eosc | FAIR-EASE
Building Interoperable Earth Science & Environmental Services

Evaluation de l'utilisation de S3 pour manipuler des données dans le domaine de l'environnement

Antoine Mahul - Mesocentre UCA

David Sarramia - UCA

Damian Smyth - Marine Institute - Ireland

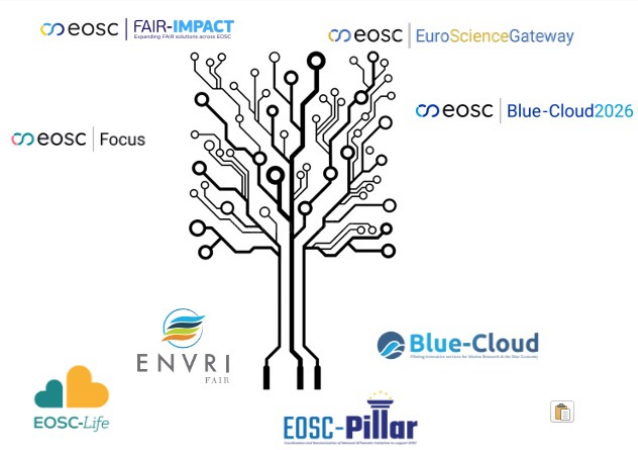
FAIR-EASE has received funding from the European Union's Horizon Europe Framework Programme (HORIZON) - under grant agreement No. 101058785.

L’ambition FAIR-EASE (2022-2025)

Le constat :

- Domaines du système terrestre : interconnectés
- Architecture numérique actuelle :
  - référentiels de données distribués,
  - dépendants du domaine.

→ Difficultés pour l'utilisation intégrée de toutes les données environnementales



FAIR-EASE = passerelles pour les sciences de la terre et de l'environnement pour l’accès aux données par

- La personnalisation et l'exploitation des services distribués et intégrés
- En coopération avec
  - Les communautés d'utilisateurs,
  - L'European Open Science Cloud (EOSC),
  - Les infrastructures de recherche.

WP5 Research communities engagement

- **Use Case 1: Earth and environment dynamics**
  - **Pilot 1: Coastal Water Dynamics**
  - **Pilot 2: Earth Critical Zones**
  - **Pilot 3: Volcano Space Observatory**
- **Use Case 2 Environmental Bio-Geochemical Assets**
  - **Pilot 4: Ocean Bio-Geochemical Observation**
- **Use Case 3 Biodiversity Observation**
  - **Pilot 5: Marine Omics Observations**

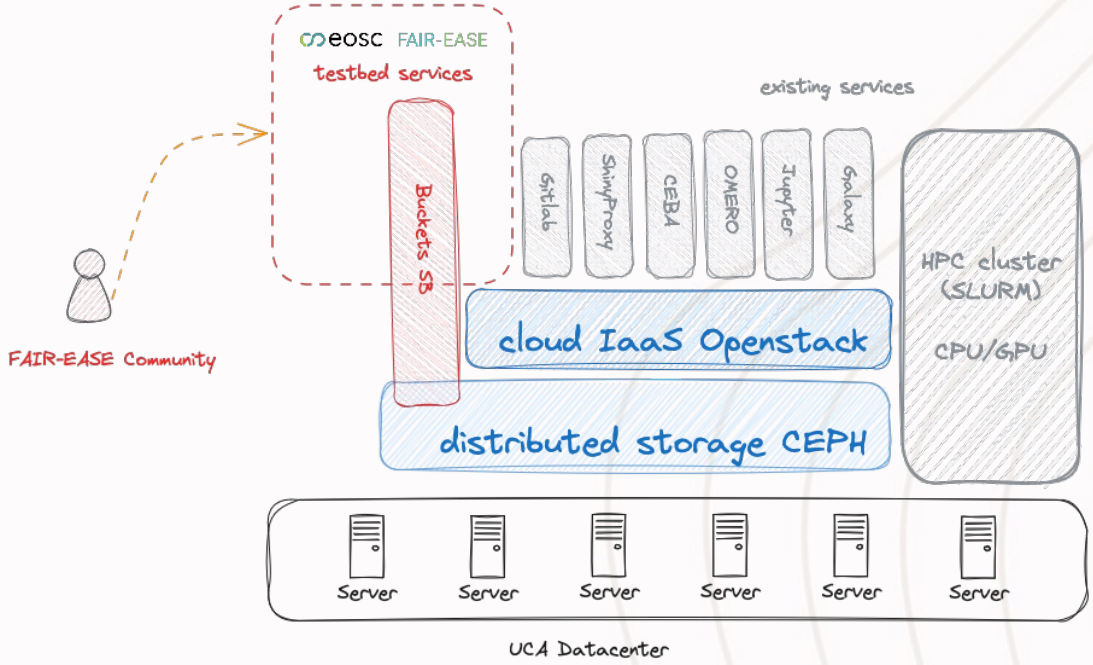
Source: German National Academy of Sciences Leopoldina (2022):
Report on Tomorrow’s Science. Earth System Science – Discovery, Diagnosis, and Solutions in Times of Global Change. Halle (Saale).

