Web Exercise 04: Working with Amazon AWS Educate Starter

Due Date: **October 18** (Thursday), 2018. 2:00pm on Blackboard. (Two weeks)

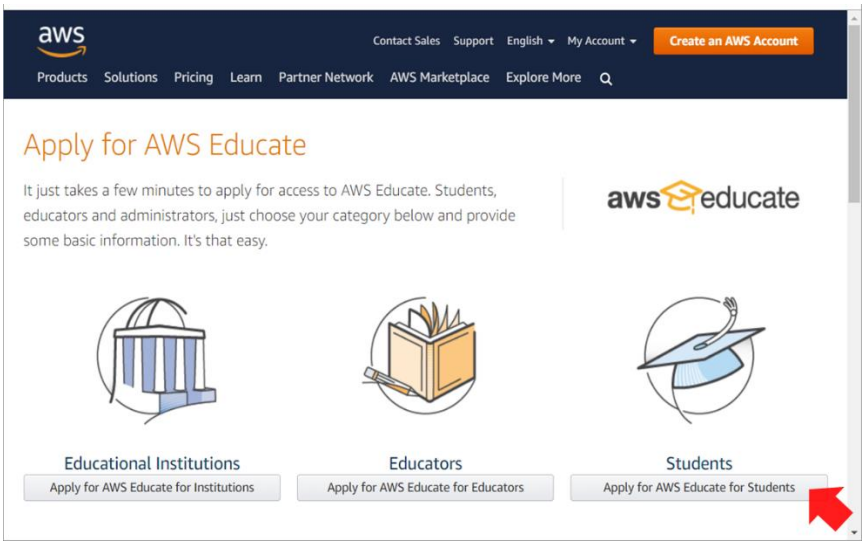
Grade: **20 points.**

1. Create Your AWS Free Account and an EC2 Instance.

Go to the following URL to establish your Amazon AWS Educate Starter account:

https://aws.amazon.com/education/awseducate/apply/

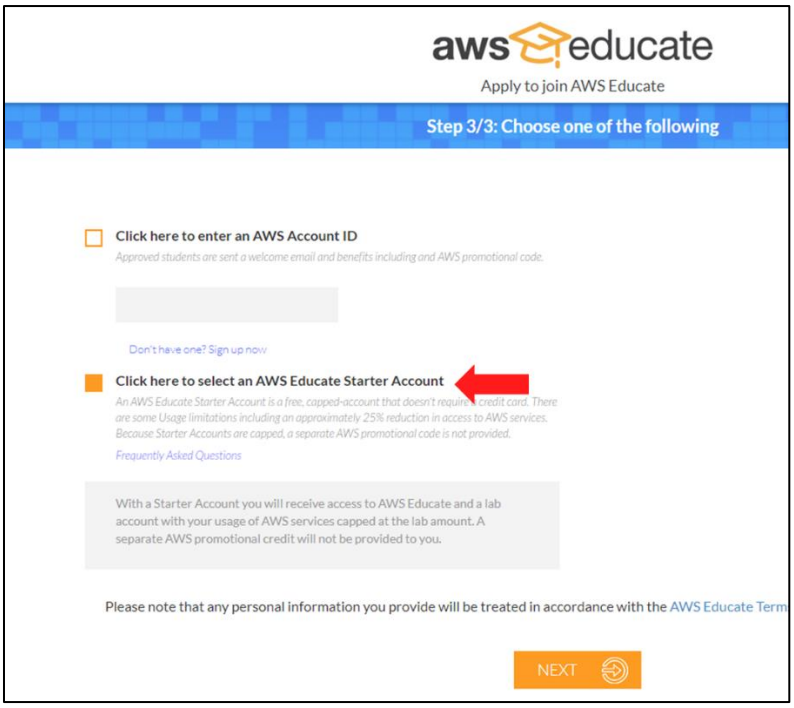
We will be creating an Amazon Web Services (**AWS**) account associated with San Diego State University, so click the "**Students: [Apply for AWS Educate for Students]**" button (the right one):



Then, complete the form.

Fill out your application. Type "**San Diego State University**" in the Institution Name. Make sure to use your SDSU mail addresses or SDSUiD mail address (such as @rohan.sdsu.edu, or **@sdsu.edu**).

The next Step is to select "**Click here to select an AWS Educate Starter Account**", then click on **NEXT**.



Your application has submitted. Check your email and **verify your email** to complete the application process. Your application will be reviewed by the Amazon staff first. Once the review is complete, you will receive a confirmation email from Amazon. Check your email and follow the instruction. Once you have received an email confirming that your account has been approved, set up your password and login first (the link inside the approval email).

Once you set up the password and login successfully. Click on the **"AWS Account"** on the top menu and

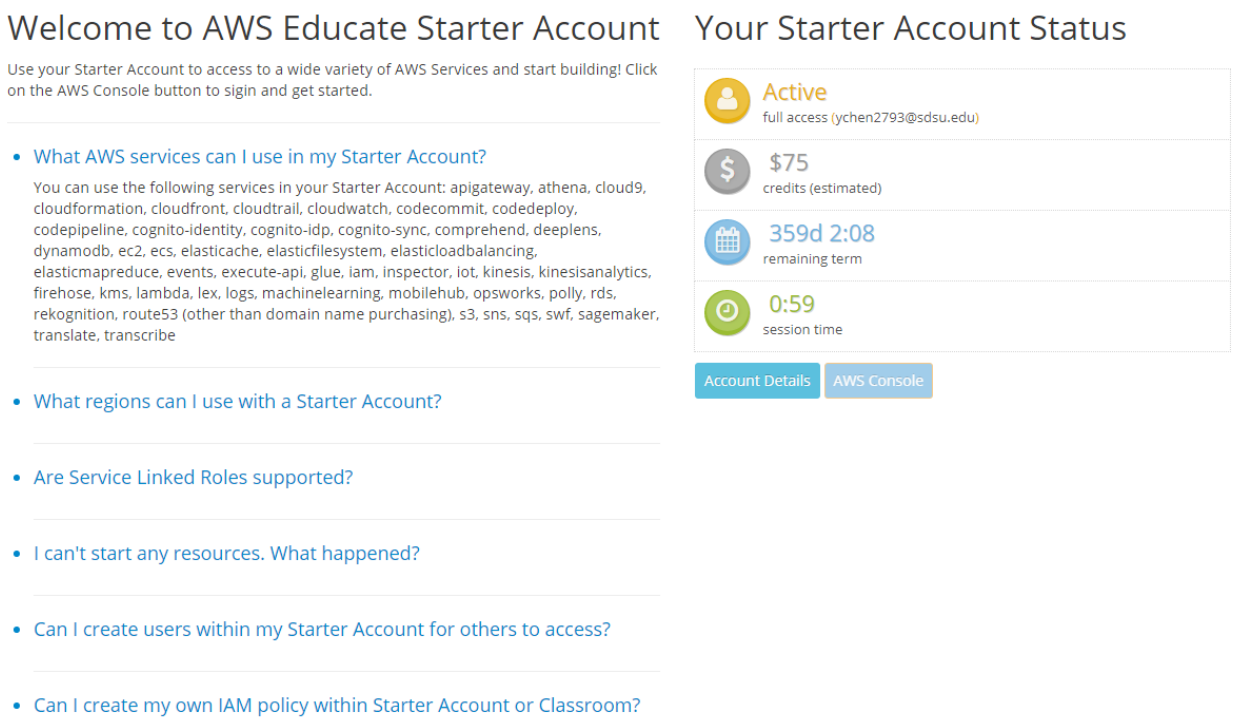
Click on **"I Agree"** on the 'Accept the Term and Condition' page.

Once you set up your AWS Educate Student Account, please use the following link to LOGIN:

https://www.awseducate.com/signin/SiteLogin

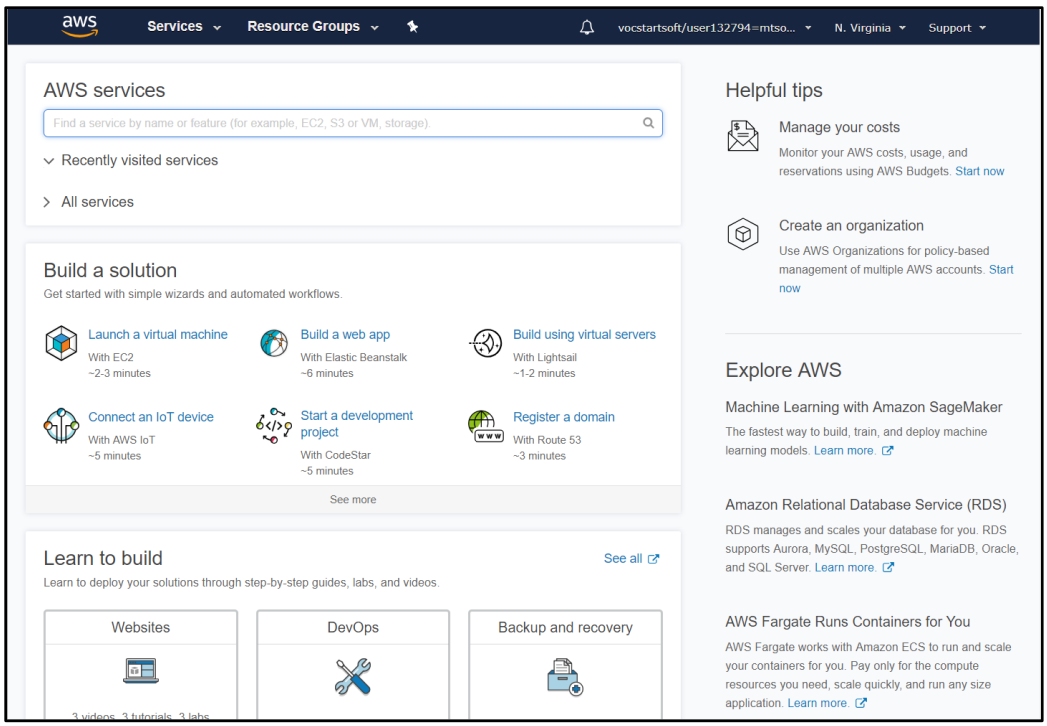
Take a look at the Introduction video.

