

HOW TO SET UP A EC2 INSTANCE ON AMAZON AWS

Rubens Zimbres

<http://github.com/RubensZimbres>

Go to your AWS Management Console, choose Services / EC2 / Instances / Launch Instance.

Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)

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.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

 **Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type** - ami-caaf84af

Amazon Linux

Free tier eligible

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

Select

64-bit

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
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of lower prices, request On-Demand instances to pay for the instance, and more.

Number of instances

1

[Launch into Auto Scaling Group](#)

Purchasing option

☐ Request Spot instances

Network

vpc-e418ec8c (default)

[Create new VPC](#)

Subnet

No preference (default subnet in any Availability Zone)

[Create new subnet](#)

Auto-assign Public IP

Use subnet setting (Enable)

IAM role

None

[Create new IAM role](#)

Shutdown behavior

Stop

Enable termination protection

☐ Protect against accidental termination

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-02ff603aac696690	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

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:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8888	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

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](#).

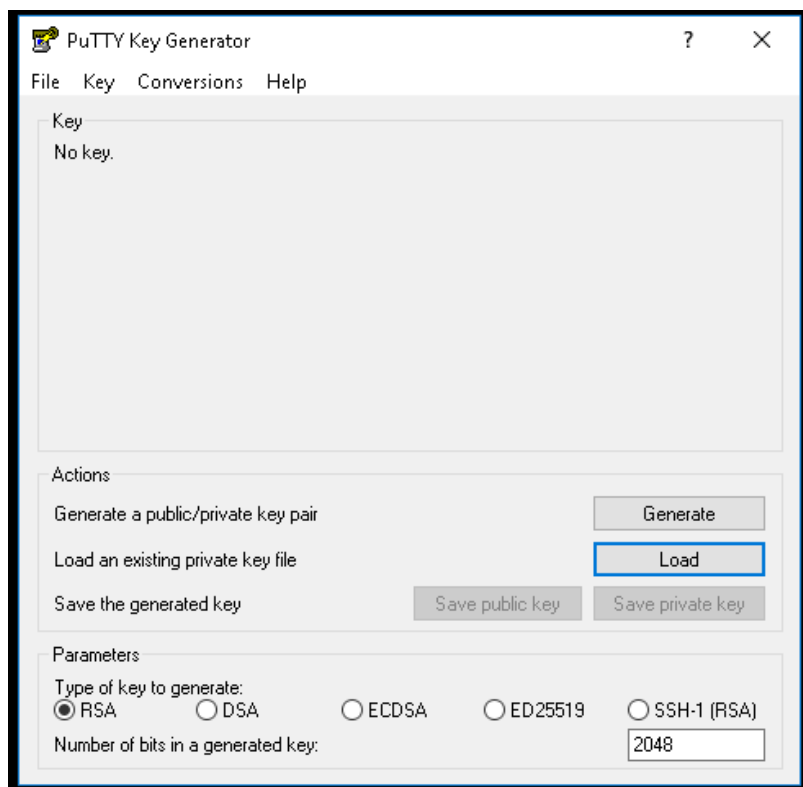
Select a key pair

☒ I acknowledge that I have access to the selected private key file (KeyPairEC2.pem), and that without this file, I won't be able to log into my instance.

