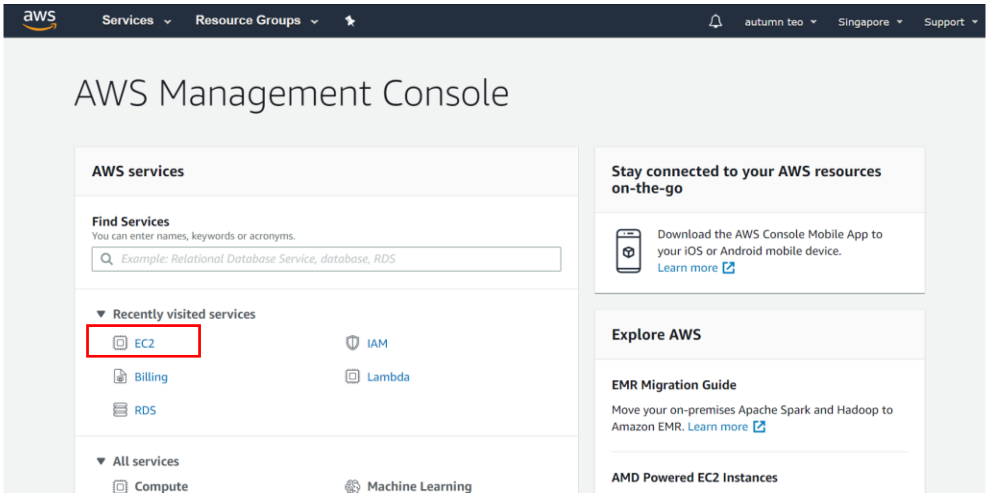
Setting up your EC2 Server

1. Go to Amazon Dashboard and select EC2



AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.
Example: Relational Database Service, database, RDS

▼ Recently visited services

EC2 | IAM
Billing | Lambda
RDS

▼ All services

Compute | Machine Learning

Stay connected to your AWS resources on-the-go

Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

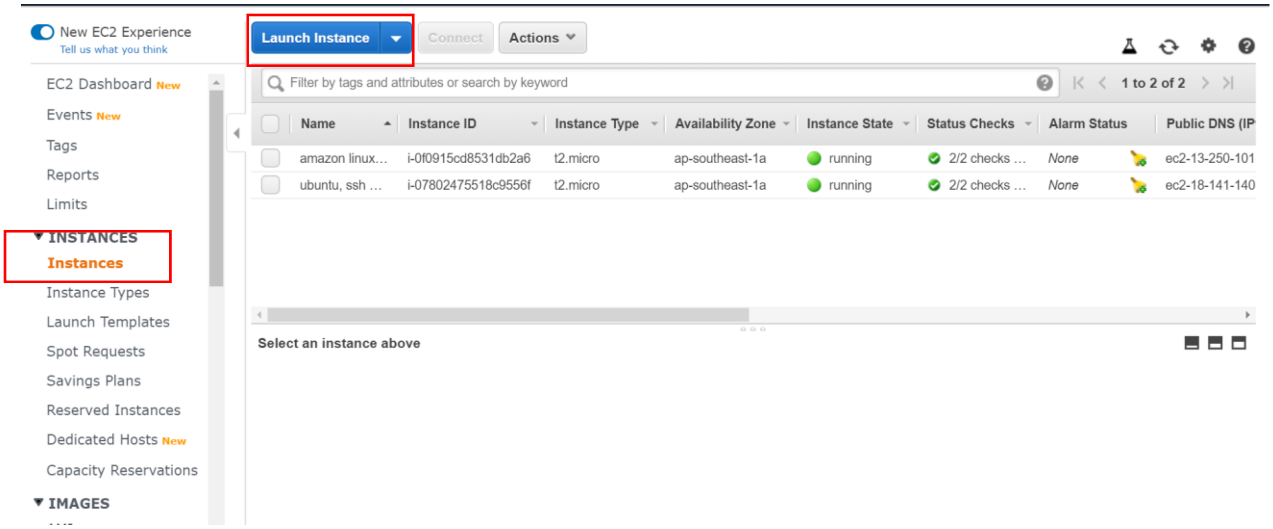
Explore AWS

EMR Migration Guide

Move your on-premises Apache Spark and Hadoop to Amazon EMR. Learn more

AMD Powered EC2 Instances

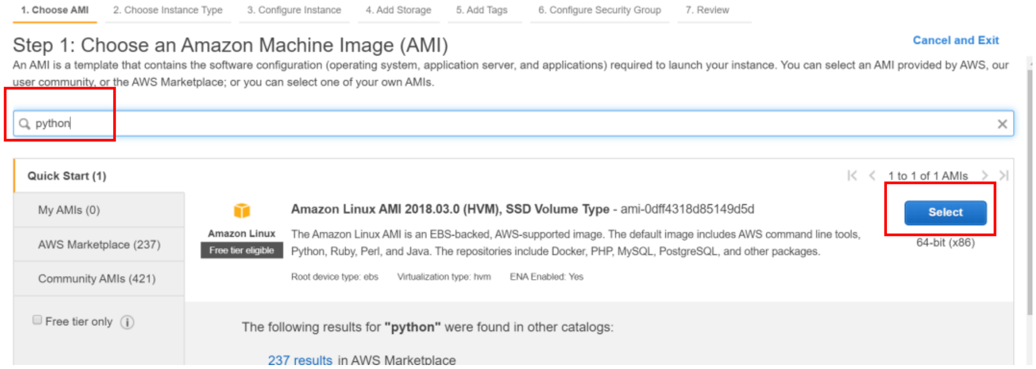
2. Select “Instances” and then “Launch Instance”



