

# PROJET 8

CONSTRUISEZ ET TESTEZ UNE  
INFRASTRUCTURE DE DONNÉES

# *SOMMAIRE*

- 1. CONTEXTE
- 2. INSTALLATION D'AIRBYTE
- 3. COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES
- 4. MISE EN PLACE D'UN ENVIRONNEMENT DOCKER
- 5. DEPLOIEMENT SUR AWS
- 6. MONITORING ET REPORTING AWS

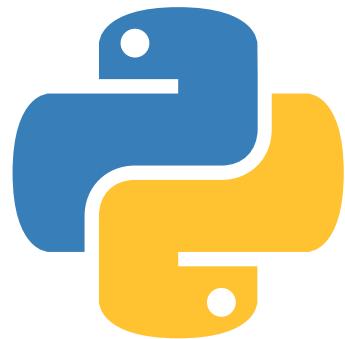
# CONTEXTE

Ce projet a été réalisé en tant que Data Engineer pour l'entreprise GreenCorp. Le but est de mettre à disposition un environnement avec des données traitées et fiables.

Ces données sont des données météorologiques qui ont pour but de fournir des informations manquantes pour la société.

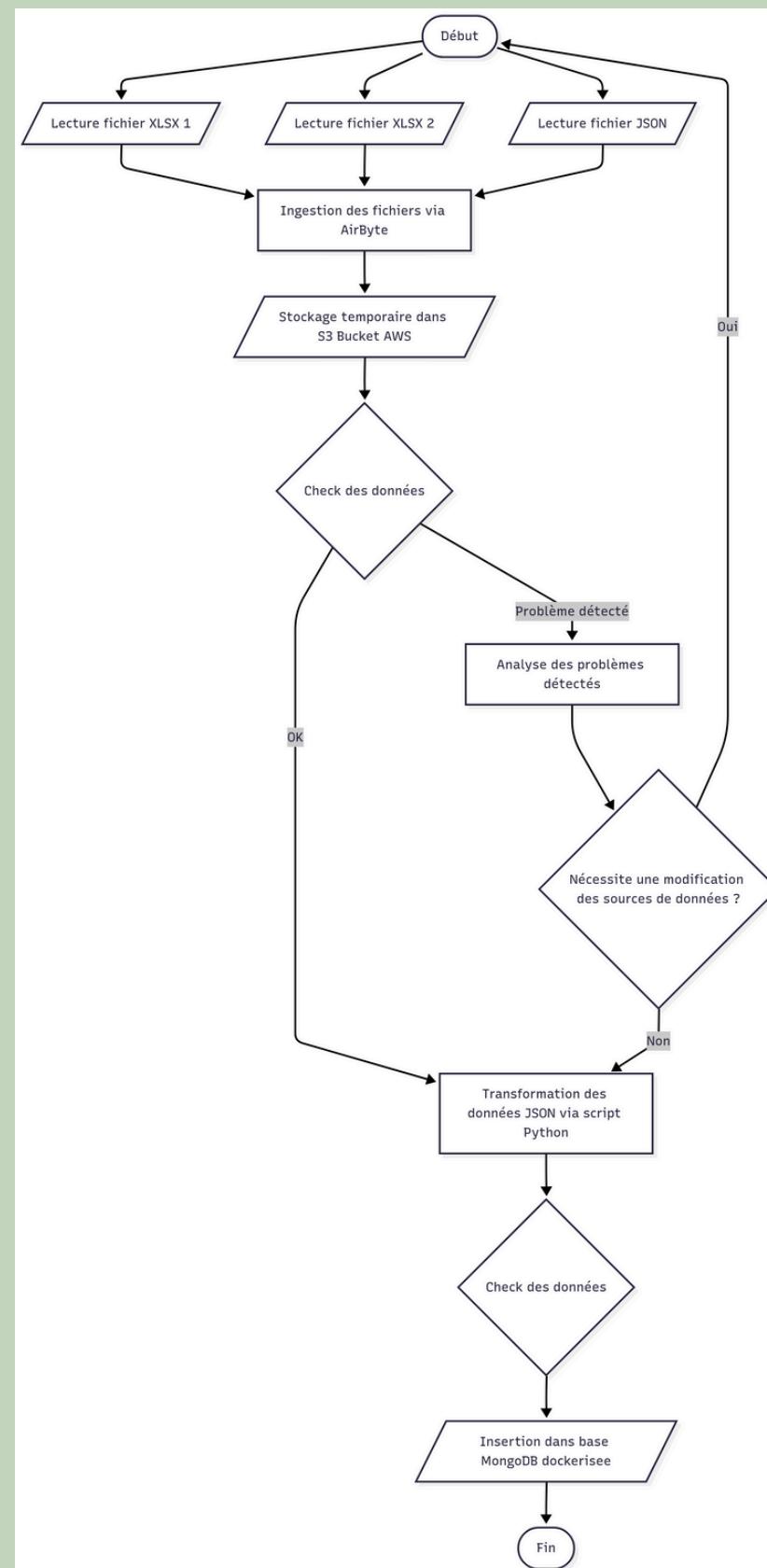
Voici les différentes étapes nécessaires pour traiter ce projet :

- Récupération des sources de données et intégrations de celles-ci dans un bucket S3 AWS avec l'aide d'AirByte
- Vérification, traitement et intégration des données dans une base de données MongoDB avec Docker
- Reproduction de l'environnement local dans une machine virtuelle EC2 associée à un cluster DocumentDB avec replica
- Monitoring et reporting de l'environnement final AWS



# CONTEXTE

Pour illustrer ce contexte, voici un logigramme du processus suivi de collecte, transformation et stockage des données





# INSTALLATION D'AIRBYTE

AirByte est un outil permettant notamment, de réaliser le processus d'ETL (extract-transform-load). Ici, nous allons l'utiliser pour faire de l'EL(extract-load). Notre instance a été mise en place via ABCTL qui est l'outil en ligne de commande conçu pour gérer AirByte.

