



RÉALISEZ UN TRAITEMENT DANS UN ENVIRONNEMENT BIG DATA SUR LE CLOUD





SOMMAIRE

1. Problématique et données utilisées

Identification du problème à résoudre ainsi que la présentation des données employées pour y parvenir.

2. Infrastructure Big Data

Description de l'écosystème et des technologies Big Data utilisées pour traiter et analyser les données.

3. Processus de traitement des données

Explication détaillée des étapes et des outils impliqués dans le pipeline de traitement des données.

4. Synthèse finale

Bilan général et récapitulatif des résultats obtenus ainsi que des perspectives envisagées.



1. Problématique et données utilisées

Contexte :

Fruits!, une start-up spécialisée dans le développement de solutions innovantes pour la récolte de fruits, s'efforce de créer des robots cueilleurs intelligents capables d'adapter leur approche en fonction de chaque espèce de fruit. En parallèle, l'entreprise développe une application mobile qui permettra de reconnaître un fruit à partir d'une simple photo et de fournir des informations à son sujet. Cette application servira également à établir une première version de l'architecture Big Data nécessaire à la gestion et l'analyse des données collectées.

Objectif :

Développer un pipeline de traitement d'images performant, capable de reconnaître les fruits à partir de photos, tout en intégrant une réduction des dimensions des images. Assurer la scalabilité du traitement des données en migrant vers un environnement Big Data sur AWS et en garantissant l'efficacité du calcul distribué avec PySpark.

Mission :

1. Construire et optimiser un pipeline de traitement d'images en reprenant les travaux existants et en y intégrant une étape de réduction des dimensions.
2. Anticiper l'augmentation des volumes de données en migrant vers une infrastructure Big Data basée sur les services Cloud AWS.
3. Tester et mettre en place un script PySpark capable de gérer le calcul distribué afin de garantir la scalabilité du traitement.



❑ Jeu de données :

Le dataset "**Fruits 360**" (version 11, datée du 04/08/2024) contient des images de fruits, légumes et fruits à coques. Il comprend un total de **94 110 images**, avec un jeu de test de **23 619 images** réparties en **141 classes**, telles que **Apple Golden 1, Banana, Kiwi, Strawberry**, entre autres. Chaque fruit est représenté par différentes variétés (par exemple, plusieurs variétés de pommes).

- **Organisation** : Chaque classe dispose de son propre répertoire, avec plusieurs photos du même fruit sous différents angles (r, r0, r1).
- **Dimensions des images** : 100x100 pixels.
- **Caractéristiques des images** : Images en couleurs avec un fond blanc uniformisé.
- **Format des fichiers** : .jpg.



- Apple Braeburn
- Apple Crimson Snow
- Apple Golden 1
- Apple Golden 2
- Apple Golden 3
- Apple Granny Smith
- Apple Pink Lady
- Apple Red 1
- Apple Red 2
- Apple Red 3
- Apple Red Delicious
- Apple Red Yellow 1
- Apple Red Yellow 2



2. Infrastructure Big Data

Description de l'écosystème et des technologies Big Data utilisées pour traiter et analyser les données.

❑ Concept principal d'une architecture Big Data : le calcul distribué

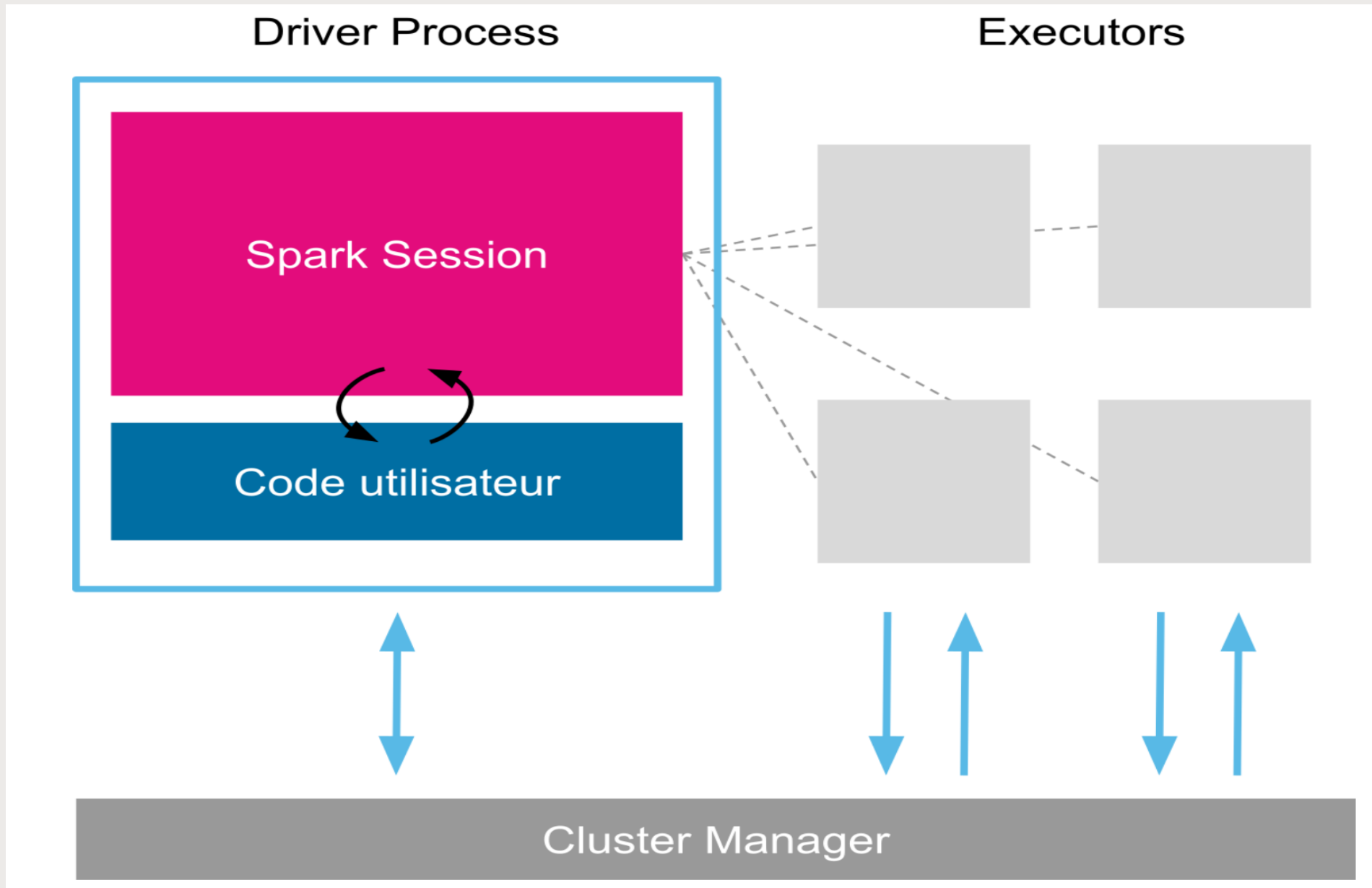
Le **cloud computing** permet de déployer des ressources de manière flexible. Le **calcul distribué** est une méthode permettant d'utiliser plusieurs unités de calcul réparties sur différents clusters afin de réduire le temps d'exécution d'un projet. Un **cluster** est un groupe de machines qui travaillent ensemble pour accomplir une tâche donnée, facilitant ainsi le **passage à l'échelle horizontale** (ajout de nouvelles machines pour augmenter la capacité de traitement).

❑ Outils de l'architecture Big Data : Apache Spark



Apache Spark est un framework qui gère et coordonne l'exécution de tâches sur des données réparties à travers un groupe d'ordinateurs (cluster). Il utilise un **gestionnaire de cluster** qui surveille les ressources disponibles pour optimiser le traitement. Le **processus de pilotage** (driver) est responsable de la gestion et de l'exécution du programme via des **exécuteurs** répartis sur les machines du cluster, appelées **workers**, qui contiennent plusieurs exécuteurs.

