Amazon Linux & Ubuntu: Mount S3 Bucket Using s3fs-fuse

Introduction

This guide explains how to mount an Amazon S3 bucket on Amazon Linux and Ubuntu using s3fs-fuse. It also includes steps to enable auto-mount after reboot, configure IAM roles, and troubleshoot common issues.

Step 1: Create an S3 Private Bucket and Upload Files

1. Go to the AWS S3 Console.
2. Click Create Bucket.
3. Enter a unique bucket name.
4. Uncheck Block all public access (if needed, keep it private).
5. Click Create Bucket.
6. Upload files by clicking Upload and selecting files.

Step 2: Create IAM Policy, Role, and Attach to Instances

Step 2.1: Create an IAM Policy

1. Go to AWS IAM Console -> Policies -> Create Policy.
2. Select the JSON tab and paste the following policy (replace YOUR_BUCKET_NAME):

IAM Policy JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::YOUR_BUCKET_NAME",
        "arn:aws:s3:::YOUR_BUCKET_NAME/*"
      ]
    }
  ]
}
```

Step 2.2: Create an IAM Role and Attach Policy

1. Go to AWS IAM Console -> Roles -> Create Role.
2. Select EC2 as the trusted entity.
3. Click Next and attach the recently created policy (S3AccessPolicy).
4. Click Next, name the role (e.g., S3AccessRole), and create it.

Step 2.3: Attach Role to EC2 Instances

1. Go to EC2 Console -> Instances.
2. Select your Amazon Linux and Ubuntu instances.
3. Click Actions -> Security -> Modify IAM Role.
4. Select S3AccessRole and click Update IAM Role.

Step 3: Launch 2 Ubuntu Instances

Step 4: Amazon Linux: S3 Mounting and Auto-Mount on Reboot

Step 4.1: Install Dependencies

```
sudo yum update -y
sudo yum install -y automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel make openssl-devel
```

Step 4.2: Install s3fs-fuse

```
git clone https://github.com/s3fs-fuse/s3fs-fuse.git
cd s3fs-fuse
./autogen.sh
./configure --prefix=/usr --with-openssl
make
sudo make install
s3fs --version  # Verify installation
```

Step 4.3: Create & Configure Mount Point

```
sudo mkdir -p /mnt/s3bucket
sudo chown ec2-user:ec2-user /mnt/s3bucket
sudo chmod 777 /mnt/s3bucket
```

Step 4.4: Mount the S3 Bucket

```
s3fs YOUR_BUCKET_NAME /mnt/s3bucket -o iam_role=auto -o allow_other -o use_cache=/tmp
```

Step 4.5: Enable Auto-Mount After Reboot

```
echo "s3fs#YOUR_BUCKET_NAME /mnt/s3bucket fuse _netdev,allow_other,use_cache=/tmp,iam_role=auto 0 0" | sudo tee -a /etc/fstab
sudo mount -a
df -h /mnt/s3bucket  # Verify mount
```

Step 5: Ubuntu: S3 Mounting and Auto-Mount on Reboot

Step 5.1: Install Dependencies

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config
```

Step 5.2: Install s3fs-fuse

```
git clone https://github.com/s3fs-fuse/s3fs-fuse.git
cd s3fs-fuse
./autogen.sh
./configure --prefix=/usr --with-openssl
make
sudo make install
s3fs --version  # Verify installation
```

Step 5.3: Enable 'allow_other' in FUSE Configuration

```
sudo nano /etc/fuse.conf
```

Add this line (or uncomment if already present):

```
user_allow_other
```

Step 5.4: Mount the S3 Bucket

```
s3fs YOUR_BUCKET_NAME /mnt/s3bucket -o iam_role=auto -o allow_other -o use_cache=/tmp
```

Step 5.5: Enable Auto-Mount After Reboot

```
echo "s3fs#YOUR_BUCKET_NAME /mnt/s3bucket fuse _netdev,allow_other,use_cache=/tmp,iam_role=auto 0 0" | sudo tee -a /etc/fstab
sudo mount -a
df -h /mnt/s3bucket  # Verify mount
```

Conclusion

Now your S3 bucket is properly mounted and will persist after reboots!