ClickHouse Cloud Live

Feb 8, 2024

ClickHouse

Agenda (60 mins)

Welcome (Tanya) - 3 mins

Cloud Platform Update (Krithika) - 5 mins

Cloud Console Update (Zach, Cristina) - 15 mins

Security & Reliability Update (Leticia) - 7 mins

Core Update (Melvyn and/or Alexey) - 10 mins

Integrations Update (Ryadh) - 10 mins

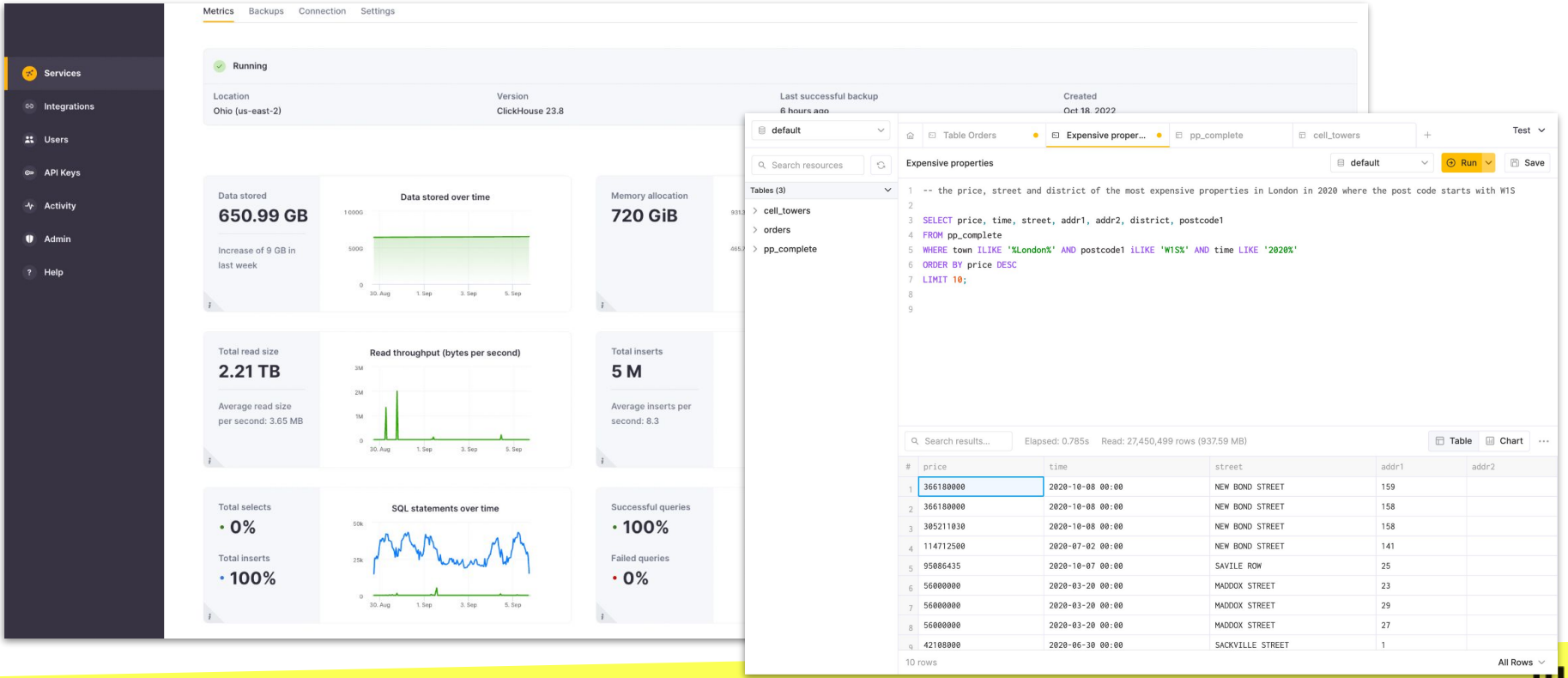
Q&A - 10 mins

For live social update links/polls, please add them to your section here.

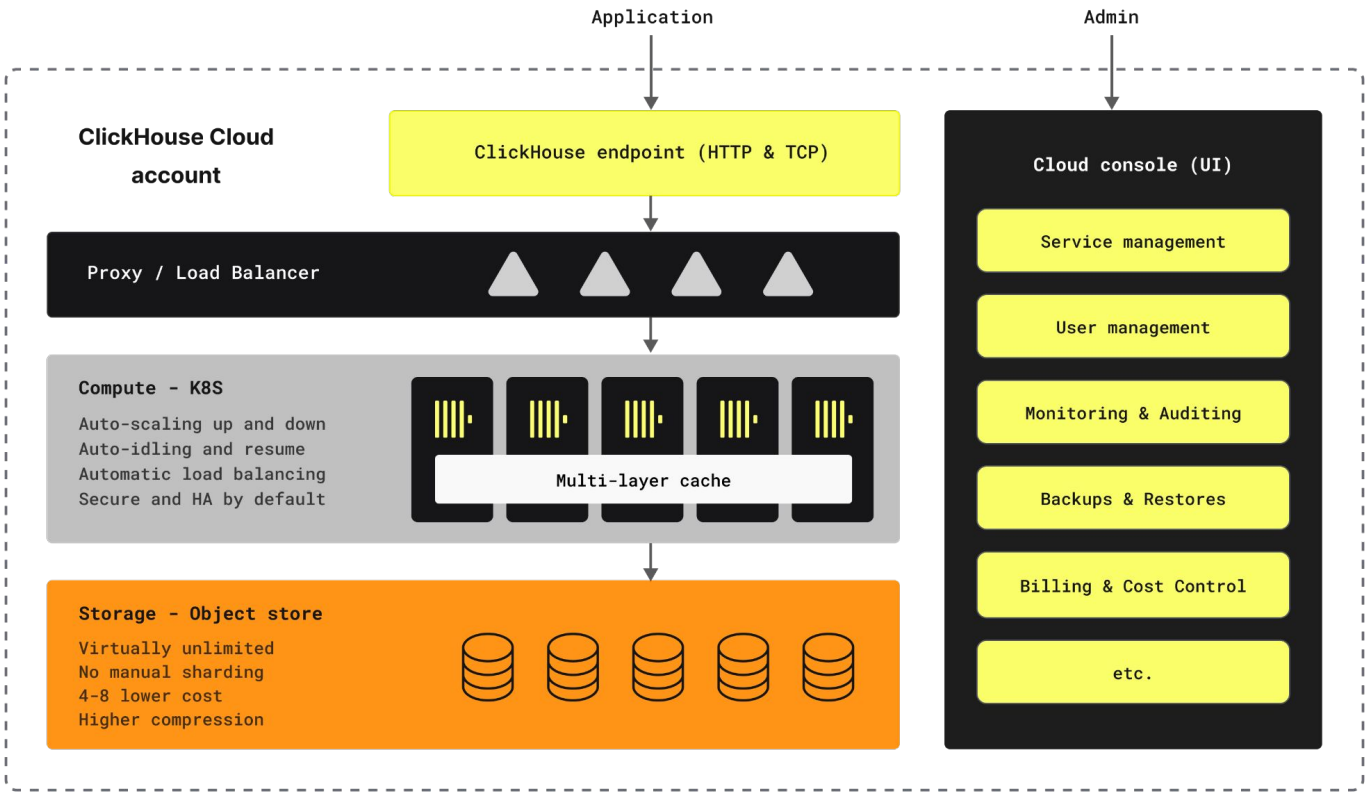
In red, topics that could be removed to save time

ClickHouse Cloud

Soon



ClickHouse Cloud Architecture



Cloud Pricing

Development

Great for smaller workloads and starter projects

$50 - $193 / Month

- ✓ Up to 1 TB storage
- ✓ 16 GiB total memory
- ✓ Burstable CPU
- ✓ Backups every 24h, retained 1 day
- ✓ Replicated across 2 AZs
- ✓ Expert support with 24h response time

Storage	Compute
$35.33 per TB/mo	$0.2160 per unit/hr

Production

Designed to handle production workloads

Usage Based

- ✓ Unlimited storage
- ✓ 24GiB+ total memory
- ✓ Dedicated CPU
- ✓ Backups every 24h, retained 1 day
- ✓ Replicated across 3 AZs
- ✓ Expert support with 1h SLA
- ✓ AWS private link support
- ✓ Automatic scaling

Storage	Compute
$47.10 per TB/mo	$0.6888 per unit/hr

Dedicated

Designed for the most demanding latency-sensitive workloads

Capacity Based

- ✓ Unlimited storage
- ✓ Custom compute options
- ✓ Dedicated environment
- ✓ Advanced isolation and security
- ✓ Customized backup controls
- ✓ Scheduled upgrades
- ✓ Uptime SLAs
- ✓ Consultative migration guidance
- ✓ Dedicated support engineers

Contact us

Available in **AWS** and **GCP**, both directly and through AWS & GCP Marketplaces

