EXPLORE DIGITAL SKILLS

Mounting an S3 bucket to an EC2 instance

Train Overview

In this train, we will learn how to mount an Amazon S3 bucket to an Amazon EC2 instance using the command-line interface.

1. Introduction
2. Process overview
3. Mounting an S3 bucket to an EC2 instance
4. Automating the mounting process

Introduction

In previous AWS tutorials, we saw how to start a remote compute instance (EC2), as well as how to **transfer data** to it (**git, scp**). These processes **work well for smaller projects**, but leave some open questions when our data becomes more complex.

- What happens when our **data collection is too large** to be stored within Git?
- How do we **ensure that our data are consistent** when working in a large team?
- Is there a way to **share our data with other parties in a secure and reliable manner**?

To answer these questions, we'll make use of the **Amazon S3 service** and learn about the process involved in **mounting an S3 bucket onto our EC2 instance.**

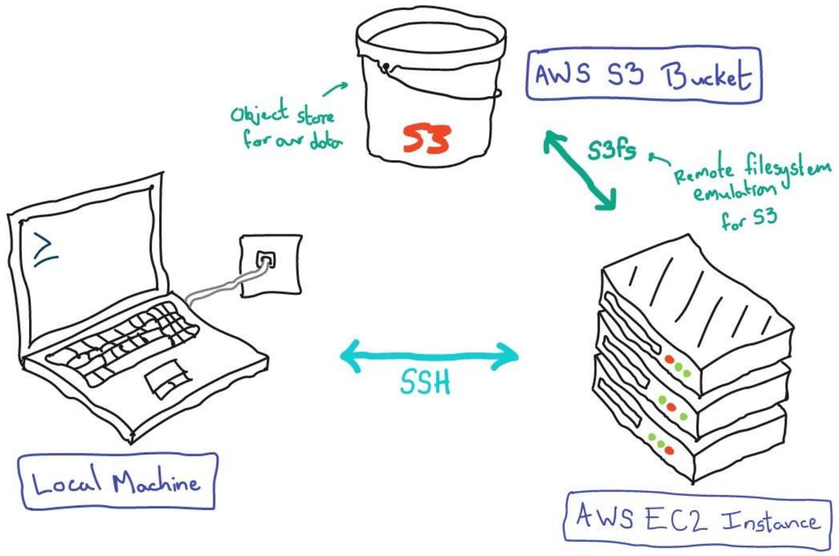
EXPLORE DIGITAL SKILLS

Process Overview

Mounting is a procedure that involves **making an operating system recognize a storage device** of some kind so that you can interact with it. In this case, we're going to make our EC2 instance recognize an EDSA owned S3 bucket so that we can read files from it.

