



starrocks에서 fe쪽에 다음과 같은 오류가 1분정도 간격으로 계속 올라와.

Thrift Error occurred during processing of message. Missing version in readMessageBegin, old client?

증상이 딱 "Thrift 포트(=FE 내부 RPC)에 HTTP/다른 프로토콜이 주기적으로 찍혀서, FE가 '버전 비어있다 / old client?'라고 성을 내는" 패턴이에요.

1분 간격이면 거의 100% 헬스체크나 모니터링 프로브가 잘못된 포트를 두드리는 겁니다.

## 왜 뜨나요?

- 이 에러 문구는 **Thrift 서버가 TBinary/TFramed 메시지를 기대하는데, 상대가 HTTP/MySQL/빈 TCP 핸드셰이크** 같은 걸 보내면 나는 전형적인 메시지입니다.
- StarRocks FE에서
  - **HTTP/REST & metrics: 8030**
  - **MySQL 프로토콜(클라이언트 접속): 9030**
  - **Thrift/RPC(내부용): 9020**  
입니다. 9020은 사람/HTTP가 두드리면 안 되는 포트예요.

## 제일 흔한 원인

- Kubernetes **liveness/readiness/startup probe**가 9020을 **httpGet**으로 찍는 경우
- Prometheus/BlackboxExporter 같은 모니터링이 9020을 HTTP로 긁는 경우
- (가끔) MySQL 클라이언트/툴이 **9030 대신 9020**으로 잘못 접속

## 바로 확인할 것 (빠른 순서)

### 1. FE Pod의 프로브

```
bash

kubectl -n <ns> describe pod <fe-pod>
# 또는 디플로이먼트/스테이트풀셋에서 probe 정의 확인
kubectl -n <ns> get deploy,statefulset <fe-name> -o yaml
```

- **httpGet/tcpSocket** 대상 포트가 **반드시 8030(HTTP)** 또는 **tcpSocket: 9030(원한다면)**이어야 합니다.
- 9020을 찍고 있으면 바꿔야 합니다.

### 2. ServiceMonitor/PodMonitor/Ingress/LB 헬스체크

```
bash

kubectl -n <ns> get servicemonitor,podmonitor -o wide | grep -i starrocks
```

```
kubectl -n <ns> get svc <fe-svc> -o yaml
```

- HTTP 스크랩 타킷은 **8030**(/metrics, /api/health)로.

### 3. 누가 두드리는지 잡기 (선택)

```
bash
```

```
# 30~60초 정도 흘러보면 소스IP가 찍힙니다
kubectl -n <ns> exec -it <fe-pod> -- sh -c 'apk add tcpdump 2>/dev/null || true; tcpdump -nn -i any port 9020 -vv -c 5'
```

- 나온 IP/Port가 Prometheus나 kubelet, LB 헬스체커면 범인 확정.

## 올바른 설정 예시

### K8s Probe (HTTP는 8030):

```
yaml

livenessProbe:
  httpGet:
    path: /api/health
    port: 8030
  initialDelaySeconds: 20
  periodSeconds: 10

readinessProbe:
  httpGet:
    path: /api/health
    port: 8030
  initialDelaySeconds: 10
  periodSeconds: 10
```

### Prometheus 스크랩 (metrics):

- 타킷: **http://<fe-service>:8030/metrics**

### MySQL 클라이언트(툴, DBeaver 등):

- 호스트: FE 서비스/주소
- 포트: **9030**
- 프로토콜: MySQL

## 혹시 다른 가능성?

- FE/CN 버전 불일치로 Thrift 핸드셰이크가 깨질 수도 있지만(정말 드뭅니다), **정확히 1분 주기**라면 거의 헬스체크/스크레이퍼 쪽입니다.
- Helm/ArgoCD 값에서 포트 매핑이 헛갈려서 FE Service가 **9020**으로 "HTTP"로 노출돼 있는 케이스도 봤습니다. Service/Ingress 포트 정의도 같이 훑어보세요.

원인만 바로잡으면 로그는 조용해집니다.