Agenda

01 **Cloud Platform**
Krithika

02 **Cloud Console**
Zach, Cristina

03 **Security & Reliability**
Leticia

04 **Core**
Melvyn, Alexey

05 **Integrations**
Ryadh

Cloud Platform Update

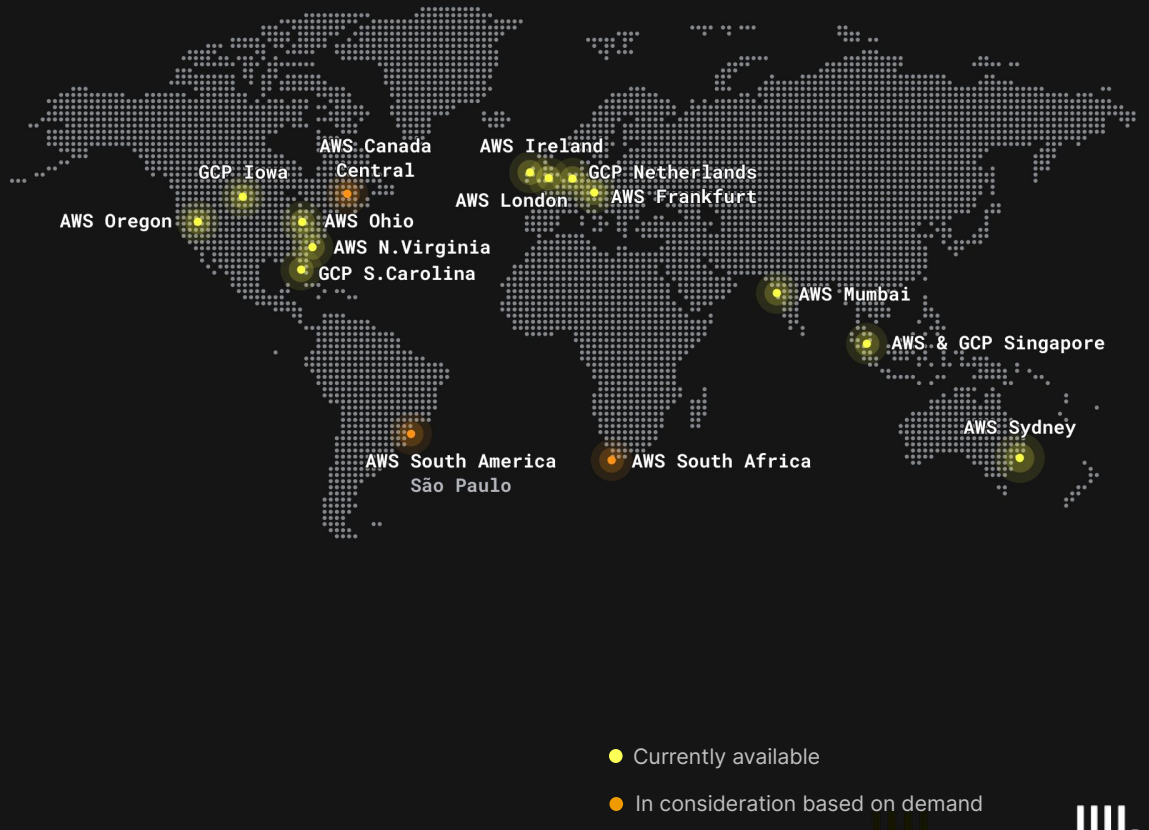
Krithika

ClickHouse

Supported Cloud Providers and Regions

- Available direct and via Marketplace
- Supported in 9 regions across United States, United Kingdom, Europe, Asia and Australia

- Available direct and via Marketplace
- Supported in 4 regions across United States, Europe and Asia



Cloud Provider Expansion

Microsoft Azure

Work in progress

Private Preview CYQ1 2024

Tracking interest in the waitlist

BYOC - AWS

Planning in progress

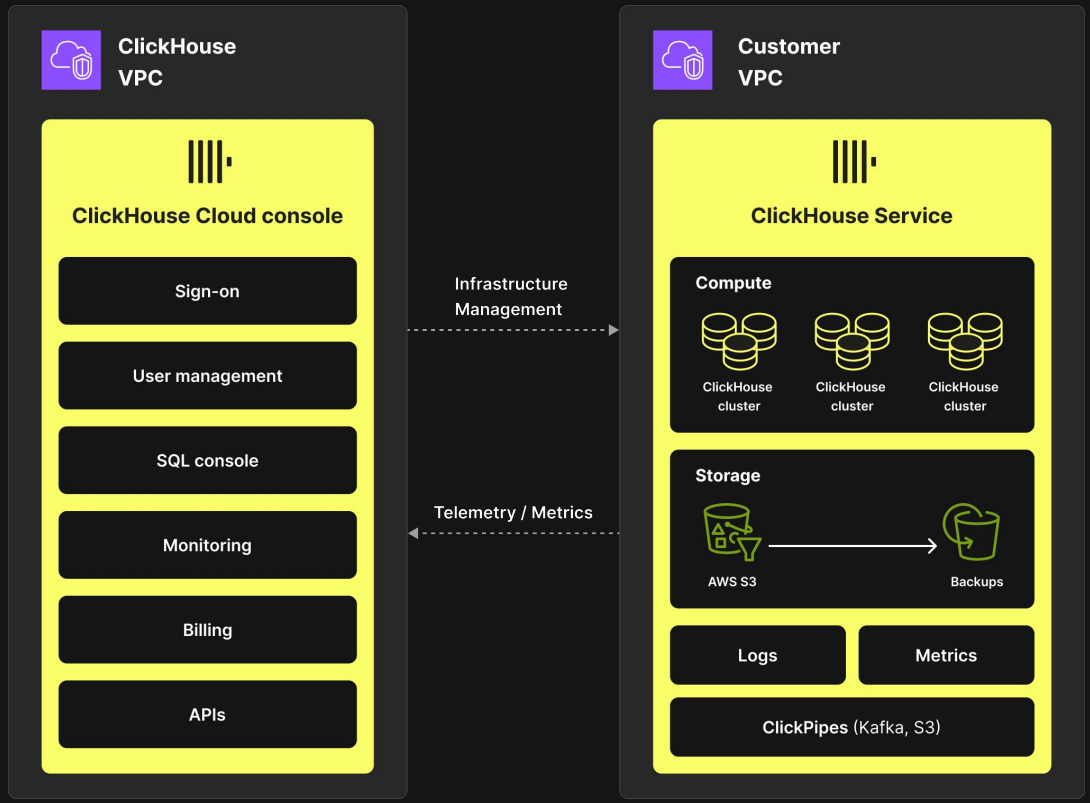
Private Preview CYQ2 2024

Recruiting design partners

Note: All timelines are tentative and subject to change.

BYOC Architecture(AWS)

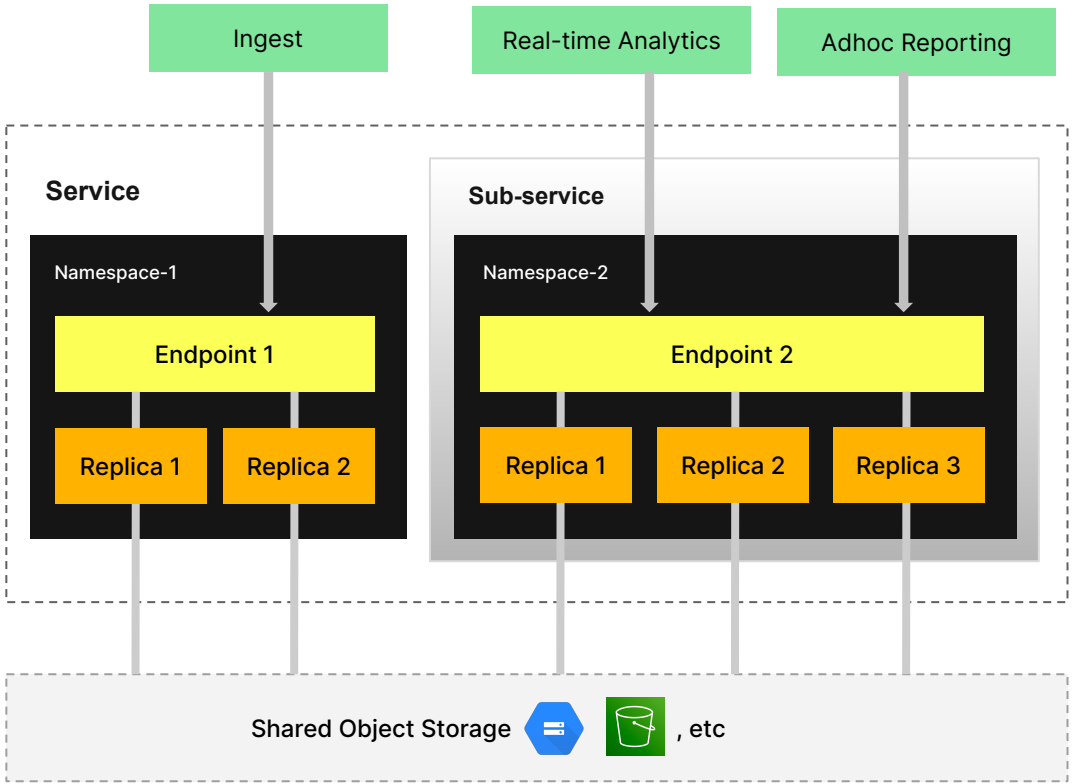
- Compute and Storage will reside in Customer' VPC
- Only service health metrics will be accessible from the ClickHouse VPC for Monitoring and Management
- All web assets will be hosted on ClickHouse VPC - includes User Management, Service Configuration, SQL Console,..
- Secure by design



Compute-Compute Separation

With Compute-Compute separation, services can allocate dedicated compute for specific operations - eg: Streaming Ingest vs Adhoc Reporting ,while sharing the same storage

- Compute units can be scaled independently
- Eliminates bottlenecks due to resource contention



