

Pivotal Container Service

Documentation

Version 1.0

Published: 9 Mar 2018

Table of Contents

Table of Contents	2
Pivotal Container Service (PKS)	4
PKS Prerequisites	6
Installing the PKS CLI	7
Installing the Kubernetes CLI	9
Installing PKS on vSphere	11
vSphere Prerequisites and Resource Requirements	12
Installing and Configuring PKS with NSX-T Integration	14
Deploying BOSH and Ops Manager to vSphere	25
Configuring Ops Manager on vSphere	29
Installing and Configuring PKS on vSphere	43
VMware Harbor Registry	49
Installing PKS on GCP	51
GCP Prerequisites and Resource Requirements	52
Preparing to Deploy PKS on GCP	54
Deploying Ops Manager to GCP	59
Configuring Ops Manager Director on GCP	61
Installing and Configuring PKS on GCP	72
Managing PKS	78
PKS API Authentication	79
Configure PKS API Access	81
Manage Users in UAA	82
Manage PKS Deployments with BOSH	83
Add Custom Workloads	84
Download Cluster Logs	85
Prepare Workloads for an Upgrade	86
Delete PKS	88
Using PKS	89
Prerequisites for Using PKS	90
Log in to PKS	91
Create a Cluster	92
Retrieve Cluster Credentials and Configuration	93
View Cluster List	94
View Cluster Details	95
View Cluster Plans	96
Using Dynamic Persistent Volumes	97
Scale Existing Clusters	98
Access the Dashboard	99
Deploy and Access Basic Workloads	100
Delete a Cluster	102
Log Out of the PKS Environment	103
Using Helm with PKS	104
Configure Tiller	105
Install Concourse Using Helm	106
Diagnosing and Troubleshooting PKS	107
Diagnostic Tools	108
Troubleshooting	109
PKS CLI	111

pks cluster	112
pks clusters	113
pks create-cluster	114
pks delete-cluster	115
pks get-credentials	116
pks login	117
pks logout	118
pks plans	119
pks resize	120
PKS Release Notes	121

Pivotal Container Service (PKS)

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

Overview

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

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

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

Features

PKS has the following features:

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

PKS Components

The PKS Controller 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

NSX-T and Harbor are optional integrations for vSphere.

The GCP Service Broker is under development.

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

Installing PKS

To install PKS, you must have a GCP or vSphere environment with Ops Manager v2.0 deployed. If you are installing PKS to vSphere, you can also configure integration with NSX-T and Harbor.

Consult the following table:

IaaS	Ops Manager v2.0	NSX-T	Harbor
vSphere	Required	Available	Available
GCP	Required	Not Available	Not Available

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

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

Managing PKS

See [Managing PKS](#) for information about managing PKS.

Using PKS

After installing PKS, you can use the PKS Command Line Interface (PKS CLI) to do things like create clusters. See [Using PKS](#) for information.

Diagnostics and Troubleshooting PKS

See [Diagnosing and Troubleshooting PKS](#) for information about diagnosing and troubleshooting issues installing or using Pivotal Container Service (PKS).

PKS Prerequisites

Page last updated:

In addition to the prerequisites detailed in the installation and usage topics, Pivotal Container Service (PKS) requires the following:

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

Installing the PKS CLI

Page last updated:

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

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

Mac OS X

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

```
$ chmod +x pkcs
```

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

```
$ mv pkcs /usr/local/bin/pkcs
```

Linux

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

```
$ chmod +x pkcs
```

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

```
$ mv pkcs /usr/local/bin/pkcs
```

Windows

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

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

Installing the Kubernetes CLI

Page last updated:

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

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

Mac OS X

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

```
$ chmod +x kubectl
```

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

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

Linux

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

```
$ chmod +x kubectl
```

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

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

Windows

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

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

Installing PKS on vSphere

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

- [vSphere Prerequisites and Resource Requirements](#)
- [Deploying BOSH and Ops Manager to vSphere](#)
- [Configuring Ops Manager on vSphere](#)
- [Installing and Configuring PKS on vSphere](#)
- [Installing and Configuring PKS with NSX-T](#)
- [Installing and Integrating VMware Harbor Registry with PKS](#) [↗](#)

vSphere Prerequisites and Resource Requirements

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

To install PKS on vSphere, you must have a vSphere environment with Ops Manager v2.0 deployed. You can also configure integration with NSX-T and Harbor.

Consult the following table for compatibility information:

IaaS	Ops Manager v2.0	NSX-T	Harbor
vSphere	Required	Available	Available
GCP	Required	Not Available	Not Available

General PKS Prerequisites

- [Pivotal Ops Manager v2.0](#)



Note: If you have Pivotal Application Service (PAS) installed, Pivotal recommends installing PKS on a separate instance of Ops Manager v2.0. This improves the security of your PAS deployment by ensuring that only PKS-related infrastructure credentials are stored in PKS. Running PKS on a separate instance of Ops Manager v2.0 helps protect your non-PKS infrastructure credentials and prevent their exposure through PKS. For more information about the infrastructure credentials specified in the PKS tile, see the [Kubernetes Cloud Provider](#) section of the *Installing and Configuring PKS* topic.

- [PKS CLI](#)
- [Kubernetes CLI](#)
- (Optional) An external TCP or HTTPS load balancer to access the PKS API
- An external TCP or HTTPS load balancer for each created cluster
- Two or more Virtual Local Area Networks (VLANs) for Ops Manager and the PKS service network

Component Version Requirements

PKS on vSphere supports the following vSphere component versions:

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

NSX-T Integration Component Version Requirements

Deploying NSX-T requires the additional following component versions:

Component	Version
VMware NSX-T	2.1

Resource Requirements

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

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

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

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

NSX-T Integration Resource Requirements

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

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

Installing PKS on vSphere with NSX-T

To install PKS on vSphere **with** NSX-T integration, follow the procedures below:

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

Installing PKS on vSphere

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

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

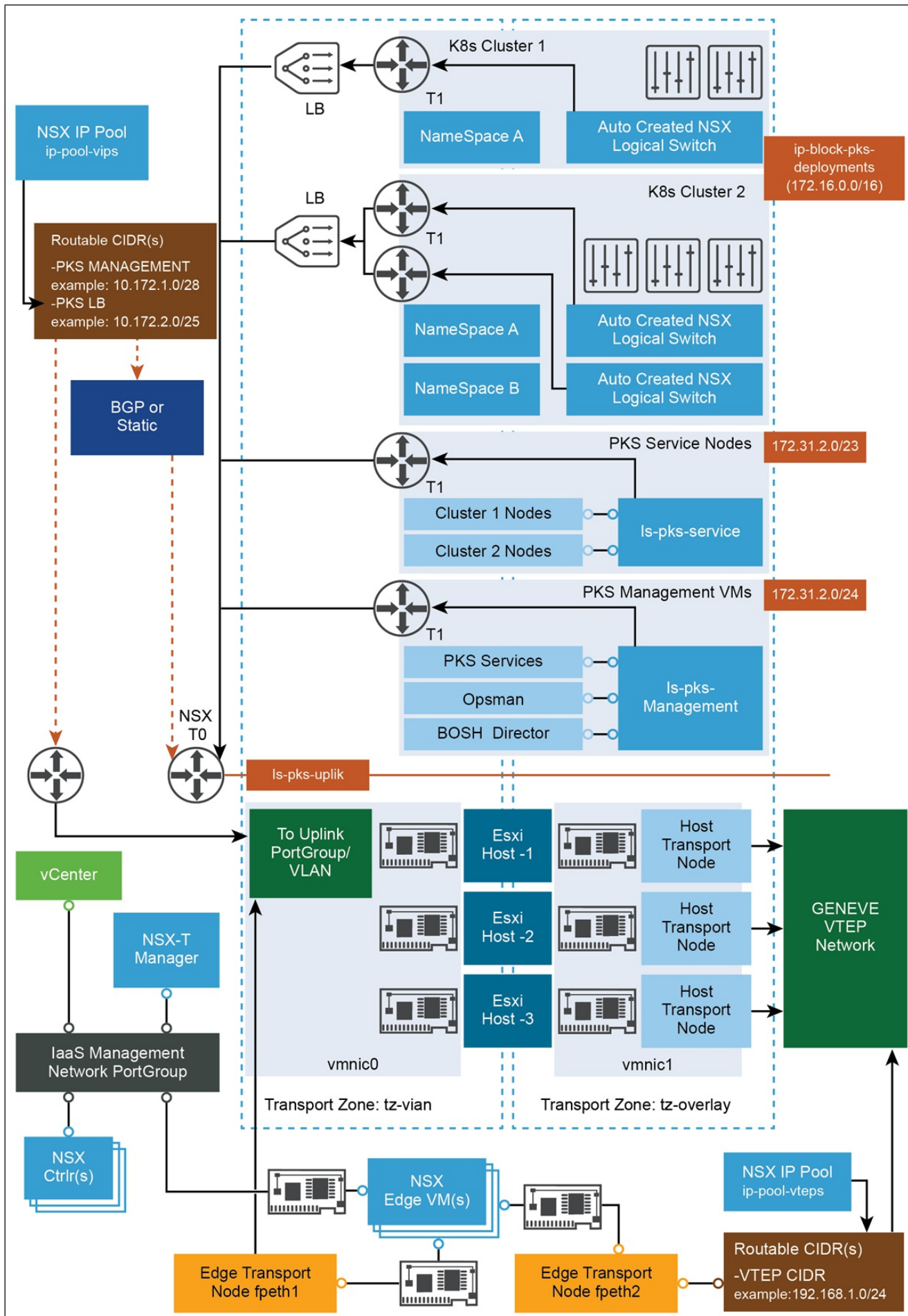
Installing and Configuring PKS with NSX-T Integration

Page last updated:

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

Deployment Architecture

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



Click [here](#) to view a larger version of this image.


The topology has the following characteristics:

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

Before You Install

Before performing the procedures in this topic:

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

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

Step 1: Pre-allocate Network Subnets

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


- **VTEP CIDR(s):** One or more of these networks will host your GENEVE Tunnel Endpoints on your NSX Transport Nodes. Size the network(s) to support all of your expected Host and Edge Transport Nodes. For example, a CIDR of `192.168.1.0/24` will provide 254 usable IPs. This will be used when creating the `ip-pool-vteps` in Step 3.
- **PKS MANAGEMENT CIDR:** This small network will be used for NAT access to PKS management components such as Ops Manager and the PKS Service VM. For example, a CIDR of `10.172.1.0/28` will provide 14 usable IPs.
- **PKS LB CIDR:** This network will provide your load balancing address space for each Kubernetes cluster created by PKS. The network will also provide IP addresses for Kubernetes API access and Kubernetes exposed services. For example, `10.172.2.0/25` will provide 126 usable IPs. This network will be used when creating the `ip-pool-vips` described in [Create NSX Network Objects](#).

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

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

Step 2: Deploy NSX-T

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

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

1. [Deploy](#) the NSX Manager.
2. [Deploy](#) NSX Controllers.
3. [Join](#) the NSX Controllers to the NSX Manager and [initialize](#) the Control Cluster.
4. [Add](#) your ESXi host(s) to the NSX-T Fabric. Each host must have at least one **free nic/vmnic** not already used by other vSwitches on the ESXi host for use with NSX Host Transport Nodes.
5. [Deploy](#) NSX Edge VMs (recommended at least two). Each deployed NSX Edge VM will require 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 T0 uplink interface will be located. Your **PKS MANAGEMENT CIDR** and **PKS LB CIDR** should be routable to this PortGroup.
- c. Connect the third 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.
- d. [Join](#) the NSX Edge VMs to the NSX-T Fabric.

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

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

3.1: Create NSX Network Objects

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

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

5. [Create](#) an NSX IP Block named `ip-block-pks-deployments`. The NSX-T Container Plug-in (NCP) and PKS will use this IP Block to assign address space to Kubernetes pods through the Container Networking Interface (CNI). Pivotal recommends using the CIDR block `172.16.0.0/16`.

3.2: Create Logical Switches

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

3.3: Create NSX Edge Objects

1. [Create](#) NSX Edge Transport Node(s).
2. Add both `tz-vlan` and `tz-overlay` NSX Transport Zones to the NSX Edge Transport Node(s). Controller Connectivity and Manager Connectivity

should be **UP**.

- Refer to the MAC addresses of the Edge VM interfaces you deployed to deploy your virtual NSX Edge(s):
 - Connect the `hs-vlan` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the second NIC from your deployed Edge VM.
 - Connect the `hs-overlay` N-VDS to the vNIC (`fp-eth#`) that matches the MAC address of the third NIC from your deployed Edge VM.
- [Create](#) an NSX Edge cluster called `edge-cluster-pks`.
- Add the NSX Edge Transport Node(s) to the cluster.

3.4: Create Logical Routers

Create T0 Logical Router for PKS

- [Create](#) a Tier-0 (T0) logical router named `t0-pks`:
 - Select `edge-cluster-pks` for the cluster.
 - Set **High Availability Mode** to **Active-Standby**. NAT rules will be applied on T0 by NCP. If not set **Active-Standby**, the router will not support NAT rule configuration.
- [Attach](#) the T0 logical router to the `ls-pks-uplink` logical switch you created previously. Create a logical router port for `ls-pks-uplink` and assign an IP address and CIDR that your environment will use to route to all PKS assigned IP pools and IP blocks.
- [Configure](#) T0 routing to the rest of your environment using the appropriate routing protocol for your environment or by using static routes. The CIDR used in `ip-pool-vips` must route to the IP you just assigned to your t0 uplink interface.

