

Conversación con Gemini

Exported: 15/6/2026 9:07:58

Jordi H.

Has dicho quiero hacer un backup de mi base de datos StarRocks 3.4. Voy a usar la sentencia backup y para ello he instalado un servidor de s3 en local (minio). He creado un bucket en minio y estoy intentando hacer un repository tal y como se explica en https://docs.starrocks.io/docs/3.4/administration/management/Backup_and_restore/#create-a-repository el resultado es con error. exactamente dice `mysql> create repository test_copies with broker on location "s3://backup-starrocks" properties ("aws.s3.access_key" = "admin", "aws.s3.secret_key"= "MinioPLEX.00", "aws.s3.endpoint" = "http://minio:9000");` ERROR 5064 (42000): Failed to create repository: failed to list remote path: s3://backup-starrocks/_starrocks_repository_test_copies/_repo_info. msg: Fail to get file status: getFileStatus on s3://backup-starrocks/_starrocks_repository_test_copies/_repo_info: software.amazon.awssdk.core.exception.SdkClientException: Received an UnknownHostException when attempting to interact with a service. See cause for the exact endpoint that is failing to resolve. If this is happening on an endpoint that previously worked,

Gemini

El error `UnknownHostException` indica un problema de resolución de nombres (DNS). StarRocks, al intentar comunicarse con el endpoint `http://minio:9000`, no logra traducir el nombre "minio" a una dirección IP. Como estás trabajando en un entorno local (posiblemente con Docker o servidores físicos distintos), aquí tienes los pasos para resolverlo priorizando la configuración de red y las propiedades de

1. Resolución del Host

Si estás ejecutando StarRocks en **Docker**, asegúrate de que ambos contenedores estén en la misma red. Si están en máquinas distintas o el despliegue es "bare metal", el servidor de StarRocks no conoce la existencia del nombre minio

- Opción rápida: Sustituye minio por la IP local del servidor donde corre MinIO. Ejemplo: `"aws.s3.endpoint" = "[http://192.168.1.50:9000](http://192.168.1.50:9000)`
- Opción recomendada (Software Libre/Linux): Añade la entrada en el archivo `/etc/hosts` de todos los nodos de StarRocks (FE y BE): Bas

CODE

```
# En el servidor de StarRocks
echo "127.0.0.1 minio" >> /etc/hosts # Si MinIO está en la misma máquina
```

2. Configuración del Repositorio

Para MinIO, que no es el S3 oficial de AWS, es fundamental forzar el uso del estilo de acceso por ruta (path style access) y asegurar que la región esté definida (aunque sea un valor ficticio), ya que el SDK de Java que utiliza StarRocks lo requiere. Prueba a recrear el repositorio con estas propiedades adicionales: S

CODE

```
CREATE REPOSITORY test_copies
WITH BROKER
ON LOCATION "s3://backup-starrocks"
PROPERTIES (
  "aws.s3.access_key" = "admin",
  "aws.s3.secret_key" = "Minioplex.00",
  "aws.s3.endpoint" = "http://[IP_DE_MINIO]:9000",
  "aws.s3.region" = "us-east-1",
  "aws.s3.enable_path_style_access" = "true",
  "aws.s3.use_instance_profile" = "false"
);
```

3. Verificaciones Críticas

- Conectividad desde el BE: StarRocks realiza las operaciones de backup desde los nodos Backend (BE). Entra en uno de tus nodos BE y lanza un curl:Bas

CODE

```
curl http://minio:9000
```

Si el curl falla con "Could not resolve host", el problema es puramente de red/DNS en el sistema operativo.

- Bucket existente: Asegúrate de que el bucket backup-starrocks ya esté creado en la consola de MinIO antes de lanzar el comando
- Uso de Broker: En la versión 3.4, si estás usando S3 directamente, a menudo no necesitas especificar un BROKER externo a menos que lo hayas configurado específicamente para sistemas de archivos legacy . Puedes intentar omitir WITH BROKER si el soporte nativo de S3 está activ

Por qué ocurre esto

StarRocks delega la escritura a sus nodos de computación. Si el Frontend (FE) acepta el comando pero el Backend no puede "ver" el nombre de host minio, el SDK de AWS lanza esa excepción de cliente al intentar listar la ruta para inicializar el repositorio. ¿Estás corriendo este laboratorio sobre Docker Compose o sobre una infraestructura de nodos dedicada

Jordi H.

