

Kubernetes (kubectl)

`kubectl` talks to a cluster's API server. Almost every command takes `-n <namespace>` (or `-A` for all namespaces) — forgetting it is the number-one reason "my pod isn't there." Set a default namespace to save the typing.

Inspect Resources

Command	Shows
<code>kubectl get pods</code>	pods in the current namespace
<code>kubectl get pods -A</code>	pods across all namespaces
<code>kubectl get pods -o wide</code>	add node & IP columns
<code>kubectl get all</code>	pods, services, deployments, etc.
<code>kubectl describe pod <name></code>	events + full detail for one object
<code>kubectl get pod <name> -o yaml</code>	the live manifest

Context & Namespace

Command	Does
<code>kubectl config get-contexts</code>	list clusters you can talk to
<code>kubectl config current-context</code>	which cluster am I on?
<code>kubectl config use-context <name></code>	switch clusters
<code>kubectl get pods -n kube-system</code>	target one namespace
<code>kubectl config set-context --current --namespace=dev</code>	set a default namespace

Know your context before you act. The same `kubectl delete` runs against whatever context is current — including prod. `kubectx / kubens` (or a shell prompt that shows the context) make the active target impossible to miss.

Apply & Manage

Command	Does
<code>kubectl apply -f app.yaml</code>	create or update from a manifest
<code>kubectl delete -f app.yaml</code>	delete what the manifest defines
<code>kubectl edit deploy <name></code>	edit a live object in \$EDITOR
<code>kubectl scale deploy <name> --replicas=3</code>	change replica count
<code>kubectl rollout restart deploy <name></code>	restart pods (e.g. after a config change)
<code>kubectl rollout status deploy <name></code>	watch a rollout finish
<code>kubectl rollout undo deploy <name></code>	roll back to the previous version

Debugging

Command	Does
<code>kubectl logs <pod></code>	print a pod's logs
<code>kubectl logs -f <pod></code>	follow logs live
<code>kubectl logs <pod> --previous</code>	logs from the crashed instance
<code>kubectl exec -it <pod> -- sh</code>	shell into a running pod
<code>kubectl port-forward <pod> 8080:80</code>	tunnel a local port to the pod
<code>kubectl get events --sort-by=.lastTimestamp</code>	recent cluster events

Nodes & Cluster

Command	Shows / does
<code>kubectl get nodes</code>	cluster nodes and their status
<code>kubectl top nodes</code>	live CPU / memory per node (needs metrics-server)
<code>kubectl top pods</code>	resource use per pod
<code>kubectl cluster-info</code>	control-plane endpoints
<code>kubectl cordon <node></code>	stop scheduling new pods onto a node
<code>kubectl drain <node></code>	evict pods to prep a node for maintenance