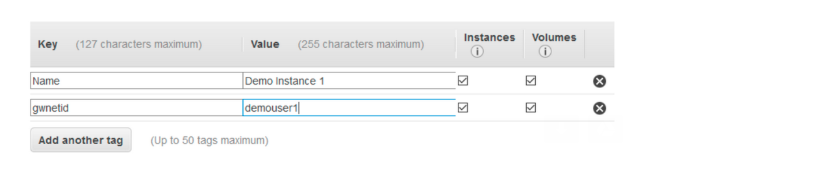
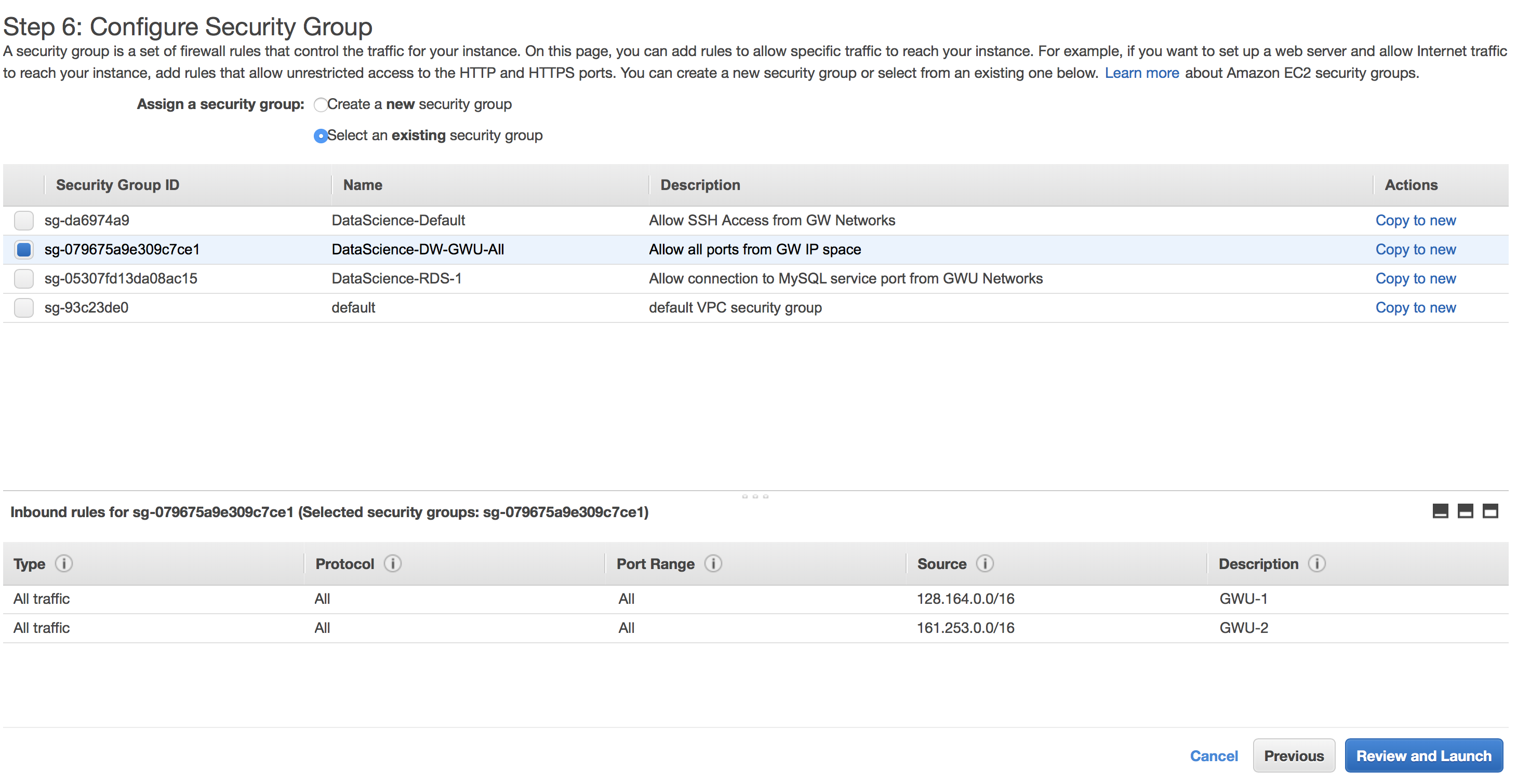
**Hadoop Lab 1**

