

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

1. Création d’une clef SSH

SSH (Secure SHell) 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 !

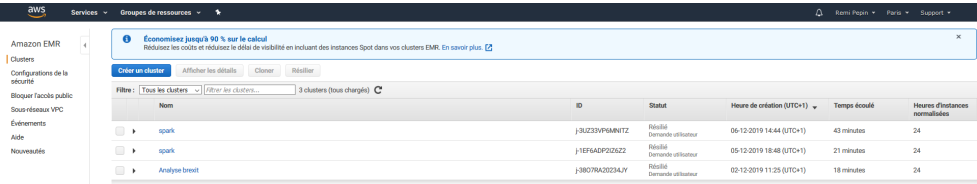
2. Conversion au format PPK

- ❑ 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’aller 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/gcknhkkoalaabfmijnogaaifnfn?hl=fr>)

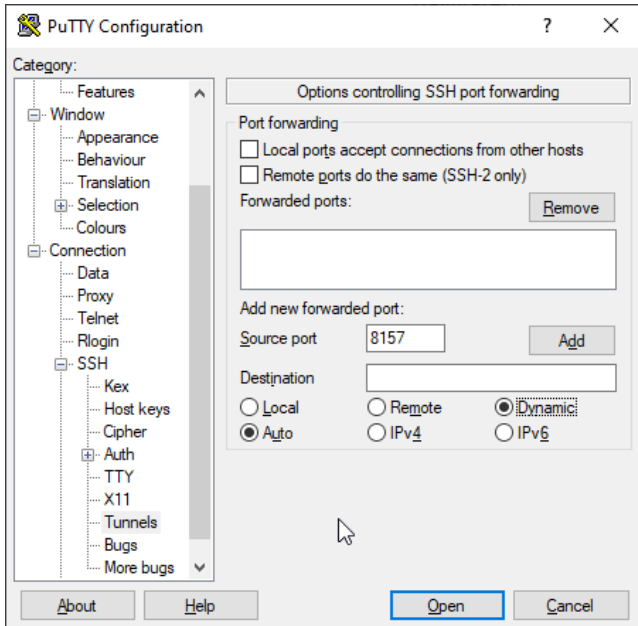
Pour firefox : lien (<https://addons.mozilla.org/fr/firefox/addon/foxyproxy-standard/>)

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

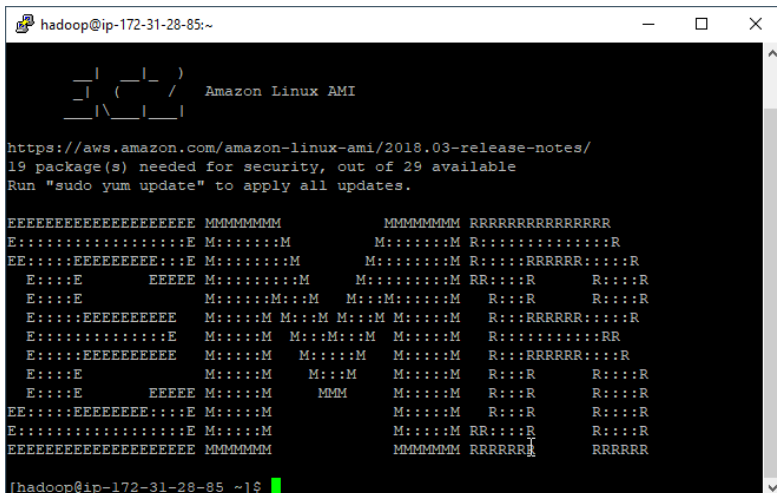
- [] Lancez PuTTY
- [] Dans la liste Category, cliquez sur Session
- [] Dans le champ Host Name, tapez **hadoop@XXXX (mailto:hadoop@XXXX)** avec XXXX le DNS public principal de votre cluster (vous le trouverez dans les informations de votre cluster sur l’interface aws)
- [] Dans la liste Category, développez 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

