Web Exercise 04: Working with Amazon AWS Educate Starter

Due Date: **October 18** (Tuesday), 2016. 3:30pm on Blackboard.

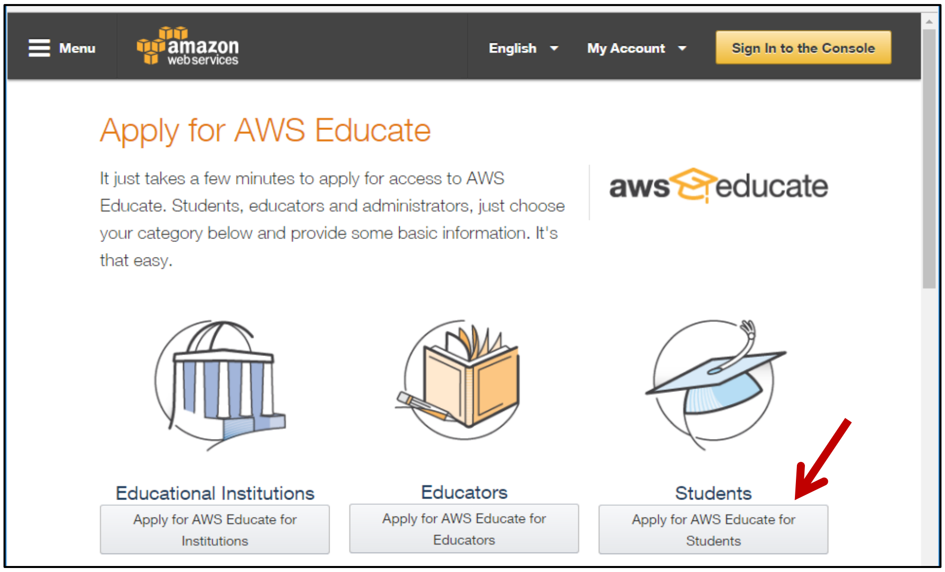
Grade: **15 points.**

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/

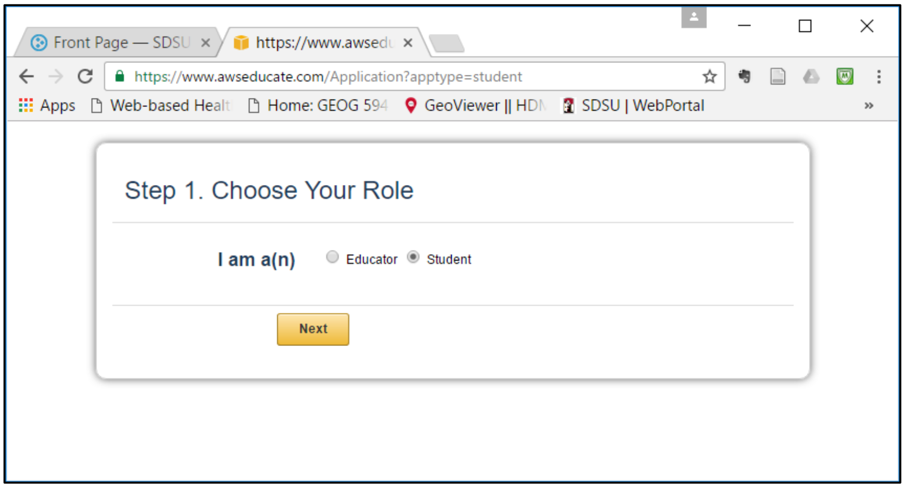
We will be creating an Amazon Web Services (**AWS**) account associated with San Diego State University, so click the "**Students: [Apply for AWS Educate for Students]**" button (the right one):



Apply for AWS Educate

It just takes a few minutes to apply for access to AWS Educate. Students, educators and administrators, just choose your category below and provide some basic information. It's that easy.

Then, select your role as "Student". Click on [Next].



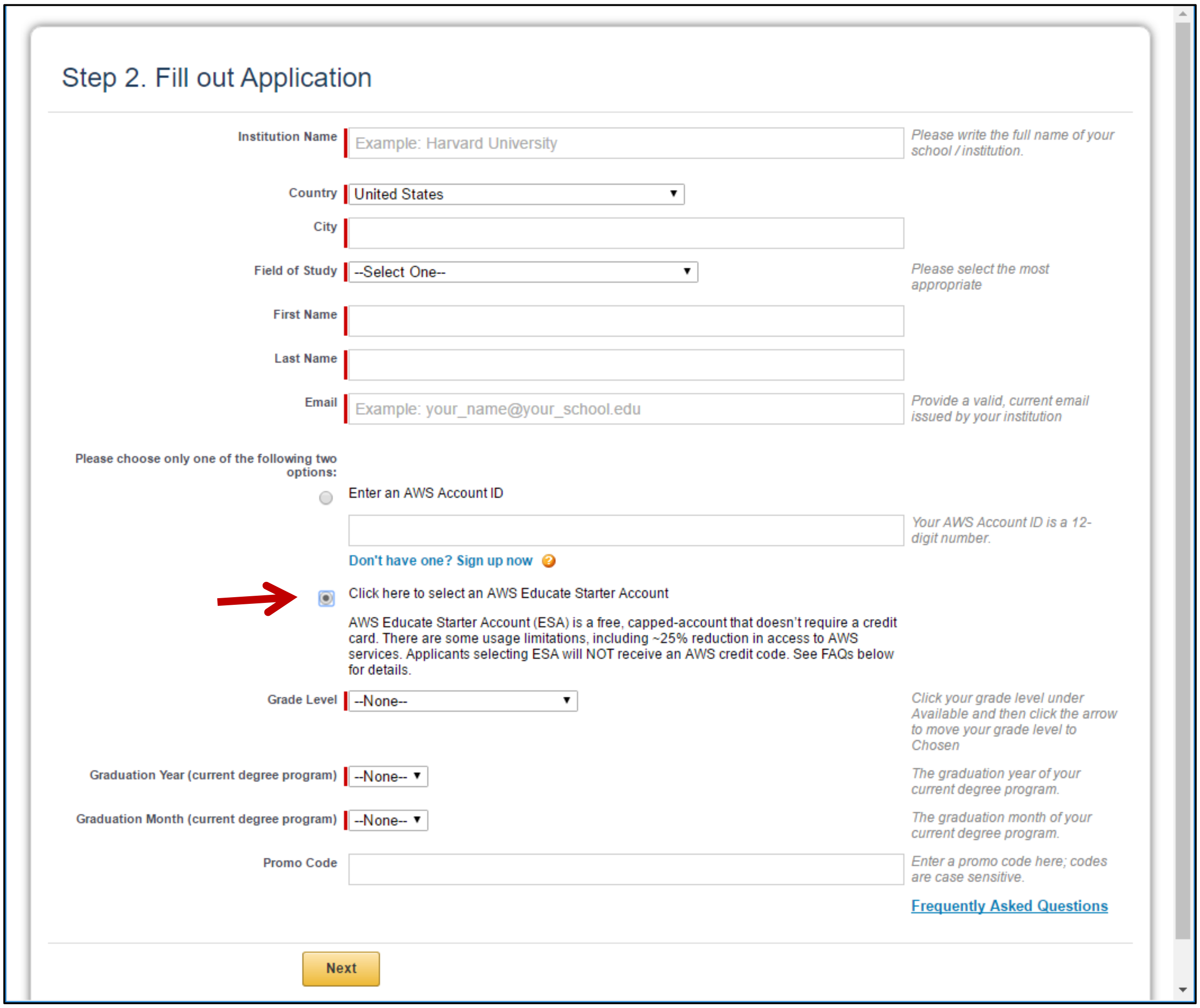
Step 1. Choose Your Role

I am a(n) ○ Educator ● Student

Next

Fill out your application. Type "**San Diego State University**" in the Institution Name. Make sure to use your SDSU mail addresses (such as @rohan.sdsu.edu, @mail.sdsu.edu, or @sdsu.edu) and select the "**Click here to select an AWS Educate Starter Account**." (Ignore the Promotion Code).

Note: **DO NOT SELECT the first option** (Enter an AWS Account ID) since it will require a credit card number to register an AWS account).



Step 2. Fill out Application

Institution Name: Example: Harvard University — *Please write the full name of your school / institution.*

Country: United States ▾

City:

Field of Study: --Select One-- ▾ — *Please select the most appropriate*

First Name:

Last Name:

Email: Example: your_name@your_school.edu — *Provide a valid, current email issued by your institution*

Please choose only one of the following two options:

○ Enter an AWS Account ID

Your AWS Account ID is a 12-digit number.

Don't have one? Sign up now

● Click here to select an AWS Educate Starter Account

AWS Educate Starter Account (ESA) is a free, capped-account that doesn't require a credit card. There are some usage limitations, including ~25% reduction in access to AWS services. Applicants selecting ESA will NOT receive an AWS credit code. See FAQs below for details.

Grade Level: --None-- ▾ — *Click your grade level under Available and then click the arrow to move your grade level to Chosen*

Graduation Year (current degree program): --None-- ▾ — *The graduation year of your current degree program.*

Graduation Month (current degree program): --None-- ▾ — *The graduation month of your current degree program.*

Promo Code: — *Enter a promo code here; codes are case sensitive.*

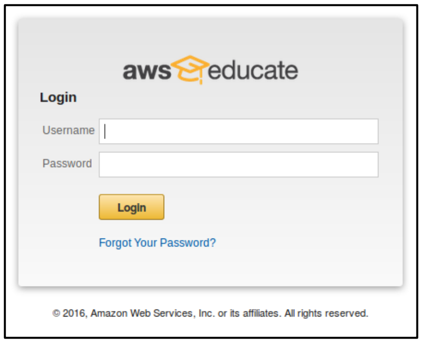
Frequently Asked Questions

After you submit the request, you will receive a confirmation email from Amazon first. After one or two days, your application will be approved by Amazon and you will be able to set up your user name and password to access your free AWS Educate Starter account.