Has dicho he probado con la ip pública y el resultado es el mismo: mysql> create repository test_copies with broker on location "s3://backup-starrocks/backup" properties ("aws.s3.access_key" = "admin" , "aws.s3.secret_key"= "Minioplex.00", "aws.s3.endpoint" = "http://minio:9000"); ERROR 5064 (42000): Failed to create repository: failed to list remote path: s3://backup-starrocks/backup/___starrocks_repository_test_copies/___repo_info. msg: Fail to get file status: getFileStatus on s3://backup-starrocks/backup/___starrocks_repository_test_copies/___repo_info: software.amazon.awssdk.core.exception.SdkClientException: Received an UnknownHostException when attempting to interact with a service. See cause for the exact endpoint that is failing to resolve. If this is happening on an endpoint that previously worked pero ya veo que tengo un problema de conectividad al exterior de los BE. dime como puedo comprobar

Si el error persiste incluso tras intentar cambiar la configuración, es muy probable que los nodos **Backend (BE)** de StarRocks estén aislados o tengan una configuración de red que les impide llegar al endpoint de MinIO. Para un arquitecto de sistemas, la clave aquí es verificar la conectividad **capa por capa** desde la perspectiva de los procesos que realmente ejecutan la carga de datos

1. Identifica dónde reside el problema (BE vs FE)

El comando lo lanzas en el **FE** (Frontend), pero la validación del repositorio y la transferencia de datos la realizan los **BE**. Accede por SSH (o docker exec) a uno de tus nodos **Backend** y ejecuta estas pruebas

A. Prueba de resolución y ruta (Capa 3 y 4)

Verifica si el BE puede siquiera "ver" la IP y el puerto de MinIO: Bash

CODE

```
# Prueba el puerto específico
nc -zv [IP_DE_MINIO] 9000
```

Si esto falla, tienes un firewall (iptables/ufw) o una regla de red (Security Group si estás en cloud) bloqueando la salida

B. Prueba de protocolo (Capa 7)

Si el puerto está abierto, comprueba si el servicio HTTP responde correctamente: Bash

CODE

```
curl -I http://[IP_DE_MINIO]:9000/minio/health/live
```

Si obtienes un 200 OK, la conectividad básica existe. Si obtienes un "Connection refused" o "No route to host", el problema es de ruteo

2. El problema del "Invisible DNS"

En el error que compartes, sigues usando "aws.s3.endpoint" = "http://minio:9000". Aunque hayas intentado con la IP en otras pruebas, si el comando que falla mantiene el nombre minio, el SDK de Java intentará resolverlo y fallará. **Puntos ciegos habituales**

1. Entornos de Contenedores: Si StarRocks está en una red Docker y MinIO en otra (o en el host), el nombre minio no se resolverá a menos que uses --add-host o una red común
2. Configuración del BE: StarRocks cachea ciertas configuraciones. Asegúrate de que el archivo be.conf no tenga restricciones de red en el parámetro priority_networks

3. Ajuste de la sentencia (Eliminando ambigüedad)

Como buena práctica de arquitectura en homelabs y entornos controlados, para evitar que el SDK de AWS intente buscar servidores DNS globales, usa la **IP Privada** (no la pública, para evitar latencia y costes de salida) y añade el flag de **Path Style**. Prueba exactamente este formato en tu consola MySQL: S

CODE

```
CREATE REPOSITORY test_copies
WITH BROKER
ON LOCATION "s3://backup-starrocks/backup"
PROPERTIES (
  "aws.s3.access_key" = "admin",
  "aws.s3.secret_key" = "Minioplex.00",
  "aws.s3.endpoint" = "http://192.168.X.X:9000", -- USA LA IP PRIVADA AQUÍ
  "aws.s3.region" = "us-east-1",
  "aws.s3.enable_path_style_access" = "true"
);
```

4. Revisión de Logs (La verdad definitiva)

Si el error en MySQL es genérico, los logs del Backend te darán el "stack trace" completo de Java. Revisa el archivo en los nodos BE:Bas

CODE

```
tail -f /ruta/a/tu/starrocks/be/log/be.INFO | grep -i "repository"
```

Pregunta técnica de diagnóstico:

¿Tu servidor de MinIO y los nodos de StarRocks están en la misma máquina física (Homelab), o estás cruzando redes (ej. StarRocks en local y MinIO en un VPS)

Jordi H.

