Showing Beautiful ClickHouse® Data with the **Altinity® Grafana Plugin**

Robert Hodges - Altinity
Eugene Klimov - Altinity

A brief message from our sponsor…

Robert Hodges

Database geek with 30+ years on DBMS. Kubernaut since 2018. Day job: Altinity CEO

Eugene Klimov

Maintains clickhouse-backup and Altinity Grafana plugin. Day job: Cloud Engineer

ClickHouse support and services including Altinity.Cloud
Authors of Altinity Kubernetes Operator for ClickHouse, Altinity clickhouse-backup and other open source projects

The Altinity Grafana Community Plugin for ClickHouse

Language
Golang + React

GitHub Project
https://github.com/Altinity/clickhouse-grafana

Downloads
16.6M

License
MIT

Recent Releases
• **v 2.5.4** - Last release for pre-Grafana 10 • **V 3.0.0** - Rewrite from AngularJS to React for Grafana 10 • **V 3.1.0** - Bug fix release

Original Author
Roman Kavronenko

Maintainer
Eugene Klimov

Introducing ClickHouse and Grafana

Introducing ClickHouse, a real-time analytic database

Understands SQL

Runs on bare metal to cloud

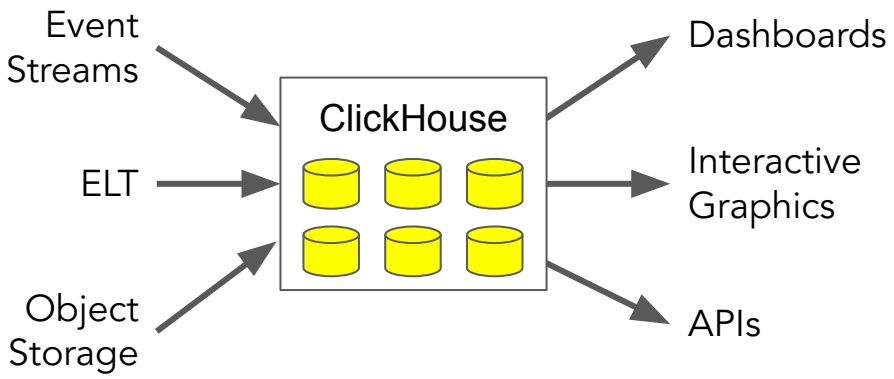
Shared nothing architecture

Stores data in columns

Parallel and vectorized execution

Scales to many petabytes

Is Open source (Apache 2.0)



Grafana is a display tool for time series data

Understands time series data

Simple installation

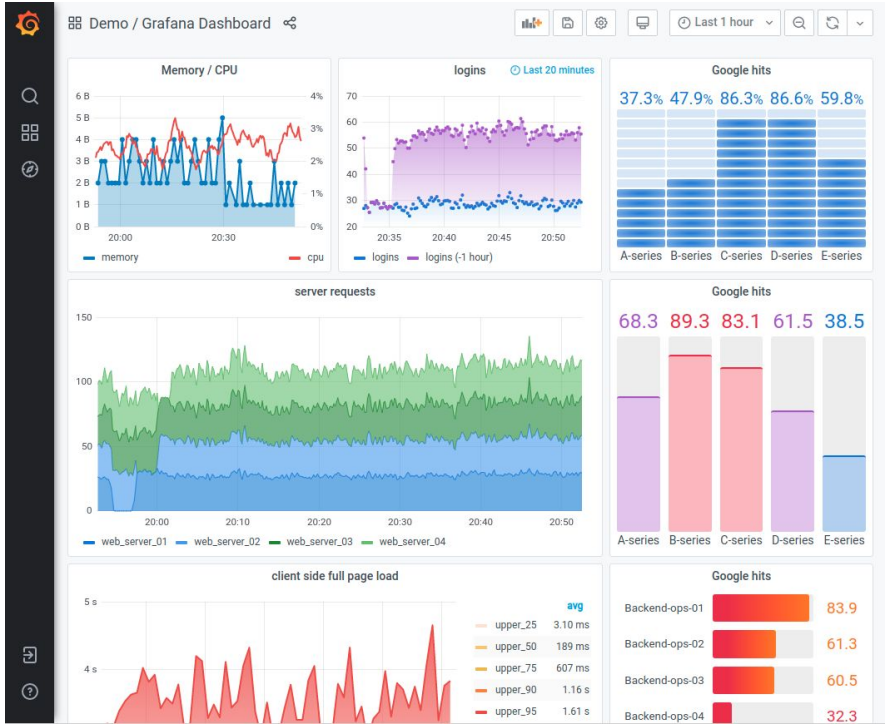
Many data sources

Lots of display plugins

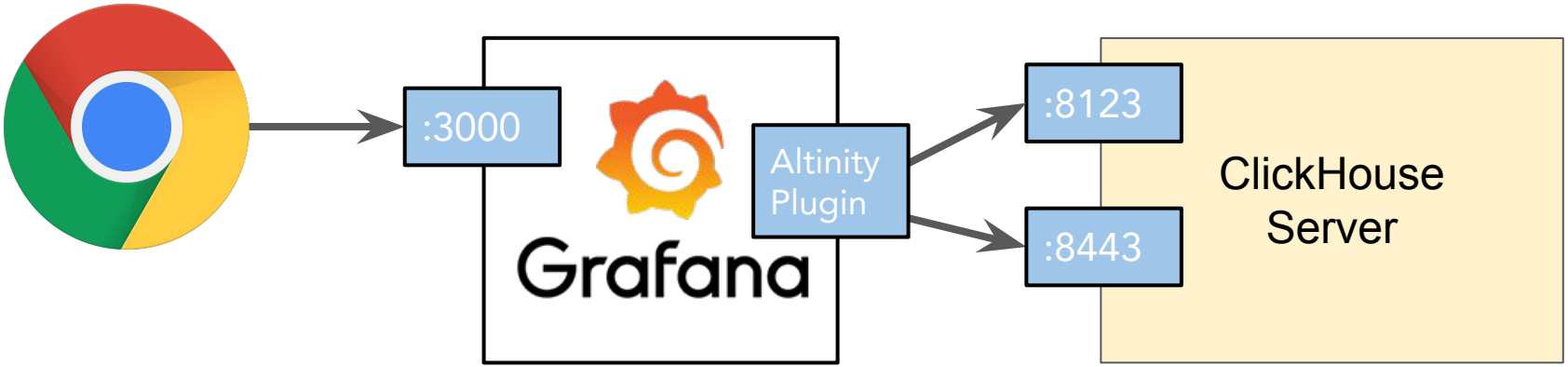
Interactive time windows

Great for monitoring dashboards

Is open source (AGPL 3.0)