필요하시면 지금 쓰시는 Helm values/ArgoCD Application YAML에서 **프로브/ServiceMonitor** 부분만 떼어 보여

주세요. 거기서 정확히 어디가 9020을 만지는지 딱 집어드릴게요.



starrocks 를 mysql cli 로 접속해서 set catalog hive; 하고 나서 테이블 목록에 보는데, 몇개가 안보여. 이때 할 수 있는게 뭐가 있지?

가능한 원인은 몇 가지로 좁혀집니다. 순서대로 점검·조치하세요. (환경: FE/CN 3.3.5, Hive 카탈로그, MinIO)

## 1) “세션/권한/컨텍스트”부터 확인

- 지금 세션이 정말 그 카탈로그·DB를 보고 있는지:

```
sql

SHOW CATALOGS;
SET CATALOG hive;          -- 이미 하셨다면 생략
SHOW DATABASES;
USE <db>;
SHOW TABLES;             -- 또는 SHOW TABLES FROM hive.<db>;
```

- 외부 카탈로그는 **USAGE 권한**이 있어야 테이블 목록이 보입니다. 필요한 경우:

```
sql

GRANT USAGE ON CATALOG hive TO 'user'@'%';
```

(외부 소스의 테이블을 보려면 USAGE가 필요함. [docs.starrocks.io](https://docs.starrocks.io))

## 2) StarRocks 쪽 메타캐시 갱신 (가장 효과적)

Hive/Glue 메타데이터는 FE에 캐시됩니다. 바깥에서 스키마/파티션/테이블이 바뀌었으면 캐시를 새로고침하세요.

- 개별 테이블 강제 갱신

```
sql

-- 카탈로그/DB를 함께 적어도 됩니다
REFRESH EXTERNAL TABLE <table_name>;
-- 특정 파티션만
REFRESH EXTERNAL TABLE <table_name> PARTITION ('dt=2025-09-09');
```

(REFRESH EXTERNAL TABLE은 Hive/Hudi 테이블 캐시를 갱신. ALTER 권한 필요. [docs.starrocks.io+1](https://docs.starrocks.io+1))

- 자동 갱신 주기 확인/튜닝(선택)

Hive 카탈로그 문서 기준 v3.3에서는 기본 메타캐시 갱신 주기가 60초입니다. 잦은 변경이 있다면 카탈로그 생성 시 `MetadataUpdateParams`를 조정하거나 FE 동적 파라미터를 튜닝할 수 있습니다. [docs.starrocks.io](https://docs.starrocks.io)

## 3) “왜 몇 개만 안 보이느냐”의 전형적 원인들

- **신규/변경 테이블의 캐시 미반영**

→ 방금 위의 **REFRESH EXTERNAL TABLE**로 해결되는 케이스가 대부분. (수동 갱신 필요 상황이 문서에 명시되어 있음) [docs.starrocks.io](https://docs.starrocks.io)

- **지원 포맷/타입 이슈**

StarRocks가 지원하는 Hive 포맷은 Parquet/ORC/Textfile/Avro/RCFile/SequenceFile 입니다(일부 타입 제한 있음). ACID 트랜잭션 테이블 등 특수 포맷/Serde는 조회 불가하거나 에러가 납니다. 우선 형식을 확인해 보세요. [docs.starrocks.io](https://docs.starrocks.io)

- **파티션명/리스트 캐시**

카탈로그를 만들 때 **enable\_cache\_list\_names=true**를 썼다면, 새 파티션/오브젝트 인지가 늦을 수 있어 수동 REFRESH가 필요합니다(문서에 언급). [docs.starrocks.io](https://docs.starrocks.io)

- **Glue/HMS 동기화 지연/버그**

Glue 사용 시 테이블 변경 반영이 늦는 이슈 리포트도 있습니다. 강제 REFRESH나 상위 레이어(예: MV) 갱신으로 반영되는지 체크하세요. [GitHub](https://github.com)

- **권한/필터링**

HMS의 권한/필터 혹(사내 Ranger/Sentry 유사)으로 특정 테이블이 숨겨질 수 있습니다. HMS 측 유저/정책을 확인하세요.

