Program 1: Secure an S3 Bucket with a Bucket Policy

Step 1: Log in to the AWS Management Console

1. Go to the AWS Management Console.
2. Sign in with your AWS account credentials.

Step 2: Navigate to Amazon S3

1. Search for S3 in the services search bar.
2. Click on S3 to open the Amazon S3 dashboard.

Step 3: Select the Bucket to Secure

1. In the S3 dashboard, locate and click on the name of the bucket.

Step 4: Open the Bucket Permissions

1. Click on the Permissions tab.
2. Scroll down to the Bucket Policy section.

Step 5: Create or Edit the Bucket Policy

1. Click on the Edit button.
2. A text box will appear for JSON bucket policy input.

Step 6: Write the Bucket Policy

1. Use the provided JSON template for a resource-based policy.
2. Replace placeholders with your specific values.

Step 7: Save the Bucket Policy

1. After writing/editing, click Save changes.
2. AWS will validate the policy syntax.

Step 8: Test the Bucket Policy

1. Verify policy by attempting bucket access.
2. Use AWS CLI, SDKs, or S3 console to test.

Step 9: Monitor and Audit

1. Use AWS CloudTrail to monitor S3 bucket access.
2. Regularly review the bucket policy for security.

Program 2: Configure S3 Versioning and Object Lock

Step 1: Prerequisites

1. AWS Account.
2. S3 Bucket.
3. IAM Permissions: Ensure permissions for versioning, object lock, upload/delete.

Step 2: Enable Versioning on the S3 Bucket

1. Log in to the AWS Management Console.
2. Navigate to Amazon S3.
3. Select the bucket.
4. Go to the Properties tab.
5. Scroll to Bucket Versioning.
6. Click Edit and enable Versioning.
7. Click Save changes.

Step 3: Enable Object Lock on the S3 Bucket

1. Note: Bucket must have never had versioning disabled.
2. Ensure versioning has always been enabled.
3. Go to the Properties tab of the bucket.

4. Scroll to the Object Lock section.
5. Click Enable.
6. Choose default retention mode (Governance or Compliance) and set a period.
7. Click Save.

Step 4: Upload an Object to the Bucket

1. Go to the Objects tab of the bucket.
2. Click Upload, select a file, and click Upload.

Step 5: Test Versioning

1. Upload a New Version of the Object.
2. View Object Versions.
3. Delete the Object.
4. Restore a Previous Version.

Step 6: Test Object Lock

1. Enable Object Lock for an Object.
2. Attempt to Delete the Object.
3. Attempt to Modify the Object.
4. Wait for the Retention Period to Expire.

Step 7: Analyze the Results

1. Versioning: Observe its behavior.
2. Object Lock: Observe its behavior.

Step 8: Clean Up

1. Delete Objects.
2. Disable Object Lock.
3. Disable Versioning.

Program 3: Implement S3 Encryption Methods

Step 1: Set Up Prerequisites

1. Create an AWS Account.
2. Install AWS CLI.
3. Configure AWS CLI with credentials.

Step 2: Create an S3 Bucket

1. Create a bucket using AWS Console or CLI.
2. CLI Command: aws s3api create-bucket --bucket my-encryption-experiment-bucket --region us-east-1 (replace bucket name).

Step 3: Upload a Sample File

1. Create a sample file.
2. Upload the file.

Step 4: Apply Different Encryption Methods *Amazon S3 supports several encryption options.*

Option 1: Server-Side Encryption with Amazon S3 (SSE-S3)

1. Enable SSE-S3.
2. Verify encryption by checking file status for ServerSideEncryption.

Option 2: Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

1. Create a KMS Key: Go to AWS KMS, create a key (default settings), and note Key ID.
2. Enable SSE-KMS.
3. Verify Encryption.

Option 3: Server-Side Encryption with Customer-Provided Keys (SSE-C)

1. Generate a Customer Key.
2. Enable SSE-C.
3. Verify Encryption.

Option 4: Client-Side Encryption

1. Encrypt the File Locally.
2. Upload the Encrypted File.
3. Decrypt the File locally after download.

Step 5: Clean Up

1. Delete the S3 Bucket.
2. Delete the KMS Key (if created).

Step 6: Document Your Findings

1. Record steps, commands, and outputs.
2. Compare pros and cons of each encryption option.

Program 4: Enforce HTTPS with S3 Bucket Policy

Step 1: Set Up Prerequisites

1. Create an AWS Account.
2. Install AWS CLI.
3. Configure AWS CLI.

Step 2: Create an S3 Bucket

1. Create a bucket using the AWS console.
2. CLI Command: aws s3api create-bucket --bucket my-secure-transport-bucket --region us-east-1 (replace bucket name).

Step 3: Upload a Sample File

1. Create sample.txt with some data.
2. Upload the file.

Step 4: Configure a Bucket Policy to Enforce HTTPS

1. Create a JSON file bucket-policy.json with the provided Deny policy for aws:SecureTransport: "false".
2. Apply the Bucket Policy.

Step 5: Test the Bucket Policy

1. Attempt to access the file via HTTP: Expect Access Denied error.
2. Access the file via HTTPS: Expect successful download.

