

1.1 COMMON COMMANDS

Name	Command
Run curl test temporarily	<code>kubectl run --rm mytest --image=yauritux/busybox-curl -it</code>
Run wget test temporarily	<code>kubectl run --rm mytest --image=busybox -it</code>
Run nginx deployment with 2 replicas	<code>kubectl run my-nginx --image=nginx --replicas=2 --port=80</code>
Run nginx pod and expose it	<code>kubectl run my-nginx --restart=Never --image=nginx --port=80 --expose</code>
Run nginx deployment and expose it	<code>kubectl run my-nginx --image=nginx --port=80 --expose</code>
Set namespace preference	<code>kubectl config set-context <context_name> --namespace=<ns_name></code>
List pods with nodes info	<code>kubectl get pod -o wide</code>
List everything	<code>kubectl get all --all-namespaces</code>
Get all services	<code>kubectl get service --all-namespaces</code>
Get all deployments	<code>kubectl get deployments --all-namespaces</code>
Show nodes with labels	<code>kubectl get nodes --show-labels</code>
Get resources with json output	<code>kubectl get pods --all-namespaces -o json</code>
Validate yaml file with dry run	<code>kubectl create --dry-run --validate -f pod-dummy.yaml</code>
Start a temporary pod for testing	<code>kubectl run --rm -i -t --image=alpine test-\$RANDOM -- sh</code>
kubectl run shell command	<code>kubectl exec -it mytest -- ls -l /etc/hosts</code>
Get system conf via configmap	<code>kubectl -n kube-system get cm kubeadm-config -o yaml</code>
Get deployment yaml	<code>kubectl -n denny-websites get deployment mysql -o yaml</code>
Explain resource	<code>kubectl explain pods , kubectl explain svc</code>
Watch pods	<code>kubectl get pods -n wordpress --watch</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>
Enable kubectl shell autocompletion	<code>echo "source <(kubectl completion bash)" >> ~/.bashrc , and reload</code>
Use minikube dockerd in your laptop	<code>eval \$(minikube docker-env)</code> , No need to push docker hub any more
Kubectl apply a folder of yaml files	<code>kubectl apply -R -f .</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>
List pods and images	<code>kubectl get pods -o='custom-columns=PODS:.metadata.name,Images:.spec.containers[*].image'</code>
List all container images	<code>list-all-images.sh</code>
kubeconfig skip tls verification	<code>skip-tls-verify.md</code>
Ubuntu install kubectl	<code>"deb https://apt.kubernetes.io/ kubernetes-xenial main"</code>
Reference	GitHub: kubernetes releases
Reference	minikube cheatsheet, docker cheatsheet, OpenShift CheatSheet

1.2 CHECK PERFORMANCE

Name	Command
Get node resource usage	<code>kubectl top node</code>

Name	Command
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/<pod-name> -n <my-namespace></code>
Delete pod by force	<code>kubectl delete pod/<pod-name> --grace-period=0 --force</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 all resources filtered by labels	<code>kubectl delete pods,services -l name=myLabel</code>
Delete resources under a namespace	<code>kubectl -n my-ns delete po,svc --all</code>
Delete persist volumes by labels	<code>kubectl delete pvc -l app=wordpress</code>
Delete state fulset only (not pods)	<code>kubectl delete sts/<stateful_set_name> --cascade=false</code>

cheatsheet.dennyzhang.com

1.4 LOG & CONF 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>
kubectl config file	<code>~/.kube/config</code>
Kubernets working dir	<code>/var/lib/kubelet/</code>
Docker working dir	<code>/var/lib/docker/ , /var/log/containers/</code>
Etd working dir	<code>/var/lib/etcd/</code>
Network cni	<code>/etc/cni/net.d/</code>
Log files	<code>/var/log/pods/</code>
log in worker node	<code>/var/log/kubelet.log , /var/log/kube-proxy.log</code>
log in master node	<code>kube-apiserver.log , kube-scheduler.log , kube-controller-manager.log</code>
Env	<code>/etc/systemd/system/kubelet.service.d/10-kubeadm.conf</code>
Env	<code>export KUBECONFIG=/etc/kubernetes/admin.conf</code>

1.5 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/<pod-name> -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>

Name	Command
List all unhealthy pods	<code>kubectl get pods --field-selector=status.phase!=Running --all-namespaces</code>
List running pods	<code>kubectl get pods --field-selector=status.phase=Running</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>
Watch pods	<code>kubectl get pods -n wordpress --watch</code>
Get pod by selector	<code>kubectl get pods --selector="app=syslog" -o jsonpath='{.items[*].metadata.name}'</code>
List pods and images	<code>kubectl get pods -o='custom-columns=PODS:.metadata.name,Images:.spec.containers[*].image'</code>
List pods and containers	<code>-o='custom-columns=PODS:.metadata.name,CONTAINERS:.spec.containers[*].name'</code>
Reference	Link: kubernetes yaml templates