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP
□	amazon linux...	i-0f0915cd8531db2a6	t2.micro	ap-southeast-1a	● running	✓ 2/2 checks ...	*None*	ec2-13-250-101
□	ubuntu, ssh ...	i-07802475518c9556f	t2.micro	ap-southeast-1a	● running	✓ 2/2 checks ...	*None*	ec2-18-141-140

Select an instance above

3. Search for Python and Select this:



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

python

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0dff4318d85149d5d

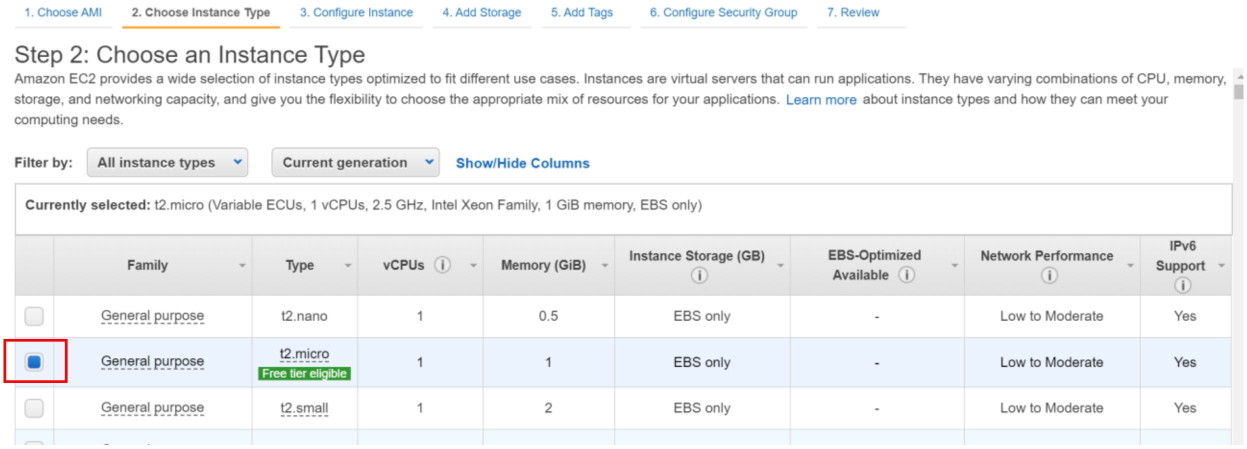
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

The following results for **"python"** were found in other catalogs:

237 results in AWS Marketplace

4. Select t2.micro (the free tier eligible) and keep selecting Next until you reach the page on Security Config



Step 2: Choose an Instance Type

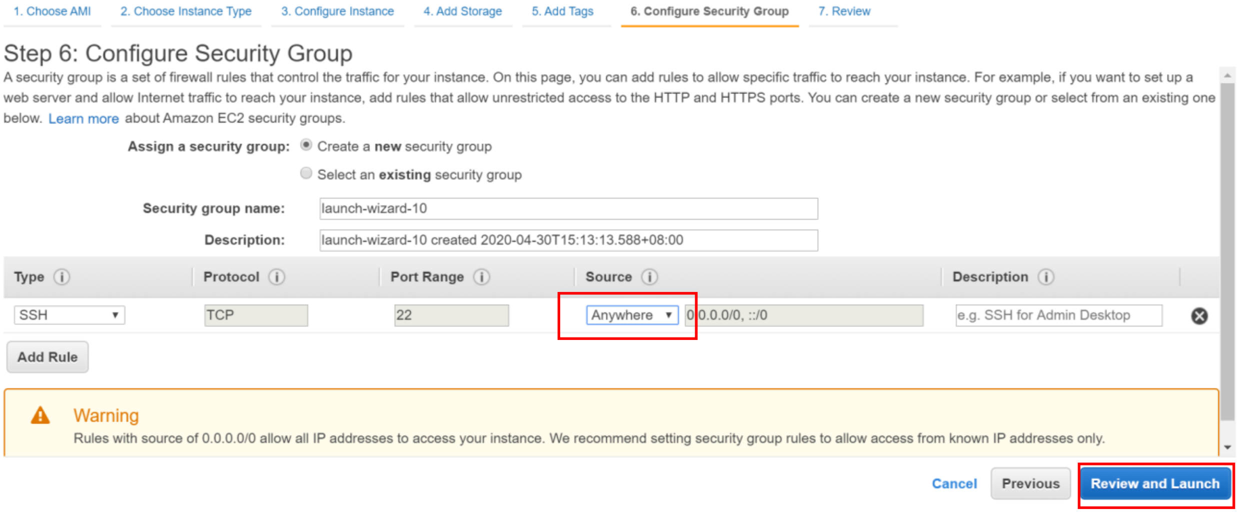
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources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

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

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	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes

5. Change the Source to "Anywhere" and select "Review and Launch"



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

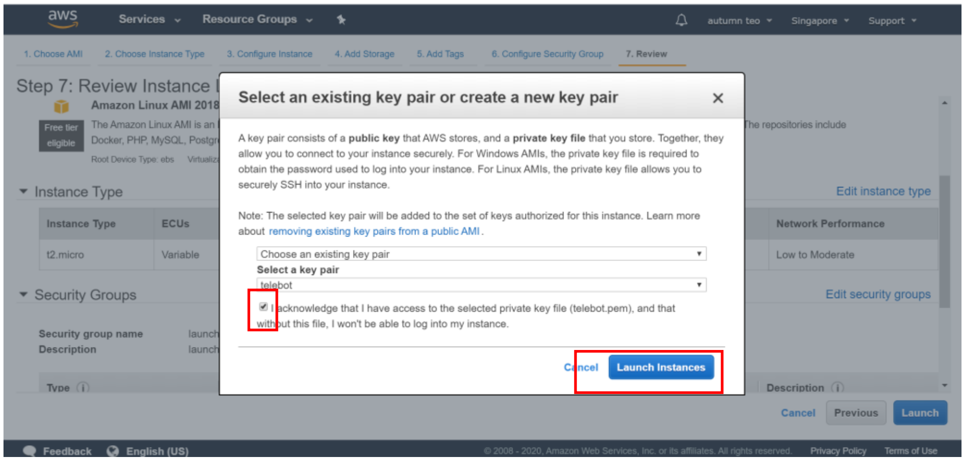
Security group name: launch-wizard-10

Description: launch-wizard-10 created 2020-04-30T15:13:13.588+08:00

Type	Protocol	Port Range	Source		Description
SSH ▾	TCP	22	Anywhere ▾	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

6. You will be prompted to create or use an existing key pair. Create a new .pem key if you do not have an existing key pair and download and save it. Then click "Launch Instance".



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.

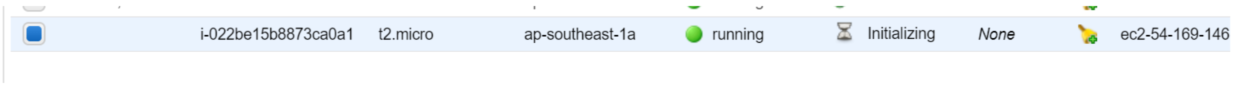
Choose an existing key pair ▾

Select a key pair

telebot ▾

☑ I acknowledge that I have access to the selected private key file (telebot.pem), and that without this file, I won't be able to log into my instance.

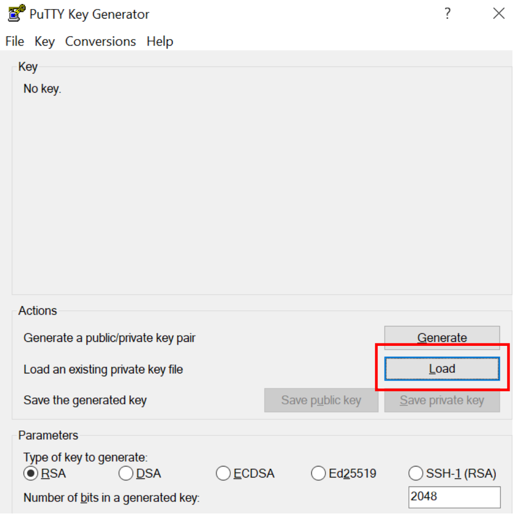
7. Go back to your “Instances” and you should see your EC2 instance initializing. Wait a while for it to run



i-022be15b8873ca0a1 t2.micro ap-southeast-1a ● running ⌛ Initializing *None* ec2-54-169-146

Creating a SSH Tunnel to your EC2 Server (To run programs etc)

8. Download/ Open Puttygen (https://www.ssh.com/ssh/putty/windows/puttygen) to convert your .pem key to .ppk. And load your .pem key (from where you saved your .pem key)



Key

No key.

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

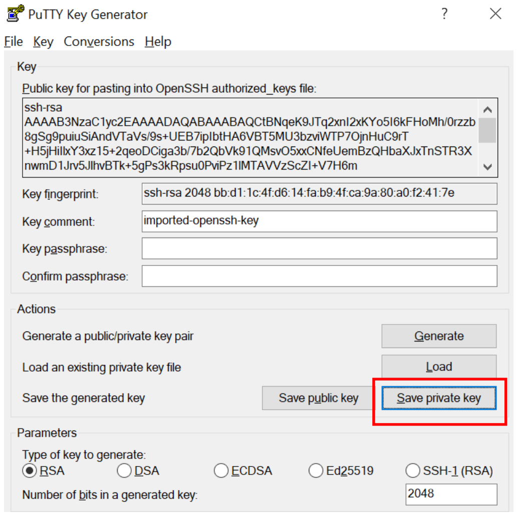
Parameters

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

9. You should see this screen. Select “Save Private Key”. You will be prompted for a passphrase, you may create it if you wish to or not. Name your .ppk key and save it. Close Puttygen.



