**Kubernetes Assignment submission**

Exercise 1: Setting Up Your Kubernetes Cluster

Objective: Set up a local Kubernetes environment using Minikube and kubectl.

1. Install Minikube and kubectl.

2. Start a Minikube cluster with minikube start.

3. Use kubectl cluster-info to verify your cluster is running.

4. List all nodes using kubectl get nodes.

Output Screenshot:

A black screen with white text

AI-generated content may be incorrect.

🔹 Exercise 2: Creating and Managing Pods

Objective: Learn how to create and manage Pods.

1. Create a simple pod using a predefined image like nginx

2. Check the pod status.

3. View pod logs

4. Expose the pod via a service

Output Screenshot:

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Exercise 3: Working with Deployments

Objective: Use Deployments for managing replicated Pods.

1. Create a deployment with using nginx image

2. Scale the deployment to 3 replicas

3. Verify the deployment

4. Update the deployment by changing the image(imperative way)

Output Screenshot:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

🔹 Exercise 4: Services and Networking

Objective: Expose your app using Kubernetes services.

1. Expose your nginx deployment using a Service:

2. Create a service of type NodePort to make it accessible externally.

3. View the service details.

4. Test access to the nginx service through the browser.

Output Screenshot:

A screen shot of a computer

AI-generated content may be incorrect.

🔹 Exercise 5: ConfigMaps and Secrets

Objective: Manage configurations using ConfigMaps and Secrets.

1. Create a ConfigMap using a key-value pair:

2. Mount the ConfigMap as environment variables in a pod.

3. Create a Secret:

4. Access the Secret in the pod via environment variables.

Output Screenshot:

A computer screen shot of a black screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Exercise 6: Persistent Volumes (PVs) and Persistent Volume Claims

(PVCs)

Objective: Use PVs and PVCs for persistent data storage.

1. Create a Persistent Volume (PV) and a Persistent Volume Claim (PVC) in YAML.

2. Apply the YAML files to create the PV and PVC.

3. Create a pod that uses the PVC to mount the volume.

4. Write data to the volume and verify its persistence by restarting the pod.

Output Screenshot:

A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

🔹 Exercise 7: StatefulSets

Objective: Use StatefulSets for managing stateful applications.

1. Deploy a StatefulSet with (create YAML for an app like MySQL).

2. View the StatefulSet.

3. Create a headless service for the StatefulSet and access the pod by its stable network

identity.

Output Screenshot:

A computer screen shot of white text

AI-generated content may be incorrect.

🔹 Exercise 8: Horizontal Pod Autoscaling (HPA)

Objective: Scale your application automatically based on metrics.

1. Create a deployment (e.g., a simple HTTP server).

2. Enable metrics server for autoscaling (e.g., Minikube).

3. Create an HPA to scale the deployment based on CPU utilization:

4. Test autoscaling by generating load on the deployment .

Output Screenshot:

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

🔹 Exercise 9: Helm Basics

Objective: Use Helm to manage Kubernetes applications.

1. Install Helm on your local machine.

2. Add a Helm chart repository:

3. Install a package from the Helm chart repository, e.g., Nginx:

4. Verify the installation using

Output Screenshot:

A screenshot of a computer program

AI-generated content may be incorrect.

🔹 Exercise 10: Debugging and Troubleshooting

Objective: Learn how to troubleshoot issues in Kubernetes.

1. Identify pod issues using describe command

2. Check the status of nodes and pods

3. View events related to the pod

4. View logs for troubleshooting(pods and deployment)

Output Screenshot:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

Capstone Project: Kubernetes Application Deployment

Objective: Deploy a production-grade multi-container application using Kubernetes.

1. Dockerize an existing application (e.g., a Python web app using Flask or FastAPI).

2. Create Kubernetes YAML files for deploying your application, including Deployment,

Service, and Persistent Volume Claims (if needed).

3. Ensure that the app scales, is resilient, and is accessible via a service (NodePort).

Output Screenshot:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a black screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.