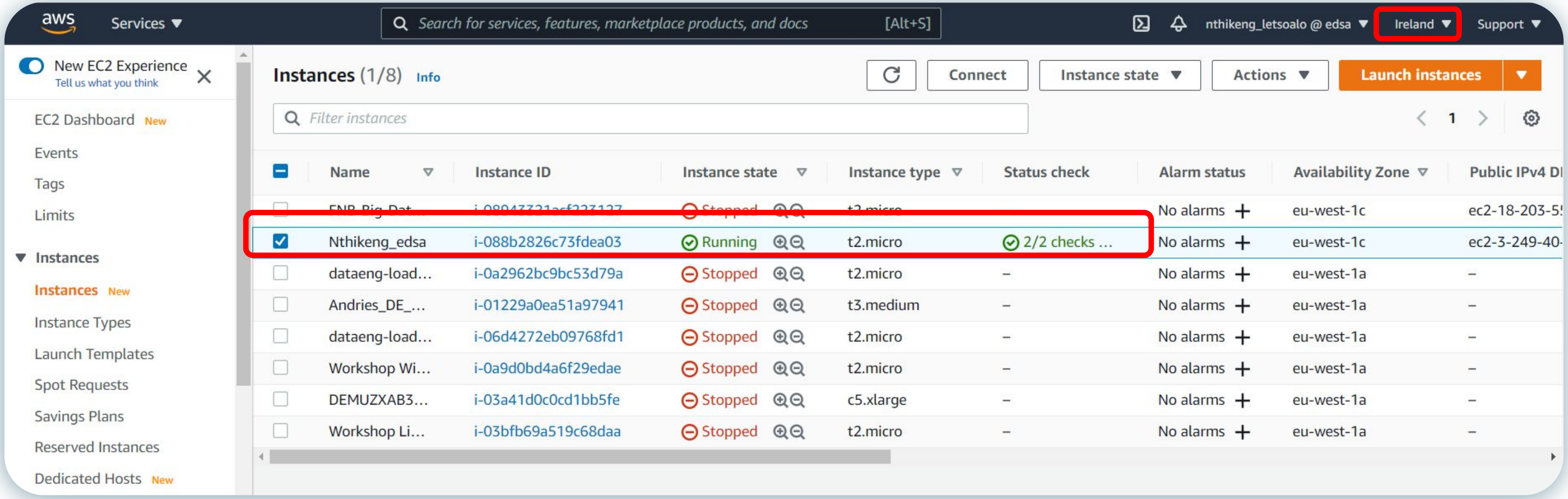
To successfully mount an S3 bucket to an EC2 instance, we'll need to do several things:

- **Give permission to our EC2 instance** to access the S3 bucket using an AWS IAM role.
- Login to our instance and **install the S3FS client application**.
- **Create a folder as a 'mount point'**. The contents of the S3 bucket will be placed inside this folder.
- **Use S3FS to recognise the S3 bucket** and mount it onto our filesystem.
- **Create a cron command** to mount this bucket every time our system restarts.



Step 1: Obtain a Running EC2 Instance

As an initial step, we **need to have access to an EC2 instance**. In this example we will be running our instance within the ‘eu-west-1’ (Ireland) region.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DI
FNB-Big-Dat...	i-08943321acf223127	Stopped	t2.micro		No alarms	eu-west-1c	ec2-18-203-5
Nthikeng_edsa	i-088b2826c73fdea03	Running	t2.micro	2/2 checks ...	No alarms	eu-west-1c	ec2-3-249-40-
dataeng-load...	i-0a2962bc9bc53d79a	Stopped	t2.micro	–	No alarms	eu-west-1a	–
Andries_DE_...	i-01229a0ea51a97941	Stopped	t3.medium	–	No alarms	eu-west-1a	–
dataeng-load...	i-06d4272eb09768fd1	Stopped	t2.micro	–	No alarms	eu-west-1a	–
Workshop Wi...	i-0a9d0bd4a6f29edae	Stopped	t2.micro	–	No alarms	eu-west-1a	–
DEMUZXAB3...	i-03a41d0c0cd1bb5fe	Stopped	c5.xlarge	–	No alarms	eu-west-1a	–
Workshop Li...	i-03bfb69a519c68daa	Stopped	t2.micro	–	No alarms	eu-west-1a	–

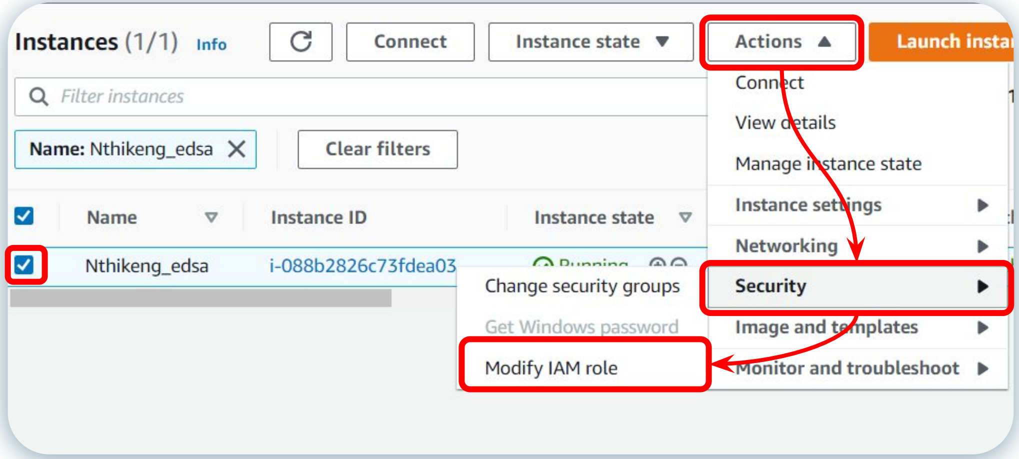
This **can be an instance that you’ve used for some time or a freshly created one**.

