Pivotal

PRODUCT DOCUMENTATION

Pivotal Container Service (PKS)

Version 1.1

Published: 4 February 2019

Table of Contents

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

- vSphere Prerequisites and Resource Requirements
- vSphere with NSX-T Prerequisites and Resource Requirements
- GCP Prerequisites and Resource Requirements

Preparing to Install PKS

To install PKS, you must deploy Ops Manager v2.1 or v2.2. You use Ops Manager to install and configure PKS.

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

Consult the following table for compatibility information:

IaaS	Ops Manager v2.1 or v2.2	NSX-T	Harbor
vSphere	Required	Available	Available
GCP	Required	Not Available	Available

For more information about compatibility and component versions, see the PKS Release Notes.

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

- vSphere
- vSphere with NSX-T Integration
- GCP

Installing PKS

For information about installing PKS, see *Installing PKS* for your IaaS:

- vSphere
- vSphere with NSX-T Integration
- GCP

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.

Backing Up and Restoring PKS

For information about using BOSH Backup and Restore (BBR) to back up and restore PKS, see Backing Up and Restoring PKS.

PKS Security

For information about security in PKS, see PKS Security.

Diagnosing and Troubleshooting PKS

For information about diagnosing and troubleshooting issues installing or using PKS, see Diagnosing and Troubleshooting PKS.

PKS Release Notes

Page last updated:

This topic contains release notes for Pivotal Container Service (PKS) v1.1.x.

v1.1.6

Release Date: September 24, 2018

Product Snapshot

Element	Details
Version	v1.1.6
Release date	September 24, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.42
Kubernetes version	v1.10.7
NSX-T version	v2.1, v2.2
NCP version	v2.2.1

What's New

- Updates stemcell to v3586.42.
- Updates Kubernetes to v1.10.7.
- The default for the Worker Persistent Disk Type has been updated to 50GB.
- The default for the Master/ETCD and Worker VM Type has been updated to 32GB disk.

Known Issues

- The default for the Master/ETCD VM Type on Plan 2 should be updated to have a minimum disk size of 32GB.

v1.1.5

Release Date: August 31, 2018

Product Snapshot

Element	Details
Version	v1.1.5
Release date	August 31, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.36
Kubernetes version	v1.10.5
NSX-T version	v2.1, v2.2
NCP version	v2.2.1

What's New

- Updates stemcell to 3586.36.
- Adds support for NSX-T v2.2.
- Updates NCP to v2.2.1.
- NSX-T Architectural Changes

Known Issues

You cannot enter whitespace into any of the fields in the PKS tile, including leading and trailing spaces and spaces between characters. Using a space in any field causes the PKS deployment to fail.

The following known issues apply to PKS deployments on vSphere with NSX-T:

> **Note**: The issues listed below pertain to NSX-T v2.2 and NCP v2.2.1. NSX-T v2.3 and NCP v2.3 include fixes for these issues. PKS support for these versions is under development for a future release.

- Updating load balancer rules fails from TLS ingress to non-TLS ingress with NCP restart.
- Stale pool found when deleting an ingress rule which is updated from non-TLS to TLS.
- Deletion of HTTPS VS pool fails after updating NCP.
- NCP crashes on restart if the load balancer has max virtual servers.
- TLS ingress certificate is not removed after deleting all related TLS ingress objects.
- SNI certificate is not updated after changing non-TLS ingress to TLS ingress with NCP restart.
- NCP error annotations are not found when updating the LBIPPool from a valid to nonexistent IPPool.
- NSX cleanup operation does not release the external IP or delete SNAT rules on the T0 router.

The following known issue applies to PKS deployments on GCP:

- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

  Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

NSX-T Architectural Changes

> **Note**: The changes in this section apply to PKS deployments on vSphere with NSX-T.

PKS v1.1.5 includes architectural changes related to its integration with NSX-T and NCP. PKS uses NCP to integrate with NSX-T. For more information about NCP, see Overview of NSX-T Container Plug-in in the VMware documentation.

NSX-T Node Agent and Kube Proxy

In PKS v1.1.4 and earlier, the NSX-T Node Agent and NSX-T Kube Proxy run as a daemon set on each worker node. In PKS v1.1.5, both the NSX-T Node Agent and the Kube Proxy run as BOSH-managed processes on each worker node.

NSX-T Container Plugin (NCP)

> **Note**: You do not need to install or configure NCP. NCP is automatically installed and configured when you deploy PKS in an NSX-T environment.

In PKS v1.1.4 and earlier, NCP runs as a Kubernetes pod on a single worker node. With PKS v1.1.5, NCP runs as a BOSH-managed process on the Kubernetes master node.

In PKS v1.1.5, if you deploy a multi-master cluster, the NCP process runs on all master nodes but is active on only a single master. If the NCP process on an

active master is unresponsive, BOSH activates another NCP process.

PKS Logs for NSX-T and NCP

In PKS v1.1.4 and earlier, you access NSX-T and NCP logs using `kubectl` commands. In PKS v1.1.5, NSX-T and NCP are BOSH-managed processes, and you access the logs for these components using BOSH.

BOSH jobs related to NSX-T integration with NCP as a BOSH process:

Location	BOSH Jobs
Master Node	`/var/vcap/sys/log/ncp`
	`/var/vcap/sys/log/pks-nsx-t-prepare-master-vm`
	`/var/vcap/sys/log/pks-nsx-t-ncp`
Worker Nodes	`/var/vcap/sys/log/nsx-kube-proxy`
	`/var/vcap/sys/log/openvswitch`
	`/var/vcap/sys/log/nsx-cni`
	`/var/vcap/sys/log/nsx-node-agent`

Run the BOSH command `bosh -d MY-DEPLOYMENT logs` to collect these logs, replacing `MY-DEPLOYMENT` with the name of your PKS deployment. For more information, see Using Logs in the BOSH documentation.

When you upgrade to PKS v1.1.5, the existing logs for NSX-T and NCP are deleted. Before you upgrade, you may want to back these logs up. For example, you may need to analyze these logs if you experience problems with your PKS deployment before upgrading, or problems related to a failed upgrade.

v1.1.4

Release Date: August 8, 2018

Product Snapshot

Element	Details
Version	v1.1.4
Release date	August 8, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.27
Kubernetes version	v1.10.5
NSX-T version	v2.1
NCP version	v2.2

What's New

- Updates stemcell to 3586.27.
- Updates Kubernetes to v1.10.5.
- Includes security enhancements.

Known Issues

- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

v1.1.3

Release Date: July 30, 2018

Product Snapshot

Element	Details
Version	v1.1.3
Release date	July 30, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.26
Kubernetes version	v1.10.4

What's New

- Updates stemcell to 3586.26.
- Telemetry information is now sent less frequently.

Known Issues

- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

  Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

v1.1.2

Release Date: July 17, 2018

Product Snapshot

Element	Details
Version	v1.1.2
Release date	July 17, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.24
Kubernetes version	v1.10.4

Security Fixes

This release includes the following security fix:

- High CVE-2018-11047: UAA accepts refresh token as access token on admin endpoints

Known Issues

- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

  Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

v1.1.1

Release Date: July 16, 2018

Product Snapshot

Element	Details
Version	v1.1.1
Release date	July 16, 2018
Compatible Ops Manager versions	v2.1.x, v2.2.x
Stemcell version	3586.24
Kubernetes version	v1.10.4

Note: PKS v1.1.1 and later can be deployed on Ops Manager v2.1 or v2.2. Pivotal recommends using Ops Manager v2.2 to deploy PKS. For added security in Ops Manager v2.2, disable the **Allow Legacy Agents** option in the **Director Config** pane of the BOSH Director tile. For more information, see the Ops Manager configuration topic for your cloud provider. For example, Configuring BOSH Director on vSphere.

What's New

- UAA and security enhancements
- NSX-T patches
- Telemetry patch
- Kubernetes 1.10.4

Bug Fixes

Ops Manager v2.1.7 and later is now supported in PKS v1.1.1. However, Pivotal recommends using Ops Manager v2.2 to deploy PKS.

Upgrade Procedure

To upgrade to PKS v1.1.1, you must upgrade from PKS v1.0.2 or later.

To upgrade to PKS v1.1.1, follow the procedures in Upgrading PKS. Pivotal recommends using Ops Manager v2.2 to deploy PKS.

For added security in Ops Manager v2.2, disable the **Allow Legacy Agents** option in the **Director Config** pane of the BOSH Director tile. For more information, see the Ops Manager configuration topic for your cloud provider. For example, Configuring BOSH Director on vSphere.

Known Issues

- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

  Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

v1.1.0

WARNING: PKS v1.1.0 is no longer available for download from Pivotal Network.

Release Date: June 28, 2018

Upgrade Procedure

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

To upgrade to PKS v1.1.0, follow the procedures in Upgrading PKS.

Features

This section describes new features introduced in PKS v1.1.0.

General Features

- Adds support for Kubernetes 1.10.3.
- Adds support for backing up and restoring PKS using BOSH Backup and Restore (BBR). For more information, see Backing Up and Restoring PKS.
- Adds support for granting PKS control plane access to clients and external LDAP groups. For more information, see the Grant Cluster Access section of *Manage Users in UAA*.
- Adds support for allowing workers to be deployed across Availability Zones (AZs).
- Adds support for network automation and node network isolation.
- Adds support for NFS by enabling rpcbind on worker nodes.
- Adds support for kube-controller-manager to issue certificates.
- Adds support for configuring HTTP/HTTPS proxy to be used by the Kubernetes control plane.
- Adds support for configuring the SecurityContextDeny admission controller. For more information, see Using Admission Controllers in the Kubernetes documentation.
- Enables the MutatingAdmissionWebhook admission controller. For more information, see Using Admission Controllers in the Kubernetes documentation.
- Enables audit logging for the API server.
- Creates logs for delete-all-cluster errands in the /var/vcap/sys/log/delete-all-clusters folder on the PKS control plane VM.
- Adds BOSH instance IDs to worker node labels.
- Hardens security by removing the ABAC authorization option for clusters.
- Hardens security by using service account IDs instead of service account keys for GCP deployments.
- Hardens security for Kubernetes system components. For example, kube-dns now uses its own configuration instead of the kubelet configuration.

vSphere Features

- Adds support for NO-NAT deployment topologies for PKS installations on NSX-T. For more information, see Deployment Topologies.
- Adds support for PKS integration with VMware Wavefront to capture metrics for clusters and pods. For more information, see the (Optional) Logging section of *Installing PKS* for your IaaS. For example, see Installing PKS on vSphere.
- Adds support for node network access using HTTP proxy for vSphere deployments. For more information, see the Networking section of *Installing PKS on vSphere*.
- Adds support for PKS integration with VMware vRealize Log Insight (vRLI) for tagged logging of the control plane, clusters, and pods. For more information, see the (Optional) Monitoring section of *Installing PKS* for your IaaS. For example, see Installing PKS on vSphere.
- Adds support for integration with VMware Analytics Cloud (VAC) to capture telemetry information.
- Hardens security by removing VM change permissions from worker nodes for vSphere deployments.
- Hardens security by removing vCenter user credentials from worker nodes for vSphere deployments.

- Adds support for Harbor Registry integration enhancements: updated Harbor tile, ability to use NFS and Google Buckets as an image store, and HTTP/HTTPS proxy servers for Clair.

Bug Fixes

- Prevents unnecessary route creation in the kube-controller-manager.
- Retains the original source IP when using Flannel.
- Disables the read-only port in the kubelet configuration.
- Disables cAdvisor in the kubelet configuration.
- For added security, the Kubernetes API server no longer tries to fix malformed requests.
- The Kubernetes API server now cleans up terminated pods more often to avoid running out of disk space.
- The Kubernetes API server now unmounts volumes of terminated pods for security reasons.
- Operators no longer have to manually delete NSX-T objects created during the life of the product. In PKS v1.1, running the `pks delete-cluster` command deletes all NSX objects.

Beta Components

- Adds support for deploying multiple Kubernetes master nodes across AZs. For information about configuring multiple masters, see the Plans section of *Installing PKS* for your IaaS. For example, see Installing PKS on vSphere.

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

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

Component Versions

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

WARNING: Ops Manager v2.1.7 and later is not supported in PKS v1.1.0.

Product Component	Version Supported	Notes
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.1.0-2.1.6	Separate download available from Pivotal Network
Stemcell	3586.24	
Kubernetes	1.10.3	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.17	Packaged in the PKS Tile
Golang	1.9.7	Packaged in the PKS Tile
NCP	2.2	Packaged in the PKS Tile
Kubernetes CLI	1.10.3	Separate download available from the PKS section of Pivotal Network
PKS CLI	1.1	Separate download available from the PKS section of Pivotal Network
VMware vSphere	6.5 U2 and 6.5 U1. Editions: • vSphere Enterprise Plus Edition • vSphere with Operations Management Enterprise Plus	vSphere versions supported for Pivotal Container Service (PKS)
VMware NSX-T	2.1 - Advanced Edition	NSX-T versions supported for Pivotal Container Service (PKS)

VMware Harbor Registry	1.5.0	Separate download available from Pivotal Network
VMware vRealize Log Insight (for vSphere deployments)	4.6	Separate download available from Pivotal Network
** 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 v1.1.0 and corresponding workarounds.

- PKS v1.1.0 does not support Ops Manager v2.1.7 and later. For more information, see Error: Duplicate Variable Name in the *Troubleshooting* topic.
- If you use PKS CLI v1.0.x with PKS tile v1.1.x, you must log in every 600 seconds to manually refresh the CLI token. Pivotal recommends upgrading to PKS CLI v1.1.x to solve this issue.
- If you upgrade PKS from v1.0.x to v1.1, you must enable the **Upgrade All Clusters** errand in the PKS tile configuration. This ensures existing clusters can perform resize or delete actions after the upgrade.
- If you use stemcell v3586.18 or later in the 3586 line of Linux stemcells when deploying PKS on GCP, you may see the following:
  - The output of the `bosh vms` command shows an error message that includes `unresponsive agent`.
  - Your PKS-provisioned Kubernetes cluster does not respond to any PKS CLI commands, such as `pks get-credentials` or `pks delete-cluster`.

  Until this issue is resolved, use stemcell v3586.16 when deploying PKS on GCP.

Cluster Security Recommendations

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

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

Reconfigure GCP Load Balancers After Master VM Recreation

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

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

Existing ABAC Clusters

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

New Default VM Type

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

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

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

PKS Concepts

Page last updated:

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

- PKS Cluster Management
- PKS API Authentication
- Load Balancers in PKS
- VM Sizing for PKS Clusters

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

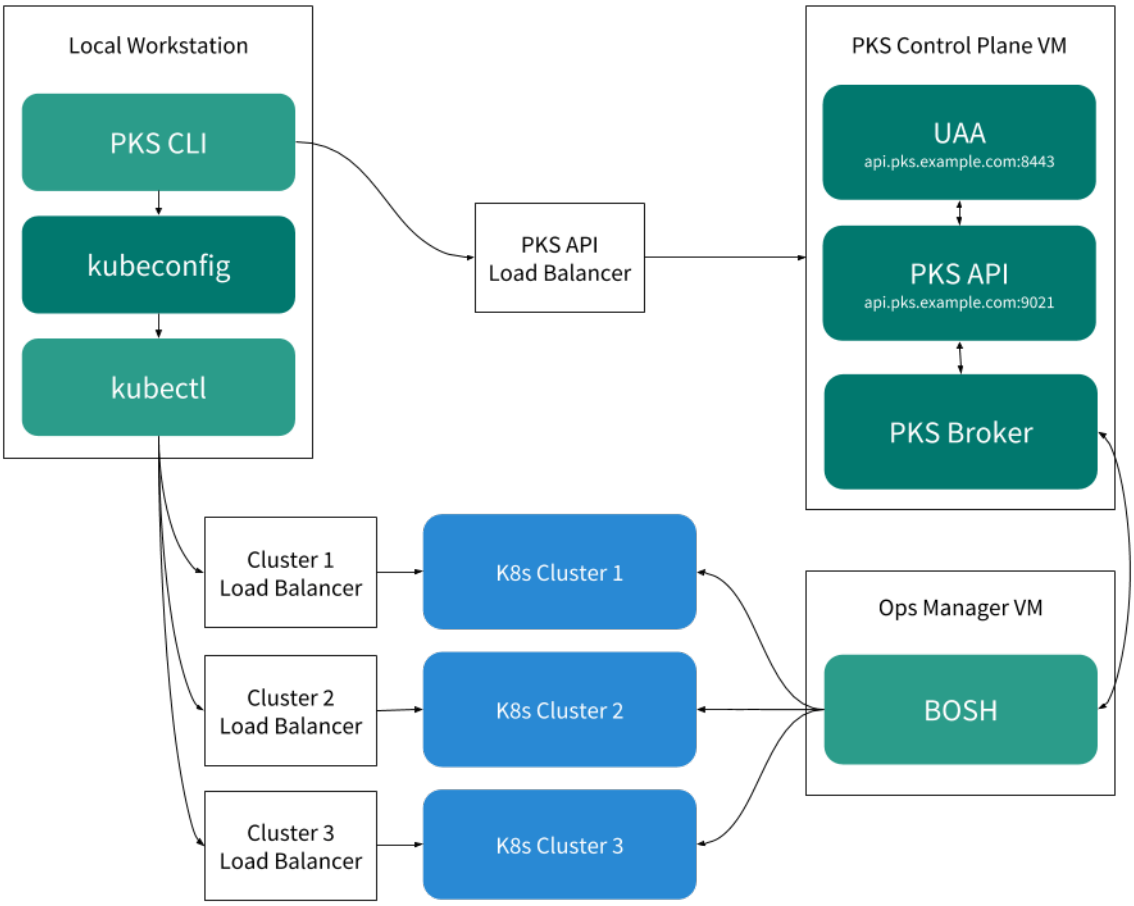
In addition, the PKS control plane can upgrade all existing clusters using the **Upgrade all clusters** BOSH errand. For more information, see Upgrade Kubernetes Clusters in *Upgrading PKS*.

PKS Control Plane Architecture

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

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

The following illustration shows how these components interact:



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

UAA

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

UAA must be configured with the appropriate users and user permissions. For more information, see Managing Users in PKS with UAA.

PKS API

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

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

PKS Broker

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

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

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

Cluster Workload Management

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

PKS API Authentication

Page last updated:

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

Authenticating PKS API Requests

Before users can log in and use the PKS CLI, you must configure PKS API access with UAA. FOr more information, see Configuring PKS API Access. 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. For more information, Grant Cluster Access in *Managing Users in PKS with UAA*.

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, `pks 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 `pks clusters`, the CLI returns a list of the clusters that the user is authorized to manage.

Routing to the PKS API Control Plane VM

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

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

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

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

For procedures that describe routing to the PKS control plane VM, see the Configure External Load Balancer section of *Installing PKS* for your IaaS.

For overview information about load balancers in PKS, see Load Balancers in PKS Deployments without NSX-T.

Load Balancers in PKS

Page last updated:

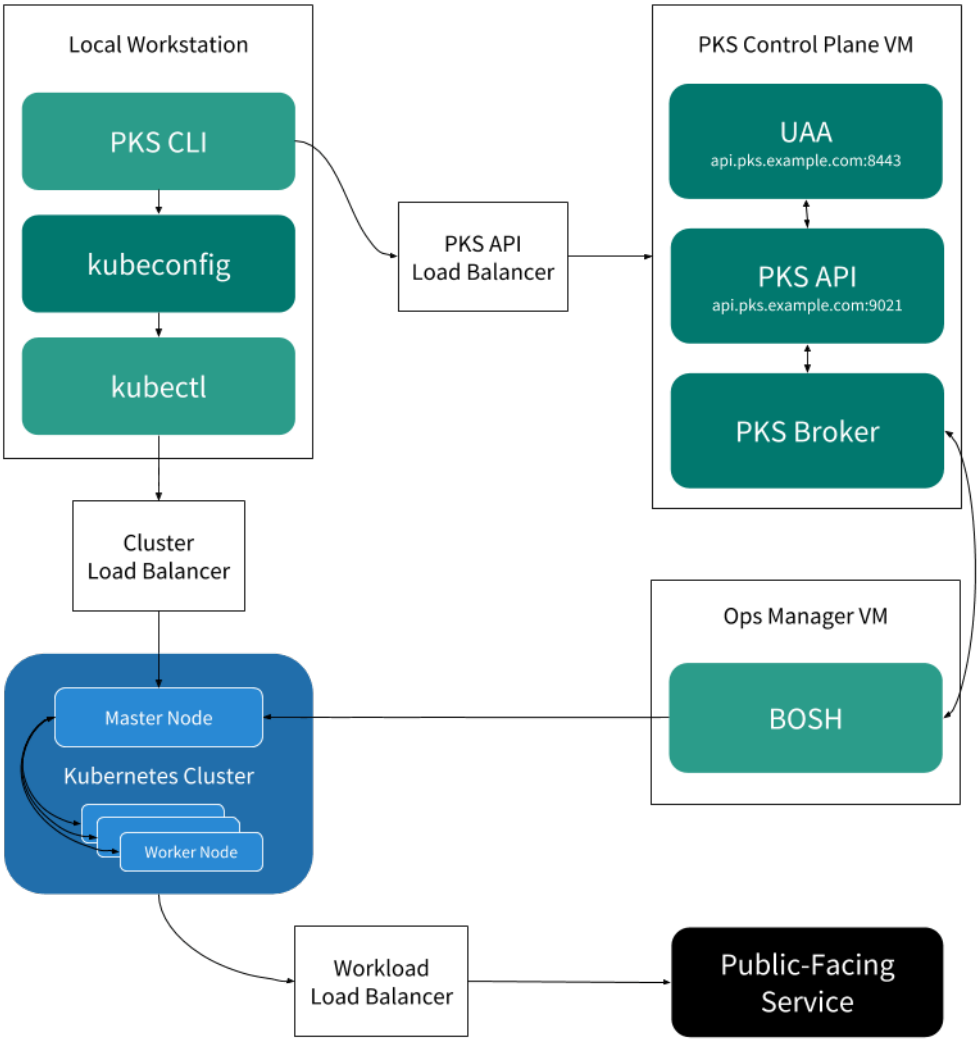
This topic describes the types of load balancers that are used in Pivotal Container Service (PKS) deployments. Load balancers differ by the type of deployment.

Load Balancers in PKS Deployments without NSX-T

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

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

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



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

About the PKS API Load Balancer

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

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

For information about configuring the PKS API load balancer, see the Configure External Load Balancer section of *Installing PKS* for your IaaS.

About Kubernetes Cluster Load Balancers

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

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

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

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

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

```
10.0.0.5 kubo.internal
```

For information about configuring a cluster load balancer, see Creating Clusters.

About Workload Load Balancers

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

For information about configuring a load balancer for your app workload, see Deploying and Accessing Basic Workloads.

Load Balancers in PKS Deployments on vSphere with NSX−T

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

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

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

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

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

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

Resizing Load Balancers

When a new Kubernetes cluster is created using the PKS API, NSX-T creates a dedicated load balancer for that new cluster. By default, the size of the load balancer is set to **Small** in NSX Manager. A Small sized load balancer is limited to a maximum of 10 NSX-T virtual servers.

Note: Pivotal recommends changing the size of your NSX-T load balancer from **Small** to **Medium** in NSX Manager. Doing so increases your virtual server limit from 10 to 100.

VM Sizing for PKS Clusters

Page last updated:

This topic describes how Pivotal Container Service (PKS) recommends you approach the sizing of VMs for cluster components.

Overview

When you configure plans in the PKS tile, you provide VM sizes for the master and worker node VMs. For more information about configuring plans, see the Plans section of *Installing PKS* for your IaaS:

- vSphere
- vSphere with NSX-T Integration
- Google Cloud Platform (GCP)

PKS determines the size of the master node VMs automatically based on the number of worker node VMs. You select the number of master nodes when you configure the plan.

For worker node VMs, you select the number and size based on the needs of your workload. The sizing of master and worker node VMs is highly dependent on the characteristics of the workload. Adapt the recommendations in this topic based on your own workload requirements.

Master Node VM Size

The master node VM size is linked to the number of worker nodes. The VM sizing shown in the following table is per master node:

Note: If there are multiple master nodes, all master node VMs are the same size. To configure the number of master nodes, see the Plans section of *Installing PKS* for your IaaS.

Number of Workers	CPU	RAM (GB)
1-5	1	3.75
6-10	2	7.5
11-100	4	15
101-250	8	30
251-500	16	60
500+	32	120

Worker Node VM Number and Size

A maximum of 100 pods can run on a single worker node. The actual number of pods that each worker node runs depends on the workload type as well as the CPU and memory requirements of the workload.

To calculate the number and size of worker VMs you require, determine the following for your workload:

- Maximum number of pods you expect to run [`p`]
- Memory requirements per pod [`m`]
- CPU requirements per pod [`c`]

Using the values above, you can calculate the following:

- Minimum number of workers [`W`] = `p / 100`
- Minimum RAM per worker = `m * 100`
- Minimum number of CPUs per worker = `c * 100`

This calculation gives you the minimum number of worker nodes your workload requires. We recommend that you increase this value to account for

failures and upgrades.

For example, increase the number of worker nodes by at least one to maintain workload uptime during an upgrade. Additionally, increase the number of worker nodes to fit your own failure tolerance criteria.

The maximum number of worker nodes that you can create for a PKS-provisioned Kubernetes cluster is 50.

Example Worker Node Requirement Calculation

An example app has the following minimum requirements:

- Number of pods [`p`] = 1000
- RAM per pod [`m`] = 1 GB
- CPU per pod [`c`] = 0.10

To determine how many worker node VMs the app requires, do the following:

1. Calculate the number of workers using `p / 100`:

    ```
    1000/100 = 10 workers
    ```

2. Calculate the minimum RAM per worker using `m * 100`:

    ```
    1 * 100 = 100 GB
    ```

3. Calculate the minimum number of CPUs per worker using `c * 100`:

    ```
    0.10 * 100 = 10 CPUs
    ```

4. For upgrades, increase the number of workers by one:

    ```
    10 workers + 1 worker = 11 workers
    ```

5. For failure tolerance, increase the number of workers by two:

    ```
    11 workers + 2 workers = 13 workers
    ```

In total, this app workload requires 13 workers with 10 CPUs and 100 GB RAM.

PKS Telemetry

Page last updated:

This topic describes the metrics that the Pivotal Container Service (PKS) tile sends when you enable the VMware Customer Experience Improvement Program (CEIP) or the Pivotal Telemetry Program (Telemetry). You can opt in or opt out of either program in the **Usage Data** pane of the PKS tile.

For more information, see the *Installing PKS* topic for your IaaS:

- vSphere
- vSphere with NSX-T Integration
- Google Cloud Platform (GCP)

Event Envelope Properties

When PKS sends metrics to CEIP or Telemetry, the tile packages the data with the following deployment information:

Property Name	Property Description	Example Data	Added in PKS Version
event	The type of event	create_cluster	v1.1
product_version	PKS tile version	1.2.0-build.40	v1.1
cloud_provider	Cloud provider for the PKS installation	GCP	v1.1
vcenter_id	vCenter ID	00000a11-22bb-3333-4c4c-555566667777	v1.1

Cluster Events

PKS sends metrics for the cluster management events shown in the table below:

Event Name	Event Description	Property Name	Property Description	Added in PKS Version
create_cluster	This event is generated when a user creates a cluster.	**user_id**	A hashed value of the username.	v1.1
		timestamp	The time when the user created the cluster.	v1.1
		plan_name	The name of the PKS plan that was used to create the cluster.	v1.1
		plan_id	The ID of the PKS plan that was used to create the cluster.	v1.1
		cluster_name	The name of the cluster.	v1.1
		cluster_id	The ID of the cluster.	v1.1
		number_of_workers	The number of worker node VMs in the cluster.	v1.1
resize_cluster	This event is generated when a cluster is resized.	**user_id**	A hashed value of the username.	v1.1
		timestamp	The time when the user created the cluster.	v1.1
		plan_name	The name of the PKS plan that was used to create the cluster.	v1.1
		plan_id	The ID of the PKS plan that was used to create the cluster.	v1.1
		cluster_name	The name of the cluster.	v1.1
		cluster_id	The ID of the cluster.	v1.1
		old_number_of_workers	The number of worker node VMs in the cluster before the resize event.	v1.1
		new_number_of_workers	The number of worker node VMs in the cluster after the resize event.	v1.1
		user_id	A hashed value of the username.	v1.1
		timestamp	The time when the user created the cluster.	v1.1

delete_cluster	This event is generated when a user deletes a cluster.	plan_name	The name of the PKS plan that was used to create the cluster.	v1.1
		plan_id	The ID of the PKS plan that was used to create the cluster.	v1.1
		cluster_name	The name of the cluster.	v1.1
		cluster_id	The ID of the cluster.	v1.1

Cluster Metrics

PKS sends both agent metrics and cluster pod metrics for each cluster.

The following table describes cluster agent metrics:

Agent Metric Name	Agent Metric Description	Example	Added in PKS Version
agentid	The unique BOSH-generated deployment name for the cluster.	service-instance_00000a11-22bb-3333-4c4c-555566667777	v1.1
isvrlienabled	If vRealize Log Insight (vRLI) is enabled, this value is true. If vRLI is disabled, this value if false.	true	v1.1
isvropsenabled	If vRealize Operations (vROps) is enabled, this value is true. If vROps is disabled, this value is false.	false	v1.1
iswavefrontenabled	If Wavefront is enabled, this value is true. If Wavefront is disabled, this value is false.	true	v1.1
vcenter_id	This is your vCenter ID.	00000a11-22bb-3333-4c4c-555566667777	v1.1

The following table describes cluster pod metrics:

Cluster Pod Metric Name	Cluster Pod Metric Description	Example	Added in PKS Version
collected_at	This timestamp represents the metric collection time on the agent.	2018-05-31 21:45:27.681 UTC	v1.1
cpu_used	This value represents how much CPU was in use at the time when the event happened.	11412427	v1.1
memory_used	This value represents how much memory was in use at the time when the event happened.	4816896	v1.1
pkst_kubernetesclusterinfo__fk	This value is a foreign key that points to an entry in the *pkst_kubernetesclusterinfo* database.	77777a66-55bb-4444-3c3c-222211110000	v1.1

PAS and PKS Deployments with Ops Manager

Page last updated:

Ops Manager is a web app that you use to deploy and manage Pivotal Application Service (PAS) and Pivotal Container Service (PKS). This topic explains why Pivotal recommends using separate installations of Ops Manager for PAS and PKS.

For more information about deploying PKS, see Installing PKS.

Security

Ops Manager deploys the PAS and PKS runtime platforms using BOSH. For security reasons, Pivotal does not recommend installing PAS and PKS on the same Ops Manager instance. For even stronger security, Pivotal recommends deploying each Ops Manager instance using a unique cloud provider account.

Tile Configuration and Troubleshooting

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.

Installing PKS

Page last updated:

You can install Pivotal Container Service (PKS) on Google Cloud Platform (GCP) or vSphere. For installation instructions, see the following:

- vSphere
- vSphere with NSX-T Integration
- GCP

vSphere

This topic lists the steps to follow when installing Pivotal Container Service (PKS) on vSphere.

Installing PKS

To install PKS, follow the instructions below:

- Prerequisites and Resource Requirements
- Preparing vSphere Before Deploying PKS
- Deploying Ops Manager on vSphere:
  - Deploying BOSH and Ops Manager v2.1 to vSphere
  - Deploying BOSH and Ops Manager v2.2 to vSphere
- Configuring Ops Manager on vSphere:
  - Configuring BOSH Director v2.1 on vSphere
  - Configuring BOSH Director v2.2 on vSphere
- Installing PKS on vSphere
- (Optional) Integrating VMware Harbor with PKS

Installing the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

vSphere Prerequisites and Resource Requirements

Page last updated:

This topic describes the prerequisites and resource requirements for installing Pivotal Container Service (PKS) on vSphere.

For prerequisites and resource requirements for installing PKS on vSphere with NSX-T integration, see vSphere with NSX-T Prerequisites and Resource Requirements.

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 VMware Harbor Registry in the *Pivotal Partner documentation*.

Prerequisites

Before installing PKS, you must install Ops Manager. You use Ops Manager to install and configure PKS.

To prepare your vSphere environment for installing Ops Manager and PKS, review the sections below and then follow the instructions in Preparing vSphere Before Deploying PKS.

vSphere Version Requirements

Ops Manager and PKS support the following vSphere component versions:

Versions	Editions
• VMware vSphere 6.5 U2* • VMware vSphere 6.5 U1	• vSphere Enterprise Plus • vSphere with Operations Management Enterprise Plus

*PKS v1.1.2 and later are compatible with vSphere 6.5 U2.

Resource Requirements

Installing Ops Manager and PKS requires the following virtual machines (VMs):

VM	CPU	RAM	Storage
Pivotal Container Service	2	8 GB	16 GB
Pivotal Ops Manager	1	8 GB	160 GB
BOSH Director	2	8 GB	16 GB

Each PKS deployment requires ephemeral VMs during installation and upgrades of PKS. After you deploy PKS, BOSH automatically deletes these VMs.

To enable PKS to dynamically create the ephemeral VMs when needed, ensure that the following resources are available in your vSphere infrastructure before deploying PKS:

Ephemeral VM	Number	CPU Cores	RAM	Ephemeral Disk
BOSH Compilation VMs	4	4	4 GB	32 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	Number	CPU Cores	RAM	Ephemeral Disk	Persistent Disk
master	1 or 3	2	4 GB	8 GB	5 GB
worker	1 or more	2	4 GB	8 GB	50 GB
errand (ephemeral)	1	1	1 GB	8 GB	none

Preparing vSphere Before Deploying PKS

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 PKS on vSphere*.

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** privileges to the service account:

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

Step 2: Grant Storage Permissions

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)

For more information about vSphere storage configurations, see vSphere Storage for Kubernetes in the VMware vSphere documentation.

Static Only Persistent Volume Provisioning

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

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

a. Grant the following privileges at the **VM Folder** level using either the vCenter UI or API:

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

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

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

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

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

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

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

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

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

4. Continue to Step 3: Create the BOSH/Ops Manager Service Account.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5. Continue to Step 3: Create the BOSH/Ops Manager Service Account.

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

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

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

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

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

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

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

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

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

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

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

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

Step 3: Create the BOSH/Ops Manager Service Account

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~~ Mark as template	~~VirtualMachine.Provisioning.DeployTemplate~~ 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

After you complete the instructions provided in this topic, install one of the following:

- Pivotal Ops Manager v2.1.x
- Pivotal Ops Manager v2.2.x

Note: You use Ops Manager to install and configure PKS. Each version of Ops Manager supports multiple versions of PKS. To confirm that your Ops Manager version supports the version of PKS that you install, see PKS Release Notes.

To install an Ops Manager version that is compatible with the PKS version you intend to use, follow the instructions in the corresponding version of the Ops Manager documentation.

Version	
Ops Manager v2.1	• Deploying BOSH and Ops Manager to vSphere • Configuring BOSH Director on vSphere
Ops Manager v2.2	• Deploying BOSH and Ops Manager to vSphere • Configuring BOSH Director on vSphere

Installing PKS on vSphere

Page last updated:

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

Prerequisites

Before performing the procedures in this topic, you must have deployed and configured Ops Manager. For more information, see vSphere Prerequisites and Resource Requirements.

If you use 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. Open the **Director Config** pane.
3. Select the **Enable Post Deploy Scripts** checkbox.
4. Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
5. Click the **Installation Dashboard** link to return to the Installation Dashboard.
6. Click **Apply Changes**.

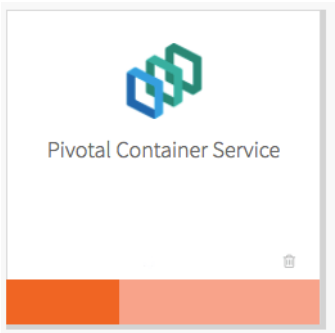
Step 1: Install PKS

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.



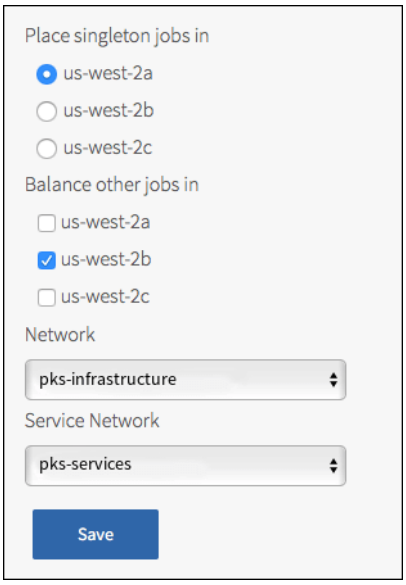
Assign AZs and Networks

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.

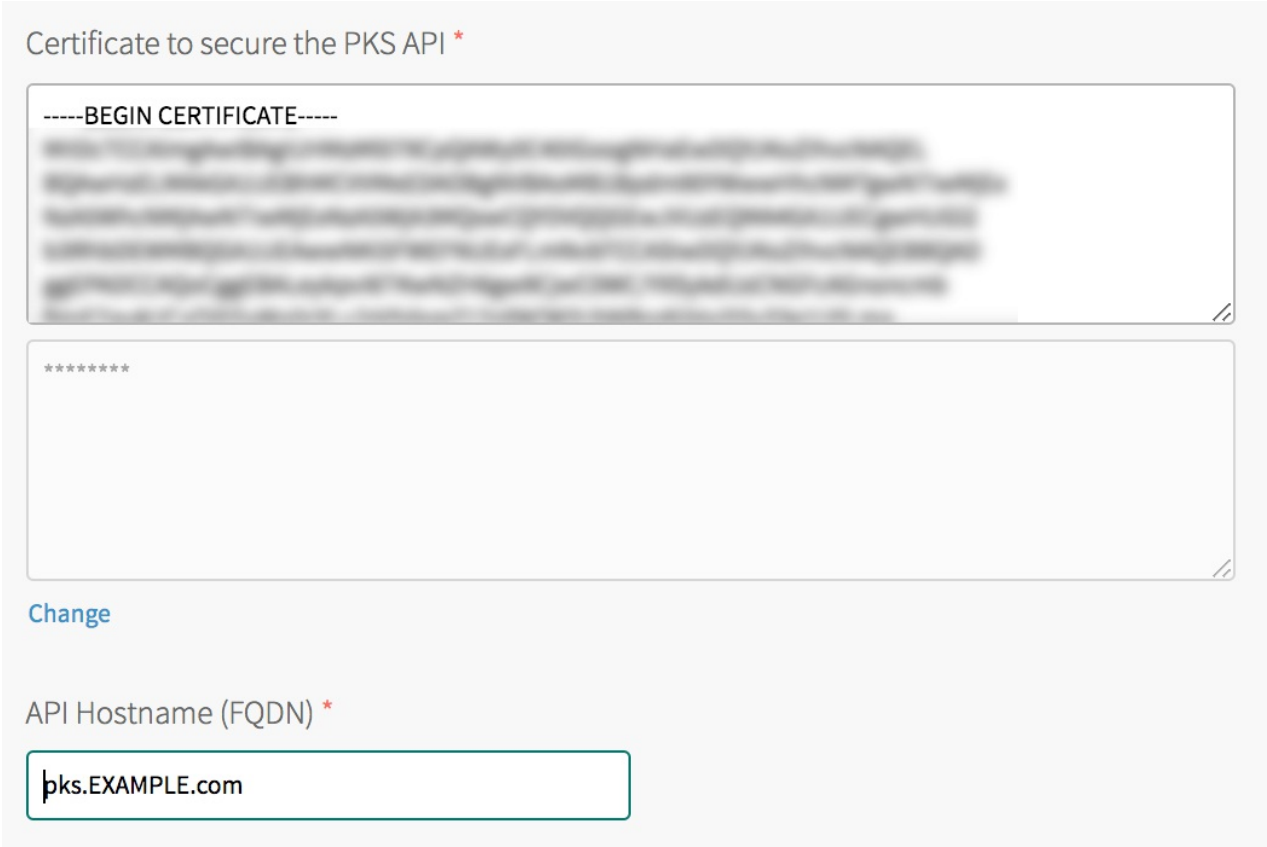


3. Under **Network**, select the infrastructure subnet you created for the PKS API VM.
4. Under **Service Network**, select the services subnet you created for Kubernetes cluster VMs.
5. Click **Save**.

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 that you supply should cover the domain that routes to the PKS API VM with TLS termination on the ingress.

If you do not have a certificate and private key pair, PKS can generate one for you by performing the following steps.

> **Note**: If you configured Ops Manager Front End without a certificate, you can use this new certificate to complete Ops Manager configuration. To configure your Ops Manager Front End certificate, see Configure Front End.

   a. Select the **Generate RSA Certificate** link.
   b. Enter the wildcard domain for your API hostname. For example, if your PKS API domain is `api.pks.example.com`, then enter `*.pks.example.com`.
   c. Click **Generate**.

3. Under **API Hostname (FQDN)**, enter a fully qualified domain name (FQDN) to access the PKS API. For example, `api.pks.example.com`.

