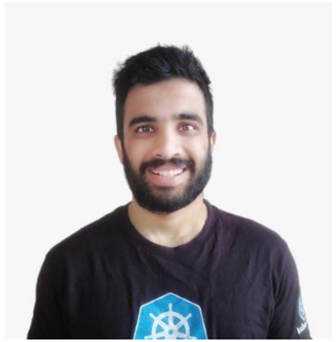
platformatory
Cloud Native Engineering Studio

Moving Druids to the house

ClickHouse Bengaluru Meetup on 8 Feb 2025

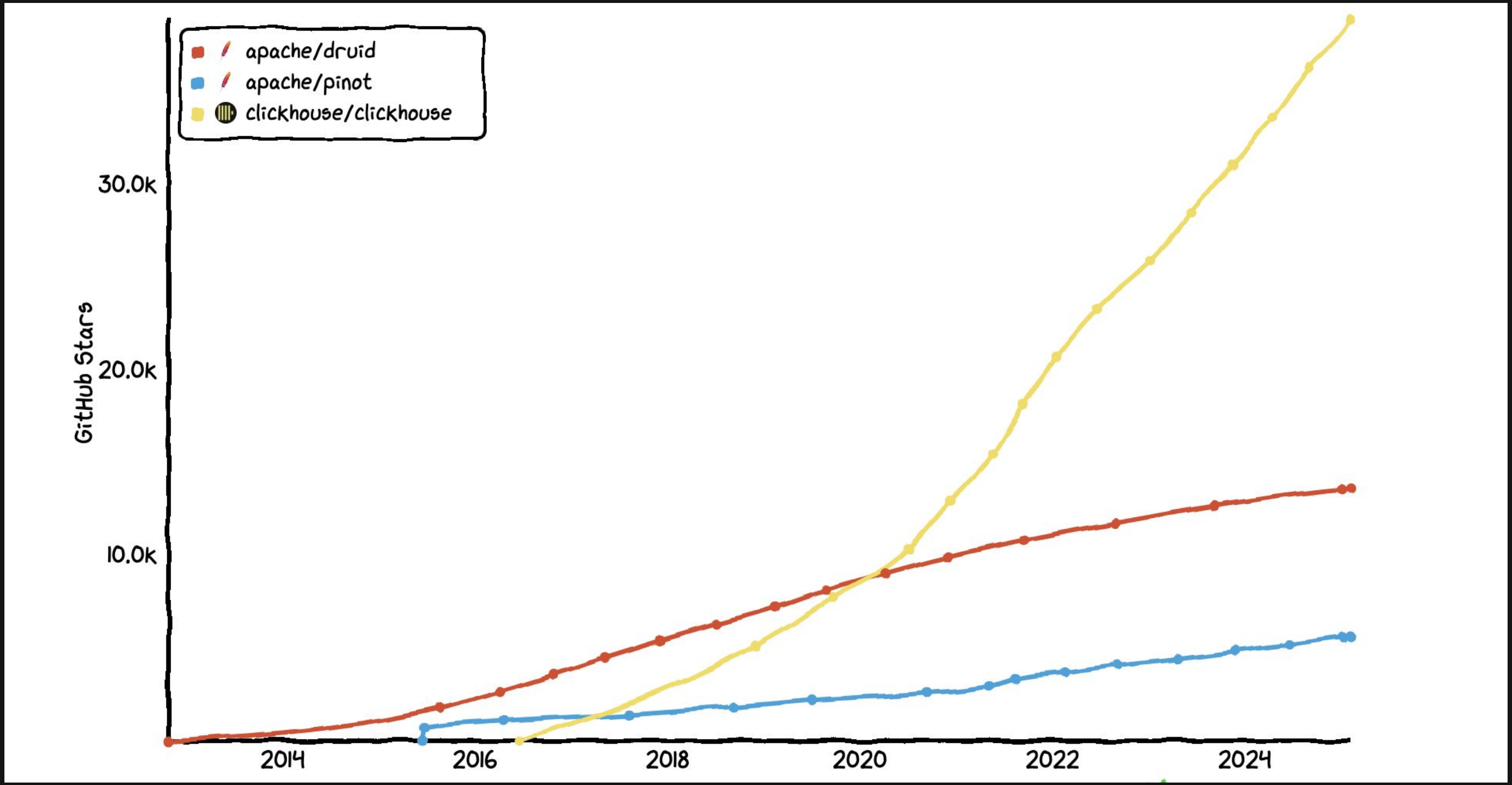
Speaker Info



- Platform engineer @ platformatory.io
- Confluent Community Catalyst, Kong Champion
- Occasional open source contributor to Cloud Native projects (k8s, ArgoCD, Tekton, Litmus, etc)
- Meetup organizer for Bengaluru Streams, Kong, Kafka, Grafana
- https://www.linkedin.com/in/avinash-upadhyaya/

