Table des matières

[I. Résumé 3](#_Toc171425692)

[I.1. Résumé du déploiement de la solution sur le cloud AWS 3](#_Toc171425693)

[I.2. Résumé de la démarche 4](#_Toc171425694)

[Points clés 5](#_Toc171425695)

[II. Upload de nos données sur S3 6](#_Toc171425696)

[II.1. Création d’un compartiment S3 6](#_Toc171425697)

[II.2. Ajout de documents dans S3 7](#_Toc171425698)

[II.3. Fichier Bootstrap-emr.sh 8](#_Toc171425699)

[III. EC2 9](#_Toc171425700)

[III.1. Règles de sécurité 9](#_Toc171425701)

[III.2. Modifier les permissions de la paire de clé 10](#_Toc171425702)

[IV. IAM 11](#_Toc171425703)

[V. EMR 12](#_Toc171425704)

[V.1. Nom et applications 12](#_Toc171425705)

[V.2. Configuration de cluster 14](#_Toc171425706)

[V.3. Dimensionnement et mise en service du cluster 16](#_Toc171425707)

[V.4. Réseaux 17](#_Toc171425708)

[V.5. Etapes 18](#_Toc171425709)

[V.6. Résiliation du cluster et remplacement des nœuds 19](#_Toc171425710)

[V.7. Actions d’amorçage 20](#_Toc171425711)

[V.8. Journaux de Cluster 21](#_Toc171425712)

[V.9. Paramètres du logiciel 22](#_Toc171425713)

[V.10. Configuration de sécurité et paire de clés EC2 23](#_Toc171425714)

[V.11. Rôle Identity and Access Management 24](#_Toc171425715)

[V.12. Création du cluster EMR 25](#_Toc171425716)

[VI. Création d’un tunnel SSH pour le maitre 27](#_Toc171425717)

[VII. FoxyProxy 29](#_Toc171425718)

[VIII. Se connecter à JupyterHub 31](#_Toc171425719)

[IX. Problème mémoire 34](#_Toc171425720)

[IX.1. Solution au problème de mémoire (HARDWARE) 35](#_Toc171425721)

[IX.2. Solution au problème de mémoire (SOFTWARE) 36](#_Toc171425722)

[IX.3. Solution au problème de mémoire (DATA) 36](#_Toc171425723)

[IX.4. Note concernant le problème de mémoire 36](#_Toc171425724)

[X. Résultats 37](#_Toc171425725)

[X.1. Chaine entière du traitement (SparkUI) 37](#_Toc171425726)

[X.2. Extraction des features – DAG SparkUi 37](#_Toc171425727)

[X.3. Visualisation SparkUI pour l’extraction des features 39](#_Toc171425728)

# I. Résumé

## I.1. Résumé du déploiement de la solution sur le cloud AWS

**1. Prestataire cloud:**

* Choix d'**Amazon Web Services (AWS)** pour son large éventail de services cloud.

**2. Solution technique:**

* Utilisation d'**Amazon EMR (Elastic MapReduce)** pour simplifier la configuration et la gestion du cluster Spark.
  + Avantages : Facilité et rapidité de mise en œuvre, solution optimisée et évolutive, sécurisée.
  + Inconvénients : Moins de flexibilité sur les versions des packages.

**3. Stockage des données:**

* Utilisation d'**Amazon S3 (Simple Storage Service)** pour un stockage de données fiable et illimité.
  + Avantages : Espace disque illimité, indépendant des serveurs, accès rapide, facile d'utilisation.
  + Inconvénient : Aucun mentionné.

**En résumé, le choix d'AWS EMR et S3 offre une solution cloud robuste, évolutive et facile à gérer pour le déploiement à grande échelle du traitement d'images.**

**Points clés:**

* AWS offre une puissance de calcul à la demande pour une flexibilité et une optimisation des coûts.
* EMR simplifie la configuration et la gestion du cluster Spark.
* S3 offre un stockage de données fiable, illimité et accessible.