4. Click **Save**.

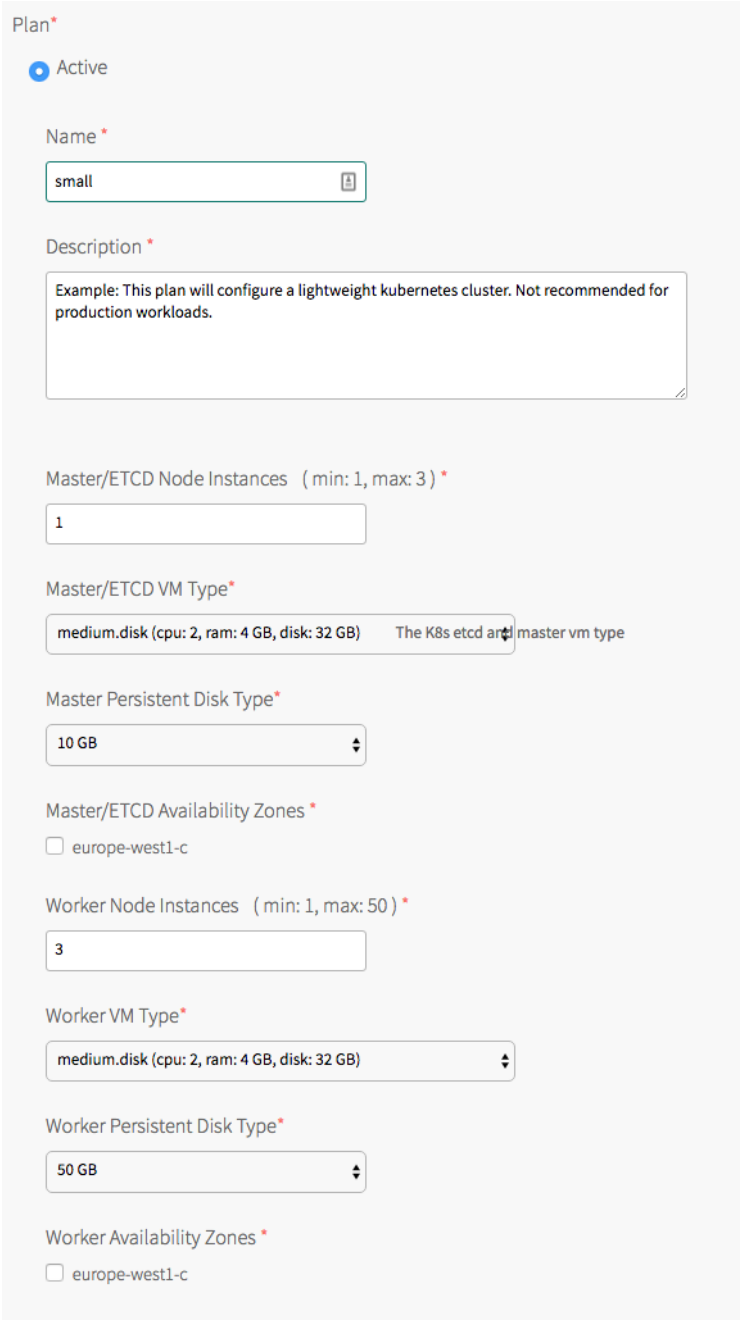
Plans

To activate a plan, perform the following steps:

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

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

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



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

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

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

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

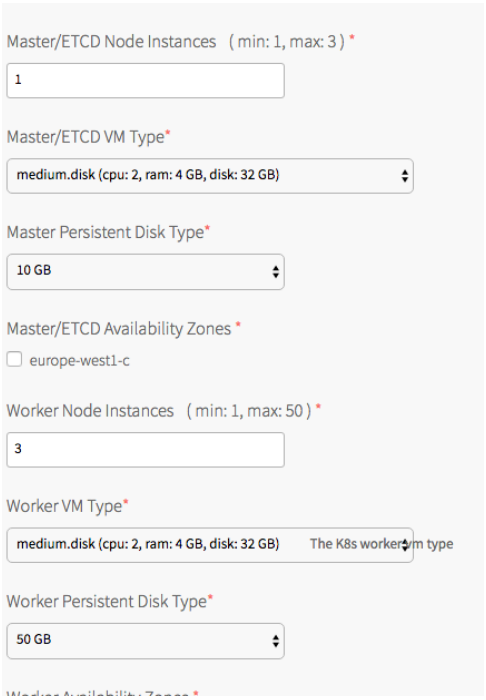
   WARNING: This feature is a beta component and is intended for evaluation and test purposes only. Do not use this feature in a production environment. Product support and future availability are not guaranteed for beta components.

6. Under **Master/ETCD VM Type**, select the type of VM to use for Kubernetes master/etcd nodes. For more information, see the Master Node VM Size section of *VM Sizing for PKS Clusters*.

7. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master node VM.

8. Under **Master/ETCD Availability Zones**, select one or more AZs for the Kubernetes clusters deployed by PKS. If you select more than one AZ, PKS deploys the master VM in the first AZ and the worker VMs across the remaining AZs.

9. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, create clusters with a minimum of three worker nodes, or two per AZ if you intend to use persistent volumes. For example, if you deploy across three AZs, you should have six worker nodes. For more information about persistent volumes, see Persistent Volumes in *Maintaining Workload Uptime*. Provisioning a minimum of three worker nodes, or two nodes per AZ is also recommended for stateless workloads.



10. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker node VMs. For more information, see the Worker Node VM Number and Size section of *VM Sizing for PKS Clusters*.

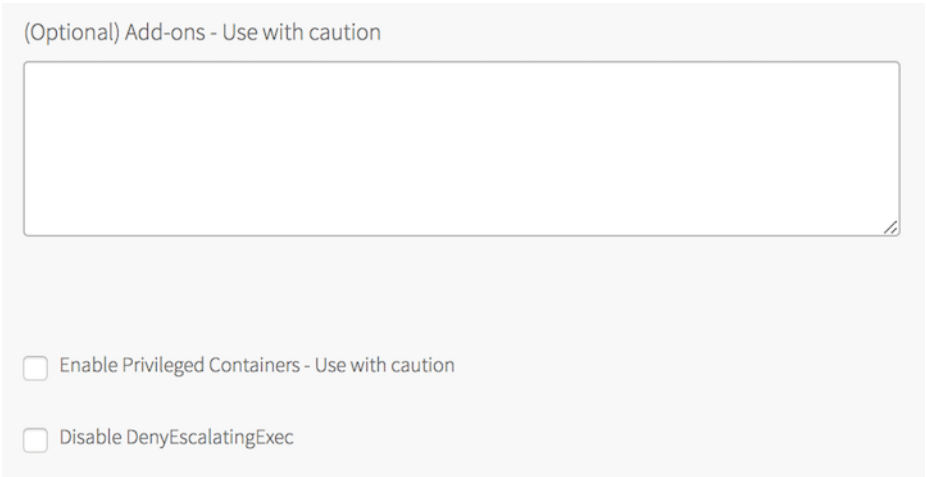
> **Note**: If you install PKS v1.1.5 or later in an NSX-T environment, we recommend that you select a **Worker VM Type** with a minimum disk size of 16 GB. The disk space provided by the default “medium” Worker VM Type is insufficient for PKS with NSX-T v1.1.5 or later.

11. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker node VMs.

12. Under **Worker Availability Zones**, select one or more AZs for the Kubernetes worker nodes. PKS deploys worker nodes equally across the AZs you select.

13. Under **Errand VM Type**, select the size of the VM that contains the errand. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.

14. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to add custom workloads to each cluster in this plan. You can specify multiple files using `---` as a separator. For more information, see Adding Custom Workloads.



15. (Optional) To allow users to create pods with privileged containers, select the **Enable Privileged Containers - Use with caution** option. For more information, see Pods in the Kubernetes documentation.

16. (Optional) To disable the admission controller, select the **Disable DenyEscalatingExec** checkbox. If you select this option, clusters in this plan can create security vulnerabilities that may impact other tiles. Use this feature with caution.

17. Click **Save**.

To deactivate a plan, perform the following steps:

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

2. Select **Plan Inactive**.

3. Click **Save**.

Kubernetes Cloud Provider

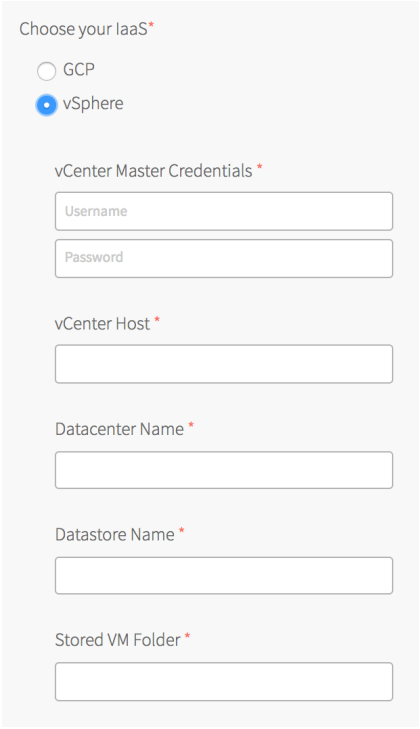
In the procedure below, you use credentials for vCenter master VMs. You must have provisioned the service account with the correct permissions. For more information, see Create the Master Node Service Account in *Preparing vSphere Before Deploying PKS*.

To configure your Kubernetes cloud provider settings, follow the procedure below:

1. Click **Kubernetes Cloud Provider**.

2. Under **Choose your IaaS**, select **vSphere**.

3. Ensure the values in the following procedure 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`. For more information about the master node service account, see Preparing vSphere Before Deploying PKS.
   b. Enter your **vCenter Host**. For example, `vcenter.CF-EXAMPLE.com`.
   c. Enter your **Datacenter Name**. For example, `CF-EXAMPLE-dc`.
   d. Enter your **Datastore Name**. For example, `CF-EXAMPLE-ds`.
   e. Enter the **Stored VM Folder** so that the persistent stores know where to find the VMs. To retrieve the name of the folder, navigate to your BOSH Director tile, click **vCenter Config**, and locate the value for **VM Folder**. The default folder name is `pcf_vms`.

> **Note**: We recommend using a shared datastore for multi-AZ and multi-cluster environments.

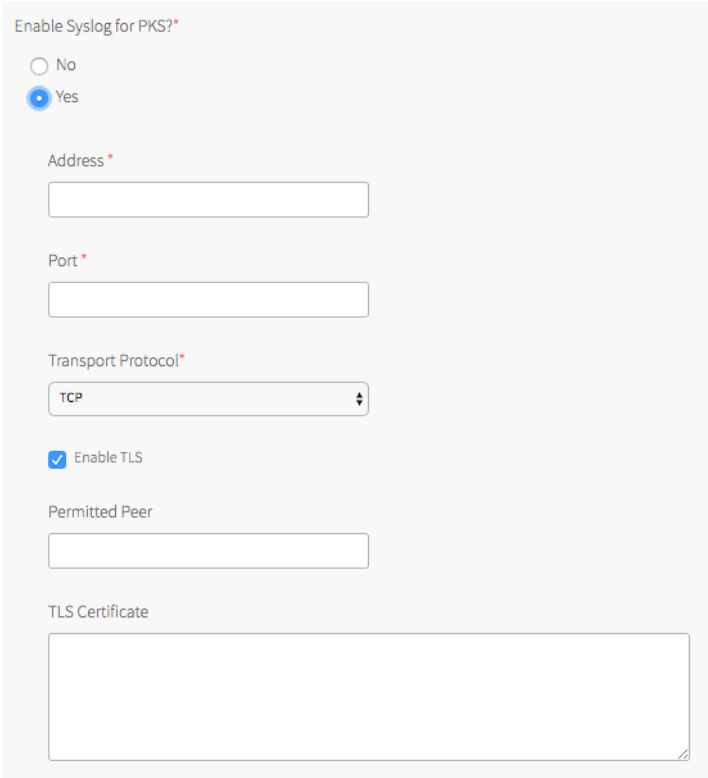
4. Click **Save**.

(Optional) Logging

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

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

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



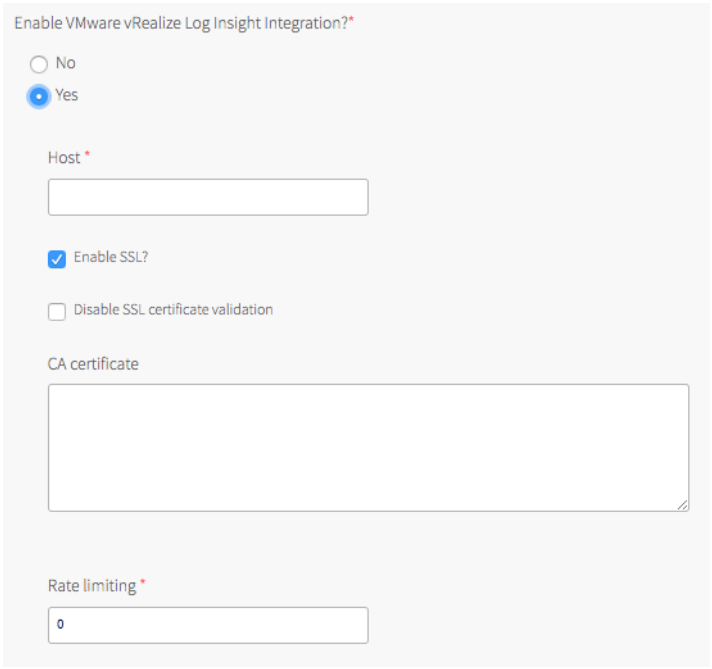
3. Under **Address**, enter the destination syslog endpoint.
4. Under **Port**, enter the destination syslog port.
5. Select a transport protocol for log forwarding.
6. (Optional) Pivotal strongly recommends that you enable TLS encryption when forwarding logs as they may contain sensitive information.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

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

7. You can manage logs using VMware vRealize Log Insight (vRLI). The integration pulls logs from all BOSH jobs and containers running in the cluster, including node logs from core Kubernetes and BOSH processes, Kubernetes event logs, and POD stdout and stderr.

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

    By default, vRLI logging is disabled. To enable and configure vRLI logging, under **Enable VMware vRealize Log Insight Integration?**, select **Yes** and



then perform the following steps:

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

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

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

- To enable certificate validation for the vRLI host, clear the **Disable SSL certificate validation** checkbox.

d. (Optional) If your vRLI certificate is not signed by a trusted CA root or other well known certificate, enter the certificate in the **CA certificate** field. Locate the PEM of the CA used to sign the vRLI certificate, copy the contents of the certificate file, and paste them into the field. Certificates must be in PEM-encoded format.
e. Under **Rate limiting**, enter a time in milliseconds to change the rate at which logs are sent to the vRLI host. The rate limit specifies the minimum time between messages before the fluentd agent begins to drop messages. The default value (0) means the rate is not limited, which suffices for many deployments.

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

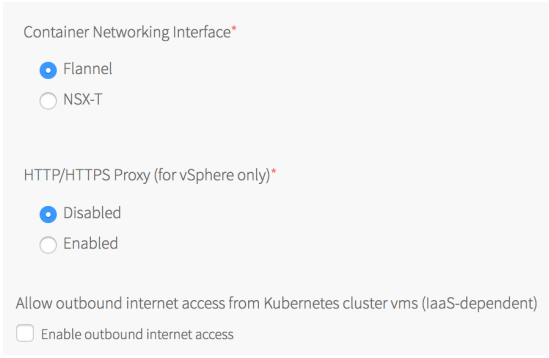
8. Click **Save**. These settings apply to any clusters created after you have saved these configuration settings and clicked **Apply Changes**. If the **Upgrade all clusters errand** has been enabled, these setting are also applied to existing clusters.

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

Networking

To configure networking, do the following:

1. Click **Networking**.

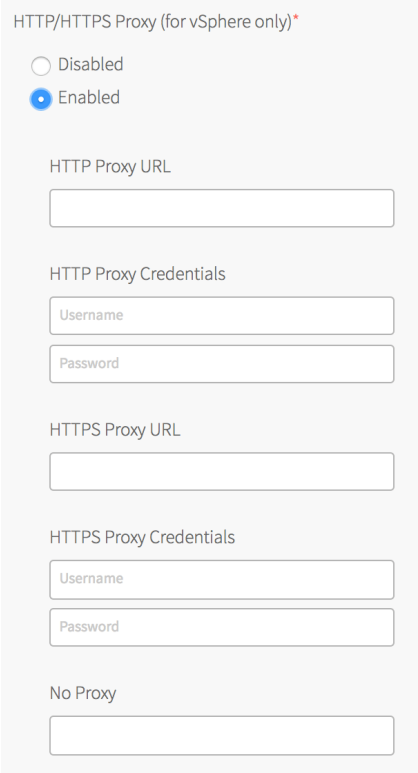


2. Under **Container Networking Interface**, select **Flannel**.

3. (Optional) Configure a global proxy for all outgoing HTTP and HTTPS traffic from your Kubernetes clusters. This setting will not set the proxy for running Kubernetes workloads or pods.

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

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



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

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

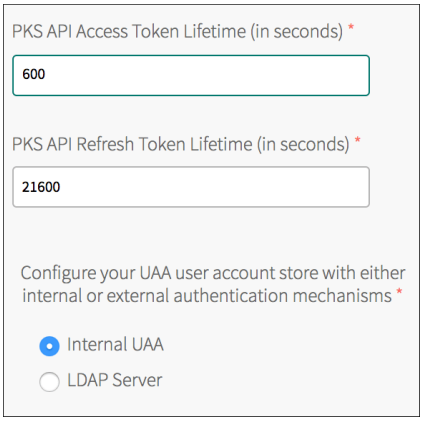
4. Under **Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)**, ignore the **Enable outbound internet access** checkbox.

5. Click **Save**.

UAA

To configure the UAA server, do the following:

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

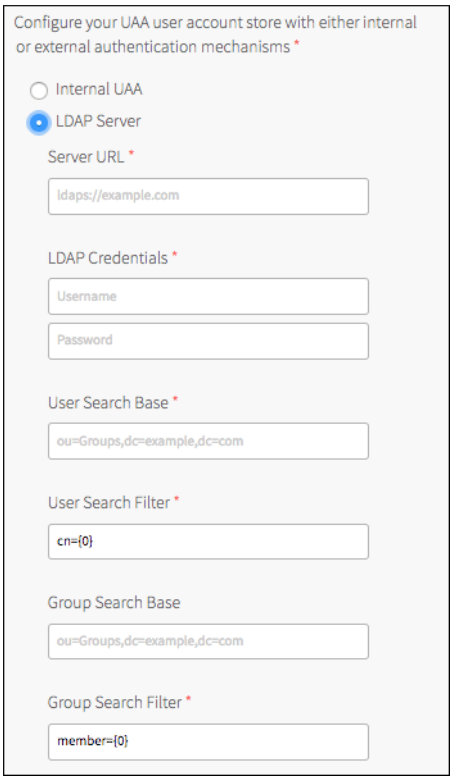


3. Under **PKS CLI Refresh Token Lifetime**, enter a time in seconds for the PKS CLI refresh token lifetime.
4. Select one of the following options:
    - To use an internal user account store for UAA, select **Internal UAA**. Click **Save** and continue to (Optional) Monitoring.
    - To use an external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

Configure LDAP as an Identity Provider

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

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



2. For **Server URL**, enter the URLs that point to your LDAP server. If you have multiple LDAP servers, separate their URLs with spaces. Each URL must include one of the following protocols:

- `ldap://`: Use this protocol if your LDAP server uses an unencrypted connection.
- `ldaps://`: Use this protocol if your LDAP server uses SSL for an encrypted connection. To support an encrypted connection, the LDAP server must hold a trusted certificate or you must import a trusted certificate to the JVM truststore.

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

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

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

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

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

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

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

   > **Note**: For information about testing and troubleshooting your LDAP search filters, see Configuring LDAP Integration with Pivotal Cloud Foundry.

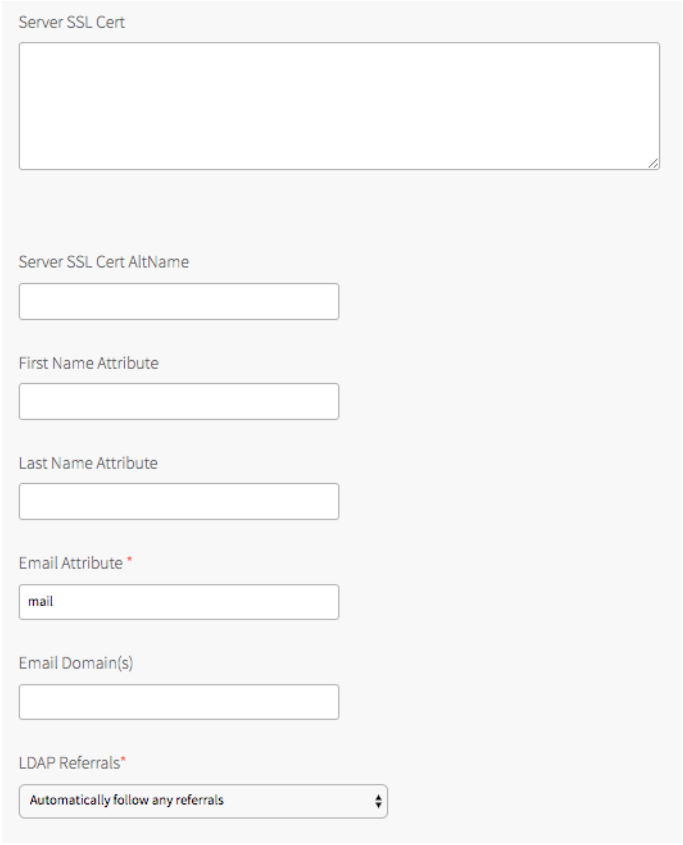
6. For **Group Search Base**, enter the location in the LDAP directory tree where the LDAP group search begins.

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

   Follow the instructions in the Grant Cluster Access to an External LDAP Group section of *Managing Users in PKS with UAA* to map the groups under this search base to roles in PKS.

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

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



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

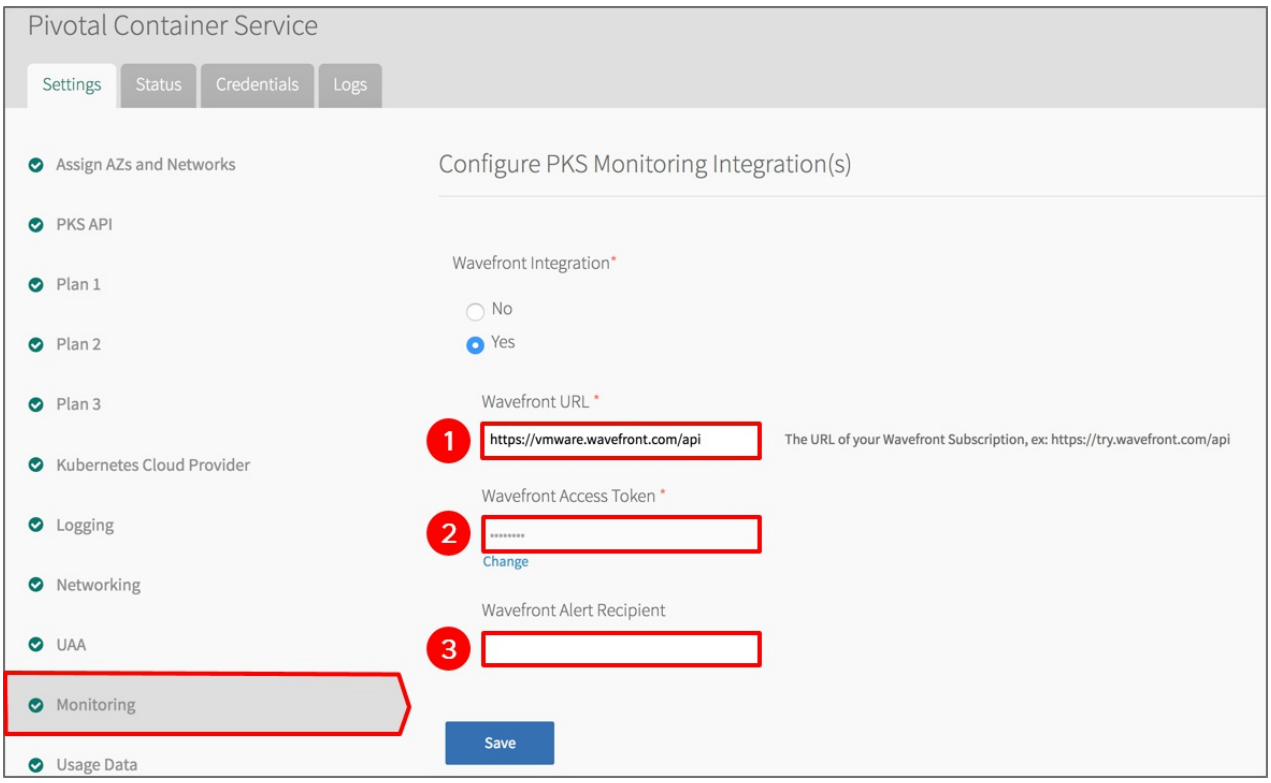
(Optional) Monitoring

You can monitor Kubernetes clusters and pods metrics externally using the integration with Wavefront by VMware.

Note: Before you configure Wavefront integration, you must have an active Wavefront account and access to a Wavefront instance. You provide your Wavefront access token during configuration and enabling errands. For additional information, see Pivotal Container Service Integration Details in the Wavefront documentation.

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

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

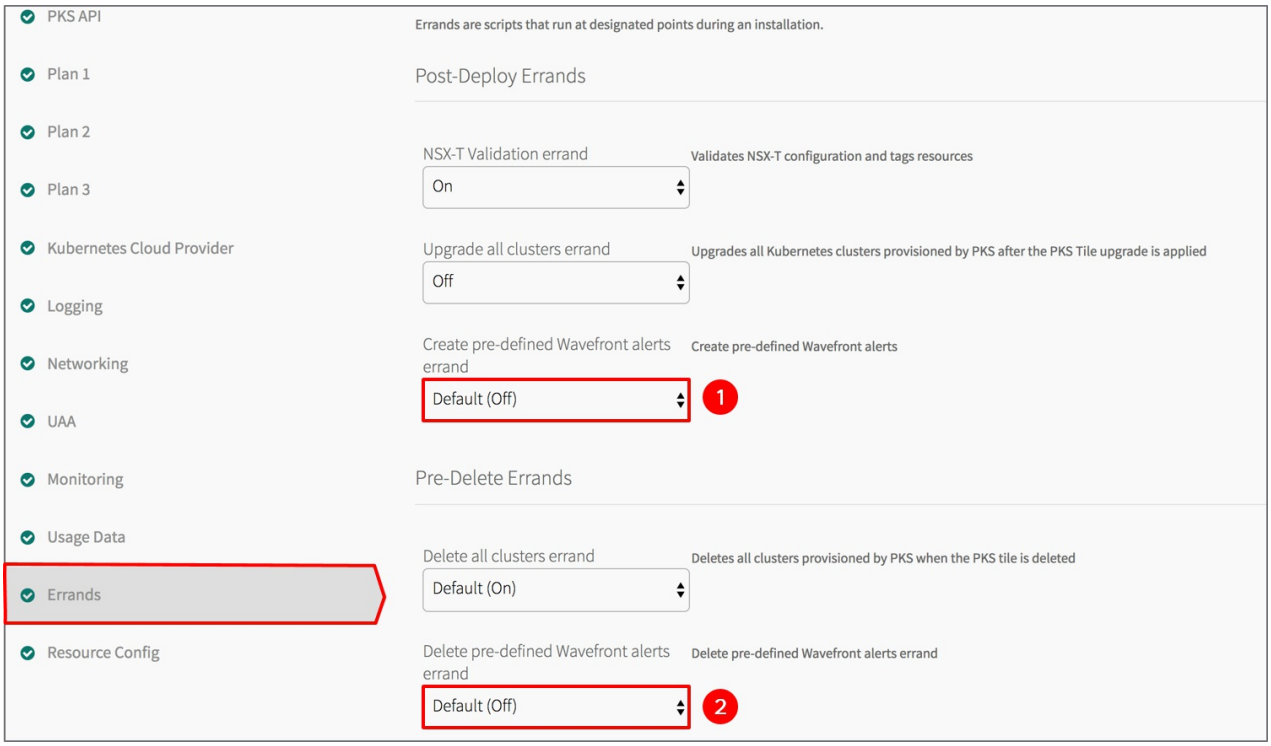


2. Under **Wavefront URL**, enter the URL of your Wavefront subscription. For example, `https://try.wavefront.com/api`.

3. Under **Wavefront Access Token**, enter the API token for your Wavefront subscription.

4. To configure Wavefront to send alerts by email, enter email addresses or Wavefront Target IDs separated by commas under **Wavefront Alert Recipient**. For example: `user@example.com,Wavefront_TargetID`. To create alerts, you must enable errands.

5. In the **Errands** tab, enable **Create pre-defined Wavefront alerts errand** and **Delete pre-defined Wavefront alerts errand**.



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

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

Wavefront.

Usage Data

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

For information about the metrics PKS sends when you opt in to CEIP or Telemetry, see PKS Telemetry.

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

To configure the **Usage Data** pane:

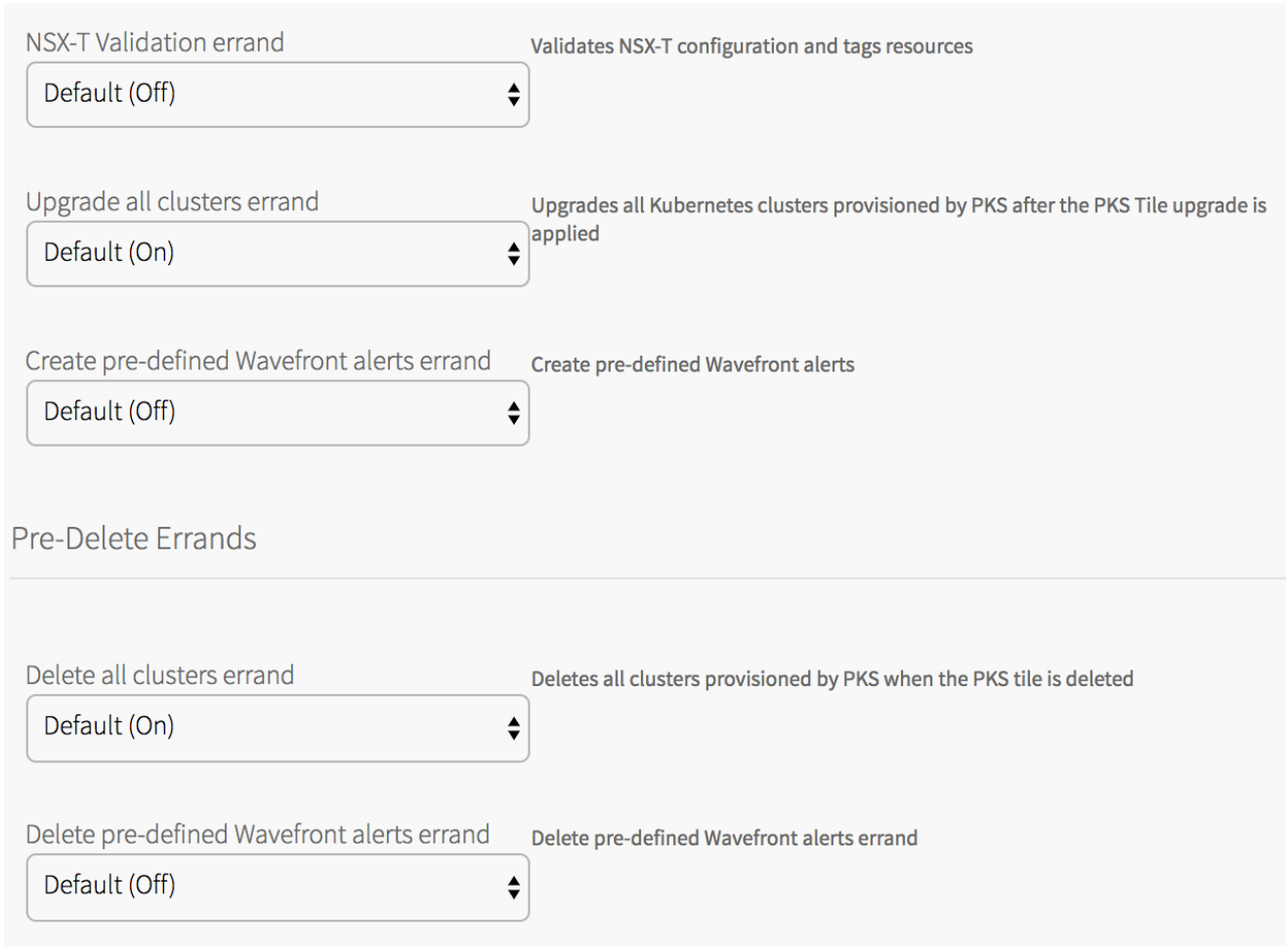
1. Select the **Usage Data** side-tab.
2. Read the Usage Data description.
3. Make your selection.
    a. To join the program, select **Yes, I want to join the CEIP and Telemetry Program for PKS**.
    b. To decline joining the program, select **No, I do not want to join the CEIP and Telemetry Program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

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

Errands

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

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



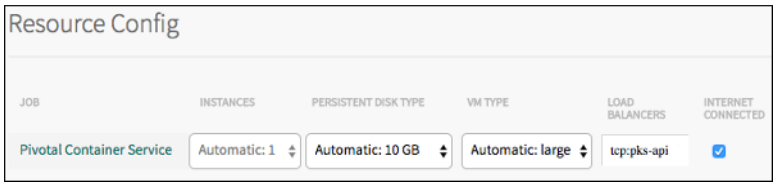
For more information about errands and their configuration state, see Managing Errands in Ops Manager.

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

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

Resource Config

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



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.

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 the 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 Creating Clusters 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

After you install the PKS tile, configure an external load balancer to access the PKS API from outside the network. You can use any external load balancer.

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

Configure your load balancer with the following information:

- IP address from Retrieve PKS API Endpoint
- Ports 8443 and 9021
- HTTPS or TCP protocol

Step 6: Install the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

Step 7: Configure PKS API Access

Follow the procedures in Configuring PKS API Access.

Step 8: Configure Authentication for PKS

Configure authentication for PKS using User Account and Authentication (UAA). For information about managing users in PKS with UAA, see Managing Users in PKS with UAA.

Next Steps

After installing PKS on vSphere, you may want to do the following:

- Integrate VMware Harbor with PKS to store and manage container images. For more information, see Integrating VMware Harbor Registry with PKS.
- Create your first PKS cluster. For more information, see Creating Clusters.

vSphere with NSX-T Integration

This topic lists the steps to follow when installing Pivotal Container Service (PKS) on vSphere with NSX-T integration.

Installing PKS

To install PKS, follow the instructions below:

- Prerequisites and Resource Requirements
- Deployment Topologies
- Preparing NSX-T Before Deploying PKS
- Deploying Ops Manager on vSphere with NSX-T
- Configuring Ops Manager on vSphere with NSX-T Integration
- Generating and Registering Certificates
- Installing PKS on vSphere with NSX-T
- (Optional) Integrating VMware Harbor with PKS

Installing the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

vSphere with NSX-T Prerequisites and Resource Requirements

Page last updated:

This topic describes the prerequisites and resource requirements for installing Pivotal Container Service (PKS) on vSphere with NSX-T integration.

For prerequisites and resource requirements for installing PKS on vSphere without NSX-T integration, see vSphere Prerequisites and Resource Requirements.

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 VMware Harbor Registry in the *Pivotal Partner documentation*.

Prerequisites

vSphere Version Requirements

PKS on vSphere supports the following vSphere component versions:

Versions	Editions
• VMware vSphere 6.5 U2* • VMware vSphere 6.5 U1	• vSphere Enterprise Plus • vSphere with Operations Management Enterprise Plus

*PKS v1.1.2 and later are compatible with vSphere 6.5 U2.

NSX-T Integration Component Version Requirements

Deploying NSX-T requires the additional following component versions:

Component	Version
VMware NSX-T	2.2 and 2.1 Advanced edition

Resource Requirements

PKS

Installing Ops Manager and PKS requires the following virtual machines (VMs):

VM	CPU	RAM	Storage
Pivotal Container Service	2	8 GB	16 GB
Pivotal Ops Manager	1	8 GB	160 GB
BOSH Director	2	8 GB	16 GB

Each PKS deployment requires ephemeral VMs during installation and upgrades of PKS. After you deploy PKS, BOSH automatically deletes these VMs.

To enable PKS to dynamically create the ephemeral VMs when needed, ensure that the following resources are available in your vSphere infrastructure before deploying PKS:

Ephemeral VM	Number	CPU Cores	RAM	Ephemeral Disk
BOSH Compilation VMs	4	4	4 GB	16 GB

Kubernetes

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	Number	CPU Cores	RAM	Ephemeral Disk	Persistent Disk
master	1 or 3	2	4 GB	8 GB	5 GB
worker	1 or more	2	4 GB	8 GB	10 GB
errand (ephemeral)	1	1	1 GB	8 GB	none

NSX-T

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

Firewall Requirements

For the firewall ports and protocols requirements for using PKS on vSphere with NSX-T integration, see Firewall Ports and Protocols Requirements.

Other Requirements

- Complete any confirmation tasks described in the VMware NSX-T Data Center Documentation to verify your configuration before proceeding to the next step.
- Comply with any requirements described in the VMware NSX-T Data Center Documentation.

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

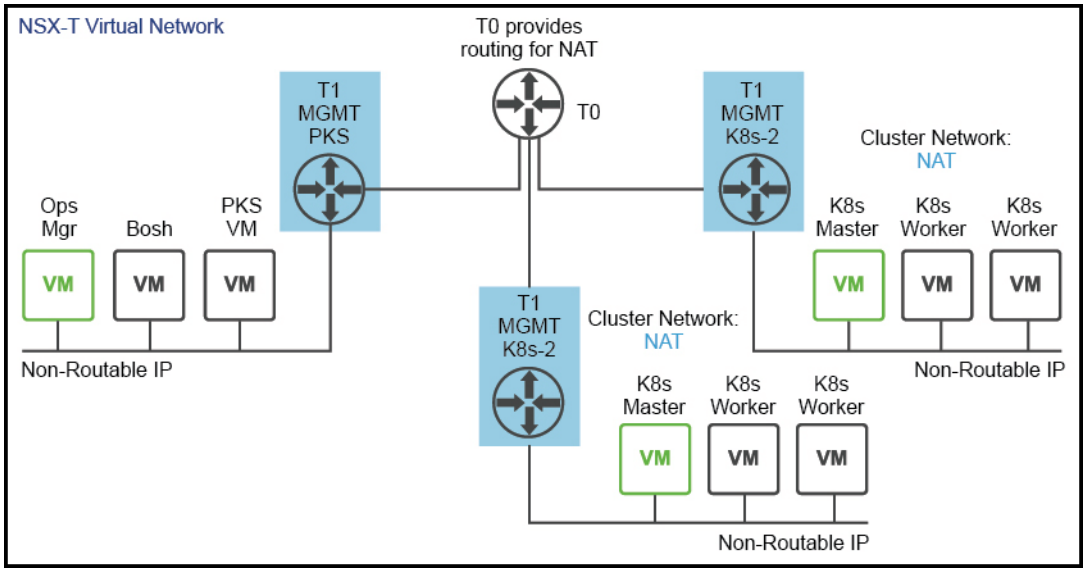
Deployment Topologies

Page last updated:

There are three supported topologies in which to deploy NSX-T with PKS.

NAT Topology

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



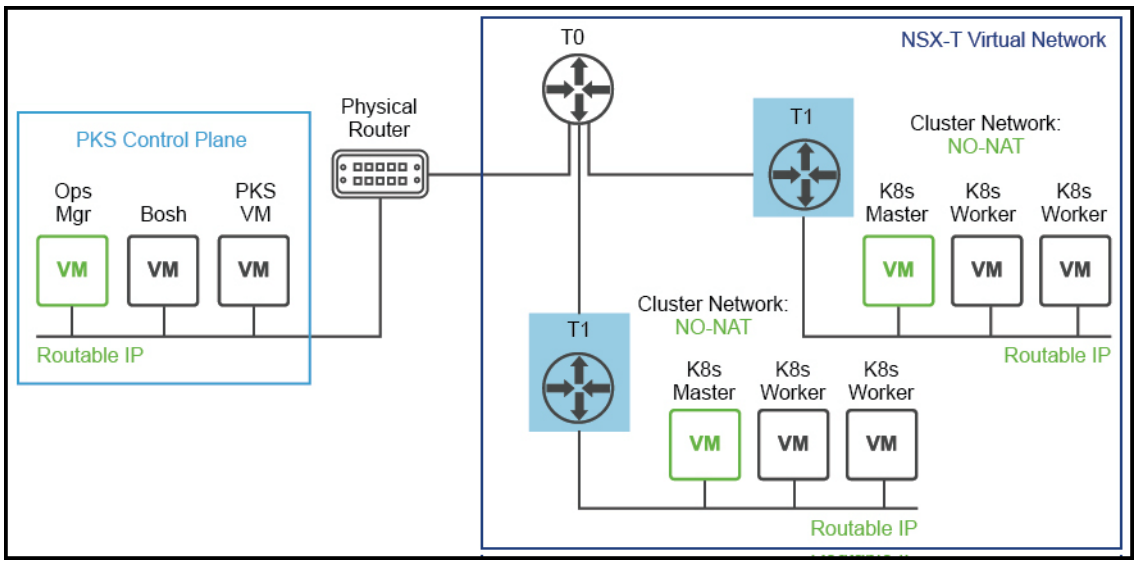
View a larger version of this image.

This topology has the following characteristics:

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

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

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



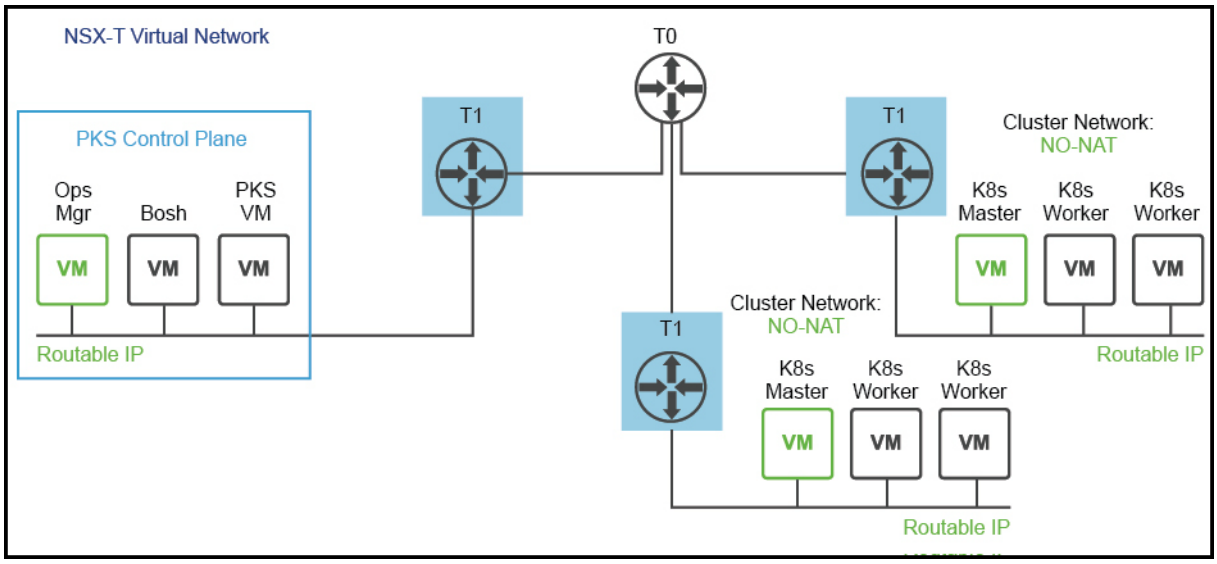
View a larger version of this image.

This topology has the following characteristics:

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

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

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



View a larger version of this image.

This topology has the following characteristics:

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

Note: PKS does not support the use of NSX-T edge clusters on bare metal.

Preparing NSX-T Before Deploying PKS

Page last updated:

Before you install Pivotal Container Service (PKS) on vSphere with NSX-T integration, you must prepare your NSX-T environment.

In addition to fulfilling the prerequisites specified in vSphere with NSX-T Prerequisites and Resource Requirements, follow the steps below.

Step 1: Plan for Network Subnets and IP Blocks

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

Plan Network CIDRs

Plan for the following network CIDRs in the IPv4 address space according to the instructions in the VMware NSX-T documentation.

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

Refer to the instructions in the VMware NSX-T documentation to ensure that your network topology enables the following communications:

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

> **Note**: Starting with PKS v1.1.5, NCP runs as a BOSH-managed process. See NSX-T Architectural Changes in the PKS v1.1.5 release notes for details.

