



JUPYTER NOTEBOOK INSTALLATION GUIDE

This is a step by step approach to install Jupyter Notebook on the EC2 instance with Amazon Linux as the OS.

Prerequisites: You must already have an EC2 machine installed on your AWS. This installation will set up the jupyter notebook on that machine. If you need to install the EC2 instance again, you can use the installation guide given in the previous module.

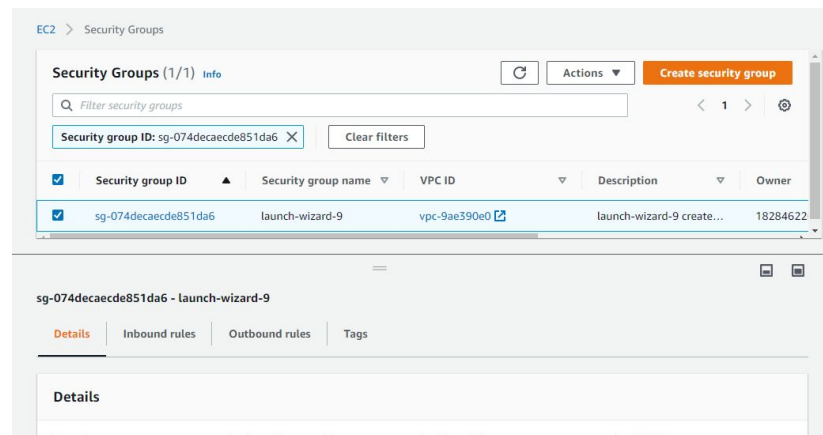
Let us begin installing jupyter notebook.

1. You will need access to the 8888 port of the EC2 instance. Let's check the security rules of the instance. You will find the security information about your instance in the description tab of the information displayed on the EC2 page.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'guduru.sai@upgrad.com @ 18...'. The left sidebar contains navigation options like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'IMAGES', and 'AMIs'. The main content area displays the details of an EC2 instance named 'Pyspark' with Instance ID 'i-023a83994f6c56c0e'. The instance is in a 'running' state, using the 't2.micro' instance type in the 'us-east-1d' availability zone. The public DNS is 'ec2-52-90-106-54.compute-1.amazonaws.com'. The 'Description' tab is selected, showing various attributes: Instance ID, Instance state (running), Instance type (t2.micro), Finding (You may not have permission to access AWS Compute Optimizer), Private DNS (ip-172-31-89-87.ec2.internal), Private IPs (172.31.89.87), Secondary private IPs, VPC ID (vpc-9ae390e0), and Subnet ID (subnet-c92d00e7). On the right side of the 'Description' tab, there are sections for 'Public DNS (IPv4)', 'IPv4 Public IP', 'IPv6 IPs', 'Elastic IPs', 'Availability zone', 'Security groups' (highlighted with a red box, showing 'launch-wizard-9. view inbound rules. view outbound rules'), 'Scheduled events', 'AMI ID', and 'Platform details' (Linux/UNIX).



- Click on launch wizard 9. You will be taken to a window like this.



- Go to inbound rules. And make sure the rules shown below are present in the list.

Inbound rules					Edit inbound rules
Type	Protocol	Port range	Source	Description - optional	
Custom TCP	TCP	8888	0.0.0.0/0	-	
SSH	TCP	22	0.0.0.0/0	-	

- If not you can add them by clicking on edit inbound rules.

- To install jupyter notebook there must be java, python3, pyspark and pip already installed in your instance. You have already installed all these while working on RDDs in the previous module. Therefore, there is no need to install them again if you are using the same instance and hence we will just run a few commands to ensure that they are installed.

- Java:

```
java -version
```

```
[ec2-user@ip-172-31-80-14 ~]$ java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
[ec2-user@ip-172-31-80-14 ~]$
```



- Python3:

```
python3 --version
```

```
[ec2-user@ip-172-31-80-14 ~]$ python3 --version
Python 3.7.6
[ec2-user@ip-172-31-80-14 ~]$
```

- Pyspark:

pyspark

```
[ec2-user@ip-172-31-80-14 ~]$ pyspark
Python 2.7.16 (default, Dec 12 2019, 23:58:22)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-6)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
20/04/16 07:47:20 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

      /_/_/_/_/_/_/_/_/_/_\
     / \ / \ / \ / \ / \ / \
    /   /   /   /   /   /   \
   /___/___/___/___/___/___\ version 2.4.4
  /___/___/___/___/___/___\
 /___/___/___/___/___/___\

Using Python version 2.7.16 (default, Dec 12 2019 23:58:22)
SparkSession available as 'spark'.
>>> _
```

- Exit the spark environment by typing

```
quit()
```

- pip

```
pip3 --version
```

```
[ec2-user@ip-172-31-80-14 ~]$ pip3 --version
pip 9.0.3 from /usr/lib/python3.7/site-packages (python 3.7)
[ec2-user@ip-172-31-80-14 ~]$
```

2. Now let's install the jupyter notebook. Use the following command



```
pip3 install jupyter --user
```

3. Run the following command to check if there are any jupyter servers running on your ec2 instance.

```
jupyter notebook list
```

You should get this output as there are no jupyter servers running currently.

```
[ec2-user@ip-172-31-89-87 ~]$ jupyter notebook list
Currently running servers:
[ec2-user@ip-172-31-89-87 ~]$
```

This tells us that there are no jupyter servers running.

4. Now let's create a configuration file to store the setting that we need to launch jupyter. Please note that this is a one time exercise. Once the file is created you can directly run jupyter notebook.