Voici une capture d'écran des différentes connexion de nos sources vers notre bucket S3 AWS (destination) :

A screenshot of the Airbyte web interface, specifically the 'Connections' page. The sidebar on the left includes links for 'Connections', 'Sources', 'Destinations', 'Builder', and 'Settings'. The main area is titled 'Connections' and shows '9 healthy · 1 failed'. A search bar and filters for 'All connections', 'All statuses', 'All sources', and 'S3' are present. Three connections are listed:

NAME	SOURCE NAME	DESTINATION NAME	FREQUENCY	LAST SYNC	ENABLED
Belgique → S3Projet8	Belgique	S3Projet8	Manual	il y a 20 jours	<input checked="" type="checkbox"/>
data_source_json → S3Projet8	data_source_json	S3Projet8	Manual	il y a 20 jours	<input checked="" type="checkbox"/>
France → S3Projet8	France	S3Projet8	Manual	il y a 20 jours	<input checked="" type="checkbox"/>

# INSTALLATION D'AIRBYTE

Voici la représentation de nos 3 sources sur la console AWS dans le module S3.

Ici on récupère les 3 sources en json :



The screenshot shows the AWS S3 console interface. The top navigation bar includes the AWS logo, a search bar, and account information for Europe (Paris) and Nicolas. The main navigation path is Amazon S3 > Compartiments > openclassrooms-airbyte-project-8 > Project8/. The left sidebar lists various S3 management options like Compartiments à usage général, Compartiments de répertoires, and Access Grants. The main content area displays a list of objects named 'Project8/' with three entries: 'belgique/' (Dossier), 'data\_source\_json/' (Dossier), and 'france/' (Dossier). Action buttons for each object include Copy URI S3, Copy URL, Download, Open, Delete, Actions, Create a folder, and Upload. A 'Copier l'URI S3' button is also visible at the top right of the object list. A search bar at the bottom allows filtering by prefix.

Nom	Type	Dernière modification	Taille	Classe de stockage
belgique/	Dossier	-	-	-
data_source_json/	Dossier	-	-	-
france/	Dossier	-	-	-

# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES



```
Analyse du fichier : ../json/info_climat.jsonl
```

```
Vérification de la présence de champs obligatoires :
```

```
Tous les champs obligatoires sont présents et contiennent des données.
```

```
Vérification des types de données :
```

```
id_station          object
dh_utc              object
temperature         object
pression            object
humidite            object
point_de_rosee       object
visibilite          object
vent_moyen          object
vent_rafales         object
vent_direction       object
pluie_3h             object
pluie_1h             object
neige_au_sol         float64
nebulosite           object
temps_omm            object
station_id           object
station_name         object
latitude             float64
longitude            float64
elevation            int64
dtype: object
```

```
Types de données listés ci-dessus.
```

```
Taux de valeurs nulles par colonne :
```

```
neige_au_sol      1.000000
temps_omm         0.992126
visibilite        0.947507
vent_rafales       0.947507
nebulosite         0.947507
pluie_3h           0.920385
pluie_1h           0.787402
vent_direction     0.020122
longitude          0.000000
latitude           0.000000
station_name       0.000000
station_id         0.000000
id_station         0.000000
dh_utc             0.000000
vent_moyen         0.000000
point_de_rosee     0.000000
humidite           0.000000
pression           0.000000
temperature         0.000000
elevation          0.000000
dtype: float64
```

```
Colonnes avec plus de 70% de valeurs nulles :
```

```
visibilite        0.947507
vent_rafales       0.947507
pluie_3h           0.920385
pluie_1h           0.787402
neige_au_sol       1.000000
nebulosite         0.947507
temps_omm          0.992126
dtype: float64
```

```
Vérification des plages géographiques :
```

```
Toutes les coordonnées sont dans des plages valides.
```

```
Vérification des doublons de station :
```

```
1143 lignes concernées par des doublons potentiels.
```

```
Stations uniques (comme après un drop_duplicates) :
```

station_id	station_name	latitude	longitude
07015	Lille-Lesquin	50.575	3.092
00052	Armentières	50.689	2.877
000R5	Bergues	50.968	2.441
STATIC0010	Hazebrouck	50.734	2.545

# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES (cas anormal de valeurs)



Vérification de la qualité des données :

1 ligne(s) avec température non exploitable :

Dew Point	Precip.	Rate.	Time	Humidity	Wind	Gust	UV	Temperature	Speed	Pressure	Solar	Precip.	Accum.
49.4 °F	0.00	in	14:59:00	57 %	NW	5.5 mph	2.0	fake value	1.3 mph	29.60 in	246 w/m²	0.00	in

1 ligne(s) avec heure non exploitable :

Dew Point	Precip.	Rate.	Time	Humidity	Wind	Gust	UV	Temperature	Speed	Pressure	Solar	Precip.	Accum.
49.4 °F	0.00	in	14h	57 %	NW	5.5 mph	2.0	65.0 °F	1.3 mph	29.60 in	246 w/m²	0.00	in

Pour vérifier que les cas anormaux sont bien repérés lors de la première vérification de données, 2 lignes avec des données aux formats différents ont été ajoutées et elles sont bien mises en valeur par le script

- Température en string (“fake value”)
- Heure dans un format non conventionnel (“14h”)

# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES



Vérification des types de données :

```
Time          object
Solar         object
UV            float64
Precip. Accum. float64
Precip. Rate. float64
date          object
station_id    object
elevation     int64
temperature   object
pression      object
humidite      object
point_de_rosee object
visibilite    object
vent_moyen    object
vent_rafales  object
vent_direction object
pluie_3h      object
pluie_1h      object
nebulosite    object
temps_omm     object
dtype: object
```

Types de données listés ci-dessus.