**Part I – Configuring EC2**

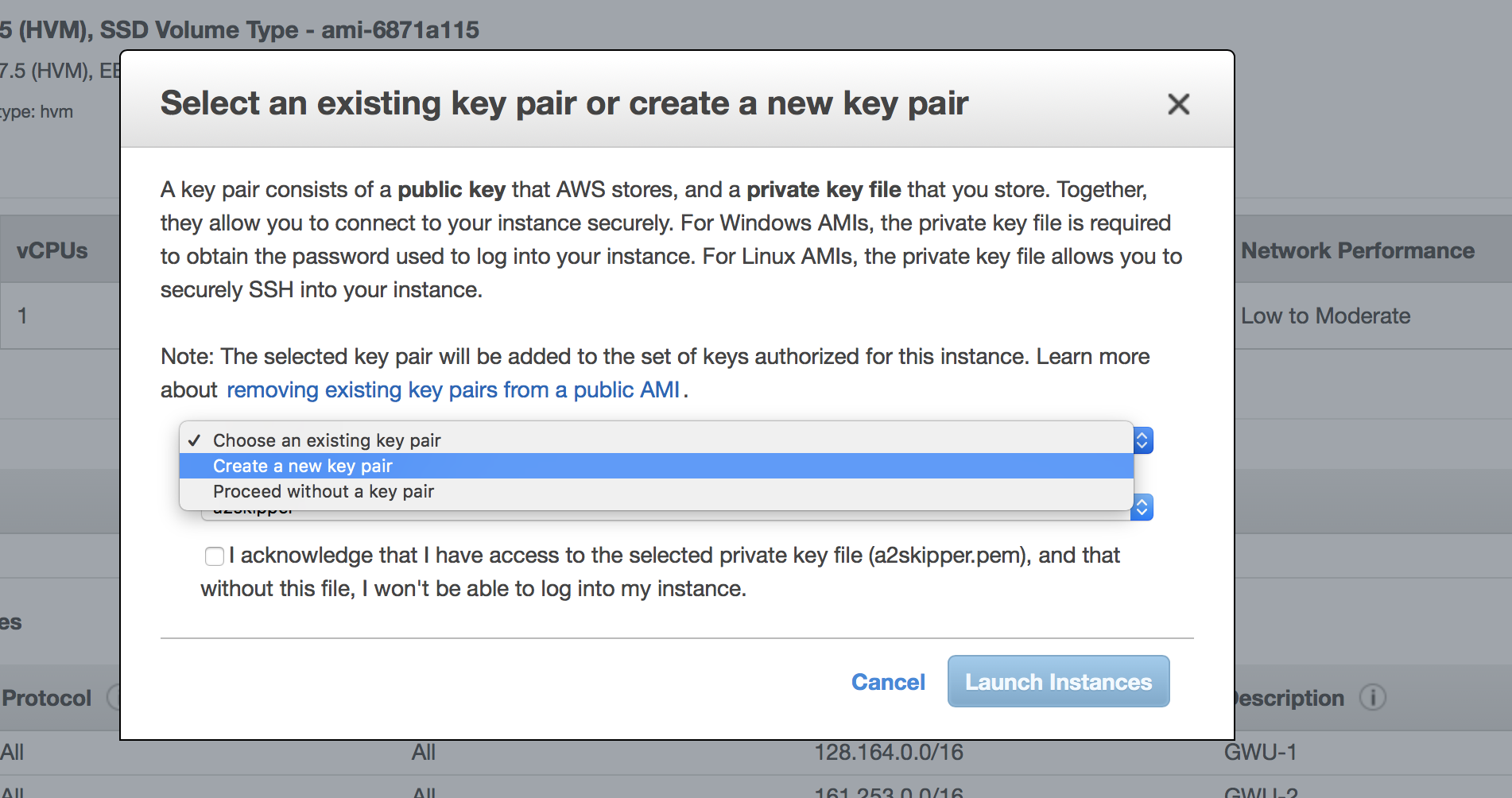
1. Follow AWS End-User Documentation
2. Use **DataScience-DW-GWU-ALL** security group
3. Launch **t2.micro** instance
4. **Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type**
5. Add a tag with YOUR netid:



1. Select existing Security group as follows 🡪 Review and Launch



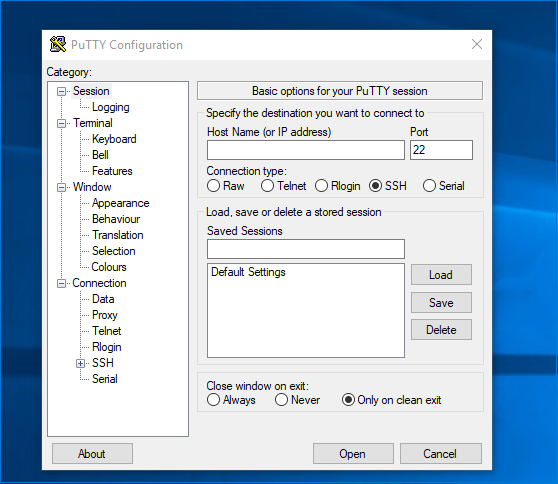
1. Create a new pair of keys:



**STEP 8 – 13 WINDOWS ONLY**

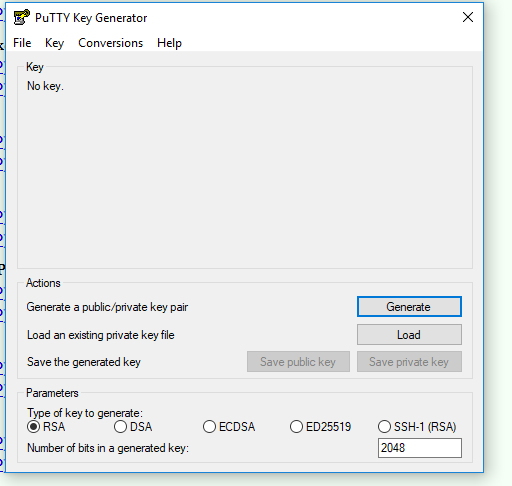
1. **DOWNLOAD Putty and puttygen :**

<https://www.putty.org/>

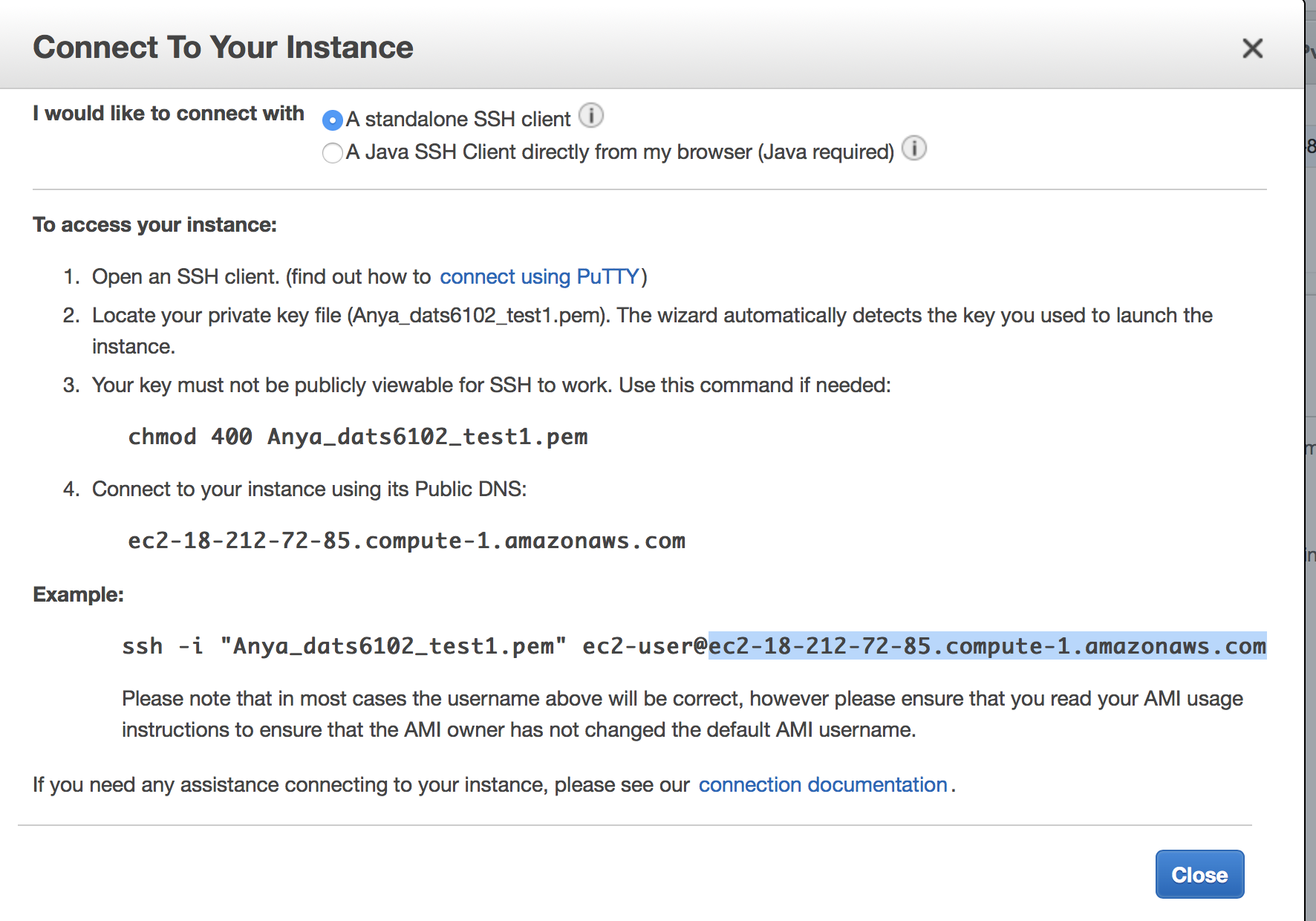


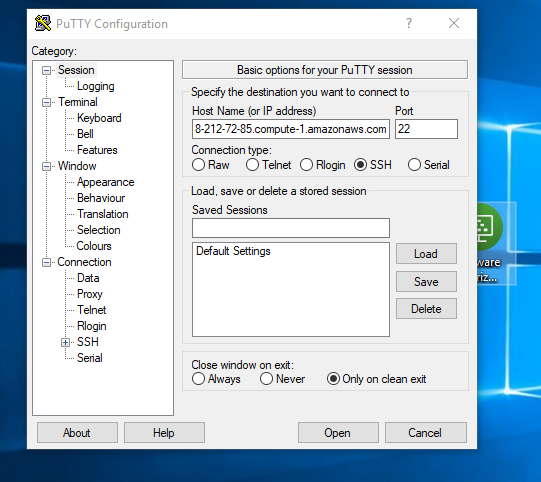
Putty doesn’t like **pem** file (your EC2 key is in this format), so we are going to use **puttygen.exe** to convert it.