Has dicho ya ha funcionado. creo que es el enable path style

Gemini

Efectivamente, en arquitecturas que usan **MinIO** o cualquier implementación de S3 que no sea el AWS oficial, el parámetro `aws.s3.enable_path_style_access` es prácticamente obligatorio

¿Por qué ha sido la solución?

Por defecto, el SDK de AWS utiliza el **Virtual-Hosted Style**. Esto significa que intenta transformar tu petición en una URL de tipo

`[http://backup-starrocks.minio:9000](http://backup-starrocks.minio:9000)` Al habilitar **Path Style**, obligas al cliente a mantener el host intacto y poner el bucket en la ruta

`http://minio:9000/backup-starrocks` Sin ese parámetro, el sistema operativo de tus BE intentaba resolver el subdominio `backup-starrocks.minio`, lo cual fallaba (de ahí el `UnknownHostException`), ya que tu DNS local o el archivo `/etc/hosts` solo conocen la entrada para `mini`

Siguientes pasos para tu Backup

Ahora que el repositorio está conectado, ya puedes ejecutar la sentencia de backup propiamente dicha. Al ser un entorno de laboratorio/arquitectura, te sugiero monitorizar estos puntos

1. Ejecución del Backup:SQL

CODE

```
BACKUP SNAPSHOT test_copies.snapshot_nombre  
TO test_copies  
ON (tu_tabla_o_db)  
PROPERTIES ("type" = "full");
```

2. Monitorización del progreso:

Como estás en un entorno de analítica, te interesará ver cómo progresa la tarea en segundo plano:SQL

CODE

```
SHOW BACKUP;
```

3. Verificación en MinIO:

Verás que StarRocks crea una estructura de directorios con metadatos y archivos .dat. Es un sistema de archivos inmutable una vez escrito el snapshot

Recomendación de Arquitecto

Puesto que estás construyendo un **vector tecnológico** en tu homelab, si planeas automatizar esto con Python o scripts de mantenimiento, asegúrate de que el usuario de MinIO tenga políticas de lectura/escritura (readwrite) específicas para ese bucket, evitando usar siempre el usuario admin para mejorar la seguridad de la arquitectura. ¿Vas a realizar backups incrementales o solo snapshots puntuales para el laboratorio?

Jordi H.

Has dicho estoy probando el backup para, cuando esté bien consolidado y comprobado el proceso, cambiar a un repositorio s3 corporativo. otra cosa, no veo la manera de crear y gestionar usuarios desde la consola de minio (uso la community edition) debo usar el cliente con CLI

Gemini

Exacto, en la versión actual de la interfaz web de MinIO (el Console), la gestión de usuarios suele estar disponible solo si has configurado un sistema de identidades externo o si la versión desplegada tiene habilitada la sección de **Identity**. Si en tu panel lateral no aparece la opción "**Identity**" (que engloba Users, Groups y Policies), la forma más potente y estándar de hacerlo bajo el espíritu del software libre es mediante el cliente oficial de línea de comandos: **mc (MinIO Clie)**

1. Instalar y configurar el cliente `mc`

Como arquitecto de sistemas, verás que mc es una herramienta tipo "navaja suiza" que imita los comandos de UNIX (ls, cp, diff). Primero, configura el alias para tu servidor local: Bas

CODE

```
mc alias set mi-minio http://[IP_DE_MINIO]:9000 admin Minioplex.00
```

2. Gestión de Usuarios vía CLI

Una vez configurado el alias, puedes gestionar los accesos de forma granular:

- Crear un nuevo usuario: Bash

CODE

```
mc admin user add mi-minio nuevo_analista password123
```

- Crear una política personalizada (JSON):

En lugar de dar permisos totales, puedes crear un archivo politica_backup.json que solo permita escribir en el bucket de StarRocks.JSO