Cloud Console Update

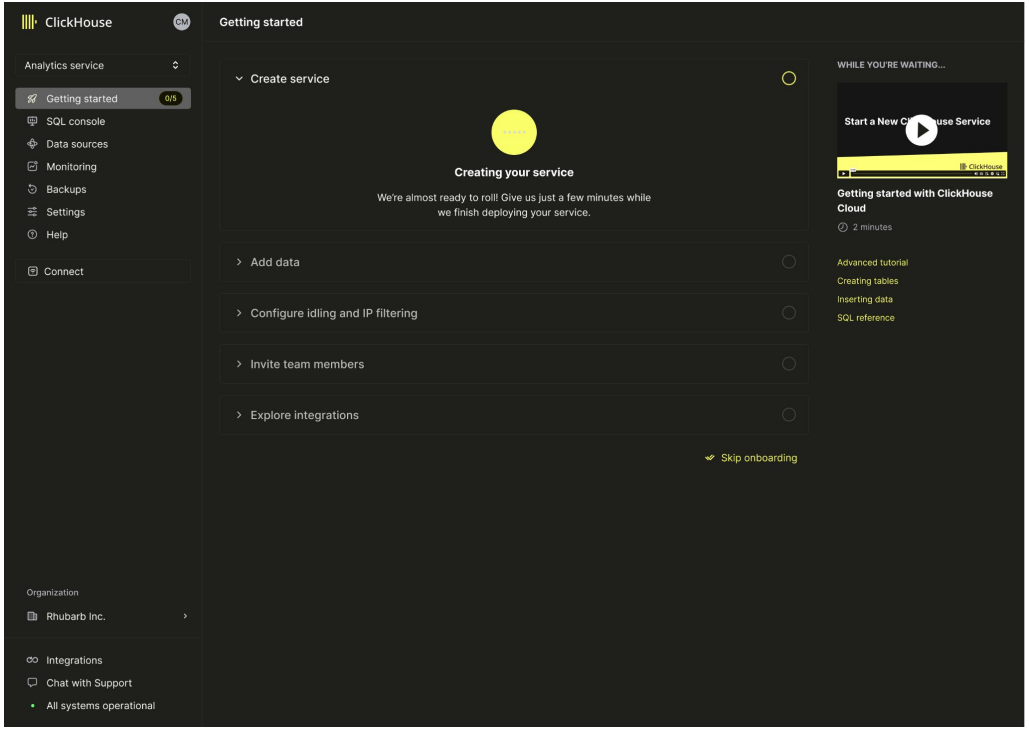
Zach, Cristina

ClickHouse

Unified Console Demo

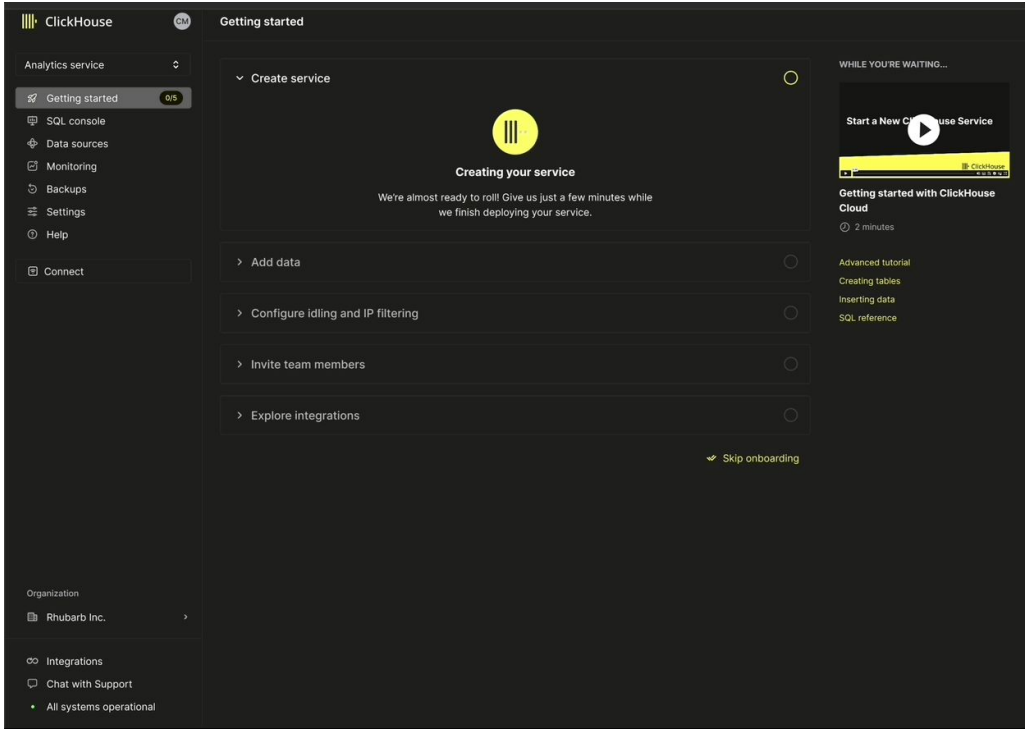
New onboarding flow

- Guided steps to help users be successful using ClickHouse Cloud
- Dedicated entry in the sidebar
- 5 steps, listed in our preferred order, but can be executed in any order
- Complementary info (videos/links) in a dedicated section on the right
- Can be skipped



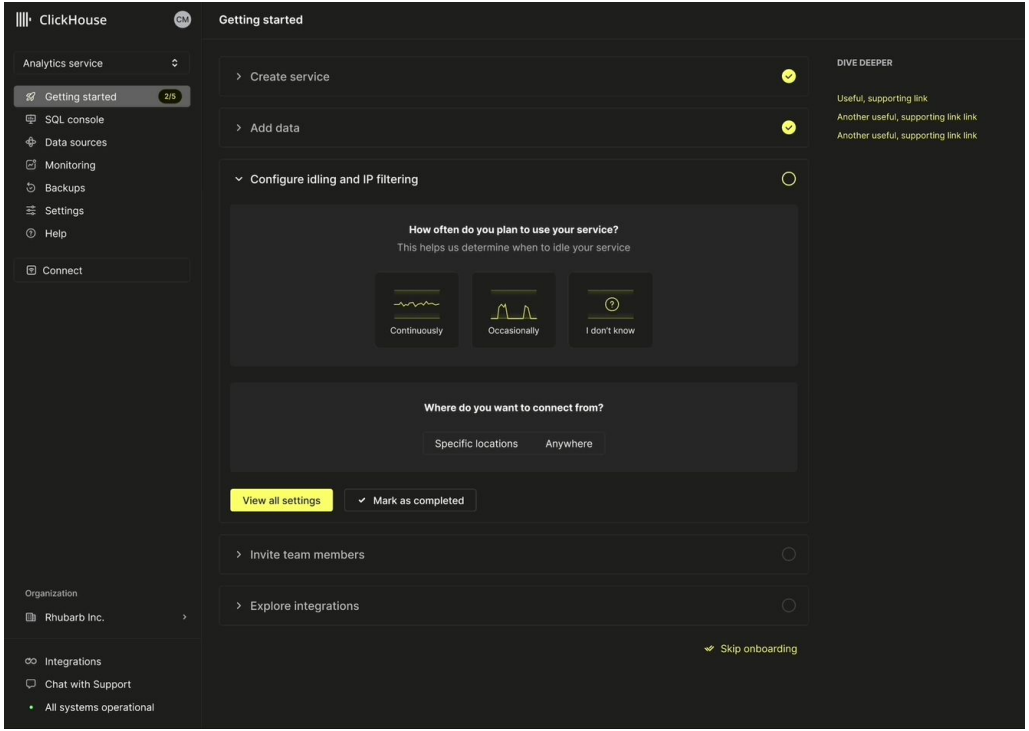
Adding data

- Only with data can one experiment the true potential of ClickHouse
- 4 main ways to add data:
  - ClickPipes (Kafka, S3, GCS, etc)
  - File Upload
  - Connect with your app/language client
  - Start with an empty table
- 2 secondary options:
  - Link to the “Data sources” page
  - View integrations



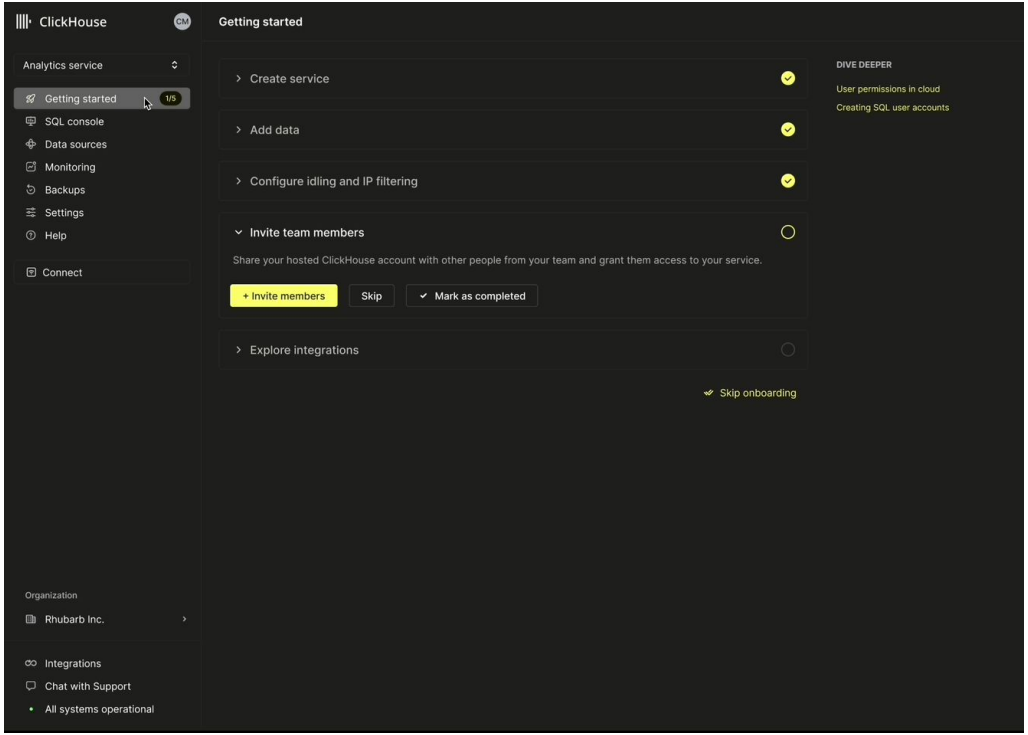
Shed light on important settings

- Configure the right idling period based on usage patterns
- Configure IP filtering for an extra layer of security
- Link to the Settings page