Questions traitées

Utilisateur

- Comprendre et utiliser S3 pour accéder et stocker des fichiers de données ?
- Créer/Manipuler des fichiers optimisés cloud ?

Tester les interactions entre Jupyter, les logiciels tiers et S3



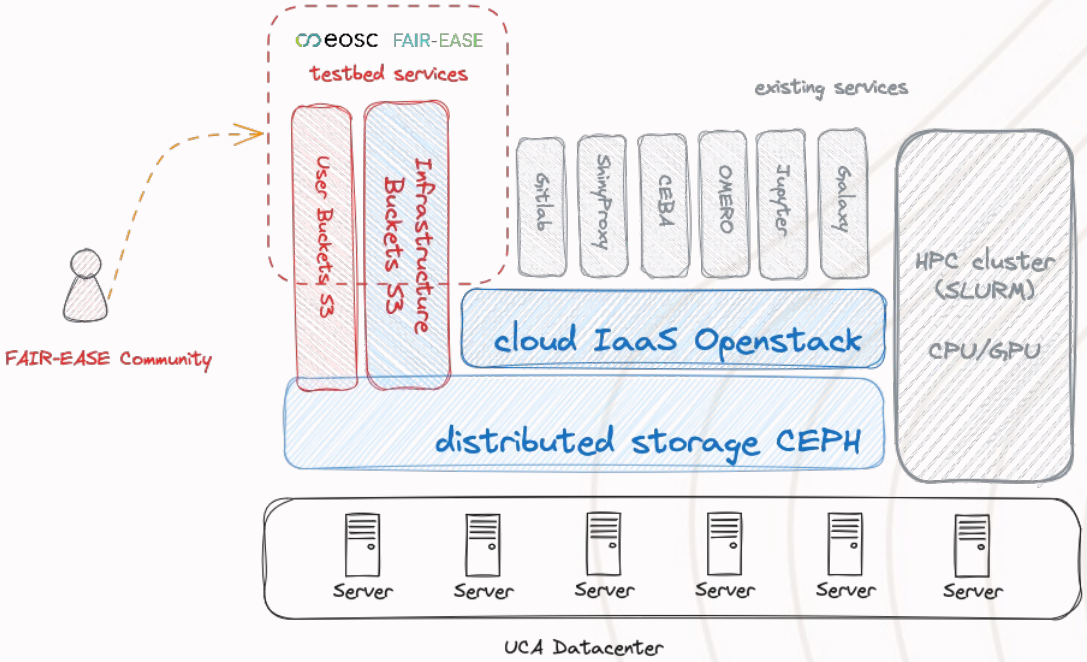
Pas de tests de performances : compréhension et usage côté utilisateur

Questions traitées

Infrastructure

- Mise à jour automatiques du catalogue de données stocké sur le S3 à l'ajout de fichiers ?
- S3 peut-il gérer les métadonnées utilisateur/FAIR-EASE ?