1. Launch **puttygen.exe** and click **Load**

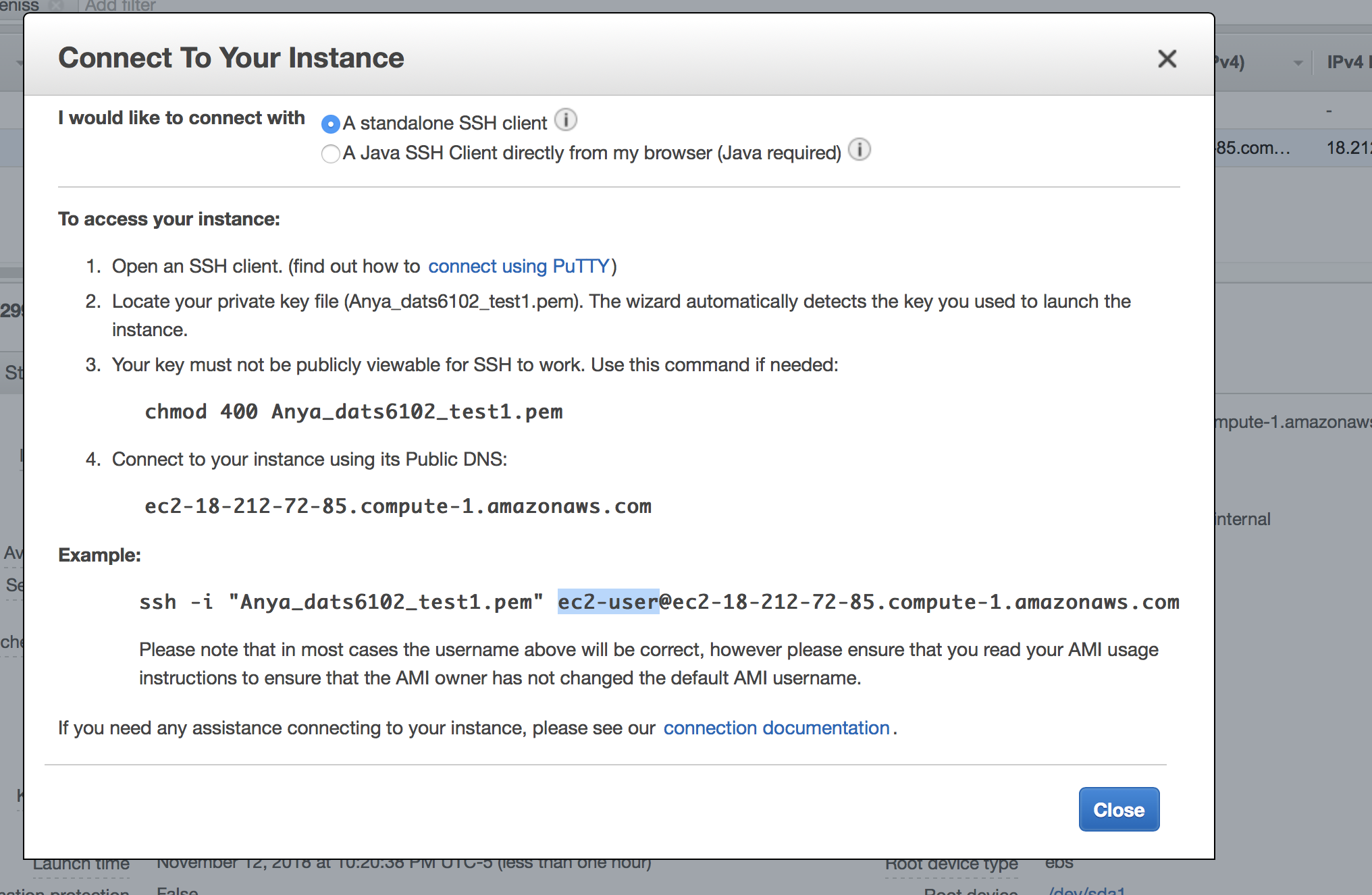
****

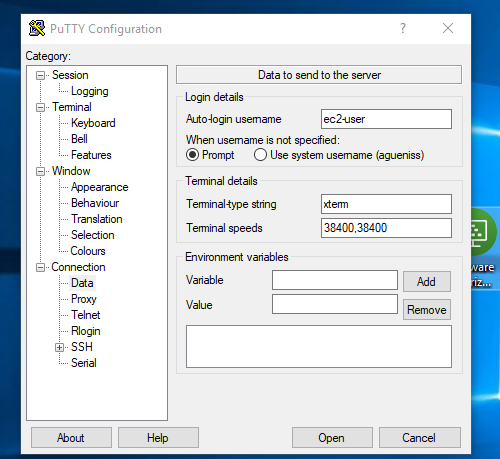
1. Save as private key (make sure its saved on Desktop)
2. Launch putty
3. Copy machine’s host name from your EC2 instance



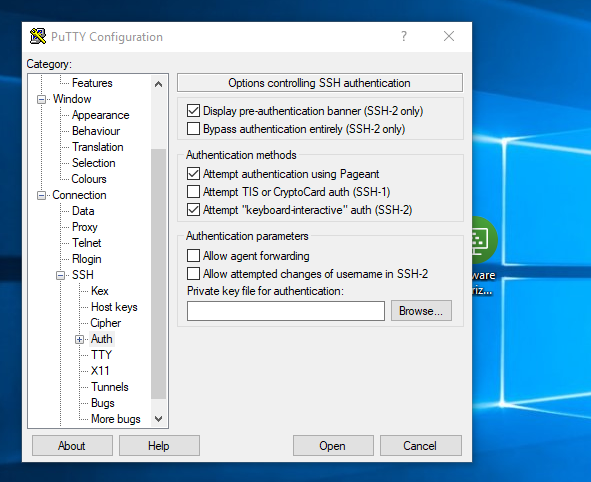


11. Go to Data section (Connection 🡪 Data) AND COPY username from the above EC2 string:

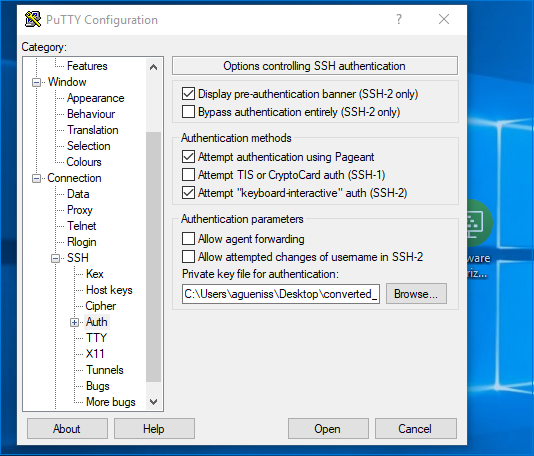




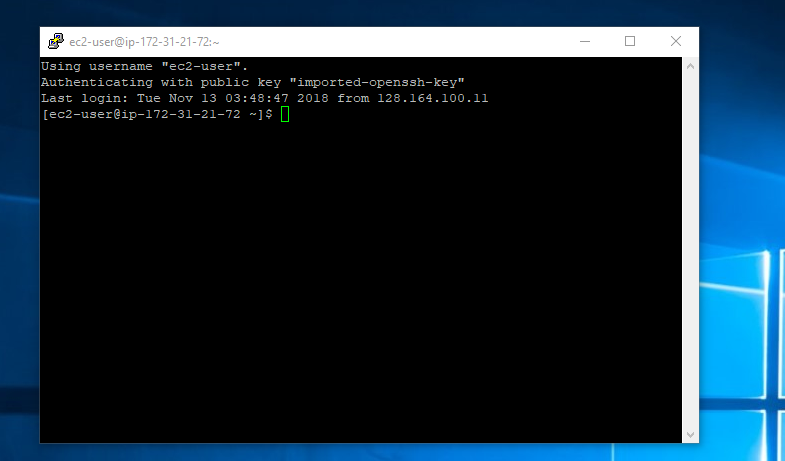
1. Now expand SSH 🡪 Auth



Click Browse 🡪 and input your saved on Desktop key:



1. Session 🡪 save session (so you don’t have to input this stuff next time) and you are set to go:



**Part II – Getting around your instance**

**Linux directory structure**

1.

# Navigate to “root” directory (Linux directory structure begins at the root and expands from #there. **/** is a directory separator)

**cd /**

2.

#Navigate to bin directory (bin is a directory for binaries and other executable programs)

#Programs are written in source code and it is human readable text, then these files are compiled #into machine readable binaries. They are called binaries because machine code is string of 0’s #and 1’s. So applications that you run will be located here

**cd /bin**

3.

# Navigate to System configuration files directory, which control how OS or application behave.

**cd /etc**

4.

