Securing Amazon S3 VPC Endpoint Communications

1. Objective

The goal of this project is to securely access data from Amazon S3 using Amazon VPC endpoints from an EC2 instance located in a private subnet. This project aims to illustrate how AWS VPC endpoints can enable secure communication with Amazon S3 without exposing private resources to the public internet. By employing a VPC endpoint policy, the solution further restricts access to specific resources, enhancing data security.

2. Prerequisites

- **AWS Account**: Set up an AWS free-tier account.
- **Basic Networking Knowledge**: Understanding of VPC, subnets, and routing.
- **Software**:
    - AWS Management Console for resource provisioning.
    - AWS CLI for command-line operations.

3. Technology Stack

- **AWS VPC**: For creating the network infrastructure with public and private subnets.
- **Amazon EC2**: For launching instances to access and test communications.
- **Amazon S3**: For storing and retrieving demo files securely.
- **AWS CLI**: For configuring and managing AWS services through the command line.

4. Step-by-Step Guide

Step 1: Set Up AWS VPC and Subnets

- **Create a VPC**:
    - Navigate to the VPC console and create a new VPC (e.g., MyVPC) with a CIDR block of 10.0.0.0/16.
- **Create Public and Private Subnets**:
    - **Public Subnet**:
        - Name: PublicSubnet
        - CIDR block: 10.0.1.0/24
    - **Private Subnet**:
        - Name: PrivateSubnet
        - CIDR block: 10.0.2.0/24
- **Attach an Internet Gateway**:
    - Create and attach an Internet Gateway (e.g., MyIGW) to the VPC.
- **Configure Route Tables**:
    - Create a route table for the public subnet that routes 0.0.0.0/0 to the Internet Gateway.
    - Leave the private subnet route table with no internet access (no NAT Gateway or Internet Gateway).

Step 2: Launch EC2 Instances

- **Launch EC2 Instances**:
  - **Public EC2 Instance**:
    - Name: PublicInstance
    - AMI: Amazon Linux 2
    - Instance Type: t3.micro (free-tier eligible)
    - Attach an IAM role with permissions for S3 access.
  - **Private EC2 Instance**:
    - Name: PrivateInstance
    - AMI: Amazon Linux 2
    - Instance Type: t3.micro
    - Attach the same IAM role as the public instance for S3 access.

Step 3: Create an S3 Bucket

- **Create an S3 Bucket**:
  - Name: my-s3-bucket-secure-access
  - Upload a demo file (e.g., demo-file.txt) for testing access from the private subnet.

Step 4: Set Up the VPC Endpoint

- **Create VPC Endpoint for S3**:
  - Navigate to the VPC console and create an endpoint for Amazon S3.
  - Select the VPC and attach the endpoint to the route table of the private subnet.

```
ap-southeast-2

{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-0c989183cee0b0772",
        "VpcEndpointType": "Gateway",
        "VpcId": "vpc-0d83f56632e948aad",
        "ServiceName": "com.amazonaws.ap-southeast-2.s3",
        "State": "available",
        "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*
\"}]}",
        "RouteTableIds": [
            "rtb-00a1f1774f58a4686"
        ],
        "SubnetIds": [],
        "Groups": [],
        "PrivateDnsEnabled": false,
        "RequesterManaged": false,
        "NetworkInterfaceIds": [],
        "DnsEntries": [],
        "CreationTimestamp": "2024-10-10T16:21:31+00:00",
        "OwnerId": "022164167735"
    }
}
```

```
[ssm-user@ip-10-0-1-4 ~] $ aws ec2 describe-vpc-endpoints --query 'VpcEndpoints[*].ServiceName'

[
    "com.amazonaws.ap-southeast-2.cloudformation",
    "com.amazonaws.ap-southeast-2.ssmmessages",
    "com.amazonaws.ap-southeast-2.ec2messages",
    "com.amazonaws.ap-southeast-2.ssm",
    "com.amazonaws.ap-southeast-2.s3"
]
```