Basic Grafana architecture



Displays the dashboard

Stores and serves dashboards; makes calls to backends

Processes SQL requests and returns results

Altinity.Cloud provides a playground for ClickHouse

Connection parameters

URL	https://github.demo.altinity.cloud:8443
User	demo
PW	demo

Datasets

airports	Airport names and locations
github_events	Full event history from Github (3.1B rows)
ontime	Airline ontime data (196M rows)
tripdata	NYC taxi commission ride data (1.3B rows)

Installing Grafana and the Altinity Plugin

Setting up Grafana locally on Ubuntu

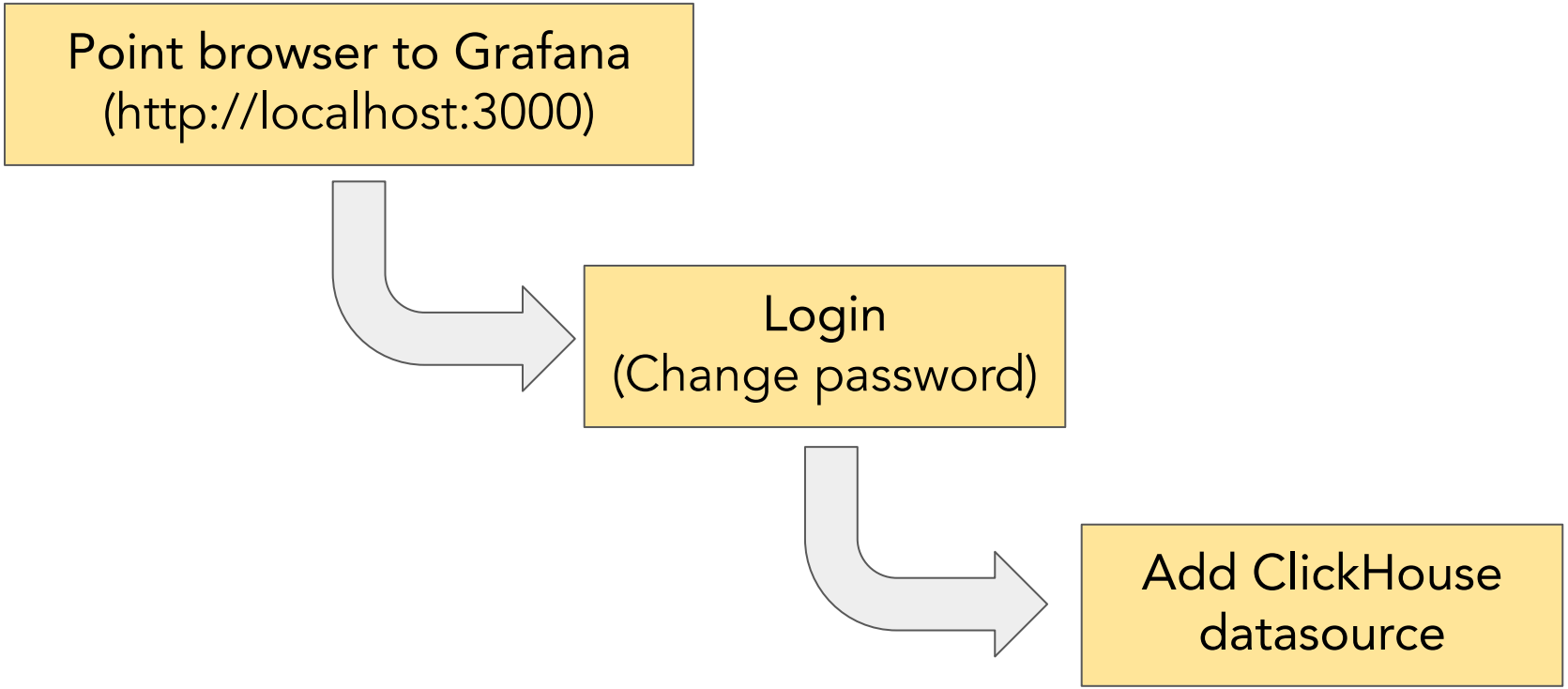
```
# Install Grafana server.
sudo echo 'deb https://packages.grafana.com/oss/deb stable main' >
/etc/apt/sources.list.d/grafana.list
curl https://packages.grafana.com/gpg.key | sudo apt-key add -
sudo apt-get -y install grafana

# Install ClickHouse plugin
sudo grafana-cli plugins install vertamedia-clickhouse-datasource
sudo systemctl restart grafana-server.service
```

