GENERAL COMMANDS

```
k get all                                          # List all major resources
k get all -A                                       # List all across namespaces
k get <resource>                                   # List resource (pods, svc, deploy)
k get <resource> -o wide                           # Detailed view
k describe <resource> <name>                       # Detailed info + events
k logs <pod>                                       # Pod logs
k logs -f <pod>                                    # Follow logs
k exec -it <pod> -- sh                             # Shell inside pod
k apply -f file.yaml                               # Apply config (create/update)
k create -f file.yaml                              # Create from file
k edit <resource> <name>                           # Edit live resource
k delete <resource> <name>                         # Delete resource
k top pod                                          # Pod metrics
k top node                                         # Node metrics
```

DEPLOYMENT COMMANDS

```
k get deploy                                       # List deployments
k rollout status deploy/<name>                     # Check rollout progress
k rollout undo deploy/<name>                       # Rollback deployment
k set image deploy/<name> <c>=<image>              # Update image
k scale deploy/<name> --replicas=N                 # Scale replicas
k autoscale deploy <name> --max=10                 # Create HPA
```

NETWORKING COMMANDS (Service, Ingress, Port-Forward)

```
k get svc                                          # List services
k expose deploy <name> --port=80 --type=NodePort   # Expose deployment
k port-forward pod/<name> 8080:80                  # Local → pod port
k get ingress                                      # List ingress
```

CONFIGMAP, SECRET & CONFIGURATION

```
k get cm                                           # List ConfigMaps
k get secrets                                      # List Secrets
k create cm <name> --from-literal=k=v              # Create ConfigMap
k create secret generic <name> --from-literal=k=v  # Create Secret
```

CLUSTER / CONTEXT COMMANDS

```
k get nodes                                        # List nodes
k get ns                                           # List namespaces
k create ns <name>                                 # Create namespace
k config get-contexts                              # List contexts
k config current-context                           # Current context
k config use-context <name>                        # Switch context
k cluster-info                                     # Cluster endpoints
```

JOBS & CRONJOBS

```
k get jobs                                      # List jobs
k create -f job.yaml                            # Create job
k delete job <name>                             # Delete job
k get pods --selector=job-name=<name>           # Pods from job
k logs <pod>                                    # Job pod logs
k get cronjobs                                  # List CronJobs
k create -f cronjob.yaml                        # Create CronJob
```

STORAGE (PV / PVC / SC)

```
k get pv                                        # List PVs
k get pvc                                       # List PVCs
k get sc                                        # List StorageClasses
k describe pv <name>                             # PV details
k describe pvc <name>                            # PVC details
k create -f pv.yaml                              # Create PV
k create -f pvc.yaml                             # Create PVC
k delete pv <name>                               # Delete PV
k delete pvc <name>                              # Delete PVC
```

RBAC (Roles, SAs, Bindings)

```
k get clusterroles                              # List ClusterRoles
k get clusterrolebindings                        # List ClusterRoleBindings
k get roles                                     # List Roles
k get rolebindings                              # List RoleBindings
k get sa                                        # List ServiceAccounts
k create -f rbac.yaml                           # Create RBAC object
k auth can-i get pods --as=system:serviceaccount:ns:sa  # Access check
```

CRD (Custom Resource Definitions)

```
k get crds                                      # List CRDs
k describe crd <name>                            # CRD schema
k create -f crd.yaml                             # Create CRD
k get <custom-resource>                          # List CR instances
k apply -f custom-resource.yaml                  # Create/update CR
```

NODE SCHEDULING (Taints, Cordon, Drain)

```
k describe node <name>                           # Node details
k taint node <name> key=value:effect             # Add taint
k taint node <name> key:effect-                  # Remove taint
k get pods -o yaml | grep tolerations -A 5       # Check tolerations
k cordon <name>                                  # Make unschedulable
k uncordon <name>                                # Make schedulable
k drain <name>                                   # Evict pods
```

NETWORK POLICY (NetPol)

```
k get netpol                                    # List netpols
k describe netpol <name>                        # Netpol details
k create -f policy.yaml                         # Create NetworkPolicy
k delete netpol <name>                          # Delete NetPol
```

TROUBLESHOOTING & HEALTH CHECK

```
k get events                                    # Recent events
k get events --sort-by=.lastTimestamp           # Sorted events
k get events --field-selector type=Warning      # Only warnings
k describe <resource> <name>                    # Detailed troubleshooting
k logs --since=1h <pod>                         # Logs from last 1 hour
k logs --previous <pod>                         # Previous container logs
k top pod                                       # Pod CPU/Memory
k top node                                      # Node CPU/Memory
k version                                       # Client/Server versions
k api-resources                                 # List API resources
```

DEBUGGING UTILITIES

```
k exec -it <pod> -- sh                          # Shell inside pod
k exec <pod> -- curl <svc>                      # In-pod network test
k port-forward <pod> 8080:80                    # Access pod locally
k debug -it <pod> --image=busybox               # Ephemeral debug container
k cp /local/file <pod>:/path                    # Copy to pod
k cp <pod>:/path /local/file                    # Copy from pod
```

MISCELLANEOUS UTILITIES

```
k config view                                   # Show kubeconfig
k explain pod.spec.containers                   # API docs
k api-resources --namespaced=true               # Namespaced resources
k api-resources --namespaced=false              # Cluster-scoped resources
k label node <name> disk=ssd                    # Add label
k label pod <name> app-                         # Remove label
k annotate pod <name> key=value                 # Add annotation
k get pod -l app=frontend                       # List by label selector
```