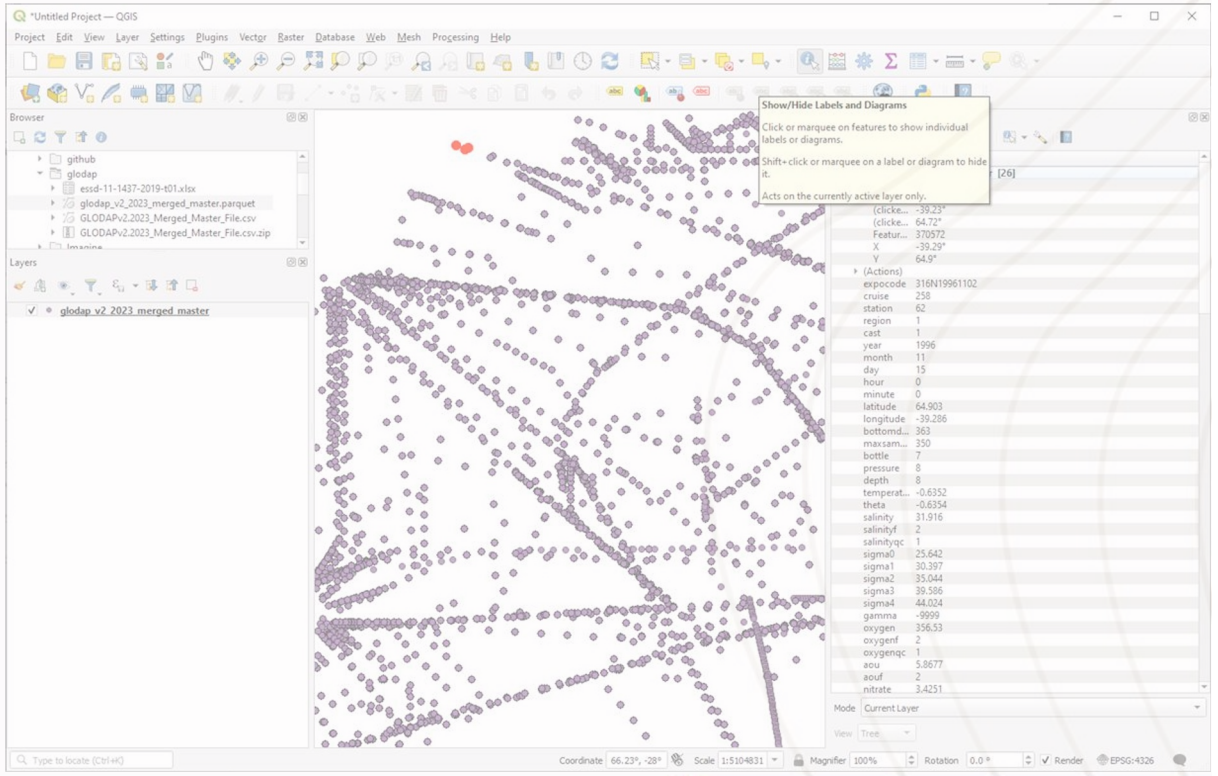
Tester les interactions avec un catalogue STAC statique



Pas de tests de performances : compréhension et usage côté utilisateur

eosc | FAIR-EASE

Utilisateur



eosc | FAIR-EASE

Côté utilisateur

Tests des interactions entre Jupyter Notebook, les logiciels tiers et S3

Question « naïve » des utilisateurs

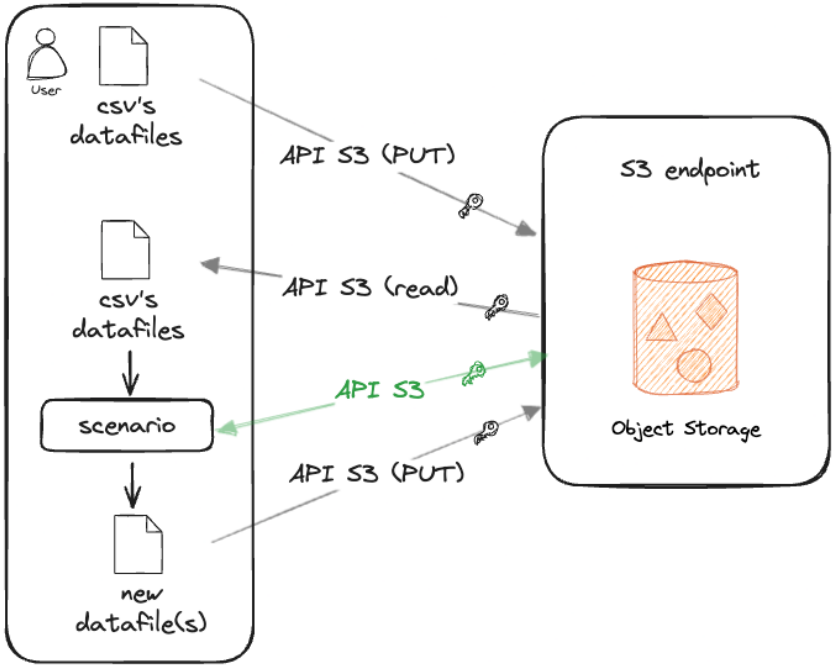
C’est quoi S3 et comment pouvons-nous l’utiliser dans nos Jupyter notebook (avec nos libraries et nos outils tiers) ?

