Soutenance Projet 7 : "Déployer un modèle dans le cloud "

Fruits!

OPENCLASSROOMS

Sommaire

I. Problématique

- Data scientist pour la société Agritech 'Fruits' qui cherche à proposer des solutions innovantes pour la récolte des fruits.
- La société souhaite dans un premier temps se faire connaître en mettant à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.
- Cette application permettrait de sensibiliser le grand public à la biodiversité des fruits.
- Le développement de l'application mobile permettra de construire une première version de l'architecture Big Data nécessaire.

- **Notre mission :**

- *Développer dans un environnement Big Data une première chaîne de traitement des données qui comprendra un preprocessing et une étape de réduction de dimension.*
- *Garder en tête que l'important est de mettre en place les premières briques de traitement qui serviront lorsqu'il faudra passer à l'échelle en termes de volume de données !*

I. Problématique

Démarche de travail :

- Analyse rapide des données disponibles.
- Récupérer un sample de ces données pour faire des essais.
- Mettre en place une chaîne de traitement en local sur machine virtuelle pour effectuer des premiers tests.
- Mettre en place une première architecture Big Data via les services AWS.
- Valider notre chaîne de traitement dans cette architecture.
- Privilégier un coût faible pour démontrer la faisabilité avant un passage à l'échelle (première brique de l'architecture).
- Envisager le passage à l'échelle pour des données plus conséquentes.

II. Les données

- Données disponibles via l'adresse :
  - https://www.kaggle.com/datasets/moltean/fruits
  - Dossier Training :
    - 131 Fruits
    - 500 images par fruits en moyenne
    - Pour des raisons de coût mémoire on utilise un petit échantillon de 3 fruits avec 2-4 images par fruits (9 images).

II. Les données

Les images sélectionnées :

III. Architecture Big Data

- Rappels:
  - On parle de Big Data lorsque la quantité de données excède la faculté d'une machine à les stocker et les analyser en temps acceptable.
  - L'idée est de distribuer les calculs entre plusieurs machines.
  - Paradigme diviser pour mieux régner / MapReduce
  - Infrastructure logiciel dédié :
    - Data locality, optimiser les déplacements de données
    - Scalability, adapter la puissance au besoin
    - Embracing failure, tolérant aux pannes

III. Architecture Big Data

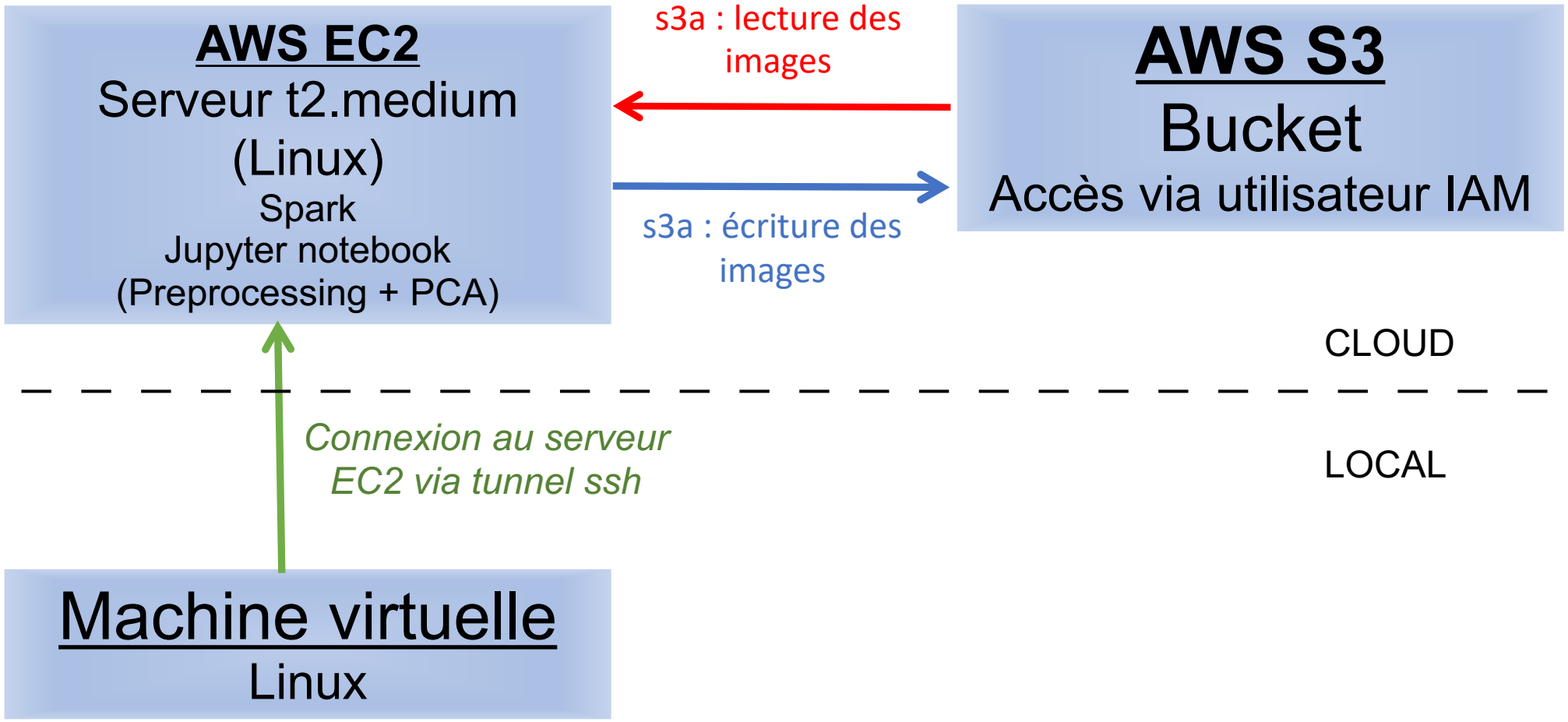
- Architecture hébergée sur Amazon Web Services (AWS) :

- Un serveur EC2 de type t2.medium (RAM 4GB)
  - Communication via ssh
  - Utilisation de Spark (via pyspark)
  - Utilisation Jupyter

- Un espace de stockage S3
  - Accès privés
  - Utilisation d'un profile utilisateur IAM pour les accès en lecture et écriture

III. Architecture Big Data

Schéma de l'architecture :



