Kubectl CLI

Sending commands to Kubernetes

What is `kubectl`?

- Command-line tool to interact with a Kubernetes cluster
- Lets you manage:
  - Pods, Deployments, Services, etc.
  - Logs and container access
  - Namespaces and contexts (clusters/users)

Basic Syntax

```
kubectl [command] [TYPE] [NAME] [flags]
```

- `command`: What to do (e.g. `get`, `create`, `delete`)
- `TYPE`: Resource type (e.g. `pod`, `svc`, `deployment`)
- `NAME`: Resource name (optional)
- `flags`: Additional options (e.g. `--namespace`, `--output`, `--context`)

Common Commands

🔍 Viewing Resources

```
kubectl get pods
kubectl get svc
kubectl get deployments
```

Add `-A` to show all namespaces:

```
kubectl get pods -A
```

📝 Creating Resources

```
kubectl apply -f app.yaml
kubectl create deployment nginx --image=nginx
```

🔧 Updating/Editing

```
kubectl edit deployment my-deployment
```

🗑️ Deleting Resources

```
kubectl delete pod my-pod
kubectl delete -f app.yaml
```

📋 Describing Resources

```
kubectl describe pod my-pod
```

📋 Apply a partial update to a resource

```
kubectl patch deployment my-app -p '{"spec":{"replicas":3}}'
```

Use Case: Quick field-level updates without full YAML

Logs & Exec

```
kubectl logs my-pod
kubectl exec -it my-pod -- /bin/bash
```

Output Customization

Format Options

```
kubectl get pods -o wide
kubectl get pods -o yaml
kubectl get pods -o json
```

Filtering by Labels

```
kubectl get pods -l app=nginx
```

Filtering by Fields

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

Working with Namespaces

Create and Use Namespaces

```
kubectl create namespace dev
kubectl get pods --namespace=dev
```

Set Default Namespace

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

Context Switching

View Current Context

```
kubectl config current-context
```

List All Contexts

```
kubectl config get-contexts
```

Switch Context

```
kubectl config use-context my-context
```

Create Custom Context

```
kubectl config set-context my-context \
  --cluster=my-cluster \
  --user=my-user \
  --namespace=default
```

Set Namespace for a Context

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

Helpful Tips

- Use `kubectl explain <resource>` to get resource documentation
- Use `watch` for real-time monitoring:

```
watch kubectl get pods
```

- Create an alias for convenience:

```
alias k=kubectl
```

✅ Summary

- `kubectl` is your primary tool for managing Kubernetes
- Understand core commands: get, apply, delete, describe, exec
- Customize with flags like `--namespace`, `-o yaml`, `--context`
- Manage access across clusters with contexts