

Pivotal Container Service (PKS)

Version 1.1

Published: 29 June 2018

Table of Contents

Table of Contents	2
Pivotal Container Service (PKS)	4
PKS Release Notes	6
PKS Concepts	9
PKS Cluster Management	10
PKS API Authentication	13
Load Balancers in PKS	14
Prerequisites	16
Installing the PKS CLI	17
Installing the Kubernetes CLI	19
Preparing to Install PKS on vSphere	21
vSphere Prerequisites and Resource Requirements	22
Firewall Ports and Protocols Requirements for vSphere with NSX-T	24
Preparing to Deploy PKS on vSphere	26
Deploying Ops Manager to vSphere	33
Configuring Ops Manager on vSphere	37
VMware Harbor Registry	50
Preparing to Install PKS on GCP	52
GCP Prerequisites and Resource Requirements	53
Preparing to Deploy PKS on GCP	54
Deploying Ops Manager to GCP	59
Configuring Ops Manager on GCP	61
Configuring a GCP Load Balancer for the PKS API	73
Configuring a GCP Load Balancer for PKS Clusters	75
Installing PKS	78
Installing and Configuring PKS	79
Installing and Configuring PKS with NSX-T Integration	98
Managing PKS	115
Configure PKS API Access	116
Manage Users in UAA	118
Configure Authentication and Enterprise SSO	121
Manage PKS Deployments with BOSH	122
Add Custom Workloads	123
Download Cluster Logs	124
Maintain Workload Uptime	125
Upgrade PKS	127
Configure the Upgrade Pipeline	130
Service Interruptions	131
Delete PKS	134
Using PKS	135
Create a Cluster	136
Retrieve Cluster Credentials and Configuration	139
View Cluster List	140
View Cluster Details	141
View Cluster Plans	142
Using Dynamic Persistent Volumes	143
Scale Existing Clusters	144
Access the Dashboard	145

Deploy and Access Basic Workloads	146
Delete a Cluster	148
Log Out of the PKS Environment	149
Using Helm with PKS	150
Configure Tiller	151
Install Concourse Using Helm	152
Backing Up and Restoring PKS	153
Install BOSH Backup and Restore	154
Back Up the PKS Control Plane	155
Restore the PKS Control Plane	158
BBR Exit Codes and Logging	161
PKS Security	162
PKS Security Disclosure and Release Process	163
Diagnosing and Troubleshooting PKS	164
Diagnostic Tools	165
Troubleshooting	166
PKS CLI	170

Pivotal Container Service (PKS)

Pivotal Container Service (PKS) enables operators to provision, operate, and manage enterprise-grade Kubernetes clusters on Pivotal Cloud Foundry (PCF).

Overview

PKS uses the [On-Demand Broker](#) to deploy [Cloud Foundry Container Runtime](#), a BOSH release that offers a uniform way to instantiate, deploy, and manage highly available Kubernetes clusters on a cloud platform using BOSH.

After operators install the PKS tile on the Ops Manager Installation Dashboard, developers can provision Kubernetes clusters using the PKS Command Line Interface (PKS CLI), and run container-based workloads on the clusters with the Kubernetes CLI, [kubectl](#).

PKS is available as part of [Pivotal Cloud Foundry](#) or as a stand-alone product.

Features

PKS has the following features:

- **Kubernetes Compatibility:** Constant compatibility with current stable release of Kubernetes
- **Production-ready:** Highly available from applications to infrastructure, with no single points of failure
- **BOSH advantages:** Built-in health checks, scaling, auto-healing and rolling upgrades
- **Fully automated operations:** Fully automated deploy, scale, patch, and upgrade experience
- **Multi-cloud:** Consistent operational experience across multiple clouds
- **GCP APIs access:** The Google Cloud Platform (GCP) Service Broker gives applications access to the Google Cloud APIs, and Google Container Engine (GKE) consistency enables the transfer of workloads from or to GCP

PKS Components

The PKS control plane contains the following components:

- An [On-Demand Broker](#) that deploys [Cloud Foundry Container Runtime](#) (CFCR), an open-source project that provides a solution for deploying and managing [Kubernetes](#) clusters using [BOSH](#).
- A Service Adapter
- The PKS API

For more information about the PKS control plane, see [PKS Cluster Management](#).

NSX-T and Harbor are optional integrations for vSphere.

The GCP Service Broker is under development.

For a detailed list of components and supported versions by a particular PKS release, see the [PKS Release Notes](#).

PKS Prerequisites

For information about the requirements for installing PKS, see [PKS Prerequisites](#).

Preparing to Install PKS

To install PKS, you must deploy Ops Manager v2.1. You use Ops Manager to install and configure PKS.

If you are installing PKS to vSphere, you can also configure integration with NSX-T and Harbor.

Consult the following table for compatibility information:

IaaS	Ops Manager v2.1	NSX-T	Harbor
vSphere	Required	Available	Available
GCP	Required	Not Available	Not Available

For information about preparing your environment before installing PKS, see the topic that corresponds to your cloud provider:

- [Preparing to Install PKS on vSphere](#)
- [Preparing to Install PKS on GCP](#)

Installing PKS

For information about installing PKS, see [Installing and Configuring PKS](#).

Upgrading PKS

For information about upgrading the PKS tile and PKS-deployed Kubernetes clusters, see [Upgrading PKS](#) [↗](#).

Managing PKS

For information about configuring authentication, creating users, and managing your PKS deployment, see [Managing PKS](#).

Using PKS

For information about using the PKS CLI to create and manage Kubernetes clusters, see [Using PKS](#).

Diagnostics and Troubleshooting PKS

For information about diagnosing and troubleshooting issues installing or using PKS, see [Diagnosing and Troubleshooting PKS](#).


PKS Release Notes

Page last updated:

v1.1.0





Release Date: June 28, 2018

Upgrade Procedure

 **Note:** The only supported upgrade path for PKS v1.1.0 is from PKS v1.0.2 and later. Do not upgrade directly to PKS v1.1.0 from v1.0.0. Instead, first upgrade PKS v1.0.0 to v1.0.2; then upgrade PKS v1.0.2 to v1.1.0. Alternatively, do a clean install of PKS v1.1.0.


To upgrade to PKS v1.1.0, follow the procedures in [Upgrade PKS](#).


Features

- Adds support for backing up and restoring PKS using BOSH Backup and Restore (BBR). See [Backing Up and Restoring PKS](#) for more information.
- Adds support for granting PKS control plane access to clients and external LDAP groups. See the [Grant Cluster Access](#) section of *Manage Users in UAA* for more information.
- Adds support for Kubernetes 1.10.3.
- Adds support for node network access via HTTP proxy for vSphere deployments. For more information, see [Networking](#).
- Adds support for PKS integration with [VMware Wavefront](#)  to capture metrics for clusters and pods. For more information, see [\(Optional\) Logging](#).
- Adds support for PKS integration with [VMware vRealize Log Insight \(vRLi\)](#)  for tagged logging of the control plane, clusters, and pods. For more information, see [\(Optional\) Monitoring](#).
- Adds support for [Harbor Registry](#)  integration enhancements: updated Harbor tile, ability to use NFS and Google Buckets as an image store, and HTTP/HTTPS proxy servers for Clair.
- Adds support for allowing workers to be deployed across Availability Zones (AZs).
- Adds support for network automation and node network isolation.
- Adds support for NO-NAT deployment topologies for PKS installations on NSX-T. See [Installing and Configuring PKS with NSX-T Integration](#) for more information.
- Adds support for integration with [VMware Analytics Cloud \(VAC\)](#)  to capture telemetry information.
- Improves security by removing ABAC authorization option for clusters.

Beta Components

- Adds support for deploying multiple Kubernetes master nodes across AZs. For information about configuring multiple masters, see the [Plans](#) section of *Installing and Configuring PKS*.

 **WARNING:** This feature is a beta component and is intended for evaluation and test purposes only. Do not use this feature in a production environment. Product support and future availability are not guaranteed for beta components.

 **WARNING:** You cannot change the number of master nodes for existing clusters. To use the multi-master feature, you must create a new plan that uses multiple master/etcd nodes and deploy a new cluster. If you are already using all three plan configurations in the PKS tile, you must delete a plan and all clusters you deployed using that plan before you can deploy a multi-master cluster.

Fixed Issues

- Operators no longer have to manually delete NSX-T objects created during the life of the product. In PKS v1.1, running the `pkcs delete-cluster` command deletes all NSX objects.

Component Versions

PKS v1.1.0 includes or supports the following component versions:

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.1	Separate download available from Pivotal Network
Stemcell	3586.24	
Kubernetes	1.10.3	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.17	Packaged in the PKS Tile
Golang	1.9.7	Packaged in the PKS Tile
NCP	2.2	Packaged in the PKS Tile
Kubernetes CLI	1.10.3	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.1	Separate download available from the PKS section of Pivotal Network
VMware vSphere	6.5 U2, 6.5 U1, and 6.5. Editions: <ul style="list-style-type: none"> vSphere Enterprise Plus Edition vSphere with Operations Management Enterprise Plus 	vSphere versions supported for Pivotal Container Service (PKS)
VMware NSX-T	2.1 - Advanced Edition	NSX-T versions supported for Pivotal Container Service (PKS)
VMware Harbor Registry	1.5.0	Separate download available from Pivotal Network
VMware vRealize Log Insight (for vSphere deployments)	4.6	Separate download available from Pivotal Network
<i>* Components marked with an asterisk have been patched to resolve security vulnerabilities or fix component behavior.</i>		

Known Issues

This section includes known issues with PKS v1.1.0 and corresponding workarounds.

- If you use PKS CLI v1.0.x with PKS tile v1.1.x, you must log in every 600 seconds to manually refresh the CLI token. Pivotal recommends upgrading to PKS CLI v1.1.x to solve this issue.
- If you upgrade PKS from v1.0.x to v1.1, you must enable the **Upgrade All Clusters** errand in the PKS tile configuration. This ensures existing clusters can perform resize or delete actions after the upgrade.

Cluster Security Recommendations

To reduce the risk of compromised clusters in your PKS deployment, the following policies are recommended:

- Ensure that only trusted operators and systems have access to clusters.
- Ensure that only trusted images are deployed to clusters.
- Maintain trusted images to consistently include current security fixes.
- Do not expose network ports to untrusted networks unless strictly required.

Reconfigure GCP Load Balancers After Master VM Recreation

If Kubernetes master node VMs are recreated for any reason, you must reconfigure your cluster load balancers to point to the new master VMs. For example, after a stemcell upgrade, BOSH recreates the VMs in your deployment.

To reconfigure your GCP cluster load balancer to use the new master VM, follow the procedure in the [Reconfiguring a GCP Load Balancer](#) section of *Configuring a GCP Load Balancer for PKS Clusters*

Existing ABAC Clusters

Attribute-based access control (ABAC) is no longer supported in v1.1. Delete any ABAC clusters before upgrading to v1.1.

New Default VM Type

In the [Resource Config](#) pane, the default **VM Type** is now **large**. This is to ensure that PKS control plane VM has sufficient resources.

If the VMs in your PKS installation use the default VM type, your VMs will use the new **large** VM type after upgrading to PKS v1.1.0.

If the VMs in your PKS installation use a custom VM type, your configuration remains the same after upgrading to PKS v1.1.0.

PKS Concepts

Page last updated:

This topic describes Pivotal Container Service (PKS) concepts. See the following sections:

- [PKS Cluster Management](#)
 - [PKS API Authentication](#)
 - [Load Balancers in PKS](#)
-

PKS Cluster Management

This topic describes how Pivotal Container Service (PKS) manages the deployment of Kubernetes clusters.

Overview

Users interact with PKS and PKS-deployed Kubernetes clusters in two ways:

- Deploying Kubernetes clusters with BOSH and managing their lifecycle. These tasks are performed using the PKS command line interface (CLI) and the PKS control plane.
- Deploying and managing container-based workloads on Kubernetes clusters. These tasks are performed using the Kubernetes CLI, `kubectl`.

Cluster Lifecycle Management

The PKS control plane enables users to deploy and manage Kubernetes clusters.

For communicating with the PKS control plane, PKS provides a command line interface, the PKS CLI. See [Installing the PKS CLI](#) for installation instructions.

PKS Control Plane Overview

The PKS control plane manages the lifecycle of Kubernetes clusters deployed using PKS. The control plane allows users to do the following through the PKS CLI:

- View cluster plans
- Create clusters
- View information about clusters
- Obtain credentials to deploy workloads to clusters
- Scale clusters
- Delete clusters

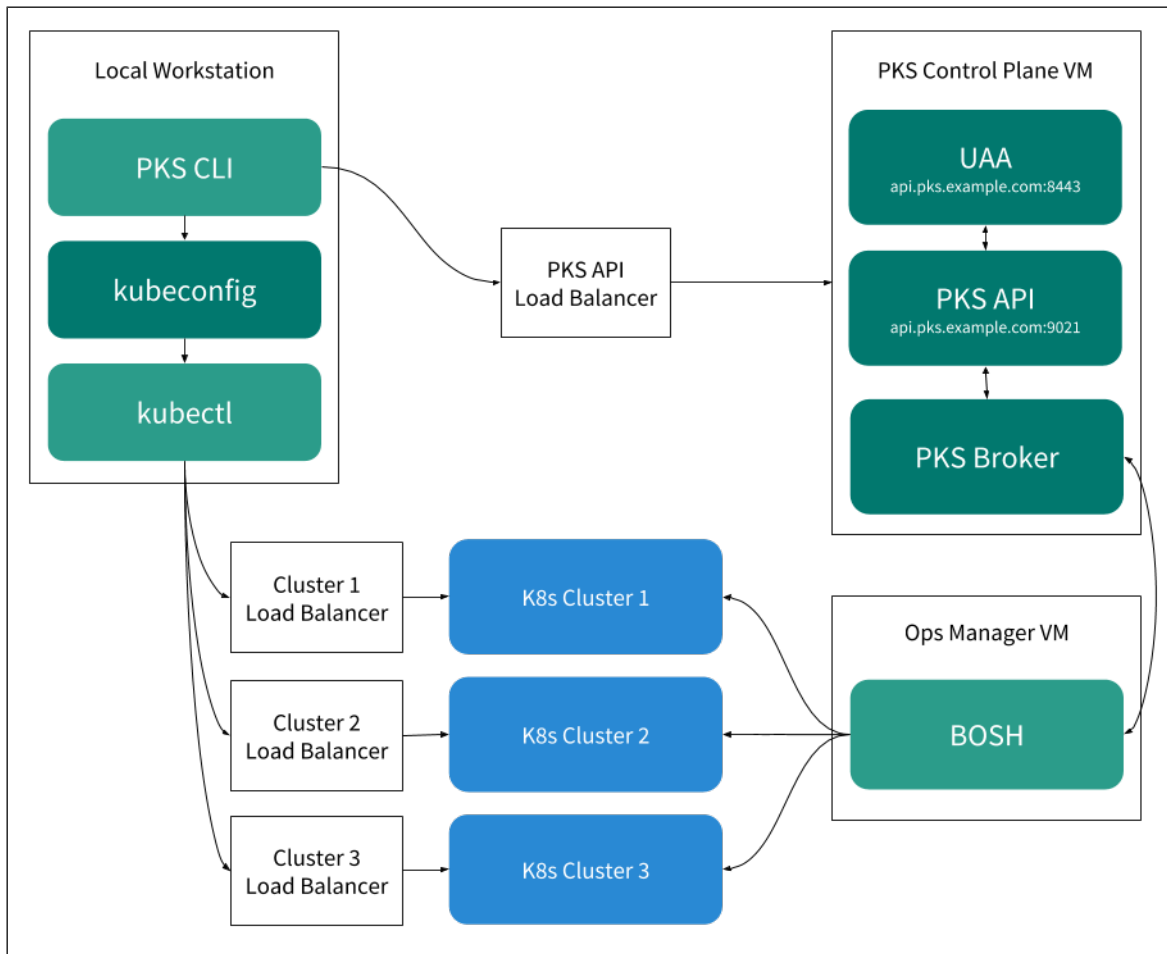
In addition, the PKS control plane can upgrade all existing clusters using the **Upgrade all clusters** BOSH errand. For more information, see [Upgrade Kubernetes Clusters](#) in *Upgrade PKS*.

PKS Control Plane Architecture

The PKS control plane is deployed on a single VM that includes the following components:

- The PKS API server
- The PKS Broker
- A User Account and Authentication (UAA) server

The following illustration shows how these components interact:



The PKS API Load Balancer is used for GCP and vSphere without NSX-T deployments. If PKS is deployed on vSphere with NSX-T, a DNAT rule is configured for the PKS API host so that it is accessible. For more information, see the [Retrieve the PKS Endpoint](#) section in *Installing and Configuring PKS with NSX-T Integration*.

UAA

When a user logs in to or logs out of the PKS API through the PKS CLI, the PKS CLI communicates with UAA to authenticate them. The PKS API permits only authenticated users to manage Kubernetes clusters. For more information about authenticating, see [PKS API Authentication](#).

UAA must be configured with the appropriate users and user permissions. For more information, see [Manage Users in UAA](#).

PKS API

Through the PKS CLI, users instruct the PKS API server to deploy, scale up, and delete Kubernetes clusters as well as show cluster details and plans. The PKS API can also write Kubernetes cluster credentials to a local kubeconfig file, which enables users to connect to a cluster through `kubectl`.

The PKS API sends all cluster management requests, except read-only requests, to the PKS Broker.

PKS Broker

When the PKS API receives a request to modify a Kubernetes cluster, it instructs the PKS Broker to make the requested change.

The PKS Broker consists of an [On-Demand Service Broker](#) and a Service Adapter. The PKS Broker generates a BOSH manifest and instructs the BOSH Director to deploy or delete the Kubernetes cluster.

For PKS deployments on vSphere with NSX-T, there is an additional component, the PKS NSX-T Proxy Broker. The PKS API communicates with the PKS NSX-T Proxy Broker, which in turn communicates with the NSX Manager to provision the Node Networking resources. The PKS NSX-T Proxy Broker then forwards the request to the On-Demand Service Broker to deploy the cluster.

Cluster Workload Management

PKS users manage their container-based workloads on Kubernetes clusters through `kubectl`. For more information about `kubectl`, see [Overview of kubectl](#) in the Kubernetes documentation.

PKS API Authentication

Page last updated:

This topic describes how the Pivotal Container Service (PKS) API works with User Account and Authentication (UAA) to manage authentication and authorization in your PKS deployment.

Authenticating PKS API Requests

Before users can log in and use the PKS CLI, you must [configure PKS API access](#) with UAA. You use the UAA Command Line Interface (UAAC) to target the UAA server and request an access token for the UAA admin user. If your request is successful, the UAA server returns the access token. The UAA admin access token authorizes you to make requests to the PKS API using the PKS CLI and [grant cluster access](#) to new or existing users.

When a user with cluster access logs in to the PKS CLI, the CLI requests an access token for the user from the UAA server. If the request is successful, the UAA server returns an access token to the PKS CLI. When the user runs PKS CLI commands, for example, `pkcs clusters`, the CLI sends the request to the PKS API server and includes the user's UAA token.

The PKS API sends a request to the UAA server to validate the user's token. If the UAA server confirms that the token is valid, the PKS API uses the cluster information from the PKS broker to respond to the request. For example, if the user runs `pkcs clusters`, the CLI returns a list of the clusters that the user is authorized to manage.

Routing to the PKS API Control Plane VM

The PKS API server and the UAA server use different port numbers on the control plane VM. For example, if your PKS API domain is `api.pks.example.com`, you can reach your PKS API and UAA servers at the following URLs:

Server	URL
PKS API	api.pks.example.com:9021
UAA	api.pks.example.com:8443

Refer to **Ops Manager > Pivotal Container Service > PKS API > API Hostname (FQDN)** for your PKS API domain.

Load balancer implementations differ by deployment environment. For PKS deployments on GCP or on vSphere without NSX-T, when you install the PKS tile, you configure a load balancer to access the PKS API. For more information, see the [Configure External Load Balancer](#) section of *Installing and Configuring PKS*.

For procedures that describe routing to the PKS control plane VM, see the [Configure External Load Balancer](#) section of *Installing and Configuring PKS*.

For overview information about load balancers in PKS, see [Load Balancers in PKS Deployments without NSX-T](#).

Load Balancers in PKS

Page last updated:

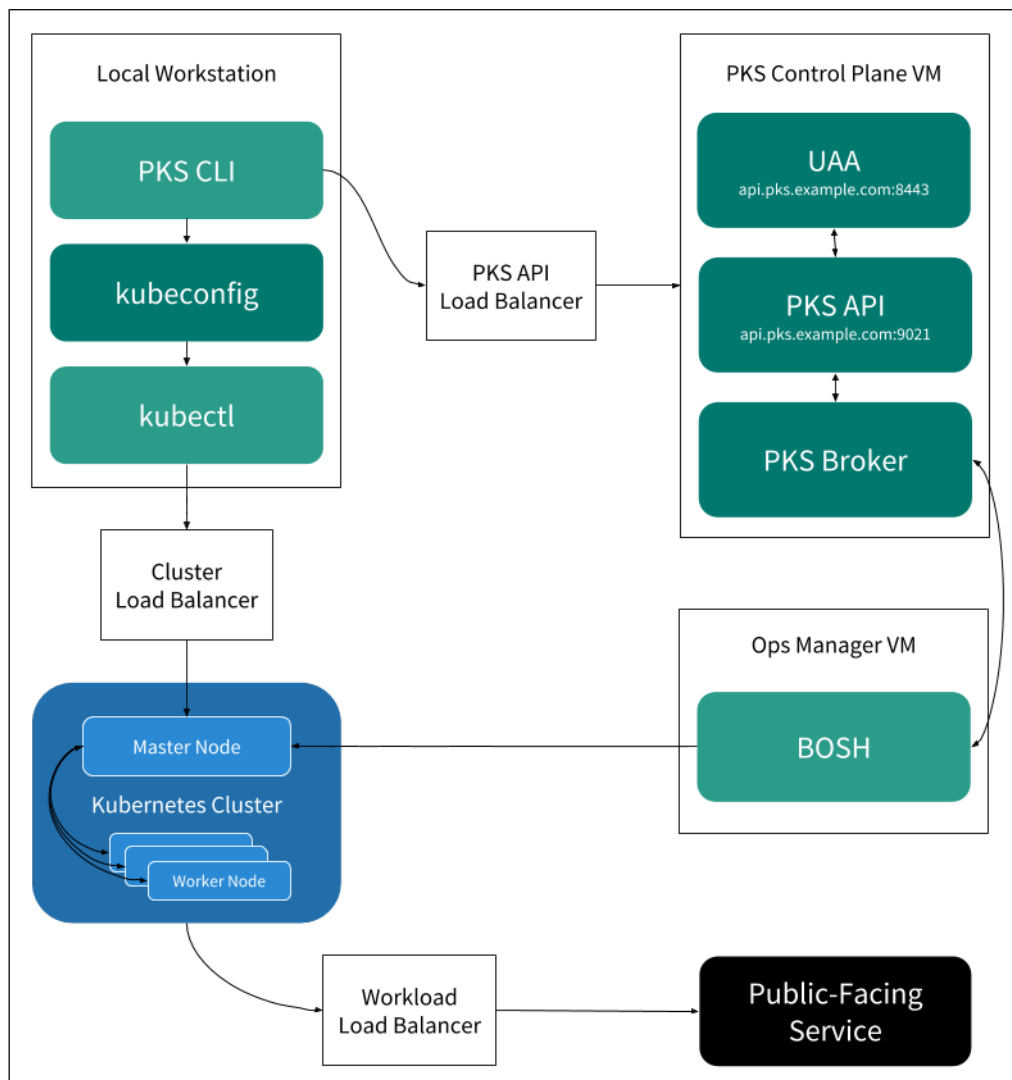
This topic describes the types of load balancers that are used in Pivotal Container Service (PKS) deployments. Load balancers differ by the type of deployment.

Load Balancers in PKS Deployments without NSX-T

For PKS deployments on GCP or vSphere without NSX-T, you can configure load balancers for the following:

- **PKS API:** Configuring this load balancer allows you to run PKS Command Line Interface (CLI) commands from your local workstation.
- **Kubernetes Clusters:** Configuring a load balancer for each new cluster allows you to run Kubernetes CLI (kubectl) commands on the cluster.
- **Workloads:** Configuring a load balancer for your application workloads allows external access to the services that run on your cluster.

The following diagram shows where each of the above load balancers can be used within your PKS deployment on GCP or on vSphere without NSX-T:



If you use either vSphere without NSX-T or GCP, you are expected to create your own load balancers within your cloud provider console. If your cloud provider does not offer load balancing, you can use any external TCP or HTTPS load balancer of your choice.

About the PKS API Load Balancer

For PKS deployments on GCP and on vSphere without NSX-T, the load balancer for the PKS API allows you to access the PKS API from outside the network.

For example, configuring a load balancer for the PKS API allows you to run PKS CLI commands from your local workstation.

For information about configuring the PKS API load balancer, see the [Configure External Load Balancer](#) section of *Installing and Configuring PKS*.

About Kubernetes Cluster Load Balancers

For PKS deployments on GCP and on vSphere without NSX-T, when you create a cluster, you must configure external access to the cluster by creating an external TCP or HTTPS load balancer. The load balancer allows the Kubernetes CLI to communicate with the cluster.

If you create a cluster in a non-production environment, you can choose not to use a load balancer. To allow kubectl to access the cluster without a load balancer, you can do one of the following:

- Create a DNS entry that points to the cluster's master VM. For example:

```
my-cluster.example.com    A    10.0.0.5
```

- On the workstation where you run kubectl commands, add the master IP address of your cluster and `kubo.internal` to the `/etc/hosts` file. For example:

```
10.0.0.5 kubo.internal
```

For information about configuring a cluster load balancer, see [Create a Cluster](#).

About Workload Load Balancers

For PKS deployments on GCP and on vSphere without NSX-T, to allow external access to your app, you can either create a load balancer or expose a static port on your workload.

For information about configuring a load balancer for your app workload, see [Deploy and Access Basic Workloads](#).

Load Balancers in PKS Deployments on vSphere with NSX-T


PKS deployments on vSphere with NSX-T do not require a load balancer configured to access the PKS API. They require only a DNAT rule configured so that the PKS API host is accessible. For more information, see [Step 9: Retrieve the PKS Endpoint](#) in *Installing and Configuring PKS with NSX-T Integration*.

NSX-T handles load balancer creation, configuration, and deletion automatically as part of the Kubernetes cluster create, update, and delete process. When a new Kubernetes cluster is created, NSX-T creates and configures a dedicated load balancer tied to it. The load balancer is a shared resource designed to provide efficient traffic distribution to master nodes as well as services deployed on worker nodes. Each application service is mapped to a virtual server instance, carved out from the same load balancer. For more information, see [Logical Load Balancer](#) in the NSX-T documentation.

Virtual server instances are created on the load balancer to provide access to the following:

- **Kubernetes API and UI services on a Kubernetes cluster.** This allows requests to be load balanced across multiple master nodes.
- **Ingress controller.** This allows the virtual server instance to dispatch HTTP and HTTPS requests to services associated with Ingress rules.
- `type:loadbalancer` **services.** This allows the server to handle TCP connections or UDP flows toward exposed services.

Load balancers are deployed in high-availability mode so that they are resilient to potential failures and able to recover quickly from critical conditions.

 **Note:** The `NodePort` Service type is not supported for PKS deployments on vSphere with NSX-T. Only `type:LoadBalancer` Services and Services associated with Ingress rules are supported on vSphere with NSX-T.

Prerequisites

Page last updated:

This topic describes the prerequisites for installing Pivotal Container Service (PKS) on vSphere or Google Cloud Platform (GCP).

General PKS Prerequisites

PKS requires the PKS Command Line Interface (PKS CLI) and the Kubernetes CLI (kubectl). See the following topics for information about installing each CLI:

- [Installing the PKS CLI](#)
- [Installing the Kubernetes CLI](#)

Resource Requirements

For information about the resource requirements for installing PKS, see the topic that corresponds to your cloud provider:

- [vSphere Prerequisites and Resource Requirements](#)
 - [GCP Prerequisites and Resource Requirements](#)
-

Installing the PKS CLI

Page last updated:

This topic describes how to install the Pivotal Container Service Command Line Interface (PKS CLI).

To install the PKS CLI, follow the procedures for your operating system to download the PKS CLI from [Pivotal Network](#). Binaries are only provided for 64-bit architectures.

Mac OS X

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **PKS CLI**.
4. Click **PKS CLI - Mac** to download the Mac OS X binary.
5. Rename the downloaded binary to `pks`.
6. On the command line, run the following command to make the PKS binary executable:

```
$ chmod +x pks
```

7. Move the binary into your `PATH`. For example:

```
$ mv pks /usr/local/bin/pks
```

Linux

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **PKS CLI**.
4. Click **PKS CLI - Linux** to download the Linux binary.
5. Rename the downloaded binary to `pks`.
6. On the command line, run the following command to make the PKS binary executable:

```
$ chmod +x pks
```

7. Move the binary into your `PATH`. For example:

```
$ mv pks /usr/local/bin/pks
```

Windows

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **PKS CLI**.
4. Click **PKS CLI - Windows** to download the Windows executable file.

5. Rename the downloaded binary to `pks.exe`.
6. Move the binary into your `PATH`.

Log in to PKS CLI

Use the command in this section to log in as an individual user. The login procedure is the same for users created in UAA or users from external LDAP groups.

On the command line, run the following command to log in to the PKS CLI:

```
pks login -a PKS-API -u USERNAME -p PASSWORD --ca-cert CERT-PATH
```

Replace the placeholder values in the command as follows:

- `PKS-API` is the domain name for the PKS API that you entered in **Ops Manager > Pivotal Container Service > PKS API > API Hostname (FQDN)**. For example, `api.pks.example.com`.
- `USERNAME` and `PASSWORD` belong to the account you created in the *Grant Cluster Access to a User* step in [Manage Users in UAA](#). If you do not use `-p` to provide a password, the CLI prompts for the password interactively. Pivotal recommends running the login command without the `-p` flag for added security.
- `CERT-PATH` is the path to your root CA certificate. Provide the certificate to validate the PKS API certificate with SSL. For example:

```
$ pks login -a api.pks.example.com -u alana \  
--ca-cert /var/tempest/workspaces/default/root_ca_certificate
```

If you are logging in to a trusted environment, you can use `-k` to skip SSL verification instead of `--ca-cert CERT-PATH`. For example:

```
$ pks login -a api.pks.example.com -u alana -k
```

Installing the Kubernetes CLI

Page last updated:

This topic describes how to install the Kubernetes Command Line Interface (kubectl).

To install kubectl, follow the procedures for your operating system to download kubectl from [Pivotal Network](#). Binaries are only provided for 64-bit architectures.

Mac OS X

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **Kubectl CLIs**.
4. Click **kubectl CLI - Mac** to download the kubectl binary.
5. Rename the downloaded binary to `kubectl`.
6. On the command line, run the following command to make the kubectl binary executable:

```
$ chmod +x kubectl
```

7. Move the binary into your `PATH`. For example:

```
$ mv kubectl /usr/local/bin/kubectl
```

Linux

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **Kubectl CLIs**.
4. Click **kubectl CLI - Linux** to download the kubectl binary.
5. Rename the downloaded binary to `kubectl`.
6. On the command line, run the following command to make the kubectl binary executable:

```
$ chmod +x kubectl
```

7. Move the binary into your `PATH`. For example:

```
$ mv kubectl /usr/local/bin/kubectl
```

Windows

1. Navigate to [Pivotal Network](#) and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Click **Kubectl CLIs**.
4. Click **kubectl CLI - Windows** to download the kubectl executable file.

5. Rename the downloaded binary to `kubect1.exe`.
 6. Move the binary into your `PATH`.
-

Preparing to Install PKS on vSphere

This topic outlines the steps for preparing to install Pivotal Container Service (PKS) on vSphere. See the following sections:

- [vSphere Prerequisites and Resource Requirements](#)
 - [Firewall Ports and Protocols Requirements for vSphere with NSX-T](#)
 - [Preparing to Deploy PKS to vSphere](#)
 - [Deploying Ops Manager to vSphere](#)
 - [Configuring Ops Manager on vSphere](#)
 - [Installing and Integrating VMware Harbor Registry with PKS](#)
-

vSphere Prerequisites and Resource Requirements

Page last updated:

This topic describes the prerequisites and resource requirements for installing Pivotal Container Service (PKS) on vSphere with or without NSX-T integration.

You can also configure integration with NSX-T and Harbor.

Component Version Requirements

vSphere Version Requirements

PKS on vSphere supports the following vSphere component versions:

Versions	Editions
<ul style="list-style-type: none"> VMware vSphere 6.5 U2 VMware vSphere 6.5 U1 VMware vSphere 6.5 GA 	<ul style="list-style-type: none"> vSphere Enterprise Plus vSphere with Operations Management Enterprise Plus

NSX-T Integration Component Version Requirements

Deploying NSX-T requires the additional following component versions:

Component	Version
VMware NSX-T	2.1 Advanced edition

Resource Requirements

Installing PKS deploys the following two virtual machines (VMs):

VM	CPU	RAM	Storage
Pivotal Container Service	1	4 GB	20 GB
Pivotal Ops Manager	1	8 GB	160 GB

Each Kubernetes cluster provisioned through PKS deploys the VMs listed below. If you deploy more than one Kubernetes cluster, you must scale your allocated resources appropriately.

VM Name	Number	CPU Cores	RAM	Ephemeral Disk	Persistent Disk
master	1	2	4 GB	8 GB	5 GB
worker	1	2	4 GB	8 GB	10 GB

NSX-T Integration Resource Requirements

Deploying NSX-T requires the additional following resources from your vSphere environment:

NSX-T Component	Instance Count	Memory per Instance	vCPU per Instance	Disk Space per Instance
NSX Manager Appliance	1	16 GB	4	140 GB
NSX Controllers	3	16 GB	4	120 GB
NSX-T Edge	1 up to 8	16 GB	8	120 GB

Installing PKS on vSphere with NSX-T

For information about the firewall ports and protocols requirements for using PKS on vSphere with NSX-T, see [Firewall Ports and Protocols Requirements for vSphere with NSX-T](#).

To install and configure PKS **with** NSX-T integration, follow the procedures below:

1. [Installing and Configuring PKS with NSX-T Integration](#)
2. (Optional) [Installing and Integrating VMware Harbor Registry with PKS](#) [↗](#)

Installing PKS on vSphere without NSX-T

To install PKS on vSphere **without** NSX-T integration, follow the procedures below:

1. [Preparing to Deploy PKS to vSphere](#)
 2. [Deploying Ops Manager to vSphere](#)
 3. [Configuring Ops Manager on vSphere](#)
 4. [Installing and Configuring PKS](#)
 5. (Optional) [Installing and Integrating VMware Harbor Registry with PKS](#) [↗](#)
-

Firewall Ports and Protocols Requirements for vSphere with NSX-T

Page last updated:


This topic describes the firewall ports and protocols requirements for using Pivotal Container Service (PKS) on vSphere with NSX-T integration.

In environments with strict inter-network access control policies, firewalls often require conduits to pass communication between system components on a different network or allow interfacing with external systems such as with enterprise applications or the public Internet.

For PKS, the recommendation is to disable security policies that filter traffic between the networks supporting the system. When that is not an option, refer to the following table, which identifies the flows between system components in a typical PKS deployment.