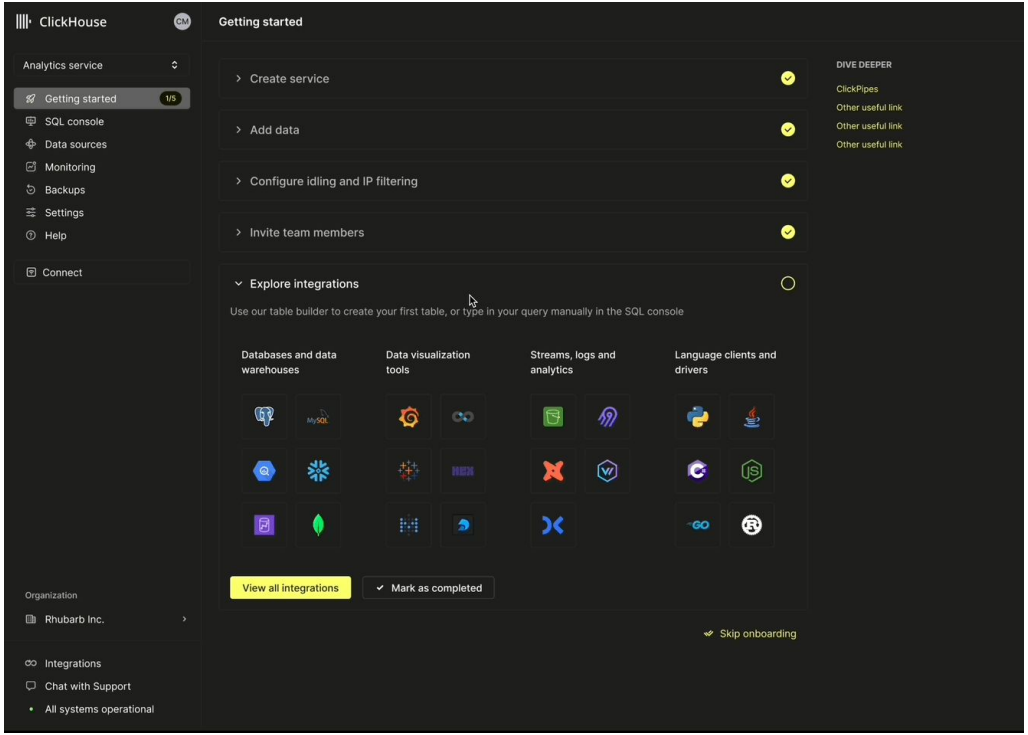
Invite team members

- Invite team members as Admins or Devs to your organization
- Can be skipped



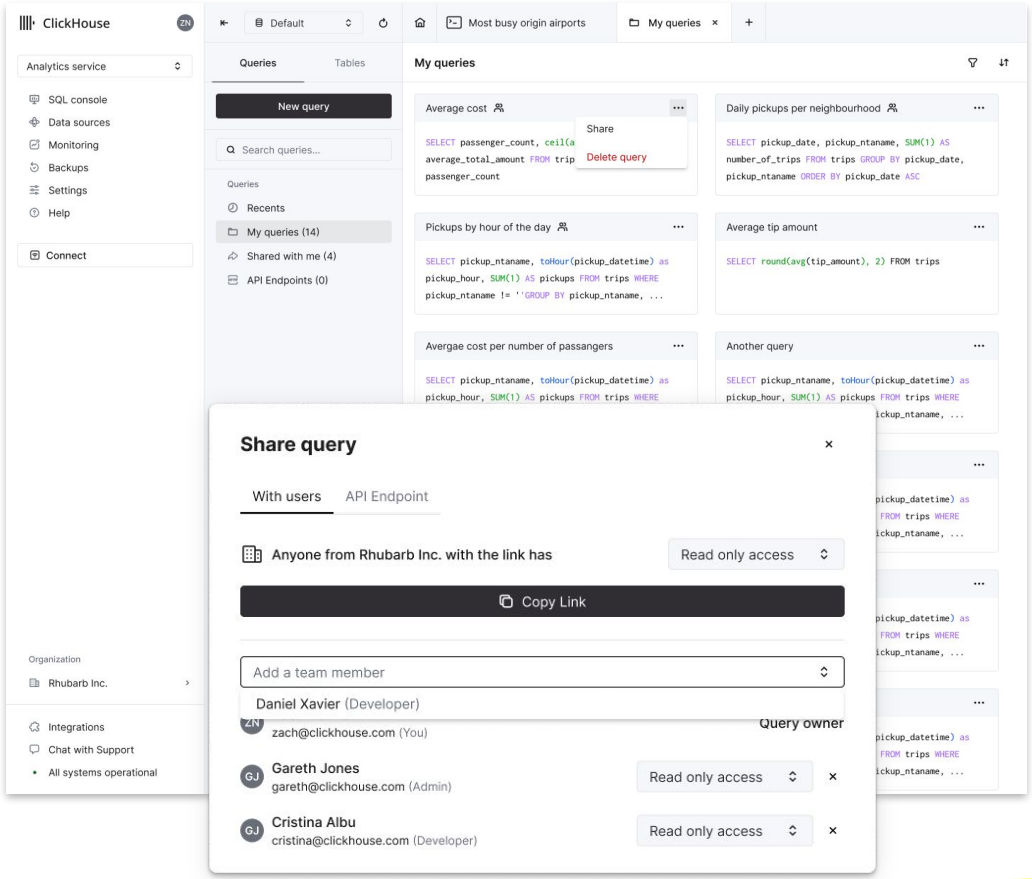
Explore integrations

- Help users discover our vast ecosystem of integrations
- 4 groups:
  - Databases and data warehouses
  - Data visualization tools
  - Streams, logs and analytics
  - Language clients and drivers



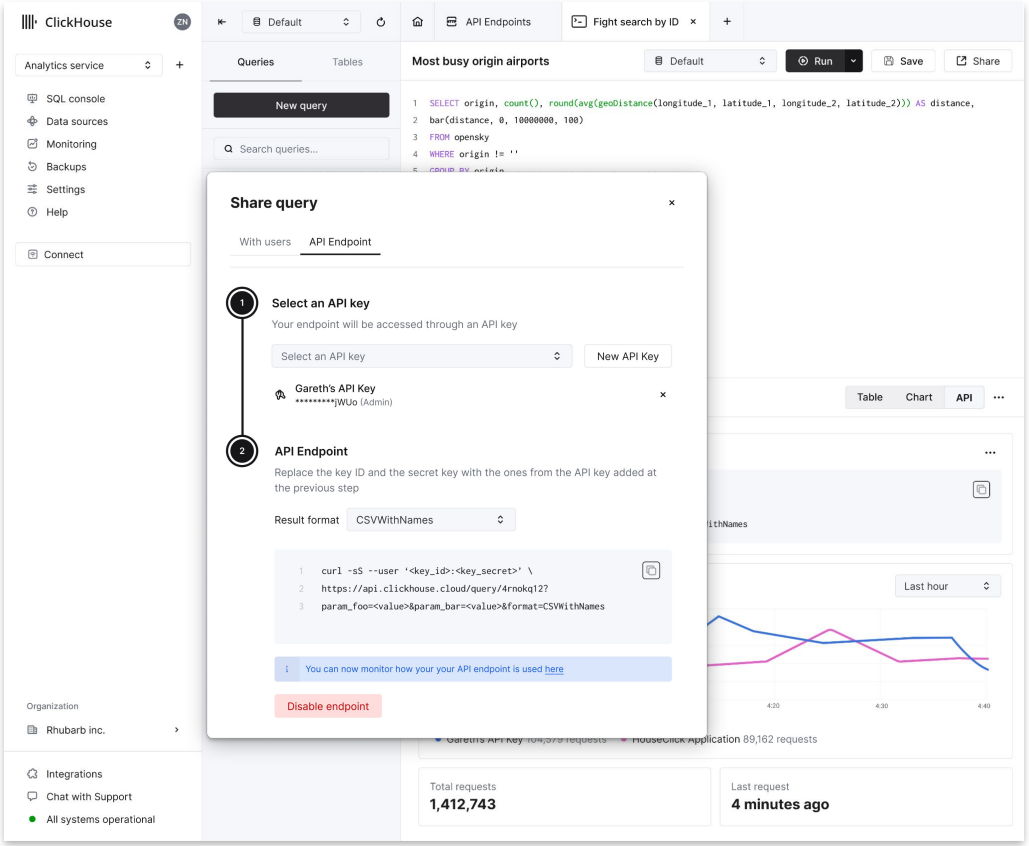
Query sharing and access control

- Share saved queries with roles or individual users on your team
- Configure access levels for each role or user (full access, read-only, etc.)