platformatory

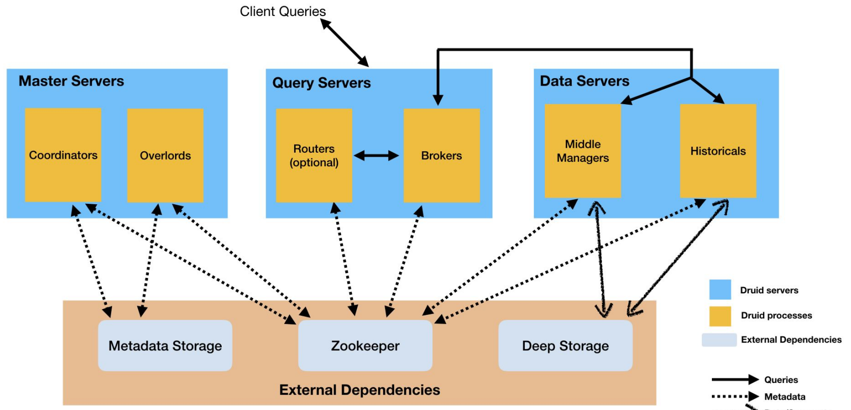
Cloud Native Engineering Studio

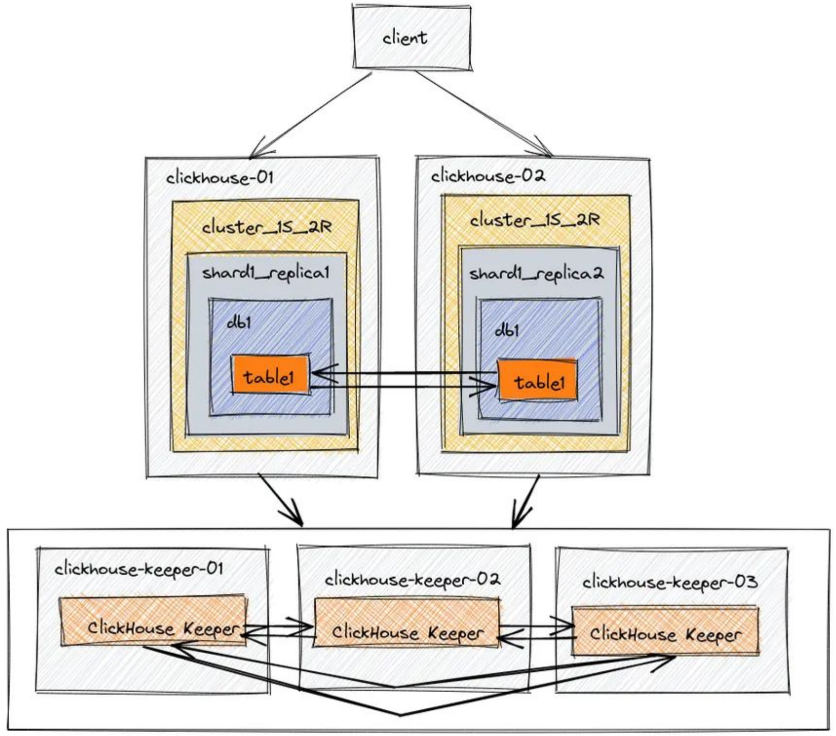


The Trifecta of Real-time OLAP

FOREST OF THE CARNUTES
NON-DRUIDS KEEP OUT
AH, WE'RE THERE!