Setting up Grafana on Docker

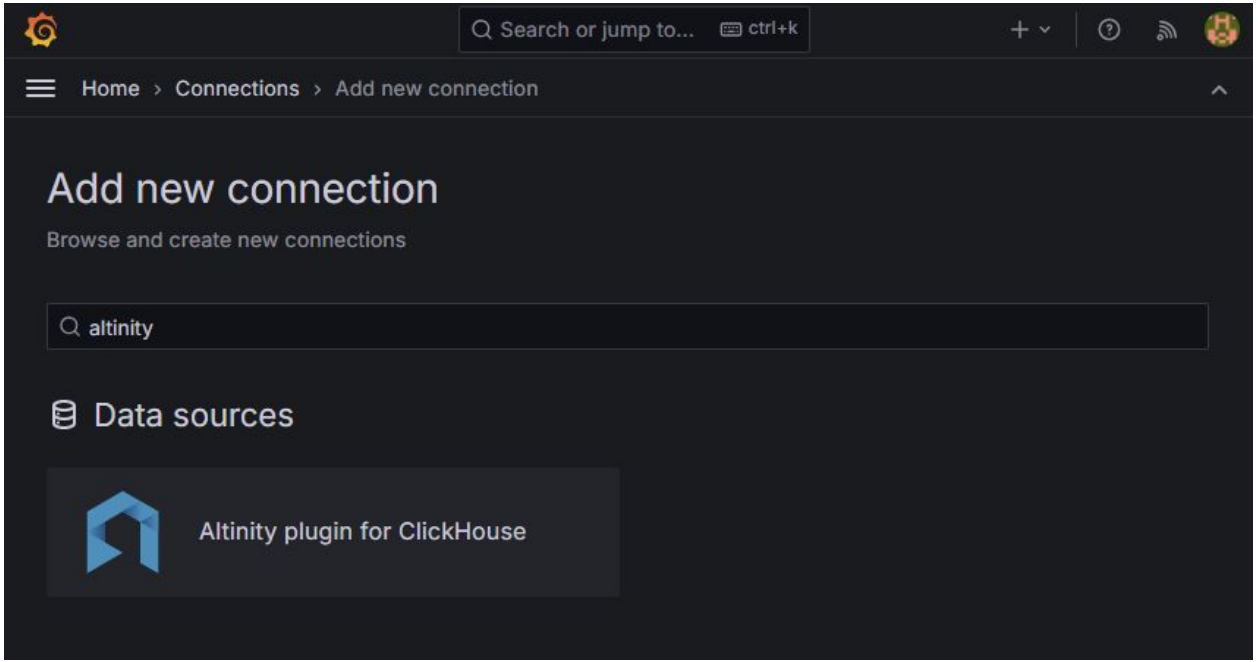
```
# Run Grafana in docker with automatic plugin installation.
docker run -d \
  -p 3001:3000 \
  --name=grafana \
  -e "GF_INSTALL_PLUGINS=vertamedia-clickhouse-datasource" \
  grafana/grafana
```

…Or if you like docker-compose, look at: https://github.com/Altinity/clickhouse-grafana/blob/master/docker-compose.yaml

Steps to connect to ClickHouse



You can install the Altinity plugin directly in Grafana



Plugin installation screen - Just press Install

Search or jump to... ctrl+k

Home › Connections › Add new connection › Altinity plugin for ClickHouse

Altinity plugin for ClickHouse

Install

Version	From	Downloads	Dependencies	Signature
3.0.0	Vertamedia	15,009,821	Grafana >=10.0.3	Signed

Altinity Grafana datasource for ClickHouse

GitHub

Overview | Version history

Altinity Grafana datasource plugin for ClickHouse (grafana Grafana 4.6+ supported)

Altinity ClickHouse datasource plugin provides a support for ClickHouse as a backend database.

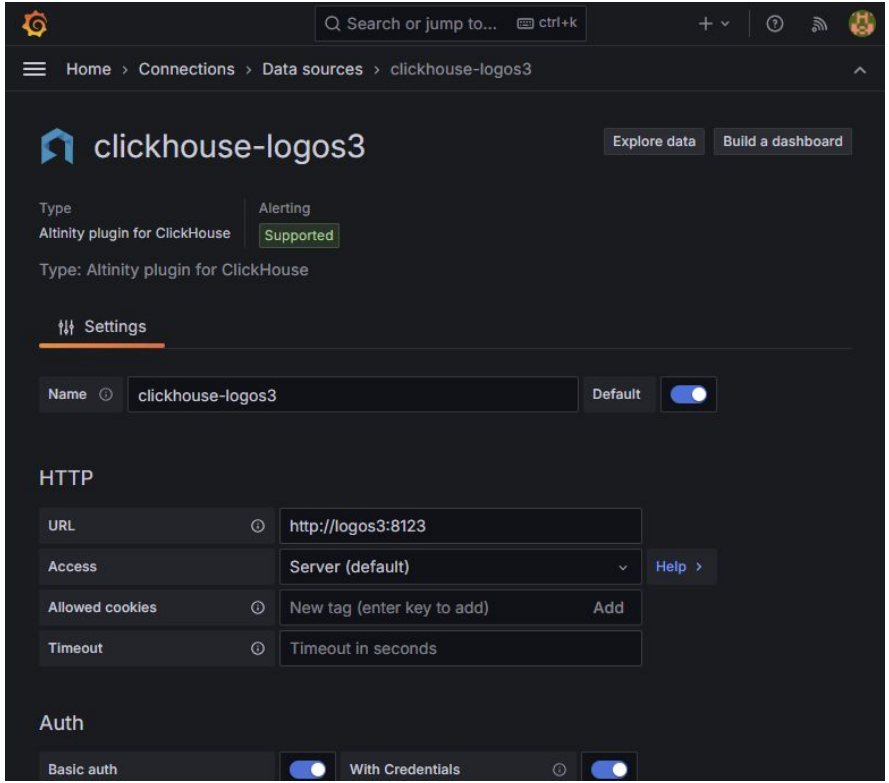
Initially plugin developed by Vertamedia, maintaned by Altinity since 2020.

Quick start

Grafana 10+ setup notes for plugin version before 3.0.0

Old versions of Altinity ClickHouse datasource plugin for Grafana written in Angular. So you can watch warning like

…Then create a data source

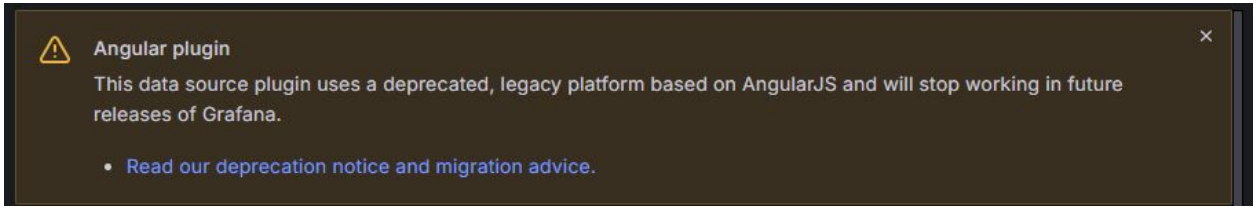


Pro tips:

- Use Server Access
- Use POST
- Enable Basic auth & With Credentials

What if you see…

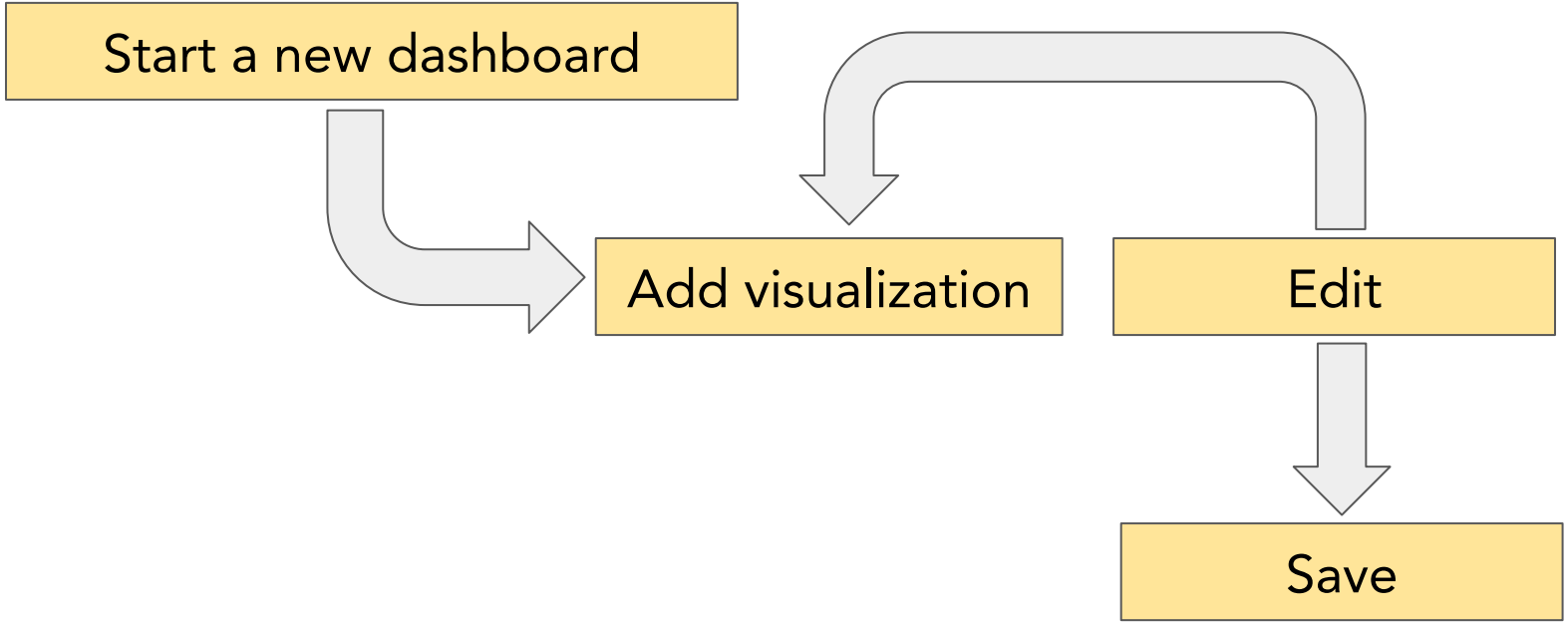
Time to upgrade to Grafana 10 or later!



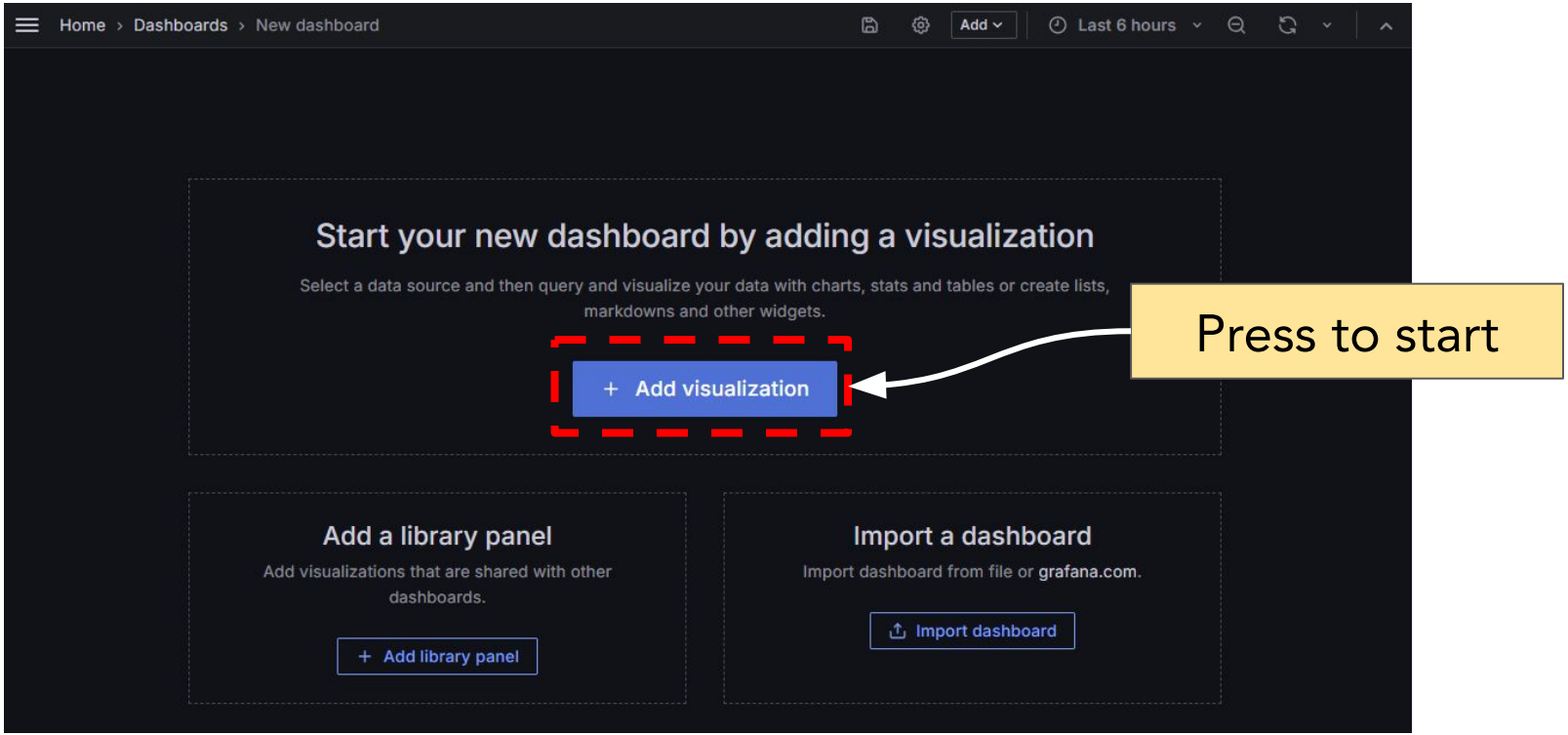
Time to upgrade the Altinity plugin to 3.0 or later!