III. Architecture Big Data

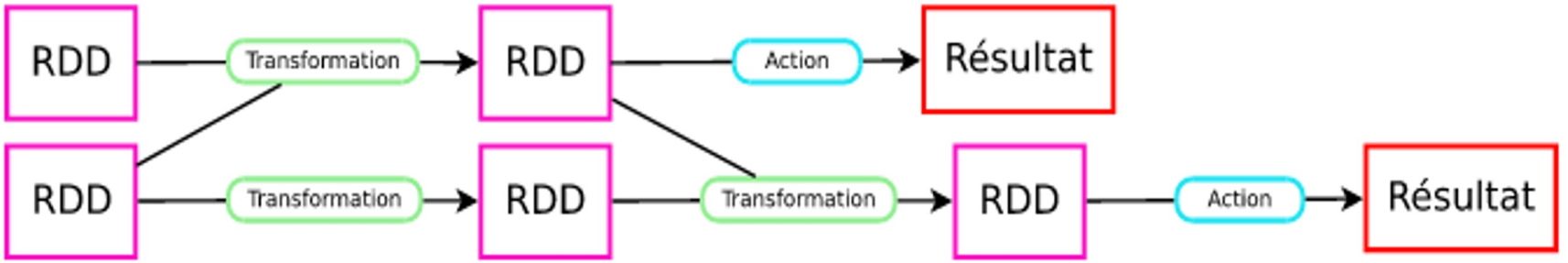
- Serveur EC2
  - Machine virtuelle sur le cloud
  - Configurations sous AWS :
    - Linux (Ubuntu 22.04)
    - Ouverture d'accès :
      - Port 22 pour le tunnel ssh
      - Port 8888 pour jupyter notebook (TCP)
      - Port 4040 pour SparkUI (TCP)
    - Installations via ssh :
      - Java 8/11
      - Spark, spark-3.3.0-bin-hadoop3
      - Jupyter (py4j)

III. Architecture Big Data

- Stockage S3 :
  - Espace de stockage sur le cloud
  - Création d'un bucket «openclassroom-p8-ceyhun»
  - Contenu privé
  - Création d'un profile utilisateur IAM (ID + clé secrète) avec droits administrateur pour y accéder
  - Ajout du dossier sample des images via commande bash

III. Architecture Big Data

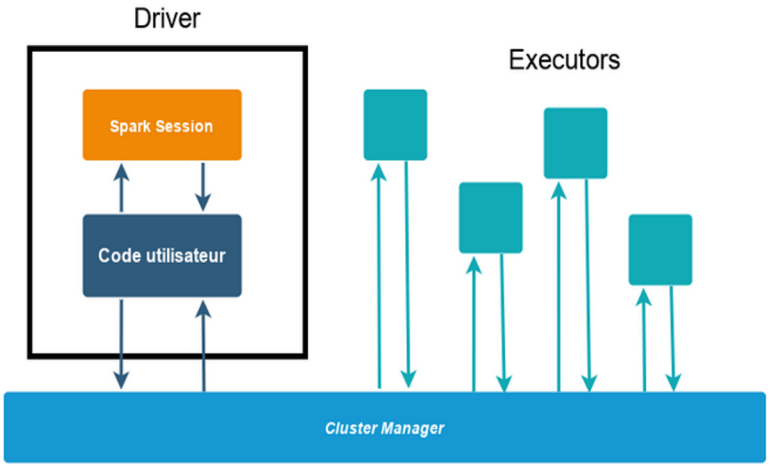
- Calcul distribué Spark :
  - Utilise la RAM pour gagner en temps d'exécution
  - Elargit le cadre map/reduce en proposant à l'utilisateur des opérations supplémentaires pouvant être réalisées de manière distribuée
  - Effectue des transformations et actions sur des RDD (Resilient Distributed Dataset)

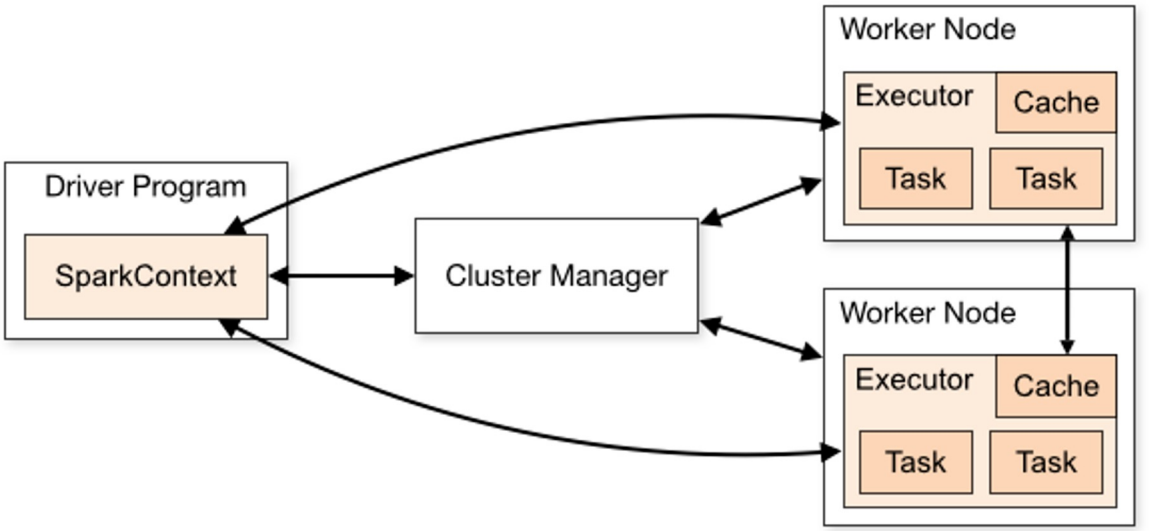


  - Permet également de passer par des données structurés de type Dataframe

III. Architecture Big Data

- Calcul distribué Spark :
  - Utilisation de PySpark (script python) en mode standalone





- Worker = machine physique
- Executor = application qui tourne sur la machine physique
- Driver Program → Distribue les tâches sur les executors
- Cluster Manager → Instancie les différents workers

IV. Chaîne de traitement

- Les étapes du traitement :
  - Lecture/chargement des images présentes sur AWS S3
  - Preprocessing (resize) pour Spark
  - Prédire avec RESNET50 model
  - Réduction de dimension des features par PCA

- Script Python via jupyter notebook (utilisation du port 8888 du serveur EC2)

IV. Chaîne de traitement

1) Chargement des images du bucket S3 dans un dataframe Spark

In [4]:
```
# Define S3 path
path_s3 = 's3a://openclassroom-p8-ceyhun/fruits_sample'
path_s3_root = 's3a://openclassroom-p8-ceyhun'
```

In [5]:
```
df_image = spark.read.format('image').load(f"{path_s3}/*")
df_image.show(5)
```

```
22/10/09 15:37:18 WARN MetricsConfig: Cannot locate configuration: tried hadoop-metrics2-s3a-
file-system.properties,hadoop-metrics2.properties
```

```
+--------------------+
|               image|
+--------------------+
|{s3a://openclassr...|
|{s3a://openclassr...|
|{s3a://openclassr...|
|{s3a://openclassr...|
|{s3a://openclassr...|
+--------------------+
```

Import data

In [9]:
```
df_image.printSchema()
```

```
root
 |-- image: struct (nullable = true)
 |    |-- origin: string (nullable = true)
 |    |-- height: integer (nullable = true)
 |    |-- width: integer (nullable = true)
 |    |-- nChannels: integer (nullable = true)
 |    |-- mode: integer (nullable = true)
 |    |-- data: binary (nullable = true)
```

In [10]:
```
print(f"count of image = {df_image.count()} length of columns = {len(df_image.columns)}")
```

```
count of image = 13 length of columns = 1
```


