Comment faire fonctionner le cluster

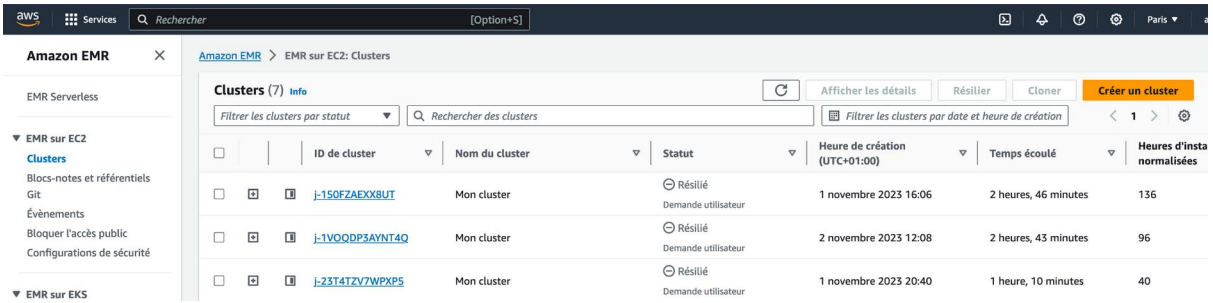
Créer/reproduire le cluster EMR

Pour créer un cluster avec EMR :

- aller sur la page EMR en tapant « EMR » dans la barre de recherche
- choisir une option parmi celles proposées ; pour notre projet, nous avons opté pour un EMR s'exécutant sur Amazon EC2
- cliquer sur « Créer un cluster »

S'ouvre alors la page de configuration de l'EMR.

Pour reproduire un EMR précédemment créé puis résilié, aller dans la rubrique EMR sur EC2 > Clusters (cf. image ci-dessous), cocher le cluster que l'on veut reproduire, puis cliquer sur cloner. S'ouvre alors la page de configuration de l'EMR.



Pour un EMR identique à celui que j'ai créé, choisir les réglages suivants :
Attention : lorsque l'on tente de cloner un cluster qui avait été créé avec ces réglages, certains se remettent automatiquement sur les valeurs par défaut et doivent être redéfinis. J'ai coloré les réglages en question en rouge.

- version d'EMR : emr-6.14.0
  > La version la plus récente au moment où j'ai créé mon EMR. Lorsque de nouvelles versions seront disponibles, il est possible que ce réglage fasse partie de ceux qui reviennent automatiquement sur une autre valeur au moment de cloner le cluster.
- choisir l'offre « Custom » et sélectionner les packages Spark, Hadoop, JupyterHub et TensorFlow
- comme option du système d'exploitation, choisir « Version Amazon Linux »
- configuration du cluster : sélectionner « Groupes d'instances »
  et « m5.xlarge » pour le type d'instance EC2
- dimensionnement et mise en service du cluster : sélectionner « Définir manuellement la taille du cluster »
  taille de l'instance(s) : entrer « 1 » pour l'unité principale, et « 2 » pour la tâche 1.

- Pour l'amorçage, charger sur S3, dans le bucket du projet, un fichier .sh qui contienne ces lignes :

```
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas==1.2.5
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
sudo python3 -m pip install jupyterlab
sudo python3 -m pip install jupyter
sudo python3 -m pip install notebook
sudo python3 -m pip install pyspark
sudo python3 -m pip install tensorflow
```

