



# Kubernetes Commands

Kubernetes (K8s) is a powerful container orchestration platform that helps automate the deployment, scaling, and management of containerized applications. Here is a list of 50 important Kubernetes commands along with brief explanations and examples:

## 1. **kubectl version**

- Display the Kubernetes client and server version.

```
kubectl version
```

## 2. **kubectl cluster-info**

- Display cluster information, including the master and services.

```
kubectl cluster-info
```

## 3. **kubectl get nodes**

- List all nodes in the cluster.

```
kubectl get nodes
```

## 4. **kubectl get pods**

- List all pods in the default namespace.

```
kubectl get pods
```

## 5. **kubectl get deployments**

- List all deployments in the default namespace.

```
kubectl get deployments
```

## 6. **kubectl describe pod [pod\_name]**

- Display detailed information about a specific pod.

```
kubectl describe pod mypod
```

## 7. **kubectl logs [pod\_name]**

- Display the logs of a specific pod.

```
kubectl logs mypod
```

## 8. **kubectl exec -it [pod\_name] -- /bin/sh**

- Start an interactive shell in a specific pod.

```
kubectl exec -it mypod -- /bin/sh
```

## 9. **kubectl create deployment [name] --image=[image]**

- Create a deployment with a specified container image.

```
kubectl create deployment myapp --image=myimage:tag
```

## 10. **kubectl expose deployment [name] --port=[port] --type=LoadBalancer**

- Expose a deployment as a service.

```
kubectl expose deployment myapp --port=80 --type=LoadBalancer
```

## 11. **kubectl scale deployment [name] --replicas=[replica\_count]**

- Scale the number of replicas for a deployment.

```
kubectl scale deployment myapp --replicas=3
```

## 12. **kubectl get svc**

- List all services in the default namespace.

```
kubectl get svc
```

### 13. **kubectl delete pod [pod\_name]**

- Delete a specific pod.

```
kubectl delete pod mypod
```

### 14. **kubectl delete deployment [name]**

- Delete a deployment and its associated pods.

```
kubectl delete deployment myapp
```

### 15. **kubectl apply -f [file]**

- Apply a configuration file to the cluster.

```
kubectl apply -f myconfig.yaml
```

### 16. **kubectl get configmaps**

- List all ConfigMaps in the default namespace.

```
kubectl get configmaps
```

### 17. **kubectl describe service [service\_name]**

- Display detailed information about a specific service.

```
kubectl describe service myservice
```

### 18. **kubectl get namespaces**

- List all namespaces in the cluster.

```
kubectl get namespaces
```

### 19. **kubectl create namespace [namespace\_name]**

- Create a new namespace.

```
kubectl create namespace mynamespace
```

## 20. **kubectl get pods -n [namespace]**

- List all pods in a specific namespace.

```
kubectl get pods -n mynamespace
```

## 21. **kubectl delete namespace [namespace\_name]**

- Delete a namespace and all its resources.

```
kubectl delete namespace mynamespace
```

## 22. **kubectl get services --sort-by=.metadata.name**

- List services and sort them by name.

```
kubectl get services --sort-by=.metadata.name
```

## 23. **kubectl rollout status deployment [deployment\_name]**

- Check the status of a deployment rollout.

```
kubectl rollout status deployment myapp
```

## 24. **kubectl get pods --field-selector=status.phase=Running**

- List pods that are in the Running phase.

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

## 25. **kubectl get events --sort-by=.metadata.creationTimestamp**

- List events sorted by creation timestamp.

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

## 26. **kubectl create secret generic [secret\_name] --from-literal=[key]=[value]**

- Create a generic secret from literal values.

```
kubectl create secret generic mysecret --from-literal=username=admin --  
from-literal=password=pass123
```

## 27. **kubectl get secrets**

- List all secrets in the default namespace.

```
kubectl get secrets
```

## 28. **kubectl describe secret [secret\_name]**

- Display detailed information about a specific secret.

```
kubectl describe secret mysecret
```

## 29. **kubectl edit deployment [deployment\_name]**

- Edit the YAML of a deployment interactively.

```
kubectl edit deployment myapp
```

## 30. **kubectl get pods -o wide**

- List pods with additional details like node information.

```
kubectl get pods -o wide
```

## 31. **kubectl get nodes -o custom-columns=NODE:.metadata.name,IP:.status.addresses[0].address**

- List nodes with custom output columns.

```
kubectl get nodes -o custom-  
columns=NODE:.metadata.name,IP:.status.addresses[0].address
```

## 32. **kubectl top pods**

- Display resource usage (CPU and memory) of pods.

```
kubectl top pods
```

### 33. **kubectl apply -f <https://url-to-yaml-file>**

- Apply a configuration file directly from a URL.

```
kubectl apply -f https://raw.githubusercontent.com/example/myconfig.yaml
```

### 34. **kubectl get pods --selector=[label\_key]=[label\_value]**

- List pods with a specific label.

```
kubectl get pods --selector=app=myapp
```

### 35. **kubectl get pods --field-selector=status.phase!=Running**

- List pods that are not in the Running phase.

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

### 36. **kubectl rollout undo deployment [deployment\_name]**

- Rollback a deployment to the previous version.

```
kubectl rollout undo deployment myapp
```

### 37. **kubectl label pod [pod\_name] [label\_key]=[label\_value]**

- Add a label to a specific pod.

```
kubectl label pod mypod environment=production
```

### 38. **kubectl get componentstatuses**

- List the health of different cluster components.

```
kubectl get componentstatuses
```

### 39. **kubectl describe node [node\_name]**

Display detailed information about a specific node.

```
kubectl describe node mynode
```

#### 40. **kubectl rollout history deployment [deployment\_name]**

- View the rollout history of a deployment.

```
kubectl rollout history deployment myapp
```

#### 41. **kubectl delete pod --selector=[label\_key]=[label\_value]**

- Delete pods with a specific label.

```
kubectl delete pod --selector=app=myapp
```

#### 42. **kubectl top nodes**

- Display resource usage (CPU and memory) of nodes.

```
kubectl top nodes
```

#### 43. **kubectl get pods --watch**

- Watch for changes to pods in real-time.

```
kubectl get pods -watch
```

#### 44. **kubectl rollout pause deployment [deployment\_name]**

- Pause a deployment to prevent further rollouts.

```
kubectl rollout pause deployment myapp
```

#### 45. **kubectl rollout resume deployment [deployment\_name]**

- Resume a paused deployment.

```
kubectl rollout resume deployment myapp
```

#### 46. **kubectl explain [resource]**

- Get information about a Kubernetes resource.

```
kubectl explain pod
```

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

- Extract specific information using JSONPath.

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

#### 48. **kubectl apply --dry-run=client -f [file]**

- Dry run to validate a configuration file without applying it.

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

#### 49. **kubectl exec -it [pod\_name] -- /bin/sh -c 'command'**

- Execute a command in a specific pod.

```
kubectl exec -it mypod -- /bin/sh -c 'ls /app'
```

#### 50. **kubectl get events --sort-by=.metadata.creationTimestamp -n [namespace]**

- List events sorted by creation timestamp in a specific namespace.

```
kubectl get events --sort-by=.metadata.creationTimestamp -n mynamespace
```

These commands cover a broad range of Kubernetes operations and are useful for managing and troubleshooting applications in a Kubernetes cluster. Note that some commands may require specific roles or permissions depending on your Kubernetes environment.