Click on the "**AWS Account**" on the Top Menu, Then click on "**Go to your AWS Educate Starter Account**". You will start to use the Vocareum (Third Party Servicer) AWS Educate Starter Account.



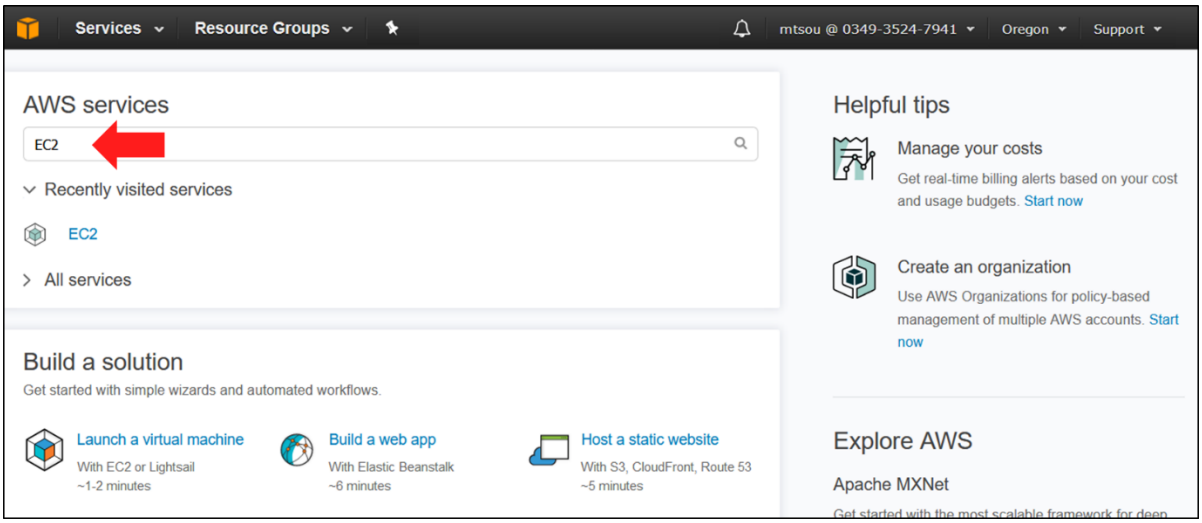
This Dashboard allows you to view your Amazon accounts and view tutorials related to Amazon's services. Since you signed up for an account that is associated with San Diego State University, you should see Your Starter account with **$75** credits and 365d remaining term.

The next step is to **start** the AWS Console. Click on the "AWS Console" button. You may need to turn off the Pop-up STOP setting in your Web browser. After you start the AWS Console, your browser will look like the following:

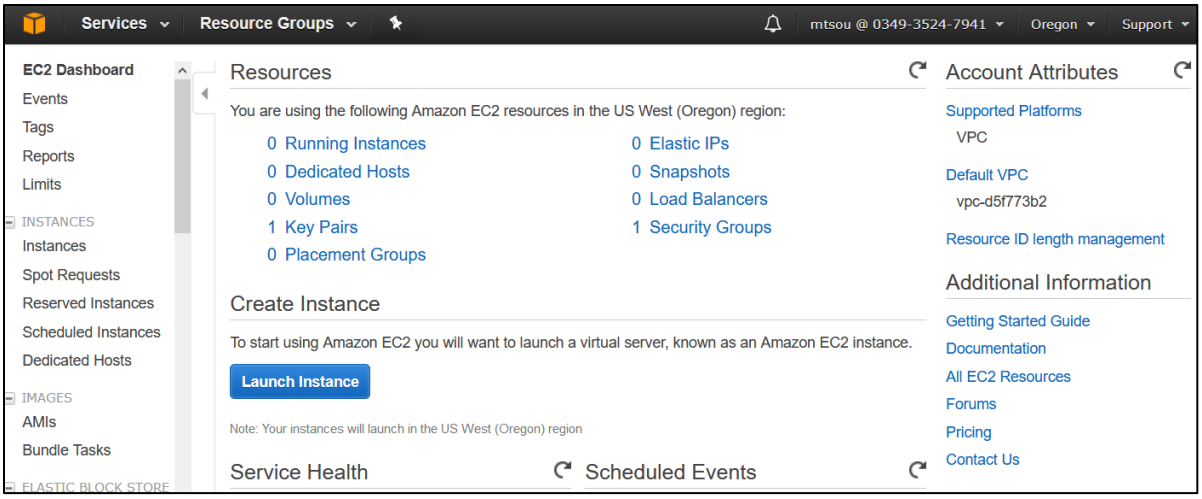


The Amazon Web Services (AWS) dashboard will let you configure and administer your Amazon services, such as EC2 (Elastic Compute Cloud). For this lab, we are only concerned with Amazon's EC2 service.

Type "**EC2**" in the AWS service text box. Then select **EC2**.

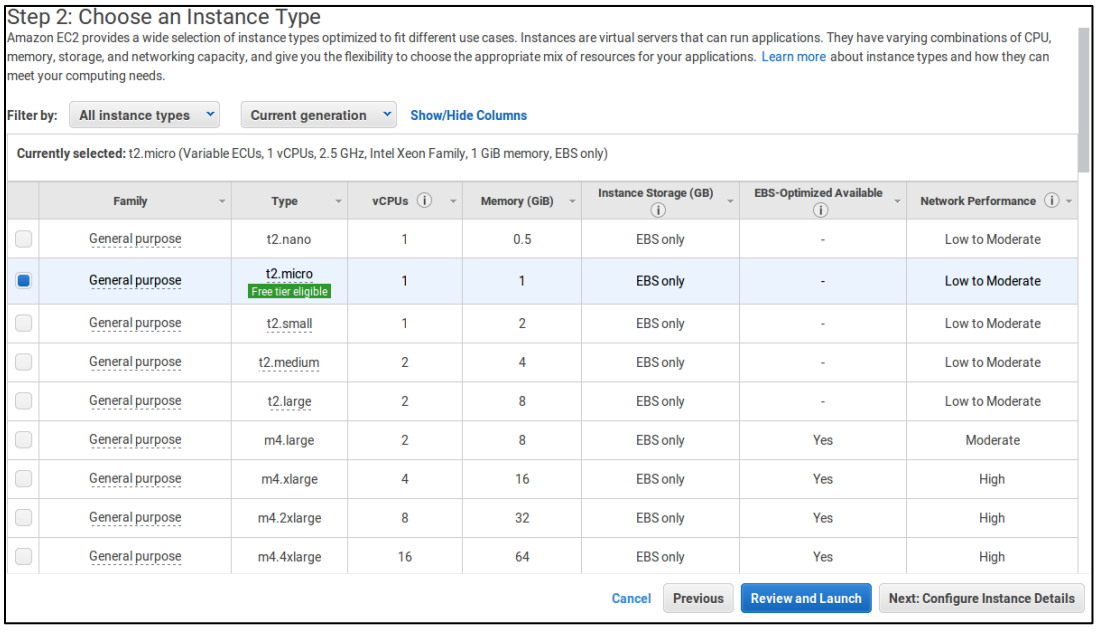


The EC2 dashboard shows us any instances of EC2 that we have created. Since we are opening a new account, there should be none present. We want to create a new instance by clicking the "**Launch Instance**" button.



On the next page, we can select the operating system for our instance. For this lab, we want to create an instance of "**Microsoft Windows Server 2012 R2 Base**", so click the "Select" button for this option.

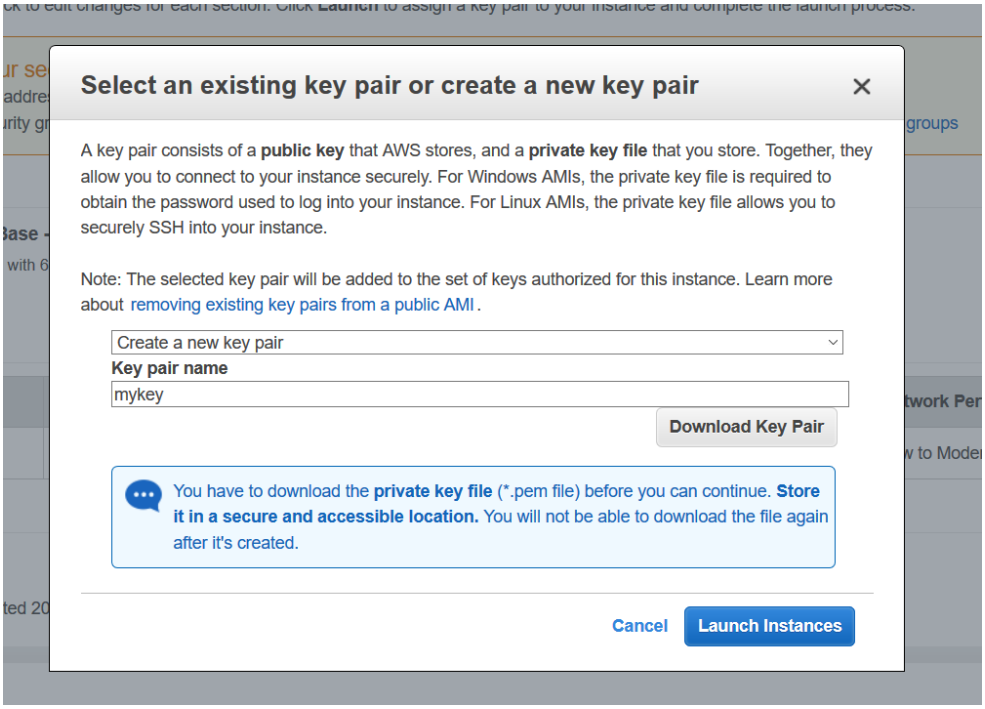
On "Step 2" we can choose the size of our instance. This allows us to configure the computing power and other system resources for our instance. Since we are using a free account, we must select the "**t2 micro" option (free tier eligible)**. Then press "**Review and Launch**" button.