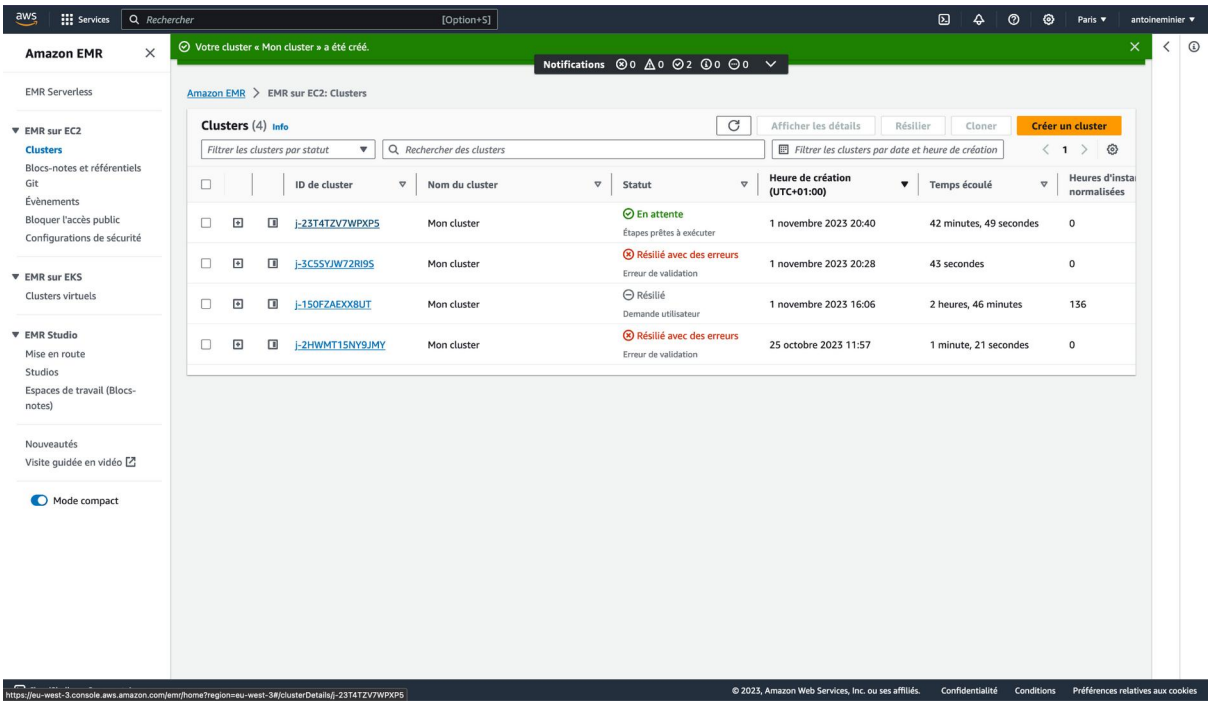
  Puis, dans la rubrique Actions d'amorçage de la page de configuration du cluster, cliquer sur « Ajouter », donner un nom au fichier d'amorçage et entrer son emplacement sur S3 (s3://nom-du-bucket/nom-du-fichier si le fichier se trouve au niveau racine, dans aucun dossier).
- Paramètres logiciels : sélectionner « Entrer la configuration » et entrer le code suivant dans l'espace dédié :

```
[
  {
        "Classification": "jupyter-s3-conf",
        "Properties": {
        "s3.persistence.bucket": "aminier-p8",
        "s3.persistence.enabled": "true"
        }
  }
]
```

