Teste SRE:

Escrevo abaixo uma descrição mais informal do que foi analisado, identificado e modificado no arquivo de Docker Compose.

A primeira coisa a ser feita é testar a execução do docker compose com o comando abaixo:

“**docker compose --profile flower up --force-recreate --always-recreate-deps -d --remove-orphans**”

Depois de um tempo tentando inicializar, o container do airflow-init “crasha”.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Sendo assim, para verificar os logs do container usei o comando:

“**docker container logs dre-3-test-main-airflow-init-1**”

Percebi que o erro era relacionado à credenciais do usuário “airflow”. Esse usuário não foi criado dentro do serviço do postgres. Alterei então a variável de ambiente do serviço ‘postgres’ desta forma:

**Antes: POSTGRES\_USER: admin**

**Depois: POSTGRES\_USER:airflow**

Mas para que isso tenha efeito, é preciso primeiro executar alguns comandos como:

“**docker compose stop**”

“**docker compose down**”

“**docker volume prune**”

“**docker volume rm dre-3-test-main\_postgres-db-volume**”

Após esses comandos, limpamos todos os nossos containers que sobraram e os volumes usados pelo postgres. Sem essa “limpeza”, o volume não seria alterado.

Após essa primeira etapa, executamos novamente o compose.

“**docker compose --profile flower up --force-recreate --always-recreate-deps -d --remove-orphans**”

Com todos os containers criados e startados, verificamos o status com o comando “**docker container ps**”.

O container do WebServer apresenta status ‘unhealthy’. Após verificar o serviço no arquivo yaml, percebi que a URL para teste estava sem uma porta configurada, apontando para ‘XXXX’. Sendo assim, modifiquei para a porta 8080.

Após a mudança, podemos subir apenas o serviço do airflow-webserver mas durante os testes, optei por sempre realizar uma inicialização ‘zerada’.

Subindo novamente o compose, todos os containers agora estão com status ‘healthy’.

Tudo recriado e agora o nosso Airflow está funcionando. Porém, ao logar na interface web, percebi que não havia nenhuma DAG encontrada.

Analisando o arquivo yaml, percebo a falta de um ‘s’ no mapeamento do volume que será o “repositório” da nossa DAG desejada.

Alterado o mapeamento:

Antes: ./dag:/opt/airflow/dags

Depois: ./dags:/opt/airflow/dags

Como não é um problema no banco de dados, não há necessidade de remover o volume novamente. Sendo assim, apenas executamos o compose up novamente.

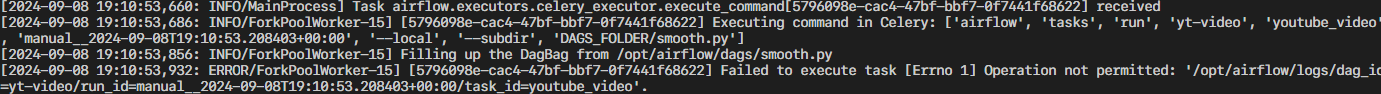
Após acessar a interface web, um erro na DAG foi encontrado. Apenas um erro de sintaxe no arquivo smooth.py. Adicionando o caractere ‘:’ no final da criação do método ‘smooth()’ na linha:

def smooth() 🡨 ‘:’

Alteração feita, DAG atualizada e pronto para testar sua execução.

A execução da DAG retorna um erro. Para verificar qual erro seria, verifiquei os logs do container de serviço “airflow-worker”.

Vendo os logs, me deparei com o erro abaixo:



*[2024-09-08 19:10:53,932: ERROR/ForkPoolWorker-15] [5796098e-cac4-47bf-bbf7-0f7441f68622] Failed to execute task [Errno 1]* ***Operation not permitted: '/opt/airflow/logs/dag\_id=yt-video/run\_id=manual\_\_2024-09-08T19:10:53.208403+00:00/task\_id=youtube\_video'.***

*[…]*

*File "/usr/local/lib/python3.7/pathlib.py", line 1291, in* ***chmod***

*self.\_accessor.chmod(self, mode)*

***PermissionError: [Errno 1] Operation not permitted: '/opt/airflow/logs/dag\_id=yt-video/run\_id=manual\_\_2024-09-08T19:10:53.208403+00:00/task\_id=youtube\_video'***

***[2024-09-08 19:10:53,944: ERROR/ForkPoolWorker-15] Task airflow.executors.celery\_executor.execute\_command[5796098e-cac4-47bf-bbf7-0f7441f68622] raised unexpected: AirflowException('Celery command failed on host: c5846d2ed83c with celery\_task\_id 5796098e-cac4-47bf-bbf7-0f7441f68622')***

Mesmo com o usuário criado e permissões das pastas OK, o erro persiste.

Verifiquei que se trata de um erro na imagem do apache/airflow:2.5.1.

Ref.: <https://github.com/apache/airflow/issues/29112>

<https://github.com/airflow-helm/charts/issues/713>

Alterei então a versão da imagem usada para 2.5.0.

Com essa imagem, a DAG executa com êxito.

Algumas configurações foram feitas mas não são necessárias para o funcionamento do ambiente, são elas:

* Alteração de variáveis de ambiente da imagem em comum:
  + AIRFLOW\_\_DATABASE\_\_SQL\_ALCHEMY\_CONN:postgresql+psycopg2://airflow:airflow@postgres/airflow

AIRFLOW\_\_DATABASE\_\_SQL\_ALCHEMY\_CONN: postgresql+psycopg2://airflow:airflow@postgres:5432/airflow

* + AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND: db+postgresql://airflow:airflow@postgres/airflow

AIRFLOW\_\_CELERY\_\_RESULT\_BACKEND: db+postgresql://airflow:airflow@postgres:5432/airflow

* Criado um arquivo .env no mesmo diretório do arquivo compose.yaml para que o AIRFLOW\_UID fosse setado. Feito isso para remover o warning ao executar o Docker compose up.
* Adicionado a configuração container\_name em todos os services para facilitar identificação dos containers e linhas de comando.