

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

## 1. Création d'une clef SSH

**SSH** (Secure **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](#).

- ☐ 1-1 : Dans la barre de recherche, cherchez "EC2" et cliquez dessus
- ☐ 1-2 : Dans le panneau 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 !

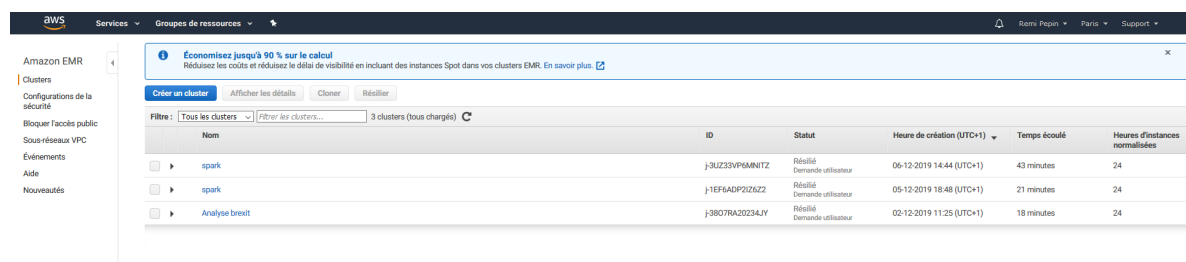
## 2. Conversion au format PPK

- ☐ 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 apparaît 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



The screenshot shows the Amazon EMR console interface. At the top, there's a navigation bar with 'AWS', 'Services', and 'Groupes de ressources'. A sidebar on the left contains links like 'Amazon EMR', 'Clusters', 'Configurations de la sécurité', 'Bloquer l'accès public', 'Sous-réseaux VPC', 'Événements', 'Aide', and 'Nouveautés'. The main content area has a header with a tip about saving up to 90% on Spot instances. Below this, there are buttons for 'Créer un cluster', 'Afficher les détails', 'Cloner', and 'Réinitialiser'. A filter bar shows 'Tous les clusters' and '3 clusters (tous chargés)'. The table below lists three clusters:

Nom	ID	Statut	Heure de création (UTC+1)	Temps écoulé	Heures d'instances normalisées
spark	j-3UJ23VPAINITZ	Réussi Demande utilisateur	06-12-2019 14:44 (UTC+1)	43 minutes	24
spark	j-1EF6ADP2IZGZ2	Réussi Demande utilisateur	05-12-2019 18:48 (UTC+1)	21 minutes	24
Analyse brexit	j-3807RA20234JY	Réussi Demande utilisateur	02-12-2019 11:25 (UTC+1)	18 minutes	24

- ☐ 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.
- ☐ Configuration des logiciels
  - ☐ Laissez la version d'emr par défaut
  - ☐ Sélectionnez comme application Spark
- ☐ Configuration du matériel
  - ☐ Type d'instance : par ex. 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)
- ☐ Sécurité et accès
  - ☐ Sélectionnez une clef SSH que vous avez déjà générée ou allez en générer une autre
  - ☐ Laissez le Rôle EMR et le Profil d'instance par défaut
- ☐ Démarrer le cluster. Le démarrage peut prendre quelques minutes
- ☐ Bravo vous avez démarré un cluster Spark en moins de 15min !

The screenshot shows the Amazon EMR console interface. The main section is titled 'Cluster : Mon cluster' and shows the cluster is in 'En attente' (Pending) state. Below this, there are several tabs: Récapitulatif, Historique de l'application, Surveillance, Matériel, Configurations, Événements, Étapes, and Actions d'amorçage. The 'Récapitulatif' tab is selected, showing details like ID, Date de création, Temps écoulé, Résiliation automatique, Protection de la désactivation, and Récapitulatif. It also shows the configuration details, including the version (emr-5.29.0), Distribution (Amazon), Applications (Ganglia 3.7.2, Spark 2.4.4, Zeppelin 0.8.2), and the network settings (Zone de disponibilité: eu-west-3b, ID de sous-réseau: subnet-638f2118).

- ☐ Avant de continuer, vérifiez si les connexions SSH sont autorisées pour votre cluster. Pour cela allez dans groupe de sécurité pour le principal

security Groups [sg-0b1a6859c333059a9] contain one or more ingress rules to ports other than [22] which allow public access.

