Kubernetes Interview Q&A with Examples

## What is a Namespace?

A namespace in Kubernetes is a way to divide cluster resources between multiple users. It helps isolate environments (like dev, test, prod) within the same cluster.  
  
Example:  
  
kubectl create namespace dev

## .kubeconfig usage

The .kube/config file contains configuration to connect to Kubernetes clusters (credentials, cluster info, contexts).  
  
Location: ~/.kube/config  
  
Example:  
  
contexts:  
- name: dev-context  
 context:  
 cluster: dev-cluster  
 user: dev-user

## Switch the cluster context from CLI

kubectl config use-context dev-context

## Check the Pod Logs

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

## Login to Running Container

kubectl exec -it <pod-name> -- /bin/bash  
kubectl exec -it <pod-name> -c <container-name> -- /bin/sh

## What is PV and PVC?

PV (Persistent Volume): Storage in the cluster.  
PVC (Persistent Volume Claim): Request for storage by user.  
  
Example:  
  
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
 name: mypvc  
spec:  
 accessModes:  
 - ReadWriteOnce  
 resources:  
 requests:  
 storage: 1Gi

## Different Types of Services

ClusterIP: Default, accessible only within cluster.  
NodePort: Exposes service on a static port on each node.  
LoadBalancer: Uses external load balancer.  
ExternalName: Maps service to external DNS.  
  
Example:  
  
apiVersion: v1  
kind: Service  
metadata:  
 name: myservice  
spec:  
 type: NodePort  
 ports:  
 - port: 80  
 targetPort: 8080  
 nodePort: 30080  
 selector:  
 app: myapp

## Configure Application Load Balancer (ALB)

For EKS: Use AWS ALB Ingress Controller.  
  
Example:  
  
apiVersion: networking.k8s.io/v1  
kind: Ingress  
metadata:  
 name: my-ingress  
 annotations:  
 alb.ingress.kubernetes.io/scheme: internet-facing  
spec:  
 rules:  
 - http:  
 paths:  
 - path: /\*  
 pathType: ImplementationSpecific  
 backend:  
 service:  
 name: myservice  
 port:  
 number: 80

## Expose Ports

kubectl expose deployment myapp --type=NodePort --port=80 --target-port=8080

## Port Forwarding

kubectl port-forward svc/myservice 8080:80  
# Now access the service via http://localhost:8080

## EKS Versions Worked On

Worked on EKS versions like 1.21, 1.22, and recently upgraded to 1.24.

## EKS Upgrade Experience

Performed in-place upgrades using AWS Console and CLI. Steps included:  
- Backing up cluster configs  
- Draining nodes  
- Upgrading EKS version  
- Updating node groups and validating workloads post-upgrade

## What are ConfigMaps?

ConfigMaps store non-sensitive key-value pairs (e.g., config settings).  
  
Example:  
  
apiVersion: v1  
kind: ConfigMap  
metadata:  
 name: app-config  
data:  
 LOG\_LEVEL: debug

## Secrets vs ConfigMaps

Secrets store sensitive data (e.g., passwords, tokens).  
  
Example:  
  
apiVersion: v1  
kind: Secret  
metadata:  
 name: db-secret  
type: Opaque  
data:  
 password: cGFzc3dvcmQ=  
  
Difference:  
- ConfigMap: Plaintext data  
- Secret: Encoded/encrypted data for sensitive info

## Role-Based Access Control (RBAC)

RBAC defines permissions for users/groups to perform actions.  
  
Example:  
  
apiVersion: rbac.authorization.k8s.io/v1  
kind: Role  
metadata:  
 namespace: dev  
 name: dev-reader  
rules:  
- apiGroups: []  
 resources: ["pods"]  
 verbs: ["get", "list"]  
---  
apiVersion: rbac.authorization.k8s.io/v1  
kind: RoleBinding  
metadata:  
 name: read-pods  
 namespace: dev  
subjects:  
- kind: User  
 name: alice  
roleRef:  
 kind: Role  
 name: dev-reader  
 apiGroup: rbac.authorization.k8s.io