

Programme d'Ingénierie en Intelligence Artificielle – OC IA P11

- Réalisez un traitement dans un environnement Big Data sur le Cloud

Titre du projet :

Chaîne de traitement Big Data et PCA sur AWS EMR : Cas d'usage pour la start-up Fruits!

Contexte rapide :

Mission de mise en place d'une architecture Big Data pour le traitement d'images de fruits.

Sommaire

- 1) Contexte & Scénario professionnel
- 2) Missions & Objectifs
- 3) Jeux de données
- 4) PySpark en bref
- 5) Applications de PySpark dans notre Projet
- 6) Chaîne de traitement
- 7) Architecture Big Data sur AWS
- 8) Création et configuration du Bucket S3
- 9) Mise en œuvre AWS EMR
- 10) Démonstration script PySpark - notebook
- 11) Retour critique & Perspectives
- 12) Conclusion & Q/R
- 13) Glossaire

1) Contexte & Scénario professionnel

La start-up « Fruits! »

- Solutions AgriTech, préservation de la biodiversité.
- Robots cueilleurs intelligents, application mobile.

Mission :

- Mettre à disposition du public une appli pour photographier un fruit & obtenir des informations.
- Architecture Big Data pour gérer un volume croissant d'images.

2) Missions & Objectifs

Basé sur un notebook existant modifié à la marge (ancien alternant).

Compléter la chaîne de traitement :

- Broadcast des poids du modèle TF (ou Keras).
- Réduction PCA en PySpark.

Contrainte RGPD :

- Data traitées et stockées en Europe - Paris (S3, EMR).

Livrables :

- Notebook PySpark, données/images, support de présentation.

3) Jeux de données

1. Échantillon de 300 images

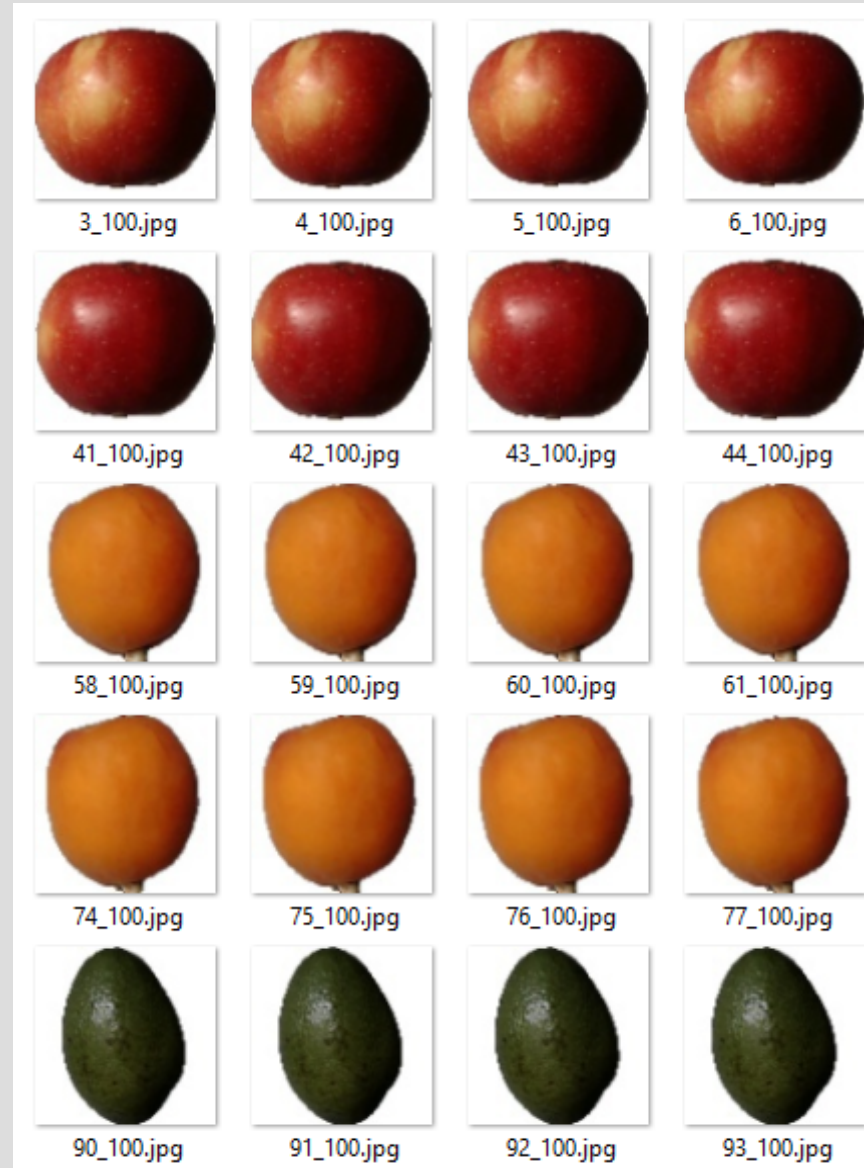
- Un échantillon de 300 **fichiers** `.jpg` (~3 Ko chacun) a été retenu, plutôt que l'intégralité du corpus initial.
- Tous les fichiers se trouvent dans `s3://ocp11-data/Test/`` et totalisent environ 900 Ko.

