

1 PKS CheatSheet

CLOUD

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

File me Issues or star this repo.

1.1 PKS Reference

Name	Summary
YouTube	YouTube: PKS Demos and Webcasts, YouTube: PKS overview
Reference	PKS Documentation, PKS Concepts, PKS 1.2 Release Notes, Managing PKS 1.2, PKS personal blog
Reference	CheatSheet: IT Infrastructure Fundamentals, VMware CheatSheet
Reference	OpenShift CheatSheet, Rancher CheatSheet
Reference	Bosh CheatSheet, Tile CheatSheet, UAA CheatSheet, CheatSheet: VMware Products

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

1.2 PKS Key Components

Name	Summary
Computing virtualization	Link: VMware vSphere Documentation
SDN networking	NCP: CNF for VMware NSX-T
VM/Cluster lifecycle management	Link: BOSH CHEATSHEET, Github: Bosh
Node healing	Link: BOSH CHEATSHEET
Container optimized OS	CloudFoundry Stemcell
Container runtime	dockerd
Container Image Compliance Scan	Harbor Clair
Docker image registry	VMware Harbor
Packages k8s cluster orchestrator	CFCR/Kubo, GitHub: kubo-release, cfc-etc-release
Component Packaging	CheatSheet: Cloudfoundry Tile & OpsManager
CSI for persistent volume	GitHub: hatchway
Reference	Link: OpenShift Key Components, Link: PKS Key Components
Reference	PKS CheatSheet, Bosh CheatSheet, Tile CheatSheet, UAA CheatSheet

1.3 VMware Product Integrations

Name	Summary
wavefront	YouTube: 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

1.4 PKS Assumptions

Name	Summary
Multiple instances of K8S	One PKS Multiple K8S clusters. Use k8s clusters or k8s namespace for multi-tenancy
No mixed versions	Versions must match between master and node hosts, excluding upgrade
Support customization mainly at PKS level	Not at k8s clusters or k8s namespace level
No hybrid cloud providers	Support multi-clouds, but only can't mix
Don't run user workload in k8s master VMs	Avoid messing up k8s control plane
Node roles	Master nodes, worker nodes, and etcd nodes.

1.5 PKS Scenarios

Name	Summary
How to run pks cli commands	run-pks-cli.md
How to run bosh cli commands, like bosh ssh	run-bosh-cli.md
How to run kubectl command	run-kubectl-in-pks.md
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
How airgap integration tests are enforced	For each node, load specific iptable rules

1.6 OpenShift Maintenance

Name	Command
Install clusters	
Upgrade clusters	Mainly in-place upgrade. Upgrade from OpsManager UI.
Node problem healing	

1.7 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

1.8 PKS Challenges

Name	Summary
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

1.9 PKS Strengths

Name	Summary
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

1.10 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-name> -e <subdomain>.pks.local -p "plan 1" -n 1</code>
Delete cluster	<code>pks delete-cluster <cluster-name></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

1.11 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

1.12 PKS footprint: in control panel

- Get process list in pks 1.2.0: ssh to the pks api vm, then `sudo monit summary`

Name	Memory (RES)
<code>pks-api</code>	1 GB
<code>uaa</code>	500 MB
<code>mysqld</code>	500 MB
<code>pks-nsx-t-osb-proxy</code>	25 MB
<code>telemetry</code>	25 MB
<code>bosh-agent</code>	17 MB
<code>bosh-dns</code>	16 MB
<code>on-demand-service-broker</code>	16 MB
<code>event-emitter</code>	10 MB
<code>galera-healthcheck</code>	7 MB
<code>bosh-dns-healthcheck</code>	6 MB
<code>cf-mysql-cluster-health-logger</code>	6 MB
<code>gra-log-purger-executable</code>	2 MB

1.13 PKS footprint: in k8s master vms

- Get process list in pks 1.2.0: ssh to k8s master vm, then `sudo monit summary`

Name	Summary
<code>blackbox syslog</code>	530 MB
<code>kube-apiserver</code>	520 MB
<code>etcd</code>	120 MB
<code>fluentd</code>	100 MB
<code>kube-controller-manager</code>	100 MB
<code>ncp</code>	70 MB
<code>kube-scheduler</code>	35 MB
<code>bosh-dns</code>	19 MB
<code>bosh-agent</code>	15 MB
<code>bosh-dns-nameserverconfig</code>	5 MB
<code>bosh-dns-health</code>	10 MB

1.14 PKS footprint: in k8s worker vms

- Get process list in pks 1.2.0: ssh to the k8s worker vm, then `sudo monit summary`

Name	Summary
fluentd	180 MB
kubelet	100 MB
cadvisor	85 MB
docker	70 MB
blackbox syslog	60 MB
metrics-server	36 MB
ovs-vswitchd open vSwitch	35 MB
kube-proxy	30 MB
bosh-dns	20 MB
bosh-agent	18 MB
bosh-dns-health	7 MB
bosh-dns-nameserver	5 MB
ovsdb-server vSwitch database	5 MB
nsx-node-agent	3 MB
nsx_kube_proxy	3 MB

1.15 PKS errands & tasks

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

1.16 PKS Troubleshooting

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

1.17 Deployment with NSX-T + NAT

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

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

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

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

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

1.20 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:

```
cluster      View the details of the cluster
clusters     Show all clusters created with PKS
create-cluster Creates a kubernetes cluster, requires cluster name, an external host name, and plan
delete-cluster Deletes a kubernetes cluster, requires cluster name
get-credentials Allows you to connect to a cluster and use kubectl
help         Help about any command
```

login	Log in to PKS
logout	Log out of PKS
plans	View the preconfigured plans available
resize	Increases the number of worker nodes for a cluster

Flags:

-h, --help	help for pks
--version	version for pks

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

1.21 More Resources

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

License: Code is licensed under MIT License.