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

This will write a file to this path: /home/ec2-user/.jupyter/jupyter_notebook_config.py

```
[ec2-user@ip-172-31-89-87 ~]$ jupyter notebook --generate-config
Writing default config to: /home/ec2-user/.jupyter/jupyter_notebook_config.py
[ec2-user@ip-172-31-89-87 ~]$
```

5. Let's add instruction to the config file we created. Go to the directory where the file is created

```
cd .jupyter/
```

- List the files in the directory

```
ls
```

- You should see the file we just created.

```
[ec2-user@ip-172-31-89-87 ~]$ cd .jupyter/
[ec2-user@ip-172-31-89-87 .jupyter]$ ls
jupyter_notebook_config.py
[ec2-user@ip-172-31-89-87 .jupyter]$
```

6. Add instructions to the configuration file

```
vi jupyter_notebook_config.py
```

- Now we will be performing the following commands:
Press i to start editing.



```
c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '0.0.0.0'
```

- The Command Prompt will look like this:

```
c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '0.0.0.0'
# Configuration file for jupyter-notebook.
#-----
# Application(SingletonConfigurable) configuration
#-----
```

Once you have edited, press **Esc** and type **:wq!** to save and exit the editor.

- Come out of the directory.

```
cd
```

7. Now run the jupyter notebook.

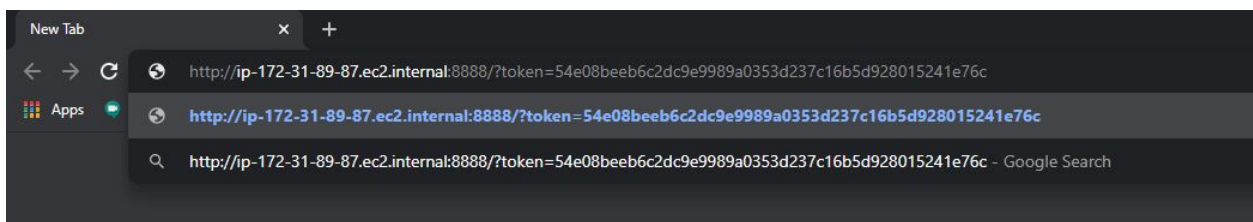
```
Jupyter notebook
```

- You will get an output like this

```
[ec2-user@ip-172-31-89-87 ~]$ jupyter notebook
[I 06:01:41.285 NotebookApp] Writing notebook server cookie secret to /home/ec2-user/.local/share/jupyter/runtime/notebook_cookie_secret
[I 06:01:41.541 NotebookApp] Serving notebooks from local directory: /home/ec2-user
[I 06:01:41.541 NotebookApp] http://ip-172-31-89-87.ec2.internal:8888/?token=54e08beeb6c2dc9e9989a0353d237c16b5d928015241e76c
[I 06:01:41.541 NotebookApp] or http://127.0.0.1:8888/?token=54e08beeb6c2dc9e9989a0353d237c16b5d928015241e76c
[I 06:01:41.542 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 06:01:41.546 NotebookApp] No web browser found: could not locate runnable browser.
[C 06:01:41.546 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/ec2-user/.local/share/jupyter/runtime/nbserver-29917-open.html Or copy and paste one of these URLs:
http://ip-172-31-89-87.ec2.internal:8888/?token=54e08beeb6c2dc9e9989a0353d237c16b5d928015241e76c
or http://127.0.0.1:8888/?token=54e08beeb6c2dc9e9989a0353d237c16b5d928015241e76c
```

8. Copy the first link out of the two which are shown in the last line and paste it in your browser but don't run it yet.





- You will need to replace the private IP link with the public IP of the EC2 instance. You can find the public IP of the instance on the right side of the EC2 page.

The screenshot shows the AWS Management Console interface. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below this is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). A single instance named 'Pyspark' is listed with Instance ID 'i-023a83994f6c56c0e', Instance Type 't2.micro', Availability Zone 'us-east-1d', Instance State 'running', Status Checks '2/2 checks ...', Alarm Status 'None', and Public DNS 'ec2-52-90-106-54.compute-1.amazonaws.com'. Below the table, the details for the selected instance are shown. The 'Description' tab is active, displaying fields for Instance ID, Instance state, Instance type, Finding, Public DNS (IPv4), IPv4 Public IP, IPv6 IPs, and Elastic IPs. The 'IPv4 Public IP' field is highlighted with a red box and contains the value '52.90.106.54'.

- Replace it in the link and now you can run.

The screenshot shows a web browser window with a single tab titled 'New Tab'. The address bar contains the URL 'http://52.90.106.54:8888/?token=54e08beeb6c2dc9e9989a0353d237c16b5d928015241e76c'. The browser's taskbar at the bottom shows icons for 'Apps' and 'Meet - C4M3 shoot...'.

- This will take you to the jupyter notebook home page.

The screenshot shows the Jupyter Notebook home page in a web browser. The browser's address bar shows the URL 'http://52.90.106.54:8888/tree'. The page has a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a list of files and folders. The list includes '0', 'spark-2.4.4-bin-hadoop2.7', 'get-pip.py', and 'spark-2.4.4-bin-hadoop2.7.tgz'. The 'Last Modified' column shows '9 months ago' for the first two items and 'an hour ago' for 'get-pip.py'. The 'File size' column shows '1.87 MB' for 'get-pip.py' and '230 MB' for 'spark-2.4.4-bin-hadoop2.7.tgz'.