**Cette approche permet un déploiement rapide, efficace et sécurisé de la solution, tout en étant capable de s'adapter à l'évolution des besoins en matière de traitement d'images.**

## I.2. Résumé de la démarche

**I. Préparation**

1. Créer un compartiment S3 et y stocker les fichiers nécessaires :
   * Notebook Jupyter
   * Bootstrap-emr.sh (contenant les librairies à installer)
   * Dossier Test (contenant les sous-dossiers et images)
2. Créer une paire de clés EC2 (.pem pour Linux, .ppk pour Windows)
3. Créer des rôles IAM avec les politiques suivantes :
   * AmazonEC2FullAccess
   * AmazonElasticMapReduceFullAccess
   * AmazonS3FullAccess

**II. Création du cluster EMR**

1. Sélectionner la version EMR 6.13.
2. Cocher les options Spark, JupyterHub et TensorFlow.
3. Choisir le type de machine et le nombre d'instances (1 pour commencer).
4. Indiquer la localisation du fichier bootstrap-emr.sh dans S3.
5. Configurer les journaux de cluster (optionnel).
6. Ajouter le JSON dans Paramètres du logiciel  
   Sélectionner la paire de clés EC2 créée précédemment.
7. Choisir les rôles IAM créés précédemment.
8. Lancer la création du cluster.

**III. Connexion à JupyterHub**

1. Obtenir le nom d'hôte du nœud principal du cluster EMR.
2. Créer un tunnel SSH vers le port 8158 du nœud principal en utilisant PuTTY et la paire de clés EC2.
3. Installer l'extension FoxyProxy dans votre navigateur et la configurer pour utiliser le proxy SOCKS5 sur localhost:8158.
4. Se connecter à JupyterHub en utilisant le lien fourni dans l'onglet "Applications" du cluster EMR.
5. Utiliser les informations d'identification par défaut :
   * Nom d'utilisateur : jovyan
   * Mot de passe : jupyter
6. Ouvrir votre notebook Jupyter et modifier le kernel pour utiliser le KERNEL PYSPARK.

**Notes importantes**

* Assurez-vous de conserver la paire de clés EC2 en lieu sûr, car elle est nécessaire pour accéder au nœud principal du cluster.
* N'oubliez pas de désactiver le cluster EMR lorsque vous ne l'utilisez pas pour éviter les frais inutiles.
* Vous pouvez modifier et étendre ce cluster en fonction de vos besoins en calcul.

## Points clés

* Ce résumé fournit les étapes essentielles pour créer un cluster EMR et utiliser JupyterHub sur AWS.
* Des instructions détaillées pour chaque étape peuvent être trouvées dans la documentation officielle d'Amazon EMR.
* Il est important de bien comprendre les concepts de base du cloud computing et d'AWS avant de commencer.
* N'hésitez pas à consulter des ressources en ligne et des tutoriels pour vous aider à démarrer.

# II. Upload de nos données sur S3

## II.1. Création d’un compartiment S3

Une image contenant texte, logiciel, Icône d’ordinateur, Logiciel multimédia

Description générée automatiquement

**Compartiment créé :** oc-p9-data-olivier-raymond (**Paramètres par défaut conservés**)

**Remarques importantes**

* Assurez-vous que le nom de votre compartiment est unique et respecte les **règles de nommage d'Amazon S3**. Les noms de compartiment doivent être constitués de 3 à 63 caractères alphanumériques minuscules, de tirets (-) ou de points (.). Ils ne peuvent pas commencer ou se terminer par un tiret ou contenir des tirets consécutifs.
* Pour plus d'informations sur les compartiments S3, veuillez consulter la documentation officielle d'Amazon S3 : <https://docs.aws.amazon.com/s3/>

## II.2. Ajout de documents dans S3

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Assurez-vous que votre compartiment S3 contient les éléments suivants :

* **Notebook Jupyter** : Pour l'exécution du code et l'analyse des données.
* **Bootstrap-emr.sh** : Un script contenant les bibliothèques requises (pandas, numpy, etc.) pour vos machines.
* **Dossier Test** : Ce dossier doit contenir tous vos sous-dossiers et images.

**Remarque importante :** Ne stockez jamais vos clés d'accès (cle.pem ou cle.ppk) dans le compartiment S3, car cela représente une faille de sécurité majeure.

## II.3. Fichier Bootstrap-emr.sh

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Le fichier Bootstrap-emr.sh** est un script Bash (format .sh pour Linux) qui permet d'installer les bibliothèques nécessaires et d'exécuter des commandes spécifiques sur vos machines.

**Personnalisation du fichier Bootstrap-emr.sh**

N'hésitez pas à adapter ce fichier en incluant des bibliothèques supplémentaires, telles que Matplotlib si vous avez besoin de générer des graphiques.

# III. EC2

## III.1. Règles de sécurité

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Avant de créer un Cluster EMR, il faut créer des règles de sécurité sur l’EC2. Dans le pannel de gauche, choisir **Réseau et sécurité >> Paires de clés.**

Pour la création de la clé, choisir :   
- Type de paire de clés : RSA  
- Format de Fichier de clé privée : .pem pour linux, .ppk pour Windows  
- Pas de balise.   
Quand la clé est créée, il faut la stocker. On l’utilisera plus tard.

