

- PDF Link: [cheatsheet-kubernetes-A4.pdf](#)
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-kubernetes>
- Category: kubernetes

File me Issues or star this repo.

See more CheatSheets from Denny: [#denny-cheatsheets](#)

## 1.1 Common Usage

### 1.1.1 yaml templates

Name	Summary
Pod yaml examples	pod-dummy.yaml, pod-nginx.yaml, pod-healthcheck-nginx.yaml
Pod yaml examples	pod-volume-redis.yaml, pod-gitclone.yaml, pod-handlers.yaml
Deployment yaml examples	deployment-nginx.yaml
Service yaml examples	service-clusterip-nginx.yaml
Volume yaml examples	pod-volume-empty-redis.yaml
Statefulset yaml examples	statefulset-nginx.yaml, statefulset-single-mysql
Statefulset yaml examples	statefulset-replicated-mysql
Serviceaccount yaml examples	serviceaccount-default.yaml

### 1.1.2 Common Commands

Name	Command
Validate yaml file with dry run	<code>kubect1 create --dry-run --validate -f pod-dummy.yaml</code>
Run wget test temporarily	<code>kubect1 run --rm mytest --image=busybox -it</code>
Run curl test temporarily	<code>kubect1 run --rm mytest --image=yauritux/busybox-curl -it</code>
Open a bash terminal in a pod	<code>kubect1 exec -it storage sh</code>
Check pod environment variables	<code>kubect1 exec redis-master-ft9ex env</code>
Get all services	<code>kubect1 get service --all-namespaces</code>
Get system conf via configmap	<code>kubect1 -n kube-system get cm kubeadm-config -o yaml</code>
Query healthcheck endpoint	<code>curl -L http://127.0.0.1:10250/healthz</code>

## 1.2 Components & Services

- Services on Master Nodes

Name	Summary
kube-apiserver	exposes the Kubernetes API from master nodes
etcd	reliable data store for all k8s cluster data
kube-scheduler	schedule pods to run on selected nodes
kube-controller-manager	node controller, replication controller, endpoints controller, and service account & token controllers

- Services on Worker Nodes

Name	Summary
kubelet	makes sure that containers are running in a pod
kube-proxy	perform connection forwarding
Container Runtime	Kubernetes supported runtimes: Docker, rkt, runc and any OCI runtime-spec implementation.

- Addons: pods and services that implement cluster features

Name	Summary
DNS	serves DNS records for Kubernetes services
Web UI	a general purpose, web-based UI for Kubernetes clusters
Container Resource Monitoring	collect, store and serve container metrics
Cluster-level Logging	save container logs to a central log store with search/browsing interface

- Tools

Name	Summary
kubect1	the command line util to talk to k8s cluster
kubeadm	the command to bootstrap the cluster
kubefed	the command line to control a Kubernetes Cluster Federation

link: [Kubernetes Components](#)

### 1.2.1 Check Performance

Name	Command
Get node resource usage	<code>kubectl top node</code>
Get node resource usage	<code>kubectl top pod</code>
List resource utilization for all containers	<code>kubectl top pod --all-namespaces --containers=true</code>

### 1.3 Pod

Name	Command
List all pods	<code>kubectl get pods</code>
List pods for all namespace	<code>kubectl get pods -all-namespaces</code>
List all critical pods	<code>kubectl get -n kube-system pods -a</code>
List pods with more info	<code>kubectl get pod -o wide</code> , <code>kubectl get pod -o yaml</code>
Get pod info	<code>kubectl describe pod srv-mysql-server</code>
List all pods with labels	<code>kubectl get pods --show-labels</code>
Get Pod initContainer status	<code>kubectl get pod --template '{{.status.initContainerStatuses}}' &lt;pod-name&gt;</code>
Get pod by selector	<code>podname=\$(kubectl get pods -n \$namespace --selector="app=syslog" -o jsonpath='{.items[*].metadata.name}')</code>
List pods with docker images	<code>kubectl get pods -o=jsonpath='{range .items[*]}{.metadata.name}:{.spec.containers[0].name}{ " "}{.spec.containers[0].image}\n'}</code>
kubectl run command	<code>kubectl exec -it -n "\$namespace" "\$podname" - sh -c "echo \$msg »/dev/termination-log"</code>

