Project 1: Implementation of a Secure AWS Environment for a Fictional Company

Detailed Report

Project Goal: Establishing a secure and scalable AWS infrastructure following best security practices.

Guide

1. **Set Up VPC:**
   - Create a new VPC and subnets using the provided commands.
   - Configure Internet gateways and route tables to manage traffic.
2. **Configure Security Groups and NACLs:**
   - Create and configure security groups for different server types.
   - Set up NACLs to control inbound and outbound traffic.
3. **Create IAM Roles and Policies:**
   - Define IAM roles and policies based on team needs.
   - Implement role-based access controls and enable MFA.
4. **Set Up Monitoring and Logging:**
   - Enable AWS CloudTrail and AWS Config for activity monitoring and logging.
   - Create alarms and notifications for security-relevant events.
5. **Implement Data Encryption:**
   - Enable encryption for stored data (S3, EBS).
   - Configure KMS keys for managing encryption.

Key Activities and Implementation:

1. **Setup a Virtual Private Cloud (VPC):**
   - Created a VPC with an IPv4 CIDR block.
   - Configured two subnets: one public and one private.

```
bash
aws ec2 create-vpc --cidr-block 10.0.0.0/16
aws ec2 create-subnet --vpc-id vpc-12345678 --cidr-block 10.0.1.0/24
--availability-zone eu-central-1a
aws ec2 create-subnet --vpc-id vpc-12345678 --cidr-block 10.0.2.0/24
--availability-zone eu-central-1b
```

2. **Configure Security Groups and NACLs:**
   - Created security groups for different applications (web server, database server).
   - Configured Network ACLs for additional security at the subnet level.

```
bash
aws ec2 create-security-group --group-name web-sg --description "Web
server security group" --vpc-id vpc-12345678
aws ec2 authorize-security-group-ingress --group-id sg-12345678 --
protocol tcp --port 80 --cidr 0.0.0.0/0
aws ec2 create-network-acl --vpc-id vpc-12345678
aws ec2 create-network-acl-entry --network-acl-id acl-12345678 --
rule-number 100 --protocol tcp --port-range From=80,To=80 --egress --
cidr-block 0.0.0.0/0 --rule-action allow
```

3. **Implement IAM:**
    - Created IAM roles and policies for various user roles (Administrator, Developer).

```
json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*"
    }
  ]
}
```

4. **Monitoring and Logging:**
    - Enabled AWS CloudTrail and AWS Config for activity monitoring and logging.

```
bash
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-
trail-bucket
aws cloudtrail start-logging --name my-trail
aws configservice put-configuration-recorder --configuration-recorder
name=default,roleARN=arn:aws:iam::123456789012:role/config-role
```

5. **Data Encryption:**
    - Enabled encryption for S3 buckets and EBS volumes.

```
bash
aws s3api put-bucket-encryption --bucket my-bucket --server-side-
encryption-configuration
'{"Rules":[{"ApplyServerSideEncryptionByDefault":{"SSEAlgorithm":"AES
256"}}]}'
aws ec2 create-volume --size 100 --region eu-central-1 --
availability-zone eu-central-1a --volume-type gp2 --encrypted
```

Result: A fully secured and scalable AWS infrastructure adhering to cloud security best practices.