- **Outils tiers :** QGIS, duckDB
- **Libraries** : pyarrow, geopandas, pandas, boto3, rasterio.
- **Format données** : csv, geoparquet (cloud optimized)
- **Dataset** : GLODAP V2.2023 ocean dataset: CSV
- **Contexte** : bucket S3 privé
- **AAI** : credentials individuels

Côté utilisateur

Tests des interactions entre Jupyter Notebook, les logiciels tiers et S3

- **Q1** : Accès à un objet dans un bucket S3 ?
- **Q2** : Interroger un objet S3 avec des outils externes (bib Python, duckDB…) ?
- **Q3** : Créer un nouvel objet basé sur des objets déjà stockés dans S3 ?
- **Q4** : Créer un nouvel objet en format optimisé cloud / un objet dans un format commun ?
- **Q5** : Télécharger un objet créé stocké dans S3 (public ou privé) ?



Côté utilisateur

Tests des interactions entre Jupyter Notebook, les logiciels tiers et S3

Q3 : Créer un nouvel objet basé sur des objets déjà stockés dans S3 ?

Réponse : OUI

Usage : partager un produit intermédiaire du scénario pour la reproductibilité

```
import pandas as pd
import numpy as np

df = pd.read_csv("s3://datascience/data/98646099999/2020.csv",
                 index_col='DATE',

usecols=['DATE','STATION','NAME','LONGITUDE','LATITUDE','ELEVATION','TMP','DEW','SLP
'],
)

df["air"] = df['TMP'].astype('str').str.partition(',')[0]
df.air = df.air.replace('+9999',np.nan).astype('float64')
df.air = df.air/10.

df.to_csv("s3://datascience/dataframe.csv")
```

Côté utilisateur

Tests des interactions entre Jupyter Notebook, les logiciels tiers et S3

Q4 : Créer un nouvel objet en format optimisé cloud / un objet dans un format commun ?

Réponse : OUI


```
df = pd.read_csv("s3://datascience/data/98646099999/2020.csv",
                 index_col='DATE',
                 usecols=['DATE','STATION','NAME','LONGITUDE','LATITUDE','ELEVATION',
)
```

...

Geo Parquet

```
import geopandas as gpd

# However, GeoPandas 1.0 will switch to use pyogrio as the default engine, since
pyogrio can provide a significant speedup compared to Fiona. We recommend to already
install pyogrio.
gpd.options.io_engine = "pyogrio"

gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.LONGITUDE, df.LATITUDE))
gdf.to_parquet("s3://datascience/geodataframe.parquet")
```

Côté utilisateur

Tests des interactions entre Jupyter Notebook, les logiciels tiers et S3

Q4 : Créer un nouvel objet en format optimisé cloud / un objet dans un format commun ?

Réponse : OUI

```
df = pd.read_csv('GLODAPv2.2023_Merged_Master_File.csv.zip',dtype={'G2expocode' :
str, 'G2Cruise' : int, 'G2region' :int , 'G2cast' :int ,'G2year': int,'G2month' :
int,'G2day' : int, 'G2hour': int, 'G2minute' : int, 'G2bottle' : int, 'G2doi' :
str})
```

```
df = pd.read_csv("s3://
```

...