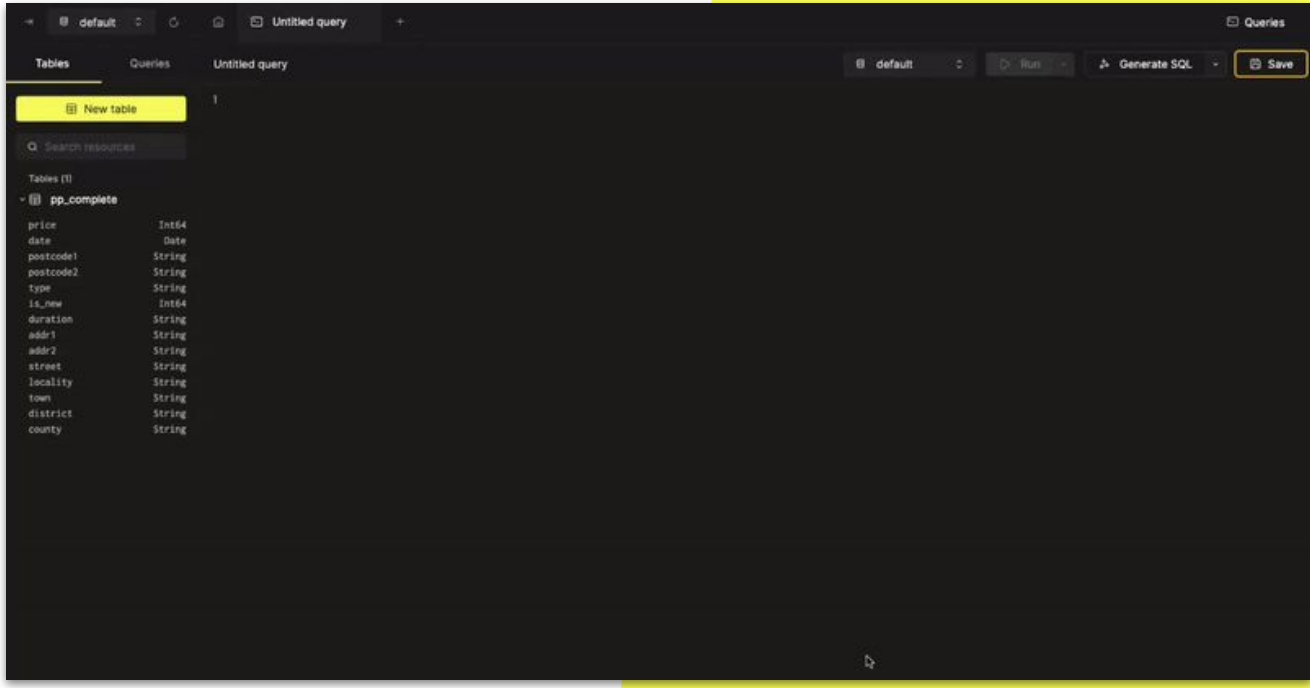
Query API endpoints

- Speed up development by building queries in the SQL console and exposing them as API endpoints
- Control access using existing ClickHouse Cloud OpenAPI keys
- Monitor endpoint request volume



Copilot-style SQL suggestions

Intelligent in-line ClickHouse SQL suggestions as you write your queries



Security & Reliability Update

Leticia

ClickHouse

Added authentication options

Strong authentication

- NIST 800-63 compliant passwords
- Multi-factor authentication options
- Social login with Google

Enterprise Single Sign-On (SSO)

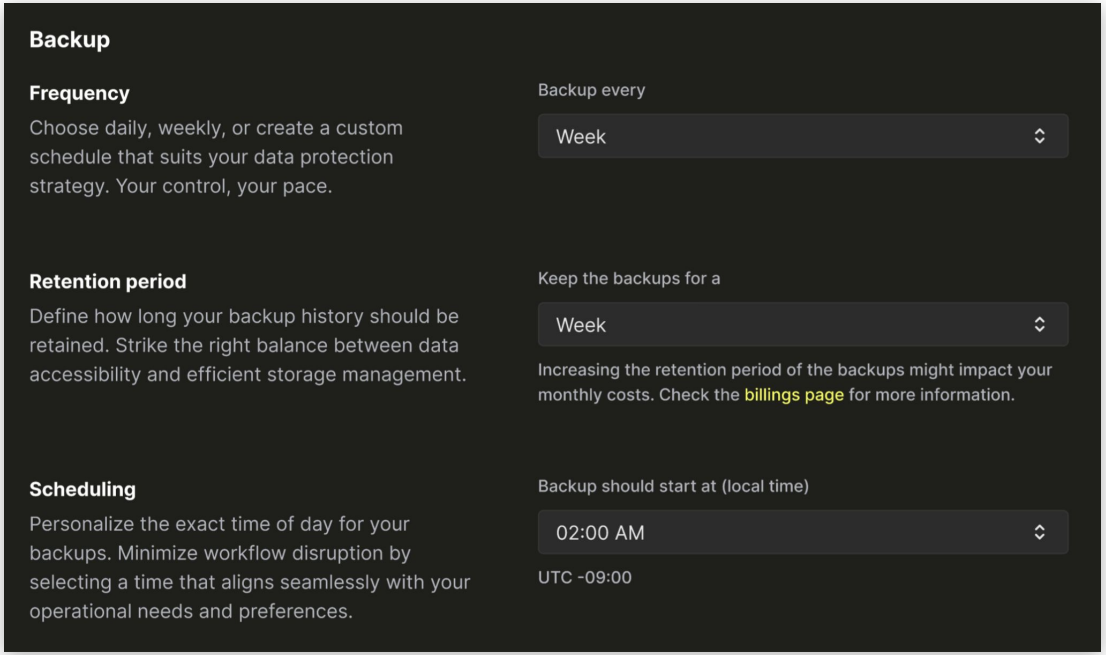
- SAML enabled (limited release)

More to come

- Social login with Microsoft
- Social login with GitHub
- SAML broad release
- SCIM enabled SSO

More control for backups

- On-demand backups
- Custom configurations for frequency and retention
- Custom scheduling
- Save to your own S3 bucket



Bring your data, all of it

HIPAA
Protected Health Information

- Secure connections with IP filters
- SHA256 hashed passwords
- Customer managed encryption
- Detailed logging

Targeting H1 2024

Federal
U.S. Federal Government

- Self-managed install
- Separation of compute + storage
- FIPS 140-3 encryption
- APIs to enable automation
- FedRAMP in cloud proposed

Planning & Scoping

PCI
Credit Card Information

- Designed for PCI 4.0
- Improved user management
- Outbound traffic control
- Granular encryption options

Planning & Scoping

Note: All timelines are tentative and subject to change.

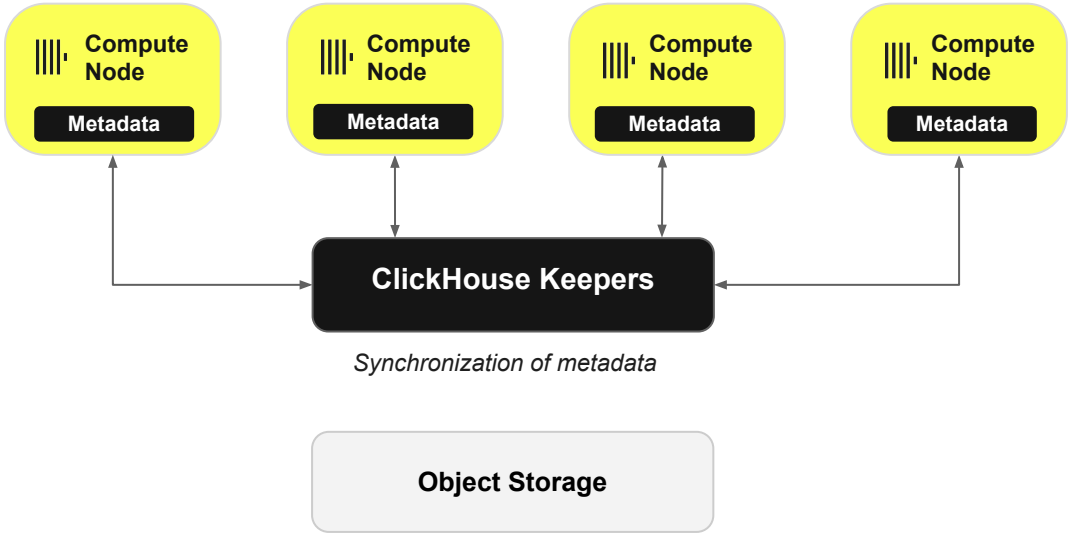
Core Update

Melvyn, Alexey

ClickHouse

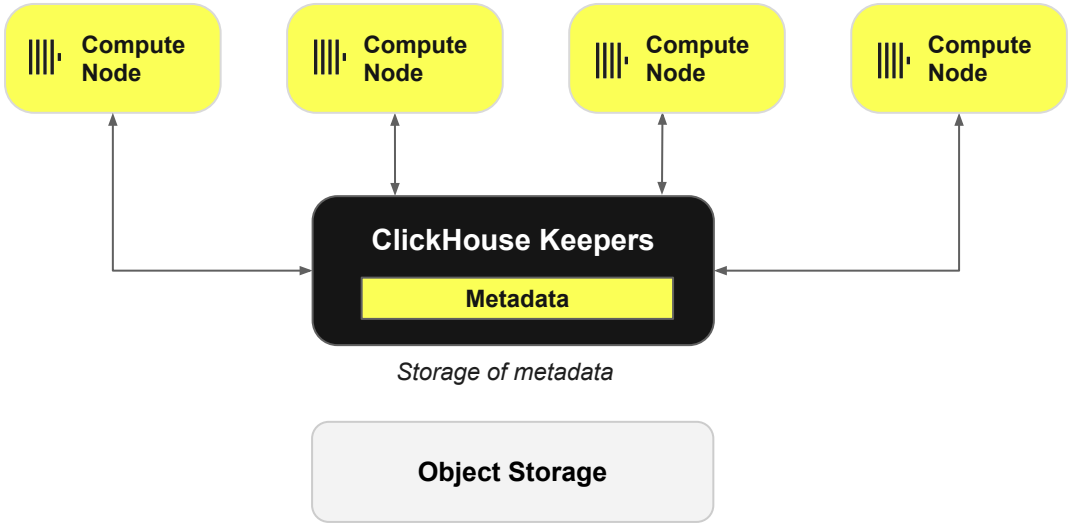
Before SharedMergeTree

- Metadata needs to be replicated to every clickhouse-server
- Adding or removing a node is a slow operation
- The higher the number of servers is, the more replication of metadata is needed, it impacts:
  - ✓ insert throughput
  - ✓ background operations