IV. Chaîne de traitement

2) Récupération des labels et remise en forme du dataframe Spark

Converting all images in Spark

In [11]:
```
image_row = 7
```

In [12]:
```
schema = StructType(df_image.select("image.*").schema.fields + [
    StructField("data_as_resized_array", ArrayType(IntegerType()), True),
    StructField("data_as_array", ArrayType(IntegerType()), True)
])

def convert_bgr_array_to_rgb_array(img_array):
    B, G, R = img_array.T
    return np.array((R, G, B)).T

def resize_img(img_data, resize=True):
    mode = 'RGBA' if (img_data.nChannels == 4) else 'RGB'
    img = Image.frombytes(mode=mode, data=img_data.data, size=[img_data.width, img_data.height]
    img = img.convert('RGB') if (mode == 'RGBA') else img
    img = img.resize([224, 224], resample=Image.Resampling.BICUBIC) if (resize) else img
    arr = convert_bgr_array_to_rgb_array(np.asarray(img))
    arr = arr.reshape([224*224*3]) if (resize) else arr.reshape([img_data.width*img_data.height

    return arr

def resize_image_udf(dataframe_batch_iterator: Iterator[pd.DataFrame]) -> Iterator[pd.DataFrame
    for dataframe_batch in dataframe_batch_iterator:
        dataframe_batch["data_as_resized_array"] = dataframe_batch.apply(resize_img, args=(True
        dataframe_batch["data_as_array"] = dataframe_batch.apply(resize_img, args=(False,), axi
        yield dataframe_batch

resized_df = df_image.select("image.*").mapInPandas(resize_image_udf, schema)
```

We can check if the data contains a converted and a resized image

IV. Chaîne de traitement

3) Utilisation de RESNET50:

Predicting with the ResNet50 model

In [17]:
```
def normalize_array(arr):
    return tf.keras.applications.resnet50.preprocess_input(arr.reshape([224,224,3]))

@pandas_udf(ArrayType(FloatType()))
def predict_batch_udf(iterator: Iterator[pd.Series]) -> Iterator[pd.Series]:
    model = ResNet50(weights=None)
    model.set_weights(bc_model_weights.value)
    for input_array in iterator:
        normalized_input = np.stack(input_array.map(normalize_array))
        preds = model.predict(normalized_input)
        yield pd.Series(list(preds))

predicted_df = resized_df.withColumn("predictions", predict_batch_udf("data_as_resized_array"))
```

In [19]:
```
tf.keras.applications.resnet50.decode_predictions(
    np.array(prediction_row.predictions).reshape(1,1000), top=5
)
```

Out[19]:
```
[[('n07742313', 'Granny_Smith', 0.9980723857879639),
  ('n07749582', 'lemon', 0.00036845251452177763),
  ('n04409515', 'tennis_ball', 0.00030365970451384783),
  ('n02782093', 'balloon', 0.00018677501066122204),
  ('n07753592', 'banana', 5.9041525673819706e-05)]]
```


IV. Chaîne de traitement

4) Utilisation de la PCA de « pyspark.ml.feature » :

```
def pca_transformation(df, n_components=2, col_image='image'):

    """
    Applique un algorithme de PCA sur l'ensemble des images pour réduire la dimension de chaque
    du jeu de données.

    Paramètres:
    df(pyspark dataFrame): contient une colonne avec les données images
    n_components(int): nombre de dimensions à conserver
    col_image(string): nom de la colonne où récupérer les données images
    """

    # Initilisation du temps de calcul
    start_time = time.time()

    # Les données images sont converties au format vecteur dense
    #ImageSchema.imageFields
    img2vec = F.udf(lambda x : Vectors.dense(x), VectorUDT())


    df = df.withColumn("data_as_vector", img2vec("data_as_resized_array"))

    standardizer = StandardScaler(withMean=True, withStd=True, inputCol="data_as_vector", outpu

    model_std = standardizer.fit(df)
    df = model_std.transform(df)
    # Entrainement de l'algorithme
    pca = PCA(k=n_components, inputCol='data_as_vector', outputCol='pcaFeatures')
    model_pca = pca.fit(df)

    # Transformation des images sur les k premières composantes
    df = model_pca.transform(df)

    df = df.filter(df.pcaFeatures.isNotNull())

    print(model_pca.explainedVariance)


    # Affiche le temps de calcul
    print("Temps d'execution {:.2f} secondes".format(time.time() - start_time))



    return df
```


IV. Chaîne de traitement

5) Enregistrement des résultats dans le bucket S3

Save Files as Parquet

In [32]:
```
result_df = df.select('img_path_s3','pcaFeatures', 'label')
result_df.show()
```

```
22/10/09 15:42:39 WARN DAGScheduler: Broadcasting large task binary with size 2.4 MiB
[Stage 34:>                                                         (0 + 1) / 1]
```

img_path_s3	pcaFeatures	label
openclassroom-p8-...	[-36292.775883125...	cabbage_white_1
openclassroom-p8-...	[-33814.861621121...	cabbage_white_1
openclassroom-p8-...	[-34564.791956497...	cabbage_white_1
openclassroom-p8-...	[-11680.891318570...	cucumber_1
openclassroom-p8-...	[-10799.869759969...	cucumber_1
openclassroom-p8-...	[-38651.840423738...	apple_granny_smith_1
openclassroom-p8-...	[-40391.651082526...	apple_granny_smith_1
openclassroom-p8-...	[-39437.741593447...	apple_granny_smith_1
openclassroom-p8-...	[-36272.882319034...	apple_granny_smith_1
openclassroom-p8-...	[-12509.175459769...	apple_rotten_1
openclassroom-p8-...	[-8275.7465326340...	apple_rotten_1
openclassroom-p8-...	[-11691.208398820...	apple_rotten_1
openclassroom-p8-...	[-9649.1109840233...	apple_rotten_1

