

Step 1: Create an S3 Bucket and Upload a File

1. Create an S3 Bucket:

- Open the **AWS Management Console**.
- Go to **S3** → Click **Create Bucket**.
- Enter a **unique bucket name** (e.g., mys3demobucket2020).
- Choose a **region** and keep other settings default.
- Click **Create bucket**.

2. Upload a File to the S3 Bucket:

- Open the **S3 bucket** you just created.
- Click **Upload** → **Add files**.
- Select a file (e.g., aws.txt).
- Click **Upload** to store the file in S3.

Step 2: Create an IAM Role for S3 Read-Only Access

1. Open the IAM Console:

- Go to the **AWS Management Console** → Open the **IAM** service.

2. Create a New IAM Role:

- Click **Roles** → **Create Role**.
- Select **AWS Service** as the trusted entity.
- Choose **EC2** as the use case.
- Click **Next**.

3. Attach the S3 Read-Only Policy:

- In the **Permissions** section, search for AmazonS3ReadOnlyAccess.
- Select it and click **Next**.

4. Name the Role:

- Enter the role name as **EC2S3ReadOnlyRole**.
- Click **Create role**.

5. Attach the IAM Role to the EC2 Instance:

- Open the **EC2 console** → Select your **EC2 instance**.
- Click **Actions** → **Security** → **Modify IAM Role**.
- Choose **EC2S3ReadOnlyRole** from the dropdown and click **Update IAM Role**.

Step 3: Verify IAM Role by Accessing S3 from EC2

1. Connect to the EC2 Instance:

- Open **Command Prompt (Windows)** or **Terminal (Mac/Linux)**.
- Use SSH to connect to the EC2 instance:

RUN : `ssh -i "C:\Users\Harshitha Basavaraju\Downloads\myec2instance.pem" ec2-user@<EC2-Public-IP>`

Type yes when prompted to confirm the connection.

```
C:\Users\Harshitha Basavaraju>ssh -i "C:\Users\Harshitha Basavaraju\Downloads\myec2instance.pem" ec2-user@35.91.156.254
The authenticity of host '35.91.156.254 (35.91.156.254)' can't be established.
ED25519 key fingerprint is SHA256:/4x3M9bACPnJUSUchXKhIWT5P2LLTXFaBeitEdRBBUQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '35.91.156.254' (ED25519) to the list of known hosts.

#_
~\_####_ Amazon Linux 2023
NN\_#####\
NN\_###|
NN\_#/ --- https://aws.amazon.com/linux/amazon-linux-2023
NN\_V~! ->
NNN
NN\_./\_
\_m/!

[ec2-user@ip-172-31-22-27 ~]$ ^C
[ec2-user@ip-172-31-22-27 ~]$ ^C
[ec2-user@ip-172-31-22-27 ~]$
[ec2-user@ip-172-31-22-27 ~]$ aws --version
```

Check if AWS CLI is Installed:

- Run the following command to verify AWS CLI installation: **aws --version**
- If installed, you should see an output like:

```
[ec2-user@ip-172-31-22-27 ~]$ aws --version
aws-cli/2.23.11 Python/3.9.21 Linux/6.1.131-143.221.amzn2023.x86_64 source/x86_64.amzn.2023
```

List S3 Buckets:

- Run the command: **aws s3 ls**
- Run the command to download a file from the S3 bucket:
aws s3 cp s3://mys3demobucket2020/aws.txt .
- Read from S3 : **aws s3 cp s3://mys3demobucket2020/aws.txt .**

```
[ec2-user@ip-172-31-22-27 ~]$ aws s3 ls s3://mys3demobucket2020
2025-04-02 17:21:02          6328 aws.txt
```

- Delete from S3: **aws s3 rm s3://mys3demobucket2020/aws.txt**

Since the IAM role EC2S3ReadOnlyRole only grants read access, the delete command should return an "Access Denied" error. This confirms that the role is correctly configured with read-only permissions.

```
[ec2-user@ip-172-31-22-27 ~]$ aws s3 rm s3://mys3demobucket2020/aws.txt
delete failed: s3://mys3demobucket2020/aws.txt An error occurred (AccessDenied) when calling the DeleteObject operation: User: arn:aws:sts::949847155882:ass
umed-role/EC2S3ReadOnlyRole/i-06a62b9a1f0afa7ce is not authorized to perform: s3:DeleteObject on resource: "arn:aws:s3::mys3demobucket2020/aws.txt" becaus
e no identity-based policy allows the s3:DeleteObject action
[ec2-user@ip-172-31-22-27 ~]$ client_loop: send disconnect: Connection reset
```