

Kubernetes CheatSheet

KUBERNETES

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

File me Issues or star this repo.

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

1.1 Common Usage

Yaml Templates: <https://cheatsheet.dennyzhang.com/cheatsheet-kubernetes-yaml>

1.1.1 Common Commands

Name	Command
List everything	<code>kubectl get all --all-namespaces</code>
Validate yaml file with dry run	<code>kubectl create --dry-run --validate -f pod-dummy.yaml</code>
Run wget test temporarily	<code>kubectl run --rm mytest --image=busybox -it</code>
Run curl test temporarily	<code>kubectl run --rm mytest --image=yauritux/busybox-curl -it</code>
Get system conf via configmap	<code>kubectl -n kube-system get cm kubeadm-config -o yaml</code>
Explain resource	<code>kubectl explain pods, kubectl explain svc</code>
Get all services	<code>kubectl get service --all-namespaces</code>
Get services sorted by name	<code>kubectl get services --sort-by=.metadata.name</code>
Get pods sorted by restart count	<code>kubectl get pods --sort-by='.status.containerStatuses[0].restartCount'</code>
Query healthcheck endpoint	<code>curl -L http://127.0.0.1:10250/healthz</code>
Open a bash terminal in a pod	<code>kubectl exec -it storage sh</code>
Check pod environment variables	<code>kubectl exec redis-master-ft9ex env</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
kubectl	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 pod resource usage	<code>kubectl top pod</code>
Get resource usage for a given pod	<code>kubectl top <podname> --containers</code>
List resource utilization for all containers	<code>kubectl top pod --all-namespaces --containers=true</code>

1.3 Resources Deletion

Name	Command
Delete pod	<code>kubectl delete pod hello-node-95913-n63qs -n \$my-namespace</code>
Delete pods by labels	<code>kubectl delete pod -l env=test</code>
Delete deployments by labels	<code>kubectl delete deployment -l app=wordpress</code>
Delete persist volumes by labels	<code>kubectl delete pvc -l app=wordpress</code>
Delete statefulset only (not pods)	<code>kubectl delete sts <stateful_set_name> --cascade=false</code>

1.4 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, 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}}' <pod-name></code>
kubectl run command	<code>kubectl exec -it -n "\$ns" "\$podname" - sh -c "echo \$msg »/dev/err.log"</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}{\n}'</code>

1.5 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>

1.6 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>

1.7 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>

1.8 StatefulSet

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

1.9 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>

1.10 Security

Name	Command
List certificates	<code>kubectl get csr</code>

1.11 Other Components

1.11.1 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>

1.11.2 Events & Metrics

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

1.11.3 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 <cluster-name></code>
List all namespaces defined	<code>kubectl get namespaces</code>
kubectl config file	<code>~/.kube/config</code>

1.11.4 Network

Name	Command
Temporarily add a port-forwarding	<code>kubectl port-forward redis-izl09 6379</code>
Add port-forwarding for deployment	<code>kubectl port-forward deployment/redis-master 6379:6379</code>
Add port-forwarding for replicaset	<code>kubectl port-forward rs/redis-master 6379:6379</code>
Add port-forwarding for service	<code>kubectl port-forward svc/redis-master 6379:6379</code>

1.11.5 Endpoint

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

1.12 Basic

1.12.1 Key Concepts

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

1.12.2 Kubernetes Critical Files

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

1.12.3 Check status

Name	Summary
List everything	kubectl get all --all-namespaces
Get cluster info	kubectl cluster-info
Get configuration	kubectl config view
Get kubectl version	kubectl version
Get component status	kubectl get componentstatus
Similar to docker ps	kubectl get nodes
Similar to docker inspect	kubectl describe pod nginx-app-413181-cn
Similar to docker logs	kubectl logs
Similar to docker exec	kubectl exec
Get services for current namespace	kubectl get svc
Get node status	kubectl describe node \$node_name

1.12.4 Kubernetes Developer Resources

Name	Summary
API Conventions	link: API Conventions

1.13 Minikube

Name	Command
Get minikube version	minikube version, link: all minikube releases
Start minikube env with a bigger machine flavor	minikube start, minikube start --memory 5120 --cpus=4
Gets all available Kubernetes versions for minikube	minikube get-k8s-versions
Start minikube with given k8s version	minikube start --kubernetes-version v1.10.0, kubectl version
minikube docker-env	eval \$(minikube docker-env)
Get minikube log	minikube logs
Get dashboard	minikube dashboard
SSH to minikube vm	minikube ssh
Get ip	minikube ip
Get cluster info	kubectl cluster-info
List addons	minikube addons list
Get service info	minikube service \$srv_name

1.14 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
```

1.15 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://marc.xn--wckerlin-0za.ch/computer/kubernetes-on-ubuntu-16-04>
- <https://codefresh.io/kubernetes-guides/kubernetes-cheat-sheet/>