Source Component	Destination Component	Destination Protocol	Destination Port	Service
Application User	K8s Cluster Worker Nodes	TCP	30000-32767	k8s nodeport
Application User	K8s Load-Balancers	TCP/UDP	varies	varies
Application User	K8s Ingress-Controllers	TCP/UDP	varies	varies
Cloud Foundry BOSH Director	Domain Name Server	UDP	53	dns
Cloud Foundry BOSH Director	vCenter Server	TCP	443	https
Cloud Foundry BOSH Director	vSphere ESXi Mgmt. vmknic	TCP	443	https
Compilation Job VMs	Domain Name Server	UDP	53	dns
Developer	Harbor Private Image Registry	TCP	4443	notary
Developer	Harbor Private Image Registry	TCP	443	https
Developer	Harbor Private Image Registry	TCP	80	http
Developer	K8s Cluster Master/Etcd Nodes	TCP	8443	uaa auth
Developer	K8s Cluster Worker Nodes	TCP	30000-32767	k8s nodeport
Developer	K8s Load-Balancers	TCP/UDP	varies	varies
Developer	K8s Ingress-Controllers	TCP/UDP	varies	varies
Domain Name Server	vCenter Server	UDP	1433	ms-sql-server
Harbor Private Image Registry	Domain Name Server	UDP	53	dns
Harbor Private Image Registry	Public CVE Source Database	TCP	443	https
Harbor Private Image Registry	Public CVE Source Database	TCP	80	http
K8s Cluster Master/Etcd Nodes	Cloud Foundry BOSH Director	TCP	4222	bosh nats server
K8s Cluster Master/Etcd Nodes	Cloud Foundry BOSH Director	TCP	25250	bosh blobstore
K8s Cluster Master/Etcd Nodes	Domain Name Server	UDP	53	dns
K8s Cluster Master/Etcd Nodes	NSX Manager Server	TCP	443	https
K8s Cluster Master/Etcd Nodes	vCenter Server	TCP	443	https
K8s Cluster Worker Nodes	Cloud Foundry BOSH Director	TCP	4222	bosh nats server
K8s Cluster Worker Nodes	Cloud Foundry BOSH Director	TCP	25250	bosh blobstore
K8s Cluster Worker Nodes	Domain Name Server	UDP	53	dns
K8s Cluster Worker Nodes	Harbor Private Image Registry	TCP	8853	bosh dns health
K8s Cluster Worker Nodes	Harbor Private Image Registry	TCP	443	https
K8s Cluster Worker Nodes	NSX Manager Server	TCP	443	https
K8s Cluster Worker Nodes	vCenter Server	TCP	443	https
NSX Controllers	Network Time Server	UDP	123	ntp
NSX Edge Management	NSX Edge TEP vNIC	UDP	3784	bfd
NSX Manager Server	Domain Name Server	UDP	53	dns
NSX Manager Server	SFTP Backup Server	TCP	22	ssh
Operator	Harbor Private Image Registry	TCP	443	https
Operator	Harbor Private Image Registry	TCP	80	http
Operator	K8s Load-Balancers	TCP	80	http
Operator	NSX Manager Server	TCP	443	https

Source Component	Destination Component	Destination Protocol	Destination Port	Service
Operator	PCF Operations Manager	TCP	22	ssh
Operator	PCF Operations Manager	TCP	443	https
Operator	PCF Operations Manager	TCP	80	http
Operator	PKS Controller	TCP	8443	uaa auth
Operator	PKS Controller	TCP	9021	pks api server
Operator	vCenter Server	TCP	443	https
Operator	vCenter Server	TCP	80	http
Operator	vSphere ESXI Mgmt. vmknic	TCP	22	ssh
PCF Operations Manager	Domain Name Server	UDP	53	dns
PCF Operations Manager	K8s Cluster Worker Nodes	TCP	22	ssh
PCF Operations Manager	Network Time Server	UDP	123	ntp
PCF Operations Manager	vCenter Server	TCP	443	https
PCF Operations Manager	vSphere ESXI Mgmt. vmknic	TCP	443	https
PKS Controller	Domain Name Server	UDP	53	dns
PKS Controller	K8s Cluster Master/Etcd Nodes	TCP	8443	uaa auth
PKS Controller	NSX Manager Server	TCP	443	https
PKS Controller	vCenter Server	TCP	443	https
vCenter Server	Domain Name Server	UDP	53	dns
vCenter Server	Network Time Server	UDP	123	ntp
vCenter Server	vSphere ESXI Mgmt. vmknic	TCP	8080	vsanvp
vCenter Server	vSphere ESXI Mgmt. vmknic	TCP	9080	io filter storage
vCenter Server	vSphere ESXI Mgmt. vmknic	TCP	443	https
vCenter Server	vSphere ESXI Mgmt. vmknic	TCP	902	ideafarm-door


 **Note:** You have the option to expose containerized applications, running in a Kubernetes cluster, for external consumption through various ports and methods. You can enable external access to applications by way of Kubernetes NodePorts, load-balancers, and ingress. Enabling access to applications through Kubernetes load-balancers and ingress controller types allow for specific port and protocol designations, while NodePort offers the least control and dynamically allocates ports from a pre-defined range of ports.

Preparing to Deploy PKS on vSphere

Page last updated:

Before you install Pivotal Container Service (PKS) on vSphere **without** NSX-T integration, you must prepare your vSphere environment. In addition to fulfilling the prerequisites specified in [vSphere Prerequisites and Resource Requirements](#), you must create the following two service accounts:

- **Master Node Service Account:** You must create a service account for Kubernetes cluster master VMs and grant it sufficient storage permissions to create load balancers and attach persistent disks to pods.
- **BOSH/Ops Manager Service Account:** You must create a service account for BOSH and Ops Manager.

 **Note:** If your Kubernetes clusters span multiple vCenters, you must set the service account privileges correctly in each vCenter.

vSphere Cloud Provider service accounts require the following:

- Read access to the folder, host, and datacenter of the cluster node VMs
- Permission to create and delete VMs within the resource pool where PKS is deployed

Step 1: Create the Master Node Service Account

1. From the vCenter console, create a service account for Kubernetes cluster master VMs.
2. Grant the following **Virtual Machine Object** privileges to the service account:

Privilege (UI)	Privilege (API)
Virtual Machine > Configuration > Advanced	VirtualMachine.Configuration.Advanced
Virtual Machine > Configuration > Settings	VirtualMachine.Configuration.Settings

Step 2: Grant Storage Permissions

Grant additional permissions to the master node service account based on your storage configuration.

Follow the procedure below that corresponds to your storage configuration:

- [Static Only Persistent Volume Provisioning](#)
- [Dynamic Persistent Volume Provisioning \(with Storage Policy-Based Volume Placement\)](#)
- [Dynamic Persistent Volume Provisioning \(without Storage Policy-Based Volume Placement\)](#)

For more information about vSphere storage configurations, see [vSphere Storage for Kubernetes](#) in the VMware vSphere documentation.

Static Only Persistent Volume Provisioning

To configure your Kubernetes master node service account using static only Persistent Volume (PV) provisioning, do the following:

1. Create a custom role that allows the service account to manage Kubernetes node VMs. Give this role a name. For example, `manage-k8s-node-vm`. For more information about custom roles in vCenter, see [Create a Custom Role](#) in the VMware vSphere documentation.
 - a. Grant the following privileges at the **VM Folder** level using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Virtual Machine > Configuration > Add existing disk	VirtualMachine.Config.AddExistingDisk
Virtual Machine > Configuration > Add new disk	VirtualMachine.Config.AddNewDisk
Virtual Machine > Configuration > Add or remove device	VirtualMachine.Config.AddRemoveDevice
Virtual Machine > Configuration > Remove disk	VirtualMachine.Config.RemoveDisk

- b. Select the **Propagate to Child Objects** checkbox.

2. (Optional) Create a custom role that allows the service account to manage Kubernetes volumes. Give this role a name. For example, `manage-k8s-`

volumes

Note: This role is required if you create a Persistent Volume Claim (PVC) to bind with a statically provisioned PV, and the reclaim policy is set to delete. When the PVC is deleted, the statically provisioned PV is also deleted.

- a. Grant the following privilege at the **Datastore** level using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Datastore > Low level file operations	Datastore.FileManagement

- b. Clear the **Propagate to Child Objects** checkbox.

3. Grant the service account the existing **Read-only** role. This role includes the following privileges at the **vCenter**, **Datacenter**, **Datastore Cluster**, and **Datastore Storage Folder** levels:

Privilege (UI)	Privilege (API)
Read-only	System.Anonymous
	System.Read
	System.View

4. Continue to [Step 3: Create the BOSH/Ops Manager Service Account](#).

Dynamic Persistent Volume Provisioning (with Storage Policy-Based Volume Placement)

To configure your Kubernetes master node service account using dynamic PV provisioning **with** storage policy-based placement, do the following:

1. Create a custom role that allows the service account to manage Kubernetes node VMs. Give this role a name. For example, `manage-k8s-node-vm`. For more information about custom roles in vCenter, see [Create a Custom Role](#) in the VMware vSphere documentation.

- a. Grant the following privileges at the **Cluster**, **Hosts**, and **VM Folder** levels using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Virtual Machine > Resource > Assign virtual machine to resource pool	Resource.AssignVMToPool
Virtual Machine > Configuration > Add existing disk	VirtualMachine.Config.AddExistingDisk
Virtual Machine > Configuration > Add new disk	VirtualMachine.Config.AddNewDisk
Virtual Machine > Configuration > Add or remove device	VirtualMachine.Config.AddRemoveDevice
Virtual Machine > Configuration > Remove disk	VirtualMachine.Config.RemoveDisk
Virtual Machine > Inventory > Create new	VirtualMachine.Inventory.Create
Virtual Machine > Inventory > Remove	VirtualMachine.Inventory.Delete

- b. Select the **Propagate to Child Objects** checkbox.

2. Create a custom role that allows the service account to manage Kubernetes volumes. Give this role a name. For example, `manage-k8s-volumes`.

- a. Grant the following privilege at the **Datastore** level using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Datastore > Allocate space	Datastore.AllocateSpace
Datastore > Low level file operations	Datastore.FileManagement

- b. Clear the **Propagate to Child Objects** checkbox.

3. Create a custom role that allows the service account to read the Kubernetes storage profile. Give this role a name. For example, `k8s-system-read-and-spbm-profile-view`.

- a. Grant the following privilege at the **vCenter** level using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Profile-driven storage view	StorageProfile.View

- b. Clear the **Propagate to Child Objects** checkbox.

4. Grant the service account the existing **Read-only** role. This role includes the following privileges at the **vCenter**, **Datacenter**, **Datastore Cluster**, and **Datastore Storage Folder** levels:

Privilege (UI)	Privilege (API)
Read-only	System.Anonymous
	System.Read
	System.View

- Continue to [Step 3: Create the BOSH/Ops Manager Service Account](#).

Dynamic Volume Provisioning (without Storage Policy-Based Volume Placement)

To configure your Kubernetes master node service account using dynamic PV provisioning **without** storage policy-based placement, do the following:

- Create a custom role that allows the service account to manage Kubernetes node VMs. Give this role a name. For example, `manage-k8s-node-vm`. For more information about custom roles in vCenter, see [Create a Custom Role](#) in the VMware vSphere documentation.
 - Grant the following privileges at the **Cluster**, **Hosts**, and **VM Folder** levels using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Virtual Machine > Configuration > Add existing disk	VirtualMachine.Config.AddExistingDisk
Virtual Machine > Configuration > Add new disk	VirtualMachine.Config.AddNewDisk
Virtual Machine > Configuration > Add or remove device	VirtualMachine.Config.AddRemoveDevice
Virtual Machine > Configuration > Remove disk	VirtualMachine.Config.RemoveDisk

- Select the **Propagate to Child Objects** checkbox.
- Create a custom role that allows the service account to manage Kubernetes volumes. Give this role a name. For example, `manage-k8s-volumes`.
 - Grant the following privilege at the **Datastore** level using either the vCenter UI or API:

Privilege (UI)	Privilege (API)
Datastore > Allocate space	Datastore.AllocateSpace
Datastore > Low level file operations	Datastore.FileManagement

- Clear the **Propagate to Child Objects** checkbox.

- Grant the service account the existing **Read-only** role. This role includes the following privileges at the **vCenter**, **Datacenter**, **Datastore Cluster**, and **Datastore Storage Folder** levels:

Privilege (UI)	Privilege (API)
Read-only	System.Anonymous
	System.Read
	System.View

Step 3: Create the BOSH/Ops Manager Service Account

- From the vCenter console, create a service account for BOSH and Ops Manager.
- Grant the permissions below to the BOSH and Ops Manager service account.

vCenter Root Privileges

Grant the following privileges on the root vCenter server entity to the service account:

Privilege (UI)	Privilege (API)
Read-only	System.Anonymous
	System.Read
	System.View
Manage custom attributes	Global.ManageCustomFields

vCenter Datacenter Privileges

Grant the following privileges on any entities in a datacenter where you deploy PKS:

Role Object

Privilege (UI)	Privilege (API)
Users inherit the Read-Only role from the vCenter root level	System.Anonymous
	System.Read
	System.View

Datastore Object

Grant the following privileges must at the datacenter level to upload and delete virtual machine files:

Privilege (UI)	Privilege (API)
Allocate space	Datastore.AllocateSpace
Browse datastore	Datastore.Browse
Low level file operations	Datastore.FileManagement
Remove file	Datastore.DeleteFile
Update virtual machine files	Datastore.UpdateVirtualMachineFiles

Folder Object

Privilege (UI)	Privilege (API)
Delete folder	Folder.Delete
Create folder	Folder.Create
Move folder	Folder.Move
Rename folder	Folder.Rename

Global Object

Privilege (UI)	Privilege (API)
Set custom attribute	Global.SetCustomField

Host Object

Privilege (UI)	Privilege (API)
Modify cluster	Host.Inventory.EditCluster

Inventory Service Object

Privilege (UI)	Privilege (API)
vSphere Tagging > Create vSphere Tag	InventoryService.Tagging.CreateTag
vSphere Tagging > Delete vSphere Tag	InventoryService.Tagging.EditTag
vSphere Tagging > Edit vSphere Tag	InventoryService.Tagging.DeleteTag

Network Object

Privilege (UI)	Privilege (API)
Assign network	Network.Assign

Resource Object

Privilege (UI)	Privilege (API)
Assign virtual machine to resource pool	Resource.AssignVMToPool
Migrate powered off virtual machine	Resource.ColdMigrate
Migrate powered on virtual machine	Resource.HotMigrate

vApp Object

Grant these privileges at the resource pool level.

Privilege (UI)	Privilege (API)
Import	VApp.Import
vApp application configuration	VApp.ApplicationConfig

Virtual Machine Object

Configuration

Privilege (UI)	Privilege (API)
Add existing disk	VirtualMachine.Config.AddExistingDisk
Add new disk	VirtualMachine.Config.AddNewDisk
Add or remove device	VirtualMachine.Config.AddRemoveDevice
Advanced	VirtualMachine.Config.AdvancedConfig
Change CPU count	VirtualMachine.Config.CPUCount
Change resource	VirtualMachine.Config.Resource
Configure managedBy	VirtualMachine.Config.ManagedBy
Disk change tracking	VirtualMachine.Config.ChangeTracking
Disk lease	VirtualMachine.Config.DiskLease
Display connection settings	VirtualMachine.Config.MksControl
Extend virtual disk	VirtualMachine.Config.DiskExtend
Memory	VirtualMachine.Config.Memory
Modify device settings	VirtualMachine.Config.EditDevice
Raw device	VirtualMachine.Config.RawDevice
Reload from path	VirtualMachine.Config.ReloadFromPath
Remove disk	VirtualMachine.Config.RemoveDisk
Rename	VirtualMachine.Config.Rename
Reset guest information	VirtualMachine.Config.ResetGuestInfo
Set annotation	VirtualMachine.Config.Annotation
Settings	VirtualMachine.Config.Settings
Swapfile placement	VirtualMachine.Config.SwapPlacement
Unlock virtual machine	VirtualMachine.Config.Unlock

Guest Operations

Privilege (UI)	Privilege (API)
Guest Operation Program Execution	VirtualMachine.GuestOperations.Execute

Guest Operation Modifications	VirtualMachine.GuestOperations.Modify
Guest Operation Queries	VirtualMachine.GuestOperations.Query

Interaction

Privilege (UI)	Privilege (API)
Answer question	VirtualMachine.Interact.AnswerQuestion
Configure CD media	VirtualMachine.Interact.SetCDMedia
Console interaction	VirtualMachine.Interact.ConsoleInteract
Defragment all disks	VirtualMachine.Interact.DefragmentAllDisks
Device connection	VirtualMachine.Interact.DeviceConnection
Guest operating system management by VIX API	VirtualMachine.Interact.GuestControl
Power off	VirtualMachine.Interact.PowerOff
Power on	VirtualMachine.Interact.PowerOn
Reset	VirtualMachine.Interact.Reset
Suspend	VirtualMachine.Interact.Suspend
VMware Tools install	VirtualMachine.Interact.ToolsInstall

Inventory

Privilege (UI)	Privilege (API)
Create from existing	VirtualMachine.Inventory.CreateFromExisting
Create new	VirtualMachine.Inventory.Create
Move	VirtualMachine.Inventory.Move
Register	VirtualMachine.Inventory.Register
Remove	VirtualMachine.Inventory.Delete
Unregister	VirtualMachine.Inventory.Unregister

Provisioning

Privilege (UI)	Privilege (API)
Allow disk access	VirtualMachine.Provisioning.DiskRandomAccess
Allow read-only disk access	VirtualMachine.Provisioning.DiskRandomRead
Allow virtual machine download	VirtualMachine.Provisioning.GetVmFiles
Allow virtual machine files upload	VirtualMachine.Provisioning.PutVmFiles
Clone template	VirtualMachine.Provisioning.CloneTemplate
Clone virtual machine	VirtualMachine.Provisioning.Clone
Customize	VirtualMachine.Provisioning.Customize
Deploy template	VirtualMachine.Provisioning.DeployTemplate
Mark as template	VirtualMachine.Provisioning.MarkAsTemplate
Mark as virtual machine	VirtualMachine.Provisioning.MarkAsVM
Modify customization specification	VirtualMachine.Provisioning.ModifyCustSpecs
Promote disks	VirtualMachine.Provisioning.PromoteDisks
Read customization specifications	VirtualMachine.Provisioning.ReadCustSpecs

Snapshot Management

Privilege (UI)	Privilege (API)
Create snapshot	VirtualMachine.State.CreateSnapshot
Remove snapshot	VirtualMachine.State.RemoveSnapshot
Rename snapshot	VirtualMachine.State.RenameSnapshot
Revert snapshot	VirtualMachine.State.RevertToSnapshot

Next Steps

To install PKS on vSphere, follow the procedures in [Deploying Ops Manager to vSphere](#).

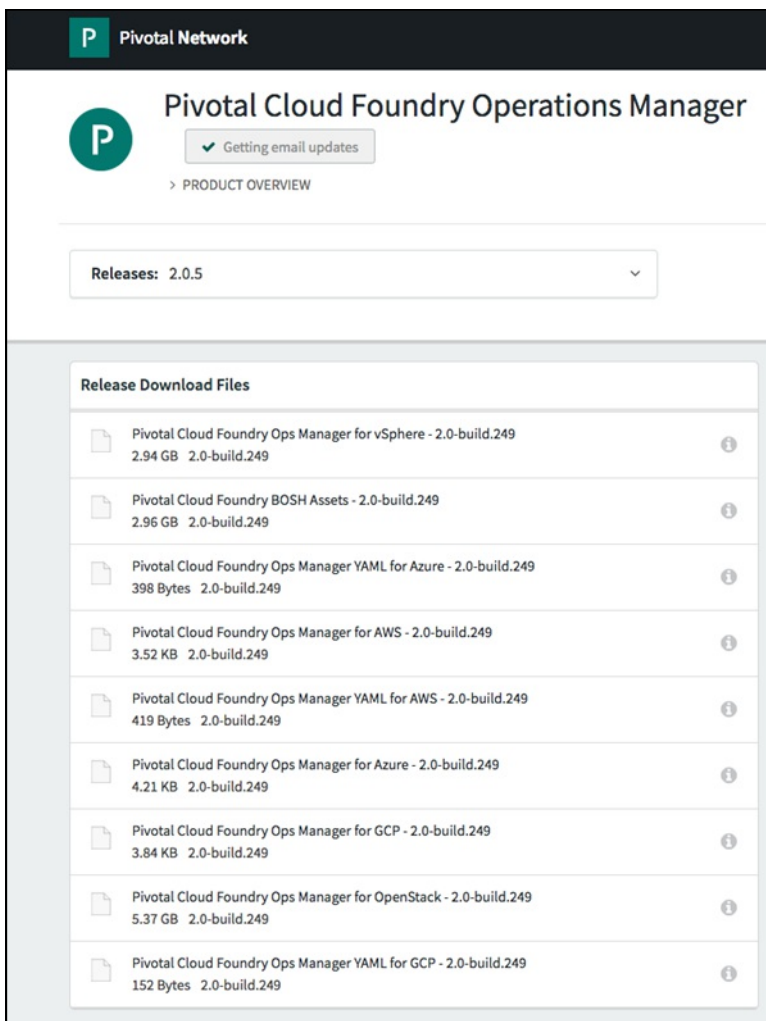
Deploying Ops Manager to vSphere

Page last updated:

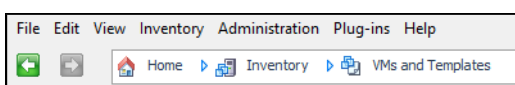
This topic provides instructions for deploying Ops Manager to VMware vSphere.

Note: With vSphere 6.5 and NSX-T 2.1, when initially deploying the Operations Manager OVF, you cannot connect directly to an NSX-T logical switch. You must first connect to a vSphere Standard (vSS) or vSphere Distributed Switch (vDS). A suggested approach is to connect to a VSS or VDS when deploying the OVF, but do not power the VM on. After the OVF deployment has completed, you can then connect the network interface to the appropriate NSX-T logical switch and power the VM on to proceed with the install.

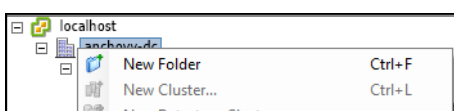
1. Before starting, refer to the known issues in the [PCF Ops Manager Release v2.1 Release Notes](#).
2. Download the [Pivotal Cloud Foundry](#) (PCF) Ops Manager `.ova` file at [Pivotal Network](#). Click the **Pivotal Cloud Foundry** region to access the PCF product page. Use the dropdown menu to select an Ops Manager release.



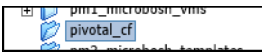
3. Log into vCenter.
4. Select the **VM and Templates** view.



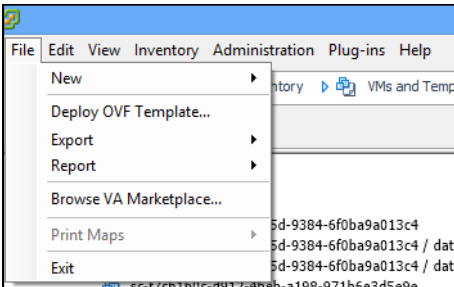
5. Right click on your datacenter and select **New Folder**.



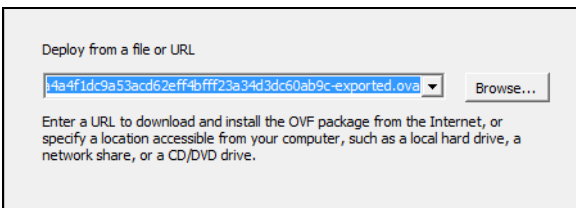
6. Name the folder `pivotal_cf` and select it.



7. Select **File > Deploy OVF Template**.



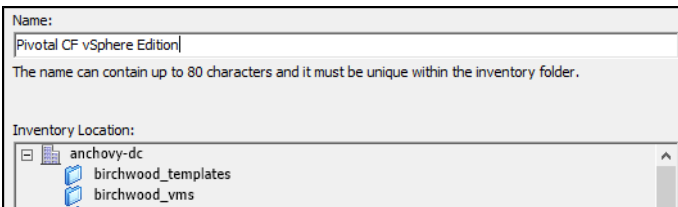
8. Select the .ova file and click **Next**.



9. Review the product details and click **Next**.

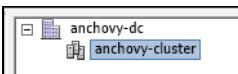
10. Accept the license agreement and click **Next**.

11. Name the virtual machine and click **Next**.



Note: The selected folder is the one you created.

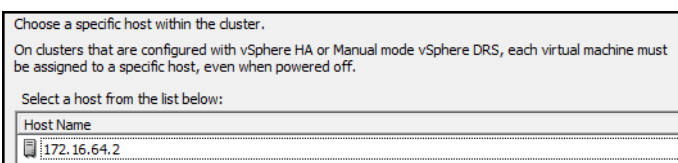
12. Select a vSphere cluster and click **Next**.



13. If prompted, select a resource pool and click **Next**.


14. If prompted, select a host and click **Next**.

Note: If your vSphere host does not support VT-X/EPT, you must disable hardware virtualization. For more information, see [PCF on vSphere Requirements](#).




15. Select a storage destination and click **Next**.

Select a destination storage for the virtual machine files:

VM Storage Profile: anchovy-ds 

Name	Drive Type	Capacity	Provisioned	Free	Type	Thin Pro
anchovy-ds	Non-SSD	5.41 TB	1.62 TB	3.98 TB	VMFS5	Supporte

16. Select a disk format and click **Next**. For more information about disk formats, see [Provisioning a Virtual Disk in vSphere](#).

 **Warning:** Ops Manager v2.1 requires a Director VM with at least 8 GB memory.

Datastore: anchovy-ds

Available space (GB): 4076.0

☒ Thick Provision Lazy Zeroed
☐ Thick Provision Eager Zeroed
☐ Thin Provision

17. Select a network from the drop down list and click **Next**.

Source Networks	Destination Networks
Network 1	<div> <div>▼</div> <div> Mattnetwork VM Network VM Network Private </div> </div>

18. Enter network information and passwords for the Ops Manager VM admin user.

Application properties - Ops Manager

Product: Ops Manager
Version: 2.0-build.91
Vendor: Pivotal

Uncategorized: 7 settings

IP Address: The IP address for the Ops Manager. Leave blank if DHCP is desired.

Netmask: The netmask for the Ops Manager's network. Leave blank if DHCP is desired.

Default Gateway: The default gateway address for the Ops Manager's network. Leave blank if DHCP is desired.

DNS: The domain name servers for the Ops Manager (comma separated). Leave blank if DHCP is desired.

NTP Servers: Comma-delimited list of NTP servers

Admin Password: This password is used to SSH into the Ops Manager. The username is 'ubuntu'.
Enter password:
Confirm password:

Custom Hostname: This will be set as the hostname on the VM. Default: 'pivotal-ops-manager'.

 **Note:** Record this network information. The IP Address will be the location of the Ops Manager interface.

19. In the **Admin Password** field, enter a default password for the ubuntu user. If you do not enter a default password, your Ops Manager will not boot up.


Admin Password: This password is used to SSH into the Ops Manager. The username is 'ubuntu'.

Enter password:


Confirm password:

20. Click **Next**.

21. Check the **Power on after deployment** checkbox and click **Finish**. Once the VM boots, the interface is available at the IP address you specified.

 **Note:** It is normal to experience a brief delay before the interface is accessible while the web server and VM start up.

22. Create a DNS entry for the IP address that you used for Ops Manager. You must use this fully qualified domain name when you log into Ops Manager in [Installing Pivotal Cloud Foundry on vSphere](#).

 **Note:** Ops Manager security features require you to create a fully qualified domain name to access Ops Manager during the initial configuration. For more information, see [PCF on vSphere Requirements](#).

Next Steps


After you complete this procedure, follow the instructions in [Configuring Ops Manager on vSphere](#).

Configuring Ops Manager on vSphere

Page last updated:

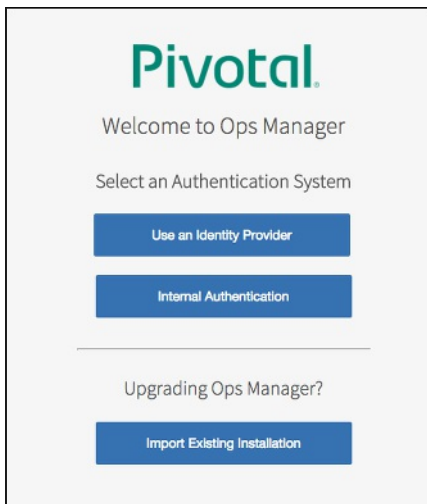
This topic describes how to configure Ops Manager for VMware vSphere.

If you are installing Pivotal Container Service (PKS) to vSphere **without** NSX-T integration, before you begin this procedure, ensure that you have successfully completed all of the steps in [Deploying Ops Manager to vSphere](#).

 **Note:** You can also perform the procedures in this topic using the Ops Manager API. For more information, see [Using the Ops Manager API](#).

Step 1: Set Up Ops Manager

1. Navigate to the fully qualified domain of your Ops Manager in a web browser.
2. The first time you start Ops Manager, you must choose one of the following:
 - [Use an Identity Provider](#): If you use an Identity Provider (IdP), an external identity server maintains your user database.
 - [Internal Authentication](#): If you use Internal Authentication, PCF maintains your user database.



Use an Identity Provider

1. Log in to your IdP console and download the IdP metadata XML. Optionally, if your IdP supports metadata URL, you can copy the metadata URL instead of the XML.
2. Copy the IdP metadata XML or URL to the Ops Manager **Use an Identity Provider** log in page.

Note: The same IdP metadata URL or XML is applied for the BOSH Director. If you use a separate IdP for BOSH, copy the metadata XML or URL from that IdP and enter it into the BOSH IdP Metadata text box in the Ops Manager log in page.

3. Enter your **Decryption passphrase**. Read the **End User License Agreement**, and select the checkbox to accept the terms.

4. Your Ops Manager log in page appears. Enter your username and password. Click **Login**.

5. Download your SAML Service Provider metadata (SAML Relying Party metadata) by navigating to the following URLs:

- **5a.** Ops Manager SAML service provider metadata: `https://OPS-MAN-FQDN:443/uaa/saml/metadata`
- **5b.** BOSH Director SAML service provider metadata: `https://BOSH-IP-ADDRESS:8443/saml/metadata`

Note: To retrieve your `BOSH-IP-ADDRESS`, navigate to the **Ops Manager Director** tile > **Status** tab. Record the **Ops Manager Director** IP address.

6. Configure your IdP with your SAML Service Provider metadata. Import the Ops Manager SAML provider metadata from Step 5a above to your IdP. If your IdP does not support importing, provide the values below.

- **Single sign on URL:** `https://OPS-MAN-FQDN:443/uaa/saml/SSO/alias/OPS-MAN-FQDN`
- **Audience URI (SP Entity ID):** `https://OP-MAN-FQDN:443/uaa`
- **Name ID:** Email Address
- SAML authentication requests are always signed

7. Import the BOSH Director SAML provider metadata from Step 5b to your IdP. If the IdP does not support an import, provide the values below.

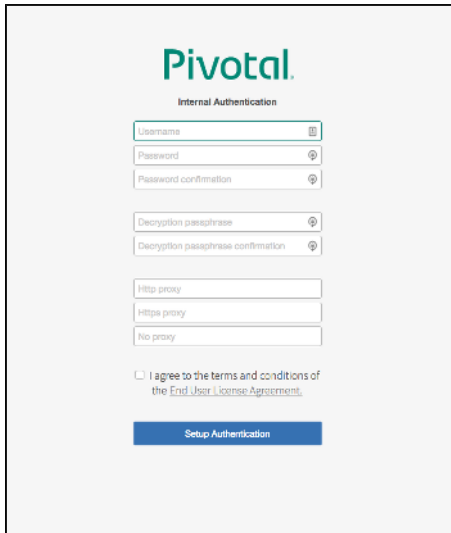
- **Single sign on URL:** `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
- **Audience URI (SP Entity ID):** `https://BOSH-IP:8443`
- **Name ID:** Email Address
- SAML authentication requests are always signed

8. Return to the **Ops Manager Director** tile, and continue with the configuration steps below.

Internal Authentication

1. When redirected to the **Internal Authentication** page, you must complete the following steps:

- Enter a **Username**, **Password**, and **Password confirmation** to create an Admin user.
- Enter a **Decryption passphrase** and the **Decryption passphrase confirmation**. This passphrase encrypts the Ops Manager datastore, and is not recoverable.
- If you are using an **HTTP proxy** or **HTTPS proxy**, follow the instructions in [Configuring Proxy Settings for the BOSH CPI](#).
- Read the **End User License Agreement**, and select the checkbox to accept the terms.



Pivotal
Internal Authentication

Username

Password

Password confirmation

Decryption passphrase

Decryption passphrase confirmation

Http proxy

Https proxy

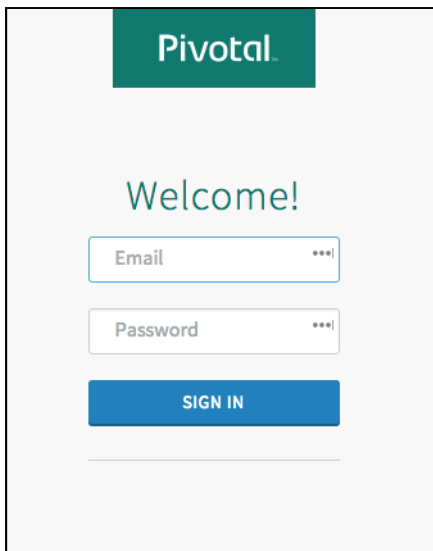
No proxy

☐ I agree to the terms and conditions of the [End User License Agreement](#).

Setup Authentication

Step 2: vCenter Config Page

1. Log in to Ops Manager with the Admin username and password you created in the previous step.



Pivotal

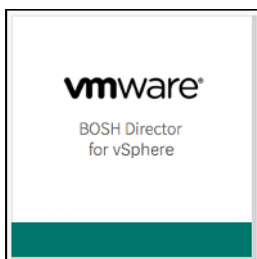
Welcome!

Email

Password

SIGN IN

2. Click the **Ops Manager Director** tile.



3. Select **vCenter Config**.

vCenter Config

Director Config

Create Availability Zones

Create Networks

Assign AZs and Networks

Security

Syslog

Resource Config

vCenter Host*

vcenter.pizza.cf-app.com

vCenter Username*

root

vCenter Password*

Change

Datacenter Name*

pizza-boxes-dc

Virtual Disk Type*

thin

Ephemeral Datastore Names (comma delimited)*

vmx5600-pizza-2

NOTE: Removing an Ephemeral Datastore after an initial deploy can result in a system outage and/or data loss.

Persistent Datastore Names (comma delimited)*

vmx5600-pizza-2

NOTE: Removing a Persistent Datastore after an initial deploy can result in a system outage and/or data loss.

☒ Standard vCenter Networking

☐ NSX Networking

NSX Address*

NSX Username*

NSX Password*

NSX CA Cert

Optional custom CA certificate(s)

VM Folder*

pikachu_vms

Template Folder*

pikachu_templates

Disk path Folder*

pikachu_disks

Save

4. Enter the following information:

- **vCenter Host:** The hostname of the vCenter that manages ESXi/vSphere.
- **vCenter Username:** A vCenter username with create and delete privileges for virtual machines (VMs) and folders.
- **vCenter Password:** The password for the vCenter user specified above.
- **Datacenter Name:** The name of the datacenter as it appears in vCenter.
- **Virtual Disk Type:** The Virtual Disk Type to provision for all VMs. For guidance on selecting a virtual disk type, see [Provisioning a Virtual Disk in vSphere](#).
- **Ephemeral Datastore Names (comma delimited):** The names of the datastores that store ephemeral VM disks deployed by Ops Manager.
- **Persistent Datastore Names (comma delimited):** The names of the datastores that store persistent VM disks deployed by Ops Manager.
- **VM Folder:** The vSphere datacenter folder (default: `pcf_vms`) where Ops Manager places VMs.
- **Template Folder:** The vSphere datacenter folder (default: `pcf_templates`) where Ops Manager places VMs.
- **Disk path Folder:** The vSphere datastore folder (default: `pcf_disk`) where Ops Manager creates attached disk images. You must not nest this folder.

5. If you are deploying to vSphere with NSX-T, select **NSX Networking - NSX-T**. For deploying on other IaaSes select **Standard vCenter Networking**.

- Click **Save**.

 **Note:** After your initial deployment, you cannot edit the VM Folder, Template Folder, and Disk path Folder names.

Step 3: Director Config Page

- Select **Director Config**.

Director Config

NTP Servers (comma delimited)*

time1.sf.cf.app.com

JMX Provider IP Address

Bosh HM Forwarder IP Address

☐ Enable VM Resurrector Plugin

☐ Enable Post Deploy Scripts

☐ Recreate all VMs


This will force BOSH to recreate all VMs on the next deploy. Persistent disk will be preserved

☐ Enable bosh deploy retries


This will attempt to re-deploy a failed deployment up to 5 times.

☐ Keep Unreachable Director VMs


- In the **NTP Servers (comma delimited)** field, enter your NTP server addresses.
- Leave the **JMX Provider IP Address** field blank.

 **Note:** Starting from PCF v2.0, BOSH-reported system metrics are available in the Loggregator Firehose by default. If you continue to use PCF JMX Bridge for consuming them outside of the Firehose, you may receive duplicate data. To prevent this duplicate data, leave the **JMX Provider IP Address** field blank.

- Leave the **Bosh HM Forwarder IP Address** field blank.

 **Note:** Starting from PCF v2.0, BOSH-reported system metrics are available in the Loggregator Firehose by default. If you continue to use the BOSH HM Forwarder for consuming them, you may receive duplicate data. To prevent duplicate data, leave the **Bosh HM Forwarder IP Address** field blank.

