KUBERNETES QUICK REFERENCE

kubectl, pods, deployments, services, configs, debugging

kubectl Basics

Cluster Info

```
kubectl cluster-info
kubectl get nodes
kubectl config current-context
kubectl config use-context my-cluster
```

Essential Commands

`kubectl get <resource>`	List resources
`kubectl describe <resource> <name>`	Detailed resource info
`kubectl create -f file.yaml`	Create resource from file
`kubectl apply -f file.yaml`	Create or update resource
`kubectl delete -f file.yaml`	Delete resource from file
`kubectl edit <resource> <name>`	Edit resource in-place
`kubectl api-resources`	List all resource types

Output Formats

`-o wide`	Extra columns (IP, node)
`-o yaml`	Full YAML output
`-o json`	Full JSON output
`-o jsonpath='{.spec}'`	Extract specific fields
`--sort-by=.metadata.name`	Sort output by field

Pods

Pod Operations

```
kubectl get pods
kubectl get pods -A                    # all namespaces
kubectl run nginx --image=nginx        # quick pod
kubectl delete pod nginx
```

Pod YAML

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  labels: { app: myapp }
spec:
  containers:
    - name: app
      image: nginx:1.27
      ports:
        - containerPort: 80
```

Pod Status Values

`Running`	All containers started
`Pending`	Waiting for scheduling or image pull
`CrashLoopBackOff`	Container keeps crashing and restarting
`ImagePullBackOff`	Cannot pull container image
`Completed`	Ran to completion (Jobs)

Deployments

Deployment YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
spec:
  replicas: 3
  selector:
    matchLabels: { app: web }
  template:
    metadata:
      labels: { app: web }
    spec:
      containers:
        - name: web
          image: nginx:1.27
          ports:
            - containerPort: 80
```

Deployment Commands

`kubectl get deploy`	List deployments
`kubectl scale deploy web --replicas=5`	Scale replicas
`kubectl set image deploy/web web=nginx:1.28`	Update image (rolling)
`kubectl rollout status deploy/web`	Watch rollout progress
`kubectl rollout undo deploy/web`	Rollback to previous revision
`kubectl rollout history deploy/web`	View revision history

Services

Service Types

`ClusterIP`	Internal only (default)
`NodePort`	Expose on each node's IP at a static port
`LoadBalancer`	External load balancer (cloud)
`ExternalName`	DNS alias to external service

Service YAML

```
apiVersion: v1
kind: Service
metadata:
  name: web-svc
spec:
  type: ClusterIP
  selector: { app: web }
  ports:
    - port: 80
      targetPort: 80
```

Quick Expose

```
kubectl expose deploy web --port=80 --type=ClusterIP
kubectl expose deploy web --port=80 --type=NodePort
kubectl get svc
```

ConfigMaps & Secrets

ConfigMap

```
kubectl create configmap app-cfg \
  --from-literal=DB_HOST=db.example.com \
  --from-file=config.ini
```

Secret

```
kubectl create secret generic db-creds \
  --from-literal=username=admin \
  --from-literal=password=s3cret
```

Using in Pods

```
# As environment variables
envFrom:
  - configMapRef: { name: app-cfg }
  - secretRef:    { name: db-creds }

# As volume mount
volumes:
  - name: cfg
    configMap: { name: app-cfg }
```

Commands

`kubectl get cm`	List ConfigMaps
`kubectl get secret`	List Secrets
`kubectl describe cm app-cfg`	Show ConfigMap data
`kubectl get secret db-creds -o yaml`	Show Secret (base64-encoded)

Namespaces

Namespace Commands

`kubectl get ns`	List namespaces
`kubectl create ns staging`	Create namespace
`kubectl delete ns staging`	Delete namespace and all resources
`kubectl get pods -n staging`	List pods in namespace
`kubectl get pods -A`	List pods across all namespaces

Set Default Namespace

```
kubectl config set-context --current \
  --namespace=staging
```

Volumes

PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: data-pvc
spec:
  accessModes: [ReadWriteOnce]
  resources:
    requests: { storage: 10Gi }
```

Mount in Pod

```
volumes:
  - name: data
    persistentVolumeClaim:
      claimName: data-pvc
containers:
  - volumeMounts:
      - name: data
        mountPath: /app/data
```

Volume Types

`emptyDir`	Temp dir, deleted with pod
`hostPath`	Mount host filesystem path
`persistentVolumeClaim`	Persistent storage (PVC)
`configMap`	Mount ConfigMap as files
`secret`	Mount Secret as files

Ingress

Ingress YAML

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-ingress
spec:
  rules:
    - host: app.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: web-svc
                port: { number: 80 }
```

Ingress Notes

`Ingress Controller`	Required (nginx-ingress, traefik, etc.)
`pathType: Prefix`	Match URL prefix
`pathType: Exact`	Match exact URL path
`TLS`	Add `tls:` section with secret name

Debugging

Diagnostic Commands

`kubectl logs <pod>`	Container stdout/stderr
`kubectl logs <pod> -c <ctr>`	Specific container logs
`kubectl logs <pod> --previous`	Logs from crashed container
`kubectl describe pod <pod>`	Events, conditions, status
`kubectl exec -it <pod> -- sh`	Shell into container
`kubectl port-forward <pod> 8080:80`	Forward local port to pod
`kubectl top pods`	CPU/memory usage (metrics-server)
`kubectl get events --sort-by=.lastTimestamp`	Cluster events timeline

Debug Pod

```
kubectl run debug --rm -it --image=busybox -- sh
# or attach ephemeral container
kubectl debug -it <pod> --image=busybox
```

Common Patterns

Labels & Selectors

```
kubectl get pods -l app=web
kubectl get pods -l 'env in (prod,staging)'
kubectl label pod myapp env=prod
```

Resource Limits

```
resources:
  requests: { cpu: 100m, memory: 128Mi }
  limits:   { cpu: 500m, memory: 256Mi }
```

Liveness & Readiness

```
livenessProbe:
  httpGet: { path: /healthz, port: 8080 }
  initialDelaySeconds: 5
  periodSeconds: 10
readinessProbe:
  httpGet: { path: /ready, port: 8080 }
```

Quick Recipes

`Dry run`	`kubectl apply -f file.yaml --dry-run=client`
`Generate YAML`	`kubectl create deploy web --image=nginx --dry-run=client -o yaml`
`Watch`	`kubectl get pods -w`
`Copy files`	`kubectl cp file.txt pod:/tmp/`
`Restart deploy`	`kubectl rollout restart deploy/web`