Project 1B: Scalable Web Infrastructure Setup on AWS (With ASG + Route53)

Objective

Deploy a scalable and secure web server infrastructure using VPC, EC2, Auto Scaling Group, NAT Gateway, Internet Gateway, custom NACLs, Route Tables, Launch Template, AMI, and Route53 domain pointing.

Step-by-Step Guide

1. VPC and Subnet Setup

- **Create a VPC:**

- CIDR: `10.0.0.0/16`

- Name: `Project1B-VPC`

- **Create Subnets:**

- **Public Subnet A:** `10.0.1.0/24` (AZ-a)

- **Public Subnet B:** `10.0.2.0/24` (AZ-b)
- **Private Subnet A:** `10.0.3.0/24` (AZ-a)

- **Private Subnet B:** `10.0.4.0/24` (AZ-b)

- **Enable Auto-assign Public IP:** For both public subnets

---

2. Internet Gateway and Routing

- **Create IGW:** Name: `Project1B-IGW`

- **Attach IGW to VPC**

- **Create Route Table:**

- Name: `Project1B-Public-RT`

- Route: `0.0.0.0/0 -> IGW`
- Associate to Public Subnet A & B

---

3. NAT Gateway for Private Subnet Access

- **Create Elastic IP**
- **Create NAT Gateway in Public Subnet A**
- **Create Private Route Table:**
- Name: `Project1B-Private-RT`
- Route: `0.0.0.0/0 -> NAT Gateway`
- Associate to Private Subnet A & B

---

4. Network ACLs (NACLs)

- **Public NACL:**

- Allow Inbound/Outbound: HTTP (80), HTTPS (443), SSH (22)

- Allow: 0.0.0.0/0

- Deny rule: `*` rule to avoid catch-all allow

- **Private NACL:**

- Allow: Ephemeral ports for NAT Gateway

- Inbound: TCP 1024-65535 from `10.0.0.0/16`
- Outbound: TCP 80/443 to `0.0.0.0/0`

---

5. Security Groups

- **Web SG:**
- Inbound: HTTP (80), SSH (22), from anywhere
- Outbound: All traffic

---

6. EC2 and Manual Setup (Base Configuration)

- Launch EC2 in Public Subnet A
- Amazon Linux 2023 AMI
- Attach Web SG
- Use SSH to connect and run:

```
sudo yum update -y
sudo yum install httpd -y
sudo systemctl enable httpd
sudo systemctl start httpd
```

```
echo "<html><h1>Webserver<br>Test Instance manual @ dipen.online</h1></
html>" > /var/www/html/index.html
```

---

7. Create AMI (Image)

- From above EC2, **Create Image**
- Name: `Project1B-WebServer-AMI`

---

8. Launch Template

- Name: `Project1B-LT`
- Use the AMI created
- Instance Type: t2.micro
- Attach Web SG
- Add User Data (Optional to auto-prepare instance)

---

9. Auto Scaling Group (ASG)

- Name: `Project1B-ASG`
- Use Launch Template: `Project1B-LT`
- Network: Project1B-VPC
- Subnets: Public Subnet A & B
- Desired Capacity: 1
- Min/Max: 1/2
- Health check type: EC2

---

10. Elastic Load Balancer (ELB)

- Type: Application Load Balancer
- Name: `Project1B-ALB`
- Scheme: Internet-facing
- Listeners: HTTP (80)
- Target Group:
- Name: `Project1B-TG`
- Protocol: HTTP

- Target type: instance
- Health Check Path: `/`
- Attach ASG to Target Group

---

11. Route53 Domain Setup

- Domain: `dipen.online`
- Hosted Zone: Already exists
- Create `A Record`:
- Name: `@`
- Alias: Yes
- Alias Target: `Project1B-ALB DNS name`

---

12. Verify

- Access `http://dipen.online`
- Should see: `Webserver - Test Instance Auto Scaling @ dipen.online`

✅Outcome

- A highly available, scalable web server architecture
- Load balanced EC2s managed via Auto Scaling
- Custom VPC, NACLs, Route Tables, NAT, and DNS (Route53) fully integrated