KUBERNETES (K8s) COMMANDS CHEAT SHEET – FULL DEVOPS EDITION

---

1. Basic Cluster Commands

`kubectl version`

Show client + server K8s version.

`kubectl cluster-info`

See cluster master & DNS info.

`kubectl get all`

List all resources (pods, svc, deployments, etc.) in default namespace.

---

2. Working With Contexts

`kubectl config get-contexts`

List all contexts (clusters).

`kubectl config use-context dev`

Switch to another cluster.

`kubectl config current-context`

Show which cluster you are using.

---

3. Namespaces

`kubectl get namespaces`

List namespaces.

`kubectl create namespace dev`

Create namespace.

`kubectl delete namespace dev`

Delete namespace.

`kubectl config set-context --current --namespace=dev`

Set default namespace.

---

4. Pods

`kubectl get pods`

List pods.

`kubectl get pods -o wide`

Show pod IP, node, etc.

`kubectl describe pod pod-name`

Detailed pod info & events.

`kubectl logs pod-name`

View logs.

`kubectl logs -f pod-name`

Follow live logs.

`kubectl exec -it pod-name -- bash`

Enter pod terminal.

`kubectl delete pod pod-name`

Delete a pod.

---

5. Deployments

`kubectl get deployments`

List deployments.

`kubectl create deployment web --image=nginx`

Create deployment.

`kubectl scale deployment web --replicas=5`

Scale number of pods.

`kubectl rollout status deployment/web`

Check deployment rollout status.

`kubectl rollout undo deployment/web`

Rollback to previous version.

`kubectl delete deployment web`

Delete deployment.

---

6. ReplicaSets

`kubectl get rs`

List ReplicaSets.

`kubectl describe rs rs-name`

ReplicaSet details.

---

7. Services

`kubectl get svc`

List services.

`kubectl expose deployment web --type=NodePort --port=80`

Expose deployment as a service.

`kubectl describe svc web`

Service details.

`kubectl get svc -o wide`

See service cluster IP & ports.

---

8. ConfigMaps

`kubectl create configmap app-config --from-literal=env=prod`

Create ConfigMap.

`kubectl create configmap myconfig --from-file=config.properties`

Create from file.

`kubectl get configmaps`

List ConfigMaps.

`kubectl describe configmap app-config`

View config details.

---

9. Secrets

`kubectl create secret generic db-secret --from-literal=password=1234`

Create secret.

`kubectl create secret generic tls-secret --from-file=server.crt --from-file=server.key`

Create TLS secret.

`kubectl get secrets`

List secrets.

`kubectl describe secret db-secret`

Describe secret.

---

10. YAML Apply, Update & Delete

`kubectl apply -f deployment.yaml`

Apply or update resource.

`kubectl delete -f deployment.yaml`

Delete resource.

`kubectl edit deployment web`

Edit resource live in editor.

---

11. Nodes & Cluster Info

`kubectl get nodes`

List nodes.

`kubectl get nodes -o wide`

Node details (OS, internal IP).

`kubectl describe node node1`

Node details (taints, capacity).

`kubectl drain node1 --ignore-daemonsets`

Drain node safely.

`kubectl cordon node1`

Mark node unschedulable.

`kubectl uncordon node1`

Mark node schedulable.

---

12. Resource Usage

`kubectl top pods`

CPU & memory usage of pods.

`kubectl top nodes`

CPU & memory usage of nodes.

(Metrics server required)

---

13. Taints & Tolerations

Add taint:

```
kubectl taint nodes node1 key=value:NoSchedule
```

Remove taint:

```
kubectl taint nodes node1 key=value:NoSchedule-
```

Used for:
✔️ Dedicate nodes
✔️ Restrict workloads
✔️ Isolation

---

14. Labels & Selectors

`kubectl label pod web app=frontend`

Add label.

`kubectl get pods -l app=frontend`

Filter with label.

`kubectl label pod web app-`

Remove label.

---

15. Port Forwarding

`kubectl port-forward pod/mypod 8080:80`

Access pod locally.

Common for debugging APIs.

---

16. Ingress

`kubectl get ingress`

List ingress rules.

`kubectl describe ingress my-ingress`

Ingress details.

---

17. StatefulSets

`kubectl get statefulsets`

List StatefulSets.

`kubectl describe statefulset mysql`

Stateful app details.

Used for:
Databases
Kafka
ElasticSearch

---

18. DaemonSets

`kubectl get daemonsets`

List DaemonSets.

Used for:
Logging agents

Monitoring agents

---

19. Jobs & CronJobs

`kubectl create job myjob --image=busybox`

Create Job.

`kubectl create cronjob backup --image=busybox --schedule="*/5 * * * *"`

Create CronJob.

`kubectl get jobs`

List jobs.