Plan IP Blocks

You must plan IP blocks for the pods and nodes that are created when PKS creates the Kubernetes cluster. IP block sizes must be a multiple of 256. For example, /24. You must allocate subnets for the IP blocks before configuring the PKS tile. For more information, see Step 3.1: Create NSX Network Objects below.

Each Kubernetes cluster owns the /24 subnet. To deploy multiple Kubernetes clusters, set the **Nodes IP Block ID** in the **Networking** pane of the PKS tile to larger than /24. The recommended size is /16.

> **Note**: You can use a smaller nodes block size for no-NAT environments with a limited number of routable subnets. For example, /20 allows up to 16 Kubernetes clusters to be created.

You configure the **Pods IP Block ID** and **Nodes IP Block ID** in the **Networking** pane of the PKS tile. For more information, see Networking in *Installing PKS on vSphere with NSX-T*.

Reserved IP Blocks

Do not use any of the IP blocks listed in this section for pods or nodes. If you create Kubernetes clusters with any of the blocks listed below, the Kubernetes worker nodes cannot reach Harbor or internal Kubernetes services.

The Docker daemon on the Kubernetes worker node uses the subnet in the following CIDR range. Do not use IP addresses in the following CIDR range:

- 172.17.0.0/16

If PKS is deployed with Harbor, Harbor uses the following CIDR ranges for its internal Docker bridges. Do not use IP addresses in the following CIDR range:

- 172.18.0.0/16
- 172.19.0.0/16
- 172.20.0.0/16
- 172.21.0.0/16
- 172.22.0.0/16

Each Kubernetes cluster uses the following subnet for Kubernetes services. Do not use the following IP block for the Nodes IP Block:

- 10.100.200.0/24

Step 2: Deploy NSX-T

Deploy NSX-T according to the instructions in the VMware NSX-T documentation.

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

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

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

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

3.1: Create NSX Network Objects

1. Create two NSX IP pools. For more information, see Create an IP Pool for Tunnel Endpoint IP Addresses in the VMware NSX-T documentation. Configuration details for the NSX IP pools:
    - 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 in the VMware NSX-T documentation. Configuration details for the NSX TZs:
    - One NSX TZ for PKS control plane Services and Kubernetes Cluster deployment overlay networks named `tz-overlay` and the associated N-VDS `hs-overlay`. Select **Standard**.
    - One NSX TZ for NSX Edge uplinks (ingress/egress) for PKS Kubernetes clusters named `tz-vlan` and the associated N-VDS `hs-vlan`. Select **Standard**.

3. If the default uplink profile is not applicable in your deployment, create your own NSX uplink host profile. For more information, see Create an Uplink Profile in the VMware NSX-T documentation.

4. Create NSX Host Transport Nodes. For more information, see Create a Host Transport Node in the VMware NSX-T documentation. Configuration details:
    - For each host in the NSX-T Fabric, create a node named `tnode-host-NUMBER`. For example, if you have three hosts in the NSX-T Fabric, create three nodes named `tnode-host-1`, `tnode-host-2`, and `tnode-host-3`.
    - Add the `tz-overlay` NSX Transport Zone to each NSX Host Transport Node.

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

5. Create NSX IP blocks. We recommend that you use separate NSX IP blocks for the node networks and the pod networks. The subnets for both nodes and pods should have a size of 256 (/24). However, if you are planning to deploy multiple Kubernetes clusters, the nodes subnet size should be /16. For more information about planning IP blocks, see the Plan IP Blocks section above. For more information about creating NSX IP blocks in NSX Manager, see Manage IP Blocks in the VMware NSX-T documentation. Configuration details:
    - One NSX IP Block named `node-network-ip-block`. PKS uses this block to assign address space to Kubernetes master and worker nodes when new clusters are deployed or a cluster increases its scale.
    - One NSX IP Block named `pod-network-ip-block`. The NSX-T Container Plug-in (NCP) uses this block to assign address space to Kubernetes pods through the Container Networking Interface (CNI).

3.2: Create Logical Switches

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

    > **Note**: This network is required for the NAT deployment topology and No-NAT with Logical Switch deployment topology. If you are deploying the No-NAT with Virtual Switch topology, you can skip this step.

2. Attach your first NSX Logical Switch to the `tz-vlan` NSX Transport Zone.

3. Attach your second and third NSX Logical Switches to the `tz-overlay` NSX Transport Zone.

> **Note**: PKS v1.0 required you to manually create the `ls-pks-service` logical switch for the PKS service network. With PKS v1.1, the service network and switch are created for you by NSX-T. When you install PKS for the first time, you are prompted to specify the service network. Specify the management network in this field. For more information, see the Assign AZs and Networks section of the NSX-T installation documentation.

3.3: Create NSX Edge Objects

1. Create NSX Edge Transport Nodes. For more information, see Create an NSX Edge Transport Node in the VMware NSX-T documentation.

2. Add both `tz-vlan` and `tz-overlay` NSX Transport Zones to the NSX Edge Transport Nodes. Controller Connectivity and Manager Connectivity should be **UP**.

3. Refer to the MAC addresses of the Edge VM interfaces you deployed to deploy your virtual NSX Edges:

a. Connect the `hs-overlay` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the second NIC from your deployed Edge VM.
b. Connect the `hs-vlan` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the third NIC from your deployed Edge VM.

4. Create an NSX Edge cluster named `edge-cluster-pks`. For more information, see Create an NSX Edge Cluster in the VMware NSX-T documentation.

5. Add the NSX Edge Transport Nodes to the cluster.

3.4: Create Logical Routers

Create T0 Logical Router for PKS

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

1. Create a Tier-0 (T0) logical router named `t0-pks`. For more information, see Create a Tier-0 Logical Router in the VMware NSX-T documentation. Configuration details:

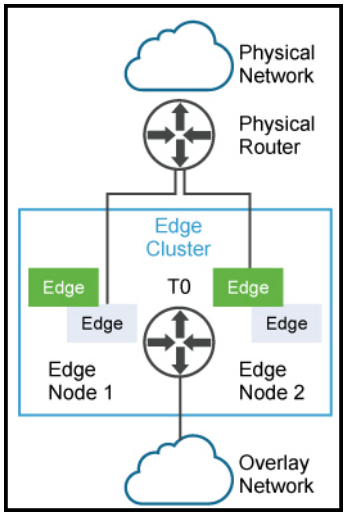
    - Select `edge-cluster-pks` for the cluster.
    - Set **High Availability Mode** to **Active-Standby**. NAT rules are be applied on T0 by NCP. If not set **Active-Standby**, the router does not support NAT rule configuration.

2. Attach the T0 logical router to the `ls-pks-uplink` logical switch you created previously. For more information, see Connect a Tier-0 Logical Router to a VLAN Logical Switch in the VMware NSX-T documentation. Create a logical router port for `ls-pks-uplink` and assign an IP address and CIDR that your environment uses to route to all PKS assigned IP pools and IP blocks.

3. Configure T0 routing to the rest of your environment using the appropriate routing protocol for your environment or by using static routes. For more information, see Tier-0 Logical Router in the VMware NSX-T documentation. The CIDR used in `ip-pool-vips` *must* route to the IP you just assigned to your t0 uplink interface.

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

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



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

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

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

Setting	First Uplink	Second Uplink
IP Address/Mask	uplink_1_ip	uplink_2_ip
URPF Mode	None (optional)	None (optional)

Transport Node	edge-TN1	edge-TN2
LS	uplink-LS1	uplink-LS1

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

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

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

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

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

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

Create T1 Logical Router for PKS Management VMs

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

    > **Note**: Skip this step if you are deploying the No-NAT with Virtual Switch topology. This Logical Router is required for the NAT deployment topology and No-NAT with Logical Switch deployment topology.

2. Create a logical router port for `ls-pks-mgmt` and assign the following CIDR block: `10.172.1.0/28`. For more information, see Connect a Tier-0 Logical Router to a VLAN Logical Switch in the VMware NSX-T documentation.

3. Configure route advertisement on the T1 as follows. For more information, see Configure Route Advertisement on a Tier-1 Logical Router in the VMware NSX-T documentation. Configuration details:
    - Enable **Status**.
    - Enable **Advertise All NSX Connected Routes**.
    - Enable **Advertise All NAT Routes**.
    - Enable **Advertise All LB VIP Routes**.

Configure NAT Rules for PKS Management VMs

> **Note**: This step applies to the NAT Topology only. Skip this step for No-NAT with Virtual Switch (VSS/VDS) Topology and No-NAT with Logical Switch (NSX-T) Topology.

Create the following NAT rules for the Manangement T0. For more information, see Tier-0 NAT in the VMware NSX-T documentation. Configuration details:

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

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

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

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

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

Deploying Ops Manager on vSphere with NSX-T

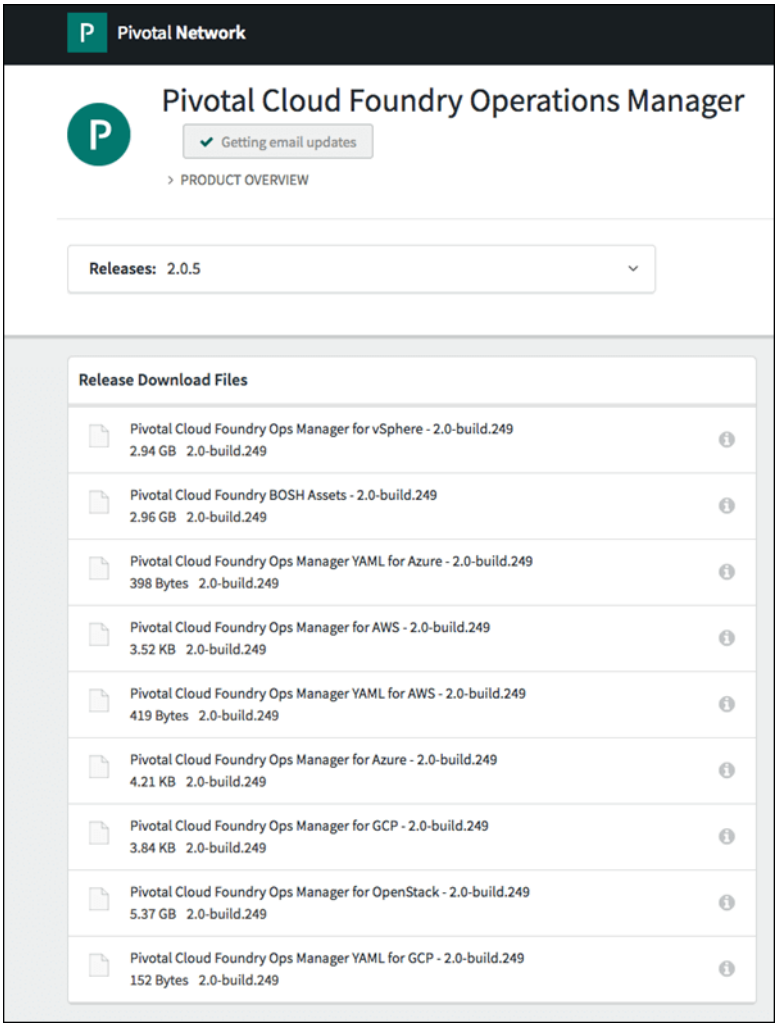
Page last updated:

This topic provides instructions for deploying Ops Manager on VMware vSphere with NSX-T integration.

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.

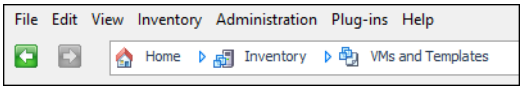
1. Before starting, review the known issues in PCF Ops Manager Release v2.1 Release Notes.

2. Download the Pivotal Cloud Foundry (PCF) Ops Manager .ova file at Pivotal Network. Click the **Pivotal Cloud Foundry** region to access the PCF product page. Use the dropdown menu to select an Ops Manager release.

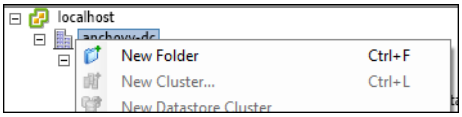


3. Log into vCenter.

4. Select the **VM and Templates** view.

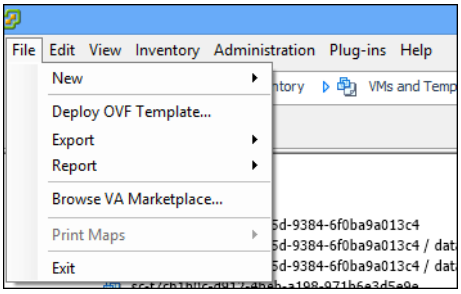


5. Right click on your datacenter and select **New Folder**.

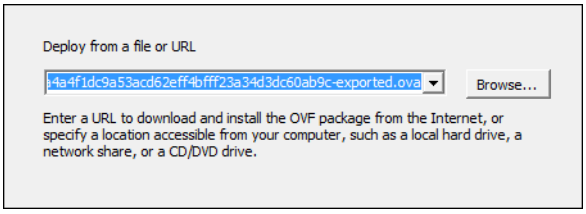


6. Name the folder `pivotal_cf` and select it.

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



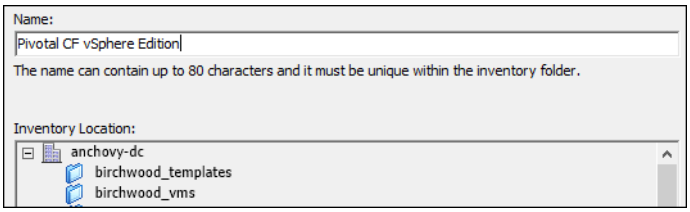
8. Select the .ova file and click **Next**.



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

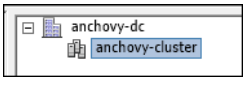
10. Accept the license agreement and click **Next**.

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



> **Note**: The selected folder is the one you created.

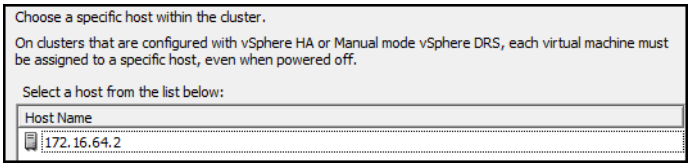
12. Select a vSphere cluster and click **Next**.



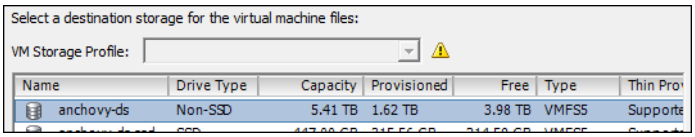
13. If prompted, select a resource pool and click **Next**.

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

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

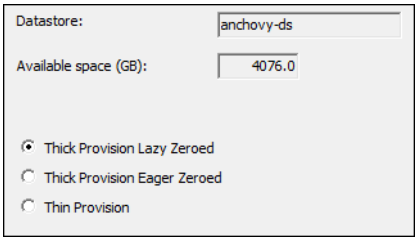


15. Select a storage destination and click **Next**.

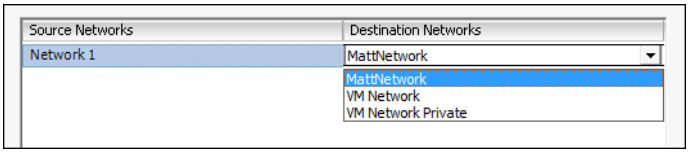


16. Select a disk format and click **Next**. For more information about disk formats, see Provisioning a Virtual Disk in vSphere.

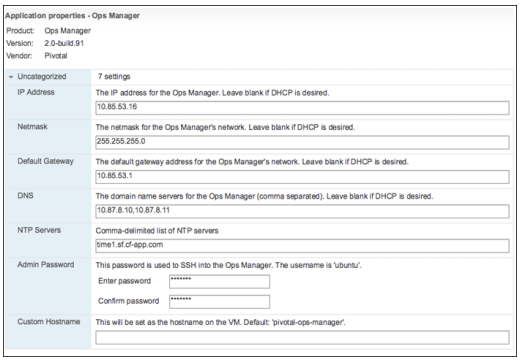
> **Warning**: Ops Manager v2.1 requires a Director VM with at least 8 GB memory.



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

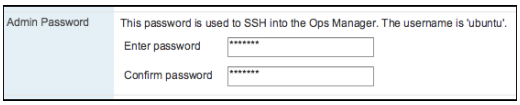


18. Enter network information and passwords for the Ops Manager VM admin user.



> **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.



20. Click **Next**.

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

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

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

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

Next Steps

After you complete this procedure, follow the instructions in Configuring Ops Manager on vSphere with NSX-T Integration.

Configuring Ops Manager on vSphere with NSX-T Integration

Page last updated:

This topic describes how to configure Ops Manager for vSphere with NSX-T integration.

Before you begin this procedure, ensure that you have successfully completed all of the steps in Deploying Ops Manager on vSphere with NSX-T.

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

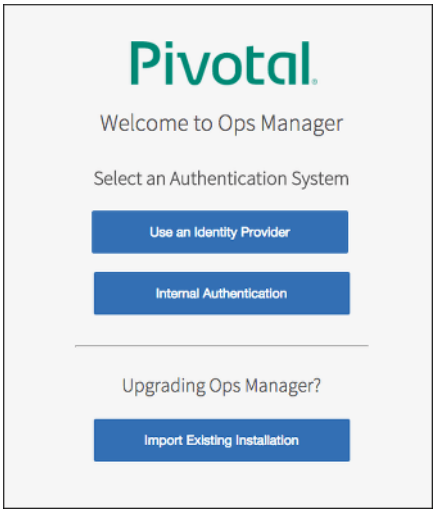
Note: If you have Pivotal Application Service (PAS) installed, we strongly recommend installing PKS on a separate instance of Ops Manager for security reasons. For more information, see PAS and PKS Deployments with Ops Manager.

1. Navigate to the fully qualified domain of your Ops Manager in a web browser.

   Note: If you are using the NAT deployment topology, you must have already deployed Ops Manager to the `ls-pks-mgmt` NSX logical switch by following the instructions in Create T1 Logical Router for PKS Management VMs in *Preparing NSX-T Before Deploying PKS*. Use the DNAT IP address to access Ops Manager.

2. The first time you start Ops Manager, you must choose one of the following:

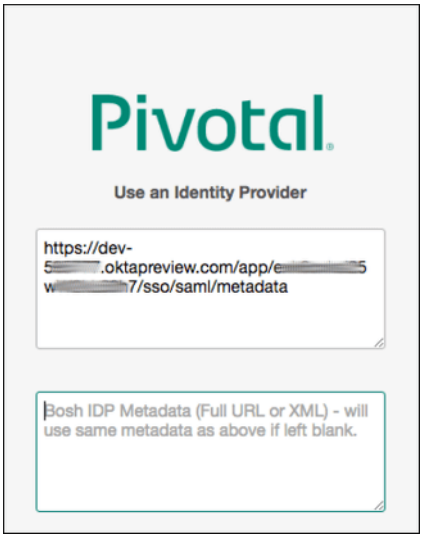
   - **Use an Identity Provider**: If you use an Identity Provider (IdP), an external identity server maintains your user database.
   - **Internal Authentication**: If you use Internal Authentication, PCF maintains your user database.



Use an Identity Provider

1. Log in to your IdP console and download the IdP metadata XML. Optionally, if your IdP supports metadata URL, you can copy the metadata URL instead of the XML.

2. Copy the IdP metadata XML or URL to the Ops Manager **Use an Identity Provider** log in page.



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

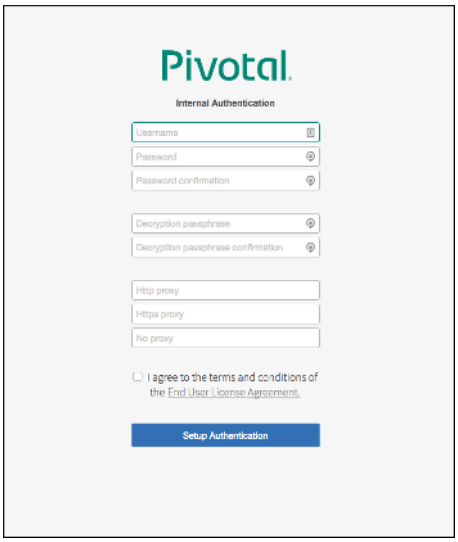
3. Enter your **Decryption passphrase**. Read the **End User License Agreement**, and select the checkbox to accept the terms.
4. Your Ops Manager log in page appears. Enter your username and password. Click **Login**.
5. Download your SAML Service Provider metadata (SAML Relying Party metadata) by navigating to the following URLs:
    - **5a.** Ops Manager SAML service provider metadata: `https://OPS-MAN-FQDN:443/uaa/saml/metadata`
    - **5b.** BOSH Director SAML service provider metadata: `https://BOSH-IP-ADDRESS:8443/saml/metadata`

    Note: To retrieve your `BOSH-IP-ADDRESS`, navigate to the **Ops Manager Director** tile > **Status** tab. Record the **Ops Manager Director** IP address.
6. Configure your IdP with your SAML Service Provider metadata. Import the Ops Manager SAML provider metadata from Step 5a above to your IdP. If your IdP does not support importing, provide the values below.
    - **Single sign on** URL: `https://OPS-MAN-FQDN:443/uaa/saml/SSO/alias/OPS-MAN-FQDN`
    - **Audience URI** (SP Entity ID): `https://OP-MAN-FQDN:443/uaa`
    - **Name ID**: Email Address
    - SAML authentication requests are always signed
7. Import the BOSH Director SAML provider metadata from Step 5b to your IdP. If the IdP does not support an import, provide the values below.
    - **Single sign on** URL: `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
    - **Audience URI** (SP Entity ID): `https://BOSH-IP:8443`
    - **Name ID**: Email Address
    - SAML authentication requests are always signed
8. Return to the **Ops Manager Director** tile, and continue with the configuration steps below.

Internal Authentication

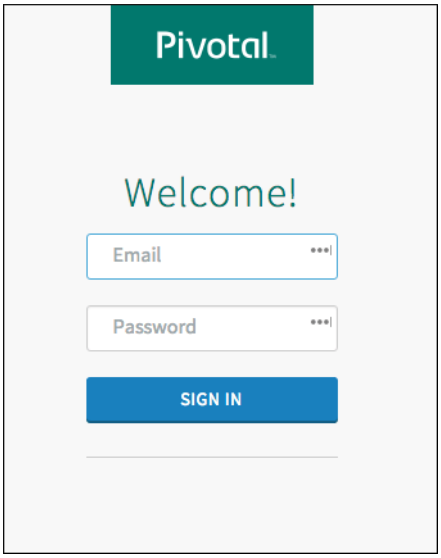
On the **Internal Authentication** page, complete the following steps:

1. Enter a **Username**, **Password**, and **Password confirmation** to create an Admin user.
2. Enter a **Decryption passphrase** and the **Decryption passphrase confirmation**. This passphrase encrypts the Ops Manager datastore, and is not recoverable.
3. If you use an **HTTP proxy** or **HTTPS proxy**, follow the instructions in Configuring Proxy Settings for the BOSH CPI in the Pivotal Cloud Foundry documentation.
4. Read the **End User License Agreement**, and select the checkbox to accept the terms.

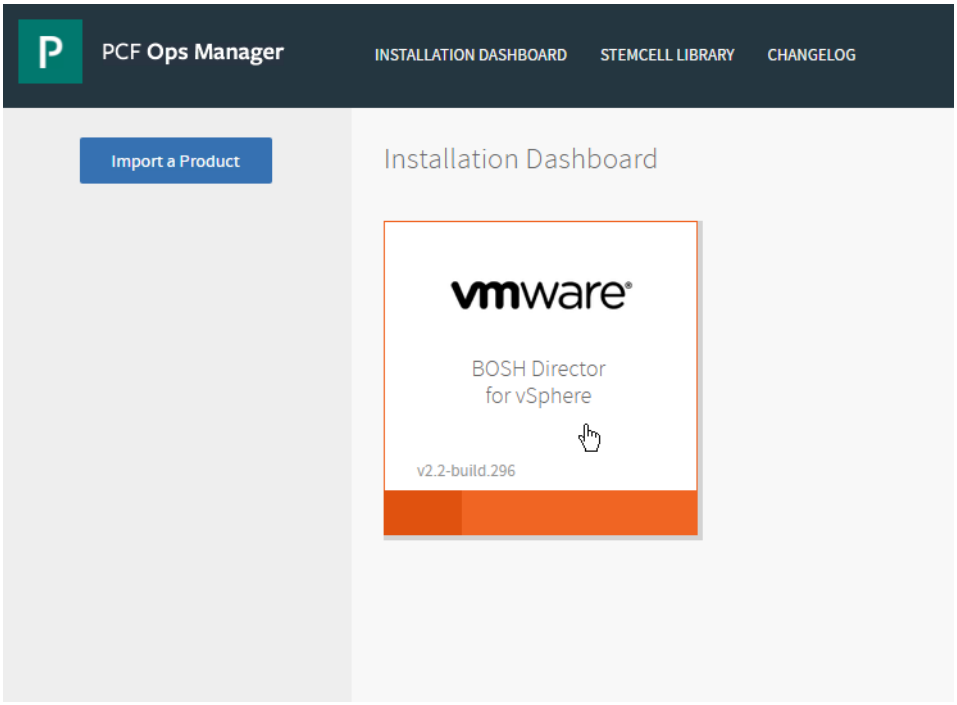


Step 2: vCenter Config Page

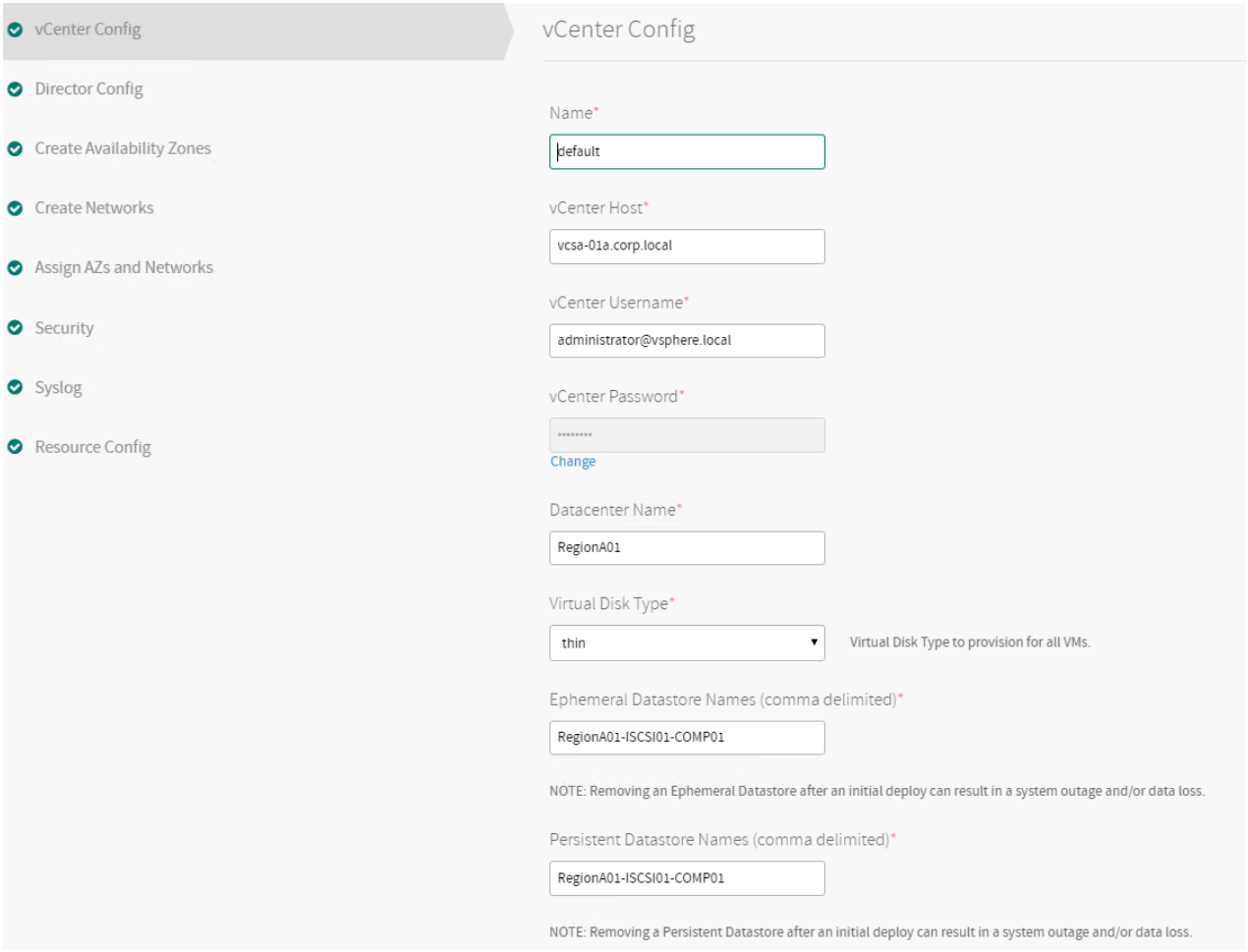
1. Log in to Ops Manager with the Admin username and password you created in the previous step.



2. Click the **BOSH Director for vSphere** tile.



3. Select **vCenter Config**.

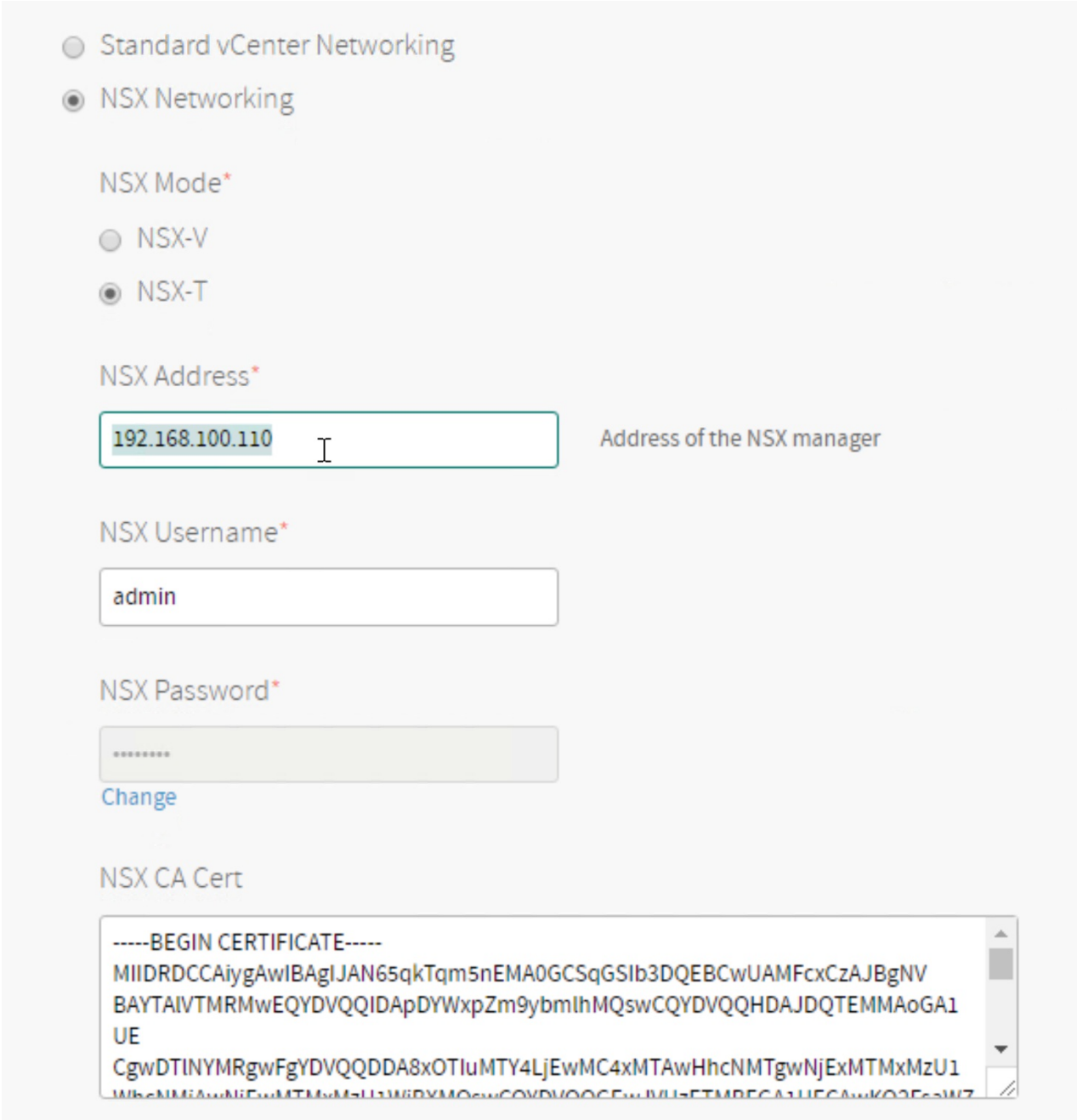


4. Enter the following information:

    - **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.

5. Select **NSX Networking**, then select **NSX-T**.



6. Configure NSX-T networking as follows:

- **NSX Address**: Enter the IP address of the NSX Manager host.
- **NSX Username** and **NSX Password**: Enter the NSX Manager username and password.
- (Optional) **NSX CA Cert**: Provide a CA certificate in PEM format that authenticates to the NSX server. If the NSX Manager generated a self-signed certificate, you can retrieve the CA certificate using OpenSSL with the command `openssl s_client -host NSX-ADDRESS -port 443 -prexit -showcerts`.

7. Configure the following folder names:

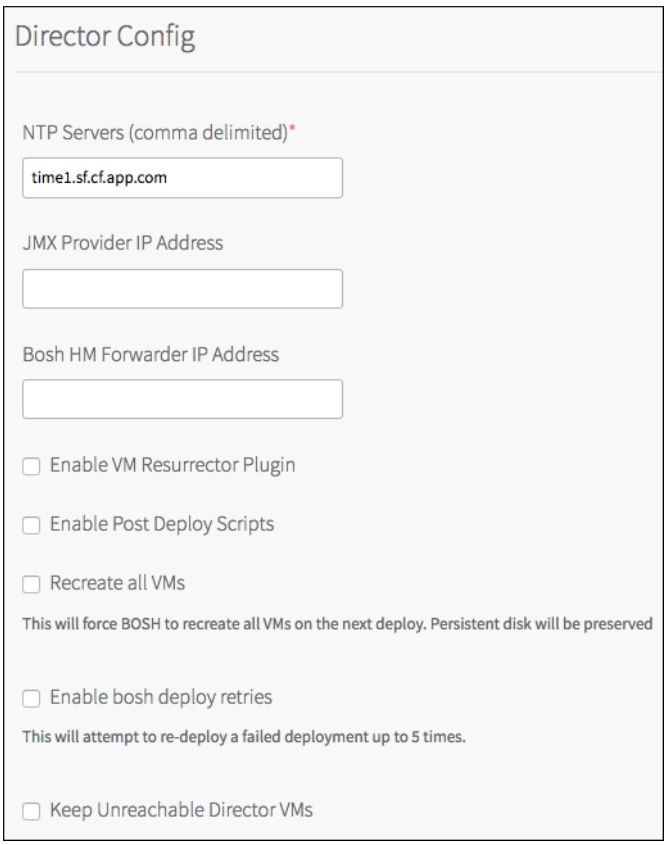
- **VM Folder**: The vSphere datacenter folder where Ops Manager places VMs. Enter `pks_vms`.
- **Template Folder**: The vSphere datacenter folder where Ops Manager places VMs. Enter `pks_templates`.
- **Disk path Folder**: The vSphere datastore folder where Ops Manager creates attached disk images. You must not nest this folder. Enter `pks_disk`.

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

8. Click **Save**.

Step 3: Director Config Page

1. Select **Director Config**.



2. In the **NTP Servers (comma delimited)** field, enter your NTP server addresses.

   Note: The NTP server configuration only updates after VM recreation. Ensure that you select the **Recreate all VMs** checkbox if you modify the value of this field.

3. Leave the **JMX Provider IP Address** field blank.

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

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

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

5. Select the **Enable VM Resurrector Plugin** to enable BOSH 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.

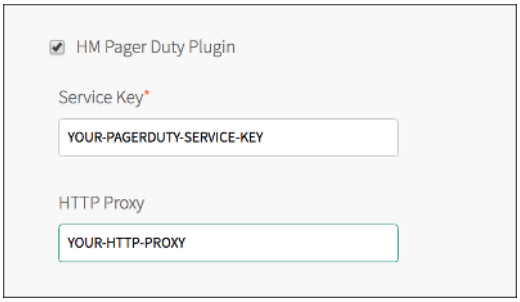
7. Select **Recreate all VMs** to force BOSH to recreate all VMs on the next deploy. This process does not destroy any persistent disk data.

8. Select **Enable bosh deploy retries** if you want Ops Manager to retry failed BOSH operations up to five times.

   Note: If you use Ops Manager v2.2, Pivotal recommends disabling **Allow Legacy Agents**. Disabling the field allows Ops Manager to implement TLS secure communications.

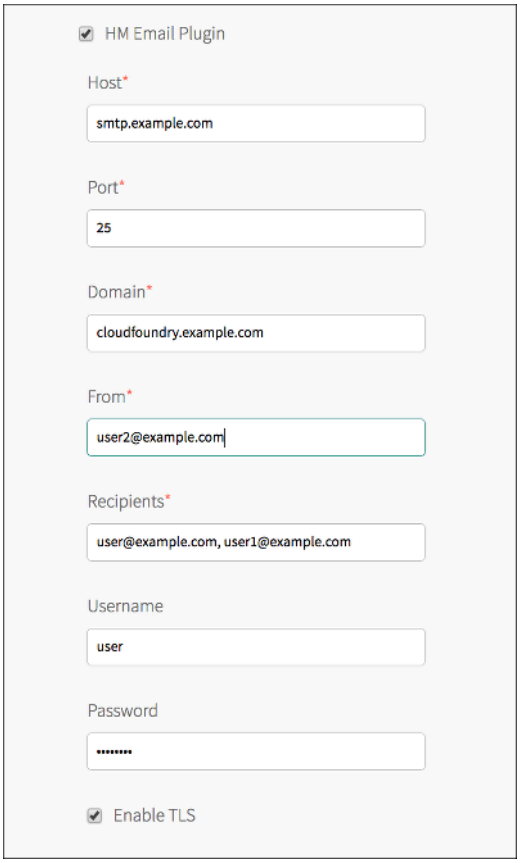
9. Select **Keep Unreachable Director VMs** if you want to preserve Ops Manager Director VMs after a failed deployment for troubleshooting purposes.

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

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

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



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

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

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

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

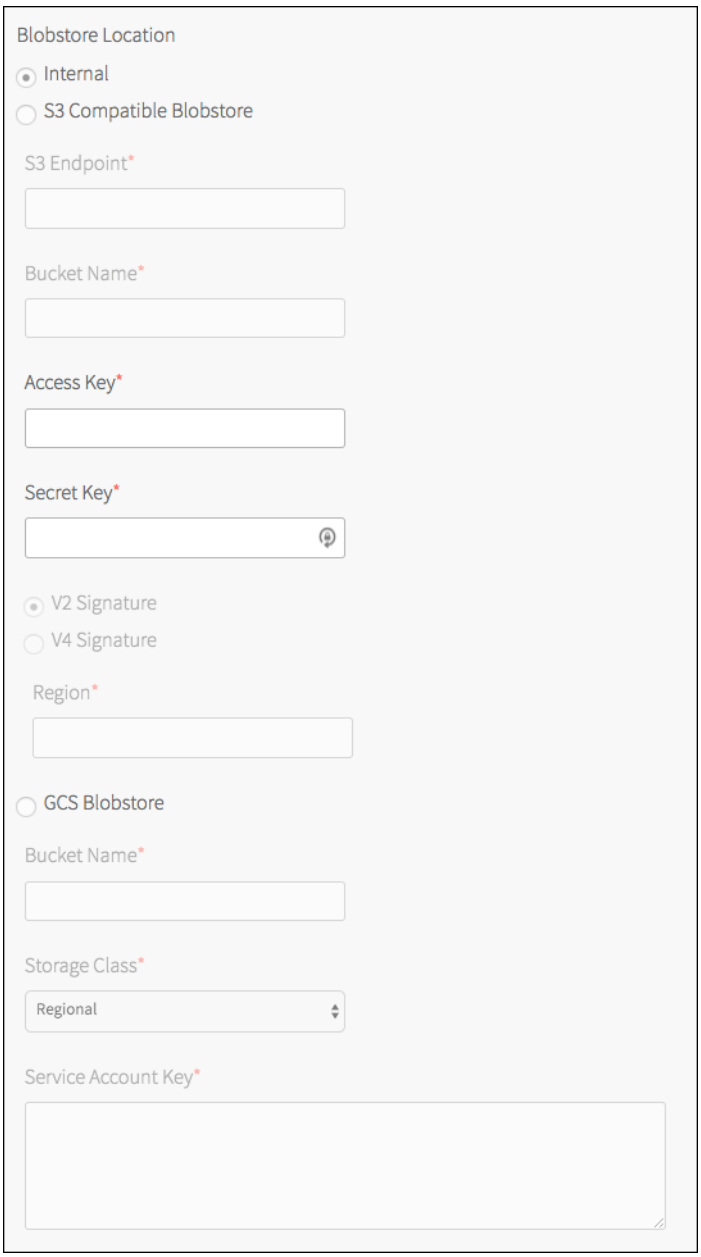
    1. **S3 Endpoint**: Navigate to the Regions and Endpoints topic in the AWS documentation. Locate the endpoint for your region in the **Amazon Simple Storage Service (S3)** table and construct a URL using your region's endpoint. For example, if you are using the `us-west-2` region, the URL you create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

    > **Note**: AWS recommends using Signature Version 4. For more information about AWS S3 Signatures, see Authenticating Requests in the AWS documentation.

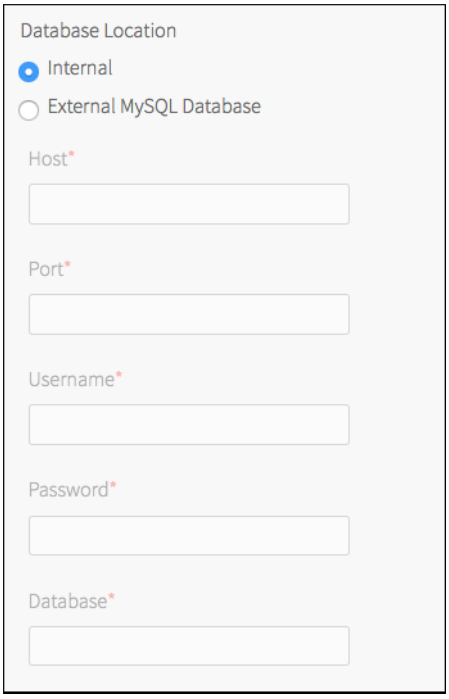
- **GCS Blobstore**: Select this option to use an external Google Cloud Storage (GCS) endpoint. To create a GCS bucket, you will need a GCS account. Follow the procedures in Creating Storage Buckets in the GCP documentation. Once you have created a GCS bucket, complete the following steps:

    1. **Bucket Name**: Enter the name of your GCS bucket.