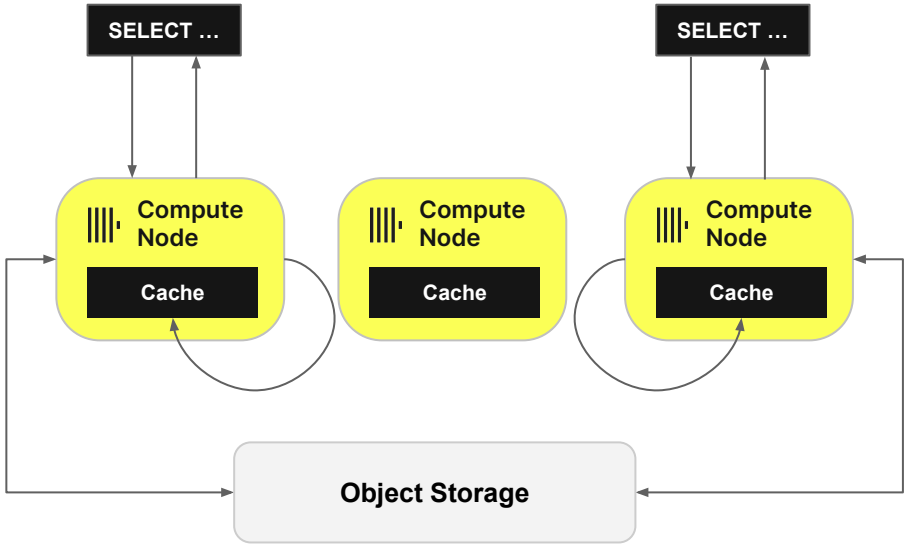
Introducing SharedMergeTree

- SharedMergeTree a cloud native replacement of ReplicatedMergeTree
- Faster scale-up and down operations
- Improved throughput for:
  - ✓ Background Merges
  - ✓ Inserts
  - ✓ Mutations



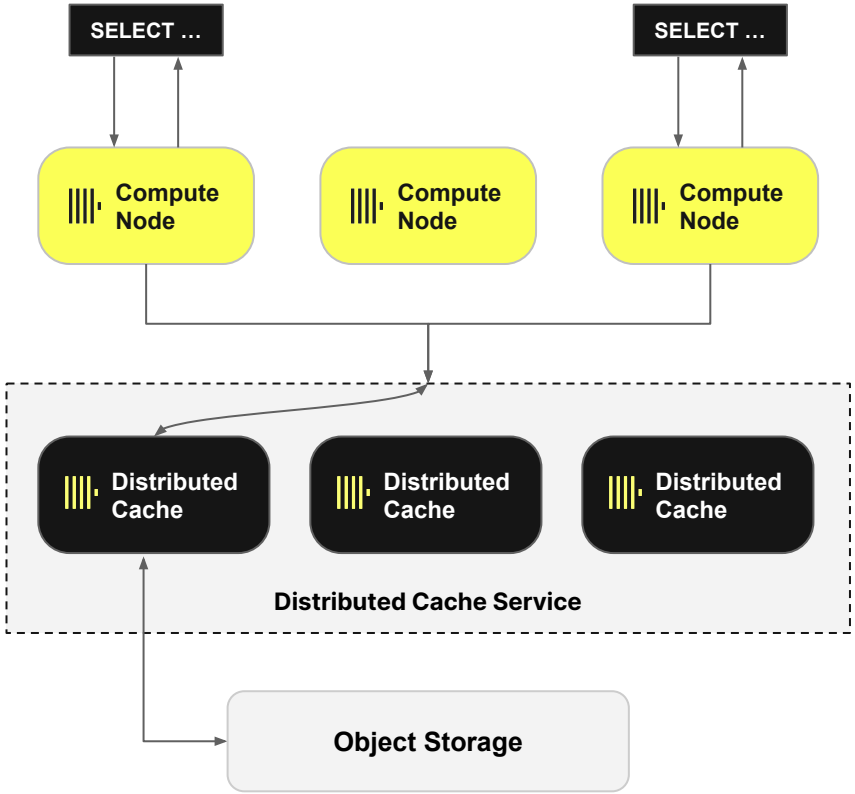
The Current State of our Cache

- Each node has their own cache
- The content of the cache varies depending on the node
- Cache needs to be repopulated after each restart, scale up/down, etc...



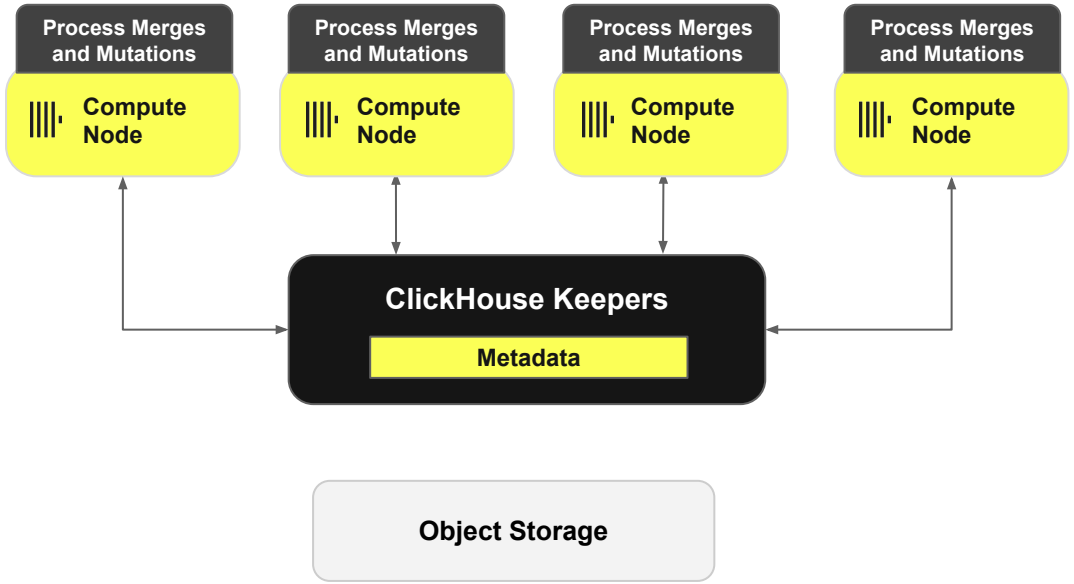
Distributed Cache

With a new distributed cache service, the cache is now separated from the compute node. It means that it does not need to be repopulated after scale up/down, restart.



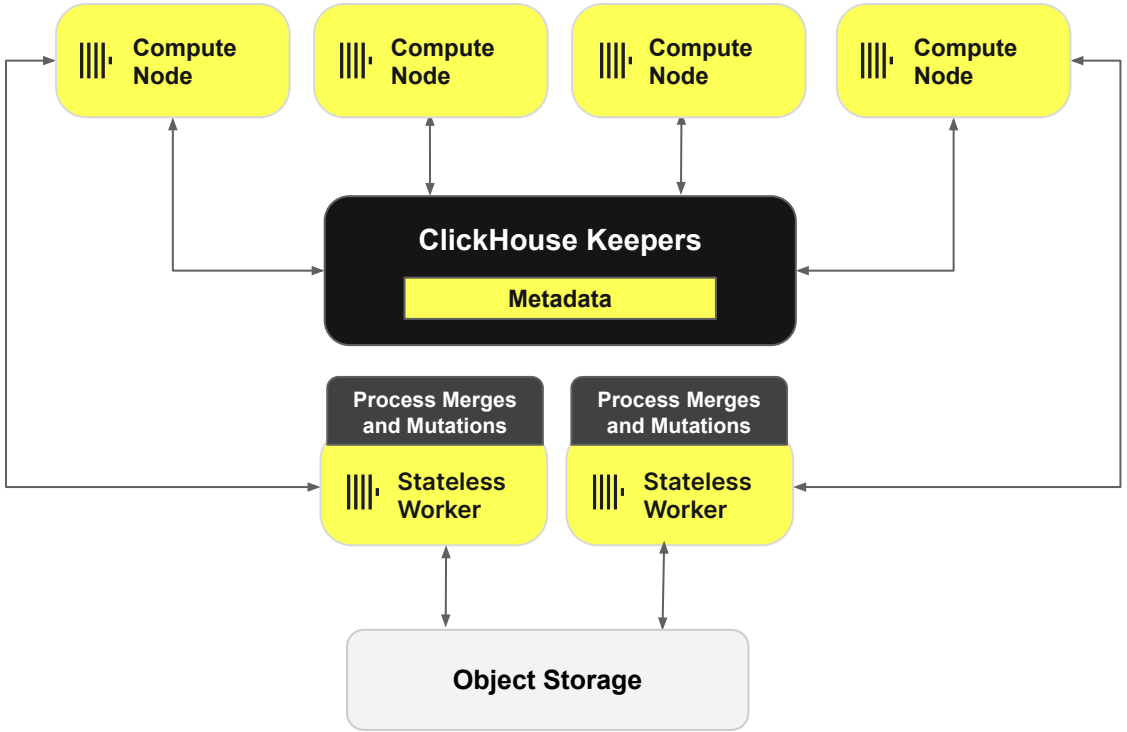
The Current State of Background Operations

- Background operations can add surprising load to a server executing end-user queries
- Resource utilization is not directly controlled by the end-user



Stateless Workers (R&D)

- Better separation of resource usage
- Compute nodes are only used for user workload
- Stateless worker are only used for background operations



Advancements In Performance

How *fast* is ClickHouse Cloud?

Insert Performance

How quickly can you insert data?

One year ago, we created a task, "Insert Scalability Challenge" to test the new **SharedMergeTree** engine.

The goal was to insert **400 billion records as quickly as possible** and find all possible ways how it could break.