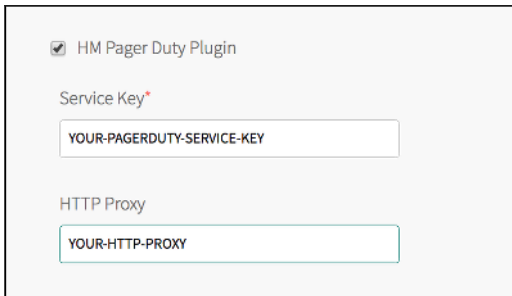
- Select the **Enable VM Resurrector Plugin** to enable Ops Manager Resurrector functionality.
- Select **Enable Post Deploy Scripts** to run a post-deploy script after deployment. This script allows the job to execute additional commands against a deployment.

 **Note:** You must enable post-deploy scripts to install PKS.

- Select **Recreate all VMs** to force BOSH to recreate all VMs on the next deploy. This process does not destroy any persistent disk data.
- Select **Enable bosh deploy retries** if you want Ops Manager to retry failed BOSH operations up to five times.

9. Select **Keep Unreachable Director VMs** if you want to preserve Ops Manager Director VMs after a failed deployment for troubleshooting purposes.

10. Select **HM Pager Duty Plugin** to enable Health Monitor integration with PagerDuty.



☒ HM Pager Duty Plugin

Service Key*

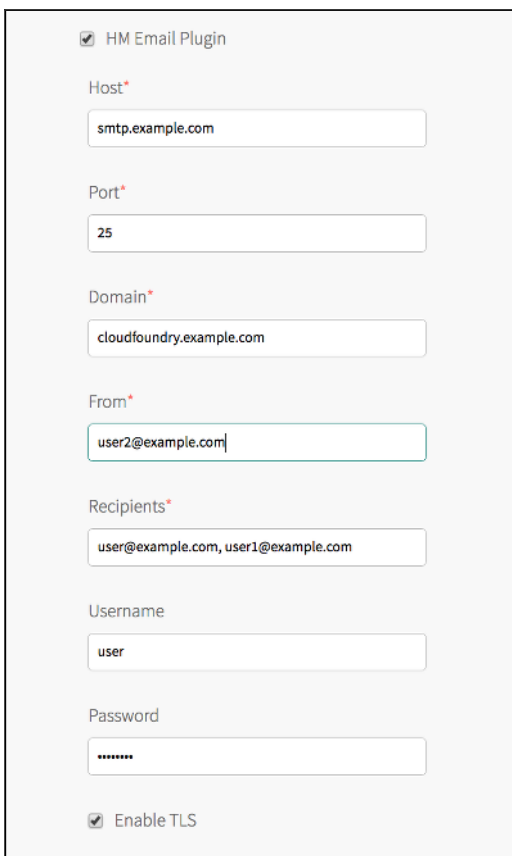
YOUR-PAGERDUTY-SERVICE-KEY

HTTP Proxy

YOUR-HTTP-PROXY

- **Service Key:** Enter your API service key from PagerDuty.
- **HTTP Proxy:** Enter an HTTP proxy for use with PagerDuty.

11. Select **HM Email Plugin** to enable Health Monitor integration with email.



☒ HM Email Plugin

Host*

smtp.example.com

Port*

25

Domain*

cloudfoundry.example.com

From*

user2@example.com

Recipients*

user@example.com, user1@example.com

Username


user

Password

☒ Enable TLS

- **Host:** Enter your email hostname.
- **Port:** Enter your email port number.
- **Domain:** Enter your domain.
- **From:** Enter the address for the sender.
- **Recipients:** Enter comma-separated addresses of intended recipients.
- **Username:** Enter the username for your email server.
- **Password:** Enter the password for your email server.
- **Enable TLS:** Select this checkbox to enable Transport Layer Security.

12. Select a **Blobstore Location** to either configure the blobstore as an internal server or an external endpoint. Because the internal server is unscalable and less secure, Pivotal recommends you configure an external blobstore.

 **Note:** After you deploy Ops Manager, you cannot change the blobstore location.

- **Internal:** Select this option to use an internal blobstore. Ops Manager creates a new VM for blob storage. No additional configuration is required.
- **S3 Compatible Blobstore:** Select this option to use an external S3-compatible endpoint. Follow the procedures in [Sign up for Amazon S3](#) and [Create a Bucket](#) in the AWS documentation. When you have created an S3 bucket, complete the following steps:

1. **S3 Endpoint:** Navigate to the [Regions and Endpoints](#) topic in the AWS documentation. Locate the endpoint for your region in the **Amazon Simple Storage Service (S3)** table and construct a URL using your region's endpoint. For example, if you are using the `us-west-2` region, the URL you create would be <https://s3-us-west-2.amazonaws.com>. Enter this URL into the **S3 Endpoint** field in Ops Manager.
2. **Bucket Name:** Enter the name of the S3 bucket.
3. **Access Key** and **Secret Key:** Enter the keys you generated when creating your S3 bucket.
4. Select **V2 Signature** or **V4 Signature**. If you select **V4 Signature**, enter your **Region**.



Note: AWS recommends using Signature Version 4. For more information about AWS S3 Signatures, see [Authenticating Requests](#) in the AWS documentation.

- **GCS Blobstore:** Select this option to use an external Google Cloud Storage (GCS) endpoint. To create a GCS bucket, you will need a GCS account. Follow the procedures in [Creating Storage Buckets](#) in the GCP documentation. Once you have created a GCS bucket, complete the following steps:
 1. **Bucket Name:** Enter the name of your GCS bucket.
 2. **Storage Class:** Select the storage class for your GCS bucket. For more information, see [Storage Classes](#) in the GCP documentation.
 3. **Service Account Key:** Follow the steps in the [Create Service Accounts](#) section to download a JSON file with a private key, and then enter the contents of the JSON file into the field.

Blobstore Location

☒ Internal
 ☐ S3 Compatible Blobstore

S3 Endpoint*

Bucket Name*

Access Key*

Secret Key*

☒ V2 Signature
 ☐ V4 Signature

Region*

☐ GCS Blobstore

Bucket Name*

Storage Class*

Regional

Service Account Key*

13. By default, PCF deploys and manages an **Internal** database for you. If you choose to use an **External MySQL Database**, complete the associated fields with information obtained from your external MySQL Database provider: **Host**, **Port**, **Username**, **Password**, and **Database**.

Database Location

☒ Internal

☐ External MySQL Database

Host*

Port*

Username*

Password*

Database*

14. (Optional) **Director Workers** sets the number of workers available to execute Director tasks. This field defaults to `5`.
15. (Optional) **Max Threads** sets the maximum number of threads that the Ops Manager Director can run simultaneously. For vSphere, the default value is `32`. Leave the field blank to use this default value. Pivotal recommends that you use the default value unless doing so results in rate limiting or errors on your IaaS.
16. Leave the **Director Hostname** field blank.
17. Ensure the **Disable BOSH DNS server for troubleshooting purposes** checkbox is not selected.

Note: BOSH DNS must be enabled in all PKS deployments. If PAS and PKS are running on the same instance of Ops Manager, you cannot use the opt-out feature of BOSH DNS for your PAS without breaking PKS. If you want to opt out of BOSH DNS in your PAS deployment, install the tile on a separate instance of Ops Manager. For more information about opting out of BOSH DNS, see [Disabling or Opting Out of BOSH DNS in PCF](#) in the Pivotal Knowledge Base and [BOSH DNS Service Discovery \(Beta\) and Opt-Out Option](#) in the *Ops Manager v2.0 Release Notes*.

18. (Optional) To set a custom banner that users see when logging in to the Director using SSH, enter text in the **Custom SSH Banner** field.

☐ Disable BOSH DNS server for troubleshooting purposes

Custom SSH Banner

19. Click **Save**.

Note: After your initial deployment, you cannot edit the Blobstore and Database locations.

Step 4: Create Availability Zone Page

Ops Manager Availability Zones correspond to your vCenter clusters and resource pools. Multiple Availability Zones allow you to provide high-availability and load balancing to your applications. When you run more than one instance of an application, Ops Manager balances those instances across all of the Availability Zones assigned to the application. At least three availability zones are recommended for a highly available installation of your chosen runtime.

1. Select **Create Availability Zones**.

2. Use the following steps to create one or more Availability Zones for your applications to use:

- Click **Add**.
- Enter a unique **Name** for the Availability Zone.
- Enter the name of an existing vCenter **Cluster** to use as an Availability Zone.
- (Optional) Enter the name of a **Resource Pool** in the vCenter cluster that you specified above. The jobs running in this Availability Zone share the CPU and memory resources defined by the pool.
- (Optional) Click **Add Cluster** to create another set of **Cluster** and **Resource Pool** fields. You can add multiple clusters. Click the trash icon to delete a cluster. The first cluster cannot be deleted.

Note: For more information about using availability zones in vSphere, see [Understanding Availability Zones in VMware Installations](#) in the PCF documentation.

3. Click **Save**.

Step 5: Create Networks Page

1. Select **Create Networks**.

2. Select **Enable ICMP checks** to enable ICMP on your networks. Ops Manager uses ICMP checks to confirm that components within your network are reachable.

3. Click **Add Network** and create the following networks:

- `pks-infrastructure`: for Ops Manager, the BOSH Director, the PKS broker, and the PKS API. If you have a large deployment with multiple tiles, you can choose to deploy the PKS broker and PKS API to a separate network named `pks-main`. See the table below for more information.
- `pks-services`: for creating the master and worker VMs for Kubernetes clusters.

Note: If you are deploying PKS with NSX-T integration, see the network configuration table in the [Configure Ops Manager](#) section of *Installing and Configuring PKS with NSX-T Integration*.


Use the values from the following table as a guide when you create each network, replacing the IP addresses with ranges that are available in your vSphere environment:


	Field	Configuration
	Name	<code>pks-infrastructure</code>

Infrastructure Network	Service Network	Leave Service Network unchecked.
	vSphere Network Name	MY-PKS-virt-net/MY-PKS-subnet-infrastructure
	CIDR	192.168.101.0/26
	Reserved IP Ranges	192.168.101.1-192.168.101.9
	DNS	192.168.101.2
Main Network (Optional)	Gateway	192.168.101.1
	Field	Configuration
	Name	pks-main
	Service Network	Leave Service Network unchecked.
	vSphere Network Name	MY-PKS-virt-net/MY-PKS-subnet-pks
	CIDR	192.168.16.0/26
	Reserved IP Ranges	192.168.16.1-192.168.16.9
	DNS	192.168.16.2
	Gateway	192.168.16.1
Service Network	Field	Configuration
	Name	pks-services
	Service Network	Select the Service Network checkbox.
	vSphere Network Name	MY-PKS-virt-net/MY-PKS-subnet-services
	CIDR	192.168.20.0/22
	Reserved IP Ranges	192.168.20.1-192.168.20.9
	DNS	192.168.20.2
	Gateway	192.168.20.1

4. Select which **Availability Zones** to use with the network.

5. Click **Save**.

 **Note:** Multiple networks allow you to place vCenter on a private network and the rest of your deployment on a public network. Isolating vCenter in this manner denies access to it from outside sources and reduces possible security vulnerabilities.

 **Note:** If you use the Cisco Nexus 1000v Switch, see more information in [Using the Cisco Nexus 1000v Switch with Ops Manager](#) in the PCF documentation.

Step 6: Assign AZs and Networks Page

1. Select **Assign AZs and Networks**.

Assign AZs and Networks

The Ops Manager Director is a single instance.

Choose the availability zone in which to place that instance. It is highly recommended that you backup this VM on a regular basis to preserve settings.

Singleton Availability Zone

AZ1

Network

Deadmines

Save

- Use the drop-down menu to select a **Singleton Availability Zone**. The Ops Manager Director installs in this Availability Zone.
- Use the drop-down menu to select a **Network** for your Ops Manager Director.
- Click **Save**.

Step 7: Security Page

- Select **Security**.

Security

Trusted Certificates

-----BEGIN CERTIFICATE-----

TH

-----END CERTIFICATE-----

These certificates enable BOSH-deployed components to trust a custom root certificate.

Generate VM passwords or use single password for all VMs

☒ Generate passwords
 ☐ Use default BOSH password

Save

- In **Trusted Certificates**, enter a custom certificate authority (CA) certificate to insert into your organization's certificate trust chain. This feature enables all BOSH-deployed components in your deployment to trust a custom root certificate. If you want to use Docker Registries for running app instances in Docker containers, use this field to enter your certificate for your private Docker Registry. For more information, see [Using Docker Registries](#) in the PCF documentation.

3. Choose **Generate passwords** or **Use default BOSH password**. Pivotal recommends that you use the **Generate passwords** option for increased security.
4. Click **Save**. To view your saved Director password, click the **Credentials** tab.

Step 8: Syslog Page

1. Select **Syslog**.

Syslog

Do you want to configure Syslog for Bosh Director?

☐ No
☒ Yes

Address*

The address or host for the syslog server

Port*

Transport Protocol*

TCP

☐ Enable TLS

Permitted Peer*

SSL Certificate*

Save

2. (Optional) To send BOSH Director system logs to a remote server, select **Yes**.
3. In the **Address** field, enter the IP address or DNS name for the remote server.
4. In the **Port** field, enter the port number that the remote server listens on.
5. In the **Transport Protocol** dropdown menu, select **TCP**, **UDP**, or **REL**P. This selection determines which transport protocol is used to send the logs to the remote server.
6. (Optional) Mark the **Enable TLS** checkbox to use TLS encryption when sending logs to the remote server.
 - In the **Permitted Peer** field, enter either the name or SHA1 fingerprint of the remote peer.
 - In the **SSL Certificate** field, enter the SSL certificate for the remote server.
7. Click **Save**.

Step 9: Resource Config Page

1. Select **Resource Config**.

JOB	INSTANCES	PERSISTENT DISK TYPE	VM TYPE
Ops Manager Director	Automatic: 1	Automatic: 50 GB	Automatic: medium.disk (cpu: 2, ram: 4 Gi)
Master Compilation Job	Automatic: 4	None	Automatic: large.cpu (cpu: 4, ram: 4 GB, di)

Save

2. Adjust any values as necessary for your deployment. Under the **Instances**, **Persistent Disk Type**, and **VM Type** fields, choose **Automatic** from the drop-down menu to allocate the recommended resources for the job. If the **Persistent Disk Type** field reads **None**, the job does not require persistent disk space.

Note: Ops Manager requires a Director VM with at least 8 GB memory.

Note: If you set a field to **Automatic** and the recommended resource allocation changes in a future version, Ops Manager automatically uses the updated recommended allocation.

3. Click **Save**.

Step 10: Complete the Ops Manager Installation

1. Click the **Installation Dashboard** link to return to the Installation Dashboard.
2. Click **Apply Changes** on the right navigation.

Next Steps

To install PKS on vSphere **with** NSX-T integration, perform the procedures in [Installing and Configuring PKS with NSX-T Integration](#).

To install PKS on vSphere **without** NSX-T integration, perform the procedures in [Installing and Configuring PKS](#).

To use Harbor to store and manage container images, see [Installing and Integrating VMware Harbor Registry](#).

VMware Harbor Registry

VMware Harbor Registry is an enterprise-class registry server that stores and distributes container images. Harbor allows you to store and manage images for use with Pivotal Container Service (PKS).

Overview

Harbor extends the open source Docker Distribution by adding the functionalities usually required by an enterprise, such as security, identity, and management. As an enterprise private registry, Harbor offers enhanced performance and security. Deploying a registry alongside the PKS environment improves image transfer efficiency.

Key Features

Harbor includes the following key features:

- **Integrated UAA Authentication:** Harbor can share UAA authentication with PAS and PKS.
- **Role-Based Access Control:** Users and repositories are organized into projects. Users can have different permissions for the images in different projects.
- **Policy-Based Image Replication:** Images can be synchronized between multiple registry instances with auto-retry on errors, offering support for load balancing, high availability, multi-datacenter, hybrid, and multi-cloud scenarios.
- **Vulnerability Scanning:** Harbor uses [Clair](#) to scan images regularly and warn users of vulnerabilities.
- **LDAP/Active Directory (AD) Support:** Harbor integrates with an existing enterprise LDAP/AD configuration for user authentication and management.
- **Image Deletion and Garbage Collection:** Images can be deleted and their space can be recycled.
- **Notary:** Image authenticity can be ensured by using Docker Notary.
- **Graphical User Portal:** Users can easily browse, search repositories, and manage projects.
- **Auditing:** All the operations to the repositories are tracked.
- **RESTful API:** RESTful APIs for most administrative operations, easy to integrate with external systems.

Versions and Compatibility

The following table provides version and compatibility information for VMware Harbor Registry.

Element	Details
Tile version	v1.5.1
Release date	June 28, 2018
Software component version	v1.5.1
Compatible Ops Manager version(s)	v2.0.x and v2.1.x
Compatible Pivotal Container Service (PKS) version(s)	v1.1.x
Compatible Pivotal Application Service (PAS) version(s)	v2.0.x and v2.1.x
IaaS support	vSphere and GCP
IPsec support?	No

Requirements

There are no special requirements for deploying VMware Harbor Registry.

Limitations

- You can configure the authentication source only once. You cannot change between UAA, LDAP, or local authentication after the initial deployment.

- Email addresses must be unique. Two users cannot have the same email address.
- Use the Google Chrome browser for the best results. There are known issues with some Firefox browser versions in this release.

If you have a feature request, questions, or information about a bug, please contact [Pivotal Cloud Foundry Feedback](#) or send an email to [Harbor](#).

License

Harbor is available under the following [VMware EULA](#) [↗](#).

Preparing to Install PKS on GCP

This topic outlines the steps for preparing to installing Pivotal Container Service (PKS) on GCP. See the following sections:

- [GCP Prerequisites and Resource Requirements](#)
 - [Preparing to Deploy PKS on GCP](#)
 - [Deploying Ops Manager to GCP](#)
 - [Configuring Ops Manager on GCP](#)
 - [Configuring a GCP Load Balancer for the PKS API](#)
 - [Configuring a GCP Load Balancer for PKS Clusters](#)
-

GCP Prerequisites and Resource Requirements

Page last updated:

This topic describes the prerequisites and resource requirements for installing Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

Resource Requirements

Installing PKS deploys the following two virtual machines (VMs):

VM	CPU	RAM	Storage
Pivotal Container Service	1	4 GB	20 GB
Pivotal Ops Manager	1	8 GB	160 GB

Each Kubernetes cluster provisioned through PKS deploys the VMs listed below. If you deploy more than one Kubernetes cluster, you must scale your allocated resources appropriately.

VM Name	Number	CPU Cores	RAM	Ephemeral Disk	Persistent Disk
master	1	2	4 GB	8 GB	5 GB
worker	1	2	4 GB	8 GB	10 GB

Installing PKS on GCP

To install PKS on GCP, follow the procedures below:

1. [Preparing to Deploy PKS on GCP](#)
 2. [Deploying Ops Manager to GCP](#)
 3. [Configuring Ops Manager on GCP](#)
 4. [Installing and Configuring PKS](#)
-

Preparing to Deploy PKS on GCP

Page last updated:

This guide describes the preparation steps required to install Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

In addition to fulfilling the prerequisites listed in the [GCP Prerequisites and Resource Requirements](#) topic, you must create resources in GCP such as a new network, firewall rules, load balancers, and a service account before deploying PKS. Follow these procedures to prepare your GCP environment.

Step 1: Enable Google Cloud APIs

Ops Manager manages GCP resources using the Google Compute Engine and Cloud Resource Manager APIs. To enable these APIs, perform the following steps:

1. Log in to the Google Developers console at <https://console.developers.google.com>.
2. In the console, navigate to the GCP project where you want to install PKS.
3. Select **Enable APIs & Services** to access the API Library.
4. In the search field, enter `Compute Engine API` and press **Enter**.
5. On the **Google Compute Engine API** page, click **Enable**.
6. In the search field, enter `Cloud Resource Manager API` and press **Enter**.
7. On the **Google Cloud Resource Manager API** page, click **Enable**.
8. To verify that the APIs have been enabled, perform the following steps:

- a. Log in to GCP:

```
$ gcloud auth login
```

- b. List your projects:

```
$ gcloud projects list
PROJECT_ID  NAME               PROJECT_NUMBER
my-project-id  my-project-name  #####
```

This command lists the projects where you enabled Google Cloud APIs.

Step 2: Create Service Accounts

In order for Kubernetes to create load balancers and attach persistent disks to pods, you must create service accounts with sufficient permissions.

You need separate service accounts for Kubernetes cluster master and worker node VMs, and a third account for BOSH and Ops Manager.

Create the Master Node Service Account

1. From the GCP Console, select **IAM & admin > Service accounts**
2. Click **Create Service Account**.
3. Enter a name for the service account, and add the following roles:
 - **Compute Engine**
 - **Storage Admin**
 - **Network Admin**
 - **Security Admin**
 - **Instance Admin (v1)**
 - **Compute Viewer**

- IAM
 - Service Account User

4. Click **Create**.

Create the Worker Node Service Account


1. From the GCP Console, select **IAM & admin > Service accounts**
2. Click **Create Service Account**.
3. Enter a name for the service account, and add the **Compute Engine > Compute Viewer** role.
4. Click **Create**.

Create the BOSH/Ops Manager Service Account

1. From the GCP Console, select **IAM & admin > Service accounts**
2. Click **Create Service Account**.
3. Enter a name for the service account, and add the following roles:
 - **Service Accounts**
 - Service Account User
 - Service Account Token Creator
 - **Compute Engine**
 - Compute Instance Admin (v1)
 - Compute Network Admin
 - Compute Storage Admin
 - **Storage**
 - Storage Admin
4. Select **Furnish a new private key** and select **JSON**.
5. Click **Create**. Your browser automatically downloads a JSON file with a private key for this account. Save this file in a secure location.

Step 3: Create a GCP Network with Subnets

1. Log in to the [GCP Console](#).
2. Navigate to the GCP project where you want to install PKS.
3. Select **VPC network**, then **CREATE VPC NETWORK**.
4. In the **Name** field, enter `your-pks-virt-net`. `your-pks` is a lower-case prefix to help you identify resources for this PKS deployment in the GCP console. Network names must be lower-case. Use the values from the following tables as a guide when you create each network, replacing the IP addresses with ranges that are available in your GCP environment.

 **Note:** Pivotal recommends using all three networks in production environments. You can combine `pks-infrastructure` and `pks-main` into a single network in non-production environments. `pks-services` always requires its own network.

- a. Under **Subnets**, complete the form as follows to create an infrastructure subnet for Ops Manager, the BOSH Director, and NAT instances:

Name	<code>MY-PKS-subnet-infrastructure-GCP-REGION</code>
Region	A region that supports three availability zones (AZs). For help selecting the correct region for your deployment, see Regions and Zones in the Google documentation.
IP address range	A CIDR ending in <code>/26</code>

Example: `192.168.101.0/26`

- b. Click **Add subnet** to add a second subnet for the PKS control plane with the following details:

Name	<code>MY-PKS-subnet-pks-GCP-REGION</code>
Region	The same region you selected for the infrastructure subnet
IP address range	A CIDR ending in <code>/26</code> Example: <code>192.168.16.0/26</code>

- c. Click **Add subnet** to add a third subnet for the Kubernetes clusters with the following details:

Name	<code>MY-PKS-subnet-services-GCP-REGION</code>
Region	The same region you selected for the previous subnets
IP address range	A CIDR in <code>/22</code> Example: <code>192.168.20.0/22</code>

5. Under **Dynamic routing mode**, leave **Regional** selected.

6. Click **Create**.

Step 4: Create NAT Instances

Use NAT instances when you want to expose only a minimal number of public IP addresses.

Creating NAT instances permits Internet access from cluster VMs. You might, for example, need this Internet access for pulling Docker images or enabling Internet access for your workloads.

1. In the console, navigate to **Compute Engine > VM instances**.
2. Click **CREATE INSTANCE**.
3. Complete the following fields:
 - o **Name**: Enter `MY-PKS-nat-gateway-pri`. This is the first, or primary, of three NAT instances you need. If you are using a single AZ, you need only one NAT instance.
 - o **Zone**: Select the first zone from your region. Example: For region `us-west1`, select zone `us-west1-a`.
 - o **Machine type**: Select `n1-standard-4`.
 - o **Boot disk**: Click **Change** and select `Ubuntu 14.04 LTS`.

4. Expand the additional configuration fields by clicking **Management, disks, networking, SSH keys**.

- a. In the **Startup script** field under **Automation**, enter the following text:


```
#!/bin/bash
sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

5. Click **Networking** to open additional network configuration fields:

- a. In the **Network tags** field, add the following: `nat-traverse` and `MY-PKS-nat-instance`.
- b. Click the pencil icon to edit the **Network interface**.
- c. For **Network**, select `your-pks-virt-net`. You created this network in [Step 3: Create a GCP Network with Subnets](#).
- d. For **Subnetwork**, select `MY-PKS-subnet-infrastructure-GCP-REGION`.
- e. For **Primary internal IP**, select `Ephemeral (Custom)`.
- f. Enter an IP address in the **Custom ephemeral IP address** field. Example: `192.168.101.2`. The IP address must meet the following requirements:
 - The IP address must exist in the CIDR range you set for the `MY-PKS-subnet-infrastructure-GCP-REGION` subnet.
 - The IP address must exist in a reserved IP range set later in Ops Manager Director. The reserved range is typically the first `.1` through `.9` addresses in the CIDR range you set for the `MY-PKS-subnet-infrastructure-GCP-REGION` subnet.
 - The IP address cannot be the same as the Gateway IP address set later in Ops Manager. The Gateway IP address is typically the first `.1` address in the CIDR range you set for the `MY-PKS-subnet-infrastructure-GCP-REGION` subnet.
- g. For **External IP**, select `Ephemeral`.
- h. Set **IP forwarding** to `On`.
- i. Click **Done**.

- Click **Create** to finish creating the NAT instance.
- To create additional NAT instances, repeat steps 2-6 using the names and zones specified in the table below.

Instance 2	Name	MY-PKS-nat-gateway-sec
	Zone	Select the second zone from your region. Example: For region <code>us-west1</code> , select zone <code>us-west1-b</code> .
	Internal IP	Select <code>Custom</code> and enter an IP address in the Internal IP address field. Example: <code>192.168.101.3</code> . As described above, this address must in the CIDR range you set for the <code>MY-PKS-subnet-infrastructure-GCP-REGION</code> subnet, must exist in a reserved IP range set later in Ops Manager Director, and cannot be the same as the Gateway IP address set later in Ops Manager.
Instance 3	Name	MY-PKS-nat-gateway-ter
	Zone	Select the third zone from your region. Example: For region <code>us-west1</code> , select zone <code>us-west1-c</code> .
	Internal IP	Select <code>Custom</code> and enter an IP address in the Internal IP address field. Example: <code>192.168.101.4</code> . As described above, this address must in the CIDR range you set for the <code>MY-PKS-subnet-infrastructure-GCP-REGION</code> subnet, must exist in a reserved IP range set later in Ops Manager Director, and cannot be the same as the Gateway IP address set later in Ops Manager.

Create Routes for NAT Instances

- In the GCP console, navigate to **VPC Networks > Routes**.
- Click **CREATE ROUTE**.
- Complete the form as follows:
 - Name:** `MY-PKS-nat-pri`
 - Network:** `your-pks-virt-net`
 - Destination IP range:** `0.0.0.0/0`
 - Priority:** `800`
 - Instance tags:** `MY-PKS`
 - Next hop:** `Specify an instance`
 - Next hop instance:** `MY-PKS-nat-gateway-pri`
- Click **Create** to finish creating the route.
- Repeat steps 2-4 to create two additional routes with the names and next hop instances specified in the table below. The rest of the configuration remains the same.

Route 2	Name: <code>MY-PKS-nat-sec</code> Next hop instance: <code>MY-PKS-nat-gateway-sec</code>
Route 3	Name: <code>MY-PKS-nat-ter</code> Next hop instance: <code>MY-PKS-nat-gateway-ter</code>

Step 5: Create Firewall Rules for the Network

GCP lets you assign [tags](#) to virtual machine (VM) instances and create firewall rules that apply to VMs based on their tags. This step assigns tags and firewall rules to Ops Manager components and VMs that handle incoming traffic.

- From the GCP console, navigate to **VPC network > Firewall rules**.
- Create firewall rules according to the table below:



Note: If you want your firewalls rules to only allow traffic within your private network, modify the **Source IP Ranges** from the table accordingly.

Firewall Rules

Rule 1	<p>This rule allows SSH from public networks.</p> <p>Name: <code>MY-PKS-allow-ssh</code></p> <p>Network: <code>your-pks-virt-net</code></p> <p>Allowed protocols and ports: <code>tcp:22</code></p> <p>Source filter: IP ranges</p> <p>Source IP ranges: <code>0.0.0.0/0</code></p> <p>Target tags: <code>allow-ssh</code></p>
Rule 2	<p>This rule allows HTTP from public networks.</p> <p>Name: <code>MY-PKS-allow-http</code></p> <p>Network: <code>your-pks-virt-net</code></p> <p>Allowed protocols and ports: <code>tcp:80</code></p> <p>Source filter: IP ranges</p> <p>Source IP ranges: <code>0.0.0.0/0</code></p> <p>Target tags: <code>allow-http</code>, <code>router</code></p>
Rule 3	<p>This rule allows HTTPS from public networks.</p> <p>Name: <code>MY-PKS-allow-https</code></p> <p>Network: <code>your-pks-virt-net</code></p> <p>Allowed protocols and ports: <code>tcp:443</code></p> <p>Source filter: IP ranges</p> <p>Source IP ranges: <code>0.0.0.0/0</code></p> <p>Target tags: <code>allow-https</code>, <code>router</code></p>
Rule 4	<p>This rule allows communication between BOSH-deployed jobs.</p> <p>Name: <code>MY-PKS-allow-pks-all</code></p> <p>Network: <code>your-pks-virt-net</code></p> <p>Allowed protocols and ports: <code>tcp;udp;icmp</code></p> <p>Source filter: Source tags</p> <p>Target tags: <code>MY-PKS</code>, <code>MY-PKS-opsman</code>, <code>nat-traverse</code></p> <p>Source tags: <code>MY-PKS</code>, <code>MY-PKS-opsman</code>, <code>nat-traverse</code></p>

3. If you are only using your GCP project to deploy PKS, then you can delete the following default firewall rules:

- o `default-allow-http`
- o `default-allow-https`
- o `default-allow-icmp`
- o `default-allow-internal`
- o `default-allow-rdp`
- o `default-allow-ssh`

Next Steps

To install PKS on GCP, follow the procedures in [Deploying Ops Manager to GCP](#).

Deploying Ops Manager to GCP

Page last updated:

This topic describes how to deploy Ops Manager for Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

After you complete this procedure, follow the instructions in the [Configuring Ops Manager on GCP](#) topic.

Step 1: Locate the Pivotal Ops Manager Installation File

1. Log in to the [Pivotal Network](#), and click on **Pivotal Cloud Foundry Operations Manager**.
2. From the **Releases** drop-down, select the release to install.
3. Select one of the following download files:
 - **Pivotal Cloud Foundry Ops Manager for GCP**
 - **Pivotal Cloud Foundry Ops Manager YAML for GCP**

When you click on the download link, your browser downloads or opens the `OpsManager_version_onGCP.pdf` or `OpsManager_version_onGCP.yml` file.

These documents provide the GCP location of the Ops Manager `.tar.gz` installation file based on the geographic location of your installation.

4. Copy the filepath string of the Ops Manager image based on your deployment location.

Step 2: Create a Private VM Image

1. Log in to the [GCP Console](#).
2. In the left navigation panel, click **Compute Engine**, and select **Images**.
3. Click **Create Image**.
4. Complete the following fields:
 - **Name:** Enter a name. For example, `opsman-pcf-gcp-2-0`.
 - **Encryption:** Leave **Automatic (recommended)** selected.
 - **Source:** Choose **Cloud Storage file**.
 - **Cloud Storage file:** Paste in the Google Cloud Storage filepath you copied from the PDF file in the [previous step](#).
5. Click **Create**. The file may take a few minutes to import.

Step 3: Create the Ops Manager VM Instance

1. Select the checkbox for the image that you created above.
2. Click **Create Instance**.
3. In the **Create an instance form**, complete the following fields:
 - **Name:** Enter a name that matches the naming conventions of your deployment.
 - **Zone:** Choose a zone from the region in which you created your network.
 - **Machine type:** Choose `n1-standard-2`.
 - Click **Customize** to manually configure the vCPU and memory. An Ops Manager VM instance requires the following minimum specifications:

Machine Spec	Minimum Value
CPU	2 vCPUs
Memory	8 GB

- **Boot disk:** Click **Change**, then perform the following steps:
 - Click **Custom images** if it is not already selected.
 - Select the **Boot disk type**. If you have an Ops Manager environment with high performance needs, select **SSD**. As an example, environments used to [develop PCF tiles](#) may benefit from a higher performing Ops Manager VM boot disk. For most environments,

however, you can select **Standard**.

- Set the **Size (GB)** of the boot disk to the minimum or higher.

Machine Spec	Minimum Value
Boot disk	100 GB

- Select the Ops Manager image you created in the previous step if it is not already selected.
- Click **Select** to save.

- Under **Identity and API access**, for the **Service account**, select the BOSH/Ops Manager service account that you created in [Step 2: Create Service Accounts](#) of *Preparing to Deploy PKS on GCP*.
- **Allow HTTP traffic**: Leave this checkbox unselected.
- **Allow HTTPS traffic**: Leave this checkbox unselected.
- **Networking**: Select the **Networking** tab, and perform the following steps:
 - Under **Network interfaces**, perform the following steps:
 - Remove the `default` network interface if this interface still exists.
 - Select the network (for example, `MY-PKS-virt-network`) that you created when preparing your environment in the [Create a GCP Network with Subnets](#) section of the *Preparing to Deploy PKS on GCP* topic.
 - Under **Subnetwork**, select the `MY-PKS-subnet-infrastructure-MY-GCP-REGION` subnet that you created when preparing your environment in the [Create a GCP Network with Subnets](#) section of the *Preparing to Deploy PKS on GCP* topic.
 - For **Primary internal IP**, select **Ephemeral (Custom)**. Enter an IP address (for example, `192.168.101.5`) in the **Custom ephemeral IP address** field. Specify the next available internal IP address located within the reserved IP address range that you will configure in Ops Manager (see [Step 5: Create Networks Page](#)). Do not use the **Gateway IP**, for example `192.168.101.1`. Confirm that the Primary Internal IP you select for OpsManager is from the infrastructure subnet you created in the previous step.
 - For **External IP**, select **Create IP address**. In the next form, enter a name for the static IP. For example, `om-public-ip`. Click **Reserve**. In the **External IP** drop-down, select the static IP address you just reserved.
 - For **Network tags**, enter `MY-PKS-opsman`, `allow-https`, and `allow-ssh`. These tags apply the firewall rules you created in [Create Firewall Rules for the Network](#) to the Ops Manager VM, allowing you to SSH into the Ops Manager VM.

4. Click **Create** to deploy the new Ops Manager VM. This may take a few moments.

5. Navigate to your DNS provider and create an entry that points the fully qualified domain name (FQDN) `opsman.MY-DOMAIN` to the `om-public-ip` external static IP address of Ops Manager that you created in a previous step. For example:

```
opsman.pks.example.com    A    300    192.168.101.5
```



Note: In order to set up Ops Manager authentication correctly, Pivotal recommends using an FQDN to access Ops Manager. Using an ephemeral IP address to access Ops Manager can cause authentication errors upon subsequent access.


Next Steps

After you complete this procedure, follow the instructions in [Configuring Ops Manager on GCP](#).

Configuring Ops Manager on GCP


Page last updated:

This topic describes how to configure Ops Manager for Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

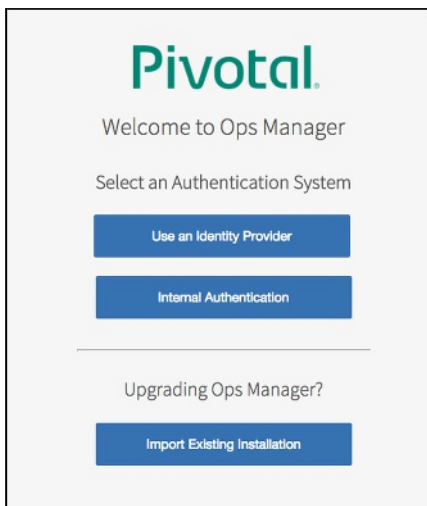
 **Note:** You can also perform the procedures in this topic using the Ops Manager API. For more information, see [Using the Ops Manager API](#).

Step 1: Access Ops Manager

1. In a web browser, navigate to the fully qualified domain name (FQDN) of Ops Manager that you set up in [Deploying Ops Manager to GCP](#). For example, `http://opsman.pks.example.com`.

 **Note:** Using an ephemeral IP address to access Ops Manager can cause authentication errors upon subsequent access. Pivotal recommends accessing Ops Manager using the FQDN.

2. When Ops Manager starts for the first time, you must choose one of the following:
 - [Use an Identity Provider](#): If you use an Identity Provider, an external identity server maintains your user database.
 - [Internal Authentication](#): If you use Internal Authentication, PCF maintains your user database.



Use an Identity Provider (IdP)

1. Log in to your IdP console and download the IdP metadata XML. Optionally, if your IdP supports metadata URL, you can copy the metadata URL instead of the XML.
2. Copy the IdP metadata XML or URL to the Ops Manager [Use an Identity Provider](#) log in page.

Note: The same IdP metadata URL or XML is applied for the BOSH Director. If you use a separate IdP for BOSH, copy the metadata XML or URL from that IdP and enter it into the BOSH IdP Metadata text box in the Ops Manager log in page.

3. Enter your **Decryption passphrase**. Read the **End User License Agreement**, and select the checkbox to accept the terms.

4. Your Ops Manager login page appears. Enter your username and password. Click **Login**.

5. Download your SAML Service Provider metadata (SAML Relying Party metadata) by navigating to the following URLs:

- **5a.** Ops Manager SAML service provider metadata: `https://OPS-MAN-FQDN:443/uaa/saml/metadata`
- **5b.** BOSH Director SAML service provider metadata: `https://BOSH-IP-ADDRESS:8443/saml/metadata`

Note: To retrieve your `BOSH-IP-ADDRESS`, navigate to the **Ops Manager Director** tile > **Status** tab. Record the **Ops Manager Director** IP address.

6. Configure your IdP with your SAML Service Provider metadata. Import the Ops Manager SAML provider metadata from Step 5a above to your IdP. If your IdP does not support importing, provide the values below.

- **Single sign on URL:** `https://OPS-MAN-FQDN:443/uaa/saml/SSO/alias/OPS-MAN-FQDN`
- **Audience URI (SP Entity ID):** `https://OP-MAN-FQDN:443/uaa`
- **Name ID:** Email Address
- SAML authentication requests are always signed

7. Import the BOSH Director SAML provider metadata from Step 5b to your IdP. If the IdP does not support an import, provide the values below.

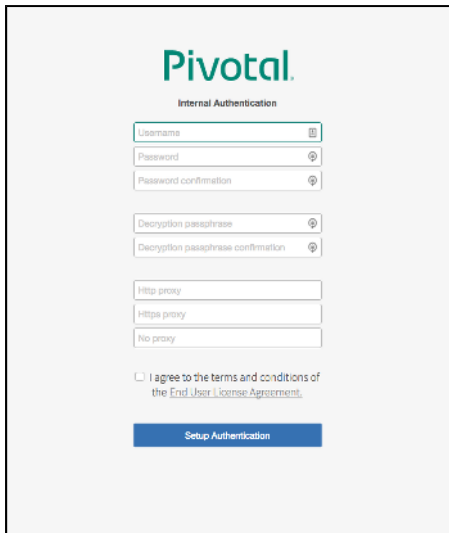
- **Single sign on URL:** `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
- **Audience URI (SP Entity ID):** `https://BOSH-IP:8443`
- **Name ID:** Email Address
- SAML authentication requests are always signed

8. Return to the **Ops Manager Director** tile, and continue with the configuration steps below.

Internal Authentication

1. When redirected to the **Internal Authentication** page, you must complete the following steps:

- Enter a **Username**, **Password**, and **Password confirmation** to create an Admin user.
- Enter a **Decryption passphrase** and the **Decryption passphrase confirmation**. This passphrase encrypts the Ops Manager datastore, and is not recoverable if lost.
- If you use an **HTTP proxy** or **HTTPS proxy**, follow the instructions in [Configuring Proxy Settings for the BOSH CPI](#).
- Read the **End User License Agreement**, and select the checkbox to accept the terms.
- Click **Setup Authentication**.



Pivotal
Internal Authentication

Username

Password

Password confirmation

Decryption passphrase

Decryption passphrase confirmation

Http proxy

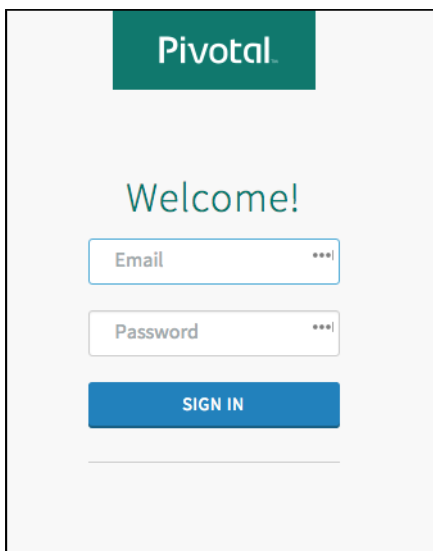
Https proxy

No proxy

☐ I agree to the terms and conditions of the [End User License Agreement](#).

Setup Authentication

2. Log in to Ops Manager with the Admin username and password that you created in the previous step.



Pivotal

Welcome!

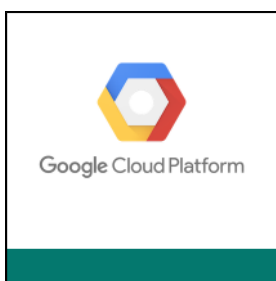
Email

Password


SIGN IN

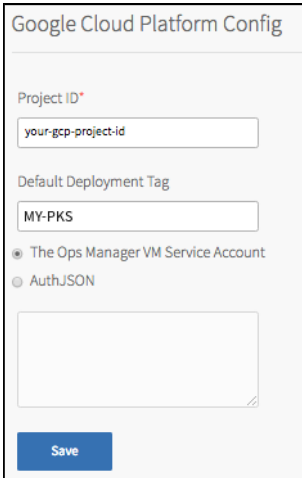
Step 2: Google Cloud Platform Config

1. Click the **Google Cloud Platform** tile within the **Installation Dashboard**.



2. Select **Google Config**. Complete the following fields:
 - **Project ID:** Enter your GCP project ID in all lower case, such as: `your-gcp-project-id`.
 - **Default Deployment Tag:** Enter the `MY-PKS` prefix that you used when creating the GCP resources for this PCF installation. This prefix matches the tags for the `allow-pks-all` firewall rule you created during [Step 5: Create Firewall Rules for the Network](#) in *Preparing to Deploy PKS on GCP*.
 - Select **AuthJSON** and enter the contents of the JSON file that you downloaded for the BOSH/Ops Manager service account that you created in [Step 2: Create Service Accounts](#) in *Preparing to Deploy PKS on GCP*.

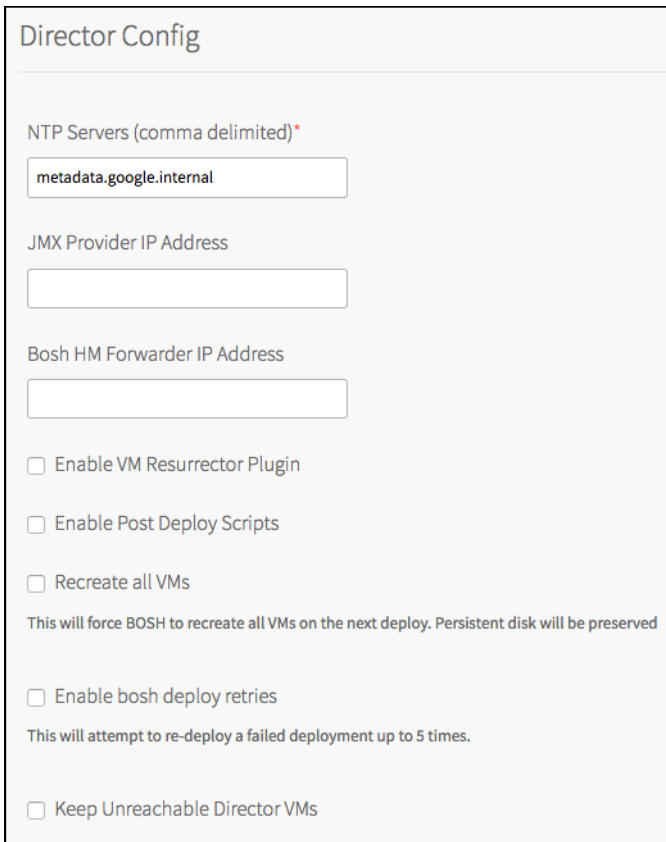
 **Note:** As an alternative, you can select **The Ops Manager VM Service Account** option to use the service account automatically created by GCP for the Ops Manager VM.



3. Click **Save**.


Step 3: Director Config Page

1. Select **Director Config** to open the **Director Config** page.

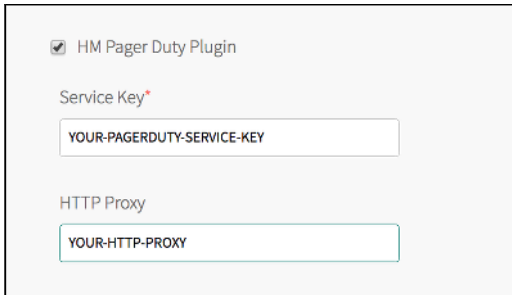


2. In the **NTP Servers (comma delimited)** field, enter `metadata.google.internal`.
3. Leave the **JMX Provider IP Address** field blank.
4. Leave the **Bosh HM Forwarder IP Address** field blank.
5. Select the **Enable VM Resurrector Plugin** checkbox to enable the Ops Manager Resurrector functionality and increase runtime availability.

6. Select **Enable Post Deploy Scripts** to run a post-deploy script after deployment. This script allows the job to execute additional commands against a deployment.

 **Note:** You must enable post-deploy scripts to install PKS.

7. (Optional) Select **Recreate all VMs** to force BOSH to recreate all VMs on the next deploy. This process does not destroy any persistent disk data.
8. Select **Enable bosh deploy retries** for Ops Manager to retry failed BOSH operations up to five times.
9. (Optional) Select **Keep Unreachable Director VMs** if you want to preserve BOSH Director VMs after a failed deployment for troubleshooting purposes.
10. (Optional) Select **HM Pager Duty Plugin** to enable Health Monitor integration with PagerDuty.



☒ HM Pager Duty Plugin

Service Key*

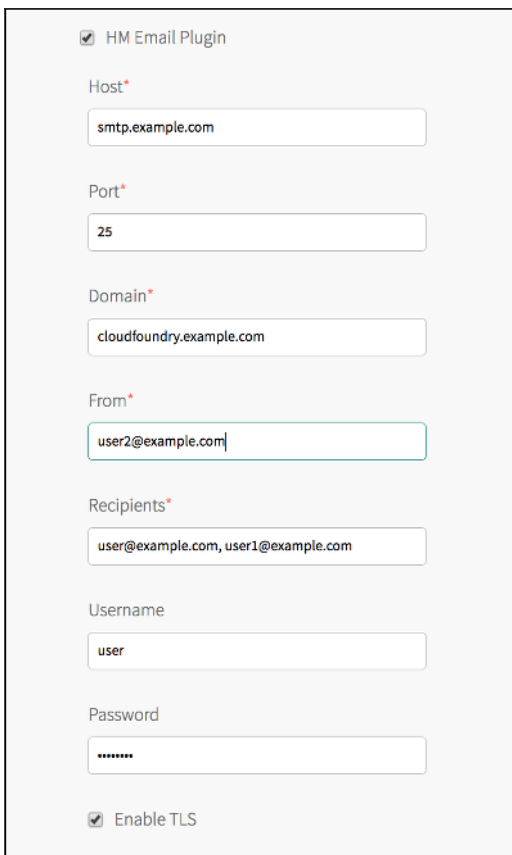
YOUR-PAGERDUTY-SERVICE-KEY

HTTP Proxy

YOUR-HTTP-PROXY

- **Service Key:** Enter your API service key from PagerDuty.
- **HTTP Proxy:** Enter an HTTP proxy for use with PagerDuty.

11. (Optional) Select **HM Email Plugin** to enable Health Monitor integration with email.



☒ HM Email Plugin

Host*

smtp.example.com

Port*

25

Domain*

cloudfoundry.example.com

From*

user2@example.com

Recipients*

user@example.com, user1@example.com

Username


user

Password


☒ Enable TLS

- **Host:** Enter your email hostname.
- **Port:** Enter your email port number.
- **Domain:** Enter your domain.
- **From:** Enter the address for the sender.
- **Recipients:** Enter comma-separated addresses of intended recipients.
- **Username:** Enter the username for your email server.
- **Password:** Enter the password password for your email server.
- **Enable TLS:** Select this checkbox to enable Transport Layer Security.

12. Select a **Blobstore Location** to either configure the blobstore as an internal server or an external endpoint. Because the internal server is unscalable and less secure, Pivotal recommends you configure an external blobstore.

 **Note:** After you deploy Ops Manager, you cannot change the blobstore location.

- **Internal:** Select this option to use an internal blobstore. Ops Manager creates a new VM for blob storage. No additional configuration is required.
- **S3 Compatible Blobstore:** Select this option to use an external S3-compatible endpoint. Follow the procedures in [Sign up for Amazon S3](#) and [Create a Bucket](#) from the AWS documentation. When you have created an S3 bucket, complete the following steps:
 1. **S3 Endpoint:** Navigate to the [Regions and Endpoints](#) topic in the AWS documentation. Locate the endpoint for your region in the **Amazon Simple Storage Service (S3)** table and construct a URL using your region's endpoint. For example, if you are using the `us-west-2` region, the URL you create would be <https://s3-us-west-2.amazonaws.com>. Enter this URL into the **S3 Endpoint** field in Ops Manager.
 2. **Bucket Name:** Enter the name of the S3 bucket.
 3. **Access Key** and **Secret Key:** Enter the keys you generated when creating your S3 bucket.
 4. Select **V2 Signature** or **V4 Signature**. If you select **V4 Signature**, enter your **Region**.

 **Note:** AWS recommends using Signature Version 4. For more information about AWS S3 Signatures, see the [Authenticating Requests](#) documentation.

- **GCS Blobstore:** Select this option to use an external Google Cloud Storage (GCS) endpoint. To create a GCS bucket, follow the procedures in [Creating Storage Buckets](#). When you have created a GCS bucket, complete the following steps:
 1. **Bucket Name:** Enter the name of your GCS bucket.
 2. **Storage Class:** Select the storage class for your GCS bucket. For more information, see [Storage Classes](#) in the GCP documentation.
 3. **Service Account Key:** Enter the contents of the JSON file associated with the service account that you created for BOSH/Ops Manager in [Step 2: Create Service Accounts](#) in *Preparing to Deploy PKS on GCP*.

Blobstore Location

☒ Internal
 ☐ S3 Compatible Blobstore

S3 Endpoint*

Bucket Name*

Access Key*

Secret Key*

☒ V2 Signature
 ☐ V4 Signature

Region*

☐ GCS Blobstore


Bucket Name*

Storage Class*

Regional

Service Account Key*

13. For **Database Location**, select **Internal**.
14. (Optional) Modify the **Director Workers** value, which sets the number of workers available to execute Director tasks. This field defaults to **5**.
15. (Optional) **Max Threads** sets the maximum number of threads that the BOSH Director can run simultaneously. Pivotal recommends that you leave the field blank to use the default value, unless doing so results in rate limiting or errors on your IaaS.
16. Leave the **Director Hostname** field blank.
17. Ensure the **Disable BOSH DNS server for troubleshooting purposes** checkbox is not selected.

 **Note:** BOSH DNS must be enabled in all PKS deployments. If PAS and PKS are running on the same instance of Ops Manager, you cannot use the opt-out feature of BOSH DNS for your PAS without breaking PKS. If you want to opt out of BOSH DNS in your PAS deployment, install the tile on a separate instance of Ops Manager. For more information about opting out of BOSH DNS, see [Disabling or Opting Out of BOSH DNS in PCF](#) on the Pivotal Support website.

18. (Optional) To set a custom banner that users see when logging in to the Director using SSH, enter text in the **Custom SSH Banner** field.

☐ Disable BOSH DNS server for troubleshooting purposes

Custom SSH Banner

19. Click **Save**.

Step 4: Create Availability Zones Page

1. Select **Create Availability Zones**.
2. Click **Add**.
3. For **Google Availability Zone**:
 - Enter one of the zones that you associated to the NAT instances. For example, if you are using the `us-central1` region and selected `us-central1-a` as one of the zones for your NAT instances, enter `us-central1-a`.
 - Click **Add**.
 - Repeat the above step for all the availability zones that you associated to instances in [Step 4: Create NAT Instances](#) in *Preparing to Deploy PKS on GCP*.

Create Availability Zones

Availability Zones

Add

▶ us-central1-b

▼ us-central1-a

Google Availability Zone*

us-central1-a

 The Google Availability Zone name


▶ us-central1-c

Save

- Click **Save**.
4. Repeat the above step for each availability zone you use in your deployment. When you are done, click **Save**.

Step 5: Create Networks Page

1. Select **Create Networks**.
2. Make sure **Enable ICMP checks** is not selected. GCP routers do not respond to ICMP pings.
3. Click **Add Network** and create the following networks:
 - `pks-infrastructure` for Ops Manager, the BOSH Director, and NAT instances.
 - `pks-main` for the PKS control plane. In non-production environments, you can choose to combine `pks-infrastructure` and `pks-main` into a single network.
 - `pks-services` for creating the master and worker VMs for Kubernetes clusters.

 **Note:** Pivotal recommends that you use the Google-provided DNS server, `169.254.169.254`, as your default DNS server. Do not use `8.8.8.8`.

Infrastructure Network	Field	Configuration
	Name	pks-infrastructure
	Service Network	Leave Service Network unchecked.
	Google Network Name	MY-PKS-virt-net/MY-PKS-subnet-infrastructure-GCP-REGION/GCP-REGION
	CIDR	192.168.101.0/26
	Reserved IP Ranges	192.168.101.1-192.168.101.9
	DNS	169.254.169.254
	Gateway	192.168.101.1
Main Network	Field	Configuration
	Name	pks-main
	Service Network	Leave Service Network unchecked.
	Google Network Name	MY-PKS-virt-net/MY-PKS-subnet-pks-GCP-REGION/GCP-REGION
	CIDR	192.168.16.0/26
	Reserved IP Ranges	192.168.16.1-192.168.16.9
	DNS	169.254.169.254
	Gateway	192.168.16.1
Service Network	Field	Configuration
	Name	pks-services
	Service Network	Select the Service Network checkbox.
	Google Network Name	MY-PKS-virt-net/MY-PKS-subnet-services-GCP-REGION/GCP-REGION
	CIDR	192.168.20.0/22
	Reserved IP Ranges	192.168.20.1-192.168.20.9
	DNS	169.254.169.254
	Gateway	192.168.20.1

Step 6: Assign AZs and Networks Page

1. Select **Assign AZs and Networks**.
2. Use the drop-down menu to select a **Singleton Availability Zone**. The BOSH Director installs in this Availability Zone.
3. Under **Network**, select the pks-infrastructure network for your BOSH Director.
4. Click **Save**.

Step 7: Security Page

1. Select **Security**.

Security

Trusted Certificates

-----BEGIN CERTIFICATE-----
TH [REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

These certificates enable BOSH-deployed components to trust a custom root certificate.

Generate VM passwords or use single password for all VMs

- ☒ Generate passwords
- ☐ Use default BOSH password

Save

2. In **Trusted Certificates**, enter a custom certificate authority (CA) certificate to insert into your organization's certificate trust chain. This feature enables all BOSH-deployed components in your deployment to trust a custom root certificate.
 - o You do not need to enter anything in this field if you are using self-signed certificates.
 - o If you want to use Docker Registries for running app instances in Docker containers, enter the certificate for your private Docker Registry in this field. For more information, see [Using Docker Registries](#).
3. Choose **Generate passwords** or **Use default BOSH password**. Pivotal recommends that you use the **Generate passwords** option for greater security.
4. Click **Save**. To view your saved Director password, click the **Credentials** tab.

Step 8: Syslog Page

1. Select **Syslog**.

Syslog

Do you want to configure Syslog for Bosh Director?

☐ No
 ☒ Yes

Address*

The address or host for the syslog server

Port*

Transport Protocol*

TCP

☐ Enable TLS

Permitted Peer*

SSL Certificate*

Save

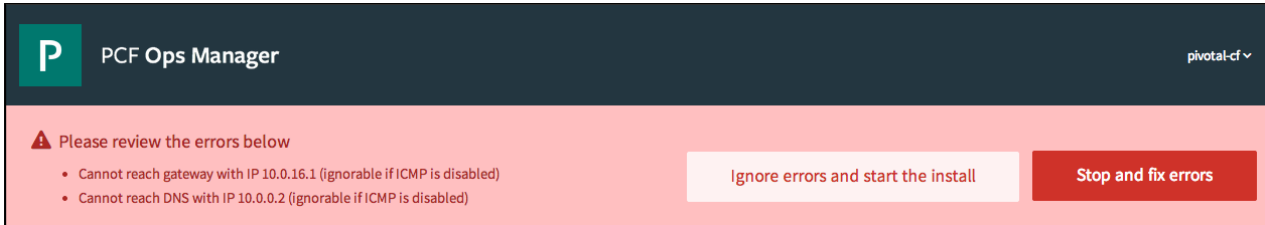
- (Optional) To send BOSH Director system logs to a remote server, select **Yes**.
- In the **Address** field, enter the IP address or DNS name for the remote server.
- In the **Port** field, enter the port number that the remote server listens on.
- In the **Transport Protocol** dropdown menu, select **TCP**, **UDP**, or **REL**. This selection determines which transport protocol is used to send the logs to the remote server.
- (Optional) Mark the **Enable TLS** checkbox to use TLS encryption when sending logs to the remote server.
 - In the **Permitted Peer** field, enter either the name or SHA1 fingerprint of the remote peer.
 - In the **SSL Certificate** field, enter the SSL certificate for the remote server.
- Click **Save**.

Step 9: Resource Config Page

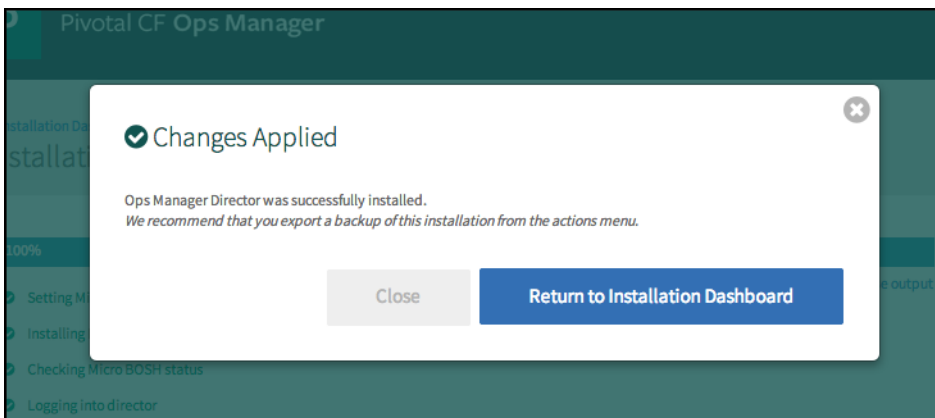
- Select **Resource Config**.
- Ensure that the **Internet Connected** checkboxes are not selected for any jobs. This checkbox gives VMs a public IP address that enables outbound Internet access. In [Preparing to Deploy PKS on GCP](#), you provisioned a Network Address Translation (NAT) box to provide Internet connectivity to your VMs. For more information about using NAT in GCP, see [Virtual Private Cloud \(VPC\) Network Overview](#) in the GCP documentation.

Step 10: Complete the Ops Manager Director Installation

1. Click the **Installation Dashboard** link to return to the Installation Dashboard.
2. Click **Apply Changes**. If the following ICMP error message appears, return to the [Network Config](#) screen, and make sure you have deselected the **Enable ICMP Checks** box. Then click **Apply Changes** again.



3. Ops Manager Director installs. This may take a few moments. When the installation process successfully completes, the **Changes Applied** window appears.



Next Steps

After you complete this procedure, follow the instructions in [Configuring a GCP Load Balancer for the PKS API](#).


Configuring a GCP Load Balancer for the PKS API

Page last updated:

This topic describes how to create a load balancer for the PKS API using Google Cloud Platform (GCP).

Before you install PKS, you must configure an external TCP load balancer to access the PKS API from outside the network. You can use any external TCP load balancer of your choice.

Refer to the procedures in this topic to create a load balancer using GCP. If you choose to use a different load balancer, use the configuration in this topic as a guide.

 **Note:** This procedure uses example commands which you should modify to represent the details of your PKS installation.

Step 1: Create a Load Balancer

To create a load balancer using GCP, perform the following steps:

1. In a browser, navigate to the [GCP console](#).
2. Navigate to **Network Services > Load balancing** and click **CREATE LOAD BALANCER**.
3. Under **TCP Load Balancing**, click **Start configuration**.
4. Select whether you want to load balance traffic from the Internet to your VMs or only between your VMs.
5. Select whether you want to place the backends for your load balancer in a single region or across multiple regions.
6. Give your load balancer a name. Pivotal recommends naming your load balancer `pkc-api`.
7. Select **Backend configuration**.
 - Under **Region**, select the region where you deployed Ops Manager.
 - Under **Backends**, leave **Select existing instances** selected. This will be automatically configured when updating the [Resource Config](#) section of the PKS tile.
 - (Optional) Select a backup pool.
 - (Optional) Select whether you want to create a health check or go without one.
 - Select a session affinity configuration.
 - (Optional) Select **Advanced configurations** to configure the **Connection draining timeout**.
8. Select **Frontend configuration**.
 - (Optional) Give your frontend a name.
 - (Optional) Give your frontend a description.
 - Select **Create IP address** to reserve an IP address for the PKS API endpoint.
 1. Enter a name for your reserved IP address. For example, `pkc-api-ip`. GCP assigns a static IP address that appears next to the name.
 2. (Optional) Enter a description.
 3. Click **Reserve**.
 - Under **Ports**, enter 8443 and 9021. Your external load balancer forwards traffic to the PKS control plane VM using the UAA endpoint on port 8443 and the PKS API endpoint on port 9021.
 - Click **Done**.
9. Click **Review and finalize** to review your load balancer configuration.
10. Click **Create**.

Step 2: Create a Firewall Rule

To create a firewall rule that allows traffic between the load balancer and the PKS API VM, do the following:

1. From the GCP console, navigate to **VPC Network > Firewall rules** and click **CREATE FIREWALL RULE**.

2. Configure the following:
 - Give your firewall rule a name.
 - (Optional) Give your firewall rule a description.
 - Under **Network**, select the VPC network you created in [Step 3: Create a GCP Network with Subnets](#) of *Preparing to Deploy PKS on GCP*.
 - Under **Priority**, enter a priority number between 0 and 65535.
 - Under **Direction of traffic**, select **Ingress**.
 - Under **Action on match**, select **Allow**.
 - Under **Targets**, select the load balancer you created in the previous section.
 - Under **Target tags**, enter `pkcs-api`.
 - Under **Source filter**, select **IP ranges**.
 - Under **Source IP ranges**, enter `0.0.0.0/0`.
 - Under **Protocols and ports**, select **Specified protocols and ports** and enter `tcp:8443,9021`.
3. Click **Create**.

Step 3: Create a Network Tag for the Firewall Rule

To apply the firewall rule to the VM that hosts the PKS API, the VM must have the `pkcs-api` tag in GCP. Do the following:

1. From the GCP console, navigate to **Compute Engine > VM instances**.
2. Locate the your PKS control plane VM.
3. Click the name of the VM to open the **VM instance details** menu.
4. Click **Edit**.
5. Verify that the **Network tags** field contains the `pkcs-api` tag. If the tag does not appear in the field, enter it now.
6. Scroll to the bottom of the screen and click **Save**.

Step 4: Create a Wildcard DNS Entry

To create a wildcard DNS entry in GCP for your PKS API domain, do the following:

1. From the GCP console, navigate to **Network Services > Cloud DNS**.
2. If you do not already have a DNS zone, click **Create zone**.
 - Give a **Zone name** and a **DNS name**.
 - Specify whether the **DNSSEC** state of the zone is **Off**, **On**, or **Transfer**.
 - (Optional) Enter a **Description**.
 - Click **Create**.
3. Click **Add record set**.
4. Under **DNS Name**, enter a subdomain for the load balancer. For example, to use `pkcs-api.pks.example.com` as your PKS API hostname, enter `pkcs-api` in this field.
5. Select a **Resource Record Type**, **TTL**, and **TTL Unit**.
6. Enter the static IP address that GCP assigned when you created the load balancer in [Step 1: Create a Load Balancer](#).
7. Click **Create**.

Next Steps

After you complete this procedure, follow the instructions in [Installing and Configuring PKS](#) to deploy PKS.

Configuring a GCP Load Balancer for PKS Clusters

A load balancer is a third party device that distributes network and application traffic across resources. You can use a load balancer to secure and facilitate access to a PKS cluster from outside the network. Using a load balancer can also prevent individual network components from being overloaded by high traffic.

The procedure below explains how to create a Google Cloud Platform (GCP) load balancer for your PKS cluster. Using a GCP load balancer is optional, but adding one to your Kubernetes cluster can make it easier to manage the cluster via the PKS API and `kubectl`.

Overview

To configure a GCP load balancer with a PKS cluster, you must connect a load balancer to the cluster, funnel inbound traffic to the load balancer, and set a firewall rule to allow traffic into the cluster itself.

The steps of this procedure are summarized below:


First, use the GCP Console to [create a load balancer](#) for the a new cluster. Configure the frontend and backend of the load balancer; the backend configuration connects the load balancer to the PKS cluster, and the frontend configuration tells inbound traffic how to contact the load balancer successfully.

Next, [create a network tag](#) for the master VMs in your PKS cluster. This lets you associate a firewall rule with the master VMs, ensuring that traffic accessing the cluster via the load balancer is subject to the firewall rule.

Finally, [configure a firewall rule](#) with permission information about which types of traffic can access your cluster.


Prerequisites

- To complete these procedures, you must have already configured a separate external load balancer to access the PKS API.
- The version of the PKS CLI you are using must match the version of the PKS tile you are installing.


 **Note:** This procedure uses example commands which you should modify to represent the details of your PKS installation.

Creating GCP Load Balancers for PKS Clusters

1. Log in to the PKS API and create a cluster. For more information, see [Create a Cluster](#).
2. Navigate to [Google Cloud Platform](#).
3. In the sidebar menu, select **Network Services** > **Load balancing**.
4. Click **Create Load Balancer**.
5. In the **TCP Load Balancing** pane, click **Start configuration**.
6. Click **Continue**. The **New TCP load balancer** menu opens.
7. Enter a **Name** for your load balancer and click **Backend configuration**. The **Backend configuration** pane opens.

 **Note:** Configure a TCP Load Balancer with a human-readable name in lower case letters, such as `your-pks-cluster-api`.

8. Configure the load balancer backend.
 - a. Choose the **Region** in which the cluster is deployed.
 - b. Click **Select existing instances**.
 - c. Select all master VMs for your cluster from the dropdown. To locate the IP addresses and VM IDs of the master VMs, see [Identify Kubernetes Cluster Master VMs](#).

 **Breaking Change:** If master VMs are recreated for any reason, such as a stemcell upgrade, you must reconfigure the load balancer to target the new master VMs. For more information, see the [Reconfiguring a GCP Load Balancer](#) section below.

- d. Specify any other configuration options you require and click **Done** to complete backend configuration.



Note: For clusters with multiple master node VMs, health checks on port 8443 are recommended.

9. Click **Frontend configuration**. The **Frontend Configuration** pane opens.
10. Configure the load balancer frontend.
 - a. Optional: Enter a human-readable name in lower case letters, such as `pks-cluster-api`.
 - b. Click **IP**.
 - c. Select **Create IP address**.
 - d. Give the IP address a human-readable name and click **Reserve**.
 - e. In the **Port** field, enter `8443`.
 - f. Click **Done** to complete frontend configuration.
11. Review your load balancer configuration and click **Create**.

Creating a Network Tag

1. In the Google Cloud Platform sidebar menu, select **Compute Engine > VM instances**.
2. Filter to find the master instances of your cluster. Type `master` in the **Filter VM Instances** search box and press **Enter**.
3. Click the name of the master instances. The **VM instance details** menu opens.
4. Click **Edit**.
5. Click in the **Network tags** field and type a human-readable name in lower case letters. Press **Enter** to create the network tag.
6. Scroll to the bottom of the screen and click **Save**.

Creating Firewall Rules for Load Balancers

1. In the Google Cloud Platform sidebar menu, select **VPC Network > Firewall Rules**.
2. Click **Create Firewall Rule**. The **Create a firewall rule** menu opens.
3. Give your firewall rule a human-readable name in lower case letters. For ease of use, you may want to align this name with the name of the load balancer you created in [Creating Load Balancers for PKS Clusters](#).
4. In the **Network** menu, select the VPC network on which you have deployed the PKS tile.
5. In the **Direction of traffic** field, select **Ingress**.
6. In the **Action on match** field, select **Allow**.
7. Confirm that the **Targets** menu is set to `Specified target tags` and enter the tag you made in [Creating a Network Tag](#) in the **Target tags** field.
8. In the **Source filter** field, choose an option to filter source traffic.
9. Based on your choice in the **Source filter** field, specify IP addresses, Subnets, or Source tags to allow access to your cluster.
10. In the **Protocols and ports** field, choose **Specified protocols and ports** and enter the port number you specified in [Creating Load Balancers for PKS Clusters](#), prepended by `tcp:`. For example: `tcp:8443`.
11. Specify any other configuration options you require and click **Done** to complete frontend configuration.
12. Click **Create**.

Reconfiguring a GCP Load Balancer

If Kubernetes master node VMs are recreated for any reason, you must reconfigure your cluster load balancers to point to the new master VMs. For

example, after a stemcell upgrade, BOSH recreates the VMs in your deployment.

To reconfigure your GCP cluster load balancer to use the new master VMs, do the following:

1. Locate the VM IDs of the new master node VMs for the cluster. For information about locating the VM IDs, see [Identify Kubernetes Cluster Master VMs](#).
 2. Navigate to the [GCP console](#).
 3. In the sidebar menu, select **Network Services > Load balancing**.
 4. Select your cluster load balancer and click **Edit**.
 5. Click **Backend configuration**.
 6. Click **Select existing instances**.
 7. Select the new master VM IDs from the dropdown. Use the VM IDs you located in the first step of this procedure.
 8. Click **Update**.
 9. In the sidebar menu, select **Cloud DNS**.
 10. Select the zone where your load balancer is deployed.
 11. Click **Add record set**.
 12. Under **DNS Name**, enter a subdomain for the load balancer. For example, to use `my-cluster.example.com` as your cluster hostname, enter `my-cluster` in this field.
 13. Select a **Resource Record Type**, **TTL**, and **TTL Unit**.
 14. Enter the IP address for the master node of the cluster. You located this IP address in the `pks cluster` output earlier in this procedure.
 15. Click **Create**.
-

Installing PKS

Page last updated:

This topic describes how to install and configure the Pivotal Container Service (PKS) tile. See the following topics:

- [Installing and Configuring PKS](#)
 - [Installing and Configuring PKS on vSphere with NSX-T Integration](#)
-

Installing and Configuring PKS

Page last updated:

This topic describes how to install and configure Pivotal Container Service (PKS).

Prerequisites

Before performing the procedures in this topic, you must have deployed and configured Ops Manager. For more information, see the prerequisites for your cloud provider:

- [GCP Prerequisites and Resource Requirements](#)
- [vSphere Prerequisites and Resource Requirements](#)

If you are using an instance of Ops Manager that you configured previously to install other runtimes, confirm the following settings before you install PKS:

1. Navigate to Ops Manager.
2. From the **Director Config** pane, do the following:
 - a. Select the **Enable Post Deploy Scripts** checkbox.
 - b. Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
3. Click the **Installation Dashboard** link to return to the Installation Dashboard.
4. Click **Apply Changes**.

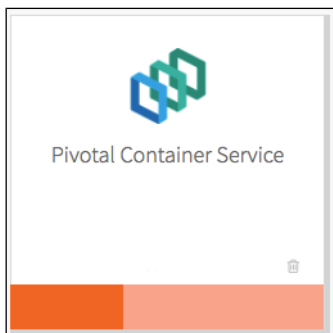
Step 1: Install PKS

To install PKS, do the following:

1. Download the product file from [Pivotal Network](#).
2. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
3. Click **Import a Product** to upload the product file.
4. Under **Pivotal Container Service** in the left column, click the plus sign to add this product to your staging area.

Step 2: Configure PKS


Click the orange **Pivotal Container Service** tile to start the configuration process.



Assign AZs and Networks

Perform the following steps:

1. Click **Assign AZs and Networks**.
2. Select the availability zone (AZ) where you want to deploy the PKS API VM as a singleton job.

 **Note:** You must select an additional AZ for balancing other jobs before clicking **Save**, but this selection has no effect in the current version of PKS.

Place singleton jobs in

☒ us-west-2a

☐ us-west-2b

☐ us-west-2c

Balance other jobs in

☐ us-west-2a

☒ us-west-2b

☐ us-west-2c

Network

pk-infrastructure

Service Network

pk-services

Save

3. Under **Network**, select the infrastructure subnet you created for the PKS API VM.
4. Under **Service Network**, select the services subnet you created for Kubernetes cluster VMs.
5. Click **Save**.

PKS API

Perform the following steps:

1. Click **PKS API**.
2. Under **Certificate to secure the PKS API**, provide your own wildcard certificate and private key pair.

Certificate to secure the PKS API ^{*}

-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----

Change

API Hostname (FQDN) ^{*}

pkcs.EXAMPLE.com

The certificate you enter here should cover the domain that routes to the PKS broker VM with TLS termination on the ingress.

(Optional) If you do not have a certificate and private key pair, you can have Ops Manager generate one for you. Perform the following steps:

- a. Select the **Generate RSA Certificate** link.
 - b. Enter the wildcard domain for your API hostname. For example, if your PKS API domain is `api.pks.example.com`, then enter `*.pks.example.com`.
 - c. Click **Generate**.
3. Under **API Hostname (FQDN)**, enter a fully qualified domain name (FQDN) to access the PKS API. For example, `api.pks.example.com`.
 4. Click **Save**.


Usage Data

VMware's Customer Experience Improvement Program (CEIP) and the Pivotal Telemetry Program (Telemetry) provides VMware and Pivotal with information that enables the companies to improve their products and services, fix problems, and advise you on how best to deploy and use our products. As part of the CEIP and Telemetry, VMware and Pivotal collect technical information about your organization's use of the Pivotal Container Service ("PKS") on a regular basis. Since PKS is jointly developed and sold by VMware and Pivotal, we will share this information with one another. Information collected under CEIP or Telemetry does not personally identify any individual.

Regardless of your selection in the **Usage Data** pane, a small amount of data is sent from Cloud Foundry Container Runtime (CFCR) to the PKS tile. However, that data is not shared externally.

To configure the **Usage Data** pane:


1. Select the **Usage Data** side-tab.
2. Read the Usage Data description.
3. Make your selection.
 - a. To join the program, select **Yes, I want to join the CEIP and Telemetry Program for PKS**.
 - b. To decline joining the program, select **No, I do not want to join the CEIP and Telemetry Program for PKS**.
4. Click **Save**.

 **Note:** If you choose to join the CEIP and Telemetry Program for PKS, open your firewall to allow outgoing access on port 443 to the following URL: <https://vcsa.vmware.com/ph-prd>.

Plans

To activate a plan, perform the following steps:

1. Click the **Plan 1**, **Plan 2**, or **Plan 3** tab.


 **Note:** A plan defines a set of resource types used for deploying clusters. You can configure up to three plans. You must configure **Plan 1**.

2. Select **Active** to activate the plan and make it available to developers deploying clusters.

Plan*

☒ Active


Name*

small 

Description*

Example: This plan will configure a lightweight kubernetes cluster. Not recommended for production workloads.


Default Cluster Authorization Mode*

RBAC 


Master/ETCD Node Instances (min: 1, max: 3)*

1

Master/ETCD VM Type*

medium (cpu: 2, ram: 4 GB, disk: 8 GB) 

Master Persistent Disk Type*

10 GB 

Master/ETCD Availability Zones*

☐ us-central1-f

☒ us-central1-a

☐ us-central1-c

3. Under **Name**, provide a unique name for the plan.

4. Under **Description**, edit the description as needed. The plan description appears in the Services Marketplace, which developers can access by using PKS CLI.

5. Under **Master/ETCD Node Instances**, select the default number of Kubernetes master/etcd nodes to provision for each cluster. You can enter either **1** or **3**. Set this value to **3** for increased master node availability.

WARNING: To change the number of master/etcd nodes, you must ensure that no existing clusters use this plan. PKS does not support changing the number of master/etcd nodes for plans with existing clusters.

WARNING: This feature is a beta component and is intended for evaluation and test purposes only. Do not use this feature in a production environment. Product support and future availability are not guaranteed for beta components.

6. Under **Master/ETCD VM Type**, select the type of VM to use for Kubernetes master/etcd nodes.
7. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master node VM.
8. Under **Master/ETCD Availability Zones**, select one or more AZs for the Kubernetes clusters deployed by PKS. If you select more than one AZ, PKS deploys the master VM in the first AZ and the worker VMs across the remaining AZs.
9. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, create clusters with a minimum of three worker nodes, or two per AZ if you intend to use persistent volumes. For example, if you deploy across three AZs, you should have six worker nodes. For more information about persistent volumes, see [Persistent Volumes](#) in *Maintain Workload Uptime*. Provisioning a minimum of three worker nodes, or two nodes per AZ is also recommended for stateless workloads.

Worker Node Instances (min: 1, max: 50) *

Worker VM Type*

Worker Persistent Disk Type*

Worker Availability Zones *
☐ us-central1-f
☒ us-central1-a
☐ us-central1-c

Errand VM Type*

10. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker node VMs.
11. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker node VMs.
12. Under **Worker Availability Zones**, select one or more AZs for the Kubernetes worker nodes. PKS deploys worker nodes equally across the AZs you select.
13. Under **Errand VM Type**, select the size of the VM that contains the errand. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.
14. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to add custom workloads to each cluster in this plan. You can specify multiple files using `---` as a separator. For more information, see [Add Custom Workloads](#).

(Optional) Add-ons - Use with caution

☐ Enable Privileged Containers - Use with caution
☐ Disable DenyEscalatingExec

15. (Optional) To allow users to create pods with privileged containers, select the **Enable Privileged Containers - Use with caution** option. For more information, see [Pods](#) in the Kubernetes documentation.
16. (Optional) To disable the admission controller, select the **Disable DenyEscalatingExec** checkbox. If you select this option, clusters in this plan can create security vulnerabilities that may impact other tiles. Use this feature with caution.
17. Click **Save**.

To deactivate a plan, perform the following steps:

1. Click the **Plan 1**, **Plan 2**, or **Plan 3** tab.
2. Select **Plan Inactive**.
3. Click **Save**.

Kubernetes Cloud Provider

To configure your Kubernetes cloud provider settings, follow the procedure for your cloud provider.

1. Click **Kubernetes Cloud Provider**.
2. Under **Choose your IaaS**, select either **vSphere** or **GCP**.
3. Follow the procedures for your cloud provider below.

vSphere

In the procedure below, you use credentials for vCenter master VMs. You must have provisioned the service account with the correct permissions. For more information, see [Create the Master Node Service Account](#).

Ensure the values in the following procedure match those in the **vCenter Config** section of the **Ops Manager** tile.

Choose your IaaS*

☐ GCP

☒ vSphere

vCenter Master Credentials *

Username

Password

vCenter Host *

Datacenter Name *

Datastore Name *

Stored VM Folder *

1. Enter your **vCenter Master Credentials**. See [Preparing to Deploy PKS on vSphere](#) for more information.
2. Enter your **vCenter Host**. For example: `vcenter.CF-EXAMPLE.com`.
3. Enter your **Datacenter Name**. For example: `CF-EXAMPLE-dc`.
4. Enter your **Datastore Name**. For example: `CF-EXAMPLE-ds`.
5. Enter the **Stored VM Folder** so that the persistent stores know where to find the VMs. To retrieve the name of the folder, navigate to your Ops Manager Director tile, click **vCenter Config**, and locate the value for **VM Folder**. The default folder name is `pcf_vms`.
6. Click **Save**.

GCP

Ensure the values in the following procedure match those in the **Google Config** section of the **Ops Manager** tile.

Choose your IaaS*

☒ GCP

GCP Project ID *

VPC Network *

GCP Master Service Account ID *

GCP Worker Service Account ID *

☐ vSphere

1. Enter your **GCP Project ID**, which is the name of the deployment in your Ops Manager environment.
2. Enter your **VPC Network**, which is the VPC network name for your Ops Manager environment.
3. Enter your **GCP Master Service Account ID**. This is the email address associated with the master node service account. For information about configuring this account, see [Create the Master Node Service Account](#).
4. Enter your **GCP Worker Service Account ID**. This is the email address associated with the worker node service account. For information about configuring this account, see [Create the Worker Node Service Account](#).
5. Click **Save**.

(Optional) Logging

You can designate an external syslog endpoint for PKS component and cluster log messages.

To specify the destination for PKS log messages, do the following:

1. Click **Logging**.
2. To enable syslog forwarding, select **Yes**.

Enable Syslog for PKS?*

☐ No

☒ Yes

Address *

Port *

Transport Protocol*


TCP

☒ Enable TLS


Permitted Peer

TLS Certificate

3. Under **Address**, enter the destination syslog endpoint.
4. Under **Port**, enter the destination syslog port.
5. Select a transport protocol for log forwarding.
6. (Optional) Pivotal strongly recommends that you enable TLS encryption when forwarding logs as they may contain sensitive information. For example, these logs may contain cloud provider credentials. To enable TLS, perform the following steps:
 - a. Under **Permitter Peer**, provide the accepted fingerprint (SHA1) or name of remote peer. For example, `*.YOUR-LOGGING-SYSTEM.com`.
 - b. Under **TLS Certificate**, provide a TLS certificate for the destination syslog endpoint.

 **Note:** You do not need to provide a new certificate if the TLS certificate for the destination syslog endpoint is signed by a Certificate Authority (CA) in your BOSH certificate store.

7. Follow the procedure for your cloud provider:
 - If you are installing PKS on GCP, click **Save** and continue to the next section.
 - If you are installing PKS on vSphere, you can manage logs using [VMware vRealize Log Insight \(vRLI\)](#). The integration pulls logs from all BOSH jobs and containers running in the cluster (node logs from core Kubernetes & BOSH processes, Kubernetes event logs, as well as POD std out and std err).

 **Note:** Before you configure the vRLI integration, you must have a vRLI license and vRLI must be installed, running, and available in your environment. You need to provide the live instance address during configuration. For instructions and additional information, see the [vRealize Log Insight documentation](#).

By default, vRLI logging is disabled. To enable and configure vRLI logging, perform the following steps:

1. Under **Enable VMware vRealize Log Insight Integration?**, select **Yes**.

Enable VMware vRealize Log Insight Integration?*

☐ No
☒ Yes

Host *

☒ Enable SSL?

☐ Disable SSL certificate validation

CA certificate

Rate limiting *

0

2. Under **Host**, enter the IP address or FQDN of the vRLI host.
3. (Optional) Select the **Enable SSL?** checkbox to encrypt the logs being sent to vRLI using SSL.
4. Choose one of the following SSL certificate validation options:

- To skip certificate validation for the vRLI host, select the **Disable SSL certificate validation** checkbox. Select this option if you are using a self-signed certificate in order to simplify setup for a development or test environment.

Note: Disabling certificate validation is not recommended for production environments.

- To enable certificate validation for the vRLI host, clear the **Disable SSL certificate validation** checkbox.
 - (Optional) If your vRLI certificate is not signed by a trusted CA root or other well known certificate, enter the certificate in the **CA certificate** field. Locate the PEM of the CA used to sign the vRLI certificate, copy the contents of the certificate file, and paste them into the field. Certificates must be in PEM-encoded format.

5. Under **Rate limiting**, enter a time in milliseconds to change the rate at which logs are sent to the vRLI host. The rate limit specifies the minimum time between messages before the fluentd agent begins to drop messages. The default value (0) means the rate is not limited, which suffices for many deployments.

Note: If your deployment is generating a high volume of logs, you can increase this value to limit network traffic. Consider starting with a lower number, such as 10, and tuning to optimize for your deployment. A large number might result in dropping too many log entries.

6. Click **Save**. This configuration applies to any clusters created after you have saved these configuration settings and clicked **Apply Changes**.

Note: The PKS tile does not validate your vRLI configuration settings. To verify your setup, look for log entries in vRLI.

Networking

To configure networking, do the following:

1. Click **Networking**.
2. Under **Container Networking Interface**, select **Flannel**.

Container Networking Interface*

☒ Flannel
 ☐ NSX-T

HTTP/HTTPS Proxy (for vSphere only)*

☒ Disabled
 ☐ Enabled

Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)

☐ Enable outbound internet access

3. (Optional) If you are installing PKS on vSphere, configure a global proxy for all outgoing HTTP and HTTPS traffic from your Kubernetes clusters.

Production environments can deny direct access to public Internet services and between internal services by placing an HTTP or HTTPS proxy in the network path between Kubernetes nodes and those services.

If your environment includes HTTP or HTTPS proxies, configuring PKS to use these proxies allows PKS-deployed Kubernetes nodes to access public Internet services and other internal services. Follow the steps below to configure a global proxy for all outgoing HTTP/HTTPS traffic from your Kubernetes clusters:

- a. Under **HTTP/HTTPS proxy**, select **Enabled**.

HTTP/HTTPS Proxy (for vSphere only)*

☐ Disabled
 ☒ Enabled

HTTP Proxy URL

HTTP Proxy Credentials

HTTPS Proxy URL

HTTPS Proxy Credentials

No Proxy

1. Under **HTTP Proxy URL**, enter the URL of your HTTP/HTTPS proxy endpoint. For example, `http://myproxy.com:1234`. 1. (Optional) If your proxy uses basic authentication, enter the username and password under **HTTP Proxy Credentials**. 1. Under **No Proxy**, enter the service network CIDR where your PKS cluster is deployed. List any additional IP addresses that should bypass the proxy.



Note: By default, the `.internal`, `10.100.0.0/8`, and `10.200.0.0/8` IP address ranges are not proxied. This allows internal PKS communication.

- (Optional) If you use GCP and do not use a NAT instance, select **Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)**. Enabling this functionality assigns external IP addresses to VMs in clusters. This setting is ignored in vSphere deployments.

- Click **Save**.

UAA

To configure the UAA server, do the following:

1. Click **UAA**.
2. Under **PKS CLI Access Token Lifetime**, enter a time in seconds for the PKS CLI access token lifetime.

PKS API Access Token Lifetime (in seconds) *

PKS API Refresh Token Lifetime (in seconds) *

Configure your UAA user account store with either internal or external authentication mechanisms *

☒ Internal UAA

☐ LDAP Server

3. Under **PKS CLI Refresh Token Lifetime**, enter a time in seconds for the PKS CLI refresh token lifetime.
4. Select one of the following options:
 - To use an internal user account store for UAA, select **Internal UAA**. Click **Save** and continue to [\(Optional\) Monitoring](#).
 - To use an external user account store for UAA, select **LDAP Server** and continue to [Configure LDAP as an Identity Provider](#).

Configure LDAP as an Identity Provider

To integrate UAA with one or more LDAP servers, configure PKS with your LDAP endpoint information as follows:

1. Under **UAA**, select **LDAP Server**.

Configure your UAA user account store with either internal or external authentication mechanisms *

☐ Internal UAA

☒ LDAP Server

Server URL *

LDAP Credentials *

User Search Base *


User Search Filter *

Group Search Base

Group Search Filter *

- For **Server URL**, enter the URLs that point to your LDAP server. If you have multiple LDAP servers, separate their URLs with spaces. Each URL must include one of the following protocols:
 - `ldap://`: Use this protocol if your LDAP server uses an unencrypted connection.
 - `ldaps://`: Use this protocol if your LDAP server uses SSL for an encrypted connection. To support an encrypted connection, the LDAP server must hold a trusted certificate or you must import a trusted certificate to the JVM truststore.

- For **LDAP Credentials**, enter the LDAP Distinguished Name (DN) and password for binding to the LDAP server. For example, `cn=administrator,ou=Users,dc=example,dc=com`. If the bind user belongs to a different search base, you must use the full DN.

 **Note:** We recommend that you provide LDAP credentials that grant read-only permissions on the LDAP search base and the LDAP group search base.


- For **User Search Base**, enter the location in the LDAP directory tree where LDAP user search begins. The LDAP search base typically matches your domain name.

For example, a domain named `cloud.example.com` may use `ou=Users,dc=example,dc=com` as its LDAP user search base.

- For **User Search Filter**, enter a string to use for LDAP user search criteria. The search criteria allows LDAP to perform more effective and efficient searches. For example, the standard LDAP search filter `cn=Smith` returns all objects with a common name equal to `Smith`.

In the LDAP search filter string that you use to configure PKS, use `{0}` instead of the username. For example, use `cn={0}` to return all LDAP objects with the same common name as the username.

In addition to `cn`, other common attributes are `mail`, `uid` and, in the case of Active Directory, `SAMAccountName`.

 **Note:** For information about testing and troubleshooting your LDAP search filters, see [Configuring LDAP Integration with Pivotal Cloud Foundry](#).

- For **Group Search Base**, enter the location in the LDAP directory tree where the LDAP group search begins.

For example, a domain named `cloud.example.com` may use `ou=Groups,dc=example,dc=com` as its LDAP group search base.

Follow the instructions in the [Grant Cluster Access to an External LDAP Group](#) section of *Creating and Managing Users with the UAA CLI (UAAC)* to map the groups under this search base to roles in PKS.

- For **Group Search Filter**, enter a string that defines LDAP group search criteria. The standard value is `member={0}`.

8. For **Server SSL Cert**, paste in the root certificate from your CA certificate or your self-signed certificate.

Server SSL Cert

Server SSL Cert AltName

First Name Attribute

Last Name Attribute

Email Attribute *

mail

Email Domain(s)


LDAP Referrals*

Automatically follow any referrals

9. For **Server SSL Cert AltName**, do one of the following:
 - If you are using `ldaps://` with a self-signed certificate, enter a Subject Alternative Name (SAN) for your certificate.
 - If you are not using `ldaps://` with a self-signed certificate, leave this field blank.
10. For **First Name Attribute**, enter the attribute name in your LDAP directory that contains user first names. For example, `cn`.
11. For **Last Name Attribute**, enter the attribute name in your LDAP directory that contains user last names. For example, `sn`.
12. For **Email Attribute**, enter the attribute name in your LDAP directory that contains user email addresses. For example, `mail`.
13. For **Email Domain(s)**, enter a comma-separated list of the email domains for external users who can receive invitations to Apps Manager.
14. For **LDAP Referrals**, choose how UAA handles LDAP server referrals to other user stores. UAA can follow the external referrals, ignore them without returning errors, or generate an error for each external referral and abort the authentication.
15. Click **Save**.

(Optional) Monitoring

You can monitor Kubernetes clusters and pods metrics externally using the integration with [Wavefront by VMware](#).

 **Note:** Before you configure the Wavefront integration, you must have an active Wavefront account and access to a Wavefront instance. You provide your Wavefront access token during configuration. For instructions and additional information, see the [Wavefront documentation](#).

By default, monitoring is disabled. To enable and configure Wavefront monitoring, do the following:

1. Under **Wavefront Integration**, select **Yes**.

Wavefront Integration *

☐ No
☒ Yes


Wavefront URL *

Wavefront Access Token *


Wavefront Alert Recipient

Save

- Under **Wavefront URL**, enter the URL of your Wavefront subscription. For example, `https://try.wavefront.com/api`.
- Under **Wavefront Access Token**, enter the API token for your Wavefront subscription.
- To configure Wavefront to send alerts by email, enter email addresses or Wavefront Target IDs separated by commas under **Wavefront Alert Recipient**. For example: `user@example.com,Wavefront_TargetID`. You also need to enable errands on order to create alerts.

 **Note:** You must enable errands to create alerts. In the **Errands** tab, enable the **Create pre-defined Wavefront alerts errand** and **Delete pre-defined Wavefront alerts errand**.

- Click **Save**. Your settings here apply to any clusters created *after* you have saved these configuration settings and applied changes.

 **Note:** The PKS tile does not validate your Wavefront configuration settings. To verify your setup, look for cluster and pod metrics in Wavefront.

Errands

Errands are scripts that run at designated points during an installation.

To configure when post-deploy and pre-delete errands for PKS are run, make a selection in the dropdown next to the errand. For a typical PKS deployment, we recommend that you leave the default settings.

NSX-T Validation errand	Validates NSX-T configuration and tags resources
Default (Off)	
Upgrade all clusters errand	Upgrades all Kubernetes clusters provisioned by PKS after the PKS Tile upgrade is applied
Default (On)	
Create pre-defined Wavefront alerts errand	Create pre-defined Wavefront alerts
Default (Off)	

Pre-Delete Errands

Delete all clusters errand	Deletes all clusters provisioned by PKS when the PKS tile is deleted
Default (On)	
Delete pre-defined Wavefront alerts errand	Delete pre-defined Wavefront alerts errand
Default (Off)	

For more information about errands and their configuration state, see [Managing Errands in Ops Manager](#).

WARNING: Because PKS uses floating stemcells, updating the PKS tile with a new stemcell triggers the rolling of every VM in each cluster. Also, updating other product tiles in your deployment with a new stemcell causes the PKS tile to roll VMs. This rolling is enabled by the **Upgrade all clusters errand**. We recommend that you keep this errand turned on because automatic rolling of VMs ensures that all deployed cluster VMs are patched. However, automatic rolling can cause downtime in your deployment.

If you upgrade PKS from 1.0.x to 1.1, you must enable the **Upgrade All Cluster** errand. This ensures existing clusters can perform resize or delete actions after the upgrade.

Resource Config

To modify the resource usage of PKS, click **Resource Config** and edit the **Pivotal Container Service** job.

JOB	INSTANCES	PERSISTENT DISK TYPE	VM TYPE	LOAD BALANCERS	INTERNET CONNECTED
Pivotal Container Service	Automatic: 1	Automatic: 10 GB	Automatic: large	tcp:pkcs-api	<input checked="" type="checkbox"/>

Note: If you experience timeouts or slowness when interacting with the PKS API, select a **VM Type** with greater CPU and memory resources for the **Pivotal Container Service** job.

If you are using GCP, enter a name for your PKS API load balancer that begins with `tcp:` in the **Load Balancers** column. For example, `tcp:pkcs-api`, where `pkcs-api` is the name that you configured in Step 6 of [Configuring a GCP Load Balancer](#). For more information, see [Configuring a GCP Load Balancer for the PKS API](#).

Step 3: Apply Changes

After configuring the tile, return to the Ops Manager Installation Dashboard and click **Apply Changes** to deploy the tile.


Step 4: Retrieve PKS API Endpoint

You must share the PKS API endpoint to allow your organization to use the API to create, update, and delete clusters. See [Create a Cluster](#) for more information.

To retrieve the PKS API endpoint, do the following:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the Pivotal Container Service tile.
3. Click the **Status** tab and locate the **Pivotal Container Service** job. The IP address of the Pivotal Container Service job is the PKS API endpoint.

Step 5: Configure External Load Balancer

 **Note:** This section applies only to PKS deployments on GCP or on vSphere without NSX-T.

If you are using GCP, continue to [Next Steps](#).

If you are using vSphere, configure an external load balancer to access the PKS API from outside the network. You can use any external load balancer of your choice.

Your external load balancer forwards traffic to the PKS API endpoint on ports 8443 and 9021. Configure the external load balancer to resolve to the domain name you set in the [PKS API](#) section of the tile configuration.

The load balancer should be configured with:

- The IP address from [Step 4: Retrieve PKS API Endpoint](#)
- Ports 8443 and 9021
- The HTTPS or TCP protocol

Next Steps

Configure authentication for PKS using either User Account and Authentication (UAA) or enterprise single sign-on (SSO).

- To create and manage users using UAA, see [Manage Users in UAA](#).
- To set up SSO configuration with Security Assertion Markup Language (SAML), see [Configure Authentication and Enterprise SSO](#).

After configuring authentication, follow the procedures in [Configure PKS API Access](#) to enable operators to create and manage clusters.

Installing and Configuring PKS with NSX-T Integration

Page last updated:

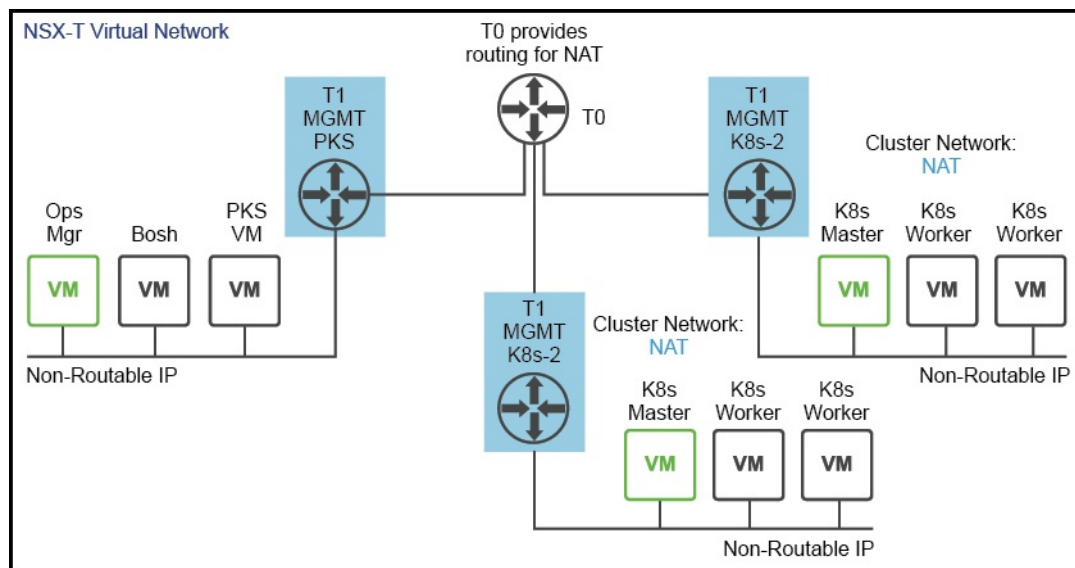
This topic describes how to install and configure Pivotal Container Service (PKS) on vSphere with NSX-T integration.

Deployment Topologies

There are three supported topologies in which to deploy NSX-T with PKS. Except where noted, the instructions in this topic describe how to set up all three options.

NAT Topology

The following figure shows a Network Address Translation (NAT) deployment:



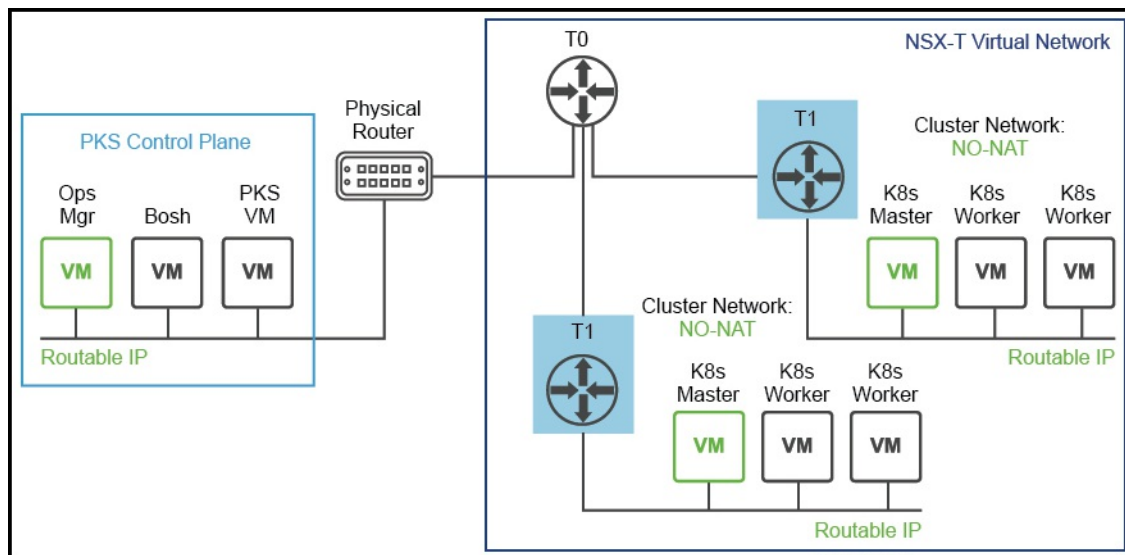
[View a larger version of this image.](#)

This topology has the following characteristics:

- PKS control plane (Ops Manager, BOSH Director, and PKS VM) components are all located on a logical switch that has undergone Network Address Translation on a T0.
- Kubernetes cluster master and worker nodes are located on a logical switch that has undergone Network Address Translation on a T0. This requires DNAT rules to allow access to Kubernetes APIs.

NO-NAT with Virtual Switch (VSS/VDS) Topology

The following figure shows a NO-NAT with Virtual Switch (VSS/VDS) deployment:



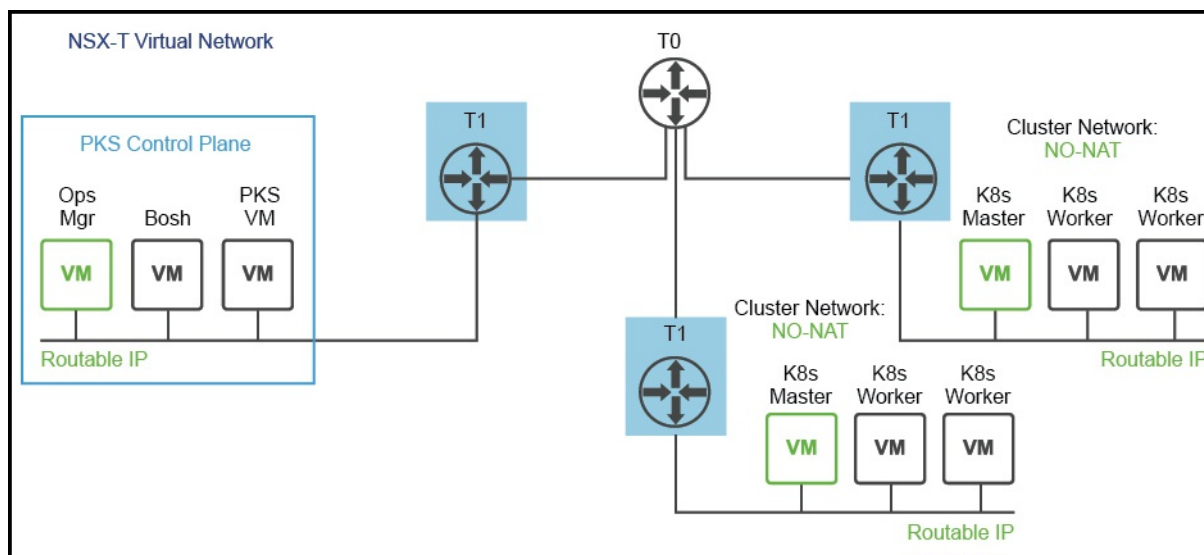
[View a larger version of this image.](#)

This topology has the following characteristics:

- PKS control plane (Ops Manager, BOSH Director, and PKS VM) components are using corporate routable IP addresses.
- Kubernetes cluster master and worker nodes are using corporate routable IP addresses.
- The PKS control plane is deployed outside of the NSX-T network and the Kubernetes clusters are deployed and managed within the NSX-T network. Since BOSH needs routable access to the Kubernetes Nodes to monitor and manage them, the Kubernetes Nodes need routable access.

NO-NAT with Logical Switch (NSX-T) Topology

The following figure shows a NO-NAT with Logical Switch (NSX-T) deployment:



[View a larger version of this image.](#)

This topology has the following characteristics:

- PKS control plane (Ops Manager, BOSH Director, and PKS VM) components are using corporate routable IP addresses.
- Kubernetes cluster master and worker nodes are using corporate routable IP addresses.
- The PKS control plane is deployed inside of the NSX-T network. Both the PKS control plane components (VMs) and the Kubernetes Nodes use corporate routable IP addresses.

Before You Install

Follow these steps before performing the procedures in this topic:

- Review the requirements described in [vSphere Prerequisites and Resource Requirements](#).
- Remember that the instructions in this section are cumulative. For each step, be sure to follow instructions precisely. Complete any confirmation tasks described in the VMware [NSX-T documentation](#) to verify your setup before proceeding to the next step.
- Comply with any requirements or best practices described in the VMware [NSX-T documentation](#).
- For firewall interoperability, see [Firewall Ports and Protocols Requirements for vSphere with NSX-T](#).

Note: When using NSX-T 2.1, creating namespaces with names longer than 40 characters may result in a truncated or hashed name in the NSX-T Manager UI.

Step 1: Plan for Network Subnets and IP Blocks

Before you install PKS on NSX-T, you should plan for the CIDRs and IP blocks that you are using in your deployment.

Plan Network CIDRs

Plan for the following network CIDRs in the IPv4 address space according to the instructions in the VMware [NSX-T documentation](#).

- **VTEP CIDR(s):** One or more of these networks host your GENEVE Tunnel Endpoints on your NSX Transport Nodes. Size the networks to support all of your expected Host and Edge Transport Nodes. For example, a CIDR of `192.168.1.0/24` provides 254 usable IPs. This is used when creating the `ip-pool-vteps` in Step 3.
- **PKS MANAGEMENT CIDR:** This small network is used to access PKS management components such as Ops Manager and the PKS Service VM. For example, a CIDR of `10.172.1.0/28` provides 14 usable IPs. For the [NO-NAT deployment topologies](#), this is a corporate routable subnet /28. For the [NAT deployment topology](#), this is a non-routable subnet /28, and DNAT needs to be configured in NSX-T to access the PKS management components.
- **PKS LB CIDR:** This network provides your load balancing address space for each Kubernetes cluster created by PKS. The network also provides IP addresses for Kubernetes API access and Kubernetes exposed services. For example, `10.172.2.0/24` provides 256 usable IPs. This network is used when creating the `ip-pool-vips` described in [Create NSX Network Objects](#), or when the services are deployed. You enter this network in the **Floating IP Pool ID** field in the **Networking** pane of the PKS tile.

Refer to the instructions in the VMware [NSX-T documentation](#) to ensure that your network topology enables the following communications:

- vCenter, NSX-T components, and ESXi hosts must be able to communicate with each other.
- The Ops Manager Director VM must be able to communicate with vCenter and the NSX Manager.
- The Ops Manager Director VM must be able to communicate with all nodes in all Kubernetes clusters.
- Each Kubernetes cluster deployed by PKS deploys a NCP pod that must be able to communicate with the NSX Manager.

Plan IP Blocks

In addition, you need to plan IP blocks for pods and nodes that are created when PKS creates the Kubernetes cluster. IP Block sizes must be a multiple of 256 (/24). You must make sure that an IP block already has any subnets allocated, and that the subnet size is 256 (/24). You configure the **Pods IP Block ID** and **Nodes IP Block ID** in the **Networking** pane of the PKS tile.


Harbor uses the following IP blocks for its internal bridges:

- 172.17.0.1/16
- 172.18.0.1/16
- 172.19.0.1/16
- 172.20.0.1/16
- 172.21.0.1/16
- 172.22.0.1/16

Note: Do not use any of the IP blocks listed above for pods or nodes. If you create Kubernetes clusters with any of the blocks listed above, the Kubernetes worker nodes cannot reach Harbor for the image pull.

Step 2: Deploy NSX-T

Deploy NSX-T according to the instructions in the VMware [NSX-T documentation](#).

 **Note:** In general, accept default settings unless instructed otherwise.

1. Deploy the NSX Manager. For more information, see [NSX Manager Installation](#) in the VMware NSX-T documentation.
2. Deploy NSX Controllers. For more information, see [NSX Controller Installation and Clustering](#) in the VMware NSX-T documentation.
3. Join the NSX Controllers to the NSX Manager. For more information, see [Join NSX Controllers with the NSX Manager](#) in the VMware NSX-T documentation.
4. Initialize the Control Cluster. For more information, see [Initialize the Control Cluster to Create a Control Cluster Master](#) in the VMware NSX-T documentation.
5. Add your ESXi hosts to the NSX-T Fabric. For more information, see [Add a Hypervisor Host to the NSX-T Fabric](#) in the VMware NSX-T documentation. Each host must have at least one **free nic/vmnic** not already used by other vSwitches on the ESXi host for use with NSX Host Transport Nodes.
6. Deploy NSX Edge VMs. We recommend at least two VMs. For more information, see [NSX Edge Installation](#) in the VMware NSX-T documentation. Each deployed NSX Edge VM requires free resources in your vSphere environment to provide 8 vCPU, 16 GB of RAM, and 120 GB of storage. When deploying, you must connect the vNICs of the NSX Edge VMs to an appropriate PortGroup for your environment by completing the following steps:
 - a. Connect the first Edge interface to your environment's PortGroup/VLAN where your Edge Management IP can route and communicate with the NSX Manager.
 - b. Connect the second Edge interface to your environment's PortGroup/VLAN where your GENEVE VTEPs can route and communicate with each other. Your **VTEP CIDR** should be routable to this PortGroup.
 - c. Connect the third Edge interface to your environment's PortGroup/VLAN where your T0 uplink interface is located.
 - d. Join the NSX Edge VMs to the NSX-T Fabric. For more information, see [Join NSX Edge with the Management Plane](#) in the VMware NSX-T documentation.


Step 3: Create the NSX-T Objects Required for PKS

Create the NSX-T objects (network objects, logical switches, NSX Edge, and logical routers) needed for PKS deployment according to the instructions in the VMware [NSX-T documentation](#).

3.1: Create NSX Network Objects

1. Create two NSX IP pools. For more information, see [Create an IP Pool for Tunnel Endpoint IP Addresses](#) in the VMware NSX-T documentation. Configuration details for the NSX IP pools:
 - o One NSX IP pool for GENEVE Tunnel Endpoints `ip-pool-vteps`, within the usable range of the **VTEP CIDR** created in Step 1, to be used with NSX Transport Nodes that you create later in this section
 - o One NSX IP pool for NSX Load Balancing VIPs `ip-pool-vips`, within the usable range of the **PKS LB CIDR** created in Step 1, to be used with the T0 Logical Router that you create later in this section
2. Create two NSX Transport Zones (TZs). For more information, see [Create Transport Zones](#) in the VMware NSX-T documentation. Configuration details for the NSX TZs:
 - o One NSX TZ for PKS control plane Services and Kubernetes Cluster deployment overlay networks named `tz-overlay` and the associated N-VDS `hs-overlay`. Select **Standard**.
 - o One NSX TZ for NSX Edge uplinks (ingress/egress) for PKS Kubernetes clusters named `tz-vlan` and the associated N-VDS `hs-vlan`. Select **Standard**.
3. If the default uplink profile is not applicable in your deployment, create your own NSX uplink host profile. For more information, see [Create an Uplink Profile](#) in the VMware NSX-T documentation.
4. Create NSX Host Transport Nodes. For more information, see [Create a Host Transport Node](#) in the VMware NSX-T documentation. Configuration details:
 - o For each host in the NSX-T Fabric, create a node named `tnode-host-NUMBER`. For example, if you have three hosts in the NSX-T Fabric, create


- three nodes named `tnode-host-1`, `tnode-host-2`, and `tnode-host-3`.
- Add the `tz-overlay` NSX Transport Zone to each NSX Host Transport Node.

 **Note:** The Transport Nodes must be placed on free host NICs not already used by other vSwitches on the ESXi host. Use the `ip-pool-vteps` IP pool that allows them to route and communicate with each other, as well as other Edge Transport Nodes, to build GENEVE tunnels.

5. Create NSX IP Blocks. We recommend that you use separate NSX IP Blocks for the node networks and the pod networks. The subnets (both nodes and pods) should have a size of 256 (/24). For more information, see [Manage IP Blocks](#) in the VMware NSX-T documentation. Configuration details:
 - One NSX IP Block named `node-network-ip-block`. PKS uses this block to assign address space to Kubernetes master and worker nodes when new clusters are deployed or a cluster increases its scale.
 - One NSX IP Block named `pod-network-ip-block`. The NSX-T Container Plug-in (NCP) uses this block to assign address space to Kubernetes pods through the Container Networking Interface (CNI).

3.2: Create Logical Switches

1. Create the following NSX Logical Switches. For more information, see [Create a Logical Switch](#) in the VMware NSX-T documentation. Configuration details for the Logical Switches:
 - One for T0 ingress/egress uplink port `ls-pks-uplink`
 - One for the PKS Management Network `ls-pks-mgmt`

 **Note:** This network is required for the [NAT deployment topology](#) and [NO-NAT with Logical Switch deployment topology](#). If you are deploying the [NO-NAT with Virtual Switch deployment topology](#), you can skip this step.

- One for the PKS Service Network `ls-pks-service`
2. Attach your first NSX Logical Switch to the `tz-vlan` NSX Transport Zone.
 3. Attach your second and third NSX Logical Switches to the `tz-overlay` NSX Transport Zone.

3.3: Create NSX Edge Objects

1. Create NSX Edge Transport Nodes. For more information, see [Create an NSX Edge Transport Node](#) in the VMware NSX-T documentation.
2. Add both `tz-vlan` and `tz-overlay` NSX Transport Zones to the NSX Edge Transport Nodes. Controller Connectivity and Manager Connectivity should be **UP**.
3. Refer to the MAC addresses of the Edge VM interfaces you deployed to deploy your virtual NSX Edges:
 - a. Connect the `hs-overlay` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the second NIC from your deployed Edge VM.
 - b. Connect the `hs-vlan` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the third NIC from your deployed Edge VM.
4. Create an NSX Edge cluster named `edge-cluster-pks`. For more information, see [Create an NSX Edge Cluster](#) in the VMware NSX-T documentation.
5. Add the NSX Edge Transport Nodes to the cluster.

3.4: Create Logical Routers

Create T0 Logical Router for PKS

T0 routers are edge routers that help route data between your non-NSX-T (such as a Physical Network) and the NSX-T network. PKS currently supports only a single T0 router per instance.

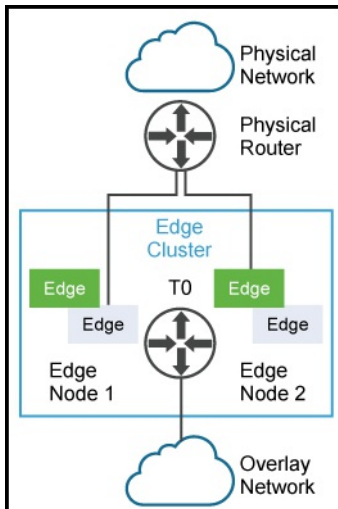
1. Create a Tier-0 (T0) logical router named `t0-pks`. For more information, see [Create a Tier-0 Logical Router](#) in the VMware NSX-T documentation. Configuration details:
 - Select `edge-cluster-pks` for the cluster.
 - Set **High Availability Mode** to **Active-Standby**. NAT rules are applied on T0 by NCP. If not set **Active-Standby**, the router does not support

NAT rule configuration.

2. Attach the T0 logical router to the `ls-pks-uplink` logical switch you created previously. For more information, see [Connect a Tier-0 Logical Router to a VLAN Logical Switch](#) in the VMware NSX-T documentation. Create a logical router port for `ls-pks-uplink` and assign an IP address and CIDR that your environment uses to route to all PKS assigned IP pools and IP blocks.
3. Configure T0 routing to the rest of your environment using the appropriate routing protocol for your environment or by using static routes. For more information, see [Tier-0 Logical Router](#) in the VMware NSX-T documentation. The CIDR used in `ip-pool-vips` *must* route to the IP you just assigned to your t0 uplink interface.

(Optional) Configure NSX Edge for High Availability (HA)

You can configure NSX Edge for high availability (HA) using Active/Standby mode to support failover, as shown in the following figure.



To configure NSX Edge for HA, complete the following steps:

Note: All IP addresses must belong to the same subnet.

Step 1: On the T0 router, create a second uplink attached to the second Edge transport node:

Setting	First Uplink	Second Uplink
IP Address/Mask	uplink_1_ip	uplink_2_ip
URPF Mode	None (optional)	None (optional)
Transport Node	edge-TN1	edge-TN2
LS	uplink-LS1	uplink-LS1

Step 2: On the T0 router, create the HA VIP:

Setting	HA VIP
VIP address	[ha_vip_ip]
Uplinks ports	uplink-1 and uplink-2

The HA VIP becomes the official IP for the T0 router uplink. External router devices peering with the T0 router *must* use this IP address.

Step 3: On the physical router, configure the next hop to point to the HA VIP address.

Step 4: You can verify your setup by running the following commands:

```
nsx-edge-n> get high-availability channels
nsx-edge-n> get high-availability channels stats
nsx-edge-n> get logical-router
nsx-edge-n> get logical-router ROUTER-UUID high-availability status
```

Create T1 Logical Router for PKS Management VMs

1. Create a Tier-1 (T1) logical router for PKS management VMs named `t1-pks-mgmt`. For more information, see [Create a Tier-1 Logical Router](#) in the VMware NSX-T documentation. Configuration details:
 - Link to the `t0-pks` logical router you created in a previous step.
 - Select `edge-cluster-pks` for the cluster.

Note: Skip this step if you are deploying the NO-NAT with Virtual Switch topology. This Logical Router is required for the [NAT deployment topology](#) and NO-NAT with Logical Switch deployment topology. .

2. Create a logical router port for `ls-pks-mgmt` and assign the following CIDR block: `10.172.1.0/28`. For more information, see [Connect a Tier-0 Logical Router to a VLAN Logical Switch](#) in the VMware NSX-T documentation.
3. Configure route advertisement on the T1 as follows. For more information, see [Configure Route Advertisement on a Tier-1 Logical Router](#) in the VMware NSX-T documentation. Configuration details:
 - Enable **Status**.
 - Enable **Advertise All NSX Connected Routes**.
 - Enable **Advertise All NAT Routes**.
 - Enable **Advertise All LB VIP Routes**.

Configure NAT Rules for PKS Management VMs

Note: This step applies to the [NAT deployment topology](#) only. Skip this step for [NO-NAT deployment topologies](#).

Create the following NAT rules for the Mgmt T0. For more information, see [Tier-0 NAT](#) in the VMware NSX-T documentation. Configuration details:

Type	For
DNAT	External -> Ops Manager
DNAT	External -> Pivotal Container Service
SNAT	Ops Manager & BOSH Director -> DNS
SNAT	Ops Manager & BOSH Director -> NTP
SNAT	Ops Manager & BOSH Director -> vCenter
SNAT	Ops Manager & BOSH Director -> ESXi
SNAT	Ops Manager & BOSH Director -> NSX-T Manager

The Destination NAT (DNAT) rule on the T0 maps an external IP address from the **PKS MANAGEMENT CIDR** to the IP where you deploy Ops Manager on the `ls-pks-mgmt` logical switch. For example, a DNAT rule that maps `10.172.1.2` to `172.31.0.2`, where `172.31.0.2` is the IP address you assign to Ops Manager when connected to `ls-pks-mgmt`. Later, you create another DNAT rule to map an external IP address from the **PKS MANAGEMENT CIDR** to the PKS endpoint.

The Source NAT (SNAT) rule on the T0 allows the PKS Management VMs to communicate with your vCenter and NSX Manager environments. For example, an SNAT rule that maps `172.31.0.0/24` to `10.172.1.1`, where `10.172.1.1` is a routable IP address from your **PKS MANAGEMENT CIDR**. For more information, see [Configure Source NAT on a Tier-1 Router](#) in the VMware NSX-T documentation.

Note: Ops Manager and BOSH must use the NFCP protocol to the actual ESX hosts to which it is uploading stemcells. Specifically, **Ops Manager & BOSH Director -> ESXi**.

Note: Limit the Destination CIDR for the SNAT rules to the subnets that contain your vCenter and NSX Manager IP addresses.

Step 4: Deploy Ops Manager

Complete the procedures in [Deploying Ops Manager to vSphere](#).

Step 5: Configure Ops Manager

Perform the following steps to configure Ops Manager for the NSX logical switches:

1. Complete the procedures in [Configuring Ops Manager on vSphere](#).

Note: If you have Pivotal Application Service (PAS) installed, we recommend installing PKS on a separate instance of Ops Manager v2.1.

- On the **vCenter Config** pane, select **NSX Networking - NSX-T**. This configuration is used for PAS and PKS. For more information, see the [Enable NSX-T Mode in the BOSH Director](#) section of *Deploying PAS with NSX-T Networking* in the PCF documentation.

Note: If you are using the [NAT deployment topology](#), you must have already deployed Ops Manager to the `ls-pks-mgmt` NSX logical switch by following the instructions above in [Create T1 Logical Router for PKS Management VMs](#). You will use the DNAT IP address to access Ops Manager.

- On the **Create Networks** pane, create the following network:

Field	Configuration
Name	<code>pks-infrastructure</code>
vSphere Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-infrastructure</code>
Description	A network for deploying the PKS control plane VMs that maps to the NSX logical switch named <code>ls-pks-mgmt</code> created for the PKS Management Network in Step 3: Create the NSX-T Objects Required for PKS .

2. Return to the Ops Manager Installation Dashboard and click **Apply Changes**.

Step 6: Generate and Register Certificates

Before you install PKS on NSX-T, you must create two certificates that you will provide in the **Networking** pane in the PKS tile. For more information, see [Networking](#).

6.1: Generate the NSX Manager Super User Principal Identity Certificate

This certificate represents a principal identity with superuser permissions that the PKS VM will use to communicate with NSX-T to manage (create, delete, and modify) node networking resources. During PKS installation on NSX-T, you will need to provide this in the NSX Manager Super User Principal Identity Certificate field on the **Networking** pane in the PKS tile. You can complete the following steps from the Ops Manager VM or from any other Linux VM. This procedure does not work on Mac OS.

Before You Begin

Export the following environment variables to complete the steps below:

```
NSX_MANAGER=<NSX_MANAGER_IP>
NSX_USER=<NSX_MANAGER_USERNAME>
NSX_PASSWORD=<NSX_MANAGER_PASSWORD>
PI_NAME="pks-nsx-t-superuser"
NSX_SUPERUSER_CERT_FILE="pks-nsx-t-superuser.crt"
NSX_SUPERUSER_KEY_FILE="pks-nsx-t-superuser.key"
NODE_ID=$(cat /proc/sys/kernel/random/uuid)
```

Step 6.1.1: Create the Super User Principal Identity Certificate

Create the Super User Principal Identity Certificate using a script or by clicking **Generate RSA Certificate** on the **Networking** tab in the PKS tile. For more information, see [Networking](#).

Create Certificate Using a Script

To create the certificate using a script, run the following command:

```
$ openssl req \
-newkey rsa:2048 \
-x509 \
-nodes \
-keyout "$NSX_SUPERUSER_KEY_FILE" \
-new \
-out "$NSX_SUPERUSER_CERT_FILE" \
-subj /CN=pks-nsx-t-superuser \
-extensions client_server_ssl \
-config <(
cat /etc/ssl/openssl.cnf\
<(printf '[client_server_ssl]\nextendedKeyUsage = clientAuth\n')
)\
-sha256 \
-days 730
```

Create Certificate from the Networking Tab

To create the certificate from the **Networking** tab in the PKS tile, follow the steps below.

1. Navigate to the **Networking** tab in the PKS tile. For more information, see [Networking](#).
2. Click **Generate RSA Certificate** and provide a wildcard domain, for example, `*.nsx.pks.vmware.local`.
3. In the Ops Manager / Linux VM where the subsequent scripts will run, create a file named `pks-nsx-t-superuser.crt` and copy the generated certificate into it.
4. In the Ops Manager / Linux VM where the subsequent scripts will run, create a file named `pks-nsx-t-superuser.key` and copy the private key into it.

Step 6.1.2: Register the Certificate

To register the certificate with NSX Manager, run the following commands:

```
cert_request=$(cat <<END
{
  "display_name": "$PI_NAME",
  "pem_encoded": "$(awk '{printf "%s\n", $0}' $NSX_SUPERUSER_CERT_FILE)"
}
END
)
```

```
curl -k -X POST \
"https://$NSX_MANAGER/api/v1/trust-management/certificates?action=import" \
-u "$NSX_USER:$NSX_PASSWORD" \
-H 'content-type: application/json' \
-d "$cert_request"
```

The response includes the `CERTIFICATE_ID` value.

Step 6.1.3: Register the Principal Identity

To register the principal identity with NSX Manager, run the following commands:

```
pi_request=$(cat <<END
{
  "display_name": "$PI_NAME",
  "name": "$PI_NAME",
  "permission_group": "superusers",
  "certificate_id": "$CERTIFICATE_ID",
  "node_id": "$NODE_ID"
}
END
)
```

```
curl -k -X POST \
  "https://${NSX_MANAGER}/api/v1/trust-management/principal-identities" \
  -u "$NSX_USER:$NSX_PASSWORD" \
  -H 'content-type: application/json' \
  -d "$pi_request"
```

Step 6.1.4: Verify the Certificate and Key

To verify that the certificate and key can be used with NSX-T, complete the following steps:


```
curl -k -X GET \
  "https://${NSX_MANAGER}/api/v1/trust-management/principal-identities" \
  --cert $(pwd)"/$NSX_SUPERUSER_CERT_FILE" \
  --key $(pwd)"/$NSX_SUPERUSER_KEY_FILE"
```

Later, when you install PKS on NSX-T, you will copy and paste the contents of the `pks-nsx-t-superuser.crt` and `pks-nsx-t-superuser.key` into the NSX Manager Super User Principal Identity Certificate field on the **Networking** pane in the PKS tile.

6.2: Generate the NSX Manager CA Certificate

This certificate is used to authenticate with the NSX Manager. You create an IP-based, self-signed certificate and register it with NSX Manager. During PKS installation on NSX-T, you will need to provide this certificate in the **NSX Manager CA Cert** field on the Networking Tab in the PKS tile.

Step 6.2.1: Generate a Self-signed Certificate

 **Note:** If you already have a CA-signed certificate, skip this section and go to 6.2.2.

1. Create a file with the certificate request parameters, replacing `NSX-MANAGER-IP-ADDRESS` with the IP address of your NSX Manager:

```
$ cat nsx-cert.cnf
[ req ]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
prompt = no
[ req_distinguished_name ]
countryName = US
stateOrProvinceName = California
localityName = CA
organizationName = NSX
commonName = NSX-MANAGER-IP-ADDRESS
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1 = NSX-MANAGER-IP-ADDRESS
```

2. Generate the certificate using openssl. Run the following command, replacing `NSX-MANAGER-IP-ADDRESS` with the IP address of your NSX Manager:

```
openssl req -newkey rsa:2048 -x509 -nodes \
-keyout nsx.key -new -out nsx.crt -subj /CN=NSX-MANAGER-IP-ADDRESS \
-reqexts SAN -extensions SAN -config <(cat ./nsx-cert.cnf \
<(printf '[SAN]subjectAltName=IP:NSX-MANAGER-IP-ADDRESS')) -sha256 -days 365
```

3. Verify that the certificate looks correct and that the NSX manager IP is in the Subject Alternative Name (SAN) by running the following command:

```
$ openssl x509 -in nsx.crt -text -noout
```

Step 6.2.2: Register the Certificate with NSX Manager

1. Log into the NSX Manager UI.
2. Import the certificate by copying `nsx.crt` and `nsx.key`. For instructions, see [Import a CA Certificate](#) in the NSX-T documentation.

3. Get the ID of the certificate. Run the following command, replacing `CERTIFICATE-NAME` with the certificate name:

```
curl --insecure -u admin:'admin_pw' -X \
GET https://NSX-Manager-IP-Address/api/v1/trust-management/certificates \
| jq '-r .results[] | select(.display_name==CERTIFICATE-NAME) | .id'
```

4. Register the certificate with NSX Manager, replacing `CERTIFICATE-ID` with the certificate ID:

```
curl --insecure -u admin:'admin_pw' -X \
POST 'https://NSX-Manager-IP-Address/api/v1/node/services/http?action=apply_certificate&certificate_id=CERTIFICATE-ID'
```

Later, when you install PKS on NSX-T, you will copy and paste the contents of the `nsx.crt` certificate into the **NSX Manager CA Cert** field on the **Networking** pane in the PKS tile.

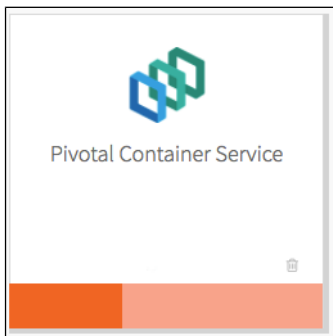
Step 7: Install and Configure PKS

Perform the following steps to install and configure PKS:

1. Install the PKS tile. For more information, see [Installing and Configuring PKS](#).
2. Click the orange **Pivotal Container Service** tile to start the configuration process.



Note: Configuration of NSX-T or Flannel **cannot** be changed after initial installation and configuration of PKS.



Assign AZs and Networks

Perform the following steps:

1. Click **Assign AZs and Networks**.
2. Select the availability zone (AZ) where you want to deploy the PKS API VM as a singleton job.



Note: You must select an additional AZ for balancing other jobs before clicking **Save**, but this selection has no effect in the current version of PKS.

Place singleton jobs in

☒ us-west-2a
 ☐ us-west-2b
 ☐ us-west-2c

Balance other jobs in

☐ us-west-2a
 ☒ us-west-2b
 ☐ us-west-2c

Network

pkc-infrastructure

Service Network

pkc-services

Save

- Under **Network**, select the PKS Management Network linked to the `ls-pkc-mgmt` NSX logical switch you created in [Step 5: Configure Ops Manager](#). This will provide network placement for the PKS API VM.
- Under **Service Network**, your selection depends on whether you are upgrading from a previous PKS version or installing an original PKS deployment.
 - If you are upgrading from a previous PKS version, select the PKS Service Network linked to the `ls-pkc-service` NSX logical switch you created in [Step 5: Configure Ops Manager](#). This will provide network placement for the on-demand Kubernetes cluster service instances created by the PKS broker.
 - If you are deploying PKS on vSphere with NSX-T for the first time, the Service Network field does not apply to PKS deployments. However, the tile requires you to make a selection. Therefore, select any network that has been configured in the Ops Manager Network configuration.
- Click **Save**.

PKS API

Perform the procedure in the [PKS API](#) section of *Installing and Configuring PKS*.

Plans

Perform the procedures in the [Plans](#) section of *Installing and Configuring PKS*.

Kubernetes Cloud Provider

Perform the procedures in the [Kubernetes Cloud Provider](#) section of *Installing and Configuring PKS*.

(Optional) Logging

Perform the procedures in the [Logging](#) section of *Installing and Configuring PKS*.

Networking

Perform the following steps:

- Click **Networking**.
- Under **Container Networking Interface**, select **NSX-T**.

Container Networking Interface*

☐ Flannel
 ☒ NSX-T

NSX Manager hostname *

NSX Manager credentials *

Username

Password

NSX Manager CA Cert

☐ Disable SSL certificate verification

☒ NAT mode

- For **NSX Manager hostname**, enter the hostname or IP address of your NSX Manager.
- For **NSX Manager Super User Principal Identity Certificate**, copy and paste the contents and private key of the Principal Identity certificate you created in [Step 6.1: Generate the NSX Manager Super User Principal Identity Certificate](#). You can create the certificate in this tab by clicking **Generate RSA Certificate**, providing a wildcard domain, for example, `*.nsx.pks.vmware.local`, and copying the generated certificate and key to the `pks-nsx-t-superuser.crt` and `pks-nsx-t-superuser.key` files. For more information, including instructions for completing the additional, required registration and verification steps, see [Step 6.1: Generate the NSX Manager Super User Principal Identity Certificate](#).
- (Optional) For **NSX Manager CA Cert**, copy and paste the contents of the NSX Manager CA certificate you created in [Step 6: Generate and Register Certificates](#). This will be used to connect to the NSX Manager.
- The **Disable SSL certificate verification** checkbox is **not** selected by default. In order to disable TLS verification, select the checkbox. You may want to disable TLS verification if you did not enter a CA certificate, or if your CA certificate is self-signed.
- If you are using a NAT deployment topology, leave the **NAT mode** checkbox selected. If you are using a NO-NAT topology, clear this checkbox. For more information, see the [Deployment Topologies](#) section above.
- Enter the following IP Block settings:

Pods IP Block ID *

Nodes IP Block ID *

T0 Router ID *

Floating IP Pool ID *

Nodes DNS *

vSphere Cluster Names *

HTTP/HTTPS Proxy (for vSphere only) *

☒ Disabled
 ☐ Enabled

Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)

☐ Enable outbound internet access

- Pods IP Block ID:** Enter the UUID of the IP block to be used for Kubernetes pods. PKS allocates IP addresses for the pods when they are created in Kubernetes. Each time a namespace is created in Kubernetes, a subnet from this IP block is allocated. The current subnet size that is created is /24, which means a maximum of 256 pods can be created per namespace.
- Nodes IP Block ID:** Enter the UUID of the IP block to be used for Kubernetes nodes. PKS allocates IP addresses for the nodes when they are created in Kubernetes. The node networks are created on a separate IP address space from the pod networks. The current subnet size that is created is /24, which means a maximum of 256 nodes can be created per cluster. For more information, including sizes and the IP blocks to avoid using, see [Plan IP Blocks](#).

- For **T0 Router ID**, enter the `t0-pks` T0 router UUID. Locate this value in the NSX-T UI router overview.
- For **Floating IP Pool ID**, enter the `ip-pool-vips` ID that you created for load balancer VIPs. For more information, see [Plan IP Blocks](#). PKS uses the floating IP pool to allocate IP addresses to the load balancers created for each of the clusters. The load balancer routes the API requests to the master nodes and the data plane.
- For **Nodes DNS**, enter one or more Domain Name Servers used by the Kubernetes nodes.
- For **vSphere Cluster Names**, enter the names of the vSphere clusters you used when creating the PKS API VM in [Assign AZs and Networks](#).
- (Optional) Configure a global proxy for all outgoing HTTP and HTTPS traffic from your Kubernetes clusters.

Production environments can deny direct access to public Internet services and between internal services by placing an HTTP or HTTPS proxy in the network path between Kubernetes nodes and those services.

If your environment includes HTTP or HTTPS proxies, configuring PKS to use these proxies allows PKS-deployed Kubernetes nodes to access public Internet services and other internal services. Follow the steps below to configure a global proxy for all outgoing HTTP/HTTPS traffic from your Kubernetes clusters:

- Under **HTTP/HTTPS proxy**, select **Enabled**.

HTTP/HTTPS Proxy (for vSphere only)*

☐ Disabled
☒ Enabled

HTTP Proxy URL

HTTP Proxy Credentials

Username

Password

HTTPS Proxy URL


HTTPS Proxy Credentials

Username

Password

No Proxy

- b. Under **HTTP Proxy URL**, enter the URL of your HTTP/HTTPS proxy endpoint. For example, `http://myproxy.com:1234`.
- c. (Optional) If your proxy uses basic authentication, enter the username and password in either **HTTP Proxy Credentials** or **HTTPS Proxy Credentials**.
- d. Under **No Proxy**, enter the service network CIDR where your PKS cluster is deployed. List any additional IP addresses that should bypass the proxy.

 **Note:** By default, the `.internal`, `10.100.0.0/8`, and `10.200.0.0/8` IP address ranges are not proxied. This allows internal PKS communication.

14. Click **Save**.

UAA

Perform the procedures in the [UAA](#) section of *Installing and Configuring PKS*.

(Optional) Monitoring

Perform the procedures in the [Monitoring](#) section of *Installing and Configuring PKS*.

Errands

Errands are scripts that run at designated points during an installation.

WARNING: You must enable the NSX-T Validation errand in order to verify and tag required NSX-T objects.

Perform the following steps:

1. Click **Errands**.

NSX-T Validation errand	Validates NSX-T configuration and tags resources
Default (Off)	
Upgrade all clusters errand	Upgrades all Kubernetes clusters provisioned by PKS after the PKS Tile upgrade is applied
Default (On)	
Create pre-defined Wavefront alerts errand	Create pre-defined Wavefront alerts
Default (Off)	
Pre-Delete Errands	
Delete all clusters errand	Deletes all clusters provisioned by PKS when the PKS tile is deleted
Default (On)	
Delete pre-defined Wavefront alerts errand	Delete pre-defined Wavefront alerts errand
Default (Off)	

2. For **Post Deploy Errands**, select **ON** for the **NSX-T Validation errand**. This errand validates your NSX-T configuration and tags the proper resources.
3. Click **Save**.

(Optional) Resource Config and Stemcell

To modify the resource usage or stemcell configuration of PKS, see the [Resource Config](#) and [Stemcell](#) sections in *Installing and Configuring PKS*.

Step 8: Apply Changes to Deploy the PKS Tile

After configuring the tile, return to the Ops Manager Installation Dashboard and click **Apply Changes** to deploy the PKS tile.

Step 9: Retrieve the PKS Endpoint

1. When the installation is completed, retrieve the PKS endpoint by performing the following steps:
 - a. From the Ops Manager Installation Dashboard, click the **Pivotal Container Service** tile.
 - b. Click the **Status** tab and record the IP address assigned to the `Pivotal Container Service` job.
2. Create a DNAT rule on the `ts-1-pks-mgmt` T1 to map an external IP from the **PKS MANAGEMENT CIDR** to the PKS endpoint. For example, a DNAT rule that maps `10.172.1.4` to `172.31.0.4`, where `172.31.0.4` is PKS endpoint IP address on the `ts-1-pks-mgmt` NSX Logical Switch.



Note: Ensure that you have no overlapping NAT rules. If your NAT rules overlap, you cannot reach Ops Manager from VMs in the vCenter network.

Developers should use the DNAT IP address when logging in with the PKS CLI. For more information, see [Using PKS](#).

Managing PKS

Page last updated:

This section describes how to manage Pivotal Container Service (PKS). See the following topics:

- [Configure PKS API Access](#)
 - [Manage Users in UAA](#)
 - [Configure Authentication and Enterprise SSO](#)
 - [Manage PKS Deployments with BOSH](#)
 - [Backing Up and Restoring the PKS Control Plane](#)
 - [Add Custom Workloads](#)
 - [Service Interruptions](#)
 - [Delete PKS](#)
-

Configure PKS API Access

Page last updated:

This topic describes how to configure access to the Pivotal Container Service (PKS) API. See [PKS API Authentication](#) for more information about how the PKS API and UAA interact with your PKS deployment.

Configure Routing to the PKS API

To configure routing to the PKS API, perform the following steps:

1. Configure an external load balancer to forward traffic to the PKS API endpoint. For more information, see the *Configure External Load Balancer* section of [Installing and Configuring PKS](#).



Note: If your PKS installation is integrated with NSX-T, map the external load balancer to the DNAT IP address assigned in the [Retrieve the PKS Endpoint](#) section of *Installing and Configuring PKS with NSX-T Integration*.

2. Configure a DNS entry that points to the load balancer and uses the domain configured in the *PKS API* section of [Installing and Configuring PKS](#).

Configure Access to the PKS API

1. Locate your Ops Manager root CA certificate.
 - If Ops Manager generated your certificate, refer to the [Retrieve the Root CA Certificate](#) section of *Managing Non-Configurable TLS/SSL Certificates*.
 - If you provided your own certificate, copy and paste the certificate you entered in the **PKS API** pane into a file.
2. Target your UAA server by running the following command:

```
uaac target https://PKS-API:8443 --ca-cert ROOT-CA-FILENAME
```

Replace the following values:

- **PKS-API** : enter the fully qualified domain name (FQDN) you use to access the PKS API. You configured this URL in the *PKS API* section of [Installing and Configuring PKS](#).
- **ROOT-CA-FILENAME** : enter the path for the certificate file you downloaded in a previous step. For example:

```
$ uaac target api.pks.example.com:8443 --ca-cert my-cert.cert
```

Including `https://` in the PKS API URL is optional.

3. Run `uaac token client get admin -s UAA-ADMIN-SECRET` to request a token from the UAA server. Replace `UAA-ADMIN-SECRET` with your UAA admin secret. Refer to **Ops Manager > Pivotal Container Service > Credentials > Pks Uaa Management Admin Client** to retrieve this value.
4. Grant cluster access to new or existing users with UAA. For more information on granting cluster access to users or creating users, see the [Grant Cluster Access to a User](#) section of *Managing Users in UAA*.

Log in to the PKS CLI

For information on logging into the PKS CLI, see the [Log in to PKS CLI](#) section of *Installing the PKS CLI*.

Log in to PKS as a Client

Use the command in this section to log in as an automated client for a script or service.

On the command line, run the following command to log in to the PKS CLI:

```
pks login -a PKS-API --client-name CLIENT-NAME --client-secret CLIENT-SECRET -k
```

Replace the placeholder values in the command as follows:

- `PKS-API` is the domain name for the PKS API that you entered in **Ops Manager > Pivotal Container Service > PKS API > API Hostname (FQDN)**. For example, `api.pks.example.com`.
- `CLIENT-NAME` is your OAuth client ID.
- `CLIENT-SECRET` is your OAuth client secret.

For example:

```
$ pks login -a api.pks.example.com \  
--client-name automated-client \  
--client-secret randomly-generated-secret -k
```

Manage Users in UAA

Page last updated:

This topic describes how to manage users in Pivotal Container Service (PKS) with User Account and Authentication (UAA). Create and manage users in UAA with the UAA Command Line Interface (UAAC).

How to Use UAAC


Use the UAA Command Line Interface (UAAC) to interact with the UAA server. You can either run UAAC commands from the Ops Manager VM or install UAAC on your local workstation.

To run UAAC commands from the Ops Manager VM, see the following SSH procedures for [vSphere](#) or [GCP](#).

To install UAAC locally, see [Component: User Account and Authentication \(UAA\) Server](#).

SSH into the Ops Manager VM on vSphere

To SSH into the Ops Manager VM on vSphere, you need the credentials used to import the PCF .ova or .ovf file into your virtualization system. You set these credentials when you installed Ops Manager.

 **Note:** If you lose your credentials, you must shut down the Ops Manager VM in the vSphere UI and reset the password. See [vCenter Password Requirements and Lockout Behavior](#) in the vSphere documentation for more information.

1. From a command line, run `ssh ubuntu@OPS-MANAGER-FQDN` to SSH into the Ops Manager VM. Replace `OPS-MANAGER-FQDN` with the fully qualified domain name of Ops Manager.
2. When prompted, enter the password that you set during the .ova deployment into vCenter. For example:

```
$ ssh ubuntu@my-opsmanager-fqdn.example.com
Password: *****
```

3. Proceed to the [Retrieve UAA Admin Credentials](#) section to manage users with UAAC.

SSH into the Ops Manager VM on GCP

To SSH into the Ops Manager VM in GCP, follow these instructions:

1. Confirm that you have installed the gcloud CLI. See the [Google Cloud Platform documentation](#) for more information.
2. From the GCP console, click **Compute Engine**.
3. Locate the Ops Manager VM in the **VM Instances** list.
4. Click the **SSH** menu button.
5. Copy the SSH command that appears in the popup window.
6. Paste the command into your terminal window to SSH to the Ops Manager VM. For example:

```
$ gcloud compute ssh om-pcf-1a --zone us-central1-b
```

7. Run `sudo su - ubuntu` to switch to the `ubuntu` user.
8. Proceed to the [Retrieve UAA Admin Credentials](#) section to manage users with UAAC.

Log in as an Admin

To retrieve the PKS UAA management admin client secret, do the following:

1. In a web browser, navigate to the fully qualified domain name (FQDN) of Ops Manager and click the **Pivotal Container Service** tile.
2. Click **Credentials**.
3. To view the secret, click **Link to Credential** next to **Pks Uaa Management Admin Client**. The client username is `admin`.
4. On the command line, run the following command to target your UAA server:

```
uaac target https://PKS-API:8443 --ca-cert ROOT-CA-FILENAME
```

Replace `PKS-API` with the URL to your PKS API server. You configured this URL in the PKS API section of [Installing and Configuring PKS](#). Replace `ROOT-CA-FILENAME` with the certificate file you downloaded in [Configure Access to the PKS API](#). For example:

```
$ uaac target api.pks.example.com:8443 --ca-cert my-cert.cert
```



Note: If you receive an `Unknown key: Max-Age = 86400` warning message, you can safely ignore it because it has no impact.

5. Authenticate with UAA using the secret you retrieved in a previous step. Run the following command, replacing `ADMIN-CLIENT-SECRET` with your PKS UAA management admin client secret:

```
uaac token client get admin -s ADMIN-CLIENT-SECRET
```

Grant Cluster Access

You can assign the following UAA scopes to users, external LDAP groups, and clients:

- `pks.clusters.manage`: accounts with this scope can create and access their own clusters.
- `pks.clusters.admin`: accounts with this scope can create and access all clusters.

Grant Cluster Access to a User

To create a new UAA user with cluster access, perform the following steps:

1. Log in as the UAA admin using the procedure [above](#).
2. To create a new user, run the following command:

```
uaac user add USERNAME --emails USER-EMAIL -p USER-PASSWORD
```

For example:

```
$ uaac user add alana --emails alana@example.com -p password
```

3. Assign a scope to the user to allow them to access Kubernetes clusters. Run `uaac member add UAA-SCOPE USERNAME`, replacing `UAA-SCOPE` with one of the UAA scopes defined [above](#). For example:

```
$ uaac member add pks.clusters.admin alana
```

Grant Control Plane Access to an External LDAP Group

For information about configuring LDAP with PKS, see the [Configure Authentication and Enterprise SSO](#) topic.

The PKS control plane enables users to deploy and manage Kubernetes clusters.

To grant control plane access to an external LDAP group, perform the following steps:

1. Log in as the UAA admin using the procedure [above](#).

2. To grant control plane access to all users in an LDAP group, run the following command:

```
uaac group map --name pks.clusters.manage GROUP-DISTINGUISHED-NAME
```

Replace `GROUP-DISTINGUISHED-NAME` with the LDAP Distinguished Name (DN) for the group. For example:

```
$ uaac group map --name pks.clusters.manage Operators
```

3. (Optional) To grant control plane access to all users in an LDAP group, run the following command:

```
uaac group map --name pks.clusters.admin GROUP-DISTINGUISHED-NAME
```

Replace `GROUP-DISTINGUISHED-NAME` with the LDAP DN for the group. For example:

```
$ uaac group map --name pks.clusters.admin cn=Administrators,ou=Groups,dc=ldap,dc=example,dc=com
```

Where:

- `cn` is the common name.
- `ou` is the organizational unit.
- `dc` is the domain component.

Grant Cluster Access to a Client

To grant cluster access to an automated client for a script or service, perform the following steps:

1. Log in as the UAA admin using the procedure [above](#).
2. Create a client with the desired scopes by running the following command:

```
uaac client add CLIENT-NAME -s CLIENT-SECRET \
--authorized_grant_types client_credentials \
--authorities UAA-SCOPES
```

Replace `CLIENT-NAME` and `CLIENT-SECRET` with the client credentials. Replace `UAA-SCOPES` with one or more of the UAA scopes defined [above](#), separated by a comma. For example:

```
$ uaac client add automated-client \
-s randomly-generated-secret
--authorized_grant_types client_credentials \
--authorities pks.clusters.admin,pks.clusters.manage
```


Configure Authentication and Enterprise SSO

Page last updated:


This topic describes Pivotal Container Service (PKS) authentication and single sign-on configuration with Lightweight Directory Access Protocol (LDAP) and Security Assertion Markup Language (SAML).

Refer to the instructions below to configure your deployment with SAML or LDAP.

Connecting PKS to either the LDAP or SAML external user store allows the User Account and Authentication (UAA) server to delegate authentication to existing enterprise user stores.

If your enterprise user store is exposed as a SAML or LDAP Identity Provider for single sign-on (SSO), you can configure SSO to allow users to access the PKS API without creating a new account or, if using SAML, without re-entering credentials.

For an explanation of the process used by the UAA Server when it attempts to authenticate a user through LDAP, see the [Configuring LDAP Integration with Pivotal Cloud Foundry](#) Knowledge Base article.

 **Note:** When integrating with an external identity provider, such as LDAP, authentication within the UAA becomes chained. An authentication attempt with a user's credentials is first attempted against the UAA user store before the external provider, LDAP. For more information, see [Chained Authentication](#) in the *User Account and Authentication LDAP Integration* GitHub documentation.

Configure PKS to Use a SAML Identity Provider


In SAML terminology, the SAML protocol communicates user data between an identity provider (IdP) and a service provider (SP).

Configure SAML as an Identity Provider for PKS

The following procedure configures a SAML IdP to designate PKS as an SP.

Download the Service Provider Metadata from `https://PKS-API:8443/saml/metadata`. Replace `PKS-API` with the fully qualified domain name of your PKS API instance. For example, `https://pks-api.pks.example.com:8443/saml/metadata`. Refer to **Ops Manager > Pivotal Container Service > PKS API > API Hostname (FQDN)** for your PKS API domain.

Consult the documentation from your Identity Provider for configuration instructions.

 **Note:** Some Identity Providers allow uploads of Service Provider Metadata. Other providers require you to manually enter the Service Provider Metadata into a form.

Manage PKS Deployments with BOSH

Page last updated:

To manage your PKS deployment with BOSH, perform the following steps:

1. Gather credential and IP address information for your BOSH Director and SSH into the Ops Manager VM. See [Advanced Troubleshooting with the BOSH CLI](#) for more information.

2. Create a BOSH alias for your PKS environment. For example:

```
$ bosh alias-env pks -e 10.0.0.3 \  
--ca-cert /var/tempest/workspaces/default/root_ca_certificate
```

3. Log in to the BOSH Director.