2. **Storage Class**: Select the storage class for your GCS bucket. For more information, see Storage Classes in the GCP documentation.
3. **Service Account Key**: Follow the steps in the Set up an IAM Service Account section of *Preparing GCP* to download a JSON file with a private key, and then enter the contents of the JSON file into the field.



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



14. (Optional) **Director Workers** sets the number of workers available to execute Director tasks. This field defaults to 5.

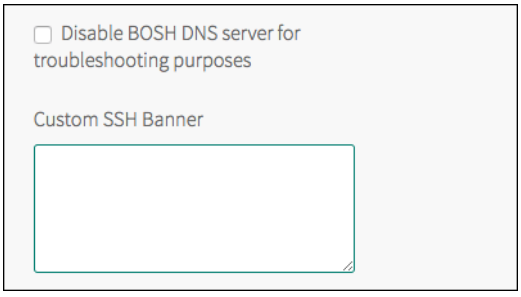
15. (Optional) **Max Threads** sets the maximum number of threads that the Ops Manager Director can run simultaneously. For vSphere, the default value is 32. Leave the field blank to use this default value. Pivotal recommends that you use the default value unless doing so results in rate limiting or errors on your IaaS.

16. Leave the **Director Hostname** field blank.

17. Ensure the **Disable BOSH DNS server for troubleshooting purposes** checkbox is not selected.

    WARNING: Do not disable BOSH DNS if you are deploying PKS.

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.



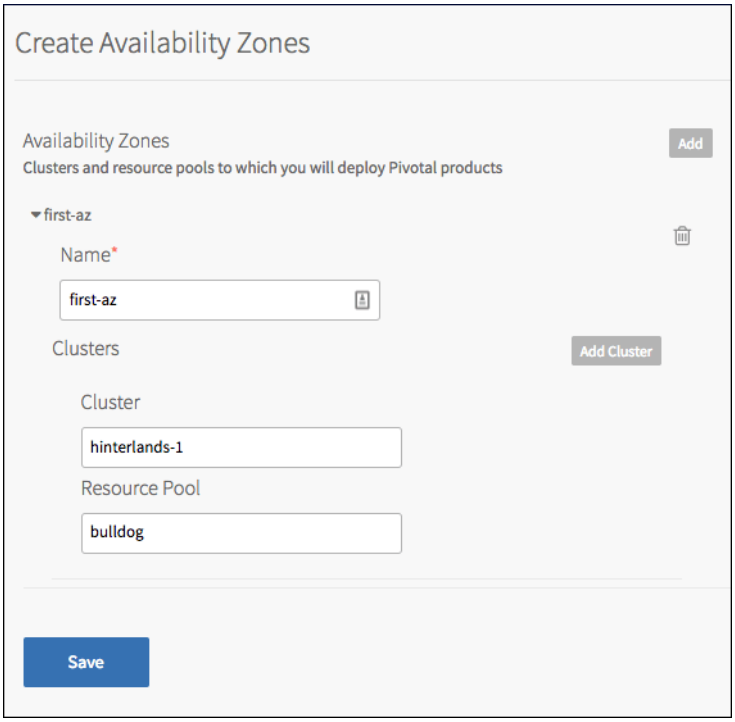
19. Click **Save**.

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

Step 4: Create Availability Zone Page

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

1. Select **Create Availability Zones**.



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

    > **Note**: For more information about using availability zones in vSphere, see Understanding Availability Zones in VMware Installations in the PCF documentation.

3. Click **Save**.

Step 5: Create Networks Page

1. Select **Create Networks**.

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

3. Click **Add Network** and create the following networks:
    - `pks-infrastructure`: Network for Ops Manager, the BOSH Director, the PKS broker, and the PKS API. The `pks-infrastructure` network maps to the NSX logical switch named `ls-pks-mgmt` created for the PKS Management Network in Create the NSX-T Objects Required for PKS in *Preparing NSX-T Before Deploying PKS*. If you have a large deployment with multiple tiles, you can deploy the PKS broker and PKS API to a separate network named `pks-main`. See the table below for more information.
    - `pks-services`: Netwok for creating the master and worker VMs for Kubernetes clusters. The CIDR should not conflict with the pod overlay network `10.200.0.0/16` or the reserved Kubernetes services CIDR of `10.100.200.0/24`.

    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	`pks-infrastructure`
	vSphere Network Name	`MY-PKS-virt-net/MY-PKS-subnet-infrastructure`

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

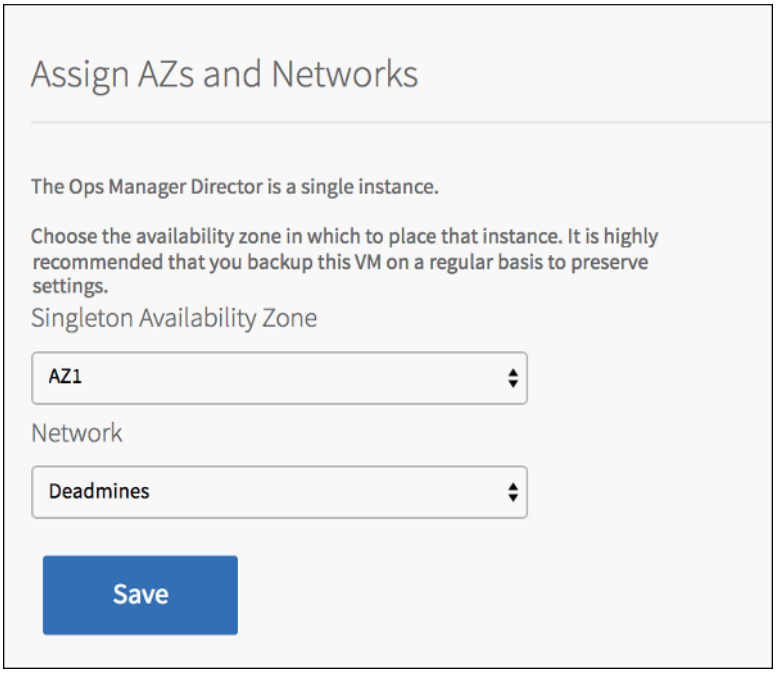
4. Select which **Availability Zones** to use with the network.
5. Click **Save**.

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

Note: If you use the Cisco Nexus 1000v Switch, see more information in Using the Cisco Nexus 1000v Switch with Ops Manager in the PCF documentation.

Step 6: Assign AZs and Networks Page

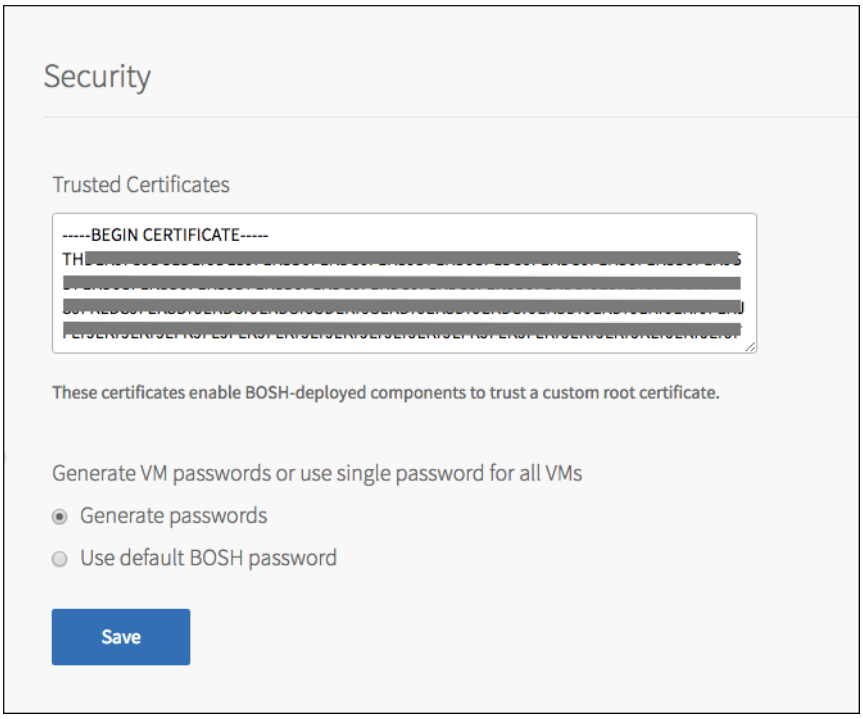
1. Select **Assign AZs and Networks**.



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

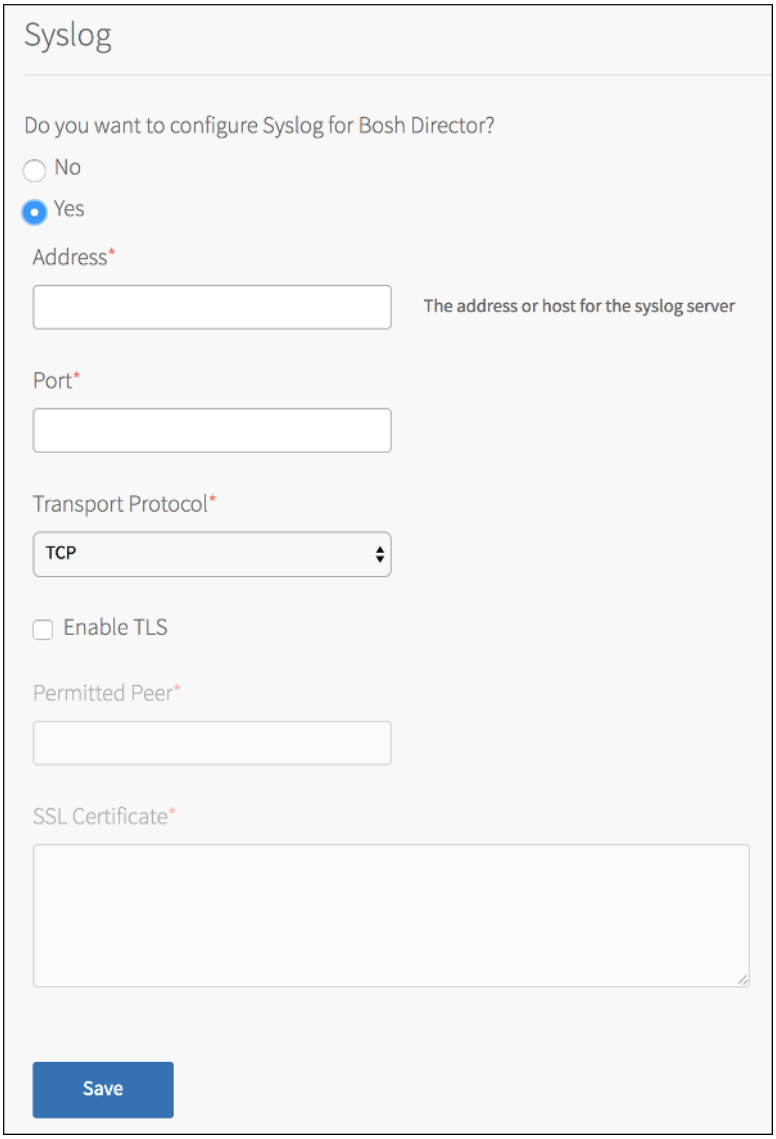
1. Select **Security**.



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

Step 8: Syslog Page

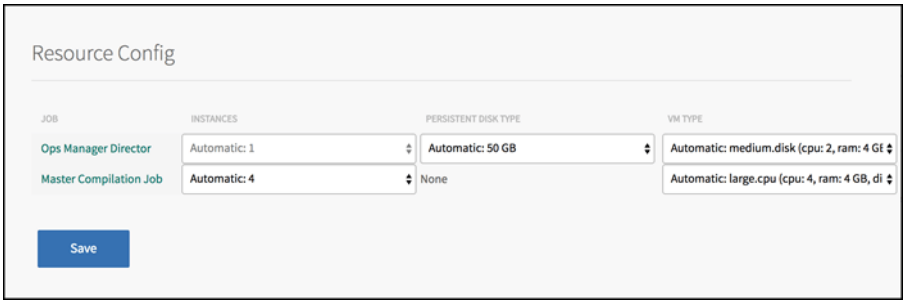
1. Select **Syslog**.



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 **RELP**. This selection determines which transport protocol is used to send the logs to the remote server.
6. (Optional) Mark the **Enable TLS** checkbox to use TLS encryption when sending logs to the remote server.
    - In the **Permitted Peer** field, enter either the name or SHA1 fingerprint of the remote peer.
    - In the **SSL Certificate** field, enter the SSL certificate for the remote server.
7. Click **Save**.

Step 9: Resource Config Page

1. Select **Resource Config**.



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

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

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

3. Click **Save**.

Step 10: Complete the Ops Manager Installation

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

Next Steps

To generate and register certificates for PKS and the NSX Manager, perform the procedures in Generating and Registering Certificates.

To install PKS on vSphere with NSX-T integration, perform the procedures in Installing PKS on vSphere with NSX-T Integration.

To integrate VMware Harbor Registry with PKS to store and manage container images, see Integrating VMware Harbor Registry with PKS.

Generating and Registering Certificates

Page last updated:

This topic describes generating and registering certificates for PKS and the NSX Manager.

Before you install PKS on NSX-T, you must create two certificates that you will provide in the **Networking** pane in the PKS tile. For more information, see Networking in *Installing PKS on vSphere with NSX-T Integration*.

NSX Manager Super User Principal Identity Certificate

This certificate represents a principal identity with super user permissions that the PKS VM will use to communicate with NSX-T to manage (create, delete, and modify) node networking resources. During PKS installation on NSX-T, you will need to provide this in the NSX Manager Super User Principal Identity Certificate field on the **Networking** pane in the PKS tile.

You can complete the following steps from the Ops Manager VM or from any other Linux VM. This procedure does not work on Mac OS.

Step 1: Export Environment Variables

Export the following environment variables:

```
NSX_MANAGER="<NSX_MANAGER_IP>"
NSX_USER="<NSX_MANAGER_USERNAME>"
NSX_PASSWORD='<NSX_MANAGER_PASSWORD>'
PI_NAME="pks-nsx-t-superuser"
NSX_SUPERUSER_CERT_FILE="pks-nsx-t-superuser.crt"
NSX_SUPERUSER_KEY_FILE="pks-nsx-t-superuser.key"
NODE_ID=$(cat /proc/sys/kernel/random/uuid)
```

Step 2: Create the Super User Principal Identity Certificate

Create the Super User Principal Identity Certificate using a script or by clicking **Generate RSA Certificate** on the **Networking** tab in the PKS tile. For more information, see Networking in *Installing PKS on vSphere with NSX-T Integration*.

Create Certificate Using a Script

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

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

Create Certificate from the Networking Tab

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

1. Navigate to the **Networking** tab in the PKS tile. For more information, see Networking in *Installing PKS on vSphere with NSX-T Integration*.

2. Click **Generate RSA Certificate** and provide a wildcard domain, for example, `*.nsx.pks.vmware.local`.
3. In the Ops Manager or Linux VM where the subsequent scripts will run, create a file named `pks-nsx-t-superuser.crt` and copy the generated certificate into it.
4. In the Ops Manager or Linux VM where the subsequent scripts will run, create a file named `pks-nsx-t-superuser.key` and copy the private key into it.

Step 3: Register the Certificate

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

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

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

The response includes the `CERTIFICATE_ID` value.

Step 4: Register the Principal Identity

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

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

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

Step 5: Verify the Certificate and Key

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

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

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

NSX Manager CA Certificate

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

Step 1: Generate a Self-Signed Certificate

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

1. Create a file for the certificate request parameters named `nsx-cert.cnf`.
2. Copy the following parameters and paste them into the file, replacing `NSX-MANAGER-IP-ADDRESS` with the IP address of your NSX Manager, and `NSX-MANAGER-COMMONNAME` with the FQDN of the NSX Manager host:

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

   For example:

```
[ req ]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
prompt = no
[ req_distinguished_name ]
countryName = US
stateOrProvinceName = California
localityName = Palo-Alto
organizationName = NSX
commonName = nsxmgr-01a.example.com
[ req_ext ]
subjectAltName=DNS:nsxmgr-01a.example.com,IP:192.0.2.40
```

3. Export the `NSX_MANAGER_IP_ADDRESS` and `NSX_MANAGER_COMMONNAME` environment variables using the IP address of your NSX Manager and the FQDN of the NSX Manager host.

   For example:

```
$ export NSX_MANAGER_IP_ADDRESS=192.0.2.40
$ export NSX_MANAGER_COMMONNAME=nsxmgr-01a.example.com
```

4. Generate the certificate using openssl. Run the following command:

```
$ openssl req -newkey rsa:2048 -x509 -nodes \
-keyout nsx.key -new -out nsx.crt -subj /CN=$NSX_MANAGER_COMMONNAME \
-reqexts SAN -extensions SAN -config <(cat ./nsx-cert.cnf \
 <(printf "[SAN]\nsubjectAltName=DNS:$NSX_MANAGER_COMMONNAME,IP:$NSX_MANAGER_IP_ADDRESS")) -sha256 -days 365
```

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

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

Step 2: Register the Certificate with NSX Manager

1. Log into the NSX Manager UI.

2. Import the certificate by copying `nsx.crt` and `nsx.key`. For instructions, see Import a Certificate in the NSX-T documentation.

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

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

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

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

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

Next Steps

To install PKS on vSphere with NSX-T integration, perform the procedures in Installing PKS on vSphere with NSX-T Integration.

To integrate VMware Harbor Registry with PKS to store and manage container images, see Integrating VMware Harbor Registry with PKS.

Installing PKS on vSphere 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.

Prerequisites

Before performing the procedures in this topic, you must have deployed and configured Ops Manager. For more information, see vSphere with NSX-T Prerequisites and Resource Requirements.

If you use 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. Open the **Director Config** pane.
3. Select the **Enable Post Deploy Scripts** checkbox.
4. Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
5. Click the **Installation Dashboard** link to return to the Installation Dashboard.
6. Click **Apply Changes**.

Step 1: Install PKS

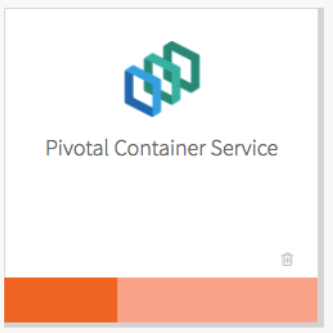
To install PKS, do the following:

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.

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



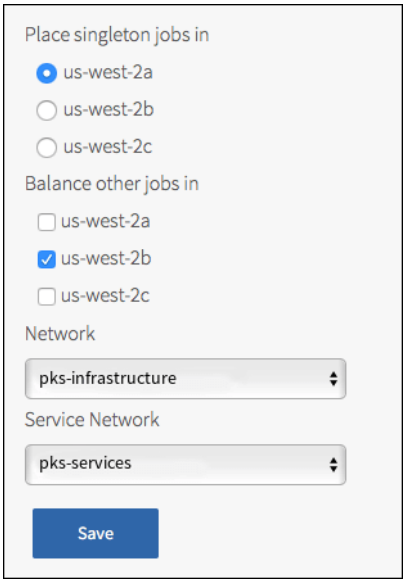
Assign AZs and Networks

Perform the following steps:

1. Click **Assign AZs and Networks**.

2. Select the availability zone (AZ) where you want to deploy the PKS API VM as a singleton job.

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



3. Under **Network**, select the PKS Management Network linked to the `ls-pks-mgmt` NSX-T logical switch you created in the Create Networks Page step of *Configuring Ops Manager on vSphere with NSX-T Integration*. This will provide network placement for the PKS API VM.

4. Under **Service Network**, your selection depends on whether you are upgrading from a previous PKS version or installing an original PKS deployment.

    - If you are deploying PKS with NSX-T for the first time, the **Service Network** field does not apply because PKS instructs NSX-T to create a new service network on-demand each time a new Kubernetes cluster is requested. However, the PKS tile requires you to make a selection. Therefore, select the same network you specified in the **Network** field.
    - If you are upgrading from a previous PKS version, select the **Service Network** linked to the `ls-pks-service` NSX-T logical switch that is created by PKS during installation. The service network provides network placement for the already existing on-demand Kubernetes cluster service instances created by the PKS broker.

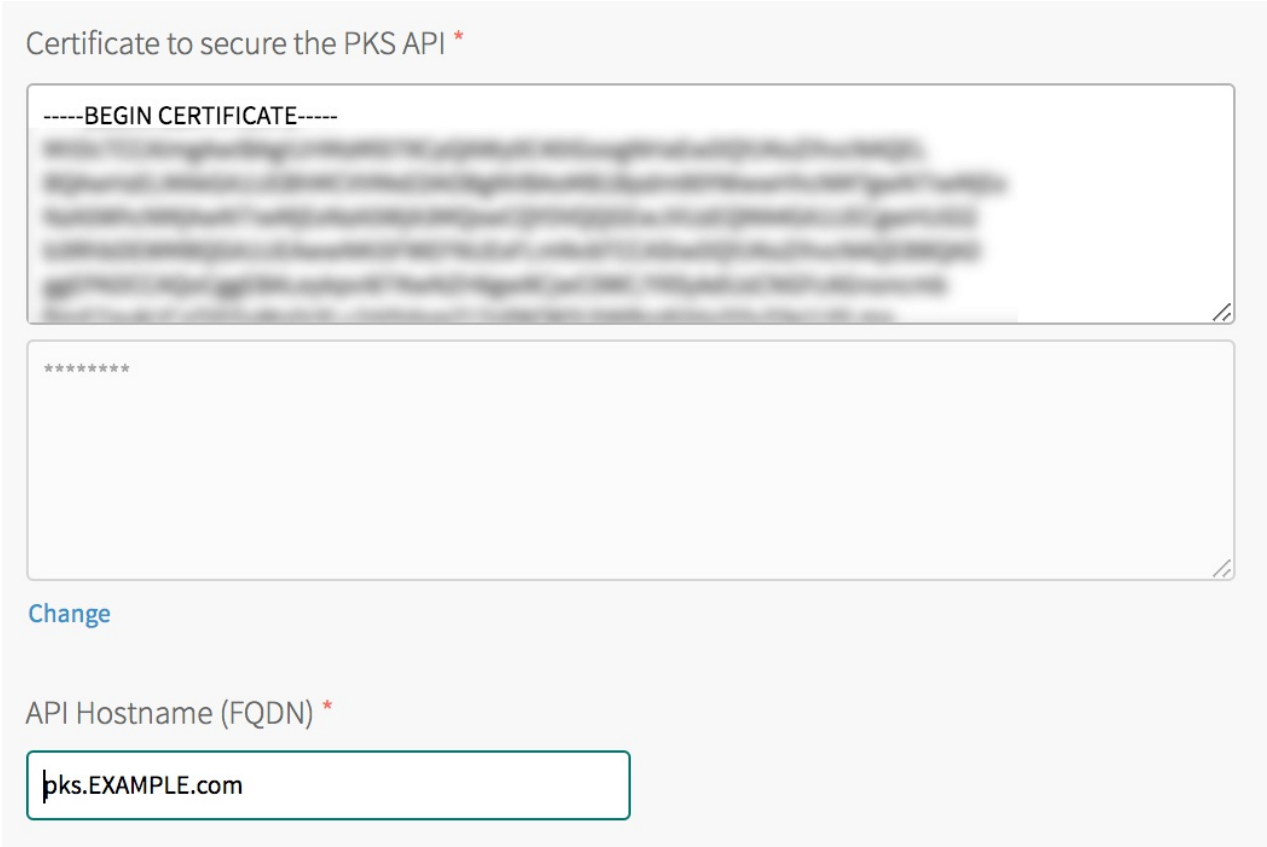
5. Click **Save**.

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 that you supply should cover the domain that routes to the PKS API VM with TLS termination on the ingress.

If you do not have a certificate and private key pair, PKS can generate one for you by performing the following steps.

> **Note**: If you configured Ops Manager Front End without a certificate, you can use this new certificate to complete Ops Manager configuration. To configure your Ops Manager Front End certificate, see Configure Front End.

   a. Select the **Generate RSA Certificate** link.
   b. Enter the wildcard domain for your API hostname. For example, if your PKS API domain is `api.pks.example.com`, then enter `*.pks.example.com`.
   c. Click **Generate**.

3. Under **API Hostname (FQDN)**, enter a fully qualified domain name (FQDN) to access the PKS API. For example, `api.pks.example.com`.

4. Click **Save**.

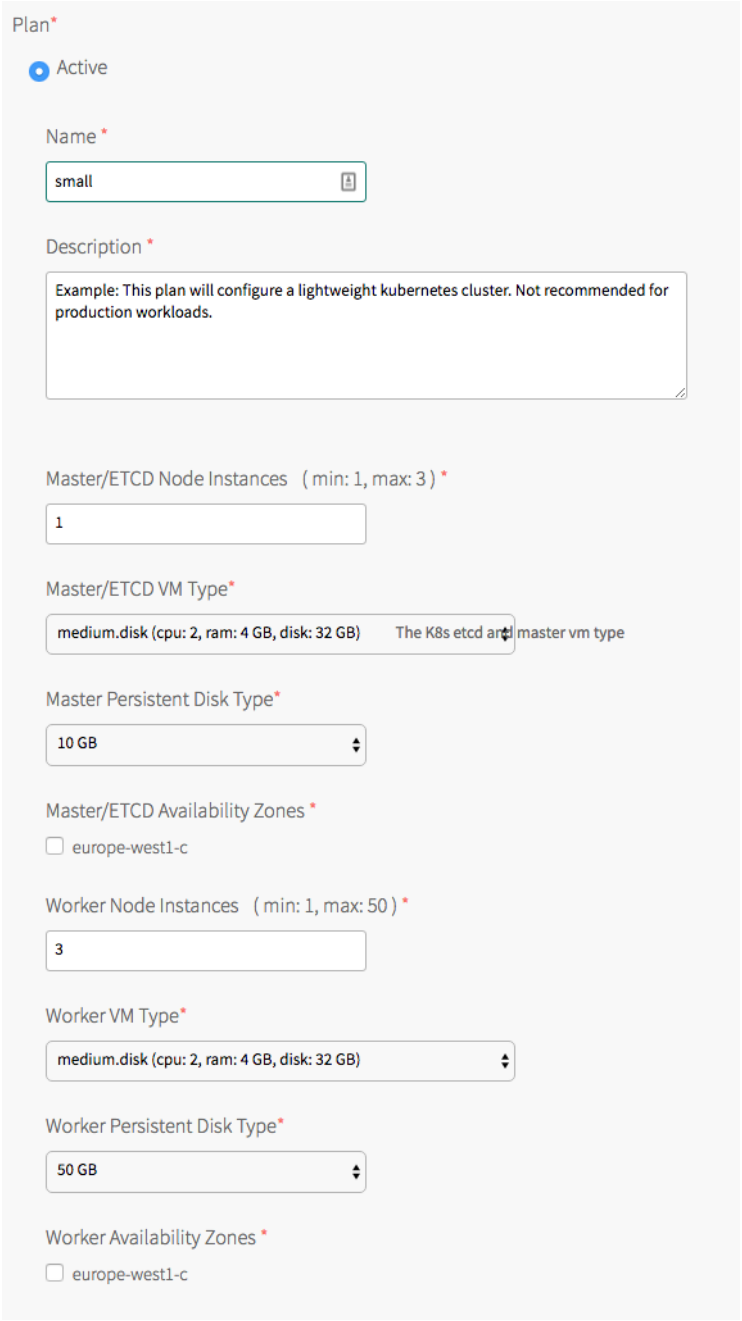
Plans

To activate a plan, perform the following steps:

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

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

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



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

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

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

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

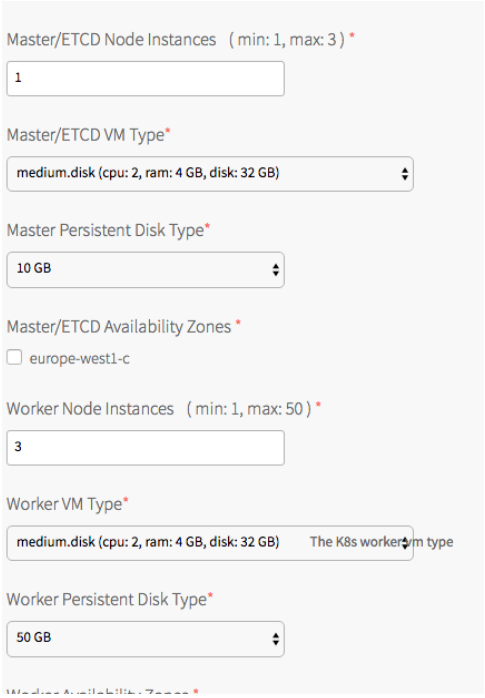
   WARNING: This feature is a beta component and is intended for evaluation and test purposes only. Do not use this feature in a production environment. Product support and future availability are not guaranteed for beta components.

6. Under **Master/ETCD VM Type**, select the type of VM to use for Kubernetes master/etcd nodes. For more information, see the Master Node VM Size section of *VM Sizing for PKS Clusters*.

7. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master node VM.

8. Under **Master/ETCD Availability Zones**, select one or more AZs for the Kubernetes clusters deployed by PKS. If you select more than one AZ, PKS deploys the master VM in the first AZ and the worker VMs across the remaining AZs.

9. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, create clusters with a minimum of three worker nodes, or two per AZ if you intend to use persistent volumes. For example, if you deploy across three AZs, you should have six worker nodes. For more information about persistent volumes, see Persistent Volumes in *Maintaining Workload Uptime*. Provisioning a minimum of three worker nodes, or two nodes per AZ is also recommended for stateless workloads.



10. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker node VMs. For more information, see the Worker Node VM Number and Size section of *VM Sizing for PKS Clusters*.

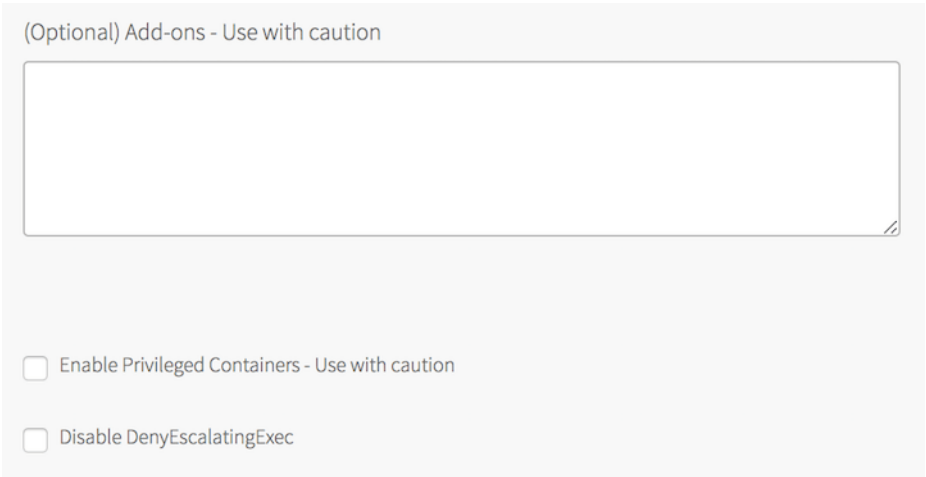
> **Note**: If you install PKS v1.1.5 or later in an NSX-T environment, we recommend that you select a **Worker VM Type** with a minimum disk size of 16 GB. The disk space provided by the default “medium” Worker VM Type is insufficient for PKS with NSX-T v1.1.5 or later.

11. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker node VMs.

12. Under **Worker Availability Zones**, select one or more AZs for the Kubernetes worker nodes. PKS deploys worker nodes equally across the AZs you select.

13. Under **Errand VM Type**, select the size of the VM that contains the errand. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.

14. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to add custom workloads to each cluster in this plan. You can specify multiple files using `---` as a separator. For more information, see Adding Custom Workloads.



15. (Optional) To allow users to create pods with privileged containers, select the **Enable Privileged Containers - Use with caution** option. For more information, see Pods in the Kubernetes documentation.

16. (Optional) To disable the admission controller, select the **Disable DenyEscalatingExec** checkbox. If you select this option, clusters in this plan can create security vulnerabilities that may impact other tiles. Use this feature with caution.

17. Click **Save**.

To deactivate a plan, perform the following steps:

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

2. Select **Plan Inactive**.

3. Click **Save**.

Kubernetes Cloud Provider

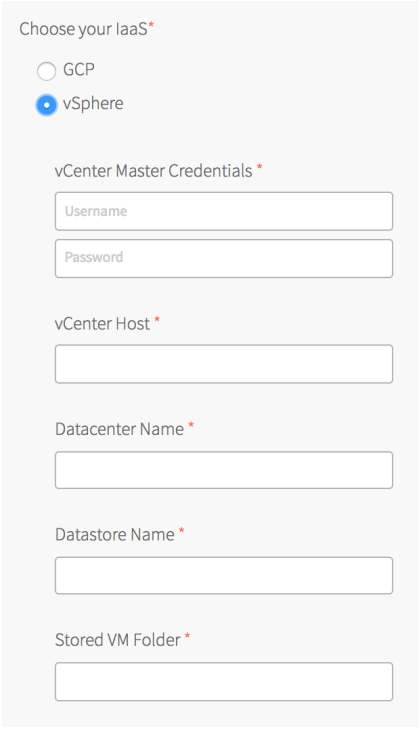
In the procedure below, you use credentials for vCenter master VMs. You must have provisioned the service account with the correct permissions. For more information, see Create the Master Node Service Account in *Preparing vSphere Before Deploying PKS*.

To configure your Kubernetes cloud provider settings, follow the procedure below:

1. Click **Kubernetes Cloud Provider**.

2. Under **Choose your IaaS**, select **vSphere**.

3. Ensure the values in the following procedure 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`. For more information about the master node service account, see Preparing vSphere Before Deploying PKS.
   b. Enter your **vCenter Host**. For example, `vcenter.CF-EXAMPLE.com`.
   c. Enter your **Datacenter Name**. For example, `CF-EXAMPLE-dc`.
   d. Enter your **Datastore Name**. For example, `CF-EXAMPLE-ds`.
   e. Enter the **Stored VM Folder** so that the persistent stores know where to find the VMs. To retrieve the name of the folder, navigate to your BOSH Director tile, click **vCenter Config**, and locate the value for **VM Folder**. The default folder name is `pcf_vms`.

> **Note**: We recommend using a shared datastore for multi-AZ and multi-cluster environments.

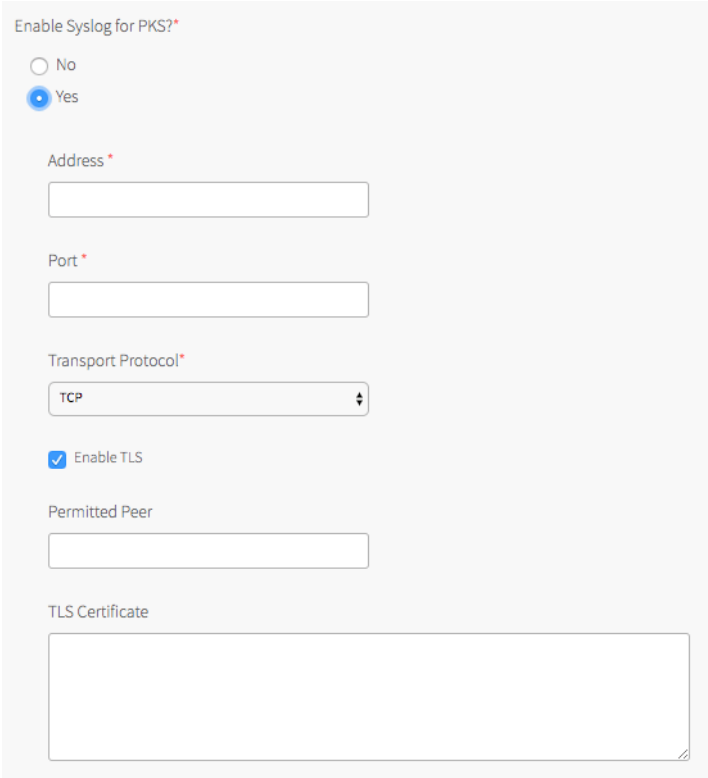
4. Click **Save**.

(Optional) Logging

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

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

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



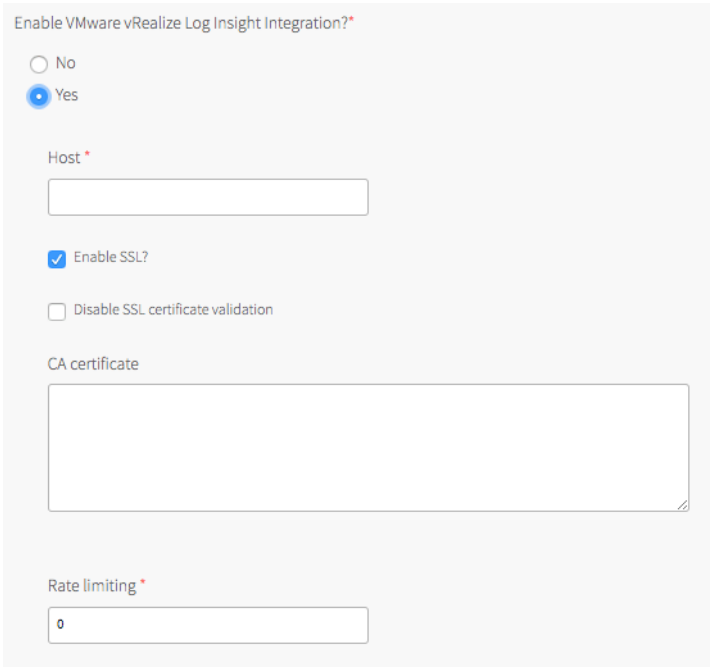
3. Under **Address**, enter the destination syslog endpoint.
4. Under **Port**, enter the destination syslog port.
5. Select a transport protocol for log forwarding.
6. (Optional) Pivotal strongly recommends that you enable TLS encryption when forwarding logs as they may contain sensitive information.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

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

7. You can manage logs using VMware vRealize Log Insight (vRLI). The integration pulls logs from all BOSH jobs and containers running in the cluster, including node logs from core Kubernetes and BOSH processes, Kubernetes event logs, and POD stdout and stderr.

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

    By default, vRLI logging is disabled. To enable and configure vRLI logging, under **Enable VMware vRealize Log Insight Integration?**, select **Yes** and



then perform the following steps:

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

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

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

- To enable certificate validation for the vRLI host, clear the **Disable SSL certificate validation** checkbox.

d. (Optional) If your vRLI certificate is not signed by a trusted CA root or other well known certificate, enter the certificate in the **CA certificate** field. Locate the PEM of the CA used to sign the vRLI certificate, copy the contents of the certificate file, and paste them into the field. Certificates must be in PEM-encoded format.
e. Under **Rate limiting**, enter a time in milliseconds to change the rate at which logs are sent to the vRLI host. The rate limit specifies the minimum time between messages before the fluentd agent begins to drop messages. The default value (0) means the rate is not limited, which suffices for many deployments.

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

8. Click **Save**. These settings apply to any clusters created after you have saved these configuration settings and clicked **Apply Changes**. If the **Upgrade all clusters errand** has been enabled, these setting are also applied to existing clusters.

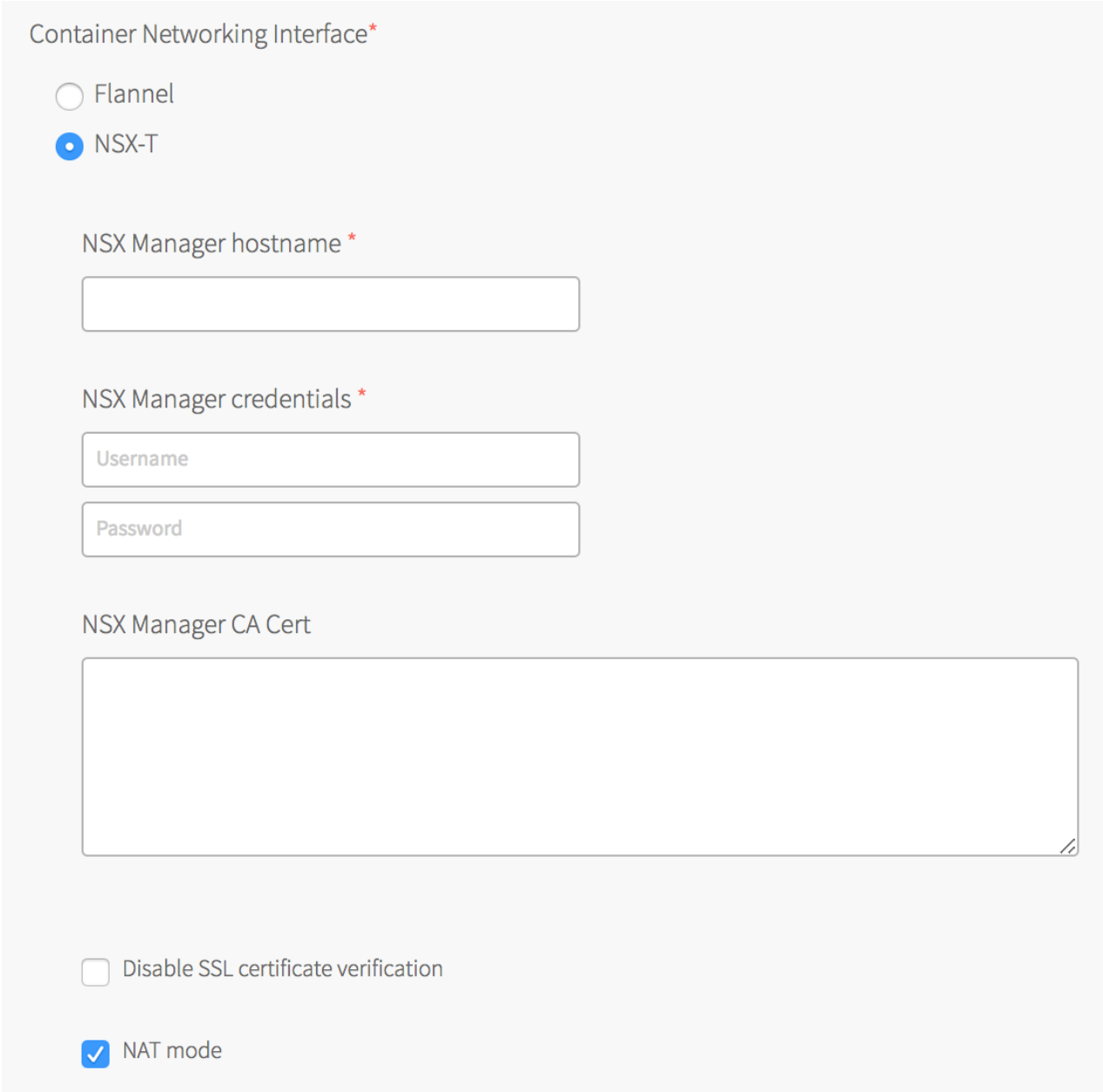
Note: The PKS tile does not validate your vRLI configuration settings. To verify your setup, look for log entries in vRLI.