Taux de valeurs nulles par colonne :

```
tempo_omm        0.998186
visibilite       0.987908
nebulosite       0.987908
pluie_3h         0.981661
pluie_1h         0.951028
UV               0.232769
Precip. Accum.  0.232769
Precip. Rate.   0.232769
Solar            0.232769
vent_rafales    0.220677
vent_direction   0.020153
vent_moyen       0.002418
Time             0.002418
point_de_rosee  0.002418
pression         0.002418
temperature      0.002418
humidite         0.002418
elevation        0.000000
station_id       0.000000
date             0.000000
```

On garde donc les colonnes selon la demande initiale. Celles qui avaient la même signification mais pas le même nom ont été merge et mis au même format (°F en °C par exemple) :

"Humidity": "humidite",  
"Temperature": "temperature",  
"Dew Point": "point\_de\_rosee",  
"Pressure": "pression",  
"Wind": "vent\_direction",  
"Speed": "vent\_moyen",  
"Gust": "vent\_rafales",  
"id\_station": "station\_id"

# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES

## AVANT TRANSFORMATION



```
1 {  
2     "_airbyte_raw_id": "6ef139ea-9340-4865-8924-644af4c7bd64",  
3     "_airbyte_extracted_at": 1748597667000,  
4     "_airbyte_generation_id": 2,  
5     "_airbyte_meta": {  
6         "changes": [],  
7         "sync_id": 136  
8     },  
9     "_airbyte_data": {  
10        "Time": "17:29:00",  
11        "Dew Point": "51.7°F",  
12        "Speed": "5.7mph",  
13        "Temperature": "57.5°F",  
14        "Humidity": "81%",  
15        "Gust": "7.8mph",  
16        "Pressure": "29.70in",  
17        "Solar": "64w/m²",  
18        "Wind": "NE",  
19        "UV": 0,  
20        "Precip. Accum.": "0.01in",  
21        "Precip. Rate.": "0.00in"  
22    }  
23 }  
24 {  
25     "Time": "17:29:00",  
26     "Solar": "64w/m²",  
27     "UV": 0.0,  
28     "Precip. Accum.": 0.254,  
29     "Precip. Rate.": 0.0,  
30     "date": "02-10-24",  
31     "station_id": "IICHTE19",  
32     "elevation": 15,  
33     "temperature": 14.17,  
34     "pression": 982.06,  
35     "humidite": 81.0,  
36     "point_de_rosee": 10.94,  
37     "visibilite": null,  
38     "vent_moyen": 8.05,  
39     "vent_rafales": 11.01,  
40     "vent_direction": "NE",  
41     "pluie_3h": null,  
42     "pluie_1h": null,  
43     "nebulosite": null,  
44     "temps_omm": null  
45 }  
46  
47 }
```

## APRES TRANSFORMATION

Réassort automatique de la date car perdu lors  
de la transition :  
XLSX → JSON (S3)

# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES

Import de ces données traitées dans la base de données MongoDB. Et voici un exemple d'un objet de chaque collection de la base une fois importée sur mongodb compass.

The screenshot shows the MongoDB Compass interface for the 'weather\_stations' collection. The top part of the image displays a terminal window with the following command and output:

```
root@cc8173934d99:/scripts# python data_import.py
4962 documents insérés dans 'backup.weather_informations'
6 documents insérés dans 'backup.weather_stations'
root@cc8173934d99:/scripts# _
```

The main Compass window shows the 'weather\_stations' collection with 6 documents. The document details are as follows:

**Document 1:**

```
_id: ObjectId('68541904e91c11fd606aada3')
station_id : "IICCHTE19"
station_name : "WeerstationBS"
latitude : 51.092
longitude : 2.999
```

**Document 2:**

```
_id: ObjectId('68541904e91c11fd606aada4')
station_id : "ILAMAD25"
station_name : "La Madeleine"
latitude : 50.659
longitude : 3.07
```

The Compass interface includes a toolbar with 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE' buttons. The bottom right corner of the interface has a red box highlighting the page number '100' and '1 - 6 of 6'.