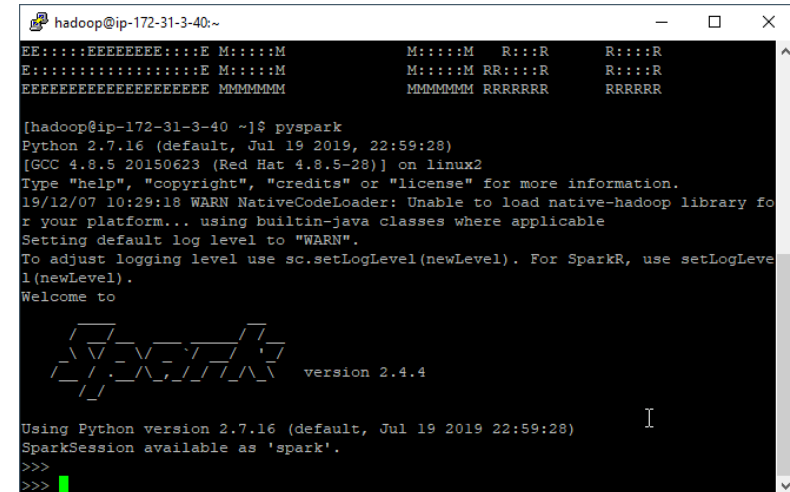
- [] Laissez le champ Destination vide, puis cliquez sur Add.
- [] Cliquez sur Open.
- [] Cliquez sur Yes pour ignorer l'alerte de sécurité.



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

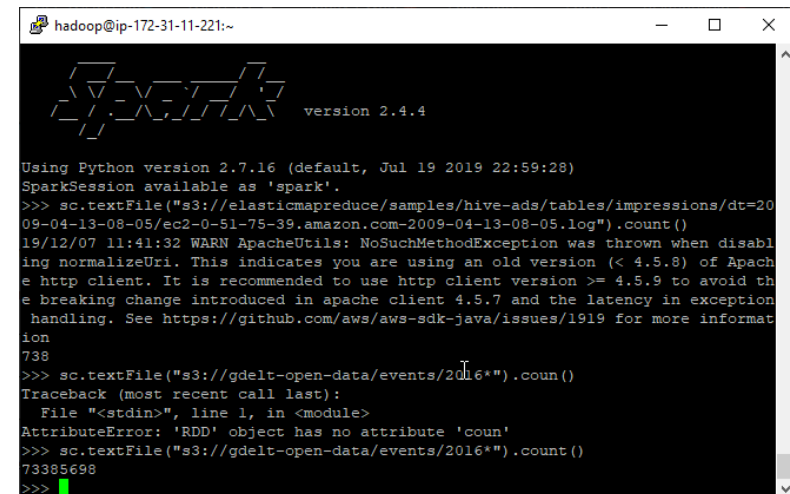
```
pyspark #pour lancer pyspark
spark-shell #pour spark-shell
```

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 ssh établie, et FoxyProxy configuré, vous pouvez désormais accéder aux interfaces de suivi du cluster.

Connexions : [Zeppelin](#), [Serveur d'historique Spark](#), [Ganglia](#), [Gestionnaire de ressources](#) ... (Tout afficher)

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 "Étape"

Cluster : spark **En attente** Cluster ready after last step completed.

Récapitulatif Historique de l'application Surveillance Matériel Configurations Événements **Étapes** Actions d'amorçage

After last step completes: Cluster waits

Ajouter une étape Cloner l'étape Cancel step

Étapes

Afficher tous les travaux interactifs | Afficher tous les travaux

Filter :	Toutes les étapes	Filter les étapes...	2 étapes (toutes chargées)		
ID	Nom	Statut	Heure de début (UTC+1)	Temps écoulé	Les fichiers journaux
s-2AMD69V19J5E0	word_count	Terminé	07-12-2019 13:02 (UTC+1)	6 minutes	contrôleur syslog* stderr stdout*
s-2NM30Y13ZTKFN	Configuration du débogage Hadoop	Terminé	07-12-2019 12:38 (UTC+1)	2 secondes	contrôleur syslog* stderr stdout*

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"

Ajouter une étape

Type d'étape Application Spark

Nom word_count

Mode de déploiement Cluster

Options Spark-submit

Emplacement de l'application* s3://ensaidataapprentissage/script_exemple.py

Arguments

Action sur échec Continuer

Exécutez votre pilote sur un nœud secondaire (mode cluster) ou sur le nœud maître en tant que client externe (mode client).