Create T1 Logical Router for PKS Management VMs

- [Create](#) a Tier-1 (T1) logical router for PKS management VMs named `t1-pks-mgmt`:
 - Link to the `t0-pks` logical router you created in a previous step.
 - Select `edge-cluster-pks` for the cluster.
- [Create](#) a logical router port for `ls-pks-mgmt` and assign the following CIDR block: `172.31.0.1/24`.
- [Configure](#) route advertisement on the T1 as follows:
 - Enable **Status**.
 - Enable **Advertise All NSX Connected Routes**.
 - Enable **Advertise All NAT Routes**.
 - Enable **Advertise All LB VIP Routes**.


Configure NAT Rules for PKS Management VMs

[Create](#) the following NAT rules:

For	Action	Settings
Mgmt T1	NO_NAT	
Mgmt Net <=> Service Net	DNAT	External -> Opsman
		External -> Pivotal Container Service
Mgmt Net <=> Service Net	SNAT	Opsman & BOSH Director -> DNS
		Opsman & BOSH Director -> NTP
		Opsman & BOSH Director -> vCenter
		Opsman & BOSH Director -> ESXi
		Opsman & BOSH Director -> NSX-T Manager

The [DNAT](#) rule on the T1 maps an external IP from the **PKS MANAGEMENT CIDR** to the IP where you will deploy Ops Manager on the `ls-pks-mgmt` logical switch. For example, a DNAT rule that maps `10.172.1.2` to `172.31.0.2`, where `172.31.0.2` is the IP address you assign to Ops Manager when connected to `ls-pks-mgmt`. Later, you will create another DNAT rule to map an external IP from the **PKS MANAGEMENT CIDR** to the PKS endpoint.

The [SNAT](#) rule on the T1 allows the PKS Management VMs to communicate with your vCenter and NSX Manager environments. For example, an SNAT rule that maps `172.31.0.0/24` to `10.172.1.1`, where `10.172.1.1` is a routable IP from your **PKS MANAGEMENT CIDR**.

 **Note:** Limit the Destination CIDR for the SNAT rules to the subnet(s) that contain your vCenter and NSX Manager IP addresses.

Create T1 Logical Router for PKS Service VMs


1. [Create](#) a Tier-1 (T1) logical router for PKS Service VMs `t1-pks-service` :
 - Link to the `t0-pks` logical switch you created in a previous step.
 - Select `edge-cluster-pks` for the cluster.
2. [Create](#) a logical router port for `ls-pks-service` and assign the following CIDR block: `172.31.2.1/23`.
3. [Configure](#) route advertisement on the T1 as follows:
 - Enable **Advertise All NSX Connected Routes**.
 - Enable **Advertise All NAT Routes**.
 - Enable **Advertise All LB VIP Routes**.

Configure NAT Rules for PKS Service VMs

[Create](#) the following NAT rules:

For	Action	Settings
Service T1	NO_NAT	
Mgmt Net <-> Service Net	SNAT	K8s Workers -> External Registries (example: DockerHub)
		K8s Workers -> DNS
		K8s Workers -> NTP
		K8s Workers -> NSX-T Manager (NCP)
		K8s Workers -> vCenter (vSphere Cloud Provider)
		K8s Workers -> External Service Endpoints for Workloads

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

 **Note:** Limit the Destination CIDR for the SNAT rules to the subnet(s) that contain your vCenter and NSX Manager IP addresses.


Step 4: Deploy Ops Manager

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


Step 5: Configure Ops Manager

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

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

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

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

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

- On the **Create Networks** page, create the following networks:


Infrastructure Network	Field	Configuration
	Name	<code>pks-infrastructure</code>
	Service Network	Leave Service Network unchecked.
	vSphere Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-infrastructure</code>
	CIDR	<code>192.168.101.0/26</code>
	Reserved IP Ranges	<code>192.168.101.1-192.168.101.9</code>
	DNS	<code>192.168.101.2</code>
	Gateway	<code>192.168.101.1</code>
Main Network	Field	Configuration
	Name	<code>pks-main</code>
	Service Network	Leave Service Network unchecked.
	vSphere Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-pks</code>
	Description	A network for deploying the PKS control plane VM(s) that maps to the NSX logical switch named <code>ls-pks-mgmt</code> created for the PKS Management Network in Step 3: Create the NSX-T Objects Required for PKS .
Service Network	Field	Configuration
	Name	<code>pks-services</code>
	Service Network	Select the Service Network checkbox.
	vSphere Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-services</code>
	Description	A service network for deploying PKS Kubernetes cluster nodes that maps to the NSX logical switch named <code>ls-pks-service</code> created for the PKS Service Network in Step 3: Create the NSX-T Objects Required for PKS .

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

Step 6: Install and Configure PKS

Perform the following steps to install and configure PKS:


- Perform the procedure in [Step 1: Install PKS](#) of *Installing and Configuring PKS on vSphere* to install the PKS tile.
- Click the orange **Pivotal Container Service** tile to start the configuration process.

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

Assign AZs and Networks

Perform the following steps:

- Click **Assign AZs and Networks**.
- Select an availability zone for your singleton jobs and one or more availability zones to balance other jobs in.

 **Note:** In PKS, Pivotal Container Service is a singleton job. This broker VM enables the creation of PKS clusters through the PKS CLI.

3. Under **Network**, select the PKS Management Network linked to the `ls-pks-mgmt` NSX logical switch you created in [Step 5: Configure Ops Manager](#). This will provide network placement for the PKS broker.
4. Under **Service Network**, select the PKS Service Network linked to the `ls-pks-service` NSX logical switch you created in [Step 5: Configure Ops Manager](#). This will provide network placement for the on-demand Kubernetes cluster service instances created by the PKS broker.
5. Click **Save**.

PKS API

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

Plans

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

Kubernetes Cloud Provider

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

Networking

Perform the following steps:

1. Click **Networking**.

Network*

☐ Flannel

☒ NSX-T

NSX Manager hostname *

NSX Manager credentials *

Username

Password

NSX Manager CA Cert

☐ Disable SSL certificate verification?

Cluster Name *

T0 router ID *

IP block ID *

Please enter the IP block ID

Floating IP pool ID *

Save


2. Under **Network**, select **NSX-T** as the **Container Network Type** to use.
3. For **NSX Manager hostname**, enter the NSX Manager hostname or IP address.
4. For **NSX Manager credentials**, enter the credentials to connect to the NSX Manager.
5. For **NSX Manager CA Cert**, optionally enter the custom CA certificate to be used to connect to the NSX Manager.
6. The **Disable SSL certificate verification?** checkbox is **not** selected by default. In order to disable TLS verification, select the checkbox. You may want to disable TLS verification if you did not enter a CA certificate, or if your CA certificate is self-signed.
7. For **vSphere Cluster Name**, enter the name of the vSphere cluster you used when creating the PKS broker in [Assign AZs and Networks](#).
8. For **T0 Router ID**, enter the `t0-pks` T0 router UUID. This can be located in the NSX-T UI router overview.
9. For **IP Block ID**, enter the `ip-block-pks-deployments` IP block UUID. This can also be located in the NSX-T UI.
10. For **Floating IP pool ID**, enter the `ip-pool-vips` Floating IP pool ID that was created for load balancer VIPs.
11. Click **Save**.

UAA


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

(Optional) Syslog

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

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

Errands

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

Perform the following steps:


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

(Optional) Resource Config and Stemcell


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

Step 7: Apply Changes and Retrieve the PKS Endpoint

1. After configuring the tile, return to the Ops Manager Installation Dashboard and click **Apply Changes** to deploy the PKS tile.
2. When the installation is completed, retrieve the PKS endpoint by performing the following steps:
 - a. From the Ops Manager Installation Dashboard, click the **Pivotal Container Service** tile.
 - b. Click the **Status** tab and record the IP address assigned to the `Pivotal Container Service` job.
3. Create a DNAT rule on the `tl-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 Logical Switch. For more information, see [Configure Destination NAT on a Tier-1 Router](#) in the VMware documentation.

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

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

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

Step 8: Deploy a Cluster and Enable NAT Access

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

1. Download the NSX scripts:

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

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

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

3. Install required packages:

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

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

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

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

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

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

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

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

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

```
$ pks clusters
```

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

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

Where:

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

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

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

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

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

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

Deploying BOSH and Ops Manager to vSphere

Page last updated:

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

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

1. In order for Kubernetes to create load balancers and attach persistent disks to pods, you must create a service account in vCenter with sufficient permissions. You can apply the default [VMware Administrator System Role](#) to the service account to achieve the appropriate permission level.

For a full list of the permissions that BOSH requires in order to deploy Ops Manager, see [vSphere Service Account Requirements](#). Apply the permissions from this list if you require a service account with minimal privileges.

2. Refer to the known issues in the [PCF Ops Manager Release v2.0 Release Notes](#) before starting.
3. 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.

Pivotal Network

Pivotal Cloud Foundry Operations Manager

✓ Getting email updates

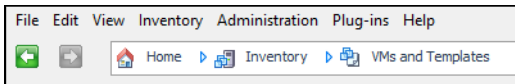
> PRODUCT OVERVIEW

Releases: 2.0.5

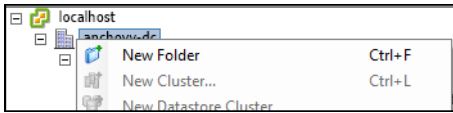
Release Download Files		
	Pivotal Cloud Foundry Ops Manager for vSphere - 2.0-build.249	2.94 GB 2.0-build.249
	Pivotal Cloud Foundry BOSH Assets - 2.0-build.249	2.96 GB 2.0-build.249
	Pivotal Cloud Foundry Ops Manager YAML for Azure - 2.0-build.249	398 Bytes 2.0-build.249
	Pivotal Cloud Foundry Ops Manager for AWS - 2.0-build.249	3.52 KB 2.0-build.249
	Pivotal Cloud Foundry Ops Manager YAML for AWS - 2.0-build.249	419 Bytes 2.0-build.249
	Pivotal Cloud Foundry Ops Manager for Azure - 2.0-build.249	4.21 KB 2.0-build.249
	Pivotal Cloud Foundry Ops Manager for GCP - 2.0-build.249	3.84 KB 2.0-build.249
	Pivotal Cloud Foundry Ops Manager for OpenStack - 2.0-build.249	5.37 GB 2.0-build.249
	Pivotal Cloud Foundry Ops Manager YAML for GCP - 2.0-build.249	152 Bytes 2.0-build.249

4. Log into vCenter.

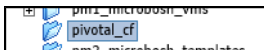
5. Select the **VM and Templates** view.



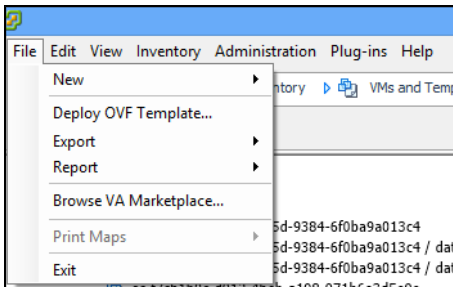
6. Right click on your datacenter and select **New Folder**.



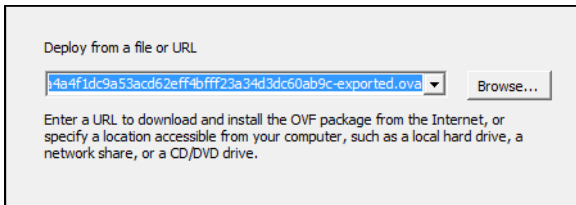
7. Name the folder `pivotal_cf` and select it.



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



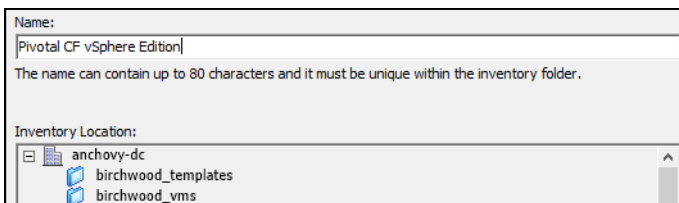
9. Select the .ova file and click **Next**.



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

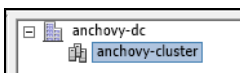
11. Accept the license agreement and click **Next**.

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



Note: The selected folder is the one you created.

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



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

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

Note: Hardware virtualization must be off if your vSphere host does not support VT-X/EPT. Refer to the [PCF on vSphere Requirements](#) topic for more information.

Choose a specific host within the cluster.


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

Select a host from the list below:

Host Name
172.16.64.2

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

Select a destination storage for the virtual machine files:

VM Storage Profile: 

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

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

Warning: Ops Manager v2.0 requires a Director VM with at least 8 GB memory.

Datastore: anchovy-ds

Available space (GB): 4076.0

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

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

Source Networks	Destination Networks
Network 1	<div> <div></div> <div>MattNetwork</div> <div> <div></div> <div>MattNetwork</div> <div>VM Network</div> <div>VM Network Private</div> </div> </div>

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

Application properties - Ops Manager

Product: Ops Manager
Version: 2.0.30.191
Vendor: Pivotal

Unategorized 7 settings

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

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

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

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

NTP Servers: Comma-delimited list of NTP servers

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

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

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

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


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

21. Click **Next**.

22. 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.