- **경로/스토리지 자격 증명**

MinIO(S3 호환) 사용 시 카탈로그의 **StorageCredentialParams**가 정확한지, 그리고 해당 버킷/프리픽스에 목록 권한이 있는지 확인. (S3-compatible 통합 가이드) [docs.starrocks.io](https://docs.starrocks.io)

## 4) 빠른 진단 SQL 세트

문제가 되는 “안 보이는” 테이블 이름을 알고 있다면, 목록엔 없더라도 직접 Fully-qualified로 찔러보세요. 에러 메시지가 힌트를 줍니다.

```
sql

-- 목록에 없는데 존재한다고 의심되는 경우
DESC hive.<db>.<table>;
SELECT * FROM hive.<db>.<table> LIMIT 1;

-- 정보 스키마에서 확인
SELECT table_catalog, table_schema, table_name
FROM information_schema.tables
WHERE table_catalog='hive' AND table_schema='<db>';
```

- DESC/SELECT가 되면 **목록 캐시 이슈** 가능성이 큼 → **REFRESH EXTERNAL TABLE** 권장.
- “포맷/타입 미지원” 류의 에러면 포맷/Serde를 확인.

## 5) 운영 팁 (지속적 반영)

- 배치/스트리밍 작업이 Hive에 테이블/파티션을 자주 추가한다면:
  - (안정) 주기적 REFRESH 스크립트를 Airflow 등에서 실행
  - (튜닝) 카탈로그의 **metastore\_cache\_refresh\_interval\_sec**, **remote\_file\_cache\_refresh\_interval\_sec**을 환경에 맞게 단축 (기본값과 동작 설명은 문서 참고) [docs.starrocks.io](https://docs.starrocks.io)

원하시면 **카탈로그 생성 SQL**(특히 MinIO 자격증명/메타캐시 파라미터)이나, 문제가 되는 **구체 테이블명 1~2개**를 알려주세요. 그 케이스로 바로 재현·수정 쿼리 묶음을 만들어 드릴게요.



테이블 목록을 갱신하고 싶은건데. 개별 테이블 refresh하는게 아니고 전체 테이블 목록을 새롭게 로딩하고 싶어. 지금 몇개 테이블이 누락되고 있거든.

좋아요, “전체 목록을 한 번에 새로고침”이 목적이시죠. 결론부터 말하면 **StarRocks 3.3 기준으로 ‘카탈로그/DB 단위 전체 리프레시’ 전용 SQL은 없습니다.** 공식 제공은 **REFRESH EXTERNAL TABLE**(테이블 단위)뿐이에요. 대신 아래 4가지 현실적인 전략이 있어요.

## A. 같은 DB의 모든 테이블을 일괄 REFRESH (실전용)

1. 먼저 목록을 긁어오고
2. 그 목록으로 **REFRESH EXTERNAL TABLE**을 루프 실행합니다.

```
sql

-- 1) 대상 DB의 테이블 목록
SHOW TABLES FROM hive.<db>;

-- 2) 실행문 자동 생성 (복사해서 실행)
SELECT CONCAT('REFRESH EXTERNAL TABLE `', table_name, ``;') AS stmt
FROM information_schema.tables
WHERE table_catalog='hive' AND table_schema='<db>';
```

생성된 **stmt**들을 그대로 실행하면, 해당 DB의 **모든 테이블 메타 캐시**가 갱신됩니다. (**REFRESH EXTERNAL TABLE**은 외부 카탈로그 캐시를 갱신하는 공식 방법입니다. [StarRocks 문서+1](#))

운영에서는 이 쿼리들을 작은 배치(예: 50~100개씩)로 나눠 실행하는 걸 권합니다.

## B. “아예 싹 갈아끼우기” - 카탈로그 드롭/재생성

외부 카탈로그는 HMS/Glue 포인터이므로 **데이터 손실 없이** 드롭/재생성으로 전체 목록을 재로딩할 수 있어요. (권한/GRANT은 다시 걸어야 할 수 있음)

```
sql

-- 주의: 카탈로그명을 정확히 확인
DROP CATALOG hive;

CREATE EXTERNAL CATALOG hive
PROPERTIES (
  "type"="hive",
  "hive.metastore.uris"="thrift://<hms-host>:9083"
  -- (S3/MinIO 쓰면 관련 자격증명/옵션 추가)
);
```

이 방법은 가장 “확실하게 목록을 초기화”합니다. 다만 카탈로그 레벨 권한을 다시 부여해야 할 수 있어요. (외부 카탈로그/캐시 동작과 REFRESH의 역할은 문서에 명시. [StarRocks 문서+1](#))

## C. 캐시 정책 조정으로 “자동 인지”를 더 자주

Hive 카탈로그의 메타 캐시·파일 캐시 정책을 더 공격적으로 바꾸면 **수동 갱신 빈도**가 줄어듭니다.