If you don’t have an EC2 instance or have forgotten how to spin one up, have a look at an earlier Introduction to EC2 tutorial for a refresher.

Step 2: Attach IAM Role to EC2 Instance

For this next step, we'll **grant our EC2 instance permission to mount the S3 bucket**. We do this by **modifying the IAM role** of an instance via the AWS console.

1. Click on the **checkbox** next to your instance to make sure it has been selected.
2. Click on the **Actions** drop-down menu.
3. Navigate to **Security.**
4. Under the **Security** sub-menu, select **Modify IAM role** - this should take you to a new page.



Instances (1/1) Info

Filter instances

Name: Nthikeng_edsa

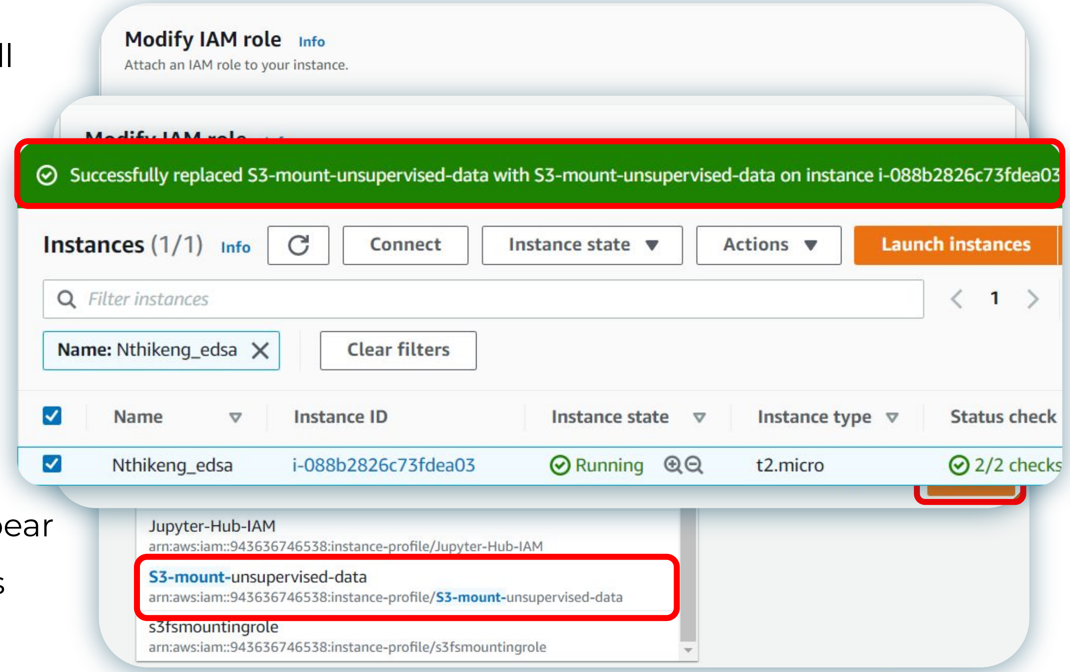
	Name	Instance ID	Instance state
✓	Nthikeng_edsa	i-088b2826c73fdea03	Running

Step 2: Attach IAM Role to EC2 Instance

For this next step, we'll **grant our EC2 instance permission to mount the S3 bucket**. We do this by **modifying the IAM role** of an instance via the AWS console.

