EXPLORE DATA SCIENCE ACADEMY

Using Data in the Cloud

Mount an S3 bucket to EC2 Instance

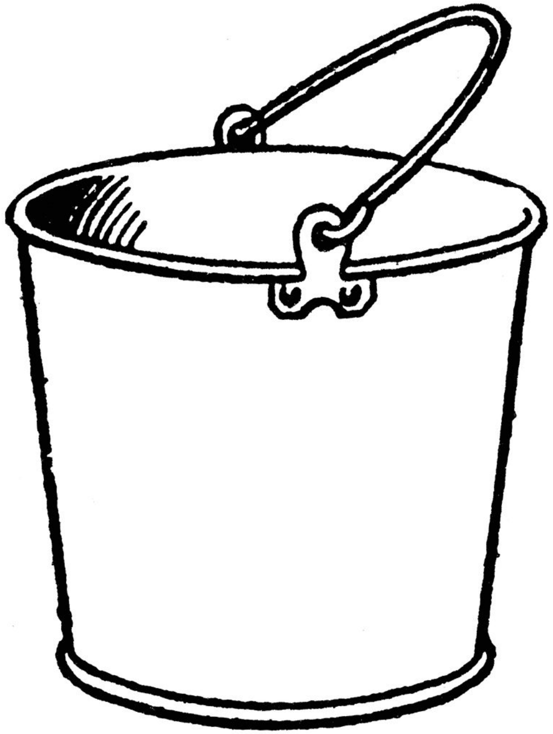
Getting to Grips with our Data

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 become more complex. These include:

- 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**?

In answer to these questions, within this tutorial we'll be learning about the **AWS S3 service**, and the process involved in **<u>mounting an S3 bucket onto our EC2 instance</u>.**

What’s an S3 bucket?



Before we go any further, let’s quickly discuss the AWS S3 service:

- S3 stands for **S**imple **S**torage **S**ervice.
- It's an Object storage service optimised to provide massive amounts of **scalability** (petabytes of data), **performant** read/write access, **reliability** and, high **security**.
- It also is **extremely cheap** to store data on S3!
- An S3 bucket is a single point of storage within the AWS cloud which can be accessed via a number of API’s, web-services, and command-line tools
- More information can be found here.

As part of your Unsupervised Predict project, **we’ve prepared an S3 bucket** containing a number of important files for your analysis.

We’ll now learn how we access this data from an EC2 instance within our Academy AWS account.

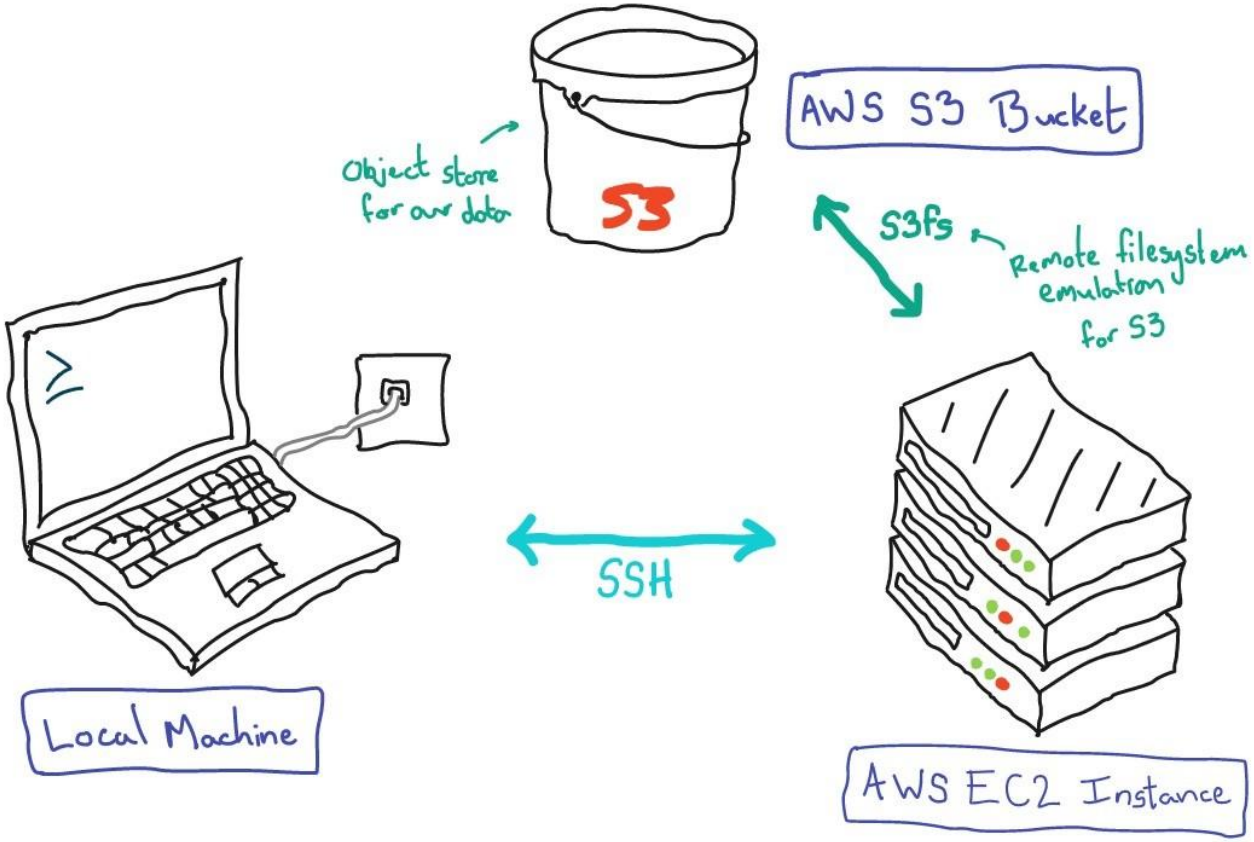
Mounting an S3 Bucket - Process Overview