**Avant de créer un cluster EMR, il est nécessaire de configurer des règles de sécurité au niveau d'EC2.**

**1. Création d'une paire de clés**

* Accédez à la console AWS et sélectionnez **Réseau et sécurité** > **Paires de clés** dans le panneau de navigation gauche.
* Cliquez sur **Créer une paire de clés**.
* Pour **Type de paire de clés**, choisissez **RSA**.
* Pour **Format de fichier de clé privée**, choisissez .pem pour Linux ou .ppk pour Windows.
* Laissez le champ **Balise** vide.
* Cliquez sur **Créer la paire de clés**.

**Points importants**

* Conservez votre clé privée en lieu sûr. Ne la partagez avec personne et ne la stockez pas dans le cloud.
* Le type de fichier de clé privée doit correspondre à votre système d'exploitation (PEM pour Linux, PPK pour Windows).
* Vous pouvez créer plusieurs paires de clés si nécessaire.

## III.2. Modifier les permissions de la paire de clé

Il est nécessaire de restreinte les permissions (read, write, etc.) de la paire de clé pour éviter l’erreur : « Permissions 0644 for 'path\_to\_KEY’ are too open. It is required that your private key files are NOT accessible by others. »

* Utilisation de la commande : **!**chmod 600 path\_to\_KEY

# IV. IAM

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

**Avant de créer un cluster EMR, il est nécessaire de configurer des rôles IAM et de leur attribuer les politiques adéquates.**

**1. Création des rôles IAM**

* Créez deux rôles IAM :
  + serviceRole pour la fonction de service Amazon EMR
  + InstanceProfile pour le profil d'instance EC2 pour Amazon EMR
* **Ne pas utiliser les rôles par défaut** créés lors de la première tentative de déploiement du cluster EMR, car ils manquent des politiques requises.

**2. Attribution des politiques aux rôles**

* Attribuez les politiques suivantes à **chaque** rôle :
  + AmazonEC2FullAccess
  + AmazonElasticMapReduceFullAccess
  + AmazonS3FullAccess

**Explication des politiques**

* AmazonEC2FullAccess donne au rôle un contrôle total sur les ressources EC2, y compris la création, la modification et la suppression des instances.
* AmazonElasticMapReduceFullAccess donne au rôle un contrôle total sur les clusters EMR, y compris la création, la modification et la suppression des clusters.
* AmazonS3FullAccess donne au rôle un contrôle total sur les compartiments S3, y compris le téléchargement, le téléchargement et la suppression d'objets.

**Points importants**

* N'accordez que les autorisations nécessaires à vos rôles. Attribuer des accès excessifs peut compromettre la sécurité de votre environnement AWS.
* Vous pouvez créer des rôles IAM supplémentaires avec des ensembles de politiques plus restreints pour des cas d'utilisation spécifiques.

# V. EMR

## V.1. Nom et applications

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

**Points importants :**

* La version 6.13.0 d'EMR est sélectionnée car elle a fait ses preuves et est stable (voir messages sur Discord OC (P9)).
* Les applications Spark, JupyterHub et TensorFlow sont sélectionnées car elles sont essentielles pour le cas d'utilisation prévu.
* Les autres applications sont désélectionnées pour éviter de surcharger le cluster avec des ressources inutilisées.

**Remarque :** N'oubliez pas de mettre à jour la version d'EMR si nécessaire.

## V.2. Configuration de cluster

Une image contenant texte, capture d’écran, nombre, Parallèle

Description générée automatiquement

C’est dans cette partie là que nous choisissons le **type de machine** que l’on souhaite. Nous avons besoin d’une machine maître (PRIMAIRE) et d’une ou plusieurs machines esclaves (UNITE PRINCIPALES).

**NOTE : Le choix du type de machine influence le prix à payer.   
Voir la doc pour choisir !**

**NOTE : La localisation des machines influence le prix à payer.**

🡪 Il faut faire attention à la localisation. Perso j’ai choisi Irlande.

**NOTE : La localisation du S3 et du cluster EMR doit être la même**



## V.3. Dimensionnement et mise en service du cluster

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Nombre d'instances d'ouvriers**

* Définissez le nombre d'instances d'ouvriers (Worker) nécessaires pour exécuter votre charge de travail.
* Pour commencer et faciliter le débogage sur AWS, commencez avec une seule instance.
* Augmentez ensuite le nombre d'instances en fonction des besoins de performance et de scalabilité de votre application.
* Envisagez l'utilisation de la mise à l'échelle automatique pour ajuster dynamiquement le nombre d'instances en fonction de la charge de travail.