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCtBNqeK9JTq2xnI2xKYo5I6kFHoMh/0rzzb
8gSg9puiuSiAndVTaVs/9s+UEB7ipIbtHA6VBT5MU3bzviWTP7OjnHuC9rT
+H5jHiIlxY3xz15+2qeoDCiga3b/7b2QbVk91QMsvO5xxCNfeUemBzQHbaXJxTnSTR3X
nwmD1Jrv5JlhvBTk+5gPs3kRpsu0PviPz1IMTAVVzScZI+V7H6m
```

Key fingerprint: ssh-rsa 2048 bb:d1:1c:4f:d6:14:fa:b9:4f:ca:9a:80:a0:f2:41:7e

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

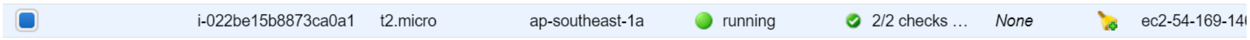
Parameters

Type of key to generate:

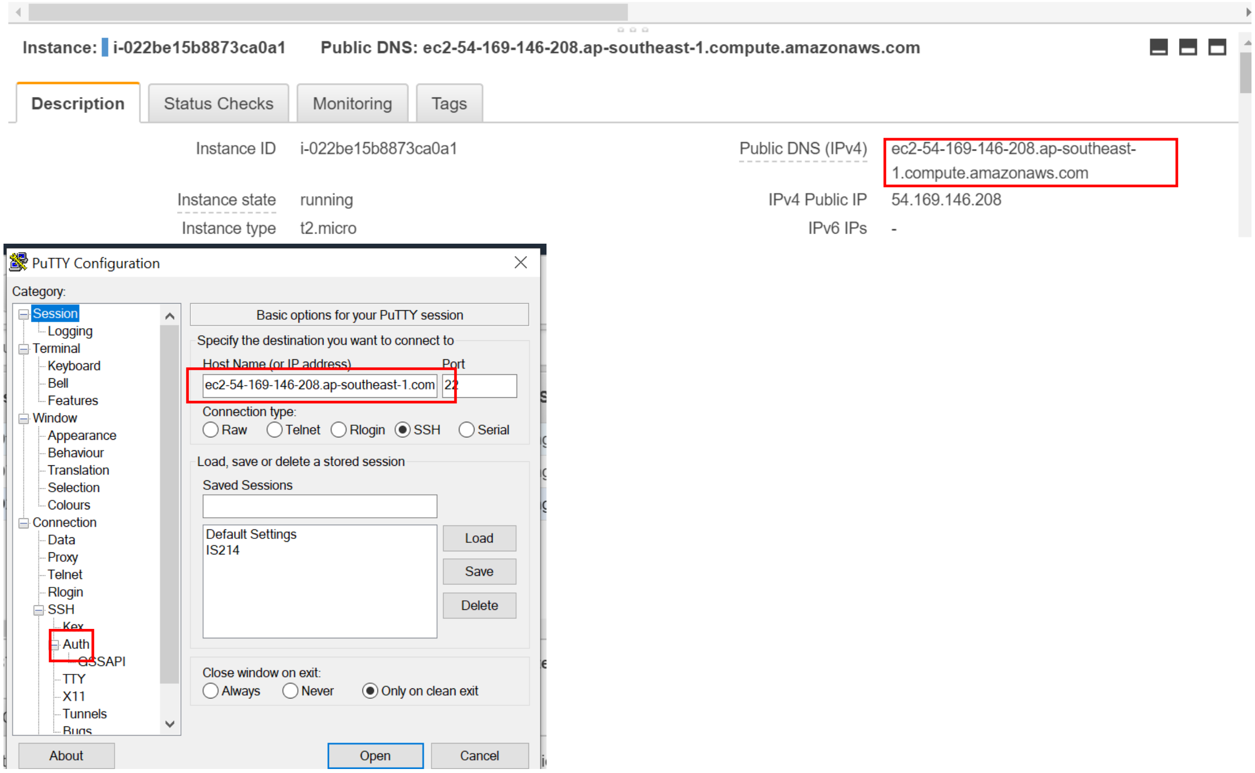
● RSA ○ DSA ○ ECDSA ○ Ed25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

10. Download/ open Putty (https://www.putty.org/). Copy the Public DNS from AWS Instance and enter it into the “Host Name”. Then select “Auth”



■	i-022be15b8873ca0a1	t2.micro	ap-southeast-1a	● running	✓ 2/2 checks ...	None	ec2-54-169-14



Instance: i-022be15b8873ca0a1 Public DNS: ec2-54-169-146-208.ap-southeast-1.compute.amazonaws.com

Instance ID i-022be15b8873ca0a1

Instance state running

Instance type t2.micro

Public DNS (IPv4) ec2-54-169-146-208.ap-southeast-1.compute.amazonaws.com

IPv4 Public IP 54.169.146.208

IPv6 IPs -

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ec2-54-169-146-208.ap-southeast-1.com

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

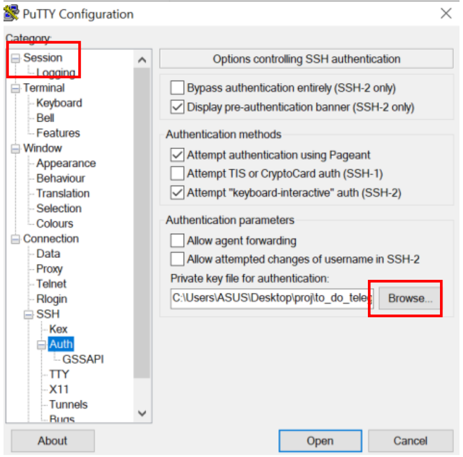
Load, save or delete a stored session