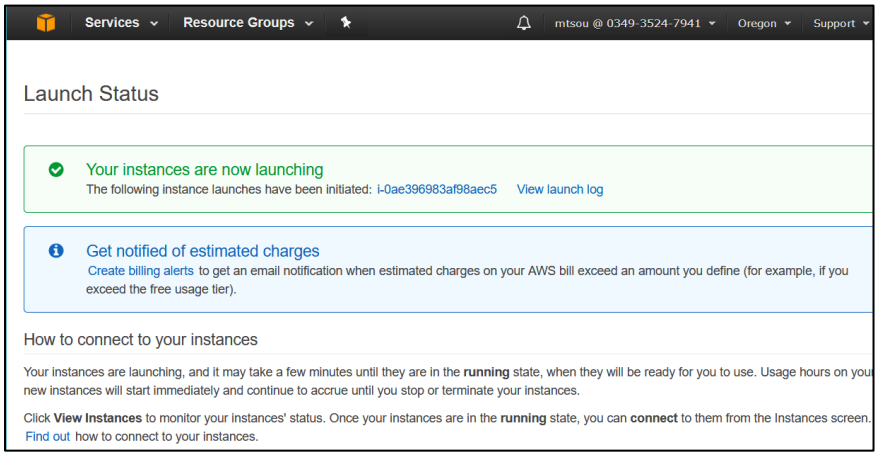
After review the Instance Launch, click on "**Launch**".

Now that we have created an instance, we need to configure it in order to make a connection through remote desktop.

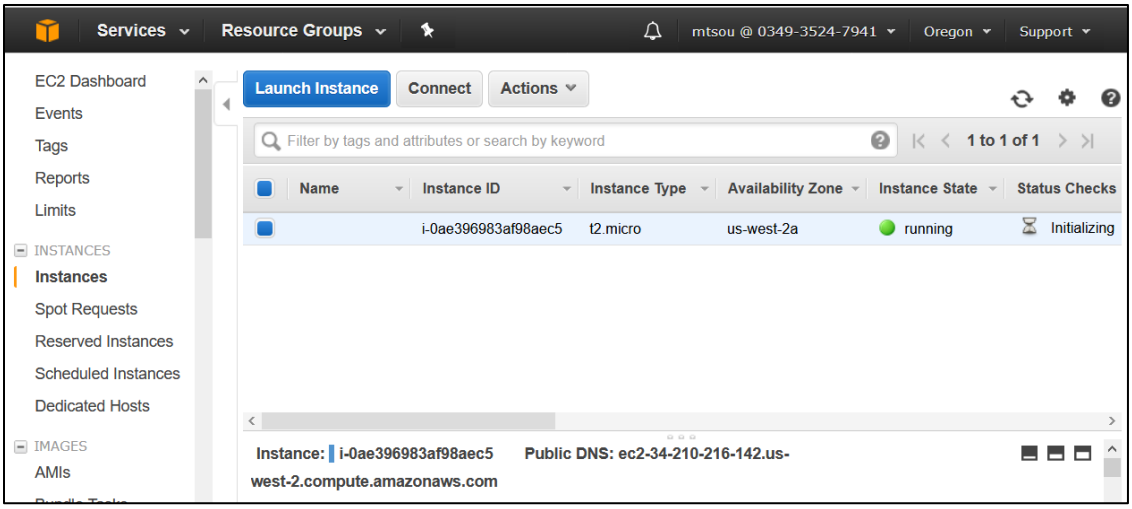
A popup window will be shown that allows you to create a key that will be used to retrieve your windows password. Select the "Create a new key pair" option and give the key a name (in this example, we have named it "**Yourname-Key**" (for example, "Ming-Key"). Next, click the "Download Key Pair" button and save the file to your local computer (the key file extension will be **.pem**).



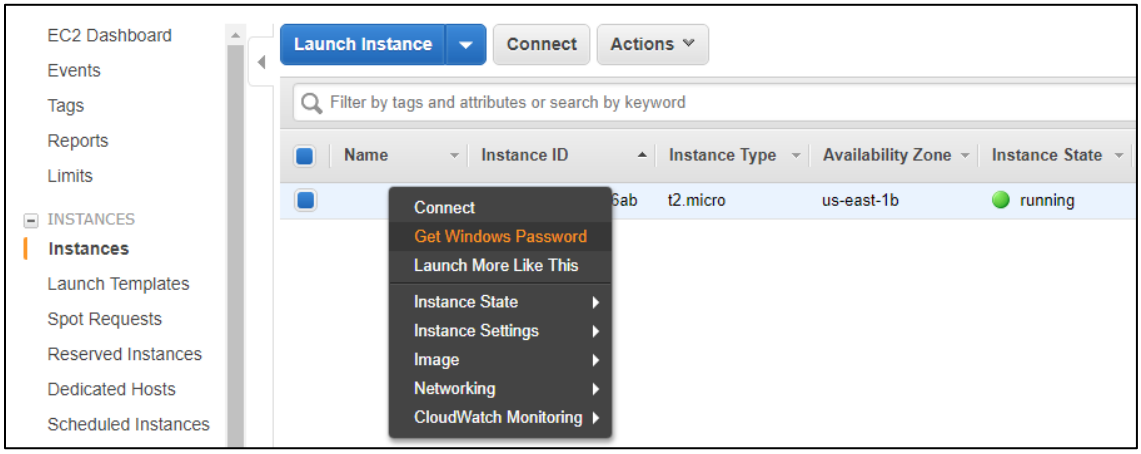
Click on "**Launch Instance**". It will show the Launch Status as the following:



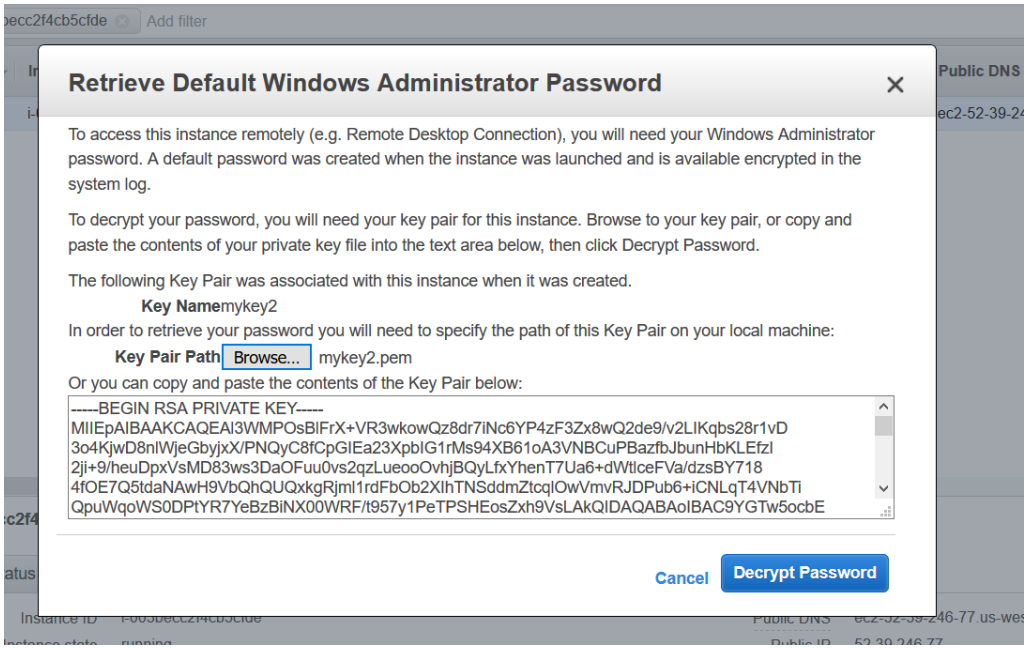
Scroll down to the end. Then click on "**View Instance**" to see the EC2 Dashboard again:



Now new instance is running. The next step is to **get the Password** to remotely login into the EC2 virtual server. **Right click on the new instance** and select the "**Get Windows Password**" option.



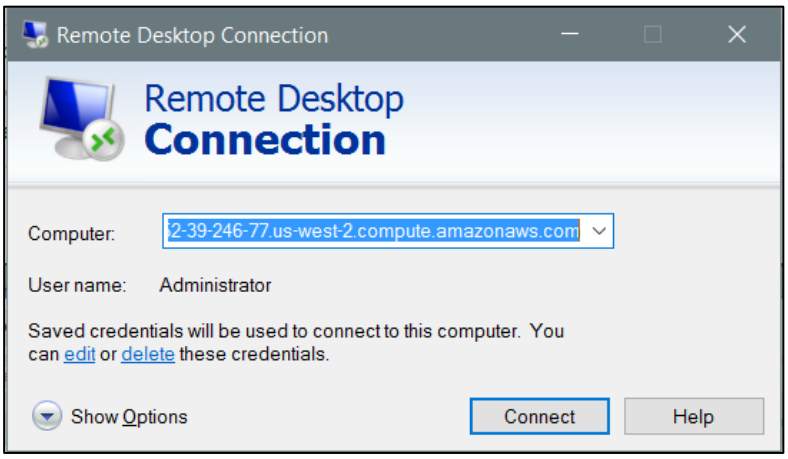
This will allow us to upload the KEY file we downloaded. (Click on the "**Choose File**" and upload the KEY file (###.pem)).