2. Périmètre du projet

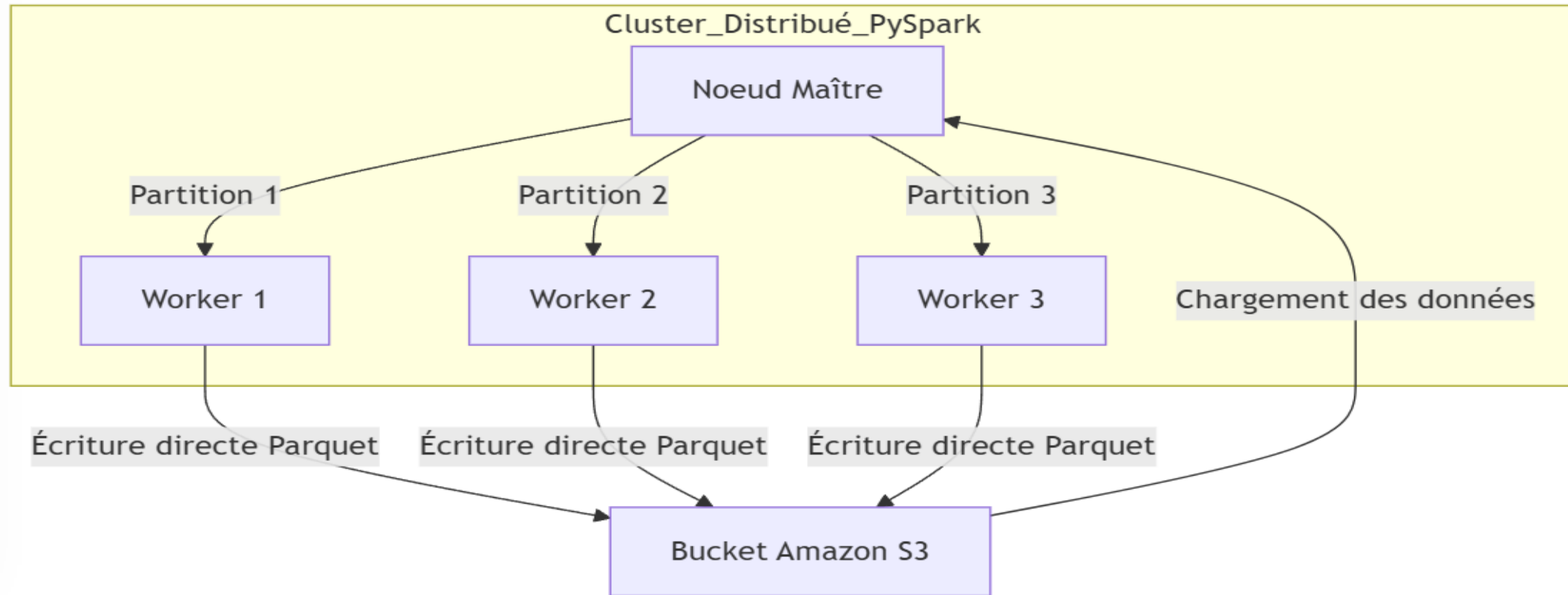
- Notre objectif se limite à **l'extraction de caractéristiques** et à **une réduction de dimension (PCA)****.
- **Aucun entraînement supervisé** n'est effectué, ce qui évite toute ambiguïté sur la finalité (pas de classification).

3. Lecture et sortie

- Les images sont lues en binaire via `binaryFile` dans PySpark.
- Les **résultats** du pipeline (PCA) sont enregistrés au **format Parquet** dans `s3://ocp11-data/Results/``.

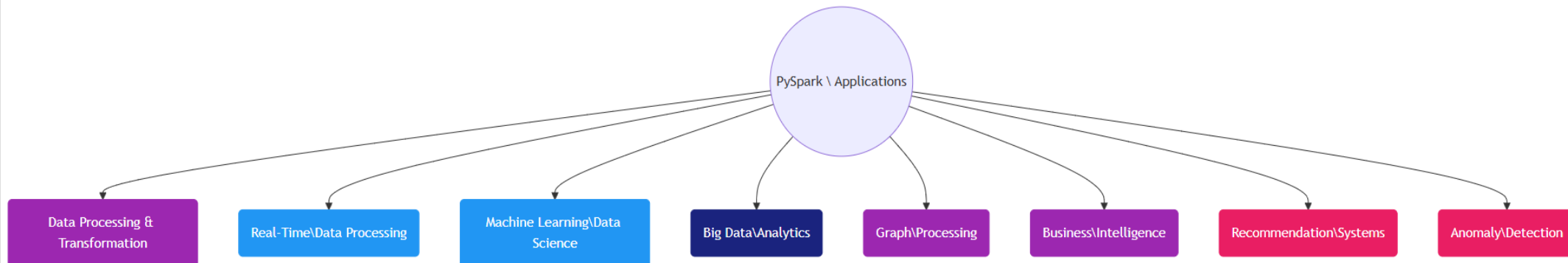


4) PySpark en bref



- Le bucket S3 alimente les partitions de données.
- Le nœud maître coordonne les tâches vers les Workers.
- Chaque Worker reçoit sa partition et réalise son traitement en parallèle.
- Chaque Worker écrit directement ses résultats en Parquet dans le bucket S3.

5) Applications de PySpark dans notre Projet



1. Data Processing et Transformation

- Dans notre cas, PySpark gère la lecture des **images brutes depuis S3**, leur conversion en DataFrame, puis leur prétraitement (extraction de features, diffusion de poids de réseau).

2. Big Data Analytics

- Le volume d'images est amené à croître rapidement. PySpark s'avère idéal pour analyser et manipuler à grande échelle (**scalabilité via EMR**).

3. Machine Learning & Data Science

- Même si nous n'entraînons pas directement de modèle final, nous utilisons PySpark pour extraire des caractéristiques (**MobileNetV2**) et effectuer la réduction de dimension (**PCA**), prérequis à tout futur modèle de reconnaissance de fruits.

4. Real-Time Data Processing (perspective future)

- À ce stade, nous **travaillons par batch**. Cependant, PySpark Streaming (ou Spark Structured Streaming) permettrait plus tard d'intégrer un flux d'images temps réel (caméras de cueillette robotisée, par exemple).

5. Graph Processing / Recommendation Systems / Anomaly Detection

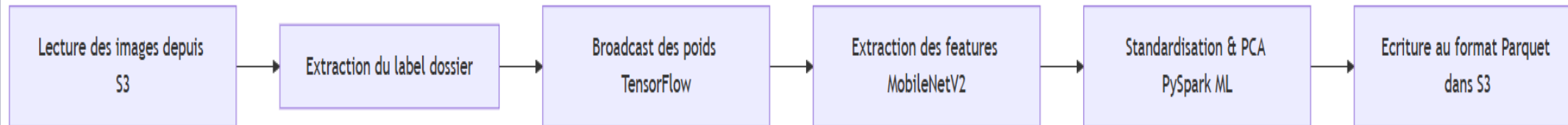
- Ces fonctionnalités de PySpark ne sont **pas exploitées dans la solution actuelle**, mais elles démontrent la polyvalence de l'outil pour d'autres cas d'usage (ex. détection de maladies sur fruits, recommandations de pratiques agricoles, etc.).