Saved Sessions

Default Settings
IS214

Close window on exit: ○ Always ○ Never ● Only on clean exit

11. Click on “Browsed” and load your saved .ppk file. Then click on “Session”.



PuTTY Configuration

Options controlling SSH authentication

□ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

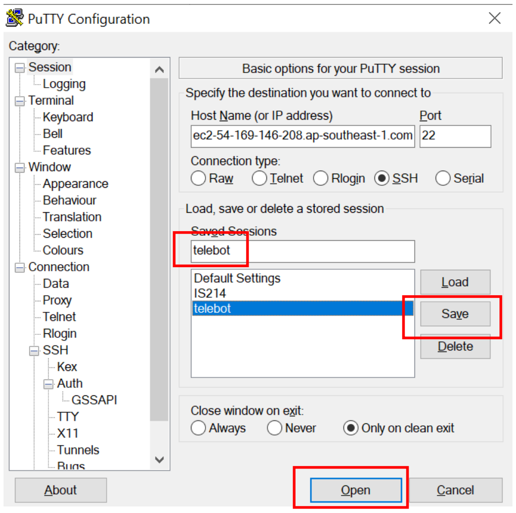
☑ Attempt authentication using Pageant
□ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

□ Allow agent forwarding
□ Allow attempted changes of username in SSH-2

Private key file for authentication:
C:\Users\ASUS\Desktop\proj\to_do_tele

12. Then name your session and click on “save”. Then select the session that you just named and select “open”.



Specify the destination you want to connect to

Host Name (or IP address): ec2-54-169-146-208.ap-southeast-1.com

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

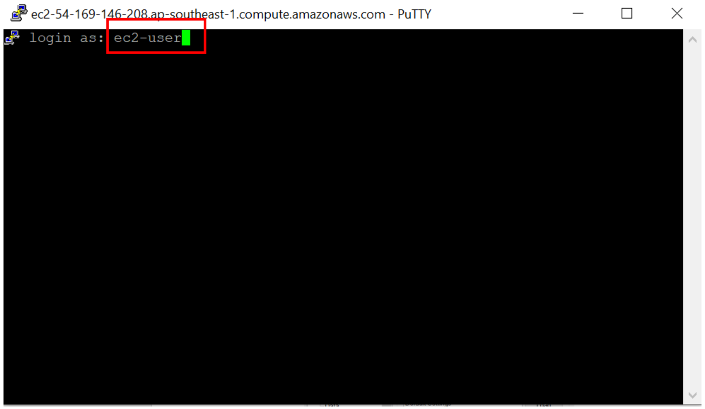
Load, save or delete a stored session

Saved Sessions: telebot

Default Settings
IS214
telebot

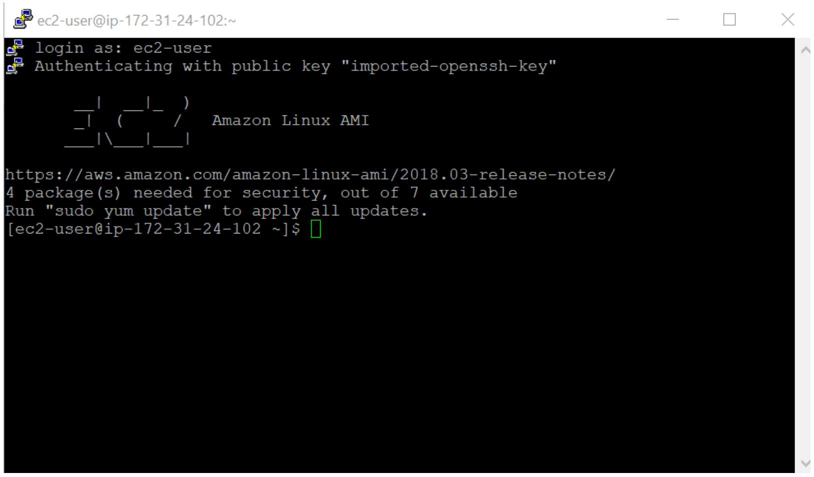
Close window on exit: ○ Always ○ Never ● Only on clean exit

13. You will be prompted with a warning, just select “Yes”. You will then be faced with a prompt. Type “ec2-user” and press enter. You will then see the login message. Minimize it, we will need it later



ec2-54-169-146-208.ap-southeast-1.compute.amazonaws.com - PuTTY