Let’s build our first dashboard!

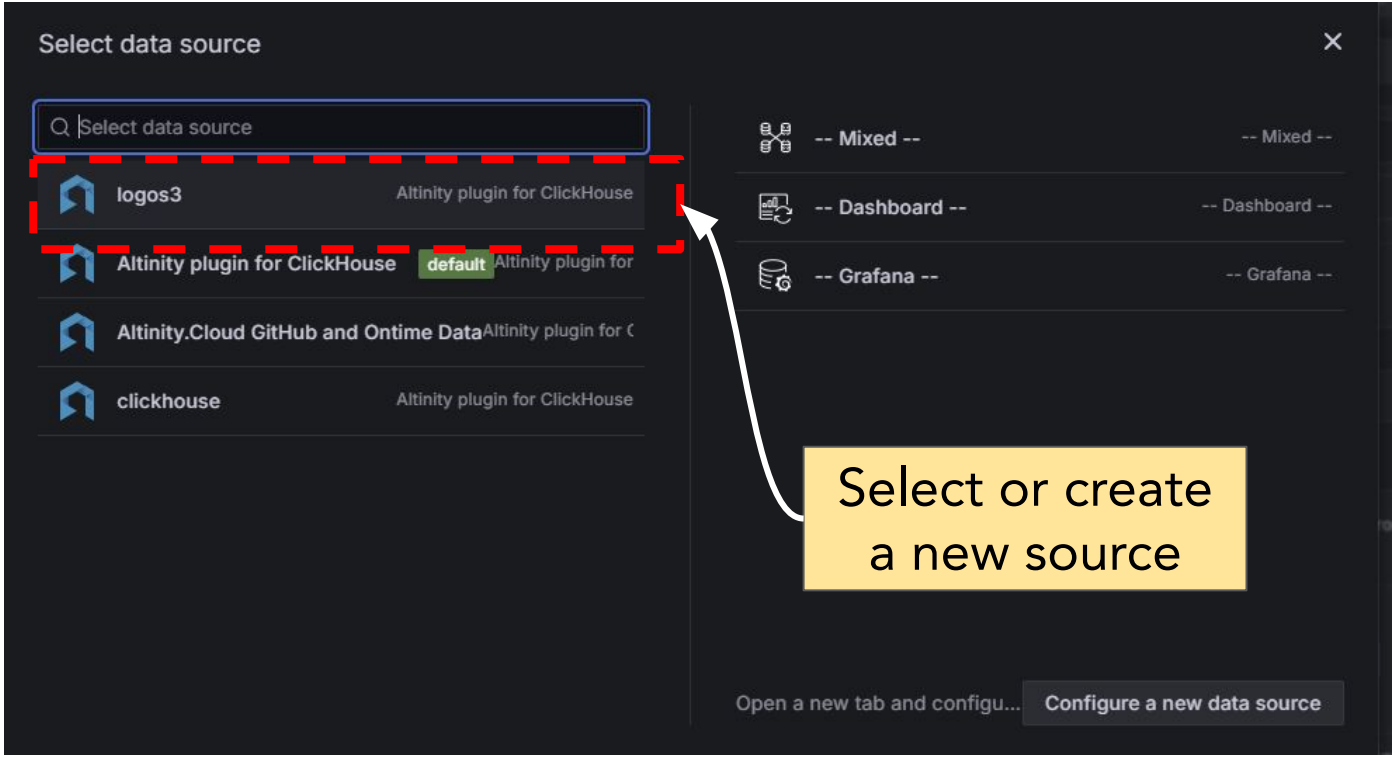
Creating a basic ClickHouse dashboard



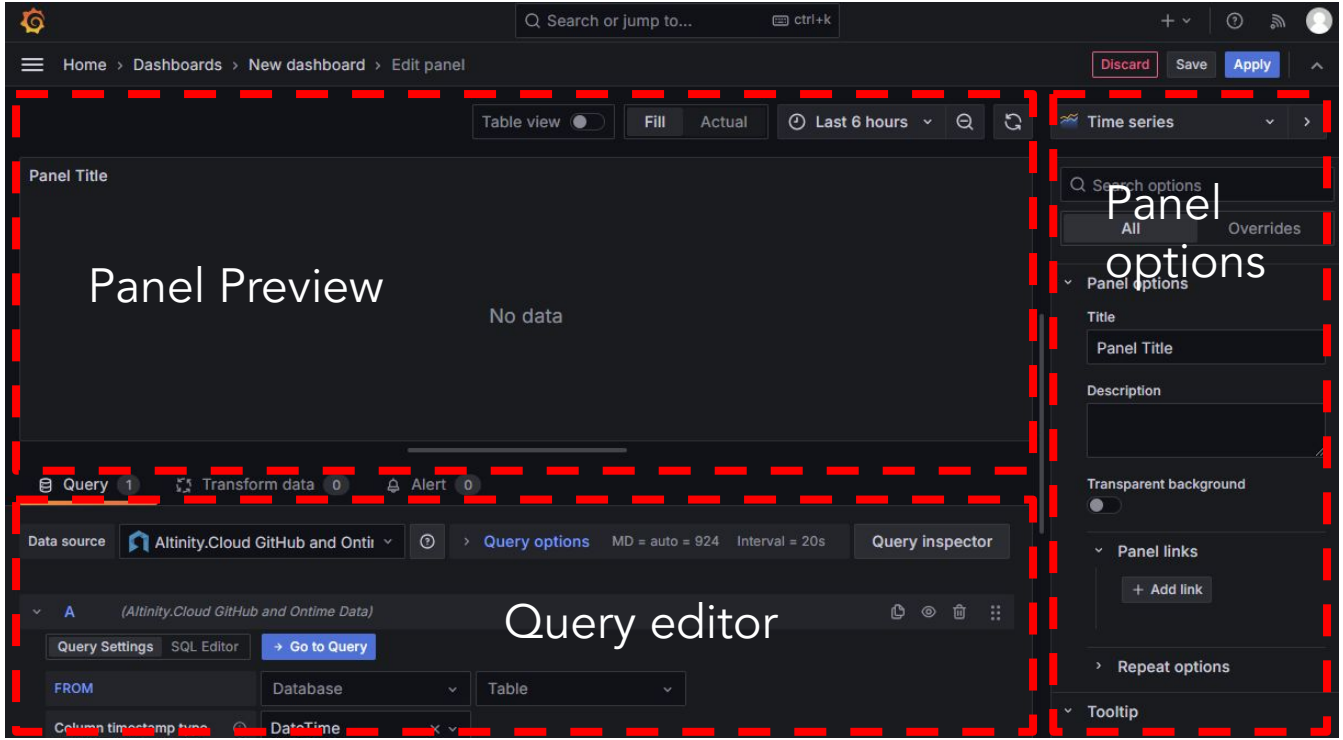
Add your first panel



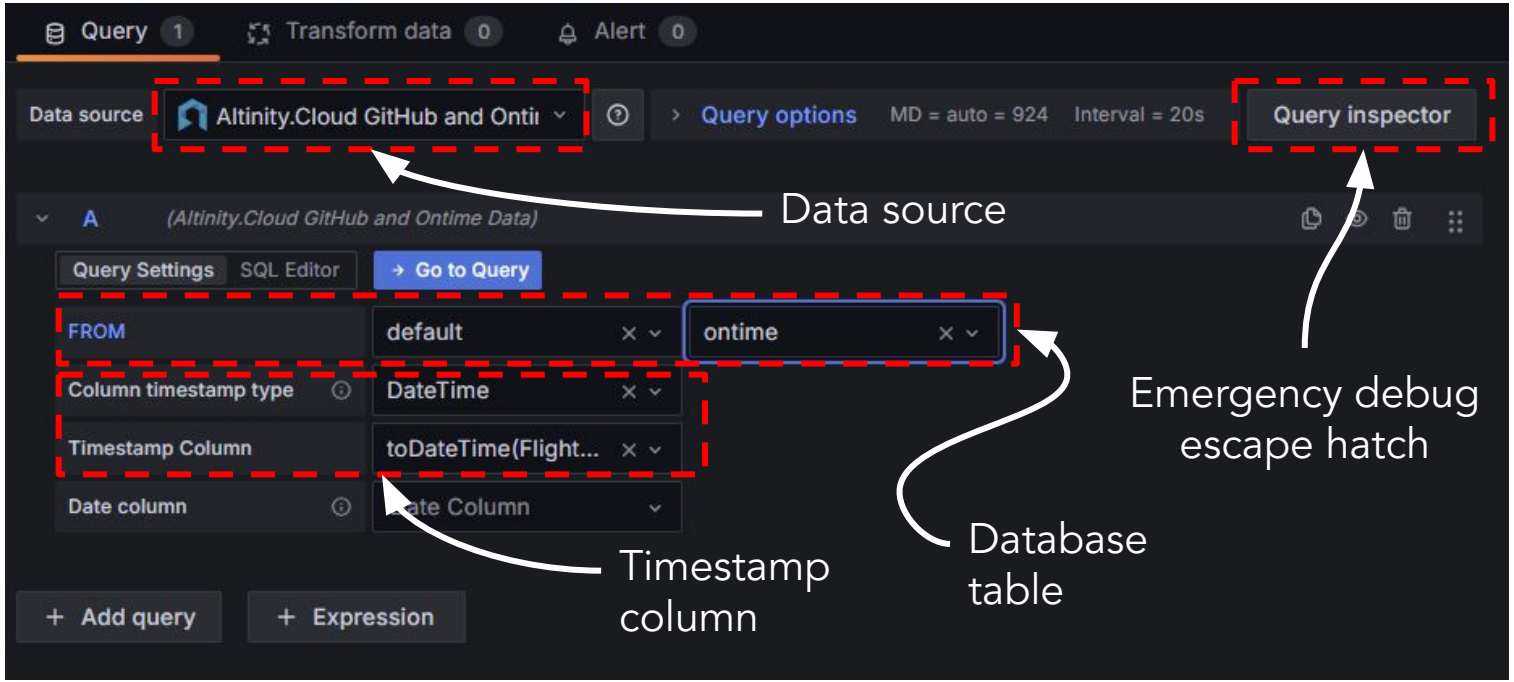
Pick a data source to feed visualizations



Creating a time series graph using ontime data



Defining the data source and time series



Editing queries in Grafana

Query 1 | Transform data 0 | Alert 0

A (Altinity.Cloud GitHub and Ontime Data)

Query Settings | SQL Editor | Run Query

```
SELECT $timeSeries as t, Carrier, count()
FROM $table
WHERE $timeFilter
GROUP BY Carrier, t ORDER BY Carrier, t
```

Query editor with macros

Extrapolation | Step | Resolution 1/1 | Format As Time series

Add metadata | Skip Comments | Round 0s | Show help

Show generated SQL

Generated SQL

Format as Time Series

Reformatted Query

```
/* grafana dashboard=New dashboard, user=1 */
SELECT (intDiv(toUInt32(toDateTime(FlightDate)), 43200) * 43200) * 1000 as t, Carrier, count()
FROM default.ontime
WHERE toDateTime(FlightDate) >= toDateTime(1577865600) AND toDateTime(FlightDate) <= toDateTime(1609488000)
GROUP BY Carrier, t ORDER BY Carrier, t
```

I broke it! Where's the debugger?

