TP 0 — Partie 2: Créer et utiliser un cluster Spark avec EMR (Elastic Map Reduce)

1. Création d'une clef SSH

SSH (**S**ecure **SH**ell) permet de se connecter de façon sécurisée à un système Unix, Linux et Windows. Pour plus d'information, je vous conseille de lire le début de cette page web (https://doc.fedora-fr.org/wiki/SSH_:_Authentification_par_cl%C3%A9)

❏ 1-1 : Dans la barre de recherche, cherchez "EC2" et cliquez dessus
❏ 1-2 : Dans le panneaux de gauche cherchez "Paires de clef" (dans la section "Réseau et sécurité") et cliquez dessus.
❏ 1-3 : Cliquez sur "Créer une paire de clés"
❏ 1-4 : Donnez lui un nom (par ex: "spark_cluster_TP"), sélectionnez le format PPK, et cliquez sur "créer"
❏ 1-5 : Enregistrez le fichier et ne le perdez pas !

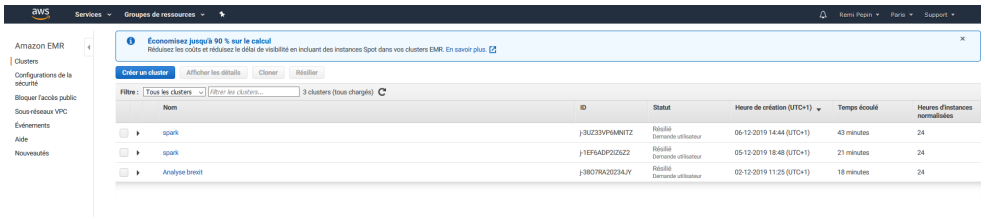
2. Conversion au format PPK

❏ 2-1 : Dans la barre de recherche windows cherchez "PuTTygen"
❏ 2-2 : Cliquez sur Load
❏ 2-3 : Allez dans le dossier où vous avez sauvegardé votre clef. Elle ne doit pas encore apparaître.
❏ 2-4 : En bas à droite sélectionnez "All Files (*.*)"
❏ 2-5 : Sélectionnez votre clef
❏ 2-6 : Un message apparait sur PuTTygen, validez le
❏ 2-7 : Cliquez sur "Save private key", puis sur "Oui" (on ne va pas mettre de passphrase)
❏ 2-8 : Sauvegardez votre clef privée .ppk
❏ 2-9 : Quittez PuTTygen

Vous avez fini de générer votre clef ssh!

3. Création d'un cluster Spark avec EMR

❏ 3-1 Sélectionnez le service EMR



Console EMR

❏ 3-2 Cliquez sur le bouton "Créer un cluster"

- Donnez le nom que vous voulez à votre cluster, par exemple Spark-TPX avec X le numéro du TP
- Laissez sélectionnée la journalisation. Cette option permet à votre cluster de stocker les log (journaux) de votre application sur votre espace S3 et ainsi faciliter le débogage. Comme vos log sont stockées sur S3, Amazon va vous facturer le stockage. Le prix de stockage sur S3 est extrêmement faible (0,023$ par Go par mois si vous avez moins de 50To), mais il peut être intéressant d'allez nettoyer vos vieilles log de temps en temps.

❏ 3-3 Configurez les logiciels

- Laissez la version d'emr par défaut
- Sélectionnez comme application Spark

❏ 3-4 Configurez le matériel

- Choisissez le type d'instance, par exemple `m5.xlarge` (4 cores avec une fréquence max de 3,1 GHz d'un Intel Xeon Platinum série 8000 avec 16Go de Ram). Prix total de 0.272$/h par instance.
- 3 Instances (ou plus selon vos envies et votre budget)

❏ 3-5 Sécurité et accès

- Sélectionnez la clef SSH que vous venez de générer
- Laissez le Rôle EMR et le Profil d'instance par défaut

❏ 3-6 Démarrer le cluster. Le démarrage peut prendre quelques minutes

Bravo vous avez démarré un cluster Spark en moins de 15min !

Détail cluster

4. Accès à l'interface de suivi du cluster

Installer FoxyProxy

Pour accéder à l'interface de suivi il est nécessaire d'installer le plugin FoxyProxy Standard sur votre navigateur.

