



Fruits!

P8 : Déployez un modèle dans le cloud

08/09/2022

DUBART Maxime

Déployez un modèle dans le cloud

**Développer une application pour la classification
d'images de fruits**

Mise en place d'une architecture Big Data

Déployez un modèle dans le cloud

Images de fruits / légumes

- 131 catégories
- env. 500 images par catégorie
- Taille 100x100 pixels



Mango



Avocado



Pear



Physalis

Déployez un modèle dans le cloud

Images de fruits / légumes

- 131 catégories
- env. 500 images par catégorie
- Taille 100x100 pixels



Mango



Avocado



Pear



Physalis

Objectif : réaliser une réduction de dimension

Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS

Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS



Stockage

Amazon S3

Réplication

Versionning

Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS

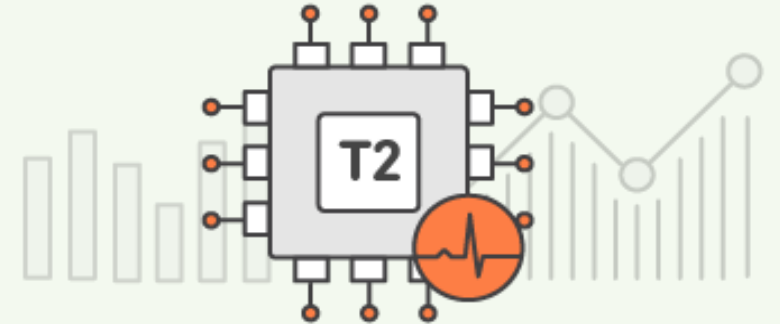


Stockage

Amazon S3

Réplication

Versionning



Calculs

Amazon EC2

(auto-)scaling

Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS



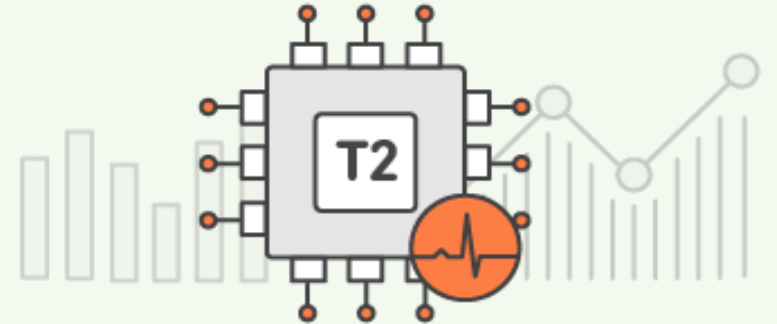
Stockage

Amazon S3
Réplication
Versionning



Access management

Amazon IAM
Lecture/Ecriture sur S3

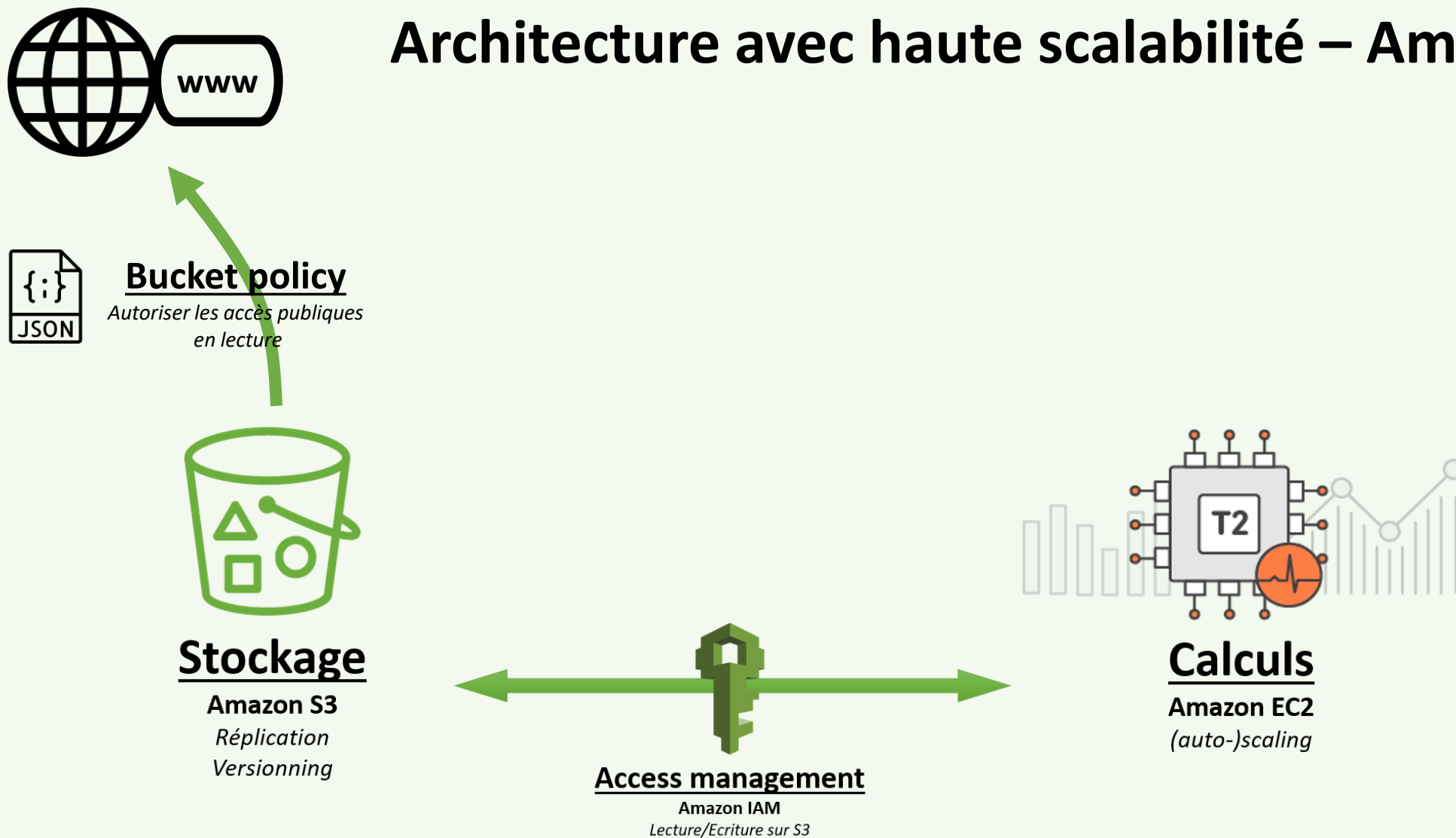


Calculs

Amazon EC2
(auto-)scaling

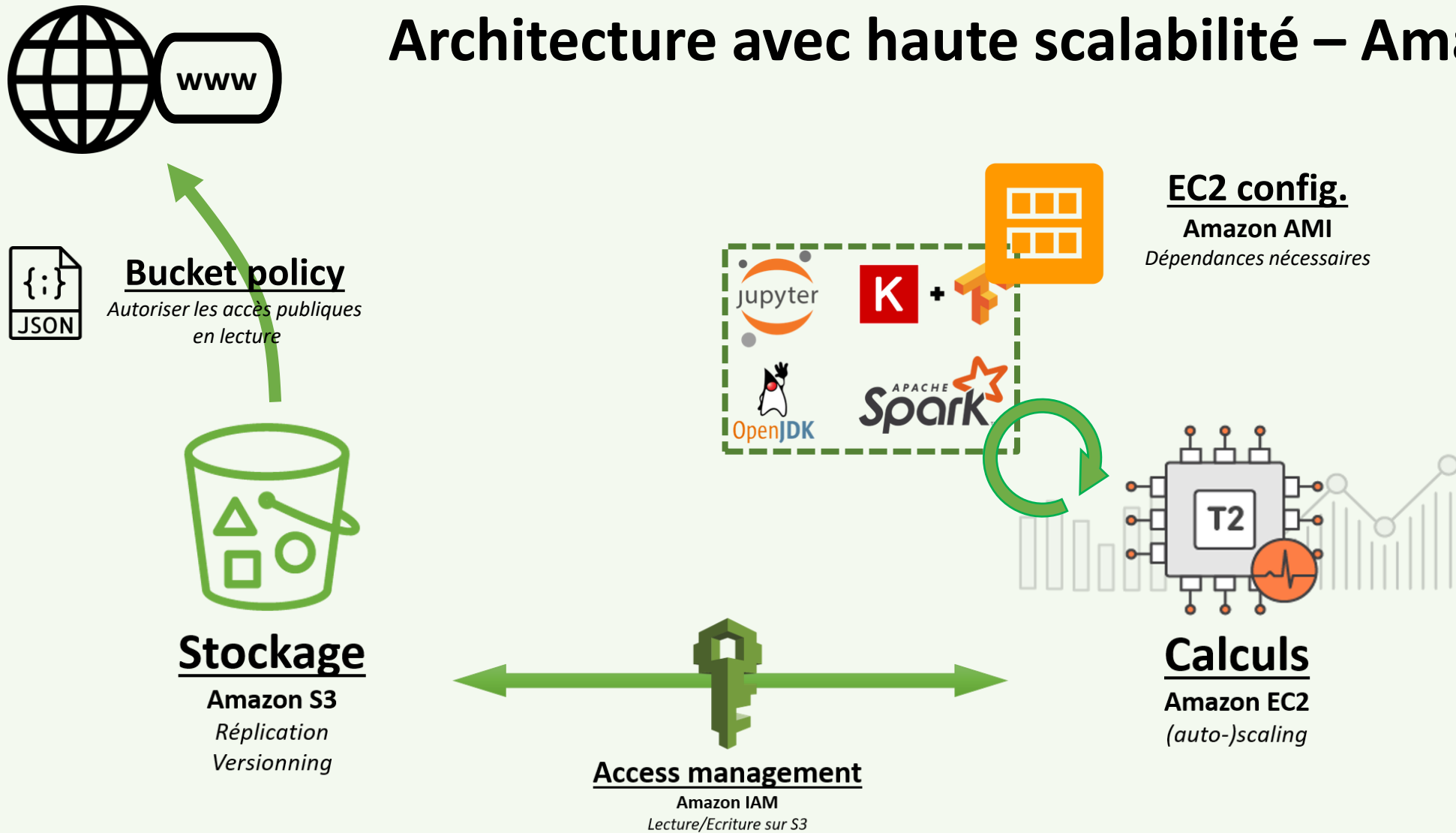
Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS



Déployez un modèle dans le cloud

Architecture avec haute scalabilité – Amazon AWS



Déployez un modèle dans le cloud

Pipeline : PySpark

Déployez un modèle dans le cloud

Pipeline : PySpark

- i. Création session Spark (+ config. pour connexion S3)

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('Fruits').getOrCreate()
spark._jsc.hadoopConfiguration().set("fs.s3a.endpoint", "s3-eu-west-3.amazonaws.com")
spark._jsc.hadoopConfiguration().set("com.amazonaws.services.s3a.enableV4", "true")
```

Déployez un modèle dans le cloud

Pipeline : PySpark

- i. Création session Spark (+ config. pour connexion S3)
- ii. Lecture du jeu de données sur S3 (lazy eval.) et formatage

```
rdd = spark.read.format("binaryFile").option("recursiveFileLookup","true").option("dropInvalid", True).load("s3a://ocr-mxdub/test_data/")
rdd = rdd.select(['content', 'path']).rdd.map(lambda x : (x[0], x[1].split('/')[4] )).toDF(['Image', 'Label'])
rdd = rdd.coalesce(2)
```

Déployez un modèle dans le cloud

Pipeline : PySpark

- i. Création session Spark (+ config. pour connexion S3)
- ii. Lecture du jeu de données sur S3 (lazy eval.) et formatage
- iii. Features extraction : `pandas_udf()` – wrapper traitement
 - i. Chargement model (VGG16) et poids
 - ii. Preprocessing images (redimensionnées, conversion BGR, centrage de chaque canal)
 - iii. Obtention des prédictions (i.e. features extraction)

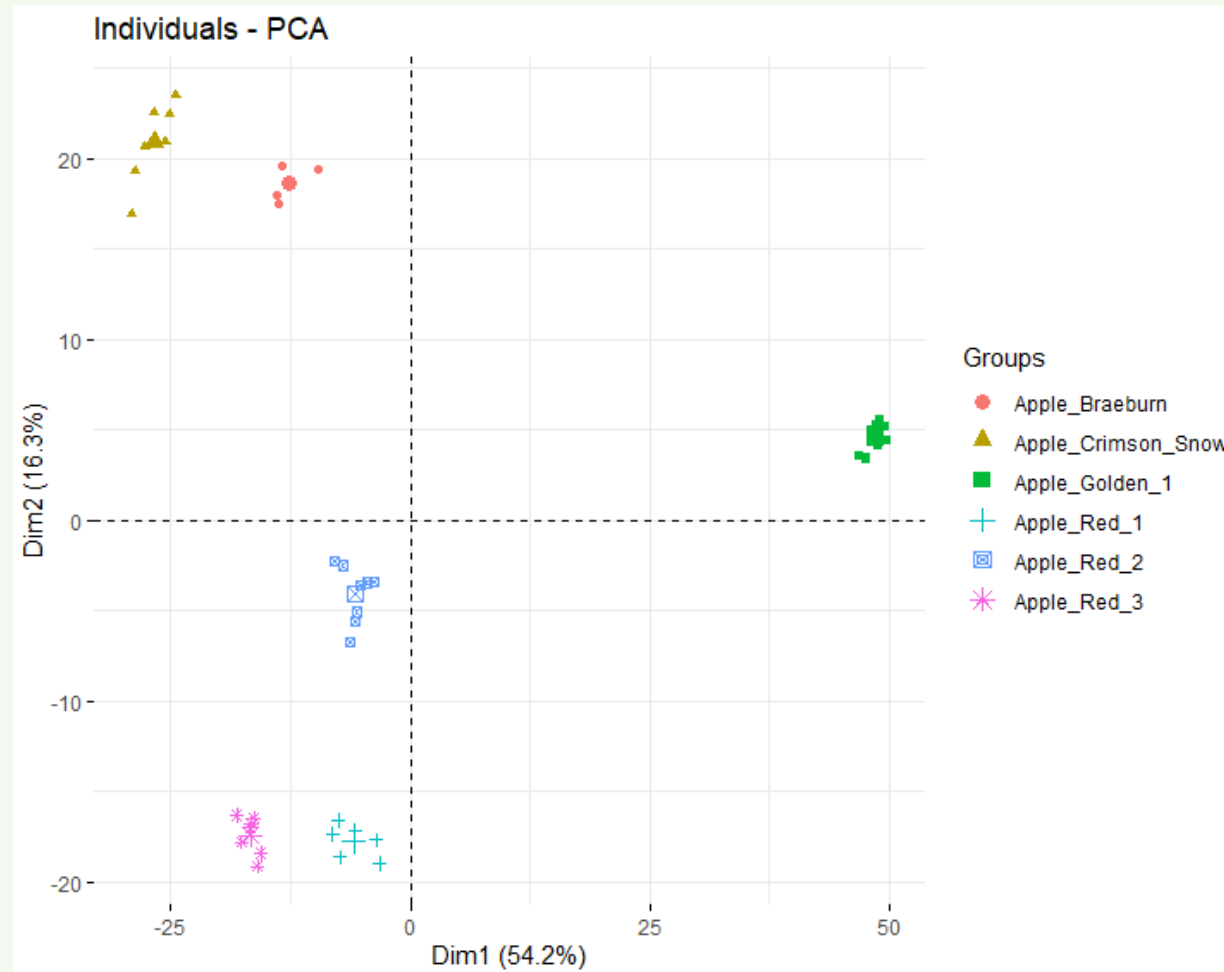
Déployez un modèle dans le cloud

Pipeline : PySpark

- i. Création session Spark (+ config. pour connexion S3)
- ii. Lecture du jeu de données sur S3 (lazy eval.) et formatage
- iii. Features extraction : `pandas_udf()` – wrapper traitement
 - i. Chargement model (VGG16) et poids
 - ii. Preprocessing images (redimensionnées, conversion BGR, centrage de chaque canal)
 - iii. Obtention des prédictions (i.e. features extraction)
- iv. Conversion en matrice et écriture sur S3