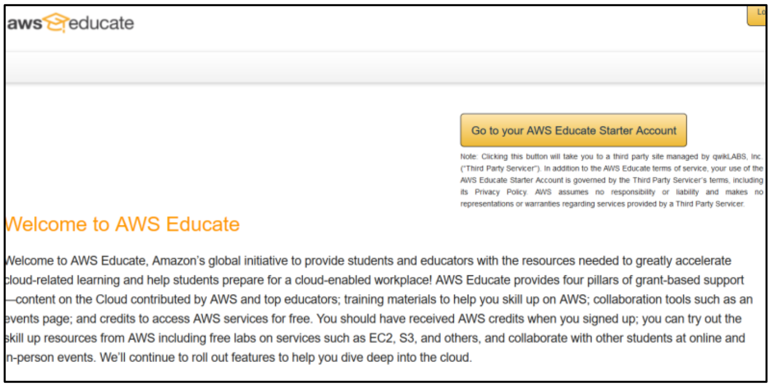
Once you have received an email confirming that your account has been approved, log into the AWS Educate Portal by going to the following URL:

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

Once you log in, you should see the follow page. Click the “Go to your AWS Educate Starter Account” button.



© 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

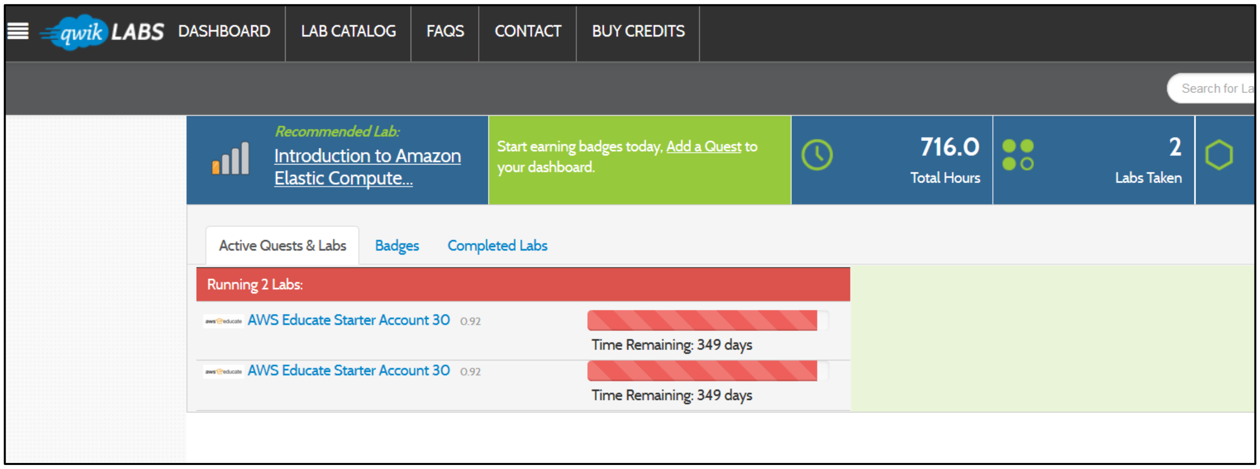


Note: Clicking this button will take you to a third party site managed by qwikLABS, Inc. ("Third Party Servicer"). In addition to the AWS Educate terms of service, your use of the AWS Educate Starter Account is governed by the Third Party Servicer's terms, including its Privacy Policy. AWS assumes no responsibility or liability and makes no representations or warranties regarding services provided by a Third Party Servicer.

Welcome to AWS Educate

Welcome to AWS Educate, Amazon’s global initiative to provide students and educators with the resources needed to greatly accelerate cloud-related learning and help students prepare for a cloud-enabled workplace! AWS Educate provides four pillars of grant-based support—content on the Cloud contributed by AWS and top educators; training materials to help you skill up on AWS; collaboration tools such as an events page; and credits to access AWS services for free. You should have received AWS credits when you signed up; you can try out the skill up resources from AWS including free labs on services such as EC2, S3, and others, and collaborate with other students at online and in-person events. We’ll continue to roll out features to help you dive deep into the cloud.

The Qwik Labs 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 one account. Click the links that says “**AWS Educate Stater Account 75.**



Running 2 Labs:

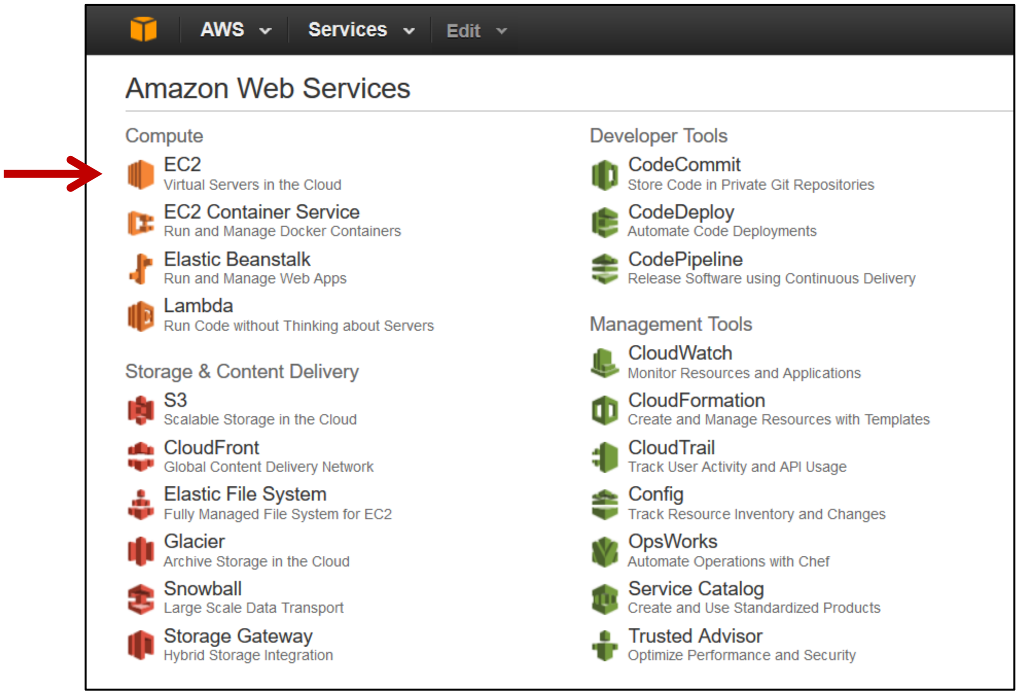
AWS Educate Starter Account 30 0.92 Time Remaining: 349 days

AWS Educate Starter Account 30 0.92 Time Remaining: 349 days

The next step is to **start** the lab. After you start the lab, the “**Open Console**” button will become active. Click on the “**Open Console**” to allow us to configure our Amazon EC2 instances.

This will take us to the Amazon Web Services dashboard. Form here you can configure and administer your Amazon services. For this lab, we are only concerned with Amazon's EC2 service.

Click the "**EC2**" icon under the "Compute" menu.



Amazon Web Services

Compute

EC2
Virtual Servers in the Cloud

EC2 Container Service
Run and Manage Docker Containers

Elastic Beanstalk
Run and Manage Web Apps

Lambda
Run Code without Thinking about Servers

Storage & Content Delivery

S3
Scalable Storage in the Cloud

CloudFront
Global Content Delivery Network

Elastic File System
Fully Managed File System for EC2

Glacier
Archive Storage in the Cloud

Snowball
Large Scale Data Transport

Storage Gateway
Hybrid Storage Integration

Developer Tools

CodeCommit
Store Code in Private Git Repositories

CodeDeploy
Automate Code Deployments

CodePipeline
Release Software using Continuous Delivery

Management Tools

CloudWatch
Monitor Resources and Applications

CloudFormation
Create and Manage Resources with Templates

CloudTrail
Track User Activity and API Usage

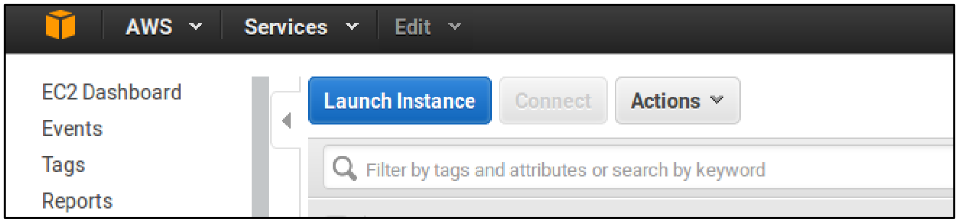
Config
Track Resource Inventory and Changes

OpsWorks
Automate Operations with Chef

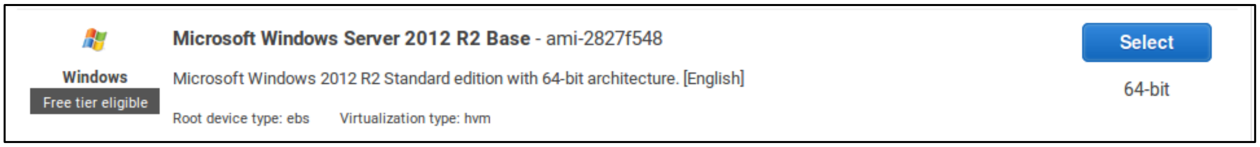
Service Catalog
Create and Use Standardized Products

Trusted Advisor
Optimize Performance and Security

This 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.

