that you have a recent backup of the cluster ready.
3. List all the nodes in the cluster.

   ```
   vkube cluster show <cluster_id>
   ```

   The output is a comma-separated list of nodes in the cluster.
4. Refresh the master nodes in the cluster.

   ```
   vkube cluster heal <cluster_id> --nodes <k8s-master_node1, k8s-master_node2, ...>
   ```

   Where *k8s-master_node1, k8s-master_node2, ...* are nodes in the comma-separated list that begin with `k8s-master`.

5 Restore the configurations of applications running on the cluster.

```
vkube job cluster recover
```

Note For applications running on Kubernetes that are deployed using a daemon set with service account, perform the following additional steps:

a Delete the service account and service.

b Redeploy the service account and service.

c If you are using local storage, restore your application data.

Stop Kubelet in Worker Nodes

The method to stop kubelet in worker nodes varies depending on the host configuration.

Host configuration without Bastion

Stop kubelet when the host configuration does not include Bastion.

1 Using SSH, log in to the VMware Integrated OpenStack with Kubernetes VM.

2 Run the command: `docker exec –it app–api bash`

3 Knowing the cluster ID, change to the directory: `cd /var/lib/vrs/terraform/<cluster_id>`

4 For each worker node IP, run:

- `ssh –i private.key ubuntu@<node_ip>`
- `systemctl stop kubelet`

Host Configuration with Bastion

Stop kubelet when the host configuration includes Bastion.

1 Using SSH, log in to the VMware Integrated OpenStack with Kubernetes VM.

2 Run the command: `docker exec –it app–api bash`

3 Knowing the cluster ID, change to the directory: `cd /var/lib/vrs/terraform/<cluster_id>`

4 Perform the following steps for each worker node IP.

- On the Bastion host, run:

  ```
  ssh –i private.key ubuntu@<node_ip>
  ```

- On other working nodes, run:

  ```
  ssh –i private.key –F ssh–bastion.conf ubuntu@<node_ip>
  ```

- Run: `systemctl stop kubelet`

Without Bastion Host

Stop kubelet without a Bastion host.

1 Using SSH, log in to the VMware Integrated OpenStack with Kubernetes VM.

2 Use the OpenStack client to assign a floating IP to each worker node.

3 Follow the procedure for Host configuration without Bastion.

4 Use the OpenStack client to unassign the floating IP assigned to each worker node.

VMware Integrated OpenStack with Kubernetes CLI Command Reference

9

The VMware Integrated OpenStack with Kubernetes CLI commands have specific syntax requirements.

To run the CLI commands, log on to the VMware Integrated OpenStack with Kubernetes control VM.

This chapter includes the following topics:

- vkube cluster update Command
- vkube cluster heal
- vkube job backup Command
- vkube job restore Command
- vkube job recover Command
- vkube job get Command
- vkube job list Command
- vkube nodegroup Command

vkube cluster update Command

Use the `vkube cluster update` command to refresh the cluster infrastructure in case of infrastructure failure.

The `vkube cluster update` command uses the following syntax.

```
vkube cluster update [--cluster-id CLUSTER_ID]
```

Parameter	Mandatory or Optional	Description
`--cluster-id` *CLUSTER_ID*	Mandatory	ID of the cluster to recover

If the cluster update succeeds, you can proceed to recover the cluster. See vkube job recover Command.

vkube cluster heal

If an OpenStack instance is in an error state, use the `vkube cluster heal` command to re-create a cluster with specific nodes.

The `vkube cluster heal` command uses the following syntax.

```
vkube cluster heal <CLUSTER_ID>[--nodes NODE_1, NODE_2...] [--show-vars]
```

Parameter	Mandatory or Optional	Description
CLUSTER_ID	Mandatory	ID of the cluster to re-create
`--nodes` *NODE_1, NODE_2,...*	Mandatory	Comma-separated list of nodes in the cluster. If unspecified, the cluster re-creation fails with an error.
`--show-vars`	Optional	Print environment variables used to heal the cluster.

If the cluster update succeeds, you can proceed to recover the cluster. See vkube job recover Command.

vkube job backup Command

Use the `vkube job backup` command to create a backup configuration of all VMware Integrated OpenStack with Kubernetes providers and clusters for each provider. You also have the option to backup the configuration of applications running on each cluster.

For a list of cases where the backup command is used, see Best Practices for Backup.

The `vkube job backup` command uses the following syntax.

```
vkube job backup [--description DESCRIPTION] [--cluster-ids CLUSTER_ID1,CLUSTER_ID2...]
```

Parameter	Mandatory or Optional	Description
`--description` *DESCRIPTION*	Mandatory	Description of the backup.
`--cluster-ids`	Optional	The clusters to backup. If unset, backup the provider only.