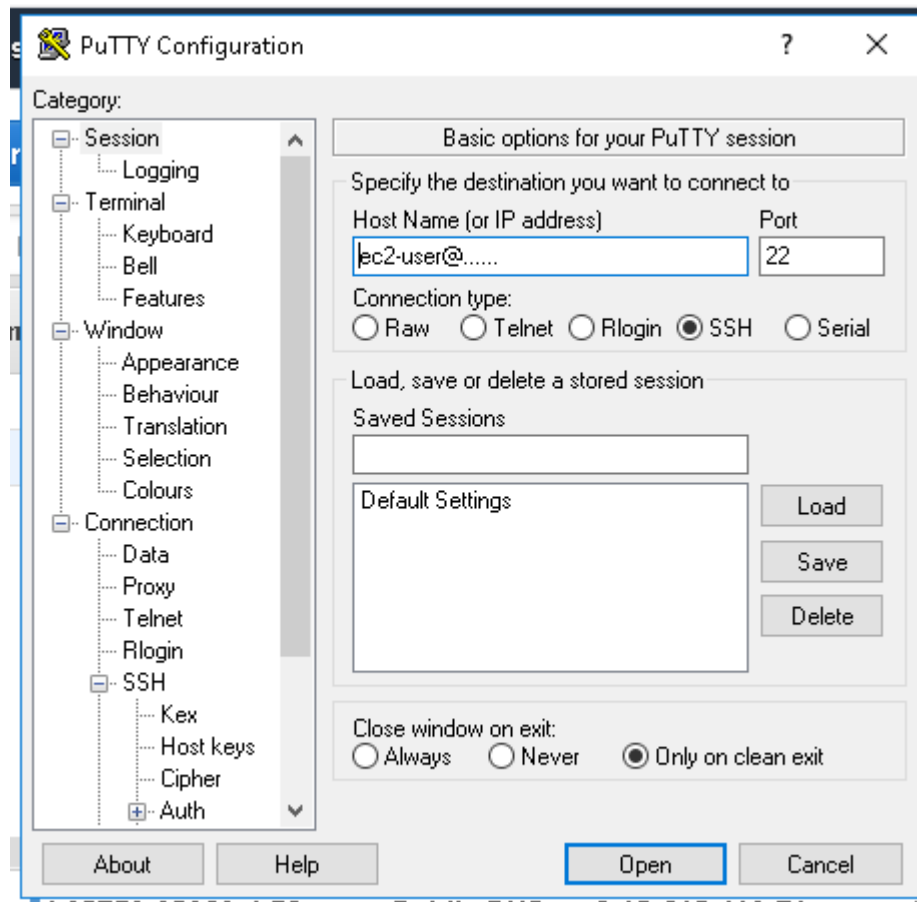
[Cancel](#)

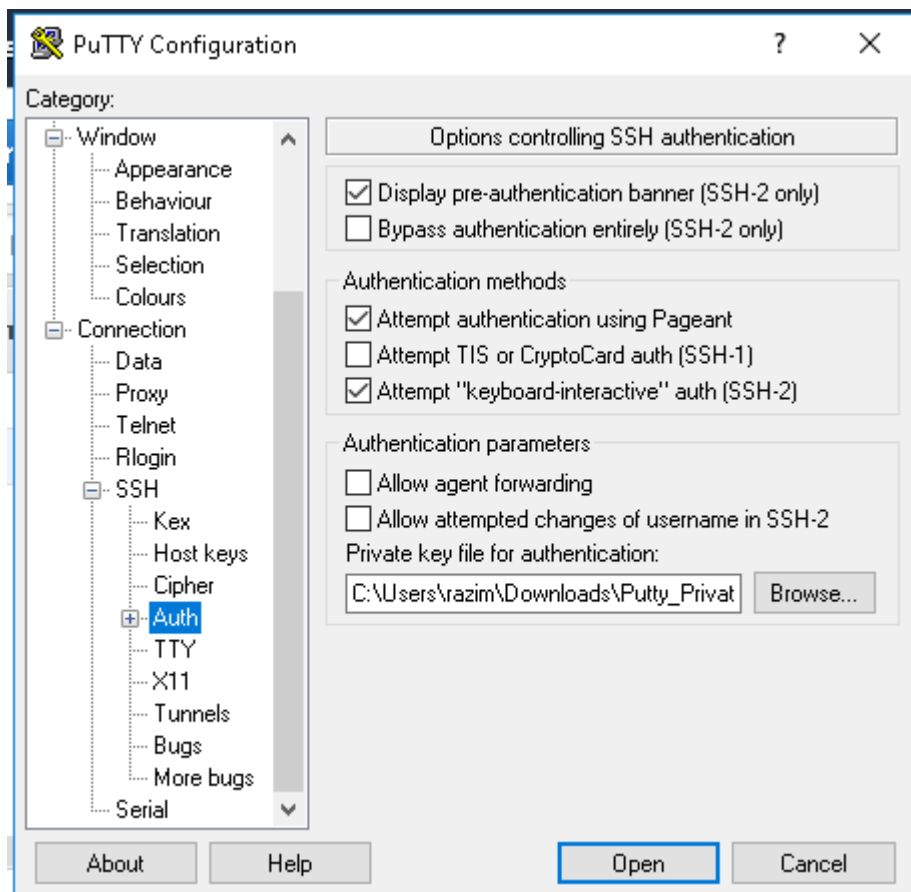
[Launch Instances](#)

