ClickHouse 101

9/20/2023

Clickhouse – using SHOW and SQL

SHOW COMMAND	SQL COMMAND
SHOW DATABASES;	SELECT name FROM system.databases [WHERE name [NOT] LIKE | ILIKE '<pattern>'] [LIMIT <N>] [INTO OUTFILE filename] [FORMAT format]
SHOW DATABASES LIKE '%pg%';	SELECT name FROM system.databases WHERE name LIKE '%pg%';
SHOW TABLES;	SELECT * FROM system.tables FORMAT Vertical;
SHOW TABLES LIKE '%facts%';	SELECT * FROM system.tables WHERE table LIKE '%facts%';
SHOW TABLES FROM system LIKE '%user%'	SELECT * FROM system.tables WHERE database='system';
SHOW COLUMNS FROM users IN system;	SELECT * FROM system.columns WHERE database = 'system' AND table = 'users';
SHOW COLUMNS FROM 'orders' LIKE 'delivery_%'	SELECT * FROM system.columns WHERE table='orders' AND name LIKE 'delivery_%';

Clickhouse System Monitoring

Task	SQL
SHOW PROCESSLIST;	SELECT query_id, user, address, elapsed, query FROM system.processes;
Check S3 Counters	SELECT * FROM system.events WHERE event ILIKE '%S3%';
Global Query Counters	SELECT * FROM system.events WHERE event ILIKE '%query%';
Check Resource Usage	SELECT * FROM system.asynchronous_metric_log WHERE event_date = '2023-09-20'
I/O Blocking Metrics	SELECT * FROM system.asynchronous_metrics WHERE metric ILIKE '%block%'
Available Space	SELECT * FROM system.asynchronous_metrics WHERE metric ILIKE '%available%'
Check TCP Connections	SELECT * FROM system.metrics WHERE metric LIKE '%Connection';
Global File Counters	SELECT * FROM system.metrics WHERE metric ILIKE '%file%';

Clickhouse System Monitoring

Task	SQL
SHOW PROCESSLIST;	SELECT query_id, user, address, elapsed, query FROM system.processes;
Check S3 Counters	SELECT * FROM system.events WHERE event ILIKE '%S3%';
Global Query Counters	SELECT * FROM system.events WHERE event ILIKE '%query%';
Check Resource Usage	SELECT * FROM system.asynchronous_metric_log WHERE event_date = '2023-09-20'
I/O Blocking Metrics	SELECT * FROM system.asynchronous_metrics WHERE metric ILIKE '%block%'
Available Space	SELECT * FROM system.asynchronous_metrics WHERE metric ILIKE '%available%'
Check Local Disk	SELECT * FROM system.disks;
Check TCP Connections	SELECT * FROM system.metrics WHERE metric LIKE '%Connection';
Global File Counters	SELECT * FROM system.metrics WHERE metric ILIKE '%file%';
Check Crash Logs	SELECT * FROM system.crash_logs (may not exist)

Cluster Status	SELECT cluster, host_name, host_address, shard_num, replica_num
	FROM system.clusters ORDER BY shard_num, replica_num;

Check Replicas	SELECT * FROM system.replicas

PostgreSQL	ClickHouse SQL
SELECT generate_series(1,1000)	SELECT * FROM system.numbers LIMIT 1000;

Clickhouse Client Command Line Flags

clickhouse client commands	
clickhouse client – server_logs_file ch.log	Download the server log
clickhouse client --ignore-error	do not stop processing in multiquery mode
clickhouse client --stacktrace	print stack traces of exceptions
clickhouse client -E	Same as adding "FORMAT Vertical" to the end of each query
clickhouse client --queries-file	Allows multiple CH-SQL commands to run from a file
clickhouse client --echo	Useful in batch mode since all commands are echo'ed out while being sent to the clickhouse server