When the **Modify IAM role** page pops up, we will do the following:

1. Under the drop-down menu for IAM role, search for **S3-mount-unsupervised-data** and select it.
2. Click **Save** to apply your new IAM role modification.
3. A **green confirmation banner** should appear at the top of your page if the IAM role was successfully applied.



Modify IAM role Info
Attach an IAM role to your instance.

Successfully replaced S3-mount-unsupervised-data with S3-mount-unsupervised-data on instance i-088b2826c73fdea03

Instances (1/1) Info | Connect | Instance state | Actions | Launch instances

Filter instances

Name: Nthikeng_edsa | Clear filters

Name	Instance ID	Instance state	Instance type	Status check
Nthikeng_edsa	i-088b2826c73fdea03	Running	t2.micro	2/2 checks

Jupyter-Hub-IAM
arn:aws:iam::943636746538:instance-profile/Jupyter-Hub-IAM
S3-mount-unsupervised-data
arn:aws:iam::943636746538:instance-profile/S3-mount-unsupervised-data
s3fsmountingrole
arn:aws:iam::943636746538:instance-profile/s3fsmountingrole

Step 3: Install Software Dependencies

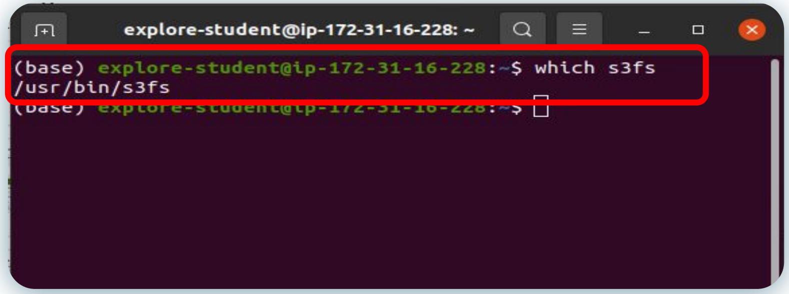
We now need to **install the S3FS client application on our instance.** The following commands are lengthy. To avoid errors during input **we recommend pasting** these from this handy GitHub gist. These commands are **identical** on **MAC(via terminal)** or **Windows (via Git Bash)**

1. Connect to your remote EC2 instance with the following command:

```
ssh explore-student@<Your EC2 Instance Public IPv4 Address>
```

2. Enter the following commands, line by line, into the terminal:

```
sudo apt-get install automake autotools-dev fuse g++ git libcurl4-gnutls-dev libfuse-dev libssl-dev libxml2-dev make pkg-config -y
git clone https://github.com/s3fs-fuse/s3fs-fuse.git
cd s3fs-fuse/
./autogen.sh
./configure --prefix=/usr --with-openssl
make
sudo make install
which s3fs
```



```
(base) explore-student@ip-172-31-16-228:~$ which s3fs
/usr/bin/s3fs
(base) explore-student@ip-172-31-16-228:~$
```

3. If the installation was successful, the last command should return the directory where the S3FS binary is located.

Terminal Commands Explained

At first glance, the installation commands might look a bit daunting. We will go through each command and briefly describe what is being done at each stage.

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

This command installs all the tools and packages that are required to successfully mount your S3 bucket.

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

Clones the Github repository that has the **S3SFS client application** to your local instance directory.

```
cd s3fs-fuse/
```

Moves to the **s3fs-fuse** folder which contains the necessary installation and configuration items.

```
./autogen.sh
```

Executes the **autogen.sh** shell script contained within the directory.

Terminal Commands Explained

At first glance, the installation commands might look a bit daunting. We will go through each command and briefly describe what is being done at each stage.

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

Configures any parameters and dependencies that are required, in our case we've also specified where the installation is to take place and that we want to use the openssl package.

```
make
```

Builds/Compiles the application from the source code.

```
sudo make install
```

Installs the built application to the appropriate directories.

```
which s3fs
```

Returns the directory where the S3FS binary is located.