6. Business Intelligence

- Les données transformées (**parquet**) peuvent être réutilisées dans des dashboards ou outils de reporting pour mesurer, par exemple, le nombre d'images par type de fruit, l'efficacité de la cueillette robotisée, etc.

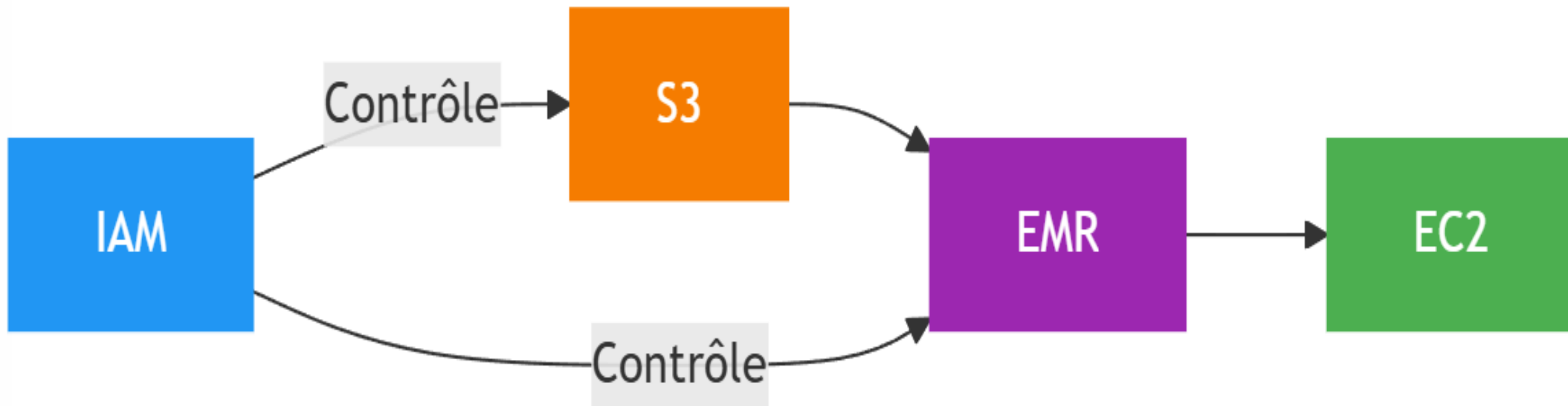
En résumé, PySpark nous offre un framework unifié pour le calcul distribué, nous permettant de passer à l'échelle et de préparer un pipeline complet de traitement d'images — depuis la lecture de fichiers jusqu'à la réduction de dimension et la mise à disposition des résultats sur AWS S3.

6) Chaîne de Traitement



D'abord, nous **chargeons** toutes les images .jpg depuis S3 de façon binaire, grâce à la fonction `binaryFile` de Spark. **Ensuite**, nous **extrayons un label (comme "Test")** simplement en analysant le chemin S3 de chaque image. **Puis**, nous **diffusons** (broadcast) sur tous les nœuds du cluster les **poids** du modèle TensorFlow (MobileNetV2), **afin que chacun puisse extraire les caractéristiques** (features) de chaque image. **Une fois ces features obtenues**, nous les **standardisons** (centrage-réduction) et nous **appliquons une PCA** (Analyse en Composantes Principales) pour réduire la dimension. **Enfin, nous stockons le résultat final**, c'est-à-dire les features réduites, dans **des fichiers Parquet directement sur S3**.

7) Architecture Big Data sur AWS



- S3 (orange) : Stocke les données (images, fichiers Parquet).
- EMR (violet) : Exécute Spark/PySpark. Il puise les données depuis S3.
- EC2 (vert) : Instances sous-jacentes fournissant la capacité de calcul aux nœuds EMR.
- IAM (bleu) : Gère et contrôle l'accès à S3 et à EMR (permissions, conformité RGPD).

7) Architecture Big Data sur AWS (suite)

1. Amazon S3

- Stocke les images (.jpg) et les résultats (fichiers Parquet).
- Durabilité et disponibilité élevées.

2. Amazon EMR

- Service managé qui exécute Spark et PySpark de manière distribuée.
- Les nœuds du cluster (Master, Core, Task) s'appuient sur **Amazon EC2** pour fournir la capacité de calcul nécessaire.

3. Amazon EC2

- Permet de choisir les types d'instances (CPU, RAM) adaptés aux besoins (ex. M5, R5, etc.).
- Facturation à l'usage, possibilité d'auto-scaling.

4. IAM

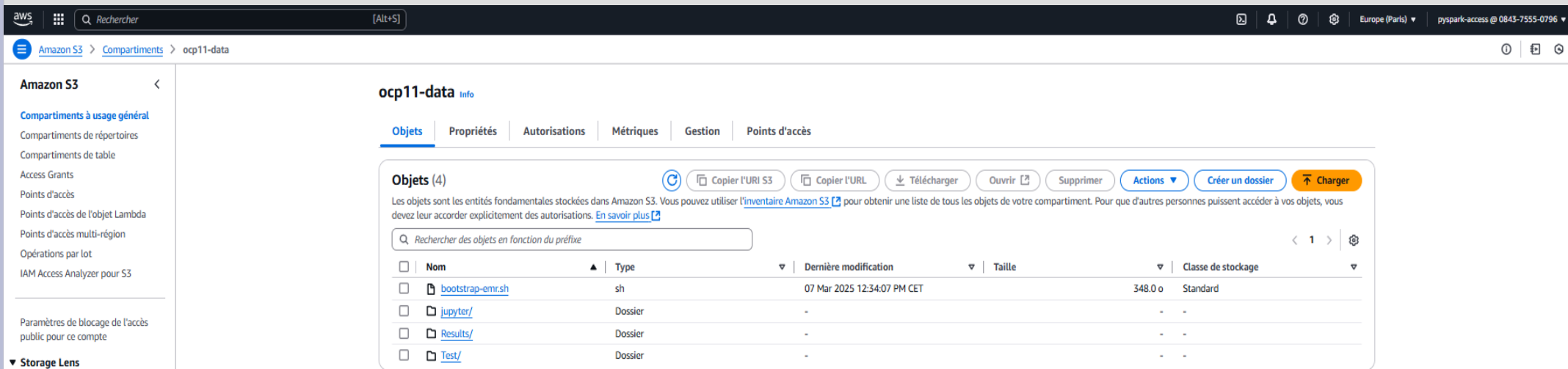
- Gère la sécurité et les autorisations (accès S3, EMR, etc.).
- Garantit le respect des normes RGPD (accès contrôlé, logs d'audit).