Checking on currently running queries

```
clickhouse :) SELECT query_id, user, address, elapsed, query   FROM system.processes ORDER BY query_id;

Query id: 45de25e1-3fff-4800-a5ac-a8d03b9127c0

Row 1:
──────
query_id: 45de25e1-3fff-4800-a5ac-a8d03b9127c0
user:     default
address:  ::ffff:10.0.2.31
elapsed:  0.00281
query:    SELECT query_id, user, address, elapsed, query   FROM system.processes ORDER BY query_id;
```

Checking on currently running queries

```
% cd dba_repo/clickhouse
% clickhouse client --host clickhouse1 -E --queries-file
current_queries.sql
Row 1:
──────
query_id: 6be19221-b4f3-4453-972c-ac7b15c1ccf9
user:     default
address:  ::ffff:10.0.2.31
elapsed:  0.00316
query:    SELECT query_id, user, address, elapsed,
query   FROM system.processes ORDER BY query_id FORMAT
Vertical;
```

What a lack of permissions looks like

```
clickhouse :) SELECT * FROM system.users;

SELECT *
FROM system.users
Query id: 242d7efb-ce9d-4e1e-a74d-07adb0f88f71
0 rows in set. Elapsed: 0.157 sec.
Received exception from server (version 23.1.3):
Code: 497. DB::Exception: Received from clickhouse1:9000. DB::Exception: default: Not enough privileges. To execute this query it's necessary to have grant SHOW USERS ON *.*. (ACCESS_DENIED)
```

Clickhouse – Log Location & Configuration

Important Directories	Description
/etc/clickhouse-client/	Config files for Clickhouse Client
/etc/clickhouse-server/	Config files for Clickhouse Server
/var/lib/clickhouse	Data Files for Clickhouse

Log File Location	Description
/var/log/clickhouse-server/clickhouse-server.err.log	Clickhouse Server Error Log
/var/log/clickhouse-server/clickhouse-server.log	Clickhouse Server Log

Information about a running Clickhouse Server	Description
/run/clickhouse-server/clickhouse-server.pid	PID of running Clickhouse Server
/var/lib/clickhouse/status	Start time and pid of running Clickhouse Server

Clickhouse TCP Ports

- **standalone clickhouse instance**

```
$ sudo lsof -P -p 27812 | grep TCP
TCP *:8123 (LISTEN)
TCP *:9000 (LISTEN)
TCP *:9004 (LISTEN)
TCP *:9005 (LISTEN)
TCP *:9009 (LISTEN)
```

TCP Port	Comment
9000	Main Listener – (Clickhouse Native)
9004	MySQL Listener
9005	PostgreSQL Listener
9009	Clickhouse Interserver port
9010	SSL/TLS – Clickhouse Interserver
9011	Clickhouse Native PROXYv1 port
9019	JDBC Bridge
9100	gRPC port
9181	Clickhouse Keeper (IMPORTANT)
9234	Clickhouse Keeper Raft
9440	SSL/TLS - Clickhouse Native
2181	Zookeeper default – Check 9181
8123	HTTP API Port (JDBC, ODBC)
8443	SSL/TLS - HTTPS API Port
9363	Promethius Metrics Port
9281	SSL/TLS – Clickhouse Keeper

Installation from RPM

- **Add the ClickHouse RPM repository**

```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://packages.clickhouse.com/rpm/clickhouse.repo
```

- **Install ClickHouse server and client**

```
sudo yum install -y clickhouse-server clickhouse-client
```

Starting and Stopping Clickhouse

- **Start ClickHouse server**

-
  ```
  sudo systemctl enable clickhouse-server
  sudo systemctl start clickhouse-server
  sudo systemctl status clickhouse-server
  clickhouse-client # or "clickhouse-client --password" if you set up a password.
  ```
- **Stop ClickHouse server**

-
  ```
  sudo systemctl stop clickhouse-server
  sudo systemctl status clickhouse-server
  ```
- **Check Status of ClickHouse server**