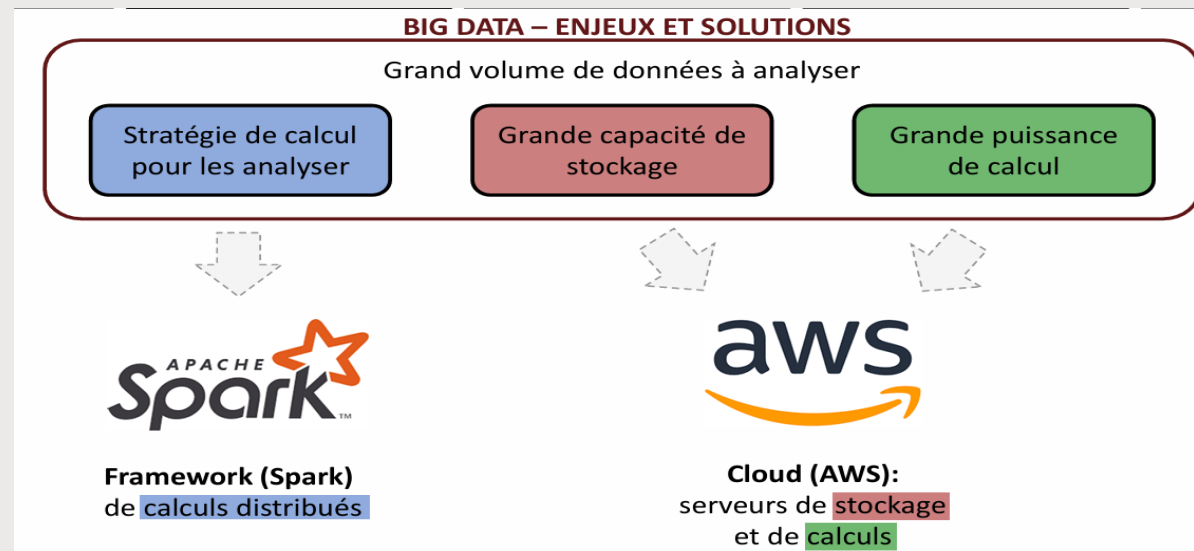
PySpark est une interface permettant d'utiliser Spark avec le langage **Python**, facilitant ainsi le développement et l'exécution de programmes distribués.





❑ Caractéristiques clés de Spark :

- **Rapidité** : Spark exécute des opérations directement en mémoire, ce qui le rend nettement plus rapide que des systèmes reposant sur le stockage disque, tels que Hadoop MapReduce.
- **API variées** : Spark offre des API dans plusieurs langages de programmation, notamment Java, Scala, Python et R, facilitant son utilisation pour divers développeurs.
- **Bibliothèques intégrées** : Spark inclut des bibliothèques pour différents usages, comme le traitement des données relationnelles avec **Spark SQL** et les algorithmes de machine learning avec **MLlib**.
- **Résilience** : Spark est hautement tolérant aux pannes grâce à son modèle basé sur les **RDDs** (Resilient Distributed Datasets), qui soutient les transformations, les actions et l'exécution différée (lazy execution).

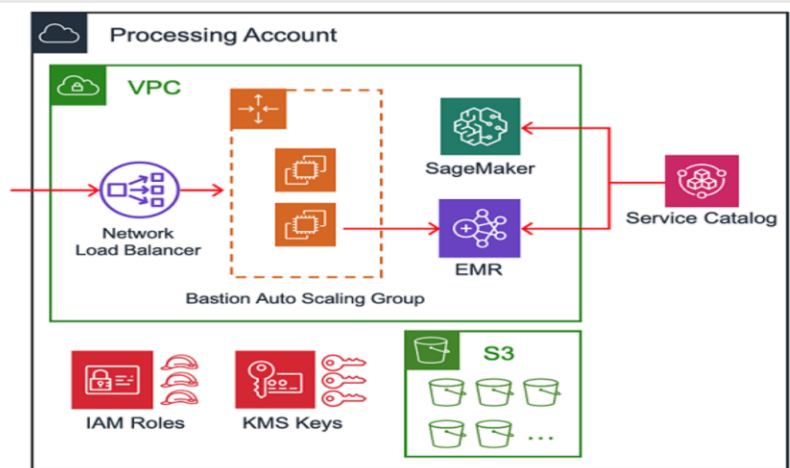


❑ Solutions AWS pour le Big Data

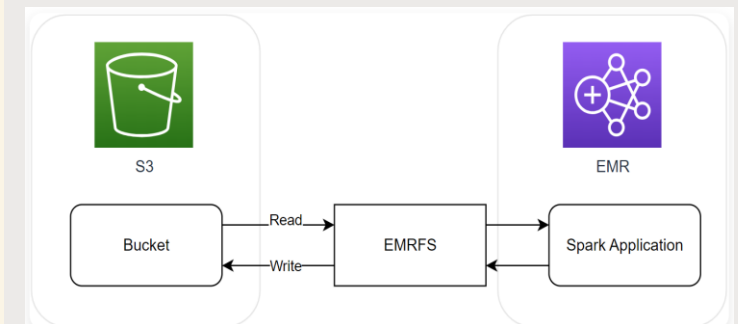
- **Fournisseur de services cloud** : AWS
- **Infrastructure technique** : Modèle **PAAS** (Platform as a Service) avec des clusters de calcul utilisant **EMR** (Elastic MapReduce) pour le traitement des données massives
- **Stockage des données** : Service de stockage **S3** (Simple Storage Service) pour entreposer et gérer les volumes de données

❑ Paramétrage de l'environnement de travail

- **AWS CLI** : Interface en ligne de commande pour interagir avec les services AWS
- **IAM** (Identity and Access Management) : Gestion des identités et des droits d'accès
- **Tunnel SSH** : Utilisé pour établir des connexions sécurisées entre les machines et le cloud AWS
- **Conformité RGPD** : Respect des normes relatives à la gestion des droits d'accès aux données conformément au **Règlement Général sur la Protection des Données (RGPD)**.



Processing Account structure upon delivery to the customer





❑ Stockage des données sur S3

Paris ▼

Khoty ▼

Amazon S3



- **Création d'un bucket :** Mise en place d'un compartiment de stockage sur S3 pour y charger les données du jeu de test.
- **Stockage du fichier d'amorçage :** Dépôt du fichier nécessaire pour l'initialisation des opérations.
- **Chargement du notebook :** Transfert du notebook dans le répertoire jupyter/ pour un accès via JupyterHub.
- **Création du répertoire Results/ :** Mise en place d'un dossier dédié pour enregistrer les résultats des traitements effectués.

Amazon S3 > Compartiments > projet9-data-scientist

projet9-data-scientist Info

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

Objets (11) Info Copier l'URI Copier l'URL Télécharger Ouvrir Supprimer **Actions ▼** Créer un dossier Charger

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](#)

<input type="checkbox"/>	Nom ▲	Type ▼	Dernière modification ▼	Taille ▼	Classe de stockage ▼
<input type="checkbox"/>	bootstrap-emr.sh	sh	01 Oct 2024 05:09:44 PM CEST	938.0 o	Standard
<input type="checkbox"/>	j-2SMVG8QT59BC9/	Dossier	-	-	-
<input type="checkbox"/>	jupyter/	Dossier	-	-	-
<input type="checkbox"/>	LICENSE	-	27 Sep 2024 11:38:36 AM CEST	1.1 Ko	Standard
<input type="checkbox"/>	Paire-de-cle-dataP9.ppk	ppk	01 Oct 2024 01:38:24 PM CEST	1.4 Ko	Standard
<input type="checkbox"/>	papers/	Dossier	-	-	-
<input type="checkbox"/>	readme.md	md	27 Sep 2024 11:58:45 AM CEST	8.0 Ko	Standard
<input type="checkbox"/>	Results/	Dossier	-	-	-
<input type="checkbox"/>	test-multiple_fruits/	Dossier	-	-	-
<input type="checkbox"/>	Test/	Dossier	-	-	-
<input type="checkbox"/>	Training/	Dossier	-	-	-