23. 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 the [Installing Pivotal Cloud Foundry on vSphere](#)  topic.



Note: Ops Manager security features require you to create a fully qualified domain name to access Ops Manager during the [initial configuration](#).



Next Steps


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

Configuring Ops Manager on vSphere

Page last updated:

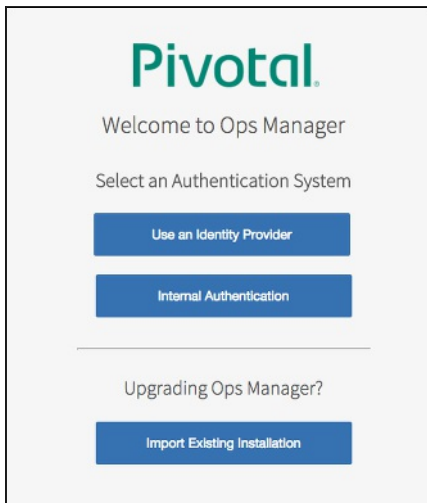
This topic describes how to configure the Ops Manager Director for VMware vSphere.

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

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

Step 1: Set Up Ops Manager

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



Use an Identity Provider

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

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

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

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

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

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

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

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

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

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

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

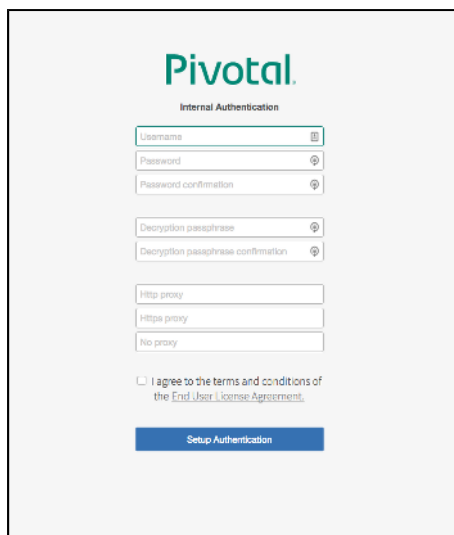
- **Single sign on URL:** `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
- **Audience URI (SP Entity ID):** `https://BOSH-IP:8443`
- **Name ID:** Email Address
- SAML authentication requests are always signed

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

Internal Authentication

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

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



Pivotal
Internal Authentication

Username

Password

Password confirmation

Decryption passphrase

Decryption passphrase confirmation

Http proxy

Https proxy

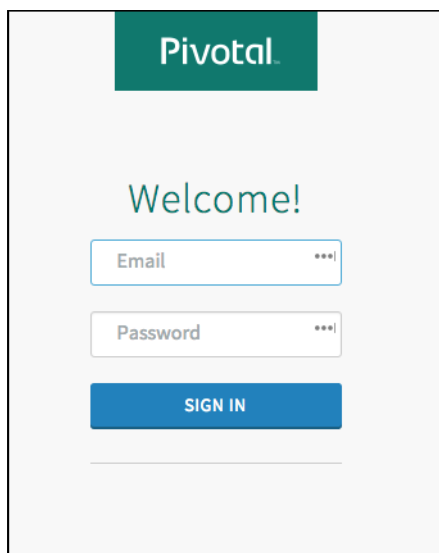
No proxy

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

Setup Authentication

Step 2: vCenter Config Page

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



Pivotal

Welcome!

Email

Password

SIGN IN

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



3. Select **vCenter Config**.

[Installation Dashboard](#)

Ops Manager Director

Settings Status Credentials

vCenter Config

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

vCenter Config

vCenter Host*

vCenter Username*

vCenter Password*

[Change](#)

Datacenter Name*

Virtual Disk Type*

Ephemeral Datastore Names (comma delimited)*

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

Persistent Datastore Names (comma delimited)*

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

☒ Standard vCenter Networking
☐ NSX Networking

NSX Address*

NSX Username*

NSX Password*

NSX CA Cert

Optional custom CA certificate(s)

VM Folder*

Template Folder*

Disk path Folder*

[Save](#)

4. Enter the following information:

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

folder.

5. Select a network configuration from one of the following:

- **Standard vCenter Networking:** This is the default option when upgrading Ops Manager.
- **NSX Networking:** Select this option to enable VMware NSX Network Virtualization.

☐ Standard vCenter Networking

☒ NSX Networking

NSX Mode*

☐ NSX-V

☒ NSX-T

NSX Address*

nsxmanager-6841155-1-pcf-danting

NSX Username*

admin

User to connect to the NSX manager

NSX Password*

[Change](#)

NSX CA Cert

```
-----BEGIN CERTIFICATE-----
MIIDmTCCAoGgAwIBAgIEShW2VDANBgkqhkiG9w0BAQsFADB9MQswCQYDVQQGEwJ
V
UzELMAkGA1UECBMCQ0ExEjAQBgNVBAcTCVBhbG8gQWx0bzEUMBIGA1UEChMLV
3
3
VYUjEFluXy4xDDAKPQNVBAcTAQFTEWDESmMCcGA1UEAxM4bW5lYXVmdj03QDQx
```

6. Configure NSX networking by entering the following information:

- **NSX Mode:** Select **NSX-T**. **NSX-V** is not supported by PKS.
- **NSX Address:** The address of the NSX Manager.
- **NSX Username:** The username to connect to the NSX Manager.
- **NSX Password:** The password for the username specified above.
- **NSX CA Cert:** A CA certificate in PEM format that authenticates to the NSX server. Provide the fully qualified domain name if the NSX Manager uses a self-signed SSL certificate.

Note: To update NSX security group and load balancer information, see the [Updating NSX Security Group and Load Balancer Information](#) topic.

7. Click **Save**.

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

Step 3: Director Config Page

1. Select **Director Config**.

Director Config

NTP Servers (comma delimited)*

time1.sf.cf.app.com

JMX Provider IP Address

Bosh HM Forwarder IP Address

☐ Enable VM Resurrector Plugin

☐ Enable Post Deploy Scripts

☐ Recreate all VMs

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

☐ Enable bosh deploy retries

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

☐ Keep Unreachable Director VMs

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



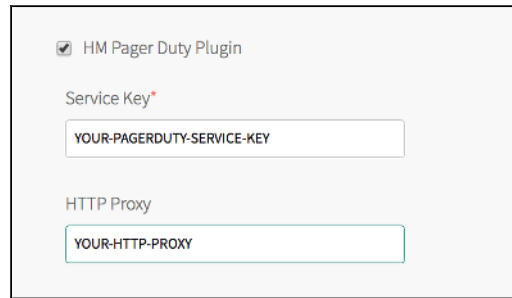
Note: Starting from PCF v2.0, BOSH-reported system metrics are available in the Loggregator Firehose by default. Therefore, if you continue to use PCF JMX Bridge for consuming them outside of the Firehose, you may receive duplicate data. To prevent this, leave the **JMX Provider IP Address** field blank.

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

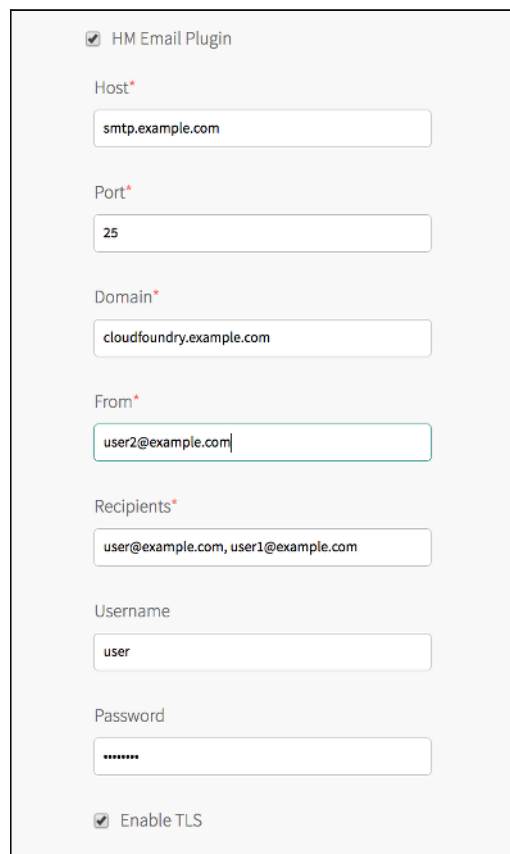


Note: Starting from PCF v2.0, BOSH-reported system metrics are available in the Loggregator Firehose by default. Therefore, if you continue to use the BOSH HM Forwarder for consuming them, you may receive duplicate data. To prevent this, leave the **Bosh HM Forwarder IP Address** field blank.

5. Select the **Enable VM Resurrector Plugin** to enable Ops Manager Resurrector functionality.
6. Select **Enable Post Deploy Scripts** to run a post-deploy script after deployment. This script allows the job to execute additional commands against a deployment.
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.
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](#) from the AWS documentation. When you have created an S3 bucket, complete the following steps:
 1. **S3 Endpoint:** Navigate to the [Regions and Endpoints](#) topic in the AWS documentation. Locate the endpoint for your region in the **Amazon Simple Storage Service (S3)** table and construct a URL using your region's endpoint. For example, if you are using the `us-west-2` region, the URL you create would be <https://s3-us-west-2.amazonaws.com>. Enter this URL into the **S3 Endpoint** field in

Ops Manager.

2. **Bucket Name:** Enter the name of the S3 bucket.
3. **Access Key** and **Secret Key:** Enter the keys you generated when creating your S3 bucket.
4. Select **V2 Signature** or **V4 Signature**. If you select **V4 Signature**, enter your **Region**.

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

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

Blobstore Location

☒ Internal
 ☐ S3 Compatible Blobstore

S3 Endpoint*

Bucket Name*

Access Key*

Secret Key*

☒ V2 Signature
 ☐ V4 Signature

Region*

☐ GCS Blobstore

Bucket Name*

Storage Class*

Regional

Service Account Key*

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

Database Location
☒ Internal
☐ External MySQL Database

Host*


Port*

Username*

Password*

Database*

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


 **Note:** BOSH DNS must be enabled in all PKS deployments. If PAS and PKS are running on the same instance of Ops Manager, you cannot use the opt-out feature of BOSH DNS for your PAS without breaking PKS. If you want to opt out of BOSH DNS in your PAS deployment, install the tile on a separate instance of Ops Manager. For more information about opting out of BOSH DNS, see [this KB article](#) and [Ops Manager v2.0 Release Notes](#).

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

☐ Disable BOSH DNS server for troubleshooting purposes

Custom SSH Banner

19. Click **Save**.

 **Note:** After your initial deployment, you will not be able to edit the Blobstore and Database locations.

Step 4: Create Availability Zone Page

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

1. Select **Create Availability Zones**.

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 the [Understanding Availability Zones in VMware Installations](#) topic.

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 three networks:

- `pks-infrastructure` : for Ops Manager and the BOSH Director.
- `pks-main` : for the PKS broker and PKS API.
- `pks-services` : for creating the master and worker VMs for Kubernetes clusters.



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



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

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:

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

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

5. Click **Save**.



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



Note: If you are using the Cisco Nexus 1000v Switch, refer to the [Using the Cisco Nexus 1000v Switch with Ops Manager](#) topic for more information.

Step 6: Assign AZs and Networks Page

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

Assign AZs and Networks

The Ops Manager Director is a single instance.

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

Singleton Availability Zone

AZ1

Network

Deadmines

Save

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

Step 7: Security Page

1. Select **Security**.

Security

Trusted Certificates

```
-----BEGIN CERTIFICATE-----
TH
-----END CERTIFICATE-----
```

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

Generate VM passwords or use single password for all VMs

- ☒ Generate passwords
- ☐ Use default BOSH password

Save

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

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

Step 8: Syslog Page

1. Select **Syslog**.

Syslog

Do you want to configure Syslog for Bosh Director?

☐ No
☒ Yes

Address*

The address or host for the syslog server

Port*

Transport Protocol*

TCP

⌵

☐ Enable TLS

Permitted Peer*

SSL Certificate*

Save

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

Step 9: Resource Config Page

1. Select **Resource Config**.


Resource Config

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

Save

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

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

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

3. Click **Save**.

Step 10: Complete the Ops Manager Director Installation

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

Next Steps

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

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

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

Installing and Configuring PKS on vSphere

Page last updated:

This topic describes how to install and configure Pivotal Container Service (PKS) on vSphere without NSX-T integration.

Before performing the procedures in this topic, consult the requirements in the [vSphere Prerequisites and Resource Requirements](#) topic.

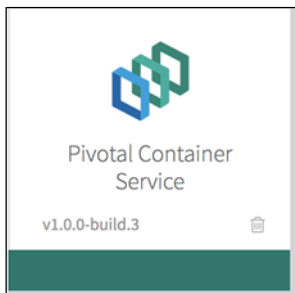
Step 1: Install PKS

Perform the following steps to install PKS:

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. From the **Director Config** page, configure the following settings:
 - Select **Enable Post Deploy Scripts**.
 - Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
4. Click **Apply Changes**.
5. Click **Import a Product** to upload the product file.
6. 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 **Pivotal Container Service** tile to start the configuration process.



