**Procédure EC2 et S3 pour Ubuntu**

**1) Configuration EC2 et ouverture notebook jupyter sous EC2**

**Ouvrir un compte AWS** et aller sur S3 et mettre en place un bucket S3 dédié au projet.

Ensuite aller sur EC2 et **choisir une instance EC2** (gratuite/payante).

Lors du **paramétrage du groupe de sécurité**, on doit ouvrir les ports suivants « depuis n’importe-où » (cf : <https://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science/5559821-lancez-une-session-de-notebook-jupyter-sur-aws>):

- HTTPS 443 en TCP 0.0.0.0, ::/0

- règle TCP 8888 en TCP pour jupyter notebook 0.0.0.0/0

- SSH 22 en TCP 0.0.0.0/0

A la fin de l’installation de l’instance EC2, on nous propose une **clé SSH** au format .pem : télécharger-là et la mettre dans le dossier ~/.ssh

Une fois fait, **démarrer l’instance**

**Ouvrir un terminal**

**Se connecter en SSH** avec la commande :

ssh -i "path/to/pem\_key" ec2-user@adresse\_ec2\_DNS\_public

exemple :

*ssh -i "~/.ssh/aws\_irlande.pem" ubuntu@ec2-34-244-205-102.eu-west-1.compute.amazonaws.com*

Ouvrir un navigateur en parallèle et aller sur [https://www.anaconda.com](https://www.anaconda.com/) et **vérifier la dernière version de anaconda** disponible pour Linux (actuellement [https://repo.anaconda.com/archive/Anaconda3-2020.11-Linuxx86\_64.sh](https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh))

**Installer Anaconda** sur EC2 : Dans le terminal SSH, taper :

wget "[https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86\_64.s](https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh)

bash Anaconda3-2020.11-Linux-x86\_64.sh

source .bashrc

**Paramètrer jupyter notebook** pour mettre un mot de passe afin de protéger l’accès à sa session:

Dans le terminal, **pour ouvrir une session jupyter taper**  : (à faire également en cas de redémarrage de l’instance)

ipython

Puis **lors du premier lancement** de l’instance uniquement, taper :

from IPython.lib import passwd

passwd()

**Entrer votre mot de passe** 2 fois puis **sauvegarder l’empreinte sha1 générée** (exemple : 'sha1:aa987ccd91a9:962184ac72e1ca562c74e5e3b4be25e80f3f0e55') puis taper :

exit

Ensuite il faut **configurer jupyter** :

jupyter notebook –-generate-config

Puis **configurer la partie certificat,** taper**:**

mkdir certs && cd certs

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

Remplir ou pas les champs demandés (vous pouvez renseigner n’importe quoi comme nom ou ville etc.)

Ensuite, **modifier le fichier de configuration jupyter** :

vim ~/.jupyter/jupyter\_notebook\_config.py (ou autre logiciel d’édition que vim)

**Ajouter les lignes suivantes** dans le fichier :

c = get\_config()

c.IPKernelApp.pylab = 'inline'

c.NotebookApp.certfile = u'/home/ubuntu/certs/mycert.pem'

c.NotebookApp.keyfile = u'/home/ubuntu/certs/mycert.key'

c.NotebookApp.ip = '\*'

c.NotebookApp.open\_browser = False

c.NotebookApp.password=u'sha1:2a9d7deafee6:c8d475f7f01b0845f2c7dc0d18d3193ef3595561' (Avec la bonne clé)

c.NotebookApp.port = 8888

Après cela, il est normalement possible de lancer le notebook avec la commande :

jupyter notebook

puis en copiant l’adresse DNS publique IPV4 de l’instance EC2 dans un navigateur :

[https://EC2\_ADRESSE\_DNS\_IPV4:8888](https://VOTRE_ADRESSE_DNS_IPV4:8888/)

exemple :

[https://*ec2-34-244-205-102.eu-west-1.compute.amazonaws.com:8888*](https://ec2-34-244-205-102.eu-west-1.compute.amazonaws.com:8888/)

Accepter le message d’alerte (car le certificat est auto-signé)

**Si pas de lancement en https :**

Changer les propriétés de /home et ~/.local/share/jupyter/ à l’utilisateur en cours:

sudo chown -R $USER /home/

sudo chown -R $USER ~/.local/share/jupyter/

**Ajouter un rôle IAM pour un accés à s3 direct depuis le notebook**

Dans le panneau de contrôle des instances :

Actions/Securité/Modifier le rôle IAM/créer un rôle IAM

**2) Installations de modules complémentaires :**

Dans l’instance EC2, via le terminal :

S’assurer que le dernier module pip est installé :

pip install pip --upgrade

Installation des différents packages Python requis pour ce projet

pip3 install findspark

pip install --no-cache-dir opencv-python

pip install --no-cache-dir tensorflow

...

**Installer Spark sous EC2**

Télécharger la version de Spark utilisée lors du développement du projet en local (ici, version 3.0.2) :

wget https://downloads.apache.org/spark/spark-3.0.2/spark-3.0.2-bin-hadoop2.7.tgz

Dézipper le package dans un répertoire créé pour cela :

sudo tar zxvf spark-3.0.2-bin-hadoop2.7.tgz -C /opt

Mettre à jour les droits et créer un lien symbolique pour rendre accès plus simple :

sudo chown -R ubuntu:ubuntu /opt/spark-3.0.2-bin-hadoop2.7

sudo ln -fs spark-3.0.2-bin-hadoop2.7 /opt/spark

Mettre à jour le fichier bash\_profile avec la variable d’environnement SPARK\_HOME :

vim ~/.bash\_profile

Ajouter:

export SPARK\_HOME=/opt/spark

export PATH=$PATH:$SPARK\_HOME/bin

mettre à jour le bash profile :

source ~/.bash\_profile

**Installer Java**

Pour moi, pour la PCA cela ne marchait pas avec la dernière version Java, donc installer la 8 :

sudo apt-get update

install openjdk-8-jre-headless

**3) Autres**

**Ajout la clé aws**

pip install awscli

aws configure

suivre les instruction sur informations clé IAM

**Copie fichier sur S3 :**

Depuis EC2, dans le terminal on tape :

aws s3 cp ~/path\_to\_file s3://path\_in\_s3

Exemple

aws s3 cp ~/p8\_samuel\_monet.ipynb s3://p8smonet/p8\_samuelmonet.ipynb

**Copie fichier en local :**

**Depuis EC2 :**

En local, dans le terminal, on tape :

scp -i ~/path\_to\_pem\_key [ec2\_user@adresse\_DNS\_publique\_EC2](mailto:ec2_user@adresse_DNS_publique_EC2):~/path\_to\_file\_to\_be\_copied ~/path\_to\_where\_to\_copy\_on\_computer

Exemple :

scp -i ~/.ssh/aws\_irlande.pem ubuntu@ec2-34-244-205-102.eu-west-1.compute.amazonaws.com:~/p8\_samuel\_monet.ipynb ~/Documents/Data/OC/Projet8

**Depuis S3 :**

En local, dans le terminal, on tape :

aws s3 cp s3://path\_to\_file\_in\_s3 ~/path\_in\_computer (avec un ajout de --recursive pour les dossiers)

Exemple :

Pour le notebook :

aws s3 cp s3://p8smonet/p8\_samuel\_monet.ipynb ~/Documents/Data/OC/Projet8/P8\_samuel\_monet/P8\_01.notebook.ipynb

Pour le fichier avec les images et les features :

aws s3 cp s3://p8smonet/more ~/Documents/Data/OC/Projet8/P8\_samuel\_monet/P8\_02\_images --recursive