- Delete Pod

Name	Command
Delete pod	kubectl delete pod hello-node-95913-n63qs
Delete pod by label	kubectl delete pod -l env=test

## 1.4 Label & Annotation

Name	Command
Filter pods by label	<code>kubectl get pods -l owner=denny</code>
Manually add label to a pod	<code>kubectl label pods dummy-input owner=denny</code>
Remove label	<code>kubectl label pods dummy-input owner-</code>
Manually add annotation to a pod	<code>kubectl annotate pods dummy-input my-url=https://www.dennyzhang.com</code>

## 1.5 Deployment & Scale

link: [Pausing and Resuming a Deployment](#)

Name	Command
Scale out	<code>kubectl scale --replicas=3 deployment/nginx-app</code>
online rolling upgrade	<code>kubectl rollout app-v1 app-v2 --image=img:v2</code>
Roll backup	<code>kubectl rollout app-v1 app-v2 --rollback</code>
List rollout	<code>kubectl get rs</code>
Check update status	<code>kubectl rollout status deployment/nginx-app</code>
Check update history	<code>kubectl rollout history deployment/nginx-app</code>
Pause/Resume	<code>kubectl rollout pause deployment/nginx-deployment, resume</code>
Rollback to previous version	<code>kubectl rollout undo deployment/nginx-deployment</code>

## 1.6 Service

Name	Command
List all services	<code>kubectl get services</code>
Get service detail	<code>kubectl get service nginx-service -o yaml</code>
Get service cluster ip	<code>kubectl get service nginx-service -o go-template='{{.spec.clusterIP}}'</code>
Get service cluster port	<code>kubectl get service nginx-service -o go-template='{{(index .spec.ports 0).port}}'</code>

## 1.7 StatefulSet

Name	Command
List statefulset	<code>kubectl get sts</code>
Scale statefulset	<code>kubectl scale sts &lt;stateful_set_name&gt; --replicas=5</code>
Delete statefulset only (not pods)	<code>kubectl delete sts &lt;stateful_set_name&gt; --cascade=false</code>

## 1.8 Volumes & Volume Claims

Name	Command
Check the mounted volumes	<code>kubectl exec storage ls /data</code>
Check persist volume	<code>kubectl describe pv pv0001</code>

## 1.9 Other Components

### 1.9.1 Log files

Name	Command
API Server.log= in master node	<code>/var.log=/kube-apiserver.log</code>
Scheduler.log= in master node	<code>/var.log=/kube-scheduler.log</code>
Controller.log= in master node	<code>/var.log=/kube-controller-manager.log</code>
Kubelet.log= in worker node	<code>/var.log=/kubelet.log</code>
Kube Proxy.log= in worker node	<code>/var.log=/kubelet-proxy.log</code>

### 1.9.2 Events & Metrics

Name	Command
View all events	<code>kubectl get events --all-namespaces</code>

### 1.9.3 Namespace & Security

Name	Command
List authenticated contexts	<code>kubectl config get-contexts</code>
List contexts	<code>kubectl config get-contexts</code>
Switch context	<code>kubectl config use-context &lt;cluster-name&gt;</code>
List all namespaces defined	<code>kubectl get namespaces</code>
kubectl config file	<code>~/.kube/config</code>

### 1.9.4 Network

Name	Command
Temporarily add a port-forwarding	<code>kubectl port-forward redis-izl09 6379</code>

### 1.9.5 Endpoint

Name	Command
List endpoints	<code>kubectl get endpoints</code>

## 1.10 Basic

### 1.10.1 Key Concepts

Name	Summary
CNCF	Cloud Native Computing Foundation
CRI	Container Runtime Interface
CNI	Container Network Interface
CSI	Container Storage Interface

