**KUBERNETES HANDS-ON EXERCISES WITH STEP-BY-STEP COMMANDS AND SOLUTIONS**

**Kubernetes hands-on exercises** with **step-by-step commands and solutions** to help you gain practical experience. These exercises range from basic to intermediate levels.

**Exercise 1: Deploy an Nginx Pod**

**Objective: Deploy a simple Nginx pod and access it.**

**Steps & Commands:**

1. **Start a Kubernetes cluster (Minikube or other cluster):**

minikube start

1. **Create an Nginx pod:**

kubectl run nginx-pod --image=nginx --restart=Never

1. **Verify the pod is running:**

kubectl get pods

1. **Check pod details:**

kubectl describe pod nginx-pod

1. **Delete the pod:**

kubectl delete pod nginx-pod

**Exercise 2: Create an Nginx Deployment and Scale It**

**Objective: Create an Nginx deployment and scale it up.**

**Steps & Commands:**

1. **Create a deployment with Nginx:**

kubectl create deployment nginx-deployment --image=nginx

1. **Check the deployment:**

kubectl get deployments

1. **Scale the deployment to 3 replicas:**

kubectl scale deployment nginx-deployment --replicas=3

1. **Check the running pods:**

kubectl get pods -o wide

1. **Delete the deployment:**

kubectl delete deployment nginx-deployment

**Exercise 3: Create a Pod using a YAML Manifest**

**Objective: Define and deploy a pod using YAML.**

**Steps & Commands:**

1. **Create a YAML file (nginx-pod.yaml):**

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

spec:

containers:

- name: nginx

image: nginx

1. **Apply the YAML file:**

kubectl apply -f nginx-pod.yaml

1. **Check if the pod is running:**

kubectl get pods

1. **Delete the pod using YAML:**

kubectl delete -f nginx-pod.yaml

**Exercise 4: Create and Use a ConfigMap**

**Objective: Store environment variables in a ConfigMap and use it in a pod.**

**Steps & Commands:**

1. **Create a ConfigMap:**

kubectl create configmap app-config --from-literal=APP\_ENV=production

1. **Verify ConfigMap:**

kubectl get configmaps app-config -o yaml

1. **Create a pod that uses the ConfigMap (nginx-config-pod.yaml):**

apiVersion: v1

kind: Pod

metadata:

name: nginx-config-pod

spec:

containers:

- name: nginx

image: nginx

env:

- name: APP\_ENV

valueFrom:

configMapKeyRef:

name: app-config

key: APP\_ENV

1. **Deploy the pod:**

kubectl apply -f nginx-config-pod.yaml

1. **Check if the pod is running:**

kubectl get pods

1. **Delete the pod and ConfigMap:**

kubectl delete -f nginx-config-pod.yaml

kubectl delete configmap app-config

**Exercise 5: Create and Use a Secret**

**Objective: Store sensitive data using Secrets and use it in a pod.**

**Steps & Commands:**

1. **Create a Secret for database credentials:**

kubectl create secret generic db-secret --from-literal=DB\_USER=admin --from-literal=DB\_PASS=password123

1. **Verify the Secret:**

kubectl get secrets

kubectl describe secret db-secret

1. **Create a Pod that uses the Secret (nginx-secret-pod.yaml):**

apiVersion: v1

kind: Pod

metadata:

name: nginx-secret-pod

spec:

containers:

- name: nginx

image: nginx

env:

- name: DB\_USER

valueFrom:

secretKeyRef:

name: db-secret

key: DB\_USER

- name: DB\_PASS

valueFrom:

secretKeyRef:

name: db-secret

key: DB\_PASS

1. **Deploy the pod:**

kubectl apply -f nginx-secret-pod.yaml

1. **Check the pod and logs:**

kubectl get pods

kubectl logs nginx-secret-pod

1. **Delete the pod and Secret:**

kubectl delete -f nginx-secret-pod.yaml

kubectl delete secret db-secret

**Exercise 6: Create and Expose a Service**

**Objective: Deploy an application and expose it using a service.**

**Steps & Commands:**

1. **Create a deployment:**

kubectl create deployment webapp --image=nginx

1. **Expose the deployment using a service:**

kubectl expose deployment webapp --type=NodePort --port=80

1. **Get service details:**

kubectl get svc webapp

1. **Access the service (Minikube users):**

minikube service webapp --url

1. **Delete the service and deployment:**

kubectl delete svc webapp

kubectl delete deployment webapp