

Practice 3 - Cloud Deployment

Title: Deploy Full Stack App on AWS with Load Balancing

Objective:

Gain experience deploying a full stack application to AWS and configure load balancing for scalability and high availability. Learn about EC2, Elastic Load Balancing (ELB), and VPC basics.

Required Materials:

- AWS Account
- AWS EC2 Instances (for frontend and backend)
- AWS Elastic Load Balancer (ALB)
- Basic knowledge of VPC, Security Groups, and Route 53 (optional for custom domain)
- Full stack application (React frontend + Node.js/Express backend + optional MongoDB database)

Steps to Follow:

1. ****Build the Full Stack Application****
 - Frontend: React app created using Create React App.
 - Backend: Node.js/Express server with API endpoints.
 - Database (optional): MongoDB (Atlas or self-hosted).
2. ****Prepare AWS Environment****
 - Launch EC2 instances for the frontend and backend.
 - Choose Amazon Linux 2 or Ubuntu as the OS.
 - Configure Security Groups to allow HTTP (port 80) and SSH (port 22) access.
3. ****Deploy Backend to EC2****

```
```bash
Connect to backend EC2 instance
ssh -i "your-key.pem" ec2-user@your-ec2-public-ip

Install Node.js and Git
sudo apt update
sudo apt install -y nodejs npm git

Clone your backend repository
git clone https://github.com/yourusername/backend.git
cd backend

Install dependencies and start the server
npm install
npm start
```
```
4. ****Deploy Frontend to EC2****

```
```bash
Connect to frontend EC2 instance
ssh -i "your-key.pem" ec2-user@your-ec2-public-ip

Install Node.js and Nginx
sudo apt update
sudo apt install -y nodejs npm nginx git

Clone your frontend repository
git clone https://github.com/yourusername/frontend.git
cd frontend
npm install
npm run build

Move build files to Nginx directory
sudo cp -r build/* /var/www/html/
sudo systemctl restart nginx
```
```

5. ****Set Up Load Balancer****
 - Go to AWS Management Console → EC2 → Load Balancers → Create Load Balancer.
 - Choose "Application Load Balancer".
 - Add your backend EC2 instances to the target group.
 - Configure listeners for HTTP (port 80).
6. ****Configure Security and Networking****
 - Ensure the Load Balancer's security group allows inbound HTTP (port 80) traffic.
 - Backend EC2 instances should only accept traffic from the Load Balancer's security group.
7. ****(Optional) Configure Route 53****
 - Register a domain or use an existing one.
 - Create an "A" record pointing to your Load Balancer's DNS name.
8. ****Test Your Deployment****
 - Access the app via Load Balancer DNS or domain name.
 - Stop one backend EC2 instance to verify that the Load Balancer routes traffic to the remaining o

Expected Output:

- A running full stack application accessible over the internet.
- AWS ALB successfully distributing traffic to backend instances.
- Demonstrated scalability and fault tolerance through load balancing.