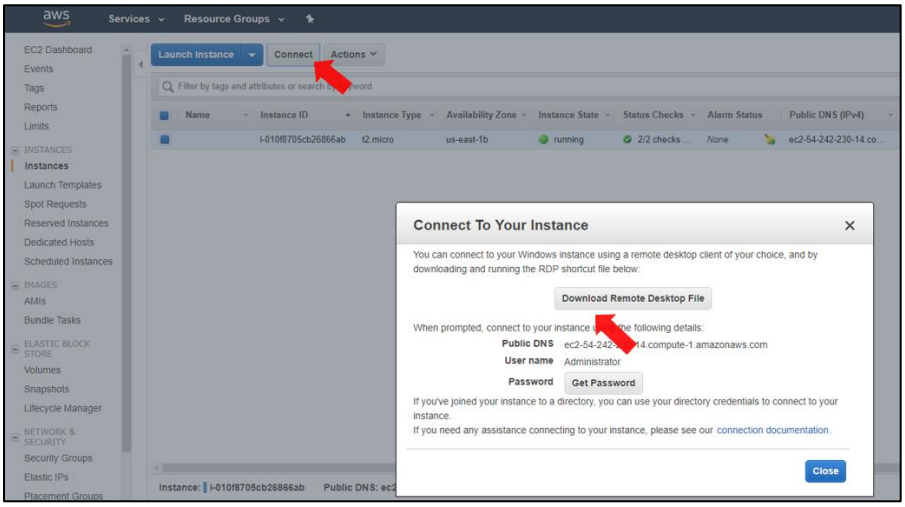
Click the "**Decrypt Password**" button, which will give us the server name, user name and password for our EC2 instance. These will be used for logging into your instance through remote desktop.

Copy the **Public DNS, User Name, and Password information** into a secured text file (or write it down).

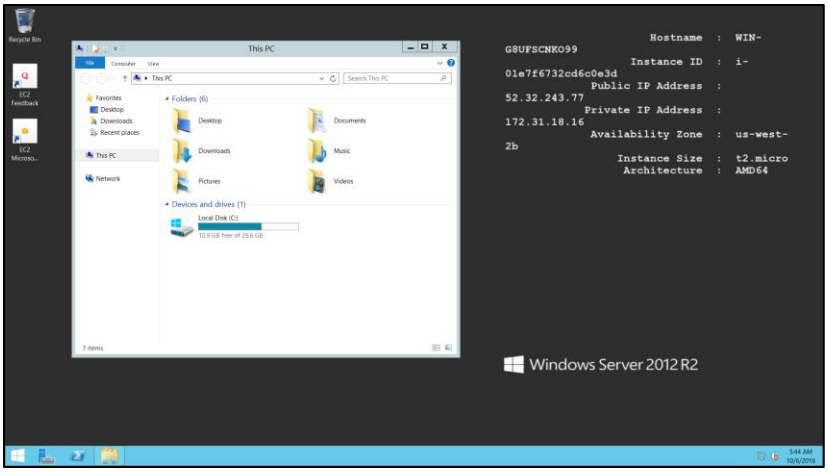
Open a Remote Desktop Connection from your local computer. Cut and paste the DNS server information for your instance. When you click "**Connect**", you will be prompted for a user name and password. Enter the information (password) from the previous step to log into the Windows Server.



Another option to use the Remote Desktop Login is to click on the [Connect] button on the top of the window. Then click on [Download Remote Desktop File]. After downloading, double click on the file to open the Remote Desktop.



After the login, you may get another yellow warning window about "The identify of the remote computer cannot be verified". Click on "Yes" to continue. Then you will see the Remote Desktop as the following:



You can **make a screen shot** of your remote desktop and save the image into your workspace. (Your final lab report will need this image).

The next step to install software and a NoSQL databases that will allow us to collect tweet and store them in a database.

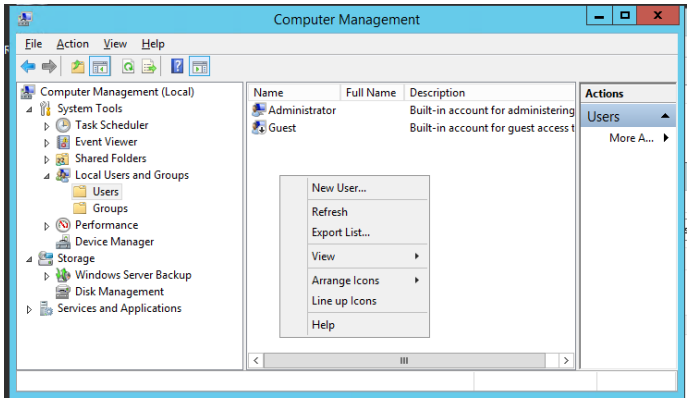
You can "**Disconnect**" your EC2 using the Remote Desktop (by clicking on the "X" on the top bar). The Server will keep running when the desktop is disconnected. You can just reconnect it when you need to continue your tasks next time. *NOTE: if you just disconnect or log out the EC2 server without shutting down, all run-time machines will cost your some credits (money).*

You can also "**Shutdown**" your EC2 server to save some cost of cloud computing. However, you need to re-start the EC2 server next time. For this exercise, please do not shut down your EC2 now.

Add one new user in this EC2 server

Our next step is to add one regular user in this EC2 server for the next exercise. Click on the Windows Icon (the lower left corner), then select the "**Administrative Tools**" icon. In the List of Tools, select "**Computer Management**". (Double click on the Computer Management). In the Computer Management Window, select "Local Users and Group", → "Users".

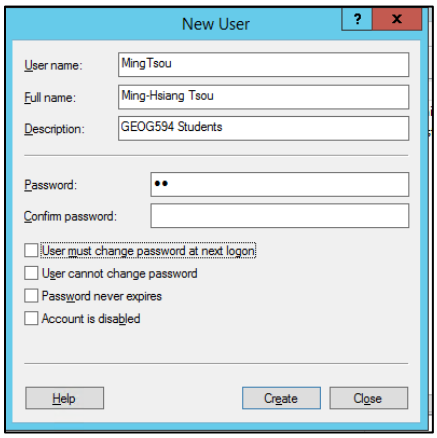
You will see the windows like this one:

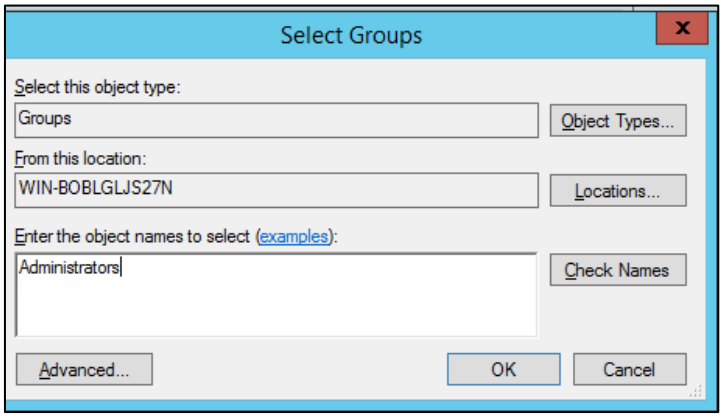


"**Right click**" on the white space under the Guest Account (See the image above) to select "**New User.**."

Fill out the new user name (for your own account or for your friends) and Password. Uncheck the box for "User must change password". Select other options you like, then click on "**Create**".

The Password must have some features (upper case, lower case) and cannot contain User names.

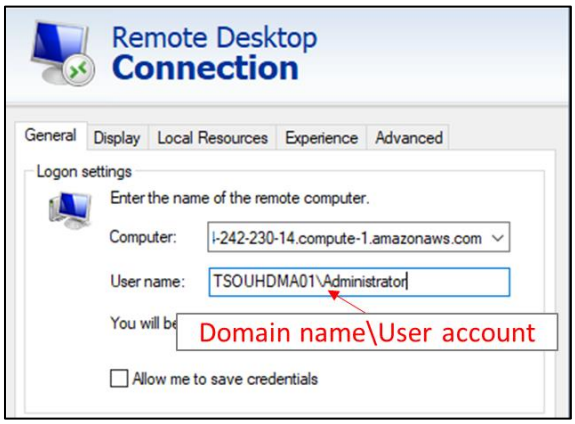




Once you create a new user name. Double click on the new account on the window to open the User Properties Window. Click on the "**Member of**" Tab, then click on "**Add…**"

Type "**Administrators**" in the Text box, then press "**OK**". Now your new Account has the "Administrators" privileges.

Now, Logout the "Administrator Account" and **login with the new user account**. (You will need to use the Remote Desktop to reconnect to EC2 Again. Use the same Domain name from the previous session).