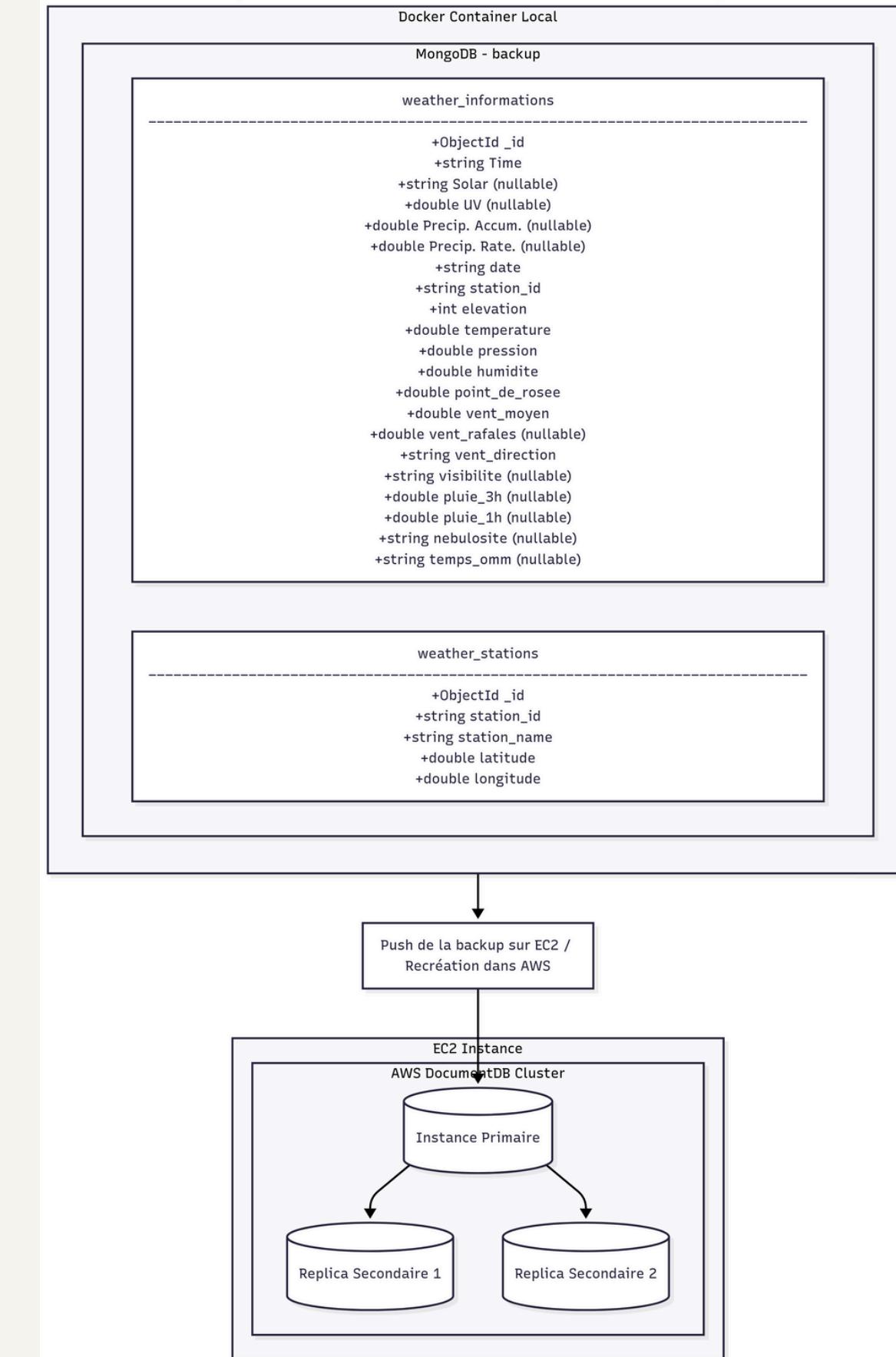
# COLLECTE/VERIFICATION ET TRANSFORMATION DES DONNEES

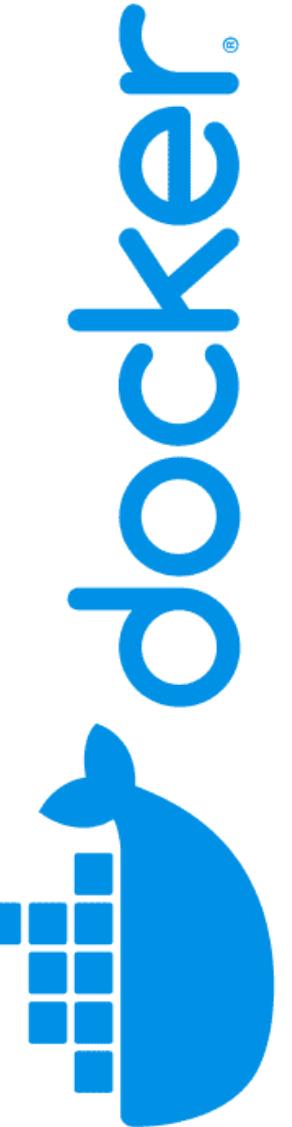
Import de ces données traitées dans la base de données MongoDB. Et voici un exemple d'un objet de chaque collection de la base une fois importée sur mongoDB compass.

The screenshot shows the MongoDB Compass interface for the 'weather\_informations' collection. The top navigation bar includes 'localhost:27018 > backup > weather\_informations' and an 'Open MongoDB shell' button. Below the navigation are tabs for 'Documents' (5.0K), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'. A search bar says 'Type a query: { field: 'value' } or Generate query'. Below the search are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. On the right, there's a pagination section with '100' selected, showing '1 - 100 of 4962' documents. A red box highlights this section. At the bottom, a code block displays a single document:

```
_id: ObjectId('68541904e91c11fd606a9a40')
Time : "00:04:00"
Solar : "0 w/m²"
UV : 0
Precip. Accum. : 0
Precip. Rate. : 0
date : "01-10-24"
station_id : "IICHTE19"
elevation : 15
temperature : 13.78
pression : 974.79
humidite : 87
point_de_rosee : 11.72
visibilite : null
vent_moyen : 11.58
vent_rafales : 14.68
vent_direction : "WSW"
pluie_3h : null
pluie_1h : null
nebulosite : null
temp_somm : null
```

# SCHEMA DE LA BASE DE DONNEES (VERS AWS)





# MISE EN PLACE D'UN ENVIRONNEMENT DOCKER

## ← DOCKER COMPOSE

```
FROM python:3.9-slim
```

```
RUN apt-get update && apt-get install -y \
    build-essential \
    curl \
    iputils-ping \
    && rm -rf /var/lib/apt/lists/*
```

```
RUN pip install --no-cache-dir pandas pymongo polars
```

```
CMD ["tail", "-f", "/dev/null"]
```

↑  
DOCKERFILE

```
❶ docker-compose.yml > ...
1  version: "3.8"
  ▷ Run All Services
2  services:
  ▷ Run Service
3    db_8:
4      container_name: mongodb_8_container
5      image: mongo
6      restart: always
7      healthcheck:
8        test: echo 'db.runCommand("ping").ok' | mongosh --quiet
9        interval: 5s
10       timeout: 5s
11       retries: 10
12      environment:
13        MONGO_INITDB_ROOT_USERNAME: root
14        MONGO_INITDB_ROOT_PASSWORD: a1B2c3D4e5
15      ports:
16        - "27018:27017"
17      networks:
18        mongodb_8_network:
19          ipv4_address: 172.35.0.8
20
  ▷ Run Service
21    python_8_app:
22      container_name: python_import_container
23      build:
24        context: .
25        dockerfile: Dockerfile
26      depends_on:
27        db_8:
28          condition: service_healthy
29      networks:
30        mongodb_8_network:
31          ipv4_address: 172.35.0.9
32      volumes:
33        - ./scripts:/scripts
34        - ./json:/json
35
36      networks:
37        mongodb_8_network:
38          driver: bridge
39          ipam:
40            config:
41              - subnet: 172.35.0.0/24
42
```