Pour google chrome : lien (https://chrome.google.com/webstore/detail/foxyproxy-standard/gcknhkkoolaabfmlnjonogaaifnjlfnp?hl=fr)

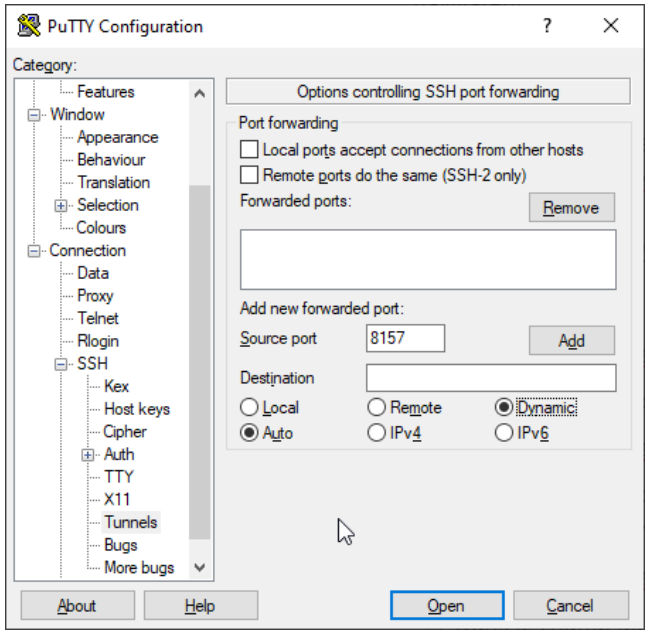
Pour firefox: lien (https://addons.mozilla.org/fr/firefox/addon/foxyproxy-standard/)

Une fois FoxyProxy installé ouvrez le plugin et importer le fichier se trouvant dans : /settings/foxyproxy-settings.json (lien (/settings/foxyproxy-settings.json))

Etablir une connection SSH avec votre cluster

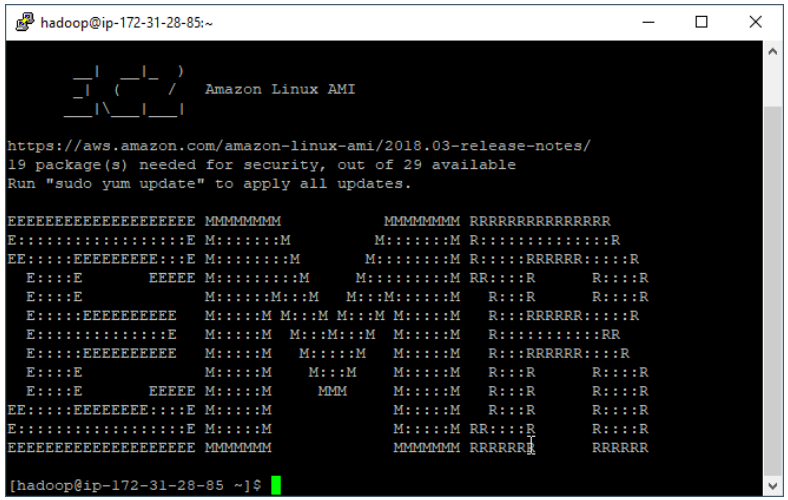
- [ ] Lancez PuTTY
- [ ] Dans la liste Category, cliquez sur Session
- [ ] Dans le champ Host Name, tapez **hadoop@XXXX (mailto:hadoop@XXXX)** avec XXXX le DNS public principal de votre cluster (vous le trouverez dans les informations de votre cluster sur l'interface aws)
- [ ] Dans la liste Category, développez Connection > SSH > Auth
- [ ] Pour le fichier de clés privées utilisé pour l'authentification, cliquez sur Browse et sélectionnez le fichier de clés privées utilisé pour lancer le cluster.
- [ ] Dans la liste Category, développez Connection > SSH, puis cliquez sur Tunnels.

- [ ] Dans le champ Source port, tapez **8157**
- [ ] Sélectionnez les options Dynamic et Auto.



Putty tunnels configuration

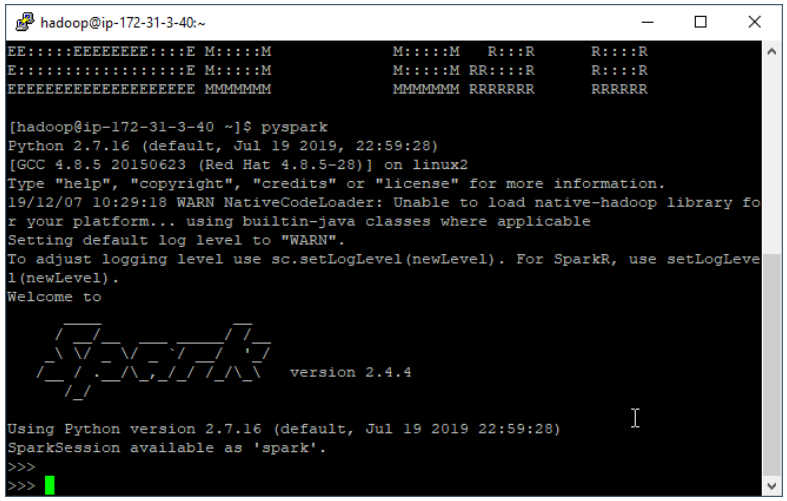
- [ ] Laissez le champ Destination vide, puis cliquez sur Add.
- [ ] Cliquez sur Open.
- [ ] Cliquez sur Yes pour ignorer l'alerte de sécurité.