CODE

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:GetBucketLocation", "s3:ListBucket"],
      "Resource": ["arn:aws:s3:::backup-starrocks"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::backup-starrocks/*"]
    }
  ]
}
```

Y subirla al servidor: Bash

CODE

```
mc admin policy create mi-minio starrocks-policy politica_backup.json
```

- Asignar la política al usuario: Bash

CODE

```
mc admin policy attach mi-minio starrocks-policy --user nuevo_analista
```

Transición al S3 Corporativo

Hacer este ejercicio con mc y políticas granulares ahora es la mejor preparación para el **S3 corporativo**.

Cuando hagas el cambio

1. Arquitectura: La lógica de "Path Style" que acabas de descubrir podría no ser necesaria en AWS S3 (que usa Virtual Hosted), pero sí en implementaciones de S3 locales (como Ceph o Dell ECS)
2. IAM: Las políticas que diseñes en MinIO con mc son compatibles en un 99% con el formato JSON de AWS IAM, por lo que ya tendrás gran parte del trabajo de seguridad hecho
3. Endpoint: Solo tendrás que actualizar el REPOSITORY en StarRocks apuntando al nuevo endpoint y cambiando las credenciales

Un detalle técnico: ¿Tienes previsto que el backup sea hacia un bucket con **Object Lock** (WORM) para evitar borrados accidentales o ataques de ransomware, o será un almacenamiento estándar

Jordi H.

Has dicho me gusta la idea de object lock. ya me cuentas luego los detalles. ahora voy a probar a hacer un backup pequeño para validar el flujo. luego me planteo si hacer backup de toda la base de datos (es la patstat , 300 GB) en un solo comando o preparar alguna estrategia de hacerlo por trozos. finalmente, creo que no tie

ne sentido la incremental porque patstat es inmutable entre versiones (las versiones se cargan enteras cada seis meses)

Gemini

Es un planteamiento muy sólido. Siendo la **PATSTAT** (que es esencialmente un "snapshot" masivo de la oficina de patentes), tienes toda la razón: la copia incremental no aporta nada si el dataset es estático. Manejar **300 GB** es una cifra trivial, pero para StarRocks es un volumen cómodo si la infraestructura acompaña. Aquí tienes algunas consideraciones de arquitectura para esa validación que vas a realizar:

1. Estrategia: ¿Un solo comando o por trozos?

Con 300 GB, mi recomendación es intentar el backup de la **base de datos completa** en un solo comando primero. StarRocks está diseñado para paralelizar estas tareas entre todos los BE (Backend Nodes)

- Ventaja del comando único: Mantienes la consistencia atómica de todo el dataset sin gestionar scripts adicionales
- Puntos de fallo: Si un nodo BE cae o la red con MinIO es inestable, el backup fallará por completo.

Si decides segmentar, no lo hagas por "trozos" de archivos, sino por **tablas**. Puedes lanzar el backup de las tablas más críticas o pesadas (como `tls201_appln` o `tls209_appln_ipc`) de forma individual para validar tiempos de transferencia

2. El factor "Timeout"

Para un backup de 300 GB, es probable que los parámetros por defecto de StarRocks se queden cortos. Antes de lanzar el backup real, revisa estos parámetros en el FE:SQ

CODE

```
ADMIN SHOW FRONTEND CONFIG LIKE "backup_result_expire_low_priority_period";
```

Si el backup tarda más de unas horas, podrías necesitar ajustar el `snapshot_timeout` en las propiedades del comando `BACKUP` para evitar que el proceso se cancele por tiempo:SQ

CODE