#Navigate to user home directory. Linux systems have multiple user accounts. Home directory is #where you separate your data from other account’s data

**cd /home**

**5.**

#Navigate to “optional/third party directory”. This directory is for software that is not bundled #with the standard operating system (e.g. Google Earth is not bundled with standard OS, so it #would be installed in opt)

**cd /opt**

**6.**

#Navigate to temp directory, which is typically cleared on reboot

**cd /tmp**

**7.**

#Navigate to user directory (user related programs, /usr/bin – binaries and executable files for the #user; /usr/lib – libraries)

**cd /usr**

**8.**

#Navigate to variable data directory (variable data, mostly log files)

**cd /var**

**9.**

#Navigate to media directory (used to mount removable media like CD-ROMs)

**cd /media**

**10.**

#Navigate to srv directory – contains data which is served by the system (e.g. /srv/www for web #server files, etc)

**cd /srv**

**11.**

#Navigate to /usr/local directory – applications that are not bundled with the operating system #are installed here. For there they will have their own sub-directory structure (e.g. #/usr/local/**program/**bin /usr/local/**program/**etc /usr/local/**program/**lib /usr/local/**program/log)**

**#** If you install some program from the web, it would be installed here in /usr/local (binary in #/bin). To modify configuration files for this specific program, you would go to #/usr/local/**program**/etc

**cd /usr/local**

Note: Applications that are not shipped with the operating systems are usually loaded to:

/opt

/usr

Most common directories that you will interact with are:

/

/bin

/etc

/home

/opt

/tmp

/usr

/var

**The Shell (command line interface) and Basic Commands**

1.

# Shell is default interface to Linux system (when you ssh to linux server, you are accessing the #shell. It takes commands and executes them). Look at your prompt (# - root user, $ regular user):

**[root@ip-172-31-21-72 local]#**

**[ec2-user@ip-172-31-21-72 home]$**

**Note: root can mean super user account OR / (root of file system)**

2.

# Navigate to your home directory. Tilde will expand to your home directory or (~user)

**cd ~**

3.

# Navigate to your home directory.

**cd ~**

4.

# Let’s practice these basic commands (commands, directories and files ARE CASE SENSITIVE)

**ls lists directory contents, displays detail information about directory (try ls -l)**

**cd changes the current directory**

**pwd displays present working directory**

**cat concatenates and displays files – displays contents of files**

**echo displays arguments on the screen**

**man dispalys online manual (q to quit it)**

**exit exits the shell or your current session**

**clear clears the screen**

5.

# Explore Man pages

**Enter move down one line**

**Space move down one page**

**g move to the top of page**

**G moved to the bottom of page**

**q quit**

6.

# Explore environmental variables (storage locations that have a name and a value; usually uppercase) $PATH controls the command search path and contains list of directories

**echo $PATH**

7.

# Explore location of a command you are executing (depends on your search $PATH)

**which cat**

8.

# Explore directory shortcuts (/ forward slash is a directory separator)

**. This directory**

**.. The parent directory**

**cd - Change to the previous directory**

9.

# Explore directory shortcuts (/ forward slash is a directory separator)

**mkdir create a directory**

**rmdir remove a directory**

**rm -rf recursively remove directory**

10.

#Explore listing directories

**ls -l list directory**

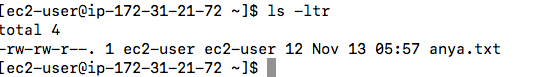
**ls -al show hidden files**

**ls -F show file types ( / directory. @link. \* executable)**

**ls -t list files by time**

**ls -r reverse order**

**ls -latr long listing including all files reverse sorted by time**



**First group of characters permissions**

**1 number of links**

**Owner name ec2-user**

**Group name ec2-user**

**Number of bytes in the file 4**

**Last modification time Nov 12**

**File name anya.txt**

11.

#Explore symbolic links. A link is a pointer to the actual file or directory, we use as if it were a file. A link can be used to create a shortcut (use for long files or directory names; use to indicate the current version of software). Observe symbolic link 🡪

**sudo su – root**

**cd /etc**

**ls - lF**

system-release -> redhat-release

12.

#Create a directory

mkdir linuxtutorialwork

13.

#Create a directory with sub-directory

#The first one is **-p** which tells mkdir to make parent directories as needed (demonstration of #what that actually means below). The second one is **-v** which makes mkdir tell us what it is #doing (as you saw in the example above, it normally does not).

mkdir -p linuxtutorialwork/foo/bar

14.

#Remove this directory

rmdir linuxtutorialwork/foo/bar

15.

#Create a blank file and copy it to a directory

touch mytest

ls -ltr

cp mytest /home/ec2-user

16.

#Create a blank file and move it to a directory

touch mytest1

ls -ltr

mv mytest /home/ec2-user

17.

#Rename a directory

mv foo to foo1

18.

#Rename a directory

rm mytest1

19.

#Open vi editor (When you run this command it opens up the file. If the file does not exist then it #will create it for you then open it up. (no need to touch files before editing them) Once you #enter vi #it will look something like this (though depending on what system you are on it may #look slightly #different).

vi newtest.txt

You always start off in edit mode so the first thing we are going to do is switch to insert mode by pressing **i**. You can tell when you are in insert mode as the bottom left corner will tell you. Type some line use # for comments

#this is a comment

And this is my first line, not a comment

* **ZZ** (Note: capitals) - Save and exit
* **:q!** - discard all changes, since the last save, and exit
* **:w** - save file but don't exit
* **:wq** - again, save and exit

Below are some of the many commands you may enter to move around the file. Have a play with them and see how they work.

* **Arrow keys** - move the cursor around
* **j, k, h, l** - move the cursor down, up, left and right (similar to the arrow keys)
* **^ (caret)** - move cursor to beginning of current line
* **$** - move cursor to end of the current line
* **nG** - move to the **n**th line (eg 5G moves to 5th line)
* **G** - move to the last line
* **w** - move to the beginning of the next word
* **nw** - move forward n word (eg 2w moves two words forwards)
* **b** - move to the beginning of the previous word
* **nb** - move back n word
* **{** - move backward one paragraph
* **}** - move forward one paragraph

Below are some of the many ways in which we may delete content within vi. Have a play with them now. (also check out the section below on undoing so that you can undo your deletes.)

* **x** - delete a single character
* **nx** - delete n characters (eg 5x deletes five characters)
* **dd** - delete the current line
* **dn** - d followed by a movement command. Delete to where the movement command would have taken you. (eg d5w means delete 5 words)

Undoing changes in vi is fairly easy. It is the character **u**.

* **u** - Undo the last action (you may keep pressing u to keep undoing)
* **U (Note: capital)** - Undo all changes to the current line

**Part III –Amazon EMR Demo**

**Section A – Connecting to EMR**

1. Connect to my EMR – download Hadoop key from blackboard (there are separate keys for Mac and Windows) and rename it to Anya\_pair.pem for Mac or Anya\_pair.ppk for Windows and save it on your desktop.