Step 4: Mount S3 bucket to Target Directory

With S3FS installed, we can now mount the S3 bucket. To do this, we first create a folder as a mount point, and then provide S3FS with details to perform the mount operation.

1. Change to your instance home directory and create a folder: **unsupervised_data** as the mount point:

```
cd ~/ && mkdir unsupervised_data.
```

2. Use the following command to perform the mount operation:

```
s3fs -o iam_role="S3-mount-unsupervised-data" -o

url="https://s3-eu-west-1.amazonaws.com" -o endpoint=eu-west-1 -o dbglevel=info -o

curldbg edsa-2020-unsupervised-predict unsupervised_data
```

3. Check that the command was successful by running:

```
ls -Rla unsupervised_data/
```

You've now officially mounted your first S3 bucket, congrats!

EXPLORE DIGITAL SKILLS

Terminal Command Explained

We will breakdown the mount command so that it becomes digestible. Unpacking it will assist with increasing your level of understanding and allowing you to personalise your own commands in the future.

`s3fs` ➔ Lets the computer know that we want to make use of the S3FS application.

`-o iam_role="S3-mount-unsupervised-data"` ➔ Specifies the IAM role.

`-o url="https://s3-eu-west-1.amazonaws.com"` ➔ Set the url to use to access Amazon S3.

`-o endpoint=eu-west-1` ➔ Specify the endpoint according to the region.

`-o dbglevel=info` ➔ Specifies the debug level, which is the level of detail you want to see as the command is executing.

`-o curldbg` ➔ Displays the debug messages from libcurl.

`edsa-2020-unsupervised-predict unsupervised_data` ➔ Name of the bucket that you want to mount to the **unsupervised_data** mount point.

Step 5: Automating the Process

While our S3 bucket is now mounted, as it currently stands **we need to repeat the steps** on the previous slide **each time our instance restarts**. Instead, to avoid this pain, **we'll create a script to run our mount command at startup with cron.**

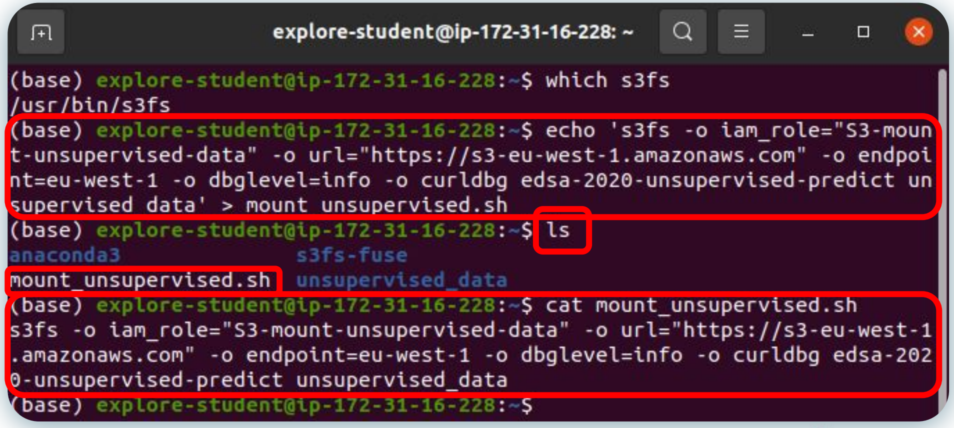
1. Create a shell script containing the mount command from the previous slide (step 4):

```
echo 's3fs -o iam_role="S3-mount-unsupervised-data" -o url="https://s3-eu-west-1.amazonaws.com"
-o endpoint=eu-west-1 -o dbglevel=info -o curldbg edsa-2020-unsupervised-predict
unsupervised_data' > mount_unsupervised.sh
```