Pourquoi AWS ?

- **Scalabilité** : possibilité d'adapter dynamiquement les ressources (EC2) en fonction du volume de données.
- **Infrastructure stable** : services managés (EMR, S3) pour simplifier l'administration.
- **Conformité européenne** : stockage et traitement dans des régions AWS situées dans l'UE (ex. eu-west-1).

8) Création et configuration du Bucket S3

Sur cette capture d'écran, on voit le bucket “ocp11-data”, qui contient “Test/” pour les images, “Results/” pour les sorties Parquet et “jupyter/” pour le notebook. Pour créer un bucket similaire, on clique sur “Create bucket”, on lui donne un nom unique, on choisit la région (ex. “eu-west-1” pour le RGPD). Ainsi, EMR peut lire les données d'entrée et y enregistrer les sorties du pipeline PySpark, tandis que le notebook stocké dans “jupyter/” peut être exécuté directement sur le cluster.



The screenshot displays the Amazon S3 console for the bucket "ocp11-data". The left sidebar shows the navigation menu with options like "Compartmentements à usage général", "Compartmentements de répertoires", "Compartmentements de table", "Access Grants", "Points d'accès", "Points d'accès de l'objet Lambda", "Points d'accès multi-région", "Opérations par lot", and "IAM Access Analyzer pour S3". The main content area shows the bucket details and a list of objects.

Amazon S3 > **Compartmentements** > ocp11-data

ocp11-data Info

Objets | Propriétés | Autorisations | Métriques | Gestion | Points d'accès

Objets (4)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

Rechercher des objets en fonction du préfixe

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	bootstrap-emr.sh	sh	07 Mar 2025 12:34:07 PM CET	348.0 o	Standard
<input type="checkbox"/>	jupyter/	Dossier	-	-	-
<input type="checkbox"/>	Results/	Dossier	-	-	-
<input type="checkbox"/>	Test/	Dossier	-	-	-

9) Mise en œuvre AWS EMR

Cette capture d'écran illustre la création (ou le clonage) d'un **cluster EMR sur AWS**, où l'on sélectionne la **version EMR (7.6.0)**, les **applications à installer** (Spark, Hadoop, Livy, TensorFlow, etc.) et la **configuration du cluster** (taille des instances, nombre de nœuds). C'est l'étape qui prépare l'environnement **Big Data** pour exécuter notre pipeline **PySpark**.

aws

Rechercher

[Alt+S]

Amazon EMR

EMR sur EC2: Clusters

Créer un cluster

Cloner « p11-cluster »

Info

▼ Nom et applications - *requis*

Info

Donnez un nom à votre cluster et choisissez les applications que vous voulez y installer.

Nom

p11-cluster

Version Amazon EMR

Info

Une version contient un ensemble d'applications susceptibles d'être installées sur votre cluster.

emr-7.6.0

▼

Offre d'applications

Spark
Interactive

Core
Hadoop

Flink

HBase

Presto

Trino

Custom

☐ AmazonCloudWatchAgent 1.300032.2

☒ Hue 4.11.0

☐ Livy 0.8.0

☒ Pig 0.17.0

☒ TensorFlow 2.16.1

☐ Zeppelin 0.11.1

☐ Flink 1.20.0

☒ Hadoop 3.4.0

☐ JupyterEnterpriseGateway 2.6.0

☐ Oozie 5.2.1

☐ Presto 0.287

☒ Tez 0.10.2

☐ ZooKeeper 3.9.2

☐ HBase 2.6.1

☒ Hive 3.1.3

☒ JupyterHub 1.5.0

☐ Phoenix 5.2.0

☒ Spark 3.5.3

☐ Trino 457

Paramètres du catalogue de données AWS Glue

Utilisez le catalogue de données AWS Glue pour fournir un metastore externe à votre application.

☐ Utiliser pour les métadonnées de table Hive

☐ Utiliser pour les métadonnées de table Spark

Options du système d'exploitation

Info

☒ Version Amazon Linux :

☐ Amazon Machine Image (AMI) personnalisée

☒ Appliquez automatiquement les dernières mises à jour Amazon Linux

Récapitulatif

Info

Nom et applications - *requis*

Nom

p11-cluster

Version Amazon EMR

emr-7.6.0

Offre d'applications

Custom (Hadoop 3.4.0, Hive 3.1.3, Hue 4.11.0, JupyterHub 1.5.0, Pig 0.17.0, Spark 3.5.3,...)

Configuration de cluster - *requis*

Groupes d'instances uniformes

Primaire (m5.xlarge), Unité principale (m5.xlarge), Tâche (m5.xlarge)

Dimensionnement et mise en service du cluster - *requis*

Configuration de mise en service

Taille du noyau: 1 instance

Taille de la tâche: 2 instances

Annuler

Cloner un cluster

9) Mise en œuvre AWS EMR (suite)

Dans la section **“Paramètres du logiciel”**, on personnalise la configuration d’applications (ici, **jupyter-conf**, **persistance activée sur S3**). On sélectionne ensuite la **paire de clés EC2 pour créer un tunnel SSH** (par exemple avec PuTTY et FoxyProxy) afin d’accéder à **JupyterHub** et exécuter le **notebook**. Le **rôle IAM du service EMR**, ainsi que le **profil d’instance**, déterminent les permissions dont dispose le **cluster pour lire/écrire sur S3** ou se connecter au service. Enfin, on clique sur **“Cloner un cluster”** pour créer ou relancer l’infrastructure, validant ainsi l’ensemble de la configuration.

▼ identifications Info

Utilisez des balises pour rechercher et filtrer les ressources, ainsi que pour suivre les coûts AWS associés à votre cluster.

Clé

Supprimer

Ajouter une nouvelle identification

Vous pouvez ajouter 40 balises supplémentaires.

▼ Paramètres du logiciel Info

Remplacez les configurations par défaut pour des applications spécifiques de votre cluster.