❑ Mise en place d'un cluster de calcul distribué avec EMR

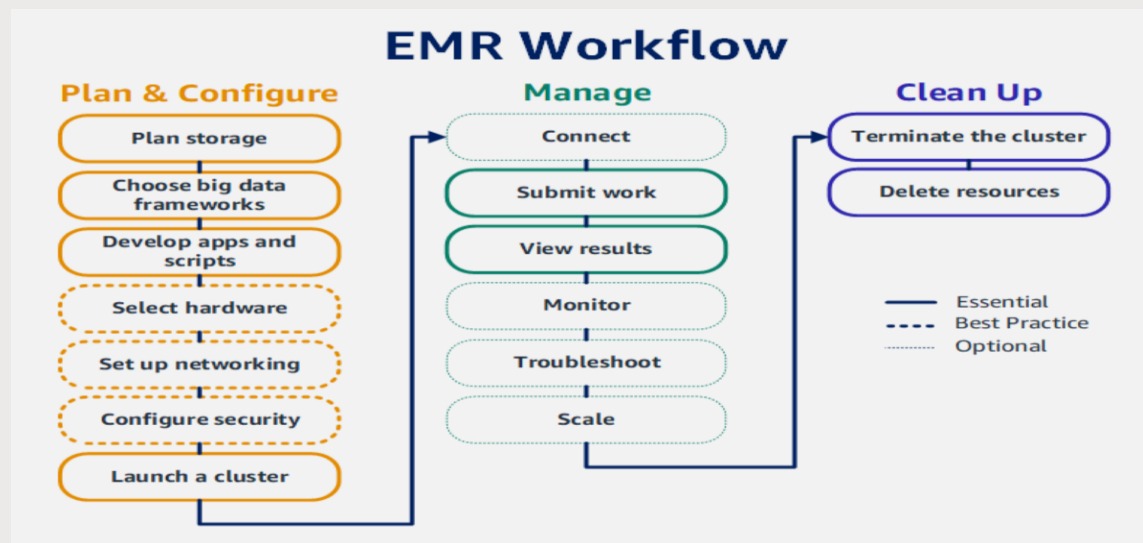
Paris ▼

Khoty ▼



EMR

- **Respect des normes RGPD** : Utilisation de serveurs localisés en Europe pour se conformer au RGPD, en sélectionnant la même **région pour les serveurs EC2 et S3** : Paris (eu-west-3).
- **Présentation d'Elastic MapReduce (EMR)** : Plateforme dédiée à l'exécution de traitements de données distribuées à grande échelle, s'appuyant sur les frameworks Hadoop et Spark. EMR utilise des instances EC2 (Elastic Compute Cloud), préinstallées avec les applications nécessaires et prêtes à être configurées.
- **Étapes de configuration du cluster EMR** :
 - Paramétrage des logiciels
 - Configuration matérielle
 - Actions d'amorçage (bootstrap) pour préparer l'environnement
 - Définition des options de sécurité pour protéger l'infrastructure et les données





❑ Configuration d'un cluster de calcul distribué avec EMR

- Configuration des logiciels :
 - Sélectionner la version d'EMR souhaitée (emr-6.7.0).
 - Choisir les applications à installer sur le cluster.
 - Conserver l'Amazon Machine Image (AMI) par défaut, basée sur Amazon Linux.
- Paramétrage de la persistance des notebooks : Configurer la persistance des notebooks créés et ouverts via JupyterHub, en utilisant une configuration au format JSON.



Nom
Cluster Spark opc-P9

Version Amazon EMR [Info](#)
Une version contient un ensemble d'applications susceptibles d'être installées sur votre cluster.
emr-6.7.0

Offre d'applications

Spark	Core Hadoop	HBase	Presto	Trino	Custom
-------	-------------	-------	--------	-------	--------

☐ Flink 1.14.2
☐ HCatalog 3.1.3
☐ Hue 4.10.0
☐ Livy 0.7.1
☐ Phoenix 5.1.2
☒ Spark 3.2.1
☐ Tez 0.9.2
☐ ZooKeeper 3.5.7

☐ Ganglia 3.7.2
☒ Hadoop 3.2.1
☐ JupyterEnterpriseGateway 2.1.0
☐ MXNet 1.8.0
☐ Pig 0.17.0
☐ Sqoop 1.4.7
☐ Trino 378

☐ HBase 2.4.4
☐ Hive 3.1.3
☒ JupyterHub 1.4.1
☐ Oozie 5.2.1
☐ Presto 0.272
☒ TensorFlow 2.4.1
☐ Zeppelin 0.10.0

Options du système d'exploitation [Info](#)

☒ Version Amazon Linux :
☐ Amazon Machine Image (AMI) personnalisée

☒ Appliquez automatiquement les dernières mises à jour Amazon Linux

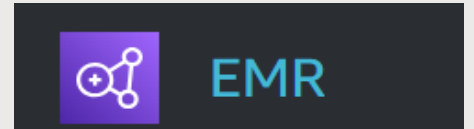
▼ Paramètres du logiciel [Info](#)
Remplacez les configurations par défaut pour des applications spécifiques de votre cluster.

☒ Entrer la configuration
☐ Charger JSON à partir d'Amazon S3

```
1 {  
2   "Classification": "jupyter-s3-conf",  
3   "Properties": {  
4     "s3.persistance.enabled": "true",  
5     "s3.persistance.bucket": "projet9-data_scientist"  
6   }  
7 }  
8  
9 }
```



❑ Configuration matérielle d'un cluster de calcul distribué avec EMR



- **Instances à configurer :**
 - **1 instance maître** (ou **master node**), responsable de la coordination du cluster.
 - **2 instances principales** (ou **core nodes**), qui assurent le traitement des données et l'exécution des tâches.
 - **1 instance de tâche** (ou **task node**), dédiée à l'exécution de tâches supplémentaires non persistantes.
- **Type d'instances :**
 - Utilisation d'instances de type **M5** (équilibré en termes de performance), avec des configurations **xlarge**, étant les options les plus économiques disponibles. Chaque instance dispose de **4 CPU virtuelles** et de **16 Go de RAM**.

▼ **Configuration de cluster - requies** [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 instanc...
Prix Spot le plus bas : 0.072 USD (eu-west-3a)

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 instanc...
Prix Spot le plus bas : 0.072 USD (eu-west-3a)

Actions ▼

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

Choisir une option

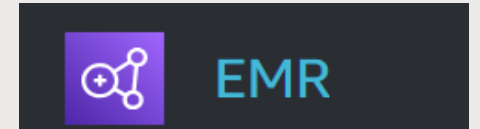
☒ **Définir manuellement la taille du cluster**
Utilisez cette option si vous connaissez vos modèles de charge de travail à l'avance.

☐ **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 personnalisée**
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 tâche groupes 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
Tâche - 1	m5.xlarge	<input type="text" value="2"/>	<input type="checkbox"/>
Unité principale	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>



❑ Amorçage du cluster de calcul distribué avec EMR

- **Objectif :** Installer les packages requis sur toutes les machines du cluster, et pas seulement sur le nœud maître (driver).
- **Étapes préparatoires :**
 - Dresser la liste des packages nécessaires à l'exécution du notebook.
 - Créer le script d'amorçage intitulé "**bootstrap-emr.sh**", contenant les commandes d'installation des packages via **pip**.
 - Charger ce script dans le compartiment S3 pour qu'il soit accessible par toutes les instances du cluster.