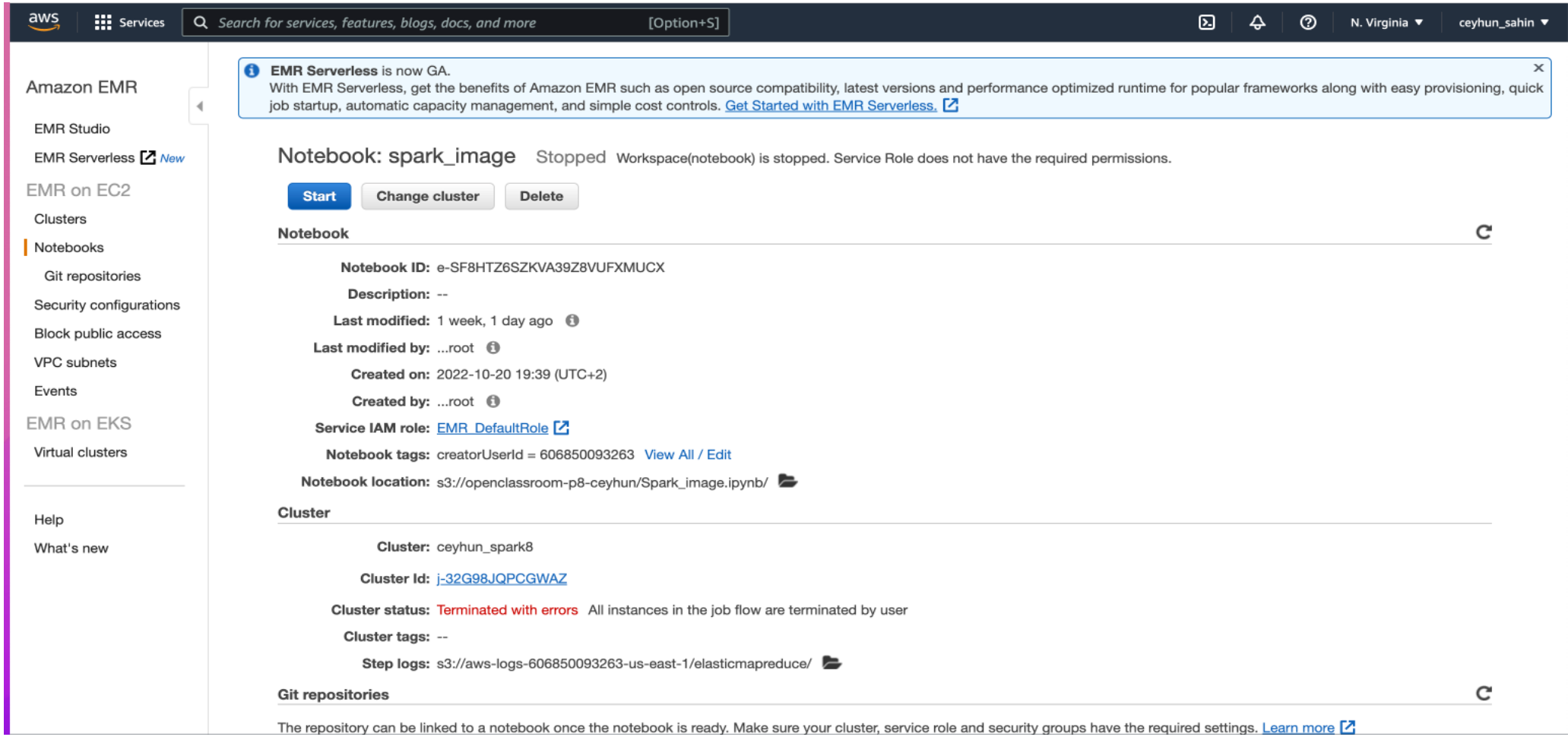
V. Conclusion

- Mise en place d'une première architecture Big Data.
- Développement d'une chaîne de traitement via pyspark fonctionnelle sur cette architecture.
- <u>Recommendations pour un passage à l'échelle :</u>
- Utiliser les outils de « cost managment » de AWS pour gérer les coût de l'infrastructure.
- Peut être voir pour améliorer la chaîne de traitement et passer par un preprocessing plus poussé (transfer learning) ?

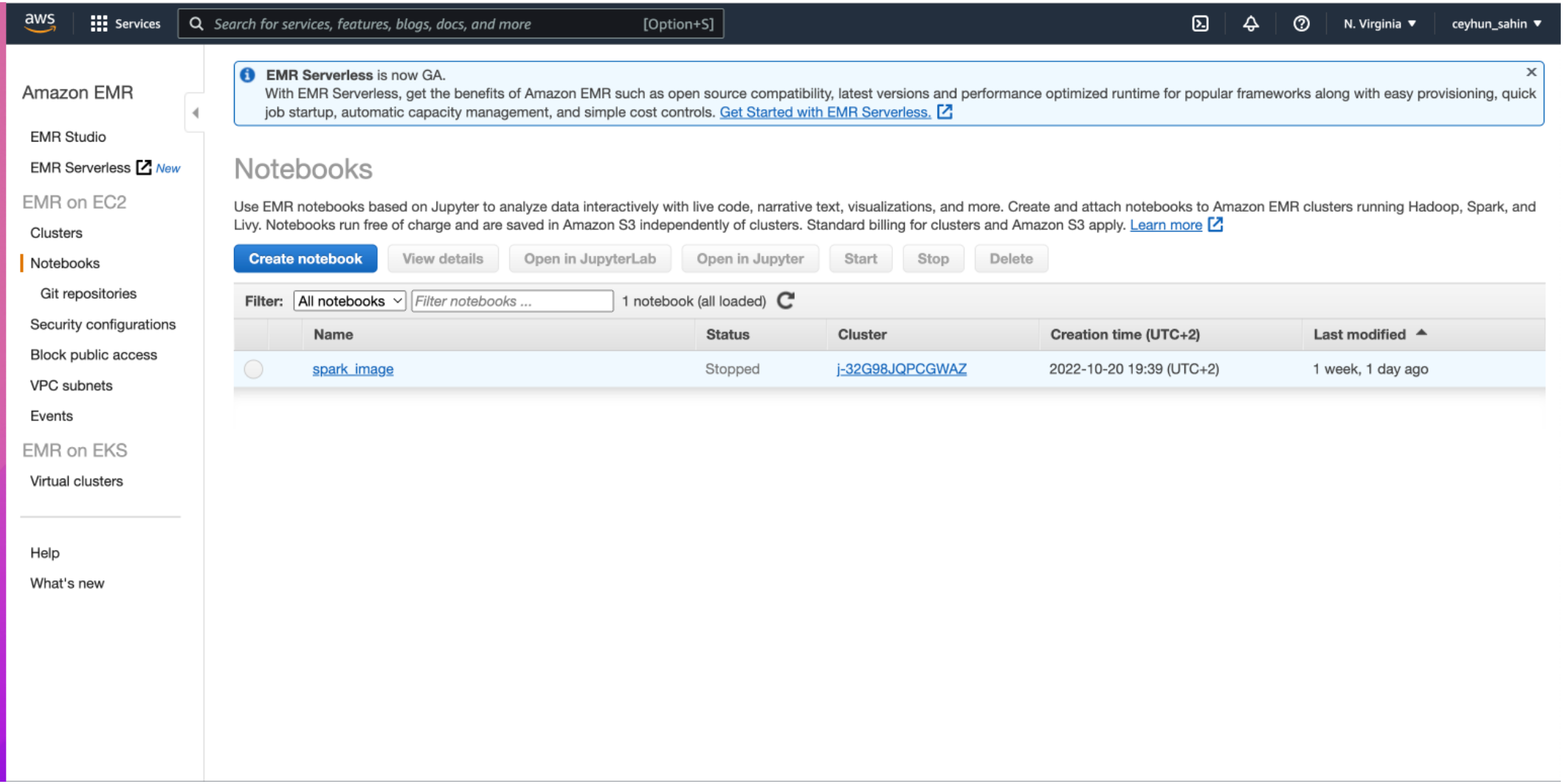
Merci pour votre attention !

Questions ?

