PwC DevOps Task03:

This project aims to setup a robust CI/CD pipeline for a three-tier application consists of:

- Frontend: **React**
- Backend: **Nodejs**
- Database: **Mongodb**

Project repo: https://github.com/abanobmorkosgad/DevOps_Task_03.git

Steps:

1- Preparing infrastructure:

- **Provision EC2 of size "t2.large" with Ubuntu AMI for:**
  1. **Jenkins server**: installation will be found in **scripts/Jenkins.sh**
  2. **Sonarqube**: Docker container setup will be found in **scripts/sonarqube.sh**
  3. **Trivy:** installation will be found in **scripts/trivy.sh**

- **Provision an EKS cluster:**
  In **terraform directory** tf files for EKS will be found:
  * VPC and subnets and IGW and NAT
  * EKS cluster and EKS node group with 2 nodes with size "t2.medium" provided with required permissions

2- Create public repo for the code

Define webhook: **http://35.91.30.41:8080/multibranch-webhook-trigger/invoke?token=githubtoken**

3- Containerization of the frontend and backend apps:

- Create **Dockerfile** in frontend dir
- Create **Dockerfile** in backend dir

4- Manifest files of k8s using the docker images:

- **Database:**
  1. **k8s_manifests/mongo/deploy.yaml:** StatefulSet of mongodb with pvc
  2. **k8s_manifests/mongo/service.yaml:** ClusterIP service of mongodb
  3. **k8s_manifests/mongo/secret.yaml:** username and password of mongodb base64

- **backend:**
  1. **k8s_manifests/ backend-deployment.yaml:** deployment of backend
  2. **k8s_manifests/ backend-service.yaml:** service of backend

- **frontend:**
  1. **k8s_manifests/ frontend-deployment.yaml:** deployment of frontend
  2. **k8s_manifests/ frontend-service.yaml:** LoadBalancer service of frontend

5- **helm charts for Backend and Frontend:**

- **pwc_chart:**
  1. **Chart.yaml:** metadata of chart
  2. **templates:**
     a. **deployment.yaml:** template of deployment
     b. **service.yaml:** template of deployment
  3. **values.yaml:** default values of deployment and service templates

- **values/backend-values.yaml, values/frontend-values.yaml:**
  *actual values of deployment and service to be replaced in template files

6- **CI pipeline (Jenkinsfile):**

Create a Multi branch pipeline and configure automatic triggering using Multibranch Scan Webhook Trigger plugin

- **Tools and Environment**
  1. **Java JDK 17**
  2. **Node.js 16**
  3. **SonarQube Scanner**
  4. **Docker**
  5. **Trivy**
  6. **AWS CLI**

  The environment variables used include AWS credentials, Docker repository information, and SonarQube scanner path.

- **Stages Description**
  1. **Build Frontend**
     - Installs Node.js dependencies and builds the frontend application.

2. **Pack Frontend**
    - Archives the built frontend application into a tar.gz file.
3. **Build Backend**
    - Installs Node.js dependencies for the backend application.
4. **SonarQube Analysis - Frontend**
    - Runs SonarQube analysis on the frontend codebase to ensure code quality.
5. **SonarQube Analysis - Backend**
    - Runs SonarQube analysis on the backend codebase to ensure code quality.
6. **Quality Gate**
    - Waits for the SonarQube Quality Gate result to ensure the code meets quality standards before proceeding.
7. **Build and Push Docker Images**
    - Builds Docker images for the frontend and backend applications and pushes them to a Docker registry.
8. **Trivy Scan and Upload Results to S3**
    - Scans the Docker images for vulnerabilities using Trivy and uploads the scan reports to an S3 bucket.
9. **Update Image Version in Helm Values**
    - Updates the image version in Helm values files to the new build version.
10. **Update GitHub Repository**
    - Commits and pushes the updated Helm values files to the GitHub repository.

