

Pivotal Container Service (PKS)

Version 1.0

Published: 16 Oct 2018

Table of Contents

Table of Contents	2
Pivotal Container Service (PKS)	4
PKS Release Notes	7
PKS Concepts	17
PKS Cluster Management	18
PKS API Authentication	21
Load Balancers in PKS	22
PKS Prerequisites	24
Installing the PKS CLI	25
Installing the Kubernetes CLI	27
Preparing to Install PKS on vSphere	29
vSphere Prerequisites and Resource Requirements	30
Firewall Ports and Protocols Requirements for vSphere with NSX-T	32
Preparing to Deploy PKS on vSphere	34
Deploying Ops Manager to vSphere	40
Configuring Ops Manager on vSphere	44
VMware Harbor Registry	56
Preparing to Install PKS on GCP	58
GCP Prerequisites and Resource Requirements	59
Preparing to Deploy PKS on GCP	60
Deploying Ops Manager to GCP	66
Configuring Ops Manager on GCP	68
Configuring a GCP Load Balancer for the PKS API	80
Configuring a GCP Load Balancer for PKS Clusters	83
Installing PKS	86
Installing and Configuring PKS	87
Installing and Configuring PKS with NSX-T Integration	93
Upgrading PKS	104
What Happens During PKS Upgrades	105
Upgrade PKS	107
Maintain Workload Uptime	109
Configure the Upgrade Pipeline	111
Managing PKS	112
Configure PKS API Access	113
Manage Users in UAA	114
Manage PKS Deployments with BOSH	116
Add Custom Workloads	117
Download Cluster Logs	118
Service Interruptions	119
Delete PKS	122
Using PKS	123
Create a Cluster	124
Retrieve Cluster Credentials and Configuration	126
View Cluster List	127
View Cluster Details	128
View Cluster Plans	129
Using Dynamic Persistent Volumes	130
Scale Existing Clusters	131

Access Dashboard	132
Deploy and Access Basic Workloads	133
Delete a Cluster	135
Log Out of the PKS Environment	136
Using Helm with PKS	137
Configure Tiller	138
Install Concourse Using Helm	139
Diagnosing and Troubleshooting PKS	140
Diagnostic Tools	141
Troubleshooting	142
PKS CLI	145
PKS Security Disclosure and Release Process	150

Pivotal Container Service (PKS)

Page last updated:

Pivotal Container Service (PKS) enables operators to provision, operate, and manage enterprise-grade Kubernetes clusters using BOSH and Pivotal Ops Manager.

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.

What PKS Adds to Kubernetes

The following table details the features that PKS adds to the Kubernetes platform.

Feature	Included in K8s	Included in PKS
Single tenant ingress	✓	✓
Secure multi-tenant ingress		✓
Stateful sets of pods	✓	✓
Multi-container pods	✓	✓
Rolling upgrades to pods	✓	✓
Rolling upgrades to cluster infrastructure		✓
Pod scaling and high availability	✓	✓
Cluster provisioning and scaling		✓
Monitoring and recovery of cluster VMs and processes		✓
Persistent disks	✓	✓
Secure container registry		✓
Embedded, hardened operating system		✓

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

On vSphere, PKS supports deploying and running Kubernetes clusters in air-gapped environments.

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](#).

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

PKS Concepts

For conceptual information about PKS, see [PKS Concepts](#).

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 v2.0 or 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 v2.0	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](#).

Diagnosing and Troubleshooting PKS

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

Please send any feedback you have to pkcs-feedback@pivotal.io.


PKS Release Notes

PKS (Pivotal Container Service) is used to create and manage on-demand Kubernetes clusters via the PKS CLI.

v1.0.4

Release Date: May 21, 2018

Upgrade Procedure

 **Note:** Upgrade to PKS v1.0.4 from either PKS v1.0.2 or PKS v1.0.3. Do not upgrade PKS v1.0.0 directly to v1.0.4. Instead, upgrade to v1.0.2, then v1.0.4. Alternatively, do a unique install of PKS v1.0.4.

To upgrade to PKS v1.0.4, follow the procedures in [Upgrade PKS](#).

Features

- Updates Kubernetes to v1.9.7.

Component Versions

PKS v1.0.4 includes or supports the following component versions:

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.0.X and 2.1.X	Separate download available from Pivotal Network
vSphere	6.5, 6.5 U1, and 6.5 U2 - 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 Harbor Registry	1.4.2	Separate download available from Pivotal Network
NSX-T	2.1 Advanced Edition	Available from VMware
Stemcell	3468.X	Floating stemcell line available to download from Pivotal Network
Kubernetes	1.9.7*	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.13	Packaged in the PKS Tile
Golang	1.9.5	Packaged in the PKS Tile
NCP	2.1.3	Packaged in the PKS Tile
Kubernetes CLI	1.9.7*	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.0.3-build.15	Separate download available from the PKS section of Pivotal Network
UAA	55	
* Components marked with an asterisk have been patched to resolve security vulnerabilities or fix component behavior.		

Known Issues

This section includes known issues with PKS v1.0.4 and corresponding workarounds.

Access to the Kubernetes API is Unavailable During Upgrades

PKS upgrades include upgrades to the master node. While the master node is undergoing an upgrade, the Kubernetes API is unavailable.

If you attempt to access the API during an upgrade, you will not be able to connect.

Stemcell Updates Cause Automatic VM Upgrading

Enabling the **Upgrade all clusters** errand allows automatic upgrading for VMs in your deployment. Pivotal recommends enabling this errand to ensure that all deployed cluster VMs are patched.

When you enable the **Upgrade all clusters** errand, the following actions can cause downtime:

- Updating the PKS tile with a new stemcell triggers updating each VM in each cluster.
- Updating other tiles in your deployment with new stemcells causes the upgrading of the PKS tile.

Upgrade Errand Fails with Failed Deployments

The **Upgrade all clusters** errand fails if any deployments are in a failed state.

To work around this issue, [delete the failed cluster](#) using the PKS CLI or [redploy the failed cluster](#) with the BOSH CLI to ensure the cluster is in a successful state.

Pods Lose Network Connectivity After VM Cold Migration

When a Kubernetes cluster worker VM goes through cold migration in vSphere, newly provisioned pods lose network connectivity.

This issue can occur under the following conditions:

- When the VM is powered off and is subject to cold migration, and the VM moves to a different ESXi host
- When the VM is powering on and is subject to Distributed Resource Scheduler (DRS) before the powerup completes
- When the vNIC of the VM is detached and reattached

To work around this issue, delete the worker VM. BOSH recreates the worker VM and restores network connectivity.


Kubernetes Cluster Creation Fails if NSX-T Manager Password Begins with Certain Special Characters

If you select NSX-T as a **Container Network Type** in PKS and your NSX-T Manager password begins with an `@`, `$`, `^`, `'`, or space character, Kubernetes cluster creation fails. To resolve this issue, reset your NSX-T Manager password so that it does not begin with any of these characters. After resetting your NSX-T Manager password, reconfigure your NSX-T Manager credentials in the PKS tile with the updated password.

v1.0.3

Release Date: May 4, 2018

Upgrade Procedure

 **Note:** The only supported upgrade path for PKS v1.0.3 is from PKS v1.0.2. Do not upgrade PKS v1.0.0 directly to v1.0.3. Instead, upgrade to v1.0.2, then v1.0.3. Alternatively, do a unique install of PKS v1.0.3.

To upgrade to PKS v1.0.3, perform the following steps:

1. Download the latest 3468.x stemcell from [Pivotal Network](#) and configure the PKS tile with the stemcell.
2. Create a new worker node service account.
 - To create the service account on GCP, see [Create the Worker Node Service Account](#).

- To create the service account on vSphere, see [Create the Worker Node Service Account](#).

3. Follow the procedures in [Upgrade PKS](#). When configuring the **Kubernetes Cloud Provider** configuration screen in the PKS tile, configure the new worker node credentials or service account key as appropriate for your IaaS.

Features

- Separates the master and worker node credentials.
- Updates Kubernetes to v1.9.6.
- Updates Golang to v1.9.5.

Component Versions

PKS v1.0.3 includes or supports the following component versions:

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.0.X and 2.1.X	Separate download available from Pivotal Network
vSphere	6.5 and 6.5 U1 - 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 Harbor Registry	1.4.1	Separate download available from Pivotal Network
NSX-T	2.1 Advanced Edition	Available from VMware
Stemcell	3468.X	Floating stemcell line available to download from Pivotal Network
Kubernetes	1.9.6*	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.13	Packaged in the PKS Tile
Golang	1.9.5*	Packaged in the PKS Tile
NCP	2.1.3*	Packaged in the PKS Tile
Kubernetes CLI	1.9.6*	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.0.3-build.15*	Separate download available from the PKS section of Pivotal Network
UAA	55*	

** Components marked with an asterisk have been patched to resolve security vulnerabilities or fix component behavior.*

Known Issues

This section includes known issues with PKS v1.0.3 and corresponding workarounds.

Access to the Kubernetes API is Unavailable During Upgrades

PKS upgrades include upgrades to the master node. While the master node is undergoing an upgrade, the Kubernetes API is unavailable.

If you attempt to access the API during an upgrade, you will not be able to connect.

Stemcell Updates Cause Automatic VM Upgrading

Enabling the **Upgrade all clusters** errand allows automatic upgrading for VMs in your deployment. Pivotal recommends enabling this errand to ensure that all deployed cluster VMs are patched.

When you enable the **Upgrade all clusters** errand, the following actions can cause downtime:

- Updating the PKS tile with a new stemcell triggers updating each VM in each cluster.
- Updating other tiles in your deployment with new stemcells causes the upgrading of the PKS tile.

Upgrade Errand Fails with Failed Deployments

The **Upgrade all clusters** errand fails if any deployments are in a failed state.

To work around this issue, [delete the failed cluster](#) using the PKS CLI or [redeploy the failed cluster](#) with the BOSH CLI to ensure the cluster is in a successful state.

Pods Lose Network Connectivity After VM Cold Migration

When a Kubernetes cluster worker VM goes through cold migration in vSphere, newly provisioned pods lose network connectivity.

This issue can occur under the following conditions:

- When the VM is powered off and is subject to cold migration, and the VM moves to a different ESXi host
- When the VM is powering on and is subject to Distributed Resource Scheduler (DRS) before the powerup completes
- When the vNIC of the VM is detached and reattached

To work around this issue, delete the worker VM. BOSH recreates the worker VM and restores network connectivity.

StatefulSets Pod Failure After Recreating a VM

When using vSphere with NSX-T integration, if you recreate a node that hosts a StatefulSets pod, the pod can get stuck in a `ContainerCreating` state. The pod emits a warning event with a `FailedCreatePodSandBox` reason. This issue affects StatefulSets pods created before PKS v1.0.3.

A fix for this bug is included in PKS v1.0.3, but the fix applies only to StatefulSets created using PKS v1.0.2 or later. After upgrading PKS to v1.0.3, manually deleting and recreating all preexisting StatefulSets pods is recommended, even if they are in a running state.

To get all StatefulSets pods, run the following command on every Kubernetes cluster using the Kubernetes admin user permissions:

```
$ kubectl get pods -l "statefulset.kubernetes.io/pod-name" \
-o wide --all-namespaces
```

For each result, delete the pod by running the following command:

```
$ kubectl delete pod POD-NAME -n POD-NAMESPACE
```

You do not need to manually recreate the deleted pods. Kubernetes detects a StatefulSet with missing pods and automatically recreates the pods.

[Kubernetes Bug] Upgrading a Cluster Affects Persistent Workload Uptime

During an upgrade to v1.0.3 on vSphere, persistent storage volumes do not reattach to pods until all worker nodes have been upgraded, which results in workload downtime until the entire cluster is upgraded.

This issue occurs when you deploy a pod with persistent storage attached, drain the node, and then immediately delete the node VM.

The expected behavior is for persistent disks to reattach to the upgraded VMs after the pod is restored. However, a Kubernetes bug prevents the disk from reattaching. PKS v1.0.3 works around this bug by attaching the volumes after all workers are upgraded.

For more information, see the [Kubernetes issue on GitHub](#).

In rare cases, pods with persistent volumes can stay in `ContainerCreating` state. If you see the error `FailedMount Unable to mount volumes for pod POD-NAME`, perform the following steps:

1. Find the problem node by running `kubectl describe pod POD-NAME`.

2. Prevent scheduling on the node that runs the pod by running `kubect1 cordon NODE-NAME`.
3. Delete pod by running `kubect1 delete pod POD-NAME`.
4. Wait for pod to be rescheduled and enter `Running` state. This may take several minutes.
5. Resume scheduling on the node that runs the pod by running `kubect1 uncordon NODE-NAME`.

Kubernetes Cluster Creation Fails if NSX-T Manager Password Begins with Certain Special Characters

If you select NSX-T as a **Container Network Type** in PKS and your NSX-T Manager password begins with an `@`, `$`, `^`, `'`, or space character, Kubernetes cluster creation fails. To resolve this issue, reset your NSX-T Manager password so that it does not begin with any of these characters. After resetting your NSX-T Manager password, reconfigure your NSX-T Manager credentials in the PKS tile with the updated password.

v1.0.2

Release Date: April 12, 2018

Upgrade Procedure

To upgrade to PKS v1.0.2, perform the following steps:

1. Download the [docker_ctl](#) script.
2. Download the [docker_ctl_update.sh](#) script.
3. Log in to the BOSH Director by running `bosh -e MY-ENVIRONMENT log-in` from a VM that can access your PKS deployment. Replace `MY-ENVIRONMENT` with the BOSH alias for your PKS environment. See [Manage PKS Deployments with BOSH](#) for more information.