**\*\*\*\*\*\*MAC Users (use Hadoop key from blackboard)\*\*\*\***

Copy the key to your desktop and

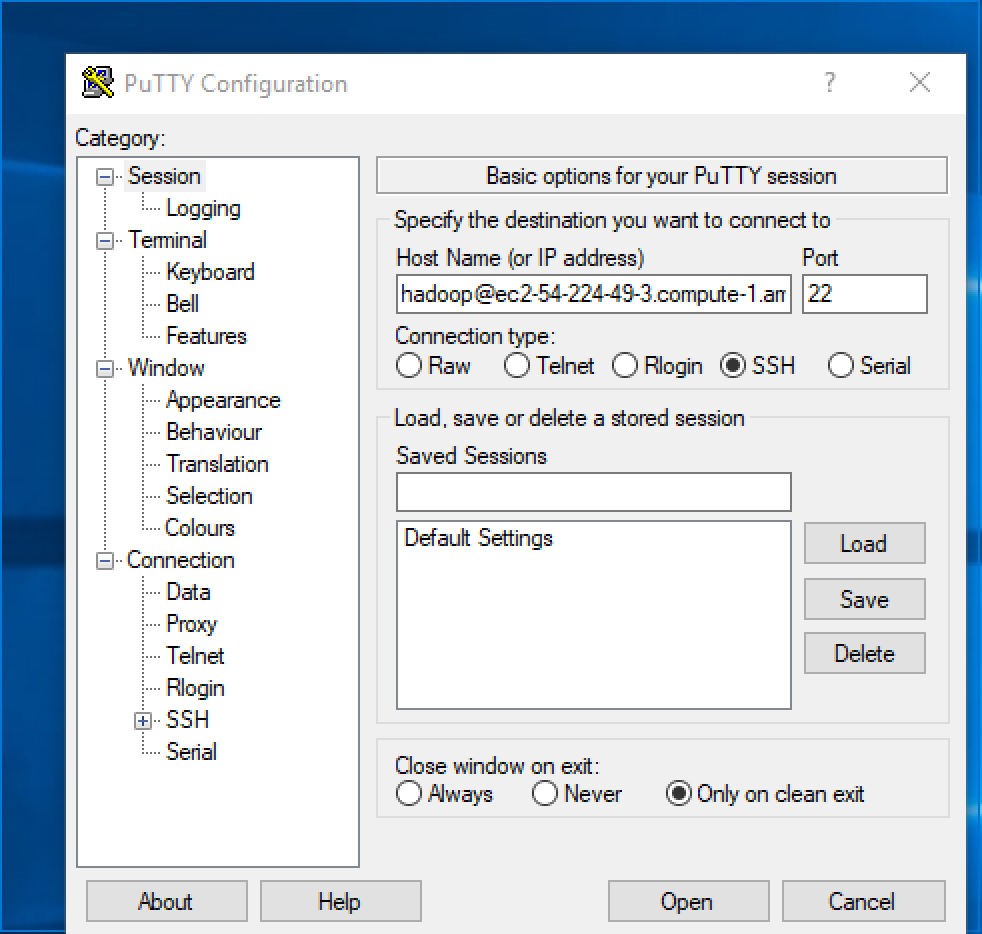
**chmod 400** **Anya\_pair.pem**

ssh -i ~/Anya\_pair.pem hadoop@ec2-3-231-158-112.compute-1.amazonaws.com

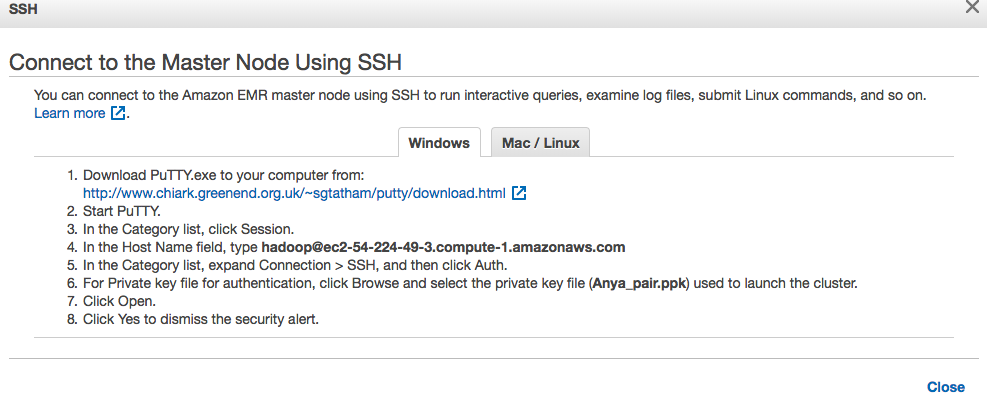
**\*\*\*\*\*\*WINDOWS USERS (use Hadoop key from blackboard)\*\*\*\***

1. **Open putty and copy paste the following into Host Name:**

hadoop@ec2-3-231-158-112.compute-1.amazonaws.com

****

**Follow these instructions starting Step 5:**

****

3.

Change directory to emr directory and list it’s contents with detail:

**cd /emr**

**pwd**

**ls -ltr**

4.

Explore the file system:

**df -h**

5.

**Explore YARN:** <http://ec2-3-231-158-112.compute-1.amazonaws.com:8088/>

**6.**

**Explore HDFS Name Node**:

<http://ec2-3-231-158-112.compute-1.amazonaws.com:50070/>

**Section B – Examining the Raw Data**

In this section, you will access the raw **Ngrams** data. You will be accessing data about 1-grams, which provides a count of the *single words* found in all the books. This data has 261 million entries, occupying 2.6 GB of disk storage. The Ngram data is accessible from Amazon S3 and can be directly accessed from Amazon EMR by creating an *external table.* This definition tells Amazon EMR how the data is formatted and where it is located.