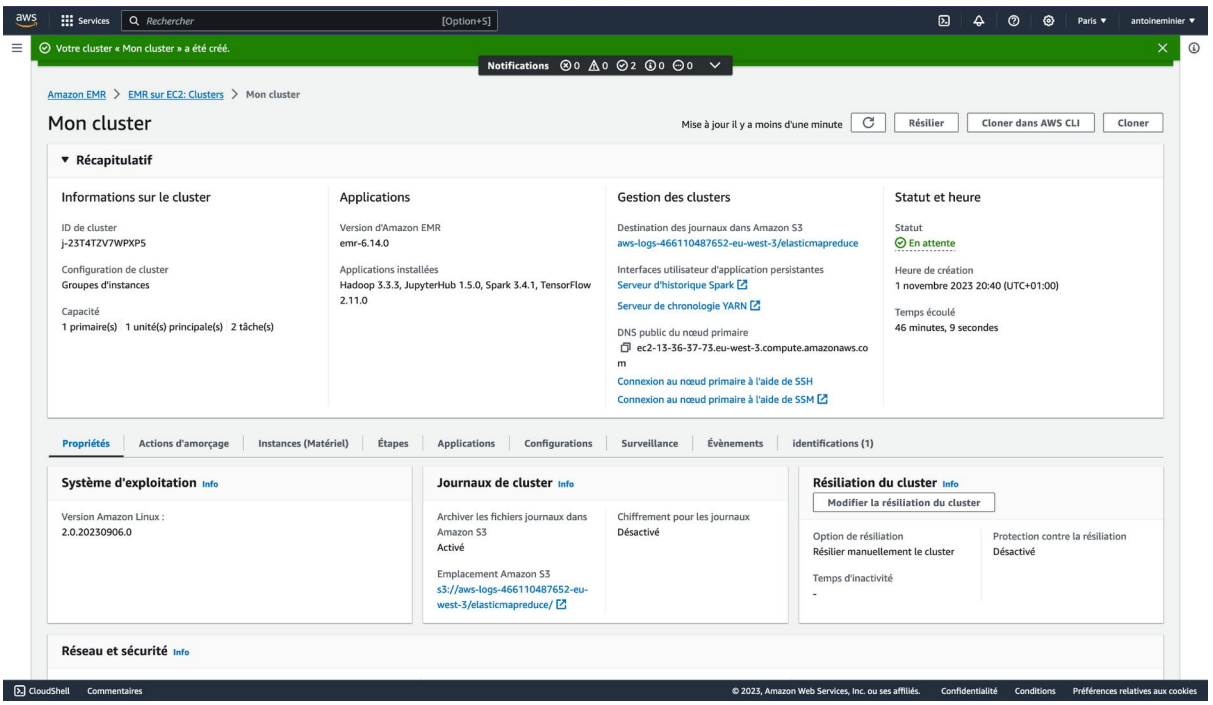
- Configuration de sécurité et paire de clés EC2 : dans « Paire de clés Amazon EC2 pour SSH sur le cluster » cliquer sur parcourir et sélectionner la paire de clés que l'on aura préalablement créée (pour cela, suivre les indications de la rubrique dédiée du cours Réalisez des calculs distribués sur des données massives, chapitre Déployez un cluster de calculs distribués)
- Rôle Identity and Access Management (IAM) :
  Dans « Fonction du service Amazon EMR », sélectionner « Créez un fonction du service » et non pas « Choisir une fonction du service existant ».
  Puis dans « Profil d'instance EC2 pour Amazon EMR », sélectionner « Choisir un profil d'instance » et non pas « Choisir un profil d'instance existant ». Juste en-dessous, dans « Accès au compartiment S3 », sélectionner « Tous les compartiments S3 de ce compte avec accès en lecture et en écriture ».

Utiliser le cluster EMR une fois qu'il est activé

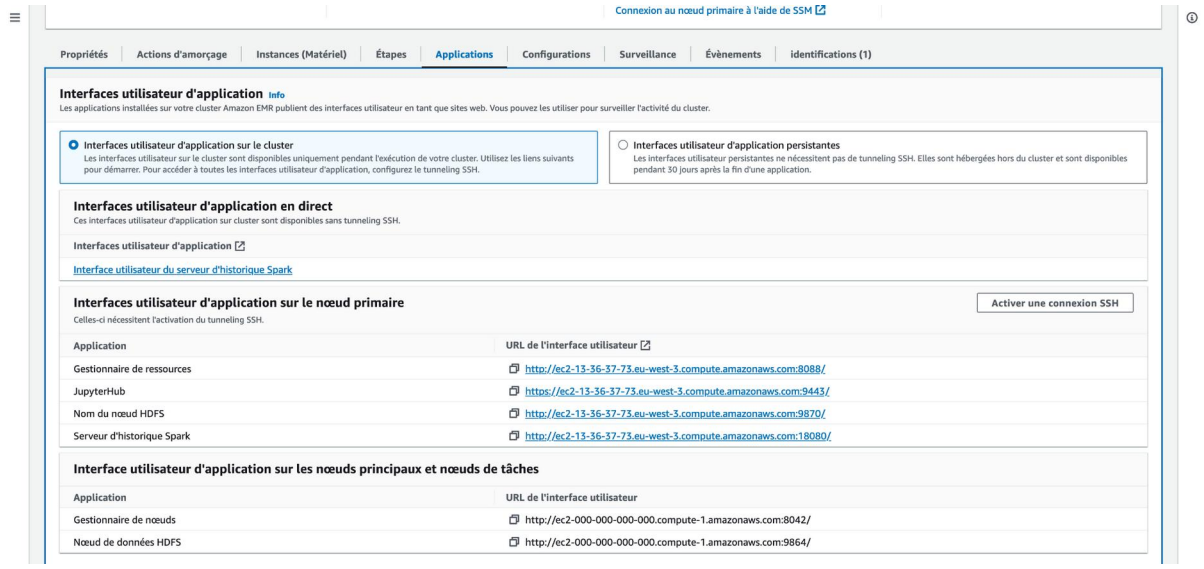
Sur la page des clusters EMR (cf. l'image ci-dessous), cliquer sur l'ID du cluster que l'on veut utiliser.



