

DevOps Assignment: One-Click Deployment (Short Version)

Objective

Create a **one-click deployment solution** that sets up a **simple REST API server** running on **private EC2 instances** behind an **Application Load Balancer (ALB)** and **Auto Scaling Group (ASG)** using IaC (Terraform or CloudFormation).

The deployment must be **repeatable, secure, and easily testable**.

Architecture Overview

```
Client → ALB (public subnets) → Target Group → ASG → EC2 (private subnets)
                                     |
                               NAT Gateway (egress)
                                     |
                               Internet Gateway
```

AWS Components Required

- VPC with **2 public** and **2 private subnets**
 - Internet Gateway (IGW)
 - NAT Gateway (in a public subnet)
 - ALB with HTTP/HTTPS listener
 - Target Group (health check: `/health` or `/`)
 - Launch Template for EC2 instances
 - Auto Scaling Group (private subnets)
 - Security Groups:
 - **ALB SG:** allow HTTP/HTTPS from internet
 - **EC2 SG:** allow traffic only from ALB SG
 - IAM Role for EC2:
 - CloudWatch Logs
 - SSM (optional, recommended)
 - Optional: Route53 record
-

Assignment Requirements

1. One-Click Deploy

Provide **one command** (Terraform apply / CloudFormation deploy / GitHub Actions button) that deploys the entire stack.

2. REST API

A small server (Node.js / Python / Go) running on port **8080** with: - `/` → returns a simple text response
- `/health` → returns `ok`

Logs should print to stdout.

3. Private EC2 Instances

- No public IPs
- Can access internet via NAT Gateway
- Installed and started via user-data

4. ALB + Target Group

- Public ALB
- Routes traffic to ASG
- Health checks configured

5. Security & Best Practices

- No hardcoded secrets
- Minimal IAM permissions
- No SSH open to world (use SSM or restrict CIDR)

6. Teardown

Provide a **destroy script** or Terraform destroy method to avoid AWS charges.

Deliverables

Your Git repository must include:

- `terraform/` or `cloudformation/` folder with IaC
- `app/` folder with API code
- `scripts/` folder with:
 - `deploy.sh`
 - `destroy.sh`
 - `test.sh` (optional)
- `README.md` with:
 - Deployment steps
 - Testing steps
 - Teardown steps
- GitHub Actions workflow (optional but preferred)
- Required screenshots (ALB, Target Group, API test, private EC2, etc.)

This is your **final, short-form assignment doc** ready to give to candidates.