Geo Parquet

```
# Creating a GeoPandas GeoDataFrame with Point geometries
from shapely.geometry import Point
gdf = gpd.GeoDataFrame(df,
                       geometry=gpd.points_from_xy(df['longitude'], df['latitude']),
                       crs='EPSG:4326')  # Set the coordinate reference system (CRS)

# write the geodataframe to a (geo) parquet file
gdf.to_parquet('glodap_v2_2023_merged_master.parquet')
```

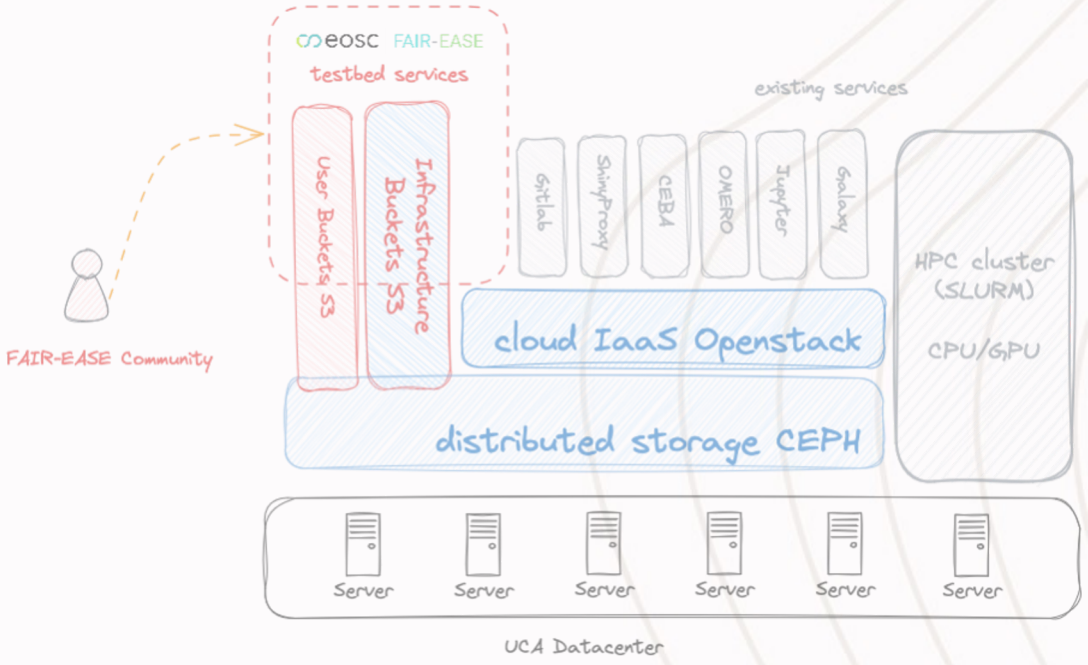
```
gdf.to_parquet("s3://
```

Côté utilisateur - RETOUR

Test des interactions entre Jupyter Notebook, les logiciels tiers et S3

- Aucun problème d'accès (identifiant individuel)
- Pas de verrou particulier pour la gestion des fichiers de données
- Les identifiants et le end point doivent parfois être utilisés explicitement (duckDB)