Insert Scalability Challenge

```
CREATE TABLE wikistat
(
    time DateTime CODEC(Delta, ZSTD(3)),
    project LowCardinality(String),
    subproject LowCardinality(String),
    path String CODEC(ZSTD(3)),
    hits UInt64 CODEC(ZSTD(3))
)
ORDER BY (path, time);

INSERT INTO wikistat WITH
    parseDateTimeBestEffort(extract(_file, '^pageviews-([\d\-]+)\.gz$')) AS time,
    splitByChar(' ', line) AS values,
    splitByChar('.', values[1]) AS projects
SELECT
    time,
    projects[1] AS project,
    projects[2] AS subproject,
    decodeURLComponent(values[2]) AS path,
    CAST(values[3], 'UInt64') AS hits
FROM s3Cluster(default, 's3://clickhouse-public-datasets/wikistat/original/pageviews*.gz', LineAsString)
WHERE length(values) >= 3;
SETTINGS parallel_distributed_insert_select = 1;
```

Insert Scalability Challenge: Requirements

1. The cluster successfully scales to 100 replicas spread across different physical hosts.
2. It does not break during the insertion. Insert query should finish successfully.
3. Health checks work during the insertion, and connections do not block.
4. The speed remains mostly uniform during the insertion.
5. All replicas participate in parts merging during the insertion.
6. The number of active data parts after the insertion is less than 5000.
7. The service does not take long to restart if you do it just after the insertion.
8. Real time queries finish in less than a second during the whole insertion process.
9. Backups work on this service.
10. The progress bar and metrics work correctly. :)

Insert Scalability Challenge

April 2023

- 👍 **400 bn rows inserted in 3230 sec** on 50 replicas
- 👍 **125 million rows/sec** insertion speed
- 👎 **didn't use all the resources** for merges
- 👎 has **too many parts** after then insertion
- 👎 the startup time is **over 6 minutes**
- 👎 SELECTs are **slow during the insertion**
- 👎 there was **no progress bar** for a query

Insert Scalability Challenge

July 2023

We made it harder:

- fault injection in the network between clickhouse-server and clickhouse-keeper;

And it didn’t pass :)

Insert Scalability Challenge

January 2024

- 💪 440 bn rows inserted in **890s on 100 replicas**
- 💪 **498 million rows/sec** insertion speed
- 💪 all replicas participate in merges
- 💪 keeper is chilling during the load
- 💪 the startup time is **2 seconds**
- 💪 realtime SELECTs are real
- 💪 the progress bar works :)

Insert Scalability Challenge

It is a pretty nice test, but customers' workloads are even better.

A case study from a customer:

- migrated **1.5 PiB** of (compressed) data **in two weeks**;
- used **20 replicas** during the insertion;
- after the initial import, scaled down to 10 replicas;
- **10 replicas serve all** the real-time load, keeping the costs low;

Select Performance

How quickly do you get the answer for a SELECT query?

— Can we get a sub-second performance on tens of billions of records scanned?

How well does ClickHouse Cloud scale?

— Does it scale to hundreds of replicas?

Does it maintain low latency for massively distributed queries?

— Does it feel snappy with a **separation of storage and compute** with object storage + multi-level cache on SSD and memory?

The best way to answer these questions is to create a demo.

Alexey DEMO

Select Performance

Speed up for parallel replicas:

Better cache locality

— it reads the same ranges from the same replicas if they are available.

Better tail latency

— faster replicas will steal tasks from slow ones.

Example from a cluster with 100 pods, 30 CPU cores each:

Processed 38.23 billion rows, 570.69 GB (**24.51 billion** rows/s., **365.90** GB/s.)

Performance of Background Operations

ClickHouse can insert data so fast, but what does it have to do to accommodate this data?

Insertions have to be coordinated in Keeper

— so now we optimized Keeper for memory usage!

Inserted data parts have to be merged in time

— a cluster can handle over 1000 merges in parallel;
— background merges are optimized for memory usage.

Bonus: adaptive mode for asynchronous inserts (coming soon).

ClickHouse Community Call (Monthly)

Highlights from the last few months 🚀

New features

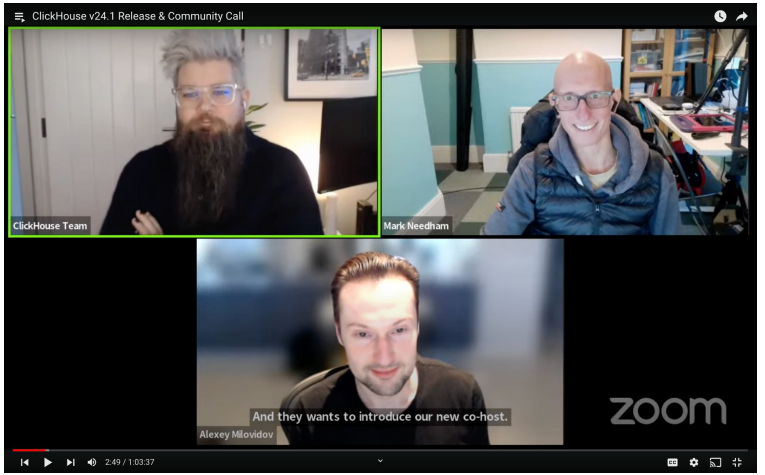
- Type inference for JSON
- S3Queue Table Engine
- Asynchronous loading of tables
- PASTE JOIN
- Temporary user credentials

Performance improvements

- Optimization for SELECT FINAL
- Optimization for external aggregations
- Improvement on the pipeline for short queries
- Condition pushdown for ORC and Parquet
- Faster window functions
- Optimization for reading from cache
- Adaptive timeout for S3

SQL compatibility improvements

- New functions: TO_DAYS, addDate, subDate
- toDayOfWeek, toYearWeek, toWeek now support String argument
- Aliases (in order to avoid breaking migrations): STD
- Support for BI tools: Information schema & prepared statements



Join us on the last Thursday each month for:
https://www.youtube.com/c/ClickHouseDB

- Coverage of new features in monthly core database releases
- Cool demos, discussion about the power of SQL, and database trivia
- Sneak peak into roadmap and AMA with Alexey, creator of ClickHouse

Integrations Update

Ryadh

ClickHouse

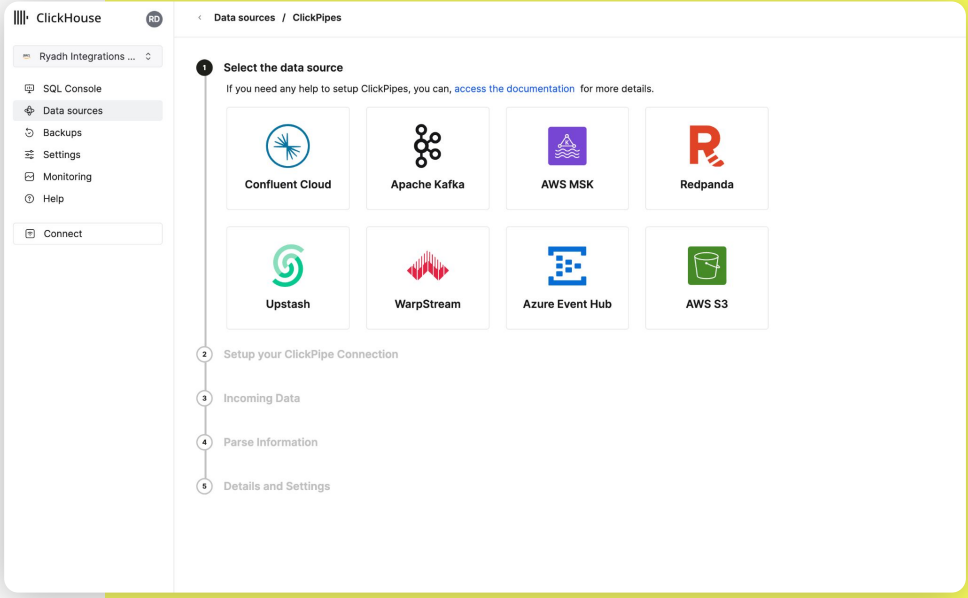
ClickHouse

ClickPipes Expansion

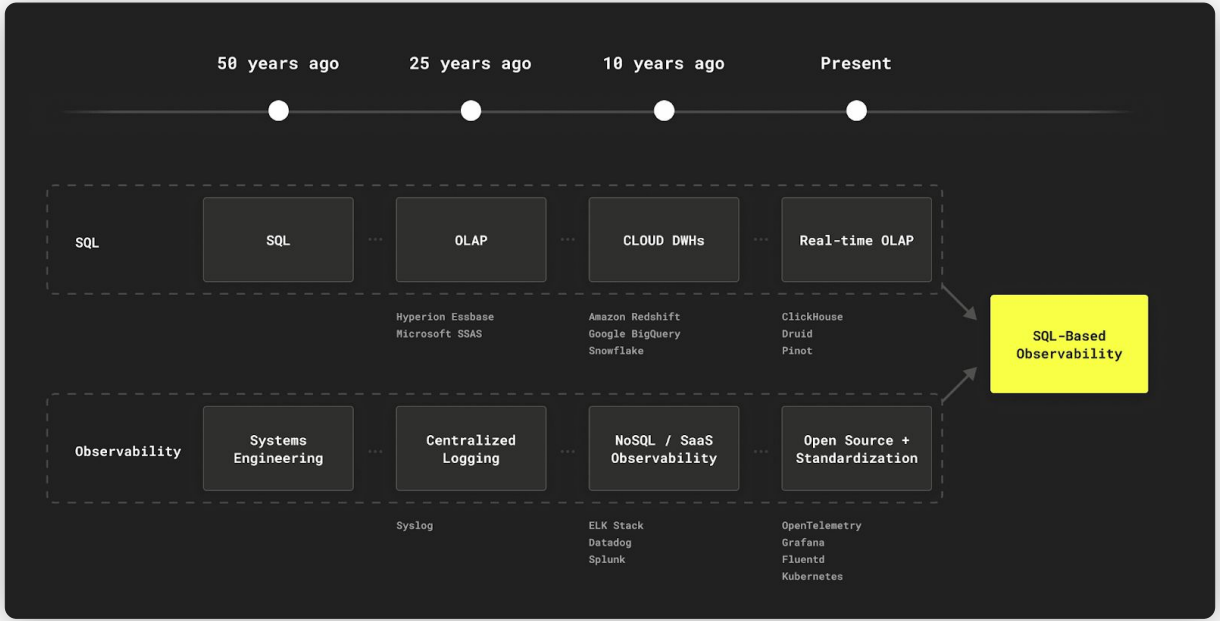
- Added support for additional Kafka flavours, 7 in total!