▼ 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				
bootstrap-amorçage	s3://projet9-data-scientist/bootstrap-emr.sh	-				

```
#!/bin/bash
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas==1.2.5
sudo python3 -m pip install -U pip
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
```

❑ Sécurité



- Sélection de la paire de clés EC2 : Choisir une paire de clés Amazon EC2 pour permettre l'accès sécurisé au cluster via SSH.
- Autorisation du profil d'instance EC2 : Configurer les permissions pour que le profil d'instance EC2 puisse accéder au bucket S3.

Paire de clés Amazon EC2 pour SSH sur le cluster Info

Q Paire-de-cle-dataP9 X Parcourir

Créer une paire de clés

[IAM](#) > [Rôles](#) > [EC2_Instance_DefaultRole](#)

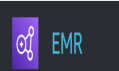
[AmazonS3FullAccess](#)

Gérées par AWS

- wheel : Optimise le processus d'installation des packages Python.
- pillow : Utilisé pour le traitement et la manipulation d'images.
- pyarrow : Permet la lecture des fichiers au format Parquet et leur conversion en DataFrame.
- boto3, s3fs, fsspec : Bibliothèques pour interagir et gérer les données sur S3.



Instanciation de l'EMR



Après plusieurs tentatives et la résolution de divers problèmes, la dernière configuration a fonctionné parfaitement de bout en bout. J'ai ensuite résilié l'instance manuellement afin de limiter les coûts.



Amazon EMR > EMR sur EC2: Clusters										
Clusters (11) Info						Afficher les détails		Résilier	Cloner	Créer un cluster
Filtrer les clusters par statut			Rechercher des clusters			Filtrer les clusters par date et heure de création		< 1 >		
		ID de cluster	Nom du cluster	Statut	Heure de création (UTC+02:00)	Temps écoulé	Heures d'instances normalisées			
<input type="checkbox"/>		j-1JB39RBADE6Z7	Cluster Spark opc-P9	⊖ Résilié Demande utilisateur	1 octobre 2024 18:54	3 heures, 25 minutes	128			
<input type="checkbox"/>		j-265SBOWJNOFP7	Cluster Spark opc-P9	⊖ Résilié Demande utilisateur	1 octobre 2024 17:12	1 heure, 2 minutes	32			
<input type="checkbox"/>		j-1OZOYQC5K8075	MonclusterP9	⊖ Résilié Demande utilisateur	1 octobre 2024 13:42	1 heure, 19 minutes	64			
<input type="checkbox"/>		j-1S1G6I5ZZGO55	Cluster Spark opc-P9	⊖ Résilié Arrêt automatique	1 octobre 2024 03:52	1 heure, 13 minutes	48			
<input type="checkbox"/>		j-2W8GQEDN25VED	Cluster Spark opc-P9	⊖ Résilié Arrêt automatique	1 octobre 2024 00:29	1 heure, 13 minutes	48			
<input type="checkbox"/>		j-34JZXHZRWQJIE	Cluster Spark opc-P9	⊖ Résilié Arrêt automatique	30 septembre 2024 20:46	1 heure, 40 minutes	48			
<input type="checkbox"/>		j-2SMVG8QT59BC9	projet9 Cluster	⊖ Résilié Arrêt automatique	29 septembre 2024 21:57	1 heure, 13 minutes	48			
<input type="checkbox"/>		j-2YDHIZL1AM4V1	mon projet 9	⊖ Résilié Arrêt automatique	28 septembre 2024 00:03	1 heure, 14 minutes	48			
<input type="checkbox"/>		j-307NSVQ05XG6V	mon projet 9	⊖ Résilié Arrêt automatique	27 septembre 2024 17:30	1 heure, 13 minutes	48			
<input type="checkbox"/>		j-2G8LS435BNE5O	mon projet 9	⊖ Résilié Arrêt automatique	27 septembre 2024 13:13	1 heure, 13 minutes	48			
<input type="checkbox"/>		j-2A6RYIEIRNE3	Cluster Spark Projet 9	⊖ Résilié Arrêt automatique	24 septembre 2024 16:33	2 heures, 28 minutes	72			

Amazon EMR > EMR sur EC2: Clusters > Cluster Spark opc-P9

Mise à jour il y a moins d'une minute

Résilier

Cloner dans AWS CLI

Cloner

Récapitulatif

Informations sur le cluster

ID de cluster
j-1JB39RBADE6Z7

Configuration de cluster
Groupes d'instances

Capacité
1 primaire(s) 1 unité(s) principale(s) 2 tâche(s)

Applications

Version d'Amazon EMR
emr-6.7.0

Applications installées
Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1

Gestion des clusters

Destination des journaux dans Amazon S3
aws-logs-180294206541-eu-west-3/elasticmapreduce

Interfaces utilisateur d'application persistantes
Serveur d'historique Spark

Serveur de chronologie YARN

DNS public du nœud primaire
ec2-15-237-27-131.eu-west-3.compute.amazonaws.com

Connexion au nœud primaire à l'aide de SSH

Statut et heure

Statut
⊖ Résilié

Heure de création
1 octobre 2024 18:54 (UTC+02:00)

Temps écoulé
3 heures, 25 minutes

Heure de fin
1 octobre 2024 22:19 (UTC+02:00)








❑ Création d'un tunnel SSH vers l'instance EC2

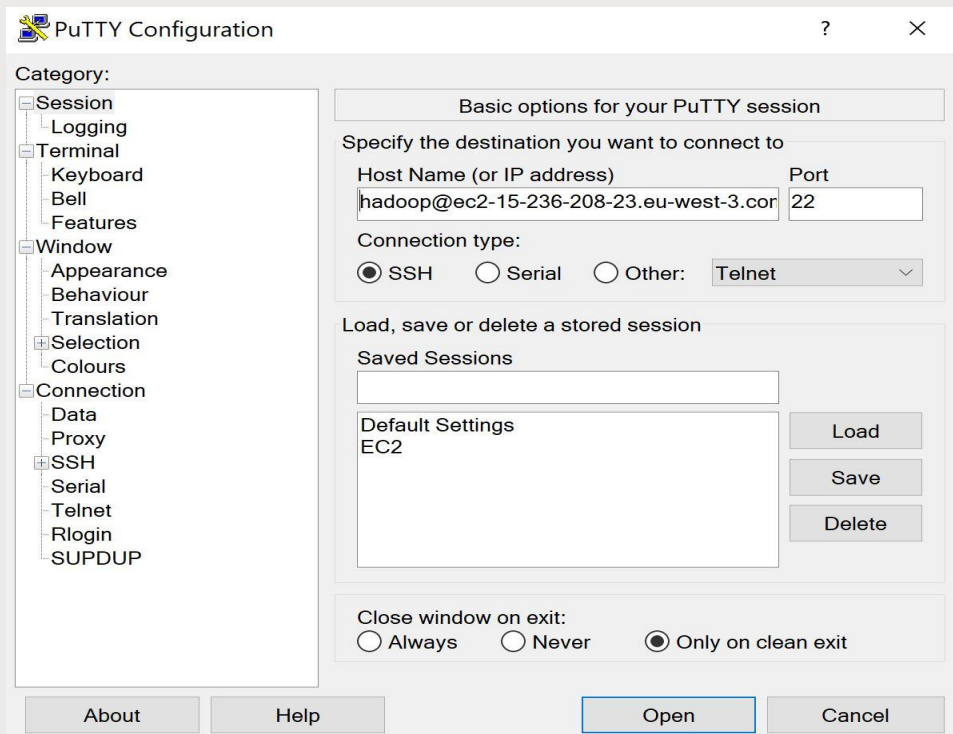
- **Connexion sécurisée :** Établir une connexion sécurisée entre la machine virtuelle (driver) et la machine locale pour accéder aux applications du serveur **EMR**, telles que :
 - **JupyterHub** pour exécuter le notebook (port 9443)
 - **Serveur d'historique Spark** pour analyser l'exécution des tâches (port 18080)
- **Modification du groupe de sécurité EC2 :**
 - Configurer le groupe de sécurité du driver EC2 pour autoriser les connexions entrantes via le port 22 (port utilisé par le serveur SSH).
- **Création du tunnel SSH vers le driver :**
 - Utiliser PuTTY (sous Windows) et une clé .ppk pour établir le tunnel SSH.
- **Redirection via FoxyProxy :**
 - Configurer **FoxyProxy** dans le navigateur pour rediriger les requêtes vers le port 8157 et emprunter le tunnel SSH.