Download Putty and load your KeyPair.pem from AWS in Putty Gen.

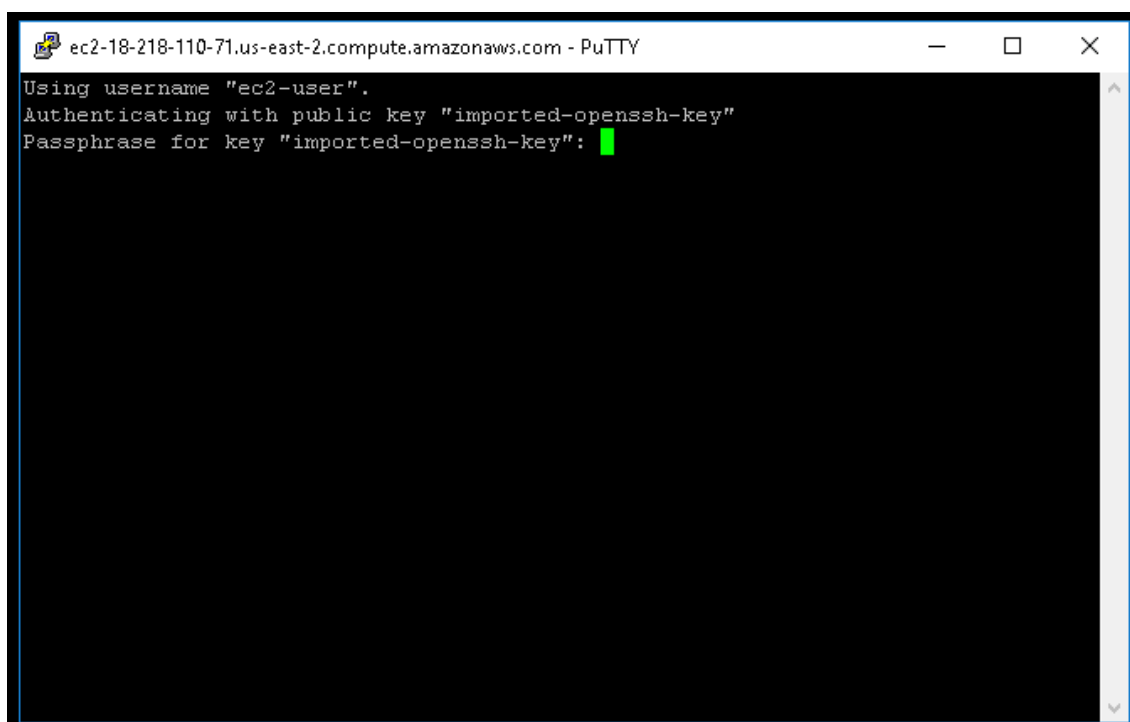


Open a Putty window and log in as ec2-user@...your Public DNS (IPv4) at AWS (bottom right)





Enter the password you created along with the KeyPair.pem



You're in

```
ec2-user@ip-172-31-21-49:~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
Passphrase for key "imported-openssh-key":  
  
  ____|  _||_ )  
  _|| ( _|| /   Amazon Linux AMI  
  ____|\_||_||  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
1 package(s) needed for security, out of 1 available  
Run "sudo yum update" to apply all updates.
```