- **Configure VPC Endpoint Policy**:
    - Set a policy that restricts access to the specific S3 bucket:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": "*",
   "Action": "s3:GetObject",
   "Resource": "arn:aws:s3:::my-s3-bucket-secure-access/*"
  }
 ]
}
```

Step 5: Verify Access

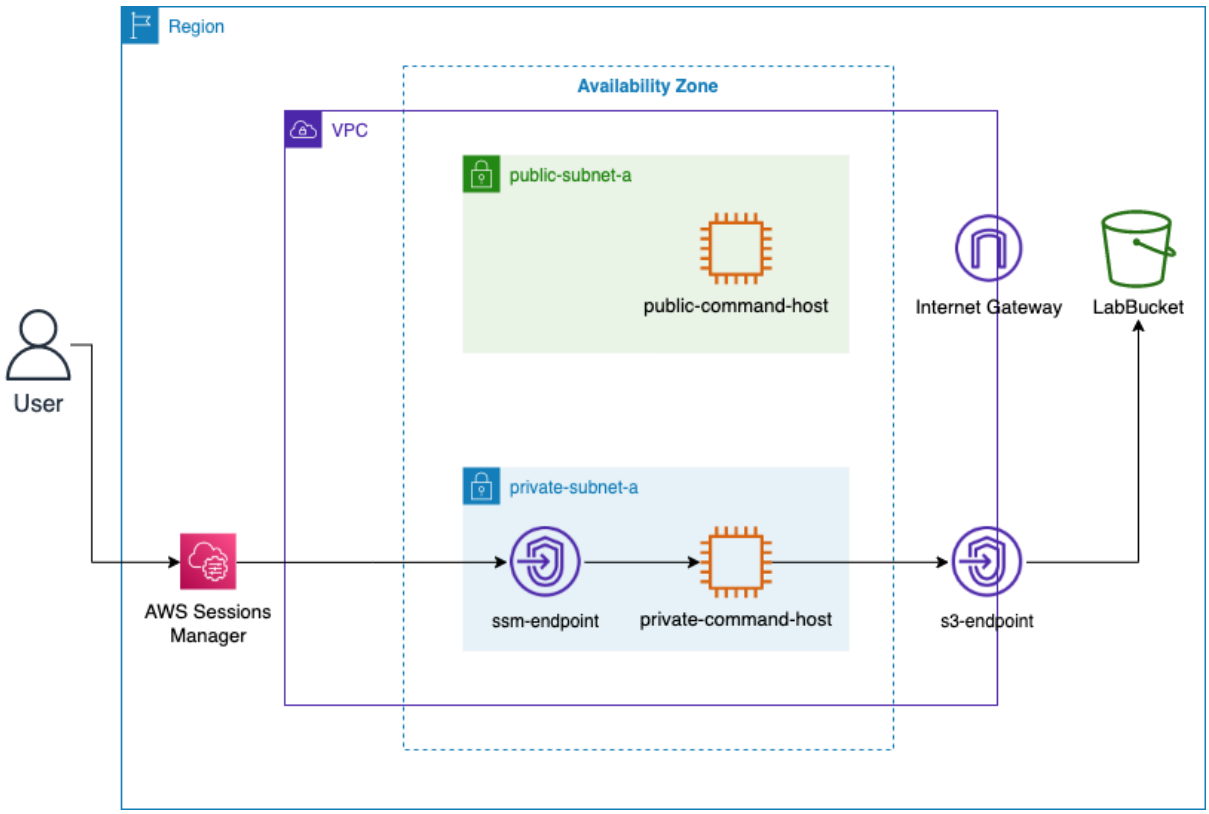
- **Log into EC2 Instances**:
    - Use SSH to connect to the PublicInstance and PrivateInstance.
- **Testing S3 Access**:
    - From the **Public EC2 instance**, verify access to the S3 bucket:

      aws s3 ls s3://my-s3-bucket-secure-access

    - From the **Private EC2 instance**, also verify access:

      aws s3 ls s3://my-s3-bucket-secure-access

5. Diagram of Project Workflow



This diagram illustrates the workflow, starting from the public instance, through the VPC endpoint, allowing the private instance to access the S3 bucket securely.

6. Conclusion

This project successfully demonstrated the use of **Amazon S3 VPC endpoints** for secure communications between an EC2 instance in a private subnet and S3. By leveraging AWS services, this solution effectively prevents exposure of private resources to the public internet, enhancing data security. The implementation of a VPC endpoint policy further reinforced access control to sensitive data, showcasing best practices in cloud security. This project provides a foundational understanding of secure network architecture in AWS, which can be beneficial for future cloud deployments.