The screenshot shows the 'Sécurité et accès' (Security and Access) tab in the Amazon EMR console. It displays the 'Groupe de sécurité' (Security Group) for the principal instance profile. The 'Groupe de sécurité' is 'sg-0b1a6859c333059a9'. The 'Groupe de sécurité' is highlighted with a red box, and the 'Groupe de sécurité' is highlighted with a red box. The 'Groupe de sécurité' is highlighted with a red box, and the 'Groupe de sécurité' is highlighted with a red box.

- Ensuite cliquez sur "ElasticMapReduce-master" et sur l'onglet "entrant" pour vérifier si les connexion SSH sont autorisées

Créer un groupe de sécuritéActions

search : sg-0b1a6859c333059a9Ajouter filtre

	Name	ID du groupe	Nom du groupe	ID de VPC	Propriétaire	Description
<input checked="" type="checkbox"/>	sg-0b1a6859c333059a9		ElasticMapReduce-master	vpc-fdc4eb87	748906290170	Master group for Elastic MapReduce created on 2020-01-21T16:23:49.061Z
<input type="checkbox"/>	sg-0d775f7676b39841a		ElasticMapReduce-slave	vpc-fdc4eb87	748906290170	Slave group for Elastic MapReduce created on 2020-01-21T16:23:49.061Z

Groupe de sécurité: sg-0b1a6859c333059a9

DescriptionEntrantSortantBalises

Modifier

Type	Protocole	Plage de ports	Source	Description
Tous les TCP	TCP	0 - 65535	sg-0b1a6859c333059a9 (ElasticMapReduce-mast	
Tous les TCP	TCP	0 - 65535	sg-0d775f7676b39841a (ElasticMapReduce-slave)	
Règle TCP personnalisée	TCP	8443	207.171.167.25/32	
Règle TCP personnalisée	TCP	8443	54.240.217.8/29	
Règle TCP personnalisée	TCP	8443	72.21.196.64/29	
Règle TCP personnalisée	TCP	8443	72.21.198.64/29	
Règle TCP personnalisée	TCP	8443	54.240.217.16/29	
Règle TCP personnalisée	TCP	8443	54.239.98.0/24	
Règle TCP personnalisée	TCP	8443	207.171.167.101/32	

- Si ce n'est pas le cas cliquez sur "Modifier", allez en bas de la fenêtre qui apparait et ajoutez la règle SSH / n'importe où. Cela vous permettra de vous connecter en SSH à votre cluster depuis n'importe quel ordinateur. Sauvegardez votre changement.

Tous les TCP

TCP

0 - 65535

Personnalis

sg-0d775f7676b39841a

par exemple SSH for Admin Desi

SSH

TCP

22

N'importe où

0.0.0.0/0, ::/0

par exemple SSH for Admin Desi

Ajouter une règle

REMARQUE : Les modifications apportées à des règles existantes se traduiront par la suppression de la règle modifiée et par la création d'une nouvelle règle avec les nouveaux détails. Le trafic lié à cette règle sera alors abandonné pendant un temps très limité jusqu'à ce que la nouvelle règle puisse être créée.