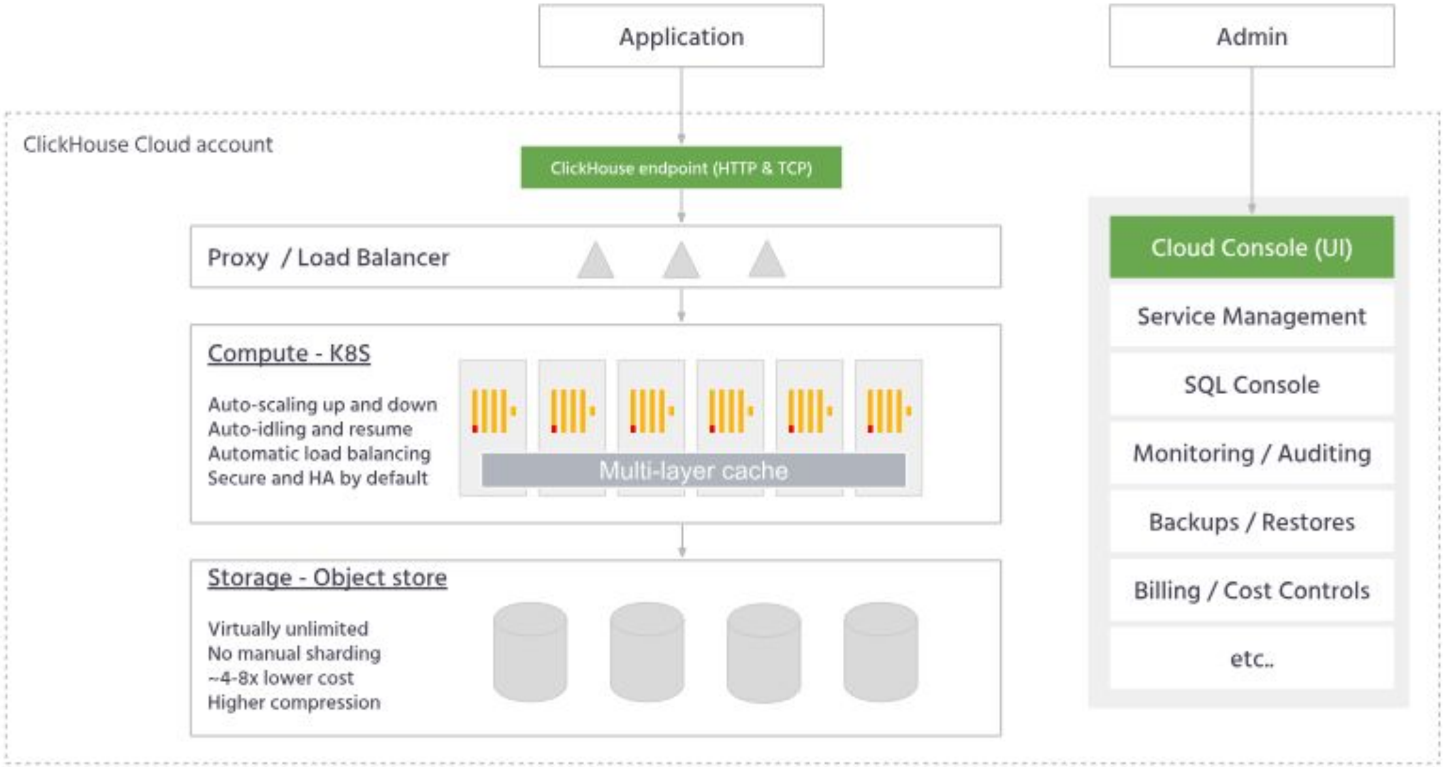
ONE DOES NOT JUST MIGRATE THEIR RTOLAP
imgflip.com





Apache Druid

ClickHouse



ClickHouse cloud is the best way to run ClickHouse.

Scoping your migration → API surface awareness

1. Querying
2. Ingestion
3. Management

Key gotchas

ClickHouse

- SQL over HTTP
- MySQL API
- Native binary API (with C++ client)
- gRPC (new)
- Table Engines are the most important detail
- Primary keys for ordering

Druid

- Druid SQL over HTTP
- JDBC (Avatica JDBC)
- Druid Native queries over HTTP
- Java APIs
- **Separate ingestion APIs → Task/submission API**
- No primary keys

Druid vs ClickHouse

Feature/Aspect	Apache Druid	Apache ClickHouse
Primary Query Interfaces	Dual: Native JSON query language and Druid SQL	Single unified SQL dialect with proprietary extensions
Query Expression	JSON for native queries; SQL for Druid SQL	Standard SQL (with additional functions for analytics, arrays, and distributed processing)
Specialized Functions	Time series–focused functions (e.g., TIME_FLOOR)	Array handling (`arrayJoin`), distributed query hints, MergeTree-specific options
Control and Flexibility	Native JSON offers detailed control over execution	SQL dialect optimized for columnar processing and high throughput
Ease of Use	SQL layer eases integration; native queries require more specific JSON structure	Unified SQL interface simplifies development and integration

Migration Overview

- Ingestion → Inventory of data sources, input formats, transformations and roll-ups
  - In ClickHouse, there's no separation between ingestion and query APIs
  - Carefully consider connectors, integrations and input formats
  - A table level mapping is crucial: choice of table family, schema parity and use of MVs
- Query → Inventory of user-facing applications, BI dashboards and clients at large
  - Druid SQL or JDBC offers scope for easier migration. Could save a lot of query rewriting time
  - Consider an API shim to perform incremental roll-out if using REST APIs
- Phased migration → across a single context of data sources and queries
- Develop a test and cut-over plan!
- Benchmark for tuning

Thank you

hello@platformatory.com

www.platformatory.io