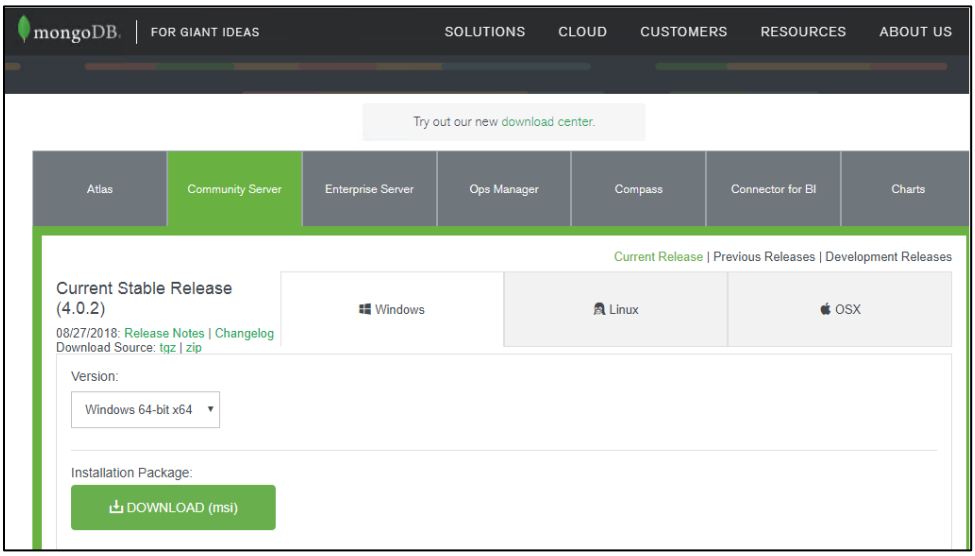
2. Install MongoDB (NoSQL) Database

After using the Remote Desktop Login to your EC2 instance with the new user account. We need to install Google Chrome. Currently, it is very difficult to install Google Chrome using the built-in Internet Explorer (In general, using a web-browser in the server is not recommended and IE has extra security

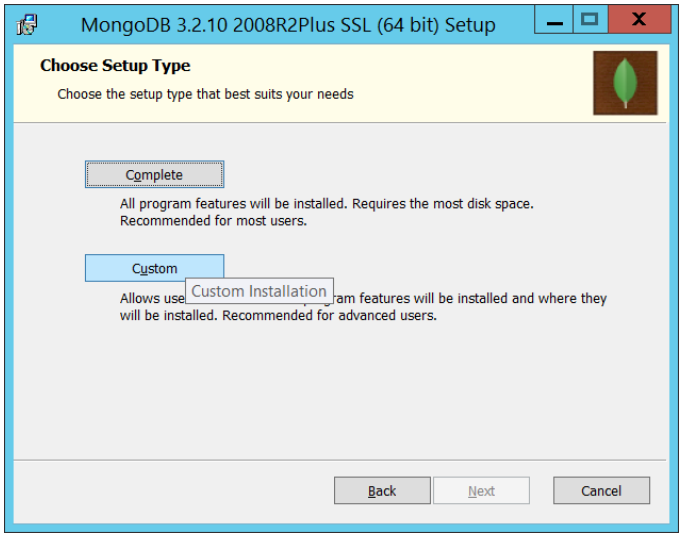
setting). The easy way is to use your local desktop machine to download the Google Chrome installer (such as "ChromeSetup"), then Drag-and-Drop into your EC2 machine. Then lunch the installer.

After installing Google Chrome, then use Google Chrome to install MongoDB. This is a database that is optimized for large collections of data that are often unstructured (for more information: https://en.wikipedia.org/wiki/MongoDB). To download MongoDB, go to https://www.mongodb.com/download-center?jmp=nav#community

Click on the "Community Server" Tab, then download the Windows installer (msi) for [Windows 64-bit x64]. You may receive a Security Alert indicating that "Do you want to run this file…" You can check the "Run" box. When you access the MongoDB website, you can add these URL into the "Trusted sites" and then continue the download process in the future.



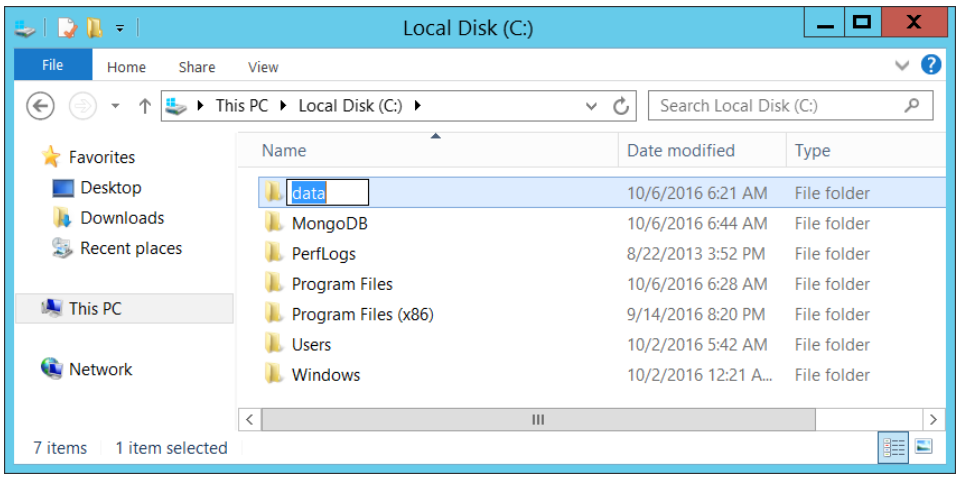
Download and Run the installer (click on accept the user agreement, etc.) and when you get to the "Choose Setup Type" page, select the "**Custom**" option.



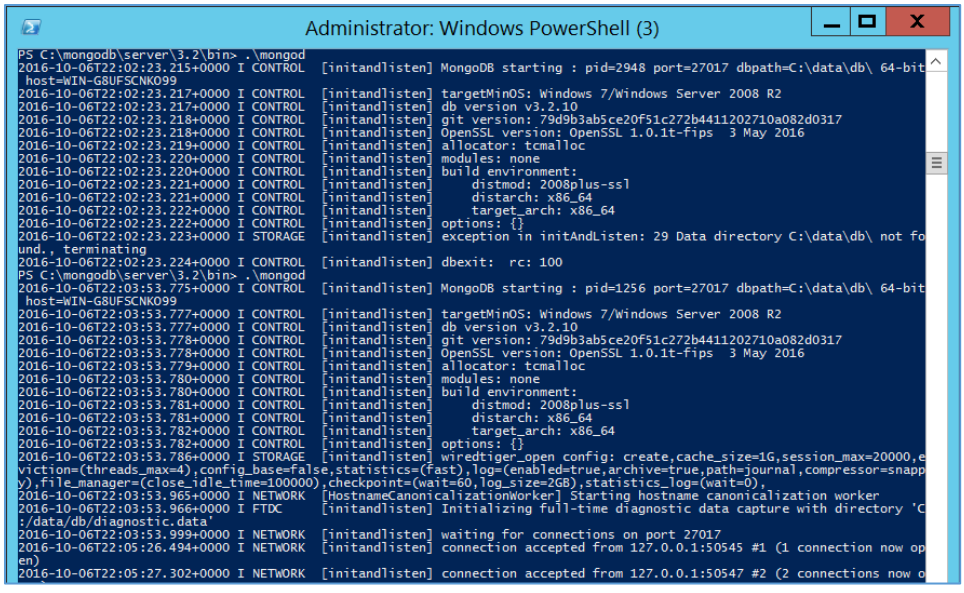
In the "Custom Setup" page, change the Location by clicking on "Browse.." and type in the Folder name "**C:\MongoDB\Server\4.0**", then click on **OK**. This will simplify the installation and running of MongoDB. Continue with the setup selecting the default options. (click on NEXT and then). Click on the [Finish] button to Exit.

Open the File Explorer (Right Click on the Start-up Menu → select "File Explorer".

Create a new folder inside the **C:** drive called "**data**." Within that 'data' folder, create a sub folder called "**db**"(C:\data\db). This is the folder in which data will be stored by MongoDB. Please make sure that the "**C:\data**" is at the same level of "C:\MongoDB" folder. Otherwise, the MongoDB may not be able to generate a database.



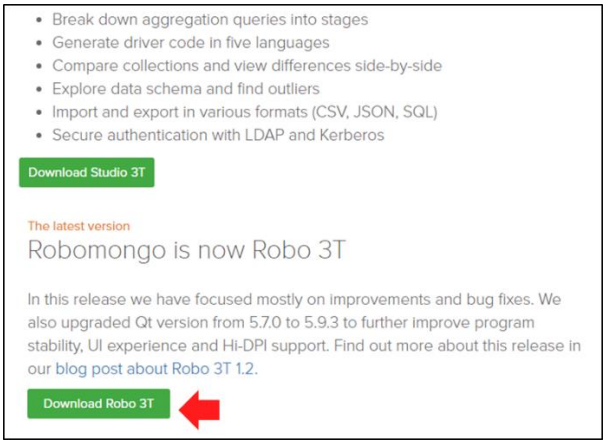
Start MongoDB by opening the [**Command Prompt (Admin)] (right click on Start-up menu)** or the Windows **PowerShell** (3) (blue icon on the Task Bar) and navigating to the folder in which you installed MongoDB (Type "**cd C:\MongoDB\server\4.0**". Navigate to the "bin" folder (Type "**cd bin**"). Then, type the command "**.\mongod**" to start the MongoDB. **Leave this window open, so MongoDB can continue to work on your EC2 machine. DO NOT close this window.**

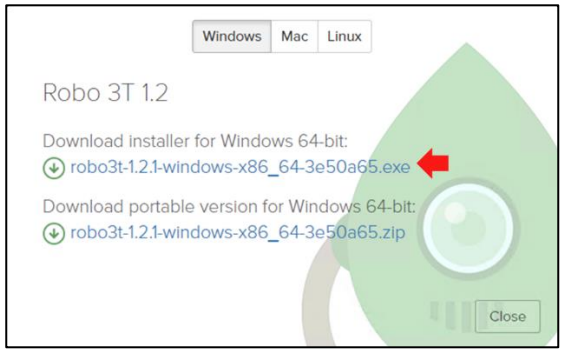


3. Install RoboMongo (a Graphic User Interface for MongoDB).

It will be helpful to install a graphical user interface for administering our MongoDB databases. Go the following page and and click on "**Download Robo 3T**" button. Click on the first link (Download **installer** for Windows 64 bit).