1.6 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://dennyzhang.com</code>

1.7 DEPLOYMENT & SCALE

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>
Reference	Link: kubernetes yaml templates , Link: Pausing and Resuming a Deployment

1.8 QUOTA & LIMITS & RESOURCE

[cheatsheet.dennyzhang.com](#)

Name	Command
List Resource Quota	<code>kubectl get resourcequota</code>
List Limit Range	<code>kubectl get limitrange</code>
Customize resource definition	<code>kubectl set resources deployment nginx -c=nginx --limits=cpu=200m</code>
Customize resource definition	<code>kubectl set resources deployment nginx -c=nginx --limits=memory=512Mi</code>
Reference	Link: kubernetes yaml templates

1.9 SERVICE

Name	Command
List all services	<code>kubectl get services</code>
List service endpoints	<code>kubectl get endpoints</code>

Name	Command
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>
Expose deployment as lb service	<code>kubectl expose deployment/my-app --type=LoadBalancer --name=my-service</code>
Expose service as lb service	<code>kubectl expose service/wordpress-1-svc --type=LoadBalancer --name=ns1</code>
Reference	Link: kubernetes yaml templates

1.10 SECRETS

Name	Command
List secrets	<code>kubectl get secrets --all-namespaces</code>
Generate secret	<code>echo -n 'mypasswd' , then redirect to <code>base64 --decode</code></code>
Get secret	<code>kubectl get secret denny-cluster-kubeconfig</code>
Get a specific field of a secret	<code>kubectl get secret denny-cluster-kubeconfig -o jsonpath="{.data.value}"</code>
Create secret from cfg file	<code>kubectl create secret generic db-user-pass --from-file=./username.txt</code>
Reference	Link: kubernetes yaml templates, Link: Secrets

1.11 STATEFULSET

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

1.12 VOLUMES & VOLUME CLAIMS

Name	Command
List storage class	<code>kubectl get storageclass</code>
Check the mounted volumes	<code>kubectl exec storage ls /data</code>
Check persist volume	<code>kubectl describe pv/pv0001</code>
Copy local file to pod	<code>kubectl cp /tmp/my <some-namespace>/<some-pod>:/tmp/server</code>
Copy pod file to local	<code>kubectl cp <some-namespace>/<some-pod>:/tmp/server /tmp/my</code>
Reference	Link: kubernetes yaml templates

1.13 EVENTS & METRICS

Name	Command
View all events	<code>kubectl get events --all-namespaces</code>
List Events sorted by timestamp	<code>kubectl get events --sort-by=.metadata.creationTimestamp</code>

1.14 NODE MAINTENANCE

Name	Command
Mark node as unschedulable	<code>kubectl cordon \$NODE_NAME</code>
Mark node as schedulable	<code>kubectl uncordon \$NODE_NAME</code>
Drain node in preparation for maintenance	<code>kubectl drain \$NODE_NAME</code>

1.15 NAMESPACE & SECURITY

Name	Command
List authenticated contexts	<code>kubectl config get-contexts , ~/.kube/config</code>
Set namespace preference	<code>kubectl config set-context <context_name> --namespace=<ns_name></code>
Load context from config file	<code>kubectl get cs --kubeconfig kube_config.yml</code>
Switch context	<code>kubectl config use-context <cluster-name></code>
Delete the specified context	<code>kubectl config delete-context <cluster-name></code>
List all namespaces defined	<code>kubectl get namespaces</code>
List certificates	<code>kubectl get csr</code>
Reference	Link: kubernetes yml templates

1.16 NETWORK

Name	Command
Temporarily add a port-forwarding	<code>kubectl port-forward redis-134 6379:6379</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>
Get network policy	<code>kubectl get NetworkPolicy</code>

1.17 PATCH

Name	Summary
Patch service to loadbalancer	<code>kubectl patch svc \$svc_name -p '{"spec": {"type": "LoadBalancer"}}'</code>

1.18 EXTENSTIONS

Name	Summary
List api group	<code>kubectl api-versions</code>
List all CRD	<code>kubectl get crd</code>
List storageclass	<code>kubectl get storageclass</code>
List all supported resources	<code>kubectl api-resources</code>

1.19 COMPONENTS & SERVICES

cheatsheet.dennyzhang.com

1.19.1 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

1.19.2 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.

1.19.3 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

1.19.4 TOOLS

Name	Summary
kubect l	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
Kubernetes Components	Link: Kubernetes Components

1.20 MORE RESOURCES

License: Code is licensed under [MIT License](#).

<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

<https://codefresh.io/kubernetes-guides/kubernetes-cheat-sheet/>