ClickHouse Automatic System Log Cleanup Configuration

- تنظیم پاکسازی خودکار لاگ‌های کلیک هاوس

تنظیم پاک‌سازی خودکار لاگ‌های کلیک هاوس

هدف

برای جلوگیری از افزایش بیش از حد حجم لاگ‌های سیستمی کلیک هاوس و پر شدن فضای دیسک، برای جدول‌های لاگ مدت نگهداری مشخص کنید.

در این تنظیم، اطلاعات لاگ به مدت هفت روز نگهداری می‌شوند و اطلاعات قدیمی‌تر به صورت خودکار حذف می‌شوند.

بررسی محل اجرای کلیک هاوس

ابتدا کانتینر کلیک هاوس را پیدا کنید:

```
docker ps
```

کانتینری که از تصویر زیر استفاده می‌کند مربوط به کلیک هاوس است:

برای مشاهده مسیرهای متصل‌شده به کانتینر، دستور زیر را اجرا کنید:

```
docker inspect clickhouse | grep -A 40 '"Mounts"'
```

به طور مثال:

```
/home/infra/clickhouse/clickhouse/data
```

اضافه کردن مسیر تنظیمات

فایل `docker-compose.yml` را باز کنید.

اگر در بخش `volumes` فقط مسیر اطلاعات وجود دارد:

```
volumes:
  - ./clickhouse/data/:/var/lib/clickhouse/
```

مسیر تنظیمات را نیز اضافه کنید:

```
volumes:
  - ./clickhouse/data/:/var/lib/clickhouse/
  - ./clickhouse/config.d/:/etc/clickhouse-server/config.d/
```

سپس مسیر تنظیمات را روی سرور ایجاد کنید:

```
mkdir -p ./clickhouse/config.d
```

قبل از ایجاد مجدد کانتینر، فایل تنظیمات پیشفرض کلیک هاوس را از داخل کانتینر به مسیر جدید منتقل کنید:

```
docker cp clickhouse:/etc/clickhouse-server/config.d/docker_related_config.xml
./clickhouse/config.d/
```

با این کار، فایل پیشفرض موجود در کانتینر پس از اتصال مسیر جدید از بین نمیرود.

ایجاد تنظیمات پاکسازی خودکار

فایل زیر را ایجاد کنید:

```
./clickhouse/config.d/system_logs_ttl.xml
```

محتوای زیر را داخل فایل قرار دهید:

```
<clickhouse>
    <query_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
    </query_log>

    <text_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
        <level>warning</level>
    </text_log>
```

```
    <trace_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
    </trace_log>

    <metric_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
        <collect_interval_milliseconds>60000</collect_interval_milliseconds>
    </metric_log>

    <asynchronous_metric_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
    </asynchronous_metric_log>

    <processors_profile_log>
        <ttl>event_date + INTERVAL 7 DAY DELETE</ttl>
    </processors_profile_log>
</clickhouse>
```

این تنظیم را برای جدول‌های زیر اعمال کنید:

```
query_log
text_log
trace_log
metric_log
asynchronous_metric_log
processors_profile_log
```

با این تنظیم، اطلاعات قدیمی‌تر از هفت روز برای حذف خودکار در نظر گرفته می‌شوند.

اعمال تنظیمات

پس از ایجاد فایل، کانتینر را مجدداً ایجاد کنید:

```
docker compose down
docker compose up -d
```

سپس لاگ راه‌اندازی کلیک هاوس را بررسی کنید:

```
docker logs --tail 100 clickhouse
```

وجود خط مشابه زیر را بررسی کنید:

```
Merging configuration file '/etc/clickhouse-server/config.d/system_logs_ttl.xml'.
```

وجود این خط نشان میدهد فایل تنظیمات خواندہ شدہ است.

بررسی اعمال مدت نگهداری

وارد کلیک هاوس شوید:

```
docker exec -it clickhouse clickhouse-client
```

سپس دستور زیر را اجرا کنید:

```
SELECT
    name,
    engine_full
FROM system.tables
WHERE database = 'system'
  AND name IN (
      'query_log',
      'text_log',
      'trace_log',
      'metric_log',
      'asynchronous_metric_log',
      'processors_profile_log'
  )
ORDER BY name;
```

برای هر جدول وجود مقدار مشابه زیر را بررسی کنید:

```
TTL event_date + toIntervalDay(7)
```

اگر این مقدار وجود داشته باشد، مدت نگهداری هفت روز روی جدول اعمال شدہ است.

بررسی حجم جدول‌ها

برای پیدا کردن جدول‌هایی که بیشترین فضای دیسک را مصرف می‌کنند، دستور زیر را اجرا کنید:

```
SELECT
    database,
    table,
    formatReadableSize(sum(bytes_on_disk)) AS size,
    sum(bytes_on_disk) AS size_bytes
FROM system.parts
WHERE active
GROUP BY
    database,
    table
ORDER BY size_bytes DESC
LIMIT 20;
```

جدول‌های سیستمی با حجم بالا را بررسی کنید.