# DEPLOIEMENT SUR AWS

Vue de l'instance EC2 fonctionnelle et sécurisée depuis la console AWS



The screenshot shows the AWS CloudFormation console. In the top navigation bar, the region is set to 'Europe (Paris)' and the user is 'Nicolas'. The main view displays a successful deployment of a stack named 'mongodb-wea...'. The stack status is 'En cours d'exécution' (Running). The instance details show it's a 't2.micro' type with a public IP of '13.39.107.41' and a private IP of '172.31.43.172'. The DNS name is 'ec2-13-39-107-41.eu-west-3.compute.amazonaws.com'. The instance is associated with an IAM role 'EC2-ECS-Instance-Role'. The security group table at the bottom lists four inbound rules: port 22 (TCP) from 92.184.108.33/32, port 27017 (TCP) from 92.184.108.33/32, port 443 (TCP) from 0.0.0.0/0, and port 80 (TCP) from 0.0.0.0/0, all belonging to the security group 'mongodb-weather-ubuntu-wizard-1'.

Nom	ID de règle du groupe de... (Security Group Rule ID)	Plage de ports (Port Range)	Protocole (Protocol)	Source (Source)	Groupes de sécurité (Security Groups)	Description
-	sgr-0195089db558bd43f	22	TCP	92.184.108.33/32	<a href="#">mongodb-weather-ubuntu-wizard-1</a>	-
-	sgr-0a041755a5de18c4d	27017	TCP	92.184.108.33/32	<a href="#">mongodb-weather-ubuntu-wizard-1</a>	-
-	sgr-043a56db0d9203e01	443	TCP	0.0.0.0/0	<a href="#">mongodb-weather-ubuntu-wizard-1</a>	-
-	sgr-075fa272700a80187	80	TCP	0.0.0.0/0	<a href="#">mongodb-weather-ubuntu-wizard-1</a>	-

# DEPLOIEMENT SUR AWS

Vue de l'instance EC2 avec connexion SSH sur un terminal local



```
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Thu Jun 19 14:56:08 UTC 2025

System load: 0.0          Processes: 107
Usage of /: 72.4% of 6.71GB  Users logged in: 0
Memory usage: 35%          IPv4 address for enX0: 172.31.43.172
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

15 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Jun 19 14:56:09 2025 from 92.184.108.33
ubuntu@ip-172-31-43-172:~$
```

# DEPLOIEMENT SUR AWS

## Utilisation de DocumentDB pour répliquer les données (dump/transfert dump/restore)

### DUMP DE LA BASE DE DONNEES LOCALE (MONGODUMP)

```
nicolas.petit@humans.local@HCKL062:~/Documents/Data_Engineering/Projet_8 $ mongodump --host 172.35.0.8 -u root -p a1B2c3D4e5 --authenticationDatabase admin --db backup --out .
2025-06-19T17:16:02.640+0200      writing backup.weather_stations to backup/weather_stations.bson
2025-06-19T17:16:02.641+0200      done dumping backup.weather_stations (6 documents)
2025-06-19T17:16:02.644+0200      writing backup.weather_informations to backup/weather_informations.bson
2025-06-19T17:16:02.661+0200      done dumping backup.weather_informations (4962 documents)
```

### SCP VERS L'INSTANCE EC2

```
nicolas.petit@humans.local@HCKL062:~/Documents/Data_Engineering/Projet_8 $ scp -i mongodb-keys.pem -r backup/ ubuntu@ec2-13-39-107-41.compute.amazonaws.com:/home/ubuntu
weather_informations.metadata.json                                         100%   187      4.6KB/s  00:00
weather_stations.bson                                                 100%   678      16.6KB/s  00:00
weather_informations.bson                                             100% 1803KB  706.9KB/s  00:02
weather_stations.metadata.json                                         100%   183      3.8KB/s  00:00
```

### RESTORE DE LA BASE DE DONNEES DANS LE CLUSTER PRINCIPAL (MONGORESTORE)

```
ubuntu@ip-172-31-43-172:~$ mongorestore --host mongodb-weather-information.cluster-c728mg42uo50.eu-west-3.docdb.amazonaws.com:27017 --ssl --sslCAFile global-bundle.pem --username ubuntu --password Mongodb123 --authenticationDatabase admin --db backup /home/ubuntu/backup/
2025-06-19T15:23:54.712+0000  The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2025-06-19T15:23:54.712+0000  building a list of collections to restore from /home/ubuntu/backup dir
2025-06-19T15:23:54.712+0000  reading metadata for backup.weather_informations from /home/ubuntu/backup/weather_informations.metadata.json
2025-06-19T15:23:54.713+0000  reading metadata for backup.weather_stations from /home/ubuntu/backup/weather_stations.metadata.json
2025-06-19T15:23:54.745+0000  restoring backup.weather_informations from /home/ubuntu/backup/weather_informations.bson
2025-06-19T15:23:54.778+0000  restoring backup.weather_stations from /home/ubuntu/backup/weather_stations.bson
2025-06-19T15:23:54.789+0000  finished restoring backup.weather_stations (6 documents, 0 failures)
2025-06-19T15:23:55.122+0000  finished restoring backup.weather_informations (4962 documents, 0 failures)
2025-06-19T15:23:55.122+0000  no indexes to restore for collection backup.weather_informations
2025-06-19T15:23:55.122+0000  no indexes to restore for collection backup.weather_stations
2025-06-19T15:23:55.122+0000  4968 document(s) restored successfully. 0 document(s) failed to restore.
```



