Set C

1. Threetier Web application using Terraform and AWS
2. How to install Prometheus and Grafana on you cluster using Terraform and helm.
3. Setup your code on a Docker container using Jenkins on AWS
4. Kubernetes Assignment: Deploying and Managing Containers with Kubernetes

Scenario: You are a DevOps engineer at "ABVC Solutions," a cloud-based software company. Your company wants to migrate its applications to Kubernetes for better scalability and management. Your task is to set up a Kubernetes cluster and deploy a sample microservices application.

Assignment Tasks:

* Kubernetes Setup
  + Set up a Kubernetes cluster using a tool of your choice (e.g., Minikube for local development, or use cloud-managed Kubernetes clusters like GKE, EKS, or AKS).
  + Ensure that kubectl is properly configured and can communicate with your cluster.
* Containerization and Docker
  + Containerize a sample microservices application using Docker. You can use a sample application or create a simple one.
  + Create Docker images for each microservice.
* Kubernetes Deployments
  + Define Kubernetes Deployment resources for your microservices in YAML files.
  + Include specifications for the number of replicas, container images, ports, and environment variables.
  + Apply the deployment files to create deployments for each microservice in your Kubernetes cluster.
* Service Discovery and Load Balancing
  + Create Kubernetes Service resources for your microservices.
  + Ensure that services expose your microservices to the cluster and set up load balancing.
  + Test the services to verify that they can discover and route traffic to the microservices correctly.
* Scaling
  + Configure Horizontal Pod Autoscaling for one of the microservices based on CPU utilization.
  + Test the autoscaling behavior by generating load on the microservice.
* Storage and Data Persistence
  + Set up a Kubernetes Persistent Volume and Persistent Volume Claim for your application.
  + Modify one of the microservices to use the persistent volume for data storage.
  + Demonstrate data persistence by creating, reading, and updating data within the microservice.