Entrer la configuration

Charger JSON à partir d'Amazon S3

1

2 {

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

4 "Properties": {

5 "s3.persistence.bucket": "ocp11-data",

6 "s3.persistence.enabled": "true"

7 }

8 }

9 }

JSON Ln 1, Col 1 0 0

▼ Configuration de sécurité et paire de clés EC2 Info

Choisissez une configuration de sécurité ou créez-en une nouvelle que vous pourrez réutiliser avec d'autres clusters.

Configuration de sécurité

Sélectionnez les paramètres de chiffrement, d'authentification, d'autorisation et de service de métadonnées d'instance de votre cluster.

Parcourir

Créer une configuration de sécurité

Paire de clés Amazon EC2 pour SSH sur le cluster Info

Parcourir

Créer une paire de clés

▼ Rôle Identity and Access Management (IAM) - requis Info

Choisissez ou créez une fonction du service et un profil d'instance pour les instances EC2 de votre cluster.

Fonction du service Amazon EMR Info

La fonction du service est un rôle IAM assumé par Amazon EMR pour mettre en service des ressources et effectuer des actions au niveau du service avec d'autres services AWS.

Choisir une fonction du service existant

Sélectionnez une fonction du service par défaut ou un rôle personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec d'autres services AWS.

Créer une fonction du service

Laissez Amazon EMR créer une nouvelle fonction du service afin que vous puissiez accorder et restreindre l'accès aux ressources d'autres services AWS.

Fonction du service

AmazonEMR-ServiceRole-20250204T125324

Profil d'instance EC2 pour Amazon EMR

Le profil d'instance attribue un rôle à chaque instance EC2 d'un cluster. Le profil d'instance doit spécifier un rôle qui peut accéder aux ressources pour vos étapes et actions d'arrimage.

Choisir un profil d'instance existant

Sélectionnez un rôle par défaut ou un profil d'instance personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec vos ressources dans Amazon S3.

Choisir un profil d'instance

Laissez Amazon EMR créer un profil d'instance afin de pouvoir spécifier un ensemble personnalisé de ressources auquel il peut accéder dans Amazon S3.

Profil d'instance

AmazonEMR-InstanceProfile-20250204T125307

Rôle d'autoscaling personnalisé - facultatif

Lorsqu'une règle d'autoscaling personnalisée se déclenche, Amazon EMR assume ce rôle pour ajouter et réduire les instances EC2. En savoir plus

Rôle d'autoscaling personnalisé

Choisir un rôle IAM

Créer un rôle IAM

Récapitulatif Info

Nom et applications - requis

Nom

p11-cluster

Version Amazon EMR

emr-7.6.0

Offre d'applications

Custom (Hadoop 3.4.0, Hive 3.1.3, Hue 4.11.0, JupyterHub 1.5.0, Pig 0.17.0, Spark 3.5.3,...)

Configuration de cluster - requis

Groupes d'instances uniformes

Primaire (m5.xlarge), Unité principale (m5.xlarge), Tâche (m5.xlarge)

Dimensionnement et mise en service du cluster - requis

Configuration de mise en service

Taille du noyau: 1 instance

Taille de la tâche: 2 instances

Annuler

Cloner un cluster

13

9) Mise en œuvre AWS EMR (suite)

Dans la configuration EMR, on distingue :

- **Le groupe “Primaire / Unité principale”** : c’est le **Master**. Il orchestre la répartition des données et la coordination des tâches Spark.
- **Les groupes “Tâche”** (“Tâche 1 sur 2” et “Tâche 2 sur 2”) : ce sont les **Workers**. Chaque nœud “Tâche” exécute en parallèle une partie du travail (lecture, traitement PySpark...).

Concrètement :

1. Primaire / Master :

- Un seul nœud prend ce rôle : il gère la planification et supervise l’exécution.

2. Tâche / Worker :

- On peut en avoir plusieurs (ici, 2). Chacun reçoit une partie des partitions de données et exécute localement les calculs (PCA, extraction de features, etc.).

Ainsi, la mention “Tâche 1 sur 2” et “Tâche 2 sur 2” indique qu’il y aura deux nœuds Workers. C’est ce qui **permet la parallélisation** du traitement dans notre pipeline PySpark.

▼ Configuration de cluster - **requis** [Info](#)

Choisissez une méthode de configuration pour les groupes de nœuds primaires, principaux et de tâches de votre cluster.

☒ **Groupes d'instances uniformes**
Choisissez le même type d'instance EC2 et la même option d'achat (à la demande ou Spot) pour tous les nœuds de votre groupe de nœuds. [En savoir plus](#)

☐ **Flottes d'instances flexibles**
Choisissez parmi la plus grande variété d'options de provisionnement pour les instances EC2 de votre cluster. Diversifiez les types d'instances et les options d'achat, et utilisez une stratégie d'allocation. [En savoir plus](#)

Groupes d'instances uniformes

Primaire
Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire
EBS uniquement stockage
Prix à la demande : 0,224 USD par inst...
Prix Spot le plus bas : 0,088 USD (eu-west...
▼

Actions ▼

☐ Utiliser la haute disponibilité
Lancez des clusters hautement disponibles et plus résilients avec trois nœuds primaires sur des instances à la demande. Cette configuration s'applique pendant toute la durée de vie de votre cluster. [En savoir plus](#)

► Configuration de nœud - **facultatif**

Unité principale
Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire
EBS uniquement stockage
Prix à la demande : 0,224 USD par inst...
Prix Spot le plus bas : 0,088 USD (eu-west...
▼

Actions ▼

► Configuration de nœud - **facultatif**

Tâche 1 sur 2 Retirer le groupe d'instances

Nom
Tâche - 2

Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire
EBS uniquement stockage
Prix à la demande : 0,224 USD par inst...
Prix Spot le plus bas : 0,088 USD (eu-west...
▼

Actions ▼

► Configuration de nœud - **facultatif**

Tâche 2 sur 2 Retirer le groupe d'instances

Nom
Tâche - 1

Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire
EBS uniquement stockage
Prix à la demande : 0,224 USD par inst...
Prix Spot le plus bas : 0,088 USD (eu-west...
▼

Actions ▼

► Configuration de nœud - **facultatif**

[Ajouter un groupe d'instances de tâches](#)

Vous pouvez ajouter jusqu'à 46 autres groupes d'instances de tâches.