# DEPLOIEMENT SUR AWS

## Vérification de la bonne intégration des données sur l'instance principale en ligne de commande (mongosh)



```
ubuntu@ip-172-31-43-172:~$ mongosh "mongodb://ubuntu:Mongodb123@mongodb-weather-informations.cluster-c728mg42uo50.eu-west-3.docdb.amazonaws.com:27017/backup?ssl=true" --tlsCAFile global-bal-bundle.pem
Current Mongosh Log ID: 68542bb8e13437c62f1e9d59
Connecting to:      mongodb://<credentials>@mongodb-weather-informations.cluster-c728mg42uo50.eu-west-3.docdb.amazonaws.com:27017/backup?ssl=true&directConnection=true&tlsCAFile=global-bal-bundle.pem&appName=mongosh+2.1.5
Using MongoDB:      5.0.0
Using Mongosh:      2.1.5
mongosh 2.5.3 is available for download: https://www.mongodb.com/try/download/shell
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
Warning: Non-Genuine MongoDB Detected
This server or service appears to be an emulation of MongoDB rather than an official MongoDB product.
Some documented MongoDB features may work differently, be entirely missing or incomplete, or have unexpected performance characteristics.
To learn more please visit: https://dochub.mongodb.org/core/non-genuine-mongodb-server-warning.
-----

rs0 [direct: primary] backup> db.weather_informations.countDocuments()
4962
rs0 [direct: primary] backup> db.weather_informations.findOne()
{
  _id: ObjectId('68541904e91c11fd606a9a40'),
  Time: '00:04:00',
  Solar: '0 w/m²',
  UV: 0,
  'Precip. Accum.': 0,
  'Precip. Rate.': 0,
  date: '01-10-24',
  station_id: 'IICHTE19',
  elevation: 15,
  temperature: 13.78,
  pression: 974.79,
  humidite: 87,
  point_de_rosee: 11.72,
  visibilite: null,
  vent_moyen: 11.58,
  vent_rafales: 14.68,
  vent_direction: 'WSW',
  pluie_3h: null,
  pluie_1h: null,
  nebulosite: null,
  temps_omm: null
}
```

# DEPLOIEMENT SUR AWS

## Vérification de la bonne intégration des données sur l'instance secondaire en ligne de commande (mongosh)



```
ubuntu@ip-172-31-43-172:~$ mongosh "mongodb://ubuntu:MongoDb123@mongodb-weather-information2.c728mg42uo50.eu-west-3.docdb.amazonaws.com:27017/backup?ssl=true" --tlsCAFile global-bundle.pem
Current Mongosh Log ID: 68542d5c0047cde022da119e
Connecting to:      mongodb://<credentials>@mongodb-weather-information2.c728mg42uo50.eu-west-3.docdb.amazonaws.com:27017/backup?ssl=true&directConnection=true&tlsCAFile=global-bundle.pem&appName=mongosh+2.1.5
Using MongoDB:      5.0.0
Using Mongosh:      2.1.5
mongosh 2.5.3 is available for download: https://www.mongodb.com/try/download/shell
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
Warning: Non-Genuine MongoDB Detected
This server or service appears to be an emulation of MongoDB rather than an official MongoDB product.
Some documented MongoDB features may work differently, be entirely missing or incomplete, or have unexpected performance characteristics.
To learn more please visit: https://dochub.mongodb.org/core/non-genuine-mongodb-server-warning.
-----

rs0 [direct: secondary] backup> db.weather_information.countDocuments()
4962
rs0 [direct: secondary] backup> db.weather_information.findOne()
{
  _id: ObjectId('68541904e91c11fd606a9a40'),
  Time: '00:04:00',
  Solar: '0 w/m²',
  UV: 0,
  'Precip. Accum.': 0,
  'Precip. Rate.': 0,
  date: '01-10-24',
  station_id: 'IICHTE19',
  elevation: 15,
  temperature: 13.78,
  pression: 974.79,
  humidite: 87,
  point_de_rosee: 11.72,
  visibilite: null,
  vent_moyen: 11.58,
  vent_rafales: 14.68,
  vent_direction: 'WSW',
  pluie_3h: null,
  pluie_1h: null,
  nebulosite: null,
  temps_omm: null
}
```