```
login as: ec2-user
```



ec2-user@ip-172-31-24-102:~

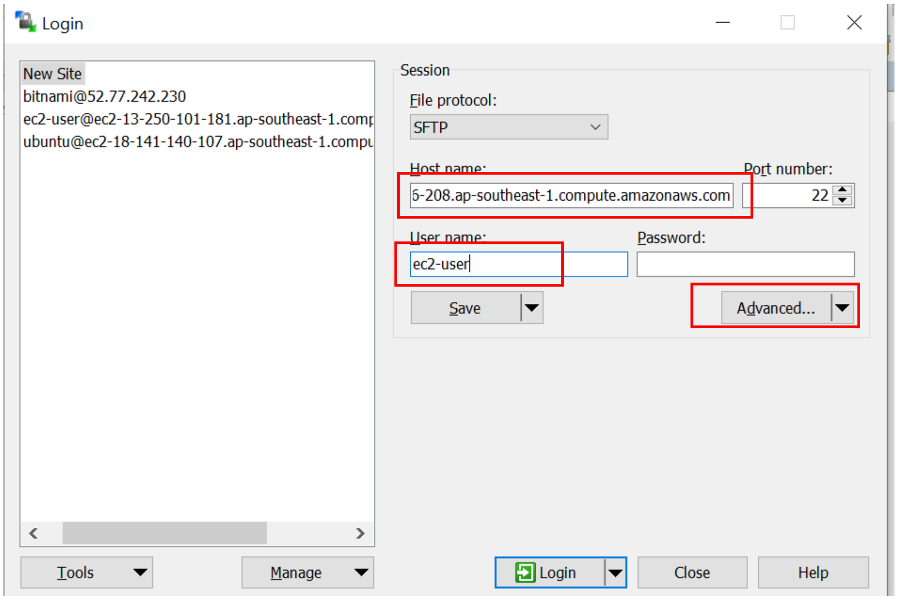
```
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-24-102 ~]$
```

Transfer your script to the EC2 Server

14. Open/ Download Winscp (https://winscp.net/eng/download.php). Enter the Public DNS of your Server (same as the one used in Step 10) in “Host Name”. Enter “ec2-user” in “User name”. Then select “Advanced…”.



Login

New Site
bitnami@52.77.242.230
ec2-user@ec2-13-250-101-181.ap-southeast-1.comp
ubuntu@ec2-18-141-140-107.ap-southeast-1.compu

Session

File protocol: SFTP

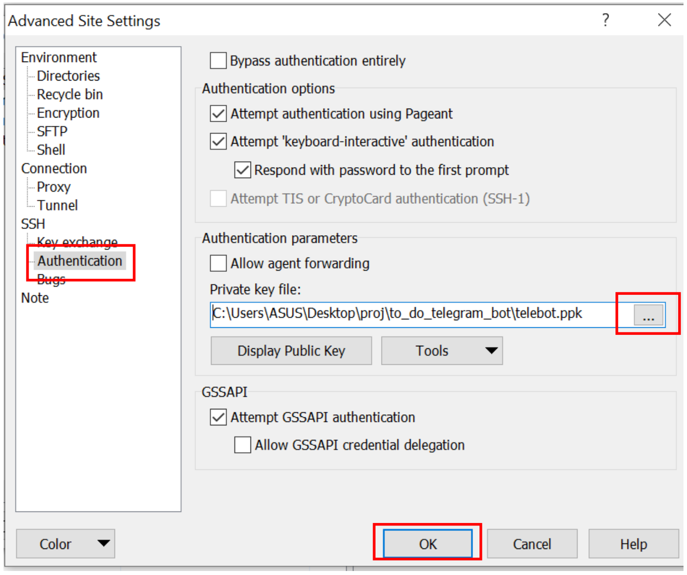
Host name: 5-208.ap-southeast-1.compute.amazonaws.com

Port number: 22

User name: ec2-user

Password:

15. Select “Authentication” and select the “…” to load your .ppk file. Then click on “OK”



Advanced Site Settings

□ Bypass authentication entirely

Authentication options

☑ Attempt authentication using Pageant

☑ Attempt 'keyboard-interactive' authentication

☑ Respond with password to the first prompt

□ Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

□ Allow agent forwarding

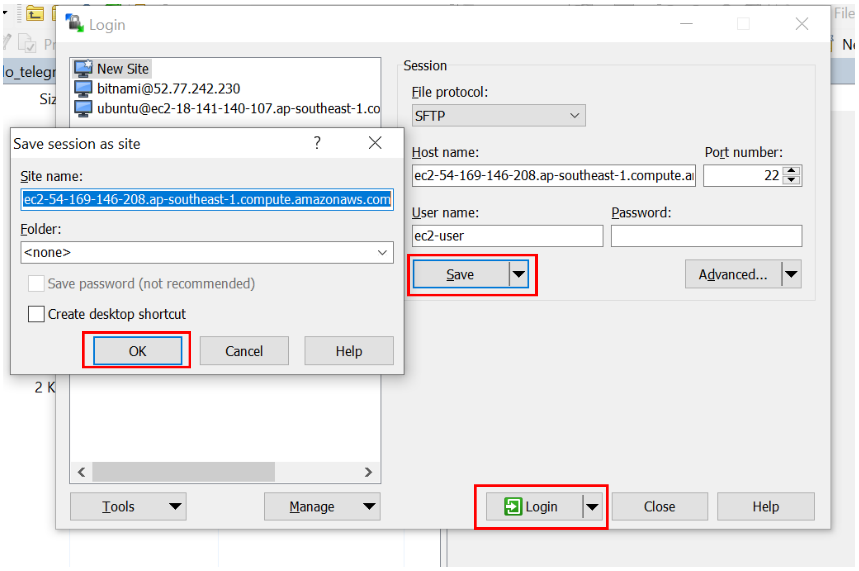
Private key file: C:\Users\ASUS\Desktop\proj\to_do_telegram_bot\telebot.ppk

GSSAPI

☑ Attempt GSSAPI authentication

□ Allow GSSAPI credential delegation

16. Click on “save” then select “ok” of the prompt. Then select “Login”.



File protocol: SFTP

Host name: ec2-54-169-146-208.ap-southeast-1.compute.ai

Port number: 22

User name: ec2-user

Password:

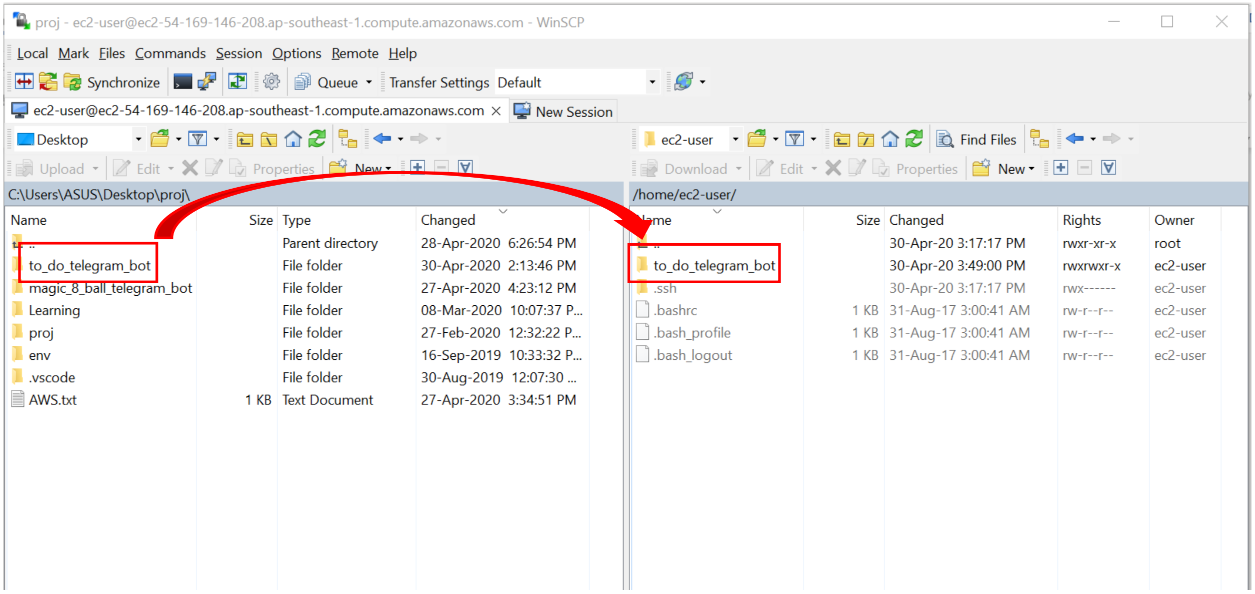
Site name: ec2-54-169-146-208.ap-southeast-1.compute.amazonaws.com

Folder: <none>

□ Save password (not recommended)

□ Create desktop shortcut

17. A prompt will open, select “Yes”. You should see something similar to the image below and you are now be able to transfer your files. The left side of your screen is your own machine and the right side is the files on your ec2 Server.