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 a EDSA owned S3 bucket so that we can read files from it.

To do this, we'll need to do several things:

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



Mounting an S3 Bucket - Step 1: Obtain a Running EC2 Instance

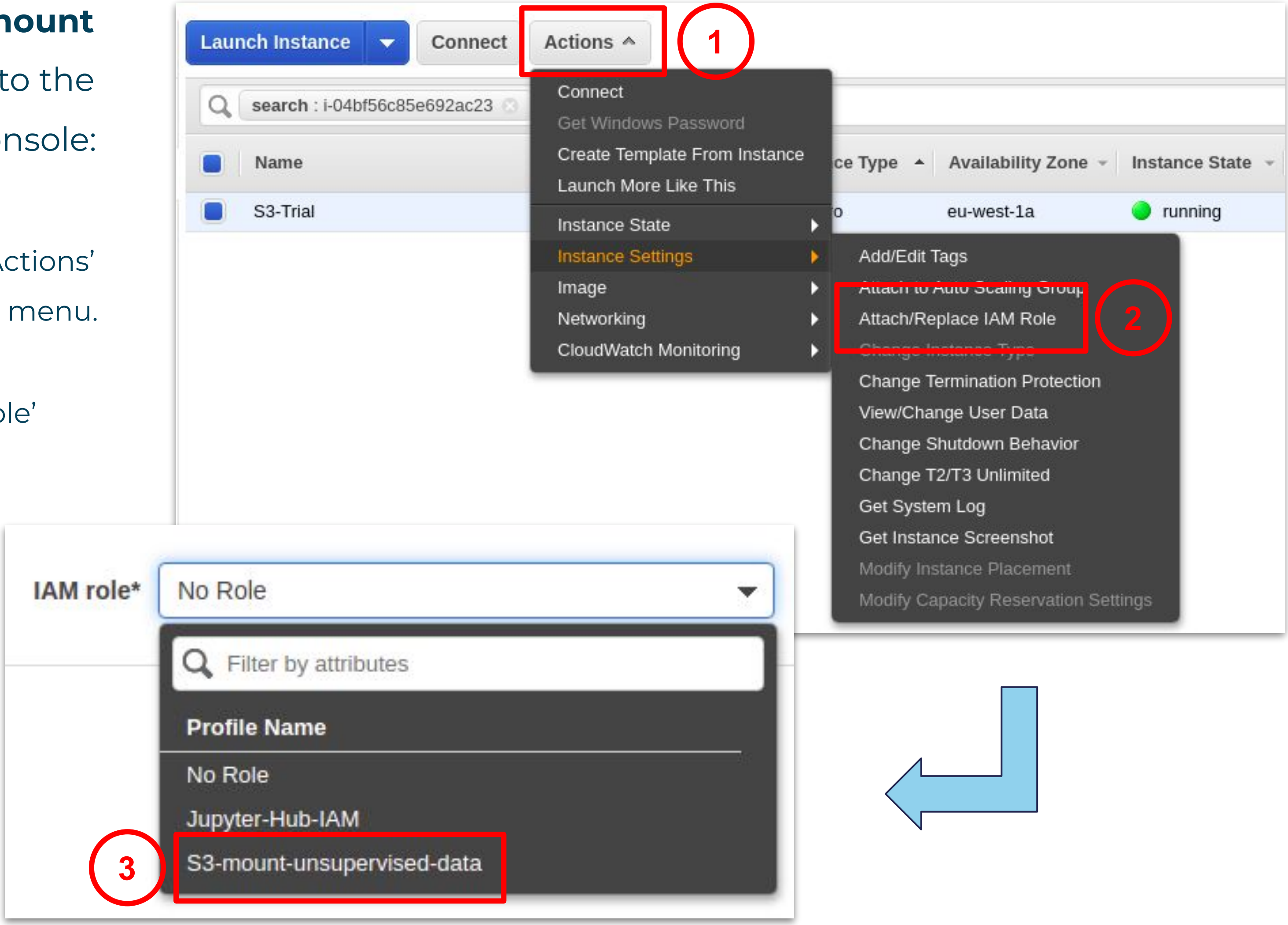
As an initial step, we **need to have access to an EC2 instance** running within the 'eu-west-1' (Ireland) region under the EDSA AWS account.