**Points importants**

* Le nombre d'instances d'ouvriers **impacte directement le coût** du cluster.
* Débutez avec un petit nombre d'instances et **augmentez progressivement** si nécessaire.
* La mise à l'échelle automatique permet d'**optimiser les coûts et les performances**.

## V.4. Réseaux

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Garder les valeurs par defaults.

## V.5. Etapes

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Idem, ne pas modifier.

## V.6. Résiliation du cluster et remplacement des nœuds

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**NOTE : Le choix de la durée de résiliation influence le prix à payer.   
Je recommande de résilier après 20 mins.**

## V.7. Actions d’amorçage

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Il s’agit d’indiquer la localisation du fichier bootstrap-emr.sh. Dans mon cas, je l’avais placé dans mon bucket s3, dans le compartiment oc-p9-data-olivier-raymond.

## V.8. Journaux de Cluster

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

J’ai gardé les valeurs par default. Vous pouvez le mettre dans votre bucket S3.   
Je me retrouve avec 2 compartiments dans mon S3 du coup.

## V.9. Paramètres du logiciel

Une image contenant texte, logiciel, Page web, Police

Description générée automatiquement

Il faut rajouter le json suivant :

[

{

"Classification": "jupyter-s3-conf",

"Properties": {

"s3.persistence.bucket": "oc-p9-data-olivier-raymond",

"s3.persistence.enabled": "true"

}

}

]

## V.10. Configuration de sécurité et paire de clés EC2

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

C’est ici que l’on indique où se situe la paire de clé de sécurité que l’on a crée dans EC2.

Pour rappel :   
- .ppk si on souhaite utiliser le Jupiter AWS sur Windows   
- .pem pour linux.

Perso, je continue sur Windows ! J’en ai marre de linux et de la machine virtuelle !

## V.11. Rôle Identity and Access Management

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

C’est ici que l’on choisit les rôles avec les politiques que l’on a créé précédemment dans IAM.

## V.12. Création du cluster EMR

Cliquer sur le bouton « Créer un cluster EMR ».

Si tous se passe bien, vous devez avoir le récapitulatif suivant.   
Ce qu’il faut regarder est le status du cluster EMR. Il va passer de « Démarrage en cours » >> « Amorçage » >> « En attente ».

Cette action peut prendre en 5 et 15 mins.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

Une image contenant texte, Police, nombre, logiciel

Description générée automatiquement

A noter qu’il y a un bouton « Connexion au nœud primaire à l’aide de SSH » qui explique les étapes de l’étape suivante (et l’obtention du HostName qui est [hadoop@ec2-52-49-124-200.eu-west-1.compute.amazonaws.com](mailto:hadoop@ec2-52-49-124-200.eu-west-1.compute.amazonaws.com) dans mon cas)

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

# VI. Création d’un tunnel SSH pour le maitre

Sur Windows, il faut télécharger l’outil PuTTY

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Il y a plusieurs choses à paramétrer :

* + Dans Sessions >> Host Name   
    Rajouter le hostname [hadoop@ec2-52-49-124-200.eu-west-1.compute.amazonaws.com](mailto:hadoop@ec2-52-49-124-200.eu-west-1.compute.amazonaws.com)
  + Dans Connexion >> SSH > Auth > Crédentials  
    Rajouter le lien vers la paire de clé de sécurité (qui est stocké sur votre PC)
  + Dans Connexion >> SSH >> Tunnels   
    Rajouter le source PORT 8158 (ne rien mettre dans destination), cliquer sur Add  
    Modifier pour Dynamic et Auto.

Finalement cliquer sur Open.   
Normalement, une Securit Alert se lance, cliquer sur Accept.

**Note : Si le port 8158 ne marche pas, en prendre un autre qui n’est pas utilisé.**

Une image contenant texte, capture d’écran, logiciel, ordinateur

Description générée automatiquement

Et nous avons créé le tunnel SSH pour la machine maitre (EMR) !!

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

# VII. FoxyProxy

Vous devez installer l’addon FoxyProxy Standard sur votre browser (disponible sur chrome, firefox etc.)