The backup process creates a tar file that can be used to restore the backup. See vkube job restore Command.

vkube job restore Command

Use the `vkube job restore` command to restore VMware Integrated OpenStack with Kubernetes provider and cluster configurations.

Note Restore is a replacement operation. So any changes made to the configuration since the time of the backup are lost during a job restore.

Before restoration, a best practice is to create a snapshot of the VMware Integrated OpenStack with Kubernetes VM. Since backups are stored on the VM, copy the backup to external storage in case you need to restore from a failure of the VM itself.

The `vkube job restore` command uses the following syntax.

```
vkube job restore [--description DESCRIPTION] [--restore-tar BACKUP_TAR_FILE_PATH]
```

Parameter	Mandatory or Optional	Description
`--description` *DESCRIPTION*	Mandatory	Description of the restore.
`--restore-tar` *BACKUP_TAR_FILE_PATH*	Mandatory	Path pointing to the backup tar file obtained from backup. See vkube job backup Command.

If the VMware Integrated OpenStack with Kubernetes deployment has an SDDC provider, the SDDC provider is put into a DATA RESTORE status. To make the provider active, run

```
vkube provider refresh sddc <PROVIDER_ID>
```

vkube job recover Command

Use the `vkube job recover` command to recover the configurations of all applications running on a cluster.

Note Recover is a replacement operation. So any changes made to the configuration since the time of the backup are lost during a job recover.

If more than half of master nodes have failed, run `vkube job recover` only after successfully updating the cluster infrastructure. See vkube cluster update Command.

The `vkube job recover` command uses the following syntax.

```
vkube job recover [--description DESCRIPTION] [--restore-tar BACKUP_TAR_FILE_PATH] [--cluster-id
CLUSTER_ID]
```

Parameter	Mandatory or Optional	Description
`--description` *DESCRIPTION*	Mandatory	Description of the recovery
`--restore-tar` *BACKUP_TAR_FILE_PATH*	Mandatory	Path pointing to the backup tar file obtained from backup.
`--cluster-id` *CLUSTER_ID*	Mandatory	The cluster to recover.

vkube job get Command

All backup, restore, or recover commands return a job ID. Use the `vkube job get` command to get the job.

The `vkube job get` command uses the following syntax.

```
vkube job get <JOB_ID>
```

Parameter	Mandatory or Optional	Description
JOB_ID	Mandatory	ID of the job to get.

vkube job list Command

Use the `vkube job list` command to list all backup, restore, or recover jobs.

The `vkube job list` command uses the following syntax.

```
vkube job list [--filter JOB_TYPE]
```

Parameter	Mandatory or Optional	Description
`--filter`*JOB_TYPE*	Mandatory	Filters for jobs of a particular type and lists all those jobs. The valid types are: **`backup`**, **`restore`**, **`recover`**, .

vkube nodegroup Command

Use the `vkube nodegroup` command to create and delete node groups for a cluster.

The `vkube nodegroup` command uses the following syntax.

```
vkube nodegroup ACTION <CLUSTER_ID> <NODEGROUP_ID>\
   [--name NAME] \
   [--node-count NODE_COUNT] \
   [--node-profile-id NODE_PROFILE_ID] \
   [--type TYPE]
   [--skip-refresh SKIP]
```

Parameter	Mandatory or Optional	Description	
`ACTION` Use one of the following positional arguments: ■ `create` ■ `delete` ■ `list` ■ `scale`	Mandatory	**`create`**	Create a node group on cluster.
		`delete`	Delete a node group from cluster.
		`list`	List available node groups per cluster.
		`scale`	Scale a node group from cluster.
CLUSTER_ID	Mandatory for `create`, `delete`, `list` and `scale` actions	ID of the cluster for the node group.	
NODEGROUP_ID	Mandatory for `delete` and `scale` actions	ID of the node group.	
`-name` *NAME*	Mandatory for `create` action	Name of node group.	
`--node-count` *NODE_COUNT*	Mandatory for `create` and `scale` action	Number of nodes.	

Parameter	Mandatory or Optional	Description
`--node-profile-id` *NODE_PROFILE_ID*	Mandatory for `create` action	Node profile ID of the node group.
`--type` *TYPE*	Mandatory for `create` action	Type of node. `Worker` is the supported node type.
`--skip-refresh` *SKIP*	Optional for `scale` action	Boolean value to refresh or skip refresh of back end resources.