Zomato Project Full Deployment Documentation

1. Instance Setup

Launch required instances on AWS:

- Application Server: Ubuntu 24.04, t2.large, 30GB EBS, security group allowing SSH, HTTP, and NodePort.
- Monitoring Server: Ubuntu 24.04, t2.large, 30GB EBS, security group allowing SSH, Prometheus port 9090, Node Exporter port 9100, Grafana port 3000.

Connect via SSH:

```
ssh ubuntu@<INSTANCE-IP>
```

2. Jenkins Installation and Configuration

2.1 Install Jenkins

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkin
sudo apt update
sudo apt install -y openjdk-17-jdk jenkins
sudo systemctl enable jenkins
sudo systemctl start jenkins
```

2.2 Access Jenkins

Open Jenkins in browser: http://<jenkins-server-ip>:8080

Use initial admin password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

2.3 Install Plugins

- Docker Pipeline
- Prometheus Metrics
- GitHub Integration
- Pipeline
- Email Extension Plugin
- SonarQube Scanner

2.4 Configure Jenkins Credentials

- Docker Hub token: Used to push images to Docker Hub
- SonarQube token: Used to connect Jenkins with SonarQube for code analysis
- Email App Password: Configured for Jenkins email notifications

2.5 Create Pipeline

Use Jenkinsfile from GitHub repo:

```
pipeline {
    agent any
    stages {
        stage('Build Docker Image') {
            steps {
                script {
                    docker.build("zomato:latest")
                }
            }
        }
        stage('Push to Docker Hub') {
            steps {
                withCredentials([string(credentialsId: 'dockerhub-token', variable: 'DOCKERHUB_TO
                    sh 'docker login -u <docker-user> -p $DOCKERHUB_TOKEN'
                    sh 'docker push zomato:latest'
                }
            }
        }
        stage('Kubernetes Deploy') {
            steps {
                sh 'kubectl apply -f Kubernetes/deployment.yaml'
                sh 'kubectl apply -f Kubernetes/service.yaml'
                sh 'kubectl apply -f Kubernetes/node-service.yaml'
            }
        }
        stage('SonarQube Scan') {
            steps {
                withSonarQubeEnv('SonarQube') {
                    sh 'mvn sonar:sonar'
                }
            }
        }
```

```
    }
}
```

3. Docker Deployment

Verify Docker installation and run Zomato image:

```
docker version
docker ps -a
docker build -t zomato:latest .
docker run -d -p 3000:3000 zomato:latest
```

Access app at http://<app-server-ip>:3000

4. Kubernetes Deployment

4.1 Install Kubernetes

```
sudo apt update
sudo apt install -y kubeadm kubelet kubectl
sudo apt-mark hold kubeadm kubelet kubectl
```

4.2 Initialize Cluster

```
sudo kubeadm init --pod-network-cidr=192.168.0.0/16
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

4.3 Install Calico Network Plugin

```
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.
```

4.4 Deploy Zomato App

```
kubectl create namespace zomato
kubectl config set-context --current --namespace=zomato
kubectl apply -f Kubernetes/deployment.yaml
```

```
kubectl apply -f Kubernetes/service.yaml
kubectl apply -f Kubernetes/node-service.yaml
```

Access Zomato app: http://<app-server-ip>:30001

5. Prometheus Setup

```
sudo useradd --system --no-create-home --shell /bin/false prometheus
wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-a
tar -xvf prometheus-2.47.1.linux-amd64.tar.gz
cd prometheus-2.47.1.linux-amd64
sudo mkdir -p /data /etc/prometheus
sudo mv prometheus promtool /usr/local/bin/
sudo mv consoles/ console_libraries/ /etc/prometheus/
sudo mv prometheus.yml /etc/prometheus/prometheus.yml
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
```

```
sudo vi /etc/systemd/system/prometheus.service
```

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/data \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-lifecycle

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl enable prometheus
sudo systemctl start prometheus
sudo systemctl status prometheus
```

6. Node Exporter Setup

```
sudo useradd --system --no-create-home --shell /bin/false node_exporter
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.lin
tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/
```

```
sudo vi /etc/systemd/system/node_exporter.service
```

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
ExecStart=/usr/local/bin/node_exporter --collector.logind

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl enable node_exporter
sudo systemctl start node_exporter
sudo systemctl status node_exporter
```

7. Grafana Installation and Dashboard

```
sudo apt-get update
sudo apt-get install -y apt-transport-https software-properties-common
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d
sudo apt-get update
sudo apt-get -y install grafana
sudo systemctl enable grafana-server
sudo systemctl start grafana-server
sudo systemctl status grafana-server
```

Access Grafana: http://<monitoring-server-ip>:3000

Default credentials: admin / admin

Add Prometheus as data source, configure dashboards for Node Exporter and Jenkins metrics.

8. Final Output and Access

- Zomato App: http://<app-server-ip>:30001
- Prometheus: http://<monitoring-server-ip>:9090
- Grafana: http://<monitoring-server-ip>:3000

Verify pods and services:

```
kubectl get pods -n zomato
kubectl get svc -n zomato
kubectl get pods -n prometheus-node-exporter
kubectl get svc -n prometheus-node-exporter
```