Assign AZs and Networks

Perform the following steps:

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

Place singleton jobs in

☒ us-west-2a

☐ us-west-2b

☐ us-west-2c

Balance other jobs in

☐ us-west-2a

☒ us-west-2b

☐ us-west-2c

Network


pk-infrastructure


Service Network

pk-services

Save

2. Select an availability zone (AZ) for your singleton jobs and one or more AZs to balance other jobs in.

 **Note:** If 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.

 **Note:** In PKS, Pivotal Container Service is a singleton job. This broker VM enables the creation of PKS clusters through the PKS CLI.

3. Under **Network**, select a subnet for the PKS broker.
4. Under **Service Network**, select a subnet for the on-demand 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 wildcard certificate and private key pair. The certificate you enter here should cover the domain that routes to the PKS broker VM with TLS termination on the ingress.

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

- a. Select the **Generate RSA Certificate** link.
 - b. Enter the wildcard domain that matches the URL used to access PKS API and UAA endpoints. For example, if your [UAA URL](#) is `api.pks.example.com`, then enter `*.pks.example.com`.
 - c. Click **Generate**.
3. 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.

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 **Default Cluster Authorization Mode**, select an authentication mode for the Kubernetes clusters. Pivotal recommends selecting **RBAC**. For more information, see the [RBAC Support in Kubernetes](#) [blog post](#).
6. Under **AZ placement**, select an AZ for the Kubernetes clusters deployed by PKS.
7. Under **ETCD/Master VM Type**, select the type of VM to use for Kubernetes etcd and master nodes.
8. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master VM.
9. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker nodes.
10. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker nodes.
11. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, Pivotal recommends creating clusters with at least 3 worker nodes.
12. Under **Errand VM Type**, select the size of the VM where the errand will run. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.
13. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to [add custom workloads](#) to each cluster in this plan. You can specify multiple files using `---` as a separator.
14. If you want users to be able to create [pods with privileged containers](#) [blog post](#), select the **Enable Privileged Containers - Use with caution** option.
15. 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

Perform the following steps:

1. Click **Kubernetes Cloud Provider**.
2. Under **Choose your IaaS**, select vSphere.
3. Perform the steps specific to vSphere.
 - Ensure the values match those in the **vCenter Config** section of the **Ops Manager** tile:

Choose your IaaS*

☐ GCP
 ☒ vSphere

vCenter Credentials *

Username

Password

vCenter Host *

Datacenter Name *

Datastore Name *

Stored VM Folder *

1. Enter your **vCenter Credentials**.
2. Enter your **vCenter Host**, such as `vcenter.cf-example.com`.
3. Enter your **Datacenter Name**, such as `cf-example-dc`.
4. Enter your **Datastore Name**, such as `cf-example-ds`.
5. Enter the **Stored VM Folder** so that the persistent stores know where to find the VMs. To retrieve the name of the folder, navigate to your Ops Manager Director tile, click **vCenter Config**, and locate the value for **VM Folder**. The default folder name is `pcf_vms`.

4. Click **Save**.

Networking

Perform the following steps:

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


UAA

Perform the following steps:

1. Click **UAA**.
2. For **UAA URL**, enter a domain name to use when accessing UAA and the PKS API. For example, `api.pks.example.com`. This domain name must match the certificate you generated when configuring the **PKS API** pane.
3. Enter the time (in seconds) for the PKS CLI access token lifetime.
4. Enter the time (in seconds) for the PKS CLI refresh token lifetime.


(Optional) Syslog

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

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

To specify the destination for PKS log messages, perform the following steps:

1. Click **Syslog**.
2. Select **Yes** to configure syslog forwarding.
3. Enter the destination syslog endpoint.
4. Enter the destination syslog port.
5. Select a transport protocol for log forwarding.
6. (Optional) Pivotal strongly recommends that you enable TLS encryption when forwarding logs as they may contain sensitive information. For example, these logs may contain cloud provider credentials. To enable TLS, perform the following steps.
 - a. Provide the accepted fingerprint (SHA1) or name of remote peer. For example, `*.YOUR-LOGGING-SYSTEM.com`.
 - b. Provide a TLS certificate for the destination syslog endpoint.

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

Errands

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

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

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


(Optional) Resource Config

To modify the resource usage of PKS, click **Resource Config** and edit the **PKS on-demand broker** job.

(Optional) Stemcell

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

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

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

Step 3: Apply Changes

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

Step 4: Retrieve PKS API Endpoint

You must share the PKS API endpoint to allow your organization to use the API to create, update, and delete clusters.

When an operator creates a cluster, they provide an IP address for the Kubernetes master host, then point the load balancer to the newly created cluster. If you use a load balancer as a service (LBaaS) tool, your LBaaS may manage cluster creation and configuration.

See [Create a Cluster](#) for more information.

Perform the following steps to retrieve the PKS API endpoint:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the PKS tile.
3. Click the **Status** tab and locate the IP address of the PKS API endpoint. This is the endpoint that developers use to create and manage clusters.

Step 5: Configure External Load Balancer

Configure your external TCP or HTTPS load balancer to resolve to the domain name used in the certificate you provided during the [PKS API](#) section of the tile configuration. Your external load balancer forwards traffic to the PKS API endpoint on port 9021 and the UAA endpoint on port 8443.

The load balancer should be configured with:

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

VMware Harbor Registry

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

Overview

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

Key Features

Harbor includes the following key features:

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

Versions and Compatibility

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

Element	Details
Tile version	v1.4.1
Release date	February 8, 2018
Software component version	v1.4.1
Compatible Ops Manager version(s)	v2.0
Compatible Pivotal Container Service (PKS) version(s)	v1.0
Compatible Pivotal Application Service (PAS) version(s)	v2.0
IaaS support	vSphere
IPsec support?	No

Requirements

There are no special requirements for deploying VMware Harbor Registry.

Limitations

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

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

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

License

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

Installing PKS on GCP

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

- [GCP Prerequisites and Resource Requirements](#)
- [Preparing to Deploy PKS on GCP](#)
- [Deploying BOSH and Ops Manager to GCP](#)
- [Configuring Ops Manager Director on GCP](#)
- [Installing and Configuring PKS on GCP](#)

GCP Prerequisites and Resource Requirements

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


To install PKS GCP, you must have a GCP environment with Ops Manager v2.0 deployed.

Consult the following table for compatibility information:

IaaS	Ops Manager v2.0	NSX-T	Harbor
vSphere	Required	Available	Available
GCP	Required	Not Available	Not Available

General PKS Prerequisites

- Ops Manager v2.0

 **Note:** If you have Pivotal Application Service (PAS) installed, Pivotal recommends installing PKS on a separate instance of Ops Manager v2.0. This improves the security of your PAS deployment by ensuring that only PKS-related infrastructure credentials are stored in PKS. Running PKS on a separate instance of Ops Manager v2.0 helps protect your non-PKS infrastructure credentials and prevent their exposure through PKS. For more information about the infrastructure credentials specified in the PKS tile, see the [Kubernetes Cloud Provider](#) section of the *Installing and Configuring PKS* topic.

- [PKS CLI](#)
- [Kubernetes CLI](#)
- (Optional) An external TCP or HTTPS load balancer to access the PKS API
- An external TCP or HTTPS load balancer for each created cluster
- Two or more Virtual Local Area Networks (VLANs) for Ops Manager and the PKS service network

Resource Requirements

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

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

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

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

Installing PKS on GCP

To install PKS on GCP, follow the procedures below:

- [Preparing to Deploy PKS on GCP](#)
- [Deploying Ops Manager to GCP](#)
- [Configuring Ops Manager Director on GCP](#)
- [Installing and Configuring PKS on GCP](#)

Preparing to Deploy PKS on GCP

Page last updated:

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

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

Step 1: Set up an IAM Service Account

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

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:
 - `roles/compute.instanceAdmin` (Compute Engine > Compute Instance Admin)
 - `roles/compute.securityAdmin` (Compute Engine > Compute Security Admin)
 - `roles/compute.networkAdmin` (Compute Engine > Compute Network Admin)
 - `roles/compute.storageAdmin` (Compute Engine > Compute Storage Admin)
 - `roles/compute.viewer` (Compute Engine > Compute Viewer)
4. Select **Furnish a new private key** and select **JSON**.
5. Click **Create**. Your browser automatically downloads a JSON file with a private key for this account. Save this file in a secure location.

Step 2: Enable Google Cloud APIs

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

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

```
$ gcloud auth login
```

- b. List your projects:

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

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

Step 3: Create a GCP Network with Subnets

1. Log in to the [GCP Console](#).
2. Navigate to the GCP project where you want to install PKS.
3. Select **VPC network**, then **CREATE VPC NETWORK**.
4. In the **Name** field, enter `MY-PKS-virt-net`. `MY-PKS` is a prefix to help you identify resources for this PKS deployment in the GCP console.
 - a. Under **Subnets**, complete the form as follows to create an infrastructure subnet for Ops Manager and NAT instances:

Name	<code>MY-PKS-subnet-infrastructure-GCP-REGION</code>
Region	A region that supports three availability zones (AZs). For help selecting the correct region for your deployment, see the Google documentation on regions and zones .
IP address range	A CIDR ending in <code>/26</code> Example: <code>192.168.101.0/26</code>

- b. Click **Add subnet** to add a second subnet for the BOSH Director, PKS API, and PKS broker with the following details:

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

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

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

5. Under **Dynamic routing mode**, leave **Regional** selected.
6. Click **Create**.

Step 4: Create NAT Instances

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

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


```
#!/bin/bash
sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```
5. Click **Networking** to open additional network configuration fields:
 - a. In the **Network tags** field, add the following: `nat-traverse` and `MY-PKS-nat-instance`.
 - b. Click the pencil icon to edit the **Network interface**.

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

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

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

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

Create Routes for NAT Instances


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

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

Step 5: Create Firewall Rules for the Network

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

1. In the **Networking** pane, select **Firewall rules**.
2. Create firewall rules according to the table below:

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

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

3. If you are only using your GCP project to deploy PKS, then you can delete the following default firewall rules:
 - o default-allow-http
 - o default-allow-https
 - o default-allow-icmp
 - o default-allow-internal
 - o default-allow-rdp
 - o default-allow-ssh

Next Steps

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

Deploying Ops Manager to GCP

Page last updated:

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

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

Step 1: Locate the Pivotal Ops Manager Installation File

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

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

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

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

Step 2: Create a Private VM Image

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

Step 3: Create the Ops Manager VM Instance

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

Machine Spec	Minimum Value
CPU	2 vCPUs
Memory	8 GB

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

however, you can select **Standard**.

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


Machine Spec	Minimum Value
Boot disk	100 GB

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

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

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

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

 **Note:** In order to set up Ops Manager authentication correctly, Pivotal recommends using a Fully Qualified Domain Name (FQDN) to access Ops Manager. Using an ephemeral IP address to access Ops Manager can cause authentication errors upon subsequent access.


Next Steps

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

Configuring Ops Manager Director on GCP


Page last updated:

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

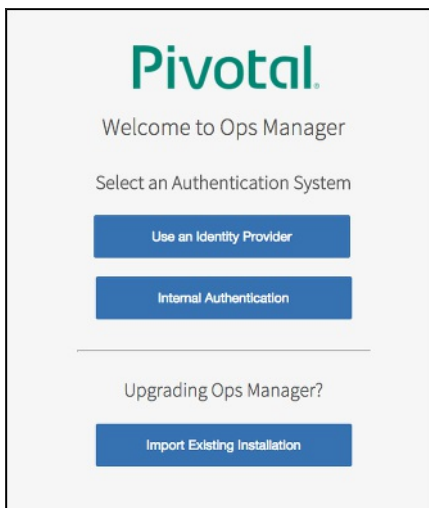
 **Note:** You can also perform the procedures in this topic using the Ops Manager API. For more information, see the [Using the Ops Manager API](#) topic.

Step 1: Access Ops Manager

1. In a web browser, navigate to the fully qualified domain name (FQDN) of Ops Manager that you set up in [Launching an Ops Manager Director Instance on GCP](#).

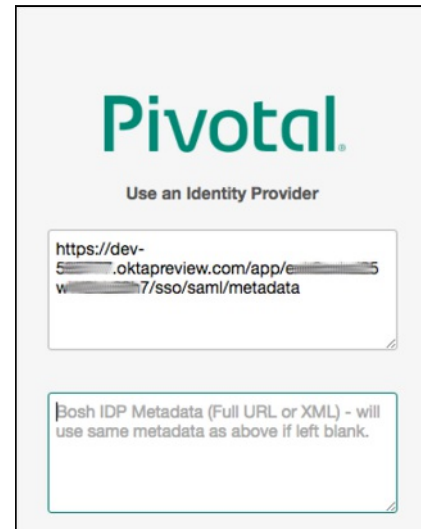
 **Note:** In order to set up Ops Manager authentication correctly, Pivotal recommends using a Fully Qualified Domain Name (FQDN) to access Ops Manager. Using an ephemeral IP address to access Ops Manager can cause authentication errors upon subsequent access.

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



Use an Identity Provider (IdP)

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



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

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

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

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

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

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

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

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

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

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

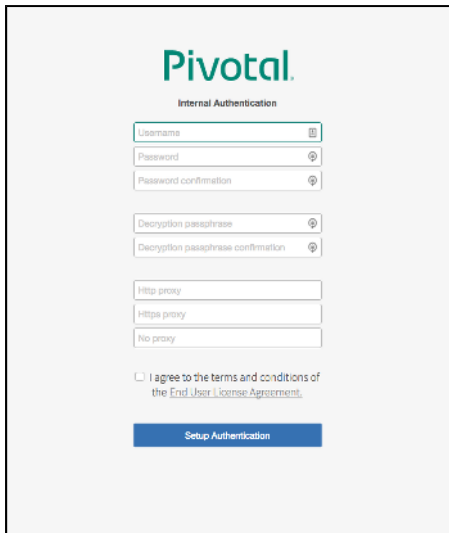
- **Single sign on URL:** `https://BOSH-IP:8443/saml/SSO/alias/BOSH-IP`
- **Audience URI (SP Entity ID):** `https://BOSH-IP:8443`
- **Name ID:** Email Address
- SAML authentication requests are always signed

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

Internal Authentication

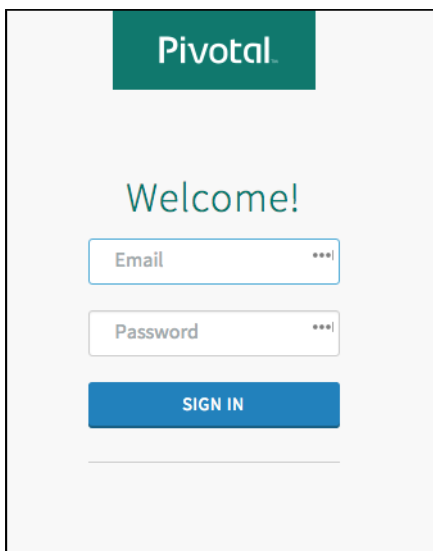
1. When redirected to the **Internal Authentication** page, you must complete the following steps:

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



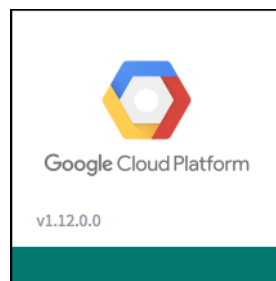
The image shows the 'Pivotal Internal Authentication' form. It includes fields for Username, Password, Password confirmation, Decryption passphrase, and Decryption passphrase confirmation. There are also fields for Http proxy, Https proxy, and No proxy. A checkbox for 'I agree to the terms and conditions of the End User License Agreement' is present, followed by a 'Setup Authentication' button.

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



The image shows the 'Pivotal Welcome!' login form. It features a 'Pivotal' logo at the top, followed by 'Welcome!'. Below this are input fields for 'Email' and 'Password', both with masked characters (***). A blue 'SIGN IN' button is at the bottom.

Step 2: Google Cloud Platform Config



1. Click the **Google Cloud Platform** tile within the **Installation Dashboard**.
2. Select **Google Config**. Complete the following fields:
 - **Project ID**: Enter your GCP project ID in all lower case, such as: `your-gcp-project-id`.
 - **Default Deployment Tag**: Enter the `MY-PKS` prefix that you used when creating the GCP resources for this PCF installation.
 - Select **AuthJSON** and in the field below enter the contents of the JSON file that you downloaded in the [Set up an IAM Service Account](#) section of the *Preparing to Deploy PKS on GCP* topic.

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

3. Click **Save**.

Step 3: Director Config Page

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

2. In the **NTP Servers (comma delimited)** field, enter `metadata.google.internal`.
3. Leave the **JMX Provider IP Address** field blank.
4. Leave the **Bosh HM Forwarder IP Address** field blank.
5. Select the **Enable VM Resurrector Plugin** checkbox to enable the Ops Manager Resurrector functionality and increase runtime availability.
6. Select **Enable Post Deploy Scripts** to run a post-deploy script after deployment. This script allows the job to execute additional commands against a deployment.

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

☒ HM Pager Duty Plugin

Service Key*

HTTP Proxy

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

☒ HM Email Plugin

Host*

Port*

Domain*

From*


Recipients*

Username

Password


☒ Enable TLS

11. (Optional) 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 password for your email server.
 - **Enable TLS:** Select this checkbox to enable Transport Layer Security.
12. Select a **Blobstore Location** to either configure the blobstore as an internal server or an external endpoint. Because the internal server is unscalable and less secure, Pivotal recommends you configure an external blobstore.

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

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

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

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

- **GCS Blobstore:** Select this option to use an external Google Cloud Storage (GCS) endpoint. To create a GCS bucket, follow the procedures in [Creating Storage Buckets](#). When you have created a GCS bucket, complete the following steps:
1. **Bucket Name:** Enter the name of your GCS bucket.
 2. **Storage Class:** Select the storage class for your GCS bucket. See [Storage Classes](#) in the GCP documentation for more information.
 3. **Service Account Key:** Enter the contents of the JSON file that you downloaded in the [Set Up an IAM Service Account](#) section of the

Blobstore Location

☒ Internal
 ☐ S3 Compatible Blobstore

S3 Endpoint*

Bucket Name*

Access Key*

Secret Key*

☒ V2 Signature
 ☐ V4 Signature

Region*

☐ GCS Blobstore

Bucket Name*


Storage Class*

Regional


Service Account Key*

Preparing to Deploy PKS on GCP topic.

13. For **Database Location**, select **Internal**.
14. (Optional) Modify the **Director Workers** value, which sets the number of workers available to execute Director tasks. This field defaults to **5**.
15. (Optional) **Max Threads** sets the maximum number of threads that the Ops Manager Director can run simultaneously. Pivotal recommends that you leave the field blank to use the default value, unless doing so results in rate limiting or errors on your IaaS.
16. (Optional) To add a custom URL for your Ops Manager Director, enter a valid hostname in **Director Hostname**. You can also use this field to configure [a load balancer in front of your Ops Manager Director](#).

 **Note:** Leave this field blank for PKS deployments.

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

 **Note:** BOSH DNS must be enabled in all PKS deployments. If PAS and PKS are running on the same instance of Ops Manager, you cannot use the opt-out feature of BOSH DNS for your PAS without breaking PKS. If you want to opt out of BOSH DNS in your PAS deployment, install the tile on a separate instance of Ops Manager. For more information about opting out of BOSH DNS, see [this KB article](#) and [Ops Manager v2.0 Release Notes](#).

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

☐ Disable BOSH DNS server for troubleshooting purposes

Custom SSH Banner

19. Click **Save**.

Step 4: Create Availability Zones Page

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

Create Availability Zones

Availability Zones

Add

▶ us-central1-b

▼ us-central1-a

Google Availability Zone*

us-central1-a

The Google Availability Zone name

▶ us-central1-c

Save

- Click **Save**.

4. Repeat the above step for all the availability zones you are using in your deployment. When you are done, click **Save**.

Step 5: Create Networks Page

- Select **Create Networks**.
- Make sure **Enable ICMP checks** is not selected. GCP routers do not respond to ICMP pings.
- Click **Add Network** and create the following three networks:
 - `pks-infrastructure` : for Ops Manager and the BOSH Director.
 - `pks-main` : for the PKS broker and PKS API.
 - `pks-services` : for creating the master and worker VMs for Kubernetes clusters.

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

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 GCP environment:

Note: Pivotal recommends using the Google-provided DNS server, `169.254.169.254`, as your default DNS server.

Note: To use a shared VPC network, enter the shared VPC host project name before the network name in the format `VPC-PROJECT-NAME/NETWORK-NAME/SUBNET-NAME/REGION-NAME`. For example, `vpc-project/opsmgr/central/us-central1`. For more information, see [Configuring a Shared VPC on GCP](#).

	Field	Configuration
Infrastructure Network	Name	<code>pks-infrastructure</code>
	Service Network	Leave Service Network unchecked.
	Google Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-infrastructure-GCP-REGION/GCP-REGION</code>
	CIDR	<code>192.168.101.0/26</code>
	Reserved IP Ranges	<code>192.168.101.1-192.168.101.9</code>
	DNS	<code>169.254.169.254</code>
	Gateway	<code>192.168.101.1</code>
	Field	Configuration
	Name	<code>pks-main</code>
	Service Network	Leave Service Network unchecked.
	Google Network Name	<code>MY-PKS-virt-net/MY-PKS-subnet-pks-GCP-REGION/GCP-REGION</code>

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

Step 6: Assign AZs and Networks Page

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

Step 7: Security Page

1. Select **Security**.

Security

Trusted Certificates

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

TH

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

Generate VM passwords or use single password for all VMs

☒ Generate passwords
 ☐ Use default BOSH password

Save

2. In **Trusted Certificates**, enter a custom certificate authority (CA) certificate to insert into your organization's certificate trust chain. This feature enables all BOSH-deployed components in your deployment to trust a custom root certificate.
 - You do not need to enter anything in this field if you are using self-signed certificates.
 - If you want to use Docker Registries for running app instances in Docker containers, enter the certificate for your private Docker Registry in this field. See the [Using Docker Registries](#) topic for more information.

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

Step 8: Syslog Page

1. Select **Syslog**.

Syslog

Do you want to configure Syslog for Bosh Director?

☐ No

☒ Yes

Address*

The address or host for the syslog server

Port*

Transport Protocol*

TCP

⌵

☐ Enable TLS

Permitted Peer*

SSL Certificate*

Save

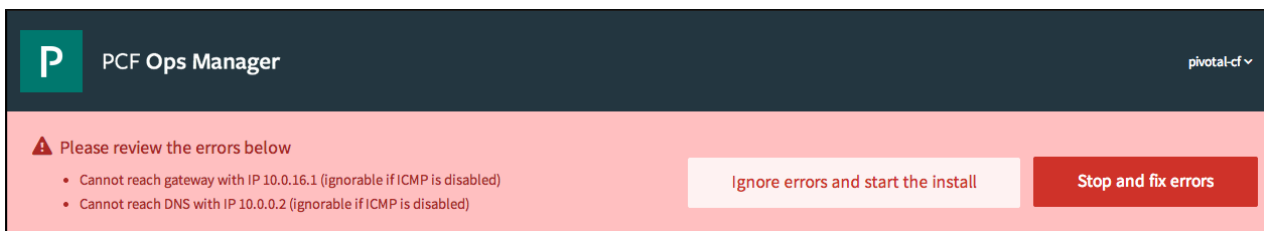
2. (Optional) To send BOSH Director system logs to a remote server, select **Yes**.
3. In the **Address** field, enter the IP address or DNS name for the remote server.
4. In the **Port** field, enter the port number that the remote server listens on.
5. In the **Transport Protocol** dropdown menu, select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

Step 9: Resource Config Page

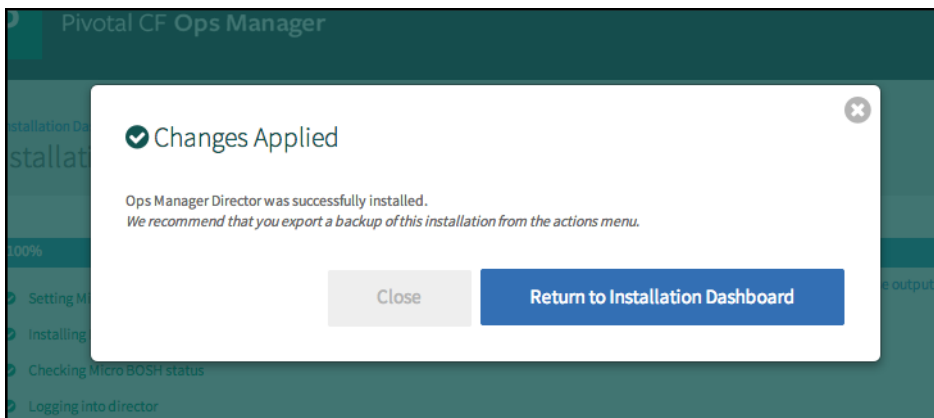
1. Select **Resource Config**.
2. Ensure that the **Internet Connected** checkboxes are not selected for any jobs. The checkbox gives VMs a public IP address that enables outbound Internet access. In [Preparing to Deploy PKS on GCP](#), you provisioned a Network Address Translation (NAT) box to provide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

Step 10: Complete the Ops Manager Director Installation

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



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



Next Steps