## 4. Accéder à l'interface de suivi du cluster

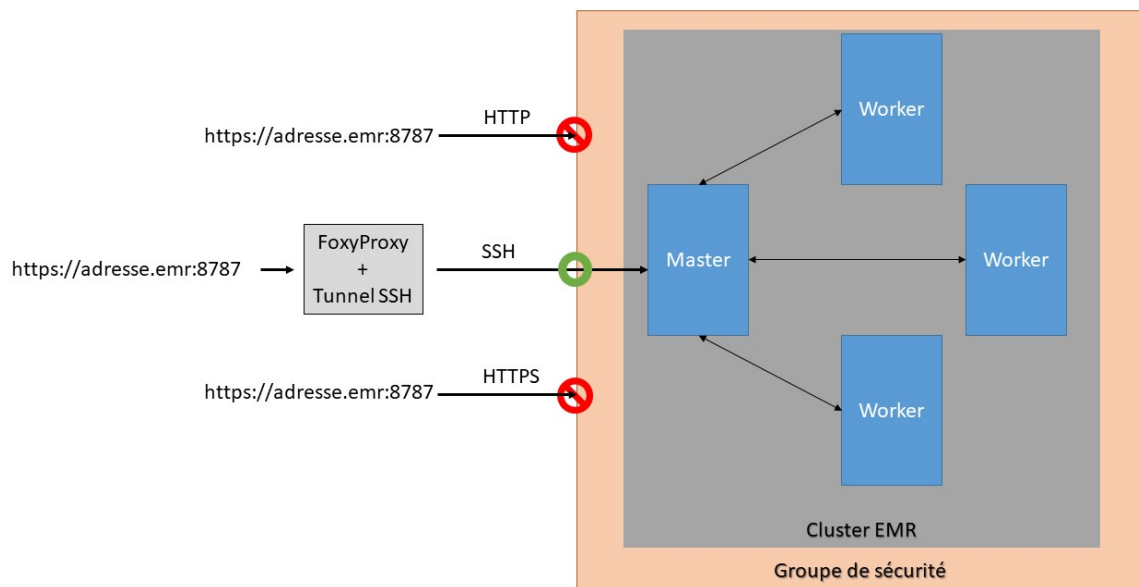
### Un peu de sécurité

Pour des raisons de sécurité, les connexions à votre cluster depuis l'extérieur sont limitées aux connexions SSH. Même s'il est possible d'autoriser plus de connexions via les "groupes de sécurité" d'aws, votre cluster ne répondra qu'à la requête SSH. Problème, votre navigateur internet ne sait pas faire des requête SSH. Et tel quel il vous ait impossible d'accéder aux interface web de votre cluster. Cela est déjà un inconvénient sérieux, mais surtout cela cela vous empêche de vous connecter avec R à votre cluster.

Pour remédier à cela nous allons faire deux choses :

- Créer un tunnel SSH entre votre ordinateur et le cluster. Cela permettra à votre ordinateur de faire passer certaines requête dans la connexion SSH établie entre le cluster et vous. Par exemple quand vous accéderez aux interfaces graphiques du cluster cela se fera via l'intermédiaire du tunnel. Le tunnel prendra toutes les requêtes faites pour l'adresse localhost:8157 pour les transmettre aux cluster.
- Installer FoxyProxy. Cette extension de navigateur permet de faire de la redirection de requête pour utiliser des proxys à la volée. Le fonctionnement est le suivant, vous paramétrez des motifs URL qui doivent être redirigés vers un certain proxy. Quand le motif est repéré, FoxyProxy redirige la requête vers le proxy associé aux motifs. Dans le cas présent, le proxy sera localhost:8157, le point d'entrée de notre tunnel.

Cette procédure n'est en aucun cas un "hack" de notre part pour accéder à des services protégés, mais bien la marche à suivre officielle proposé par amazon. **Il est obligatoire de la respecter pour pouvoir utiliser R avec votre cluster**



## Installer FoxyProxy

Pour google chrome : [lien](#)

Pour firefox: [lien](#)

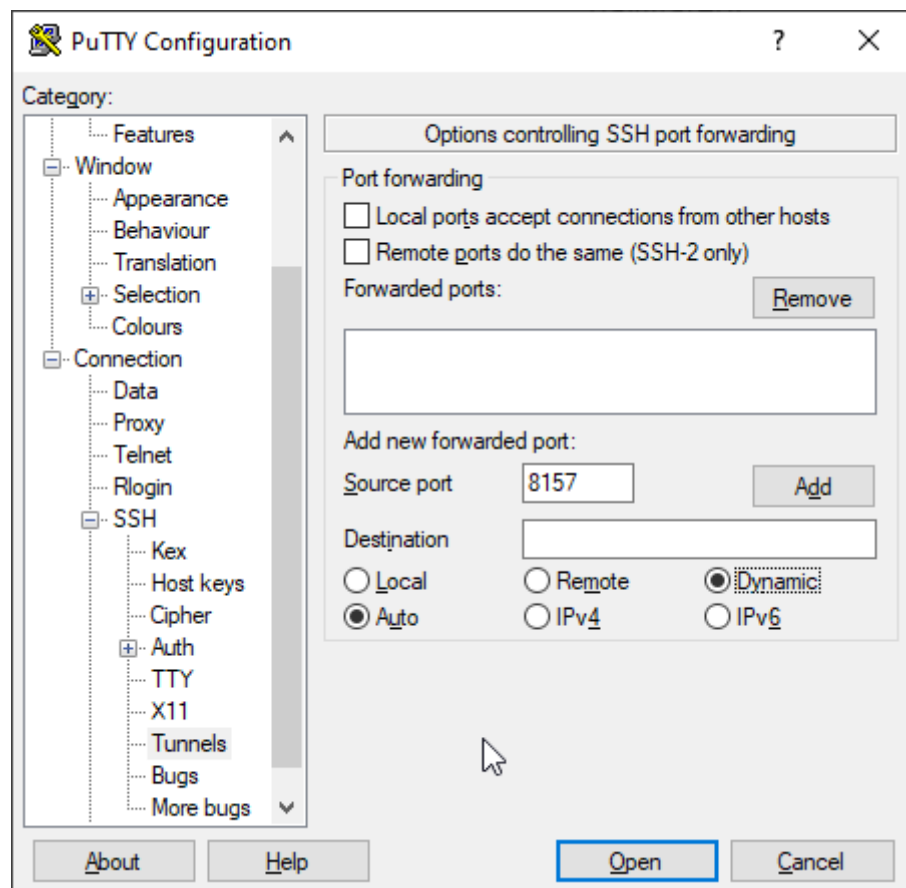
Une fois FoxyProxy installé, ouvrez le plugin et importer le fichier se trouvant dans :