# MONITORING ET REPORTING AWS



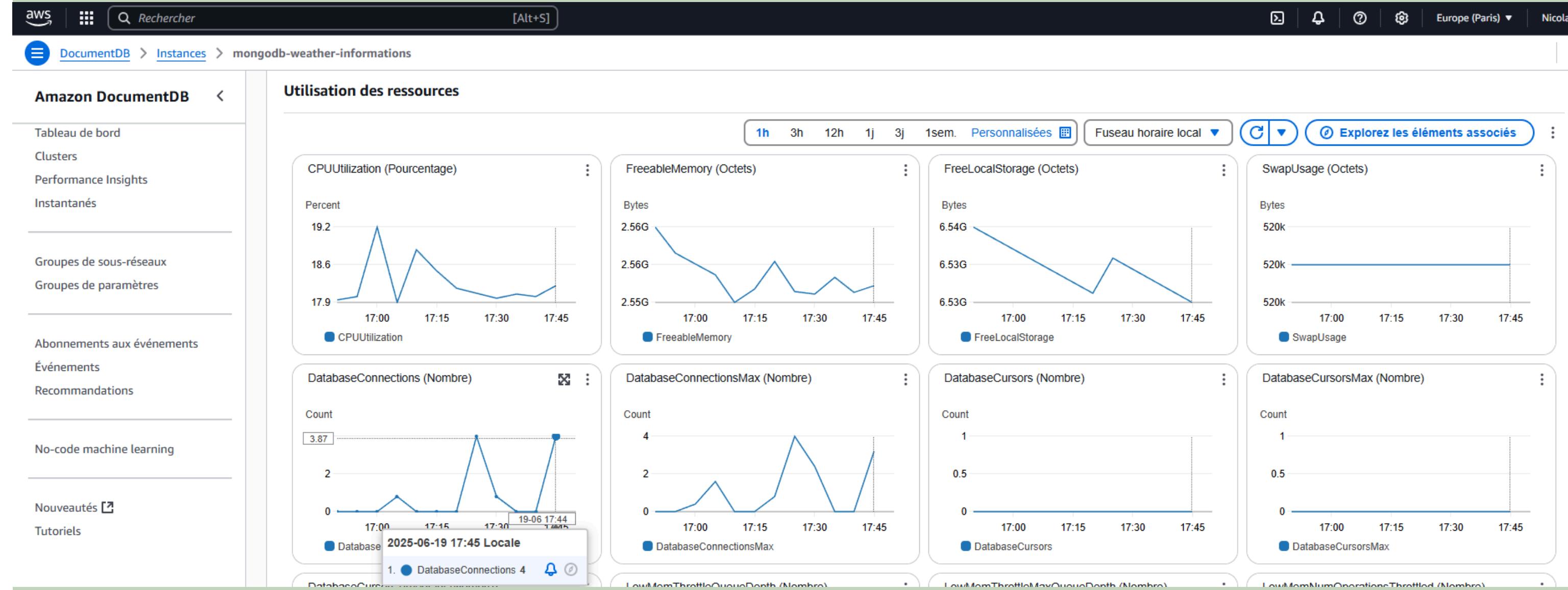
The screenshot shows the AWS DocumentDB Clusters management interface. On the left sidebar, under 'Clusters', there are four instances listed: 'mongodb-weather-information' (Cluster régional), 'mongodb-weather-information' (Instance principale), 'mongodb-weather-information2' (Instance de réplica), and 'mongodb-weather-information3' (Instance de réplica). Each instance is marked as 'Available'. The main table provides detailed information for each instance, including their status, role, engine version, region, and current activity. The CPU usage for the first instance is shown as 18.23% with 0 connections.

Identifiant du cluster	Statut	Rôle	Version du moteur	Région et zone d...	État de l'insta...	CPU	Activité actuelle
<a href="#">mongodb-weather-information</a>	Available	Cluster régional	5.0.0	eu-west-3	-	-	-
<a href="#">mongodb-weather-information</a>	Available	Instance principale	5.0.0	eu-west-3b	En bon état	18.23%	0 Connexions
<a href="#">mongodb-weather-information2</a>	Available	Instance de réplica	5.0.0	eu-west-3a	En bon état	17.13%	4 Connexions
<a href="#">mongodb-weather-information3</a>	Available	Instance de réplica	5.0.0	eu-west-3c	En bon état	18.41%	0 Connexions

## REMARQUES :

- On peut remarquer que toutes les instances sont en bon état et sur une zone de disponibilité différente
- On peut voir en direct le nombre de connexions
- On peut voir les performances du CPU pour chaque instance

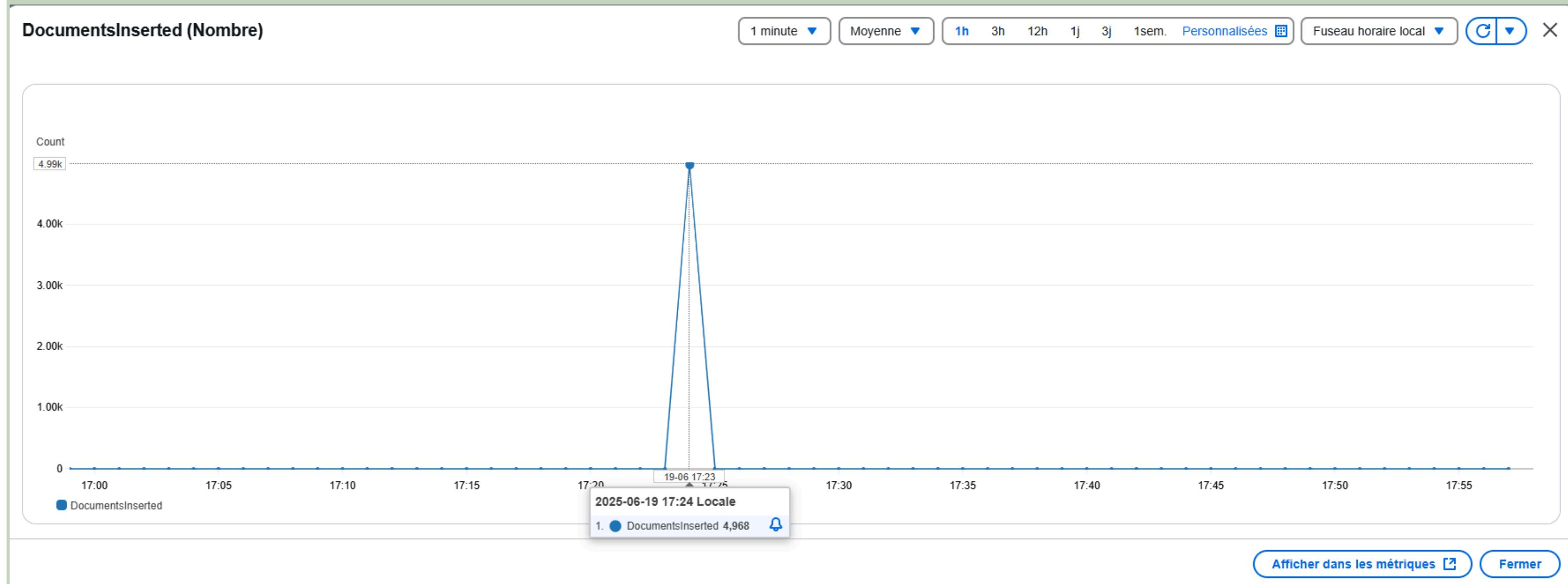
# MONITORING ET REPORTING AWS



## REMARQUES :

- On peut rentrer dans le détail de l'utilisation des ressources et donc de la gestion des performances de chaque instance avec CloudWatch
- Plus de détails sur les dates et heures des évènements