Interfaces utilisateur d'application sur le nœud primaire

Activer une connexion SSH

Celles-ci nécessitent l'activation du tunneling SSH.

Application	URL de l'interface utilisateur 
Gestionnaire de ressources	 http://ec2-15-236-208-23.eu-west-3.compute.amazonaws.com:8088/
JupyterHub	 https://ec2-15-236-208-23.eu-west-3.compute.amazonaws.com:9443/
Nom du nœud HDFS	 http://ec2-15-236-208-23.eu-west-3.compute.amazonaws.com:9870/
Serveur d'historique Spark	 http://ec2-15-236-208-23.eu-west-3.compute.amazonaws.com:18080/



```

🖥️ Using username "hadoop".
🖥️ Authenticating with public key "Paire-de-cle-dataP9"

```

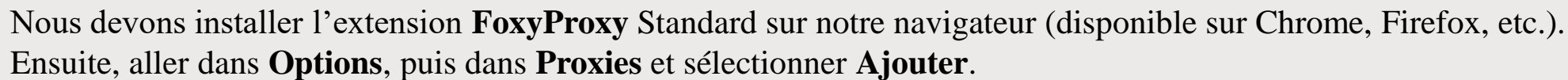
```
A newer release of "Amazon Linux" is available.
  Version 2023.5.20240903:
  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
```

```

#
~\#### Amazon Linux 2023
~~~\#####\
~~~\####|
~~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~~V~'-'>
~~~~
~~~.
~~~\
~~~\m/'
Last login: Tue Oct 1 11:47:13 2024

```

EEEEEEEEEEEEEEEEEEEE	MMMMMMM	MMMMMMM	RRRRRRRRRRRRRRR			
E:::::::::::::::::E	M::::::::M	M::::::::M	R:::::::::::::::::R			
EE:::::EEEEEEEE:::E	M:::::::::M	M:::::::::M	R:::::RRRRRR:::::R			
E:::::E	EEEE	M:::::::::M	M:::::::::M	RR::::R	R:::::R	
E:::::E	M:::::M:::M	M:::M:::::M	R:::R	R:::::R		
E:::::EEEEEEEE	M:::::M	M:::M	M:::::M	R:::RRRRRR:::::R		
E:::::::::::::::::E	M:::::M	M:::M:::M	M:::::M	R:::::::::::::::::RR		
E:::::EEEEEEEE	M:::::M	M:::::M	M:::::M	R:::RRRRRR:::::R		
E:::::E	M:::::M	M:::M	M:::::M	R:::R	R:::::R	
E:::::E	EEEE	M:::::M	MM	M:::::M	R:::R	R:::::R
EE:::::EEEEEEEE:::E	M:::::M	M:::::M	R:::R	R:::::R		
E:::::::::::::::::E	M:::::M	M:::::M	RR:::R	R:::::R		
EEEEEEEEEEEEEEEEEEEE	MMMMMMM	MMMMMMM	RRRRRR	RRRRR		



Paramètres à configurer :

- **Type** : SOCKS5
- **HostName** : localhost
- **Port** : 8157 (ce port doit correspondre à celui utilisé précédemment pour la configuration SSH).

Proxy_8157_EMR

Nom ou Description (optionnel)

Proxy_8157_EMR

Hostname

localhost

Type

SOCKS5

Port

8157

Country

Nom d'utilisateur (optionnel)

username

City

city

Mot de passe (optionnel)

Couleur

PAC URL

PAC URL

Proxy DNS

Store Locally

Quick Add

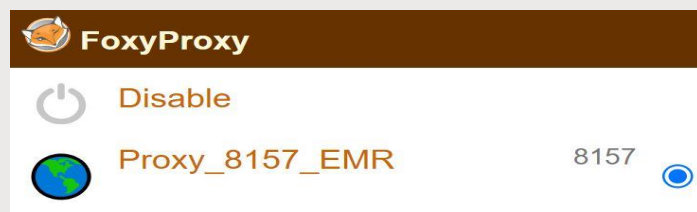
Include

Type

Nom ou Description (optionnel)

Modèles

- L'option devrait maintenant apparaître, et nous devons l'activer.





❑ Exécution du script PySpark

- **Lancement manuel** : Exécution du script directement sur le cluster via **JupyterHub**, en utilisant le kernel **PySpark**.
- **Concepts Spark appliqués** : Mise en œuvre des notions clés de **Spark**, notamment les **transformations**, les **actions** et l'**exécution différée** (lazy execution), illustrées par l'utilisation des commandes comme **.show()** et **yield**.
- Le cluster a traité les **22 688 images**.
- L'étape la plus longue a été la **réduction des dimensions**, qui a duré environ **17 minutes** (correspondant à la cellule 20, ou statement 19), et a nécessité **7 jobs**

Les informations par défaut sont :

Username : jovyan

Password : jupyter

Sign in

Username:

Password:

Sign in

Ouverture Pyspark et Librairies

▶ # L'exécution de cette cellule démarre l'application Spark

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1727802056491_0001	pyspark	idle	Link	Link	None	✓

```
Entrée [20]: ▶ # Application de L'algorithme PCA (Principal Component Analysis) sur les vecteurs de caractéristiques

pca = PCA(k=PCA_K, inputCol = 'features_vectors', outputCol = 'pca_vectors')
model = pca.fit(features_df)
df_pca = model.transform(features_df)
df_pca.show(5, True)
```

11 (19)	Groupe de travail pour l'instruction 19 showString à NativeMethodAccessorImpl.java:0	01/10/2024 18:17:59	22 s	1/1 (1 sautés)	1/1 (709 sautés)
10 (19)	Groupe de travail pour l'instruction 19 showString à NativeMethodAccessorImpl.java:0	01/10/2024 18:12:54	5,1 min	1/1	709/709
9 (19)	Groupe de travail pour l'instruction 19 treeAggregate à RowMatrix.scala:156	01/10/2024 18:09:49	2,9 minutes	2/2 (1 sautés)	28/28 (709 sautés)
8 (19)	Groupe de travail pour l'instruction 19 est vide à RowMatrix.scala:426	01/10/2024 18:09:31	17 s	1/1 (1 sautés)	1/1 (709 sautés)
7 (19)	Groupe de travail pour l'instruction 19 treeAggregate à Statistics.scala:58	01/10/2024 18:06:36	2,9 minutes	2/2 (1 sautés)	28/28 (709 sautés)
6 (19)	Groupe de travail pour l'instruction 19 premier à RowMatrix.scala:62	01/10/2024 18:06:09	27 s	1/1 (1 sautés)	1/1 (709 sautés)
5 (19)	Groupe de travail pour l'instruction 19 premier sur PCA.scala:44	01/10/2024 18:00:29	5,7 minutes	2/2	710/710



❑ Analyse de l'historique d'exécution : Spark Web UI (Spark History Server, port 18080)

- **Vue détaillée des tâches** : La **Spark Web UI** offre une vision précise de l'exécution des différentes tâches sur les machines du cluster. Lors du traitement, Spark segmente l'application en **jobs**, **stages**, et **tasks**.
- **Persistance de l'historique** : L'historique de l'exécution est conservé, permettant de suivre l'application de manière persistante.
- **Parallélisation et optimisation des ressources** : Cette interface fournit la preuve de la bonne parallélisation des calculs et du dimensionnement adéquat des ressources pour garantir une exécution optimale.