This **can be an instance which you've used for some time, or a freshly created one**. If you don't have an EC2 instance and have forgotten how to spin one up, have a look at the video tutorial below for a refresher.

Mounting an S3 Bucket - Step 2: Attach IAM Role to Instance

We'll **next grant our EC2 instance permission to mount the S3 bucket.** We do this by attaching an IAM Role to the instance via the AWS console:

1. With your EC2 instance selected, click on the 'Actions' drop-down menu.
2. Navigate to 'Instance Settings' → 'Attach/Replace IAM Role'
3. Under the drop-down menu for IAM role, select 'S3-mount-unsupervised-data'
4. Click 'Apply'. A green confirmation prompt should display if the IAM role was successfully applied.



Mounting an S3 Bucket - Step 3: Install Software Dependencies

We now need to **install the S3FS client application on our instance.**

NOTE: The following commands are lengthy. To avoid errors during input **we recommend pasting** these from this handy GitHub gist.

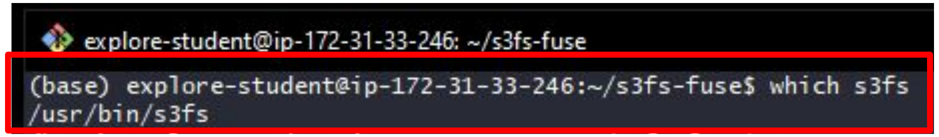
These commands are identical on MAC (via terminal) or Windows (via Git Bash)

1. Connect to your remote instance with the following command. Revisit Part 1 here if you've forgotten:

```
ssh explore-student@<EC2 Instance 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
```



3

```
explore-student@ip-172-31-33-246: ~/s3fs-fuse
(base) explore-student@ip-172-31-33-246:~/s3fs-fuse$ which s3fs
/usr/bin/s3fs
```

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

Mounting an S3 Bucket - Step 4: Mount S3 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. This is outlined below:

1. Change to your instance's home directory and create a folder called `unsupervised_data` as the mount point:

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

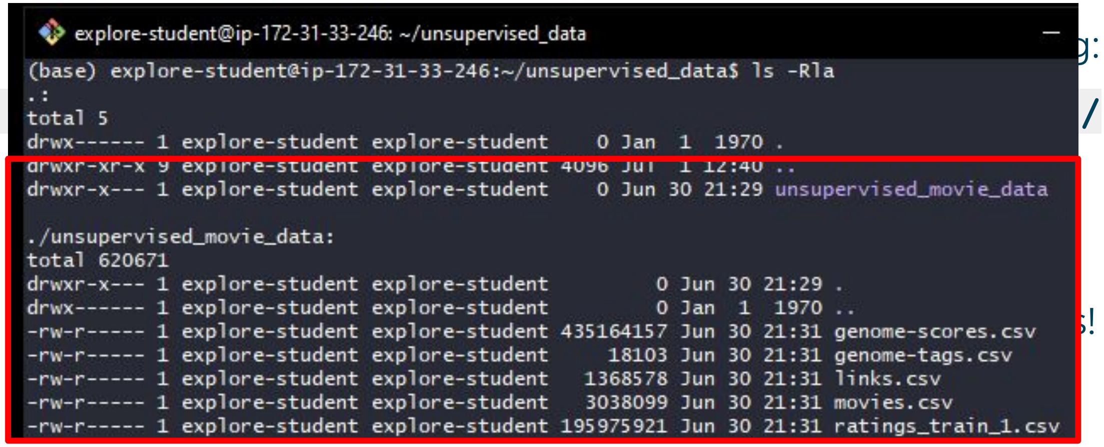
2. Use the following command to perform the mount operation:

   ```
   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 g:

   ```
   ls -Rla /
   ```

You've now officially mounted 🎉 your s!