### 1.10.2 Kubernetes Critical Files

Name	Comment
Config folder	<code>/etc/kubernetes/</code>
Certificate files	<code>/etc/kubernetes/pki/</code>
Credentials to API server	<code>/etc/kubernetes/kubelet.conf</code>
Superuser credentials	<code>/etc/kubernetes/admin.conf</code>
Kubernetes working dir	<code>/var/lib/kubelet/</code>
Docker working dir	<code>/var/lib/docker/</code>
Etd working dir	<code>/var/lib/etcd/</code>
Network cni	<code>/etc/cni/net.d/</code>
Docker container log	<code>/var/log/containers/</code>
Log files	<code>/var/log/pods/</code>
Env	<code>export KUBECONFIG=/etc/kubernetes/admin.conf</code>
Env	<code>/etc/systemd/system/kubelet.service.d/10-kubeadm.conf</code>

### 1.10.3 Check status

Name	Summary
Start a service	<code>kubectl run hello --image=my_img --port=80</code>
Similar to <code>docker ps</code>	<code>kubectl get nodes</code>
Similar to <code>docker inspect</code>	<code>kubectl describe pod nginx-app-413181-cn</code>
Similar to <code>docker logs</code>	<code>kubectl logs</code>
Similar to <code>docker exec</code>	<code>kubectl exec</code>
Get deployment info	<code>kubectl get deploy</code>
Delete service	<code>kubectl delete service nginxservice</code>
Get kubectl version	<code>kubectl version</code>
Get cluster info	<code>kubectl cluster-info</code>
Get configuration	<code>kubectl config view</code>
Get component status	<code>kubectl get componentstatus</code>
Get node status	<code>kubectl describe node \$node_name</code>
Get services for current namespace	<code>kubectl get svc</code>

### 1.10.4 Kubernetes Developer Resources

Name	Summary
API Conventions	<a href="#">link: API Conventions</a>

## 1.11 Minikube

[link: minikube in GitHub](#)

Name	Command
Start minikube env	<code>minikube start</code>
minikube docker-env	<code>eval \$(minikube docker-env)</code>
Get dashboard	<code>minikube dashboard</code>
ssh to minikube vm	<code>minikube ssh</code>
Get ip	<code>minikube ip</code>
Get cluster info	<code>kubectl cluster-info</code>
List addons	<code>minikube addons list</code>
Get service info	<code>minikube service \$srv_name</code>

TODO: rolling-update command is imperative, better use Deployments rollout. It's declarative.

### 1.12 Misc scripts

- Tail pod log by label

```
namespace="mynamespace"
mylabel="app=mylabel"
kubectl get pod -l "$mylabel" -n "$namespace" | tail -n1 \
| awk -F' ' '{print $1}' | xargs -I{} \
  kubectl logs -n "$namespace" -f {}
```

- Get node hardware resource utilization

```
kubectl get nodes --no-headers \
| awk '{print $1}' | xargs -I {} \
sh -c 'echo {}; kubectl describe node {} | grep Allocated -A 5'
```

```
kubectl get nodes --no-headers | awk '{print $1}' | xargs -I {} \
sh -c 'echo {}; kubectl describe node {} | grep Allocated -A 5 \
| grep -ve Event -ve Allocated -ve percent -ve -- ; echo'
```

- Apply the configuration in manifest.yaml and delete all the other configmaps that are not in the file.

```
kaubectl apply --prune -f manifest.yaml --all --prune-whitelist=core/v1/ConfigMap
```

## 1.13 More Resources

License: Code is licensed under MIT License.

- Useful links

- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- [https://github.com/kubecamp/kubernetes\\_in\\_2\\_days](https://github.com/kubecamp/kubernetes_in_2_days)
- <https://marc.xn--wckerlin-0za.ch/computer/kubernetes-on-ubuntu-16-04>
- <https://codefresh.io/kubernetes-guides/kubernetes-cheat-sheet/>