- [ ] Une fois connectez en ssh à votre cluster vous pouvez lancer spark-shell ou pySpark avec

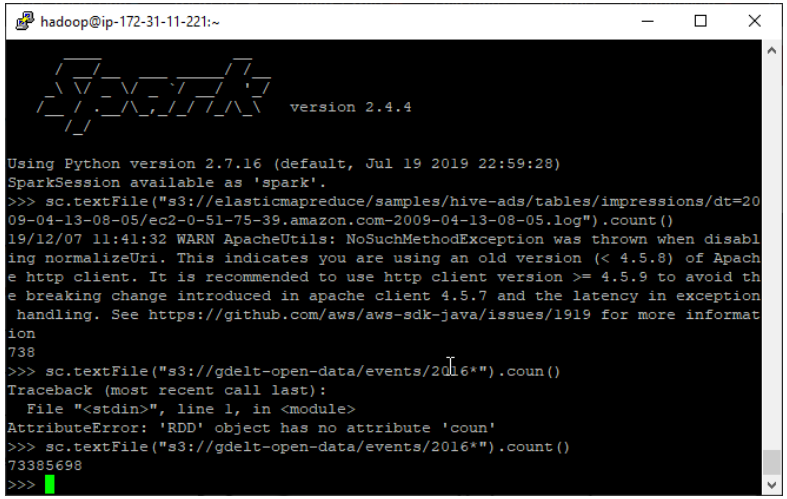
```
pyspark #pour lancer pyspark
spark-shell #pour spark-shell
```

Si vous préférez écrire votre code en python, il est nécessaire de lancer spark-shell avant pour charger toutes les bibliothèques java nécessaires.



- [ ] Vous pouvez désormais écrire du code spark en interactif. Par exemple voici un petit script python qui compte le nombre de lignes dans un fichier public stocké sur s3.

```
>> sc
<pyspark.context.SparkContext object at 0x7fe7e659fa50>
>>> textfile = sc.textFile("s3://gdelt-open-data/events/2016*")
>>> textfile.count()
73385698
```



Voici le même script scala en plus condensé

```
sc.textFile("s3://gdelt-open-data/events/2016*").count()
```

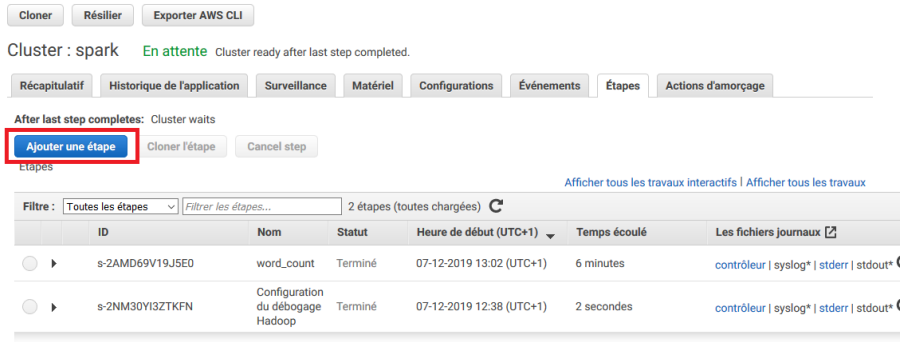
Ouvrir les interfaces de suivi

Une fois la connexion shh établie, et FoxyPproxy configuré, vous pouvez désormais accéder aux interfaces de suivi du cluster.