```
BACKUP SNAPSHOT test_copies.full_patstat  
TO test_copies  
PROPERTIES ("timeout" = "36000"); -- Tiempo en segundos (ej. 10 horas)
```

3. Rendimiento en el Homelab

Al ser un entorno local, el cuello de botella suele ser la **escritura en disco del servidor MinIO** o la **red de 1Gbps** (si los nodos son físicos)

- 300 GB a través de una red de 1 Gbps tardarán, en condiciones ideales, unas ~40-50 minutos.
- Si usas discos mecánicos en el homelab para el bucket de MinIO, vigila la latencia de escritura.

Adelanto sobre el Object Lock (WORM)

Como mencionaste que te interesa, para cuando pases al S3 corporativo o para tu mc local: el **Object Lock** evitará que incluso un usuario con permisos de administrador pueda borrar o sobrescribir los archivos de backup durante un periodo de retención (ej. 30 días). Esto es la defensa definitiva contra errores de script o ataques que intenten borrar tus históricos de PATSTAT

Prueba de validación rápida

Para el "backup pequeño" que vas a lanzar ahora:

1. Crea una tabla pequeña de test con unos pocos miles de filas.
2. Lanza el BACKUP.
3. Crucial: Haz un RESTORE en una base de datos de prueba distinta para confirmar que el flujo es circular y el dato es íntegro

SQL

CODE

```
-- Ejemplo de restauración en DB de test
RESTORE SNAPSHOT db_test.snapshot_valida
FROM test_copies
PROPERTIES ("backup_timestamp" = "2026-04-30-10-30-00");
```

¿Qué capacidad de throughput tienes estimada entre tus nodos de StarRocks y el servidor de MinIO?

Jordi H.

Has dicho no he entendido como funciona el comando restore. ya lo estudiaré y te pediré ayuda luego. sobre el throughput, esto va desde una maquina virtual con un disco QEMU a través del comando de starrocks, todo ejecutándose en el mismo servidor que corre el docker de minio con un volumen montado en mi raid 5 de 12 TB que también soporta los discos de las VM. en realidad es de disco a disco, desde el mismo raid5 al mismo raid5. dame algun comando de cómo probar el throu

Gemini

Al estar todo en la misma máquina física y compartiendo el mismo **RAID 5**, el cuello de botella no será la red, sino la **latencia de entrada/salida (I/O Wait)** y la contención del bus de datos, ya que el RAID estará leyendo o (desde la VM/StarRocks) y escribiendo (hacia el volumen de MinIO) simultáneamente. Para medir el rendimiento real de este flujo, aquí tienes cómo probar

1. Prueba de "Fuerza Bruta" con `mc` (Desde la VM)

Antes de involucrar a StarRocks, mide cuánto tarda en mover un archivo grande desde el disco de la VM hacia MinIO. Esto te dará el **techo máximo** de velocidad: Bas

CODE

```
# Crea un archivo de prueba de 10GB dentro de la VM
dd if=/dev/zero of=test_10gb.img bs=1G count=10 status=progress

# Médelo con el cliente de MinIO (mc)
time mc cp test_10gb.img mi-minio/backup-starrocks/
```

- Qué observar: Si la velocidad es estable o si cae drásticamente (esto último indicaría que la caché de escritura del RAID se ha llenado)

2. Monitorización en tiempo real (I/O Contention)

Mientras lanzas el backup de StarRocks, abre otra terminal en el **host físico** para ver cómo sufre el RAID. La herramienta clave aquí es iostat (del paquete sysstat): Bas

CODE

```
# Muestra estadísticas cada 2 segundos centrado en discos  
iostat -xz 2
```

- Fíjate en %util: Si llega al 100%, el RAID 5 está al límite de su capacidad de maniobra.
- Fíjate en await: Si los milisegundos suben mucho, las peticiones de StarRocks están "haciendo cola" porque el disco está ocupado escribiendo en MinIO

3. Prueba de Red interna (Docker Bridge)

Aunque sea la misma máquina, los paquetes pasan por el stack de red de Docker. Puedes medir este "salto" con iperf3

- En el contenedor de MinIO: iperf3 -s
- Desde la VM de StarRocks: iperf3 -c [IP_del_contenedor_Minio]

Consideración para PATSTAT (300 GB) en RAID 5

El RAID 5 es excelente para lectura, pero tiene una **penalización en escritura** (Write Penalty) debido al cálculo de paridad. Como StarRocks leerá 300 GB de datos y MinIO escribirá otros 300 GB **en el mismo conjunto d**

1. Lectura: StarRocks leerá los archivos .dat de sus tablets.
2. Escritura: MinIO recibirá los datos y los escribirá, generando un flujo masivo de actualizaciones de paridad en el RAID

Mi consejo de arquitectura: Si tienes la posibilidad de usar un SSD dedicado (aunque sea pequeño) como **caché de escritura** para ese RAID o para el volumen de MinIO, notarás una mejora sustancial. Si no, prepárate para que el backup de la PATSTAT sea un proceso de varias horas donde el sistema se sentirá al go lento. ¿Tienes configurada alguna caché (tipo LVM Cache o ZFS intent log) sobre ese RAID

Jordi H.

Has dicho de donde saco el cliente mc de minio? por cierto, tengo instalado el midnight commander y se invoca como mc, ya me dices..

Gemini

Tienes toda la razón en advertirlo: el conflicto de nombres es un clásico en sistemas Linux. **Midnight** ** (mc) es el veterano de las consolas, mientras que el cliente de MinIO utiliza el mismo binario. Para evitar que tu sistema se vuelva loco (o que acabes abriendo el gestor de archivos cuando quieres subir un backup), aquí tienes la mejor forma de gestionarlo:

1. Descarga del binario oficial

MinIO distribuye su cliente como un único binario estático. Lo ideal es descargarlo en /usr/local/bin/ pero con un nombre distinto para evitar el conflicto. Ejecuta estos comandos en tu VM (como root o con sudo): Bas

CODE