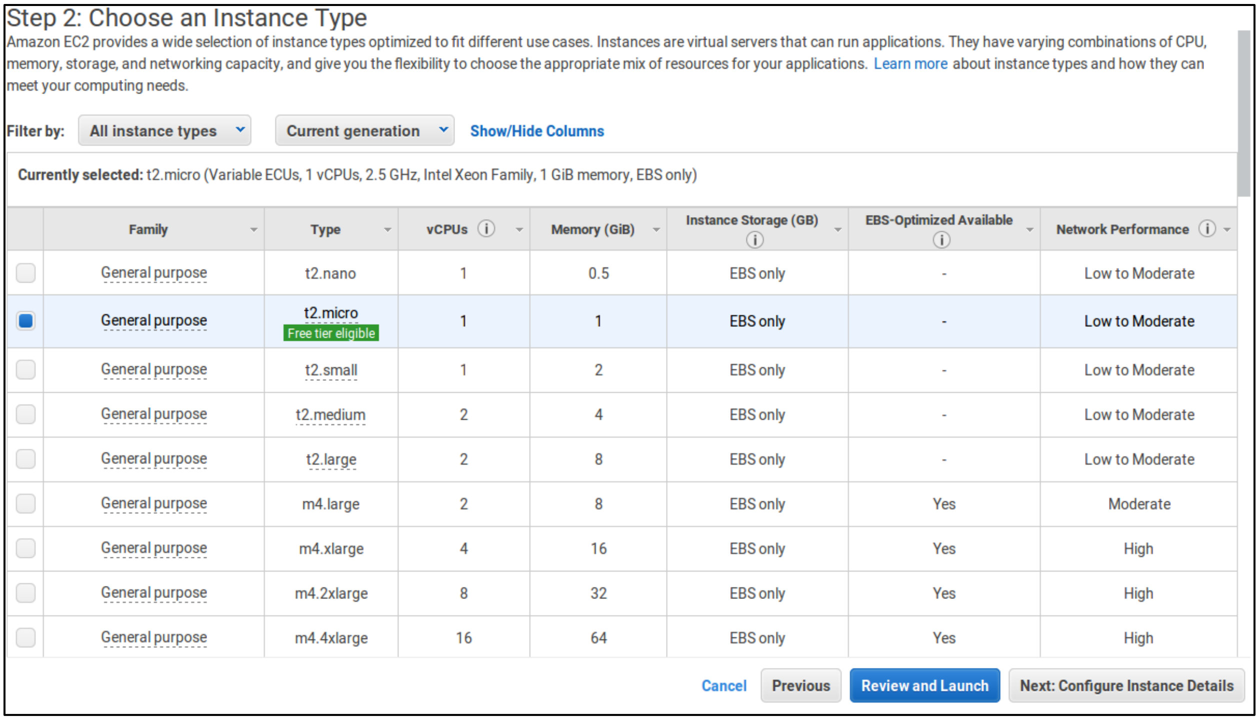


Microsoft Windows Server 2012 R2 Base - ami-2827f548

Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]

Root device type: ebs Virtualization type: hvm

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.



Step 2: Choose an Instance Type

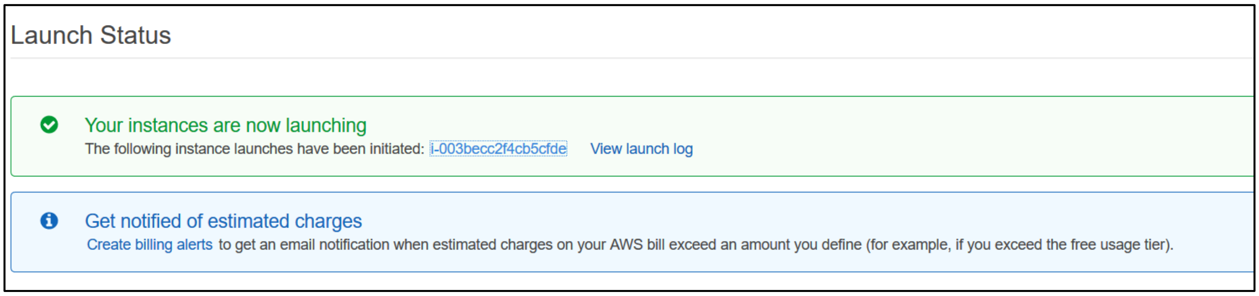
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types · Current generation

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
□	General purpose	m4.4xlarge	16	64	EBS only	Yes	High

Once we have launched our instance, a page will tell us our launch status. In order to view our instances, click the hyperlink containing the instance number (here it's i-003becc2f4cb5cfde).



Launch Status

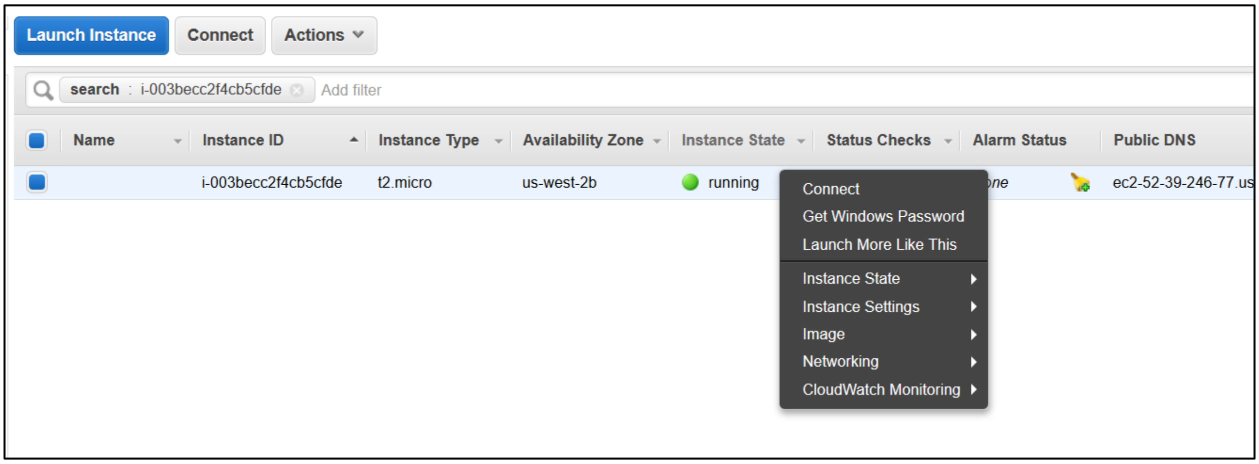
Your instances are now launching

The following instance launches have been initiated: i-003becc2f4cb5cfde View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

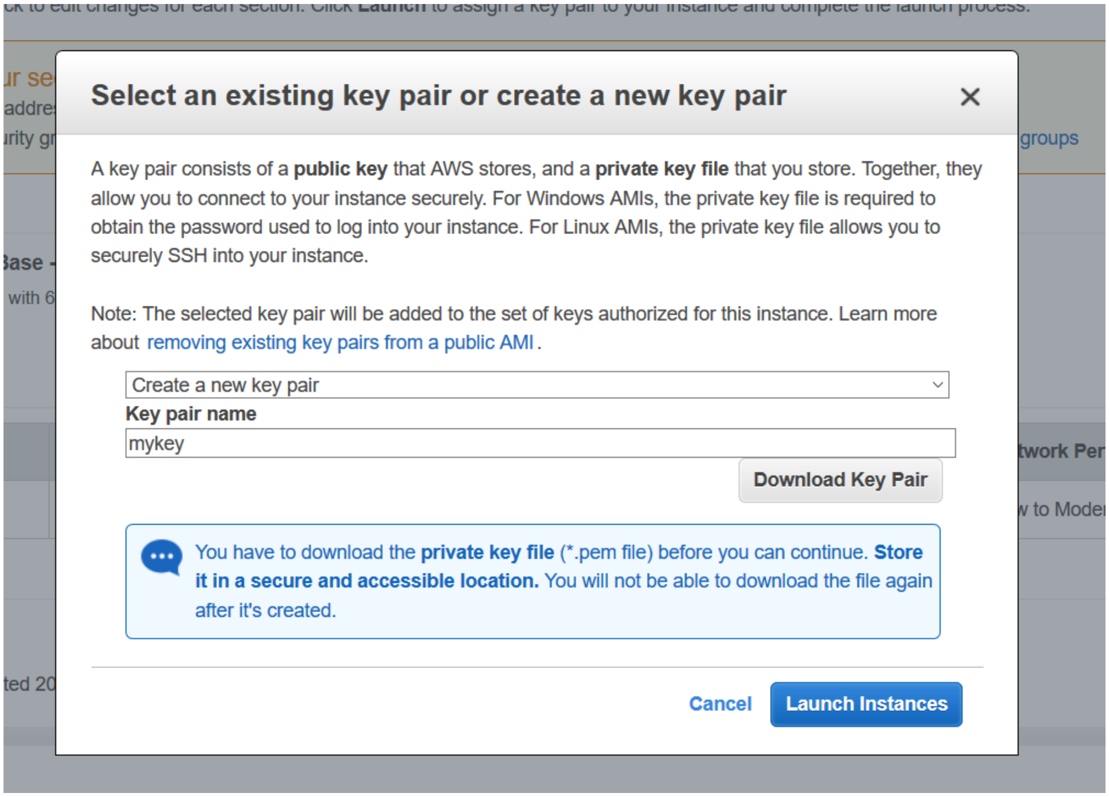
Now that we have created an instance, we need to configure it in order to make a connection through remote desktop. Right click the instance we just created and choose the "**Get Windows Password**" option.



search : i-003becc2f4cb5cfde Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-003becc2f4cb5cfde	t2.micro	us-west-2b	running		ne	ec2-52-39-246-77.us

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.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