Type:

`ssh -i /home/ec2-user/KeyPairEC2.pem ec2-user@...your_Public DNS (IPv4) at AWS`

Download Anaconda3

```
wget https://repo.continuum.io/archive/Anaconda3-4.4.0-Linux-x86_64.sh
```

Install Anaconda 3

```
bash Anaconda3-4.4.0-Linux-x86_64.sh
```

```
vim .bashrc
```

CTRL+C below

```
export PATH="/home/ec2-user/anaconda3/bin:$PATH"
```

```
type "i" SHIFT + INSERT
```

```
ESC + ":wq"
```

```
which python /usr/bin/python
```

```
source .bashrc
```

Create a Jupyter Password:

```
ipython  
from IPython.lib import passwd  
passwd()
```

```
IPython 5.3.0 -- An enhanced Interactive Python.
?      -> Introduction and overview of IPython's
%quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??'

In [1]: from IPython.lib import passwd

In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'sha1:49d9a3c06898:20ce3bbca0ccf25117653aec8
```

Type Exit

Save this code:

```
'sha1:49d9a3c06898:20ce3bbca0ccf25117653aec8a7a93...'
```

Create a brand new Jupyter config file:

```
jupyter notebook --generate-config
```

mkdir certs

cd certs

Create a new key to be used as a certificate:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
mycert.pem -out mycert.pem
```

```
ec2-user@ip-172-31-21-49:~/certs
In [3]: exit
[ec2-user@ip-172-31-21-49 ~]$ jupyter notebook --generate-config
Writing default config to: /home/ec2-user/.jupyter/jupyter_notebook_config.py
[ec2-user@ip-172-31-21-49 ~]$ mkdir certs
[ec2-user@ip-172-31-21-49 ~]$ cd certs
[ec2-user@ip-172-31-21-49 certs]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem
Generating a 1024 bit RSA private key
.....+++++
...+++++
writing new private key to 'mycert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:BR
State or Province Name (full name) []:Sao Paulo
Locality Name (eg, city) [Default City]:Sao Paulo
Organization Name (eg, company) [Default Company Ltd]:
```

NOW, THE FOLLOWING STEP IS THE TRICKY PART:

If you mess with the VIM configuration, that is, your jupyter notebook config, the whole thing won't work. So, if you make any mistakes in the following step, go back, type:

```
jupyter notebook --generate-config
```

and generate a brand new config file and then start VIM again.

Edit the Jupyter config file:

```
vim .jupyter/jupyter_notebook_config.py
```

CTRL+C the code below with your sha1 key, go to Putty. To insert text in VIM, type "i":

```
c = get_config()
c.IPKernelApp.pylab = 'inline'
c.NotebookApp.certfile = '/home/ec2-user/certs/mycert.pem'
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.password = 'sha1:49d9a3c06898:20ce3bbca0ccf25117653ae.....'
c.NotebookApp.port = 8888
```

SHIFT+INSERT to paste into VIM

ESC and type “:wq” to exit

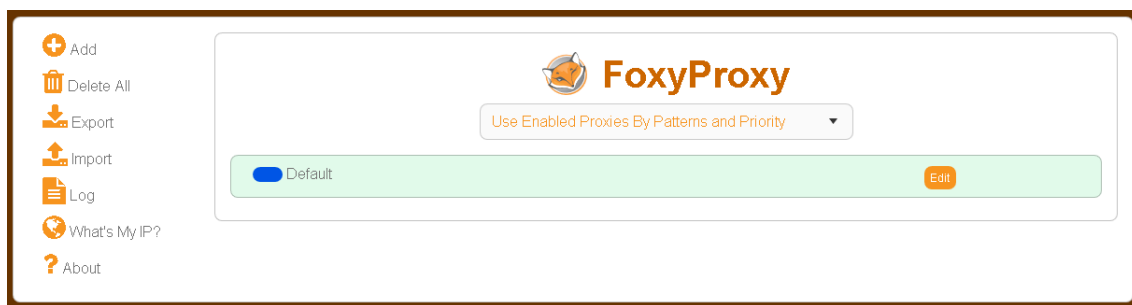
Type:

```
mkdir Notebooks
```

```
cd Notebooks
```

Now you have to config your browser:

Setup FoxyProxy: Download, add as extension:



Config Foxy Proxy .xml file:

Save the following code as foxyproxy-settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<foxyproxy>
```

```
<proxies>
```

```
<proxy name="ec2-socks-proxy" id="2322596116" notes="" fromSubscription="false" enabled="true"
mode="manual" selectedTabIndex="2" lastresort="false" animatedIcons="true" includeInCycle="true"
color="#0055E5" proxyDNS="true" noInternalIPs="false" autoconfMode="pac"
clearCacheBeforeUse="false" disableCache="false" clearCookiesBeforeUse="false" rejectCookies="false">
```

```
</matches>
```



```

    <match enabled="true" name="*ec2*.amazonaws.com*" pattern="*ec2*.amazonaws.com*"
isRegex="false" isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />

    <match enabled="true" name="*ec2*.compute*" pattern="*ec2*.compute*" isRegex="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false" />

    <match enabled="true" name="*.compute.internal*" pattern="*.compute.internal*"
isRegex="false" isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false"/>

    <match enabled="true" name="*.ec2.internal* " pattern="*.ec2.internal*" isRegex="false"
isBlackList="false" isMultiLine="false" caseSensitive="false" fromSubscription="false"/>

    </matches>

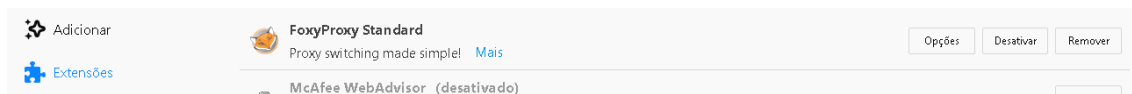
    <manualconf host="localhost" port="8888" socksversion="5" isSocks="true" username=""
password="" domain="" />

</proxy>

</proxies>

</foxyproxy>

```



Restart Firefox

Back to Putty, type:

jupyter notebook

You will get

```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Last login: Sun Jan 14 10:49:46 2018 from 179.209.47.139

  _ | _ | _ |
  _ | ( _ | /   Amazon Linux AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-21-49 ~]$ cd Notebooks
[ec2-user@ip-172-31-21-49 Notebooks]$ jupyter notebook
[I 12:52:23.256 NotebookApp] Serving notebooks from local directory: /
ser/Notebooks
[I 12:52:23.257 NotebookApp] 0 active kernels
[I 12:52:23.257 NotebookApp] The Jupyter Notebook is running at: https
addresses on your system]:8888/
[I 12:52:23.257 NotebookApp] Use Control-C to stop this server and shut
down all kernels (twice to skip confirmation).

█
```

Go to Firefox and enter your PublicDNS address, with **https://** before and **:8888** after

<https://ec2-54-144-47-200.compute-1.amazonaws.com:8888/>

Your browser will show the following screen:



Your connection is not secure

The owner of [REDACTED] has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

[Learn more...](#)

[Go Back](#)

[Advanced](#)

☐ Report errors like this to help Mozilla identify and block malicious sites

fritz.box uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: [SEC_ERROR_UNKNOWN_ISSUER](#)

[Add Exception...](#)

Add security exception

Enter password

WE'RE DONE !