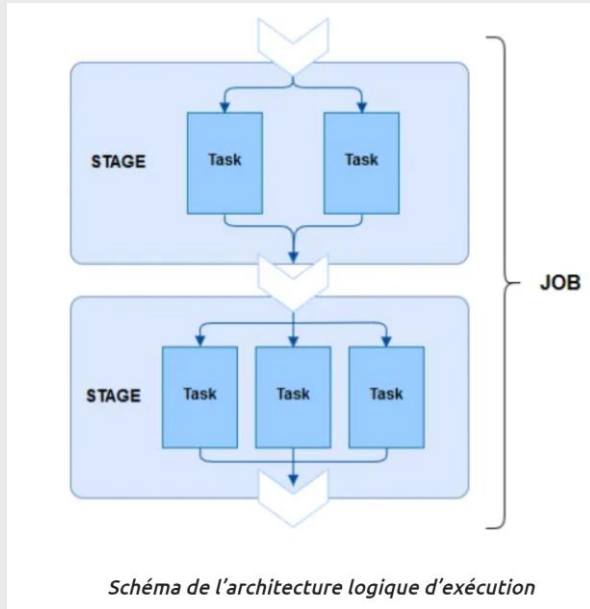
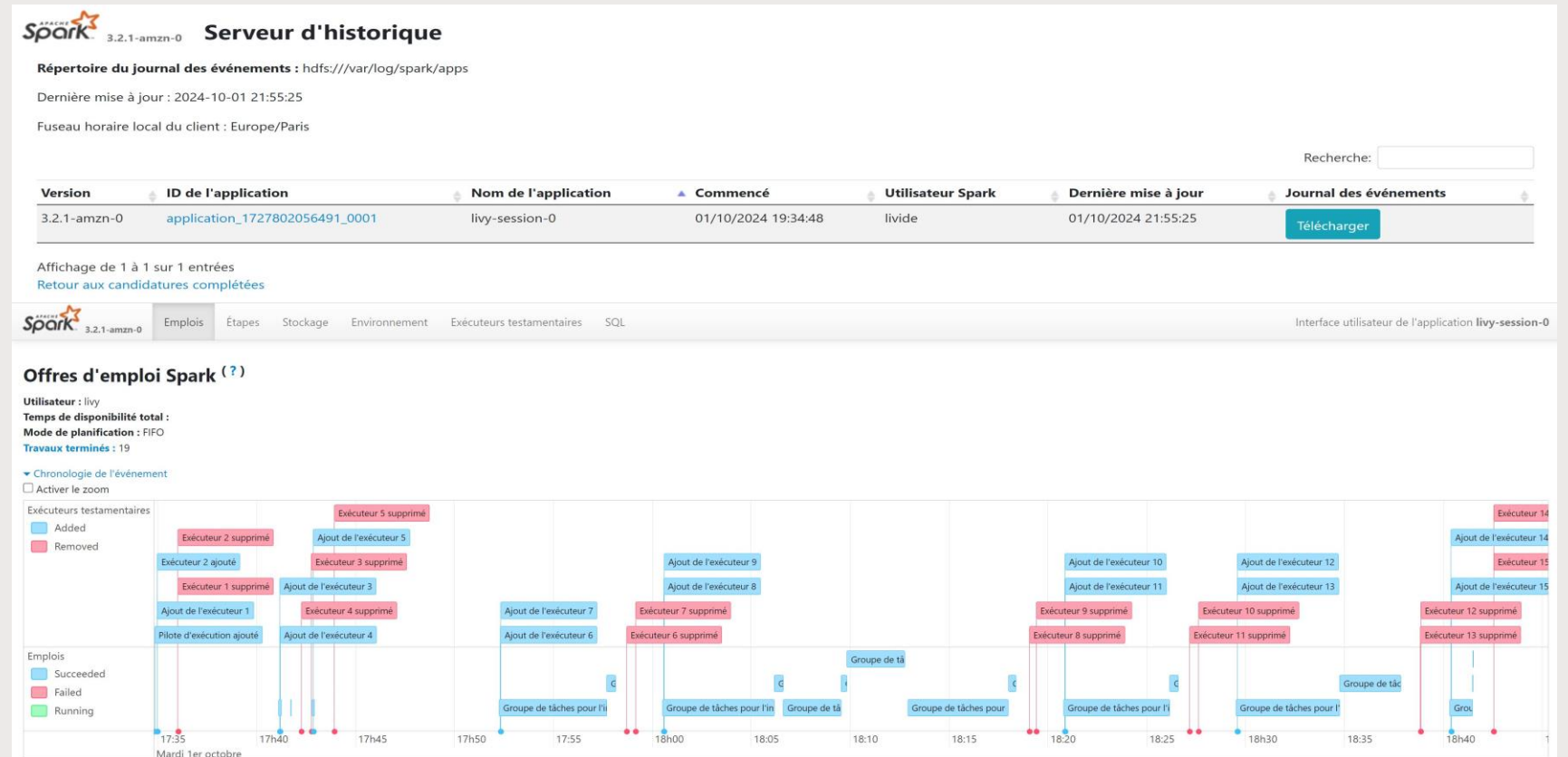


Schéma de l'architecture logique d'exécution

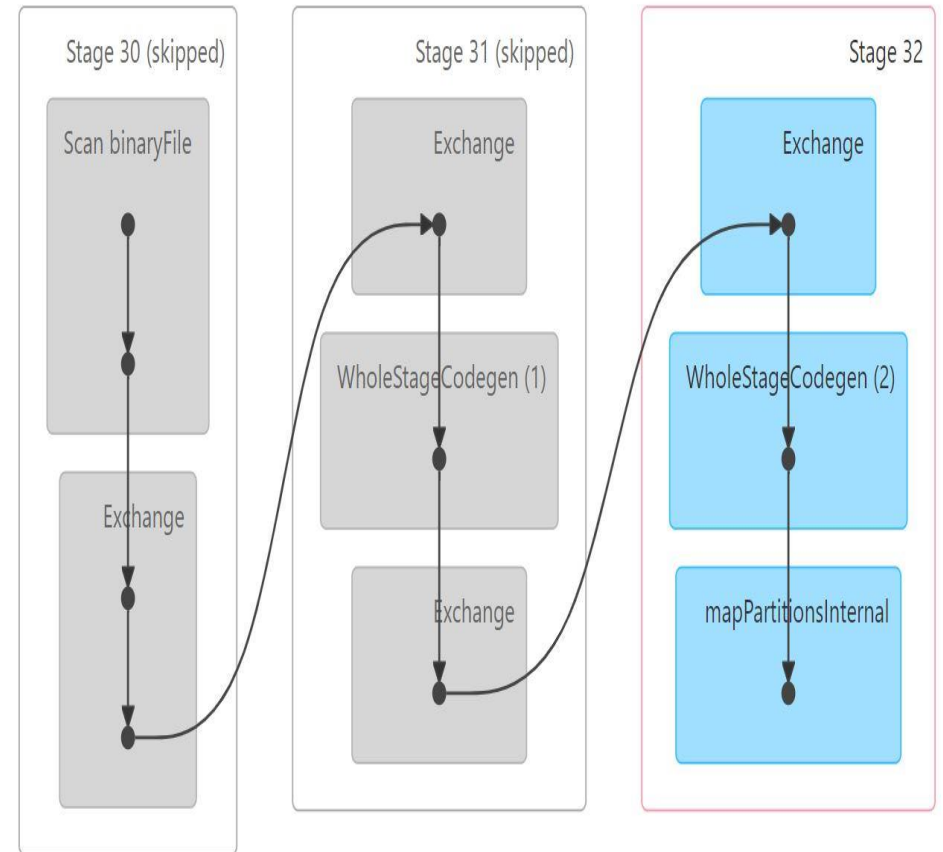




❑ DAG = Directed Acyclic Graph

- **Exchange** : Représente la redistribution des données à travers les **workers** pour assurer l'équilibrage de la charge de travail dans les différentes phases du traitement.
- **WholeStageCodegen** : Technique d'optimisation utilisée par Spark pour générer du code à la volée, ce qui permet de réduire les surcharges associées aux opérations répétitives et d'améliorer l'efficacité des transformations en une seule étape.
- **mapPartitionsInternal** : Représente la parallélisation des traitements à travers les différentes partitions de données, optimisant l'exécution en distribuant les calculs entre les **workers**.