به نسخه‌های شماره‌دار جدول‌های لاگ توجه کنید. برای مثال:

```
trace_log_0
trace_log_1
text_log_0
text_log_1
metric_log_0
metric_log_1
```

این جدول‌ها ممکن است مربوط به نسخه‌های قبلی جدول‌های سیستمی باشند.

تشخیص جدول‌های قدیمی

برای تشخیص نسخه فعال و نسخه‌های قدیمی `trace_log`، زمان آخرین اطلاعات موجود در هر جدول را بررسی کنید:

```
SELECT
    'trace_log' AS name,
    max(event_time) AS newest,
    count() AS rows
FROM system.trace_log

UNION ALL

SELECT
    'trace_log_0',
    max(event_time),
    count()
FROM system.trace_log_0

UNION ALL

SELECT
    'trace_log_1',
    max(event_time),
    count()
FROM system.trace_log_1;
```

جدولی را که اطلاعات جدید دریافت می‌کند به عنوان جدول فعال در نظر بگیرید.

اگر نسخه‌های شماره‌دار دیگر اطلاعات جدید دریافت نمی‌کنند و اطلاعات آن‌ها موردنیاز نیست، آن‌ها را حذف کنید.

حذف نسخه‌های قدیمی

برای حذف نسخه‌های قدیمی از دستورهای زیر استفاده کنید:

```
DROP TABLE IF EXISTS system.trace_log_0;
DROP TABLE IF EXISTS system.trace_log_1;
```

اگر کلیک هاوس به دلیل بزرگ بودن جدول اجازه حذف ندادو خطای محدودیت حجم نمایش داده شد، محدودیت حذف را برای همان نشست بردارید:

```
SET max_table_size_to_drop = 0;
```

سپس دوباره دستور حذف را اجرا کنید:

```
DROP TABLE system.trace_log_1;
```

قبل از حذف هر جدول شماره‌دار، از غیرفعال بودن و عدم نیاز به اطلاعات آن اطمینان حاصل کنید.

بررسی آزاد شدن فضای دیسک

پس از حذف جدول‌های قدیمی، حجم مسیر اطلاعات کلیک هاوس را بررسی کنید:

```
du -sh /home/infra/clickhouse/clickhouse/data
```

فضای کلی دیسک را نیز بررسی کنید:

```
df -h /home/infra/clickhouse/clickhouse/data
```

برای بررسی مجدد حجم جدول‌ها، دستور زیر را اجرا کنید:

```
SELECT
    database,
    table,
    formatReadableSize(sum(bytes_on_disk)) AS size
FROM system.parts
WHERE active
GROUP BY
    database,
    table
ORDER BY sum(bytes_on_disk) DESC;
```

راه‌اندازی خودکار پاک‌سازی عملکرد آزمایشی

برای آزمایش عملکرد پاک‌سازی بدون تغییر تنظیمات اصلی، یک جدول آزمایشی ایجاد کنید:

```
CREATE TABLE default.ttl_test
(
```

```
    ts DateTime,
    value UInt32
)
ENGINE = MergeTree
ORDER BY ts
TTL ts + INTERVAL 1 MINUTE DELETE;
```

دو رکورد با زمان متفاوت وارد کنید:

```
INSERT INTO default.ttl_test VALUES
(now() - INTERVAL 5 MINUTE, 1),
(now(), 2);
```

اطلاعات جدول را بررسی کنید:

```
SELECT *
FROM default.ttl_test
ORDER BY ts;
```

در این مرحله باید هر دو رکورد نمایش داده شوند.

سپس ادغام نهایی را اجرا کنید:

```
OPTIMIZE TABLE default.ttl_test FINAL;
```

دوباره اطلاعات جدول را بررسی کنید:

```
SELECT *
FROM default.ttl_test
ORDER BY ts;
```

رکوردی که پنج دقیقه از زمان آن گذشته است باید حذف شود و فقط رکورد جدید باقی بماند.

پس از پایان آزمایش، جدول را حذف کنید:

```
DROP TABLE default.ttl_test;
```

بررسی دوره‌ای

برای بررسی وضعیت لاگ‌های سیستمی، به صورت دوره‌ای حجم جدول‌ها را با دستور زیر کنترل کنید:

```
SELECT
    database,
    table,
    formatReadableSize(sum(bytes_on_disk)) AS size
FROM system.parts
WHERE active
GROUP BY
    database,
    table
ORDER BY sum(bytes_on_disk) DESC;
```

در صورت مشاهده جدول‌های شماره‌دار مانند موارد زیر:

```
trace_log_0
trace_log_1
text_log_0
metric_log_1
```

ابتدا آخرین زمان ثبت اطلاعات در آن‌ها را بررسی کنید و تنها در صورت غیرفعال بودن، آن‌ها را حذف کنید.

با ثابت نگه داشتن تنظیمات جدول‌های لاگ، اطلاعات جدید در جدول‌های اصلی ذخیره می‌شوند و اطلاعات قدیمی‌تر از هفت روز نیز به وسیله مدت نگهداری تعیین‌شده حذف می‌شوند.