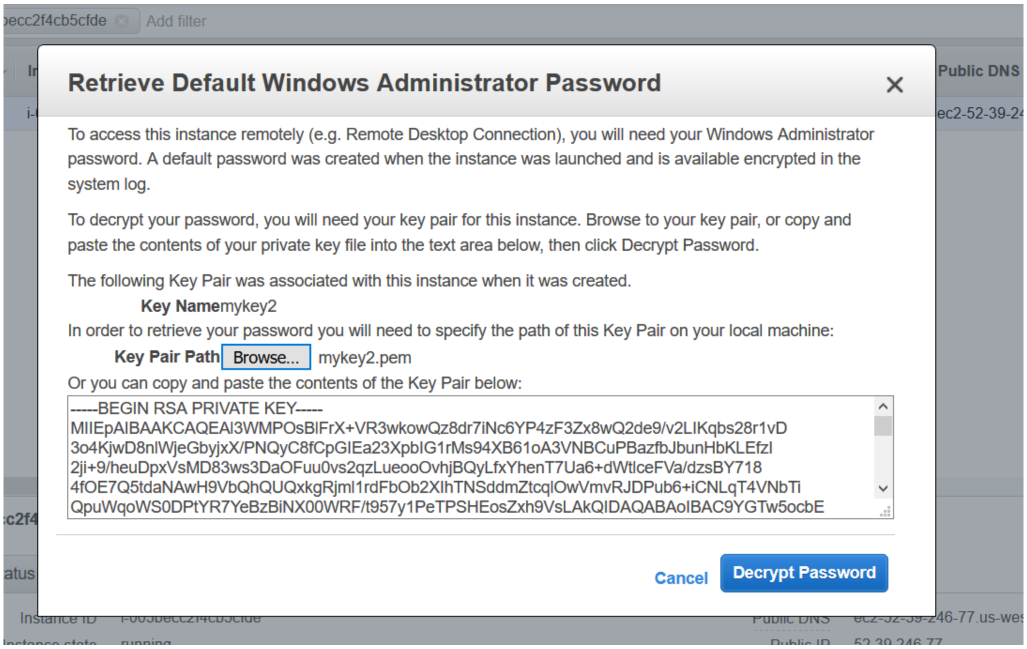
Create a new key pair

Key pair name

mykey

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

As before, we can right click on our instance and select the "Get Windows Password" option. This will allow us to upload the file we downloaded.



Retrieve Default Windows Administrator Password

To access this instance remotely (e.g. Remote Desktop Connection), you will need your Windows Administrator password. A default password was created when the instance was launched and is available encrypted in the system log.

To decrypt your password, you will need your key pair for this instance. Browse to your key pair, or copy and paste the contents of your private key file into the text area below, then click Decrypt Password.

The following Key Pair was associated with this instance when it was created.

Key Namemykey2

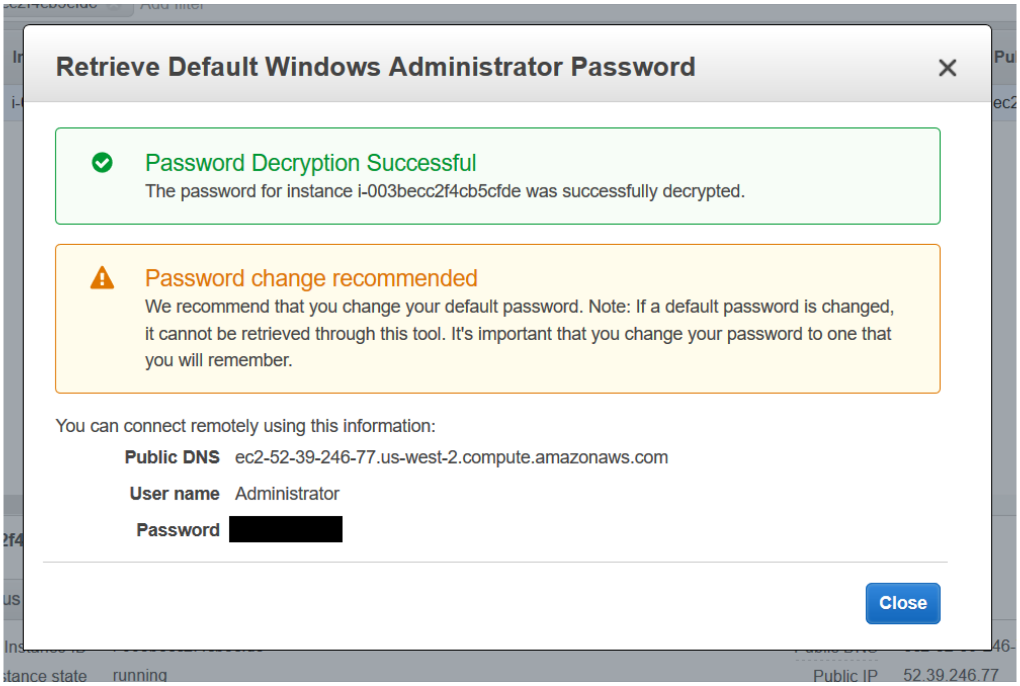
In order to retrieve your password you will need to specify the path of this Key Pair on your local machine:

Key Pair Path Browse... mykey2.pem

Or you can copy and paste the contents of the Key Pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAl3WMPOsBlFrX+VR3wkowQz8dr7iNc6YP4zF3Zx8wQ2de9/v2LIKqbs28r1vD
3o4KjwD8nlWjeGbyjxX/PNQyC8fCpGIEa23XpbIG1rMs94XB61oA3VNBCuPBazfbJbunHbKLEfzl
2ji+9/heuDpxVsMD83ws3DaOFuu0vs2qzLueooOvhjBQyLfxYhenT7Ua6+dWtlceFVa/dzsBY718
4fOE7Q5tdaNAwH9VbQhQUQxkgRjml1rdFbOb2XIhTNSddmZtcqlOwVmvRJDPub6+iCNLqT4VNbTi
QpuWqoWS0DPtYR7YeBzBiNX00WRF/t957y1PeTPSHEosZxh9VsLAkQIDAQABAoIBAC9YGTw5ocbE
```

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 our instance through remote desktop.



Retrieve Default Windows Administrator Password

Password Decryption Successful

The password for instance i-003becc2f4cb5cfde was successfully decrypted.

Password change recommended

We recommend that you change your default password. Note: If a default password is changed, it cannot be retrieved through this tool. It's important that you change your password to one that you will remember.

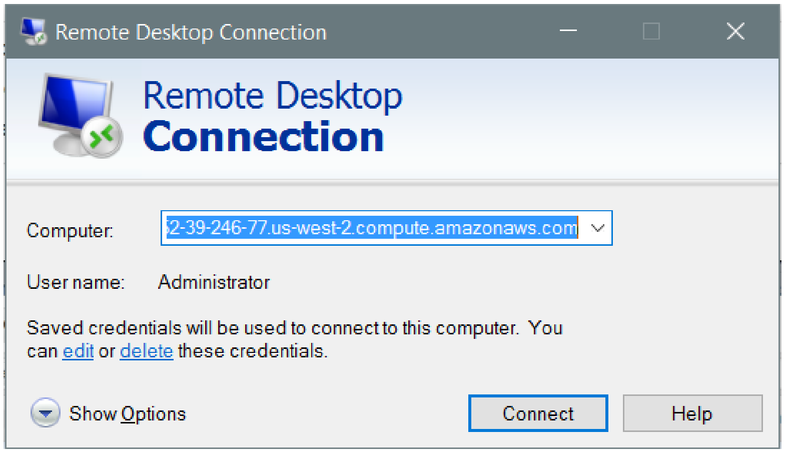
You can connect remotely using this information:

Public DNS ec2-52-39-246-77.us-west-2.compute.amazonaws.com

User name Administrator

Password [redacted]

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.



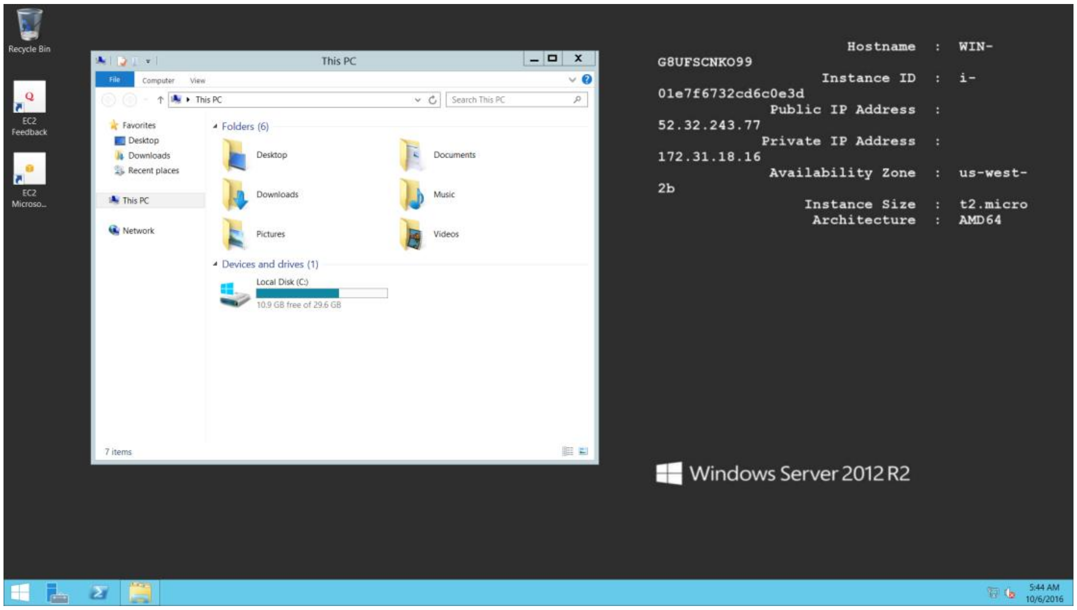
Computer: 2-39-246-77.us-west-2.compute.amazonaws.com

User name: Administrator

Saved credentials will be used to connect to this computer. You can edit or delete these credentials.

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 can see the Remote Desktop as the following:



```
                        Hostname    :   WIN-
G8UFSCNKO99
                     Instance ID    :   i-
01e7f6732cd6c0e3d
               Public IP Address    :
52.32.243.77
              Private IP Address    :
172.31.18.16
               Availability Zone    :   us-west-
2b
                   Instance Size    :   t2.micro
                    Architecture    :   AMD64
```

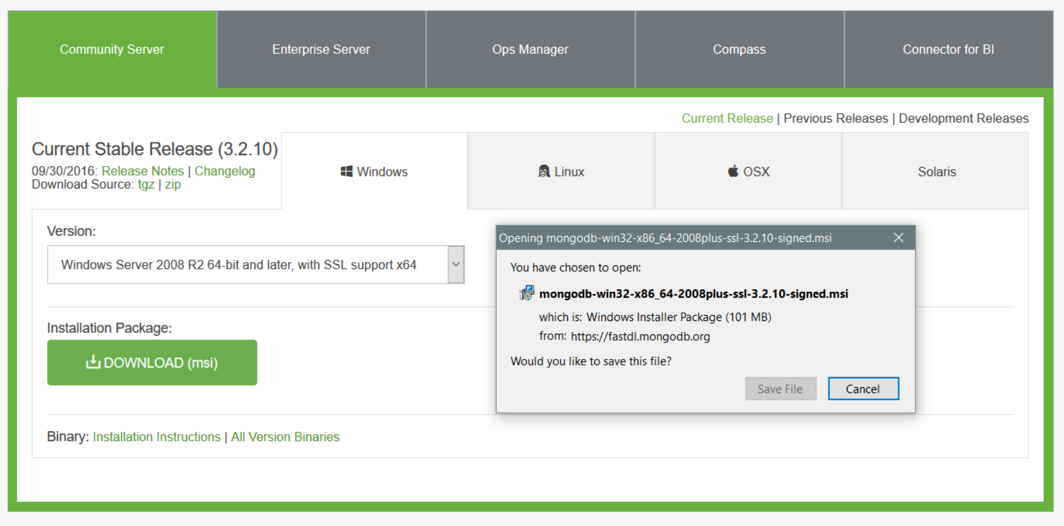
The next step to install software and a NoSQL databases that will allow us to collect tweet and store them in a database.

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.

2. Install MongoDB (NoSQL) Database

After using the Remote Desktop Login to your EC2 instance. Open the **Internet Explorer** inside your EC2 instance.

First we want 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 (from IE) and download the Windows installer for x64. You may receive a Security Alert indicating that "You are about to view pages…" You can check the box and then click on [OK]. When you access the MongoDB website, you can add these URL into the "Trusted sites" and then continue the download process.



Current Stable Release (3.2.10)
09/30/2016: Release Notes | Changelog
Download Source: tgz | zip

Version: Windows Server 2008 R2 64-bit and later, with SSL support x64

Installation Package:

Opening mongodb-win32-x86_64-2008plus-ssl-3.2.10-signed.msi

You have chosen to open:

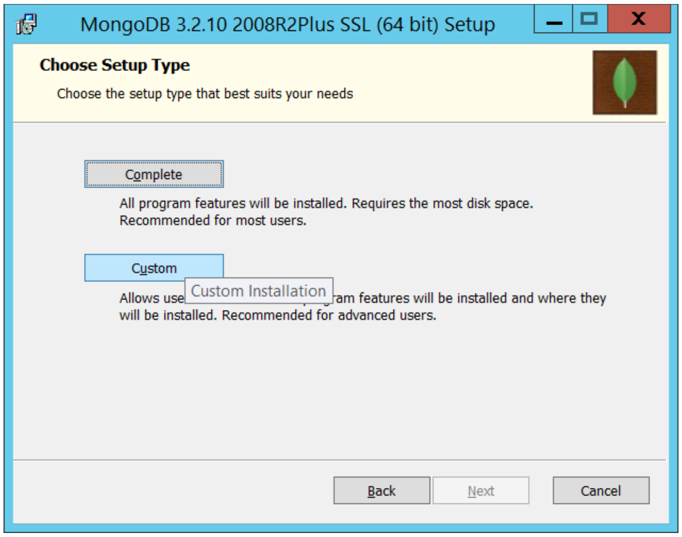
mongodb-win32-x86_64-2008plus-ssl-3.2.10-signed.msi

which is: Windows Installer Package (101 MB)
from: https://fastdl.mongodb.org

Would you like to save this file?

Binary: Installation Instructions | All Version Binaries

Download and Run the installer and when you get to the "Choose Setup Type" page, select the "**Custom**" option.



MongoDB 3.2.10 2008R2Plus SSL (64 bit) Setup

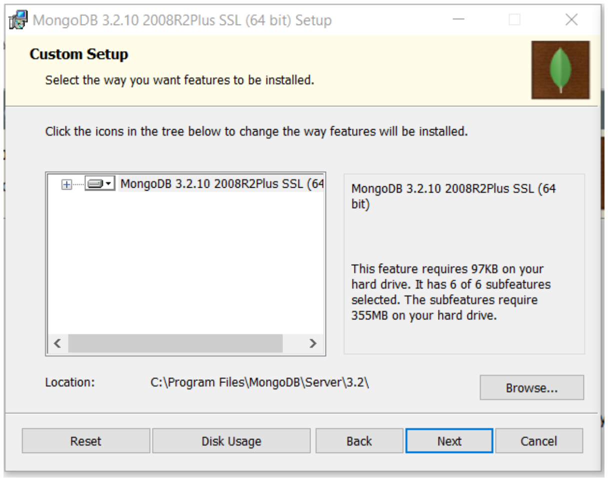
Choose Setup Type

Choose the setup type that best suits your needs

All program features will be installed. Requires the most disk space. Recommended for most users.

Allows use ram features will be installed and where they will be installed. Recommended for advanced users.

In the "Custom Setup" page, change the Location by clicking on "Browse.." and type in the Folder name "**C:\MongoDB\Server\3.2\**", 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.



MongoDB 3.2.10 2008R2Plus SSL (64 bit) Setup

Custom Setup

Select the way you want features to be installed.

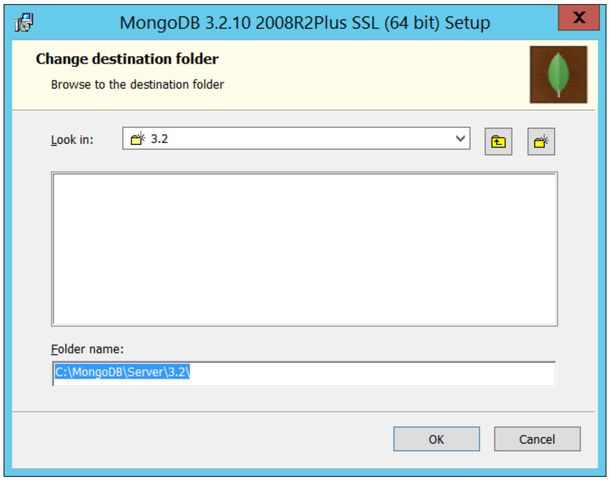
Click the icons in the tree below to change the way features will be installed.

MongoDB 3.2.10 2008R2Plus SSL (64

MongoDB 3.2.10 2008R2Plus SSL (64 bit)

This feature requires 97KB on your hard drive. It has 6 of 6 subfeatures selected. The subfeatures require 355MB on your hard drive.

Location: C:\Program Files\MongoDB\Server\3.2\



MongoDB 3.2.10 2008R2Plus SSL (64 bit) Setup

Change destination folder

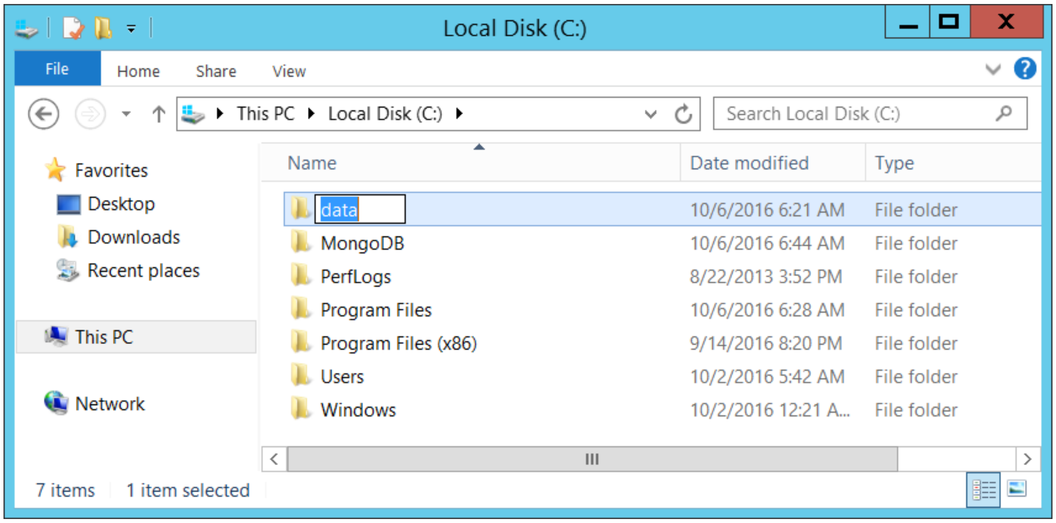
Browse to the destination folder

Look in: 3.2

Folder name: C:\MongoDB\Server\3.2\

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 folder, create a sub folder called "**db**". This is the folder in which data will be stored by MongoDB. Please make sure that the "**C:/data/db**" is at the same level of "MongoDB" folder. Otherwise, the MongoDB may not be able to generate a database.



Local Disk (C:)

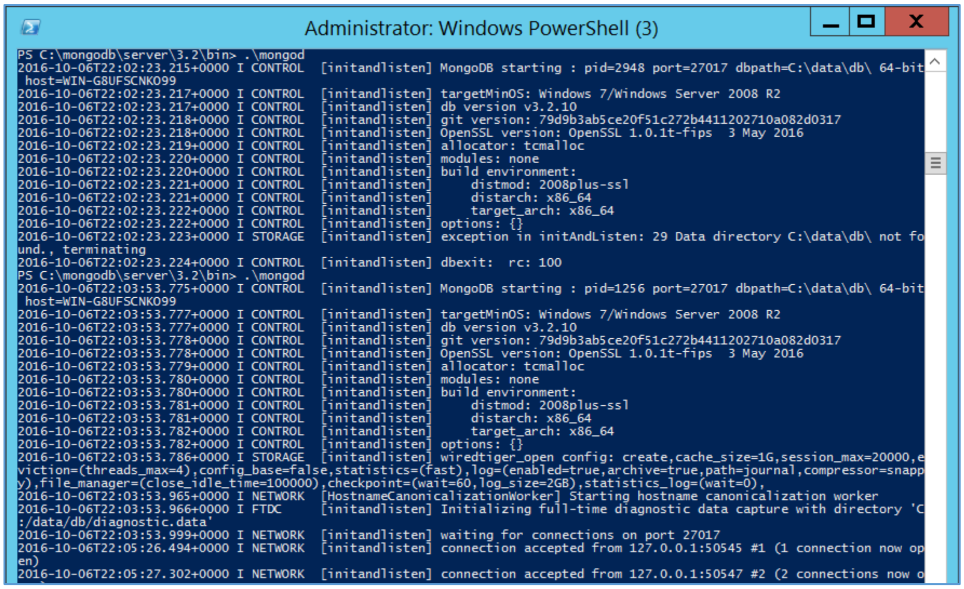
This PC ▸ Local Disk (C:) ▸

Search Local Disk (C:)

Name	Date modified	Type
data	10/6/2016 6:21 AM	File folder
MongoDB	10/6/2016 6:44 AM	File folder
PerfLogs	8/22/2013 3:52 PM	File folder
Program Files	10/6/2016 6:28 AM	File folder
Program Files (x86)	9/14/2016 8:20 PM	File folder
Users	10/2/2016 5:42 AM	File folder
Windows	10/2/2016 12:21 A...	File folder

7 items 1 item selected

Start MongoDB by opening the [**command line interface] (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\3.2\**". 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.**

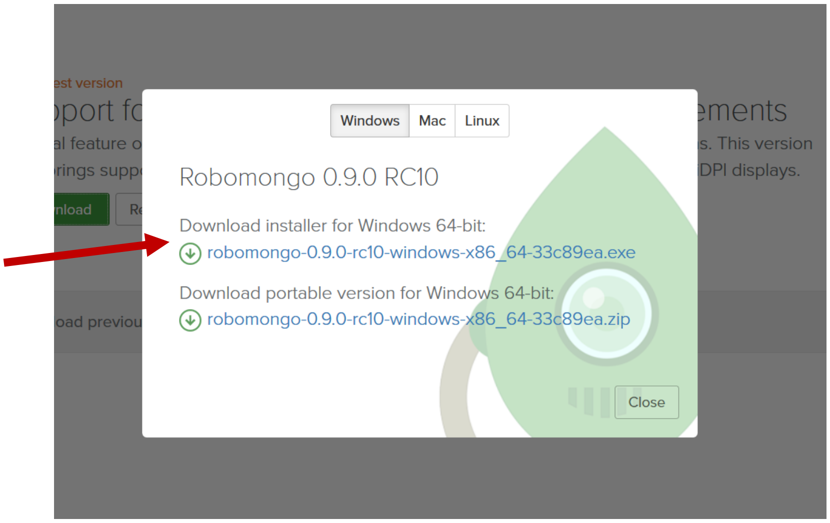