**1.**

Access hive (type hive at the command prompt):

**hive**

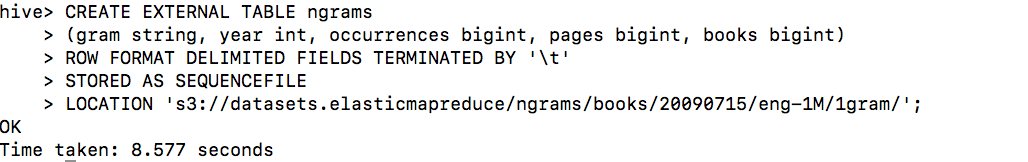
**2.**

Copy and paste this statement to create the data structure and identify the input data location. Append the name of the table **TO YOUR NETID, so it looks like ngrams\_YOURNETID**:

CREATE EXTERNAL TABLE ngrams\_YOURNETID

(gram string, year int, occurrences bigint, pages bigint, books bigint) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS SEQUENCEFILE LOCATION 's3://datasets.elasticmapreduce/ngrams/books/20090715/eng-1M/1gram/';

The output should look similar to this:



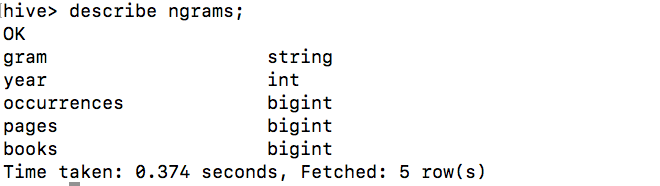
**The ngrams table has been created. Good first step is to view structure of this table.**

**3.**

Describe ngrams table:

**DESCRIBE ngrams\_YOURNETID;**

**Output should look like that**

****

* **Gram** is the word (or set of words) I the n-gram
* **Year** is the year that the n-gram appeared in publication
* **Occurrences** is a count of the number of times the n-gram appeared in the given year
* **Pages** is the number of pages on which the n-gram appeard in the given year
* **Books** is the number of books in which the n-gram appeared in the given year

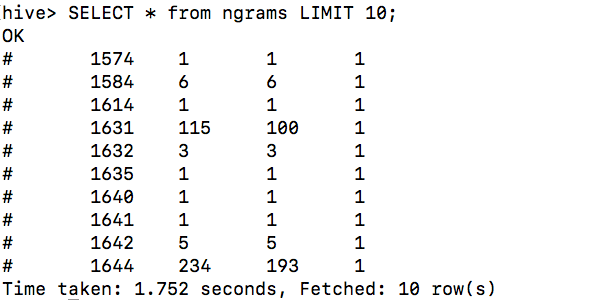
**4.**

Data scientist normally wants to see a sample of the data. This can be done by using SQL SELECT statement

Select first 10 records from ngrams table:

**SELECT \* from ngrams\_YOURNETID LIMIT 10;**

**Output should look like that:**

****

**It appears that the data in its current form is very raw. The first column (gram) contains punctuation marks and other special characters (eg #) and has both uppercase and lowercase characters. This makes it very difficult to analyze.**

**Data scientists frequently face similar data quality problems. Before they can analyze the data, they need to normalize it to a clean format.**

**5. Normalizing data**

To cleanse the data, you will:

* Exclude words under three characters long
* Exclude words that contain non-alphabetical characters (but hyphens and apostrophes are acceptable)
* Convert all characters to lowercase

First you will create a new table to store the results of the normalized data. Copy and paste this statement to create a table to store the results of the normalization (with just the columns of interest). **DON’T FORGET TO APPEND YOUR TABLE WITH YOUR NETID**

CREATE TABLE normalized\_YOURNETID (gram string, year int, occurrences bigint);

Next, you will SELECT the data from the ngrams\_YOURNETID table and then INSERT it into the new normalized table. The query will:

* Only use data between 1990 and 2005
* Convert the n-grams to lowercase
* Only include words of three or more characters
* Only include words that contain alphabethical characters, apostrophes and hyphens.

Copy and paste this statement to select data from the ngrams\_YOURNETID table and insert it itno the normalized table:

INSERT OVERWRITE TABLE normalized\_YOURNETID

SELECT lower(gram), year, occurrences

FROM ngrams\_YOURNETID

WHERE year BETWEEN 1990 AND 2005 AND gram REGEXP "^[A-Za-z+\'-]{3,}$";

**This query takes about 2 minutes to complete. While it is running, Hive provides details of how many processes are executing on the Amazon EMR cluster. While the statement is running, continue to next section. There is no need to wait.**

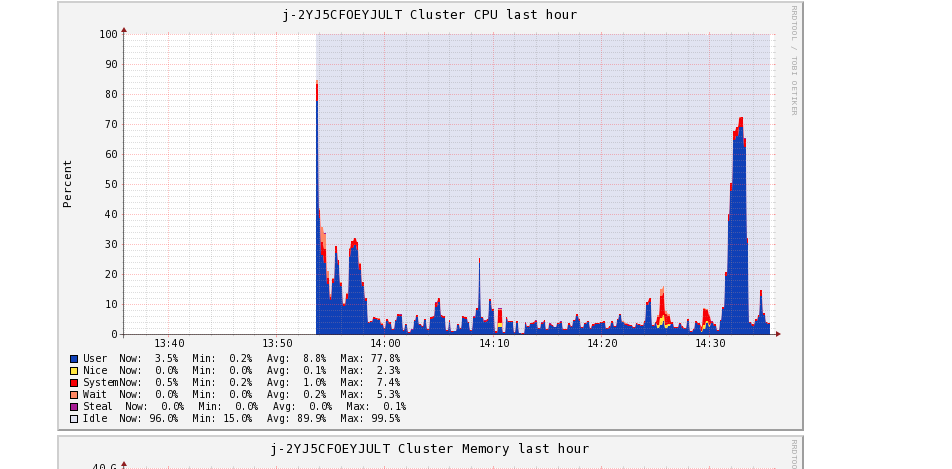
**6. Monitor job execution and cluster usage**

While the query is running, you can use Gangila to monitor your Amazon EMR cluster. Gangila is a scalable, distributed monitoring tool for high-performance computing systems, clusters and networks. The software is used to view metrics such as CPU load averages and network utilization for many nodes.

In your browser enter the following URL:

<http://ec2-54-224-49-3.compute-1.amazonaws.com/ganglia/>

Examine the statistics for our cluster. Click on links and charts to drill-down into more data.



Metrics from Gangilla can help us greatly improve our cluster performance and reduce costs.

**7. Examine normalized data**

Your Hive job should now be complete. A clean version of the data will now be available in the normalized table. You will now examine the data.