```
features_long = features_df.select([col("Label")] + [(col("features")[i]).alias("feat_{}".format(i)) for i in range(feats_size_VGG16)])
features_long.coalesce(1).write.mode("overwrite").option("header", "true").csv("s3a://ocr-mxdub/results/")
# features_long.collect()
```

Déployez un modèle dans le cloud



Déployez un modèle dans le cloud

- i. Déploiement aisé des données dans le cloud S3 (via awscli)**
- ii. Définition des rôles utilisateurs importants
(Identity & Access Management, IAM)**
- iii. Définition des politiques d'accès aux buckets S3**
- iv. Choix du serveur supportant la charge (ici t2.large) et auto-scaling**

Déployez un modèle dans le cloud

- i. Déploiement aisé des données dans le cloud S3 (via awscli)
- ii. Définition des rôles utilisateurs importants
(Identity & Access Management, IAM)
- iii. Définition des politiques d'accès aux buckets S3
- iv. Choix du serveur supportant la charge (ici t2.large) et auto-scaling

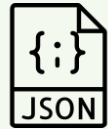
Notes

- Traitement n'est pas distribué sur plusieurs serveurs ici (cf. EMR)
- Permet aisément le passage à une plus large échelle (e.g. changement du type de serveur)

Déployez un modèle dans le cloud

Merci

Déployez un modèle dans le cloud



Bucket policy

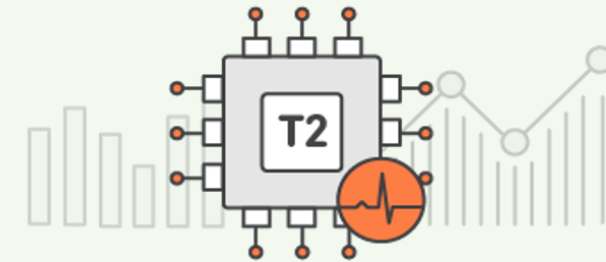
Autoriser les accès publics en lecture



Stockage

Amazon S3
Réplication
Versionning

Architecture avec haute scalabilité – Amazon AWS



Calculs

Amazon EC2
(auto-)scaling



Access management

Amazon IAM
Lecture/Ecriture sur S3