DevOps Shack

TOP 30 Kubernetes Commands Asked in MNC Interviews

1. How do you check the version of Kubernetes installed?

Command:

```
kubectl version --short
```

Explanation:

- **This command displays the client and server versions of Kubernetes.**
- **The `--short` flag ensures a concise output.**
- **It helps verify compatibility between your `kubectl` version and the cluster's API version.**

2. How do you list all pods running in a specific namespace?

Command:

```
kubectl get pods -n <namespace>
```

Explanation:

- **`kubectl get pods`: Fetches all pods.**
- **`-n <namespace>`: Restricts the query to a specific namespace.**
- **If no namespace is specified, it defaults to the `default` namespace.**

Example:

```
kubectl get pods -n kube-system
```

- **This is crucial in troubleshooting since pods in different namespaces might have different statuses.**

3. How do you get detailed information about a specific pod?

Command:

```
kubectl describe pod <pod-name> -n <namespace>
```

Explanation:

- **Provides detailed information about the pod, such as:**
  - **Labels, annotations.**
  - **Events (like image pull errors, crash loops).**
  - **Containers and their statuses.**
- **Useful for debugging issues related to pods.**

Example:

```
kubectl describe pod nginx-pod -n default
```

4. How do you access the logs of a pod?

Command:

```
kubectl logs <pod-name> -n <namespace>
```

Explanation:

- Fetches logs of the primary container in a pod.

For multi-container pods, specify the container:

```
kubectl logs <pod-name> -c <container-name>
```

- Add the `--tail` or `--since` flags to limit the logs.

Example:

```
kubectl logs nginx-pod -n default
```

5. How do you execute a command inside a running pod?

Command:

```
kubectl exec -it <pod-name> -n <namespace> -- <command>
```

Explanation:

- `-it`: Interactive terminal mode.
- `--`: Separates `kubectl` options from the command being executed.

Example (access a shell):

```
kubectl exec -it nginx-pod -n default -- /bin/
```

- Often used to debug containers in real time.

6. How do you expose a deployment as a service?

Command:

```
kubectl expose deployment <deployment-name>
--type=<service-type> --port=<port>
```

Explanation:

- **Creates a service to expose the deployment externally or internally.**
- **Service types:**
  - `ClusterIP` **(default): Internal access within the cluster.**
  - `NodePort`**: Exposes the service on a static port on each node.**
  - `LoadBalancer`**: Creates an external load balancer.**

Example:

```
kubectl expose deployment nginx-deployment --type=NodePort
--port=80
```

7. How do you scale a deployment?

Command:

```
kubectl scale deployment <deployment-name>
--replicas=<number>
```

Explanation:

- **Adjusts the number of pod replicas for a deployment.**
- **Ensures high availability or resource optimization.**

Example:

```
kubectl scale deployment nginx-deployment --replicas=5
```

- **Use** `kubectl get pods` **to verify new pods are created.**

8. How do you create a resource from a YAML file?

Command:

```
kubectl apply -f <file.yaml>
```

Explanation:

- **Applies configurations defined in the YAML file.**
- **If the resource exists, it updates it; otherwise, it creates it.**

Example:

```
kubectl apply -f deployment.yaml
```

- **Commonly used for infrastructure as code (IaC) practices.**

9. How do you delete a resource in Kubernetes?

Command:

```
kubectl delete <resource-type> <resource-name>
```

Explanation:

- **Deletes a resource, such as a pod, deployment, or service.**

To delete all resources of a type:

```
kubectl delete <resource-type> --all
```

Example:

```
kubectl delete pod nginx-pod
```

10. How do you view cluster-wide resources like nodes?

Command:

```
kubectl get nodes
```

Explanation:

- Lists all worker and master nodes in the cluster.
- Shows node statuses (`Ready`, `NotReady`).

Example:

```
NAME              STATUS   ROLES    AGE   VERSION
ip-192-168-1-1    Ready    master   30d   v1.23.3
```

11. How do you debug a node or pod issue?

Commands:

```
# Check node status
kubectl describe node <node-name>

# Check pod events
kubectl describe pod <pod-name>
```

Explanation:

- Node-level issues (e.g., CPU, memory) can affect pods.
- Pod-level events (e.g., image pull errors, restarts) give insights into issues.

12. How do you view resource utilization in the cluster?

Command:

```
kubectl top nodes
kubectl top pods
```

Explanation:

- **Displays CPU and memory utilization of nodes and pods.**
- **Requires the Metrics Server to be installed in the cluster.**

Example:

```
kubectl top pods -n default
```

13. How do you get all resources in a namespace?

Command:

```
kubectl get all -n <namespace>
```

Explanation:

- **Lists all resources (pods, services, deployments, etc.) in a namespace.**
- **Useful for gaining a holistic view of the namespace.**

14. How do you update a Kubernetes resource?

Command:

```
kubectl edit <resource-type> <resource-name>
```

Explanation:

- **Opens the resource definition in the default editor (e.g., `vim`).**
- **On save, Kubernetes updates the resource.**