```
(base) explore-student@ip-172-31-33-246:~/unsupervised_data$ ls -Rla
.:
total 5
drwx------ 1 explore-student explore-student    0 Jan  1  1970 .
drwxr-xr-x 9 explore-student explore-student 4096 Jul  1 12:40 ..
drwxr-x--- 1 explore-student explore-student    0 Jun 30 21:29 unsupervised_movie_data

./unsupervised_movie_data:
total 620671
drwxr-x--- 1 explore-student explore-student         0 Jun 30 21:29 .
drwx------ 1 explore-student explore-student         0 Jan  1  1970 ..
-rw-r----- 1 explore-student explore-student 435164157 Jun 30 21:31 genome-scores.csv
-rw-r----- 1 explore-student explore-student     18103 Jun 30 21:31 genome-tags.csv
-rw-r----- 1 explore-student explore-student   1368578 Jun 30 21:31 links.csv
-rw-r----- 1 explore-student explore-student   3038099 Jun 30 21:31 movies.csv
-rw-r----- 1 explore-student explore-student 195975921 Jun 30 21:31 ratings_train_1.csv
```

Mounted S3 content

Mounting an S3 Bucket - 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`** using the following steps:

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

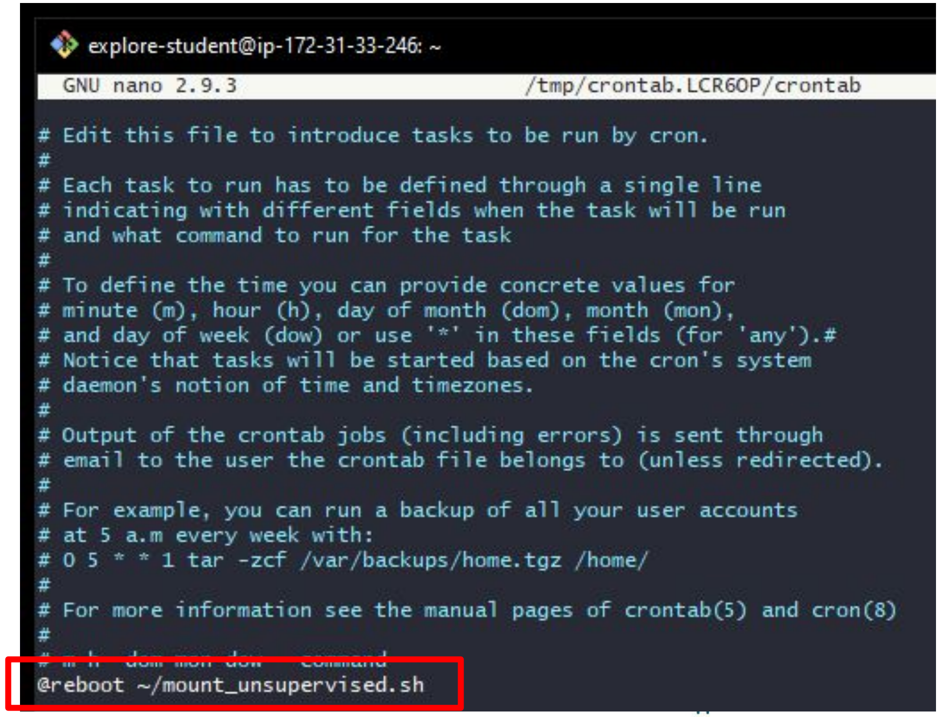
   ```
   echo 's3fs -o iam_role="S3-mount-unsupervised-data" -o url="https://s3-eu-west-1.amazonaws.com" -o endpoint=eu-west-1 -o dbglevel=info -o curldbg edsa-2020-unsupervised-predict unsupervised_data' > mount_unsupervised.sh
   ```

2. Make the script executable:

   ```
   sudo chmod +x mount_unsupervised.sh
   ```

3. 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`

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



```
explore-student@ip-172-31-33-246: ~
 GNU nano 2.9.3                         /tmp/crontab.LCR6OP/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# 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
```

That’s a Wrap!

If you’ve made it to this slide then you’ve probably finished mounting your S3 bucket to your instance - Fantastic!

Before you go diving into your data, however, we have some parting notes for you:

- 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.