30-DAY KUBERNETES MASTERY PLAN

Kubernetes Cheat Sheet: Daily Reference

Hendrawan

Monday 12th May, 2025

1 Introduction

Kubernetes automates containerized application deployment, scaling, and management. `kubectl` is the command-line tool for interacting with clusters. This cheat sheet, authored by Hendrawan, provides a comprehensive, user-friendly reference for daily Kubernetes tasks, aligned with v1.33. Install `kubectl` via Kubernetes Docs.

2 Setup and Configuration

2.1 Autocomplete

- **BASH**: `source <(kubectl completion bash)`
  Add to `~/.bashrc`
- **ZSH**: `source <(kubectl completion zsh)`
  Add to `~/.zshrc`
- **FISH**: `kubectl completion fish | source`
  Save to `~/.config/fish/completions/kubectl.fish`

2.2 Contexts and Namespaces

- List contexts: `kubectl config get-contexts`
- Switch context: `kubectl config use-context <name>`
- Set namespace: `kubectl config set-context -current -namespace=<name>`

2.3 Cluster Info

- Cluster details: `kubectl cluster-info`
- Version: `kubectl version`

3 Managing Resources

3.1 Pods

- Create: `kubectl run <name> -image=<image>`
  Example: `kubectl run nginx -image=nginx`
- List: `kubectl get pods`
  All namespaces: `kubectl get pods -A`
- Describe: `kubectl describe pod <name>`
- Logs: `kubectl logs <name>` (`-f` for streaming)
- Exec: `kubectl exec -it <name> - sh`
- Delete: `kubectl delete pod <name>`

3.2 Deployments

- Create: `kubectl create deployment <name> -image=<image>`
  Example: `kubectl create deployment web -image=nginx`
- List: `kubectl get deployments`
- Describe: `kubectl describe deployment <name>`
- Scale: `kubectl scale deployment <name> -replicas=<n>`
  Example: `kubectl scale deployment web -replicas=3`
- Rollout status: `kubectl rollout status deployment <name>`
- Rollout undo: `kubectl rollout undo deployment <name>`
- Delete: `kubectl delete deployment <name>`

3.3 Services

- Create: `kubectl expose deployment <name> -type=LoadBalancer`
  Example: `kubectl expose deployment web -type=ClusterIP -port=80`
- List: `kubectl get services`
- Describe: `kubectl describe service <name>`
- Delete: `kubectl delete service <name>`

3.4 ConfigMaps

- Create: `kubectl create configmap <name> -from-literal=<key>=<value>`
  Example: `kubectl create configmap app-config -from-literal=env=prod`
- List: `kubectl get configmaps`
- Describe: `kubectl describe configmap <name>`
- Delete: `kubectl delete configmap <name>`

3.5 Secrets

- Create: `kubectl create secret generic <name> -from-literal=<key>=<value>`
  Example: `kubectl create secret generic db-secret -from-literal=password=secret`
- List: `kubectl get secrets`
- Describe: `kubectl describe secret <name>`

- Delete: `kubectl delete secret <name>`

4 Advanced Commands

4.1 Namespaces

- Create: `kubectl create namespace <name>`
  Example: `kubectl create namespace dev`
- List: `kubectl get namespaces`
- Use: `kubectl config set-context -current -namespace=<name>`
- Delete: `kubectl delete namespace <name>`

4.2 Persistent Volumes (PV)

- List: `kubectl get pv`
- Describe: `kubectl describe pv <name>`

4.3 Persistent Volume Claims (PVC)

- List: `kubectl get pvc`
- Describe: `kubectl describe pvc <name>`

4.4 Ingress

- List: `kubectl get ingress`
- Describe: `kubectl describe ingress <name>`

4.5 Network Policies

- List: `kubectl get networkpolicies`
- Describe: `kubectl describe networkpolicy <name>`

5 Troubleshooting

- Events: `kubectl get events`
  Tip: Use to diagnose cluster issues.
- Describe: `kubectl describe <resource> <name>`
  Example: `kubectl describe pod nginx`
- Logs: `kubectl logs <pod-name>`
  Streaming: `kubectl logs -f <pod-name>`
- Exec: `kubectl exec -it <pod-name> - sh`
  Tip: Debug inside a pod.

6 Helm Basics

- Install: `helm install <release> <chart>`
  Example: `helm install my-app bitnami/nginx`
- List: `helm list`
- Upgrade: `helm upgrade <release> <chart>`
- Uninstall: `helm uninstall <release>`

7 Best Practices

- **Labels**: Use for selection: `kubectl get pods -l app=nginx`
- **Namespaces**: Organize resources: `kubectl create namespace dev`
- **ConfigMaps/Secrets**: Store configurations securely.
- **Ingress**: Use for external access: `kubectl apply -f ingress.yaml`
- **Monitoring**: Use Prometheus/Grafana: `helm install prometheus prometheus-community/prometheus`
- **Logging**: Use EFK: `helm install efk fluent/elasticsearch-kibana`

8 Reference Tables

8.1 Resource Types

Full Name	Abbreviation
pod	po
service	svc
deployment	deploy
statefulset	sts
daemonset	ds
job	
cronjob	
configmap	cm
secret	
persistentvolume	pv
persistentvolumeclaim	pvc
ingress	
networkpolicy	

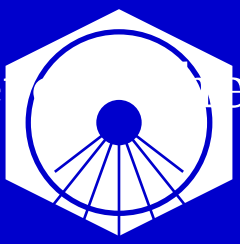
8.2 Common Flags

Flag	Description
-o wide	Show additional info
-o yaml	Output in YAML
–all-namespaces (-A)	List all namespaces
–dry-run=client	Simulate changes
-f <file>	Specify file
-l <label>	Filter by label

9 Useful Tools

- **kubectx**: Manage contexts
  Install: `brew install kubectx`
  Usage: `kubectx`, `kubectx <context>`
- **kubens**: Manage namespaces
  Install: Same as kubectx
  Usage: `kubens`, `kubens <namespace>`
- **stern**: Tail logs from multiple pods
  Install: `go get github.com/wercker/stern`
  Usage: `stern <pod-pattern>`

Kubernetes Cheat Sheet



- Essential Commands
- YAML Examples
- Quick Reference

By Hendrawan

Published on Monday 12th May, 2025