Infrastructure



eosc | FAIR-EASE

Côté infrastructure

??? et S3

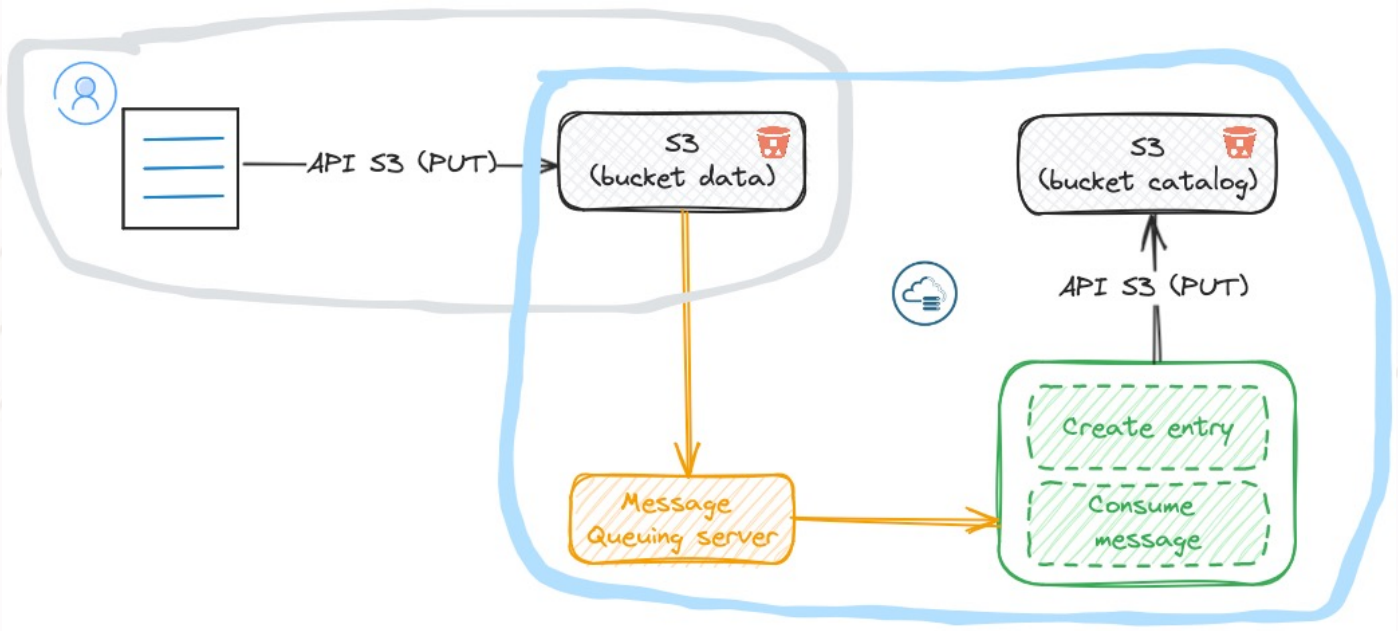
Question « naïve » côté infrastructure

Gestion d'un catalogue de données géospatiales sur S3 ?

- **Outils tiers :** STAC, RabbitMQ
- **Libraries** : pySTAC, rasterio, boto3
- **Format données** : geotiff (cloud optimized)
- **Dataset** : Copernicus DEM - Global and European Digital Elevation Model
- **Contexte** : bucket S3 privé et public
- **AAI** : credentials individuels

Côté infrastructure

Gestion catalogue et S3



Q1 : Manipuler un catalogue STAC statique directement sur S3 ?

Q2 : Utiliser les notifications S3 pour mettre à jour automatiquement un catalogue statique ?

Q3 : Gérer automatiquement les métadonnées de S3 vers STAC ?

Côté infrastructure

Gestion catalogue et S3

Q2: Utiliser les notifications S3 pour mettre à jour automatiquement un catalogue statique ?

Réponse : OUI, avec un serveur de messagerie dans le bon réseau et du code Python

- **Etape 1:**
  - un serveur RabbitMQ (pour la prise en charge du protocole de message AMQP)
  - un objet de type exchange définit dans RabbitMQ, nommé 'fairease'
- **Etape 2:**
  - un topic dans S3 pour publier des événements vers l'exchange « fairease »
  - 2 buckets : le catalogue, les données (avec des notifications S3 à la création d'objet)
- **Etape 3: code python**
  - Ouvre le catalogue à partir du bucket S3
  - Consomme les messages AMQP reçus sur l'exchange « fairease »
  - Récupère les informations d'événement S3 depuis les messages AMQP (uniquement création) et les métadonnées de l'objet
  - Lit l'objet GeoTIFF pour récupérer les coordonnées du polygone
  - Crée d'un item et d'un asset dans le catalogue STAC (avec ses données géographiques et temporelles)
  - Enregistre le catalogue S3


```
{
 "Records": [
   {
    "eventVersion": "2.2",
    "eventSource": "ceph:s3",
    "awsRegion": "fr-clermont-mesocentre",
    "eventTime": "2024-04-03T08:05:24.828135Z",
    "eventName": "ObjectCreated:Put",
    "userIdentity": {
      "principalId": "770af59709e34bb3ac574bfdc2a85eb0"
    },
    "requestParameters": {
      "sourceIPAddress": ""
    },
    "responseElements": {
      "x-amz-request-id": "9c81178a-7853-4b88-bfe0-d5d116fda944.1333150835.6316920358329002430",
      "x-amz-id-2": "4f764473-uca-oscar-fr-clermont-mesocentre"
    },
    "s3": {
      "s3SchemaVersion": "1.0",
      "configurationId": "data",
      "bucket": {
        "name": "uca-eoscfe-data",
        "ownerIdentity": {
          "principalId": "770af59709e34bb3ac574bfdc2a85eb0"
        },
        "arn": "arn:aws:s3:::uca-eoscfe-data",
        "id": "9c81178a-7853-4b88-bfe0-d5d116fda944.1333138354.60"
      },
      "object": {
        "key": "Copernicus_DSM_COG_30_N00_00_E006_00_DEM.tif",
        "size": 490119,
        "eTag": "0c3f5310131359cfa2dfb028bf3510fd",
        "versionId": "",
        "sequencer": "C40D0D66D24C5F36",
        "metadata": [
          {
            "key": "x-amz-content-sha256",
            "val": "UNSIGNED-PAYLOAD"
          },
          {
            "key": "x-amz-date",
            "val": "20240403T080524Z"
          },
          {
            "key": "x-amz-meta-fairease.catalog.mediatype",
            "val": "COG"
          }
        ],
        "tags": []
      }
    },
    "eventId": "1712131524.912215.0c3f5310131359cfa2dfb028bf3510fd",
    "opaqueData": ""
   }
 ]
}
```