- **Environment Variables and Credentials**

    - AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY: AWS credentials for S3 and other AWS services.
    - SCANNER_HOME: Path to the SonarQube scanner.
    - REPO_SERVER: Docker repository server.
    - REPO_NAME_BACKEND and REPO_NAME_FRONTEND: Docker repository names for the backend and frontend images.
    - IMAGE_VERSION: Version of the Docker image, set to the build number.
    - GitHub credentials for pushing changes to the repository.

Important Notes

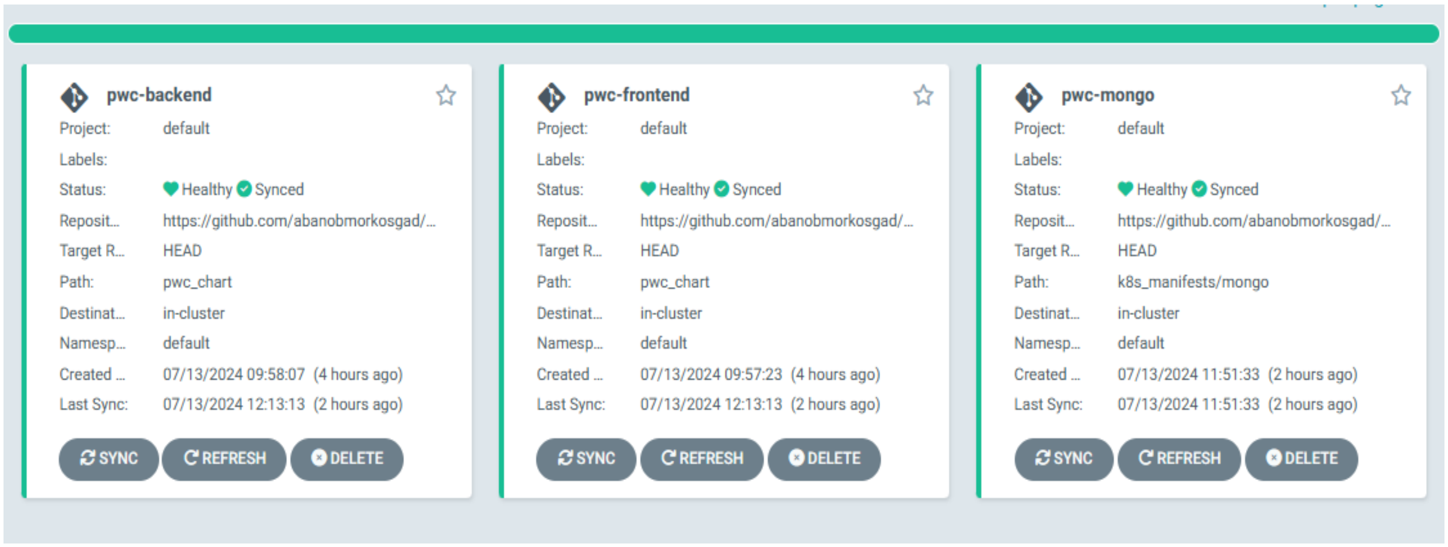
- Ensure that the required tools and plugins (SonarQube, Docker, Trivy, etc.) are installed and configured in Jenkins.
- Sensitive information such as AWS credentials and GitHub credentials should be managed securely using Jenkins credentials.
- Ignore committer strategy needed to avoid infinity loop of pipeline triggers.

6- **CD with ArgoCD:**

- **Installation of argoCD in cluster will be found in scripts/argocd.sh**
- **Accesing the Dashboard of argocd:**
  - **Application for Frontend helm chart:**
    PATH: pwc_chart
    VALUES FILES: ../values/backend-values.yaml

