Document for Machine Learning Server setup for R & Python

Pre-requisites:

- Setup a EC2 Linux Ubuntu instance and assign it an elastic IP (so that it's endpoint remain always fixed)
- Configure the security-group of the EC2 instance to allow all outgoing requests and following specific port incoming request:

Type	Protocol	Port Range	Source	Description
Custom TCP	TCP	8888	Anywhere	Jupyetr Notebook
Custom TCP	TCP	8787	Anywhere	R Studio
Custom TCP	TCP	443	Anywhere	Jupyetr
SSH	SSH	22	Anywhere	Remote Login

- Remote login to EC2 instance using Putty or Terminal

1. **Installing R-studio server on AWS EC2:**

   $ sudo apt-get update

   $ sudo apt-get install r-base

   $ sudo apt-get install gdebi-core

   $ wget https://download2.rstudio.org/rstudio-server-1.1.423-amd64.deb

   $ sudo gdebi rstudio-server-1.1.423-amd64.deb

   $ sudo rstudio-server verify-installation

   $ sudo adduser rstudio

Now restart the EC2 instance and go to the following URL to access R Studio Server:

http://<your-elastic-ip-for-ec2-instance>:8787

Enter your username and password to login.

Reference:

https://www.rstudio.com/products/rstudio/download-server/

https://support.rstudio.com/hc/en-us/articles/200552306-Getting-Started

2. Installing Jupyter Notebook server on AWS EC2:

$ sudo apt-get update

$ sudo adduser jupyter

$ wget https://repo.continuum.io/archive/Anaconda3-5.1.0-Linux-x86_64.sh

$ bash Anaconda3-5.1.0-Linux-x86_64.sh

$ cd

$ source .bashrc

$ cd /home/< YOUR-USERNAME >/.jupyter/

$ jupyter notebook --no-browser

$ jupyter notebook --generate-config

$ jupyter notebook --NotebookApp.allow_password_change=False

$ jupyter notebook password (Enter the password to be used)

Using the function notebook.auth.security.passwd(), find hashed password:

$ ipython

```
In [1]: from notebook.auth import passwd
In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'sha1:67c9e60bb8b6:9ffede0825894254b2e042ea597d771089e11aed'
```

(Copy this password and keep it somewhere safe)

$ cd ..

$ mkdir certs

$ cd certs

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

$ cd /home/<YOUR-USERNAME>/.jupyter/

Update the following file:

$ **vim jupyter_notebook_config.py** (use the password obtained in above step):

```
c = get_config()
c.IPKernelApp.pylab = 'inline'
# change the url to your certfile as per your username
c.NotebookApp.certfile = u'/home/ubuntu/certs/mycert.pem'
c.NotebookApp.ip = '*'
```

```
c.NotebookApp.open_browser = False
# Your password below will be whatever you copied earlier
c.NotebookApp.password =
u'sha1:8c4a3b995028:e78436aebcd81aef276011aee63707f81b5edf63'
c.NotebookApp.port = 8888
```

Now, Create the file:

$ **vi /etc/systemd/system/jupyter.service**

```
[Unit]
Description=Jupyter Workplace

[Service]
Type=simple
PIDFile=/run/jupyter.pid
ExecStart=/home/ubuntu/anaconda3/bin/jupyter-notebook --
config=/home/ubuntu/.jupyter/jupyter_notebook_config.py
User=ubuntu
Group=ubuntu
WorkingDirectory=/home/ubuntu
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Then, run the commands :

$ sudo systemctl enable jupyter.service
$ sudo systemctl daemon-reload
$ sudo systemctl restart jupyter.service

Now restart the EC2 instance and go to the following URL to access Jupyter Notebook Server:

https://<your-elastic-ip-for-ec2-instance>:8888

Enter your username and password to login.

Reference:

http://jupyter-notebook.readthedocs.io/en/latest/public_server.html

https://hackernoon.com/aws-ec2-part-4-starting-a-jupyter-ipython-notebook-server-on-aws-549d87a55ba9

NOTE: If any additional package is required to be installed (remember packages can be installed from the notebooks itself, but if due to any reason that fails, then only use this method), use terminal or Putty to access EC2 instance command line and there you can install packages manually.