Return to SSH session. Copy/paste this statement to view first 20 rows of data

SELECT \* FROM normalized\_YOURNETID LIMIT 20;

You can now see that words are displayed, together with the year that they appeared in books and the number of occurrences of the word in that year. A data scientist would next be interested in viewing the most popular words in all books.

Copy and paste this statement to view the 50 most-used words in all books across all years:

-- Display the 50 most-used words

SELECT gram, sum(occurrences) as total\_occurrences

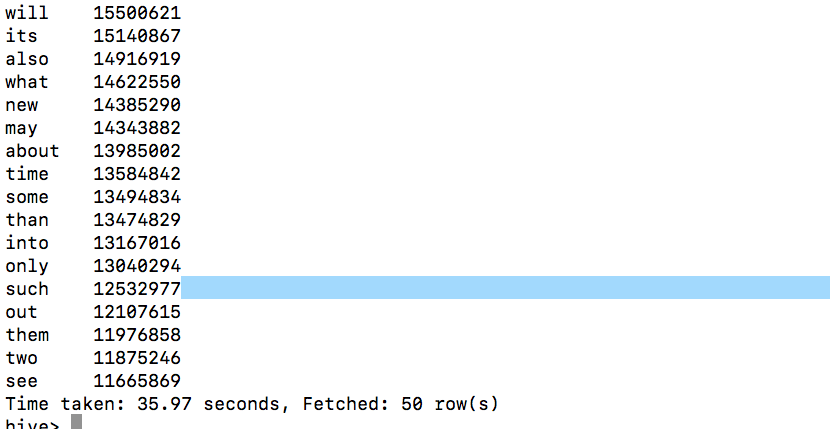
FROM normalized\_YOURNETID

GROUP BY gram

ORDER BY total\_occurrences

DESC LIMIT 50;

**Output:**



These short words would be expected to appear often, but longer words are likely to be more interesting.

Copy/paste this statement to view the 50 most-used words longer than 10 characters:

-- Display the 50 most-used words longer than 10 characters

SELECT gram, sum(occurrences) as total\_occurrences

FROM normalized\_YOURNETID

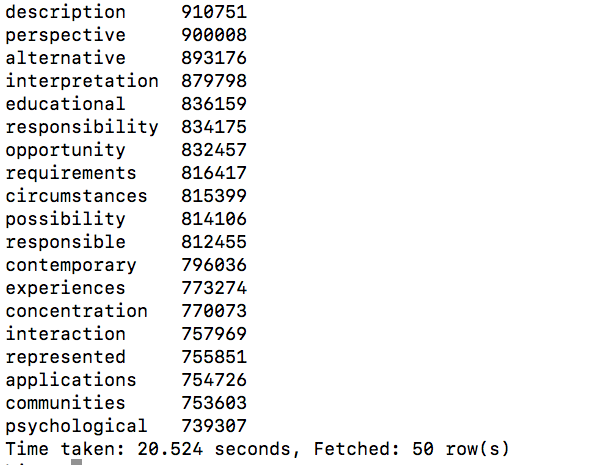
WHERE length(gram) > 10

GROUP BY gram

ORDER BY total\_occurrences

DESC LIMIT 50;

**Output:**



**Each of these queries is analyzing over 20 million records!**

**8. Calculate word usage ratios**

A data scientist might be interested in how word usage varies between years. The easiest way to examine this is to create a new table that contains a ration of the relative occurance of each word, compared to the total number of words stored for the year.

For example, the words *cloud* appears 10,000 times in one year and 13,000 times the next year, it is unclear whether this is because the word became more popular or whether more books were included in the database. By calculating a *ratio* it is possible to compare the relative frequency of a word compared to all other words in a given year.

Copy/paste this Hive statement and create **ratios\_YOURNETID** table:

CREATE TABLE ratios\_YOURNETID (gram string, year int, occurrences bigint, ratio double);

Copy/paste this statement to populate the ratios table with data calculated from the **normalized\_YOURNETID** table:

INSERT OVERWRITE TABLE ratios

SELECT

a.gram, a.year,

sum(a.occurrences) AS occurrences,

sum(a.occurrences) / b.total AS ratio

FROM normalized\_YOURNETID a

JOIN (SELECT year, sum(occurrences) AS total

FROM normalized\_YOURNETID GROUP BY year) b

ON (a.year = b.year) GROUP BY a.gram, a.year, b.total;

This statement is using an INNER JOIN to calculate the ratio of word occurences relative to all words used in the same year.

This statement takes about 3 minutes to execute. You may monitor this via Ganglia while it is executing.

You are now able to query, for each year and word, the change in the ratio from the previous year. This is done by joining the ratios table to itself to compare values between years.

Copy/paste this statement to calculate the difference of ratios year over year. The statement filters out all words with less than 1000 occurrences to avoid having extreme increases for very rare words.

-- Words that most increased in popularity each year

SELECT year, gram, occurrences, CONCAT(CAST(increase AS INT), 'x increase') as increase

FROM ( SELECT y2.gram, y2.year, y2.occurrences, y2.ratio / y1.ratio as increase,

rank() OVER (PARTITION BY y2.year ORDER BY y2.ratio / y1.ratio DESC) AS rank FROM ratios y2

JOIN ratios y1 ON y1.gram = y2.gram

and y2.year = y1.year + 1

WHERE y2.year BETWEEN 1991 and 2005 AND y1.occurrences > 1000

AND y2.occurrences > 1000 ) grams

WHERE rank = 1

ORDER BY year;

Copy/paste this statement:

-- Occurrences of 'internet' in books by year?

SELECT year, occurrences F

ROM ratios

WHERE gram = 'internet'

ORDER BY year;

The results show the explosion in use of the internet around 1994/1995 when the World Wide Web was invented (internet itself has been around a lot longer than that)

Finally, for a bit of fun, the following query finds the most popular words of each length for the year 2000.

Copy/paste this statement:

-- Most popular words of each length

SELECT DISTINCT length, gram

FROM ( SELECT length(gram) AS length,

gram, rank() OVER (partition by length(gram)

order by occurrences desc) AS rank FROM ratios ) x

WHERE rank = 1

ORDER BY length;

These queries provide an idea of the capabilities of Amazon EMR and Hive to analyze massive quantities of data.