

Compte rendu TP Pipeline CI/CD

Membres du groupe : Sonia AYEM, Zaineb BOUDHRIQUA et Samy Lacombe

Partie 1 : Installation des prérequis

Installation de git :

```
root@ip-172-31-84-73:/home/ubuntu# git --version
git version 2.34.1
```

Installation de nodeJS et NPM :

```
root@ip-172-31-84-73:/home/ubuntu# node -v
v12.22.9
root@ip-172-31-84-73:/home/ubuntu# npm -v
8.5.1
root@ip-172-31-84-73:/home/ubuntu#
```

Nous avons ensuite créé un fichier de déploiement docker avec un fichier docker compose pour déployer le une bdd PostgreSQL:

```
GNU nano 6.2
version: '3.9'

services:
  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: ynov
      POSTGRES_DB: ynov
    ports:
      - "5432:80"
```

Nous avons ici notre conteneur qui fonctionne correctement

```
root@ip-172-31-84-73:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
118fc40657c5   postgres  "docker-entrypoint.s..." About an hour ago Up About an hour 5432/tcp, 0.0.0.0:5432->80/tcp, :::5432->80/tcp  ubuntu_db_1
```

Nous avons mis en suite en place un bucket S3 à l'aide de votre vidéo :

Compartiments (1) [Info](#)

Les compartiments sont des conteneurs pour les données stockées dans S3. [En savoir plus](#)

Q

Rechercher des compartiments par nom

Nom

▲

Région AWS

▼

Accéder

▼

Date de création

▼

○

ynobuckets3

USA Est (Virginie du Nord) us-east-1

Public

external

20 Apr 2023 02:58:08 PM CEST

Créer un compartiment

Et voici les permissions avec l'aide de votre readme

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1625306057759",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::ynovbuckets3",
        "arn:aws:s3:::ynovbuckets3/*"
      ]
    }
  ]
}
```

Nous avons cloné le repos github sur notre machine puis configurer le fichier set_env.sh avec les identifiants de nos bases S3 et postgresSQL.

```
GNU nano 6.2

# This file is used for convenience of local development.
# DO NOT STORE YOUR CREDENTIALS INTO GIT
export POSTGRES_USERNAME=ynov
export POSTGRES_PASSWORD=ynov
export POSTGRES_HOST=ynov
export POSTGRES_DB=ynov
export AWS_BUCKET=ynovbuckets3
export AWS_REGION=us-east-1
export AWS_PROFILE=default
export JWT_SECRET=testing
export URL=http://localhost:5432
```

Après avoir installé notre app, nous pouvons la lancer avec npm run dev

```

root@ip-172-31-64-73:/home/ubuntu/ynov-ap/m2/udagram-api-feed# npm run dev
> udagram-api@2.0.0 dev
> ts-node-dev --respawn --transpile-only ./src/server.ts

[INFO] 14:02:22 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.1, typescript ver. 4.9.3)
Initialize database connection...
(node:9677) UnhandledPromiseRejectionWarning: SequelizeConnectionError: Connection terminated unexpectedly
    at Client._connectionCallback (/home/ubuntu/ynov-ap/m2/udagram-api-feed/node_modules/sequelize/src/dialects/postgres/connection-manager.js:193:20)
    at Connection.<anonymous> (/home/ubuntu/ynov-ap/m2/udagram-api-feed/node_modules/pg/lib/client.js:144:18)
    at Object.onceWrapper (events.js:420:35)
    at Connection.emit (events.js:314:20)
    at Connection.EventEmitter.emit (domain.js:483:12)
    at Socket.<anonymous> (/home/ubuntu/ynov-ap/m2/udagram-api-feed/node_modules/pg/lib/connection.js:107:12)
    at Socket.emit (events.js:326:22)
    at Socket.EventEmitter.emit (domain.js:483:12)
    at endReadableNT (stream_readable.js:1241:12)
    at processTicksAndRejections (internal/process/task_queues.js:84:21)
(node:9677) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled promise rejection, use the CLI flag "--unhandled-rejections=strict" (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode). (rejection id: 1)
(node:9677) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.

```

Par manque de temps et en ayant rencontré quelques problèmes lors de notre projet, nous n'avons pas pu avancer plus. Nous essayerons de finir ce TP ce vendredi 21/04 si nous avons le temps.