카탈로그 생성 시/이후에 다음 속성을 검토하세요:

- **metastore\_cache\_ttl\_sec** : Hive 메타 캐시 TTL(기본 86400s)
- **remote\_file\_cache\_refresh\_interval\_sec** : 파일 목록 캐시 갱신 주기(기본 60s)
- **enable\_cache\_list\_names** : 파티션 이름 캐시 사용 여부

변경 의도: “새 테이블/파티션 인지가 늦다 → TTL 단축 또는 갱신 주기 단축”

단, 너무 낮추면 HMS/스토리지 부하가 커집니다. (관련 옵션과 주기 동작은 Hive 카탈로그 문서 참조)

[StarRocks 문서](#)

또한 FE에는 **백그라운드 메타 갱신 만료 제어** 파라미터가 있습니다. 오랫동안 접근이 없던 카탈로그는 자동 갱신을 멈출 수 있는데, 주기를 조정할 수 있어요. [StarRocks 문서](#)

## D. 최후수단: FE 재시작(캐시 강제 무효화)

서비스 영향이 가능하므로 비권장이나, 급할 때는 **FE 재시작으로 캐시를 비우고** 목록을 다시 로딩하게 만들 수 있습니다. (메타 복구/재기동 시 캐시가 비워짐. 운영 가이드는 별도 문서) [StarRocks 문서](#)

## 체크리스트 — “왜 몇 개만 안 보이느냐”

- **HMS/Glue에 테이블이 실제로 등록되어 있는지** (파일만 있고 테이블 생성/등록이 안 된 케이스) → 먼저 HMS 쪽을 확인
- **권한**: 외부 카탈로그에 **USAGE** 권한 없으면 **SHOW TABLES**가 비어 보일 수 있어요. (외부 소스 조회에는 **USAGE** 권한 필요) [StarRocks 문서](#)
- **캐시 타이밍**: 새로 만든 직후면 A/B/C 중 하나로 반영 촉진
- **특수 포맷/ACID 테이블**: 비표준/미지원 포맷이면 목록/쿼리에서 누락·오류가 납니다(별도 확인 필요).

## 현장용 스니펫 (셸에서 일괄 REFRESH)

bash

```
mysql -h <fe-host> -P9030 -uroot -p'<pw>' -Nse \
"SELECT CONCAT('REFRESH EXTERNAL TABLE `', table_name, '`;')
FROM information_schema.tables
WHERE table_catalog='hive' AND table_schema='<db>' " \
| mysql -h <fe-host> -P9030 -uroot -p'<pw>'
```

필요하시면 지금 쓰시는 **카탈로그 생성 SQL**(MinIO 자격/옵션 포함) 보여주세요.  
TTL/주기 값 튜닝과, “스크립트로 DB 단위 일괄 REFRESH”를 환경에 맞게 딱 맞춰드릴게요.



starrocks 에서 테이블을 조회하는데 kudu 타입이 아닌데 KuduMetadata.getTable 에서 찾으려고 하지?

네, 그 증상은 **해당 테이블의 HMS 메타데이터가 'Kudu 테이블'로 인식될 때** FE가 내부에서 `KuduMetadata.getTable()` 경로로 타는 전형적인 패턴입니다. “Kudu 타입이 아닌데 왜 Kudu로 보냐?”의 대부분 원인은 **Hive Metastore 쪽에 Kudu 관련 표식이 남아** 있어서예요.