Networking

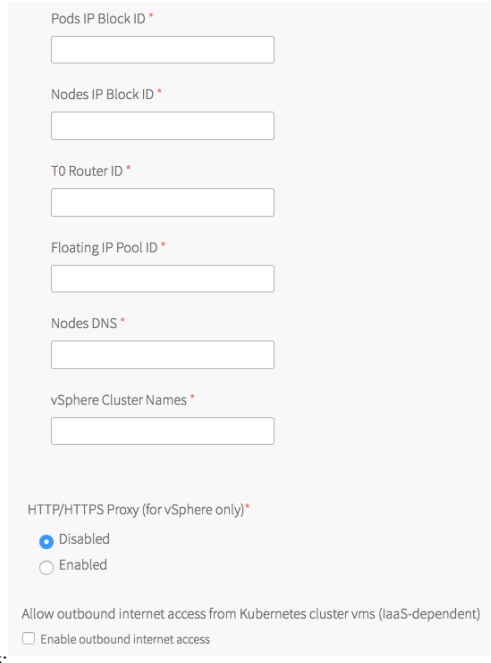
To configure networking, do the following:

1. Click **Networking**.

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



a. For **NSX Manager hostname**, enter the hostname or IP address of your NSX Manager.
b. For **NSX Manager Super User Principal Identify Certificate**, copy and paste the contents and private key of the Principal Identity certificate you created in the NSX Manager Super User Principal Identity Certificate section of *Generating and Registering Certificates*. You can create the certificate in this tab by clicking **Generate RSA Certificate**, providing a wildcard domain, for example, `*.nsx.pks.vmware.local`, and copying the generated certificate and key to the `pks-nsx-t-superuser.crt` and `pks-nsx-t-superuser.key` files. For more information, including instructions for completing the additional, required registration and verification steps, see NSX Manager Super User Principal Identity Certificate in *Generating and Registering Certificates*.
c. (Optional) For **NSX Manager CA Cert**, copy and paste the contents of the NSX Manager CA certificate you created in Generating and Registering Certificates. This will be used to connect to the NSX Manager.
d. 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.
e. If you are using a NAT deployment topology, leave the **NAT mode** checkbox selected. If you are using a No-NAT topology, clear this checkbox. For more information, see Deployment Topologies.



f. Enter the following IP Block settings:

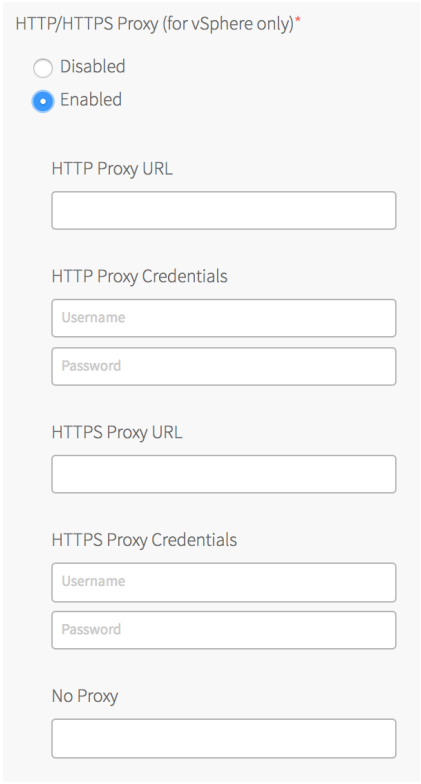
- **Pods IP Block ID**: Enter the UUID of the IP block to be used for Kubernetes pods. PKS allocates IP addresses for the pods when they are created in Kubernetes. Each time a namespace is created in Kubernetes, a subnet from this IP block is allocated. The current subnet size that is created is /24,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

g. For **T0 Router ID**, enter the `t0-pks` T0 router UUID. Locate this value in the NSX-T UI router overview.
h. For **Floating IP Pool ID**, enter the `ip-pool-vips` ID that you created for load balancer VIPs. For more information, see Plan Network CIDRs. PKS uses the floating IP pool to allocate IP addresses to the load balancers created for each of the clusters. The load balancer routes the API requests to the master nodes and the data plane.
i. For **Nodes DNS**, enter one or more Domain Name Servers used by the Kubernetes nodes.
j. For **vSphere Cluster Names**, enter a comma-separated list of the vSphere clusters where you will deploy Kubernetes clusters. The NSX-T precheck errand uses this field to verify that the hosts from the specified clusters are available in NSX-T. You can specify clusters in this format: `cluster1,cluster2,cluster3`.

3. (Optional) Configure a global proxy for all outgoing HTTP and HTTPS traffic from your Kubernetes clusters.

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

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



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

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

4. Under **Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)**, ignore the **Enable outbound internet access** checkbox.

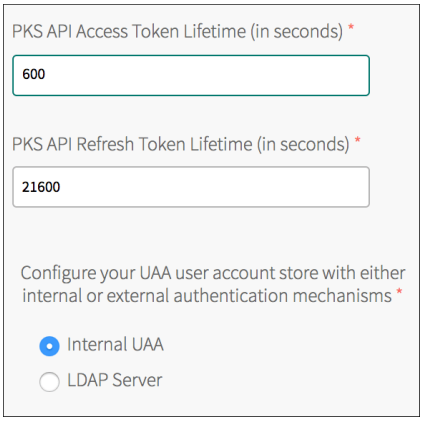
5. Click **Save**.

UAA

To configure the UAA server, do the following:

1. Click **UAA**.

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



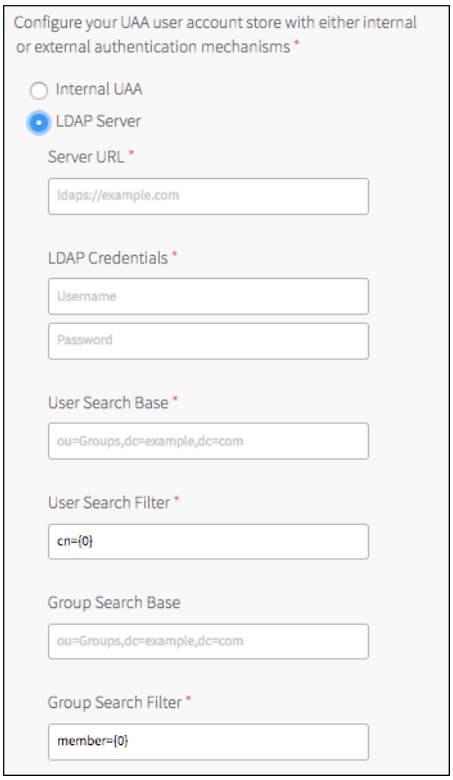
3. Under **PKS CLI Refresh Token Lifetime**, enter a time in seconds for the PKS CLI refresh token lifetime.

4. Select one of the following options:
    - To use an internal user account store for UAA, select **Internal UAA**. Click **Save** and continue to (Optional) Monitoring.
    - To use an external user account store for UAA, select **LDAP Server** and continue to Configure LDAP as an Identity Provider.

Configure LDAP as an Identity Provider

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

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



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

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

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

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

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

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

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

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

Note: For information about testing and troubleshooting your LDAP search filters, see Configuring LDAP Integration with Pivotal Cloud Foundry.

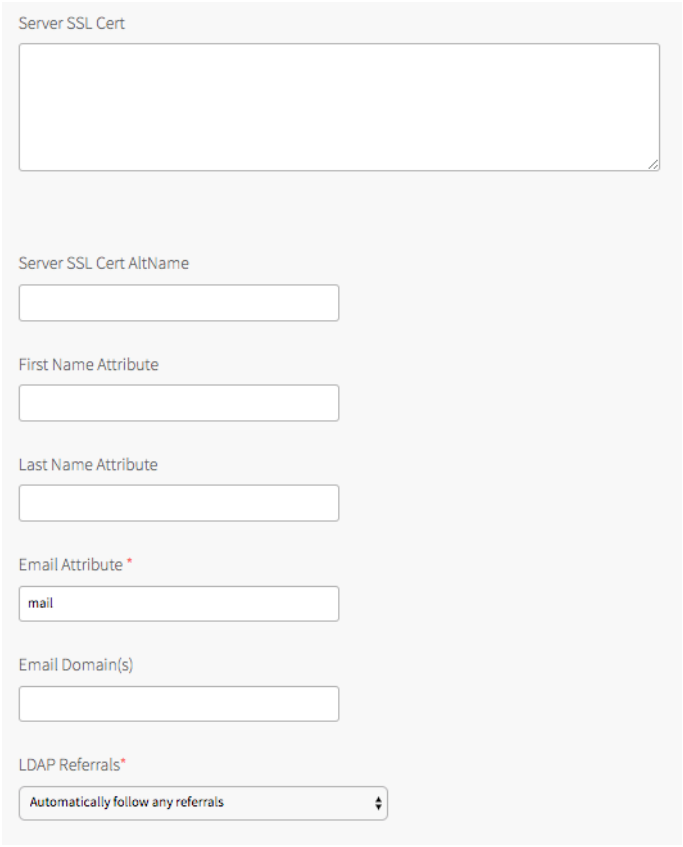
6. For **Group Search Base**, enter the location in the LDAP directory tree where the LDAP group search begins.

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

   Follow the instructions in the Grant Cluster Access to an External LDAP Group section of *Managing Users in PKS with UAA* to map the groups under this search base to roles in PKS.

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

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



9. For **Server SSL Cert AltName**, do one of the following:

   - If you are using `ldaps://` with a self-signed certificate, enter a Subject Alternative Name (SAN) for your certificate.
   - If you are not using `ldaps://` with a self-signed certificate, leave this field blank.

10. For **First Name Attribute**, enter the attribute name in your LDAP directory that contains user first names. For example, `cn`.

11. For **Last Name Attribute**, enter the attribute name in your LDAP directory that contains user last names. For example, `sn`.

12. For **Email Attribute**, enter the attribute name in your LDAP directory that contains user email addresses. For example, `mail`.

13. For **Email Domain(s)**, enter a comma-separated list of the email domains for external users who can receive invitations to Apps Manager.

14. For **LDAP Referrals**, choose how UAA handles LDAP server referrals to other user stores. UAA can follow the external referrals, ignore them without returning errors, or generate an error for each external referral and abort the authentication.

15. Click **Save**.

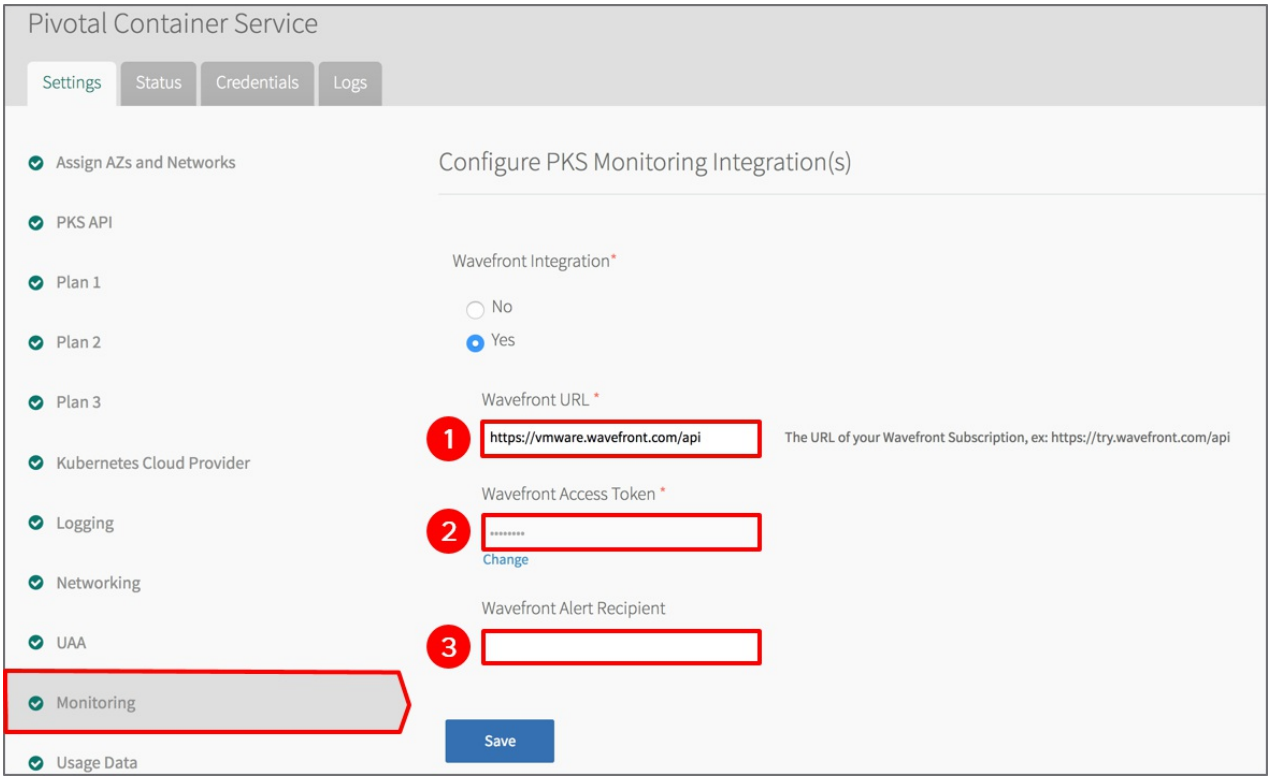
(Optional) Monitoring

You can monitor Kubernetes clusters and pods metrics externally using the integration with Wavefront by VMware.

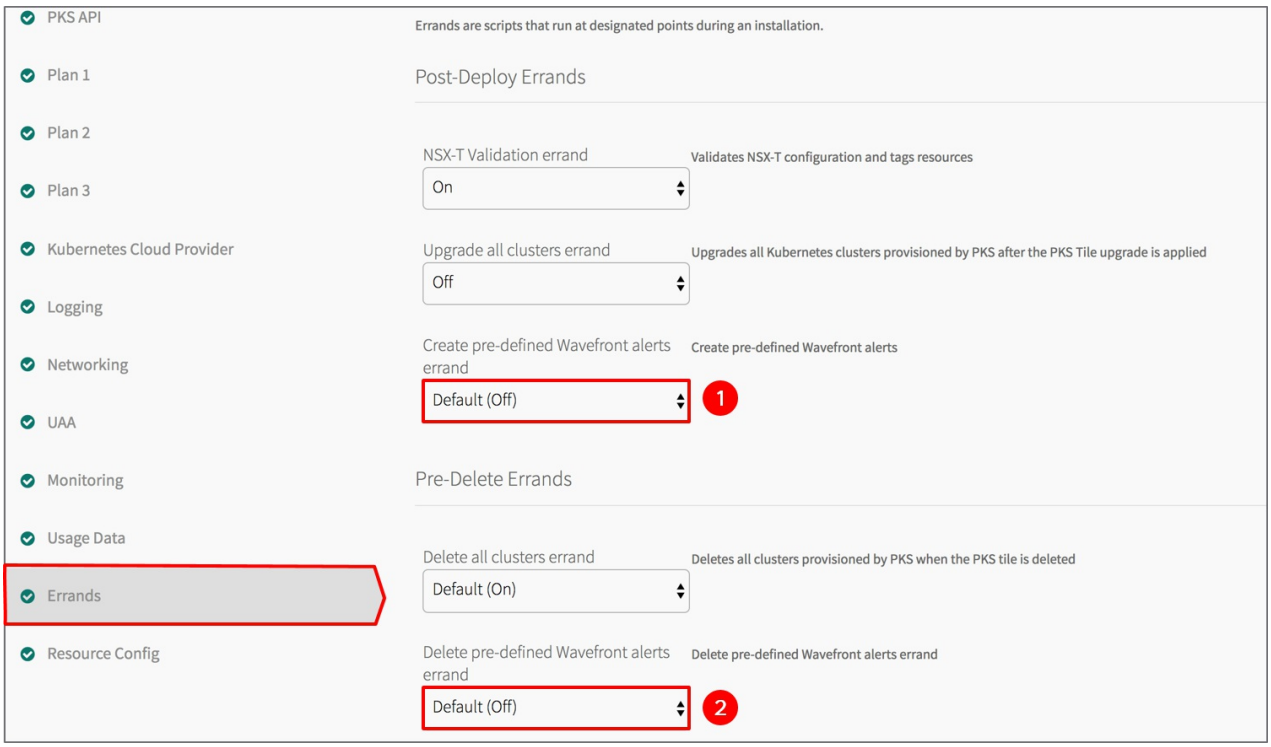
Note: Before you configure Wavefront integration, you must have an active Wavefront account and access to a Wavefront instance. You provide your Wavefront access token during configuration and enabling errands. For additional information, see Pivotal Container Service Integration Details in the Wavefront documentation.

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

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



2. Under **Wavefront URL**, enter the URL of your Wavefront subscription. For example, `https://try.wavefront.com/api`.
3. Under **Wavefront Access Token**, enter the API token for your Wavefront subscription.
4. To configure Wavefront to send alerts by email, enter email addresses or Wavefront Target IDs separated by commas under **Wavefront Alert Recipient**. For example: `user@example.com,Wavefront_TargetID`. To create alerts, you must enable errands.
5. In the **Errands** tab, enable **Create pre-defined Wavefront alerts errand** and **Delete pre-defined Wavefront alerts errand**.



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

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

Usage Data

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

For information about the metrics PKS sends when you opt in to CEIP or Telemetry, see PKS Telemetry.

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

To configure the **Usage Data** pane:

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

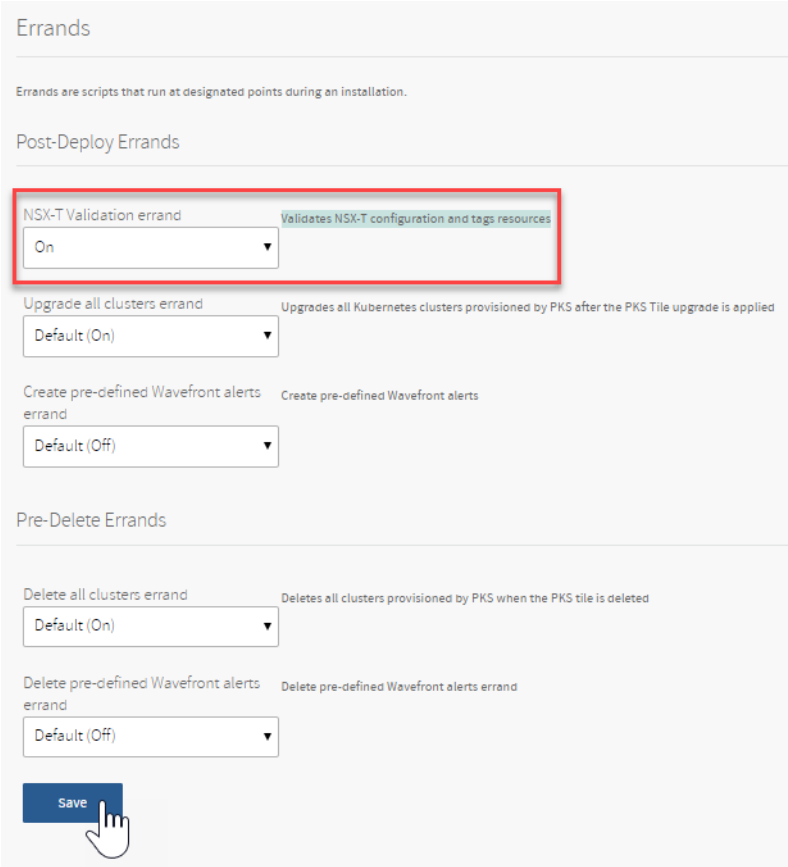
> **Note:** If you join the CEIP and Telemetry Program for PKS, open your firewall to allow outgoing access to `https://vcsa.vmware.com/ph-prd` on port `443`.

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.

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



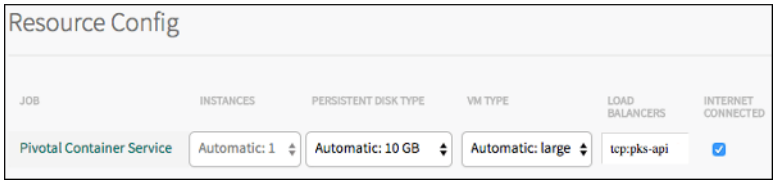
For more information about errands and their configuration state, see Managing Errands in Ops Manager.

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

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

Resource Config

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



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.

Step 3: Apply Changes

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

Step 4: Install the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

Step 5: Share the 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 Creating Clusters for more information.

1. When the installation is completed, retrieve the PKS endpoint by performing the following steps:
    a. From the Ops Manager Installation Dashboard, click the **Pivotal Container Service** tile.
    b. Click the **Status** tab and record the IP address assigned to the `Pivotal Container Service` job.
2. Create a DNAT rule on the `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.

    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.

Step 6: Configure PKS API Access

Follow the procedures in Configuring PKS API Access.

Step 7: Configure Authentication for PKS

Configure authentication for PKS using User Account and Authentication (UAA). For information about managing users in PKS with UAA, see Managing Users in PKS with UAA.

Next Steps

After installing PKS on vSphere with NSX-T integration, you may want to do one or more of the following:

- Integrate VMware Harbor with PKS to store and manage container images. For more information, see Integrating VMware Harbor Registry with PKS.
- Create your first PKS cluster. For more information, see Creating Clusters.

Google Cloud Platform (GCP)

This topic lists the steps to follow when installing Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

See the following topics:

- Prerequisites and Resource Requirements
- Deploying Ops Manager on GCP:
  - Preparing GCP
  - Deploying BOSH and Ops Manager to GCP
  - Configuring BOSH Director on GCP
- Creating Service Accounts in GCP for PKS
- Configuring a GCP Load Balancer for the PKS API
- Installing PKS on GCP

Installing the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

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).

Prerequisites

Before you install PKS, you must install one of the following:

- Ops Manager v2.1.0 or later
- Ops Manager v2.2.0 or later

Note: You use Ops Manager to install and configure PKS. Each version of Ops Manager supports multiple versions of PKS. To confirm that your Ops Manager version supports the version of PKS that you install, see PKS Release Notes.

You must also create service accounts for Kubernetes master and worker nodes and create a load balancer to access the PKS API.

Install and Configure Ops Manager

To install an Ops Manager version that is compatible with the PKS version you intend to use, follow the instructions in the corresponding version of the Ops Manager documentation.

Version	
Ops Manager v2.1	• Preparing to Deploy PCF on GCP • Deploying BOSH and Ops Manager to GCP • Configuring BOSH Director on GCP
Ops Manager v2.2	• Preparing to Deploy PCF on GCP • Deploying BOSH and Ops Manager to GCP • Configuring BOSH Director on GCP

Create Service Accounts for Kubernetes

After you install and configure Ops Manager, you must create service accounts for Kubernetes master and worker node VMs in your PKS deployment. To create the service accounts, follow the procedures in Creating Service Accounts in GCP for PKS.

Create a Load Balancer for the PKS API

Before you install PKS, you must create an external TCP load balancer so that you can access the PKS API from outside the network. This load balancer allows you to run `pks` commands from your local workstation. You must create the load balancer before you install PKS, and then complete the load balancer configuration after you install PKS.

To create a load balancer in GCP, follow the procedures in Creating a GCP Load Balancer for the PKS API.

Resource Requirements

Installing Ops Manager and PKS requires the following virtual machines (VMs):

VM	CPU	RAM	Storage
Pivotal Container Service	2	8 GB	16 GB
Pivotal Ops Manager	1	8 GB	160 GB

BOSH Director	2	8 GB	16 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	32 GB	5 GB
worker	1	2	4 GB	32 GB	50 GB

Creating Service Accounts in GCP for PKS

Page last updated:

This topic describes the steps required to create service accounts for Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

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. 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**
        - **Compute Instance Admin (v1)**
        - **Compute Network Admin**
        - **Compute Security Admin**
        - **Compute Storage Admin**
        - **Compute Viewer**
    - **Service Accounts**
        - **Service Account User**
4. Click **Create**.

Create the Worker Node Service Account

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

After you create both service accounts for Kubernetes, follow the procedures in Installing PKS on GCP.

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. Under **Internet facing or internal only**, select **From Internet to my VMs**.
5. Under **Multiple regions or single region**, select **Single region only**.
6. Click **Continue**.
7. Name your load balancer. Pivotal recommends naming your load balancer `pks-api`.
8. Select **Backend configuration**.
    - Under **Region**, select the region where you deployed Ops Manager.
    - Under **Backends**, select **Select existing instances**. This is automatically configured when updating the Resource Config section of the PKS tile.
    - (Optional) Under **Backup pool**, select a backup pool. If you select a backup pool, set a **Failover ratio**.
    - (Optional) Under **Health check**, select whether or not you want to create a health check.
    - Under **Session affinity**, select a session affinity configuration.
    - (Optional) Select **Advanced configurations** to configure the **Connection draining timeout**.
9. Select **Frontend configuration**.
    - (Optional) Name your frontend.
    - (Optional) Click **Add a description** and provide 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**.
10. Click **Review and finalize** to review your load balancer configuration.
11. 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:
    - Name your firewall rule.
    - (Optional) Provide a description for your firewall rule.
    - Under **Network**, select the VPC network you created in the Create a GCP Network with Subnets step of *Preparing 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 `pks-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 PKS on GCP 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 `pks-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 `pks-api` tag. Add the tag if it does not appear in the field.
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**.
    - Provide 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 `pks-api.example.com` as your PKS API hostname, enter `pks-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 Create a Load Balancer.

8. Click **Create**.

Next Steps

After you complete this procedure, follow the instructions in Installing PKS on GCP.

Installing PKS on GCP

Page last updated:

This topic describes how to install and configure Pivotal Container Service (PKS) on Google Cloud Platform (GCP).

Prerequisites

Before performing the procedures in this topic, you must have deployed and configured Ops Manager. For more information, see GCP Prerequisites and Resource Requirements.

If you use 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. Open the **Director Config** pane.
3. Select the **Enable Post Deploy Scripts** checkbox.
4. Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
5. Click the **Installation Dashboard** link to return to the Installation Dashboard.
6. Click **Apply Changes**.

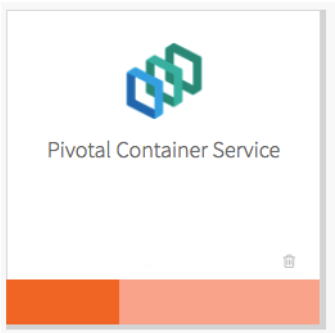
Step 1: Install PKS

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.



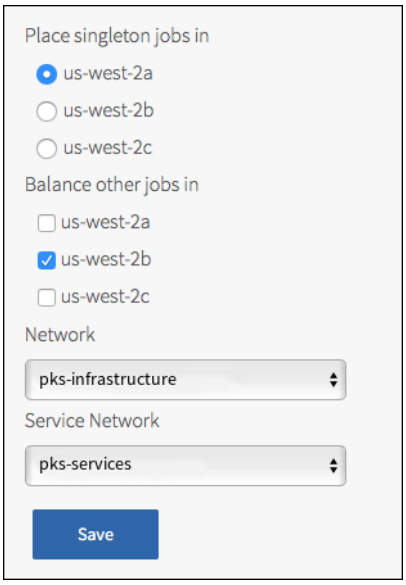
Assign AZs and Networks

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.

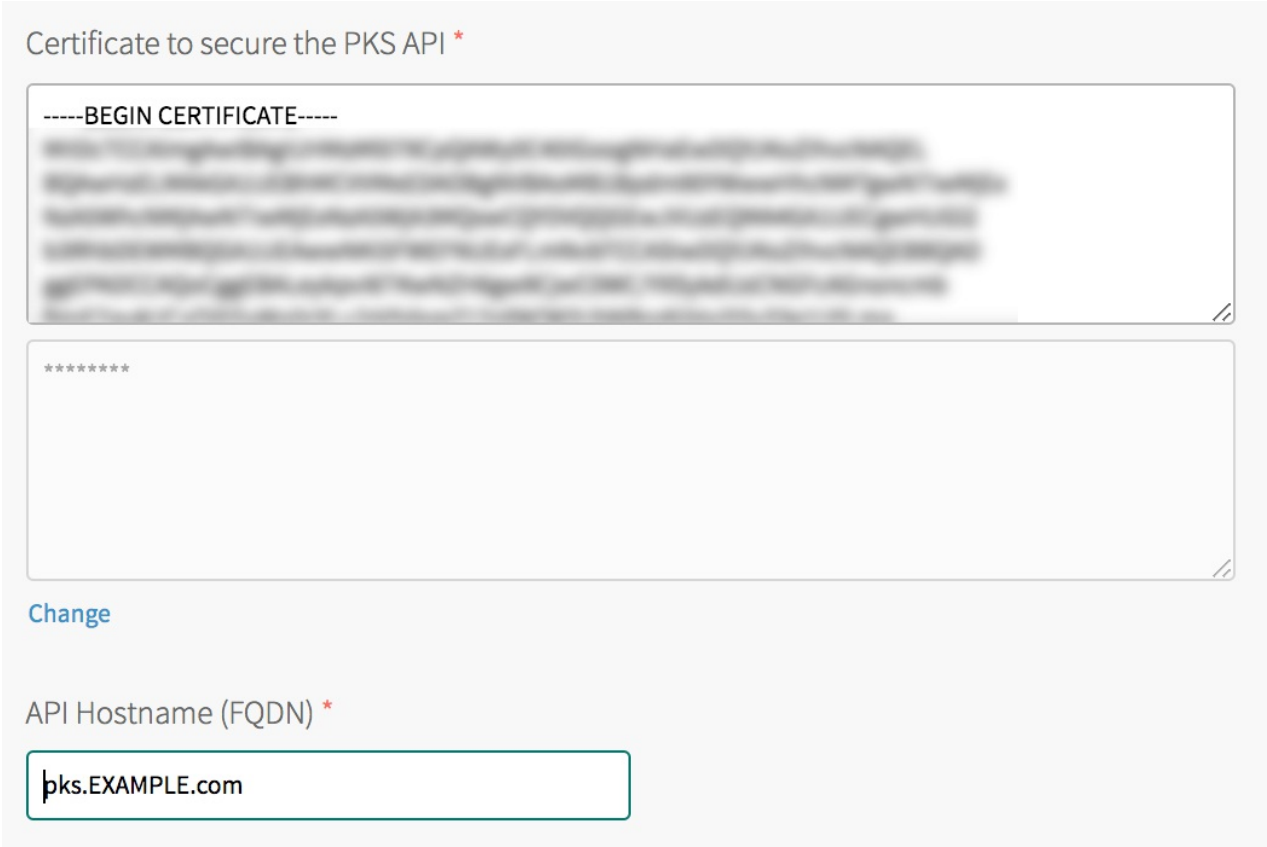


3. Under **Network**, select the infrastructure subnet you created for the PKS API VM.
4. Under **Service Network**, select the services subnet you created for Kubernetes cluster VMs.
5. Click **Save**.

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 that you supply should cover the domain that routes to the PKS API VM with TLS termination on the ingress.

If you do not have a certificate and private key pair, PKS can generate one for you by performing the following steps.

> **Note**: If you configured Ops Manager Front End without a certificate, you can use this new certificate to complete Ops Manager configuration. To configure your Ops Manager Front End certificate, see Configure Front End.

   a. Select the **Generate RSA Certificate** link.
   b. Enter the wildcard domain for your API hostname. For example, if your PKS API domain is `api.pks.example.com`, then enter `*.pks.example.com`.
   c. Click **Generate**.

3. Under **API Hostname (FQDN)**, enter a fully qualified domain name (FQDN) to access the PKS API. For example, `api.pks.example.com`.

4. Click **Save**.

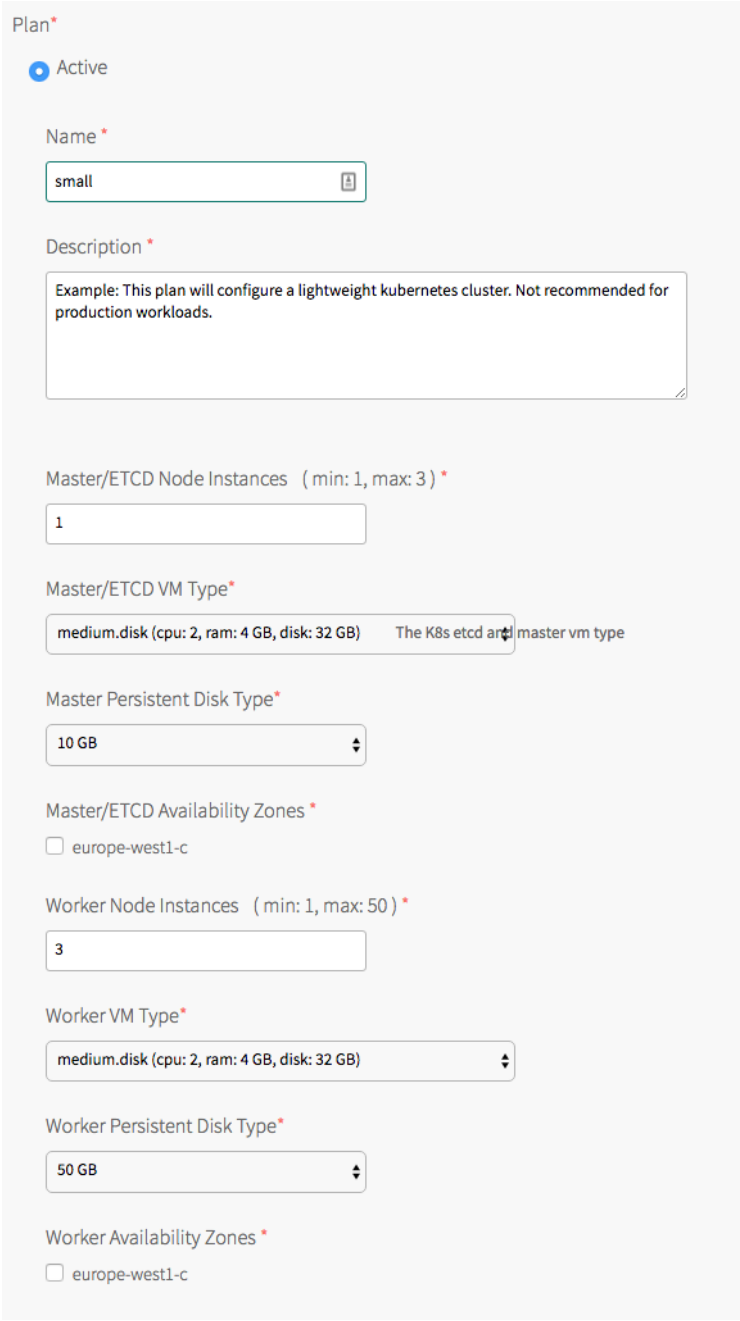
Plans

To activate a plan, perform the following steps:

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

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

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



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

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

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

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

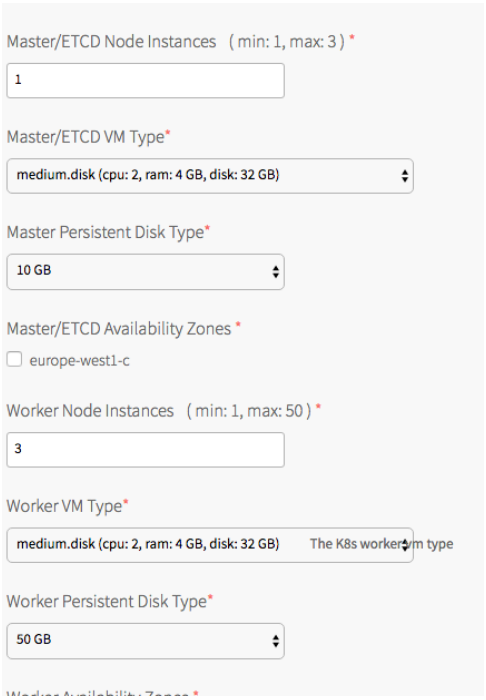
   WARNING: This feature is a beta component and is intended for evaluation and test purposes only. Do not use this feature in a production environment. Product support and future availability are not guaranteed for beta components.

6. Under **Master/ETCD VM Type**, select the type of VM to use for Kubernetes master/etcd nodes. For more information, see the Master Node VM Size section of *VM Sizing for PKS Clusters*.

7. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master node VM.

8. Under **Master/ETCD Availability Zones**, select one or more AZs for the Kubernetes clusters deployed by PKS. If you select more than one AZ, PKS deploys the master VM in the first AZ and the worker VMs across the remaining AZs.

9. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, create clusters with a minimum of three worker nodes, or two per AZ if you intend to use persistent volumes. For example, if you deploy across three AZs, you should have six worker nodes. For more information about persistent volumes, see Persistent Volumes in *Maintaining Workload Uptime*. Provisioning a minimum of three worker nodes, or two nodes per AZ is also recommended for stateless workloads.



10. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker node VMs. For more information, see the Worker Node VM Number and Size section of *VM Sizing for PKS Clusters*.

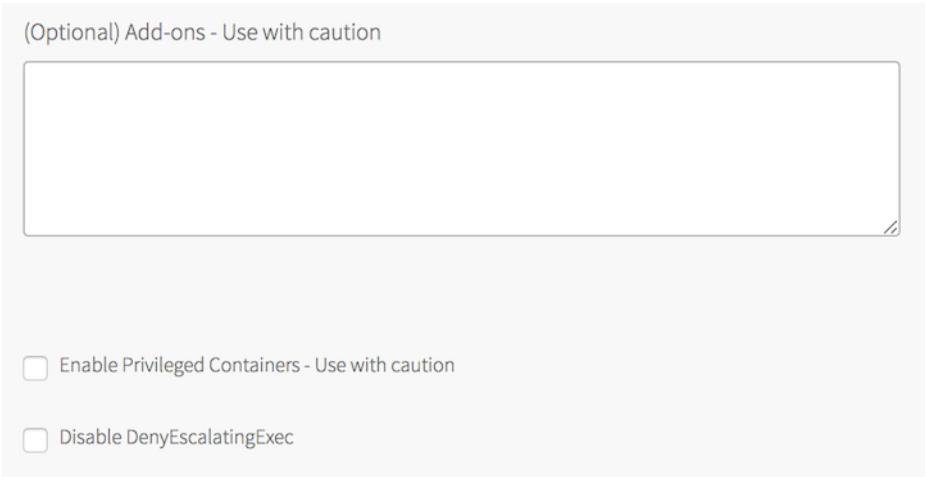
> **Note**: If you install PKS v1.1.5 or later in an NSX-T environment, we recommend that you select a **Worker VM Type** with a minimum disk size of 16 GB. The disk space provided by the default “medium” Worker VM Type is insufficient for PKS with NSX-T v1.1.5 or later.

11. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker node VMs.

12. Under **Worker Availability Zones**, select one or more AZs for the Kubernetes worker nodes. PKS deploys worker nodes equally across the AZs you select.

13. Under **Errand VM Type**, select the size of the VM that contains the errand. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.

14. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to add custom workloads to each cluster in this plan. You can specify multiple files using `---` as a separator. For more information, see Adding Custom Workloads.



15. (Optional) To allow users to create pods with privileged containers, select the **Enable Privileged Containers - Use with caution** option. For more information, see Pods in the Kubernetes documentation.

16. (Optional) To disable the admission controller, select the **Disable DenyEscalatingExec** checkbox. If you select this option, clusters in this plan can create security vulnerabilities that may impact other tiles. Use this feature with caution.

17. Click **Save**.

To deactivate a plan, perform the following steps:

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

2. Select **Plan Inactive**.

3. Click **Save**.

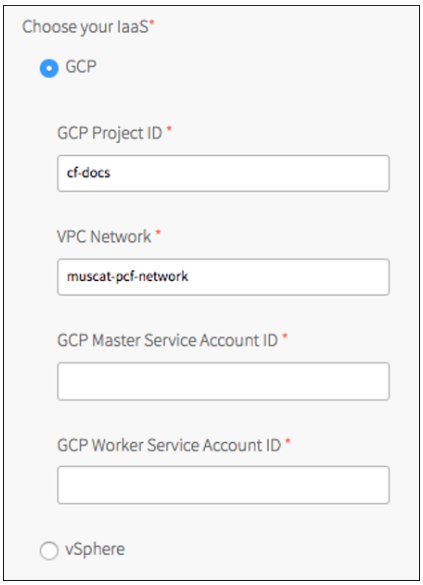
Kubernetes Cloud Provider

To configure your Kubernetes cloud provider settings, follow the procedures below:

1. Click **Kubernetes Cloud Provider**.

2. Under **Choose your IaaS**, select **GCP**.

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



   a. Enter your **GCP Project Id**, which is the name of the deployment in your Ops Manager environment. To find the project ID, go to **BOSH Director for GCP > Google Config > Project ID**.
   b. Enter your **VPC Network**, which is the VPC network name for your Ops Manager environment.
   c. Enter your **GCP Master Service Account ID**. This is the email address associated with the master node service account. For information about configuring this account, see Create the Master Node Service Account in *Creating Service Accounts in GCP for PKS*.
   d. Enter your **GCP Worker Service Account ID**. This is the email address associated with the worker node service account. For information about configuring this account, see Create the Worker Node Service Account in *Creating Service Accounts in GCP for PKS*.

4. Click **Save**.

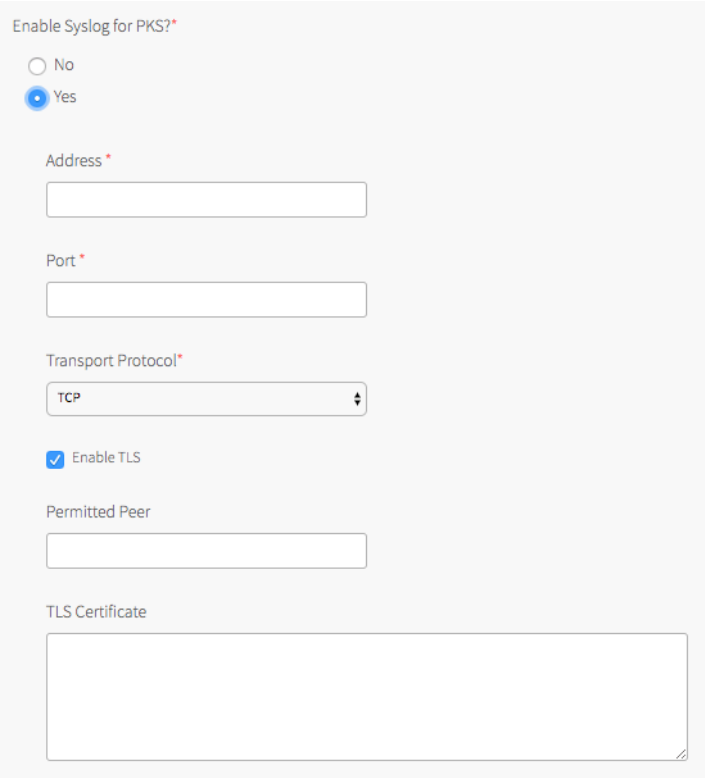
(Optional) Logging

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

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

1. Click **Logging**.

2. To enable syslog forwarding, select **Yes**.



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

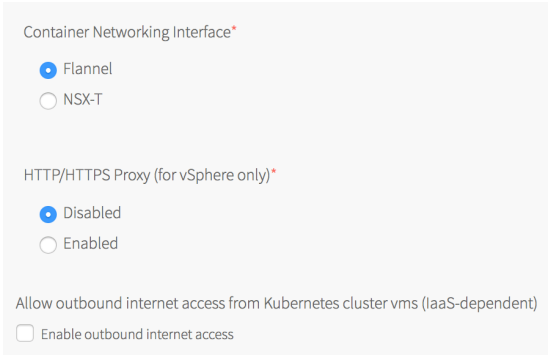
    > **Note**: You do not need to provide a new certificate if the TLS certificate for the destination syslog endpoint is signed by a Certificate Authority (CA) in your BOSH certificate store.

