

**### \*\*Project Overview: AWS VPC Setup for Secure Web Application\*\***

In this project, I have designed and deployed a secure and scalable web application infrastructure using AWS Virtual Private Cloud (VPC). The infrastructure follows best practices by segregating resources into public and private subnets, leveraging a reverse proxy for external access, and a NAT gateway for secure outbound internet access from private subnets.

**#### \*\*1. VPC Configuration\*\***

The Virtual Private Cloud (VPC) is the foundation of the infrastructure, where all networking is contained within a dedicated, isolated environment.

- \*\*VPC CIDR Block\*\*: `10.0.0.0/16`

- This CIDR block provides ample IP addresses for both public and private resources.

#### \*\*2. Subnet Design\*\*

The VPC is divided into multiple availability zones (AZs) for high availability and fault tolerance. I’ve used both public and private subnets to organize resources based on their accessibility needs.

**- \*\*Public Subnets\*\*:**

- DEV-Public-Subnet-01: `10.0.100.0/24`

- DEV-Public-Subnet-02: `10.0.101.0/24`

- DEV-Public-Subnet-03: `10.0.102.0/24`

- These subnets host resources that require internet access, such as the \*\*Reverse Proxy Server\*\*, ensuring they can communicate with the outside world while still being secure.

**- \*\*Private Subnets\*\*:**

- DEV-Private-Subnet-01: `10.0.0.0/24`

- DEV-Private-Subnet-02: `10.0.1.0/24`

- DEV-Private-Subnet-03: `10.0.2.0/24`

- These subnets are isolated from direct internet access and are used for sensitive resources such as the \*\*Web Server\*\* and \*\*source code\*\*.

**#### \*\*3. Internet Gateway and Public Subnets\*\***

- \*\*Internet Gateway (IGW)\*\*: I have attached an \*\*Internet Gateway (DEV-IGW)\*\* to the VPC, allowing resources in the public subnet to access the internet.

- \*\*Reverse Proxy Server\*\*: The reverse proxy server in the public subnet handles all incoming traffic, forwarding requests to the backend web server in the private subnet. This adds an additional layer of security and load balancing.

**#### \*\*4. Private Subnets and NAT Gateway\*\***

- \*\*NAT Gateway\*\*: The NAT Gateway in the public subnet provides outbound internet access for resources in the private subnet, allowing the \*\*Web Server\*\* to download updates or connect to external services without being exposed to direct inbound traffic.

- \*\*Private Web Server\*\*: The \*\*Web Server\*\* is housed in the private subnet and is only accessible via the reverse proxy, which adds a layer of security while ensuring necessary connectivity.

**#### \*\*5. Route Tables\*\***

To control traffic flow within the VPC, I’ve created dedicated route tables for the public and private subnets.

**- \*\*Public Route Table (Dev-Public-RT)\*\*:** Associated with the public subnets, this table ensures that the resources in the public subnet can communicate with the internet via the \*\*Internet Gateway\*\*.

**- \*\*Private Route Table (Dev-Private-RT)\*\*:** The private route table directs outbound traffic from the private subnet to the internet through the \*\*NAT Gateway\*\*, ensuring that the web server remains secure while having the necessary outbound connectivity.

**#### \*\*6. Security Groups and Network ACLs\*\***

To enforce strict security measures, I’ve configured \*\*Security Groups\*\* and \*\*Network ACLs (Access Control Lists)\*\*. The web server in the private subnet is only accessible by the reverse proxy, and I’ve restricted inbound traffic to only the necessary ports and IP ranges.

---

**### \*\*Application Flow\*\***

1. \*\*User Access\*\*: Users initiate a request via the \*\*Reverse Proxy\*\*, which is publicly accessible.

2. \*\*Reverse Proxy\*\*: The reverse proxy forwards these requests to the \*\*Web Server\*\* located in the private subnet, ensuring secure internal communication.

3. \*\*Outbound Internet Access\*\*: The \*\*Web Server\*\* can make outbound requests via the \*\*NAT Gateway\*\* (e.g., for software updates) while remaining inaccessible from the public internet.

4. \*\*High Availability\*\*: The infrastructure spans multiple availability zones, ensuring redundancy and fault tolerance.

---

This project demonstrates my ability to design a secure, scalable, and highly available web application infrastructure on AWS. By utilizing public and private subnets, reverse proxy servers, and NAT gateways, I’ve ensured that the application remains secure, accessible, and ready for future scaling.

---

Does this version work for you? Let me know if you'd like to adjust any part!