

# Airflow StarRocks 접속 오류



airflow 2.9 에서 starrocks 를 connection id 로 접근하는데, 접속이 실패했다고 하네.  
mysql cli 로 starrocks 에 접속할때는 접속이 잘되는데  
(예. mysql -u pdpdev -ppdpdev pdpdev 이런식으로)



가능성 좁혀봅시다. mysql CLI 는 붙는데 Airflow `conn\_id` 로만 실패라면, 보통은 “드라이버/포트/호스트/권한” 네 군데에서 빠깃합니다. 아래 순서대로 체크해보세요. (쿠버네티스/Helm/ArgoCD 환경 기준)

## 0) StarRocks 접속 기본값 상기

- FE MySQL 프로토콜 포트: **9030**
- FE Thrift/HTTP 포트: 9020/8030 (이걸로는 Airflow가 못 붙습니다)
- Kubernetes 서비스 DNS 예: `starrocks-fe.<네임스페이스>.svc.cluster.local`

## 1) Airflow Connection 설정을 “정석”으로

Airflow UI → **Admin** → **Connections** 에서:

- **Conn Type:** `MySQL`
- **Host:** `starrocks-fe.<ns>.svc.cluster.local`
- **Schema:** `pdpdev`
- **Login/Password:** `pdpdev / pdpdev` (예시)
- **Port:** `9030`
- **Extra:** 비우세요(SSL 안 쓰면). 타임아웃만 줄 거면:

```
{"connect_timeout": 10, "read_timeout": 30, "charset": "utf8mb4"}
```

URI 방식으로 한 번에 넣고 싶으면:



```
mysql+pymysql://pdpdev:pdpdev@starrocks-fe.<ns>.svc.cluster.local:9030/pdpdev?
charset=utf8mb4
```

포인트는 ``mysql+**pymysql**`` 입니다. (C 확장인 ``mysqlclient`` 가 이미지에 없거나 깨져있어도 PyMySQL로 우회됩니다.)

### 필수 패키지

- ``apache-airflow-providers-mysql`` (Airflow 2.9용)
- ``PyMySQL``

이미지에 없다면 Helm values에 추가(예: ``env`` 또는 커스텀 이미지에 ``pip install``).

## 2) Airflow 파드 안에서 “로컬” 테스트

웹서버/스케줄러 파드 중 하나에 들어가서 같은 조건으로 붙여봅니다.

```
kubectl exec -it deploy/airflow-webserver -n <ns> -- bash
```

```
# 2-1. 포트 확인 (9030)
```

```
nc -vz starrocks-fe.<ns>.svc.cluster.local 9030
```

```
# 2-2. mysql 클라이언트로 붙기 (있다면)
```

```
mysql -h starrocks-fe.<ns>.svc.cluster.local -P 9030 -u pdpdev -ppdpdev pdpdev
-e "select 1;"
```

```
# 2-3. PyMySQL로 붙기 (드라이버 검증)
```

```
python - <<'PY'
import pymysql, sys
try:
    conn = pymysql.connect(host="starrocks-fe.<ns>.svc.cluster.local",
                           port=9030,
                           user="pdpdev", password="pdpdev", database="pdpdev",
                           charset="utf8mb4", connect_timeout=10,
                           read_timeout=30)
    with conn.cursor() as c:
        c.execute("select 1")
        print("OK:", c.fetchone())
    conn.close()
except Exception as e:
    print("FAIL:", e)
```

```
sys.exit(1)
PY

# 2-4. Airflow 자체 테스트 (2.5+ 지원)
airflow connections test starrocks_conn_id
```

여기서 붙고 Airflow UI만 실패한다? → Conn 설정/드라이버 매칭 문제일 확률이 큼니다.  
여기서도 실패한다? → 네트워크/DNS/Netpol/권한 쪽을 의심하세요.

### 3) 네트워크/서비스/DNS 체크

- ``kubectl -n <starrocks-ns> get svc`` 로 FE 서비스가 9030/TCP를 노출하는지.
- Airflow 네임스페이스에 **NetworkPolicy** 가 있어 egress가 막히지 않았는지:

```
kubectl -n <airflow-ns> get netpol
```

막혀있다면 FE 서비스로의 9030 허용 규칙을 추가.