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

Installing and Configuring PKS on GCP

Page last updated:

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

Before performing the procedures in this topic, consult the [GCP Prerequisites and Resource Requirements](#) topic.

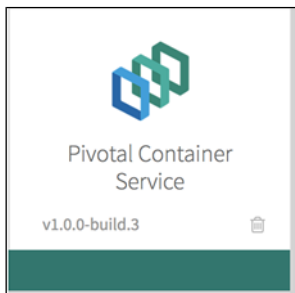
Step 1: Install PKS

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. From the **Director Config** page, configure the following settings:
 - Select **Enable Post Deploy Scripts**.
 - Clear the **Disable BOSH DNS server for troubleshooting purposes** checkbox.
4. Click **Apply Changes**.
5. Click **Import a Product** to upload the product file.
6. 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

To assign AZs and networks, do the following:

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

Place singleton jobs in

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

Balance other jobs in

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

Network

Service Network

Save

2. Select an availability zone (AZ) for your singleton jobs and one or more AZs to balance other jobs in.

Note: If 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.

Note: In PKS, Pivotal Container Service is a singleton job. This broker VM enables the creation of PKS clusters through the PKS CLI.

3. Under **Network**, select a subnet for the PKS broker.
4. Under **Service Network**, select a subnet for the on-demand 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 wildcard certificate and private key pair. The certificate you enter here should cover the domain that routes to the PKS broker VM with TLS termination on the ingress.

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

- a. Select the **Generate RSA Certificate** link.
- b. Enter the wildcard domain that matches the URL used to access PKS API and UAA endpoints. For example, if your [UAA URL](#) is `api.pks.example.com`, then enter `*.pks.example.com`.
- c. Click **Generate**.

3. Click **Save**.

This is the wildcard domain you use when setting up your DNS entry for your load balancer in [Step 5](#).

Plans

To activate a plan, do the following:

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

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

2. Select **Active** to activate the plan and make it available to developers deploying clusters.
3. Under **Name**, provide a unique name for the plan.
4. Under **Description**, edit the description as needed. The plan description appears in the Services Marketplace, which developers can access by using PKS CLI.
5. Under **AZ placement**, select an AZ for the Kubernetes clusters deployed by PKS.
6. Under **Default Cluster Authorization Mode**, select an authentication mode for the Kubernetes clusters. Pivotal recommends selecting **RBAC**. For more information, see the [RBAC Support in Kubernetes](#) blog post.
7. Under **ETCD/Master VM Type**, select the type of VM to use for Kubernetes etcd and master nodes.
8. Under **Master Persistent Disk Type**, select the size of the persistent disk for the Kubernetes master VM.
9. Under **Worker VM Type**, select the type of VM to use for Kubernetes worker nodes.
10. Under **Worker Persistent Disk Type**, select the size of the persistent disk for the Kubernetes worker nodes.
11. Under **Worker Node Instances**, select the default number of Kubernetes worker nodes to provision for each cluster. For high availability, Pivotal recommends creating clusters with at least 3 worker nodes.
12. Under **Errand VM Type**, select the size of the VM where the errand will run. The smallest instance possible is sufficient, as the only errand running on this VM is the one that applies the **Default Cluster App** YAML configuration.
13. (Optional) Under **(Optional) Add-ons - Use with caution**, enter additional YAML configuration to [add custom workloads](#) to each cluster in this plan. You can specify multiple files using `---` as a separator.
14. If you want users to be able to create [pods with privileged containers](#), select the **Enable Privileged Containers - Use with caution** option.
15. Click **Save**.

To deactivate a plan, do the following:

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

Kubernetes Cloud Provider

To configure your Kubernetes cloud provider settings, do the following:

1. Click **Kubernetes Cloud Provider**.
2. Under **Choose your IaaS**, select **GCP**.
3. Perform the steps specific to GCP.
 - Ensure the values match those in the **Google Config** section of the **Ops Manager** tile:
 1. Enter your **GCP Project Id**, which is the name of the deployment in your Ops Manager environment.
 2. Enter your **VPC Network**, which is the VPC network name for your Ops Manager environment.
 3. Enter your **GCP Service Key**, which you created using [the instructions from the GCP prerequisites](#).
4. Click **Save**.

Networking

To configure networking, do the following:

1. Click **Networking**.
2. Under **Network**, select **Flannel**.

3. Click **Save**.

UAA

To configure UAA, do the following:

1. Click **UAA**.
2. For **UAA URL**, enter a domain name to use when accessing UAA and the PKS API. For example, `api.pks.example.com`. This domain name must match the certificate you generated when configuring the [PKS API](#) pane.
3. Enter the time (in seconds) for the PKS CLI access token lifetime.
4. Enter the time (in seconds) for the PKS CLI refresh token lifetime.

(Optional) Syslog

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

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

1. Click **Syslog**.
2. Select **Yes** to configure syslog forwarding.
3. Enter the destination syslog endpoint.
4. Enter the destination syslog port.
5. Select a transport protocol for log forwarding.
6. (Optional) Pivotal strongly recommends that you enable TLS encryption when forwarding logs as they may contain sensitive information. For example, these logs may contain cloud provider credentials. To enable TLS, perform the following steps.
 - a. Provide the accepted fingerprint (SHA1) or name of remote peer. For example, `*.YOUR-LOGGING-SYSTEM.com`.
 - b. Provide a TLS certificate for the destination syslog endpoint.



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

Errands

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

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

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

(Optional) Resource Config

To modify the resource usage of PKS, click **Resource Config** and edit the **PKS on-demand broker** job.

If you created a PKS API load balancer, you should give it a name in the **Load Balancers** column. For more information, see [Step 5: Configure External Load Balancer](#).

(Optional) Stemcell

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

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

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

Step 3: Apply Changes

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

Step 4: Retrieve PKS API Endpoint

You must share the PKS API endpoint to allow your organization to use the API to create, update, and delete clusters.

When an operator creates a cluster, they provide an IP address for the Kubernetes master host, then point the load balancer to the newly created cluster. If you use a load balancer as a service (LBaaS) tool, your LBaaS may manage cluster creation and configuration.

See [Using PKS](#) for more information.

To retrieve the PKS API endpoint, do the following:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the PKS tile.
3. Click the **Status** tab and locate the IP address of the PKS API endpoint. This is the endpoint that developers use to create and manage clusters.

Step 5: Configure External Load Balancer

Configure your external TCP load balancer to resolve to the domain name used in the certificate you provided during the [PKS API](#) section of the tile configuration.

Create Load Balancer for PKS API

1. Navigate to **Network Services > Load balancing** and click **Create load balancer**.
2. Under **TCP Load Balancing**, click **Start configuration**.
3. Select whether you want to load balance traffic from the Internet to your VMs or only between your VMs.
4. Select whether you want to place the backends for your load balancer in a single region or across multiple regions.
5. Give your load balancer a name.
6. Configure your backend.
 - Give your backend a name.
 - Select a region.
 - Select backends from either instance groups or VM instances in the target pool's region.
 - (Optional) Select a backup pool.
 - (Optional) Select whether you want to create a health check or go without one.
 - Select a session affinity. **None** will enable successive requests from a particular client IP to go to any instance in the pool. **Client IP** will send requests from the client IP to the same instance. **Client IP and protocol** will send requests from the client IP with the same protocol to the same instance.
7. Configure your frontend.
 - Give your frontend a name.

- Select **Create IP address** to reserve the IP address from [Step 4: Retrieve PKS API Endpoint](#).
- Specify 8443 and 9021 as your ports. Your external load balancer forwards traffic to the UAA endpoint on port 8443 and the PKS API endpoint on port 9021.
- Click **Done**.

8. Click **Create**.

Create Firewall Rule for Load Balancer

1. Navigate to **VPC Network > Firewall rules** and click **Create firewall rule**.
2. Configure the following:
 - Give your firewall rule a name.
 - Select the network to which the firewall rule will apply.
 - Specify the priority of this firewall rule.
 - Select **Ingress** as the **Direction of traffic**.
 - Select **Allow** as the **Action on match**.
 - Select the load balancer you just created as your target.
Note: Make sure that the PKS VM hosting the PKS API and Broker has the `pks-api` network tag. If it does not have the `pks-api` tag, add it so the above firewall rule will apply to it.
 - Use `0.0.0.0/0` as your source IP range.
 - Under **Protocols and ports**, select **Specified protocols and ports** and specify `tcp:8443,9021`.
3. Click **Apply Changes**.

Create Wildcard DNS Entry to Load Balancer

To create a DNS entry for the wildcard domain and certificate you generated in [Step 3](#), do the following:

1. Navigate to **Network Services > Cloud DNS**.
2. If you do not already have a DNS zone, click **Create zone**.
 - Give a zone name and a DNS name.
 - Specify whether the **DNSSEC** state of the zone is Off, On, or Transfer.
 - Click **Create**.
3. Add a DNS record for your load balancer.

Managing PKS

Page last updated:

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

- [Configure PKS API Access](#)
- [Manage Users in UAA](#)
- [Manage PKS Deployments with BOSH](#)
- [Add Custom Workloads](#)
- [Download Cluster Logs](#)
- [Prepare Workloads for an Upgrade](#)
- [Delete PKS](#)

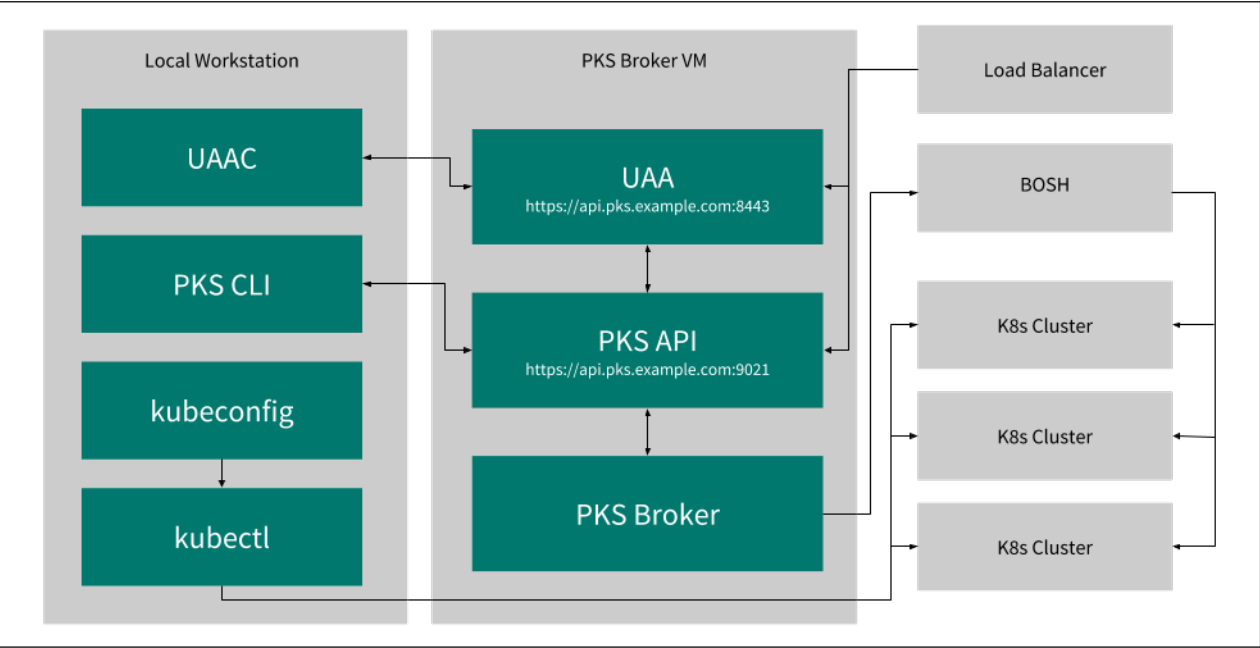
PKS API Authentication

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.

Overview

PKS deploys a broker VM that contains the PKS API server, a UAA server, and the PKS broker. To communicate with these services, you install command-line tools for [PKS](#), [UAA](#), and [Kubernetes](#) on your local workstation. See the following diagram for more information about how these services interact:



Authenticating PKS API Requests

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

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

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

Routing to the PKS API Broker VM

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

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

Refer to **Ops Manager > Pivotal Container Service > UAA > UAA URL** for your UAA domain.

See the [Configure Routing to the PKS API](#) section of the *Configure PKS API Access* topic for procedures for routing to the PKS broker VM.

Configure PKS API Access

Page last updated:

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

Configure Routing to the PKS API

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

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



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

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

Configure Access to the PKS API

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

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

4. Run `uaac token client get admin -s UAA-ADMIN-SECRET` to request a token from the UAA server. Replace `UAA-ADMIN-SECRET` with your UAA admin secret. Refer to **Ops Manager > Pivotal Container Service > Credentials > Uaa Admin Secret** to retrieve this value.
5. Grant cluster access to new or existing users with UAA. See the [Grant Cluster Access](#) section of *Managing Users in UAA* for more information.
6. Run `pks login -a UAA-URL -u USERNAME -p PASSWORD -k` to log in to the PKS CLI. Replace the `UAA-URL` with the URL of your UAA server, `USERNAME` with your username, and `PASSWORD` with your password. For example:

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

Manage Users in UAA

Page last updated:

Create and manage users in UAA with the [UAA Command Line Interface \(UAAC\)](#).

Retrieve UAA Admin Credentials

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

1. In a web browser, navigate to the fully qualified domain name (FQDN) of Ops Manager and click the **Pivotal Container Service** tile.
2. Click **Credentials**.
3. To view the UAA admin client password, click **Uaa Admin Secret**. The client username is `admin`.

With this, you will be able to create a UAA user account.

Grant Cluster Access to a User

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

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

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

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

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

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

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

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

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

For example:

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

Manage PKS Deployments with BOSH

Page last updated:

To manage your PKS deployment with BOSH, perform the following steps:

1. Gather credential and IP address information for your BOSH Director and SSH into the Ops Manager VM. See [Advanced Troubleshooting with the BOSH CLI](#) for more information.

2. Create a BOSH alias for your PKS environment. For example:

```
$ bosh alias-env pks -e 10.0.0.3 --ca-cert /var/tempest/workspaces/default/root_ca_certificate
```

3. Log in to the BOSH Director.

```
$ bosh -e pks log-in
```

4. Follow the procedures in the [Use the BOSH CLI for Troubleshooting](#) topic to manage your PKS deployment with BOSH.

Add Custom Workloads

Page last updated:

To apply custom Kubernetes workloads to every cluster created on a plan, add YAML to the tile config under **Default Cluster Apps**. Use this configuration to define what a cluster includes out of the box. For example, use custom workloads to configure metrics or logging.

Download Cluster Logs

To download cluster logs, perform the following steps:

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use the BOSH CLI v2 to log in to the BOSH Director from the Ops Manager VM. For more information, see [Advanced Troubleshooting with the BOSH CLI](#).

2. After logging in to the BOSH Director, identify the name of your PKS deployment. For example:

```
$ bosh -e pks deployments
```

Your PKS deployment name begins with `pivotal-container-service` and includes a BOSH-generated hash.

3. Identify the names of the VMs you want to retrieve logs from by listing all VMs in your deployment. For example:

```
$ bosh -e pks -d pivotal-container-service-aa1234567bc8de9f0a1c vms
```

4. Download the logs from the VM. For example:

```
$ bosh -e pks -d pivotal-container-service-aa1234567bc8de9f0a1c logs pks/0
```

See the [View Log Files](#) section of the *Diagnostic Tools* topic for information about using cluster logs to diagnose issues in your PKS deployment.

Prepare Workloads for an Upgrade

This topic describes how to manage workloads during an upgrade in Pivotal Container Service (PKS).

To prevent downtime during a PKS upgrade, configure the following settings in your deployment manifest:

- Set the number of [workload replicas](#)
- Define an [anti-affinity rule](#)

Set Workload Replicas

Upgrades run on a single VM at a time. To prevent downtime during an upgrade, Pivotal recommends running your workload on at least two worker VMs. While one worker VM runs an upgrade, additional worker VMs continue to run your workload.

To run 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:

Key-Value Pair	Description
<pre>spec: replicas: 3</pre>	Set this value to at least 2 to have at least two instances of your workload running at any time.
<pre>app: APP-NAME</pre>	Use this app name when you define the anti-affinity rule later in the spec.

Define an Anti-Affinity Rule

To distribute your workload across multiple worker VMs, you must use anti-affinity rules. See the [Kubernetes documentation](#) for more information about anti-affinity rules.

To define an anti-affinity rule, add the `spec.template.spec.affinity` section to your deployment manifest:

```
kind: Deployment
metadata:
  #...
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: APP-NAME
    spec:
      containers:
        - name: MY-APP
          image: MY-IMAGE
          ports:
            - containerPort: 12345
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: "app"
                    operator: In
                    values:
                      - APP-NAME
              topologyKey: "kubernetes.io/hostname"
```

See the following table for more information:

Key-Value Pair	Description
<pre>matchExpressions: - key: "app"</pre>	This value matches <code>spec.template.metadata.labels.app</code> .
<pre>values: - APP-NAME</pre>	This value matches the <code>APP-NAME</code> you defined earlier in the spec.

Delete PKS


To delete PKS, perform the following steps:

1. Navigate to the Ops Manager Installation Dashboard.
2. Click the trash icon on the PKS tile.
3. Click **Confirm** in the dialog box that appears.
4. By default, deleting the PKS tile will also delete all the clusters created by PKS. To preserve the clusters, click the **Delete all clusters** errand under **Pending Changes** and select **Off**.
5. Click **Apply Changes**.

Using PKS

Page last updated:

This topic describes how to use Pivotal Container Service (PKS).

 **Note:** Because PKS does not currently support the Kubernetes Service Catalog or the GCP Service Broker, binding clusters to Kubernetes services is not supported.


See the following sections:

- [Prerequisites](#)
- [Log in to PKS](#)
- [Create a Cluster](#)
- [Retrieve Cluster Credentials and Configuration](#)
- [View Cluster List](#)
- [View Cluster Details](#)
- [View Cluster Plans](#)
- [Using Dynamic Persistent Volumes](#)
- [Scale Existing Clusters](#)
- [Access the Dashboard](#)
- [Deploy and Access Basic Workloads](#)
- [Delete a Cluster](#)
- [Log Out of the PKS Environment](#)

Prerequisites for Using PKS

The procedures for using PKS have the following prerequisites:

- You must have a Pivotal Cloud Foundry (PCF) deployment with Ops Manager v2.0 or later and PKS installed.
- You must have the [PKS CLI](#) installed.
- You must have the [Kubernetes CLI](#) installed. For more information about using `kubectl`, see the [kubectl documentation](#).
- You must have an external TCP or HTTPS load balancer configured to forward traffic to the PKS API endpoint. For more information, see the Configure External Load Balancer section of *Installing and Configuring PKS on GCP or vSphere*.
- 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 [Apply Changes and Retrieve the PKS Endpoint](#) section of *Installing and Configuring PKS with NSX-T Integration*.

Log in to PKS

Page last updated:

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

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

Replace the placeholder values in the command as follows:

- `PKS_API` is the domain name you entered in **Ops Manager > Pivotal Container Service > UAA > UAA URL**. For example, `api.pks.example.com`.
- `USERNAME` and `PASSWORD` belong to the account you created in the *Grant Cluster Access to a User* step in the [Manage Users in UAA](#) topic.
- `CERT-PATH` is the path to your root CA certificate. Provide the certificate to validate the PKS API certificate with SSL. If you are logging in to a trusted environment, you can use `-k` to skip SSL verification instead of `--ca-cert CERT-PATH`.

The following example command uses SSL validation:

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

The following example command skips SSL validation:

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

Create a Cluster

Follow the steps below to create a Kubernetes cluster using the PKS CLI.

1. Locate the external hostname for accessing the Kubernetes API. Use one of the following methods, depending on your PKS installation:
 - If your PKS installation is **integrated with NSX-T**, use the NAT IP from the `ip-pool-vips` NSX IP pool. For more information, see [Enable NAT Access](#) in *Installing and Configuring PKS with NSX-T Integration*.
 - If your PKS installation is **not integrated with NSX-T**, [create an external load balancer](#) and record its IP address or hostname. Each new cluster requires its own TCP or HTTPS load balancer to allow external access. When you provide the external hostname later in this procedure, you can either use the load balancer IP address or a hostname from the domain you specified for the PKS API. For more information, see the PKS API section of *Installing and Configuring PKS on GCP or vSphere*.
2. Grant cluster access to a new or existing user in UAA. See the [Grant Cluster Access to a User](#) section of *Manage Users in UAA* for more information.
3. On the command line, run the following command to log in:

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

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

4. Run the following command to create a cluster:

```
pkcs 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` is a unique name for your cluster.
- `HOSTNAME` is the external hostname for accessing the Kubernetes API. Use the hostname you located earlier in this procedure.
- `PLAN-NAME` is the name of the plan you want to use to create the cluster.
- `WORKER-NODES` is the number of worker nodes for the cluster. For high availability, Pivotal recommends creating clusters with at least 3 worker nodes. The maximum value is 50. This flag is optional.

For example:

```
$ pkcs create-cluster my-cluster --external-hostname 10.0.0.1 --plan large --num-nodes 3
```

5. Track the cluster creation process by running `pkcs cluster CLUSTER-NAME`. Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pkcs cluster my-cluster
```

The cluster status appears in the **Last Action State** column. If the 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.

6. When cluster creation is complete, configure the external hostname.
 - a. Run `pkcs cluster CLUSTER-NAME` to view cluster details. Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pkcs cluster my-cluster
```

- b. Locate the master IP address for the cluster in the `kubernetes_master_ips` row.
- c. Configure your external TCP or HTTPS load balancer to point to the master IP address.

7. To access your cluster, run `pkcs get-credentials CLUSTER-NAME`. This command creates a local `kubeconfig` that allows you to manage the cluster. See [Retrieve Cluster Credentials and Configuration](#) for more information.
8. 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.

Retrieve Cluster Credentials and Configuration

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

The `pks get-credentials` command performs the following actions:

- Fetch the cluster's kubeconfig
- Add the cluster's kubeconfig to the existing kubeconfig
- Create a new kubeconfig, if none exists
- Switch the context to the `CLUSTER-NAME` provided

When you run `pks get-credentials CLUSTER-NAME`, PKS sets the context to the cluster you provide as the `CLUSTER-NAME`. PKS binds your username to the cluster and populates the kubeconfig file on your local workstation with cluster credentials and configuration.

The default path for your kubeconfig is `$HOME/.kube/config`.

If you access multiple clusters, you can choose to use a custom kubeconfig file for each cluster. To save cluster credentials to a custom kubeconfig, use the `KUBECONFIG` environment variable when you run `pks get-credentials`. For example:

```
$ KUBECONFIG=/path/to/my-cluster.config pks get-credentials my-cluster
```

Retrieve Cluster Credentials

Perform the following steps to populate your local kubeconfig with cluster credentials and configuration:

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

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

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

2. Run the following command:

```
pks get-credentials CLUSTER-NAME
```

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

```
$ pks get-credentials my-cluster
```

Run kubectl Commands

After PKS populates your kubeconfig, you can use the Kubernetes Command Line Interface (kubectl) to run commands against your Kubernetes clusters.

See [Installing the Kubernetes CLI](#) for information about installing kubectl.

For information about using kubectl, refer to the [Kubernetes documentation](#).

View Cluster List

Follow the steps below to view the list of deployed Kubernetes cluster with the PKS CLI.

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

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

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

2. Run the following command to view the list of deployed clusters, including cluster names and status:

```
$ pks clusters
```

View Cluster Details

Follow the steps below to view the details of an individual cluster using the PKS CLI.

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

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

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

2. Run the following command to view the details of an individual cluster:

```
pkcs cluster CLUSTER-NAME
```

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

```
$ pkcs cluster my-cluster
```

View Cluster Plans

Follow the steps below to view information about the available plans for deploying a cluster using the PKS CLI.

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

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

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

2. Run the following command to view information about the available plans for deploying a cluster:

```
$ pkcs plans
```

The response lists details about the available plans, including plan names and descriptions:

Name	ID	Description
default		Default plan for K8s cluster

Using Dynamic Persistent Volumes

When using PKS, you can choose to pre-provision persistent storage or create on-demand persistent storage volumes. Refer to the [Kubernetes documentation](#) for more information about storage management.

Perform the steps in this section to define a PersistentVolumeClaim that you can apply to newly-created pods.

1. Download the StorageClass spec for your cloud provider.

- **GCP:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-gcp.yml
```

- **vSphere:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-vsphere.yml
```

2. Apply the spec by running `kubectl create -f STORAGE-CLASS-SPEC.yml`. Replace `STORAGE-CLASS-SPEC` with the name of the file you downloaded in the previous step. For example:

```
$ kubectl create -f storage-class-gcp.yml
```

3. Run the following command to download the example PersistentVolumeClaim:

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/persistent-volume-claim.yml
```

4. Run the following command to apply the PersistentVolumeClaim:

```
$ kubectl create -f persistent-volume-claim.yml
```


- To confirm you applied the PersistentVolumeClaim, run the following command:

```
$ kubectl get pvc -o wide
```

5. To use the dynamic persistent volume, create a pod that uses the PersistentVolumeClaim. See the [pv-guestbook.yml configuration file](#) as an example.

Scale Existing Clusters

Follow the steps below to scale up an existing cluster using the PKS CLI.


 **Note:** You cannot scale the number of worker nodes down. You can only scale the number of worker nodes up.

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

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

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

2. Run the following command below to scale up your cluster. You cannot scale the number of worker nodes down.

 **Note:** This command may roll additional VMs in the cluster, affecting workloads if the worker nodes are at capacity. This issue will be resolved in a future release of PKS.