Volume racine EBS

Le volume racine EBS s'applique aux systèmes d'exploitation et aux applications que vous installez sur le cluster. [Contraintes relatives au rapport de volume racine EBS](#)

Taille (Gio)	IOPS	Débit (Mio/s)
20	3000	125
15 - 100 GiB par volume. Volume SSD polyvalent (gp3)	5000- 16000 IOPS par volume. Choisissez un rapport maximum de 500:1 entre les IOPS et la taille du volume.	125- 1000 Mio/s par volume. Choisissez un rapport maximum de 0,25:1 entre le débit et les IOPS.

9) Mise en œuvre AWS EMR (suite)

Configuration finale du cluster EMR – Points clés

- **Instances m5.large** : Choix équilibré entre CPU, mémoire et coût pour le Master et les Workers, adapté à notre pipeline PySpark (traitement d'images et PCA).

- **Clé SSH (Key Pair)** : Permet de **se connecter** (par exemple, via **JupyterHub**) au nœud Master et **tester le notebook** directement sur le cluster.

- **Script de bootstrap** : ``bootstrap-emr.sh`` (**stocké sur S3**) pour installer toute dépendance ou config nécessaire avant le démarrage réel du cluster.

- **Paramètres réseau** : Sélection du VPC, du sous-réseau et des groupes de sécurité (pare-feu) garantissant un accès maîtrisé et conforme RGPD.

- **Résiliation & logs** : Option d'arrêter le cluster automatiquement après usage, et possibilité d'archiver les journaux pour diagnostiquer ou auditer les traitements. 15

▼ **Dimensionnement et mise en service du cluster - requis** [info](#)
Choisissez la manière dont Amazon EMR doit dimensionner votre cluster.

Choisir une option

☒ Définir manuellement la taille du cluster
Utilisez cette option si vous connaissez vos modèles de charge de travail à l'avance.

☐ Utiliser la mise à l'échelle gérée par EMR
Surveillez les principales métriques de charges de travail afin qu'EMR puisse optimiser la taille du cluster et l'utilisation des ressources.

☐ Utiliser un autoscaling personnalisé
Pour dimensionner de manière programmatique les unités principales et les nœuds de tâches, créez des politiques d'autoscaling personnalisées.

Configuration de mise en service
Définissez la taille de votre noyau et tâchesgroupes d'instance. Amazon EMR tente de fournir cette capacité lorsque vous lancez votre cluster.

Nom	Type d'instance	Taille de l'instance(s)	Utiliser l'option d'achat Spot
Unité principale	m5.xlarge	1	<input type="checkbox"/>
Tâche - 2	m5.xlarge	1	<input type="checkbox"/>
Tâche - 1	m5.xlarge	1	<input type="checkbox"/>

▼ **Réseaux - requis** [info](#)
Choisissez les paramètres réseau qui déterminent la façon dont vous et les autres entités communiquez avec votre cluster.

Cloud privé virtuel (VPC) [info](#)
vpc-0a09f1c50fa743ea [Parcourir](#) [Créer un VPC](#)

Sous-réseau [info](#)
subnet-0014a1e187e425e23 [Parcourir](#) [Créer un sous-réseau](#)

► **Groupes de sécurité EC2 (pare-feu)**

► **Étapes (0)** [info](#) [Supprimer](#) [Modifier](#) [Ajouter](#)
Utilisez des commandes et des scripts pour indiquer à votre cluster où trouver vos données et comment les traiter. Les étapes s'exécutent de manière consécutive, sauf si vous activez l'option Simultanéité.

▼ **Résiliation du cluster et remplacement des nœuds** [info](#)
Choisissez des paramètres de résiliation et protégez votre cluster contre un arrêt accidentel.

Option de résiliation

☒ Résilier manuellement le cluster

☐ Résilier automatiquement le cluster à la fin de la dernière étape

☐ Résilier automatiquement le cluster après le temps d'inactivité (Recommandé)

☐ Utiliser la protection contre la résiliation
Protégez votre cluster contre les résiliations accidentelles. Si cette option est activée, vous devez d'abord désactiver la protection pour résilier le cluster. Nous vous recommandons d'activer la protection contre la résiliation pour vos clusters de longue durée.

Remplacement des nœuds défectueux - nouveau [info](#)

☒ Activer
Amazon EMR interrompt les processus sur les nœuds défectueux afin de minimiser les pertes de données et les interruptions de travail. Il remplace rapidement ces nœuds défectueux par de nouvelles instances EC2 pour assurer le bon fonctionnement de vos tâches en cours d'exécution.

☐ Désactiver
Amazon EMR ajoute les nœuds défectueux à une liste de dénombrement tout en les conservant dans le cluster, ce qui vous permet de continuer à y accéder pour la résolution de problèmes.

▼ **Actions d'amorçage (1)** [info](#) [Supprimer](#) [Modifier](#) [Ajouter](#)
Utilisez les actions d'amorçage pour installer des logiciels ou personnaliser la configuration de votre instance.

Nom	Emplacement Amazon S3	Arguments
<input type="radio"/> Install dependencies	s3://ocp11-data/bootstrap-emr.sh	-

► **Journaux de cluster** [info](#)
Choisissez où et comment stocker vos fichiers journaux.

9) Mise en œuvre AWS EMR (suite)

Dans la section **“Paramètres du logiciel”**, on personnalise la configuration d’applications (ici, **jupyter-conf**, **persistance activée sur S3**). On sélectionne ensuite la **paire de clés EC2** pour créer un tunnel SSH (par exemple avec PuTTY et FoxyProxy) afin d’accéder à **JupyterHub** et exécuter le **notebook**. Le **rôle IAM** du service **EMR**, ainsi que le **profil d’instance**, déterminent les permissions dont dispose le **cluster pour lire/écrire sur S3** ou se connecter au service. Enfin, on clique sur **“Cloner un cluster”** pour créer ou relancer l’infrastructure, validant ainsi l’ensemble de la configuration.

▼ identifications Info

Utilisez des balises pour rechercher et filtrer les ressources, ainsi que pour suivre les coûts AWS associés à votre cluster.

Clé

Supprimer

Ajouter une nouvelle identification

Vous pouvez ajouter 40 balises supplémentaires.

▼ Paramètres du logiciel Info

Remplacez les configurations par défaut pour des applications spécifiques de votre cluster.