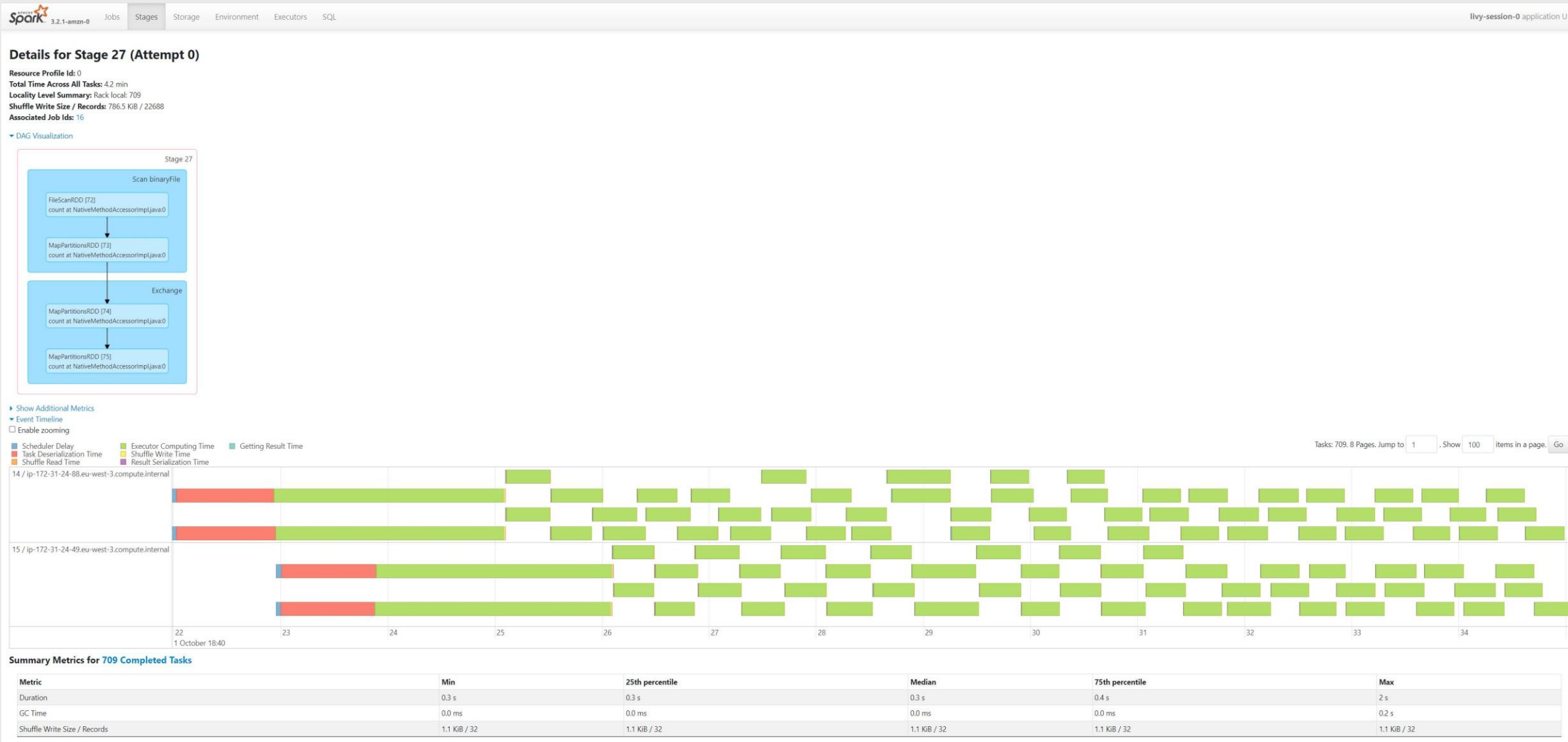
▼ DAG Visualization





Extraction des features – DAG SparkUI

Visualisation SparkUI pour l'extraction des features





❑ Résultats finaux du traitement : stockage des données

Les résultats du traitement sont enregistrés sous forme de **24 fichiers** au format **Parquet**, correspondant aux **24 partitions** générées. Le format Parquet, optimisé pour les performances grâce à son architecture basée sur les colonnes, permet une lecture rapide et efficace des données. De plus, il utilise la compression **Snappy** pour réduire la taille des fichiers tout en maintenant des performances élevées.

Amazon S3 > Compartiments > projet9-data-scientist > Results/

Results/

Copier l'URI S3

Objets

Propriétés

Objets (25) Info

🔄

Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Actions

Créer un dossier

Charger

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

< 1 > ⚙️

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	_SUCCESS	-	01 Oct 2024 08:37:50 PM CEST	0 o	Standard
<input type="checkbox"/>	part-00000-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:23 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00001-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:23 PM CEST	11.5 Mo	Standard
<input type="checkbox"/>	part-00002-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:23 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00003-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:23 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00004-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:53 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00005-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:53 PM CEST	11.3 Mo	Standard
<input type="checkbox"/>	part-00006-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:52 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00007-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:35:52 PM CEST	11.4 Mo	Standard
<input type="checkbox"/>	part-00008-608534b8-f865-45ed-83ce-597e1cf6737c-c000.snappy.parquet	parquet	01 Oct 2024 08:36:21 PM CEST	11.3 Mo	Standard

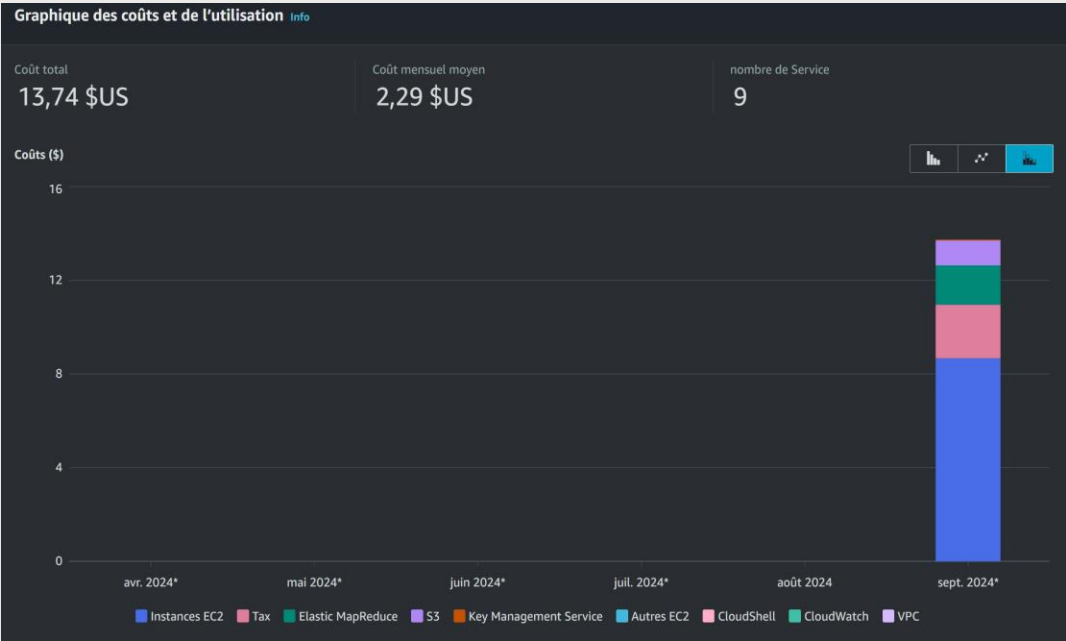


❑ Facturation AWS : exploration, suivi et réduction des coûts

Le graphique montre un coût total de **13,74 \$US**, avec un coût mensuel moyen de **2,29 \$US**. Ce coût résulte de l'utilisation de **9 services AWS** distincts, principalement concentrés sur le mois de septembre 2024.