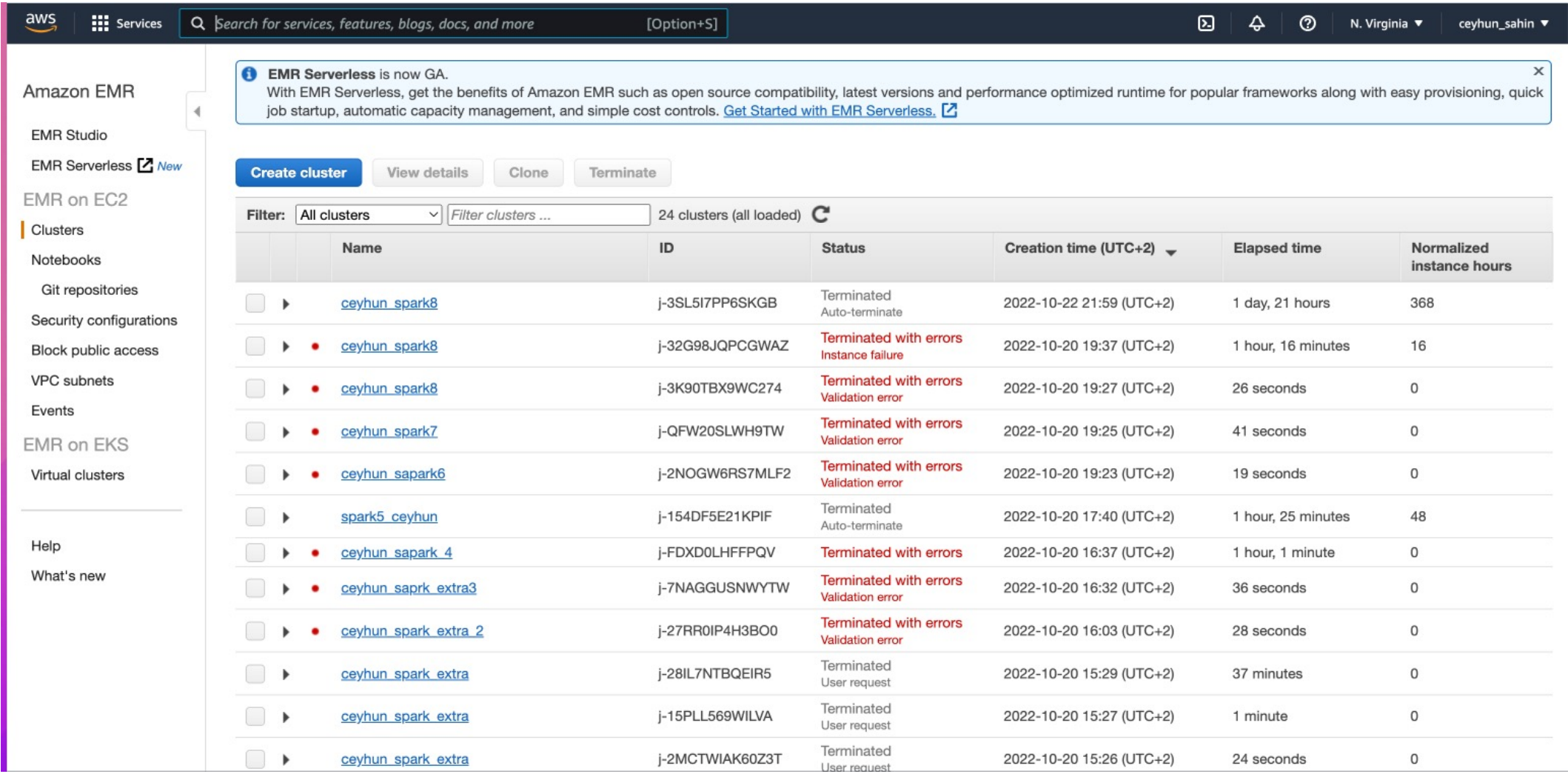
EMR


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
ceyhun_sahin
Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Notebook: spark_image Stopped Workspace(notebook) is stopped. Service Role does not have the required permissions.
Start
Change cluster
Delete
Notebook
Notebook ID: e-SF8HTZ6SZKVA39Z8VUFXMUCX
Description: --
Last modified: 1 week, 1 day ago
Last modified by: ...root
Created on: 2022-10-20 19:39 (UTC+2)
Created by: ...root
Service IAM role: EMR_DefaultRole
Notebook tags: creatorUserId = 606850093263 View All / Edit
Notebook location: s3://openclassroom-p8-ceyhun/Spark_image.ipynb/
Cluster
Cluster: ceyhun_spark8
Cluster Id: j-32G98JQPCGWAZ
Cluster status: Terminated with errors All instances in the job flow are terminated by user
Cluster tags: --
Step logs: s3://aws-logs-606850093263-us-east-1/elasticmapreduce/
Git repositories
The repository can be linked to a notebook once the notebook is ready. Make sure your cluster, service role and security groups have the required settings. Learn more