### 4) 계정 권한/호스트 매칭

StarRocks에서:

```
SHOW GRANTS FOR 'pdpdev'@'%';
```

- Airflow 파드의 소스 IP/호스트가 허용 범위(`%` 또는 해당 CIDR)인지 확인.
- 필요 시:

```
CREATE USER 'pdpdev'@ '%' IDENTIFIED BY 'pdpdev';
GRANT ALL ON pdpdev.* TO 'pdpdev'@ '%';
FLUSH PRIVILEGES;
```

### 5) 흔한 에러 → 빠른 처방

- **Timeout / can't connect**: 잘못된 호스트명, 9030 미개방, Netpol 차단, 서비스/엔드포인트 없음.
- **Access denied**: 계정/비밀번호 틀림, 또는 ``'user'@'host'`` 매칭 불일치.

- **ImportError: MySQLdb:** `mysqlclient` 미설치 → URI를 `mysql+pymysql://...` 로 교체하고 `PyMySQL` 설치.
- **SSL 관련 오류:** FE가 TLS 안 쓰는데 Airflow Extra에 SSL 옵션이 들어간 경우 → Extra 비우기.
- **Unknown system variable** 계열: 드물지만 드라이버가 MySQL 전용 옵션을 던질 때 → Extra 옵션 제거/단순화.

## 6) DAG 코드에서의 사용 예 (SQL provider 공용)

`SqlOperator` 를 쓰는 경우:

```
from airflow.providers.common.sql.operators.sql import SQLExecuteQueryOperator

SQLExecuteQueryOperator(
    task_id="poke_starrocks",
    conn_id="starrocks_conn_id",
    sql="SELECT 1",
)
```

또는 혹 직접:

```
from airflow.providers.mysql.hooks.mysql import MySQLHook
hook = MySQLHook(mysql_conn_id="starrocks_conn_id")
records = hook.get_records("SELECT 1")
```

## 7) 최종 체크리스트 (짧게)

- ☐ FE 서비스 9030 reachable (파드 안에서 `nc -vz host 9030`)
- ☐ Airflow Conn: `mysql+pymysql://...:9030/...`
- ☐ `apache-airflow-providers-mysql` + `PyMySQL` 설치
- ☐ `SHOW GRANTS` 로 권한/호스트 확인
- ☐ Netpol/네임스페이스/서비스 이름 오기 체크

원하시면 현재 Airflow 이미지/values.yaml 스니펫 알려주세요. 드라이버/패키지 설치 위치나 커맨드 (예: `pip install PyMySQL apache-airflow-providers-mysql==<버전>`)까지 맞춰 드릴게요.



connection test dag 에서 로그 메시지에 "Access denied" 라고 나오더라고요.



24s 동안 생각함

오케이, "Access denied"면 네트워크는 붙었고 인증/권한 매칭에서 막힌 겁니다. 보통은 `user@host` 매칭이 안 맞아요. Airflow 파드의 **Pod IP 대역(10.x/172.16-31.x 등)** 이 `GRANT` 대상에 안 들어가 있으면 바로 이 에러가 납니다.

아래 순서대로 바로 조치해 보세요.

## 1) 에러 메시지로 host 확인

Airflow 로그에 보통 이렇게 찍혀요:

```
Access denied for user 'pdpdev'@'10.244.7.53' (using password: YES)
```

따옴표 뒤 IP/호스트(`10.244.7.53`)가 **StarRocks**가 인식한 클라이언트 **host**입니다.

## 2) StarRocks에서 현재 권한 상태 보기

FE에 mysql CLI로 접속해서:

```
-- 어떤 엔트리가 있는지
SHOW GRANTS FOR 'pdpdev'@'%';
SHOW GRANTS FOR 'pdpdev'@'10.%';
SHOW GRANTS FOR 'pdpdev'@'172.16.%';
-- (필요하면) 전체 사용자 호스트 목록 훑기
-- StarRocks는 MySQL 호환이라 아래도 대체로 통합니다
SHOW ALL GRANTS;
```

만약 위가 비어있거나, `localhost` 같은 것만 있고 Pod IP 대역이 없다면 3)로 가세요.