Liens connexion interfaces cluster

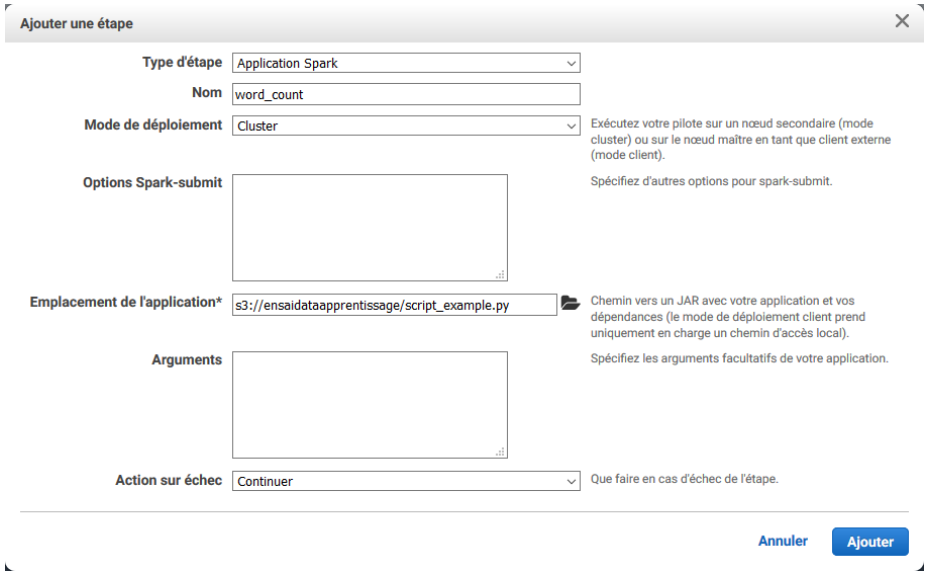
5. Lancer un job avec un script

- [ ] Upload sur S3 le script que vous voulez utiliser. Par exemple le fichier exemple (/exemple/script_exemple.py) suivant.
- [ ] Sur l'interface de votre cluster sélectionnez l'onglet "Etape"



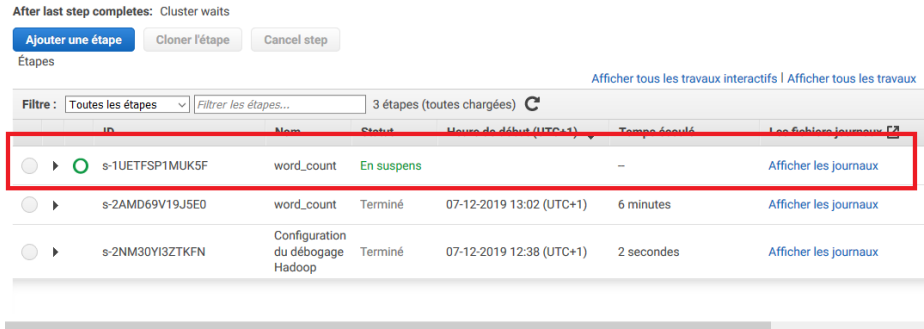
Ecran étape cluster

- [ ] Ajouter une étape
  - [ ] Type étape : application Spark
  - [ ] Nom de l'application : word_count
  - [ ] Mode de déploiement : cluster
  - [ ] Emplacement de l'application : allez chercher sur s3 le script uploadé plus tôt
  - [ ] "Ajouter"



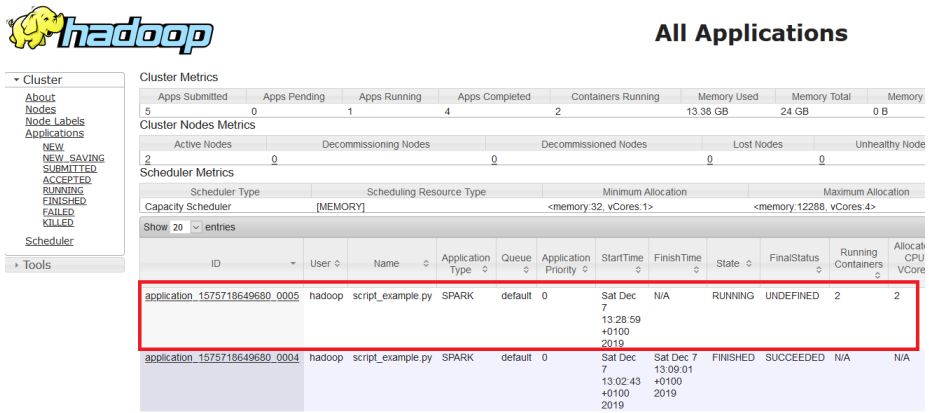
Ecran ajout d'une étape

- [ ] Vous allez voir votre script apparaitre dans les étapes de votre cluster. Son exécution peut prendre quelques minutes.

