ZomatoClone: Secure Deployment with DevSecOps CI/CD

Prerequisites

1. ***Tools Required:***
    - *Git*
    - *Docker*
    - *Kubernetes*
    - *Jenkins or GitHub Actions (for CI/CD)*
    - *Static Analysis Tools (e.g., SonarQube, CodeQL)*
    - *IaC Tools (e.g., Terraform)*
    - *Security Tools (e.g., Snyk, Trivy, AquaSec)*
    - *Monitoring Tools (e.g., Prometheus, Grafana)*
2. ***Accounts and Permissions:***
    - *Cloud provider account (AWS, GCP, or Azure)*
    - *Access to a Kubernetes cluster*
    - *Admin rights for creating CI/CD pipelines*
3. ***Secure Secrets Management:***
    - *Use tools like HashiCorp Vault or Kubernetes Secrets to manage sensitive data.*

Step-by-Step Deployment

Step 1: Code Development

- *Develop the ZomatoClone application, ensuring adherence to secure coding standards.*
- *Organize the project with a clear folder structure (e.g., `/src`, `/tests`, `/config`).*
- *Include a `README.md` file for project details.*

Step 2: Version Control

- *Initialize a Git repository and push the code to a platform like GitHub or GitLab.*
- *Protect the main branch by enabling branch protection rules (e.g., requiring pull request reviews).*

Step 3: Static Code Analysis

- *Integrate a static code analysis tool like SonarQube or CodeQL into your workflow.*
- *Configure the tool to scan code automatically during pull requests.*
- *Fix any identified vulnerabilities or code smells before proceeding.*

Step 4: Containerization

- *Create a* `Dockerfile` *for the application:*

FROM node:16
WORKDIR /app
COPY package.json ./*
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
Build the Docker image and test it locally:
docker build -t zomato-clone:latest .

- *docker run -p 3000:3000 zomato-clone:latest*
- *Push the Docker image to a secure container registry (e.g., Docker Hub, AWS ECR).*

Step 5: CI/CD Pipeline Setup

- *Use Jenkins, GitHub Actions, or GitLab CI/CD to automate the process.*
- *Example GitHub Actions Workflow:*

```
name: CI/CD Pipeline

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Set up Node.js
        uses: actions/setup-node@v3
        with:
          node-version: 16

      - name: Install dependencies
        run: npm install

      - name: Run tests
        run: npm test

      - name: Build Docker image
        run: |
          docker build -t zomato-clone:latest .
```

```
        echo "${{ secrets.DOCKER_PASSWORD }}" | docker login -u
"${{ secrets.DOCKER_USERNAME }}" --password-stdin
        docker push zomato-clone:latest

  deploy:
    runs-on: ubuntu-latest
    needs: build

    steps:
      - name: Deploy to Kubernetes
        run: |
          kubectl apply -f k8s/deployment.yaml
          kubectl apply -f k8s/service.yaml
```

Step 6: Infrastructure as Code

- *Write a Terraform script to provision secure infrastructure:*

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "zomato_clone" {
  ami           = "ami-0c55b159cbfafe1f0"
  instance_type = "t2.micro"

  tags = {
    Name = "ZomatoCloneServer"
  }
}
```

Run Terraform commands to apply changes:
terraform init
terraform plan

- *terraform apply*

Step 7: Dynamic and Runtime Security Scanning

- *Integrate runtime security tools like AquaSec or Trivy into the pipeline.*
- *Example Trivy scan command:*
  trivy image zomato-clone:latest
- *Address any identified vulnerabilities.*

Step 8: Monitoring and Logging

- *Deploy monitoring tools like Prometheus and Grafana to monitor application performance.*
- *Enable logging for security events and application activity using ELK stack (Elasticsearch, Logstash, Kibana).*