```
PS C:\mongodb\server\3.2\bin> .\mongod
2016-10-06T22:02:23.215+0000 I CONTROL  [initandlisten] MongoDB starting : pid=2948 port=27017 dbpath=C:\data\db\ 64-bit
 host=WIN-G8UFSCNK099
2016-10-06T22:02:23.217+0000 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2016-10-06T22:02:23.217+0000 I CONTROL  [initandlisten] db version v3.2.10
2016-10-06T22:02:23.218+0000 I CONTROL  [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2016-10-06T22:02:23.218+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1t-fips  3 May 2016
2016-10-06T22:02:23.219+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2016-10-06T22:02:23.220+0000 I CONTROL  [initandlisten] modules: none
2016-10-06T22:02:23.220+0000 I CONTROL  [initandlisten] build environment:
2016-10-06T22:02:23.221+0000 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2016-10-06T22:02:23.221+0000 I CONTROL  [initandlisten]     distarch: x86_64
2016-10-06T22:02:23.222+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2016-10-06T22:02:23.222+0000 I CONTROL  [initandlisten] options: {}
2016-10-06T22:02:23.223+0000 I STORAGE  [initandlisten] exception in initAndListen: 29 Data directory C:\data\db\ not fo
und., terminating
2016-10-06T22:02:23.224+0000 I CONTROL  [initandlisten] dbexit:  rc: 100
PS C:\mongodb\server\3.2\bin> .\mongod
2016-10-06T22:03:53.775+0000 I CONTROL  [initandlisten] MongoDB starting : pid=1256 port=27017 dbpath=C:\data\db\ 64-bit
 host=WIN-G8UFSCNK099
2016-10-06T22:03:53.777+0000 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2016-10-06T22:03:53.777+0000 I CONTROL  [initandlisten] db version v3.2.10
2016-10-06T22:03:53.778+0000 I CONTROL  [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2016-10-06T22:03:53.778+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1t-fips  3 May 2016
2016-10-06T22:03:53.779+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2016-10-06T22:03:53.780+0000 I CONTROL  [initandlisten] modules: none
2016-10-06T22:03:53.780+0000 I CONTROL  [initandlisten] build environment:
2016-10-06T22:03:53.781+0000 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2016-10-06T22:03:53.781+0000 I CONTROL  [initandlisten]     distarch: x86_64
2016-10-06T22:03:53.782+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2016-10-06T22:03:53.782+0000 I CONTROL  [initandlisten] options: {}
2016-10-06T22:03:53.786+0000 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=1G,session_max=20000,e
viction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-10-06T22:03:53.965+0000 I NETWORK  [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2016-10-06T22:03:53.966+0000 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory 'C
:/data/db/diagnostic.data'
2016-10-06T22:03:53.999+0000 I NETWORK  [initandlisten] waiting for connections on port 27017
2016-10-06T22:05:26.494+0000 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50545 #1 (1 connection now op
en)
2016-10-06T22:05:27.302+0000 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:50547 #2 (2 connections now o
```

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 select the 64-bit .ngo to install this tool into your EC2. (Use all default setting).

https://robomongo.org/download



Robomongo 0.9.0 RC10

Download installer for Windows 64-bit:
robomongo-0.9.0-rc10-windows-x86_64-33c89ea.exe

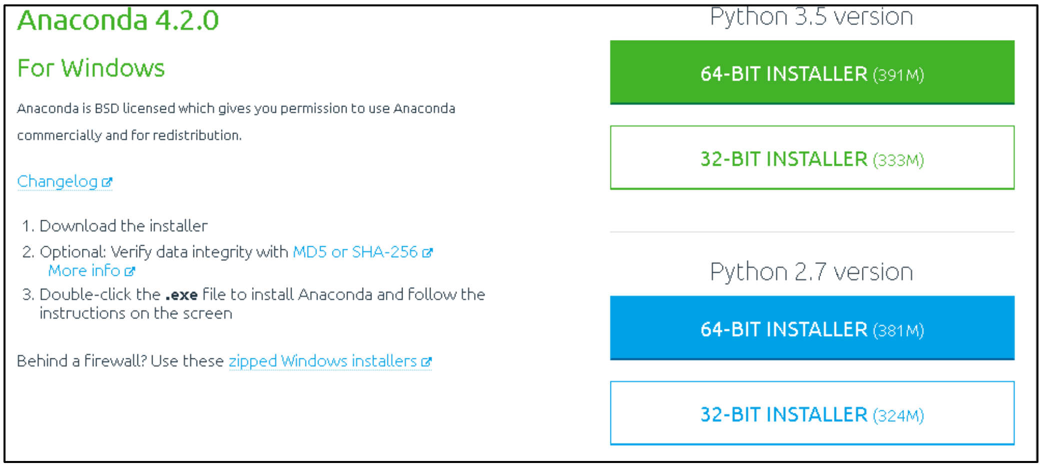
Download portable version for Windows 64-bit:
robomongo-0.9.0-rc10-windows-x86_64-33c89ea.zip

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, 64-Bit Installer**. (Use all default setting).

https://www.continuum.io/downloads



Anaconda 4.2.0

For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

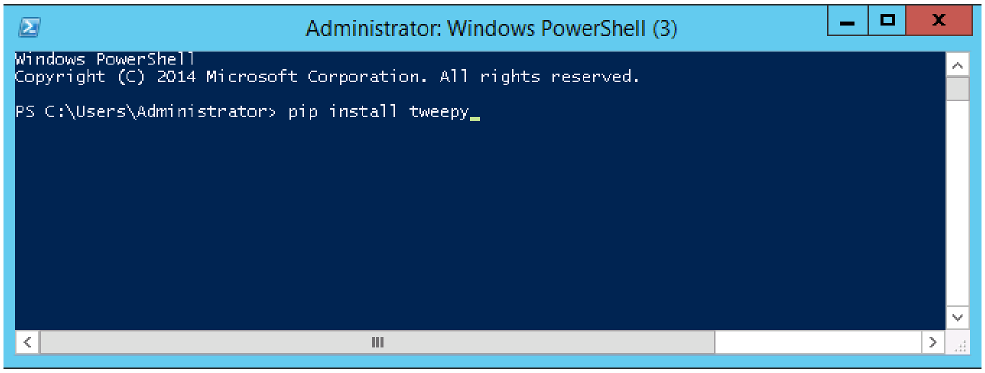
Changelog

1. Download the installer
2. Optional: Verify data integrity with MD5 or SHA-256 More info
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these zipped Windows installers

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

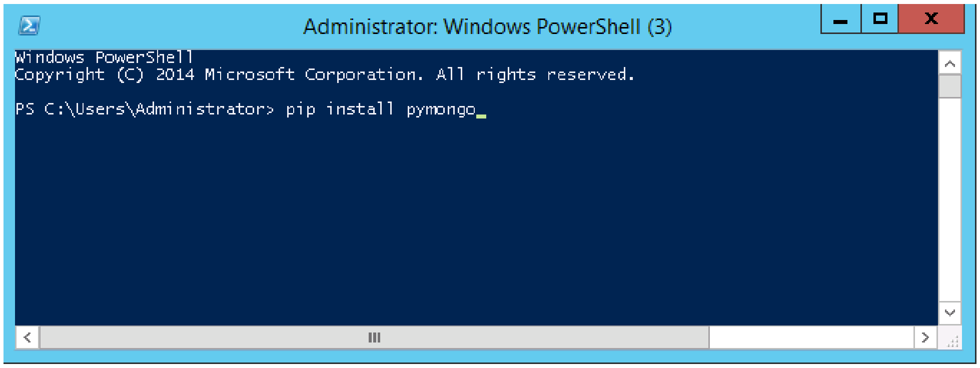
pip install tweepy



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> 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



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> pip install pymongo_
```