```
# Descarga el binario (arquitectura x86_64)
curl https://dl.min.io/client/mc/release/linux-amd64/mc \
--create-dirs \
-o /usr/local/bin/mcli

# Dale permisos de ejecución
chmod +x /usr/local/bin/mcli
```

Nota: He sugerido renombrarlo a mcli (MinIO CLI). Así, cuando quieras usar el gestor de archivos clásico usarás mc, y para gestionar tus buckets usarás mcli

2. Configuración inicial

Una vez descargado y renombrado, el primer paso es configurar el acceso a tu servidor local. Al usar mcli, la sintaxis sería:Bas

CODE

```
mcli alias set local_minio http://[TU_IP]:9000 admin Minioplex.00
```

3. Comandos útiles para tu laboratorio

Ahora puedes usarlo para inspeccionar qué está haciendo StarRocks por debajo:

- Listar el contenido del bucket:
mcli ls local_minio/backup-starrocks
- Ver el espacio ocupado:
mcli du local_minio/backup-starrocks
- Monitorizar logs de acceso (muy útil para debug):
mcli admin trace local_minio

Sobre el throughput y el RAID 5

Como mencionabas que todo corre en el mismo RAID, el uso de mcli te va a permitir hacer una prueba de estrés controlada sin cargar la base de datos:Bas

CODE

```
# Prueba de subida midiendo tiempo
time mcli cp un_archivo_grande.iso local_minio/backup-starrocks/
```

Si ves que al ejecutar esto el **Midnight Commander** (mc) original se queda "congelado" o tarda en responder al navegar por carpetas, es la confirmación definitiva de que el RAID 5 está sufriendo por la **latencia de escritura**. Al ser una PATSTAT de 300 GB, este estrés será prolongado. Si el sistema se vuelve inestable durante el backup, podrías limitar el ancho de banda del cliente MinIO, aunque lo ideal en un homelab es simplemente dejarlo correr en horas de baja actividad. ¿Has conseguido ya que el comando BACKUP de StarRocks termine con éxito en la tabla pequeña de prueba?

Jordi H.

Has dicho 3,2 gb en 30 segundos desde mcli. creo que nos vamos a 45 minutos para 200 GB... voy a hacer un backup de una sola tabla

Gemini

Esa velocidad de **~100 MB/s** (3,2 GB en 30 segundos) es una señal excelente. Confirma que estás saturando prácticamente el límite de una red de 1 Gbps (si hubiera red de por medio) o que el RAID 5 está manteniendo un flujo de escritura secuencial muy sano a pesar de la paridad. Efectivamente, tus cálculos son precisos

- $\$200 \text{ GB} \div 100 \text{ MB/s} \approx 2000 \text{ segundos}$.
- $\$2000 \div 60 \approx \mathbf{\$33 \text{ minutos}}$ (en condiciones ideales).
- Con la sobrecarga de metadatos de StarRocks y la contención del RAID, esos 45 minutos que estimas son una previsión de arquitecto muy realista

Consejos para el Backup de una sola tabla

Al lanzar el comando para una única tabla, asegúrate de usar la sintaxis correcta para no incluir toda la base de datos por error: SQ

CODE