If you choose to log in from the Ops Manager VM, perform the following steps:

- a. Run `sudo apt-get update`.
 - b. Run `sudo apt-get install jq`.
4. Run `export BOSH_ENVIRONMENT=MY-ENVIRONMENT`. Replace `MY-ENVIRONMENT` with the BOSH alias for your PKS environment.
 5. Run the `docker_ctl_update.sh` script. This script contains the fix to correctly unmount Docker overlays. See the corresponding [known issue](#) for more information.
 6. Download the latest 3468.x stemcell from [Pivotal Network](#) and configure the PKS tile with the stemcell.
 7. Follow the procedures in [Upgrade PKS](#).

Features

- Updates Kubernetes to v1.9.5.
- Updates Golang to v1.9.4.

Fixed Issues

General

- Worker nodes are now drained before they stop in order to minimize workload downtime during a rolling upgrade.
- UAA credentials and vCenter passwords no longer appear in [BOSH logs](#).
- BOSH DNS no longer causes worker nodes to fail after a manual restart.
- The Kubernetes Controller Manager certificate no longer contains additional whitespace.

- Drain user now has additional permissions to remove replication controller-owned pods.
- Unmounting Docker overlay volumes no longer causes BOSH unmount failures.
- Addresses upgrade issues in constrained environments.

vSphere

- vSphere NSX-T integration now works with BOSH [stemcell v3468.25](#) and later.
- For vSphere with NSX-T, the pod logical switch port (LSP) is now updated when you recreate the VM that hosts the pod. See [StatefulSets](#) in the Kubernetes documentation and the [known issue](#) below for more information.
- Added support for [special characters](#) `#`, `&`, `;`, `"`, `'`, `^`, `\`, space (), `%`, and `!` in vCenter passwords in the Kubernetes Cloud Provider tile configuration page.
- Drain script now deletes nodes to fix a vSphere issue where node names changed between 1.9.2 and 1.9.5.

Component Versions

PKS v1.0.2 includes or supports the following component versions:

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.0.X and 2.1.X	Separate download available from Pivotal Network
vSphere	6.5 and 6.5 U1 - 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 Harbor Registry	1.4.1	Separate download available from Pivotal Network
NSX-T	2.1 Advanced Edition	Available from VMware
Stemcell	3468.X*	Floating stemcell line available to download from Pivotal Network
Kubernetes	1.9.5*	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.13	Packaged in the PKS Tile
Golang	1.9.4*	Packaged in the PKS Tile
NCP	2.1.2*	Packaged in the PKS Tile
Kubernetes CLI	1.9.5*	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.0.2-build.4*	Separate download available from the PKS section of Pivotal Network

* Components marked with an asterisk have been patched to resolve security vulnerabilities or fix component behavior.

Known Issues

This section includes known issues with PKS v1.0.2 and corresponding workarounds.

Access to the Kubernetes API is Unavailable During Upgrades

PKS upgrades include upgrades to the master node. While the master node is undergoing an upgrade, the Kubernetes API is unavailable.

If you attempt to access the API during an upgrade, you will not be able to connect.

Volume Unmount Failure After Stemcell Upgrade

During an upgrade to PKS v1.0.2, BOSH can fail to unmount the `/var/vcap/store` volume on worker nodes. This is due to an issue with the Docker BOSH

release installed by the PKS v1.0.0 tile.

In this version of the BOSH release, Docker occasionally fails to unmount all overlays when stopping a node. When you upgrade the stemcell for the PKS tile, BOSH recreates VMs and can fail to correctly unmount Docker overlays.

To avoid this issue, follow the steps in the [Upgrade Procedure](#) section when you upgrade the PKS tile. The `docker_ctl_update.sh` script correctly unmounts Docker overlays by replacing the `docker_ctl` script on all worker nodes that have Docker deployed.

Stemcell Updates Cause Automatic VM Upgrading

Enabling the **Upgrade all clusters** errand allows automatic upgrading for VMs in your deployment. Pivotal recommends enabling this errand to ensure that all deployed cluster VMs are patched.

When you enable the **Upgrade all clusters** errand, the following actions can cause downtime:

- Updating the PKS tile with a new stemcell triggers updating each VM in each cluster.
- Updating other tiles in your deployment with new stemcells causes the upgrading of the PKS tile.

Upgrade Errand Fails with Failed Deployments

The **Upgrade all clusters** errand fails if any deployments are in a failed state.

To work around this issue, [delete the failed cluster](#) using the PKS CLI or [redploy the failed cluster](#) with the BOSH CLI to ensure the cluster is in a successful state.

Pods Lose Network Connectivity After VM Cold Migration

When a Kubernetes cluster worker VM goes through cold migration in vSphere, newly provisioned pods lose network connectivity.

This issue can occur under the following conditions:

- When the VM is powered off and is subject to cold migration, and the VM moves to a different ESXi host
- When the VM is powering on and is subject to Distributed Resource Scheduler (DRS) before the powerup completes
- When the vNIC of the VM is detached and reattached

To work around this issue, delete the worker VM. BOSH recreates the worker VM and restores network connectivity.

StatefulSets Pod Failure After Recreating a VM

When using vSphere with NSX-T integration, if you recreate a node that hosts a StatefulSets pod, the pod can get stuck in a `ContainerCreating` state. The pod emits a warning event with a `FailedCreatePodSandBox` reason. This issue affects StatefulSets pods created before PKS v1.0.2.

A fix for this bug is included in PKS v1.0.2, but the fix applies only to StatefulSets created using PKS v1.0.2 or later. After upgrading PKS to v1.0.2, manually deleting and recreating all preexisting StatefulSets pods is recommended, even if they are in a running state.

To get all StatefulSets pods, run the following command on every Kubernetes cluster using the Kubernetes admin user permissions:

```
$ kubectl get pods -l "statefulset.kubernetes.io/pod-name" \
-o wide --all-namespaces
```

For each result, delete the pod by running the following command:

```
$ kubectl delete pod POD-NAME -n POD-NAMESPACE
```

You do not need to manually recreate the deleted pods. Kubernetes detects a StatefulSet with missing pods and automatically recreates the pods.

[Kubernetes Bug] Upgrading a Cluster Affects Persistent Workload Uptime

During an upgrade to v1.0.2 on vSphere, persistent storage volumes do not reattach to pods until all worker nodes have been upgraded, which results in

workload downtime until the entire cluster is upgraded.

This issue occurs when you deploy a pod with persistent storage attached, drain the node, and then immediately delete the node VM.

The expected behavior is for persistent disks to reattach to the upgraded VMs after the pod is restored. However, a Kubernetes bug prevents the disk from reattaching. PKS v1.0.2 works around this bug by attaching the volumes after all workers are upgraded.

For more information, see the [Kubernetes issue on GitHub](#).

In rare cases, pods with persistent volumes can stay in `ContainerCreating` state. If you see the error `FailedMount Unable to mount volumes for pod POD-NAME`, perform the following steps:

1. Find the problem node by running `kubectl describe pod POD-NAME`.
2. Prevent scheduling on the node that runs the pod by running `kubectl cordon NODE-NAME`.
3. Delete pod by running `kubectl delete pod POD-NAME`.
4. Wait for pod to be rescheduled and enter `Running` state. This may take several minutes.
5. Resume scheduling on the node that runs the pod by running `kubectl uncordon NODE-NAME`.

Kubernetes Cluster Creation Fails if NSX-T Manager Password Begins with Certain Special Characters

If you select NSX-T as a **Container Network Type** in PKS and your NSX-T Manager password begins with an `@`, `$`, `^`, `'`, or space character, Kubernetes cluster creation fails. To resolve this issue, reset your NSX-T Manager password so that it does not begin with any of these characters. After resetting your NSX-T Manager password, reconfigure your NSX-T Manager credentials in the PKS tile with the updated password.

v1.0.0

Release Date: February 8, 2018

Features

- Create, resize, delete, list, and show clusters through the PKS CLI
- Native support for NSX-T and Flannel
- Easily obtain kubeconfigs to use each cluster
- Use kubectl to view the Kubernetes dashboard
- Define plans that pre-configure VM size, authentication, default number of workers, and addons when creating Kubernetes clusters
- User/Admin configurations for access to PKS API
- Centralized logging through syslog

Component Versions

PKS v1.0.0 includes or supports the following component versions:

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.0.0 - 2.0.5	Separate download available from Pivotal Network
vSphere	6.5 and 6.5 U1 - 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 Harbor Registry	1.4.1	Separate download available from Pivotal Network
NSX-T	2.1 Advanced Edition	Available from VMware

Stemcell	3468.21	Separate download available from Pivotal Network
Kubernetes	1.9.2	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.13	Packaged in the PKS Tile
NCP	2.1.0.1	Packaged in the PKS Tile
Kubernetes CLI	1.9.2	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.0.0-build.3	Separate download available from the PKS section of Pivotal Network

Known Issues

This section includes known issues with PKS v1.0.0 and corresponding workarounds.

Access to the Kubernetes API is Unavailable During Upgrades

PKS upgrades include upgrades to the master node. While the master node is undergoing an upgrade, the Kubernetes API is unavailable.

If you attempt to access the API during an upgrade, you will not be able to connect.

Special Characters

In PKS v1.0.0, special characters, such as #, &, :, ", ', ^, \, space (), !, and % cannot be used in vCenter passwords. To resolve this issue, reset your password so that it does not include any of the special characters listed above. After resetting your password in vCenter, reconfigure your credentials in the PKS tile with the updated password. [PKS v1.0.2](#) adds support for the special characters listed above.

Stemcell Incompatibility with NSX-T

When deploying PKS v1.0.0 using NSX-T as the networking layer with a stemcell other than 3468.21, Kubernetes cluster deployments fail. [PKS v1.0.2](#) adds support for stemcells v3468.25 and later.

Stemcell Updates Cause Automatic VM Upgrading

Enabling the **Upgrade all clusters** errand allows automatic upgrading for VMs in your deployment. Pivotal recommends enabling this errand to ensure that all deployed cluster VMs are patched.

When you enable the **Upgrade all clusters** errand, the following actions can cause downtime:

- Updating the PKS tile with a new stemcell triggers the rolling of each VM in each cluster.
- Updating other tiles in your deployment with new stemcells causes the rolling of the PKS tile.

Upgrade Errand Fails with Failed Deployments

The **Upgrade all clusters** errand fails if any deployments are in a failed state.

To work around this issue, [delete the failed cluster](#) using the PKS CLI or [redploy the failed cluster](#) with the BOSH CLI to ensure the cluster is in a successful state.

Syslog Security Recommendations

BOSH Director logs contain sensitive information that should be considered privileged. For example, these logs may contain cloud provider credentials in PKS v1.0.0. If you choose to forward logs to an external syslog endpoint, using TLS encryption is strongly recommended to prevent information from being intercepted by a third party.

Please send any feedback you have to pks-feedback@pivotal.io.

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](#)

Please send any feedback you have to pbs-feedback@pivotal.io.

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

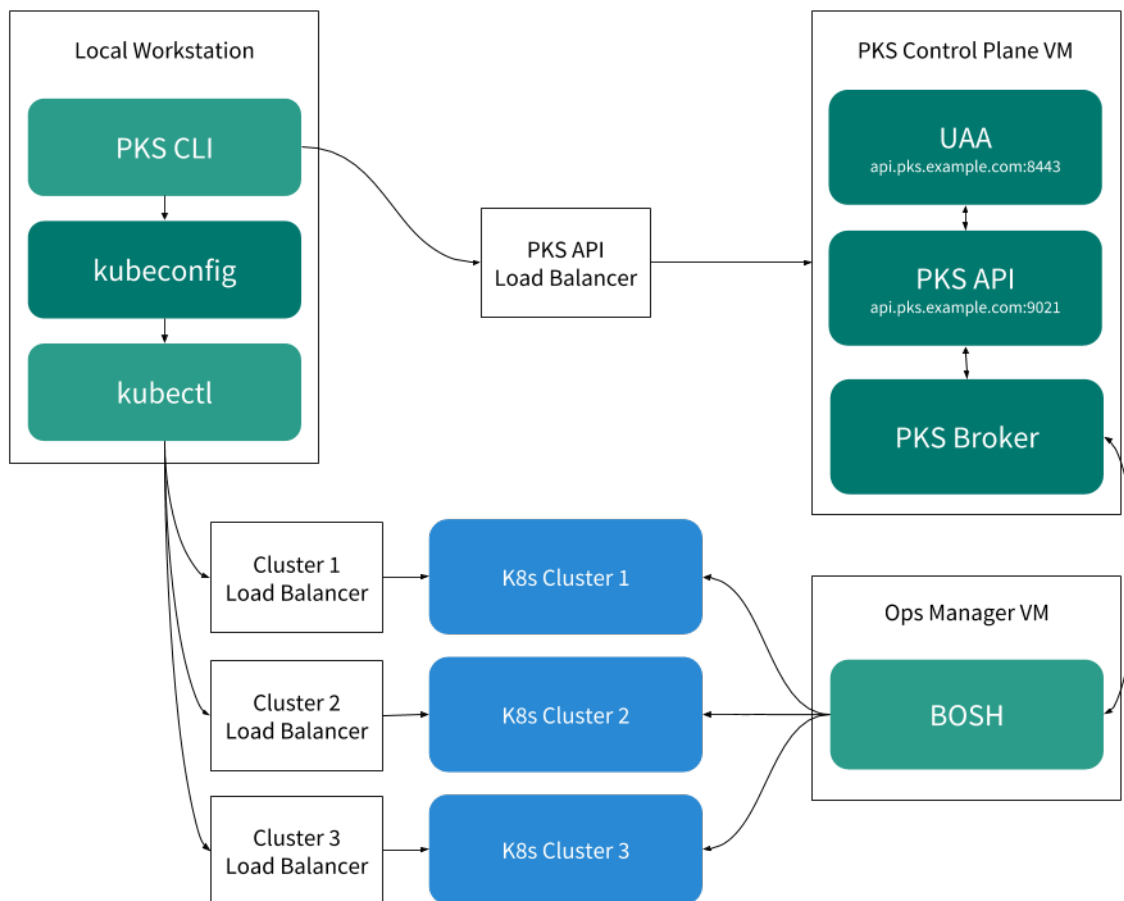
In addition, the PKS control plane can upgrade all existing clusters using the **Upgrade all clusters** BOSH errand. For more information, see [Upgrade Kubernetes Clusters](#) in *Upgrade PKS*.