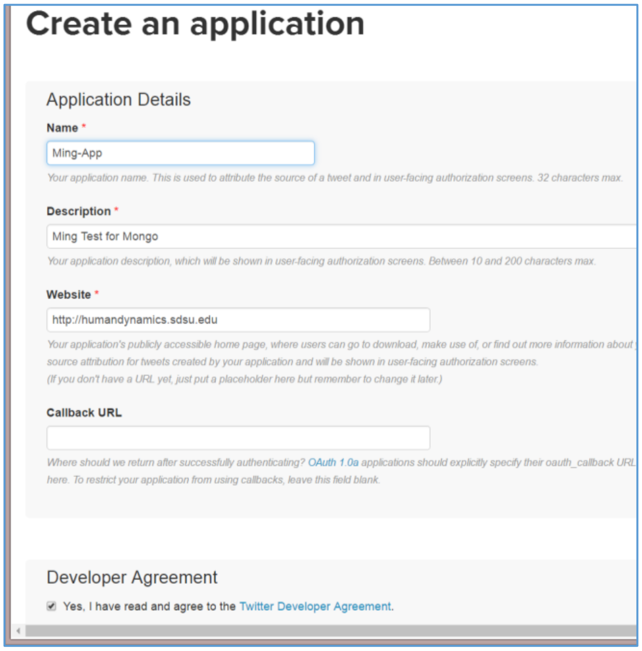
Next we will create a Python script that downloads tweets from the Twitter API and places them into MongoDB.

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.

1. Log on to Twitter Developer (**https://dev.twitter.com/**) and sign in with your Twitter account.
2. Next, use the same web browser to open this URL**:** https://apps.twitter.com/
3. Click on “**Create New App**” button.
4. Fill out the application form. (for the Website URL, If you don't have a URL yet, just put a placeholder here but remember to change it later.) Make sure to check the YES for the Developer Agreement.



Create an application

Application Details

Name *

Ming-App

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Ming Test for Mongo

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://humandynamics.sdsu.edu

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about
source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

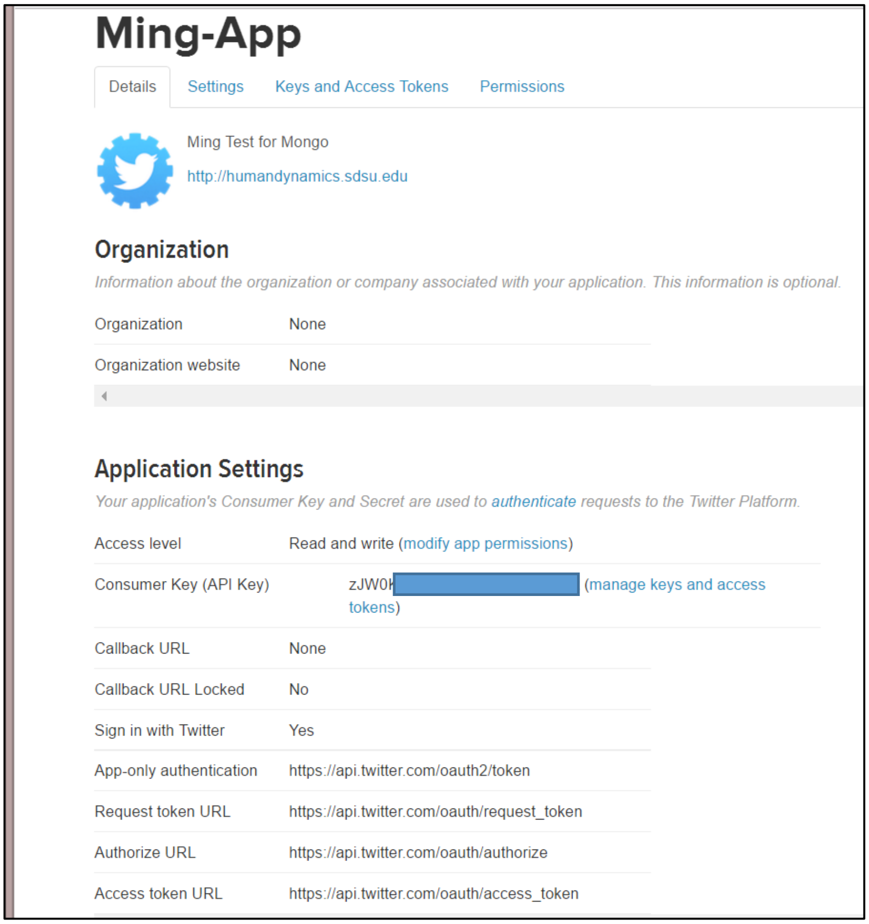
Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL
here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

☑ Yes, I have read and agree to the Twitter Developer Agreement.

5. Click on "**Create your Twitter application**" button. (Note: you need to add your mobile phone to your Twitter profile before creating an application.)

After that, you will see the following page:



Ming-App

Ming Test for Mongo

http://humandynamics.sdsu.edu

Organization

Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
Consumer Key (API Key)	zJW0[redacted] (manage keys and access tokens)
Callback URL	None
Callback URL Locked	No
Sign in with Twitter	Yes
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

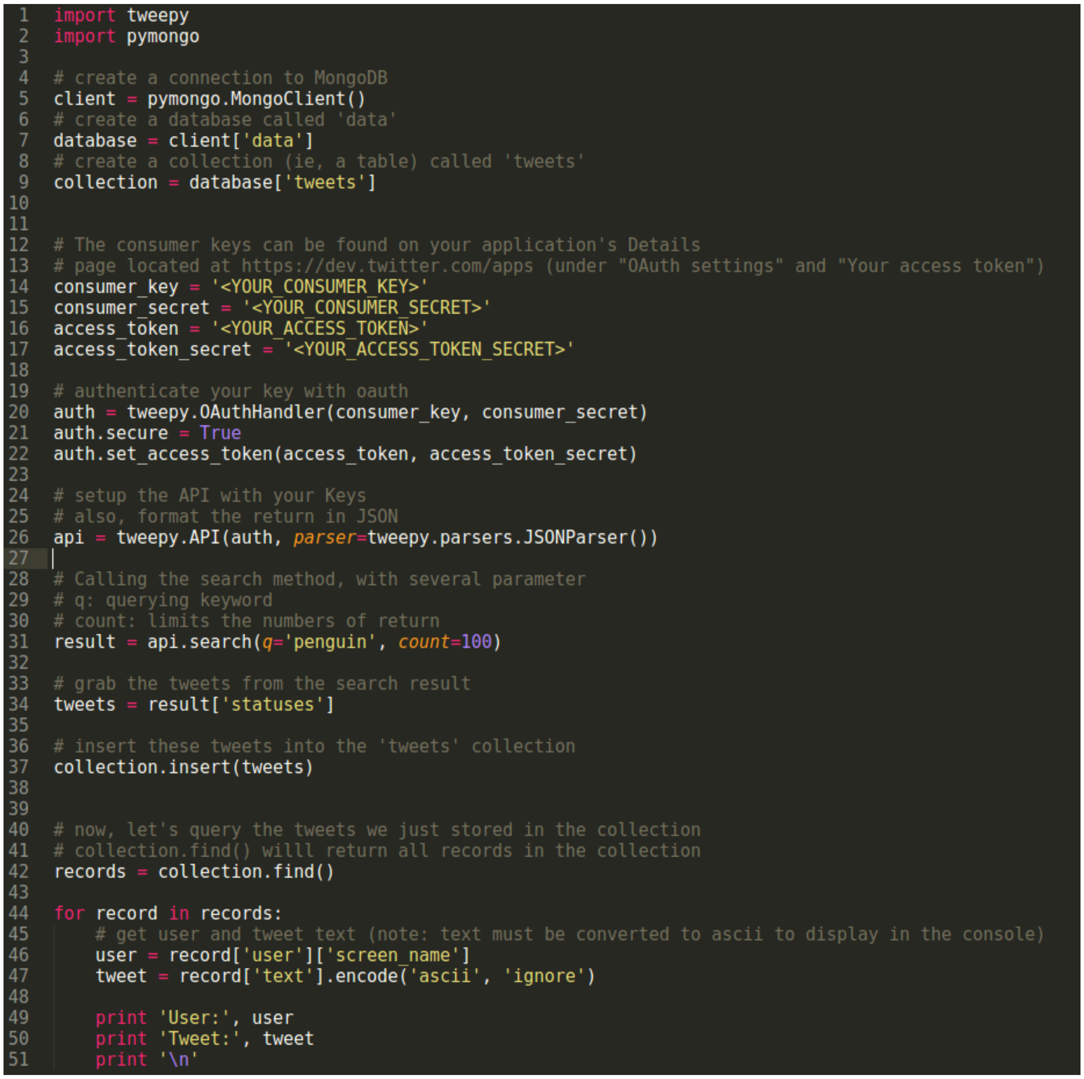
Click on "**Keys and Access Tokens**" Tab on the upper menu (under your application name). Save the Consumer key (API Key) and Consumer secret (API Secret) into your local file (in Word or text file). In the bottom of the page, click on "**Create my access token**". Copy and Save your Access Token and Access Token Secret into your local 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/)** or **ATOM** (https://atom.io/).