`kubectl get cronjobs`

List CronJobs.

---

20. Pod Debugging

`kubectl describe pod web`

Check events.

`kubectl logs pod --previous`

Check logs of crashed container.

`kubectl exec -it web -- sh`

Debug inside pod.

`kubectl get pod web -o yaml`

View complete pod spec.

---

21. Troubleshooting Cluster

`kubectl get events`

Cluster-wide events.

`kubectl get endpoints`

Check service endpoints.

`kubectl get componentstatus`

Master component status (older K8s).

`kubectl get networkpolicies`

Check network restrictions.

---

22. Network Policies

`kubectl get netpol`

List policies.

`kubectl describe netpol`

Network policy details.

Used for:
Restrict pod-to-pod communication
Zero-trust networking

---

23. Storage

`kubectl get pv`

List persistent volumes.

`kubectl get pvc`

List persistent volume claims.

`kubectl describe pv pv-name`

PV details.

`kubectl describe pvc pvc-name`

PVC details.

---

24. Service Accounts & RBAC

`kubectl get serviceaccounts`

List service accounts.

`kubectl create serviceaccount dev-sa`

Create SA.

`kubectl get clusterrole`

List cluster roles.

`kubectl get clusterrolebinding`

List bindings.

Used for:
✔️ Access control
✔️ Least privilege
✔️ Pod-to-AWS auth (IRSA)

---

25. Deleting Everything

`kubectl delete all --all`

Delete pods, svc, deployments in namespace.

`kubectl delete namespace dev`

Delete entire namespace.

---

26. Useful Shortcuts

`kubectl get po`

Short for pods.

`kubectl get deploy`

Short for deployments.

`kubectl get svc`

Short for service.

`kubectl get ing`

Short for ingress.

`kubectl get cm`

Short for configmap.

`kubectl get no`

Short for nodes.

---

27. Apply with Dry Run (very important)

`kubectl apply -f app.yaml --dry-run=client`

Check if YAML is valid without applying.

---

28. Debugging Node Issues

`kubectl describe node node1`

Check taints, disk pressure, memory pressure.

`kubectl get nodes -o json | jq '.items[].status.conditions'`

Detailed node conditions.

---

29. Extract Pod YAML

`kubectl get pod web -o yaml > pod.yaml`

Export pod definition.

Useful for reproducing or modifying pods.

---

30. Kubernetes API Access

`kubectl proxy`

Start API server proxy for debugging.

31. Advanced Pod Debugging

`kubectl exec -it pod --container app -- sh`

Exec into a specific container inside a multi-container pod.

`kubectl logs pod -c app`

View logs of a specific container.

`kubectl logs pod --previous`

Check logs of a container that crashed.

`kubectl describe pod pod | grep -i error`

Search for errors in pod events.

---

32. Ephemeral Debug Container (K8s 1.23+)

`kubectl debug podname -it --image=busybox`

Attach a temporary debug container.

`kubectl debug node/node1 -it --image=busybox`

Debug an entire node.

Used for:
✔️ CrashLoopBackOff
✔️ ImagePullBackOff
✔️ Node-level debugging

---

33. ImagePull & Crash Issues

`kubectl get pod pod -o jsonpath='{.status.containerStatuses[].state.waiting.reason}'`

Get reason for failed pod.

Common issues:

- ImagePullBackOff
- ErrImagePull
- CrashLoopBackOff

---

34. Rollouts & History

`kubectl rollout history deployment web`

View deployment history.

`kubectl rollout undo deployment web --to-revision=2`

Rollback to a specific version.

`kubectl rollout pause deployment web`

Pause rollout (for debugging).

`kubectl rollout resume deployment web`

Resume rollout.

---

35. Advanced Scaling

`kubectl scale deployment web --replicas=0`

Scale down to zero.

`kubectl scale statefulset mysql --replicas=3`

Scale StatefulSet (ordered).

---

36. YAML Dry Run & Diff

`kubectl diff -f app.yaml`

Show changes before applying.

`kubectl apply -f app.yaml --dry-run=server`

Server-side validation without apply.

Useful for CI/CD pipelines.

---

37. Patching (Very Important for DevOps)

Patch resources inline:

```
kubectl patch deployment web -p '{"spec": {"replicas": 4}}'
```

Patch with strategic merge:

```
kubectl patch svc web --patch-file patch.yaml
```

Used for:
✔️ Hotfixes
✔️ On-the-fly changes
✔️ CI/CD automation

---

38. Node Health & Troubleshooting

`kubectl get nodes -o json | jq '.items[].status.conditions[] | select(.status=="False")'`

Find unhealthy nodes.

`kubectl describe node node1`

Check disk pressure, memory pressure.

`kubectl get pods --all-namespaces -o wide --field-selector spec.nodeName=node1`

List all pods on a node.