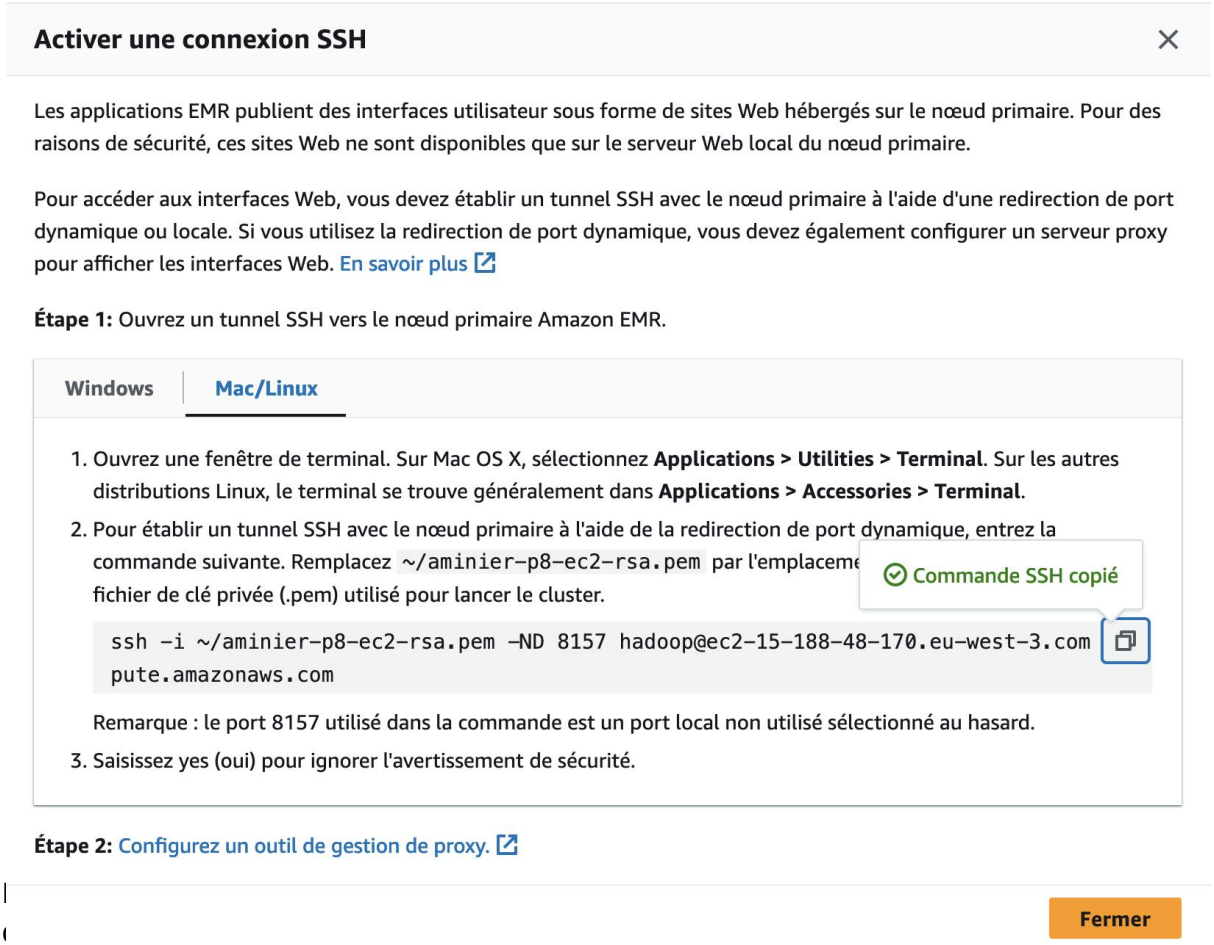
On arrive sur la page du cluster (cf. l'image ci-dessous).