7. Click **Save**.

Networking

To configure networking, do the following:

1. Click **Networking**.

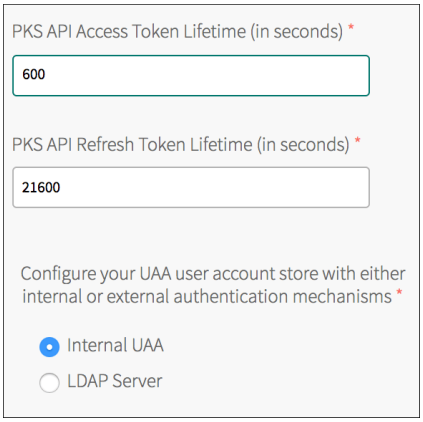


2. Under **Container Networking Interface**, select **Flannel**.
3. (Optional) If you do not use a NAT instance, select **Allow outbound internet access from Kubernetes cluster vms (IaaS-dependent)**. Enabling this functionality assigns external IP addresses to VMs in clusters.
4. Click **Save**.

UAA

To configure the UAA server, do the following:

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

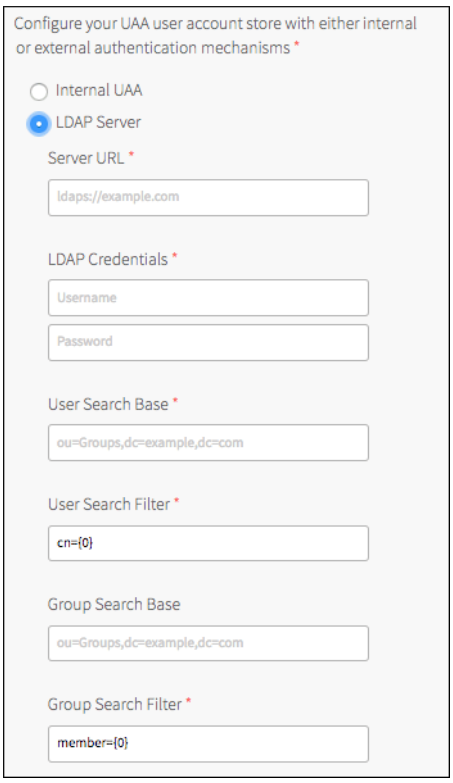


3. Under **PKS CLI Refresh Token Lifetime**, enter a time in seconds for the PKS CLI refresh token lifetime.
4. Select one of the following options:
    - To use an internal user account store for UAA, select **Internal UAA**. Click **Save** and continue to (Optional) Monitoring.
    - To use an external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

Configure LDAP as an Identity Provider

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

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



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

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

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

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

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

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

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

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

    Note: For information about testing and troubleshooting your LDAP search filters, see Configuring LDAP Integration with Pivotal Cloud Foundry.

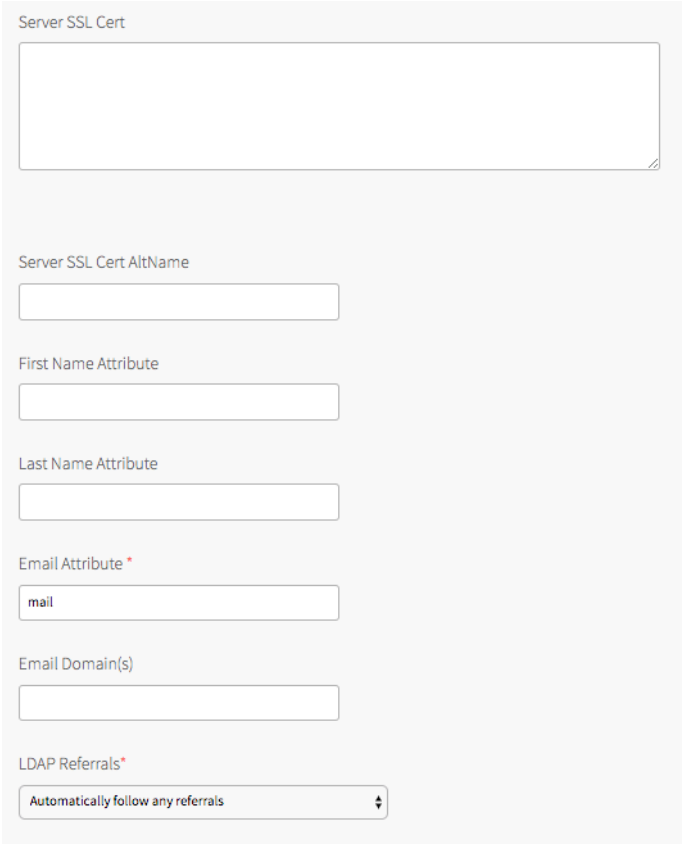
6. For **Group Search Base**, enter the location in the LDAP directory tree where the LDAP group search begins.

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

    Follow the instructions in the Grant Cluster Access to an External LDAP Group section of *Managing Users in PKS with UAA* to map the groups under this search base to roles in PKS.

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

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



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

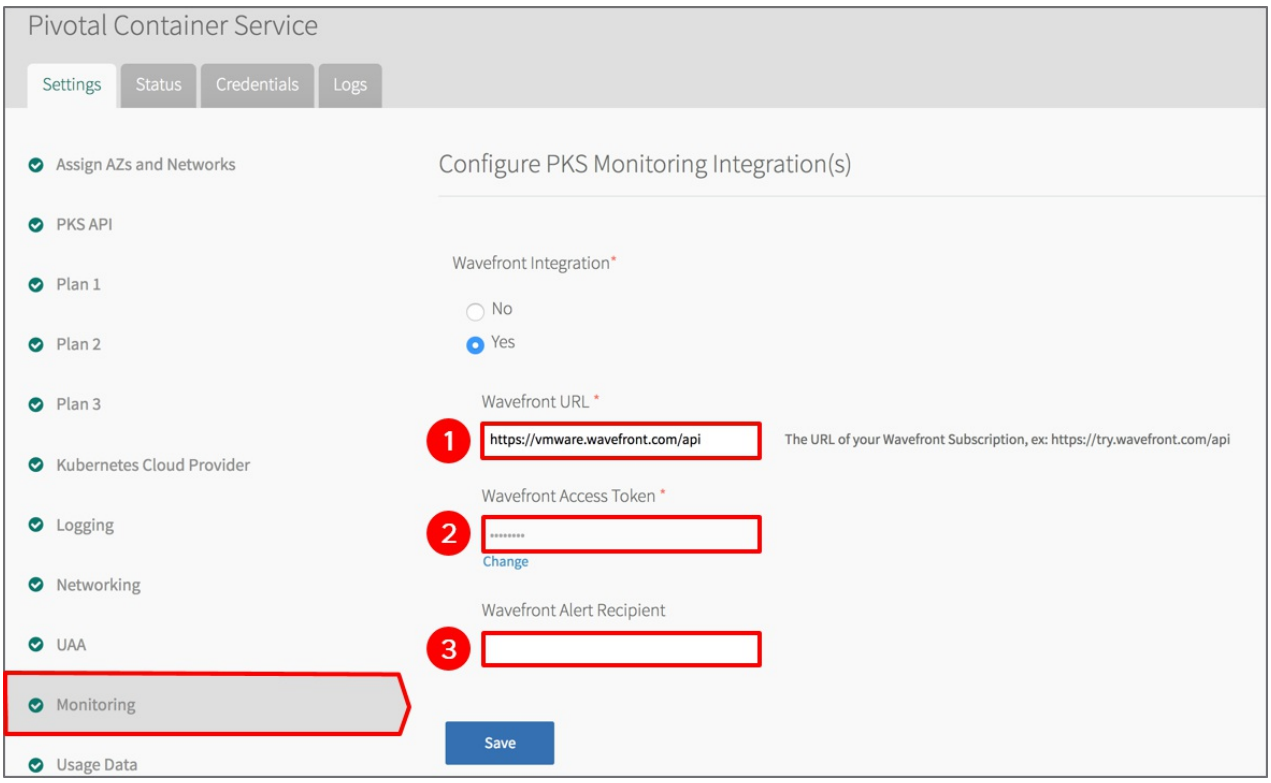
(Optional) Monitoring

You can monitor Kubernetes clusters and pods metrics externally using the integration with Wavefront by VMware.

Note: Before you configure Wavefront integration, you must have an active Wavefront account and access to a Wavefront instance. You provide your Wavefront access token during configuration and enabling errands. For additional information, see Pivotal Container Service Integration Details in the Wavefront documentation.

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

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

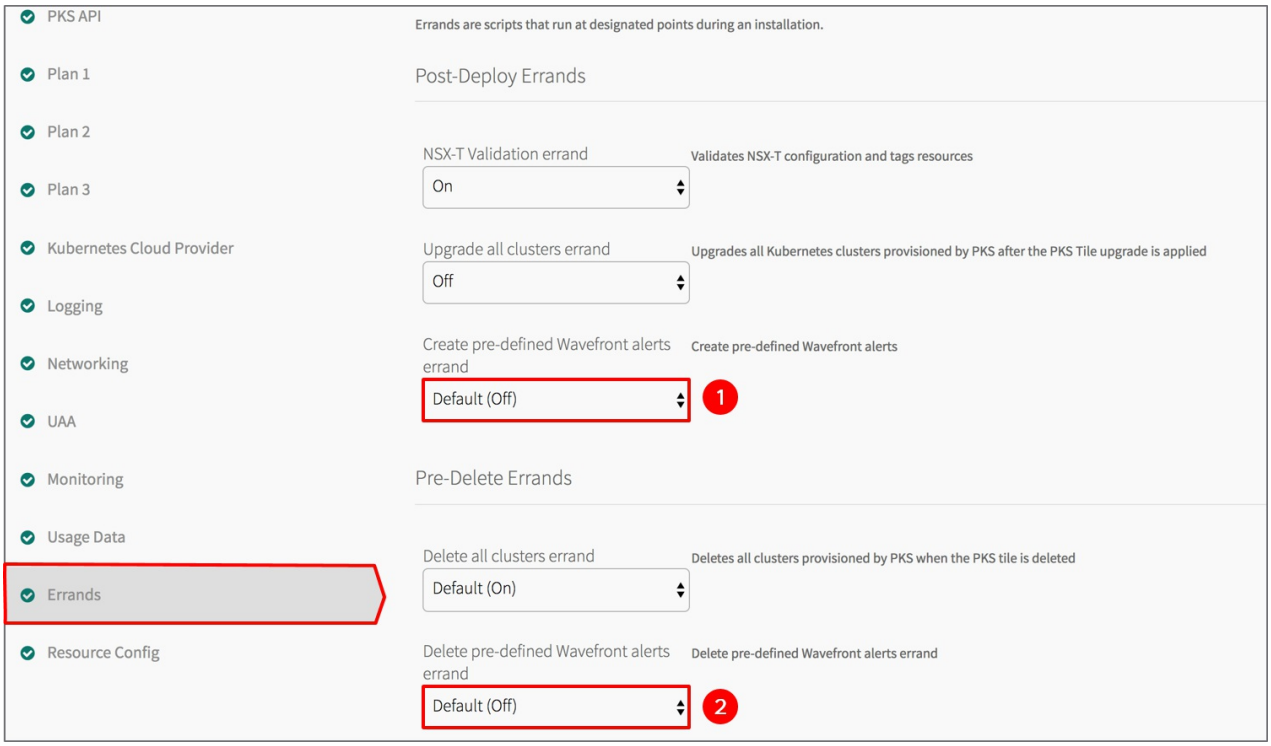


2. Under **Wavefront URL**, enter the URL of your Wavefront subscription. For example, `https://try.wavefront.com/api`.

3. Under **Wavefront Access Token**, enter the API token for your Wavefront subscription.

4. To configure Wavefront to send alerts by email, enter email addresses or Wavefront Target IDs separated by commas under **Wavefront Alert Recipient**. For example: `user@example.com,Wavefront_TargetID`. To create alerts, you must enable errands.

5. In the **Errands** tab, enable **Create pre-defined Wavefront alerts errand** and **Delete pre-defined Wavefront alerts errand**.



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

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

Wavefront.

Usage Data

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

For information about the metrics PKS sends when you opt in to CEIP or Telemetry, see PKS Telemetry.

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

To configure the **Usage Data** pane:

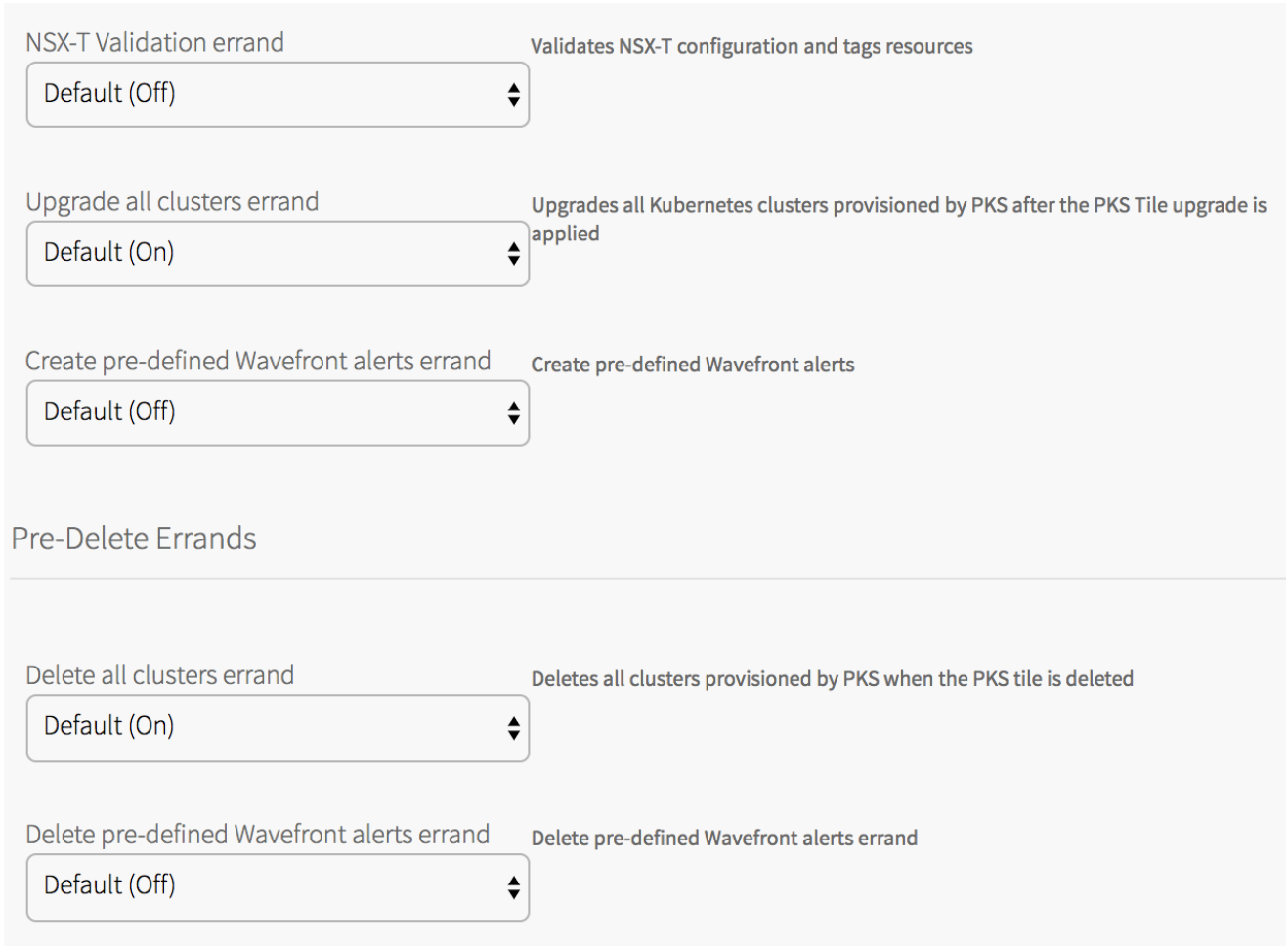
1. Select the **Usage Data** side-tab.
2. Read the Usage Data description.
3. Make your selection.
    a. To join the program, select **Yes, I want to join the CEIP and Telemetry Program for PKS**.
    b. To decline joining the program, select **No, I do not want to join the CEIP and Telemetry Program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

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

Errands

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

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



For more information about errands and their configuration state, see Managing Errands in Ops Manager.

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

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

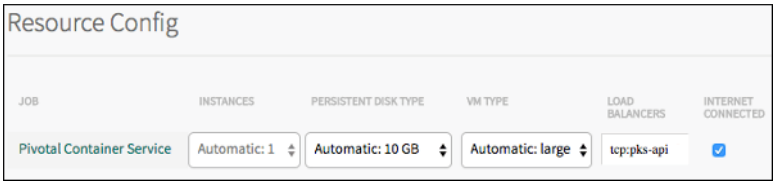
Resource Config

To modify the resource usage of PKS and specify your PKS API load balancer, follow the steps below:

1. Select **Resource Config**.
2. In the **Load Balancers** column, enter a name for your PKS API load balancer that begins with `tcp:`. For example, `tcp:pks-api`, where `pks-api` is the name that you configured in the Create a Load Balancer section of *Creating a GCP Load Balancer for the PKS API*.

   Note: After you click **Apply Changes** for the first time, BOSH assigns the PKS VM an IP address. BOSH uses the name you provide in the **Load Balancers** column to locate your load balancer, and then connect the load balancer to the PKS VM using its new IP address.

3. (Optional) Edit resources used by the **Pivotal Container Service** job.



4. In the **Load Balancers** column, enter a name for your PKS API load balancer that begins with `tcp:`. For example, `tcp:pks-api`, where `pks-api` is the name that you configured in the Create a Load Balancer step *Configuring a GCP Load Balancer for the PKS API*.

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.

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 the 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 Creating Clusters 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

Follow the procedure in the Create a Network Tag for the Firewall Rule section of *Configuring a GCP Load Balancer for the PKS API*.

Step 6: Install the PKS and Kubernetes CLIs

The PKS and Kubernetes CLIs help you interact with your PKS-provisioned Kubernetes clusters and Kubernetes workloads. To install the CLIs, follow the instructions below:

- Installing the PKS CLI
- Installing the Kubernetes CLI

Step 7: Configure PKS API Access

Follow the procedures in Configuring PKS API Access.

Step 8: Configure Authentication for PKS

Configure authentication for PKS using User Account and Authentication (UAA). For information about managing users in PKS with UAA, see Managing Users in PKS with UAA.

Next Steps

After installing PKS on GCP, you may want to do one or more of the following:

- Create a load balancer for your PKS clusters. For more information, see Creating and Configuring a GCP Load Balancer for PKS Clusters.
- Create your first PKS cluster. For more information, see Creating Clusters.

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. Select your desired release version from the **Releases** dropdown.
4. Click **PKS CLI**.
5. Click **PKS CLI - Mac** to download the Mac OS X binary.
6. Rename the downloaded binary file to `pks`.
7. On the command line, run the following command to make the PKS binary act as an executable file:

```
$ chmod +x pks
```

8. Move the binary file into your `PATH`.

   For example, you can run the following command:

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

Linux

1. Navigate to Pivotal Network and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Select your desired release version from the **Releases** dropdown.
4. Click **PKS CLI**.
5. Click **PKS CLI - Linux** to download the Linux binary.
6. Rename the downloaded binary file to `pks`.
7. On the command line, run the following command to make the PKS binary executable:

```
$ chmod +x pks
```

8. Move the binary file into your `PATH`.

   For example, you can run the following command:

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

Windows

1. Navigate to Pivotal Network and log in.
2. Click **Pivotal Container Service (PKS)**.
3. Select your desired release version from the **Releases** dropdown.
4. Click **PKS CLI**.
5. Click **PKS CLI - Windows** to download the Windows executable file.
6. Rename the downloaded binary file to `pks.exe`.
7. Move the binary file into your `PATH`.

Log in to PKS CLI

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

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

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

Replace the placeholder values in the command as follows:

- `PKS-API` is the domain name for the PKS API that you entered in **Ops Manager** > **Pivotal Container Service** > **PKS API** > **API Hostname (FQDN)**. For example, `api.pks.example.com`.
- `USERNAME` and `PASSWORD` belong to the account you created in the Grant Cluster Access to a User step in *Managing Users in PKS with UAA*. If you do not use `-p` to provide a password, the PKS CLI prompts for the password interactively. Pivotal recommends running the login command without the `-p` flag for added security.
- `CERTIFICATE-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 CERTIFICATE-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 your `~/.pks` directory. You can use the `PKS_HOME` environment variable to override this location and use `creds.yml` from any directory.

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 `kubectl.exe`.

6. Move the binary into your `PATH`.

Upgrading PKS Overview

Page last updated:

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

- What Happens During PKS Upgrades
- Upgrading PKS
- Upgrading PKS with NSX-T
- Maintaining Workload Uptime
- Configuring the Upgrade Pipeline

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 Upgrading 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 Maintaining 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 the Upgrade the PKS Tile section of *Upgrading 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.

Upgrading 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.

Breaking Change: PKS v1.1 does not support ABAC. Delete any ABAC clusters before upgrading to v1.1. For more information, see Existing ABAC Clusters in the *PKS v1.1 Release Notes*.

WARNING: Do not manually upgrade your Kubernetes version. The PKS service includes the compatible Kubernetes version.

Prepare to Upgrade

Before you begin upgrading the PKS tile, consider your workload capacity and uptime requirements. If workers are operating too close to their capacity, the PKS upgrade can fail. View your workload resource usage in Dashboard. For more information, see Accessing 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 `pks-resize` or create a cluster using a larger plan. For more information, see Scaling 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 Maintaining Workload Uptime.

Step 1: Upgrade Ops Manager

PKS v1.1 requires Ops Manager v2.1.

1. To upgrade to the required Ops Manager version, follow the procedure detailed in: Upgrade Ops Manager and Installed Products to v2.1.
2. Operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see About Workload Upgrades in *Maintaining Workload Uptime* and Creating Clusters.

You can monitor the PKS control plane VM by clicking the **Pivotal Container Service tile**, selecting **Status** tab, and reviewing the **Pivotal Container Service** VM's data points. If any data points are at capacity, scale your deployment accordingly.

Step 2: Upgrade the PKS Tile

To upgrade PKS, you follow the same Ops Manager process that you use to install the tile for the first time. Your configuration settings migrate to the new version automatically. To perform an upgrade:

1. Review the Release Notes for the version you are upgrading to.
2. Download the desired version of the product from Pivotal Network.
3. Navigate to the Ops Manager Installation Dashboard and click **Import a Product** to upload the product file.
4. Under the **Import a Product** button, click **+** next to **Pivotal Container Service**. This adds the tile to your staging area.
5. Click the newly-added **Pivotal Container Service** tile.
6. Optional: To upgrade all PKS-deployed Kubernetes clusters when you upgrade the PKS tile, follow the next steps:
    a. Click **Errands**.
    b. Under **Post-Deploy Errands**, set the **Upgrade all clusters errand** to **Default (On)**. The errand upgrades a single Kubernetes cluster at a time. Upgrading PKS Kubernetes clusters can temporarily interrupt the service, as described below.

        Note: If you upgrade PKS from 1.0.x to 1.1, you must enable the **Upgrade All Cluster** errand. This ensures existing clusters can perform

resize or delete actions after the upgrade.

(Optional) To monitor the **Upgrade all clusters errand** using the BOSH CLI, do the following:

i. Log in to the BOSH Director by running `bosh -e MY-ENVIRONMENT log-in` from a VM that can access your PKS deployment. For more information, see Managing 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. For example, if you use NSX-T, follow the instructions in Upgrade NSX-T before clicking **Apply Changes** for the PKS tile upgrade.

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.

1. 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.

2. Click **Apply Changes**.

3. Operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see About Workload Upgrades in *Maintaining Workload Uptime* and Creating Clusters.

You can monitor the PKS control plane VM by clicking the**Pivotal Container Service tile**, selecting **Status** tab, and reviewing the **Pivotal Container Service** VM's data points. If any data points are at capacity, scale your deployment accordingly.

Step 3: Upgrade NSX-T (Optional)

If you are deploying PKS on vSphere with NSX-T integration, NSX-T v2.1 is required.

To upgrade PKS with NSX-T, make the following configuration changes to adapt your deployment to new features that have been added in PKS 1.1.0.

1. Create the **NSX Manager Super User Principal Identity Certificate** by following the procedure in Generating and Registering Certificates.

2. Select the NAT option if Network Address Translation needs to be enforced for the Kubernetes nodes. Clearing this option allows the nodes to have globally routable IP addresses. For more information, see NAT Topology.

3. PKS 1.1 allows you to specify dedicated IP blocks for node and pod networking. Create these IP blocks according to the instructions in Plan IP Blocks of *Preparing NSX-T Before Deploying PKS*. Enter one or more domain servers used by Kubernetes nodes. These domain servers will be used by the nodes that are created on the Node Networks that are dynamically generated at the time of cluster creation.

   Note: When upgrading NSX-T for PKS, you must use a different CIDR range for the node IP block than the one you used for the service network.

4. (Optional) To configure a global proxy for all outgoing HTTP/HTTPS traffic from your Kubernetes clusters, do the following:

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

5. Make sure that the **Enable outbound internet access** checkbox is not selected. This setting is not applicable to vSphere without NSX-T integrations.

Step 4: Upgrade vCenter (Optional)

1. If you are deploying PKS on vSphere, consult vSphere Version Requirements and upgrade vSphere if necessary.
2. Operators should add additional workloads and create an additional cluster to ensure that the PKS control plane is still functional. For more information on performing those actions, see About Workload Upgrades in *Maintaining Workload Uptime* and Creating Clusters.

You can monitor the PKS control plane VM by clicking the **Pivotal Container Service tile**, selecting **Status** tab, and reviewing the **Pivotal Container Service** VM's data points. If any data points are at capacity, scale your deployment accordingly.

Upgrade Kubernetes Clusters

If you set the **Upgrade all clusters errand** to **Default (On)**, your PKS-deployed Kubernetes clusters are upgraded automatically when the PKS tile upgrade runs.

If you set the **Upgrade all clusters errand** to **Off**, you can upgrade all PKS-deployed Kubernetes clusters by setting the **Upgrade all clusters errand** to **On** and clicking **Apply Changes**.

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

Service Interruptions

Service changes and upgrades and failures at the process, VM, and IaaS level can cause outages in the PKS service, as described below.

Read this section if:

- You are experiencing a service interruption and are wondering why.
- You are planning to update or change a Kubernetes cluster and want to know if it might cause a service interruption.

Stemcell or Service Upgrade

An operator updates a stemcell version or the PKS tile version.

- **Impact:** The PKS API experiences downtime while the new stemcell is applied to the Pivotal Container Service VM.
  - **Required Actions:** None. If the update deploys successfully, apps reconnect automatically.
- **Impact:** Workloads running on single node clusters experience downtime.
  - **Required Actions:** None. If the update deploys successfully, workloads resume automatically. For more information, see Maintaining Workload Uptime.

Upgrading PKS with NSX-T

Page last updated:

This topic explains how to upgrade the Pivotal Container Service (PKS) tile for environments using vSphere with NSX-T.

PKS v1.1.5 supports NSX-T 2.2 and vSphere 6.5 U2. For details, see the VMware Product Interoperability Matrix for PKS in the VMware documentation.

We recommend that you upgrade to PKS v1.1.5 and NSX-T 2.2 to take advantage of key features and important architectural changes. For more information, see NSX-T Architectural Changes in the PKS v1.1.5 release notes.

Note: When you upgrade PKS on vSphere with NSX-T, workloads in your Kubernetes cluster are unavailable while the NSX Edge nodes run the upgrade. Configure NSX Edge for high availability using Active/Standby mode to avoid workload downtime. For more information, see the Configure NSX Edge for High Availability (HA) section of *Preparing NSX-T Before Deploying PKS*.

Prepare to Upgrade

Before you begin upgrading the PKS tile, follow the steps below:

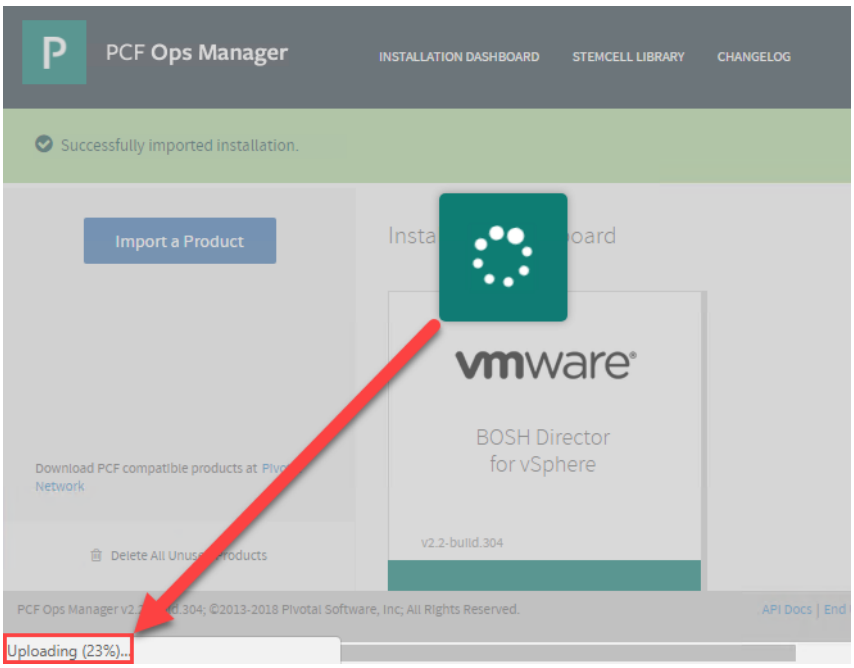
1. Review the Release Notes for the version you are upgrading to.
2. Verify the health of your Kubernetes environment by following the steps below:
    a. To verify that all nodes are in a ready state, run `kubectl get nodes` for all Kubernetes contexts.
    b. To verify that all pods are running, run `kubectl get pods --all-namespaces` for all Kubernetes contexts.
    c. To verify that all the processes are in a running state, run `bosh -d MY-DEPLOYMENT instances --ps` for each deployment, replacing `MY-DEPLOYMENT` with the deployment name. Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.
3. Make sure there are no issues with vSphere by following the steps below:
    a. Verify that datastores have enough space.
    b. Verify that hosts have enough memory.
    c. Verify that there are no alarms.
    d. Verify that hosts are in a good state.
4. (Optional) Back up the environment using the procedures in the following topics:
    - Backup PKS
    - Backup NSX-T
    - Backup vCenter

    Note: If you choose not to back up PKS, NSX-T, or vCenter, we recommend backing up the NSX-T and NSX-T Container Plugin (NCP) logs. For more information, see PKS Logs for NSX-T and NCP below.

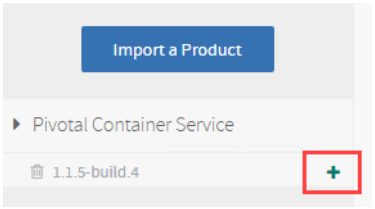
Upgrade the PKS Tile

To upgrade PKS, you follow the same Ops Manager process that you use to install the tile for the first time. Your configuration settings migrate to the new version automatically. Follow the steps below 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**.
4. Browse to the PKS product file and select it. Uploading the file takes several minutes.

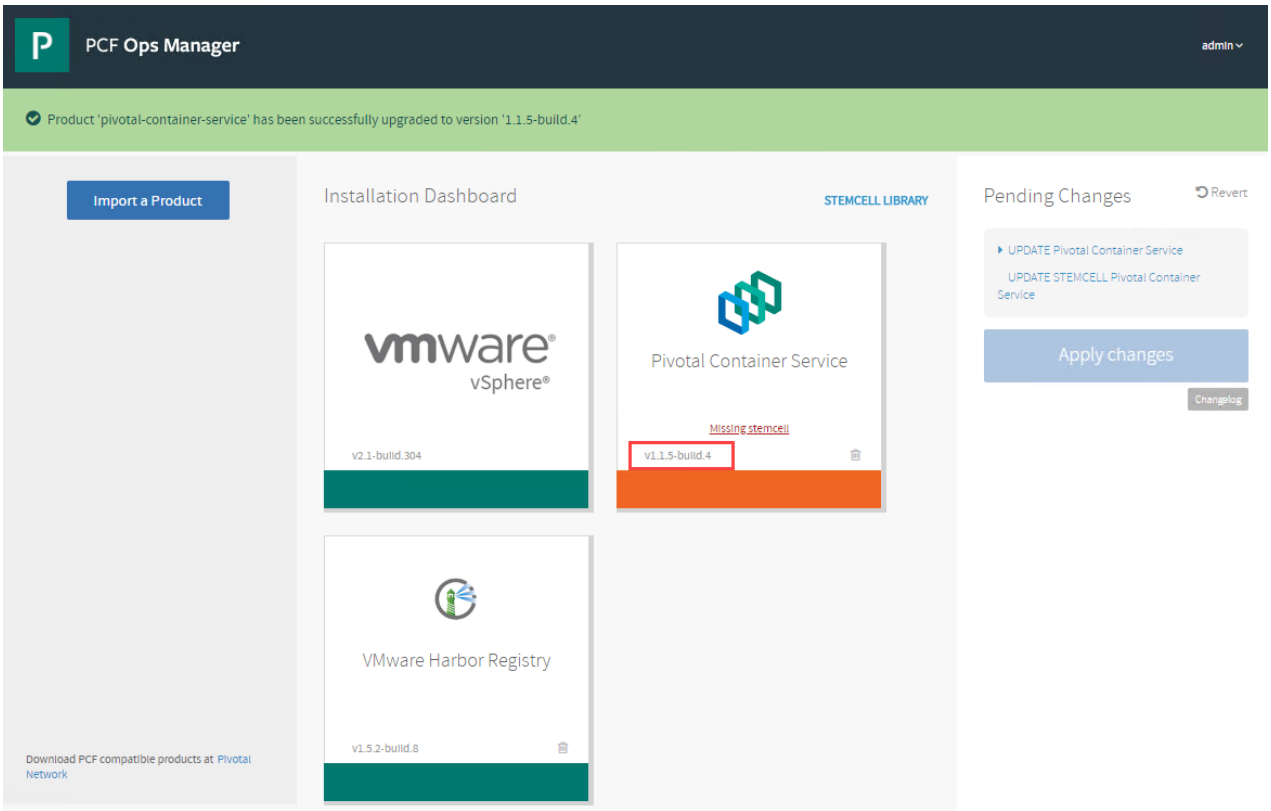


5. Under the **Import a Product** button, click **+** next to **Pivotal Container Service**. This adds the tile to your staging area.

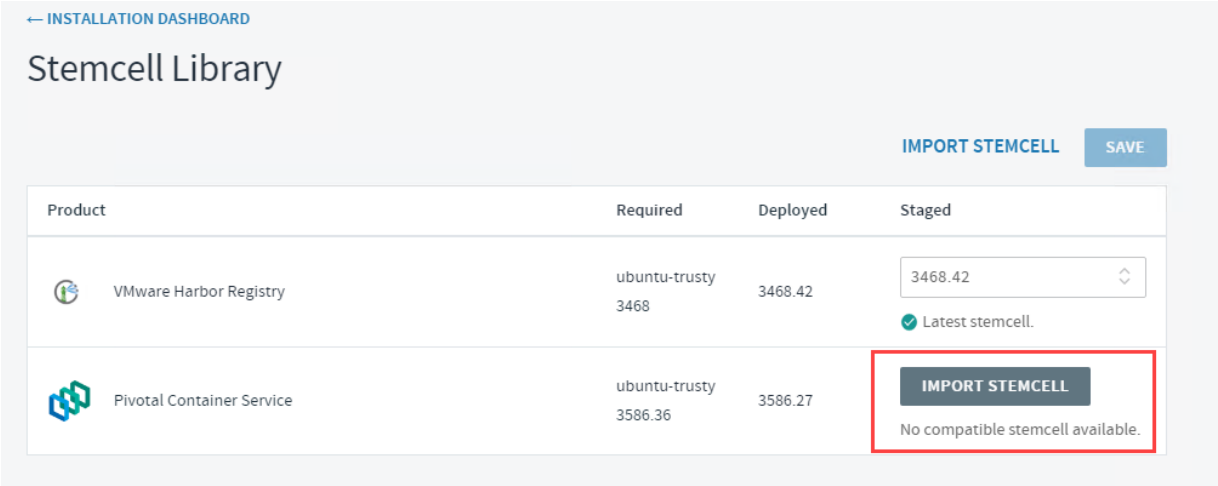


6. Ops Manager adds the PKS tile to your staging area.

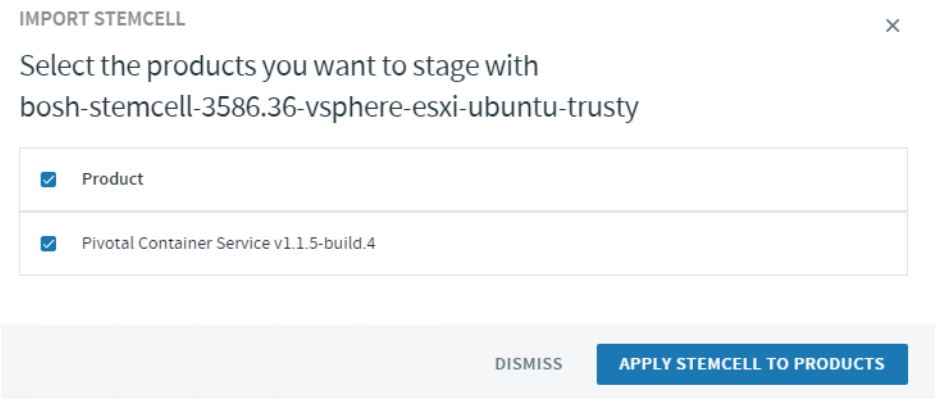
7. (Optional) If the stemcell is not current, click the **Missing stemcell** link and follow the steps below:



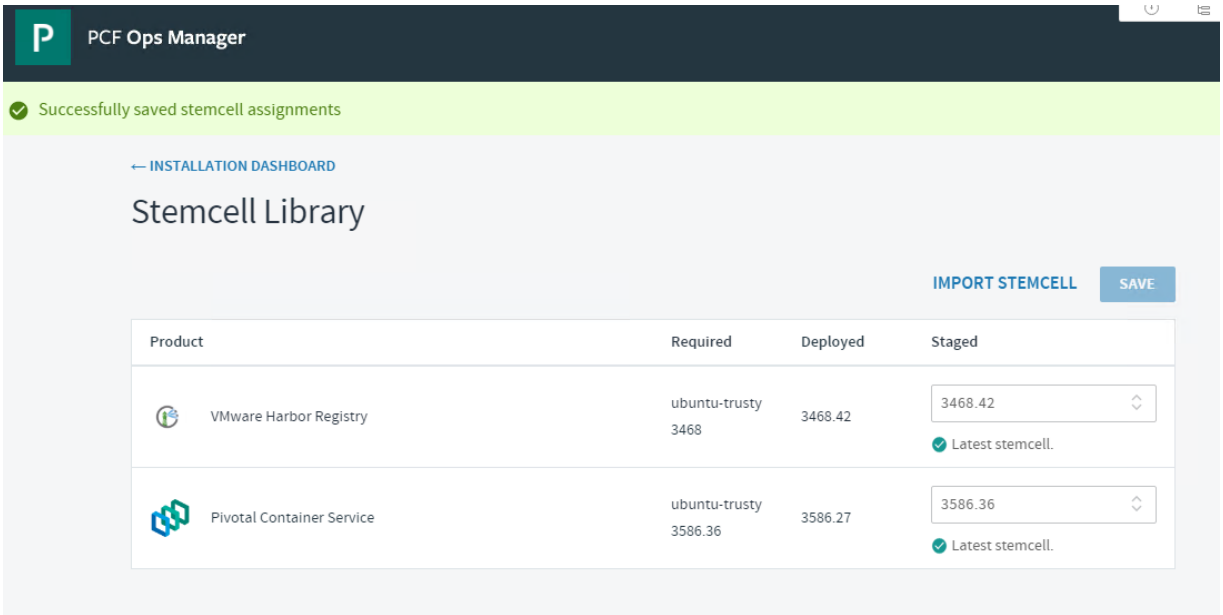
a. On the Stemcell Library page, click **Import Stemcell**.