2. If you use the `ls` command you should see a new shell script created in your directory.

3. If you wish to see the contents of the script you can use the below command:

```
cat mount_unsupervised.sh
```



```
(base) explore-student@ip-172-31-16-228:~$ which s3fs
/usr/bin/s3fs
(base) explore-student@ip-172-31-16-228:~$ echo 's3fs -o iam_role="S3-moun
t-unsupervised-data" -o url="https://s3-eu-west-1.amazonaws.com" -o endpoi
nt=eu-west-1 -o dbglevel=info -o curldbg edsa-2020-unsupervised-predict un
supervised_data' > mount_unsupervised.sh
(base) explore-student@ip-172-31-16-228:~$ ls
anaconda3                s3fs-fuse
mount_unsupervised.sh    unsupervised_data
(base) explore-student@ip-172-31-16-228:~$ cat mount_unsupervised.sh
s3fs -o iam_role="S3-mount-unsupervised-data" -o url="https://s3-eu-west-1
.amazonaws.com" -o endpoint=eu-west-1 -o dbglevel=info -o curldbg edsa-202
0-unsupervised-predict unsupervised_data
(base) explore-student@ip-172-31-16-228:~$
```

Step 5: Automating the Process

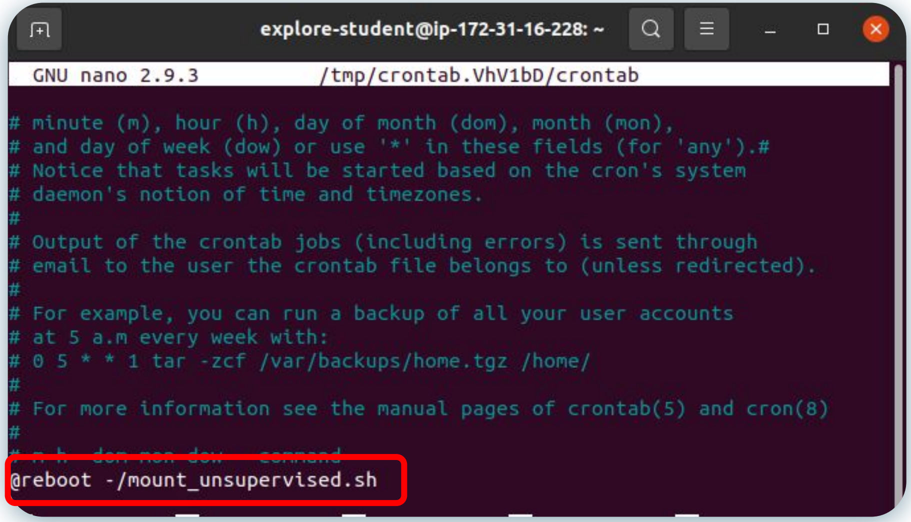
While our S3 bucket is now mounted, as it currently stands **we need to repeat the steps** on the previous slide **each time our instance restarts**. Instead, to avoid this pain, **we'll create a script to run our mount command at startup with cron.**

4. Make the script executable:

   `sudo chmod +x mount_unsupervised.sh`

5. Run `crontab -e`. From the provided options, choose to use a text editor you are comfortable with. Insert the following line at the end of the file: `@reboot ~/mount_unsupervised.sh`

6. Save your changes and exit the file. Reboot your instance to test that the automated mounting is working correctly.



```
explore-student@ip-172-31-16-228: ~

  GNU nano 2.9.3              /tmp/crontab.VhV1bD/crontab

# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot ~/mount_unsupervised.sh
```

Conclusion

What have we learnt?

We have successfully learned how to mount an Amazon S3 bucket to an EC2 instance using the s3fs client application via the command-line interface.

We took it a step further and automated the process so that the S3 bucket gets mounted every time the machine is started.

General Notice

You'll notice that **your mounted directory has *read-only permissions***. This is intentional, as all students are accessing the same S3 bucket, and implies that **you cannot write or save work within this directory** - be careful of this fact!

While the data you have access to now are not confidential, they often will be. This means that you need to **take care not to download data onto a personal machine**, or to share them amongst other individuals.