---

39. Events & Clusterd-level Issues

`kubectl get events`

Show all events.

`kubectl get events --sort-by='.lastTimestamp'`

Sort events by time.

`kubectl get --raw /metrics`

Access kube-apiserver metrics.

---

40. Service Debugging

`kubectl get endpoints web`

Check if service has endpoints.

`kubectl describe svc web`

Check wrong targetPort / selector.

`kubectl run test --image=busybox -it --rm -- wget web`

Test service connectivity.

---

41. Ingress Debugging

`kubectl get ingress`

List ingress.

`kubectl describe ingress web-ingress`

Check rules & events.

`kubectl get pods -n ingress-nginx`

Check ingress controller health.

---

42. Network Policy Debugging

`kubectl get netpol`

List network policies.

`kubectl describe netpol allow-app`

View network rules.

Test connectivity:

```
kubectl run test --rm -it --image=busybox -- wget http://pod-ip
```

---

43. Persistent Storage Commands

`kubectl get storageclass`

List storage classes.

`kubectl describe storageclass gp2`

Details of storage class.

`kubectl get pv`

List persistent volumes.

`kubectl get pvc`

List persistent volume claims.

`kubectl describe pvc my-pvc`

PVC status.

---

44. Logs & Audit

`kubectl logs pod --since=1h`

Logs from the last 1 hour.

`kubectl logs pod --tail=50`

Show last 50 lines.

`kubectl logs pod --timestamps`

Show logs with timestamps.

---

45. Node-to-Pod Debugging

`kubectl run debug --rm -it --image=busybox --command -- ping <pod-ip>`

Test pod connectivity.

`kubectl get pod -o wide`

Find pod IP.

---

46. Copy Files To & From Pod

`kubectl cp pod:/app/logs ./logs`

Copy from pod → local.

`kubectl cp file.txt pod:/app/`

Copy local → pod.

---

47. ConfigMaps & Secrets Debugging

`kubectl get configmaps -o yaml`

View ConfigMap values.

`kubectl get secret db-secret -o yaml`

Base64 encoded output.

Decode:

```
echo 'cGFzc3dvcmQ=' | base64 --decode
```

---

48. K8s Useful JSONPath Queries

Get pod image:

```
kubectl get pod web -o jsonpath='{.spec.containers[*].image}'
```

Get pod node:

```
kubectl get pod web -o jsonpath='{.spec.nodeName}'
```

---

49. Resource Quotas & Limits

`kubectl get resourcequota`

List quotas.

`kubectl describe resourcequota rq1`

Quota details.

---

50. LimitRanges

`kubectl get limitrange`

List pod limit ranges.

Used for:
✔️ default CPU/memory
✔️ maximum/minimum allowed resources

---

51. Service Account with Pod

`kubectl describe sa my-sa`

Show token & secrets.

`kubectl describe pod pod | grep ServiceAccount`

Check which service account pod uses.

---

52. RBAC Debugging

Test user permission:

```
kubectl auth can-i get pods --as user1
```

Test namespace-specific permission:

```
kubectl auth can-i delete pods -n dev
```

---

53. Port & Connectivity Debugging (Must Know)

Test port:

```
kubectl run test --rm -it --image=busybox -- nc -zv web 80
```

DNS check:

```
kubectl run test --rm -it --image=busybox -- nslookup web
```

---

54. Horizontal Pod Autoscaling

`kubectl autoscale deployment web --cpu-percent=50 --min=2 --max=10`

Create autoscaler.

`kubectl get hpa`

List HPAs.

---

55. Cluster Autoscaler Debugging

`kubectl -n kube-system get pods | grep autoscaler`

Check autoscaler pod.

---

56. Advanced Resource Filtering (field selectors)

Get pods scheduled on a specific node:

```
kubectl get pods --field-selector spec.nodeName=node1
```

Get pods that failed:

```
kubectl get pods --field-selector=status.phase=Failed
```

Get pods that are not running:

```
kubectl get pods --field-selector=status.phase!=Running
```

Get pods created in last 5 minutes:

```
kubectl get pods --sort-by=.metadata.creationTimestamp
```

---

57. Advance JSON & YAML Output Formatting

Get only pod names:

```
kubectl get pods -o jsonpath='{.items[*].metadata.name}'
```

Get node internal IPs:

```
kubectl get nodes -o jsonpath='{.items[*].status.addresses[?
(@.type=="InternalIP")].address}'
```

Get pod image names:

```
kubectl get pods -o jsonpath='{.items[*].spec.containers[*].image}'
```

---

58. Temporary BusyBox Pod (for debugging)

(one of the most used commands by DevOps engineers)

```
kubectl run tmp --rm -it --image=busybox -- sh
```