- <https://github.com/katossky/panorama-bigdata/blob/master/settings/foxyproxy-settings.json> pour firefox
  - <https://github.com/katossky/panorama-bigdata/blob/master/settings/foxyproxy-settings.xml> pour chrome

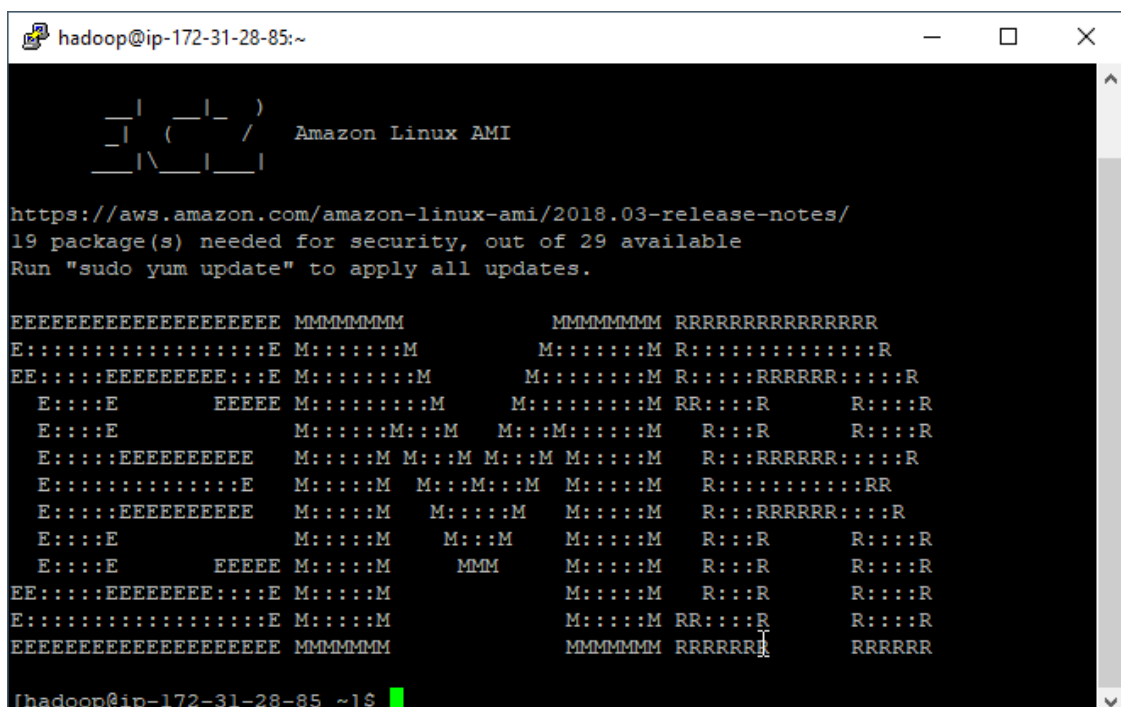
## Etablir une connexion SSH avec votre cluster

*(La marche à suivre est également disponible si vous cliquez sur "activez la connexion web" depuis la page de votre cluster)*

- ☐ Lancez PuTTY
- ☐ Dans la liste Category, cliquez sur Session
- ☐ Dans le champ Host Name, tapez **hadoop@[DNS public]** avec [DNS public] 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.