Entrer la configuration

Charger JSON à partir d'Amazon S3

1

2 {

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

4 "Properties": {

5 "s3.persistence.bucket": "ocp11-data",

6 "s3.persistence.enabled": "true"

7 }

8 }

9 }

JSON Ln 1, Col 1

▼ Configuration de sécurité et paire de clés EC2 Info

Choisissez une configuration de sécurité ou créez-en une nouvelle que vous pourrez réutiliser avec d'autres clusters.

Configuration de sécurité

Sélectionnez les paramètres de chiffrement, d'authentification, d'autorisation et de service de métadonnées d'instance de votre cluster.

Parcourir

Créer une configuration de sécurité

Paire de clés Amazon EC2 pour SSH sur le cluster Info

Parcourir

Créer une paire de clés

▼ Rôle Identity and Access Management (IAM) - requis Info

Choisissez ou créez une fonction du service et un profil d'instance pour les instances EC2 de votre cluster.

Fonction du service Amazon EMR Info

La fonction du service est un rôle IAM assumé par Amazon EMR pour mettre en service des ressources et effectuer des actions au niveau du service avec d'autres services AWS.

Choisir une fonction du service existant

Sélectionnez une fonction du service par défaut ou un rôle personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec d'autres services AWS.

Créer une fonction du service

Laissez Amazon EMR créer une nouvelle fonction du service afin que vous puissiez accorder et restreindre l'accès aux ressources d'autres services AWS.

Fonction du service

AmazonEMR-ServiceRole-20250204T125324

Profil d'instance EC2 pour Amazon EMR

Le profil d'instance attribue un rôle à chaque instance EC2 d'un cluster. Le profil d'instance doit spécifier un rôle qui peut accéder aux ressources pour vos étapes et actions d'arrimage.

Choisir un profil d'instance existant

Sélectionnez un rôle par défaut ou un profil d'instance personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec vos ressources dans Amazon S3.

Choisir un profil d'instance

Laissez Amazon EMR créer un profil d'instance afin de pouvoir spécifier un ensemble personnalisé de ressources auquel il peut accéder dans Amazon S3.

Profil d'instance

AmazonEMR-InstanceProfile-20250204T125307

Rôle d'autoscaling personnalisé - facultatif

Lorsqu'une règle d'autoscaling personnalisée se déclenche, Amazon EMR assume ce rôle pour ajouter et réduire les instances EC2. En savoir plus

Rôle d'autoscaling personnalisé

Choisir un rôle IAM

Créer un rôle IAM

Récapitulatif Info

Nom et applications - requis

Nom

p11-cluster

Version Amazon EMR

emr-7.6.0

Offre d'applications

Custom (Hadoop 3.4.0, Hive 3.1.3, Hue 4.11.0, JupyterHub 1.5.0, Pig 0.17.0, Spark 3.5.3,...)

Configuration de cluster - requis

Groupes d'instances uniformes

Primaire (m5.xlarge), Unité principale (m5.xlarge), Tâche (m5.xlarge)

Dimensionnement et mise en service du cluster - requis

Configuration de mise en service

Taille du noyau: 1 instance

Taille de la tâche: 2 instances

Annuler

Cloner un cluster

16

9) Mise en œuvre AWS EMR (suite)

Une fois le cluster démarré, la **console EMR** affiche le récapitulatif : on y voit le numéro d'ID, la version installée (**EMR 7.6.0**), les journaux S3, ainsi que les rôles et autorisations IAM associés. Cela confirme que notre environnement Big Data est opérationnel, prêt à recevoir notre pipeline PySpark (**lecture d'images, PCA**) et à écrire les résultats au format **Parquet sur S3**

The screenshot displays the AWS EMR console interface for a cluster named 'p11-cluster'. The top navigation bar shows the AWS logo, a search bar, and various utility links. The main content area is titled 'p11-cluster' and includes a 'Résumé' (Summary) section with four columns of information:

- Informations sur le cluster:** ID de cluster (j-2T9MOZ7DSHZ3Z), ARN du cluster (arn:aws:elasticmapreduce:eu-west-3:084375550796:cluster/j-2T9MOZ7DSHZ3Z), Configuration de cluster (Groupes d'instances), and Capacité (1 primaire(s) | 1 unité(s) principale(s) | 2 tâche(s)).
- Applications:** Version d'Amazon EMR (emr-7.6.0) and Applications installées (Hadoop 3.4.0, Hive 3.1.3, Hue 4.11.0, JupyterHub 1.5.0, Pig 0.17.0, Spark 3.5.3, TensorFlow 2.16.1, Tez 0.10.2).
- Gestion des clusters:** Destination des journaux dans Amazon S3 (aws-logs-084375550796-eu-west-3/elasticmapreduce), Interfaces utilisateur d'application persistantes (Serveur d'historique Spark, Serveur de chronologie YARN, Interface utilisateur Tez), DNS public du nœud primaire (ec2-15-237-96-179.eu-west-3.compute.amazonaws.com), and Connexion au nœud primaire à l'aide de SSH.
- Statut et heure:** Statut (Résilié), Heure de création (8 mars 2025 07:52 (UTC+01:00)), Temps écoulé (3 heures, 58 minutes), and Heure de fin (8 mars 2025 11:50 (UTC+01:00)).

Below the summary, there are several tabs for cluster management: Propriétés, Actions d'amorçage, Instances (Matériel), Étapes, Applications, Configurations, Surveillance, Évènements, and identifications (1). The 'Propriétés' tab is active, showing details for the 'Système d'exploitation' (Version Amazon Linux: 2023.6.20250218.2), 'Journaux de cluster' (Archiver les fichiers journaux dans Amazon S3: Actif, Emplacement Amazon S3: s3://aws-logs-084375550796-eu-west-3/elasticmapreduce/), 'Résiliation du cluster et remplacement des nœuds' (Option de résiliation: Résilier manuellement le cluster, Temps d'inactivité: -, Protection contre la résiliation: Désactivé, Remplacement des nœuds défectueux: Actif), 'Réseau et sécurité' (Cloud privé virtuel (VPC): vpc-0a09f1c50fea743ea, Sous-réseau(x) et zone(s) de disponibilité: subnet-0014a1e187e425e23), 'Configuration de sécurité' (Configuration de sécurité: Aucun, Paire de clés EC2: p11-cluster), and 'Autorisations' (Fonction du service pour Amazon EMR: AmazonEMR-ServiceRole-20250204T125324, Profil d'instance EC2: AmazonEMR-InstanceProfile-20250204T125307, Rôle d'autoscaling personnalisé: Non configuré).