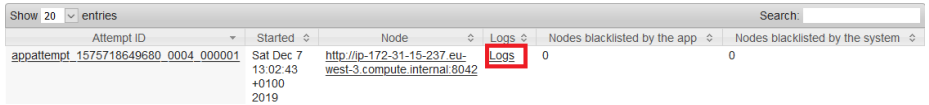


Ecran après ajout d'une étape

- [ ] Pour voir le résultat retournez dans la l'onglet "Récapitulatif" puis cliquez sur "Gestionnaire de ressource"
- [ ] Sur l'interface d'Hadoop sélectionnez votre application, puis en bas de la nouvelle page cliquez sur Logs



Ecran de gestion hadoop



Choix des logs

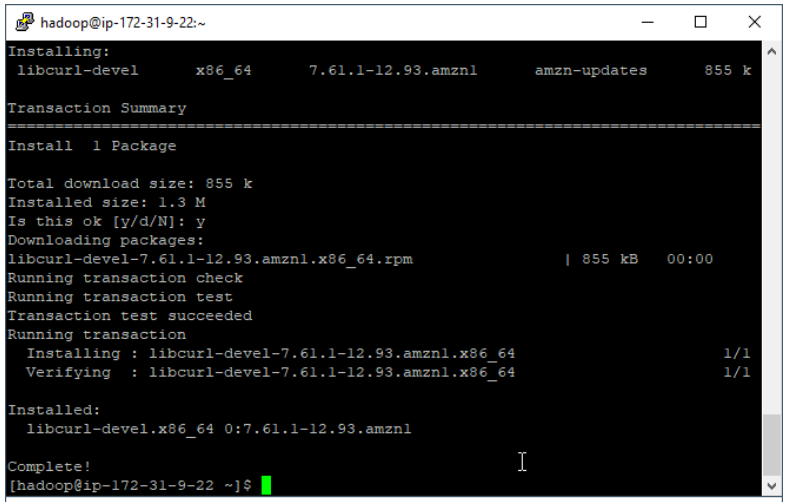
- [ ] En bas de la page de log vous trouverez votre résultat

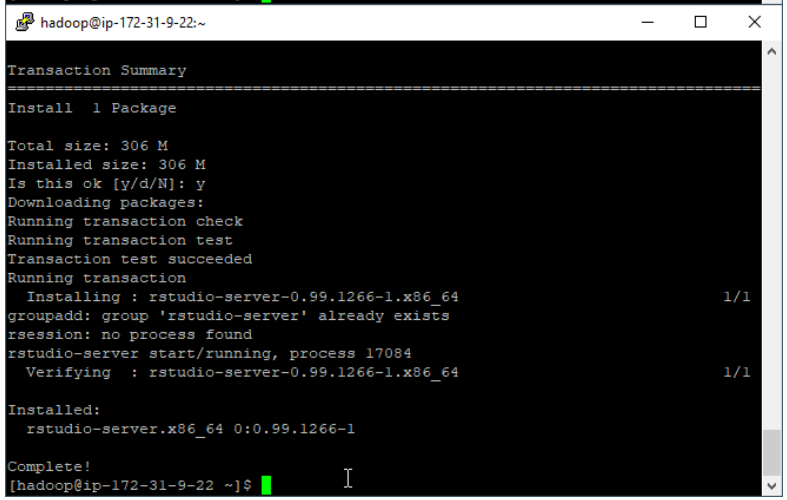
Log finale

6. Se connecter avec Rstudio et sparklyR

- [ ] Connectez-vous en SSH à votre cluster EMR
- [ ] Installez Rstudio server

```
sudo yum install libcurl-devel openssl-devel # used for devtools
wget -P /tmp https://s3.amazonaws.com/rstudio-dailybuilds/rstudio-server-rhel-0.99.1266-x86_64.rpm
sudo yum install --nogpgcheck /tmp/rstudio-server-rhel-0.99.1266-x86_64.rpm
```





- [ ] Créez un user pour Rstudio

```
# Make User
sudo useradd -m rstudio-user
sudo passwd rstudio-user
```

- [ ] Créez un dossier dans HDFS pour votre user

```
# Create new directory in hdfs
hadoop fs -mkdir /user/rstudio-user
hadoop fs -chmod 777 /user/rstudio-user
```

- [ ] Connectez-vous à l'interface web de Rstudio server avec l'adresse suivante https://master-node-public-DNS:8787 (https://master-node-public-DNS:8787) puis connectez vous avec l'utilisation rstudio-user et le mot de passe que vous avez choisi.
- [ ] Vous pouvez commencer à coder. Voici un script exemple : lien (exemple/script_exemple_R.R)