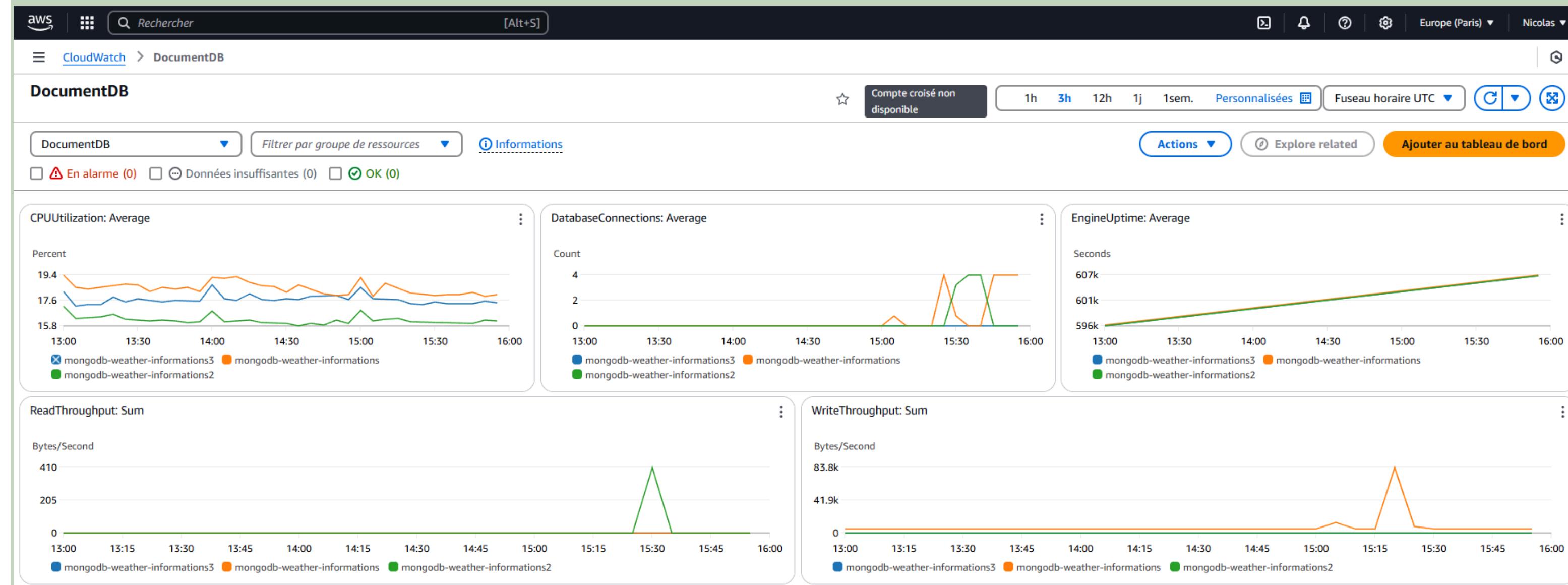
# MONITORING ET REPORTING AWS



## REMARQUES :

- Vérification de l'intégration du bon nombre de documents directement avec CloudWatch
- Possibilité de traquer par dates et heures les différents documents intégrés

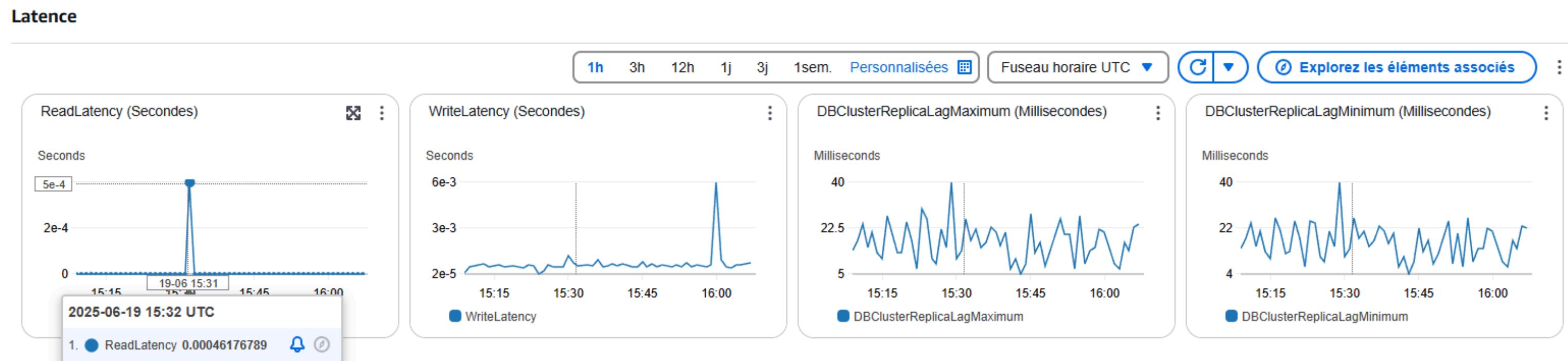
# MONITORING ET REPORTING AWS



## REMARQUES :

- Possibilité d'accéder à un dashboard CloudWatch du cluster DocumentDB
- Visibilité des métriques par instance avec une comparaison directe entre chacune d'elles

# MONITORING ET REPORTING AWS



## REMARQUES:

- Vision des temps de latence d'écriture et de lecture
- Métriques facilement accessible pour chaque instance pour détecter des problèmes de performance

# *MERCI!*

CONSTRUISEZ ET TESTEZ UNE  
INFRASTRUCTURE DE DONNÉES