Create the following program 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.

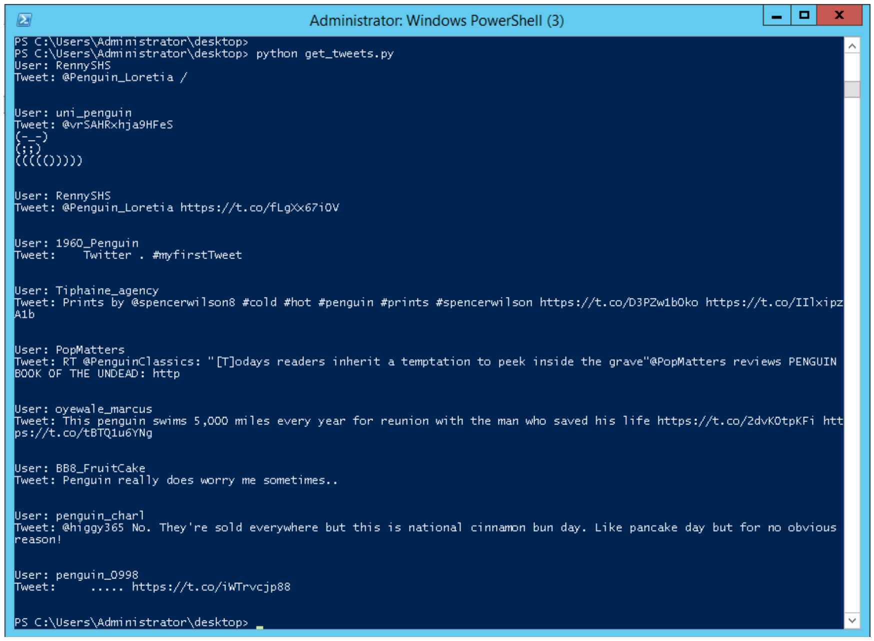


```
import tweepy
import pymongo

# 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'
```

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 command to download 100 tweets related to your keyword and store them in MongoDB:

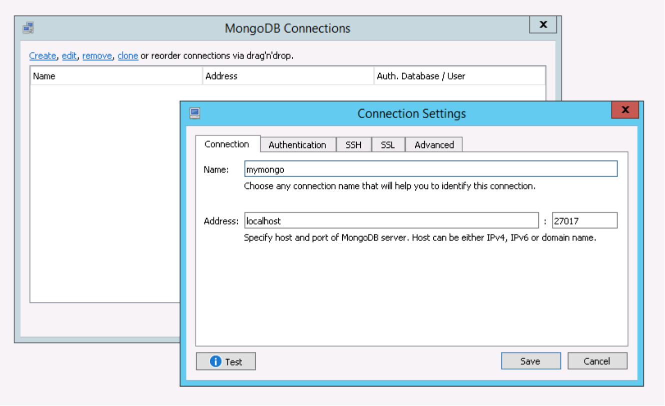
```
python C:\Scripts\get_tweets-yourname.py
```



```
PS C:\Users\Administrator\desktop>
PS C:\Users\Administrator\desktop> python get_tweets.py
User: RennySHS
Tweet: @Penguin_Loretia /


User: uni_penguin
Tweet: @vrSAHRxhja9HFeS
(-_-)
(;;)
((((()))))


User: RennySHS
Tweet: @Penguin_Loretia https://t.co/fLgXx67iOV


User: 1960_Penguin
Tweet:    Twitter . #myfirstTweet


User: Tiphaine_agency
Tweet: Prints by @spencerwilson8 #cold #hot #penguin #prints #spencerwilson https://t.co/D3PZw1bOko https://t.co/IIlxipz
A1b


User: PopMatters
Tweet: RT @PenguinClassics: "[T]odays readers inherit a temptation to peek inside the grave"@PopMatters reviews PENGUIN
BOOK OF THE UNDEAD: http


User: oyewale_marcus
Tweet: This penguin swims 5,000 miles every year for reunion with the man who saved his life https://t.co/2dvKOtpKFi htt
ps://t.co/tBTQ1u6YNg


User: BB8_FruitCake
Tweet: Penguin really does worry me sometimes..


User: penguin_charl
Tweet: @higgy365 No. They're sold everywhere but this is national cinnamon bun day. Like pancake day but for no obvious
reason!


User: penguin_0998
Tweet:     ..... https://t.co/iWTrvcjp88


PS C:\Users\Administrator\desktop> _
```

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 RoboMongo. 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.



Create, edit, remove, clone or reorder connections via drag'n'drop.

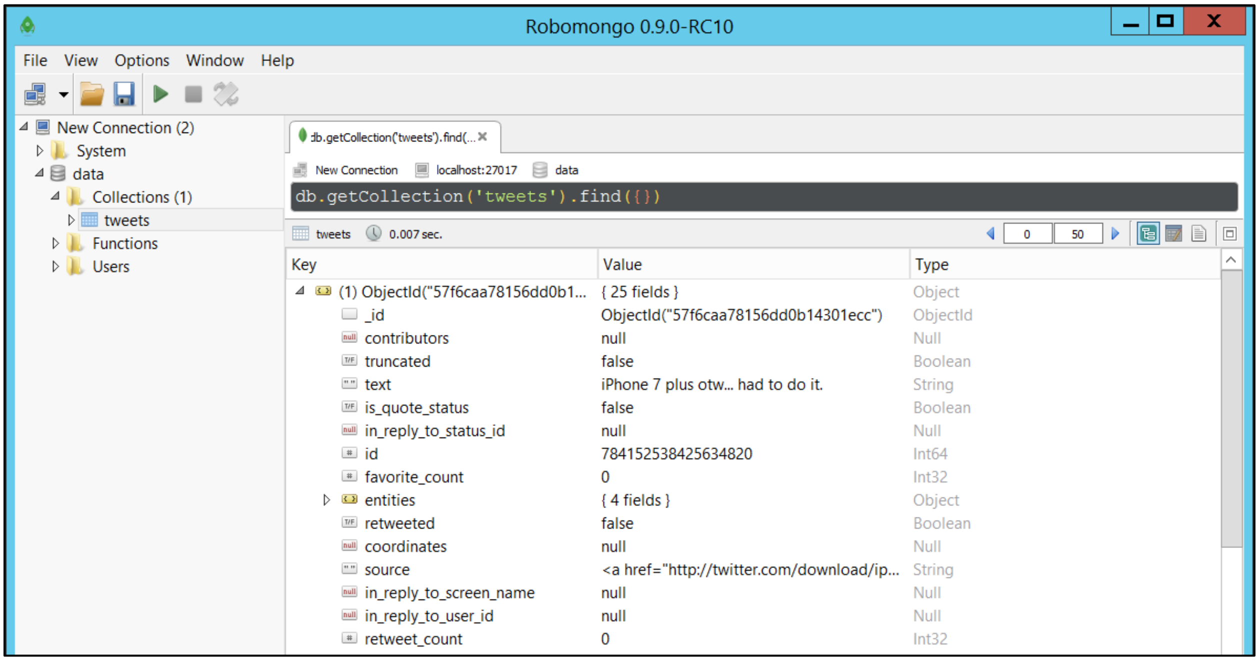
Name: mymongo

Choose any connection name that will help you to identify this connection.

Address: localhost : 27017

Specify host and port of MongoDB server. Host can be either IPv4, IPv6 or domain name.

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



```
db.getCollection('tweets').find({})
```

Key	Value	Type
(1) ObjectId("57f6caa78156dd0b1...	{ 25 fields }	Object
_id	ObjectId("57f6caa78156dd0b14301ecc")	ObjectId
contributors	null	Null
truncated	false	Boolean
text	iPhone 7 plus otw... had to do it.	String
is_quote_status	false	Boolean
in_reply_to_status_id	null	Null
id	784152538425634820	Int64
favorite_count	0	Int32
entities	{ 4 fields }	Object
retweeted	false	Boolean
coordinates	null	Null
source	<a href="http://twitter.com/download/ip...	String
in_reply_to_screen_name	null	Null
in_reply_to_user_id	null	Null
retweet_count	0	Int32

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. What are the differences between traditional SQL databases and NoSQL databases?
3. 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. 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.