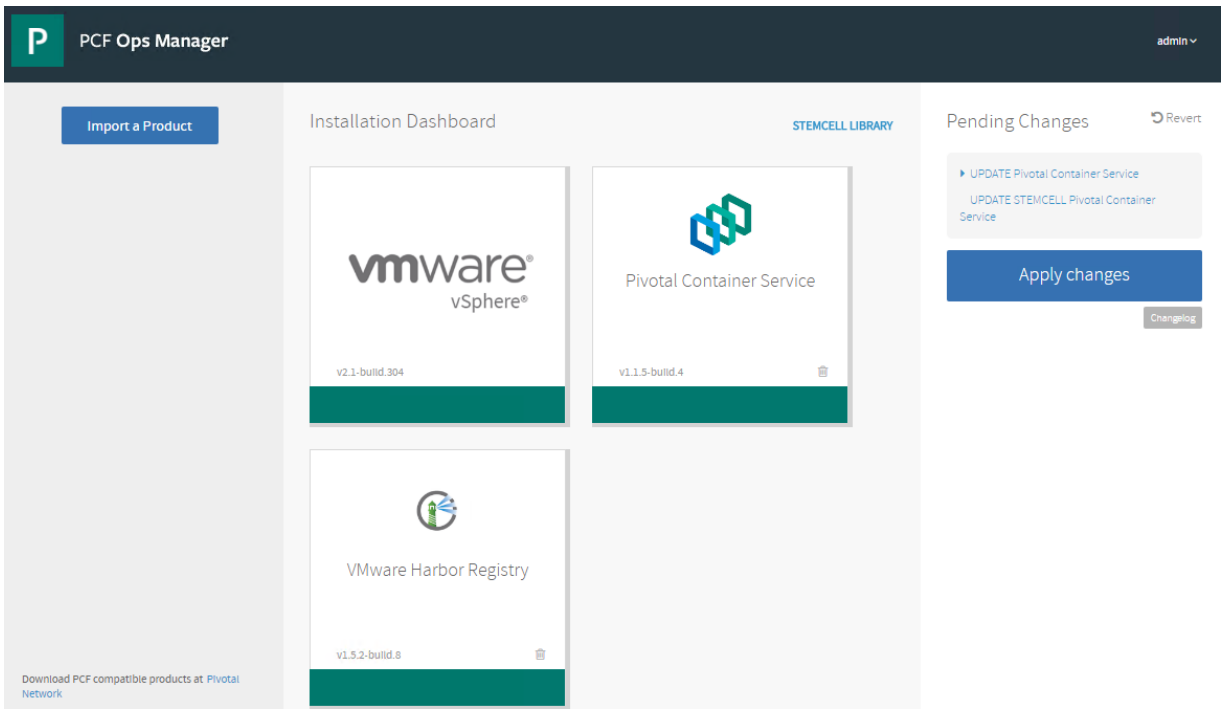
b. Select the PKS product and click **Apply Stemcell to Products**.



c. Verify that Ops Manager successfully applied the stemcell.



d. Select the **Installation Dashboard** link to return to the Installation Dashboard.



Increase the Kubernetes Worker Node VM Size

The default Kubernetes worker node VM size provides insufficient disk space for PKS v1.1.5 or later on vSphere with NSX-T.

Note: If you do not increase the size of the Kubernetes worker node VM before you upgrade, the VM can run out of ephemeral disk space and cause the upgrade to fail.

Follow the steps below to increase the Kubernetes worker node VM size:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the **Pivotal Container Service** tile.
3. Click **Plan 1**.
4. Under **Worker VM Type**, select a VM type with a minimum disk size of 16 GB.
5. (Optional) If you have additional plans configured, repeat this procedure for each plan.

Verify NSX-T Manager CA Certificate Settings

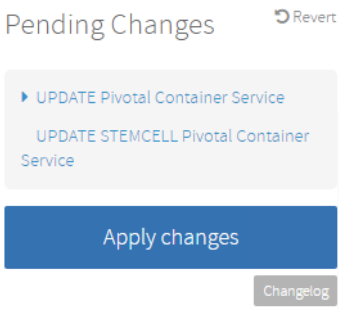
Follow the steps below:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the **Pivotal Container Service** tile.
3. Click **Networking**.
4. Under **NSX Manager CA Cert**, choose one of the following options:
    - Confirm that you have a valid NSX-T Manager CA certificate.
    - Select **Disable SSL certificate verification**.

Note: You cannot choose both options. If you provide an NSX Manager CA certificate and also disable SSL certificate verification, the PKS upgrade fails.

Apply Changes to the PKS Tile

Return to the Installation Dashboard and do one of the following:



- **Apply Changes**: Under **Pending Changes**, click **Apply Changes**.
- **Review Pending Changes [BETA]**:

    1. Click **Review Pending Changes**.
    2. Click **Pivotal Container Service**.
    3. Click **Apply Changes**.

Verify the Upgrade

After you apply changes to the PKS tile and the upgrade is complete, verify that your Kubernetes environment is healthy and confirm that NCP is running on the master node VM.

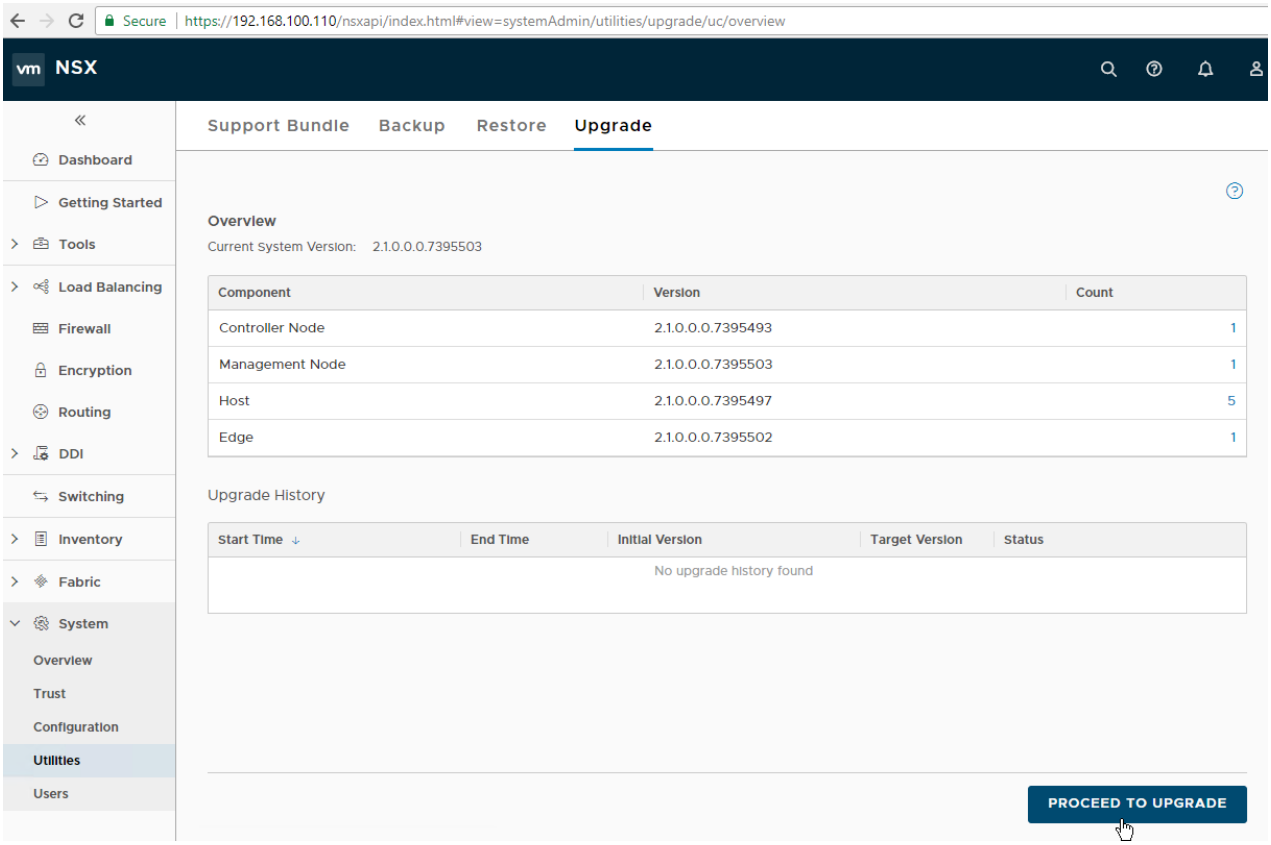
To verify the health of your Kubernetes environment and NCP, see Verifying Deployment Health.

(Optional) Upgrade NSX-T

Upgrading to NSX-T from 2.1 to 2.2 is optional but strongly recommended. Support for NSX-T 2.1 will be removed in a future PKS release.

Follow the steps below to upgrade NSX-T:

1. Log in to the NSX Manager UI and navigate to **System > Utilities > Upgrade**.
2. Click **Proceed to Upgrade** and follow the instructions. The NSX-T Upgrade wizard walks you through the process of upgrading from NSX-T 2.1 to NSX-T 2.2.



For more information, see Upgrading NSX-T in the VMware documentation.

(Optional) Upgrade vSphere

Upgrade vSphere from version 6.5 or 6.5 U1 to 6.5 U2.

For more information, see Upgrading vSphere in an NSX Environment in the VMware documentation.

Maintaining Workload Uptime

Page last updated:

This topic describes how you can maintain workload uptime for Kubernetes clusters deployed with Pivotal Container Service (PKS).

To maintain workload uptime, configure the following settings in your deployment manifest:

1. Configure workload replicas to handle traffic during rolling upgrades.
2. Define an anti-affinity rule to evenly distribute workloads across the cluster.

To increase uptime, you can also refer to the documentation for the services that run on your clusters, and configure your workload based on the recommendations of the software vendor.

About Workload Upgrades

The PKS tile contains an errand that upgrades all Kubernetes clusters. Upgrades run on a single VM at a time. While one worker VM runs an upgrade, the workload on that VM goes down. The additional worker VMs continue to run replicas of your workload, maintaining the uptime of your workload.

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 and using multiple replicas of your workloads spread across those VMs. You must edit your manifest to define the replica set and configure an anti-affinity rule to ensure that the replicas run on separate worker nodes.

Set Workload Replicas

Set the number of workload replicas to handle traffic during rolling upgrades. To replicate your workload on additional worker VMs, deploy the workload using a replica set.

Edit the `spec.replicas` value in your deployment manifest:

```
kind: Deployment
metadata:
  # ...
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: APP-NAME
```

See the following table for more information about this section of the manifest:

Key-Value Pair	Description
`spec:` `  replicas: 3`	Set this value to at least 3 to have at least three instances of your workload running at any time.
`app: APP-NAME`	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
`podAntiAffinity: requiredDuringSchedulingIgnoredDuringExecution`	• When you set `podAntiAffinity` to the `requiredDuringSchedulingIgnoredDuringExecution` value, the pod is eligible to be scheduled only on worker nodes that are not running a replica of this pod. If the requirement cannot be met, scheduling fails. • Alternatively, you can set `podAntiAffinity` to the `preferredDuringSchedulingIgnoredDuringExecution` value. With this rule, the scheduler tries to schedule pod replicas on different worker nodes. If it is not possible, the scheduler assigns more than one pod to the same worker node.
`matchExpressions: - key: "app"`	This value matches `spec.template.metadata.labels.app`.
`values: - APP-NAME`	This value matches the `APP-NAME` you defined earlier in the spec.

Multi-AZ Worker

Kubernetes evenly spreads pods in a replication controller over multiple Availability Zones (AZs). For more granular control over scheduling pods, add an `Anti-Affinity Rule` to the deployment spec by replacing `"kubernetes.io/hostname"` with `"failure-domain.beta.kubernetes.io/zone"`.
For more information on scheduling pods, see Advanced Scheduling in Kubernetes on the Kubernetes Blog.

Persistent Volumes

Persistent volumes cannot be attached across AZs. Therefore, when persistent volumes are created, the `PersistentVolumeLabel` admission controller automatically adds AZ labels to them. The scheduler then ensures that pods that claim a given volume are only placed into the same AZ as that volume.

If an AZ goes down, the persistent volume along with its data also goes down and cannot be automatically re-attached. To preserve your persistent volume data in the event of a fallen AZ, your persistent workload needs to have a failover mechanism in place.

For example, to ensure the uptime of your persistent volumes during a cluster upgrade, Pivotal recommends that you have at least two nodes per AZ. By configuring your workload as suggested, Kubernetes reschedules pods in the other node of the same AZ while BOSH is performing the upgrade.

Configuring the Upgrade Pipeline

Page last updated:

This topic describes how to set up a Concourse pipeline to perform automatic upgrades of a Pivotal Container Service (PKS) installation.

When you configure the upgrade pipeline, the pipeline upgrades your installation when a new PKS release becomes available on Pivotal Network.

By default, the pipeline upgrades when a new major patch version is available.

For more information about configuring and using Concourse for continuous integration (CI), see the Concourse documentation.

Download the Upgrade Pipeline

Perform the following steps:

1. From a browser, log in to Pivotal Network.
2. Navigate to the **PCF Platform Automation with Concourse** product page to download the upgrade-tile pipeline.

   Note: If you cannot access PCF Platform Automation with Concourse on Pivotal Network, contact Pivotal Support.

3. (Optional) Edit params.yml to configure the pipeline.
   - For example, edit the `product_version_regex` value to follow minor version updates.
4. Set the pipeline using the `fly` CLI for Concourse. See the upgrade-tile pipeline documentation for more information.

Managing PKS

Page last updated:

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

- Configuring PKS API Access
- Managing Users in PKS with UAA
- Managing PKS Deployments with BOSH
- Configuring a GCP Load Balancer for PKS Clusters
- Adding Custom Workloads
- Verifying Deployment Health
- Service Interruptions
- Viewing and Exporting Usage Data
- Downloading Cluster Logs
- Deleting PKS

Configuring 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** pane into a file.
2. Target your UAA server by running the following command:

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

    Replace the following values:

    - `PKS-API`: enter the fully qualified domain name (FQDN) you use to access the PKS API. You configured this URL in the PKS API section of *Installing PKS* for your IaaS. For example, see Installing PKS on vSphere.
    - `ROOT-CA-FILENAME`: enter the path for the certificate file you downloaded in a previous step. For example:

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

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

Log in to the PKS CLI as a User

For information about logging in to the PKS CLI as a user, see the Log in to PKS CLI section of *Installing the PKS CLI*.

Log in to PKS as an Automated Client

On the command line, run the following command to log in to the PKS CLI as an automated client for a script or service:

```
pks login -a PKS-API --client-name CLIENT-NAME --client-secret CLIENT-SECRET --ca-cert CERTIFICATE-PATH
```

Where:

- `PKS-API` is the domain name for the PKS API that you entered in **Ops Manager** > **Pivotal Container Service** > **PKS API** > **API Hostname (FQDN)**. For example, `api.pks.example.com`.
- `CLIENT-NAME` is your OAuth client ID.
- `CLIENT-SECRET` is your OAuth client secret.
- `CERTIFICATE-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 \
--client-name automated-client \
--client-secret randomly-generated-secret \
--ca-cert /var/tempest/workspaces/default/root_ca_certificate
```

Managing Users in PKS with UAA

Page last updated:

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

How to Use UAAC

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

To run UAAC commands from the Ops Manager VM, see the following SSH procedures for vSphere or Google Cloud Platform (GCP).

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

SSH into the Ops Manager VM on vSphere

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

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

1. From a command line, run the following command to SSH into the Ops Manager VM:

   ```
   ssh ubuntu@OPS-MANAGER-FQDN
   ```

   Where `OPS-MANAGER-FQDN` is the fully qualified domain name (FQDN) 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 with UAAC.

SSH into the Ops Manager VM on GCP

To SSH into the Ops Manager VM in GCP, do the following:

1. Confirm that you have installed the gcloud CLI. See Downloading gcloud in 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 with UAAC.

Log in as a UAA 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 of Ops Manager and click the **Pivotal Container Service** tile.
2. Click **Credentials**.
3. To view the secret, click **Link to Credential** next to **Pks Uaa Management Admin Client**. The client username is `admin`.
4. On the command line, run the following command to target your UAA server:

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

   Where:

   - `PKS-API` is the URL to your PKS API server. You configured this URL in the PKS API section of *Installing PKS* for your IaaS. For example, see Installing PKS on vSphere.
   - `ROOT-CA-FILENAME` is the certificate file you downloaded in Configuring PKS API Access.

   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. Run the following command to authenticate with UAA using the secret you retrieved in a previous step:

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

   Where `ADMIN-CLIENT-SECRET` is your PKS UAA management admin client secret.

Grant PKS Access

PKS access gives users the ability to deploy and manage Kubernetes clusters. As an Admin user, you can assign the following UAA scopes to users, external LDAP groups, and clients:

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

Grant PKS Access to a User

You can create a new UAA user with PKS access by performing the following steps:

1. Log in as the UAA admin using the procedure in Log in as a UAA Admin.
2. To create a new user, run the following command:

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

   For example:

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

3. Run the following command to assign a scope to the user to allow them to access Kubernetes clusters:

   ```
   uaac member add UAA-SCOPE USERNAME
   ```

   Where `UAA-SCOPE` is one of the UAA scopes defined in Grant PKS Access. For example:

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

Grant Control Plane Access to an External LDAP Group

Connecting PKS to a LDAP external user store allows the User Account and Authentication (UAA) server to delegate authentication to existing enterprise user stores.

Note: When integrating with an external identity provider such as LDAP, authentication within the UAA becomes chained. UAA first attempts to authenticate with a user's credentials against the UAA user store before the external provider, LDAP. For more information, see Chained Authentication in the *User Account and Authentication LDAP Integration* GitHub documentation.

For more information about the process used by the UAA Server when it attempts to authenticate a user through LDAP, see the Configuring LDAP Integration with Pivotal Cloud Foundry Knowledge Base article.

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

To grant control plane access to an external LDAP group, perform the following steps:

1. Log in as the UAA admin using the procedure in Log in as a UAA Admin.
2. To assign the `pks.clusters.manage` scope to all users in an LDAP group, run the following command:

   ```
   uaac group map --name pks.clusters.manage GROUP-DISTINGUISHED-NAME
   ```

   Where `GROUP-DISTINGUISHED-NAME` is the LDAP Distinguished Name (DN) for the group. For example:

   ```
   $ uaac group map --name pks.clusters.manage cn=operators,ou=groups,dc=example,dc=com
   ```

   For more information about LDAP DNs, see the LDAP DNs and RDNs in the LDAP documentation.
3. (Optional) To assign the `pks.clusters.admin` scope to all users in an LDAP group, run the following command:

   ```
   uaac group map --name pks.clusters.admin GROUP-DISTINGUISHED-NAME
   ```

   Where `GROUP-DISTINGUISHED-NAME` is the LDAP DN for the group. For example:

   ```
   $ uaac group map --name pks.clusters.admin cn=operators,ou=groups,dc=example,dc=com
   ```

Grant Cluster Access to a Client

To grant cluster access to an automated client for a script or service, perform the following steps:

1. Log in as the UAA admin using the procedure Log in as a UAA Admin.
2. Run the following command to create a client with the desired scopes:

   ```
   uaac client add CLIENT-NAME -s CLIENT-SECRET \
   --authorized_grant_types client_credentials \
   --authorities UAA-SCOPES
   ```

   Where:

   - `CLIENT-NAME` and `CLIENT-SECRET` are the client credentials.
   - `UAA-SCOPES` is with one or more of the UAA scopes defined in Grant Cluster Access, separated by a comma. For example:

   ```
   $ uaac client add automated-client \
   -s randomly-generated-secret
   --authorized_grant_types client_credentials  \
   --authorities pks.clusters.admin,pks.clusters.manage
   ```

Managing PKS Deployments with BOSH

Page last updated:

This topic describes how to manage your Pivotal Container Service (PKS) deployment using BOSH.

Set a BOSH Environment Alias

To set a BOSH alias for your PKS deployment environment, follow the steps below:

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. To create a BOSH alias for your PKS environment, run the following command:

   ```
   bosh alias-env ENVIRONMENT \
   -e BOSH-DIRECTOR-IP \
   --ca-cert /var/tempest/workspaces/default/root_ca_certificate
   ```

   Where:

   - `ENVIRONMENT` is an alias of your choice. For example, `pks`.
   - `BOSH-DIRECTOR-IP` is the BOSH Director IP address you located in the first step. For example, `10.0.0.3`.

   For example:

   ```
   $ bosh alias-env pks -e 10.0.0.3 \
   --ca-cert /var/tempest/workspaces/default/root_ca_certificate
   ```

3. To log in to the BOSH Director using the alias you set, run `bosh -e ENVIRONMENT login`.

   For example:

   ```
   $ bosh -e pks login
   ```

SSH into the PKS VM

To SSH into the PKS VM using BOSH, follow the steps below:

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use BOSH CLI to log in to the BOSH Director from the Ops Manager VM. For more information, see Advanced Troubleshooting with the BOSH CLI.

2. Identify the name of your PKS deployment by running `bosh -e ENVIRONMENT deployments`, where `ENVIRONMENT` is the alias you set in Set a BOSH Environment Alias.

   For example:

   ```
   $ bosh -e pks deployments
   ```

   Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. Identify the name of your PKS VM by running `bosh -e ENVIRONMENT -d DEPLOYMENT vms`, where `DEPLOYMENT` is your PKS deployment name.

   For example:

   ```
   $ bosh -e pks -d pivotal-container-service/a1b2c333d444e5f66a77 vms
   ```

   Your PKS VM name begins with `pivotal-container-service` and includes a BOSH-generated hash. This value is different from the hash in your PKS deployment name.

4. SSH into the PKS VM by running `bosh -e ENVIRONMENT -d DEPLOYMENT ssh PKS-VM`, where `PKS-VM` is your PKS VM name.

For example:

```
$ bosh -e pks \
 -d pivotal-container-service/a1b2c333d444e5f66a77 \
 ssh pivotal-container-service/000a1111-222b-3333-4cc5-de66f7a8899b
```

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 `pks 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 in *Creating Clusters*.
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**.

Adding 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
```

Verifying Deployment Health

Page last updated:

This topic describes how to verify the health of your Pivotal Container Service (PKS) deployment.

For the BOSH CLI commands in this topic, replace the text as follows:

- `MY-ENV`: the alias you set for your BOSH Director. For more information, see Managing PKS Deployments with BOSH.
- `MY-DEPLOYMENT`: the name of your PKS deployment. PKS deployment names begin with `pivotal-container-service` and include a unique BOSH-generated hash.
- `VM-NAME`: your Kubernetes master node VM name.
- `ID`: your Kubernetes master node VM ID. This is a unique BOSH-generated hash.

Verify Kubernetes Health

Verify the health of your Kubernetes environment by following the steps below:

1. To verify that all nodes are in a ready state, run the following command for all Kubernetes contexts:

   ```
   kubectl get nodes
   ```

2. To verify that all pods are running, run the following command for all Kubernetes contexts:

   ```
   kubectl get pods --all-namespaces
   ```

3. To verify that all the processes are in a running state, run the following command for each deployment:

   ```
   bosh -d MY-DEPLOYMENT instances --ps
   ```

   For example:

   ```
   $ bosh -d pivotal-container-service/a1b2c333d444e5f66a77 instances --ps
   ```

Verify NCP Health (NSX-T Only)

In PKS v1.1.5 and later on vSphere with NSX-T, NCP runs as a BOSH host process. Each Kubernetes master node VM has one NCP process running. If your cluster has multiple master nodes, one NCP process is active while the others are on standby. For more information, see Architectural Changes.

Verify NCP health by following the steps below:

1. From the Ops Manager VM, run the following command:

   ```
   bosh -e MY-ENV login
   ```

   For example:

   ```
   $ bosh -e pks login
   ```

2. To locate the Kubernetes master node VM name and ID, run the following command:

   ```
   bosh -e MY-ENV -d MY-DEPLOYMENT vms
   ```

   For example:

   ```
   $ bosh -e pks -d pivotal-container-service/a1b2c333d444e5f66a77 vms
   ```

   Your PKS API VM name begins with `pivotal-container-service` and includes a BOSH-generated hash. This value is different from the deployment hash.

3. To SSH into the Kubernetes master node VM, run the following command:

   ```
   bosh -e MY-ENV -d MY-DEPLOYMENT ssh VM-NAME/ID
   ```

   For example:

   ```
   $ bosh -e pks \
   -d pivotal-container-service/a1b2c333d444e5f66a77 \
   ssh pivotal-container-service/000a1111-222b-3333-4cc5-de66f7a8899b
   ```

4. From the master node VM, run the following command:

   ```
   monit summary
   ```

   Verify that you see `Process: 'ncp'` is `running`.

5. To check if the NCP process is active or on standby, run the following command:

   ```
   /var/vcap/jobs/ncp/bin/nsxcli -c get ncp-master status
   ```

6. To restart the NCP process, run the following command:

   ```
   monit restart ncp
   ```

7. To verify that the NCP process restarts successfully, run the following command:

   ```
   monit summary
   ```

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

Viewing and Exporting Usage Data

Page last updated:

Note: The procedures in this topic apply to PKS v1.2 and earlier only. To view usage data in later versions of PKS, see the Viewing Usage Data topic in the documentation for PKS v1.3 or later.

This topic describes how operators can view and export usage information from their Pivotal Container Service (PKS) deployment. Operators can use this data to calculate billed usage, perform customer chargebacks, and generate usage reports.

The PKS database stores the following pod usage data:

- **Watermark**: the number of pods that run at a single time.
- **Consumption**: the memory and CPU usage of pods.

To extract either type of usage data from your PKS deployment, you must `bosh ssh` into your PKS VM.

About Usage Data

This section describes the usage data records you can view and export from the PKS VM. The agent pod collects both watermark and consumption data for the deployment and sends the data to the PKS aggregator agent. The aggregator agent then stores the data in the PKS database. You can access the PKS database from the PKS VM and export the usage data for your deployment.

Watermark Data

The PKS database stores comma-separated watermark data about the number of pods that run simultaneously in your PKS deployment. You can view the watermark data on the command line or export it to a comma-separated values (`.csv`) file.

The following is an example of a watermark usage data export:

```
id,collect_time,day,month,year,agent,agent_pod_cnt,total_pod_cnt,high_watermark_since_install_pod_cnt,high_watermark_since_install_date
1,2018-06-08 01:16:22,7,6,2018,Service-instance_61916de4-8abe-4ec7-a67b-e2568c83dbe0,1,1,1,2018-06-08 01:16:22
2,2018-06-09 01:16:24,8,6,2018,service-instance_61916de4-8abe-4ec7-a67b-e2568c83dbe0,1,1,1,2018-06-08 01:16:22
3,2018-06-10 01:16:34,9,6,2018,service-instance_61916de4-8abe-4ec7-a67b-e2568c83dbe0,1,1,1,2018-06-08 01:16:22
4,2018-06-12 01:12:25,11,6,2018,service-instance_61916de4-8abe-4ec7-a67b-e2568c83dbe0,1,1,1,2018-06-12 01:12:25
5,2018-06-26 01:16:38,25,6,2018,service-instance_748709f7-41be-4c5f-9123-78874caeb602,3,3,3,2018-06-26 01:16:38
6,2018-06-28 01:16:59,27,6,2018,service-instance_748709f7-41be-4c5f-9123-78874caeb602,3,3,3,2018-06-28 01:16:59
```

The following table describes the database fields related to watermark usage data:

Field Name	Description
id	Unique record identifier
collect_time	The date when the agent collects the record
day	The day that corresponds to the record
month	The month that corresponds to the record
year	The year that corresponds to the record
agent	The name of the pod that contains the agent
agent_pod_cnt	The maximum number of pods in the cluster on the given day
total_pod_cnt	The maximum number of pods in the deployment on the given day
high_watermark_since_install_pod_cnt	The maximum number of pods in the deployment since installation
high_watermark_since_install_date	The date when the agent logs the maximum number of pods in the deployment

Consumption Data

The PKS database stores resource consumption data for all pods in a deployment. You can view the uptime and pod count for each cluster in your PKS deployment as well as memory and CPU usage for each pod by accessing the database on the command line.

The following table describes the database fields related to consumption usage data:

Field Name	Description
id	Unique record identifier
agent	Unique cluster name
collect_time	The date when the agent collects the record
pod_id	Unique pod identifier
pod_name	Unique pod name
memory_used	Pod memory usage
cpu_used	Pod CPU usage
pod_cnt	The number of pods in the cluster

SSH into the PKS VM

To SSH into your PKS VM using BOSH, follow the steps below:

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use BOSH CLI to log in to the BOSH Director from the Ops Manager VM. For more information, see Advanced Troubleshooting with the BOSH CLI.

2. Identify the name of your PKS deployment by running `bosh -e ENVIRONMENT deployments`, where `ENVIRONMENT` is the alias you set in Set a BOSH Environment Alias.

   For example:

   ```
   $ bosh -e pks deployments
   ```

   Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. Identify the name of your PKS VM by running `bosh -e ENVIRONMENT -d DEPLOYMENT vms`, where `DEPLOYMENT` is your PKS deployment name.

   For example:

   ```
   $ bosh -e pks -d pivotal-container-service/a1b2c333d444e5f66a77 vms
   ```

   Your PKS VM name begins with `pivotal-container-service` and includes a BOSH-generated hash. This value is different from the hash in your PKS deployment name.

4. SSH into the PKS VM by running `bosh -e ENVIRONMENT -d DEPLOYMENT ssh PKS-VM`, where `PKS-VM` is your PKS VM name.

   For example:

   ```
   $ bosh -e pks \
    -d pivotal-container-service/a1b2c333d444e5f66a77 \
    ssh pivotal-container-service/000a1111-222b-3333-4cc5-de66f7a8899b
   ```

View and Export Watermark Usage Data

To view and export the watermark usage data for your PKS deployment, follow the steps below:

1. From the PKS VM, create a new file named `print-watermark.sh`.

2. Paste the following contents into the `print-watermark.sh` file:

```
#! /bin/bash

set -e

db_username="$(grep 'DBName: telemetry' -A2 /var/vcap/jobs/mysql/config/mariadb_ctl_config.yml | grep 'User' | tr -d ' ' | cut -d':' -f2)"
db_password="$(grep 'DBName: telemetry' -A2 /var/vcap/jobs/mysql/config/mariadb_ctl_config.yml | grep 'Password' | tr -d ' ' | cut -d':' -f2)"

mysql_cmd="/var/vcap/packages/mariadb/bin/mysql -u$db_username -h 127.0.0.1 -p$db_password"

watermark_select_result="$($mysql_cmd --execute="use telemetry; select * from pkswatermark order by collect_time")"

watermark_csv="$(echo "$watermark_select_result" | tr '\t' ',')"

echo "$watermark_csv"
```

3. To print all watermark data to the terminal window, run the following command:

```
bash print-watermark.sh
```

> **Note**: To print only the most recent watermark data entries, append `| tail -nNUMBER` to the above command. For example, to display the five most recent watermarks, run `bash print-watermark.sh | tail -n5`.

4. (Optional) To write the data to a `.csv` file, run the following command:

```
bash print-watermark.sh > watermarks-$(date -u +"%Y-%m-%dT%H:%M:%SZ").csv
```

View Consumption Usage Data

To view the consumption data for your PKS deployment, follow the steps below:

1. On the command line, connect to your PKS database. You can locate your database credentials in `/var/vcap/jobs/mysql/config/mariadb_ctl_config.yml`.
2. To view the `pksdata` table, run `describe pksdata;`.

   For example:

```
MariaDB [telemetry]> describe pksdata;
+--------------+------------+------+-----+-------------------+-----------------------------+
| Field        | Type       | Null | Key | Default           | Extra                       |
+--------------+------------+------+-----+-------------------+-----------------------------+
| id           | int(11)    | NO   | PRI | NULL              | auto_increment              |
| agent        | char(253)  | YES  |     | NULL              |                             |
| collect_time | timestamp  | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| pod_id       | char(253)  | YES  |     | NULL              |                             |
| pod_name     | char(253)  | YES  |     | NULL              |                             |
| memory_used  | bigint(20) | NO   |     | NULL              |                             |
| cpu_used     | bigint(20) | NO   |     | NULL              |                             |
| pod_cnt      | bigint(20) | NO   |     | NULL              |                             |
+--------------+------------+------+-----+-------------------+-----------------------------+
```

3. Continue to the following sections to run specific queries.

View Pod Counts by Cluster in a Given Time Window

To view the pod counts by cluster for a given time window, run the following query:

```
select agent,count(distinct pod_name) from pksdata where collect_time between
'BEGINNING-TIMESTAMP' and 'ENDING-TIMESTAMP' group by agent;
```

Where `BEGINNING-TIMESTAMP` and `ENDING-TIMESTAMP` represent the beginning and ending times for your search window. Use the `YYYY-MM-DD HH:MM:SS` format for both timestamps.

For example:

```
MariaDB [telemetry]> select agent,count(distinct pod_name) from pksdata where collect_time between '2018-08-14 00:00:00' and '2018-08-14 01:00:00' group by agent;
+------------------------------------------------------+--------------------------+
| agent                                                | count(distinct pod_name) |
+------------------------------------------------------+--------------------------+
| service-instance_1a8617eb-a582-45b5-8749-3b18fb1d661c | 9                        |
| service-instance_1c025915-0ef7-4621-b1eb-aff1c549feb9 | 8                        |
| service-instance_209caeb3-87da-47b7-81db-8b978249f80a | 7                        |
| service-instance_3eb8c734-dce5-4de3-b971-88792d5262d0 | 12                       |
| service-instance_483fa035-c2ee-47c4-b2bd-79329155d6b2 | 3                        |
| service-instance_666ded63-7265-4af8-bfbb-b51f4b1e3f0a | 2                        |
| service-instance_789f7c3f-33f6-4914-88f5-0359c39bf856 | 24                       |
| service-instance_998f7e11-940e-4fdd-9abb-117370dcaaf3 | 12                       |
| service-instance_e3139ecc-5567-4232-9707-1c16e3cdf571 | 10                       |
+------------------------------------------------------+--------------------------+
```

View Running Pod Hours for the Current Day

To view the running pod count by cluster for the current day, run the following query:

```
select agent,timestampdiff(HOUR,min(collect_time), max(collect_time)) + 1 as
"hours today", pod_name from pksdata where collect_time > curdate() -1 group by pod_name;
```

For example:

```
MariaDB [telemetry]> select agent,timestampdiff(HOUR,min(collect_time), max(collect_time)) + 1 as "hours today", pod_name from pksdata where collect_time > curdate() -1 group by pod_na
+------------------------------------------------------+-------------+--------------------------------------------------+
| agent                                                | hours today | pod_name                                         |
+------------------------------------------------------+-------------+--------------------------------------------------+
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-create-bucket-mtlpr                   |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-minio-7d6647dcdd-f62wx                |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |           1 | my-release-spinnaker-clouddriver-c4d56c6b-fd227  |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-spinnaker-deck-5f7c94d6b8-qm49n       |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-spinnaker-echo-75655dd8bb-hkc2t       |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-spinnaker-front50-8f58449b6-pwpdq     |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |           1 | my-release-spinnaker-gate-c66c8996-xxt5h         |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-spinnaker-igor-59d69c6c69-2xcrp       |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |           1 | my-release-spinnaker-orca-7c8ffd56d9-pmr7d       |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-spinnaker-rosco-7759ffcc65-jsw66      |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | my-release-upload-build-image-d4f9f              |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |           1 | nginx-77v62                                      |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |           1 | nginx-kkrm8                                      |
| service-instance_db568933-c59d-409e-a488-a716641e55cd |           1 | nginx-mqr9x                                      |
| service-instance_db568933-c59d-409e-a488-a716641e55cd |           1 | nginx-rbnlm                                      |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |          21 | wordpress-mysql-bcc89f687-vjfvc                  |
+------------------------------------------------------+-------------+--------------------------------------------------+
```

View Pods by Running Time

To view the running pod count by cluster for each hour in the current day, run the following query:

```
select agent,timestampdiff(HOUR,min(collect_time), max(collect_time)) + 1 as
"hours today", pod_name from pksdata where collect_time > curdate() -1 group by pod_name;
```

For example:

```
MariaDB [telemetry]> select agent,hour(collect_time) as hour,count(distinct pod_name) from pksdata group by agent,hour;
+------------------------------------------------------+------+--------------------------+
| agent                                                | hour | count(distinct pod_name) |
+------------------------------------------------------+------+--------------------------+
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    0 |                       12 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    1 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    2 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    3 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    4 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    5 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    6 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    7 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    8 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |    9 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   10 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   11 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   12 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   13 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   14 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   15 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   16 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   17 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   18 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   19 |                        9 |
| service-instance_b43c891c-6a4d-42ef-8bcd-402c0433f89e |   20 |                       11 |
| service-instance_db568933-c59d-409e-a488-a716641e55cd |   20 |                        2 |
+------------------------------------------------------+------+--------------------------+
```

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. SeeMaintaining Workload Uptime for more information.
- **Kubernetes control plane**: The Kubernetes master VM is recreated during the upgrade, so `kubectl` 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 Maintaining Workload Uptime for more information.

Required Actions

None. BOSH brings the process back automatically using `monit`. If the process resumes cleanly and without manual intervention, the worker recovers automatically, and the scheduler resumes scheduling new pods on this worker.

VM Process Failure on the Pivotal Container Service VM

A process, such as the PKS API server, crashes on the pivotal-container-service VM.

Impact

- **PKS control plane**: Depending on the process and what it was doing, the PKS control plane may experience 60-120 seconds of downtime. Until the process resumes, the following can occur:
  - The PKS API or UAA may be inaccessible
  - Use of the PKS CLI is interrupted
  - Metrics or logging may stop
  - Other features may be interrupted

Required Actions

None. BOSH brings the process back automatically using `monit`. If the process resumes cleanly, the PKS control plane recovers automatically and the PKS CLI resumes working.

VM Failure

A PKS VM fails and goes offline due to either a virtualization problem or a host hardware problem.

Impact

- **If the BOSH Resurrector is enabled**, BOSH detects the failure, recreates the VM, and reattaches the same persistent disk and IP address. Downtime depends on which VM goes offline, how quickly the BOSH Resurrector notices, and how long it takes the IaaS to create a replacement VM. The BOSH Resurrector usually notices an offline VM within one to two minutes. For more information about the BOSH Resurrector, see the BOSH documentation.
- **If the BOSH Resurrector is not enabled**, some cloud providers, such as vSphere, have similar resurrection or high availability (HA) features. Depending on the VM, the impact can be similar to a key process on that VM going down as described in the previous sections, but the recovery time is longer while the replacement VM is created. See the sections for process failures on the cluster worker, cluster master, and PKS VM sections for more information.

Required Actions

When the VM comes back online, no further action is required for the developer to continue operations.

AZ Failure

An availability zone (AZ) goes offline entirely or loses connectivity to other AZs (net split).

Impact

The control plane and clusters are inaccessible. The extent of the downtime is unknown.

Required Actions

When the AZ comes back online, the control plane recovers in one of the following ways:

- **If BOSH is in a different AZ**, BOSH recreates the VMs with the last known persistent disks and IPs. If the persistent disks are gone, the disks can be restored from your last backup and reattached. Pivotal recommends manually checking the state of VMs and databases.
- **If BOSH is in the same AZ**, follow the directions for region failure.

Region Failure

An entire region fails, bringing all PKS components offline.

Impact

The entire PKS deployment and all services are unavailable. The extent of the downtime is unknown.

Required Actions

The PKS control plane can be restored using BOSH Backup and Restore (BBR). For more information, see Restoring the PKS Control Plane. Each cluster may need to be restored manually from backups.

Deleting PKS

To delete PKS, perform the following steps:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the trash icon on the PKS tile.
3. Click **Confirm** in the dialog box that appears.
4. By default, deleting the PKS tile will also delete all the clusters created by PKS. To preserve the clusters, click the **Delete all clusters** errand under **Pending Changes** and select **Off**.
5. Click **Apply Changes**.

Using PKS

Page last updated:

This topic describes how to use Pivotal Container Service (PKS).

Note: Because PKS does not currently support the Kubernetes Service Catalog or the GCP Service Broker, binding clusters to Kubernetes services is not supported.

The procedures for using PKS have the following prerequisites:

- You must have an external TCP or HTTPS load balancer configured to forward traffic to the PKS API endpoint. For more information, see the Configure External Load Balancer section of *Installing PKS* for your IaaS.
- You must know the address of your PKS API endpoint and have a UAA-created user account that has been granted PKS cluster access. For more information, see Manage Users in UAA.

  Note: If your PKS installation is integrated with NSX-T, use the DNAT IP address assigned in the Retrieve the PKS Endpoint section of *Installing PKS on vSphere with NSX-T Integration*.

See the following sections:

- Creating Clusters
- Retrieving Cluster Credentials and Configuration
- Viewing Cluster Lists
- Viewing Cluster Details
- Viewing Cluster Plans
- Using Dynamic Persistent Volumes
- Scaling Existing Clusters
- Accessing Dashboard
- Deploying and Accessing Basic Workloads
- Deleting Clusters
- Logging Out of the PKS Environment

Creating Clusters

Page last updated:

This topic describes how to create a Kubernetes cluster with Pivotal Container Service (PKS) using the PKS Command Line Interface (CLI).

Configure Cluster Access

Cluster access configuration differs by the type of PKS deployment.

vSphere with NSX-T

PKS deploys a load balancer automatically when clusters are created. The load balancer is configured automatically when workloads are being deployed on these Kubernetes clusters. For more information, see Load Balancers in PKS Deployments with NSX-T.

vSphere without NSX-T or GCP

When you create a Kubernetes cluster, you must configure external access to the cluster by creating an external TCP or HTTPS load balancer. This load balancer allows you to run PKS CLI commands on the cluster from your local workstation. For more information, see Load Balancers in PKS Deployments without NSX-T.

You can configure any load balancer of your choice. If you use GCP or vSphere without NSX-T, you can create a load balancer using your cloud provider console. For information about configuring a GCP load balancer for PKS clusters, see Configuring a GCP Load Balancer for PKS Clusters.

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

Create the load balancer before you create the cluster. 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 IP address of the cluster's master VM after cluster creation.

To locate the IP addresses and VM IDs of the master VMs, see Identify the Kubernetes Cluster Master VM below.

Create a Kubernetes Cluster

Perform the following steps:

1. Grant cluster access to a new or existing user in UAA. See the Grant Cluster Access to a User section of *Managing Users in PKS with UAA* for more information.

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

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. For more information about the `pks get-credentials` command, see Retrieving Cluster Credentials and Configuration.

7. Run `kubectl cluster-info` to confirm you can access your cluster using the Kubernetes CLI.

See Managing PKS for information about checking cluster health and viewing cluster logs.

Identify Kubernetes Cluster Master VMs

> **Note**: This section applies only to PKS deployments on GCP or on vSphere without NSX-T. Skip this section if your PKS deployment is on vSphere with NSX-T. For more information, see Load Balancers in PKS.

To reconfigure the load balancer or DNS record for an existing cluster, you may need to locate VM ID and IP address information for the cluster's master VMs. Use the information you locate in this procedure when configuring your load balancer backend.

To locate the IP addresses and VM IDs for the master VMs of an existing cluster, do the following:

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

    ```
    pks login -a PKS-API -u USERNAME -k
    ```

    For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

2. To locate the cluster ID and master node IP addresses, run `pks cluster CLUSTER-NAME`. From the output of this command, record the following items:

    - **UUID**: This value is your cluster ID.
    - **Kubernetes Master IP(s)**: This value lists the IP addresses of all master nodes in the cluster.

3. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use the BOSH CLI to log in to the BOSH Director from the Ops Manager VM. For more information, see Advanced Troubleshooting with the BOSH CLI.

4. Identify the name of your cluster deployment. For example:

    ```
    $ bosh -e pks deployments
    ```

    Your cluster deployment name begins with `service-instance` and includes the UUID you located in a previous step.

5. Identify the master VM IDs by listing the VMs in your cluster. For example:

    ```
    $ bosh -e pks -d service-instance-aa1234567bc8de9f0a1c vms
    ```

    Your master VM IDs appear in the **VM CID** column.

6. Use the information you gathered in this procedure to configure your load balancer backend. For example, if you use GCP, use the master VM IDs from Reconfiguring a GCP Load Balancer in *Configuring a GCP Load Balancer for PKS Clusters*.

