

1 PKS CheatSheet

KUBERNETES

- PDF Link: [cheatsheet-pks-A4.pdf](#), Category: kubernetes
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-pks-A4>
- Related posts: Bosh CheatSheet, OpenShift CheatSheet, #denny-cheatsheets

File me Issues or star this repo.

1.1 PKS Reference

Name	Command
YouTube	YouTube: PKS overview, YouTube: Pivotal Container Service (PKS) Demos and Webcasts
Reference	PKS Documentation, PKS Concepts, PKS 1.2 Release Notes, Managing PKS 1.2, PKS personal blog
Reference	Bosh CheatSheet, Tile CheatSheet, UAA CheatSheet
Reference	OpenShift CheatSheet, Rancher CheatSheet, PKS CheatSheet
Reference	Link: Get bosh/pks tile credential from om

<https://raw.githubusercontent.com/dennyzhang/cheatsheet-pks-A4/master/pks-highlevel.png>

1.2 PKS Strengths

Name	Command
Kubernetes Federation	Multiple clusters on-demand. Not only one kubernetes cluster for your infra
End-to-end integration	Monitoring and logging works out of box
VM LCM: auto healing	VM health check and auto-replacement
Less vendor lock-in	Vanilla Kubernetes; Any infra; Any OS
Networking with NSX-T	Advanced CNI
Image registry & security	Image sign, audit, replication; vulnerabilities scan

1.3 Key Components

Name	Command
Computing virtualization	Link: VMware vSphere Documentation
NCP: CNI for Vmware NSX-T	Help to create pod networks in PKS k8s clusters
CFCR/Kubo	Deploy/manage fleets of k8s clusters. GitHub: kubo-release, cfc-etc-release
Bosh: VM lifecycle management	Link: BOSH CHEATSHEET, Github: Bosh
Packaging: tile, ops manager	CheatSheet: Cloudfoundry Tile & OpsManager
CSI for persistent volume	GitHub: hatchway

1.4 VMware Product Integrations

Name	Command
wavefront	YouTube: Pivotal Container Service (PKS) and VMware Wavefront
log insight	YouTube: PKS and VMware vRealize Log Insight
vrops	YouTube: VMware vRealize Operations
vcd (VMware vCloud Director)	YouTube: VMware vCloud Director Overview

1.5 PKS Scenarios

Name	Summary
How to run pks cli commands	run-pks-cli.md
How to run bosh cli commands	run-bosh-cli.md
How to run kubectl command	run-kubectl.md
How airgap integration tests are enforced	For each node, load specific iptable rules
How PKS supports k8s master HA	GitHub: kubo-release, GitHub: cfc-etc-release
Workflow of how PKS creates a k8s cluster	Link: Create a Kubernetes Cluster
Deployment for env without internet	Enforce air-gapped integration deployment tests

1.6 What PKS Adds to Kubernetes

Name	Summary
Secure multi-tenant ingress	NSX-T
Secure container registry	VMware Harbor
Rolling upgrades to cluster infrastructure	IaaS: bosh VM upgrade
Cluster provisioning and scaling	IaaS: VM lifecycle management
Monitoring and recovery of cluster VMs and processes	IaaS: VM lifecycle management
Embedded, hardened operating system	Linux release for OS hardening
Log sink	K8S Namespace multi-tenancy

1.7 PKS Challenges

Name	Command
Faster for typical use cases	Create k8s clusters, resize k8s cluster, create pods, etc
Tile & OpsManager is not agile	It slows down everything. The development, testing and deployment.
Extend PKS API layer	Easy to add more functionalities for PKS admins
UX of PKS CLI	The usage of pks cli could be more intuitive
Improve PKS control panel HA	Online rolling upgrade for opsmanager, uaa, pks api, etc
Better storage support of PV	HA for PV, and support more CSI providers
Cleanup for stale resources	When operations have failed, need to do the cleanup in a safe way
More built-in security supports	PKS supports most common security enhancements, but it doesn't provides them