```
pks resize CLUSTER-NAME --num-nodes WORKER-NODES
```

Replace the placeholder values in the command as follows:

- o `CLUSTER-NAME` is the name of your cluster.
- o `WORKER-NODES` is the number of worker nodes for the cluster. The maximum number of worker nodes is 50. For example:

```
$ pks resize my-cluster --num-nodes 5
```

Access the Dashboard

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot containerized applications, and manage the cluster and its resources. Dashboard also provides information about the state of Kubernetes resources in the cluster.

You must have `kubectrl` credentials to access Dashboard. This requirement prevents unauthorized admin access to the Kubernetes cluster through a browser.

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

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

Deploy and Access Basic Workloads


You can deploy and access workloads in a number of ways. This PKS release focuses on `routing_mode: external` and does not include a bundled load balancing component. Select an option based on your PKS deployment:

- [No Load Balancer Abstraction Configured / vSphere without NSX-T](#)
- [Load Balancer Abstraction Configured / GCP or vSphere with NSX-T](#)
- [External Load Balancer](#)

No Load Balancer Abstraction Configured / vSphere without NSX-T

If you use vSphere without NSX-T or configuring a load balancer abstraction, follow the steps below to deploy and access basic workloads.

1. Expose the workload using a Service with `type: NodePort`.
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. Retrieve the IP address for a worker node with a running NGINX pod.


 **Note:** If you deployed more than four worker nodes, some worker nodes may not contain a running NGINX pod. Select a worker node that contains a running NGINX pod.

You can retrieve the IP address for a worker node with a running NGINX pod in one of the following ways:

- On the command line, run `kubectl get nodes`. Select a node name, then locate the node name in the vCenter or GCP Console to find the IP address.
 - On the Ops Manager command line, run `bosh vms` to find the IP address.
5. On the command line, run `kubectl get svc nginx`. Find the node port number in the `3XXXX` range.
 6. On the command line of a server with network connectivity and visibility to the IP address of the worker node, run `curl http://NODE-IP:NODE-PORT` to access the app. Replace `NODE-IP` with the IP address of the worker node, and `NODE-PORT` with the node port number.

Load Balancer Abstraction Configured / GCP or vSphere with NSX-T

If you use GCP or vSphere with NSX-T, follow the steps below to deploy and access basic workloads.

 **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`.
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 the GCP CloudProvider interacts with GCP to create 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:PORT` with the IP address of the load balancer, and `PORT` with the port number.

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`.

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.

Delete a Cluster

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

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

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

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

2. Run `pks delete-cluster CLUSTER-NAME` to delete a cluster. Replace `CLUSTER-NAME` with the unique name for your cluster. For example:

```
$ pks delete-cluster my-cluster
```

Log Out of the PKS Environment

On the command line, run `pkc logout` to log out of your PKS environment.

After logging out, you must run `pkc login` before you can run any other `pkc` commands.

Using Helm with PKS

Page last updated:

This topic describes how you can use the package manager [Helm](#) for your Kubernetes apps running on Pivotal Container Service (PKS).

Helm includes of the following components:

Component	Role	Location
<code>helm</code>	Client	Runs on your local workstation
<code>tiller</code>	Server	Runs inside your Kubernetes cluster

Helm packages are called [charts](#). Here are some examples of charts you can use:

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

This topic includes a procedure for installing [Concourse](#) using Helm. For more charts, see the Kubernetes [charts repository](#) on GitHub.

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

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

Configure Tiller

Tiller runs inside the Kubernetes cluster and requires access to the Kubernetes API. If you use role-based access control (RBAC) in PKS, perform the steps in this section to grant Tiller permission to access the API.

1. Create a service account for Tiller and bind it to the `cluster-admin` role by adding the following section to `rbac-config.yaml`:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: kube-system
```

2. Apply the service account and role by running the following command:

```
$ kubectl create -f rbac-config.yaml
```

3. Download and install the [Helm CLI](#).

4. Deploy Helm using the service account by running the following command:

```
$ helm init --service-account tiller
```

5. Run `helm ls` to verify that the permissions are configured.

To apply more granular permissions to the Tiller service account, see the [Helm RBAC](#) documentation.

Install Concourse Using Helm

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

1. Download the StorageClass spec for your cloud provider.

- **GCP:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-gcp.yml
```

- **vSphere:**

```
$ wget https://raw.githubusercontent.com/cloudfoundry-incubator/kubo-ci/master/specs/storage-class-vsphere.yml
```

2. Apply the spec by running `kubectl create -f STORAGE-CLASS-SPEC.yml`. Replace `STORAGE-CLASS-SPEC` with the name of the file you downloaded in the previous step. For example:

```
$ kubectl create -f storage-class-gcp.yml
```

3. Install the Concourse Helm chart by running `helm install stable/concourse` with the following options:
 - `--name APP-NAME` : (Optional) Replace `APP-NAME` with a name you provide for the installed chart.
 - `--set persistence.worker.storageClass=STORAGE-CLASS` : Replace `STORAGE-CLASS` with your StorageClass to apply the spec to the Concourse worker persistent volumes.
 - `--set postgresql.persistence.storageClass=STORAGE-CLASS` : Replace `STORAGE-CLASS` with your StorageClass to apply the spec to the PostgreSQL database persistent volumes.

For example:

```
$ helm install --name my-concourse --set persistence.worker.storageClass=ci-storage,postgresql.persistence.storageClass=ci-storage stable/concourse
```

4. Forward the port number so that you can access Concourse from localhost. By default, the Concourse chart does not expose services outside the cluster.

- a. Export the pod name as an environment variable. For example:

```
$ export POD_NAME=$(kubectl get pods --namespace default -l "app=concourse-web" -o jsonpath="{.items[0].metadata.name}")
```

- b. Forward the port number by running the following command:

```
$ kubectl port-forward --namespace default $POD_NAME 8080:8080
```

5. Navigate to `http://127.0.0.1:8080` in your browser to access Concourse. Use the default credentials to log in.
6. Log in to your Concourse instance from the command line by running `fly -t MY-CONCOURSE login -c http://127.0.0.1:8080`. For example:

```
$ fly -t ci-helm login -c http://127.0.0.1:8080
```

For more configuration options, see the [Concourse Helm chart](#) documentation.

Diagnosing and Troubleshooting PKS

This topic is intended to provide assistance when diagnosing and troubleshooting issues installing or using Pivotal Container Service (PKS).

See the following sections:

- [Diagnostic Tools](#)
- [Troubleshooting](#)

Diagnostic Tools

Verify PKS CLI Version

The Pivotal Container Service (PKS) CLI interacts with your PKS deployment through the PKS API endpoint. You create, manage, and delete Kubernetes clusters on your PKS deployment by entering commands in the PKS CLI. The PKS CLI is under active development and commands may change between versions.

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

```
$ pkcs --version
PKS CLI version: 1.0.0-build.3
```

View Log Files

Log files contain error messages and other information you can use to diagnose issues with your PKS deployment. Follow the steps below to access PKS log files.

1. Gather credential and IP address information for your BOSH Director, SSH into the Ops Manager VM, and use 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. 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 `kubernetes-api` log file.
 - On a PKS worker VM, examine the `kubelet` log file.

Troubleshooting

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 Dashboard](#) in a browser or using the `kubectl` command-line tool.
- [Heapster](#) does not start.
- Pods cannot resolve DNS names, and error messages report the service `kube-dns` is invalid. If `kube-dns` is not deployed, the cluster typically fails to start.

Explanation

The Kubernetes features and functions listed above are provided by the following PKS add-ons:

- **Kubernetes Dashboard** `kubernetes-dashboard`
- **Heapster:** `heapster`
- **DNS Resolution:** `kube-dns`

To enable these add-ons, Ops Manager must run scripts after deploying PKS. You must configure Ops Manager to automatically run these post-deploy scripts.

Solution

Perform the following steps to configure Ops Manager to run post-deploy scripts to deploy the missing add-ons to your cluster.

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Click the Ops Manager v2.0 tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.



Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.

5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pkcs delete-cluster` on the command line to delete the cluster. For more information, see the [Delete Cluster](#) section of *Using PKS*.
9. On the command line, enter `pkcs create-cluster` to recreate the cluster. For more information, see the [Create Cluster](#) section of the *Using PKS*.

Error: Failed Jobs

Symptom

In stdout or log files, you see an error message referencing `post-start scripts failed` or `Failed Jobs`.

Explanation

After deploying PKS, Ops Manager runs scripts to start a number of jobs. You must configure Ops Manager to automatically run these post-deploy scripts.

Solution

Perform the following steps to configure Ops Manager to run post-deploy scripts.

1. Navigate to `https://YOUR-OPS-MANAGER-FQDN/` in a browser to log in to the Ops Manager Installation Dashboard.
2. Click the Ops Manager v2.0 tile.
3. Select **Director Config**.
4. Select **Enable Post Deploy Scripts**.



Note: This setting enables post-deploy scripts for all tiles in your Ops Manager installation.

5. Click **Save**.
6. Click the **Installation Dashboard** link to return to the Installation Dashboard.
7. Click **Apply Changes**.
8. After Ops Manager finishes applying changes, enter `pkcs delete-cluster` on the command line to delete the cluster. For more information, see the [Delete Cluster](#) section of *Using PKS*.
9. On the command line, enter `pkcs create-cluster` to recreate the cluster. For more information, see the [Create Cluster](#) section of the *Using PKS*.

Error: No Such Host

Symptom

In stdout or log files, you see an error message that includes `lookup vm-WORKER-NODE-GUID on IP-ADDRESS: no such host`.

Explanation

This error occurs on GCP when the Ops Manager Director tile uses 8.8.8.8 as the DNS server. When this IP range is in use, the master node cannot locate the route to the worker nodes.

Solution

Use the Google internal DNS range, 169.254.169.254, as the DNS server.

PKS CLI

Page last updated:

This topic describes how to use the Pivotal Container Service Command Line Interface (PKS CLI) to interact with the PKS API.

The [PKS CLI](#) is used to create, manage, and delete Kubernetes clusters. To deploy workloads to a Kubernetes cluster created using the PKS CLI, use the Kubernetes CLI, [kubectl](#).

Current Version: 1.0.0-build3

See the following sections:

- [pks cluster](#): View cluster details
- [pks clusters](#): View all clusters
- [pks create-cluster](#): Create a Kubernetes cluster
- [pks delete-cluster](#): Delete a Kubernetes cluster
- [pks get-credentials](#): Connect to a cluster and use kubectl
- [pks login](#): Log in to PKS
- [pks logout](#): Log out of the PKS API
- [pks plans](#): View available preconfigured plans
- [pks resize](#): Increase the number of worker nodes for a cluster

pks cluster

View the details of the cluster

Synopsis

Run this command to see details of your cluster such as name, host, port, ID, number of worker nodes, last operation, etc.

```
pks cluster [flags]
```

Examples

```
pks cluster my-cluster
```

Options

```
-h, --help  help for cluster  
--json     Return the PKS-API output as json
```


pks clusters

Show all clusters created with PKS

Synopsis

This command describes the clusters created via PKS, and the last action taken on the cluster

```
pks clusters [flags]
```

Examples

```
pks clusters
```

Options

```
-h, --help  help for clusters
--json      Return the PKS-API output as json
```

pks create-cluster

Creates a kubernetes cluster, requires cluster name and an external host name

Synopsis

Create-cluster requires a cluster name, as well as an external hostname. External hostname can be a loadbalancer, from which you access your kubernetes API (aka, your cluster control plane)

```
pks create-cluster <cluster-name> [flags]
```

Examples

```
pks create-cluster my-cluster --external-hostname example.hostname --plan production
```

Options

```
-e, --external-hostname string  Address from which to access Kubernetes API
-h, --help                    help for create-cluster
--json                        Return the PKS-API output as json
-n, --num-nodes string         Number of worker nodes
-p, --plan string              Preconfigured plans. Run pks list-plans for more details
```

pks delete-cluster

Deletes a kubernetes cluster, requires cluster name

Synopsis

Delete-cluster requires a cluster name.

```
pks delete-cluster <cluster-name> [flags]
```

Examples

```
pks delete-cluster my-cluster
```

Options

```
-h, --help  help for delete-cluster
```

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 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 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
```

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

```
pks resize <cluster-name> [flags]
```

Examples

```
pks resize my-cluster --num-nodes 5
```

Options

```
-h, --help           help for resize  
-n, --num-nodes int32 Number of worker nodes (default 1)
```


PKS Release Notes

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

v1.0.1

Release Date: March 6, 2018

- **[Bug Fix]** vSphere NSX-T integration works with BOSH stemcell v3468.25 and later.
- **[Bug Fix]** Adds support for special characters `#`, `&`, `;`, `"`, `'`, `^`, `\`, space (), and `%` in vCenter passwords.
- **[Bug Fix]** Worker nodes are drained before they stop in order to minimize workload downtime during a rolling upgrade.
- **[Security Fix]** UAA credentials and vCenter passwords do not appear in BOSH logs.
- **[Bug Fix]** BOSH DNS no longer causes worker nodes to fail after a manual restart.
- Updates Kubernetes to v1.9.3.
- Updates Golang to v1.9.4.

v1.0.0

Release Date: February 8, 2018

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

Product Component	Version Supported	Notes
vSphere	6.5 and 6.5 U1 - Editions <ul style="list-style-type: none"> • vSphere Enterprise Plus Edition • vSphere with Operations Management Enterprise Plus 	vSphere versions supported for Pivotal Container Service (PKS)
VMware Harbor Registry	1.4.1	Separate download available from Pivotal Network
NSX-T	2.1 Advanced Edition	Available from VMware
Pivotal Cloud Foundry Operations Manager (Ops Manager)	2.0.X	Separate download available from Pivotal Network
Stemcell	3468.21	Separate download available from Pivotal Network
Kubernetes	1.9.2	Packaged in the PKS Tile (CFCR)
CFCR (Kubo)	0.13	Packaged in the PKS Tile
NCP	2.1.0.1	Packaged in the PKS Tile
PKS CLI	1.0.0-build.3	Separate download available from the PKS section of Pivotal Network
Kubernetes CLI	1.9.2	Separate download available from the PKS section of Pivotal Network
* Components marked with an asterisk have been patched to resolve security vulnerabilities or fix component behavior.		

Features

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

- Centralized logging through syslog

Known Issues

Special Characters

In PKS v1.0.0, special characters, such as #, &, :, ", ', ^, \, space (), !, and % cannot be used in vCenter passwords. PKS v1.0.1 adds support for the special characters listed above except for !.

Stemcell Incompatibility

When deploying the PKS v1.0.0 using NSX-T as the networking layer with a stemcell other than x3468.21, Kubernetes cluster deployments fail. PKS v1.0.1 adds support for stemcells v3468.25 and later.

Stemcell Updates Cause Automatic VM Rolling

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

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

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

Upgrade Errand Fails with Failed Deployments

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

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

Syslog Security Recommendations

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