Automated Deployment & Monitoring Using Jenkins, Kubernetes, Prometheus, Grafana, and CloudWatch

1. Project Overview

This project automates the deployment and monitoring of a Maven-based Java web application using modern DevOps tools. The workflow involves:

- **CI/CD Pipeline** using Jenkins.
- **Containerization** with Docker.
- **Orchestration** via Kubernetes on AWS EKS.
- **Monitoring & Logging** using Prometheus, Grafana, and CloudWatch.

---

2. Infrastructure Setup using Terraform

Terraform provisions the required AWS infrastructure:

- **AWS EKS Cluster** with nodes.
- **VPC, Subnets, Security Groups** for networking.
- **IAM roles** for Jenkins and Kubernetes.
- **CloudWatch Setup** to monitor logs and metrics.

Terraform Files:

- `provider.tf` → AWS Provider Configuration.
- `main.tf` → EKS Cluster, VPC, and Subnets.
- `monitoring.tf` → Configures Prometheus, Grafana, and CloudWatch.
- `output.tf` → Outputs values like EKS endpoint, CloudWatch logs.
- `vars.tf` → Stores variable values.

---

3. Jenkins Setup & CI/CD Pipeline

Jenkins automates the entire software delivery lifecycle.

Pipeline Stages

1. **Fetch Code from GitHub**
2. **Build with Maven**
3. **Build & Push Docker Image** (Locally stored Dockerfile)
4. **Deploy to Kubernetes using kubectl & Helm**

5. **Monitor using Prometheus & Grafana**
6. **Log Management with CloudWatch**

---

4. Kubernetes Deployment Configuration

Kubernetes manages the application in the AWS EKS cluster.

Deployment & Service YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mavenwebapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mavenwebapp
  template:
    metadata:
      labels:
        app: mavenwebapp
    spec:
      containers:
      - name: mavenwebapp
        image: myrepo/mavenwebapp:latest
        ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: mavenwebapp-service
spec:
  selector:
    app: mavenwebapp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
  type: LoadBalancer
```

---

5. Monitoring Setup: Prometheus, Grafana, and CloudWatch

Prometheus Setup

Prometheus collects real-time metrics from Kubernetes.

Helm Installation

```
helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts
helm repo update
helm install prometheus prometheus-community/kube-prometheus-stack --
namespace monitoring --create-namespace
```

Prometheus Service Check

```
kubectl get pods -n monitoring
kubectl get svc -n monitoring
```

Grafana Setup

Grafana visualizes metrics from Prometheus.

Access Grafana

```
kubectl port-forward svc/prometheus-grafana 3000:80 -n monitoring
```

- Open **`http://localhost:3000`**
- Default Login: **admin/admin**
- Import Dashboards from Grafana.com

Reset Admin Password (If Needed)

```
kubectl exec -it prometheus-grafana-xxxxxx -n monitoring -- grafana-cli
admin reset-admin-password newpassword
```

AWS CloudWatch Integration

CloudWatch provides centralized logging and metrics.

CloudWatch Agent Installation

```
sudo yum install amazon-cloudwatch-agent
sudo systemctl enable amazon-cloudwatch-agent
sudo systemctl start amazon-cloudwatch-agent
```

CloudWatch Logs Setup

```
aws logs create-log-group --log-group-name /eks/monitoring
aws logs create-log-stream --log-group-name /eks/monitoring --log-stream-
name node-metrics
```

Send Logs to CloudWatch

```
{
  "agent": {
    "metrics_collection_interval": 60,
    "logfile": "/var/log/cloudwatch-agent.log"
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/kubelet.log",
            "log_group_name": "/eks/monitoring",
            "log_stream_name": "{instance_id}"
          }
        ]
      }
    }
  }
}
```

6. Project Summary

Tool	Purpose
Terraform	AWS Infrastructure Setup
Jenkins	CI/CD Pipeline
Docker	Containerization
Kubernetes	Orchestration on AWS EKS
Prometheus	Monitoring & Metrics Collection
Grafana	Visualization & Dashboards
CloudWatch	Log Management & Monitoring

Outcome

- **Automated Deployment:** Jenkins ensures continuous integration and delivery.
- **Efficient Monitoring:** Prometheus & Grafana provide real-time insights.
- **Centralized Logging:** CloudWatch tracks system logs for debugging.

This document provides a comprehensive overview of the DevOps setup, including **Jenkins CI/CD, Docker, Kubernetes, Prometheus, Grafana, and AWS CloudWatch monitoring.**