DevOps Shack

Kubernetes Scenario-based Interview Questions

Scenario: You have a pod named web-server running in the production namespace. You need to find out the IP address of this pod.

Answer: `kubectl get pod web-server -n production -o=jsonpath='{.status.podIP}'`

Scenario: You need to scale a deployment named nginx-deployment to 5 replicas.

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

Scenario: You want to expose a service named nginx-service on port 80 using NodePort.

Answer: `kubectl expose service nginx-service --type=NodePort --port=80`

Scenario: A pod named database is stuck in Pending state. How do you troubleshoot it?

Answer: `kubectl describe pod database`

Scenario: You need to update the image of a deployment named nginx-deployment to nginx:latest.

Answer: `kubectl set image deployment/nginx-deployment nginx=nginx:latest`

Scenario: You have a pod named nginx with the label app=web. You need to delete this pod.

Answer: `kubectl delete pod -l app=web`

Scenario: You want to view the logs of a container named nginx in a pod named nginx-pod.

Answer: `kubectl logs nginx-pod -c nginx`

Scenario: You have a deployment named app-deployment and you want to check the rollout status.

Answer: `kubectl rollout status deployment/app-deployment`

Scenario: You want to create a namespace named development.

Answer: `kubectl create namespace development`

Scenario: You need to execute a command inside a running container named nginx in a pod named nginx-pod.

Answer: `kubectl exec -it nginx-pod -- /bin/bash`

Scenario: You need to create a persistent volume named pv-data with 1Gi capacity.

Answer: `kubectl apply -f - <<EOF`

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-data
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  hostPath:
    path: /mnt/data
EOF
```

Scenario: You want to schedule a pod named nginx on a specific node named node-1.

Answer: `kubectl patch pod nginx -p '{"spec": {"nodeName": "node-1"}}'`

Scenario: You have a deployment named app-deployment and you want to rollback to the previous version.

Answer: `kubectl rollout undo deployment/app-deployment`

Scenario: You need to create a secret named db-secret with a username and password.

Answer: `kubectl create secret generic db-secret --from-literal=username=myuser --from-literal=password=mypassword`

Scenario: You want to list all the pods in the cluster along with their IP addresses.

Answer: `kubectl get pods -o wide`

Scenario: You need to label a node named node-1 with environment=production.

Answer: `kubectl label node node-1 environment=production`

Scenario: You want to create a deployment named nginx-deployment with 3 replicas using a YAML file.

Answer (create nginx-deployment.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Apply the YAML file:

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

Scenario: You need to add a label app=nginx to a pod named nginx-pod.

Answer: `kubectl label pod nginx-pod app=nginx`

Scenario: You have a service named nginx-service and you want to delete it.

Answer: `kubectl delete service nginx-service`

Scenario: You need to list all the persistent volumes in the cluster.

Answer: `kubectl get pv`

Scenario: You want to check the resource utilization of pods in the production namespace.

Answer: `kubectl top pods -n production`

Scenario: You need to delete all the pods in the development namespace.

Answer: `kubectl delete pods --all -n development`

Scenario: You want to view the YAML definition of a pod named nginx-pod.

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

Scenario: You need to create a service named nginx-service to expose port 80 on TCP.

Answer: `kubectl create service tcp nginx-service --tcp=80:80`

Scenario: You have a pod named nginx and you want to change its resource requests to 0.5 CPU and 512Mi memory.

Answer: `kubectl set resources pod nginx --requests=cpu=0.5,memory=512Mi`

Scenario: You need to create a role named nginx-reader with read-only access to pods in the default namespace.

Answer (create nginx-reader-role.yaml):

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: default
  name: nginx-reader
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

Apply the YAML file:

```
kubectl apply -f nginx-reader-role.yaml
```

Scenario: You want to check the status of all the nodes in the cluster.

Answer: `kubectl get nodes`

Scenario: You need to check the version of Kubernetes server.

Answer: `kubectl version`

Scenario: You want to check the details of a specific event in the cluster.

Answer: `kubectl describe event <event-name>`

Scenario: You have a deployment named app-deployment and you want to update its environment variable DB_HOST to db-server.

Answer: `kubectl set env deployment/app-deployment DB_HOST=db-server`

Scenario: You want to drain a node named node-1 for maintenance.

Answer:

```
kubectl drain node-1 --ignore-daemonsets
```

Scenario: You have a deployment named app-deployment and you want to expose it as a service named app-service on port 8080.

Answer: `kubectl expose deployment app-deployment --name=app-service --port=8080`

Scenario: You need to create a deployment named nginx-deployment with a specific label app

=nginx.

Answer (create nginx-deployment.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Apply the YAML file:

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

Scenario: You want to check the logs of the previous instance of a pod named nginx.

Answer: `kubectl logs nginx –previous`

Scenario: You have a service named nginx-service and you want to find out its external IP.

Answer: `kubectl get svc nginx-service -o=jsonpath='{.status.loadBalancer.ingress[0].ip}'`

Scenario: You need to create a service named nginx-service to expose the nginx pods on port 8080.

Answer: `kubectl expose pod nginx --name=nginx-service --port=8080`

Scenario: You have a deployment named app-deployment and you want to annotate it with description="App Deployment".

Answer: `kubectl annotate deployment app-deployment description="App Deployment"`

Scenario: You need to create a role binding named nginx-read-access to bind the nginx-reader role to a user named john.

Answer (create nginx-read-access.yaml):

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: nginx-read-access
subjects:
- kind: User
  name: john
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: nginx-reader
  apiGroup: rbac.authorization.k8s.io
```

Apply the YAML file:

```
kubectl apply -f nginx-read-access.yaml
```

Scenario: You want to check the status of all the pods in the default namespace.

Answer: `kubectl get pods -n default`

Scenario: You need to create a deployment named nginx-deployment with a specific namespace production.

Answer (create nginx-deployment.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: production
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Apply the YAML file:

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

Scenario: You want to view the logs of the container named nginx in a pod named nginx-pod since a specific timestamp.

Answer: `kubectl logs nginx-pod -c nginx --since-time=<timestamp>`

Scenario: You have a service named nginx-service and you want to update it to use TCP port 8080.

Answer: `kubectl edit svc nginx-service`

Then change the port to 8080 in the opened editor.

Scenario: You need to create a deployment named nginx-deployment with specific resource limits: CPU=0.5 and memory=512Mi.

Answer (create nginx-deployment.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        resources:
          requests:
            cpu: 500m
            memory: 512Mi
```

Apply the YAML file:

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

Scenario: You have a namespace named development and you want to delete all the resources in it.

Answer: `kubectl delete namespace development --grace-period=0 –force`

Scenario: You need to create a deployment named nginx-deployment with specific labels env=prod and app=nginx.

Answer (create nginx-deployment.yaml):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
```

```
    labels:
      env: prod
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx:latest
      ports:
      - containerPort: 80
```

Apply the YAML file:

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

Scenario: You want to view the details of a specific node named node-1.

Answer: `kubectl describe node node-1`

Scenario: You need to create a service account named deployer in the default namespace.

Answer: `kubectl create serviceaccount deployer`

Scenario: You want to check the status of a specific pod named nginx in the production namespace.

Answer: `kubectl get pod nginx -n production`

Scenario: You have a deployment named app-deployment and you want to check the rollout history.

Answer: `kubectl rollout history deployment/app-deployment`

Scenario: You want to check the status of a specific service named nginx-service.

Answer: `kubectl get service nginx-service`