EMR

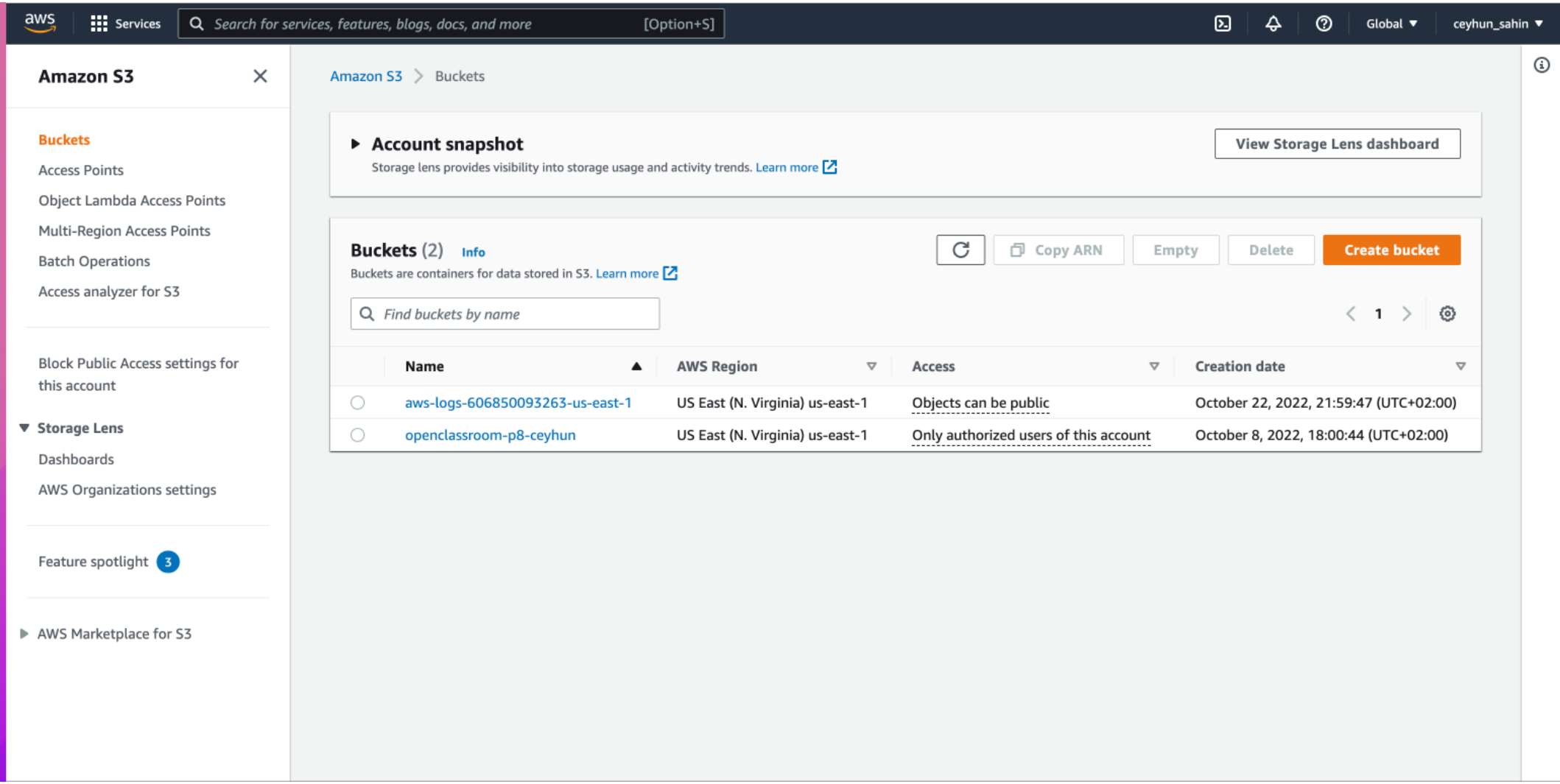

aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
ceyhun_sahin
Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Notebooks
Use EMR notebooks based on Jupyter to analyze data interactively with live code, narrative text, visualizations, and more. Create and attach notebooks to Amazon EMR clusters running Hadoop, Spark, and Livy. Notebooks run free of charge and are saved in Amazon S3 independently of clusters. Standard billing for clusters and Amazon S3 apply. Learn more
Create notebook
View details
Open in JupyterLab
Open in Jupyter
Start
Stop
Delete
Filter:
All notebooks
Filter notebooks ...
1 notebook (all loaded)
Name
Status
Cluster
Creation time (UTC+2)
Last modified
spark_image
Stopped
j-32G98JQPCGWAZ
2022-10-20 19:39 (UTC+2)
1 week, 1 day ago

EMR

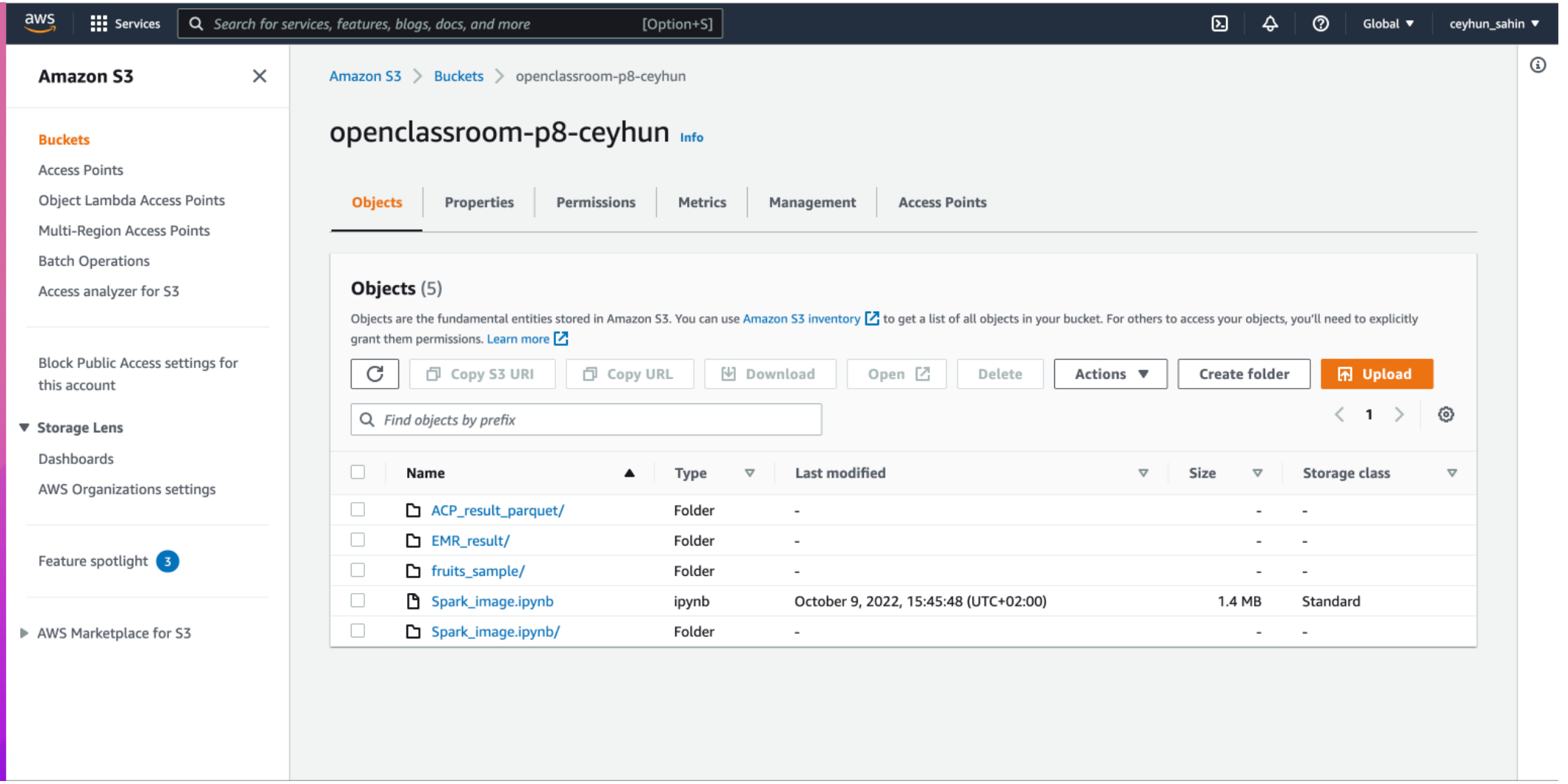


Name	ID	Status	Creation time (UTC+2)	Elapsed time	Normalized instance hours
ceyhun_spark8	j-3SL5I7PP6SKGB	Terminated Auto-terminate	2022-10-22 21:59 (UTC+2)	1 day, 21 hours	368
ceyhun_spark8	j-32G98JQPCGWAZ	Terminated with errors Instance failure	2022-10-20 19:37 (UTC+2)	1 hour, 16 minutes	16
ceyhun_spark8	j-3K90TBX9WC274	Terminated with errors Validation error	2022-10-20 19:27 (UTC+2)	26 seconds	0
ceyhun_spark7	j-QFW20SLWH9TW	Terminated with errors Validation error	2022-10-20 19:25 (UTC+2)	41 seconds	0
ceyhun_sapark6	j-2NOGW6RS7MLF2	Terminated with errors Validation error	2022-10-20 19:23 (UTC+2)	19 seconds	0
spark5_ceyhun	j-154DF5E21KPIF	Terminated Auto-terminate	2022-10-20 17:40 (UTC+2)	1 hour, 25 minutes	48
ceyhun_sapark_4	j-FDXD0LHFFPQV	Terminated with errors	2022-10-20 16:37 (UTC+2)	1 hour, 1 minute	0
ceyhun_saprk_extra3	j-7NAGGUSNWYTW	Terminated with errors Validation error	2022-10-20 16:32 (UTC+2)	36 seconds	0
ceyhun_spark_extra_2	j-27RR0IP4H3BO0	Terminated with errors Validation error	2022-10-20 16:03 (UTC+2)	28 seconds	0
ceyhun_spark_extra	j-28IL7NTBQEIR5	Terminated User request	2022-10-20 15:29 (UTC+2)	37 minutes	0
ceyhun_spark_extra	j-15PLL569WILVA	Terminated User request	2022-10-20 15:27 (UTC+2)	1 minute	0
ceyhun_spark_extra	j-2MCTWIAK60Z3T	Terminated User request	2022-10-20 15:26 (UTC+2)	24 seconds	0

