AWS Cloud Security Configuration Project

Introduction

This project focuses on designing and implementing a secure cloud infrastructure using Amazon Web Services (AWS). The objective is to create a secure environment for hosting a web server and a database server while adhering to security best practices. This includes setting up subnets, configuring security groups, deploying a LAMP stack, and securing access through SSH hardening.

Project Steps and Configurations

Section 1: Subnet Creation

Objective: To create two subnets - one public for the web server and one private for the database server.

Steps Performed:
1. Configured a public subnet for the web server.
2. Configured a private subnet for the database server.
Outcome: Subnets were successfully created.

Section 2: Launch Instances

Objective: To launch two instances in their respective subnets.

Steps Performed:
1. Launched an Ubuntu instance in the public subnet for the web server.
2. Launched an Ubuntu instance in the private subnet for the database server.
Outcome: Instances were successfully launched and verified.

Section 3: Configuring Traffic

Objective: To configure network security rules to allow only necessary traffic.

Steps Performed:
1. Allowed inbound SSH traffic to the web server.
2. Configured inbound HTTP/HTTPS traffic for the web server.
3. Allowed ping traffic from the web server to the database server.
Outcome: Network security rules were successfully implemented.

Section 4: LAMP Stack Deployment

Objective: To install and configure the LAMP stack on the web server.

Steps Performed:
1. Installed Apache, MySQL, and PHP on the Ubuntu web server.

2. Configured the server and verified the default Apache "It Works" page.
Outcome: LAMP stack was successfully deployed and verified.

Section 5: SSH Configuration

Objective: To secure server access by configuring SSH settings.

Steps Performed:
1. Disabled password authentication in the SSH configuration file:

```
PasswordAuthentication no
```

2. Enabled public/private key authentication:

```
PubkeyAuthentication yes
```

3. Allowed root login securely with a public key:

```
PermitRootLogin yes
```

4. Restarted the SSH service to apply changes:

```
sudo systemctl restart sshd
```

5. Added a public key for the root user and verified root login:

```
ssh -i /path/to/private-key.pem root@<server-ip>
```

Outcome: SSH configuration was successfully completed. Password authentication was disabled, and root login via public key was enabled and verified.

Conclusion

This project demonstrates the ability to design and implement secure cloud solutions using AWS. Key achievements include the creation of secure subnets, traffic configuration, deployment of a LAMP stack, and hardening server access with SSH. These configurations align with security best practices and ensure a secure environment for hosting web and database servers.