If you want to install Install Keras and Tensorflow in Jupyter

cd anaconda3

pip install keras==2.0.8

```
[ec2-user@ip-172-31-21-49 anaconda3]$ pip install keras==2.0.8
Collecting keras==2.0.8
  Downloading Keras-2.0.8-py2.py3-none-any.whl (276kB)
    100% |#####| 276kB 424kB/s
Requirement already satisfied: scipy>=0.14 in ./lib/python3.6/site-packages (from keras==2.0.8)
Requirement already satisfied: numpy>=1.9.1 in ./lib/python3.6/site-packages (from keras==2.0.8)
Requirement already satisfied: pyyaml in ./lib/python3.6/site-packages (from keras==2.0.8)
Requirement already satisfied: six>=1.9.0 in ./lib/python3.6/site-packages (from keras==2.0.8)
Installing collected packages: keras
Successfully installed keras-2.0.8
[ec2-user@ip-172-31-21-49 anaconda3]$
```

pip install tensorflow==1.3.0 (for Python 3)

```
[I 11:27:35.203 NotebookApp] Shutting down kernels
[ec2-user@ip-172-31-21-49 anaconda3]$ pip install tensorflow==1.4.0
Collecting tensorflow==1.4.0
  Downloading tensorflow-1.4.0-cp36-cp36m-manylinux1_x86_64.whl (41.2MB)
    100% |#####| 41.2MB 3.0kB/s
Collecting tensorflow-tensorboard<0.5.0,>=0.4.0rc1 (from tensorflow==1.4.0)
  Downloading tensorflow-tensorboard-0.4.0rc3-py3-none-any.whl (1.7MB)
    100% |#####| 1.7MB 73kB/s
Collecting protobuf>=3.3.0 (from tensorflow==1.4.0)
  Downloading protobuf-3.5.1-cp36-cp36m-manylinux1_x86_64.whl (6.4MB)
    100% |#####| 6.4MB 20kB/s
Requirement already satisfied: six>=1.10.0 in ./lib/python3.6/site-packages (from tensorflow==1.4.0)
Requirement already satisfied: numpy>=1.12.1 in ./lib/python3.6/site-packages (from tensorflow==1.4.0)
Collecting enum34>=1.1.6 (from tensorflow==1.4.0)
  Downloading enum34-1.1.6-py3-none-any.whl
Requirement already satisfied: wheel>=0.26 in ./lib/python3.6/site-packages (from tensorflow==1.4.0)
Requirement already satisfied: bleach==1.5.0 in ./lib/python3.6/site-packages (from tensorflow-tensorboard<0.5.0,>=0.4.0rc1->tensorflow==1.4.0)
Collecting html5lib==0.9999999 (from tensorflow-tensorboard<0.5.0,>=0.4.0rc1->tensorflow==1.4.0)
  Downloading html5lib-0.9999999.tar.gz (889kB)
    100% |#####| 890kB 146kB/s
Collecting markdown>=2.6.8 (from tensorflow-tensorboard<0.5.0,>=0.4.0rc1->tensorflow==1.4.0)
  Downloading Markdown-2.6.11-py2.py3-none-any.whl (78kB)
    100% |#####| 81kB 1.0MB/s
```

Type:

jupyter notebook

The screenshot shows a Jupyter Notebook interface with the following components:

- Header:** Jupyter logo, "Untitled" title, "Last Checkpoint: 6 minutes ago (unsaved changes)", and a "Logout" button.
- Menu Bar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help.
- Toolbar:** Includes buttons for saving, undo, redo, and a dropdown menu currently set to "Code".
- Code Cell (In [4]):**

```
import keras
import keras
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Merge, Lambda, GlobalAveragePooling1D, GlobalAveragePooling2D, UpSampling1D, UpSampling2D
from keras.optimizers import SGD
from scipy.interpolate import spline
from keras.callbacks import LearningRateScheduler
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
from keras.layers.normalization import BatchNormalization
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from sklearn import datasets
import keras.backend as K
from keras.layers.core import Reshape
from keras.callbacks import ModelCheckpoint
```
- Code Cell (In [5]):**

```
import tensorflow as tf
```
- Code Cell (In [6]):**

```
model=Sequential()
```
- Code Cell (In [7]):**

```
2+2
```
- Output:** The output of the last cell is "Out[7]: 4".