Côté infrastructure

Gestion catalogue et S3

Q3: Peux-t-on gérer des metadonnées automatiquement S3 -> STAC

Réponse : OUI, metadonnées ajoutées à la création de l'objet S3 (mais pas après), disponible dans le message AMQP envoyé par S3 vers le server RabbitMQ

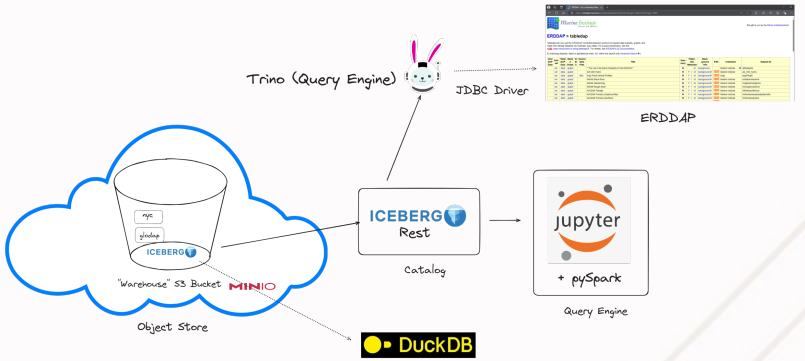
- **Etapes** :
  - En ligne de commande, il faut ajouter les metadonnées dans la commande d'upload de la donnée
    - metadonnée 'fairease.catalog.mediatype'
    - $ aws s3 cp --content-type image/tiff --metadata "fairease.catalog.mediatype=COG" Copernicus_DSM_COG_30_N00_00_E006_00_DEM.tif
    - s3://uca-eoscfe-data/Copernicus_DSM_COG_30_N00_00_E006_00_DEM.tif

Côté infrastructure - RETOUR

Gestion catalogue et S3

- Nécessité d’un message AMQP
- Déploiement d’outils nécessaires sur l'infrastructure (RabbitMQ)
- Notification configurée au niveau du bucket
- Gestion simple des métadonnées testée
- Enregistrement automatique pour un catalogue STAC statique

Retours et suite



- Fait
  - Notebook avec jeu de données océan GLODAP V2.2023 : CSV → geoparquet par Geopandas dans un notebook jupyter; requêtes locales : DuckDb, QGIS.
  - Tutoriels pour les interactions d'accés et d'outils S3
  - Tests catalogue STAC
  - Benchmark (Ifremer) : Minio, DuckDB / Argo (Easy One, General)
  - Exploration Apache Iceberg (Marine Institute) : Minio
- La suite
  - Tests sur des fichiers COG
  - Interaction S3/Galaxy
  - Galaxy pulsar

eosc | FAIR-EASE
Building Interoperable Earth Science & Environmental Services

Merci à toutes les personnes impliquées !!!