https://robomongo.org/download



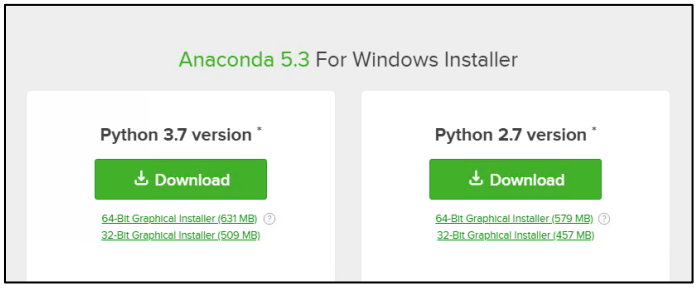


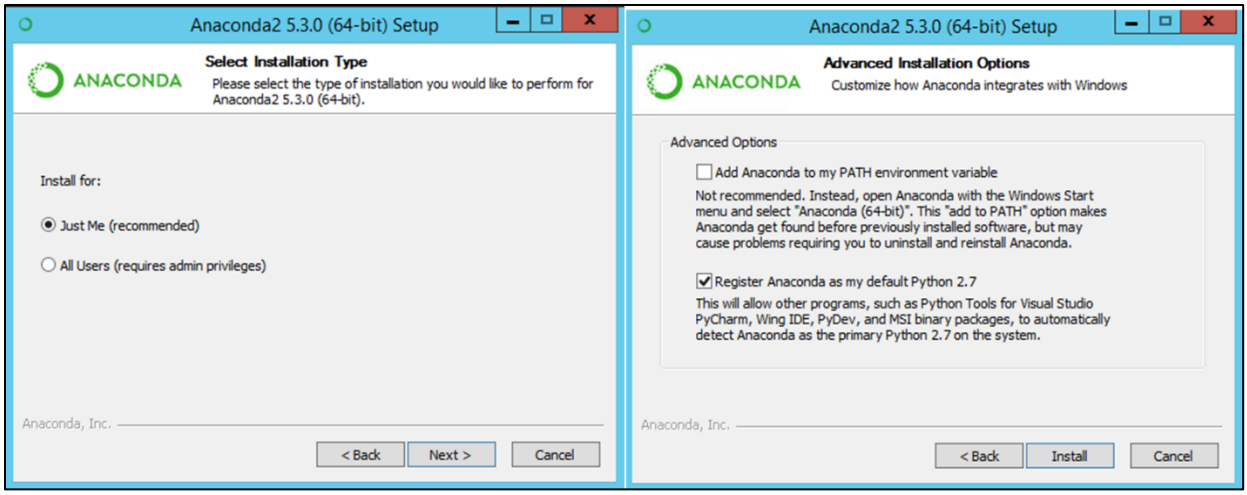
Install the Robo 3T in your EC2 machine (accept all default settings). When you run the Robo 3T first time, you will need to accept their open source license agreement.

4. Install Python and related Libraries.

Now we want to install Python and libraries that will allow us to download tweets and insert them into MonoDB. First download **Anaconda**, which is a distribution of Python that includes many libraries that are useful for data science and scientific computing. Go to the following page to download the installer for Anaconda and click the button for **Python 2.7 version, 64-Bit Installer**. (Select – "**Just Me**" during the installation)

https://www.anaconda.com/download/





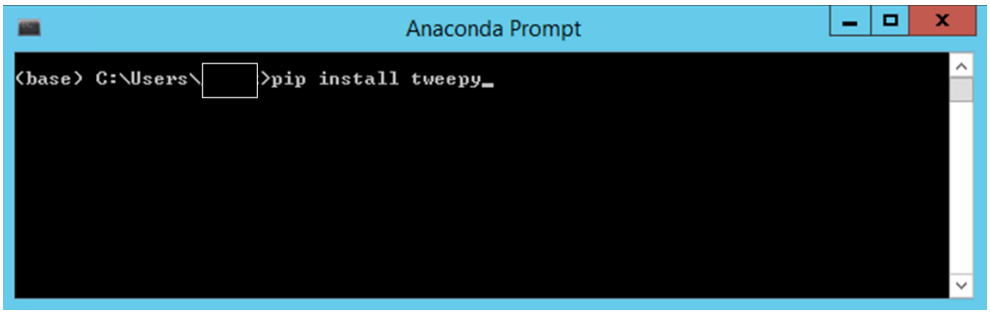
The Python 2.7 will be installed on **C:\Users\yourusername\Anaconda2** folder.

Go to the Windows start button and open "**Anaconda Prompt".**

You will see "<base> C:\Users\yourusername>" on the window.

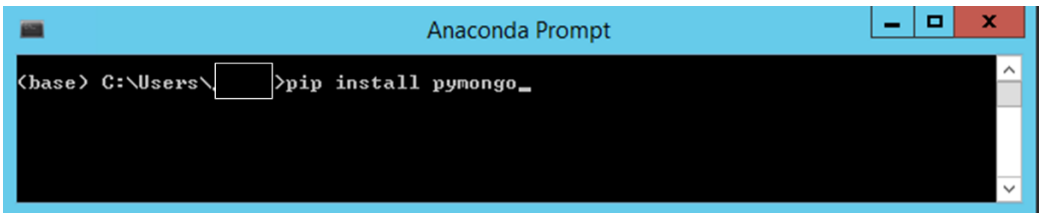
Next, enter the following command to install "tweepy," which is a python library that will allow us to collect tweets from the Twitter API:

pip install tweepy



Next, we want to install "pymongo," which is a Python library that allows us to interface with MongoDB. Type the following command to install "pymongo":

pip install pymongo



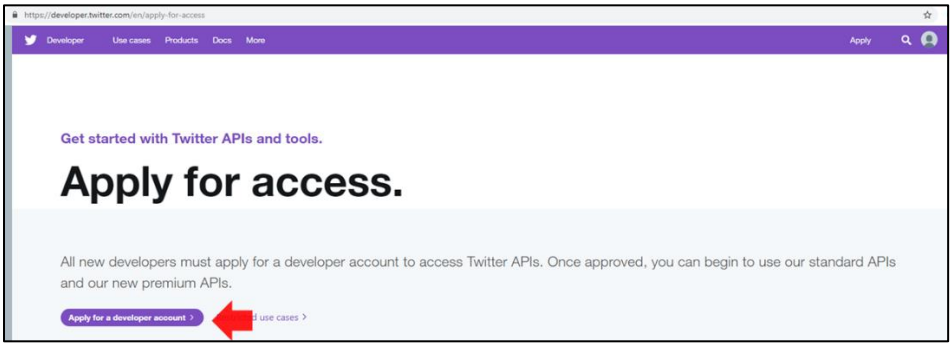
Next we will create a Python script that downloads tweets from the Twitter API and places them into MongoDB.

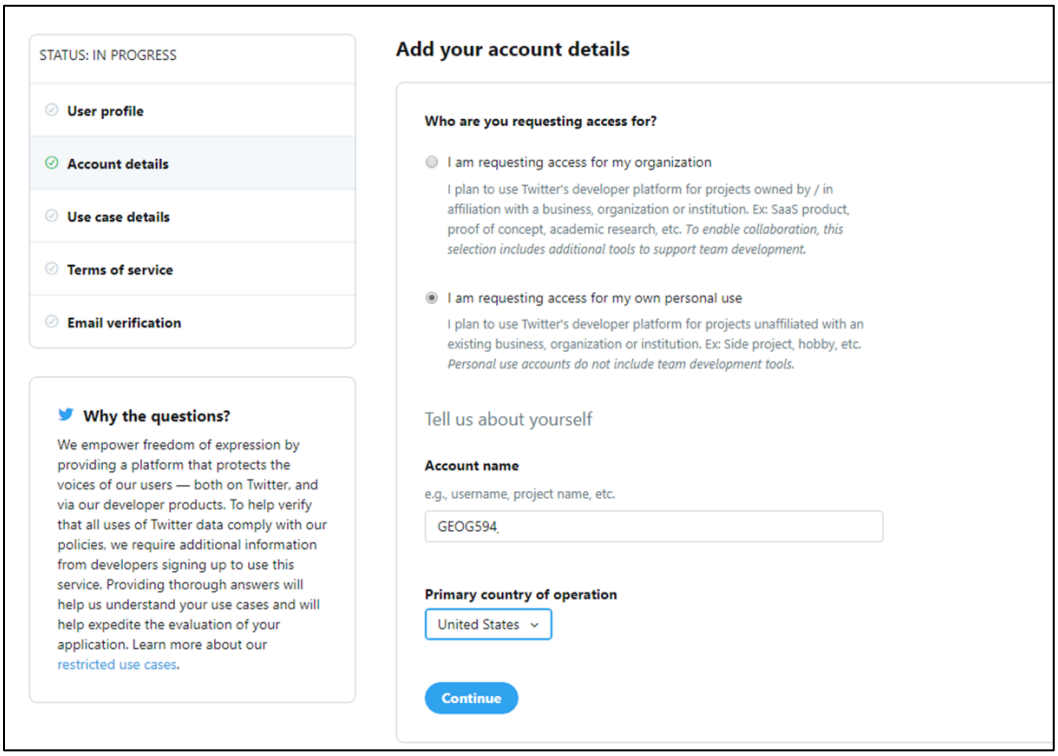
5. Get the Twitter API Keys/Tokens.