Example:

```
kubectl edit deployment nginx-deployment
```

15. How do you get the configuration of an existing resource as YAML?

Command:

```
kubectl get <resource-type> <resource-name> -o yaml
```

Explanation:

- **Outputs the resource's current configuration in YAML format.**

Example:

```
kubectl get pod nginx-pod -o yaml
```

16. How do you create a namespace?

Command:

```
kubectl create namespace <namespace-name>
```

Explanation:

- **Creates a logical isolation within the cluster.**
- **Namespaces are useful for organizing resources by environment (e.g., Dev, QA, Prod).**

Example:

```
kubectl create namespace dev
```

17. How do you delete a namespace?

Command:

```
kubectl delete namespace <namespace-name>
```

Explanation:

- **Deletes the namespace and all resources within it.**
- **Be cautious, as this action is irreversible.**

Example:

```
kubectl delete namespace dev
```

18. How do you apply a rollout for a deployment update?

Command:

```
kubectl rollout restart deployment <deployment-name>
```

Explanation:

- **Triggers a rolling update for the specified deployment.**
- **Ensures zero-downtime updates by replacing pods one at a time.**

Example:

```
kubectl rollout restart deployment nginx-deployment
```

19. How do you check the rollout status of a deployment?

Command:

```
kubectl rollout status deployment <deployment-name>
```

Explanation:

- **Monitors the progress of a rolling update.**
- **Useful to ensure all pods are updated without errors.**

Example:

```
kubectl rollout status deployment nginx-deployment
```

20. How do you undo a deployment to a previous revision?

Command:

```
kubectl rollout undo deployment <deployment-name>
```

Explanation:

- **Rolls back the deployment to its last stable revision.**
- **Use `--to-revision=<number>` to specify a particular revision.**

Example:

```
kubectl rollout undo deployment nginx-deployment
```

21. How do you view the history of a deployment?

Command:

```
kubectl rollout history deployment <deployment-name>
```

Explanation:

- **Lists all revisions of a deployment along with details.**

Example:

```
kubectl rollout history deployment nginx-deployment
```

22. How do you port-forward a service or pod to access it locally?

Command:

```
kubectl port-forward <pod-or-service-name>
<local-port>:<target-port>
```

Explanation:

- **Maps a local port to a pod or service for testing/debugging.**

Example:

```
kubectl port-forward pod/nginx-pod 8080:80
```

23. How do you expose the Kubernetes dashboard?

Command:

```
kubectl proxy
```

Explanation:

- **Starts a local proxy to access cluster services like the dashboard.**
- **Accessible at `http://localhost:8001`.**
- **Requires the Kubernetes Dashboard to be installed.**

24. How do you create a ConfigMap?

Command:

```
kubectl create configmap <configmap-name>
--from-literal=<key>=<value>
```

Explanation:

- ConfigMaps store configuration data as key-value pairs.

Example:

```
kubectl create configmap app-config
--from-literal=environment=dev
```

25. How do you create a Secret?

Command:

```
kubectl create secret generic <secret-name>
--from-literal=<key>=<value>
```

Explanation:

- Secrets store sensitive data like passwords, tokens, or keys.
- Data is base64 encoded.

Example:

```
kubectl create secret generic db-secret
--from-literal=username=admin
--from-literal=password=securepass
```

26. How do you attach a ConfigMap or Secret to a Pod?

Command: Add it to the Pod YAML file.

yaml

```
envFrom:
  - configMapRef:
      name: <configmap-name>
```

```
  - secretRef:
      name: <secret-name>
```

Explanation:

- Use `configMapRef` and `secretRef` in the Pod's environment variables.

Example:
yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: app
spec:
  containers:
    - name: app-container
      image: nginx
      envFrom:
        - configMapRef:
            name: app-config
        - secretRef:
            name: db-secret
```

27. How do you drain a node before maintenance?

Command:

```
kubectl drain <node-name> --ignore-daemonsets
--delete-emptydir-data
```

Explanation:

- Evicts all pods from the node to make it ready for maintenance.
- The `--ignore-daemonsets` flag skips DaemonSets.

Example:

```
kubectl drain worker-node-1 --ignore-daemonsets
--delete-emptydir-data
```

28. How do you uncordon a node?

Command:

```
kubectl uncordon <node-name>
```

Explanation:

- Marks a node as schedulable after maintenance.

Example:

```
kubectl uncordon worker-node-1
```

29. How do you debug a CrashLoopBackOff issue?

Commands:

```
# View logs
kubectl logs <pod-name>
# Describe the pod
kubectl describe pod <pod-name>
# Check the events
kubectl get events -n <namespace>
```

Explanation:

- `CrashLoopBackOff` indicates the container is repeatedly failing.
- Use logs and events to identify the root cause (e.g., missing ConfigMap, secret, or incorrect image).

30. How do you check the endpoints of a service?

Command:

```
kubectl get endpoints <service-name>
```

Explanation:

- Lists the IP addresses of pods backing the service.
- Helps ensure that the service is routing traffic correctly.

Example:

```
kubectl get endpoints nginx-service
```