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

For more information about how these components interact, see the following diagram:



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.

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.

Please send any feedback you have to pks-feedback@pivotal.io.

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, `pkc 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 `pkc 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 > UAA > UAA URL** for your PKS API domain.

When you install the PKS tile, you configure a load balancer for the PKS API. This load balancer allows you to run PKS CLI commands from your local workstation. For more information, see the *Configure External Load Balancer* section of [Installing and Configuring PKS](#).

Please send any feedback you have to pkc-feedback@pivotal.io.

Load Balancers in PKS

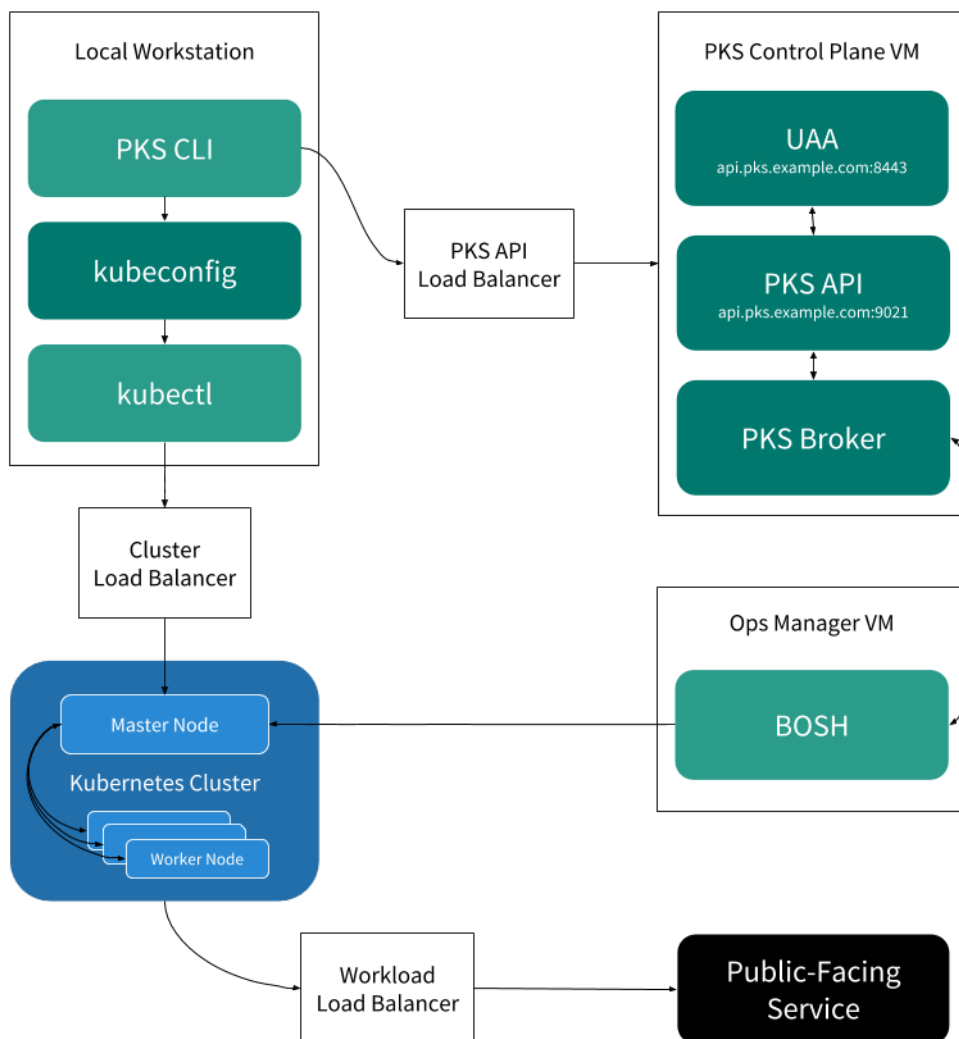
Page last updated:

This topic describes the types of load balancers that are used in Pivotal Container Service (PKS).

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:



If you use either vSphere with NSX-T or GCP, you can create 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

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

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 `kubect`l 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 `kubect`l 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

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](#).

Please send any feedback you have to pks-feedback@pivotal.io.

PKS 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](#)

Please send any feedback you have to pbs-feedback@pivotal.io.