```
$ bosh -e pks log-in
```

4. Follow the procedures in the [Use the BOSH CLI for Troubleshooting](#) topic to manage your PKS deployment with BOSH.
-


Add Custom Workloads

Page last updated:

To apply custom Kubernetes workloads to every cluster created on a plan, add a YAML file to the tile config under **Default Cluster Apps**.

Custom workloads define what a cluster includes out of the box.

For example, you can use custom workloads to configure metrics or logging.

The following example YAML file comes from the [Kubernetes documentation](#) .

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template: # create pods using pod definition in this template
    metadata:
      # unlike pod-nginx.yaml, the name is not included in the meta data as a unique name is
      # generated from the deployment name
    labels:
      app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

Download Cluster Logs

To download cluster logs, perform the following steps:

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use the BOSH CLI v2+ to log in to the BOSH Director from the Ops Manager VM. For more information, see [Advanced Troubleshooting with the BOSH CLI](#).

2. After logging in to the BOSH Director, identify the name of your PKS deployment. For example:

```
$ bosh -e pks deployments
```

Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. Identify the names of the VMs you want to retrieve logs from by listing all VMs in your deployment. For example:

```
$ bosh -e pks -d pivotal-container-service-aa1234567bc8de9f0a1c vms
```

4. Download the logs from the VM. For example:

```
$ bosh -e pks \  
-d pivotal-container-service-aa1234567bc8de9f0a1c logs pks/0
```

See the [View Log Files](#) section of the *Diagnostics Tools* topic for information about using cluster logs to diagnose issues in your PKS deployment.

Maintain Workload Uptime

Page last updated:

This topic describes how you can maintain workload uptime for Kubernetes clusters deployed with Pivotal Container Service (PKS).

To maintain workload uptime, configure the following settings in your deployment manifest:

1. Configure [workload replicas](#) to handle traffic during rolling upgrades.
2. Define an [anti-affinity rule](#) to evenly distribute workloads across the cluster.

To increase uptime, you can also refer to the documentation for the services that run on your clusters, and configure your workload based on the recommendations of the software vendor.

About Workload Upgrades

The PKS tile contains an errand that upgrades all Kubernetes clusters. Upgrades run on a single VM at a time. While one worker VM runs an upgrade, the workload on that VM goes down. The additional worker VMs continue to run replicas of your workload, maintaining the uptime of your workload.

To prevent workload downtime during a cluster upgrade, Pivotal recommends running your workload on at least three worker VMs and using multiple replicas of your workloads spread across those VMs. You must edit your manifest to define the replica set and configure an anti-affinity rule to ensure that the replicas run on separate worker nodes.

Set Workload Replicas

Set the number of workload replicas to handle traffic during rolling upgrades. To replicate your workload on additional worker VMs, deploy the workload using a replica set.

Edit the `spec.replicas` value in your deployment manifest:

```
kind: Deployment
metadata:
  # ...
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: APP-NAME
```

See the following table for more information about this section of the manifest:

Key-Value Pair	Description
<pre>spec: replicas: 3</pre>	Set this value to at least 3 to have at least three instances of your workload running at any time.
<pre>app: APP-NAME</pre>	Use this app name when you define the anti-affinity rule later in the spec.

Define an Anti-Affinity Rule

To distribute your workload across multiple worker VMs, you must use anti-affinity rules. If you do not define an anti-affinity rule, the replicated pods can be assigned to the same worker node. See the [Kubernetes documentation](#) for more information about anti-affinity rules.

To define an anti-affinity rule, add the `spec.template.spec.affinity` section to your deployment manifest:

```
kind: Deployment
metadata:
  # ...
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: APP-NAME
    spec:
      containers:
        - name: MY-APP
          image: MY-IMAGE
          ports:
            - containerPort: 12345
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: "app"
                    operator: In
                    values:
                      - APP-NAME
              topologyKey: "kubernetes.io/hostname"
```

See the following table for more information:

Key-Value Pair	Description
<pre>matchExpressions: - key: "app"</pre>	This value matches <code>spec.template.metadata.labels.app</code> .
<pre>values: - APP-NAME</pre>	This value matches the <code>APP-NAME</code> you defined earlier in the spec.

Multi-AZ Worker

Kubernetes evenly spreads pods in a replication controller over multiple Availability Zones (AZs). For more granular control over scheduling pods, add an `Anti-Affinity Rule` to the deployment spec by replacing `"kubernetes.io/hostname"` with `"failure-domain.beta.kubernetes.io/zone"`. For more information on scheduling pods, see [Advanced Scheduling in Kubernetes](#) on the Kubernetes Blog.

Persistent Volumes

Persistent volumes cannot be attached across AZs. Therefore, when persistent volumes are created, the `PersistentVolumeLabel` admission controller automatically adds AZ labels to them. The scheduler then ensures that pods that claim a given volume are only placed into the same AZ as that volume.


If an AZ goes down, the persistent volume along with its data also goes down and cannot be automatically re-attached. To preserve your persistent volume data in the event of a fallen AZ, your persistent workload needs to have a failover mechanism in place.


For example, to ensure the uptime of your persistent volumes during a cluster upgrade, Pivotal recommends that you have at least two nodes per AZ. By configuring your workload as suggested, Kubernetes reschedules pods in the other node of the same AZ while BOSH is performing the upgrade.

Upgrade PKS

Page last updated:

This topic explains how to upgrade the Pivotal Container Service (PKS) tile and existing Kubernetes clusters. It also explains the service interruptions that can result from service changes and upgrades and from failures at the process, VM, and IaaS level.

 **Breaking Change:** PKS v1.1 does not support ABAC. Delete any ABAC clusters before upgrading to v1.1. For more information, see [Existing ABAC Clusters](#) in the *PKS v1.1 Release Notes*.

 **WARNING:** Do not manually upgrade your Kubernetes version. The PKS service includes the compatible Kubernetes version.

Prepare to Upgrade

Before you begin upgrading the PKS tile, consider your workload capacity and uptime requirements. If workers are operating too close to their capacity, the PKS upgrade can fail. View your workload resource usage in Dashboard. For more information, see [Access the Dashboard](#).

If your clusters are near capacity for your existing infrastructure, Pivotal recommends scaling up your clusters before you upgrade. Scale up your cluster by running `pkcs-resize` or create a cluster using a larger plan. For more information, see [Scale Existing Clusters](#).

To prevent workload downtime during a cluster upgrade, Pivotal recommends running your workload on at least three worker VMs, using multiple replicas of your workloads spread across those VMs. For more information, see [Maintain Workload Uptime](#).

Step 1: Upgrade Ops Manager

PKS v1.1 requires Ops Manager v2.1.


1. To upgrade to the required Ops Manager version, follow the procedure detailed in: [Upgrade Ops Manager and Installed Products to v2.1](#).
2. At this time, operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see [About Workload Upgrades](#) and [Create a Cluster](#).

You can monitor the PKS control plane VM by clicking the **Pivotal Container Service** tile, selecting **Status** tab, and reviewing the **Pivotal Container Service** VM's data points. If any data points are at capacity, scale your deployment accordingly.

Step 2: Upgrade the PKS Tile

To upgrade PKS, you follow the same Ops Manager process that you use to install the tile for the first time. Your configuration settings migrate to the new version automatically. To perform an upgrade:

1. Review the [Release Notes](#) for the version you are upgrading to.
2. Download the desired version of the product from [Pivotal Network](#).
3. Navigate to the Ops Manager Installation Dashboard and click **Import a Product** to upload the product file.
4. Under the **Import a Product** button, click + next to **Pivotal Container Service**. This adds the tile to your staging area.
5. Click the newly-added **Pivotal Container Service** tile.
6. Optional: To upgrade all PKS-deployed Kubernetes clusters when you upgrade the PKS tile, follow the next steps:
 - a. Click **Errands**.
 - b. Under **Post-Deploy Errands**, set the **Upgrade all clusters errand** to **Default (On)**. The errand upgrades a single Kubernetes cluster at a time. Upgrading PKS Kubernetes clusters can temporarily interrupt the service, as described [below](#).

 **Note:** If you upgrade PKS from 1.0.x to 1.1, you must enable the **Upgrade All Cluster** errand. This ensures existing clusters can perform resize or delete actions after the upgrade.

(Optional) To monitor the **Upgrade all clusters errand** using the BOSH CLI, do the following:

- i. Log in to the BOSH Director by running `bosh -e MY-ENVIRONMENT log-in` from a VM that can access your PKS deployment. For more information, see [Manage PKS Deployments with BOSH](#).
- ii. Run `bosh -e MY-ENVIRONMENT tasks`.
- iii. Locate the task number for the errand in the # column of the BOSH output.
- iv. Run `bosh task TASK-NUMBER`, replacing `TASK-NUMBER` with the task number you located in the previous step.

c. Click **Save**.

WARNING: If you set the **Upgrade all clusters errand** to **Off**, your Kubernetes cluster version will fall behind the PKS tile version. If your clusters fall more than one version behind the tile, you can no longer upgrade the clusters. You must upgrade your clusters to match the PKS tile version before the next tile upgrade.

7. Review the other configuration panes. Click **Save** on any panes where you make changes.

Note: When you upgrade PKS, you must place singleton jobs in the AZ you selected when you first installed the PKS tile. You cannot move singleton jobs to another AZ.

8. Return to the Installation Dashboard. Under **Pending Changes**, click **INSTALL Pivotal Container Service**. If you changed **Post-Deploy Errands**, confirm that the **Post-Deploy Errands** setting matches the configuration you set in the previous step.

9. Click **Apply Changes**.

10. At this time, operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see [About Workload Upgrades](#) and [Create a Cluster](#).

You can monitor the PKS control plane VM by clicking the **Pivotal Container Service tile**, selecting **Status** tab, and reviewing the **Pivotal Container Service VM's** data points. If any data points are at capacity, scale your deployment accordingly.

Step 3: Upgrade NSX-T (Optional)

If you are deploying PKS on vSphere with NSX-T integration, NSX-T v2.1 is required.

To upgrade PKS with NSX-T, make the following configuration changes to adapt your deployment to new features that have been added in PKS 1.1.0.

1. Create the **NSX Manager Super User Principal Identity Certificate** by following the procedure in [Step 6: Generate and Register Certificates in Installing and Configuring PKS with NSX-T Integration](#).
2. Select the NAT option if Network Address Translation needs to be enforced for the Kubernetes nodes. Clearing this option would allow the nodes to have globally routable IP addresses. For more information, see [NAT Topology](#).
3. PKS 1.1 allows you to specify a dedicated IP block for Node and Pod Networking. Create these IP blocks according to the instructions in [Plan IP Blocks](#). Enter one or more domain servers used by Kubernetes nodes. These domain servers will be used by the nodes that are created on the Node Networks that are dynamically generated at the time of cluster creation. 1.(Optional) To configure a global proxy for all outgoing HTTP/HTTPS traffic from your Kubernetes clusters, do the following:
 - o Under HTTP/HTTPS proxy, select **Enabled**.
 - o Under HTTP Proxy URL, enter the URL of your HTTP/HTTPS proxy endpoint. For example, <http://myproxy.com:1234>. (Optional) If your proxy uses basic authentication, enter the username and password in either HTTP Proxy Credentials or HTTPS Proxy Credentials.
 - o Under No Proxy, enter the service network CIDR where your PKS cluster is deployed. List any additional IP addresses that should bypass the proxy
4. Make sure that the **Enable outbound internet access** checkbox is not selected. This setting is not applicable to vSphere without NSX-T integrations.

Step 4: Upgrade vCenter (Optional)

1. If you are deploying PKS on vSphere, consult [vSphere Version Requirements](#) and upgrade vSphere if necessary.
2. At this time, operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see [About Workload Upgrades](#) and [Create a Cluster](#).

You can monitor the PKS control plane VM by clicking the **Pivotal Container Service tile**, selecting **Status** tab, and reviewing the **Pivotal Container Service VM's** data points. If any data points are at capacity, scale your deployment accordingly.

Upgrade Kubernetes Clusters

If you set the **Upgrade all clusters errand** to **Default (On)**, your PKS-deployed Kubernetes clusters are upgraded automatically when the PKS tile upgrade runs.

If you set the **Upgrade all clusters errand** to **Off**, you can upgrade all PKS-deployed Kubernetes clusters by setting the **Upgrade all clusters errand** to **On** and clicking **Apply Changes**.

⚠ Note: If you upgrade PKS from 1.0.x to 1.1, you must enable the **Upgrade All Cluster** errand. This ensures existing clusters can perform resize or delete actions after the upgrade.

Service Interruptions

Service changes and upgrades and failures at the process, VM, and IaaS level can cause outages in the PKS service, as described below.

Read this section if:

- You are experiencing a service interruption and are wondering why.
- You are planning to update or change a Kubernetes cluster and want to know if it might cause a service interruption.

Stemcell or Service Upgrade

An operator updates a stemcell version or the PKS tile version.

- **Impact:** The PKS API experiences downtime while the new stemcell is applied to the Pivotal Container Service VM.
 - **Required Actions:** None. If the update deploys successfully, apps reconnect automatically.
- **Impact:** Workloads running on single node clusters experience downtime.
 - **Required Actions:** None. If the update deploys successfully, workloads resume automatically. For more information, see [Maintain Workload Uptime](#).

Configure the Upgrade Pipeline

Page last updated:

This topic describes how to set up a Concourse pipeline to perform automatic upgrades of a Pivotal Container Service (PKS) installation.

When you configure the upgrade pipeline, the pipeline upgrades your installation when a new PKS release becomes available on Pivotal Network.

By default, the pipeline upgrades when a new major patch version is available.

For more information about configuring and using Concourse for continuous integration (CI), see the [Concourse documentation](#).

Download the Upgrade Pipeline

Perform the following steps:

1. From a browser, log in to [Pivotal Network](#).
2. Navigate to the **PCF Platform Automation with Concourse** product page to download the upgrade-tile pipeline.



Note: If you cannot access PCF Platform Automation with Concourse on Pivotal Network, contact Pivotal Support.

3. (Optional) Edit [params.yml](#) to configure the pipeline.
 - For example, edit the `product_version_regex` value to follow minor version updates.
 4. Set the pipeline using the `fly` CLI for Concourse. See the [upgrade-tile pipeline documentation](#) for more information.
-

Service Interruptions

Page last updated:

This topic describes events in the lifecycle of a Kubernetes cluster deployed by Pivotal Container Service (PKS) that can cause temporary service interruptions.

Stemcell or Service Update

An operator updates the stemcell version or PKS version.

Impact

- **Workload:** If you run the recommended configuration, no workload downtime is expected since the VMs are upgraded one at a time. See [Maintain Workload Uptime](#) for more information.
- **Kubernetes control plane:** The Kubernetes master VM is recreated during the upgrade, so `kubect1` and the Kubernetes control plane experience a short downtime.

Required Actions

None. If the update deploys successfully, the Kubernetes control plane recovers automatically.

VM Process Failure on a Cluster Master

A process, such as the scheduler or the Kubernetes API server, crashes on the cluster master VM.

Impact

- **Workload:** If the scheduler crashes, workloads that are in the process of being rescheduled may experience up to 120 seconds of downtime.
- **Kubernetes control plane:** Depending on the process and what it was doing when it crashed, the Kubernetes control plane may experience 60-120 seconds of downtime. Until the process resumes, the following can occur:
 - Developers may be unable to deploy workloads
 - Metrics or logging may stop
 - Other features may be interrupted

Required Actions

None. BOSH brings the process back automatically using `monit`. If the process resumes cleanly and without manual intervention, the Kubernetes control plane recovers automatically.

VM Process Failure on a Cluster Worker

A process, such as Docker or `kube-proxy`, crashes on a cluster worker VM.

Impact

- **Workload:** If the cluster and workloads follow the recommended configuration for the number of workers, replica sets, and pod anti-affinity rules, workloads should not experience downtime. The Kubernetes scheduler reschedules the affected pods on other workers. See [Maintain Workload Uptime](#) for more information.

Required Actions

None. BOSH brings the process back automatically using `monit`. If the process resumes cleanly and without manual intervention, the worker recovers automatically, and the scheduler resumes scheduling new pods on this worker.

VM Process Failure on the Pivotal Container Service VM

A process, such as the PKS API server, crashes on the pivotal-container-service VM.

Impact

- **PKS control plane:** Depending on the process and what it was doing, the PKS control plane may experience 60-120 seconds of downtime. Until the process resumes, the following can occur:
 - The PKS API or UAA may be inaccessible
 - Use of the PKS CLI is interrupted
 - Metrics or logging may stop
 - Other features may be interrupted


Required Actions

None. BOSH brings the process back automatically using `monit`. If the process resumes cleanly, the PKS control plane recovers automatically and the PKS CLI resumes working.

VM Failure

A PKS VM fails and goes offline due to either a virtualization problem or a host hardware problem.

Impact

- **If the BOSH Resurrector is enabled**, BOSH detects the failure, recreates the VM, and reattaches the same persistent disk and IP address. Downtime depends on which VM goes offline, how quickly the BOSH Resurrector notices, and how long it takes the IaaS to create a replacement VM. The BOSH Resurrector usually notices an offline VM within one to two minutes. For more information about the BOSH Resurrector, see the [BOSH documentation](#) .
- **If the BOSH Resurrector is not enabled**, some cloud providers, such as vSphere, have similar resurrection or high availability (HA) features. Depending on the VM, the impact can be similar to a key process on that VM going down as described in the previous sections, but the recovery time is longer while the replacement VM is created. See the sections for process failures on the [cluster worker](#), [cluster master](#), and [PKS VM](#) sections for more information.

Required Actions

When the VM comes back online, no further action is required for the developer to continue operations.

AZ Failure

An availability zone (AZ) goes offline entirely or loses connectivity to other AZs (net split).

Impact

The control plane and clusters are inaccessible. The extent of the downtime is unknown.

Required Actions

When the AZ comes back online, the control plane recovers in one of the following ways:

- If BOSH is in a different AZ, BOSH recreates the VMs with the last known persistent disks and IPs. If the persistent disks are gone, the disks can be restored from your last backup and reattached. Pivotal recommends manually checking the state of VMs and databases.
- If BOSH is in the same AZ, follow the directions for [region failure](#).

Region Failure

An entire region fails, bringing all PKS components offline.

Impact

The entire PKS deployment and all services are unavailable. The extent of the downtime is unknown.

Required Actions

The PKS control plane can be restored using [BOSH Backup and Restore](#) (BBR). Each cluster may need to be restored manually from backups.

Delete PKS


To delete PKS, perform the following steps:

1. Navigate to the Ops Manager Installation Dashboard.
 2. Click the trash icon on the PKS tile.
 3. Click **Confirm** in the dialog box that appears.
 4. By default, deleting the PKS tile will also delete all the clusters created by PKS. To preserve the clusters, click the **Delete all clusters** errand under **Pending Changes** and select **Off**.
 5. Click **Apply Changes**.
-

Using PKS


Page last updated:

This topic describes how to use Pivotal Container Service (PKS).

 **Note:** Because PKS does not currently support the Kubernetes Service Catalog or the GCP Service Broker, binding clusters to Kubernetes services is not supported.

The procedures for using PKS have the following prerequisites:

- You must have an external TCP or HTTPS load balancer configured to forward traffic to the PKS API endpoint. For more information, see the *Configure External Load Balancer* section of [Installing and Configuring PKS](#).
- You must know the address of your PKS API endpoint and have a UAA-created user account that has been granted PKS cluster access. For more information, see [Manage Users in UAA](#).

 **Note:** If your PKS installation is integrated with NSX-T, use the DNAT IP address assigned in the [Retrieve the PKS Endpoint](#) section of *Installing and Configuring PKS with NSX-T Integration*.

See the following sections:

- [Create a Cluster](#)
 - [Retrieve Cluster Credentials and Configuration](#)
 - [View Cluster List](#)
 - [View Cluster Details](#)
 - [View Cluster Plans](#)
 - [Using Dynamic Persistent Volumes](#)
 - [Scale Existing Clusters](#)
 - [Access the Dashboard](#)
 - [Deploy and Access Basic Workloads](#)
 - [Delete a Cluster](#)
 - [Log Out of the PKS Environment](#)
-

Create a Cluster

Page last updated:

This topic describes how to create a Kubernetes cluster with Pivotal Container Service (PKS) using the PKS Command Line Interface (CLI).

Configure Cluster Access

Cluster access configuration differs by the type of PKS deployment.


vSphere with NSX-T

PKS deploys a load balancer automatically when clusters are created. The load balancer is configured automatically when workloads are being deployed on these Kubernetes clusters. For more information, see [Load Balancers in PKS Deployments with NSX-T](#).

vSphere without NSX-T or GCP

When you create a Kubernetes cluster, you must configure external access to the cluster by creating an external TCP or HTTPS load balancer. This load balancer allows you to run PKS CLI commands on the cluster from your local workstation. For more information, see [Load Balancers in PKS Deployments without NSX-T](#).

You can configure any load balancer of your choice. If you use GCP or vSphere without NSX-T, you can create a load balancer using your cloud provider console. For information about configuring a GCP load balancer for PKS clusters, see [Configuring a GCP Load Balancer for PKS Clusters](#).

 **Note:** You can configure GCP load balancers only for PKS clusters that are deployed on GCP.

Create the load balancer before you create the cluster, then point the load balancer to the IP address of the master virtual machine (VM) after cluster creation. If the cluster has multiple master nodes, you must configure the load balancer to point to all master VMs for the cluster.

If you are creating a cluster in a non-production environment, you can choose to create a cluster without a load balancer. Create a DNS entry that points to the IP address of the cluster's master VM after cluster creation.

To locate the IP addresses and VM IDs of the master VMs, see [Identify the Kubernetes Cluster Master VM](#) below.

Create a Kubernetes Cluster

Perform the following steps:

1. Grant cluster access to a new or existing user in UAA. See the [Grant Cluster Access to a User](#) section of *Manage Users in UAA* for more information.
2. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

3. Run the following command to create a cluster:

```
pks create-cluster CLUSTER-NAME \
--external-hostname HOSTNAME \
--plan PLAN-NAME \
[--num-nodes WORKER-NODES]
```

Replace the placeholder values in the command as follows:

- `CLUSTER-NAME`: Enter a unique name for your cluster.
- `HOSTNAME`: Enter an external hostname for your cluster. You can use any fully qualified domain name (FQDN) or IP address you own. For example, `my-cluster.example.com` or `10.0.0.1`.
- `PLAN-NAME`: Choose a plan for your cluster. Run `pks plans` to list your available plans.

- (Optional) `WORKER-NODES` : Choose the number of worker nodes for the cluster. If you do not specify a number of worker nodes, the default value is 3. For high availability, Pivotal recommends creating clusters with at least 3 worker nodes. The maximum value is 50.

For example:

```
$ pks create-cluster my-cluster \
--external-hostname my-cluster.example.com \
--plan large --num-nodes 3
```

4. Track the cluster creation process by running `pks cluster CLUSTER-NAME`. Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pks cluster my-cluster
Name:          my-cluster
Plan Name:     large
UUID:         01a234bc-d56e-7f89-01a2-3b4cde5f6789
Last Action:   CREATE
Last Action State: succeeded
Last Action Description: Instance provisioning completed
Kubernetes Master Host: my-cluster.example.com
Kubernetes Master Port: 8443
Worker Instances: 3
Kubernetes Master IP(s): 192.168.20.7
```

If the value for **Last Action State** is `error`, troubleshoot cluster creation by logging in to the BOSH Director and running `bosh tasks`. See [Advanced Troubleshooting with the BOSH CLI](#) for more information.

5. Depending on your deployment:
 - For GCP and vSphere without NSX-T, configure external access to the cluster's master nodes using either DNS records or an external load balancer. Use the output from the `pks cluster` command to locate the master node IP addresses and ports.

 **Note:** For clusters with multiple master node VMs, health checks on port 8443 are recommended.


- For vSphere with NSX-T, choose one of the following:
 - Specify the hostname or FQDN and register the FQDN with the IP provided by PKS after cluster deployment. You can do this using `resolv.conf` or via DNS registration.
 - Specify a temporary placeholder value for FQDN, then replace the FQDN in the `kubeconfig` with the IP address assigned to the load balancer dedicated to the cluster.

To retrieve the IP address to access the Kubernetes API and UI services, use the `pks cluster CLUSTER-NAME` command.

6. To access your cluster, run `pks get-credentials CLUSTER-NAME`. This command creates a local `kubeconfig` that allows you to manage the cluster. For more information about the `pks get-credentials` command, see [Retrieve Cluster Credentials and Configuration](#).
7. Run `kubectl cluster-info` to confirm you can access your cluster using the Kubernetes CLI.

See [Managing PKS](#) for information about checking cluster health and viewing cluster logs.

Identify Kubernetes Cluster Master VMs

 **Note:** This section applies only to PKS deployments on GCP or on vSphere without NSX-T. Skip this section if your PKS deployment is on vSphere with NSX-T. For more information, see [Load Balancers in PKS](#).

To reconfigure the load balancer or DNS record for an existing cluster, you may need to locate VM ID and IP address information for the cluster's master VMs. Use the information you locate in this procedure when configuring your load balancer backend.

To locate the IP addresses and VM IDs for the master VMs of an existing cluster, do the following:

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. To locate the cluster ID and master node IP addresses, run `pks cluster CLUSTER-NAME`. From the output of this command, record the following items:

- **UUID:** This value is your cluster ID.
 - **Kubernetes Master IP(s):** This value lists the IP addresses of all master nodes in the cluster.
3. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use the BOSH CLI to log in to the BOSH Director from the Ops Manager VM. For more information, see [Advanced Troubleshooting with the BOSH CLI](#).
 4. Identify the name of your cluster deployment. For example:

```
$ bosh -e pks deployments
```

Your cluster deployment name begins with `service-instance` and includes the UUID you located in a previous step.

5. Identify the master VM IDs by listing the VMs in your cluster. For example:

```
$ bosh -e pks -d service-instance-aa1234567bc8de9f0a1c vms
```

Your master VM IDs appear in the **VM CID** column.

6. Use the information you gathered in this procedure to configure your load balancer backend. For example, if you use GCP, use the master VM IDs from the previous step in [Reconfiguring a GCP Load Balancer](#).
-

Retrieve Cluster Credentials and Configuration

This topic describes how to use the `pks get-credentials` command in Pivotal Container Service (PKS) using the PKS Command Line Interface (CLI).

The `pks get-credentials` command performs the following actions:

- Fetch the cluster's kubeconfig
- Add the cluster's kubeconfig to the existing kubeconfig
- Create a new kubeconfig, if none exists
- Switch the context to the `CLUSTER-NAME` provided

When you run `pks get-credentials CLUSTER-NAME`, PKS sets the context to the cluster you provide as the `CLUSTER-NAME`. PKS binds your username to the cluster and populates the kubeconfig file on your local workstation with cluster credentials and configuration.

The default path for your kubeconfig is `$HOME/.kube/config`.

If you access multiple clusters, you can choose to use a custom kubeconfig file for each cluster. To save cluster credentials to a custom kubeconfig, use the `KUBECONFIG` environment variable when you run `pks get-credentials`. For example:

```
$ KUBECONFIG=/path/to/my-cluster.config pks get-credentials my-cluster
```

Retrieve Cluster Credentials

Perform the following steps to populate your local kubeconfig with cluster credentials and configuration:

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. Run the following command:

```
pks get-credentials CLUSTER-NAME
```

Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pks get-credentials my-cluster
```

Run kubectl Commands

After PKS populates your kubeconfig, you can use the Kubernetes Command Line Interface (kubectl) to run commands against your Kubernetes clusters.

See [Installing the Kubernetes CLI](#) for information about installing kubectl.

For information about using kubectl, refer to the [Kubernetes documentation](#).

View Cluster List

Follow the steps below to view the list of deployed Kubernetes cluster with the PKS CLI.

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. Run the following command to view the list of deployed clusters, including cluster names and status:

```
$ pks clusters
```

View Cluster Details

Follow the steps below to view the details of an individual cluster using the PKS CLI.

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. Run the following command to view the details of an individual cluster:

```
pks cluster CLUSTER-NAME
```

Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pks cluster my-cluster
```

View Cluster Plans

Follow the steps below to view information about the available plans for deploying a cluster using the PKS CLI.

1. On the command line, run the following command to log in:

```
pkcs login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pkcs login` command.

2. Run the following command to view information about the available plans for deploying a cluster:

```
$ pkcs plans
```

The response lists details about the available plans, including plan names and descriptions:

Name	ID	Description
default		Default plan for K8s cluster

Using Dynamic Persistent Volumes

When using PKS, you can choose to pre-provision persistent storage or create on-demand persistent storage volumes. Refer to the [Kubernetes documentation](#) for more information about storage management.

Perform the steps in this section to define a PersistentVolumeClaim that you can apply to newly-created pods.

1. Download the StorageClass spec for your cloud provider.

- **GCP:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-gcp.yml
```

- **vSphere:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-vsphere.yml
```

2. Apply the spec by running `kubectl create -f STORAGE-CLASS-SPEC.yml`. Replace `STORAGE-CLASS-SPEC` with the name of the file you downloaded in the previous step. For example:

```
$ kubectl create -f storage-class-gcp.yml
```

3. Run the following command to download the example PersistentVolumeClaim:

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/persistent-volume-claim.yml
```

4. Run the following command to apply the PersistentVolumeClaim:

```
$ kubectl create -f persistent-volume-claim.yml
```


- To confirm you applied the PersistentVolumeClaim, run the following command:

```
$ kubectl get pvc -o wide
```

5. To use the dynamic persistent volume, create a pod that uses the PersistentVolumeClaim. See the [pv-guestbook.yml configuration file](#) as an example.

Scale Existing Clusters

Follow the steps below to scale up an existing cluster using the PKS CLI.


 **Note:** You cannot scale the number of worker nodes down. You can only scale the number of worker nodes up.

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. Run the following command below to scale up your cluster. You cannot scale the number of worker nodes down.

 **Note:** This command may roll additional VMs in the cluster, affecting workloads if the worker nodes are at capacity. This issue will be resolved in a future release of PKS.

```
pks resize CLUSTER-NAME --num-nodes WORKER-NODES
```

Replace the placeholder values in the command as follows:


- o `CLUSTER-NAME` is the name of your cluster.
- o `WORKER-NODES` is the number of worker nodes for the cluster. The maximum number of worker nodes is 50. For example:

```
$ pks resize my-cluster --num-nodes 5
```


Access the Dashboard

The Dashboard is a web-based Kubernetes user interface. Use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot containerized applications, and manage the cluster and its resources. Dashboard also provides information about the state of Kubernetes resources in the cluster. You can use Dashboard to manage the cluster at scale, including initiating rolling updates, restarting pods, and deploying new applications.

To access the dashboard, follow the steps in [Accessing the Dashboard UI](#) in the Kubernetes Web UI (Dashboard) documentation.

 **Note:** You must have `kubectl` credentials to access Dashboard. This requirement prevents unauthorized admin access to the Kubernetes cluster through a browser.

Deploy and Access Basic Workloads

Page last updated:

This topic describes how to deploy and access basic workloads in Pivotal Container Service (PKS).


If you use Google Cloud Platform (GCP) or vSphere with NSX-T integration, your cloud provider can configure a load balancer for your workload.

If you use vSphere without NSX-T, you can choose to configure your own external load balancer or expose static ports to access your workload without a load balancer.

- [Access Workloads Using an Internal Load Balancer](#)
- [Access Workloads Using an External Load Balancer](#)
- [Access Workloads without a Load Balancer](#)

Access Workloads Using an Internal Load Balancer

If you use GCP or vSphere with NSX-T, follow the steps below to deploy and access basic workloads using a load balancer configured by your cloud provider.

 **Note:** This approach creates a dedicated load balancer for each workload. This may be an inefficient use of resources in clusters with many apps.

1. Expose the workload using a Service with `type: LoadBalancer`. See the [Kubernetes documentation](#) for more information about the `LoadBalancer` Service type.
2. Download the spec for a basic NGINX app from the [cloudfoundry-incubator/kubo-ci](#) GitHub repository.
3. Run `kubectl create -f nginx.yml` to deploy the basic NGINX app. This command creates three pods (replicas) that span three worker nodes.
4. Wait until your cloud provider creates a dedicated load balancer and connects it to the worker nodes on a specific port.
5. Run `kubectl get svc nginx` and retrieve the load balancer IP address and port number.
6. On the command line of a server with network connectivity and visibility to the IP address of the worker node, run `curl http://EXTERNAL-IP:PORT` to access the app. Replace `EXTERNAL-IP` with the IP address of the load balancer and `PORT` with the port number.

