## **Kubernetes Cheat Sheet**

### **CONTEXT & CONFIGURATION**

kubectl config view # Show kubeconfig details

kubectl config get-contexts # List all available contexts

kubectl config current-context # Show current context

kubectl config use-context <context-name> # Switch to a specific context

kubectl config set-context <name> --cluster=<cluster> --user=<user> # Create new context

kubectl config unset contexts.<context-name> # Remove a context

### **CLUSTER INFORMATION**

kubectl cluster-info # Show cluster information

kubectl version # Show client and server version

kubectl api-resources # List all resource types

kubectl api-versions # List API versions

kubectl get all --all-namespaces # List all resources in all namespaces

### **NODES**

kubectl get nodes # List all nodes

kubectl describe node <node-name> # Show detailed info about node

kubectl cordon <node-name> # Mark node unschedulable

kubectl uncordon <node-name> # Mark node schedulable

kubectl drain <node-name> --ignore-daemonsets # Evict pods from node safely

kubectl delete node <node-name> # Remove node from cluster

### **NAMESPACES**

kubectl get namespaces # List all namespaces

kubectl create namespace <name> # Create a namespace

kubectl delete namespace <name> # Delete a namespace

kubectl config set-context --current --namespace=<name> # Set default namespace

### **PODS**

kubectl get pods # List pods

kubectl get pods -n <namespace> # List pods in specific namespace

kubectl describe pod <pod-name> # Show pod details

kubectl logs <pod-name> # Show logs of pod

kubectl logs <pod-name> -c <container-name> # Logs of specific container in pod

kubectl exec -it <pod-name> -- /bin/bash # SSH into pod

kubectl delete pod <pod-name> # Delete pod

### **DEPLOYMENTS**

kubectl get deployments # List deployments

kubectl describe deployment <name> # Deployment details

kubectl create deployment <name> --image=<image> # Create deployment

kubectl delete deployment <name> # Delete deployment

kubectl scale deployment <name> --replicas=<n> # Scale deployment

kubectl rollout restart deployment/<name> # Restart deployment

kubectl rollout status deployment/<name> # Rollout status

kubectl rollout undo deployment/<name> # Rollback deployment

kubectl set image deployment/<DEPLOYMENT\_NAME> <CONTAINER\_NAME>=<NEW\_IMAGE>:<TAG> #update the image

### **SERVICES**

kubectl get svc # List services

kubectl describe svc <name> # Service details

kubectl expose pod <pod-name> --type=NodePort --port=<port> # Expose pod

kubectl delete svc <name> # Delete service

kubectl expose deploy/<name> --port=80 --target-port=8080

#Create a ClusterIP service

kubectl expose deploy/<name> --type=LoadBalancer --port=80

#Create a LoadBalancer service

### **CONFIGMAPS & SECRETS**

kubectl get configmaps # List configmaps

kubectl create configmap <name> --from-literal=key=value # Create configmap

kubectl get secrets # List secrets

kubectl create secret generic <name> --from-literal=key=value # Create secret

kubectl describe secret <name> # Show secret details

### **VOLUMES, PVs & PVCs**

kubectl get pv # List Persistent Volumes

kubectl get pvc # List Persistent Volume Claims

kubectl describe pvc <name> # Show details

### **SERVICE ACCOUNT & SECURITY CONTEXT**

kubectl create serviceaccount <name> # Create service account

kubectl get serviceaccounts # List service accounts

### **YAML GENERATION**

kubectl create deployment nginx --image=nginx --dry-run=client -o yaml # Generate YAML without creating

kubectl get pod <name> -o yaml # Get YAML of resource

### **GO TEMPLATE FORMATTING**

kubectl get pods -o go-template='{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}'

# List only pod names

kubectl get nodes -o go-template='{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}'

# List only node names

kubectl get pods -o go-template='{{range .items}}{{if eq .status.phase "Running"}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' #Filter Pods in "Running" state

export NODE\_PORT="$(kubectl get services/kubernetes-bootcamp -o go-template='{{(index .spec.ports 0).nodePort}}')" #get nodeport of service

**APPLY / CREATE / DELETE RESOURCES FROM YAML FILE**

kubectl apply -f <file.yaml> # Create or update resources from YAML file

kubectl apply -f <directory/> # Apply all YAMLs in directory

kubectl create -f <file.yaml> # Create resources from YAML (error if exists)

kubectl delete -f <file.yaml> # Delete resources defined in YAML

kubectl delete -f <directory/> # Delete all resources in directory

kubectl replace -f <file.yaml> # Replace existing resource

**PORT FORWARDING & PROXY**

kubectl port-forward svc/<service-name> <local-port>:<service-port> # Forward local port to service

kubectl port-forward pod/<pod-name> <local-port>:<pod-port> # Forward local port to pod

kubectl proxy # Proxy to Kubernetes API server

**Label Commands**

kubectl label pods <pod-name> env=prod #Add label env=prod to a pod

kubectl label pods <pod-name> env=staging --overwrite #Update an existing label

kubectl label pods <pod-name> tier- #Remove the label tier

kubectl get pods -l app=nginx #pods with label nginx

kubectl get pods -l '!tier' #Pods without the tier label

## **HELM COMMANDS**

### **Repository Management**

helm repo add <repo-name> <url> # Add a Helm repository

helm repo add bitnami https://charts.bitnami.com/bitnami # Example

helm repo update # Update all added repositories

helm repo list # List configured repositories

helm search repo <keyword> # Search charts in added repos

helm search repo nginx # Example: Search for "nginx" charts

helm search hub <keyword> # Search Artifact Hub for charts

### **Release Management**

helm install <release-name> <chart> [flags] # Install a chart

helm install my-app bitnami/nginx --namespace=dev -f values.yaml

helm upgrade <release-name> <chart> [flags] # Upgrade a release

helm upgrade my-app bitnami/nginx --set replicaCount=3 --atomic

helm uninstall <release-name> [flags] # Uninstall a release

helm uninstall my-app --keep-history # Keep release history

helm list -A # List releases in all namespaces

helm rollback <release-name> <revision> # Rollback to a specific version

helm rollback my-app 2 # Example

helm status <release-name> # Show release status/details

### **Common Flags**

-f/--values values.yaml # Specify values file

--set key1=val1,key2=val2 # Override values inline

--namespace dev # Target namespace

--dry-run --debug # Simulate install/upgrade

--wait # Wait for resources to be ready

--atomic # Auto-rollback on failure

--version 4.1.2 # Specify chart version

### 

### **Chart Operations**

helm show values <chart> # Display default values

helm show values bitnami/nginx > values.yaml # Save default values

helm template <chart> [flags] # Render templates locally

helm template my-chart ./mychart -f prod-values.yaml

helm lint ./mychart # Validate chart syntax

helm package ./mychart # Package chart to .tgz

helm create my-new-chart # Scaffold new chart

### **Debugging & Inspection**

helm get all <release-name> # All release info (manifest, hooks, values)

helm get manifest my-app # Show generated Kubernetes resources

helm get values my-app # Show values used for release

helm get hooks my-app # List release hooks

helm history my-app # Show release revision history

### **Dependency Management**

helm dependency list ./mychart # List chart dependencies

helm dependency update ./mychart # Download dependencies

helm dependency build ./mychart # Rebuild from Chart.lock