```
BACKUP SNAPSHOT test_copies.backup_tabla_especifica
TO test_copies
ON (nombre_de_la_tabla)
PROPERTIES ("timeout" = "3600"); -- 1 hora por seguridad
```

Qué vigilar durante estos 45 minutos

Como vas a mover un volumen considerable (aunque sea una sola tabla de la PATSTAT), te sugiero monitorizar dos cosas

1. Estado de los BE (Backend): StarRocks divide la tabla en tablets. Cada BE leerá sus tablets y las enviará a MinIO. Si tienes varios BE, verás que MinIO recibe múltiples conexiones simultáneas. Puedes verlo con: Ba

CODE

```
mcli admin trace local_minio
```

2. Consumo de CPU en el Host: El cálculo de la paridad del RAID 5 durante una escritura sostenida de 45 minutos elevará la temperatura y el uso de CPU del host físico. Si el servidor tiene otras funciones críticas, asegúrate de que tiene "aire" suficiente.
3. Verificación Final:

Una vez que SHOW BACKUP te diga FINISHED, no te fíes solo del estado. Usa tu nuevo comando mcli para ver el tamaño real en el bucket: Bas

CODE

```
mcli du -h local_minio/backup-starrocks
```

Si esta tabla de prueba sale bien, ya tienes el "plano" para el despliegue corporativo. ¿Qué tabla has elegido para la prueba? Si es una de las de "enlaces" (como `tls212_citation`), suelen ser ideales porque tienen mucha cardinalidad y ponen a prueba la generación de índices en el backup

Jordi H.

Has dicho 8gb en 4 minutos. he usado la tabla `tls212`

Eso nos da una media de unos **33 MB/s** ($\frac{2.000 \text{ MB}}{60 \text{ segundos}}$ aproximadamente). Es una velocidad significativamente menor que la prueba sintética con mcli, lo cual es normal: StarRocks no solo está "copiando" bits; está empaquetando tablets, calculando checksums y gestionando metadatos mientras lee de sus propios volúmenes. A este ritmo, los 300 GB de la PATSTAT completa tardarían una **2,5 horas**. Si que siendo un tiempo muy razonable para un backup de este calibre en un home

Análisis de la tabla `tls212\_citation`

Has elegido una tabla excelente para el test. tls212 contiene las citas de las patentes y, aunque no es la más pesada en gigabytes comparada con las de abstractos o títulos, tiene una **densidad de filas enorme**. Al ser una tabla con muchas claves externas y relaciones, para StarRocks esto signific

- Muchos índices que respaldar.
- Una estructura de almacenamiento (columnar) muy eficiente pero que requiere CPU para ser leída y enviada al repositorio S3

Verificación de la integridad

Para cerrar el ciclo de "arquitecto de sistemas" y quedarte tranquilo antes de ir a por los 300 GB, te sugiero hacer la prueba de fuego: **el Restore parcial**. No necesitas borrar nada. Puedes restaurar esa tabla con un nombre diferente en una base de datos de pruebas para verificar que los datos han sobrevivido al viaje al RAID 5:

CODE

```
-- 1. Crea una DB de pruebas
CREATE DATABASE patstat_test;

-- 2. Restaura solo esa tabla desde el snapshot
RESTORE SNAPSHOT patstat_test.test_restore
FROM test_copies
ON (tls212 AS tls212_restored)
PROPERTIES ("backup_timestamp" = "2026-04-30-XX-XX-XX"); -- Mira el timestamp en SHOW BACKUP
```

Un detalle sobre el RAID 5 y el I/O Wait

Si durante esos 4 minutos has notado que el sistema respondía bien, significa que el RAID 5 está gestionando correctamente la carga. Si planeas automatizar esto, podrías considerar programarlo para horas donde no lances procesos de analítica pesados en Python, para evitar que la lectura del backup compita con tus queries de arquitectura. ¿El comando SHOW BACKUP\G te muestra el estado como FINISHED y te da un backup_timestamp corre