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

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 you entered in **Ops Manager > Pivotal Container Service > UAA > UAA URL**. 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](#).
- `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
```

Upon successful login, the PKS CLI generates a `creds.yml` file containing the API endpoint, CA certificate (if applicable), refresh token, and access token.

By default, `creds.yml` is saved in the `~/.pks` directory. You can use the `PKS_HOME` environment variable to override this location and use `creds.yml` from any directory.

Please send any feedback you have to pks-feedback@pivotal.io.

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`.

Please send any feedback you have to pks-feedback@pivotal.io.

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](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

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.

PKS supports air-gapped deployments on vSphere with or without NSX-T integration.

You can also configure integration with the Harbor tile, an enterprise-class registry server for container images. For more information, see the [VMware Harbor Registry](#) documentation.

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

NSX-T Integration Version Requirements

Deploying NSX-T requires the additional following component versions:

Component	Version
VMware NSX-T	2.1

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](#) [↗](#)

About Deploying PAS and PKS

The Pivotal Application Service (PAS) and PKS runtime platforms are both deployed by Ops Manager using BOSH. You can deploy both PAS and PKS using the same Ops Manager instance in a development or test environment, but we recommend that you deploy production installations of PAS and PKS to separate Ops Manager instances. For increased security, we recommend deploying each Ops Manager instance using a unique cloud provider account.

Separate installations of Ops Manager allow you to customize and troubleshoot runtime tiles independently. You may choose to configure Ops Manager with different settings for your PAS and PKS deployments. For example, PKS and many PAS features depend on BOSH DNS.

If you deploy PAS to a separate Ops Manager instance, you can disable BOSH DNS for troubleshooting purposes. PAS can run without BOSH DNS, but key features such as secure service credentials with CredHub, service discovery for container-to-container networking, and NSX-T integration do not work when BOSH DNS is disabled.

If you deploy PAS and PKS to the same Ops Manager instance, you cannot disable BOSH DNS without breaking your PKS installation along with the PAS features that depend on BOSH DNS.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 via 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.

Please send any feedback you have to pks-feedback@pivotal.io.

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 in vSphere:


- **Master Node Service Account:** You must create a service account for Kubernetes cluster master VMs.
- **BOSH/Ops Manager Service Account:** You must create a service account for BOSH and Ops Manager.

After you create the service accounts listed above, you must grant them privileges in vSphere. Pivotal recommends configuring each service account with the least permissive privileges and unique credentials.

For the master node service account, you can create a custom role in vSphere based on your storage configuration. Kubernetes master node VMs require storage permissions to create load balancers and attach persistent disks to pods. Creating a custom role allows vSphere to apply the same privileges to all Kubernetes master node VMs in your PKS installation.

When you configure the **Kubernetes Cloud Provider** pane of the PKS tile, you enter the master node service account credentials in the **vSphere Master Credentials** fields. For more information, see the [Kubernetes Cloud Provider](#) section of *Installing and Configuring PKS*.

For the BOSH/Ops Manager service account, you can apply privileges directly to the service account without creating a role. You can also apply the default [VMware Administrator System Role](#) to the service account to achieve the appropriate permission level.

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

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** permissions to the service account:

Privilege (UI)	Privilege (API)
Advanced	VirtualMachine.Configuration.Advanced
Settings	VirtualMachine.Configuration.Settings

Step 2: Grant Additional Storage Permissions

Kubernetes master node VM 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

Grant these permissions to the master node service account based on your storage configuration using one of the procedures below:

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

See [vSphere Storage for Kubernetes](#) in the VMware documentation for more information.


Storage Permissions for Service Accounts

The following tables describe the minimum permissions required by the master node service account based on your storage configuration.

Static Only Persistent Volume Provisioning

Roles	Privileges	Entities	Propagate to Children
-------	------------	----------	-----------------------

manage-k8s-node-vm	<ul style="list-style-type: none"> VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.RemoveDisk 	VM Folder	Yes
manage-k8s-volumes	Datastore.FileManagement (Low level file operations)	Datastore	No
Read-only (pre-existing default role)	<ul style="list-style-type: none"> System.Anonymous System.Read System.View 	<ul style="list-style-type: none"> vCenter Datacenter Datastore Cluster Datastore Storage Folder 	No

 **Note:** Datastore.FileManagement is only required for the role `manage-k8s-volumes` if a Persistent Volume Claim (PVC) is created to bind with a statically provisioned Persistent Volume (PV), and the reclaim policy set to delete. When the PVC is deleted, the statically provisioned PV is also deleted.

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

Roles	Privileges	Entities	Propagate to Children
manage-k8s-node-vm	<ul style="list-style-type: none"> Resource.AssignVMToPool VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Inventory.Create VirtualMachine.Inventory.Delete 	<ul style="list-style-type: none"> Cluster Hosts VM Folder 	Yes
manage-k8s-volumes	<ul style="list-style-type: none"> Datastore.AllocateSpace Datastore.FileManagement (Low level file operations) 	Datastore	No
k8s-system-read-and-spbm-profile-view	StorageProfile.View (Profile-driven storage view)	vCenter	No
Read-only (pre-existing default role)	<ul style="list-style-type: none"> System.Anonymous System.Read System.View 	<ul style="list-style-type: none"> Datacenter Datastore Cluster Datastore Storage Folder 	No


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

Roles	Privileges	Entities	Propagate to Children
manage-k8s-node-vm	<ul style="list-style-type: none"> VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.RemoveDisk 	VM Folder	Yes
manage-k8s-volumes	<ul style="list-style-type: none"> Datastore.AllocateSpace Datastore.FileManagement (Low level file operations) 	Datastore	No
	<ul style="list-style-type: none"> System.Anonymous 	<ul style="list-style-type: none"> vCenter Datacenter 	

Read-only (pre-existing default role)	<ul style="list-style-type: none"> System.Read System.View 	<ul style="list-style-type: none"> Datastore Cluster Datastore Storage Folder 	No
---------------------------------------	--	---	----

Step 3: Create the BOSH/Ops Manager Service Account

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

 **Note:** The privileges listed in this section describe the minimum required permissions to deploy BOSH. You can also apply the default [VMware Administrator System Role](#) to the service account to achieve the appropriate permission level, but the default role includes more privileges than those listed below.

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](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.

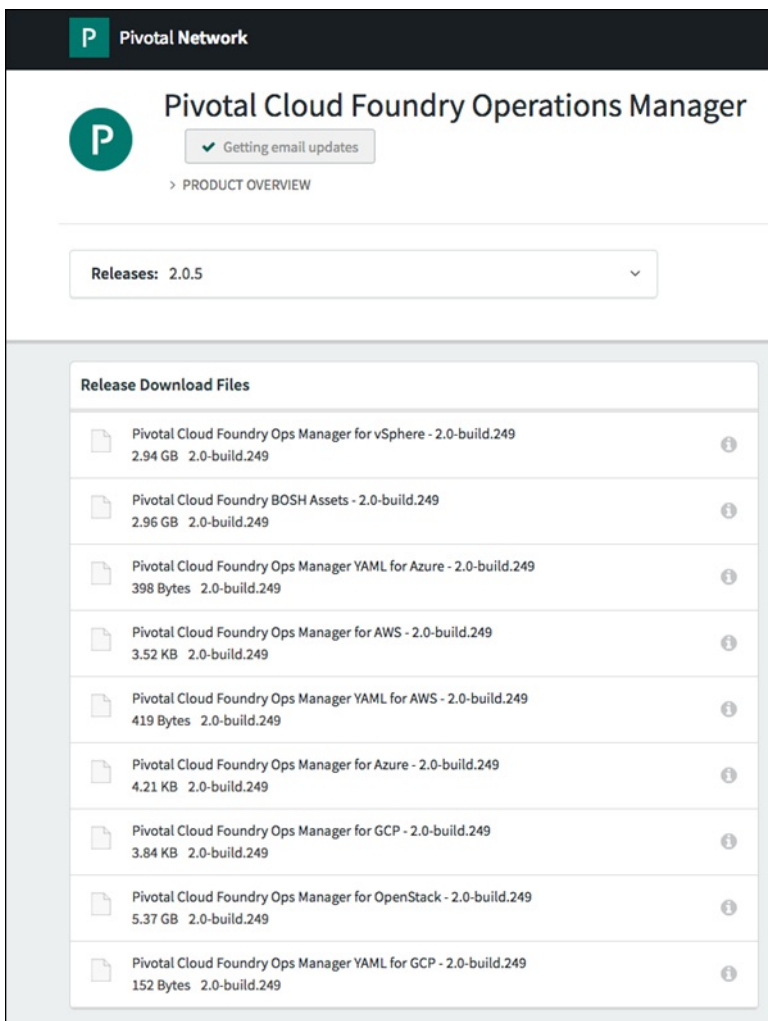
Deploying Ops Manager to vSphere

Page last updated:

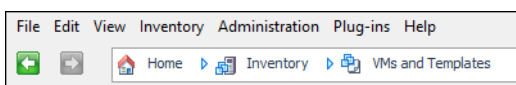
This topic provides instructions for deploying Ops Manager to VMware vSphere.

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. This issue is resolved in VMware vCenter Server 6.7. For more information about this issue, see the [VMware Knowledge Base](#).

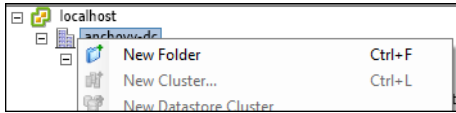
1. Before starting, refer to the known issues in the [PCF Ops Manager Release v2.0 Release Notes](#).
2. Download the [Pivotal Cloud Foundry](#) (PCF) Ops Manager `.ova` file at [Pivotal Network](#). Click the **Pivotal Cloud Foundry** region to access the PCF product page. Use the dropdown menu to select an Ops Manager release.



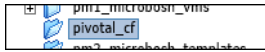
3. Log into vCenter.
4. Select the **VM and Templates** view.



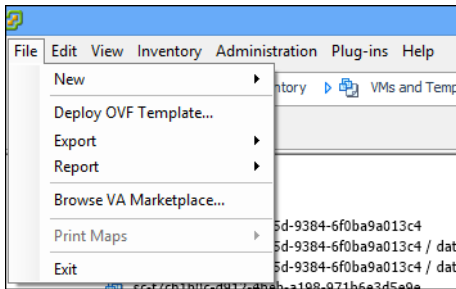
5. Right click on your datacenter and select **New Folder**.



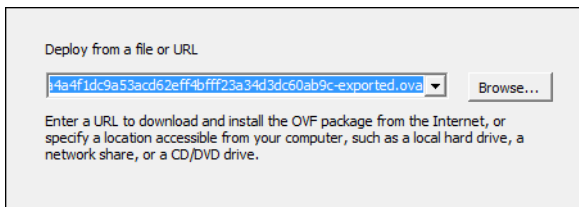
6. Name the folder `pivotal_cf` and select it.



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



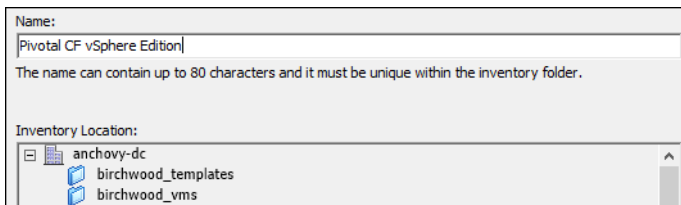
8. Select the .ova file and click **Next**.



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

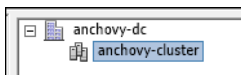
10. Accept the license agreement and click **Next**.

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, hardware virtualization must be ****off****. For more information, see [PCF on vSphere Requirements](#).

Choose a specific host within the cluster.


On clusters that are configured with vSphere HA or Manual mode vSphere DRS, each virtual machine must be assigned to a specific host, even when powered off.

Select a host from the list below:

Host Name
172.16.64.2


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

Select a destination storage for the virtual machine files:

VM Storage Profile: ▼ 

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 on vSphere](#).

 **Warning:** Ops Manager v2.0 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	▼ Mattnetwork VM Network VM Network Private

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](#). 

Next Steps

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


Please send any feedback you have to pks-feedback@pivotal.io.

Configuring Ops Manager on vSphere

Page last updated:

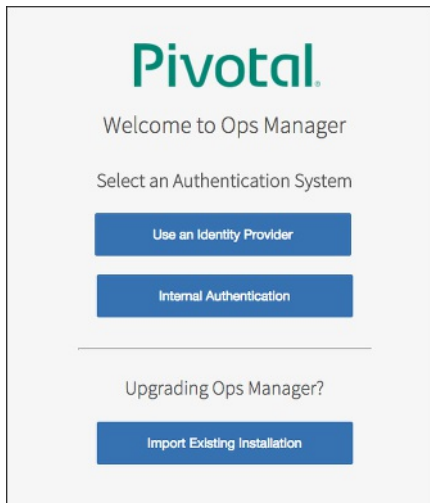
This topic describes how to configure Ops Manager for VMware vSphere.

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, 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 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.

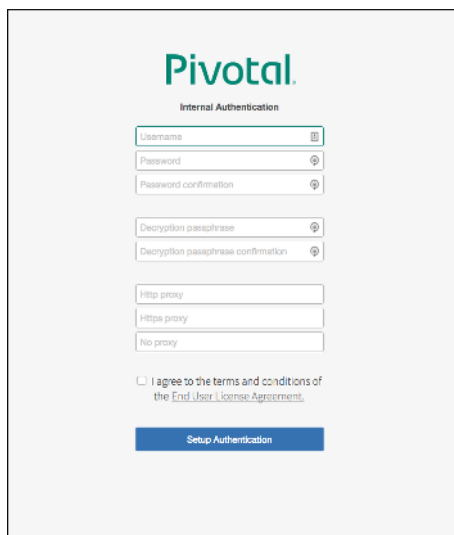
- **Single sign on URL:** `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
- **Audience URI (SP Entity ID):** `https://BOSH-IP:8443`
- **Name ID:** Email Address
- SAML authentication requests are always signed

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

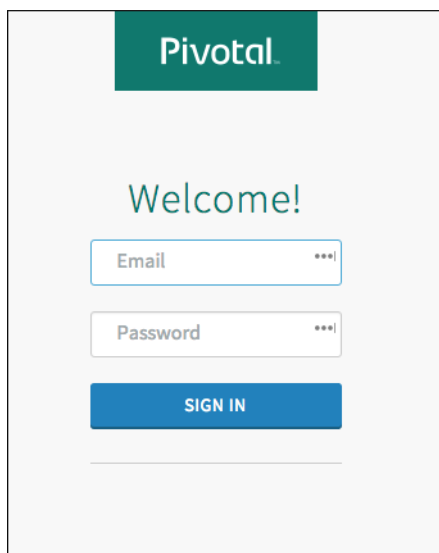
No proxy

☐ 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**.

[Installation Dashboard](#)

Ops Manager Director

Settings Status Credentials

vCenter Config

☒ vCenter Config
☒ Director Config
☒ Create Availability Zones
☒ Create Networks
☒ Assign AZs and Networks
☒ Security
☒ Syslog
☒ Resource Config

vCenter Config

vCenter Host*

vCenter Username*

vCenter Password*

[Change](#)

Datacenter Name*

Virtual Disk Type*

Ephemeral Datastore Names (comma delimited)*

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

Persistent Datastore Names (comma delimited)*

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*

Template Folder*


Disk path Folder*

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. Select **Standard vCenter Networking**. This is the default option when upgrading Ops Manager. This configuration is utilized for PAS only. You configure NSX-T integration for PKS within the PKS tile.
 6. Click **Save**.

 **Note:** After your initial deployment, you will not be able to edit the VM Folder, Template Folder, and Disk path Folder names.

Step 3: Director Config Page

1. 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


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

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

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

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

5. Select the **Enable VM Resurrector Plugin** to enable Ops Manager Resurrector functionality.
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. 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** if you want Ops Manager to retry failed BOSH operations up to five times.
 - **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. After 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*

9. 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*

10. (Optional) **Director Workers** sets the number of workers available to execute Director tasks. This field defaults to `5`.
11. (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.
12. Leave the **Director Hostname** field blank.
13. 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 \(Pivotal Knowledge Base article\)](#) and [BOSH DNS Service Discovery \(Beta\) and Opt-Out Option](#) in the *Ops Manager v2.0 Release Notes*.

14. 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

15. Click **Save**.

Note: After your initial deployment, you will not be able to 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**.

Create Availability Zones

Availability Zones
Clusters and resource pools to which you will deploy Pivotal products

▼ first-az

Name*

first-az A unique name for this availability zone

Clusters

Cluster

hinterlands-1

Resource Pool

bulldog

Save

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](#).

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:

- `pk-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 `pk-main`. See the table below for more information.
- `pk-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	

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

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 are using the Cisco Nexus 1000v Switch, see more information in [Using the Cisco Nexus 1000v Switch with Ops Manager](#).

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

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

Step 7: Security Page

1. 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

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. 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](#).

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**. 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.

- Click **Save**.

Step 9: Resource Config Page

- 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

- 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.

- Click **Save**.

Step 10: Complete the Ops Manager Installation

- Click the **Installation Dashboard** link to return to the Installation Dashboard.
- 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 with PKS](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.

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:

- **Replicate projects:** Harbor supports images replication to replicate repositories from one Harbor instance to another.
- **Manage role by LDAP group:** Harbor administrators can import an LDAP/AD group to Harbor and assign project roles to it.
- **Manage Labels:** Harbor provides labels to isolate image resources globally or at the project level.
- **Manage Helm Charts:** Harbor provides management of Helm charts isolated by projects and controlled by RBAC.
- **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 enterprise LDAP/AD systems 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 tables provide version and compatibility information for VMware Harbor Registry.

Element	Details
Tile version	v1.6.0
Release date	September 27, 2018
Software component version	v1.6.0
Compatible Ops Manager version(s)	v2.1.x & v2.2.x
Compatible Pivotal Container Service (PKS) version(s)	v1.1.x & v1.2.x
Compatible Pivotal Application Service (PAS) version(s)	v2.1.x and v2.2.x
IaaS support	vSphere, AWS, & GCP
IPsec support?	No

Element	Details
Tile version	v1.5.2
Release date	July 23, 2018
Software component version	v1.5.2

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.

Feedback

If you have a feature request, questions, or information about a bug, 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 install 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](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

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](#)

About Deploying PAS and PKS

The Pivotal Application Service (PAS) and PKS runtime platforms are both deployed by Ops Manager using BOSH. You can deploy both PAS and PKS using the same Ops Manager instance in a development or test environment, but we recommend that you deploy production installations of PAS and PKS to separate Ops Manager instances. For increased security, we recommend deploying each Ops Manager instance using a unique cloud provider account.

Separate installations of Ops Manager allow you to customize and troubleshoot runtime tiles independently. You may choose to configure Ops Manager with different settings for your PAS and PKS deployments. For example, PKS and many PAS features depend on BOSH DNS.

