Synthèse - DAG Airflow Flask API

Document généré le 24/10/2025 à 08:46

Ce document présente une synthèse complète du DAG **Airflow_Lab2_Flask**, son fonctionnement, l'architecture mise en place, ainsi que les procédures de dépannage en cas de blocage.

1. Objectif du DAG

Le DAG **Airflow_Lab2_Flask** a pour objectif de démarrer et maintenir actif un serveur Flask qui expose une API web. Cette API permet de consulter l'état d'exécution d'un autre DAG Airflow (par défaut **Airflow_Lab2**) via une interface web accessible sur le port 5555.

2. Architecture et Composants

Composant	Description	Localisation
DAG Airflow	Orchestrateur du déploiement Flask	dags/Flask_API.py
Serveur Flask	API web exposant l'état des DAG runs	Fonction start_flask_app()
API REST Airflow	Source des données d'état	http://localhost:8080/api/v2
Port réseau	Point d'accès à l'API Flask	5555 (TCP)

3. Fonctionnement Détaillé

Étape 1 : Démarrage du DAG

Le DAG est déclenché manuellement (schedule=None). Il n'y a qu'une seule tâche : start_Flask_API.

Étape 2 : Vérification du port

La fonction verifier_port_disponible() teste si le port 5555 est libre.

Étape 3 : Gestion de l'état du port

Si le port est occupé, la fonction tester_api_existante() vérifie si c'est notre API Flask qui tourne déjà. Si oui, la tâche entre en mode surveillance (boucle de 30 secondes). Si non, une erreur est levée.

Étape 4 : Lancement de Flask

Si le port est libre, Flask démarre sur 0.0.0.0:5555 avec app.run(). Le serveur reste actif et écoute les requêtes HTTP.

Étape 5 : Routes exposées

L'API Flask expose 4 routes : / (redirection), /success, /failure, et /health. Ces routes interrogent l'API REST d'Airflow pour récupérer l'état du DAG cible.

4. Problèmes Courants et Solutions

Problème : Le DAG tourne en rond et ne s'arrête pas
Cause : La tâche Flask reste en état "running" car le serveur Flask (app.run()) est une boucle bloquante qui ne se termine pas. C'est le comportement normal attendu pour un service web long-running.

Solution : Pour arrêter proprement le DAG, il faut :

```
# 1. Identifier le processus Flask
lsof -i :5555

# 2. Tuer le processus (remplacer PID par le numéro trouvé)
kill -9 <PID>

# 3. Marquer le run comme échoué dans Airflow
airflow dags list-runs Airflow_Lab2_Flask --state running
airflow dags set-run-state Airflow_Lab2_Flask <run_id> failed

# 4. Nettoyer les tâches bloquées
airflow tasks clear Airflow_Lab2_Flask --yes
```

Problème : Port 5555 déjà utilisé lors du redémarrage
Cause : Un processus Flask orphelin continue de tourner après l'arrêt du DAG.

Solution : Libérer le port avant de relancer le DAG avec la commande kill -9.

5. Configuration

Variable	Valeur par défaut	Description
AIRFLOW_WEBSERVER	http://localhost:8080	URL du serveur Airflow
AIRFLOW_USERNAME	airflow	Utilisateur pour l'API REST
AIRFLOW_PASSWORD	airflow	Mot de passe pour l'API REST
TARGET_DAG_ID	Airflow_Lab2	DAG dont on surveille l'état

FLASK_PORT	5555	Port d'écoute du serveur Flask

Note : Ces variables peuvent être modifiées via des variables d'environnement avant le lancement d'Airflow.

6. Commandes Utiles

Action	Commande
Lister les DAGs	`airflow dags list | grep Flask`
Voir l'état des runs	`airflow dags list-runs Airflow_Lab2_Flask`
Déclencher le DAG	`airflow dags trigger Airflow_Lab2_Flask`
Mettre en pause	`airflow dags pause Airflow_Lab2_Flask`
Reprendre	`airflow dags unpause Airflow_Lab2_Flask`
Voir les logs	`airflow tasks logs Airflow_Lab2_Flask start_Flask_API <run_id>`
Tester l'API Flask	`curl http://localhost:5555/health`

7. Conclusion

Le DAG **Airflow_Lab2_Flask** est conçu pour fonctionner comme un service long-running qui expose une API Flask. Son comportement "en boucle" est normal et attendu. Pour l'arrêter, il est nécessaire de tuer manuellement le processus Flask et de nettoyer l'état dans Airflow.

Points clés à retenir :
• Le DAG ne se termine pas automatiquement (c'est voulu)
• Toujours vérifier si le port 5555 est libre avant de relancer
• Utiliser les commandes de nettoyage en cas de blocage
• L'API Flask doit être accessible sur http://localhost:5555