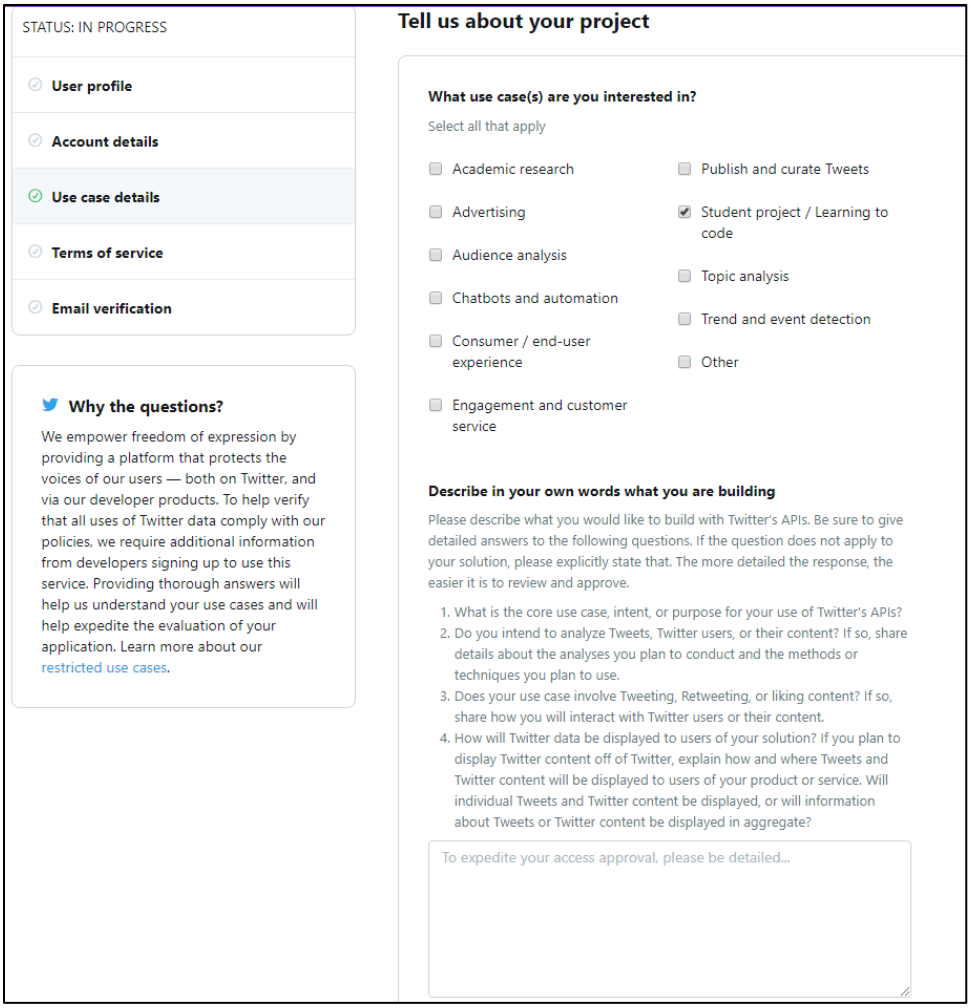
First, we will need to sign up for a Twitter account (if you don't already have one) and obtain some information that will allow us to connect to the Twitter API. Make sure to add your mobile phone number to your Twitter Profile before the following step.

Here are the simple procedure to get your Twitter API key/Tokens (next page).

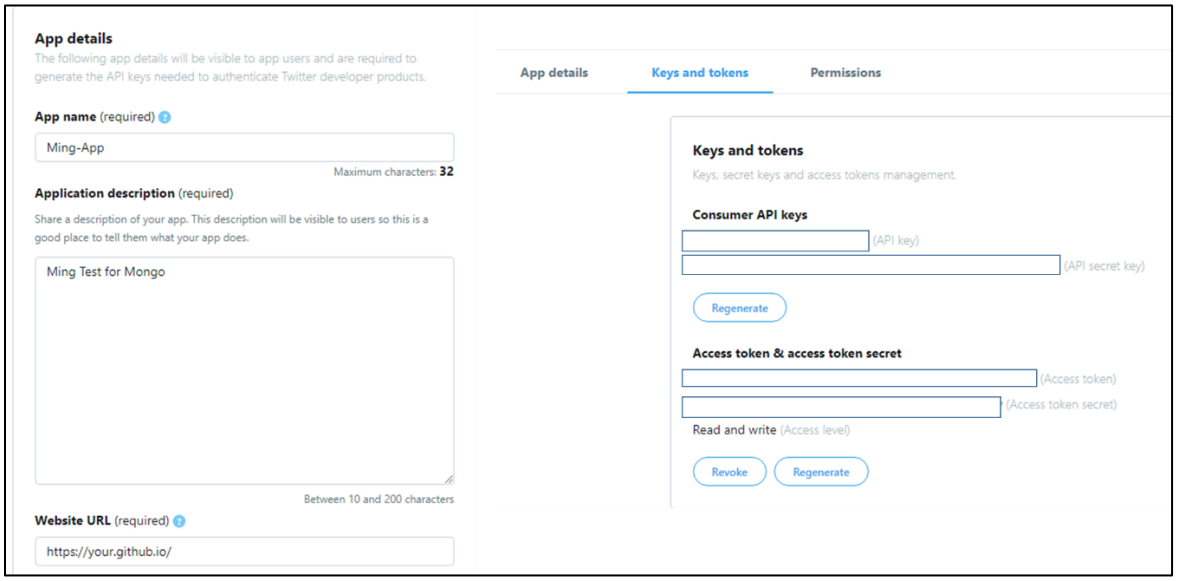
1. Log on to Twitter Developer (**https://developer.twitter.com/**) and sign in with your Twitter account.
2. Click on "Apply" next to Sign-in button.
3. Click on **"Apply for a developer account"** and **fill out a form and** submit the application.
4. Confirm your email to complete your application.







5. Once your developer account is approved, click on "**Create an App**" button.
6. Fill out the application form. (For the Website URL, you can put your GitHub homepage or if you don't have a URL yet, just put a placeholder here but remember to change it later.)
7. Please provide a detail "**Application description**", such as "*This app is to collect SDSU related keywords for analyzing the SDSU students' activities in Fall 2018 semesters*".
8. Click on "**Create your Twitter application**" button. (Note: you need to add your mobile phone to your Twitter profile before creating an application.)
9. Click on "**Keys and Access Tokens**" Tab on the upper menu (under your application name).
10. In the bottom of the page, click on "**Create my access token**". Copy and Save your Consumer key (API Key), Consumer secret (API Secret), Access Token, and Access Token Secret into your local file (in Word or text file). You will need to use the four items in the next step.



6. Create a Python Program to Collect Tweets.

Next, we want to create a Python script, called "**get_tweets.py**", which establishes a connection to the Twitter API, downloads tweets, and stores them in MongoDB:

Open a programing text editing tool, like **notepad++ (**[**https://notepad-plus-plus.org/**](https://notepad-plus-plus.org/)**)** or **ATOM** (https://atom.io/). Create the python program (**see the next page**) and save it (as "**get_tweets-yourname.py**") into your EC2 instance C:\Scripts folder (You will need to create this new folder first). Make sure to type the Twitter **API KEY, Secret, Token, and Token_Secret** into this python program.

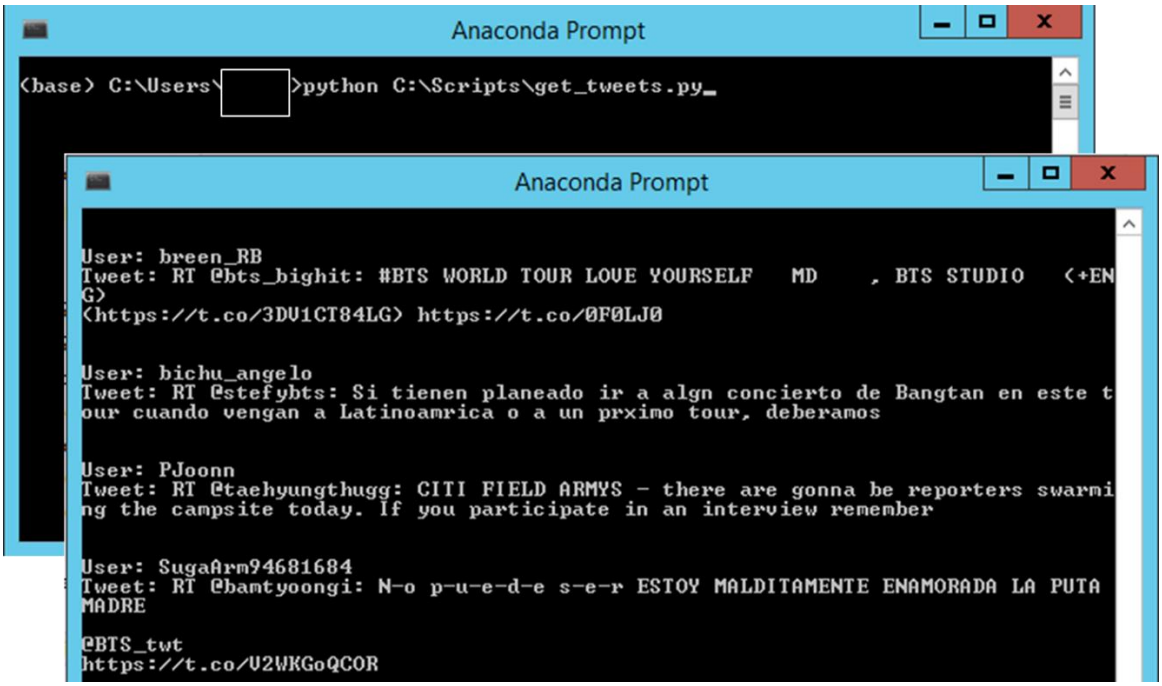
This program will search 100 Twitter messages containing "penguin" and then save the results into the MongoDB database. **Please choose our own preferred search keyword (replacing "penguin" with your own keyword, such as "SDSU" or "iPhone".**