10) Démonstration script PySpark - notebook

Notebook PySpark : Lecture depuis S3 (images), Application du broadcast + featurization, PCA & stockage Parquet.

aws

Rechercher

[Alt+S]

Amazon S3 > Compartiments > ocp11-data > jupyter/ > jovyan/ > Preudhomme_Patrice_1_022025.ipynb

Amazon S3

Compartiments à usage général

Compartiments de répertoires

Compartiments de table

Access Grants

Points d'accès

Points d'accès de l'objet Lambda

Points d'accès multi-région

Opérations par lot

IAM Access Analyzer pour S3

Paramètres de blocage de l'accès public pour ce compte

▼ Storage Lens

Tableaux de bord

Groupes Storage Lens

Paramètres AWS Organizations

Fonctionnalité spot 11

► AWS Marketplace pour S3

Preudhomme_Patrice_1_022025.ipynb

Info

Copier l'URI S3

Télécharger

Ouvrir

Actions d'objet

Propriétés

Autorisations

Versions

Présentation de l'objet

Propriétaire

612bc059af26b3586cba1ae445eeb1ad47013802bcad13f8b7fdec6db3bf0938

Région AWS

Europe (Paris) eu-west-3

Dernière modification

08 Mar 2025 11:47:42 AM CET

Taille

163.2 Ko

Type

ipynb

Clé

jupyter/jovyan/Preudhomme_Patrice_1_022025.ipynb

URI S3

s3://ocp11-data/jupyter/jovyan/Preudhomme_Patrice_1_022025.ipynb

Amazon Resource Name (ARN)

arn:aws:s3:::ocp11-data/jupyter/jovyan/Preudhomme_Patrice_1_022025.ipynb

Balise d'entité (Etag)

46b06a891dcd3d3070829b146351c3

URL de l'objet

https://ocp11-data.s3.eu-west-3.amazonaws.com/jupyter/jovyan/Preudhomme_Patrice_1_022025.ipynb

Présentation de la gestion des objets

Les propriétés de compartiment et les configurations de gestion des objets suivantes ont un impact sur le comportement de cet objet.

Propriétés du compartiment

Gestion des versions de compartiment

Lorsque cette option est activée, plusieurs variantes d'un objet peuvent être stockées dans le compartiment pour faciliter la récupération en cas d'actions involontaires de l'utilisateur et de défaillances de l'application.

Désactivé

La gestion des versions n'est pas activée pour le compartiment « ocp11-data ».

Nous vous recommandons d'activer la gestion des versions des compartiments pour vous protéger contre le remplacement ou la suppression involontaires d'objets. En savoir plus

Activer la gestion des versions de compartiment

Configurations de gestion

Statut de réplication

Lorsqu'une règle de réplication est appliquée à un objet, le statut de réplication indique la progression de l'opération.

-

Afficher les règles de réplication

Règle d'expiration

Vous pouvez utiliser une configuration de cycle de vie pour définir des règles d'expiration afin de planifier la suppression de cet objet après une période prédéfinie.

-

Date d'expiration

L'objet sera supprimé définitivement à cette date.

-

CloudShell Commentaires

© 2025, Amazon Web Services, Inc. ou ses affiliés. Confidentialité Conditions Préférences relatives aux cookies

11) Retour critique & Perspectives

Retour sur la solution :

- Bien adaptée pour le volume actuel.
- Possibilité de l'étendre à plus de données (scalabilité).

Perspectives :

Tester des instances EC2 moins coûteuses (ex. m5.large pour les nœuds workers) afin de mieux maîtriser les coûts. **Activer l'auto-scaling du cluster EMR** pour absorber dynamiquement les pics d'activité sans surdimensionner le cluster. Enfin, intégrer des **alertes ou limites de coûts** via CloudWatch pour anticiper les dépassements éventuels. Ces améliorations optimiseront davantage le rapport performance-coût tout en renforçant la robustesse et le contrôle de la solution Big Data sur AWS.

12) Conclusion & Q/R

Objectifs atteints :

- Traitement Big Data (EMR, Spark).
- PCA pour réduire la dimension.
- Données & résultats sur S3, conformes RGPD.

Intérêt pour “Fruits!” :

- Base pour la future application de reconnaissance de fruits.

13) Glossaire

AWS

Amazon Web Services : plateforme de services cloud (stockage, calcul, bases de données...).

S3

Simple Storage Service : service de stockage objet.

EC2

Elastic Compute Cloud : machines virtuelles à la demande (nœuds EMR, serveurs divers).

EMR

Elastic MapReduce : service géré pour exécuter Hadoop, Spark, Hive, etc. à grande échelle.

IAM

Identity and Access Management : gestion des identités et permissions (accès cloud).

Big Data et PySpark

Big Data : ensemble de techniques et outils pour stocker et traiter des volumes de données trop importants pour un système classique.

Spark : moteur de calcul distribué en mémoire, très rapide pour le Big Data.

PySpark : API Python de Spark pour écrire le code de traitement distribué.

TensorFlow et Keras

TensorFlow : bibliothèque open source de machine learning (spécialisée dans le deep learning).

Keras : API haut niveau de TensorFlow pour définir et entraîner plus simplement des modèles.

13) Glossaire (suite)

MobileNetV2

Modèle pré-entraîné de vision par ordinateur, léger et rapide, utilisé pour extraire des représentations intermédiaires (features) d'images.

PCA

Principal Component Analysis ou Analyse en Composantes Principales : technique de réduction de dimension qui conserve l'essentiel de la variance des données tout en diminuant le nombre de dimensions.

RGPD

Règlement Général sur la Protection des Données : réglementation européenne imposant la localisation des données en Europe et la protection de la vie privée.

Parquet

Format de stockage en colonnes, binaire et compressé, optimisé pour le traitement distribué (Spark, Hive...).

Broadcast (Spark)

Mécanisme pour diffuser un objet (modèle, variables...) depuis le driver vers tous les nœuds du cluster sans le recharger à chaque exécution.

Python 3

Langage de programmation utilisé dans PySpark, standard pour l'analyse et le machine learning.