1.8 PKS cli

Name	Command
Check cli version	<code>pks --version</code>
List all pks clusters	<code>pks clusters</code>
Create cluster	<code>pks create-cluster <cluster-name1> -e myk8s1.pks.com -p "plan 1" --num-nodes 1</code>
Delete cluster	<code>pks delete-cluster <cluster-name1></code>
Check cluster status	<code>pks cluster cluster1</code>
Get cluster kubectl credential	<code>pks get-credentials <cluster_name></code>
Set kubectl context	<code>kubectl config use-context <cluster_name></code>
List all available plans	<code>pks plans</code>
pks login	<code>pks login -a <api.test.com> -u <username> -p <passwd> -k</code>
pks login	<code>pks login -k --username <username> --password <password> --api <myk8s1.test.com></code>
Default roles in pks UAA	<code>pks.clusters.manage</code> , <code>pks.clusters.admin</code> , Link: Manage Users in UAA

1.9 Deployment Diagram

Name	Summary
Bosh director vm	VM manager
Ops manager vm	Package manager
PKS API server vm	See below
Build-in process in k8s master vm	See below
Build-in process in k8s worker vm	See below

1.10 PKS footprint: in control panel

Name	Summary
Get process list in pks 1.2.0	ssh to the pks api vm, then <code>sudo monit summary</code>
pks-api	
uua	
broker	
pks-nsx-t-osb-proxy	
mariadb_ctrl	
galera-healthcheck	
gra-log-purger-executable	
cluster_health_logger	
telemetry	
event-emitter	
bosh-dns	
bosh-dns-resolvconf	
bosh-dns-healthcheck	

1.11 PKS footprint: in k8s master vms

Name	Summary
Get process list in pks 1.2.0	ssh to k8s master vm, then <code>sudo monit summary</code>
kube-apiserver	
kube-controller-manager	
kube-scheduler	
etcd	
blackbox	
nep	
bosh-dns	
bosh-dns-resolvconf	
bosh-dns-healthcheck	
pks-helpers-bosh-dns-resolvconf	

1.12 PKS footprint: in k8s worker vms

Name	Summary
Get process list in pks 1.2.0	ssh to the k8s worker vm, then <code>sudo monit summary</code>
docker	
kubelet	
kube-proxy	
blackbox	
nsx-node-agent	
ovs-vswitchd	open vSwitch
ovsdb-server	vSwitch database
nsx-kube-proxy	
bosh-dns	
bosh-dns-resolvconf	
bosh-dns-healthcheck	
pks-helpers-bosh-dns-resolvconf	

1.13 PKS errands & tasks

Name	Command
kubeconfig	Github: kubo-deployment/bin/set_kubeconfig
apply-specs	<code>bosh -d cfcf run-errand apply-specs</code>

1.14 PKS Troubleshooting

Name	Command
Log files in pks vms	<code>/var/vcap/sys/log</code>
Reference	Link: PKS Troubleshoot

1.15 Deployment with NSX-T + NAT

<https://raw.githubusercontent.com/dennyzhang/cheatsheet-pks-A4/master/pks-nsxt-nat.png>

1.16 Deployment with NSX-T + No-NAT + vswitch

<https://raw.githubusercontent.com/dennyzhang/cheatsheet-pks-A4/master/pks-nsxt-no-nat-virtual-switch.png>

1.17 Deployment with NSX-T + No-NAT + logical switch

<https://raw.githubusercontent.com/dennyzhang/cheatsheet-pks-A4/master/pks-nsxt-no-nat-logical-switch.png>

1.18 PKS CLI Online Help

```
[ec2-user@ip-172-31-33-176 ~]$ pks --help
```

The Pivotal Container Service (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`.

Version: 1.1.1-build.8

Usage:

`pks [command]`

Available Commands:

<code>cluster</code>	View the details of the cluster
<code>clusters</code>	Show all clusters created with PKS
<code>create-cluster</code>	Creates a kubernetes cluster, requires cluster name, an external host name, and plan
<code>delete-cluster</code>	Deletes a kubernetes cluster, requires cluster name
<code>get-credentials</code>	Allows you to connect to a cluster and use kubectl
<code>help</code>	Help about any command
<code>login</code>	Log in to PKS
<code>logout</code>	Log out of PKS
<code>plans</code>	View the preconfigured plans available
<code>resize</code>	Increases the number of worker nodes for a cluster

Flags:

<code>-h, --help</code>	help for pks
<code>--version</code>	version for pks

Use "`pks [command] --help`" for more information about a command.

1.19 More Resources

<https://docs.pivotal.io/runtimes/pks/1-2/index.html>

License: Code is licensed under MIT License.