P8 DÉPLOYEZ UN MODÈLE DANS LE CLOUD

Formation Data Scientist
OpenClassrooms


CRÉATION, CONFIGURATION DE L'ENVIRONNEMENT SUR AWS
ENVIRONNEMENT CLOUD
CONCLUSION
CONTEXT
MODELISATION
SCRIPT PYSPARK, EXTRACTION DE FEATURE PAR TRANSFERT LEARNING, RÉDUCTION DE DIMENSION

CONTEXT

CONTEXT

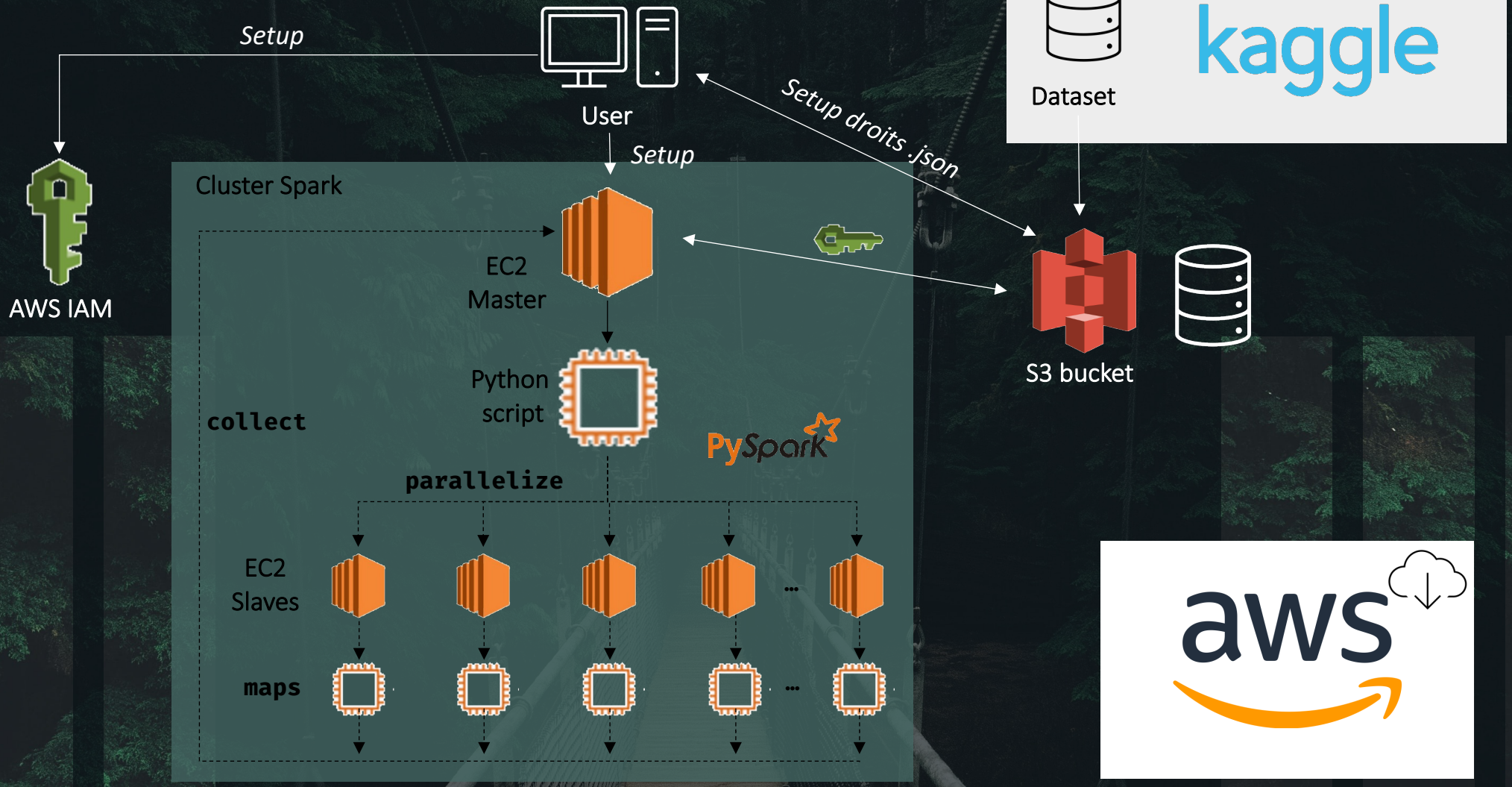
Fruits!