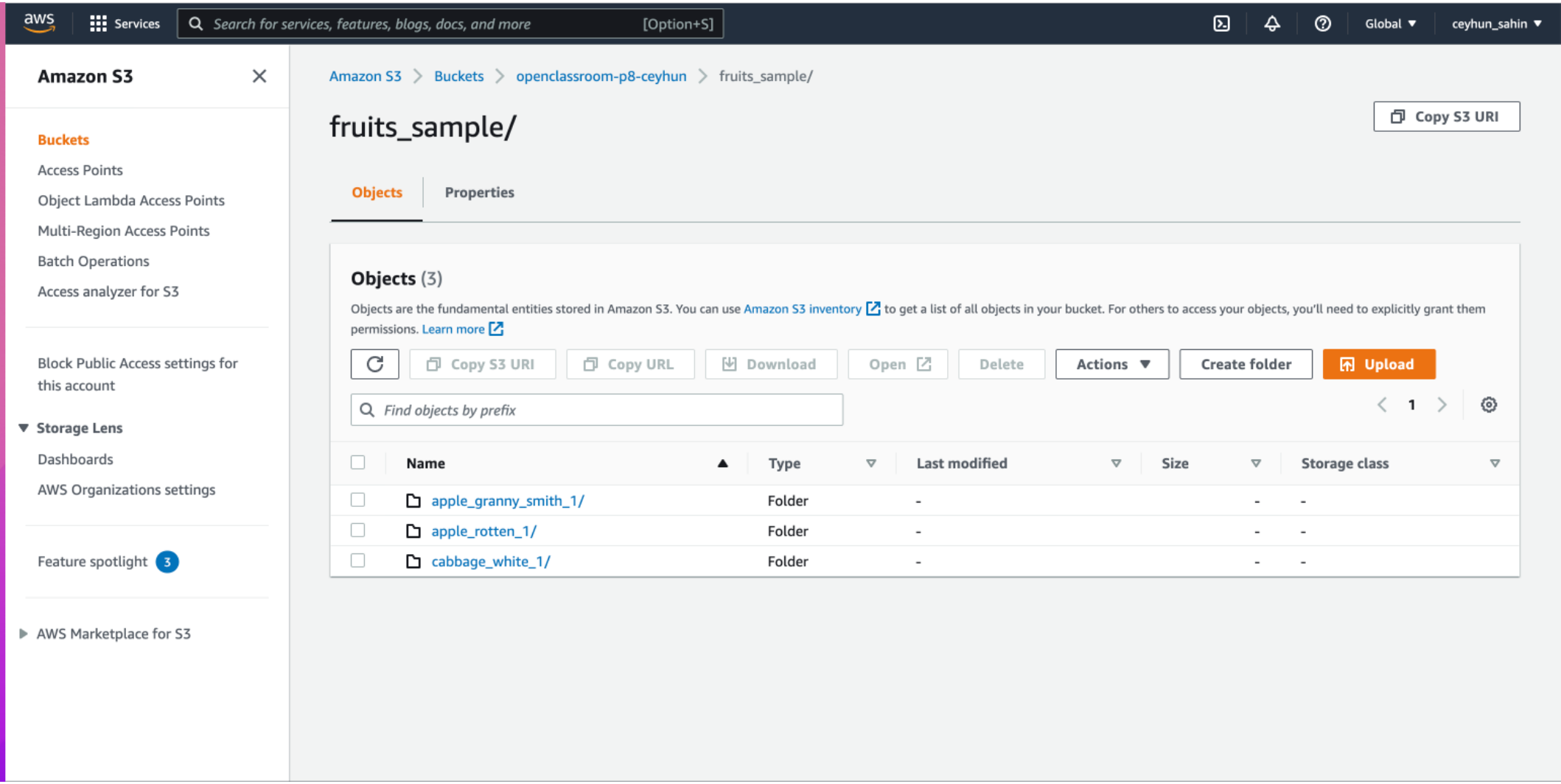
S3 Bucket


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
Global
ceyhun_sahin
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (2) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
Access
Creation date
aws-logs-606850093263-us-east-1
US East (N. Virginia) us-east-1
Objects can be public
October 22, 2022, 21:59:47 (UTC+02:00)
openclassroom-p8-ceyhun
US East (N. Virginia) us-east-1
Only authorized users of this account
October 8, 2022, 18:00:44 (UTC+02:00)