## 왜 Kudu로 인식될까?

StarRocks는 외부 카탈로그(Hive/Unified 등)에서 테이블을 볼 때, `storage_handler`나 **테이블 프로퍼티**를 보고 포맷을 판단합니다. 만약 HMS에

- `storage_handler = org.apache.hadoop.hive.kudu.KuduStorageHandler` 이거나
- TBLPROPERTIES에 `kudu.table`, `kudu.master_addresses` 같은 키가 있으면,  
해당 테이블을 Kudu로 간주하고 Kudu 메타 경로로 조회합니다. [hive.apache.orgkudu.apache.org](http://hive.apache.orgkudu.apache.org)

또한 StarRocks 3.3에서는 **Kudu 카탈로그/Unified 카탈로그가 Kudu를 지원**하므로, 이런 표식이 있으면 자연스럽게 Kudu 플러그인을 타게 됩니다. [StarRocks 문서+2](#)

## 무엇을 확인/조치하면 좋을까?

### 1. HMS에서 해당 테이블의 메타를 직접 확인

- Hive/Beeline 등에서:

```
sql

USE <db>;
DESCRIBE FORMATTED <table>;
SHOW TBLPROPERTIES <table>;
```

- 여기서 `KuduStorageHandler` 또는 `kudu.*` 프로퍼티가 보이면 Kudu로 등록/남은 상태입니다. (Impala가 만든 Kudu 테이블은 HMS에 Kudu 메타를 심습니다.) [kudu.apache.orgdocs.cloudera.com](http://kudu.apache.orgdocs.cloudera.com)

### 2. 정말 Kudu 테이블이 아니라면(메타가 잘못된 경우)

- 안전한 방법은 **테이블을 올바른 포맷으로 재생성**하는 겁니다.  
예: 원래 Parquet/ORC라면 적절한 SerDe/INPUTFORMAT으로 새 **EXTERNAL TABLE**을 만들고, 기존 메타를 드롭.
- 실수로 남은 Kudu 프로퍼티만 제거하려면 HMS를 직접 편집해야 하는데, 이는 환경마다 제약이 있어 **재생성**이 보통 더 깔끔합니다. (Cloudera 계열 환경에선 Kudu 메타가 타 도구에서 읽기 난해한 케이스도 있습니다.) [docs.cloudera.com](http://docs.cloudera.com)

### 3. 진짜 Kudu 테이블이라면

- StarRocks에서 Kudu를 제대로 조회하려면 **\*\*Kudu 카탈로그(또는 Unified 카탈로그 내 Kudu 설정)\*\***가 올바르게 구성돼야 합니다. 보통 `kudu.master_addresses` 등 연결 정보를 카탈로그에 명시합니다.  
[StarRocks 문서](#)
- 설정이 없거나 접근이 막혀 있으면 Kudu 경로로 진입한 뒤 에러가 납니다. 카탈로그 생성 문서를 따라 파라미터를 채워주세요. [StarRocks 문서](#)

#### 4. StarRocks 쪽에서 포맷 인식 우회(임시)

- **Unified 카탈로그**에서 혼합 포맷 때문에 오인식이 난다면, 문제 테이블만 **Hive 카탈로그**로 **별도 연결**해 테스트하거나, 테이블을 뷰로 감싸는 등 임시 우회를 해볼 수 있습니다. (근본 해결은 HMS 메타 정정)  
[StarRocks 문서+1](#)

## 빠른 체크리스트

- 해당 테이블의 HMS에 **Kudu StorageHandler/프로퍼티가 존재하는가?** → 존재하면 Kudu 경로 탐니다.  
[hive.apache.org](https://hive.apache.org)
- 진짜 Kudu면 **StarRocks Kudu(또는 Unified) 카탈로그 설정**이 되어 있는가? → `kudu.master_addresses` 등 필수값 점검. [StarRocks 문서](#)
- Kudu가 아니면 **테이블 재생성**으로 Kudu 표식을 제거. (남은 Kudu 표식은 다른 도구에서도 잡음을 유발합니다.)  
[docs.cloudera.com](https://docs.cloudera.com)

원하시면 문제가 되는 테이블의 **DESCRIBE FORMATTED/SHOW TBLPROPERTIES** 결과를 붙여 주세요.

그걸 기준으로 “Kudu로 오인식되는 필드”를 콕 집어서, **재생성 DDL** 또는 **카탈로그 설정 예시**까지 바로 정리해 드리겠습니다.