

DevOps Engineer Technical Code Challenge

Challenge Overview:

In this challenge, you'll enhance and deploy a demo online store application on an Kubernetes (preferably AKS), using a blend of infrastructure as code, containerization, and CI/CD automation.

Submission Guidelines:

1. **GitHub Repository:** Create a new GitHub repository for your solution, including all relevant files (Dockerfiles, Kubernetes manifests, CI/CD YAML files, etc.).
Note: Do NOT include any files from the AKS Demo repository
2. **README:** Write a README that explains:
 - How to run the application locally (using Docker or Kubernetes).
 - How to configure and run the CI/CD pipeline.
 - How to deploy the application to Kubernetes

Environment setup

Use the following repository for Online store demo <https://github.com/Azure-Samples/aks-store-demo> and the following manifest [aks-store-quickstart.yaml](#) as starting point.

Challenge Steps:

1. Complete the Kubernetes manifest with Ingress controller

- **Task:** Add Ingress controller definition to the Kubernetes manifest

2. Create Kubernetes cluster by Terraform

- **Task:** Create the cluster using Terraform

3. Create CI/CD for the project

- **Task:** Implement a CI/CD pipeline using **yaml** , preferably Azure DevOps:
 - **CI**
 - **Building** the Docker images for the frontend and backend.
 - **Testing** the application before deploying.
 - **Pushing** the Docker images to **Azure Container Registry (ACR)** or another container registry.
 - **CD**
 - **Deploying** the updated images to an AKS cluster.

4. Bonus steps

- **Task:** Create Helm chart, include resource limits and improve inter-service security.
 - Create a **Helm chart** to manage the application's deployment.
 - Include **resource requests** and **limits** for containers.
 - implement **network policies** to limit inter-service communication within the Kubernetes cluster