Coming soon

- Amazon S3 connector for bulk data loading
- AVRO format support for Kafka
- Amazon Kinesis connector



SQL-Based Observability 🔭

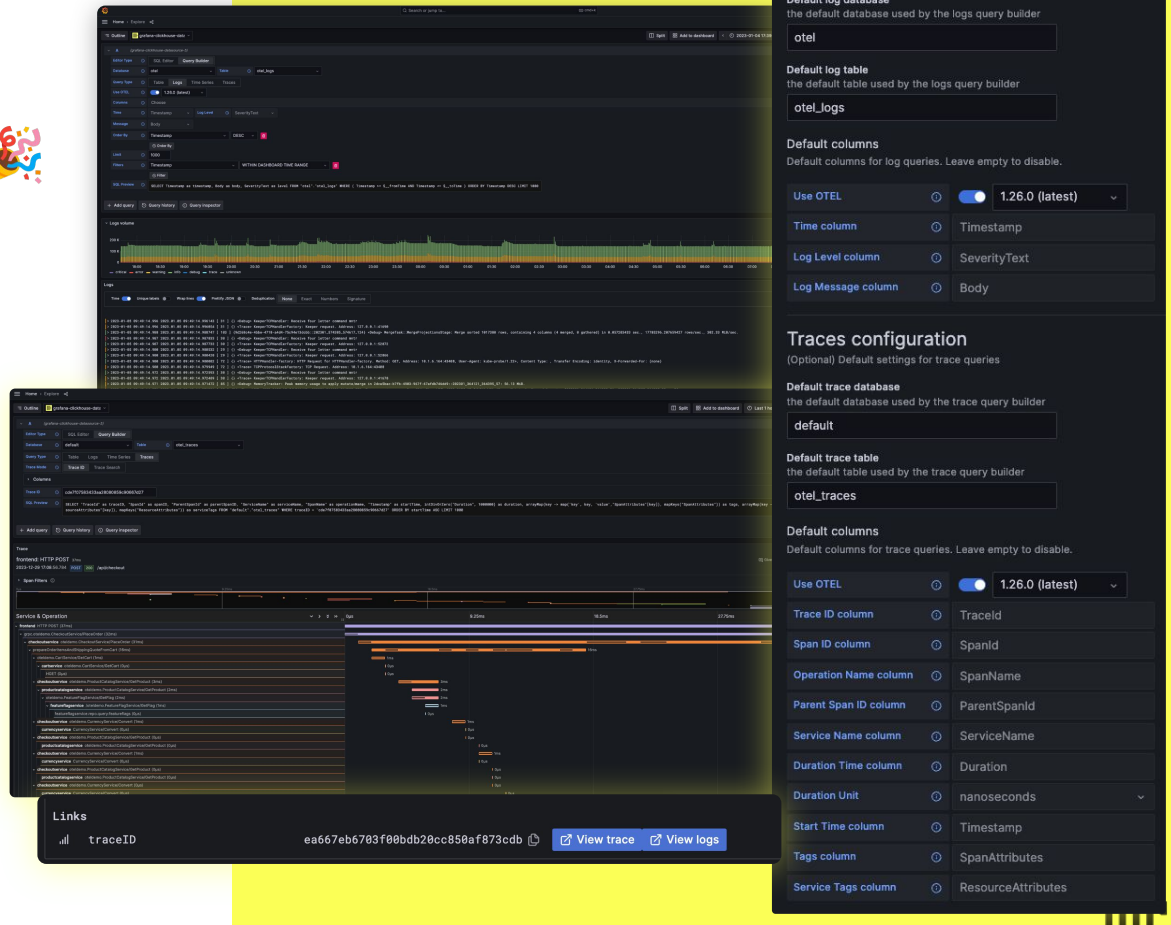


https://clickhouse.com/blog/the-state-of-sql-based-observability

ClickHouse

Grafana datasource V4 🎉

- Improved UI/UX for query builder, datasource configuration, logs and traces explorer
- Improved support for the OpenTelemetry standard
- Aligned with the SQL-Observability workflow with a guided experience



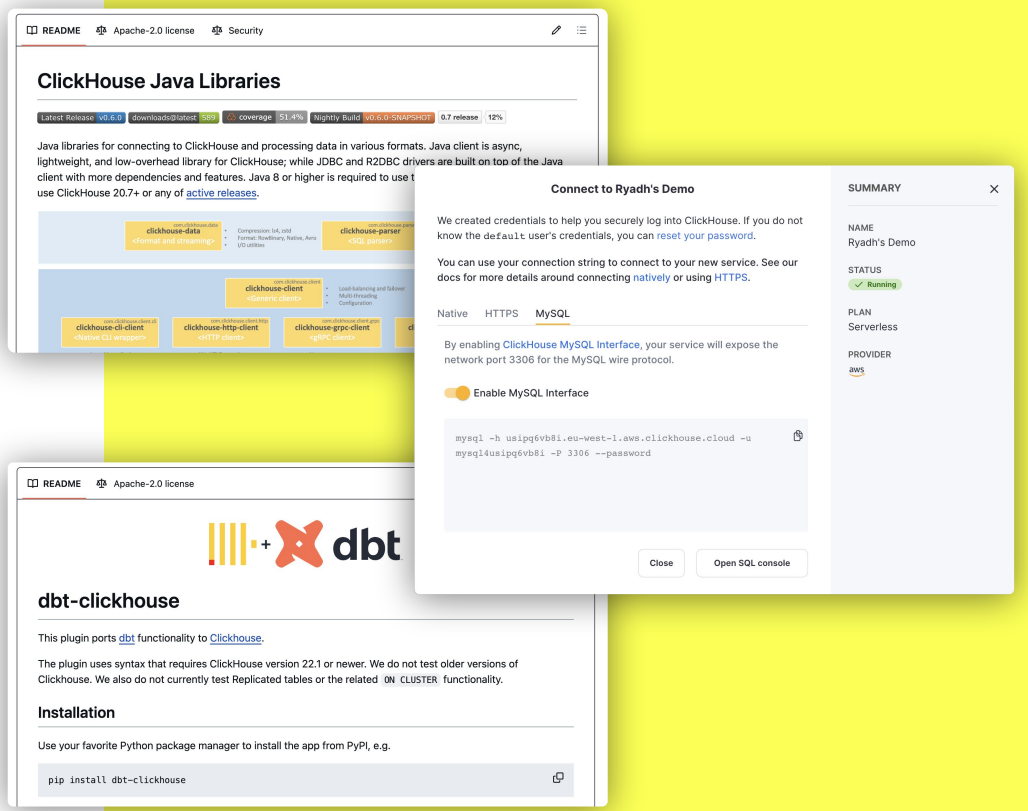
ClickHouse

Grafana Plugin DEMO

ClickHouse

Also happening in integrations …

- Extended supported Data Types for Apache Beam
- Kafka Connector Sink performance improvements
- MySQL Interface Improvements
- DBT upgrade to 1.7 with support for Materialized Views
- clickhouse-java : refactoring in progress
- Golang/Python/JS: reliability and performance improvements



Ecosystem Integrations status and roadmap

	Available	Recent/Next
Data Ingestion / Orchestration	🔥 DBT Adapter (core) 🔥 MySQL Interface (core) 🔥 Fivetran (partner) 🔶 OpenTelemetry (community) ✅ Airbyte (partner) ✅ Vector (community) ✅ PostgreSQL (core) ✅ MySQL (core) ✅ FluentBit (community) ✅ EMQX (partner) ✅ Apache Nifi (community) ✅ Apache Spark (community) ✅ Apache Flink (community)	🟡ClickPipes - S3 / GCS 🟡ClickPipes - Avro for Kafka 🟡ClickPipes - Kinesis 🔶ClickPipes - CDC Postgres 🔶 Apache Beam (community) 🔶 Google Dataflow (partner) ✅ClickPipes - Azure Event Hub ✅ ClickHouse-Kafka-Connect GA ✅ Apache Airflow (community) ✅ RisingWave (community) ✅ Apache Sea Tunnel (community)
Language Clients	✅ Node.js (Added browser support) (core) ✅ Common Lisp (go Julio!) ✅ Golang (core) ✅ Python (core) ✅ C++ ✅ PHP (community) ✅ Ruby (community) ✅ Rust (community) ✅ R (community) ✅ Erlang (community)	🔥 Java (Refactoring) (core) 🔶 C# .NET (community) 🔶 ODBC (Maintenance) (core)

Legend

✅ Completed
🟡 In development
🔶 Coming Next / Evaluating
🔥 Recently improved

Ecosystem Integrations status and roadmap

	Available	Recent/Next
Data Visualization	🔥 Tableau Desktop (core) 🔥 Google Looker Studio (partner) ✅ Superset / Preset (partner) ✅ Deepnote (partner) ✅ HEX (partner) ✅ Mode (partner) ✅ Metabase (core) ✅ Cube.js (partner) ✅ EXPLO (partner) ✅ Mode.com (partner) ✅ Redash (partner) ✅ Sisense (partner) ✅ Tooljet (partner) ✅ Zing Data (partner) ✅ Observable (partner) ✅ OpenBlocks (community)	🔥 Grafana (Redesign v4) (partner) 🟡 Tableau Online (partner) 🟡 Google Looker (partner) 🔥 AWS QuickSight (partner) 🔥 PowerBI (core) 🔶 Azure Power BI (partner)
SQL Clients	✅ ClickHouse Client ✅ DBeaver (partner) ✅ Datagrip (partner) ✅ JupySQL (community)	

Legend

✅ Completed
🟡 In development
🔶 Coming Next / Evaluating
🔥 Recently improved

See 110+ integrations on
https://clickhouse.com/docs/en/integrations

Thanks for joining!

 clickhousedb.slack.com

 @ClickHouseDB