If you deploy PAS to a separate Ops Manager instance, you can disable BOSH DNS for troubleshooting purposes. PAS can run without BOSH DNS, but key features such as secure service credentials with CredHub, service discovery for container-to-container networking, and NSX-T integration do not work when BOSH DNS is disabled.

If you deploy PAS and PKS to the same Ops Manager instance, you cannot disable BOSH DNS without breaking your PKS installation along with the PAS features that depend on BOSH DNS.

Please send any feedback you have to pbs-feedback@pivotal.io.

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. Pivotal recommends configuring each service account with the least permissive privileges and unique credentials.

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. 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.

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. 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.

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.

 **Note:** Pivotal recommends confirming the permissions of your Master Node Service Account, Worker Node Service Account, and BOSH/Ops Manager Service Account after you create them. To verify these account permissions, run the `gcloud auth list` command. For more information, see [gcloud auth](#) in the Google Cloud documentation.

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: <code>192.168.101.0/26</code>

- 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:
 - **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.
 - **Zone:** Select the first zone from your region. Example: For region `us-west1`, select zone `us-west1-a`.
 - **Machine type:** Select `n1-standard-4`.
 - **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**.

6. Click **Create** to finish creating the NAT instance.

7. To create additional NAT instances, repeat steps 2-6 using the names and zones specified in the table below.

Instance 2	Name	<code>MY-PKS-nat-gateway-sec</code>
	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	<code>MY-PKS-nat-gateway-ter</code>
	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


1. In the GCP console, navigate to **VPC Networks > Routes**.
2. Click **CREATE ROUTE**.
3. 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`
4. Click **Create** to finish creating the route.
5. 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.

1. From the GCP console, navigate to **VPC network > Firewall rules**.
2. 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: MY-PKS-allow-ssh</p> <p>Network: your-pks-virt-net</p> <p>Allowed protocols and ports: tcp:22</p> <p>Source filter: IP ranges</p> <p>Source IP ranges: 0.0.0.0/0</p> <p>Target tags: allow-ssh</p>
Rule 2	<p>This rule allows HTTP from public networks.</p> <p>Name: MY-PKS-allow-http</p> <p>Network: your-pks-virt-net</p> <p>Allowed protocols and ports: tcp:80</p> <p>Source filter: IP ranges</p> <p>Source IP ranges: 0.0.0.0/0</p> <p>Target tags: allow-http, router</p>
Rule 3	<p>This rule allows HTTPS from public networks.</p> <p>Name: MY-PKS-allow-https</p> <p>Network: your-pks-virt-net</p> <p>Allowed protocols and ports: tcp:443</p> <p>Source filter: IP ranges</p> <p>Source IP ranges: 0.0.0.0/0</p> <p>Target tags: allow-https, router</p>
Rule 4	<p>This rule allows communication between BOSH-deployed jobs.</p> <p>Name: MY-PKS-allow-pks-all</p> <p>Network: your-pks-virt-net</p> <p>Allowed protocols and ports: tcp;udp;icmp</p> <p>Source filter: Source tags</p> <p>Target tags: MY-PKS, MY-PKS-opsman, nat-traverse</p> <p>Source tags: MY-PKS, MY-PKS-opsman, nat-traverse</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.
- Click **Create** to deploy the new Ops Manager VM. This may take a few moments.
 - 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](#).

Please send any feedback you have to pks-feedback@pivotal.io.

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).

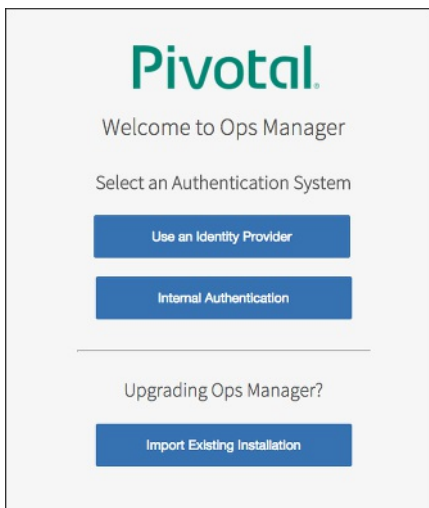
 **Note:** You can also perform the procedures in this topic using the Ops Manager API. For more information, see [Using the Ops Manager API](#).

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.

Pivotal

Use an Identity Provider

https://dev-55555.oktapreview.com/app/e77777/sso/saml/metadata

Bosh IDP Metadata (Full URL or XML) - will use same metadata as above if left blank.

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:

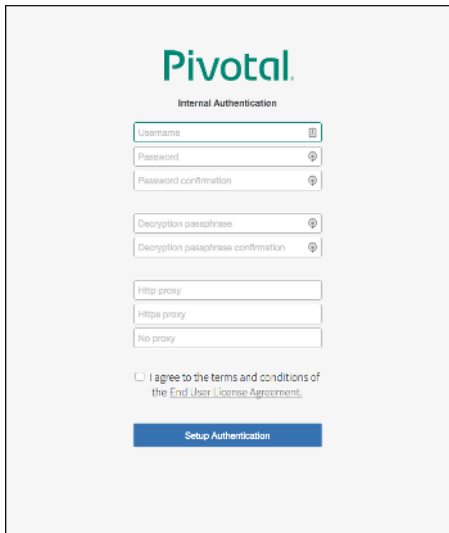
- **5a.** Ops Manager SAML service provider metadata: `https://OPS-MAN-FQDN:443/uaa/saml/metadata`
- **5b.** BOSH Director SAML service provider metadata: `https://BOSH-IP-ADDRESS:8443/saml/metadata`

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

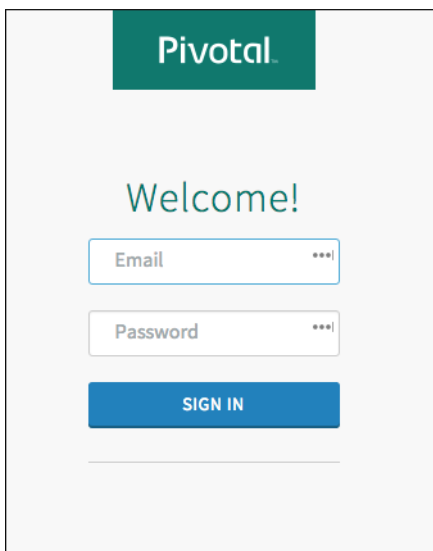
Https proxy

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

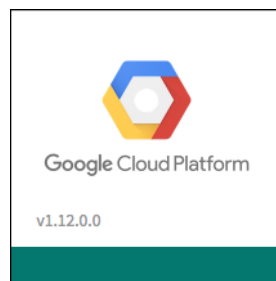
Welcome!

Email


Password

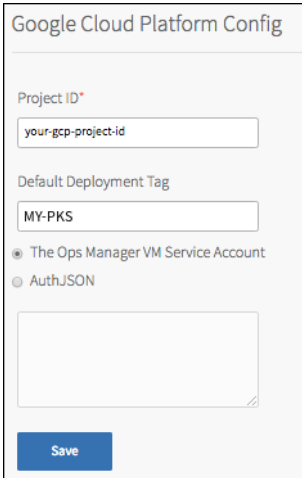
SIGN IN

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*.

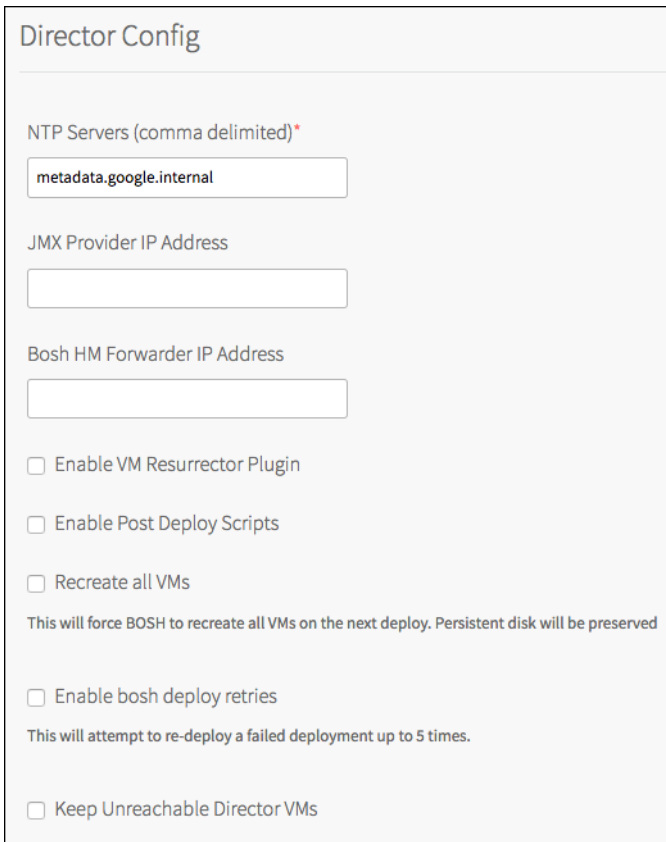
 **Note:** As an alternative, you can select **The Ops Manager VM Service Account** option to use the service account automatically created by GCP for the Ops Manager VM.



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*

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*

Port*

Domain*

From*

Recipients*

Username


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 [Creating 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 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**.

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:
 - `pkc-infrastructure` for Ops Manager, the BOSH Director, and NAT instances.
 - `pkc-main` for the PKS control plane. In non-production environments, you can choose to combine `pkc-infrastructure` and `pkc-main` into a single network.
 - `pkc-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	<code>pks-infrastructure</code>
	Service Network	Leave Service Network unchecked.
	Google Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-infrastructure-GCP-REGION/GCP-REGION</code>
	CIDR	<code>192.168.101.0/26</code>
	Reserved IP Ranges	<code>192.168.101.1-192.168.101.9</code>
	DNS	<code>169.254.169.254</code>
	Gateway	<code>192.168.101.1</code>
Main Network	Field	Configuration
	Name	<code>pks-main</code>
	Service Network	Leave Service Network unchecked.
	Google Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-pks-GCP-REGION/GCP-REGION</code>
	CIDR	<code>192.168.16.0/26</code>
	Reserved IP Ranges	<code>192.168.16.1-192.168.16.9</code>
	DNS	<code>169.254.169.254</code>
	Gateway	<code>192.168.16.1</code>
Service Network	Field	Configuration
	Name	<code>pks-services</code>
	Service Network	Select the Service Network checkbox.
	Google Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-services-GCP-REGION/GCP-REGION</code>
	CIDR	<code>192.168.20.0/22</code>
	Reserved IP Ranges	<code>192.168.20.1-192.168.20.9</code>
	DNS	<code>169.254.169.254</code>
	Gateway	<code>192.168.20.1</code>

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]

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.
 - You do not need to enter anything in this field if you are using self-signed certificates.
 - 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**.

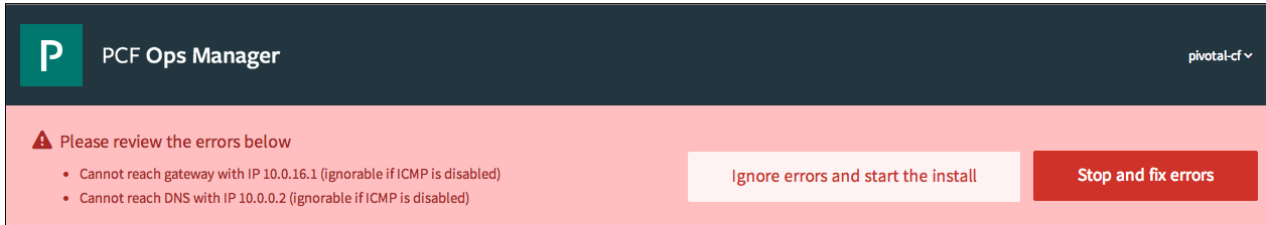
Step 9: Resource Config Page

- 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

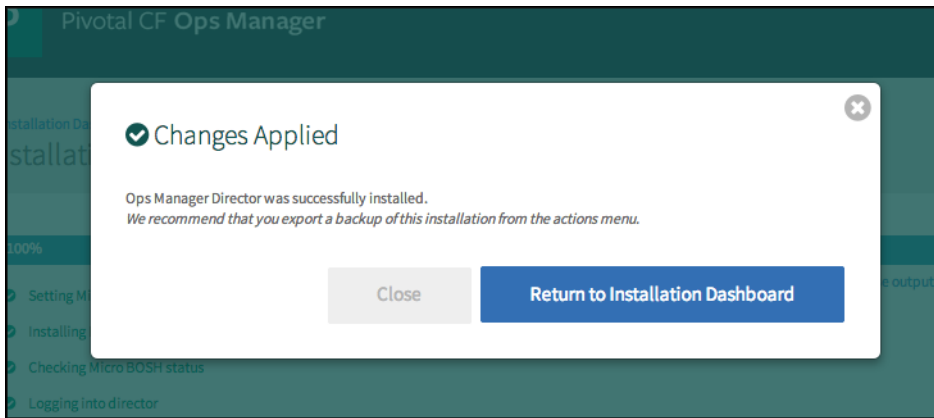
your VMs. For more information about using NAT in GCP, see [Virtual Private Cloud \(VPC\) Network Overview](#) in the GCP documentation.

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](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.


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 `pks-api`.
7. Select **Backend configuration**.
 - Under **Region**, select the region where you deployed Ops Manager.
 - Under **Backends**, leave **Select existing instance groups** selected. You point the PKS API instance to the backend later in this procedure.
 - (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, `pks-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 **Specified target tags**.
- Under **Target tags**, enter `pkc-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: Install PKS

Follow the instructions in [Installing and Configuring PKS](#) to deploy PKS. After you finish installing PKS, continue to the following sections to complete the PKS API load balancer configuration.

Step 4: 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 `pkc-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 `pkc-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 5: 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 `pkc-api.pkc.example.com` as your PKS API hostname, enter `pkc-api` in this field.
5. Under **Resource Record Type**, select **A** to create a DNS address record.
6. Enter a value for **TTL** and select a **TTL Unit**.
7. Enter the static IP address that GCP assigned when you created the load balancer in [Step 1: Create a Load Balancer](#).
8. Click **Create**.

Next Steps

Follow the procedures in [Configure PKS API Access](#).

Configure authentication for PKS using User Account and Authentication (UAA). To create and manage users using UAA, see [Manage Users in UAA](#).

Please send any feedback you have to pbs-feedback@pivotal.io.

Configuring a GCP Load Balancer for PKS Clusters

Page last updated:

This topic describes how to configure a Google Cloud Platform (GCP) load balancer for a Kubernetes cluster deployed by Pivotal Container Service (PKS).

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

You can configure GCP load balancers only for PKS clusters that are deployed on GCP.

Prerequisites

- To complete these procedures, you must have already configured a load balancer to access the PKS API. For more information, see [Creating a GCP Load Balancer for the PKS API](#).
- The version of the PKS CLI you are using must match the version of the PKS tile you are installing.

Configure GCP Load Balancer

Follow the procedures in this section to create and configure a load balancer for PKS-deployed Kubernetes clusters using GCP. Modify the example commands in these procedures to match your PKS installation.

Step 1: Create a GCP Load Balancer

Perform the following steps to create a GCP load balancer for your PKS clusters:

1. Navigate to the [Google Cloud Platform console](#).
2. In the sidebar menu, select **Network Services > Load balancing**.
3. Click **Create a Load Balancer**.
4. In the **TCP Load Balancing** pane, click **Start configuration**.
5. Click **Continue**. The **New TCP load balancer** menu opens.
6. Give the load balancer a name. For example, `my-cluster`.
7. Click **Frontend configuration** and configure the following settings:
 - a. Click **IP**.
 - b. Select **Create IP address**.
 - c. Give the IP address a name. For example, `my-cluster-ip`.
 - d. Click **Reserve**. GCP assigns an IP address.
 - e. In the **Port** field, enter `8443`.
 - f. Click **Done** to complete frontend configuration.
8. Review your load balancer configuration and click **Create**.


Step 2: Create the Cluster

Follow the procedures in the [Create a Kubernetes Cluster](#) section of *Creating Clusters*. Use the GCP-assigned IP address from the previous step as the external hostname when you run the `pkcs create-cluster` command.

Step 3: Configure Load Balancer Backend

Perform the following steps to configure the backend of the load balancer:

1. Navigate to the [Google Cloud Platform console](#).
2. In the sidebar menu, select **Network Services** > **Load balancing**.
3. Select the load balancer you created for the cluster and select **Configure**.
4. Click **Backend configuration** and configure the following settings:
 - a. 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](#) in *Creating Clusters*.

 **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.

- b. Specify any other configuration options you require and click **Update** to complete backend configuration.

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

Step 4: Access the Cluster

Perform the following steps to complete cluster configuration:

1. From your local workstation, 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 [Retrieving Cluster Credentials and Configuration](#).
2. 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.

Step 5: Create a Network Tag

Perform the following steps to create 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**.

Step 6: Create Firewall Rules

Perform the following steps to create firewall rules:

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 [Step 1: Create a GCP Load Balancer](#).
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 [Step 5: Create 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 [Step 1: Create a GCP Load Balancer](#), 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**.

Reconfigure 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**.

Please send any feedback you have to pks-feedback@pivotal.io.

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](#)

Please send any feedback you have to pbs-feedback@pivotal.io.

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**.

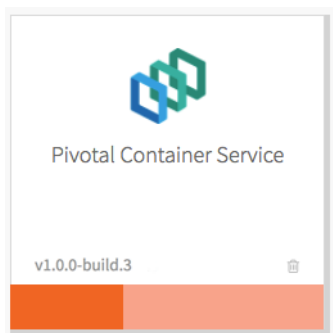
Step 1: Install PKS

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

pkc-infrastructure

Service Network

pkc-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 certificate and private key pair. The certificate you enter here should cover the domain that routes to the PKS API 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. Click **Save**.

Plans

To activate a plan, do the following:

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. Configuring **Plan 1** is required.

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

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 **AZ placement**, select an AZ for the Kubernetes clusters deployed by PKS.
6. Under **Default Cluster Authorization Mode**, select an authentication mode for the Kubernetes clusters. Pivotal recommends selecting **RBAC**. For more information, see [Authorization Overview](#) in the Kubernetes documentation.
7. Under **ETCD/Master VM Type**, select the type of VM to use for Kubernetes etcd and master nodes.
8. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master VM.
9. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker nodes.
10. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker nodes.
11. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, Pivotal recommends creating clusters with at least 3 worker nodes.
12. Under **Errand VM Type**, select the size of the VM where the errand will run. 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.
13. (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.
14. If you want users to be able to create pods with privileged containers, select the **Enable Privileged Containers - Use with caution** option. For more information, see [Pods](#) in the Kubernetes documentation.
15. Click **Save**.

To deactivate a plan, do the following:

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 will use credentials for vCenter master and worker VMs. You must have provisioned the service account associated with each type of VM with the correct permissions. For more information, see [Create the Master Node Service Account](#) and [Create the Worker Node Service Account](#).

Perform the following steps:

1. Click **Kubernetes Cloud Provider**.
2. Under **Choose your IaaS**, select vSphere.
3. Perform the steps specific to vSphere. Ensure the values match those in the **vCenter Config** section of the **Ops Manager** tile:
 - a. Enter your **vCenter Master Credentials**. Enter the username using the format `user@CF-EXAMPLE.com`.
 - b. Enter your **vCenter Worker Credentials**. Enter the username using the format `user@CF-EXAMPLE.com`.
 - c. Enter your **vCenter Host**. For example, `vcenter.CF-EXAMPLE.com`.

- d. Enter your **Datacenter Name**. For example, `CF-EXAMPLE-dc`.
- e. Enter your **Datastore Name**. For example, `CF-EXAMPLE-ds`.
- f. Enter the **Stored VM Folders** 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`.

4. Click **Save**.

GCP

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

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 Key**. For information about configuring this key, see [Create the Master Node Service Account](#).
4. Enter your **GCP Worker Service Account Key**. For information about configuring this key, see [Create the Worker Node Service Account](#).
5. Click **Save**.

Networking

To configure networking, do the following:

1. Click **Networking**.
2. Under **Network**, select the Container Network Interface to use.
 - For **Flannel**, no additional fields are required.
 - For **NSX-T**, see [Installing and Configuring PKS with NSX-T Integration](#).
3. Click **Save**.

UAA

To configure the UAA server, do the following:

1. Click **UAA**.
2. Under **UAA URL**, enter a fully qualified domain name (FQDN) to access UAA on the PKS broker VM. This URL must belong to the domain you provided in the [PKS API](#) section. For example, if you provided a certificate for `*.pks.example.com`, enter `api.pks.example.com` for the UAA URL.
3. Under **PKS CLI Access Token Lifetime**, enter a time in seconds for the PKS CLI access token lifetime.