- ☐ 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.*

```
hadoop@ip-172-31-3-40:~  
EE::::EEEEEEEE::::E M::::M      M::::M      R:::R      R::::R  
E::::::::::::::::::E M::::M      M::::M      RR::::R      R::::R  
EEEEEEEEEEEEEEEEEEEE MMMMMM      MMMMMM      RRRRRRR      RRRRRR  
  
[hadoop@ip-172-31-3-40 ~]$ pyspark  
Python 2.7.16 (default, Jul 19 2019, 22:59:28)  
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
19/12/07 10:29:18 WARN NativeCodeLoader: Unable to load native-hadoop library fo  
r your platform... using builtin-java classes where applicable  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve  
l(newLevel).  
Welcome to  
  
          version 2.4.4  
  
Using Python version 2.7.16 (default, Jul 19 2019 22:59:28)  
SparkSession available as 'spark'.  
>>>  
>>>
```

- ☐ 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
```

```
hadoop@ip-172-31-11-221:~  
          version 2.4.4  
  
Using Python version 2.7.16 (default, Jul 19 2019 22:59:28)  
SparkSession available as 'spark'.  
>>> sc.textFile("s3://elasticmapreduce/samples/hive-ads/tables/impressions/dt=20  
09-04-13-08-05/ec2-0-51-75-39.amazon.com-2009-04-13-08-05.log").count()  
19/12/07 11:41:32 WARN ApacheUtils: NoSuchMethodException was thrown when disabl  
ing normalizeUri. This indicates you are using an old version (< 4.5.8) of Apach  
e http client. It is recommended to use http client version >= 4.5.9 to avoid th  
e breaking change introduced in apache client 4.5.7 and the latency in exception  
handling. See https://github.com/aws/aws-sdk-java/issues/1919 for more informat  
ion  
738  
>>> sc.textFile("s3://gdelt-open-data/events/2016*").count()  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
AttributeError: 'RDD' object has no attribute 'coun'  
>>> sc.textFile("s3://gdelt-open-data/events/2016*").count()  
73385698  
>>>
```

Voici le même script scala en plus condensé

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

- ☐ 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.

![Accueil Pyspark](../img/setup-emr/pyspark\_emr.png)

- ☐ 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://gdelit-open-data/events/2016*")
>>> textfile.count()
73385698
```

![Résultat pyspark](../img/setup-emr/pyspark\_script.png)

Voici le même script scala en plus condensé

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

## Ouvrir les interfaces de suivi

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

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

## 5. Lancer un job avec un script

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

Cloner Résilier Exporter AWS CLI

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](#)

	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	<a href="#">contrôleur</a>   <a href="#">syslog*</a>   <a href="#">stderr</a>   <a href="#">stdout*</a>
▶	s-2NM30Y13ZTKFN	Configuration du débogage Hadoop	Terminé	07-12-2019 12:38 (UTC+1)	2 secondes	<a href="#">contrôleur</a>   <a href="#">syslog*</a>   <a href="#">stderr</a>   <a href="#">stdout*</a>

- ☐ 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\_example.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

- ☐ 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

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-2NM30YI3ZTKFN	Configuration du débogage Hadoop	Terminé	07-12-2019 12:38 (UTC+1)	2 secondes	Afficher les journaux

- ☐ 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 Submitted: 5, Apps Pending: 0, Apps Running: 1, Apps Completed: 4, Containers Running: 2, Memory Used: 13.38 GB, Memory Total: 24 GB, Memory: 0 B
Cluster Nodes Metrics
Active Nodes: 2, Decommissioning Nodes: 0, Decommissioned Nodes: 0, Lost Nodes: 0, Unhealthy Node: 0
Scheduler Metrics
Capacity Scheduler, Scheduling Resource Type: [MEMORY], Minimum Allocation: <memory:32, vCores:1>, Maximum Allocation: <memory:12288, vCores:4>
Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocat 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



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

- ☐ 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 Length: 321
73385698
498554164      20060103      200601  2006   2006.0082      BUS      CORPORATION
```

## 6. Se connecter avec Rstudio et sparklyR

- ☐ Connectez-vous en SSH à votre cluster EMR (vous pouvez réutiliser la connexion avec le tunnel faite plus tôt)
- ☐ Installez Rstudio server

```
sudo yum install libcurl-devel openssl-devel # used for devtools
wget https://download2.rstudio.org/server/centos6/x86_64/rstudio-server-rhel-1.2.5033-x86_64.rpm
sudo yum install rstudio-server-rhel-1.2.5033-x86_64.rpm
```

![yum install](../img/setup-emr/rstudio\_yum\_install.png)

![r server install](../img/setup-emr/rstudio\_server\_install.png)

- ☐ 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 avec [master-node-public-DNS] le DNS public de votre cluster. 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](#)

<!--

## Liens utiles

- [Documentation officielle spark EMR](#)
- [Getting Started with PySpark on AWS EMR](#)
- [Creating PySpark DataFrame from CSV in AWS S3 in EMR](#)
- [Connection avec Rstudio](#)

-->