Par défaut, on se trouve dans l'onglet « Propriétés » ; cliquer sur l'onglet « Applications ». Descendre à la rubrique « Interfaces utilisateur d'application sur le nœud primaire », puis cliquer sur “Activer une connexion SSH” en haut à droite de la rubrique.



Copier la commande SSH surlignée en gris.



contenant la paire de clé pour la connexion SSH.

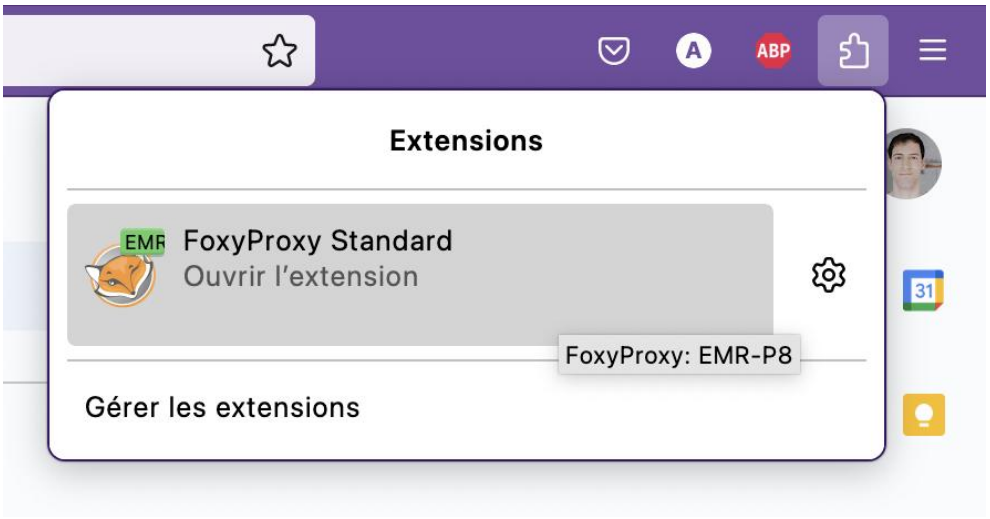
Puis coller la commande précédemment récupérée d'Amazon pour la connexion SSH. Enlever de cette commande ~/ et le N dans -ND pour n'avoir plus, dans notre exemple, que :

ssh -i aminier-p8-ec2-rsa.pem -D 8157 hadoop@ec2-13-36-37-73.eu-west-3.compute.amazonaws.com

Changer aussi le port selon ce que l'on souhaite ; par exemple 5000 à la place de 8157. Exécuter la commande, répondre yes à la question qui s'affiche en retour.

Si l'on utilise Firefox, installer l'extension FoxyProxy, et la régler selon les indications données dans le notebook donné en ressource pour le P8 (si l'on n'utilise pas Firefox, utiliser un autre proxy). Indiquer le même port que celui que l'on aura choisi d'utiliser pour la commande SSH.

Une fois installée et réglée, dans Firefox, dans les extensions, cliquer sur FoxyProxy.



Puis activer le proxy en cliquant sur le nom que l'on aura défini lors du réglage, en vert (dans l'exemple, « EMR-P8 »).

Revenir ensuite sur la page du cluster, et cliquer sur le lien de l'url qui correspond à JupyterHub.
Passer les avertissements de sécurité.
S'ouvre alors l'interface JupyterHub avec le contenu du dossier Jupyter de notre bucket S3.
Si des identifiants sont d'abord demandés, entrer les identifiants par défaut : XXXX en nom d'utilisateur, et jupyter en mot de passe.

Une fois le cluster résilié, désactiver le proxy.