## 3) 호스트 패턴을 Kubernetes 대역으로 열기 (가장 빠른 해결책)

MySQL 계열은 CIDR이 아니라 와일드카드를 씁니다. K8s에서 흔한 대역을 열어주세요.

```
-- 10.x 대역
CREATE USER IF NOT EXISTS 'pdpdev'@'10.%' IDENTIFIED BY 'pdpdev';
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'10.%';
```

```
-- 172.16~31.x 대역(클러스터에 따라 사용)
CREATE USER IF NOT EXISTS 'pdpdev'@'172.16.%' IDENTIFIED BY 'pdpdev';
CREATE USER IF NOT EXISTS 'pdpdev'@'172.17.%' IDENTIFIED BY 'pdpdev';
CREATE USER IF NOT EXISTS 'pdpdev'@'172.18.%' IDENTIFIED BY 'pdpdev';
CREATE USER IF NOT EXISTS 'pdpdev'@'172.19.%' IDENTIFIED BY 'pdpdev';
CREATE USER IF NOT EXISTS 'pdpdev'@'172.20.2%' IDENTIFIED BY 'pdpdev'; --
172.20~29
CREATE USER IF NOT EXISTS 'pdpdev'@'172.30.2%' IDENTIFIED BY 'pdpdev'; --
172.30~31
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'172.%';

-- 가장 단순하게는 임시로 전체 허용(문제 해결 확인용)
CREATE USER IF NOT EXISTS 'pdpdev'@'%' IDENTIFIED BY 'pdpdev';
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'%';

FLUSH PRIVILEGES;
```

운영에선 너무 넓은 `%` 대신 실제 클러스터 **Pod IP** 대역만 남기고 정리하는 걸 추천합니다.

## 4) 충돌/우선순위 점검

MySQL 매칭은 “더 구체적인 host가 우선”입니다.

예: ``'pdpdev'@'localhost'`와 ``'pdpdev'@'%'`가 동시에 있고 비번이 다르면 혼동이 납니다. 아래로 정리하세요.

```
-- 실제 쓰지 않는 엔트리 제거 예시
DROP USER IF EXISTS 'pdpdev'@'localhost';
```

## 5) 스키마 권한 누락도 혼함

유저는 맞는데 DB 권한이 없어서 “Access denied”가 날 수도 있어요.

```
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'%';
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'10.%';
GRANT ALL ON `pdpdev`.* TO 'pdpdev'@'172.%';
FLUSH PRIVILEGES;
```

## 6) Airflow 쪽 두 가지 재확인

- **Conn Type/URI:** `mysql+pymysql://pdpdev:pdpdev@starrocks-fe.<ns>.svc.cluster.local:9030/pdpdev`

- 비번에 특수문자 있으면 URI 인코딩 필요합니다. 애매하면 UI 필드로 입력하세요.

- **연결 테스트:**

```
kubectl exec -it deploy/airflow-webserver -n <ns> -- bash
python - <<'PY'
```

```
import pymysql;
p = dict(host="starrocks-fe.<ns>.svc.cluster.local", port=9030, user="pdpdev",
password="pdpdev", database="pdpdev", charset="utf8mb4")
conn = pymysql.connect(**p); conn.close(); print("OK")
PY
airflow connections test <your_conn_id>
```

# 7) 아직도 Access denied면

- **\*\*로그로 정답 보기\*\***: FE의 `fe.audit.log`/`fe.warn.log`에 실패 원인이 남습니다.  
``bash

# 예시(리소스 이름은 환경에 맞게)

```
kubectl logs sts/starrocks-fe -c fe -n <starrocks-ns> | grep -i "access denied"
```

- 에러 메시지에 찍힌 `@'호스트'` 그대로를 GRANT 대상에 추가하세요:

```
CREATE USER 'pdpdev'@'10.244.%' IDENTIFIED BY 'pdpdev';
GRANT ALL ON `pdpdev`. * TO 'pdpdev'@'10.244.%;
FLUSH PRIVILEGES;
```

필요하시면, Airflow `values.yaml`에 들어갈 `env`/`requirements` 스니펫이랑, StarRocks에 적용할 **최소-권한 GRANT 세트**도 맞춰 드릴게요. 로그에 찍힌 `@'` 뒷부분 한 줄만 복사해 주시면 바로 깔끔하게 정리해 드립니다.