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	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
Container runtime	dockerd
SDN networking	NCP: CNI for Vmware NSX-T
VM/Cluster lifecycle management	Link: BOSH CHEATSHEET, Github: Bosh
CSI for persistent volume	GitHub: hatchway
Docker image registry	VMware Harbor
Packaging: tile, ops manager	CheatSheet: Cloudfoundry Tile & OpsManager
Packages k8s cluster orchestrator	CFCR/Kubo, GitHub: kubo-release, cfcr-etcd-release
Container optimized OS	CloudFoundry Stemcell
Reference	Link: OpenShift Key Components, Link: PKS Key Components

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 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-in-pks.md
How PKS supports k8s master HA	GitHub: kubo-release, GitHub: cfcr-etcd-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

1.5 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

1.6 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

1.7 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

1.8 PKS cli

Name	Command
Check cli version	`pks --version`
List all pks clusters	`pks clusters`
Create cluster	`pks create-cluster <cluster-name> -e <subdomain>.pks.local -p "plan 1" -n 1`
Delete cluster	`pks delete-cluster <cluster-name>`
Check cluster status	`pks cluster cluster1`
Get cluster kubectl credential	`pks get-credentials <cluster_name>`
Set kubectl context	`kubectl config use-context <cluster_name>`
List all available plans	`pks plans`
pks login	`pks login -a <api.test.com> -u <username> -p <passwd> -k`
pks login	`pks login -k --username <username> --password <password> --api <myk8s1.test.com>`
Default roles in pks UAA	`pks.clusters.manage`, `pks.clusters.admin`, 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

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

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

1.11 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
blackbox syslog	530 MB
kube-apiserver	520 MB
etcd	120 MB
fluentd	100 MB
kube-controller-manager	100 MB
ncp	70 MB
kube-scheduler	35 MB
bosh-dns	19 MB
bosh-agent	15 MB
bosh-dns-nameserverconfig	5 MB
bosh-dns-health	10 MB

1.12 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-namesever	5 MB
ovsdb-server vSwitch database	5 MB
nsx-node-agent	3 MB
nsx_kube_proxy	3 MB

1.13 PKS errands & tasks

Name	Command
kubeconfig	`Github: kubo-deployment/bin/set_kubeconfig`
apply-specs	`bosh -d cfcr run-errand apply-specs`

1.14 PKS Troubleshooting

Name	Summary
Log files in pks vms	`/var/vcap/sys/log`
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:
  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.
```

1.19 More Resources

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