Cloud Solution Architecture / Senior DevOps Engineer

1. Cloud Provider: AWS

Primary AWS Services Used:

- **EKS (Elastic Kubernetes Service):** For container orchestration
- **ALB (Application Load Balancer):** For HTTP/HTTPS load balancing
- **EC2:** Managed EKS with a Bastion Host Server
- **RDS / PostgreSQL:** For managed database services

2. Infrastructure as Code (IaC) with Terraform

Terraform Responsibilities:

➔ Provision and manage the following:
  - VPC, subnets, route tables
  - EKS Cluster
  - Security groups and IAM roles
  - RDS/PostgreSQL
  - ALB configuration

Suggested Terraform Folder Structure:

```
terraform/
   ➔ main.tf
   ➔ variables.tf
   ➔ outputs.tf
   ➔ eks/eks-cluster.tf
   ➔ vpc/vpc-setup.tf
   ➔ rds/rds-instance.tf
```

3. Containerization with Docker

Frontend (React/Vue/Angular etc.):

Dockerfile

```
FROM node:20
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["npm", "start"]
```

Backend (Node.js):

Dockerfile

```
FROM node:20
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD ["node", "server.js"]
```

4. Kubernetes Deployment on EKS

Kubernetes Manifests Structure:

```
k8s/
  ➔ namespace.yaml
  ➔ frontend-deployment.yaml
  ➔ frontend-service.yaml
  ➔ backend-deployment.yaml
  ➔ backend-service.yaml
  ➔ ingress.yaml                        # For ALB Ingress Controller
  ➔ hpa.yaml                            # Horizontal Pod Autoscaler
```

Ingress Setup:

- ➔ Use **AWS ALB Ingress Controller** (or AWS Load Balancer Controller) to manage routing and external access to services.

5. CI/CD Pipeline (GitHub Actions)

Workflow: .github/workflows/deploy.yml

```yaml
name: CI/CD Pipeline

on:
  push:
    branches:
      - main

jobs:
  build_and_push:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout Code
        uses: actions/checkout@v2

      - name: Build and Push Docker Images
        run: |
          docker build -t johnsmith/my-frontend:frontend-v1 ./frontend
          docker build -t johnsmith/my-backend:backend-v1 ./backend
          docker push johnsmith/my-frontend:frontend-v1
          docker push johnsmith/my-backend:backend-v1

  deploy:
    needs: build_and_push
    runs-on: ubuntu-latest
    steps:
      - name: Configure kubectl with EKS
        run: |
          aws eks update-kubeconfig --region us-east-1 --name my-cluster

      - name: Deploy Kubernetes Manifests
        run: |
          kubectl apply -f k8s/
```

6. IAM Roles and Security

- ➔ Assign IAM roles to Kubernetes service accounts for fine-grained AWS access.
- ➔ **Security Groups Configuration:**
    - ◆ ALB: Allow HTTP/HTTPS traffic (80/443).
    - ◆ EC2/EKS Nodes: Restrict SSH and other ports.
    - ◆ RDS: Allow connections only from the backend.
- ➔ **IAM Policies:**
    - ◆ Follow the least **privilege** principle.
    - ◆ Create scoped policies for EKS worker nodes

7. Auto-Scaling

EKS-Level Scaling Setup:

- ➔ **Horizontal Pod Autoscaler (HPA):**
    - ◆ Automatically scale pods based on CPU/memory usage or custom metrics.
    - ◆ Define in hpa.yaml.

- ➔ **Cluster Autoscaler:**
    - ◆ Auto-scales EC2 worker nodes (or Fargate profiles) based on pod scheduling needs.

8. Load Balancer Configuration

AWS Application Load Balancer (ALB):

- ➔ **Ingress Controller:**
    - ◆ Use **AWS ALB Ingress Controller** (or newer AWS Load Balancer Controller).

- ➔ **Features to Enable:**
    - ◆ **Health Checks:** Ensure routing only to healthy pods.
    - ◆ **SSL Termination:**
        - ● Use **AWS Certificate Manager (ACM)** for TLS certificates.

Github repository structure:

- buliptech-project/
  - terraform/
    - `main.tf`
    - `variables.tf`
    - `outputs.tf`
    - `eks/eks-cluster.tf`
    - `vpc/vpc-setup.tf`
    - `rds/rds-instance.tf`
  - k8s/
    - namespace.yaml
    - frontend-deployment.yaml
    - frontend-service.yaml
    - backend-deployment.yaml
    - backend-service.yaml
    - postgres-deployment.yaml
    - ingress.yaml
    - Hpa.yaml
  - frontend/
    - Dockerfile
  - backend/
    - Dockerfile
  - .github/
    - workflows/
      - deploy.yml
  - README.md