Navigate to your folder (on the left hand side of the screen) containing your script and drag it to the right side of the screen to transfer it.



C:\Users\ASUS\Desktop\proj\

Name	Size	Type	Changed
..		Parent directory	28-Apr-2020 6:26:54 PM
to_do_telegram_bot		File folder	30-Apr-2020 2:13:46 PM
magic_8_ball_telegram_bot		File folder	27-Apr-2020 4:23:12 PM
Learning		File folder	08-Mar-2020 10:07:37 P...
proj		File folder	27-Feb-2020 12:32:22 P...
env		File folder	16-Sep-2019 10:33:32 P...
.vscode		File folder	30-Aug-2019 12:07:30 ...
AWS.txt	1 KB	Text Document	27-Apr-2020 3:34:51 PM

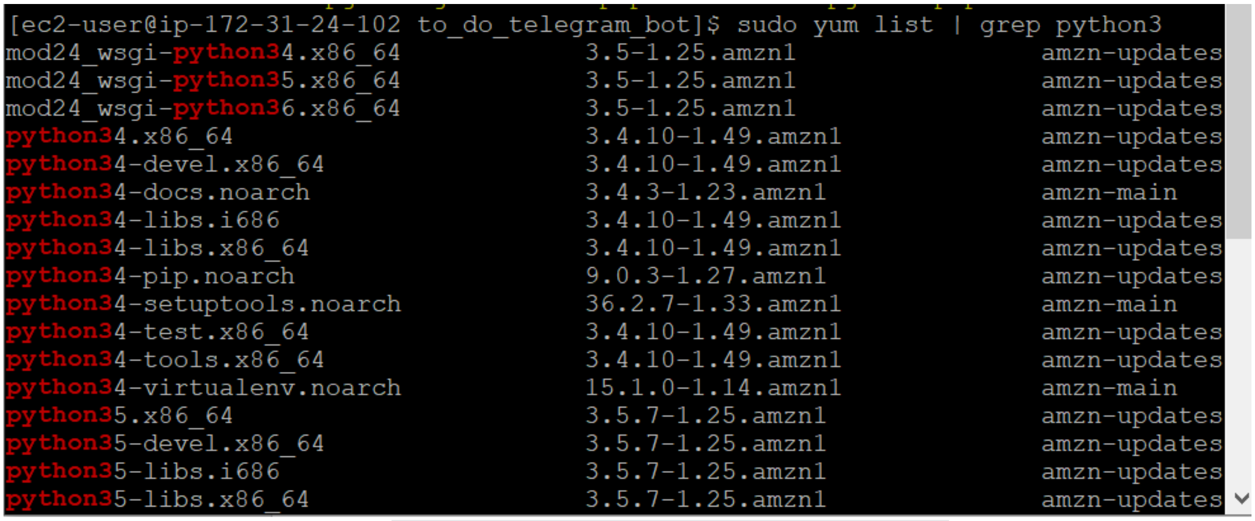
/home/ec2-user/

Name	Size	Changed	Rights	Owner
..		30-Apr-20 3:17:17 PM	rwxr-xr-x	root
to_do_telegram_bot		30-Apr-20 3:49:00 PM	rwxrwxr-x	ec2-user
.ssh		30-Apr-20 3:17:17 PM	rwx------	ec2-user
.bashrc	1 KB	31-Aug-17 3:00:41 AM	rw-r--r--	ec2-user
.bash_profile	1 KB	31-Aug-17 3:00:41 AM	rw-r--r--	ec2-user
.bash_logout	1 KB	31-Aug-17 3:00:41 AM	rw-r--r--	ec2-user

Running the Telegram Bot on the Server

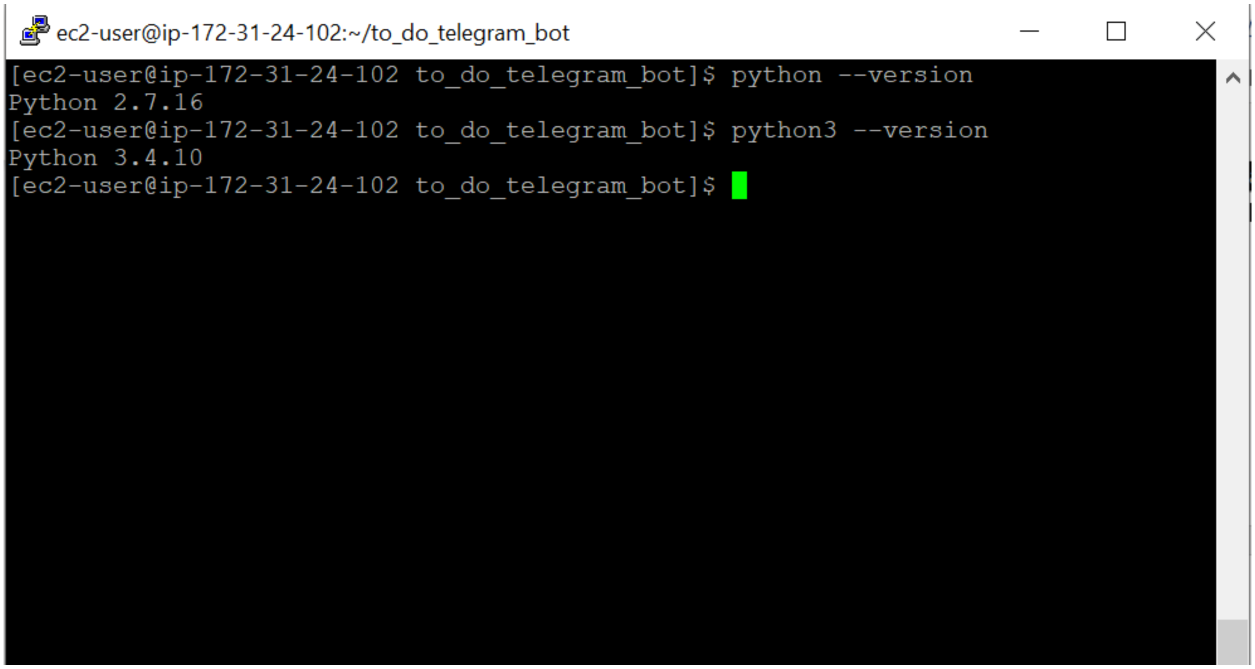
18. Open your Putty application and change folder to the folder containing your script by typing “cd to_do_telegram_bot”

19. The current default python version is 2.7. As the script is coded in python3, we need to change the default version to python3. Type `sudo yum list | grep python3` to check the python version available.



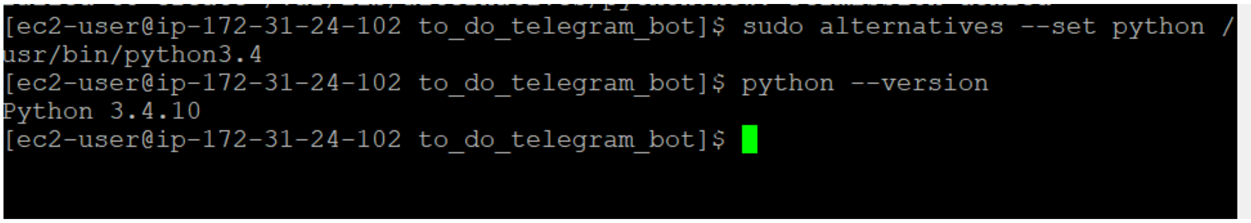
```
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$ sudo yum list | grep python3
mod24_wsgi-python34.x86_64            3.5-1.25.amzn1              amzn-updates
mod24_wsgi-python35.x86_64            3.5-1.25.amzn1              amzn-updates
mod24_wsgi-python36.x86_64            3.5-1.25.amzn1              amzn-updates
python34.x86_64                       3.4.10-1.49.amzn1           amzn-updates
python34-devel.x86_64                 3.4.10-1.49.amzn1           amzn-updates
python34-docs.noarch                  3.4.3-1.23.amzn1            amzn-main