4. Under **PKS CLI Refresh Token Lifetime**, enter a time in seconds for the PKS CLI refresh token lifetime.
5. Click **Save**.


(Optional) Syslog

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 **Syslog**.
2. Select **Yes** to configure syslog forwarding.
3. Enter the destination syslog endpoint.
4. 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. Provide the accepted fingerprint (SHA1) or name of remote peer. For example, `*.YOUR-LOGGING-SYSTEM.com`.
 - b. 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.

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, Pivotal recommends that you leave the default settings.

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


Resource Config

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

 **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`. For more information, see [Configuring a GCP Load Balancer for the PKS API](#).

(Optional) Stemcell

 **Note:** The **Stemcell** pane appears in Ops Manager v2.0 and earlier only. In Ops Manager v2.1 and later, manage stemcells using the **Stemcell Library**. For more information, see [Importing and Managing Stemcells](#) in the Pivotal Cloud Foundry documentation.

To edit the stemcell configuration, click **Stemcell**. Click **Import Stemcell** to import a new stemcell.

PKS uses floating stemcells. Floating stemcells allow upgrades to the minor versions of stemcells but not the major versions. For example, a stemcell can float from `1234.56` to `1234.99` but not from `1234.991` to `1235.0`. For more information on floating stemcells, see [Understanding Floating Stemcells](#).

 **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**. Pivotal recommends 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.

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

If you are using GCP, continue to [Step 4: Create a Network Tag for the Firewall Rule](#) of *Configuring a GCP Load Balancer for the PKS API*.

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 9021 and 8443. 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

Follow the procedures in [Configure PKS API Access](#).

Configure authentication for PKS using User Account and Authentication (UAA). To create and manage users using UAA, see [Manage Users in UAA](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.

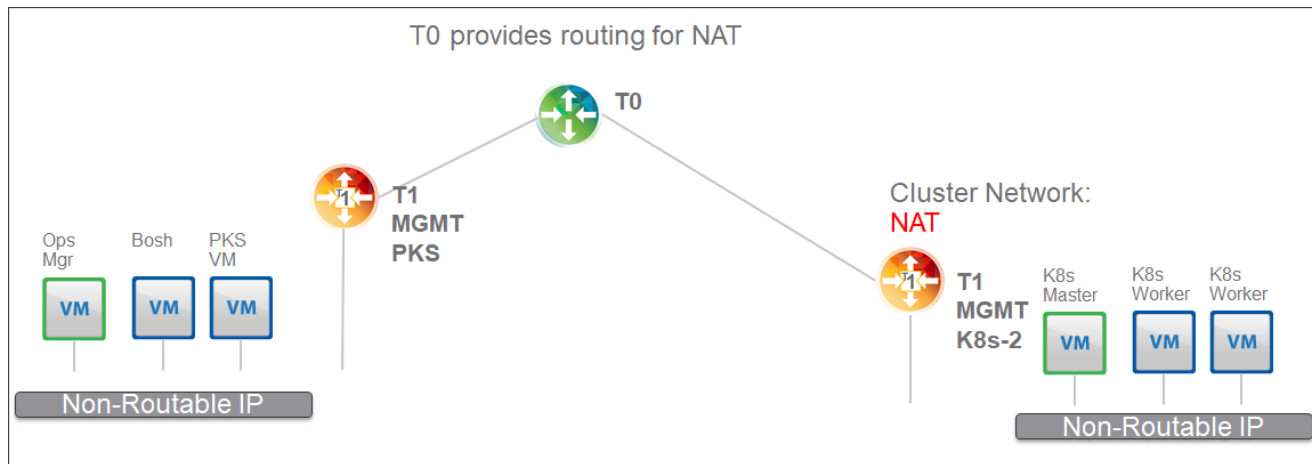
Installing and Configuring PKS with NSX-T Integration

Page last updated:

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

Deployment Architecture

The instructions in this topic deploy NSX-T with PKS using the Network Address Translation (NAT) topology. The following figure shows the NAT deployment architecture:



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

This topology has the following characteristics:

- The BOSH Director, Ops Manager, and the PKS service instance are all located on a logical switch NAT'd behind a T1 logical router.
- All Kubernetes cluster nodes are located on a logical switch NAT'd behind a T1 logical router. This will require NAT rules to allow access to Kubernetes APIs.

Before You Install

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 [NSX-T documentation](#) to verify your setup before proceeding to the next step.
- Comply with any requirements or best practices described in the [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/hashed name.

Step 1: Pre-allocate Network Subnets

Determine and pre-allocate the following network CIDRs in the IPv4 address space according to the instructions in the [NSX-T documentation](#). Ensure that the CIDRs are routable in your environment.

- **VTEP CIDR(s):** One or more of these networks will host your GENEVE Tunnel Endpoints on your NSX Transport Nodes. Size the network(s) to support all of your expected Host and Edge Transport Nodes. For example, a CIDR of `192.168.1.0/24` will provide 254 usable IPs. This will be used when creating the `ip-pool-vteps` in Step 3.
- **PKS MANAGEMENT CIDR:** This small network will be used for NAT access to PKS management components such as Ops Manager and the PKS Service

VM. For example, a CIDR of `10.172.1.0/28` will provide 14 usable IPs.


- **PKS LB CIDR:** This network provides 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, `10.172.2.0/25` provides 126 usable IPs. This network is used when creating the `ip-pool-vips` described in [3.1: Create NSX Network Objects](#).

Refer to the instructions in the [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 will deploy a NCP pod that must be able to communicate with the NSX Manager.

Step 2: Deploy NSX-T

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

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


1. Deploy the NSX Manager. For more information, see [NSX Manager Installation](#).
2. Deploy NSX Controllers. For more information, see [NSX Controller Installation and Clustering](#).
3. Join the NSX Controllers to the NSX Manager. For more information, see [Join NSX Controllers with the NSX Manager](#).
4. Initialize the Control Cluster. For more information, see [Initialize the Control Cluster to Create a Control Cluster Master](#).
5. Add your ESXi host(s) to the NSX-T Fabric. For more information, see [Add a Hypervisor Host to the NSX-T Fabric](#). 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. Pivotal recommends at least two VMs. For more information, see [NSX Edge Installation](#). 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 will be located. Your **PKS MANAGEMENT CIDR** and **PKS LB CIDR** should be routable to this PortGroup.
 - d. Join the NSX Edge VMs to the NSX-T Fabric. For more information, see [Join NSX Edge with the Management Plane](#).

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 [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](#). Configuration details for the NSX IP pools follow:
 - 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
 - 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](#). Configuration details for the NSX TZs follow:
 - One NSX TZ for PKS control plane Services and Kubernetes Cluster deployment overlay network(s) called `tz-overlay` and the associated N-VDS `hs-overlay`. Select **Standard**.

- One NSX TZ for NSX Edge uplinks (ingress/egress) for PKS Kubernetes cluster(s) called `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](#).
 - 4. Create NSX Host Transport Nodes. For more information, see [Create a Host Transport Node](#). Configuration details follow:
 - 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 will allow them to route and communicate with each other, as well as other Edge Transport Nodes, to build GENEVE tunnels.
- 5. Create an NSX IP Block named `ip-block-pks-deployments` (for more information, see [Manage IP Blocks](#)). The NSX-T Container Plug-in (NCP) and PKS will use this IP Block to assign address space to Kubernetes pods through the Container Networking Interface (CNI). Pivotal recommends using the CIDR block `172.16.0.0/16`.

3.2: Create Logical Switches

1. Create the following NSX-T Logical Switches. For more information, see [Create a Logical Switch](#). Configuration details for the Logical Switches follow:
 - One for T0 ingress/egress uplink port `ls-pks-uplink`
 - One for the PKS Management Network `ls-pks-mgmt`
 - One for the PKS Service Network `ls-pks-service`
2. Attach your first NSX-T Logical Switch to the `tz-vlan` NSX Transport Zone.
3. Attach your second and third NSX-T Logical Switches to the `tz-overlay` NSX Transport Zone.

3.3: Create NSX Edge Objects

1. Create NSX Edge Transport Node(s). For more information, see [Create an NSX Edge Transport Node](#).
2. Add both `tz-vlan` and `tz-overlay` NSX Transport Zones to the NSX Edge Transport Node(s). 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 Edge(s):
 - 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 called `edge-cluster-pks`. For more information, see [Create an NSX Edge Cluster](#).
5. Add the NSX Edge Transport Node(s) to the cluster.

3.4: Create Logical Routers

Create T0 Logical Router for PKS

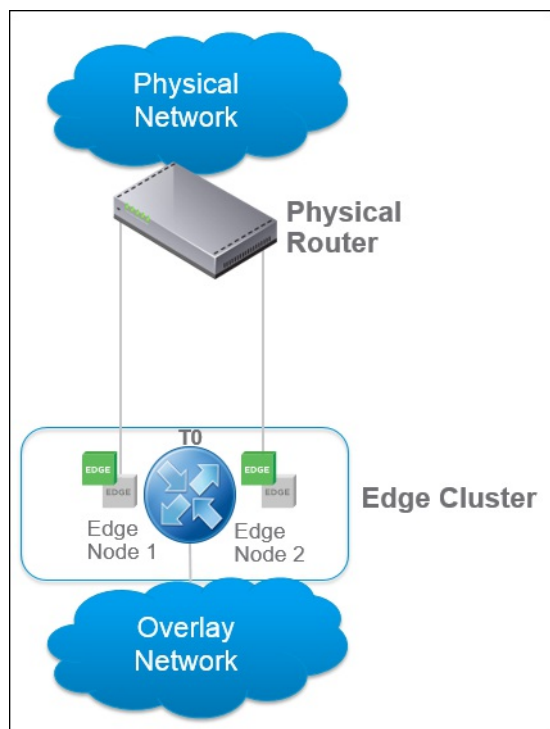
1. Create a Tier-0 (T0) logical router named `t0-pks`. See [Create a Tier-0 Logical Router](#) for more information. Configuration details follow:
 - Select `edge-cluster-pks` for the cluster.
 - Set **High Availability Mode** to **Active-Standby**. NAT rules will be applied on T0 by NCP. If not set **Active-Standby**, the router will 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](#). Create a logical router port for `ls-pks-uplink` and assign an IP address and CIDR that your environment will use 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](#). 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 will become 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](#). Configuration details follow:
 - Link to the `t0-pks` logical router you created in a previous step.
 - Select `edge-cluster-pks` for the cluster.
2. Create a logical router port for `ls-pks-mgmt` and assign the following CIDR block: `172.31.0.1/24`. For more information, see [Connect a Tier-0 Logical Router to a VLAN Logical Switch](#).
3. Configure route advertisement on the T1 as follows. For more information, see [Configure Route Advertisement on a Tier-1 Logical Router](#). Configuration details follow:
 - 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

Create the following NAT rules for the Mgmt T1. For more information, see [Tier-1 NAT](#). Configuration details follow:

Type	For
NO_NAT	Mgmt Net <-> Service Net
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 [DNAT](#) rule on the T1 maps an external IP from the **PKS MANAGEMENT CIDR** to the IP where you will 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 will create another DNAT rule to map an external IP from the **PKS MANAGEMENT CIDR** to the PKS endpoint.

The [SNAT](#) rule on the T1 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 from your **PKS MANAGEMENT CIDR**.

Note: Ops Manager and BOSH need to 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 subnet(s) that contain your vCenter and NSX Manager IP addresses.

Create T1 Logical Router for PKS Service VMs

1. Create a Tier-1 (T1) logical router for PKS Service VMs `t1-pks-service`. For more information, see [Create a Tier-1 Logical Router](#). Configuration details follow:
 - Link to the `t0-pks` logical switch you created in a previous step.
 - Select `edge-cluster-pks` for the cluster.
2. Create a logical router port for `ls-pks-service` and assign the following CIDR block: `172.31.2.1/23`. For more information, see [Connect a Tier-0 Logical Router to a VLAN Logical Switch](#).
3. Configure route advertisement on the T1 as follows. For more information, see [Configure Route Advertisement on a Tier-1 Logical Router](#). Configuration details follow:
 - Enable **Advertise All NSX Connected Routes**.
 - Enable **Advertise All NAT Routes**.

- Enable **Advertise All LB VIP Routes**.

Configure NAT Rules for PKS Service VMs

Create the following NAT rules for the Service T1. For more information, see [Tier-1 NAT](#). Configuration details follow:

Type	For
NO_NAT	Mgmt Net <-> Service Net
SNAT	K8s Workers -> External Registries (for example, DockerHub)
SNAT	K8s Workers -> DNS
SNAT	K8s Workers -> NTP
SNAT	K8s Workers -> NSX-T Manager (NCP)
SNAT	K8s Workers -> vCenter (vSphere Cloud Provider)
SNAT	K8s Workers -> External Service Endpoints for Workloads

The [SNAT](#) rule allows the Kubernetes Cluster VMs to communicate with your environment's NSX Manager and allows the NCP pod on each cluster to communicate with your NSX Manager. For example, a SNAT rule that maps `172.31.2.0/23` to `10.172.1.3`, where `10.172.1.3` is a routable IP from your **PKS MANAGEMENT CIDR**.

Note: Limit the Destination CIDR for the SNAT rules to the subnet(s) 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-T logical switches:

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

Note: If you have Pivotal Application Service (PAS) installed, Pivotal recommends installing PKS on a separate instance of Ops Manager v2.0.

- On the **vCenter Config** page, select **Standard vCenter Networking**. This configuration is utilized for PAS only. You configure NSX-T integration for PKS in a later step.

Note: Using this NAT topology, you must have already deployed Ops Manager to the `ls-pks-mgmt` NSX-T 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** page, create the following networks:

Infrastructure Network	Field	Configuration
	Name	<code>pks-infrastructure</code>
	Service Network	Leave Service Network unchecked.
	vSphere Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-infrastructure</code>
	Description	A network for deploying the PKS control plane VM(s) that maps to the NSX-T 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 .
	Field	Configuration
	Name	<code>pks-services</code>
	Service	Select the Service Network checkbox.

Service Network	Network vSphere Network Name	MY-PKS-virt-net/MY-PKS-subnet-services
	Description	A service network for deploying PKS Kubernetes cluster nodes that maps to the NSX-T logical switch named <code>ls-pks-service</code> created for the PKS Service Network in Step 3: Create the NSX-T Objects Required for PKS .

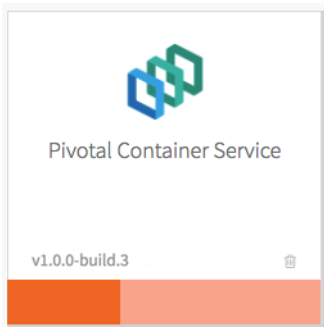
- Return to the Ops Manager Installation Dashboard and click **Apply Changes**.

Step 6: Install and Configure PKS

Perform the following steps to install and configure PKS:

- Install the PKS tile. For more information, see [Install PKS](#).
- 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:

- Click **Assign AZs and Networks**.
- 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.

- Under **Network**, select the PKS Management Network linked to the `ls-pks-mgmt` NSX-T logical switch you created in [Step 5: Configure Ops Manager](#). This will provide network placement for the PKS API VM.
- Under **Service Network**, select the PKS Service Network linked to the `ls-pks-service` NSX-T 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.
- Click **Save**.

PKS API

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

Plans

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

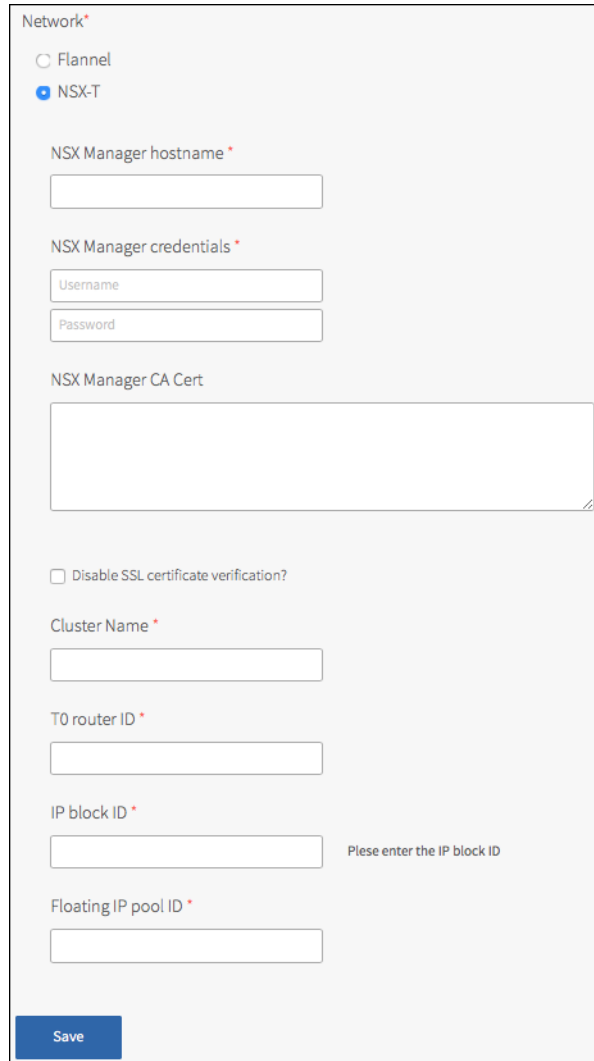
Kubernetes Cloud Provider

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

Networking

Perform the following steps:

1. Click **Networking**.



The screenshot shows a 'Network' configuration form. At the top, there are two radio buttons: 'Flannel' (unselected) and 'NSX-T' (selected). Below this, there are several input fields: 'NSX Manager hostname' (required), 'NSX Manager credentials' (with sub-fields for 'Username' and 'Password'), and 'NSX Manager CA Cert' (a large text area). There is a checkbox for 'Disable SSL certificate verification?'. Below that are 'Cluster Name', 'T0 router ID', 'IP block ID', and 'Floating IP pool ID', all marked as required. A note next to the 'IP block ID' field says 'Please enter the IP block ID'. At the bottom right is a blue 'Save' button.

2. Under **Network**, select **NSX-T** as the **Container Network Type** to use.
3. For **NSX Manager hostname**, enter the NSX Manager hostname or IP address.
4. For **NSX Manager credentials**, enter the credentials to connect to the NSX Manager.
5. For **NSX Manager CA Cert**, optionally enter the custom CA certificate to be used to connect to the NSX Manager.
6. 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.
7. For **vSphere Cluster Name**, enter the name of the vSphere cluster that corresponds to the AZ where you deployed the PKS control plane VM.
8. For **T0 Router ID**, enter the `t0-pks` T0 router UUID. This can be located in the NSX-T UI router overview.
9. For **IP Block ID**, enter the `ip-block-pks-deployments` IP block UUID. This can also be located in the NSX-T UI.
10. For **Floating IP pool ID**, enter the `ip-pool-vips` Floating IP pool ID that was created for load balancer VIPs.

11. Click **Save**.

UAA

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

Syslog

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

Errands

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**.
2. For **Post Deploy Errands**, select **ON** for the **NSX-T Validation errand**. This errand will validate your NSX-T configuration and will tag 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 7: Apply Changes and Retrieve the PKS Endpoint

1. After configuring the tile, return to the Ops Manager Installation Dashboard and click **Apply Changes** to deploy the PKS tile.
2. 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.
3. Create a DNAT rule on the `t1-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 `ls-pks-mgmt` NSX-T Logical Switch. For more information, see [Configure Destination NAT on a Tier-1 Router](#).

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](#).

WARNING: The PKS CLI is under active development and commands may change. To ensure you have installed the latest version, we recommend that you re-install the PKS CLI before you use it. For more information, see [Installing the PKS CLI](#).

Step 8: Deploy a Cluster and Enable NAT Access

In the current version of PKS, NSX-T does not automatically configure a NAT for the master node of each Kubernetes cluster. As a result, you must perform the following procedure for each cluster to enable your developers to use kubectl:

1. Download the NSX scripts:

```
$ wget https://storage.googleapis.com/pks-releases/nsx-helper-pkg.tar.gz
```

2. Untar the `nsx-helper-pkg.tar.gz` file:

```
$ tar -xvzf nsx-helper-pkg.tar.gz
```

3. Install required packages:

```
$ sudo apt-get install git
$ sudo apt-get install -y httpie
$ sudo apt-get install jq
```

4. One of the files from the tarball is `nsx-cli.sh`. Make the script executable:

```
$ chmod 755 nsx-cli.sh
```

5. Set your NSX Manager admin user, password, and IP address as environment variables named `NSX_MANAGER_USERNAME`, `NSX_MANAGER_PASSWORD`, and `NSX_MANAGER_IP`. For example:

```
$ export NSX_MANAGER_USERNAME="admin-user"
$ export NSX_MANAGER_PASSWORD="admin-password"
$ export NSX_MANAGER_IP="192.0.2.1"
```

6. Execute the `nsx-cli` script with the following command:

```
$ ./nsx-cli.sh ipam allocate
```

Developers can use this IP address as the `--external-hostname` value to create a cluster via the PKS CLI. For more information, see [Using PKS](#).

7. Collect the Cluster UUID after cluster has been successfully created.

```
$ pks clusters
```

8. Use the `nsx-cli` script to create a NAT rule to allow access to the Kubernetes API for the cluster. Execute the following command:

```
$ ./nsx-cli.sh nat create-rule CLUSTER-UUID MASTER-IP NAT-IP
```

Where:

- `CLUSTER-UUID` is the ID of the cluster retrieved in the previous step.
- `MASTER-IP` is the IP address that BOSH has assigned to the master node of the cluster. To retrieve this value, use BOSH CLI v2+ to log in to your BOSH Director and list all instances with `bosh -e YOUR-ENV instances`. For more information, see [Commands](#) in the BOSH documentation.
- `NAT-IP` is the NAT IP from the `ip-pool-vips` NSX IP pool retrieved above.

Step 9: Clean NSX-T Objects After Deletion of a Cluster

In the current version of PKS, NSX-T does not automatically delete NSX-T objects created during the life of the product. After a cluster is deleted, you **must** perform the following task using the `nsx-cli.sh` script downloaded in the previous step (see [Step 8: Deploy a Cluster and Enable NAT Access](#)).

Configuration details follow:

1. Delete the Kubernetes Cluster using the PKS CLI. For more information, see [Delete a Cluster](#).
2. Execute the `nsx-cli` script with the following command:

```
$ ./nsx-cli.sh cleanup CLUSTER-UUID false
```

Where `CLUSTER-UUID` is the ID of the cluster you deleted.

Please send any feedback you have to pks-feedback@pivotal.io.

Upgrading PKS

Page last updated:

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

- [What Happens During PKS Upgrades](#)
- [Upgrade PKS](#)
- [Maintain Workload Uptime](#)
- [Configure the Upgrade Pipeline](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

What Happens During PKS Upgrades

This topic explains what happens to Kubernetes clusters provisioned by Pivotal Container Service (PKS) during PKS upgrades.

Introduction

PKS enables you to upgrade either the PKS tile and all PKS-provisioned Kubernetes clusters or only the PKS tile.

- [Upgrades of the PKS Tile and PKS-Provisioned Clusters](#)
- [Upgrades of the PKS Tile Only](#)

During an upgrade of the PKS tile, your configuration settings are automatically migrated to the new tile version. For upgrading instructions, see [Upgrade PKS](#).

⚠ WARNING: If you upgrade the PKS tile from v1.0.x to v1.1, you must upgrade both the PKS tile and all PKS-provisioned Kubernetes clusters. This ensures existing clusters can run `resize` or `delete` commands after the upgrade.

Canary Instances and max_in_flight

The PKS tile is a BOSH deployment. When you deploy or upgrade a product using BOSH, two things that can affect the deployment are the number of canary instances and the value of the `max_in_flight` variable.

BOSH-deployed products can set a number of canary instances to upgrade first, before the rest of the deployment VMs. BOSH continues the upgrade only if the canary instance upgrade succeeds. If the canary instance encounters an error, the upgrade stops running and other VMs are not affected. The PKS tile uses one canary instance when deploying or upgrading PKS.

The `max_in_flight` variable limits how many instances of a component can restart simultaneously during updates or upgrades. This variable is set to `1` and is not configurable in PKS. Because the value is set to `1`, only one component restarts at a time.

Upgrades of the PKS Tile and PKS-Provisioned Clusters

During an upgrade of the PKS tile and PKS-provisioned clusters, the following occurs:

1. The PKS API server is recreated. For more information, see [PKS API Server](#).
2. Each of your Kubernetes clusters is recreated, one at a time. This includes the following stages for each cluster:
 - a. Master nodes are recreated. For more information, see [Master Nodes](#).
 - b. Worker nodes are recreated. For more information, see [Worker Nodes](#).

💡 Note: When PKS is set to upgrade both the PKS tile and PKS-provisioned clusters, updating any stemcell in your deployment rolls every VM in each Kubernetes cluster. This ensures that all the VMs are patched. With the recommended resource configuration described above, no workload downtime is expected. For information about maintaining your Kubernetes workload uptime, see [Maintain Workload Uptime](#).

PKS API Server

When the PKS API server is recreated, you cannot interact with the PKS control plane or manage Kubernetes clusters. These restrictions prevent you from performing the following actions:


- Logging in through the PKS CLI
- Retrieving information about clusters
- Creating and deleting clusters
- Resizing clusters

Recreating the PKS API server does not affect deployed Kubernetes clusters and their workloads. You can still interact with them through the Kubernetes Command Line Interface, `kubectl`.

For more information about the PKS control plane, see [PKS Control Plane Overview](#) in *PKS Cluster Management*.


Master Nodes

When PKS recreates a single-master cluster during an upgrade, you cannot interact with your cluster, use `kubectl`, or push new workloads.

 **Note:** To avoid this loss of functionality, Pivotal recommends using multi-master clusters.

Worker Nodes

When PKS recreates worker nodes, the upgrade runs on a single VM at a time. During the upgrade, the VM stops running containers. If your workloads run on a single VM, your apps will experience downtime.

 **Note:** To avoid downtime for stateless workloads, Pivotal recommends using at least one worker node per availability zone (AZ). For stateful workloads, Pivotal recommends using a minimum of two worker nodes per AZ.

Upgrades of the PKS Tile Only

During an upgrade of the PKS tile only, the PKS API server is recreated.


When the PKS API server is recreated, you cannot interact with the PKS control plane or manage Kubernetes clusters. These restrictions prevent you from performing the following actions:

- Logging in through the PKS CLI
- Retrieving information about clusters
- Creating and deleting clusters
- Resizing clusters

Recreating the PKS API server does not affect deployed Kubernetes clusters and their workloads. You can still interact with them through the Kubernetes Command Line Interface, `kubectl`.

To upgrade the PKS tile only, set the **Upgrade all clusters errand** to **Off** before you begin the upgrade. For more information, see [Step 2: Upgrade the PKS Tile of Upgrade PKS](#).

For more information about the PKS control plane, see [PKS Control Plane Overview](#) in *PKS Cluster Management*.

 **Note:** When PKS is set to upgrade only the PKS tile and not the clusters, the Kubernetes cluster version falls behind the PKS tile version. If the clusters fall more than one version behind the tile, PKS cannot upgrade the clusters. The clusters must be upgraded to match the PKS tile version before the next tile upgrade.

Please send any feedback you have to pbs-feedback@pivotal.io.

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.

For conceptual information about upgrading the PKS tile and PKS-provisioned Kubernetes clusters, see [What Happens During PKS Upgrades](#).

WARNING: Do not manually upgrade your Kubernetes clusters outside of the PKS tile upgrade. The PKS tile 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. See [Access the Dashboard](#) for more information.

If your cluster is near capacity for your existing infrastructure, Pivotal recommends scaling your cluster before you upgrade. 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 `pkc-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](#).

Upgrade the PKS Tile

To upgrade PKS, you follow the same Ops Manager process that you use to install the service 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](#).

(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**.

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**.

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](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 you run on your clusters and configure your workload based on the best practices recommended by 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.

 **Note:** Ensure that your pods are bound to a *ReplicaSet* or *Deployment*. Naked pods are not rescheduled in the event of a node failure. For more information, see [Configuration Best Practices](#) in the Kubernetes documentation.

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. 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.

Please send any feedback you have to pbs-feedback@pivotal.io.

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.

About the Upgrade Pipeline

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 and Configure 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.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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](#)
- [Manage PKS Deployments with BOSH](#)
- [Add Custom Workloads](#)
- [Download Cluster Logs](#)
- [Service Interruptions](#)
- [Delete PKS](#)

Please send any feedback you have to pbs-feedback@pivotal.io.

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 Access to the PKS API

1. Locate your Ops Manager root CA certificate.
 - If Ops Manager generated your certificate, refer to the [Retrieve the Ops Manager Root Certificate](#) [↗](#) section of *Managing Certificates*.
 - If you provided your own certificate, copy and paste the certificate you entered in the **PKS API** page into a file.
2. Locate the URL of your UAA server. You configured this URL in the UAA section of [Installing and Configuring PKS](#).
3. Run `uaac target UAA-URL --ca-cert ROOT-CA-FILENAME` to target the UAA server. Replace `UAA-URL` with the URL of your UAA server and `ROOT-CA-FILENAME` with the certificate file you downloaded in a previous step. For example:

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

4. 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 > Uaa Admin Secret** to retrieve this value.
5. 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*.
6. Run `pks login -a UAA-URL -u USERNAME -p PASSWORD -k` to log in to the PKS CLI. Replace the `UAA-URL` with the URL of your UAA server, `USERNAME` with your username, and `PASSWORD` with your password. For example:

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

Please send any feedback you have to pks-feedback@pivotal.io.

Manage Users in UAA

Page last updated:

This topic describes how to manage users with User Account and Authentication (UAA) in Pivotal Container Service (PKS).

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 the [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 [Log in as an Admin](#) section to manage users.

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 [Log in as an Admin](#) section to manage users.

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 **Uaa Admin Secret**. 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 of your UAA server. You configured this URL in the UAA 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. 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 to a User

To allow a user to access clusters in PKS, do the following using UAAC:

1. Target your UAA server using `uaac target https://UAA-URL:8443`. Replace `UAA-URL` with the domain name you configured in the **UAA** pane of the PKS tile. For example:

```
$ uaac target https://api.pks.example.com:8443
```

2. Authenticate with UAA using the secret you retrieved in the previous section. Run the following command, replacing `UAA-ADMIN-SECRET` with your UAA admin secret:

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

3. (Optional) Create a user by running `uaac user add USERNAME --emails USER-EMAIL -p USER-PASSWORD`. For example:

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

4. Assign a scope to a user to allow them to access Kubernetes clusters. Run `uaac member add UAA-SCOPE USERNAME`, replacing `UAA-SCOPE` with one of the following UAA scopes:

- `pks.clusters.admin`: Users with this scope have full access to all clusters.
- `pks.clusters.manage`: Users with this scope can only access clusters they create.

For example:

```
$ uaac member add pks.clusters.admin alana
```

Please send any feedback you have to pks-feedback@pivotal.io.

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.

Please send any feedback you have to pkcs-feedback@pivotal.io.


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
```

Please send any feedback you have to pks-feedback@pivotal.io.

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 *Diagnostic Tools* topic for information about using cluster logs to diagnose issues in your PKS deployment.

Please send any feedback you have to pks-feedback@pivotal.io.

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 [Auto-healing Capabilities](#) in 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

You must reinstall the PKS tile. Each cluster may need to be restored manually from backups.

Please send any feedback you have to pbs-feedback@pivotal.io.

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**.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 [Step 7: Apply Changes and 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](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

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

When you create a cluster, you must configure external access to the cluster by creating an external TCP or HTTPS load balancer. 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.

You can configure any load balancer of your choice. If you use vSphere with NSX-T or GCP, 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](#).

Create the load balancer before you create the cluster. Use the load balancer IP address as the external hostname, and 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 cluster's master VM after cluster creation.

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 -p PASSWORD --ca-cert CERT-PATH
```

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`. If you created an external load balancer, use its IP address.
- `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
```

 **Note:** It can take up to 30 minutes to create a cluster.

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 **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.

- For **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.
- For **GCP**, use the output from the `pks cluster` command to locate the master node IP addresses and ports, and then continue to [Step 3: Configure Load Balancer Backend](#) in *Configuring a GCP Load Balancer for PKS Clusters*.

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

6. To access your cluster, run `pks get-credentials CLUSTER-NAME`. This command creates a local `kubeconfig` that allows you to manage the cluster. See [Retrieve Cluster Credentials and Configuration](#) for more information.

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.

Please send any feedback you have to pkcs-feedback@pivotal.io.

Retrieve Cluster Credentials and Configuration

This topic describes how to use the `pkcs get-credentials` command in Pivotal Container Service (PKS) using the PKS Command Line Interface (CLI).

The `pkcs 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 `pkcs 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 `pkcs get-credentials`. For example:

```
$ KUBECONFIG=/path/to/my-cluster.config pkcs 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:

```
pkcs login -a PKS_API -u USERNAME -p PASSWORD --ca-cert CERT-PATH
```

See [Log in to the PKS CLI](#) for more information about the `pkcs login` command.

2. Run the following command:

```
pkcs get-credentials CLUSTER-NAME
```

Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pkcs 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](#).

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 -p PASSWORD --ca-cert CERT-PATH
```

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
```

Please send any feedback you have to pks-feedback@pivotal.io.

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 -p PASSWORD --ca-cert CERT-PATH
```

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
```

Please send any feedback you have to pks-feedback@pivotal.io.

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 -p PASSWORD --ca-cert CERT-PATH
```

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

Please send any feedback you have to pkcs-feedback@pivotal.io.

Using Dynamic Persistent Volumes

When using PKS, you can choose to pre-provision persistent storage or create on-demand persistent storage volumes. Refer to [Persistent Volumes](#) in 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.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 -p PASSWORD --ca-cert CERT-PATH
```

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:

- `CLUSTER-NAME` is the name of your cluster.
- `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
```

Please send any feedback you have to pbs-feedback@pivotal.io.

Access Dashboard

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized apps to a Kubernetes cluster, troubleshoot containerized apps, 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 apps.

To access Dashboard, follow the steps in *Accessing the Dashboard UI* in the [Kubernetes Web UI \(Dashboard\)](#) [↗](#) documentation.

You must have `kubectl` credentials to access Dashboard. Requiring these credentials prevents unauthorized admin access to the Kubernetes cluster through a browser.

Accessing Dashboard

Follow the steps below to access Dashboard for a Kubernetes cluster.

1. Retrieve the `kubectl` credentials. As a PKS operator or developer, you may already have access to `kubectl` credentials. If you do not, follow the instructions in [Retrieve Cluster Credentials and Configuration](#).
2. After retrieving `kubectl` credentials, run `kubectl proxy` on a command line. Do not exit or close the terminal.
3. Dashboard is now available. In a web browser, browse to `http://localhost:8001/ui` to access Dashboard.

Please send any feedback you have to pbs-feedback@pivotal.io.

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`. For more information about the `NodePort` Service type, see [Services](#) in the Kubernetes documentation.
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`. For more information about the `NodePort` Service type, see [Services](#) in the Kubernetes documentation.
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.

Please send any feedback you have to pks-feedback@pivotal.io.

Delete a Cluster

Follow the steps below to delete a cluster using the PKS CLI.

1. On the command line, run the following command to log in:

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

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
```

Please send any feedback you have to pks-feedback@pivotal.io.

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.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 of 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](#). Here are some examples of charts you can use:

- [Concourse](#) for CI/CD pipelines
- [Datadog](#) for monitoring
- [MySQL](#) for storage

This topic includes instructions to [Install Concourse Using Helm](#). For more charts, see the Kubernetes [charts repository](#) on GitHub.

If you want to to use Helm with PKS, see the following sections:

- [Configure Tiller](#)
- [Install Concourse Using Helm](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 [Role-based Access Control](#) in the Helm documentation.

Please send any feedback you have to pkcs-feedback@pivotal.io.

Install Concourse Using Helm

Page last updated:

Perform the steps in this section to install Concourse using Helm.

 **Note:** Concourse requires privileged containers. You must deploy a cluster using a plan that allows privileged containers before installing the Concourse chart. For information about configuring plans, see the [Plans](#) section of *Installing and Configuring PKS*.

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.

Please send any feedback you have to pbs-feedback@pivotal.io.

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](#)

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 `pkcs --version` to determine the version of PKS CLI installed locally. For example:

```
$ pkcs --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.

Please send any feedback you have to pkcs-feedback@pivotal.io.

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 `pks 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 task TASK-NUMBER`. For example:

```
$ bosh -e pks task 23
```

For more information about troubleshooting failed BOSH tasks, see [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 v2.0 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 [Delete a Cluster](#) in *Using PKS*.
9. On the command line, enter `pkcs create-cluster` to recreate the cluster. For more information, see [Create a Cluster](#) in *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 v2.0 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 [Delete a Cluster](#).
 9. On the command line, enter `pkcs create-cluster` to recreate the cluster. For more information, see [Create a Cluster](#).
-

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.
-

Please send any feedback you have to pkcs-feedback@pivotal.io.

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: 1.0.0-build3

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
-n, --num-nodes string	Number of worker nodes
-p, --plan string	Preconfigured plans. Run pks list-plans for more details

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
```

Please send any feedback you have to pbs-feedback@pivotal.io.

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 CF CR

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 CF CR.

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](#).

Please send any feedback you have to pbs-feedback@pivotal.io.