DEV-SEC-OPS PROJECTS

- PROJECT: - Secure CI/CD Pipeline with Jenkins, Docker, Trivy, Snyk, and Kubernetes.

- Objective: - Set up a **secure DevSecOps CI/CD pipeline** that includes.
    1. Code Analysis (SAST) using SonarQube
    2. Dependency Scanning using Snyk
    3. Container Scanning using Trivy
    4. Security Testing with OWASP ZAP
    5. Deployment to Kubernetes

- Tech Stack: -
    1. Jenkins
    2. Docker
    3. SonarQube (SAST)
    4. Snyk (Dependency Scanning)
    5. Trivy (Container Scanning)
    6. OWASP ZAP (DAST)
    7. Kubernetes (Deployment)
    8. Helm
- Pipeline Steps: -
    1. Code Checkout → Jenkins pulls the source code from GitHub.
    2. Static Code Analysis (SAST) → SonarQube scans the code.
    3. Dependency Scanning → Snyk checks for vulnerabilities in dependencies.
    4. Build Docker Image → Create a secure containerized application.
    5. Container Scanning → Trivy scans the image for vulnerabilities.
    6. Security Testing (DAST) → OWASP ZAP performs penetration testing.
    7. Deploy to Kubernetes → Helm deploys to Kubernetes.

- **COMPLETE JENKINS FILE FOR SECURE CI/CD.**

```
pipeline {
    agent any

    environment {
        IMAGE_NAME = "secure-app"
        IMAGE_TAG = "latest"
        REGISTRY = "your-dockerhub-username"
    }

    stages {
        stage('Checkout Code') {
            steps {
                git branch: 'main', url: 'https://github.com/your-repo/secure-
app.git'
            }
        }

        stage('SAST Analysis') {
            steps {
                sh 'sonar-scanner -Dsonar.projectKey=secure-app -
Dsonar.sources=./src -Dsonar.host.url=http://sonarqube:9000'
            }
        }

        stage('Dependency Scanning') {
            steps {
                sh 'snyk test --all-projects'
            }
        }

        stage('Build Docker Image') {
            steps {
                sh 'docker build -t
$REGISTRY/$IMAGE_NAME:$IMAGE_TAG .'
            }
        }

        stage('Container Scanning') {
            steps {
```

```
                sh 'trivy image
$REGISTRY/$IMAGE_NAME:$IMAGE_TAG'
            }
        }

        stage('Push Docker Image') {
            steps {
                sh 'docker push
$REGISTRY/$IMAGE_NAME:$IMAGE_TAG'
            }
        }

        stage('Deploy to Kubernetes') {
            steps {
                sh 'helm upgrade --install secure-app helm/secure-app'
            }
        }

        stage('DAST Testing with OWASP ZAP') {
            steps {
                sh 'zap-cli quick-scan --start-options "-config
api.disablekey=true" http://your-app-url.com'
            }
        }
    }
}
```

- **Deployment Files (Helm Charts).**

1. Kubernetes Deployment (deploy.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: secure-app

spec:
  replicas: 2
  selector:
    matchLabels:
      app: secure-app
  template:
    metadata:
      labels:
        app: secure-app

    spec:
      containers:
      - name: secure-app
        image: your-dockerhub-username/secure-app:latest
        ports:
        - containerPort: 8080
```

- Kubernetes Service (service.yaml)

```
apiVersion: v1
kind: Service
metadata:
  name: secure-app

spec:
  selector:
    app: secure-app
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
  type: LoadBalancer
```