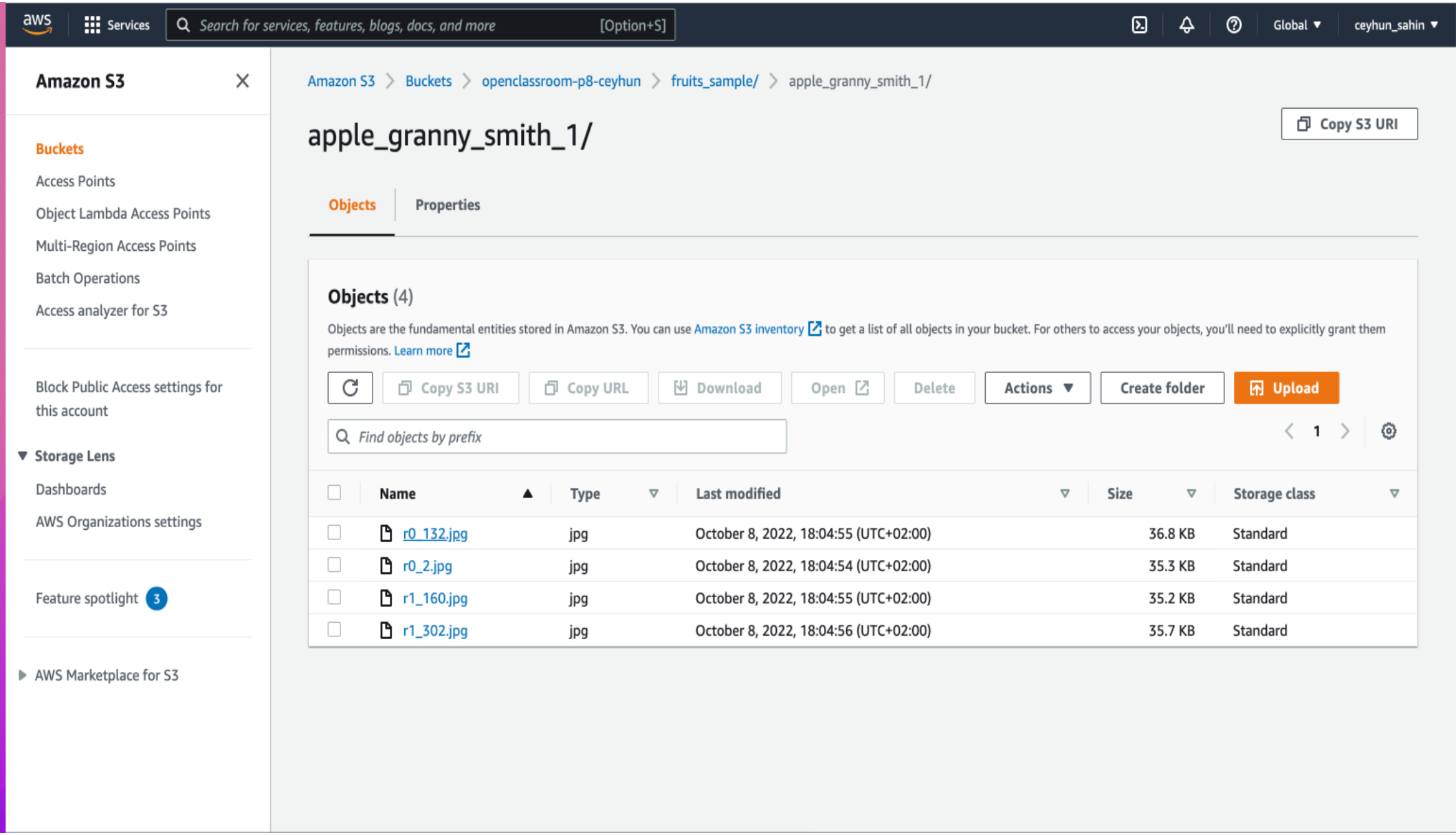
S3 Bucket


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
Global
ceyhun_sahin
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets > openclassroom-p8-ceyhun
openclassroom-p8-ceyhun Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (5)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
ACP_result_parquet/ Folder - - -
EMR_result/ Folder - - -
fruits_sample/ Folder - - -
Spark_image.ipynb ipynb October 9, 2022, 15:45:48 (UTC+02:00) 1.4 MB Standard
Spark_image.ipynb/ Folder - - -

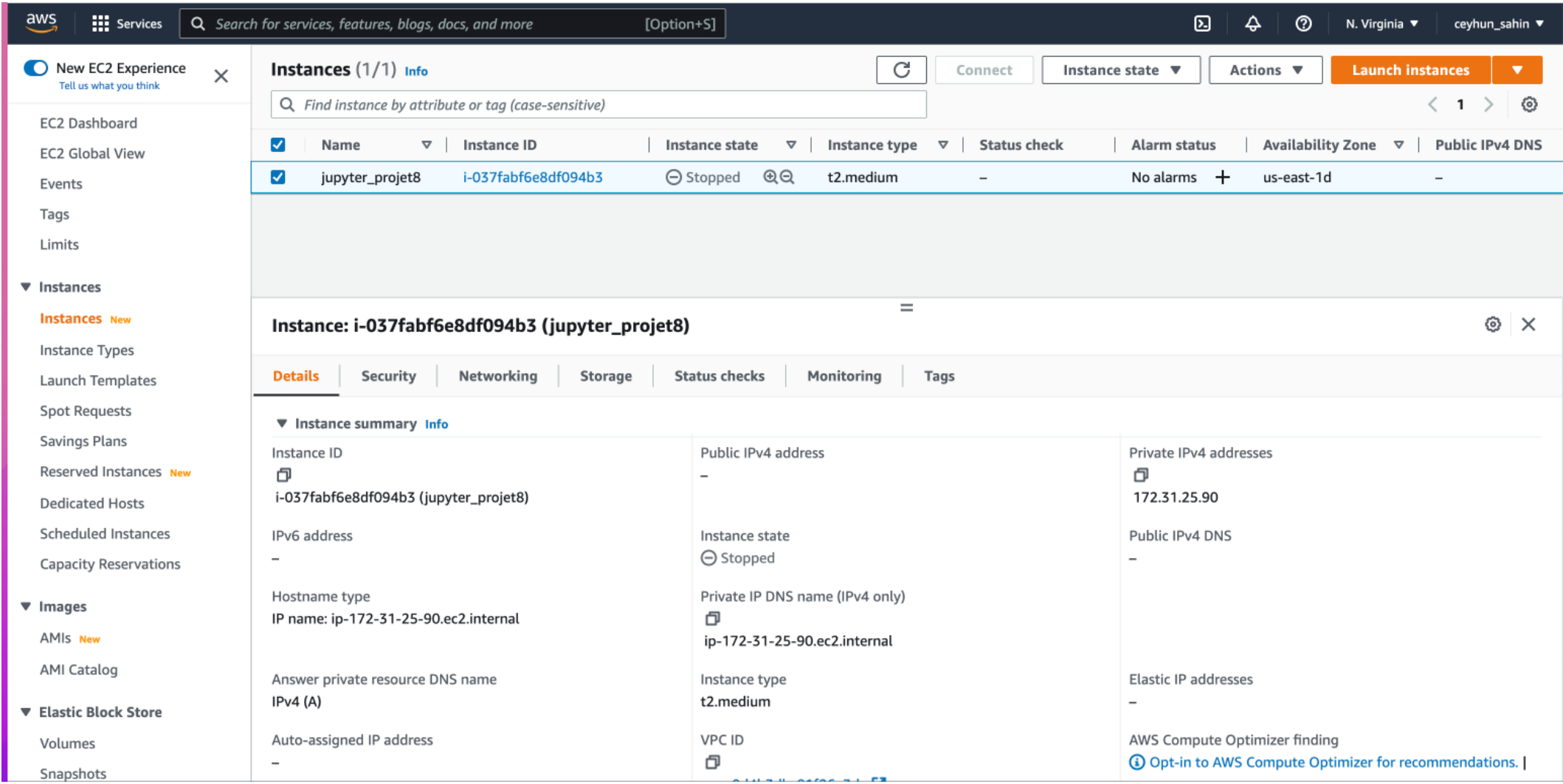
S3 Bucket


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
Global
ceyhun_sahin
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets > openclassroom-p8-ceyhun > fruits_sample/
fruits_sample/
Copy S3 URI
Objects
Properties
Objects (3)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
apple_granny_smith_1/ Folder - - -
apple_rotten_1/ Folder - - -
cabbage_white_1/ Folder - - -

S3 Bucket


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
Global
ceyhun_sahin
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets > openclassroom-p8-ceyhun > fruits_sample/ > apple_granny_smith_1/
apple_granny_smith_1/
Copy S3 URI
Objects
Properties
Objects (4)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
r0_132.jpg jpg October 8, 2022, 18:04:55 (UTC+02:00) 36.8 KB Standard
r0_2.jpg jpg October 8, 2022, 18:04:54 (UTC+02:00) 35.3 KB Standard
r1_160.jpg jpg October 8, 2022, 18:04:55 (UTC+02:00) 35.2 KB Standard
r1_302.jpg jpg October 8, 2022, 18:04:56 (UTC+02:00) 35.7 KB Standard

EC2


aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
ceyhun_sahin
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
jupyter_projet8
i-037fabf6e8df094b3
Stopped
t2.medium
–
No alarms
us-east-1d
–
Instance: i-037fabf6e8df094b3 (jupyter_projet8)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-037fabf6e8df094b3 (jupyter_projet8)
IPv6 address
–
Hostname type
IP name: ip-172-31-25-90.ec2.internal
Answer private resource DNS name
IPv4 (A)
Auto-assigned IP address
–
Public IPv4 address
–
Instance state
Stopped
Private IP DNS name (IPv4 only)
ip-172-31-25-90.ec2.internal
Instance type
t2.medium
VPC ID
Private IPv4 addresses
172.31.25.90
Public IPv4 DNS
–
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.