- Le service **EC2** représente la plus grande part des dépenses, suivi par des services comme **Elastic MapReduce (EMR)**, **S3**, et des services auxiliaires tels que **CloudWatch**, **CloudShell**, et le **Key Management Service**.

Il est essentiel de surveiller ces coûts pour optimiser l'utilisation des ressources AWS et identifier les moyens de réduire les dépenses, en particulier sur les services EC2.



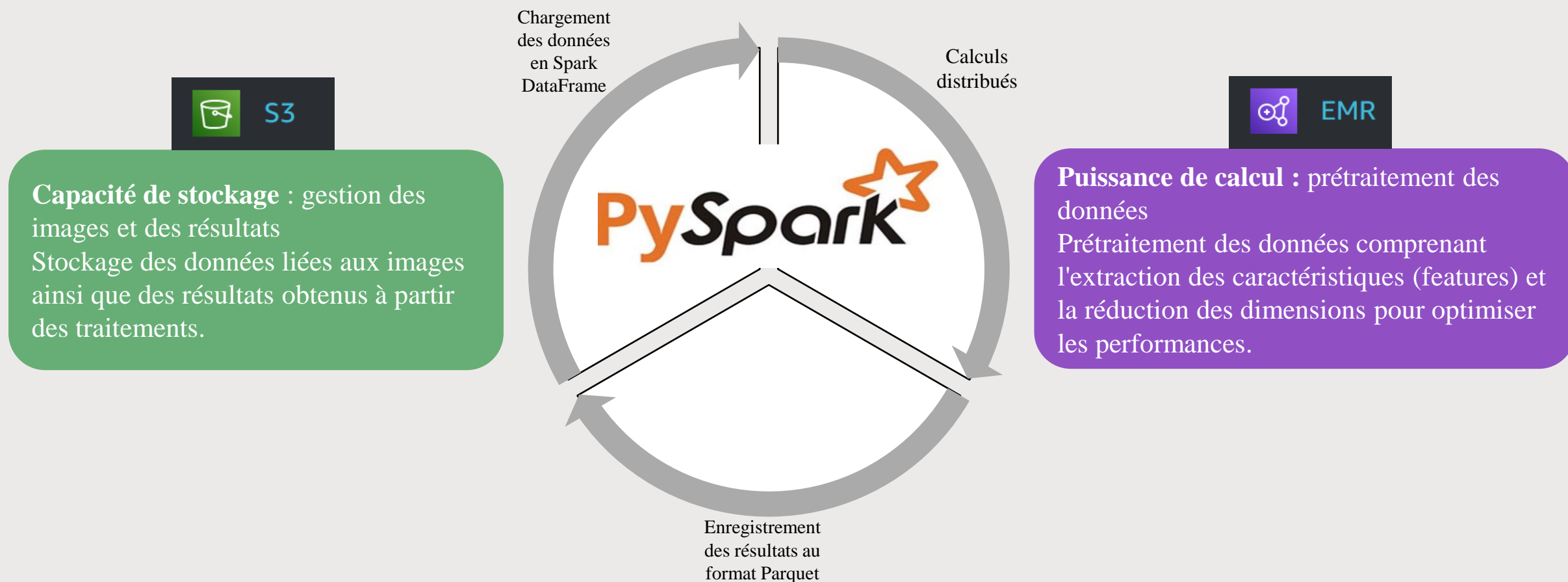
Répartition des coûts et de l'utilisation Télécharger au format CSV

🔍 Trouver des données sur les coûts et l'utilisation

	Total	avril 2024*	mai 2024*	juin 2024*	juillet 2024*	août 2024	septembre 2024*
Total des coûts	13,74 \$US	0,00 \$US	0,00 \$US	0,00 \$US	0,00 \$US	0,00 \$US	13,74 \$US
Instances EC2	8,67 \$US	-	-	-	-	-	8,67 \$US
Tax	2,29 \$US	-	-	-	-	0,00 \$US	2,29 \$US
Elastic MapReduce	1,68 \$US	-	-	-	-	-	1,68 \$US
S3	1,06 \$US	-	-	-	-	0,00 \$US	1,06 \$US
Key Management Service	0,04 \$US	-	-	-	-	-	0,04 \$US
Autres EC2	0,00 \$US	-	-	-	-	-	0,00 \$US
CloudShell	0,00 \$US	-	-	-	-	0,00 \$US	0,00 \$US
CloudWatch	0,00 \$US	-	-	-	-	0,00 \$US	0,00 \$US
VPC	0,00 \$US	-	-	-	-	-	0,00 \$US



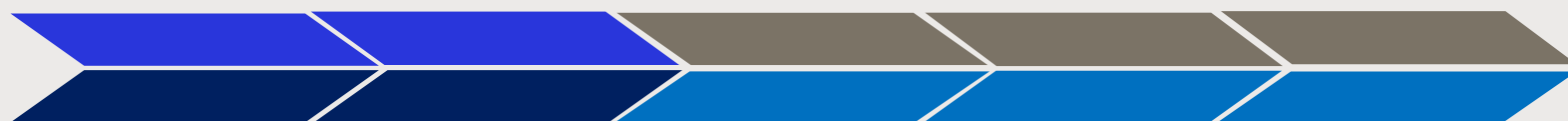
3. Processus de traitement des données





❑ Étapes du pipeline de traitement

- Dans ce projet, nous avons développé un pipeline de traitement de données capable de manipuler et d'analyser de grandes quantités d'images, en exploitant la puissance du calcul distribué avec **PySpark** et l'infrastructure scalable d'**AWS**. Le processus commence par le chargement des images, suivi d'une extraction des caractéristiques clés, incluant la réduction des dimensions pour optimiser les performances. Les résultats sont ensuite enregistrés au format Parquet pour un stockage et une gestion efficace.



Chargement des images

Importer les
images pour le
traitement

Préparation du modèle

Préparer un
modèle
d'extraction de
caractéristiques
avec Transfer
Learning.

Diffusion des poids du modèle

Appliquer les
poids du modèle
pour améliorer
l'efficacité du
traitement.

Extraction des caractéristiques

Extraire les
caractéristiques
visuelles des
images et
réduire leurs
dimensions.

Enregistrement des résultats

Sauvegarder les
résultats après
le traitement.





4. Synthèse finale

L'utilisation du cloud AWS offre une infrastructure puissante et scalable, permettant de gérer et de traiter un très grand volume de données, notamment dans le cadre de projets collaboratifs impliquant plusieurs équipes responsables de différents composants (stockage, EMR, EC2, etc.). Toutefois, ces avantages s'accompagnent de coûts non négligeables : un cluster EMR EC2, même avec deux exécuteurs à bas coût, peut engendrer des frais d'environ 1\$ par heure, et le nombre de **requêtes gratuites (20 000)** est rapidement dépassé.

Pour anticiper l'augmentation future de la charge de travail liée au traitement des images de fruits, le projet a été divisé en deux phases. La première phase consistait à développer et valider localement le pipeline de traitement sur Google Colaboratory, en y intégrant une étape clé de réduction de dimensions. Cette étape a permis de se familiariser avec **Spark** et le calcul distribué, tout en assurant le bon fonctionnement de la solution.

La deuxième phase a consisté à déployer un véritable cluster de calcul sur AWS en utilisant **EMR**, **EC2**, et **S3**, tout en configurant les librairies essentielles comme **Spark**, **Hadoop**, **JupyterHub** et **TensorFlow**. Ce cluster a permis d'exécuter le traitement sur l'ensemble des images du jeu de test, tout en organisant efficacement l'accès aux données et aux résultats dans le Cloud.

Ce projet a donc démontré non seulement la faisabilité technique d'un traitement à grande échelle sur le Cloud, mais aussi la nécessité de surveiller attentivement les coûts afin d'assurer la viabilité économique de telles infrastructures.