Access Workloads Using an External Load Balancer

All deployments can use an external load balancer. To use an external load balancer, follow the steps below to deploy and access basic workloads.



1. Expose every workload and app using a Service with `type: NodePort`. See the [Kubernetes documentation](#) for more information about the `NodePort` Service type.
2. Map each node port exposed in the worker nodes that you need to an external port in your external load balancer. The process to map these ports depends on your load balancer. See your external load balancer documentation for more information.
3. For each app, run `curl http://LOAD-BALANCER-IP:EXTERNAL-PORT`. Replace `LOAD-BALANCER-IP` with the IP address of your external load balancer and `EXTERNAL-PORT` with the external port number.

Access Workloads without a Load Balancer

If you use vSphere without NSX-T integration, you do not have a load balancer configured by your cloud provider. You can choose to [configure your own external load balancer](#) or follow the procedures in this section to access your workloads without a load balancer.

If you do not use an external load balancer, you can configure the NGINX service to expose a static port on each worker node. From outside the cluster, you can reach the service at `http://NODE-IP:NODE-PORT`.

To expose a static port on your workload, perform the following steps:

1. Download the spec for a basic NGINX app from the [cloudfoundry-incubator/kubo-ci](#)  GitHub repository.
2. Run `kubectl create -f nginx.yml` to deploy the basic NGINX app. This command creates three pods (replicas) that span three worker nodes.
3. Expose the workload using a Service with `type: NodePort`. See the [Kubernetes documentation](#)  for more information about the `NodePort` Service type.
4. Retrieve the IP address for a worker node with a running NGINX pod.



Note: If you deployed more than four worker nodes, some worker nodes may not contain a running NGINX pod. Select a worker node that contains a running NGINX pod.

You can retrieve the IP address for a worker node with a running NGINX pod in one of the following ways:

- On the command line, run `kubectl get nodes`. Select a node name, then locate the node name in the vCenter or GCP Console to find the IP address.
 - On the Ops Manager command line, run `bosh vms` to find the IP address.
5. On the command line, run `kubectl get svc nginx`. Find the node port number in the `3XXXX` range.
 6. On the command line of a server with network connectivity and visibility to the IP address of the worker node, run `curl http://NODE-IP:NODE-PORT` to access the app. Replace `NODE-IP` with the IP address of the worker node, and `NODE-PORT` with the node port number.
-

Delete a Cluster

Follow the steps below to delete a cluster using the PKS CLI. In PKS v1.1, running the `pks delete-cluster` command automatically deletes all NSX objects.

1. On the command line, run the following command to log in:

```
pks login -a PKS-API -u USERNAME -k
```

See [Log in to the PKS CLI](#) for more information about the `pks login` command.

2. Run `pks delete-cluster CLUSTER-NAME` to delete a cluster. Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pks delete-cluster my-cluster
```

Log Out of the PKS Environment

On the command line, run `pkcs logout` to log out of your PKS environment.

After logging out, you must run `pkcs login` before you can run any other `pkcs` commands.

Using Helm with PKS

Page last updated:

This topic describes how you can use the package manager [Helm](#) for your Kubernetes apps running on Pivotal Container Service (PKS).

Helm includes the following components:

Component	Role	Location
helm	Client	Runs on your local workstation
tiller	Server	Runs inside your Kubernetes cluster

Helm packages are called **charts**. See [Charts](#) in the Helm documentation for more information.

Examples of charts:

- [Concourse](#) for CI/CD pipelines
- [Datadog](#) for monitoring
- [MySQL](#) for storage

This topic includes a procedure for installing [Concourse](#) using Helm. For more charts, see the Kubernetes [charts repository](#) on GitHub.

If you want to use Helm with PKS, see the following sections:

- [Configure Tiller](#)
 - [Install Concourse Using Helm](#)
-

Configure Tiller

Tiller runs inside the Kubernetes cluster and requires access to the Kubernetes API. If you use role-based access control (RBAC) in PKS, perform the steps in this section to grant Tiller permission to access the API.

1. Create a service account for Tiller and bind it to the `cluster-admin` role by adding the following section to `rbac-config.yaml`:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: kube-system
```

2. Apply the service account and role by running the following command:

```
$ kubectl create -f rbac-config.yaml
```

3. Download and install the [Helm CLI](#).

4. Deploy Helm using the service account by running the following command:

```
$ helm init --service-account tiller
```

5. Run `helm ls` to verify that the permissions are configured.

To apply more granular permissions to the Tiller service account, see the [Helm RBAC](#) documentation.

Install Concourse Using Helm

Perform the steps in this section to install Concourse using Helm.

1. Download the StorageClass spec for your cloud provider.

- **GCP:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-gcp.yml
```

- **vSphere:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-vsphere.yml
```

2. Apply the spec by running `kubectl create -f STORAGE-CLASS-SPEC.yml`. Replace `STORAGE-CLASS-SPEC` with the name of the file you downloaded in the previous step. For example:

```
$ kubectl create -f storage-class-gcp.yml
```

3. Install the Concourse Helm chart by running `helm install stable/concourse` with the following options:
 - `--name APP-NAME` : (Optional) Replace `APP-NAME` with a name you provide for the installed chart.
 - `--set persistence.worker.storageClass=STORAGE-CLASS` : Replace `STORAGE-CLASS` with your StorageClass to apply the spec to the Concourse worker persistent volumes.
 - `--set postgresql.persistence.storageClass=STORAGE-CLASS` : Replace `STORAGE-CLASS` with your StorageClass to apply the spec to the PostgreSQL database persistent volumes.

For example:

```
$ helm install --name my-concourse --set persistence.worker.storageClass=ci-storage,postgresql.persistence.storageClass=ci-storage stable/concourse
```

4. Forward the port number so that you can access Concourse from localhost. By default, the Concourse chart does not expose services outside the cluster.

- a. Export the pod name as an environment variable. For example:

```
$ export POD_NAME=$(kubectl get pods --namespace default -l "app=concourse-web" -o jsonpath="{.items[0].metadata.name}")
```

- b. Forward the port number by running the following command:

```
$ kubectl port-forward --namespace default $POD_NAME 8080:8080
```

5. Navigate to `http://127.0.0.1:8080` in your browser to access Concourse. Use the default credentials to log in.
6. Log in to your Concourse instance from the command line by running `fly -t MY-CONCOURSE login -c http://127.0.0.1:8080`. For example:

```
$ fly -t ci-helm login -c http://127.0.0.1:8080
```

For more configuration options, see the [Concourse Helm chart](#) documentation.

Backing Up and Restoring PKS

Page last updated:

This section describes how to back up and restore the Pivotal Container Service (PKS) control plane. PKS uses the Cloud Foundry [BOSH Backup and Restore](#) framework to back up and restore PKS deployments.

The PKS control plane includes the following components:

- UAA MySQL database
- PKS API MySQL database

BOSH Backup and Restore (BBR) backs up the PKS control plane components. BBR does not back up cluster data or deployed applications.

BBR orchestrates triggering the backup or restore process on the PKS BOSH deployment, and transfers the backup artifacts to and from the PKS BOSH deployment.

For more information about installing and using BBR, see [Install BOSH Backup and Restore](#), [Back Up the PKS Control Plane](#), and [Restore the PKS Control Plane](#).

See [BBR Exit Codes and Logging](#) for information about troubleshooting BBR.

Install BOSH Backup and Restore

Page last updated:


This topic describes how to install BOSH Backup and Restore (BBR).

You install the `bbr` binary on a jumpbox, and then run `bbr` from the jumpbox to [back up](#) and [restore](#) your PKS deployment.

Step 1: Set Up Your Jumpbox

Set up your jumpbox with the following settings:

- The jumpbox must be able to communicate with the network that contains your PKS deployment. You can use the Ops Manager VM as your jumpbox.
- The jumpbox must have sufficient space for the backup.
- BBR connects to the VMs at their private IP address, so the jumpbox needs to be in the same network as the deployed VMs. BBR does not support SSH gateways.
- BBR copies the backed-up data from the VMs to the jumpbox, so ensure you have minimal network latency between them to reduce transfer times.

 **Note:** BBR uses SSH to orchestrate the backup of your PKS instances using port 22 by default.

Step 2: Transfer BBR to Your Jumpbox

Perform the following steps to transfer the `bbr` binary to your jumpbox:

1. Download the [latest BBR release](#).
2. Change the permissions of `bbr` in order to make it executable:

```
$ chmod a+x bbr
```

3. SCP the binary to your jumpbox:

```
$ scp LOCAL_PATH_TO_BBR/bbr JUMPBOX_USER/JUMPBOX_ADDRESS
```

If your jumpbox has access to the internet, you can also SSH into your jumpbox and use `wget`:

```
$ ssh JUMPBOX_USER/JUMPBOX_ADDRESS -i YOUR_CERTIFICATE.pem
$ wget BBR_RELEASE_URL
$ chmod a+x bbr
```

Back Up the PKS Control Plane

Page last updated:

This topic describes how to use BOSH Backup and Restore (BBR) to back up a PKS deployment.

To perform a restore, see [Restore the PKS Control Plane](#).

Prerequisites

If you want to use the result of the backup to restore to a destination environment, verify that the current environment and the destination environment are compatible. For more information, see [Compatibility of Restore](#).

Before you begin backing up your PKS deployment, perform the following steps:

- Download the root CA certificate for your PKS deployment:
 - From the Ops Manager Installation Dashboard, click your username in the top right corner.
 - Navigate to **Settings > Advanced**.
 - Click **Download Root CA Cert**.
- Locate your PKS BOSH deployment name:
 - From the Ops Manager Installation Dashboard, click the Director tile.
 - Click the **Credentials** tab.
 - Navigate to **Bosh Commandline Credentials** and click **Link to Credential**.
 - Copy the credential value.
 - From the command line, run the following command to retrieve your PKS BOSH deployment name, replacing `BOSH-CLI-CREDENTIALS` with the credential value you copied in the previous step:

```
BOSH-CLI-CREDENTIALS deployments | grep pivotal-container-service
```

Your PKS BOSH deployment name begins with `pivotal-container-service` and includes a unique identifier.

Back Up a PKS Deployment

Perform the following steps to check that your BOSH Director is reachable and has a deployment that can be backed up:

- SSH into your jumpbox.
For general information about the jumpbox, see [Install BOSH Backup and Restore](#).

- Run the BBR pre-backup check:
`BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET bbr deployment --target BOSH-TARGET --username BOSH-CLIENT --deployment DEPLOYMENT-NAME --ca-cert PATH-TO-BOSH-SERVER-CERT pre-backup-check`

Where:

Credential	Location
<code>BOSH-CLIENT-SECRET</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_CLIENT_SECRET</code> .
<code>BOSH-TARGET</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_ENVIRONMENT</code> . You must be able to reach the target address from the workstation where you run <code>bbr</code> commands.
<code>BOSH-CLIENT</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_CLIENT</code> .
<code>PATH-TO-BOSH-CA-CERT</code>	Use the path to the root CA certificate you downloaded in the Prerequisites section.
<code>DEPLOYMENT-NAME</code>	Use the PKS BOSH deployment name you located in the Prerequisites section.


For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \  
bbr deployment \  
--target bosh.example.com \  
--username admin \  
--deployment cf-acceptance-0 \  
--ca-cert bosh.ca.cert \  
pre-backup-check
```

3. If the pre-backup check command fails, do the following:
 - Run the command again adding the `--debug` flag to enable debug logs.
For more information, see [Exit Codes and Logging](#).
 - Make the fix suggested in the output and run the pre-backup check again.
For example, the deployment you selected might not have the correct backup scripts, or the connection to the BOSH Director failed.
4. If the pre-backup check succeeds, run the BBR backup command from your jumpbox to back up your BOSH deployment:


```
BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET nohup bbr deployment --target BOSH-DIRECTOR-IP --username BOSH-CLIENT --deployment DEPLOYMENT-NAME --ca-cert  
PATH-TO-BOSH-SERVER-CERT backup
```


Replace the placeholder values with the same values as above.

 **Note:** If you want to include the manifest in the backup artifact, add the `--with-manifest` flag. However, be aware that the backup artifact then includes credentials that you must keep secret.

For example,

```
$ BOSH_CLIENT_SECRET=p455w0rd \  
nohup bbr deployment \  
--target bosh.example.com \  
--username admin \  
--deployment cf-acceptance-0 \  
--ca-cert bosh.ca.cert \  
backup
```

 **Note:** The BBR backup command can take a long time to complete. You can run it independently of the SSH session so that the process can continue running even if your connection to the jumpbox fails. The command above uses `nohup`, but you could also run the command in a `screen` or `tmux` session.

5. If the command completes successfully, follow the steps in [What To Do with Your Backup Artifact](#) below.
6. If the backup command fails, do the following:
 - Run the command again adding the `--debug` flag to enable debug logs. For more information, see [Exit Codes and Logging](#).
 - Follow the steps in [Recovering from a Failing Command](#).

Recovering from a Failing Command

If the backup fails, follow these steps:

1. Ensure all the parameters in the command are set.
2. Ensure the BOSH Director credentials are valid.
3. If you are backing up a deployment, ensure the deployment you specify in the BBR command exists.
4. Ensure that the jumpbox can reach the BOSH Director.
5. Consult [Exit Codes and Logging](#).
6. If you see the error message `Directory /var/vcap/store/bbr-backup already exists on instance`, run the appropriate cleanup command.
See [Clean Up after a Failed Backup](#) below.
7. If the backup artifact is corrupted discard the failing artifacts and rerun the backup.

Cancel a Backup

Backups can take a long time. If you need to cancel a backup, for example if you realize that the backup is going to fail or that your developers need to push an app in a hurry, follow these steps:

1. Terminate the BBR process by pressing Ctrl-C and typing `yes` to confirm.
2. Because stopping a backup can leave the system in an unusable state and prevent additional backups, follow the procedures in [Clean Up after a Failed Backup](#) below.

Clean Up after a Failed Backup

If your backup process fails, it might leave the BBR backup folder on the instance, causing any subsequent attempts to backup to fail. In addition, BBR might not have run the post-backup scripts, leaving the instance in a locked state.

Follow the steps below to use the BBR cleanup script to tidy up after a failed backup attempt.

1. If the PKS deployment backup failed, run the following command:

```
BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET bbr deployment --target BOSH-TARGET --username BOSH-CLIENT --deployment DEPLOYMENT-NAME --ca-cert PATH-TO-BOSH-CA-CERT backup-cleanup
```

For example,

```
$ BOSH_CLIENT_SECRET=p455w0rd \
bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.crt \
backup-cleanup
```

2. If the cleanup script fails, consult the following table to match the exit codes to an error message.

Value	Error
0	Success
1	General failure
8	The post-backup unlock failed. Your deployment might be in a bad state and require attention.
16	The cleanup failed. This is a non-fatal error indicating that the utility has been unable to clean up open BOSH SSH connections to the deployment VMs. Manual cleanup might be required to clear any hanging BOSH users and connections.

For more information about how to interpret the exit code, see [Exit Codes](#).

Good Practices for Managing Your Backup Artifact

Keep your backup artifact safe by following these steps:

1. Move the backup artifact off the jumpbox to your storage space.
BBR stores each backup in a subdirectory named `DEPLOYMENT-TIMESTAMP` within the current working directory. The backup created by BBR consists of a folder with the backup artifacts and metadata files.
2. Compress and encrypt the backup artifacts when storing them.
3. Make redundant copies of your backup and store them in multiple locations.
This minimizes the risk of losing your backups in the event of a disaster.
4. Each time you redeploy PKS, test your backup artifact by following the procedures in [Restore the PKS Control Plane](#).

Restore the PKS Control Plane

Page last updated:

This topic describes how to use BOSH Backup and Restore (BBR) to restore a PKS deployment.

To back up a PKS deployment with BBR, see the [Back Up the PKS Control Plane](#) topic.


The steps in this topic allow you to restore a PKS deployment.

Compatibility of Restore

This section describes the restrictions for a backup artifact to be restorable to another environment. This section is for guidance only, and Pivotal highly recommends that operators validate their backups by using the backup artifacts in a restore.

The restrictions for a backup artifact to be restorable are the following:

- **Topology:** BBR requires the BOSH topology of a deployment to be the same in the restore environment as it was in the backup environment.
- **Naming of instance groups and jobs:** For any deployment that implements the backup and restore scripts, the instance groups and jobs must have the same names.
- **Number of instance groups and jobs:** For instance groups and jobs that have backup and restore scripts, there must be the same number of instances.
- **Limited validation:** BBR puts the backed up data into the corresponding instance groups and jobs in the restored environment, but cannot validate the restore beyond that. For example, if the MySQL encryption key is different in the restore environment, the BBR restore might succeed although the restored MySQL database is unusable.

 **Note:** A change in VM size or underlying hardware should not affect BBR's ability to restore data, as long as adequate storage space to restore the data exists.

Step 1: Recreate VMs

Before restoring a PKS deployment, you must create the VMs that constitute the deployment.

In a disaster recovery scenario, you can re-create the deployment with your PKS deployment manifest. If you used the `--with-manifest` flag when running the BBR backup command, your backup artifact includes a copy of your manifest.

Step 2: Transfer Artifacts to Jumpbox


Move your BBR backup artifact from your safe storage location to the jumpbox.

For instance, you could SCP the backup artifact to your jumpbox:

```
$ scp LOCAL_PATH_TO_BACKUP_ARTIFACT JUMPBOX_USER/JUMPBOX_ADDRESS
```

If it is encrypted, decrypt it.

Step 3: Restore

 **Note:** The BBR restore command can take a long time to complete. You can run it independently of the SSH session so that the process can continue running even if your connection to the jumpbox fails. The command above uses `nohup`, but you could also run the command in a `screen` or `tmux` session.

Use the optional `--debug` flag to enable debug logs. See the [Exit Codes and Logging](#) topic for more information.

Perform the following steps to restore a PKS deployment:

1. Ensure the PKS deployment backup artifact is in the folder you will run BBR from.
2. Download the root CA certificate for your PKS deployment:
 - a. From the Ops Manager Installation Dashboard, click your username in the top right corner.
 - b. Navigate to **Settings > Advanced**.
 - c. Click **Download Root CA Cert**.
3. Locate your PKS BOSH deployment name:
 - a. From the Ops Manager Installation Dashboard, click the Director tile.
 - b. Click the **Credentials** tab.
 - c. Navigate to **Bosh Commandline Credentials** and click **Link to Credential**.
 - d. Copy the credential value.
 - e. From the command line, run the following command to retrieve your PKS BOSH deployment name, replacing `BOSH-CLI-CREDENTIALS` with the credential value you copied in the previous step:

```
BOSH-CLI-CREDENTIALS deployments | grep pivotal-container-service
```

Your PKS BOSH deployment name begins with `pivotal-container-service` and includes a unique identifier.

4. Run the BBR restore:

```
$ BOSH_CLIENT_SECRET=BOSH_CLIENT_SECRET \
nohup bbr deployment \
--target BOSH_TARGET \
--username BOSH_CLIENT \
--deployment DEPLOYMENT_NAME \
--ca-cert PATH_TO_BOSH_SERVER_CERT \
restore \
--artifact-path PATH_TO_DEPLOYMENT_BACKUP
```

Replace the placeholder values as follows:

Credential	Location
<code>BOSH-CLIENT-SECRET</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_CLIENT_SECRET</code> .
<code>BOSH-TARGET</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_ENVIRONMENT</code> . You must be able to reach the target address from the workstation where you run <code>bbr</code> commands.
<code>BOSH-CLIENT</code>	In the BOSH Director tile, navigate to Credentials > Bosh Commandline Credentials . Record the value for <code>BOSH_CLIENT</code> .
<code>PATH-TO-BOSH-CA-CERT</code>	Use the path to the root CA certificate you downloaded in a previous step.
<code>DEPLOYMENT-NAME</code>	Use the PKS BOSH deployment name you located in a previous step.

If the command fails, try the steps in [Recovering from a Failing Command](#).

Recovering from a Failing Command

1. Ensure all the parameters in the command are set.
2. Ensure the BOSH Director credentials are valid.
3. Ensure the specified BOSH deployment exists.
4. Ensure that the jumpbox can reach the BOSH Director.
5. Ensure the source BOSH deployment is compatible with the target BOSH deployment.
6. If you see the error message `Directory /var/vcap/store/bbr-backup already exists on instance`, run the relevant commands from the [Clean Up After Failed Restore](#) section of this topic.
7. See the [Exit Codes and Logging](#) topic.

Cancel a Restore

If you need to cancel a restore, perform the following steps:

1. Terminate the BBR process by pressing Ctrl-C and typing `yes` to confirm.
2. Stopping a restore can leave the system in an unusable state and prevent future restores. Perform the procedures in the [Clean Up After Failed Restore](#) section to enable future restores.

Clean Up After Failed Restore

If your restore process fails, then the process may leave the BBR restore folder on the instance. As a result, any subsequent restore attempts may also fail. In addition, BBR may not have run the post-restore scripts, which can leave the instance in a locked state.

In order to resolve these issues, run the BBR cleanup script.

To clean up after a failed restore, run the following command:

```
$ BOSH_CLIENT_SECRET=BOSH_CLIENT_SECRET \  
  bbr deployment \  
  --target BOSH_TARGET \  
  --username BOSH_CLIENT \  
  --deployment DEPLOYMENT_NAME \  
  --ca-cert PATH_TO_BOSH_CA_CERT \  
  restore-cleanup
```

If the cleanup script fails, consult the following table to match the exit codes to an error message.

Value	Error
0	Success
1	General failure
8	The post-restore unlock failed. Your deployment may be in a bad state and require attention.
16	The cleanup failed. This is a non-fatal error indicating that the utility has been unable to clean up open BOSH SSH connections to the deployment VMs. Manual cleanup may be required to clear any hanging BOSH users and connections.

For more information about how to interpret the exit code, see the [Exit Codes](#) section of the [Exit Codes and Logging](#) topic.

BBR Exit Codes and Logging

This topic provides information about the exit codes returned by BBR and BBR logging. Use this information when troubleshooting a failed backup or restore using BBR.

Exit Codes

The exit code returned by BBR indicates the status of the backup or restore. The following table matches exit codes to error messages.

Value	Error
0	Success
1	General failure
4	The pre-backup lock failed.
8	The post-backup unlock failed. Your BOSH deployment or BOSH Director may be in a bad state and require attention.
16	The cleanup failed. This is a non-fatal error indicating that the utility has been unable to clean up open BOSH SSH connections to the deployment VMs. Manual cleanup may be required to clear any hanging BOSH users and connections.

If multiple failures occur, your exit code reflects a combination of values. Use bitwise AND to determine which failures occurred.

For example, the exit code `5` indicates that the pre-backup lock failed and a general error occurred.

To check that a bit is set, use bitwise AND, as demonstrated by the following example of exit code `20`:

```
20 & 1 == 1 # false
20 & 4 == 4 # true; lock failed
20 & 8 == 8 # false
20 & 16 == 16 # true; cleanup failed
```

Exit code `20` indicates that the pre-backup lock failed and cleanup failed.

Logging

By default, BBR displays the following:

- The backup and restore scripts that it finds
- When it starts or finishes a stage, such as `pre-backup scripts` or `backup scripts`
- When the process is complete
- When any error occurs

BBR writes any errors associated with stack traces to a file in of the form `bbr-TIMESTAMP.err.log` in the current directory.

If more logging is needed, use the optional `--debug` flag to print the following information:

- Logs about the API requests made to the BOSH server
- All commands executed on remote instances
- All commands executed on local environment
- Standard in and standard out streams for the backup and restore scripts when they are executed

PKS Security

Page last updated:

This section includes security topics for Pivotal Container Service (PKS).

See the following topic:

- [PKS Security Disclosure and Release Process](#)
-

PKS Security Disclosure and Release Process

Page last updated:

This topic describes the processes for disclosing security issues and releasing related fixes for Pivotal Container Service (PKS), Kubernetes, Cloud Foundry Container Runtime (CFCR), VMware NSX, and VMware Harbor.

Security Issues in PKS

Pivotal and VMware provide security coverage for PKS. Please report any vulnerabilities directly to [Pivotal Application Security Team](#) or the [VMware Security Response Center](#).

Security fixes are provided in accordance with the [PCF Security Release Policy](#) and the [Pivotal Support Lifecycle Policy](#).

Where applicable, security issues may be coordinated with the responsible disclosure process for the open source security teams in Kubernetes and Cloud Foundry projects.

Security Issues in Kubernetes

Pivotal and VMware follow the Kubernetes responsible disclosure process to work within the Kubernetes project to report and address suspected security issues with Kubernetes.

This process is discussed in [Kubernetes Security and Disclosure Information](#).

When the Kubernetes project releases security fixes, PKS releases fixes according to the [PCF Security Release Policy](#) and the [Pivotal Support Lifecycle Policy](#).

Security Issues in CFCR

Pivotal and VMware follow the Cloud Foundry responsible disclosure process to work within the Cloud Foundry Foundation to report and address suspected security issues with CFCR.

This process is discussed in [Cloud Foundry Security](#).

When the Cloud Foundry Foundation releases security fixes, PKS releases fixes according to the [PCF Security Release Policy](#) and the [Pivotal Support Lifecycle Policy](#).

Security Issues in VMware NSX

Security issues in VMware NSX are coordinated with the [VMware Security Response Center](#).

Security Issues in VMware Harbor

Security issues in VMware Harbor are coordinated with the [VMware Security Response Center](#).

Diagnosing and Troubleshooting PKS

This topic is intended to provide assistance when diagnosing and troubleshooting issues installing or using Pivotal Container Service (PKS).

See the following sections:

- [Diagnostic Tools](#)
 - [Troubleshooting](#)
-

Diagnostic Tools

Verify PKS CLI Version

The Pivotal Container Service (PKS) CLI interacts with your PKS deployment through the PKS API endpoint. You create, manage, and delete Kubernetes clusters on your PKS deployment by entering commands in the PKS CLI. The PKS CLI is under active development and commands may change between versions.

Run `pks --version` to determine the version of PKS CLI installed locally. For example:

```
$ pks --version
PKS CLI version: 1.0.0-build.3
```

View Log Files

Log files contain error messages and other information you can use to diagnose issues with your PKS deployment. Follow the steps below to access PKS log files.

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use BOSH CLI v2+ to log in to the BOSH Director from the Ops Manager VM. For more information, see [Advanced Troubleshooting with the BOSH CLI](#).
2. After logging in to the BOSH Director, identify the name of your PKS deployment. For example:

```
$ bosh -e pks deployments
```

Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. On a command line, run `bosh -e pks -d YOUR-DEPLOYMENT-NAME vms` to list the virtual machines (VMs) in your PKS deployment. For example:

```
$ bosh -e pks -d pivotal-container-service-aa1234567bc8de9f0a1c vms
```

4. Run `bosh -e pks -d YOUR-DEPLOYMENT-NAME ssh VM-NAME/GUID` to ssh into a PKS VM.
 - To access logs on the master VM, replace `VM-NAME/GUID` with the name of the PKS master VM, and `GUID` with the GUID of the master VM.
 - To access logs on a worker VM, replace `VM-NAME/GUID` with the name of a PKS worker VM, and `GUID` with the GUID of the same worker VM.
5. Run `sudo su` to act as super user on the PKS VM.
6. Navigate to `/var/vcap/sys/log` on the PKS VM:

```
$ cd /var/vcap/sys/log
```

7. Examine the following file:
 - On the PKS master VM, examine the `kube-apiserver` log file.
 - On a PKS worker VM, examine the `kubelet` log file.

Troubleshooting

Page last updated:

PKS API is Slow or Times Out

Symptom

When you run PKS CLI commands, the PKS API times out or is slow to respond.

Explanation

The PKS API control plane VM requires more resources.

Solution

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Select the **Pivotal Container Service** tile.
3. Select the **Resource Config** page.
4. For the **Pivotal Container Service** job, select a **VM Type** with greater CPU and memory resources.
5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.

Cluster Creation Fails

Symptom

When creating a cluster, you run `pkcs cluster CLUSTER-NAME` to monitor the cluster creation status. In the command output, the value for **Last Action State** is `error`.

Explanation

There was an error creating the cluster.

Diagnostics

1. Log in to the BOSH Director and run `bosh tasks`. The output from `bosh tasks` provides details about the tasks that the BOSH Director has run. See [Manage PKS Deployments with BOSH](#) for more information about logging in to the BOSH Director.
2. In the BOSH command output, locate the task that attempted to create the cluster.
3. Find more information about the task by running `bosh -e MY-ENVIRONMENT tasks TASK-NUMBER`. For example:

```
$ bosh -e pks tasks 23
```



See the [BOSH documentation](#) for more information about troubleshooting failed BOSH tasks.

Cannot Access Add-On Features or Functions

Symptom

You cannot access a feature or function provided by a Kubernetes add-on.

Examples include the following:

- You cannot access the Kubernetes [Web UI \(Dashboard\)](#)  in a browser or using the `kubectl` command-line tool.
- [Heapster](#)  does not start.
- Pods cannot resolve DNS names, and error messages report the service `kube-dns` is invalid. If `kube-dns` is not deployed, the cluster typically fails to start.

Explanation

The Kubernetes features and functions listed above are provided by the following PKS add-ons:

- **Kubernetes Dashboard** `kubernetes-dashboard`
- **Heapster:** `heapster`
- **DNS Resolution:** `kube-dns`

To enable these add-ons, Ops Manager must run scripts after deploying PKS. You must configure Ops Manager to automatically run these post-deploy scripts.

Solution

Perform the following steps to configure Ops Manager to run post-deploy scripts to deploy the missing add-ons to your cluster.

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Click the Ops Manager v2.1 tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.



Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.

5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pks delete-cluster` on the command line to delete the cluster. For more information, see the [Delete a Cluster](#) section of *Using PKS*.
9. On the command line, enter `pks create-cluster` to recreate the cluster. For more information, see the [Create a Cluster](#) section of the *Using PKS*.

Error: Failed Jobs

Symptom

In stdout or log files, you see an error message referencing `post-start scripts failed` or `Failed Jobs`.

Explanation

After deploying PKS, Ops Manager runs scripts to start a number of jobs. You must configure Ops Manager to automatically run these post-deploy scripts.

Solution

Perform the following steps to configure Ops Manager to run post-deploy scripts.

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Click the Ops Manager v2.1 tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.



Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.

5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pkcs delete-cluster` on the command line to delete the cluster. For more information, see the [Delete a Cluster](#) section of *Using PKS*.
9. On the command line, enter `pkcs create-cluster` to recreate the cluster. For more information, see the [Create a Cluster](#) section of *Using PKS*.

Error: No Such Host

Symptom

In stdout or log files, you see an error message that includes `lookup vm-WORKER-NODE-GUID on IP-ADDRESS: no such host`.

Explanation

This error occurs on GCP when the Ops Manager Director tile uses 8.8.8.8 as the DNS server. When this IP range is in use, the master node cannot locate the route to the worker nodes.

Solution

Use the Google internal DNS range, 169.254.169.254, as the DNS server.

Error: FailedMount

Symptom

In Kubernetes log files, you see a `Warning` event from kubelet with `FailedMount` as the reason.

Explanation

A persistent volume fails to connect to the Kubernetes cluster worker VM.

Diagnostics

- In your cloud provider console, verify that volumes are being created and attached to nodes.
- From the Kubernetes cluster master node, check the controller manager logs for errors attaching persistent volumes.
- From the Kubernetes cluster worker node, check kubelet for errors attaching persistent volumes.

Resurrecting VMs Causes Incorrect Permissions in vSphere HA

Symptoms

Output resulting from the `bosh vms` command alternates between showing that the VMs are `failing` and showing that the VMs are `running`. The operator must run the `bosh vms` command multiple times to see this cycle.

Explanation

The VMs' permissions are altered during the restarting of the VM so operators have to reset permissions every time the VM reboots or is redeployed.

VMs cannot be successfully resurrected if the resurrection state of your VM is set to `off` or if the the vSphere HA restarts the VM before BOSH is aware that the VM is down. For more information on VM resurrection, see [Resurrection](#) in the Cloud Foundry BOSH documentation.

Solution

Run the following command on all of your master and worker VMs:

```
bosh -environment BOSH-DIRECTOR-NAME -deployment DEPLOYMENT-NAME ssh INSTANCE-GROUP-NAME -c "sudo /var/vcap/jobs/kube-controller-manager/bin/pre-start; sudo /var/vcap/jobs/kube-apiserver/bin/post-start"
```


Where:

- `BOSH-DIRECTOR-NAME` is your BOSH Director name.
- `DEPLOYMENT-NAME` is the name of your BOSH deployment.
- `INSTANCE-GROUP-NAME` is the name of the BOSH instance group you are referencing.

The above command, when applied to each VM, gives your VMs the correct permissions.

PKS CLI

Page last updated:

This topic describes how to use the Pivotal Container Service Command Line Interface (PKS CLI) to interact with the PKS API.

The [PKS CLI](#) is used to create, manage, and delete Kubernetes clusters. To deploy workloads to a Kubernetes cluster created using the PKS CLI, use the Kubernetes CLI, [kubectl](#).

Current Version: 1.1.0-build297

pks login

Login to PKS

Synopsis

The login command requires -a to target the IP of your PKS API, -u for username and -p for password

```
pks login [flags]
```

Examples

```
pks login -a <API> -u <USERNAME> -p <PASSWORD> [--ca-cert <PATH TO CERT> | -k]
```

Options

-a, --api string	The PKS API server URI
--ca-cert string	Path to CA Cert for PKS API
-h, --help	help for login
-p, --password string	Password
-k, --skip-ssl-verification	Skip SSL Verification
-u, --username string	Username

pks get-credentials

Allows you to connect to a cluster and use kubectl

Synopsis

Run this command in order to update a kubeconfig file so you can access the cluster through kubectl

```
pks get-credentials <CLUSTER-NAME> [flags]
```

Examples

```
pks get-credentials my-cluster
```

Options

```
-h, --help  help for get-credentials
```

pks cluster

View the details of the cluster

Synopsis

Run this command to see details of your cluster such as name, host, port, ID, number of worker nodes, last operation, etc.

```
pks cluster [flags]
```

Examples

```
pks cluster my-cluster
```

Options

```
-h, --help  help for cluster
--json      Return the PKS-API output as json
```

pks clusters

Show all clusters created with PKS

Synopsis

This command describes the clusters created via PKS, and the last action taken on the cluster

```
pks clusters [flags]
```

Examples

```
pks clusters
```

Options

```
-h, --help  help for clusters
--json      Return the PKS-API output as json
```

pks create-cluster

Creates a kubernetes cluster, requires cluster name and an external host name

Synopsis

Create-cluster requires a cluster name, as well as an external hostname. External hostname can be a loadbalancer, from which you access your kubernetes API (aka, your cluster control plane)

```
pks create-cluster <CLUSTER-NAME> [flags]
```

Examples

```
pks create-cluster my-cluster --external-hostname example.hostname --plan production
```

Options

```
-e, --external-hostname string  Address from which to access Kubernetes API
-h, --help                    help for create-cluster
--json                        Return the PKS-API output as json
--non-interactive              Don't ask for user input
-n, --num-nodes string         Number of worker nodes
-p, --plan string              Preconfigured plans. Run pks plans for more details
--wait                        Wait for the operation to finish
```

pks delete-cluster

Deletes a kubernetes cluster, requires cluster name

Synopsis

Delete-cluster requires a cluster name.

```
pks delete-cluster <CLUSTER-NAME> [flags]
```

Examples

```
pks delete-cluster my-cluster
```

Options

```
-h, --help                help for delete-cluster
--non-interactive          Don't ask for user input
--wait                    Wait for the operation to finish
```

pks plans

View the preconfigured plans available

Synopsis

This command describes the preconfigured plans available

```
pks plans [flags]
```

Examples

```
pks plans
```

Options

```
-h, --help    help for plans
--json       Return the PKS-API output as json
```

pks resize

Increases the number of worker nodes for a cluster

Synopsis

Resize requires a cluster name, and the number of desired worker nodes. Users can only scale UP clusters, to a maximum of 50 worker nodes and not scale down. By default, the resize command prompts for interactive confirmation.

```
pks resize <CLUSTER-NAME> [flags]
```

Examples

```
pks resize my-cluster --num-nodes 5
```

Options

```
-h, --help          help for resize
--json              Return the PKS-API output as json. Only applicable when used with --wait flag
--non-interactive    Don't ask for user input
-n, --num-nodes int32 Number of worker nodes (default 1)
--wait              Wait for the operation to finish
```

pks logout

Logs user out of the PKS API

Synopsis

Logs user out of the PKS API. Does not remove kubeconfig credentials or kubectl access.

```
pks logout [flags]
```

Examples

```
pks logout
```

Options

```
-h, --help  help for logout
```
