**How to Mount S3 bucket in AWS**

A S3 bucket can be mounted in a AWS instance as a file system known as S3fs. S3fs is a FUSE file-system that allows you to mount an Amazon S3 bucket as a local file-system. It behaves like a network attached drive, as it does not store anything on the Amazon EC2, but user can access the data on S3 from EC2 instance.

Filesystem in Userspace (FUSE) is a simple interface for userspace programs to export a virtual file-system to the Linux kernel. It also aims to provide a secure method for non privileged users to create and mount their own file-system implementations.

S3fs-fuse project is written in python backed by Amazons Simple Storage Service. Amazon offers an open API to build applications on top of this service, which several companies have done, using a variety of interfaces (web, rsync, fuse, etc).

Follow the below steps to mount your S3 bucket to your Linux Instance.

This Tutorial assumes that you have a running Linux EC2 instance on AWS with root access and a bucket created in S3 which is to be mounted on your Linux Instance. You will also require Access and Secret key pair with sufficient permissions of S3 or else an IAM access to generate or Create it.

We will perform the steps as a root user. You can also use sudo command if you are a normal user with sudo access. So lets get started.

Step-1:- If you are using a new centos or ubuntu instance. Update the system.

-> For CentOS or Red Hat

#yum update all

-> For Ubuntu

#apt-get update

Step-2:- Install the dependencies.

-> In CentOS or Red Hat

#sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config

Step-3:- Clone s3fs source code from git.

#git clone https://github.com/s3fs-fuse/s3fs-fuse.git

Step-4:- Now change to source code directory, and compile and install the code with the following commands:

#cd s3fs-fuse

#./autogen.sh

#./configure --prefix=/usr --with-openssl

#make

#sudo make install

Step-5:- Use below command to check where s3fs command is placed in O.S. It will also tell you the installation is ok.

#which s3fs

Step-6:- Getting the access key and secret key.

You will need AWS Access key and Secret key with appropriate permissions in order to access your s3 bucket from your EC2 instance. You can easily manage your user permissions from IAM (Identity and Access Management) Service provided by AWS. Create an IAM user with S3 full access(or with a role with sufficient permissions) or use root credentials of your Account. Here we will use the root credentials for simplicity.

Go to AWS Menu -> Your AWS Account Name -> My Security Credentials. Here your IAM console will appear. You have to go to Users > Your Account name and under permissions Tab, check whether you have sufficient access on S3 bucket. If not, you can manually assign an existing “S3 Full-Access” policy or create a new policy with sufficient permissions.

Now go to Security Credentials Tab and Create Access Key. A new Access Key and Secret Key pair will be generated. Here you can see access key and secret key (secret key is visible when you click on show tab) which you can also download. Copy these both keys separately.

Note that you can always use an existing access and secret key pair. Alternatively, you can also create a new IAM user and assign it sufficient permissions to generate the access and secret key.

Step-7 :- Create a new file in /etc with the name passwd-s3fs and Paste the access key and secret key in the below format .

#touch /etc/passwd-s3fs

#vim /etc/passwd-s3fs

#Your\_accesskey:Your\_secretkey

Step-8:- change the permission of file

#sudo chmod 640 /etc/passwd-s3fs

Step-9:- Now create a directory or provide the path of an existing directory and mount S3bucket in it.

If you have a simple bucket without dot(.) in the bucket name, use the commands used in point “a” or else for bucket with dot(.) in bucket name, follow step “b”:

a) Bucket name without dot(.):

#mkdir /mys3bucket

#s3fs your\_bucketname -o use\_cache=/tmp -o allow\_other -o uid=1001 -o mp\_umask=002 -o multireq\_max=5 /mys3bucket

where, “your\_bucketname” = the name of your S3 bucket that you have created on AWS S3, use\_cache = to use a directory for its cache purpose, allow\_other= to allow other users to write to the mount-point, uid= uid of the user/owner of the mountpoint (can also add “-o gid=1001” for group), mp\_umask= to remove other users permission. multireq\_max= parameter to send request to s3 bucket, /mys3bucket= mountpoint where the bucket will be mounted.

You can make an entry in /etc/rc.local to automatically remount after reboot. Find the s3fs binary file by “which” command and make the entry before the “exit 0” line as below.

#which s3fs

/usr/local/bin/s3fs

#vi /etc/rc.local

#/usr/local/bin/s3fs your\_bucketname -o use\_cache=/tmp -o allow\_other -o uid=1001 -o mp\_umask=002 -o multireq\_max=5 /mys3bucket

b) Bucket name with dot(.):

# mkdir /mys3bucket

#s3fs your\_bucketname /mys3bucket -o use\_cache=/tmp -o allow\_other -o uid=1001 -o mp\_umask=002 -o multireq\_max=5 -o use\_path\_request\_style -o url=https://s3-{{aws\_region}}.amazonaws.com

where, “your\_bucketname” = the name of your S3 bucket that you have created on AWS S3, use\_cache = to use a directory for its cache purpose, allow\_other= to allow other users to write to the mount-point, uid= uid of the user/owner of the mountpoint (can also add “-o gid=1001” for group), mp\_umask= to remove other users permission. multireq\_max= parameter to send request to s3 bucket, /mys3bucket= mountpoint where the bucket will be mounted .

Remember to replace “{{aws\_region}}” with your bucket region (example: eu-west-1).

You can make an entry in /etc/rc.local to automatically remount after reboot. Find the s3fs binary file by “which” command and make the entry before the “exit 0” line as below.

#which s3fs /usr/local/bin/s3fs

#vi /etc/rc.local

# s3fs your\_bucketname /mys3bucket -o use\_cache=/tmp -o allow\_other -o uid=1001 -o mp\_umask=002 -o multireq\_max=5 -o use\_path\_request\_style -o url=https://s3-{{aws\_region}}.amazonaws.com

To debug at any point, add “-o dbglevel=info -f -o curldbg” in the s3fs mount command.

Step-10:- Check mounted s3 bucket. Output will be similar as shown below but Used size may differ.

#df -Th

“or”

# df -Th /mys3bucket

Filesystem Type Size Used Avail Use% Mounted on

s3fs fuse.s3fs 256T 0 256T 0% /mys3bucket

If it shows the mounted file system, you have successfully mounted the S3 bucket on your EC2 Instance. You can also test it further by creating a test file.

# cd /mys3bucket

#echo "this is a test file to check s3fs" >> test.txt

#ls

This change should also reflect on S3 bucket. So Login to your S3 bucket to verify if the test file is present or not.

Note : If you already had some data in s3bucket and it is not visible, then you have to set permission in ACL at the S3 AWS management console for that s3 bucket.

Also, If you get any s3fs error such as “transport end point is not connected”, you have to unmount and remount the file-system. You can also do so through a custom script to detect and perform remount automatically.

Congrats!! You have successfully mounted your S3 bucket to your EC2 instance. Any files written to /mys3bucket will be replicated to your Amazon S3 bucket.