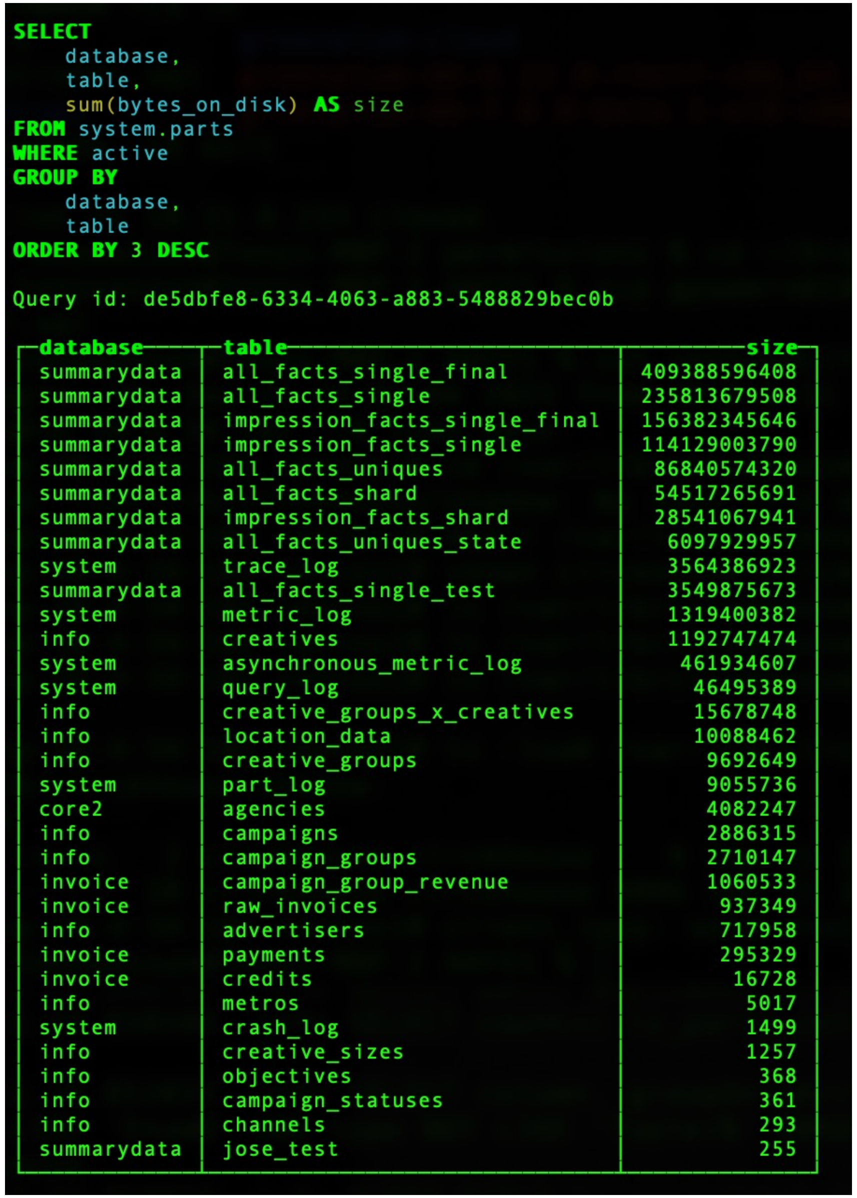
-
  ```
  sudo systemctl status clickhouse-server
  ```

Checking on the location of a Database

- Database – Contains tables loaded from the Engine
- Engine – ReplicatedMergeTree or S3Storage or PostgresDBQuery
- Schema – Just prepend the schemaname to the table name
- Storage – Can be S3, local files or temporary (not persisted)