Used for:
✔️ DNS testing
✔️ Connecting to services
✔️ Checking network restrictions

---

59. Check Pod Events Script

Helpful command to sort events:

```
kubectl get events --sort-by='.lastTimestamp'
```

Best for:
✔️ CrashLoopBackOff
✔️ Pod scheduling failure
✔️ Image pull issues

---

60. Get Pod Environment Variables

```
kubectl exec -it <pod> -- printenv
```

Get env from container:

```
kubectl get pod <pod> -o jsonpath='{.spec.containers[*].env}'
```

---

61. Debugging Service DNS

Test DNS resolution:

```
kubectl run dns-test --rm -it --image=busybox -- nslookup svc-name
```

Test cluster DNS:

```
kubectl run dns-test --rm -it --image=busybox -- nslookup kubernetes.default
```

---

62. Debug Service Connectivity

Test port connectivity:

```
kubectl run test --rm -it --image=busybox -- nc -zv svc-name 80
```

Test via curl:

```
kubectl run test --rm -it --image=curlimages/curl -- curl http://svc-name
```

---

63. Debug Node Ports

Check nodeport access:

```
curl <node-ip>:<node-port>
```

---

64. Pod Security / User Permissions

Find which user the pod runs as:

```
kubectl get pod pod -o
jsonpath='{.spec.containers[*].securityContext.runAsUser}'
```

---

65. Copy Kubernetes Manifest From Live Resource

Extract current running deployment YAML:

```
kubectl get deploy web -o yaml > web.yaml
```

Extract current configmap YAML:

```
kubectl get cm app-config -o yaml > configmap.yaml
```

---

66. Validating YAML

Client-side validation

```
kubectl apply -f app.yaml --dry-run=client
```

Server-side validation

```
kubectl apply -f app.yaml --dry-run=server
```

Lint YAML (if installed):

```
kubeval app.yaml
```

---

67. Kubernetes API Access (Raw)

View cluster components metrics:

```
kubectl get --raw /metrics
```

Access API paths:

```
kubectl get --raw /api
```

---

68. Node Disk / Memory Pressure Debugging

Check node conditions:

```
kubectl describe node node1 | grep -i pressure
```

Look for:

- DiskPressure
- MemoryPressure
- PIDPressure

---

69. Node Logs (Master & Worker Debugging)

Kubelet logs:

```
journalctl -u kubelet -f
```

Container runtime logs:

```
journalctl -u containerd -f
```

---

70. Restarting Pods Properly

Delete pod safely (deployment recreates it):

```
kubectl delete pod pod-name
```

Force delete stuck pod:

```
kubectl delete pod pod-name --force --grace-period=0
```

---

71. Restart Deployment (Without Editing)

```
kubectl rollout restart deployment web
```

---

72. Checking Cluster Authentication

Test if user can perform action:

```
kubectl auth can-i create pods
```

Test as specific user:

```
kubectl auth can-i delete pods --as bob
```

---

73. Get Logs From ALL Pods of a Deployment

```
kubectl logs -l app=web --all-containers=true
```

Labels must match deployment selector.

---

74. Debugging Network Policies

List policies:

```
kubectl get netpol
```

Connectivity test across namespaces:

```
kubectl run test --rm -it --image=busybox -- sh
```

Inside:

```
wget http://pod-ip
```

---

75. ConfigMap Reload Troubleshooting

Pods do NOT automatically reload ConfigMaps unless:

✔️ Pod restarts
✔️ Sidecar reloaders (e.g., Reloader, ConfigMap reloader)
✔️ Using projected volumes

Check if mounted as volume:

```
kubectl get pod pod -o jsonpath='{.spec.volumes[*].configMap.name}'
```

---

76. Secret Decoding & Validation

Decode a secret:

```
echo 'cGFzc3dvcmQ=' | base64 --decode
```

Encode new value:

```
echo -n 'mypassword' | base64
```

---

77. Checking Pod Storage Paths

Check mounted volumes:

```
kubectl get pod pod -o jsonpath='{.spec.volumes}'
```

Check volume mount path:

```
kubectl get pod pod -o jsonpath='{.spec.containers[*].volumeMounts}'
```

---

78. Live Pod Debug Session

```
kubectl debug --image=busybox pod-name
```

This creates a temporary container INSIDE pod for debugging.

---

79. Upgrade Kubernetes Without Downtime (Cluster Admin)

Check nodes before upgrade:

```
kubectl get nodes
```

Upgrade node one-by-one:

```
kubectl drain node1 --ignore-daemonsets
```

Then update node OS/Kubelet.

```
kubectl uncordon node1
```

---

80. Advanced kube-proxy Debugging

View kube-proxy rules:

```
iptables -L -t nat
```

Inspect kube-proxy logs:

```
journalctl -u kube-proxy -f
```