

Metrika data, published only when tested

Python, ClickHouse and dbt for a pharmaceutical client

A dashboard should never see half a night of data.

For a pharmaceutical client moving reporting off Google services: raw Yandex Metrika data, checked before every publish and reconciled with the source.

THE QUESTION

Can the report trust the numbers, and tell a broken load from a quiet website?

THE MODEL

Visits and hits in ClickHouse, rebuilt and tested with dbt

THE OUTPUT

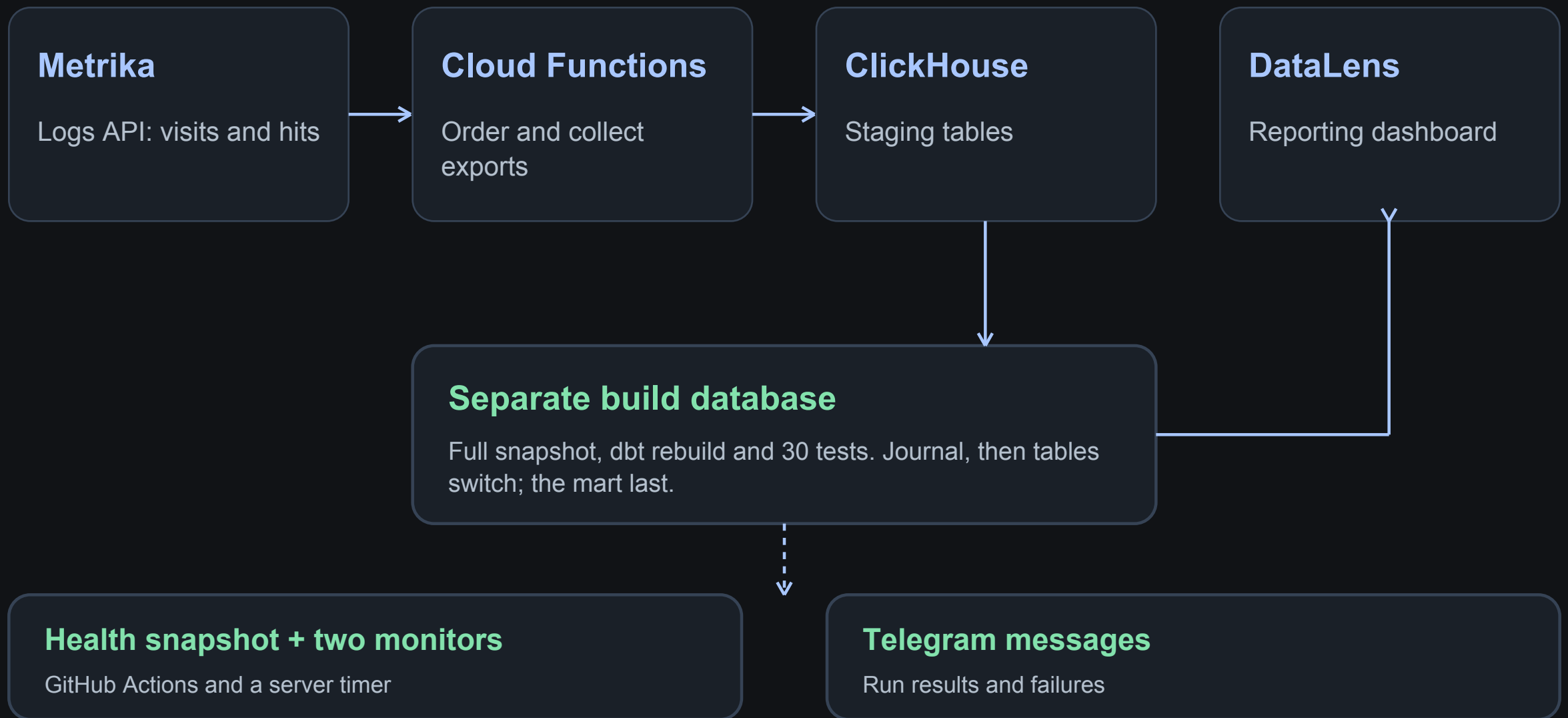
A tested reporting mart for DataLens

MY CONTRIBUTION

Pipeline design, Python, dbt models, release safety and reconciliation.

Collect, build, test, then switch

Solid arrows: data. Dashed: monitoring and alerts. Illustration of the documented design.



Two Yandex Cloud functions and three timers, described in Terraform.

Release, reconcile, explain

Three decisions behind data people can trust.

01 Publish whole releases

New data never lands on the working tables. It is built and tested in a separate database, a journal records the switch, and the previous version stays for rollback.

02 Reconcile against the source

A coverage test passed only because it used the same join as the mart. I compared 2026 with Metrika for each of 250 days, recovered 18 missing hits on five dates and explained the remaining gaps.

03 Flag bots with readable rules

The Logs API has no robot flag. Two explainable rules mark suspicious cookies and subnets; in a one-week named check all 8 robots Metrika labelled were flagged.

Make a failed run diagnosable

Scheduled processing needs a recovery path as well as a happy path.

One writer, recoverable switches

An exclusive lock allows one writer. After a failure, recovery compares journal identifiers and performs only the missing switches.

Two independent monitors

A health snapshot is checked hourly by GitHub Actions and every ten minutes by a server timer; no successful publish for 26 hours is an error.

Alerts without secrets

Successes and failures go to Telegram; the bot token is stripped from error text before logging.

Infrastructure as code

Terraform describes the cloud as it actually runs, after an audit found the README had drifted from production.

What the checks established

Figures come from the repository and reconciliation records, not a new measurement.

250

days of 2026 compared
with Metrika, unsampled

30

dbt tests before each
publish

34

isolated tests of release,
failure and recovery

18

missing hits recovered,
nothing deleted

Delivered

Scheduled Logs API collection, a tested dbt mart, a release journal with rollback, source reconciliation, two monitors and Telegram alerts.

Evidence boundary

Client under NDA; traffic volumes not published. Reconciliation covers visits and pageviews, not users or attribution. The release went live on 13 September 2026; later operation, dashboard use and business impact were not verified.