CONTEXT

Problématique : développer dans un environnement Big Data et une preuve de concept qui comprendra le preprocessing et une étape de réduction de dimension

ENVIRONNEMENT CLOUD

ENVIRONNEMENT CLOUD



ENVIRONNEMENT CLOUD

AWS Infrastructure as a Services

IAM

Création d'un user <ADMIN>

Paire de clés
Clé Access
Clé Secret

EC2

Set up de l'instance

Choix de solution
Taille: t2.medium
Stockage: 30 Go
OS: Linux 20.x, 64-bits

Groupe de sécurité (TCP SSH)

Set up une Elastic IP (IP statique)

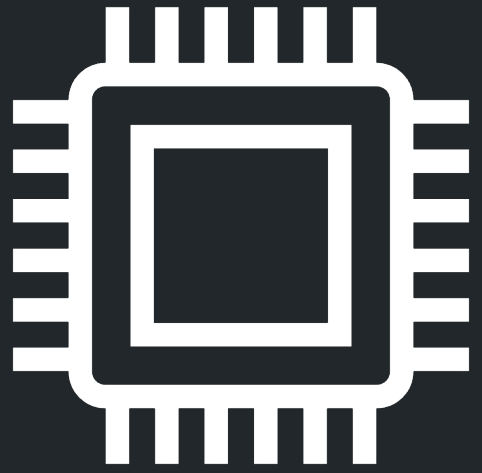
Communication avec l'instance avec protocole SSH (SCP)

S3

Création du bucket

Chargement des données dans des dossiers (= objets)

Gestion des droits
création d'une stratégie .json avec le générateur



MODELISATION

MODELISATION

PySpark

Setup spark session / context

Import données

Preprocessing

ResNet50

Extraction de features

Postprocessing (standard scaler)

PCA

kaggle

MODELISATION

- findspark: localise spark dans l'EC2
- boto3: gestion communication EC2-S3
- Ouverture d'une spark session

Config clés AMI et module hadoop-s3

- Création d'un context spark

Capable de communiqué avec S3

→ Import des données dans une dataframe

```
# Pyspark
import findspark
findspark.init()
import pyspark
```

```
# Check content of S3 bucket
connexion = boto3.client('s3')
contents = connexion.list_objects(Bucket='p8-devenv')['Contents']
for file in contents:
    print(file['Key'])
```

```
    os.environ['PYSPARK_SUBMIT_ARGS'] = '--packages com.amazonaws:aws-java-sdk-pom:1.10.34,org.apache.hadoop:hadoop-aws:2.7.2 pyspark-shell'

    accessKeyId='<YOUR_ACC_KEY>'
    secretAccessKey='<YOUR_SEC_KEY>'

    spark = (SparkSession
             .builder.master('local[*]')
             .appName('P8 - Déployez un modèle dans le cloud')
             .config('spark.hadoop.fs.s3a.access.key', accessKeyId)
             .config('spark.hadoop.fs.s3a.secret.key', secretAccessKey)
             .config('spark.hadoop.fs.s3a.impl', 'org.apache.hadoop.fs.s3a.S3AFileSystem')
             .getOrCreate()
             )

    sc = spark.sparkContext
    sc.setSystemProperty('com.amazonaws.services.s3.enableV4', 'true')
    sc._jsc.hadoopConfiguration().set("fs.s3a.endpoint", "s3.eu-west-1.amazonaws.com")
```

MODELISATION

Pre-processing using ResNet50

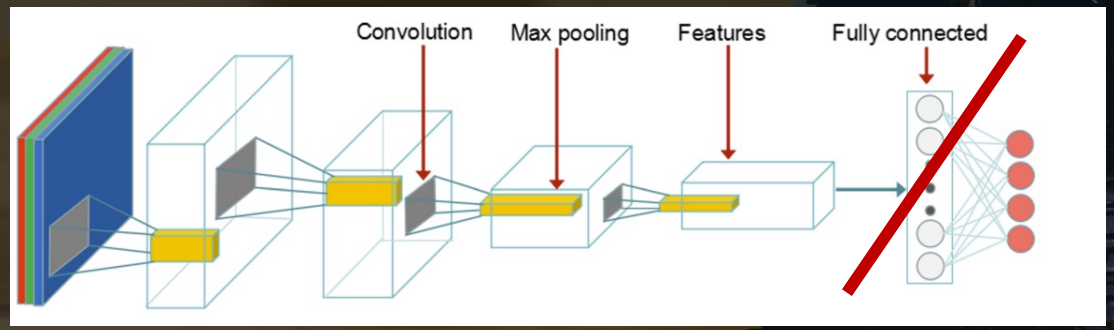
Utilisation de pandas UDF pour appliquer des fonctions sur dataframe