Spécifiez d'autres options pour spark-submit.

Chemin vers un JAR avec votre application et vos dépendances (le mode de déploiement client prend uniquement en charge un chemin d'accès local).

Spécifiez les arguments facultatifs de votre application.

Que faire en cas d'échec de l'étape.

Annuler Ajouter

Ecran ajout d'une étape

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

After last step completes: Cluster waits

Ajouter une étape Cloner l'étape Cancel step

Étapes

Afficher tous les travaux interactifs | Afficher tous les travaux

Filter : Toutes les étapes Filter les étapes... 3 étapes (toutes chargées)

ID	Nom	Statut	Heure de début (UTC+1)	Temps écoulé	Les fichiers journaux
s-1UETFP1MUK5F	word_count	En suspens		--	Afficher les journaux
s-2AMD69V19J5E0	word_count	Terminé	07-12-2019 13:02 (UTC+1)	6 minutes	Afficher les journaux
s-2NM30Y13ZTKFN	Configuration du débogage Hadoop	Terminé	07-12-2019 12:38 (UTC+1)	2 secondes	Afficher les journaux

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



All Applications

Cluster

About
Nodes
Node Labels
Applications

NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted0Apps Pending1Apps Running4Apps Completed2Containers Running13.38 GBMemory Used24 GBMemory Total0 B

Cluster Nodes Metrics

Active Nodes0Decommissioning Nodes0Decommissioned Nodes0Lost Nodes0Unhealthy Nodes0

Scheduler Metrics

Scheduler TypeScheduling Resource TypeMinimum AllocationMaximum Allocation

Capacity Scheduler[MEMORY]<memory 32, vCores 1><memory 12288, vCores 4>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU / VCore
application_1575718649680_0005	hadoop	script_example.py	SPARK	default	0	Sat Dec 7 13:28:59 +0100 2019	N/A	RUNNING	UNDEFINED	2	2
application_1575718649680_0004	hadoop	script_example.py	SPARK	default	0	Sat Dec 7 13:02:43 +0100 2019	Sat Dec 7 13:09:01 +0100 2019	FINISHED	SUCCEEDED	N/A	N/A

Ecran de gestion hadoop

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattempt_1575718649680_0004_000001	Sat Dec 7 13:02:43 +0100 2019	http://ip-172-31-15-237.eu-west-3.compute.internal:8042	Logs	0	0

Choix des logs

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

Log Type: stdout
 Log Upload Time: Sat Dec 07 12:09:02 +0000 2019

Log Content: 327					
73385698	20060103	200601 2006	2006.0082	BUS	CORPORATION
498554164					

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
```

```
hadoop@ip-172-31-9-22:~
Installing:
  libcurl-devel      x86_64      7.61.1-12.93.amzn1      amzn-updates      855 k

Transaction Summary
=====
Install 1 Package

Total download size: 855 k
Installed size: 1.3 M
Is this ok [y/d/N]: y
Downloading packages:
libcurl-devel-7.61.1-12.93.amzn1.x86_64.rpm | 855 kB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libcurl-devel-7.61.1-12.93.amzn1.x86_64      1/1
  Verifying  : libcurl-devel-7.61.1-12.93.amzn1.x86_64      1/1

Installed:
  libcurl-devel.x86_64 0:7.61.1-12.93.amzn1

Complete!
[hadoop@ip-172-31-9-22 ~]$
```

```
hadoop@ip-172-31-9-22:~
Transaction Summary
=====
Install 1 Package

Total size: 306 M
Installed size: 306 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : rstudio-server-0.99.1266-1.x86_64      1/1
groupadd: group 'rstudio-server' already exists
rsession: no process found
rstudio-server start/running, process 17084
  Verifying : rstudio-server-0.99.1266-1.x86_64      1/1

Installed:
  rstudio-server.x86_64 0:0.99.1266-1

Complete!
[hadoop@ip-172-31-9-22 ~]$
```

- [] 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 \(exemple/script_exemple_R.R\)](#)