Size of each table on Disk

```
SELECT database, table
, sum(bytes_on_disk) AS size
FROM system.parts
WHERE active
GROUP BY database, table
ORDER BY 3 DESC
FORMAT Pretty;
```



```
SELECT
    database,
    table,
    sum(bytes_on_disk) AS size
FROM system.parts
WHERE active
GROUP BY
    database,
    table
ORDER BY 3 DESC

Query id: de5dbfe8-6334-4063-a883-5488829bec0b

┌─database────┬─table─────────────────────────┬─────────size─┐
│ summarydata │ all_facts_single_final        │ 409388596408 │
│ summarydata │ all_facts_single              │ 235813679508 │
│ summarydata │ impression_facts_single_final │ 156382345646 │
│ summarydata │ impression_facts_single       │ 114129003790 │
│ summarydata │ all_facts_uniques             │  86840574320 │
│ summarydata │ all_facts_shard               │  54517265691 │
│ summarydata │ impression_facts_shard        │  28541067941 │
│ summarydata │ all_facts_uniques_state       │   6097929957 │
│ system      │ trace_log                     │   3564386923 │
│ summarydata │ all_facts_single_test         │   3549875673 │
│ system      │ metric_log                    │   1319400382 │
│ info        │ creatives                     │   1192747474 │
│ system      │ asynchronous_metric_log       │    461934607 │
│ system      │ query_log                     │     46495389 │
│ info        │ creative_groups_x_creatives   │     15678748 │
│ info        │ location_data                 │     10088462 │
│ info        │ creative_groups               │      9692649 │
│ system      │ part_log                      │      9055736 │
│ core2       │ agencies                      │      4082247 │
│ info        │ campaigns                     │      2886315 │
│ info        │ campaign_groups               │      2710147 │
│ invoice     │ campaign_group_revenue        │      1060533 │
│ invoice     │ raw_invoices                  │       937349 │
│ info        │ advertisers                   │       717958 │
│ invoice     │ payments                      │       295329 │
│ invoice     │ credits                       │        16728 │
│ info        │ metros                        │         5017 │
│ system      │ crash_log                     │         1499 │
│ info        │ creative_sizes                │         1257 │
│ info        │ objectives                    │          368 │
│ info        │ campaign_statuses             │          361 │
│ info        │ channels                      │          293 │
│ summarydata │ jose_test                     │          255 │
└─────────────┴───────────────────────────────┴──────────────┘
```

Checking Free Space on Local Disks

```
clickhouse :) SELECT free_space,* FROM system.disks;

Row 1:
──────
free_space:       839891140608
name:             default
path:             /var/lib/clickhouse/
free_space:       839891140608
total_space:      2147472093184
unreserved_space: 839891140608
keep_free_space:  0
type:             local
is_encrypted:     0
is_read_only:     0
is_write_once:    0
is_remote:        0
is_broken:        0
cache_path:
```

Debugging Clickhouse via SQL

- SELECT
- arrayStringConcat(arrayMap(x -> demangle(addressToSymbol(x)), trace), '\n') AS trace_functions,
- count()
- FROM system.stack_trace
- GROUP BY trace_functions
- ORDER BY count()
- DESC
- SETTINGS allow_introspection_functions=1
- FORMAT Vertical;

Debugging for a Stack Trace (Alternate)

- To be done on clickhouse-server

```
for i in $(ls -1 /proc/$(pidof clickhouse-server)/task/); do kill -TSTP $i; done
```

clickhouse-backup

```
[centos@ip-10-4-34-182 ~]$ clickhouse-backup
NAME:
   clickhouse-backup - Tool for easy backup of ClickHouse with cloud support

USAGE:
   clickhouse-backup <command> [-t, --tables=<db>.<table>] <backup_name>

DESCRIPTION:
   Run as 'root' or 'clickhouse' user

COMMANDS:
   tables              List of tables, exclude skip_tables
   create              Create new backup
   create_remote       Create and upload new backup
   upload              Upload backup to remote storage
   list                List of backups
   download            Download backup from remote storage
   restore             Create schema and restore data from backup
   restore_remote      Download and restore
   delete              Delete specific backup
   default-config      Print default config
   print-config        Print current config merged with environment variables
   clean               Remove data in 'shadow' folder from all 'path' folders available from 'system.disks'
   clean_remote_broken  Remove all broken remote backups
   watch               Run infinite loop which create full + incremental backup sequence to allow efficient backup sequences
   server              Run API server
   help, h             Shows a list of commands or help for one command


GLOBAL OPTIONS:
   --config value, -c value  Config 'FILE' name. (default: "/etc/clickhouse-backup/config.yml") [$CLICKHOUSE_BACKUP_CONFIG]
   --help, -h                show help
   --version, -v             print the version
```