Objectif : extraction de features

Preprocessing :

- Resizing: 100 x 100 px → 224 x 224 px
- RGB → BGR
- Zero-centering each channel

Load ResNet50 weights and predict features :



Peng *et al.*, *European Radiology*, **2019**

Post-processing before PCA

- **Objectif** : formater les features pour la PCA
- Vector dense → création de n (feature) colonnes
- Standard Scaler

PCA

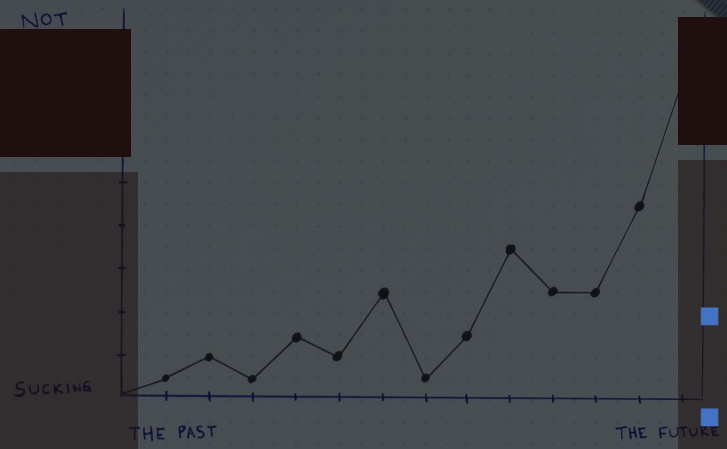
- Nombre PCs = 10
- Parallélisation = 5 CPUs
- Fit-transform time ~ 5 min

CONCLUSION

CONCLUSION

Set up solution Big Data dans le cloud

- Mise en place EC2, S3, IAM
- Configuration EC2
- Gestion des droits S3
- Principal difficulté : config EC2 pour un environnement spark fonctionnel



PoC preprocessing et PCA

- Communication avec S3
- Utilisation de pandas udf pour calculs sur dataframe
- Parallélisation sur plusieurs CPU
- Principal difficulté : peu de tutoriels online pour traitement d'image

QUESTIONS

```
##### connexion server
ssh -i "EC2_P8_pyspark.pem" ubuntu@ec2-3-250-200-52.eu-west-1.compute.amazonaws.com

#### Installation Spark Hadoop ect ..
sudo apt update

#Java
sudo apt install openjdk-8-jre-headless


#If the instance does not have python install -> Anaconda
'''sudo apt install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6
wget -P /tmp https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh
sha256sum /tmp/Anaconda3-2020.02-Linux-x86_64.sh
bash /tmp/Anaconda3-2020.02-Linux-x86_64.sh
source ~/.profile'''

#Scala
sudo apt install scala

#Spark
wget https://downloads.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop2.7.tgz

tar xvf spark-*
sudo mv spark-3.2.1-bin-hadoop2.7 /opt/spark

echo "export SPARK_HOME=/opt/spark" >> ~/.profile
echo "export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin" >> ~/.profile
echo "export PYSPARK_PYTHON=/usr/bin/python3.10" >> ~/.profile
source ~/.profile

#AWS cli
sudo apt install awscli
aws configure

sudo apt install python3-pip

pip install pandas tensorflow pillow findspark pyarrow boto3 fsspec s3fs seaborn

#### Install SWAP taille swap 2x RAM
https://shurn.me/blog/2017-02-13/swap-space-in-ec2-ubuntu

#### Send file to EC2
scp -i "EC2_P8_pyspark.pem" /Users/franck/Documents/P8_deployez_modele_cloud/pysparkApp_cloud.py ubuntu@ec2-54-173-139-254.compute-1.amazonaws.com:~
```