Deployment Guide: 2-Tier Web Architecture on AWS

This document explains the step-by-step deployment of a 2-Tier Web Architecture using AWS services, where:

- A Private EC2 hosts a simple website using Apache2.
- A Public EC2 acts as a Reverse Proxy using Apache2 to forward traffic to the private instance.

Tools & Services Used:

- Amazon VPC
- EC2 Instances
- Internet Gateway
- NAT Gateway
- Apache2 (Web Server + Reverse Proxy)
- MobaXterm (for SSH access)

STEP 1: Create a Custom VPC

1. Go to VPC Dashboard -> Create VPC
2. Set:
   - Name: MyCustomVPC
   - IPv4 CIDR: 10.0.0.0/16
3. Click Create

STEP 2: Create Two Subnets

Public Subnet:

- Name: PublicSubnet
- CIDR block: 10.0.1.0/24

Private Subnet:

- Name: PrivateSubnet

- CIDR block: 10.0.2.0/24

STEP 3: Internet Gateway (IGW)

1. Create and name MyIGW

2. Attach it to MyCustomVPC

STEP 4: Route Tables & NAT Gateway

Public Route Table:

- Associate with: PublicSubnet

- Route: 0.0.0.0/0 -> MyIGW

Private Route Table:

- Associate with: PrivateSubnet

- Allocate Elastic IP and create NAT Gateway in PublicSubnet

- Route: 0.0.0.0/0 -> NAT Gateway

STEP 5: Launch EC2 Instances

Public EC2 (Reverse Proxy):

- AMI: Ubuntu 20.04

- Subnet: PublicSubnet

- Public IP: Enabled

- SG: Allow SSH (22), HTTP (80)

Private EC2 (Web Server):

- AMI: Ubuntu 20.04

- Subnet: PrivateSubnet

- Public IP: Disabled

- SG: Allow SSH from 10.0.1.0/24, HTTP from Public EC2

STEP 6: Deploy Website on Private EC2

- SSH into Private EC2 from Public EC2

- sudo apt update && sudo apt install apache2 -y

- echo "<h1>Welcome to the PRIVATE EC2 Website</h1>" | sudo tee /var/www/html/index.html

- Test with: curl localhost

STEP 7: Configure Reverse Proxy on Public EC2

- sudo apt update && sudo apt install apache2 -y

- Enable modules: sudo a2enmod proxy && sudo a2enmod proxy_http

- Edit config: /etc/apache2/sites-available/000-default.conf

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost

    ProxyPreserveHost On

    ProxyPass / http://10.0.2.155/

    ProxyPassReverse / http://10.0.2.155/

</VirtualHost>
```

- Restart Apache: sudo systemctl restart apache2

STEP 8: Test the Setup

- Visit: http://<Public-EC2-Public-IP>

- Should see: Welcome to the PRIVATE EC2 Website

Supporting Files:

- architecture/
- screenshots/
- user-data/

Key Learnings:

- Built real-world 2-Tier architecture using AWS
- Configured private web hosting securely
- Setup Apache reverse proxy for controlled access