python34-libs.i686                    3.4.10-1.49.amzn1           amzn-updates
python34-libs.x86_64                  3.4.10-1.49.amzn1           amzn-updates
python34-pip.noarch                   9.0.3-1.27.amzn1            amzn-updates
python34-setuptools.noarch            36.2.7-1.33.amzn1           amzn-main
python34-test.x86_64                  3.4.10-1.49.amzn1           amzn-updates
python34-tools.x86_64                 3.4.10-1.49.amzn1           amzn-updates
python34-virtualenv.noarch            15.1.0-1.14.amzn1           amzn-main
python35.x86_64                       3.5.7-1.25.amzn1            amzn-updates
python35-devel.x86_64                 3.5.7-1.25.amzn1            amzn-updates
python35-libs.i686                    3.5.7-1.25.amzn1            amzn-updates
python35-libs.x86_64                  3.5.7-1.25.amzn1            amzn-updates
```

20. To change python version, type `sudo yum install python34 python34-pip` to install python 3.4. then enter “y” when prompted to download the packages. But the default python version is still 2.7.



```
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$ python --version
Python 2.7.16
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$ python3 --version
Python 3.4.10
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$
```

21. To make python 3 the default version, enter `sudo alternatives --set python /usr/bin/python3.4`
(Note: Using alias python = python3 will stop working once you log out/close of the current session)



```
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$ sudo alternatives --set python /
usr/bin/python3.4
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$ python --version
Python 3.4.10
[ec2-user@ip-172-31-24-102 to_do_telegram_bot]$
```

22. To ensure the your script runs as long as the server is up, enter `nohup python app.py &`
23. Before running the script, we need to first download the relevant packages, enter `sudo pip install requests`
24. To run the script, enter `python app.py`