Retrieving Cluster Credentials and Configuration

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

The `pks get-credentials` command performs the following actions:

- Fetch the cluster's kubeconfig
- Add the cluster's kubeconfig to the existing kubeconfig
- Create a new kubeconfig, if none exists
- Switch the context to the `CLUSTER-NAME` provided

When you run `pks get-credentials CLUSTER-NAME`, PKS sets the context to the cluster you provide as the `CLUSTER-NAME`. PKS binds your username to the cluster and populates the kubeconfig file on your local workstation with cluster credentials and configuration.

The default path for your kubeconfig is `$HOME/.kube/config`.

If you access multiple clusters, you can choose to use a custom kubeconfig file for each cluster. To save cluster credentials to a custom kubeconfig, use the `KUBECONFIG` environment variable when you run `pks get-credentials`. For example:

```
$ KUBECONFIG=/path/to/my-cluster.config pks get-credentials my-cluster
```

Retrieve Cluster Credentials

Perform the following steps to populate your local kubeconfig with cluster credentials and configuration:

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

2. Run the following command:

   ```
   pks get-credentials CLUSTER-NAME
   ```

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

   ```
   $ pks get-credentials my-cluster
   ```

Run kubectl Commands

After PKS populates your kubeconfig, you can use the Kubernetes Command Line Interface (kubectl) to run commands against your Kubernetes clusters.

See Installing the Kubernetes CLI for information about installing kubectl.

For information about using kubectl, refer to the Kubernetes documentation.

Viewing Cluster Lists

Follow the steps below to view the list of deployed Kubernetes cluster with the PKS CLI.

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

2. Run the following command to view the list of deployed clusters, including cluster names and status:

   ```
   $ pks clusters
   ```

Viewing Cluster Details

Follow the steps below to view the details of an individual cluster using the PKS CLI.

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

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
   ```

Viewing 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:

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

2. Run the following command to view information about the available plans for deploying a cluster:

   ```
   $ pks plans
   ```

   The response lists details about the available plans, including plan names and descriptions:

   ```
   Name     ID  Description
   default      Default plan for K8s cluster
   ```

Using Dynamic Persistent Volumes

When using PKS, you can choose to pre-provision persistent storage or create on-demand persistent storage volumes. Refer to the Kubernetes documentation for more information about storage management.

Perform the steps in this section to define a PersistentVolumeClaim that you can apply to newly-created pods.

1. Download the StorageClass spec for your cloud provider.
    - **GCP**:

      ```
      $ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-gcp.yml
      ```

    - **vSphere**:

      ```
      $ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-vsphere.yml
      ```

2. Apply the spec by running `kubectl create -f STORAGE-CLASS-SPEC.yml`. Replace `STORAGE-CLASS-SPEC` with the name of the file you downloaded in the previous step. For example:

   ```
   $ kubectl create -f storage-class-gcp.yml
   ```

3. Run the following command to download the example PersistentVolumeClaim:

   ```
   $ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/persistent-volume-claim.yml
   ```

4. Run the following command to apply the PersistentVolumeClaim:

   ```
   $ kubectl create -f persistent-volume-claim.yml
   ```

    - To confirm you applied the PersistentVolumeClaim, run the following command:

      ```
      $ kubectl get pvc -o wide
      ```

5. To use the dynamic persistent volume, create a pod that uses the PersistentVolumeClaim. See the pv-guestbook.yml configuration file as an example.

Scaling Existing Clusters

Follow the steps below to scale up an existing cluster using the PKS CLI.

Note: You cannot scale the number of worker nodes down. You can only scale the number of worker nodes up.

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

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
     ```

Accessing Dashboard

Page last updated:

This topic describes how to access Dashboard, a web-based Kubernetes user interface, for your Pivotal Container Service (PKS) deployment.

Overview

You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot containerized apps, manage the cluster and its resources, initiate rolling updates, and restart pods. Dashboard also provides information about the state of Kubernetes resources in the cluster.

Prerequisites

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

Access Dashboard

To access Dashboard, navigate to `http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/` in a browser.

Use Dashboard

For information on how to use Dashboard, see the Web UI (Dashboard) topic of the Kubernetes documentation.

Deploying and Accessing 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

If you use GCP or vSphere with NSX-T, follow the steps below to deploy and access basic workloads using a load balancer configured by your cloud provider.

Note: This approach creates a dedicated load balancer for each workload. This may be an inefficient use of resources in clusters with many apps.

1. Expose the workload using a Service with `type: LoadBalancer`. See the Kubernetes documentation for more information about the `LoadBalancer` Service type.
2. Download the spec for a basic NGINX app from the cloudfoundry-incubator/kubo-ci GitHub repository.
3. Run `kubectl create -f nginx.yml` to deploy the basic NGINX app. This command creates three pods (replicas) that span three worker nodes.
4. Wait until your cloud provider creates a dedicated load balancer and connects it to the worker nodes on a specific port.
5. Run `kubectl get svc nginx` and retrieve the load balancer IP address and port number.
6. On the command line of a server with network connectivity and visibility to the IP address of the worker node, run `curl http://EXTERNAL-IP:PORT` to access the app. Replace `EXTERNAL-IP` with the IP address of the load balancer and `PORT` with the port number.

Access Workloads Using an External Load Balancer

All deployments can use an external load balancer. To use an external load balancer, follow the steps below to deploy and access basic workloads.

1. Expose every workload and app using a Service with `type: NodePort`. See the Kubernetes documentation for more information about the `NodePort` Service type.
2. Map each node port exposed in the worker nodes that you need to an external port in your external load balancer. The process to map these ports depends on your load balancer. See your external load balancer documentation for more information.
3. For each app, run `curl http://LOAD-BALANCER-IP:EXTERNAL-PORT`. Replace `LOAD-BALANCER-IP` with the IP address of your external load balancer and `EXTERNAL-PORT` with the external port number.

Access Workloads without a Load Balancer

If you use vSphere without NSX-T integration, you do not have a load balancer configured by your cloud provider. You can choose to configure your own external load balancer or follow the procedures in this section to access your workloads without a load balancer.

If you do not use an external load balancer, you can configure the NGINX service to expose a static port on each worker node. From outside the cluster, you can reach the service at `http://NODE-IP:NODE-PORT`.

To expose a static port on your workload, perform the following steps:

1. Download the spec for a basic NGINX app from the cloudfoundry-incubator/kubo-ci GitHub repository.

2. Run `kubectl create -f nginx.yml` to deploy the basic NGINX app. This command creates three pods (replicas) that span three worker nodes.

3. Expose the workload using a Service with `type: NodePort`. See the Kubernetes documentation for more information about the `NodePort` Service type.

4. Retrieve the IP address for a worker node with a running NGINX pod.

   > **Note**: If you deployed more than four worker nodes, some worker nodes may not contain a running NGINX pod. Select a worker node that contains a running NGINX pod.

   You can retrieve the IP address for a worker node with a running NGINX pod in one of the following ways:

   - On the command line, run `kubectl get nodes -L spec.ip`.
   - 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.

Deleting Clusters

Follow the steps below to delete a cluster using the PKS CLI. In PKS v1.1, running the `pks delete-cluster` command automatically deletes all NSX objects.

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

   ```
   pks login -a PKS-API -u USERNAME -k
   ```

   For more information about the `pks login` command, see the Log in to the PKS CLI section of *Installing the PKS CLI*.

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
   ```

Logging Out of the PKS Environment

On the command line, run `pks logout` to log out of your PKS environment.

After logging out, you must run `pks login` before you can run any other `pks` commands.

Using Helm with PKS

Page last updated:

This documentation 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**. For more information, see Charts in the Helm documentation.

Examples of charts:

- Concourse for CI/CD pipelines
- Datadog for monitoring
- MySQL for storage

For more charts, see the Helm Charts repository on GitHub.

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

- Configuring Tiller

Configuring Tiller

Tiller runs inside the Kubernetes cluster and requires access to the Kubernetes API. If you use role-based access control (RBAC) in PKS, perform the steps in this section to grant Tiller permission to access the API.

1. Create a service account for Tiller and bind it to the `cluster-admin` role by adding the following section to `rbac-config.yaml`:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
  - kind: ServiceAccount
    name: tiller
    namespace: kube-system
```

2. Apply the service account and role by running the following command:

```
$ kubectl create -f rbac-config.yaml
```

3. Download and install the Helm CLI.

4. Deploy Helm using the service account by running the following command:

```
$ helm init --service-account tiller
```

5. Run `helm ls` to verify that the permissions are configured.

To apply more granular permissions to the Tiller service account, see the Helm RBAC documentation.

Backing Up and Restoring PKS

Page last updated:

This section describes how to back up and restore the Pivotal Container Service (PKS) control plane. PKS uses the Cloud Foundry BOSH Backup and Restore framework to back up and restore the PKS control plane.

The PKS control plane includes the following components:

- UAA MySQL database
- PKS API MySQL database

BOSH Backup and Restore (BBR) backs up the PKS control plane components. BBR does not back up cluster data or deployed applications.

BBR orchestrates triggering the backup or restore process on the PKS BOSH deployment, and transfers the backup artifacts to and from the PKS BOSH deployment.

For more information about installing and using BBR, see the following topics:

- Installing BOSH Backup and Restore
- Backing Up the PKS Control Plane
- Restoring the PKS Control Plane.

See BBR Logging for information about troubleshooting BBR.

Installing BOSH Backup and Restore

Page last updated:

This topic describes how to install BOSH Backup and Restore (BBR).

To install BBR, you copy the `bbr` executable to a jumpbox.

Once installed on your jumpbox, you can run `bbr` commands to back up and restore your PKS deployment.

For more information, see Backing up the PKS Control Plane and Restoring the PKS Control Plane.

Prerequisite

You must have a jumpbox before you can install BBR to the jumpbox. A jumpbox is a separate, hardened server on your network that provides a controlled means of access to the VMs other computers on your network.

See the jumpbox-deployment GitHub repository for an example jumpbox deployment.

Step 1: Configure Your Jumpbox

Configure your jumpbox to meet the following requirements:

- Your jumpbox must be able to communicate with the network that contains your PKS deployment. You can use the Ops Manager VM as your jumpbox.
- Your jumpbox must have sufficient space for the backup.
- Your jumpbox must be in the same network as the deployed VMs because BBR connects to the VMs at their private IP addresses. BBR does not support SSH gateways.
- BBR copies the backed-up data from the VMs to the jumpbox, so you should have minimal network latency between the VMs and the jumpbox to reduce transfer times.

Note: BBR uses SSH to orchestrate the backup of your PKS instances using port 22 by default.

Step 2: Transfer BBR to Your Jumpbox

Perform the following steps to transfer the `bbr` binary to your jumpbox:

1. Download the latest BOSH Backup and Restore release from Pivotal Network.
2. To add executable permissions to the `bbr` binary file, run `chmod a+x bbr`:

   ```
   $ chmod a+x bbr
   ```

3. To securely copy the `bbr` binary file to your jumpbox, run the following command:

   ```
   scp LOCAL-PATH-TO-BBR/bbr JUMPBOX-USER/JUMPBOX-ADDRESS
   ```

   If your jumpbox has access to the internet, you can instead SSH into your jumpbox and use `wget`:

   ```
   $ ssh JUMPBOX-USER/JUMPBOX-ADDRESS -i YOUR-CERTIFICATE.pem
   $ wget BBR-RELEASE-URL
   $ chmod a+x bbr
   ```

Backing up the PKS Control Plane

Page last updated:

This topic describes how to use BOSH Backup and Restore (BBR) to back up the PKS control plane.

To perform a restore, see Restoring the PKS Control Plane.

Prerequisites

If you want to use the result of the backup to restore to a destination environment, verify that the current environment and the destination environment are compatible. For more information, see the Compatibility of Restore section of *Restoring the PKS Control Plane*.

Before you begin backing up the PKS control plane, perform the following steps:

1. Download the root CA certificate for your PKS deployment as follows:
    a. On the Ops Manager Installation Dashboard, in the top right corner, click your username.
    b. Navigate to **Settings** > **Advanced**.
    c. Click **Download Root CA Cert**.
2. Locate and record your PKS BOSH deployment name as follows:
    a. On the Ops Manager Installation Dashboard, click the Director tile.
    b. In the Director tile, click the **Credentials** tab.
    c. Navigate to **Bosh Commandline Credentials** and click **Link to Credential**.
    d. Copy the credential value.
    e. On the command line, run the following command to retrieve your PKS BOSH deployment name. Replace `BOSH-CLI-CREDENTIALS` with the credential value you copied in the previous step:

```
BOSH-CLI-CREDENTIALS deployments | grep pivotal-container-service
```

> **Note**: Your PKS BOSH deployment name begins with “pivotal-container-service” and includes a unique identifier.

Connect to Your Jumpbox

You can establish a connection to your jumpbox in one of the following ways.

- Connect with SSH
- Connect with BOSH_ALL_PROXY

For general information about the jumpbox, see Installing BOSH Backup and Restore.

Connect with SSH

SSH into your jumpbox. If you connect to your jumpbox with SSH, you must run the BBR commands in the following sections from within your jumpbox.

Connect with BOSH_ALL_PROXY

Set and use `BOSH_ALL_PROXY`. Using `BOSH_ALL_PROXY` opens an SSH tunnel with SOCKS5 to the jumpbox. This tunnel allows you to forward requests to the BOSH Director through the jumpbox from your local machine.

Use one of the following methods to create the tunnel:

- **Tunnel created by BOSH CLI**: To provide the BOSH CLI with the SSH credentials it needs to create the tunnel, run the following command:

```
export BOSH_ALL_PROXY=ssh+socks5://jumpbox@jumpbox-ip:12345?private_key=jumpbox.key
```

- **Tunnel established separately**:

1. To establish the tunnel and make it available on a local port, run the following command:

```
ssh -4 -D 12345 -fNC jumpbox@jumpbox-ip -i jumpbox.key
```

2. To provide the BOSH CLI with access to the tunnel through use of the `BOSH_ALL_PROXY` environment variable, run the following command:

```
export BOSH_ALL_PROXY=socks5://localhost:12345
```

Note: Using `BOSH_ALL_PROXY` can result in longer backup and restore times due to network performance degradation. Because all operations must pass through the proxy, moving backup artifacts can be significantly slower.

Back up the PKS Control Plane

1. Run the BBR pre-backup check to confirm that your BOSH Director is reachable and has a deployment that can be backed up:

```
BOSH_CLIENT_SECRET=SECRET=BOSH-CLIENT-SECRET \
bbr deployment \
--target BOSH-TARGET \
--username BOSH-CLIENT \
--deployment DEPLOYMENT-NAME \
--ca-cert PATH-TO-BOSH-SERVER-CERT \
pre-backup-check
```

Replace the placeholder text using the information in the following table.

Placeholder Text	Instructions
`BOSH-CLIENT-SECRET`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT_SECRET`.
`BOSH-TARGET`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_ENVIRONMENT`. You must be able to reach the target address from the workstation where you run `bbr` commands.
`BOSH-CLIENT`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT`.
`DEPLOYMENT-NAME`	Use the PKS BOSH deployment name that you located in the Prerequisites section.
`PATH-TO-BOSH-CA-CERT`	Use the path to the root CA certificate that you downloaded in the Prerequisites section.

For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \
bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.cert \
pre-backup-check
```

2. If the pre-backup check command fails, perform the following actions:

   - Run the command again, adding the `--debug` flag to enable debug logs. For more information, see BBR Logging.
   - Make any correction suggested in the output and run the pre-backup check again. For example, the deployment that you selected might not have the correct backup scripts, or the connection to the BOSH Director failed.

3. If the pre-backup check succeeds, run the BBR backup command from your jumpbox to back up the PKS control plane:

```
BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET \
nohup bbr deployment \
--target BOSH-TARGET \
--username BOSH-CLIENT \
--deployment DEPLOYMENT-NAME \
--ca-cert PATH-TO-BOSH-SERVER-CERT \
backup
```

Replace the placeholder text using the information in the following table. These are the same values as shown in the previous table.

Placeholder Text	Instructions
`BOSH-CLIENT-SECRET`	In your BOSH Director tile, navigate to **Credentials** > **Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT_SECRET`.
`BOSH-TARGET`	In your BOSH Director tile, navigate to **Credentials** > **Bosh Commandline Credentials**. Record the value for `BOSH_ENVIRONMENT`. You must be able to reach the target address from the workstation where you run `bbr` commands.
`BOSH-CLIENT`	In your BOSH Director tile, navigate to **Credentials** > **Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT`.
`DEPLOYMENT-NAME`	Use the PKS BOSH deployment name that you located in the Prerequisites section.
`PATH-TO-BOSH-CA-CERT`	Use the path to the root CA certificate that you downloaded in the Prerequisites section.

Note: If you want to include the manifest in the backup artifact, add the `--with-manifest` flag. However, be aware that the backup artifact then includes credentials that you must keep secret.

For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \
nohup bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.cert \
backup
```

Note: The BBR backup command can take a long time to complete. You can run it independently of the SSH session so that the process can continue running even if your connection to the jumpbox fails. The command above uses `nohup`, but you can run the command in a `screen` or `tmux` session instead.

4. If the command completes successfully, follow the steps in Manage Your Backup Artifact below.
5. If the backup command fails, perform the following actions:
    - Run the command again, adding the `--debug` flag to enable debug logs. For more information, see BBR Logging.
    - Follow the steps in Recover from a Failing Command.

Recover from a Failing Command

If the backup fails, follow these steps:

1. Ensure that you set all the parameters in the backup command.
2. Ensure the BOSH Director credentials are valid.
3. Ensure the deployment that you specify in the BBR command exists.
4. Ensure that the jumpbox can reach the BOSH Director.
5. Consult BBR Logging.
6. If you see the error message `Directory /var/vcap/store/bbr-backup already exists on instance`, run the appropriate cleanup command. See Clean up After a Failed Backup below.
7. If the backup artifact is corrupted, discard the failing artifacts and run the backup again.

Cancel a Backup

Backups can take a long time. If you need to cancel a backup, for example if you realize that the backup is going to fail or that your developers need to push an app in a hurry, follow these steps:

1. Terminate the BBR process by pressing Ctrl-C and typing `yes` to confirm.

2. Because stopping a backup can leave the system in an unusable state and prevent additional backups, follow the procedures in Clean up After a Failed Backup below.

Clean up After a Failed Backup

If your backup process fails, it might leave the BBR backup folder on the instance, causing any subsequent attempts to backup to fail. In addition, BBR might not have run the post-backup scripts, leaving the instance in a locked state.

If the PKS control plane backup failed, run the following command to use the BBR cleanup script to clean up:

```
BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET \
bbr deployment \
--target BOSH-TARGET \
--username BOSH-CLIENT \
--deployment DEPLOYMENT-NAME \
--ca-cert PATH-TO-BOSH-CA-CERT \
backup-cleanup
```

Replace the placeholder text using the information in the following table. These are the same values as shown in the previous table.

Placeholder Text	Instructions
`BOSH-CLIENT-SECRET`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT_SECRET`.
`BOSH-TARGET`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_ENVIRONMENT`. You must be able to reach the target address from the workstation where you run `bbr` commands.
`BOSH-CLIENT`	In your BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT`.
`DEPLOYMENT-NAME`	Use the PKS BOSH deployment name that you located in the Prerequisites section.
`PATH-TO-BOSH-CA-CERT`	Use the path to the root CA certificate that you downloaded in the Prerequisites section.

Note: If you want to include the manifest in the backup artifact, add the `--with-manifest` flag. However, be aware that the backup artifact then includes credentials that you must keep secret.

For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \
bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.crt \
backup-cleanup
```

Manage Your Backup Artifact

Keep your backup artifact safe by following these steps:

1. Move the backup artifact off the jumpbox to your storage space. BBR stores each backup in a subdirectory named `DEPLOYMENT-TIMESTAMP` within the current working directory. The backup created by BBR consists of a folder with the backup artifacts and metadata files.

2. Compress and encrypt the backup artifacts when storing them.

3. Make redundant copies of your backup and store them in multiple locations. This minimizes the risk of losing your backups in the event of a disaster.

4. Each time you redeploy PKS, test your backup artifact by following the procedures in Restoring the PKS Control Plane.

Restoring the PKS Control Plane

Page last updated:

This topic describes how to use BOSH Backup and Restore (BBR) to restore the PKS control plane.

To back up the PKS control plane with BBR, see Backing up the PKS Control Plane.

Compatibility of Restore

This section describes the restrictions for a backup artifact to be restorable to another environment. This section is for guidance only, and Pivotal highly recommends that operators validate their backups by using the backup artifacts in a restore.

The restrictions for a backup artifact to be restorable are the following:

- **Topology**: BBR requires the BOSH topology of a deployment to be the same in the restore environment as it was in the backup environment.
- **Naming of instance groups and jobs**: For any deployment that implements the backup and restore scripts, the instance groups and jobs must have the same names.
- **Number of instance groups and jobs**: For instance groups and jobs that have backup and restore scripts, the same number of instances must exist..
- **Limited validation**: BBR puts the backed up data into the corresponding instance groups and jobs in the restored environment, but cannot validate the restore beyond that. For example, if the MySQL encryption key is different in the restore environment, the BBR restore might succeed although the restored MySQL database is unusable.

Note: A change in VM size or underlying hardware should not affect the ability for BBR restore data, as long as adequate storage space to restore the data exists.

Step 1: Recreate VMs

Before restoring the PKS control plane, you must create the VMs that constitute the deployment.

In a disaster recovery scenario, you can re-create the control plane with your PKS deployment manifest. If you used the `--with-manifest` flag when you ran the BBR backup command, your backup artifact includes a copy of your manifest.

Step 2: Transfer Artifacts to Jumpbox

Transfer your BBR backup artifact from your safe storage location to the jumpbox.

For example, you could run the following command to SCP the backup artifact to your jumpbox:

```
scp LOCAL-PATH-TO-BACKUP-ARTIFACT JUMPBOX-USER/JUMPBOX-ADDRESS
```

If this artifact is encrypted, you must decrypt it.

Step 3: Restore

Note: The BBR restore command can take a long time to complete. You can run it independently of the SSH session so that the process can continue running even if your connection to the jumpbox fails. The command above uses `nohup`, but you run the command in a `screen` or `tmux` session instead.

Perform the following steps to restore the PKS control plane. You can use the optional `--debug` flag to enable debug logs. See the BBR Logging topic for more information.

1. Ensure the PKS deployment backup artifact is in the folder from which you run BBR.
2. Download the root CA certificate for your PKS deployment as follows:

a. On the Ops Manager Installation Dashboard, in the top right corner, click your username.
b. Navigate to **Settings** > **Advanced**.
c. Click **Download Root CA Cert**.

3. Locate and record your PKS BOSH deployment name as follows:

a. On the Ops Manager Installation Dashboard, click the Director tile.
b. In the Director tile, click the **Credentials** tab.
c. Navigate to **Bosh Commandline Credentials** and click **Link to Credential**.
d. Copy the credential value.
e. On the command line, run the following command to retrieve your PKS BOSH deployment name. Replace `BOSH-CLI-CREDENTIALS` with the credential value you copied in the previous step:

```
BOSH-CLI-CREDENTIALS deployments | grep pivotal-container-service
```

Note: Your PKS BOSH deployment name begins with “pivotal-container-service” and includes a unique identifier.

4. Run the BBR restore command to restore the PKS control plane:

```
BOSH_CLIENT_SECRET=BOSH-CLIENT-SECRET \
nohup bbr deployment \
--target BOSH-TARGET \
--username BOSH-CLIENT \
--deployment DEPLOYMENT-NAME \
--ca-cert PATH-TO-BOSH-SERVER-CERT \
restore \
--artifact-path PATH-TO-DEPLOYMENT-BACKUP
```

Replace the placeholder values as follows:

Credential	Location
`BOSH-CLIENT-SECRET`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT_SECRET`.
`BOSH-TARGET`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_ENVIRONMENT`. You must be able to reach the target address from the workstation where you run `bbr` commands.
`BOSH-CLIENT`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT`.
`DEPLOYMENT-NAME`	Use the PKS BOSH deployment name that you recorded in a previous step.
`PATH-TO-BOSH-CA-CERT`	Use the path to the root CA certificate that you downloaded in a previous step.
`PATH-TO-DEPLOYMENT-BACKUP`	Use the path to the PKS control plane backup that you want to restore.

For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \
nohup bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.crt \
restore \
--artifact-path /home/cf-abcd1234abcd1234abcd-abcd1234abcd1234
```

If the command fails, follow the steps in Recover from a Failing Command.

Recover from a Failing Command

1. Ensure that you set all the parameters in the command.
2. Ensure that the BOSH Director credentials are valid.
3. Ensure that the specified BOSH deployment exists.
4. Ensure that the jumpbox can reach the BOSH Director.

5. Ensure the source BOSH deployment is compatible with the target BOSH deployment.

6. If you see the error message `Directory /var/vcap/store/bbr-backup already exists on instance`, run the relevant commands from the Clean up After Failed Restore section of this topic.

7. See the BBR Logging topic.

Cancel a Restore

If you must cancel a restore, perform the following steps:

1. Terminate the BBR process by pressing Ctrl-C and typing `yes` to confirm.

2. Perform the procedures in the Clean up After Failed Restore section to enable future restores. Stopping a restore can leave the system in an unusable state and prevent future restores.

Clean up After Failed Restore

If your restore process fails, then the process may leave the BBR restore folder on the instance. As a result, any subsequent restore attempts may also fail. In addition, BBR may not have run the post-restore scripts, which can leave the instance in a locked state.

To resolve these issues, run the BBR cleanup script with the following command:

```
BOSH-CLIENT-SECRET=BOSH-CLIENT-SECRET \
bbr deployment \
--target BOSH-TARGET \
--username BOSH-CLIENT \
--deployment DEPLOYMENT-NAME \
--ca-cert PATH-TO-BOSH-CA-CERT \
restore-cleanup
```

Replace the placeholder values as follows:

Credential	Location
`BOSH-CLIENT-SECRET`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT_SECRET`.
`BOSH-TARGET`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_ENVIRONMENT`. You must be able to reach the target address from the workstation where you run `bbr` commands.
`BOSH-CLIENT`	In the BOSH Director tile, navigate to **Credentials > Bosh Commandline Credentials**. Record the value for `BOSH_CLIENT`.
`DEPLOYMENT-NAME`	Use the PKS BOSH deployment name that you recorded in a previous step.
`PATH-TO-BOSH-CA-CERT`	Use the path to the root CA certificate that you downloaded in a previous step.

For example:

```
$ BOSH_CLIENT_SECRET=p455w0rd \
bbr deployment \
--target bosh.example.com \
--username admin \
--deployment cf-acceptance-0 \
--ca-cert bosh.ca.crt \
restore-cleanup
```

BBR Logging

This topic provides information about BBR logging. Use this information when troubleshooting a failed backup or restore using BBR.

Understand Logging

By default, BBR displays the following:

- The backup and restore scripts that it finds
- When it starts or finishes a stage, such as `pre-backup scripts` or `backup scripts`
- When the process is complete
- When any error occurs

BBR writes any errors associated with stack traces to a file in of the form `bbr-TIMESTAMP.err.log` in the current directory.

If more logging is needed, use the optional `--debug` flag to print the following information:

- Logs about the API requests made to the BOSH server
- All commands executed on remote instances
- All commands executed on local environment
- Standard in and standard out streams for the backup and restore scripts when they are executed

PKS Security

Page last updated:

This section includes security topics for Pivotal Container Service (PKS).

See the following topic:

- PKS Security Disclosure and Release Process

PKS Security Disclosure and Release Process

Page last updated:

This topic describes the processes for disclosing security issues and releasing related fixes for Pivotal Container Service (PKS), Kubernetes, Cloud Foundry Container Runtime (CFCR), VMware NSX, and VMware Harbor.

Security Issues in PKS

Pivotal and VMware provide security coverage for PKS. Please report any vulnerabilities directly to Pivotal Application Security Team or the VMware Security Response Center.

Security fixes are provided in accordance with the PCF Security Release Policy and the Pivotal Support Lifecycle Policy.

Where applicable, security issues may be coordinated with the responsible disclosure process for the open source security teams in Kubernetes and Cloud Foundry projects.

Security Issues in Kubernetes

Pivotal and VMware follow the Kubernetes responsible disclosure process to work within the Kubernetes project to report and address suspected security issues with Kubernetes.

This process is discussed in Kubernetes Security and Disclosure Information.

When the Kubernetes project releases security fixes, PKS releases fixes according to the PCF Security Release Policy and the Pivotal Support Lifecycle Policy.

Security Issues in CFCR

Pivotal and VMware follow the Cloud Foundry responsible disclosure process to work within the Cloud Foundry Foundation to report and address suspected security issues with CFCR.

This process is discussed in Cloud Foundry Security.

When the Cloud Foundry Foundation releases security fixes, PKS releases fixes according to the PCF Security Release Policy and the Pivotal Support Lifecycle Policy.

Security Issues in VMware NSX

Security issues in VMware NSX are coordinated with the VMware Security Response Center.

Security Issues in VMware Harbor

Security issues in VMware Harbor are coordinated with the VMware Security Response Center.

Diagnosing and Troubleshooting PKS

This topic is intended to provide assistance when diagnosing and troubleshooting issues installing or using Pivotal Container Service (PKS).

See the following sections:

Diagnostic Tools

Verify PKS CLI Version

The Pivotal Container Service (PKS) CLI interacts with the your PKS deployment through the PKS API endpoint. You create, manage, and delete Kubernetes clusters on your PKS deployment by entering commands in the PKS CLI. The PKS CLI is under active development and commands may change between versions.

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

```
$ pks --version
PKS CLI version: 1.0.0-build.3
```

View Log Files

Log files contain error messages and other information you can use to diagnose issues with your PKS deployment. Follow the steps below to access PKS log files.

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use BOSH CLI v2+ to log in to the BOSH Director from the Ops Manager VM. For more information, see Advanced Troubleshooting with the BOSH CLI.

2. After logging in to the BOSH Director, identify the name of your PKS deployment. For example:

   ```
   $ bosh -e pks deployments
   ```

   Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. On a command line, run `bosh -e pks -d YOUR-DEPLOYMENT-NAME vms` to list the virtual machines (VMs) in your PKS deployment. For example:

   ```
   $ bosh -e pks -d pivotal-container-service-aa1234567bc8de9f0a1c vms
   ```

4. Run `bosh -e pks -d YOUR-DEPLOYMENT-NAME ssh VM-NAME/GUID` to ssh into a PKS VM.

   - To access logs on the master VM, replace `VM-NAME/GUID` with the name of the PKS master VM, and `GUID` with the GUID of the master VM.
   - To access logs on a worker VM, replace `VM-NAME/GUID` with the name of a PKS worker VM, and `GUID` with the GUID of the same worker VM.

5. Run `sudo su` to act as super user on the PKS VM.

6. Navigate to `/var/vcap/sys/log` on the PKS VM:

   ```
   $ cd /var/vcap/sys/log
   ```

7. Examine the following file:

   - On the PKS master VM, examine the `kube-apiserver` log file.
   - On a PKS worker VM, examine the `kubelet` log file.

Troubleshooting

Page last updated:

PKS API is Slow or Times Out

Symptom

When you run PKS CLI commands, the PKS API times out or is slow to respond.

Explanation

The PKS API control plane VM requires more resources.

Solution

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Select the **Pivotal Container Service** tile.
3. Select the **Resource Config** page.
4. For the **Pivotal Container Service** job, select a **VM Type** with greater CPU and memory resources.
5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.

Cluster Creation Fails

Symptom

When creating a cluster, you run `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 Managing 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. To retrieve more information about the task, run the following command:

   ```
   bosh -e MY-ENVIRONMENT task TASK-NUMBER
   ```

   Where:

   - `MY-ENVIRONMENT` is the name of your BOSH environment.
   - `TASK-NUMBER` is the number of the task that attempted to create the cluster.

   For example:

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

See the BOSH documentation for more information about troubleshooting failed BOSH tasks.

Cannot Re-Create a Cluster that Failed to Deploy

Symptom

After cluster creation fails, you cannot re-run `pks create-cluster` to attempt creating the cluster again.

Explanation

PKS does not automatically clean up the failed BOSH deployment. Running `pks create-cluster` using the same cluster name creates a name clash error in BOSH.

Solution

Perform the following steps to clean up the BOSH deployment:

1. Run the following command:

   ```
   bosh -e MY-ENVIRONMENT delete-deployment -d DEPLOYMENT-NAME
   ```

   Where:

   - `MY-ENVIRONMENT` is the name of your BOSH environment.
   - `DEPLOYMENT-NAME` is the name of your BOSH deployment.

   > **Note**: If necessary, you can append the `--force` flag to delete the deployment.

2. Run the following command:

   ```
   pks delete-cluster CLUSTER-NAME
   ```

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

If you are using PKS v1.1.4 or earlier, contact VMware Customer Support to obtain a cleanup script.

Cannot Access Add-On Features or Functions

Symptom

You cannot access a feature or function provided by a Kubernetes add-on.

Examples include the following:

- You cannot access the Kubernetes Web UI (Dashboard) in a browser or using the kubectl command-line tool.
- Heapster does not start.
- Pods cannot resolve DNS names, and error messages report the service `kube-dns` is invalid. If `kube-dns` is not deployed, the cluster typically fails to start.

Explanation

The Kubernetes features and functions listed above are provided by the following PKS add-ons:

- **Kubernetes Dashboard** `kubernetes-dashboard`
- **Heapster**: `heapster`
- **DNS Resolution**: `kube-dns`

To enable these add-ons, Ops Manager must run scripts after deploying PKS. You must configure Ops Manager to automatically run these post-deploy scripts.

Solution

Perform the following steps to configure Ops Manager to run post-deploy scripts to deploy the missing add-ons to your cluster.

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Click the Ops Manager v2.1 tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.

   Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.
5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pks delete-cluster` on the command line to delete the cluster. For more information, see Deleting Clusters.
9. On the command line, enter `pks create-cluster` to recreate the cluster. For more information, see Creating Clusters.

Resurrecting VMs Causes Incorrect Permissions in vSphere HA

Symptoms

Output resulting from the `bosh vms` command alternates between showing that the VMs are `failing` and showing that the VMs are `running`. The operator must run the `bosh vms` command multiple times to see this cycle.

Explanation

The VMs' permissions are altered during the restarting of the VM so operators have to reset permissions every time the VM reboots or is redeployed.

VMs cannot be successfully resurrected if the resurrection state of your VM is set to `off` or if the the vSphere HA restarts the VM before BOSH is aware that the VM is down. For more information on VM resurrection, see Resurrection in the Cloud Foundry BOSH documentation.

Solution

Run the following command on all of your master and worker VMs:

```
bosh -environment BOSH-DIRECTOR-NAME -deployment DEPLOYMENT-NAME ssh INSTANCE-GROUP-NAME -c "sudo /var/vcap/jobs/kube-controller-manager/bin/pre-start; sudo /var/vcap/jobs/kube-apiserver/bin/post-start"
```

Where:

- `BOSH-DIRECTOR-NAME` is your BOSH Director name.
- `DEPLOYMENT-NAME` is the name of your BOSH deployment.
- `INSTANCE-GROUP-NAME` is the name of the BOSH instance group you are referencing.

The above command, when applied to each VM, gives your VMs the correct permissions.

Worker Node Hangs Indefinitely

Symptoms

After making your selection in the **Upgrade all clusters errand** section, the worker node might hang indefinitely. For more information on monitoring the **Upgrade all clusters errand** using the BOSH CLI, see Upgrading PKS.

Explanation

During the PKS tile upgrade process, worker nodes are cordoned and drained. This drain is dependent on Kubernetes being able to unschedule all pods. If Kubernetes is unable to unschedule a pod, then the drain hangs indefinitely. One reason why Kubernetes may be unable to unschedule the node is if the `PodDisruptionBudget` object has been configured in a way that allows 0 disruptions and only a single instance of the pod has been scheduled.

In your spec file, the `.spec.replicas` configuration sets the total amount of replicas that are available in your application. `PodDisruptionBudget` objects can specifies the amount of replicas, proportional to that total, that must be available in your application, regardless of downtime. Operators can configure `PodDisruptionBudget` objects for each application using their spec file.

Some apps deployed using Helm-Charts may have a default `PodDisruptionBudget` set. For more information on configuring `PodDisruptionBudget` objects using a spec file, see Specifying a PodDisruptionBudget in the Kubernetes documentation.

Solution

Configure `.spec.replicas` to be greater than the `PodDisruptionBudget` object.

When the number of replicas configured in `.spec.replicas` is greater than the number of replicas set in the `PodDisruptionBudget` object, disruptions can occur.

For more information, see How Disruption Budgets Work in the Kubernetes documentation. For more information on workload capacity and uptime requirements in PKS, see Prepare to Upgrade in *Upgrading PKS*.

Cannot Authenticate to an OpenID Connect-Enabled Cluster

Symptom

When you authenticate to an OpenID Connect-enabled cluster using an existing kubeconfig file, you see an authentication or authorization error.

Explanation

ID or refresh token contained in the kubeconfig file for the cluster may have expired.

Solution

1. Upgrade the PKS CLI to v1.2.0 or later. To download the PKS CLI, navigate to Pivotal Network. For more information, see Installing the PKS CLI.
2. Obtain a kubeconfig file that contains the new tokens by running the following command:

   ```
   pks get-credentials CLUSTER-NAME
   ```

   Where `CLUSTER-NAME` is the name of your cluster.
3. Connect to the cluster using kubectl.

If you continue to see an authentication or authorization error, verify that you have sufficient access permissions for the cluster.

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 BOSH Director tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.

   Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.

5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pks delete-cluster` on the command line to delete the cluster. For more information, see Deleting Clusters.
9. On the command line, enter `pks create-cluster` to recreate the cluster. For more information, see Creating Clusters.

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.

Error: Duplicate Variable Name

Symptom

In PKS Broker log files, you see an error message that includes `Duplicate variable name '/dns_api_tls_ca'`.

Explanation

This error may occur if you use Ops Manager v2.1.7 and later with PKS v1.1.0.

Solution

PKS v1.1.0 does not support Ops Manager v2.1.7 and later. You must use Ops Manager v2.1.0-2.1.6 with PKS v1.1.0.

Error: Duplicate Logical Ports (NSX-T)

Symptom

On a PKS deployments using NSX-T, when you upgrade PKS 1.1.x the Ops Manager upgrade task fails with the message `Failed Jobs: pks-nsx-t-prepare-master-vm`.

Explanation

Each logical port must be identified with a different BOSH ID. If the same BOSH ID identifies more than one logical port, the `pks-nsx-t-prepare-master-vm` job fails.

During a PKS upgrade, VM migration can trigger **Fast Suspend and Resume (FSR)**. When FSR is triggered during an upgrade, NSX-T 2.1 might not delete the existing logical port. When BOSH creates a new VM for the cluster, BOSH may attempt to assign the same BOSH ID to the new VM.

Solution

Manually delete the existing logical port and resume the upgrade.

PKS CLI

Page last updated:

This topic describes how to use the Pivotal Container Service Command Line Interface (PKS CLI) to interact with the PKS API.

The PKS CLI is used to create, manage, and delete Kubernetes clusters. To deploy workloads to a Kubernetes cluster created using the PKS CLI, use the Kubernetes CLI, kubectl.

Current Version: 1.1.0-build297

pks login

Login to PKS

Synopsis

The login command requires -a to target the IP of your PKS API, -u for username and -p for password

```
pks login [flags]
```

Examples

```
pks login -a <API> -u <USERNAME> -p <PASSWORD> [--ca-cert <PATH TO CERT> | -k]
```

Options

```
  -a, --api string              The PKS API server URI
      --ca-cert string          Path to CA Cert for PKS API
  -h, --help                    help for login
  -p, --password string         Password
  -k, --skip-ssl-verification   Skip SSL Verification
  -u, --username string         Username
```

pks get-credentials

Allows you to connect to a cluster and use kubectl

Synopsis

Run this command in order to update a kubeconfig file so you can access the cluster through kubectl

```
pks get-credentials <CLUSTER-NAME> [flags]
```

Examples

```
pks get-credentials my-cluster
```

Options

```
-h, --help   help for get-credentials
```

pks cluster

View the details of the cluster

Synopsis

Run this command to see details of your cluster such as name, host, port, ID, number of worker nodes, last operation, etc.

```
pks cluster [flags]
```

Examples

```
pks cluster my-cluster
```

Options

```
-h, --help   help for cluster
    --json   Return the PKS-API output as json
```

pks clusters

Show all clusters created with PKS

Synopsis

This command describes the clusters created via PKS, and the last action taken on the cluster

```
pks clusters [flags]
```

Examples

```
pks clusters
```

Options

```
-h, --help   help for clusters
    --json   Return the PKS-API output as json
```

pks create-cluster

Creates a kubernetes cluster, requires cluster name and an external host name

Synopsis

Create-cluster requires a cluster name, as well as an external hostname. External hostname can be a loadbalancer, from which you access your kubernetes API (aka, your cluster control plane)

```
pks create-cluster <CLUSTER-NAME> [flags]
```

Examples

```
pks create-cluster my-cluster --external-hostname example.hostname --plan production
```

Options

```
  -e, --external-hostname string   Address from which to access Kubernetes API
  -h, --help                       help for create-cluster
      --json                       Return the PKS-API output as json
      --non-interactive            Don't ask for user input
  -n, --num-nodes string           Number of worker nodes
  -p, --plan string                Preconfigured plans. Run pks plans for more details
      --wait                       Wait for the operation to finish
```

pks delete-cluster

Deletes a kubernetes cluster, requires cluster name

Synopsis

Delete-cluster requires a cluster name.

```
pks delete-cluster <CLUSTER-NAME> [flags]
```

Examples

```
pks delete-cluster my-cluster
```

Options

```
  -h, --help              help for delete-cluster
      --non-interactive   Don't ask for user input
      --wait              Wait for the operation to finish
```

pks plans

View the preconfigured plans available

Synopsis

This command describes the preconfigured plans available

```
pks plans [flags]
```

Examples

```
pks plans
```

Options

```
-h, --help   help for plans
    --json   Return the PKS-API output as json
```

pks resize

Increases the number of worker nodes for a cluster

Synopsis

Resize requires a cluster name, and the number of desired worker nodes. Users can only scale UP clusters, to a maximum of 50 worker nodes and not scale down By default, the resize command prompts for interactive confirmation.

```
pks resize <CLUSTER-NAME> [flags]
```

Examples

```
pks resize my-cluster --num-nodes 5
```

Options

```
-h, --help             help for resize
    --json             Return the PKS-API output as json. Only applicable when used with --wait flag
    --non-interactive  Don't ask for user input
-n, --num-nodes int32  Number of worker nodes (default 1)
    --wait             Wait for the operation to finish
```

pks logout

Logs user out of the PKS API

Synopsis

Logs user out of the PKS API. Does not remove kubeconfig credentials or kubectl access.

```
pks logout [flags]
```

Examples

```
pks logout
```

Options

```
-h, --help   help for logout
```