To run this script, open a command prompt and navigate to the folder where you saved your script into a new folder "**C:\Scripts**" (you will create this new folder first). Assuming you have called your file "**get_tweets-yourname.py**", you can run the following python command to download tweets related to your keyword and store them in MongoDB:

python C:\Scripts\get_tweets-yourname.py

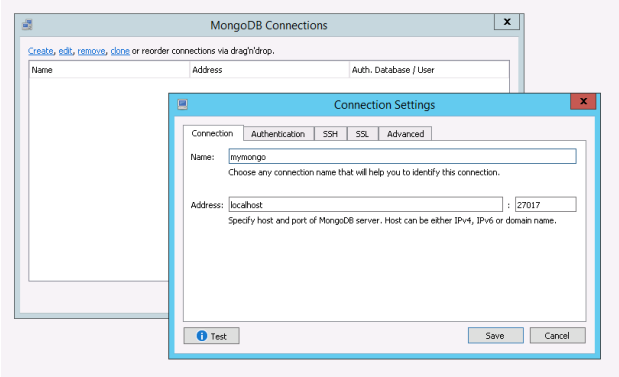
“get_tweets.py”

```python
import tweepy
import pymongo

# create a connection to MongoDB
client = pymongo.MongoClient()
# create a database called 'data'
database = client['data']
# create a collection (ie, a table) called 'tweets'
collection = database['tweets']


# The consumer keys can be found on your application's Details
# page located at https://dev.twitter.com/apps (under "OAuth settings" and "Your access token")
consumer_key = '<YOUR_CONSUMER_KEY>'
consumer_secret = '<YOUR_CONSUMER_SECRET>'
access_token = '<YOUR_ACCESS_TOKEN>'
access_token_secret = '<YOUR_ACCESS_TOKEN_SECRET>'

# authenticate your key with oauth
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.secure = True
auth.set_access_token(access_token, access_token_secret)

# setup the API with your Keys
# also, format the return in JSON
api = tweepy.API(auth, parser=tweepy.parsers.JSONParser())

# Calling the search method, with several parameter
# q: querying keyword
# count: limits the numbers of return
result = api.search(q='penguin', count=100)

# grab the tweets from the search result
tweets = result['statuses']

# insert these tweets into the 'tweets' collection
collection.insert(tweets)


# now, let's query the tweets we just stored in the collection
# collection.find() willl return all records in the collection
records = collection.find()

for record in records:
    # get user and tweet text (note: text must be converted to ascii to display in the console)
    user = record['user']['screen_name']
    tweet = record['text'].encode('ascii', 'ignore')

    print 'User:', user
    print 'Tweet:', tweet
    print '\n'
```

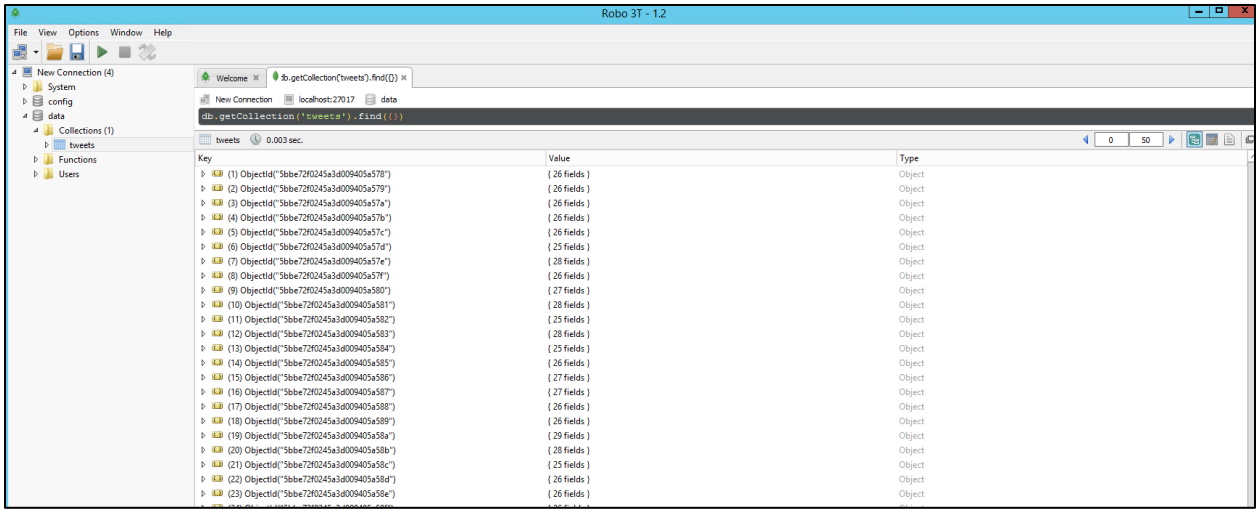


Note: Some of the tweets in the screenshot above do not have the selected keyword in the text. This is because the Twitter API will return tweets if the user's screen name also contains the keyword (even if the text does not).

Next, we can open up Robo 3T. The first time this program is opened it will ask you to create a MongoDB connection. Click the "**Create**" link and enter a name for our connection. The default settings (localhost) should be fine to configure this connection. Click "**Save**" and then connect to your connection.



In the Python script we created above, we stored the tweets in the "tweets" collection within the "data" database. Navigate to this collection within Robo 3T. From here we can inspect the tweets we stored using the Python script: data → Collection(1) → tweets



Take a close look at the content of one selected tweet inside the MongoDB. What are the KEYS in the databases? What are the Values associated with each KEY? What type of Value are they? Read the Data Model Introduction in MongoDB online document:

https://docs.mongodb.com/manual/core/data-modeling-introduction/

After finishing all steps and tasks, you can just keep your free EC2 Instance running, since it is a free-tier instance. If you are paying the usage of EC2 instance, you can "STOP" it when you are not running the server to save money (similar to "shutdown" your server). When you need to re-use the Instance, you can "Start" the instance again.

If you select "STOP" for your instance, you will see a warning message that "Any data on the ephemeral storage of your instance will be lost". Since we are not using any ephemeral storage in this exercise. It is fine to STOP the instance. (Note, but you will need to "re-launch" the MongoDB database when you re-START this instance next time).

After finishing this Web Exercise, Please use your own words to answer the following questions (next page): (DO NOT COPY any web resources or Wikipedia texts. We will check your answers with Blackboard tools to verify that your responses are uniquely yours.) By submitting your answers (paper) to Blackboard, you agree: (1) that you are submitting your paper to be used and stored as part of the SafeAssign™ services in accordance with the Blackboard Privacy Policy; (2) that your institution may use your paper in accordance with your institution's policies; and (3) that your use of SafeAssign will be without recourse against Blackboard Inc. and its affiliates.

SafeAssign accepts files in .doc, .docx, .docm, .ppt, .pptx, .odt, .txt, .rtf, .pdf, and .html file formats only. Files of any other format will not be checked through SafeAssign.

LAB-4 Additional Assignment:

1. What are the advantages and disadvantage of using Amazon EC2 for building a research project website?
2. Attached the screenshot of your EC2 Virtual Server.
3. What are the differences between traditional SQL databases and NoSQL databases?
4. What is your selected keyword in the **get_tweets-yourname.py** to collect tweets? How many tweets did you get? **List the content of first 3 tweets in your report. (Please include all metadata elements in the example, including ID, source, text, location, etc.) Discuss the search results and how to improve the keyword search for your subject.**

Please submit your LAB-4 Answers (in a MS Word or a PDF file format only) to the Blackboard System BEFORE the DUE DATE/TIME.

Additional Step (Optional):

In the **get_tweets.py,** we only collected 100 tweets per keyword search. The twitter Search API only allows up to 100 tweets per query. To increase the number of search result tweets, we will send our query to the Search API is through Tweepy's **Cursor** method. The Cursor will automatically send queries to the Search API until we have collected the maximum number of tweets that we specified, or until we reach the end of the Search API database. You can try to modify your get_tweets.py to the next page codes and test if you can get 1000 tweets for your selected keywords.

```python
##Python 2.7
import tweepy
import pymongo

# create a connection to MongoDB
client = pymongo.MongoClient()

# create a database called 'data'
database = client['data']

# create a collection (i.e., a table) called 'tweets'
collection = database['tweets']

consumer_key = 'Type Yours'
consumer_secret= 'Type Yours'
access_token = 'Type Yours'
access_token_secret='Type Yours'

# authenticate your key with Oauth
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.secure= True
auth.set_access_token(access_token,access_token_secret)

# setup the API with your Keys
api = tweepy.API(auth, wait_on_rate_limit=True)

max_tweets= 1000  #limits the number of return
query='BTS'  #querying keyword

#The twitter Search API allows up to 100 tweets per query.
#So, we will send our query to the Search API is through Tweepy's Cursor method.
#The Cursor will automatically send queries to the Search API until we have collected the maximum number of tweets
#that we specified, or until we reach the end of the Search API database.

for status in tweepy.Cursor(api.search, q=query).items(max_tweets):
    tweets = status._json # convert the result to JSON
    collection.insert(tweets) #Insert these tweets into the 'tweets' collection
    user = tweets['user']['screen_name']
    tweet = tweets['text'].encode('ascii', 'ignore')

    print'User:', user,'\n', 'Tweet:',tweet, '\n'
```