Step 6: Verify the Bucket Policy

1. Use AWS CLI to check and ensure the applied policy matches.

Step 7: Clean Up

1. Delete the S3 Bucket and its contents.

Step 8: Document Your Findings

1. Record steps, commands, and outputs.
2. Explain how the policy enforces HTTPS and prevents HTTP access.

Program 5: Configure EC2 Security Groups

Step 1: Understand Security Groups

- Security Groups are virtual firewalls for EC2 instances.
- They operate at the instance level.

- Rules are stateful (inbound allows corresponding outbound).

Step 2: Access the AWS Management Console

1. Log in to the AWS Management Console.
2. Navigate to the EC2 Dashboard.

Step 3: Create a Security Group

1. In EC2, go to Network & Security > Security Groups.
2. Click Create Security Group.
3. Provide Name, Description, and VPC.
4. Click Create.

Step 4: Configure Inbound Rules

1. Select the new Security Group.
2. Click Inbound Rules tab.
3. Click Edit Inbound Rules.
4. Add rules: e.g., SSH (port 22) from specific IP (192.168.1.0/24).
5. Click Save Rules.

Step 5: Configure Outbound Rules

1. Click Outbound Rules tab.
2. Click Edit Outbound Rules.
3. Add rules: e.g., Allow all traffic (0.0.0.0/0).
4. Click Save Rules.

Step 6: Associate the Security Group with EC2 Instances

1. Go to Instances in EC2 Dashboard.
2. Select the instance.
3. Click Actions > Security > Change Security Groups.
4. Select the Security Group.
5. Click Assign Security Groups.

Step 7: Test the Configuration

1. Verify Security Group rules work as expected.
2. Double-check rules if access is denied.

Step 8: Monitor and Update Security Groups

1. Regularly review and update Security Group rules.
2. Use AWS CloudTrail to monitor changes.
3. Consider AWS Config for compliance.

Program 6: Launch and Connect to an EC2 Instance

Step 1: Sign in to AWS Console

1. Go to https://aws.amazon.com.
2. Click Sign In to the AWS Management Console.

Step 2: Navigate to EC2 Dashboard

1. Search for EC2 and select it.
2. This opens the EC2 Dashboard.

Step 3: Launch an Instance

1. Click "Launch Instance" button.
2. Enter a name for your instance.

Step 4: Choose an Amazon Machine Image (AMI)

1. Choose an AMI (e.g., Amazon Linux 2 AMI).

2. Click Select.

Step 5: Choose an Instance Type

1. Select the instance type.
2. Click Next: Configure Instance Details.

Step 6: Configure Key Pair

1. Create or select a key pair.
2. Download and save the .pem file securely (needed for SSH).
3. Find your machine's IPv4 address via ipconfig in Command Prompt.

Step 7: Configure Network Settings (Security Group)

1. Click Edit security groups.
2. Create a new security group.
3. Add inbound rules.
4. Click Add Rule for each.
5. **Configure Security Group (Allow Network Traffic):**
    - Create or use existing security group.
    - Add Inbound rules:
        - SSH (22): Protocol TCP, Port 22, Source: Your IP (recommended) or Anywhere.
        - HTTP (80): Protocol TCP, Port 80, Source: Anywhere (0.0.0.0/0).
        - HTTPS (443): Protocol TCP, Port 443, Source: Anywhere.

Step 8: Launch the Instance

1. Review settings.
2. Click Launch Instance.

Step 9: Access Your EC2 Instance *Once the instance is running:*

1. Copy the Public IPv4 address.
2. Open terminal/command prompt.
3. Run: chmod 400 your-key.pem.
4. Run: ssh -i "your-key.pem" ec2-user@your-public-ip.
5. Alternatively, select your instance, click "Connect", choose SSH client, and follow instructions.

Program 7: Block Traffic with Network ACLs (NACLs)

Step 1: Sign In and Navigate to VPC Dashboard

1. Sign in to the AWS Console.
2. Search for VPC and select it.

Step 2: Find or Create a Network ACL

1. In the left sidebar, click "Network ACLs".
2. Select an existing NACL or click "Create Network ACL".

Step 3: Associate NACL with Your Subnet *If it's a new NACL:*

1. After creation, go to "Subnet associations" tab.
2. Click "Edit subnet associations".
3. Select the subnet where your EC2 instance is running.
4. Click Save.

Step 4: Add Inbound and Outbound Deny Rules for the TCP Port
Inbound Rule

1. Go to "Inbound Rules" tab.

2. Click "Edit inbound rules", then "Add new rule".
3. Configure: Rule number (e.g., 100), Type: Custom TCP, Protocol: TCP (6), Port range: 8080 (or port to block), Source: 0.0.0.0/0, Allow/Deny: DENY.

Outbound Rule

1. Go to "Outbound Rules" tab.
2. Click "Edit outbound rules", then "Add new rule".
3. Configure: Rule number (e.g., 100), Type: Custom TCP, Protocol: TCP (6), Port range: 8080, Destination: 0.0.0.0/0, Allow/Deny: DENY.

Step 5: Test the Port Blocking

1. Test with curl or telnet from another server.
2. Use online port scanning tools to check if the port is blocked.