Une image contenant texte, capture d’écran, logiciel, multimédia

Description générée automatiquement

Aller dans Options puis dans Proxies >> Ajouter.  
Choisir   
- Type = SOCKS5  
- HostName = localhost  
- Port = 8158 **(ce port doit correspondre à celui que vous avez utilisé dans la partie précédente pour SSH).**

Une image contenant texte, logiciel, Logiciel multimédia, Logiciel de graphisme

Description générée automatiquement

Il doit apparaitre maintenant **(vous devez l’activer)**

Une image contenant texte, capture d’écran, logiciel, multimédia

Description générée automatiquement

# VIII. Se connecter à JupyterHub

Dans EMR, dans notre Cluster EMR, aller dans l’onglet « applications ». Vous trouverez le lien vers JupyterHub.

Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement

Vous aurez un message indiquant que votre connexion n’est pas privée, cliquer sur Paramètres avancés >> Continuer vers le site … (dangereux).

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Vous pouvez maintenant vous connecter à JupyterHub. Les informations par défaut sont :  
**Username** : jovyan  
**Password**  : jupyter

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Il faut ensuite rajouter notre notebook puis l’ouvrir.

La dernière étape est de modifier le kernel du notebook pour **utiliser le KERNEL PYSPARK**!

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Normalement, le notebook est fonctionnel. Du calcul distribué est effectué en utilisant plusieurs machines grâce à AWS (Cloud Computing).

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Si la session Spark démarre sans problème, les messages suivants doivent s’afficher.   
Le lien vers le webui de Spark est présent.

# IX. Problème mémoire

Une image contenant texte, capture d’écran, logiciel, ligne

Description générée automatiquement

Il se peut que vous rencontriez un problème mémoire lors de votre run Jupyter.   
D’après Gemini :   
**Error Breakdown :**

* **Application Name :** application\_1720342513270\_0001
* **Application Type :** Spark
* **Error Message :** Application failed due to Application Master (AM) Container exiting with exit code 137. This code signifies a memory-related issue.
* **Diagnostics :**
  + Container killed on request (by YARN)
  + Exit code 137 (out-of-memory)
  + Killed by external signal (likely YARN due to memory constraints)

**Possible Causes :**

* **Insufficient Memory Allocation :** The Spark application likely ran out of memory allocated to the AM container.
* **Memory-Intensive Operations :** The application itself might be performing memory-heavy computations.

**Note : Il n’est pas indiqué quelle est la mémoire nécessaire au bon déroulement du processus.**

## IX.1. Solution au problème de mémoire (HARDWARE)

Puisqu’on ne sait pas quelle est la mémoire nécessaire au bon fonctionnement du notebook, j’ai décidé de modifier la partie « Dimensionnement et mise en service du cluster ».

A l’origine, j’avais demandé de « Définir manuellement la taille du cluster » avec 1 machine maître et 1 machine esclaves.   
Cette fois-ci, je demande 1 machine maitre, 1 machine esclave et 6 machines de tâches.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## IX.2. Solution au problème de mémoire (SOFTWARE)

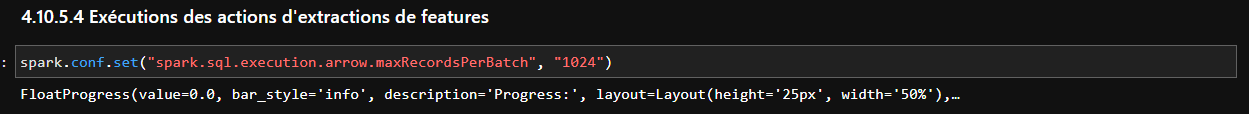
Une autre solution pour le problème mémoire peut être de modifier les lignes de codes suivantes :

* Passage de 20 à 30 parquets :

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

* Utilisation de maxRecordsPerBatch : 1024 :



## IX.3. Solution au problème de mémoire (DATA)

Diminuer le dataset (garder 1 image sur 4 par exemple si l’on prend le dossier Training (68 000 images)).

## IX.4. Note concernant le problème de mémoire

* Pas de problème pour réaliser l’extraction de features des images sur les 68k images.
* Par contre, le problème survient lors du PCA. Il faut se limiter à 20 ou 25% de 68k images avec 6 machines de tâches.   
  Pour avoir les 100%, il faut utiliser 6\*4 = 24 machines (en considérant le problème de performance comme étant linéaire) (pas testé)

# X. Résultats

## Une image contenant texte, ligne, diagramme, Tracé Description générée automatiquementX.1. Chaine entière du traitement (SparkUI)

## Une image contenant texte, capture d’écran, Police, Parallèle Description générée automatiquementX.2. Extraction des features – DAG SparkUI

## Une image contenant texte, capture d’écran, diagramme, ligne Description générée automatiquementX.3. Visualisation SparkUI pour l’extraction des features