Press 'Query inspector' to debug

Press to [re-]run query

Query and errors

Data

Inspect: Airline flights

1 queries with total query time of 1.12 s

Data Stats JSON Query

Query inspector

Query inspector allows you to view raw request and response. To collect this data Grafana needs to issue a new query. Click refresh button below to trigger a new query.

Refresh + Expand all Copy to clipboard

```
▾Object
  traceId: undefined
  ▾request: Object
    url: "api/datasources/proxy/uid/ddp4ei37o6ww0d"
    method: "POST"
    data: "/* grafana dashboard=New dashboard, user=1 */
    SELECT (intDiv(toUInt32(toDateTime(FlightDate)), 43200) * 43200) * 10
    00 as t, Carrier, count()
    FROM default.ontime
    WHERE toDateTime(FlightDate) >= toDateTime(1577865600) AND toDateTime
    (FlightDate) <= toDateTime(1609488000)
    GROUP BY Carrier, t ORDER BY Carrier, t FORMAT JSON"
    hideFromInspector: false
  ▾response: Object
    ▾meta: Array[3]
      ▸0: Object
      ▸1: Object
      ▸2: Object
    ▾data: Array[6125]
      ▸0: Object
      ▸1: Object
```

Debug SQL on ClickHouse from browser

https://github.demo.trial.altinity.cloud:8443 demo ••••

```
SELECT (intDiv(toUInt32(toDateTime(FlightDate)), 21600) * 21600) * 1000 as t, Carrier, count() Flights
  FROM default.ontime
 WHERE toDateTime(FlightDate) >= toDateTime(1598276750)
 GROUP BY t, Carrier ORDER BY t, Carrier LIMIT 3
```

Run (Ctrl+Enter) ✓ Elapsed: 0.008 sec, read 5163702 rows.

t	Carrier	Flights
1598313600000	9E	538
1598313600000	AA	1293
1598313600000	AS	299

Summary of basic tips

- For Time Series Graphs:
  - Set time series type: DateTime/DateTime64/Timestamp
  - Select the column or…
    - If you are using Date types, enter something like toDateTime(FlightDate) to convert
  - Use $timeSeries and $timeFilter to get automatic handling of time ranges
  - Pick 'Time Series' visualization to display
- For all visualizations:
  - Add query with macros in editor
  - Check the generated code with "Show generated SQL" button
  - Use 'Query inspector' to debug
  - Put misbehaving queries into your favorite ClickHouse client

Advanced tricks with Grafana and ClickHouse

Altinity

Use function macros to generate queries

Same result as previous query!

```
SELECT t, groupArray((Carrier, Flights)) AS
groupArr FROM
( SELECT (intDiv(toUInt32(toDateTime(FlightDate)),
43200) * 43200) * 1000 AS t, Carrier, count()
Flights FROM default.ontime WHERE
toDateTime(FlightDate) >= toDateTime(1577865600)
AND toDateTime(FlightDate) <=
toDateTime(1609488000) GROUP BY t, Carrier ORDER BY
t, Carrier) GROUP BY t ORDER BY t
```

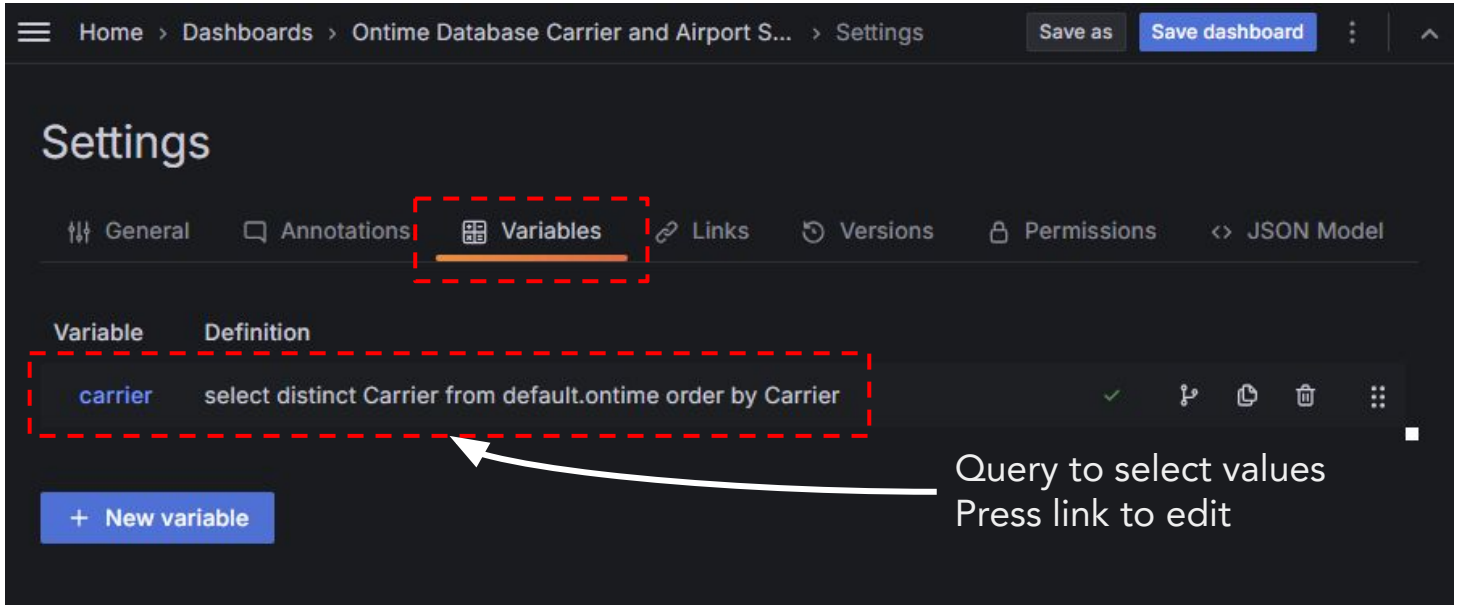
Sundry function macros for ClickHouse

$rate(columns…)	Computes change rate per interval on each column
$columns(key, value)	Produces key-value arrays per interval
$rateColumns(key, value)	Combines $rate and $columns
$perSecond(columns…)	Compute change rate per interval on counters
$perSecondColumns(key, value)	Combines $perSecond and $columns