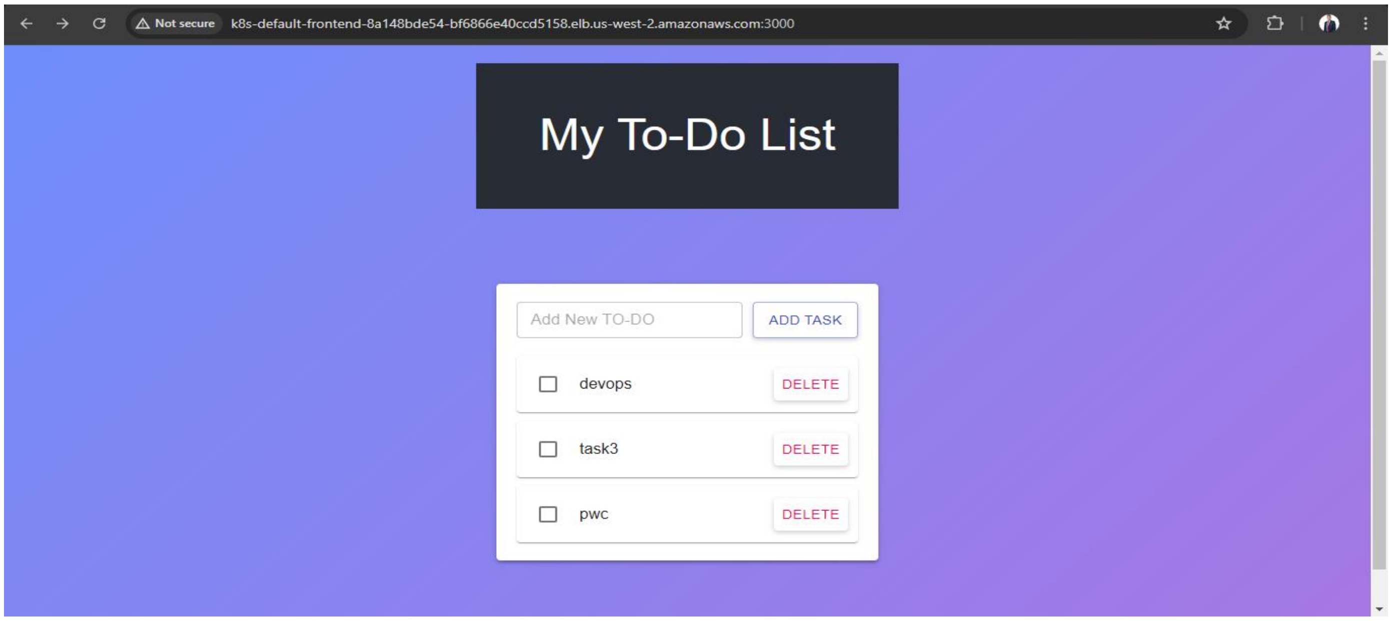
  - **Application for Backend helm chart:**
    PATH: pwc_chart
    VALUES FILES: ../values/frontend-values.yaml

  - **Application for mongodb manifest files:**
    PATH: k8s_manifests/mongo



	pwc-backend	pwc-frontend	pwc-mongo
Project:	default	default	default
Labels:			
Status:	Healthy Synced	Healthy Synced	Healthy Synced
Reposit...	https://github.com/abanobmorkosgad/...	https://github.com/abanobmorkosgad/...	https://github.com/abanobmorkosgad/...
Target R...	HEAD	HEAD	HEAD
Path:	pwc_chart	pwc_chart	k8s_manifests/mongo
Destinat...	in-cluster	in-cluster	in-cluster
Namesp...	default	default	default
Created ...	07/13/2024 09:58:07 (4 hours ago)	07/13/2024 09:57:23 (4 hours ago)	07/13/2024 11:51:33 (2 hours ago)
Last Sync:	07/13/2024 12:13:13 (2 hours ago)	07/13/2024 12:13:13 (2 hours ago)	07/13/2024 11:51:33 (2 hours ago)

Accessing Frontend Loadbalancer DNS name:



Not secure k8s-default-frontend-8a148bde54-bf6866e40ccd5158.elb.us-west-2.amazonaws.com:3000

My To-Do List

devops DELETE
task3 DELETE
pwc DELETE