Tip: Use function macro to generate query, then edit to suit

Defining selection variables

Enter from dashboard settings icon



Now you can use the selection in queries

```
SELECT
  $timeSeries as t,
  sum(ArrDel15)/count(*)*100.0 as "Carrier Average"
FROM $table
WHERE Carrier = '$carrier' AND $timeFilter
GROUP BY t ORDER BY t
```

Variables work in titles as well

Examples of using multiple queries for pie charts

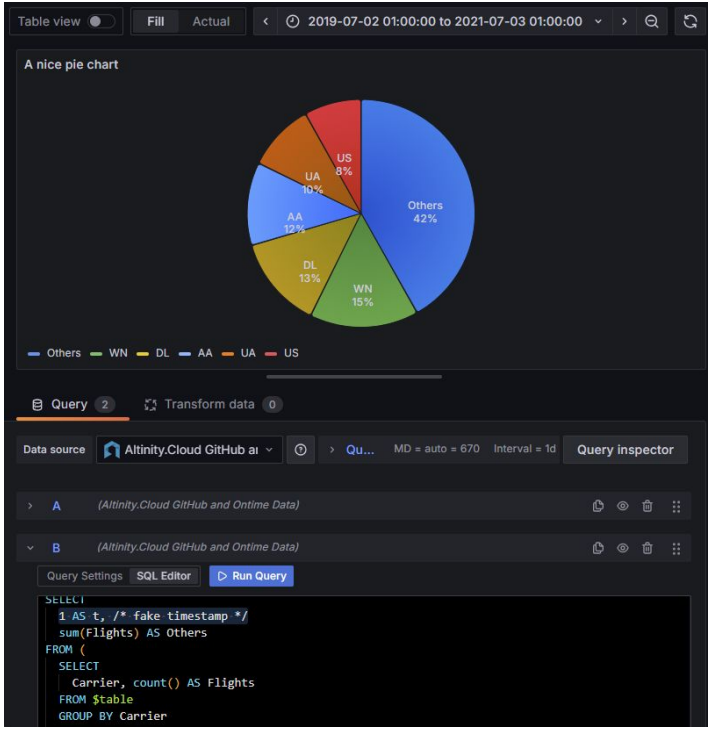
Top 5 carriers by flights

```
SELECT
  1 AS t, /* fake timestamp */
  Carrier, count() AS Flights
FROM $table
GROUP BY Carrier
ORDER by Flights DESC
LIMIT 5
```

All the rest…

```
SELECT
  1 AS t, /* fake timestamp */
  sum(Flights) AS Others
FROM (
  SELECT
    Carrier, count() AS Flights
  FROM $table
  GROUP BY Carrier
  ORDER by Flights DESC
  LIMIT 5,1000
)
```

A nice pie chart with 2 queries



More fun things to try in Grafana

- Alerts
- Geographic displays
- Flame graphs
- Traces, logs, and metrics for observability

Just to whet your appetite – Here's a traces example

Home › Dashboards › FlameGraph and Tracing support

Add | Share | Last 3 minutes

Traces Example

clickhouse: HTTPHandler 2m 2s

2024-06-20 18:11:30.022

Give feedback | Trace ID | Export

Span Filters

185 spans | Prev | Next

0µs | 30.48s | 1m 1s | 1m 31s

Service & Operation | 0µs | 30.48s | 1m | 1m

flamegraph example

Search...

Top Table | Flame Graph | Both

Symbol	Self	Total
clock_gettime	0	1
DB::DataPartsLock::DataPartsLock(std::__1::mutex&)	0	1
DB::CompressedReadBufferBase::readCompressedData(unsigne...	0	1
DB::CompressedReadBufferFromFile::nextImpl()	0	1
DB::ReadBuffer::readStrict(char*, unsigned long)	0	2
DB::SerializationLowCardinality::deserializeBinaryBulkStatePrefix...	0	1

111532 | 112 K samples (Count)

all (111532)

unknown (98103)

void* std::__1::__thread_proxy[abi:v15000]<std::__1::tuple<std::__1::unique_ptr<std::__1::__thread_struc

void std::__1::__function::__policy_invoker<void ()>::__call_impl<std::__1::__function::__default_alloc

DB::BackgroundSchedulePool::threadFunction() (98103)

unknown (98103)

Wrapping up

Roadmap for further development on Grafana

- Extended library of sample visualization examples
- Add streaming support with WATCH SQL statement
- Improvements and bug fixes in code editor
- Improvements performance for ad hoc query selector
- Checkout https://github.com/Altinity/clickhouse-grafana/issues for our public roadmap

It's open source! Add your own issues and PRs!!

More information

- Altinity blog: https://altinity.com/blog
- Altinity Grafana plugin docs: https://grafana.com/grafana/plugins/vertamedia-clickhouse-datasource
- Code: https://github.com/Altinity/clickhouse-grafana
- Altinity.Cloud: https://altinity.com/cloud-database/
- Grafana Cloud: https://grafana.com/products/cloud/
- Meet us on Slack!
  - Invite link at https://altinity.com

Thank you!

https://altinity.com

Altinity.Cloud
Altinity Stable Builds for ClickHouse
Altinity Kubernetes Operator
(and of course)
Altinity Grafana plugin for ClickHouse!