OurSQL Foundation

NOV 3 · WEBINAR

Beyond the Proxy: The Expanding ProxySQL Ecosystem





René Cannaò & Alkin Tezuysal

CEO & Chief Strategy Officer, ProxySQL

10:00 AM ET · 16:00 CET

PROXYSQL.CLOUD®

ProxySQL.Cloud, by example

The rules, manifests, phases, logs and numbers you actually see when a managed ProxySQL® fleet runs inside your own cloud account.

MySQL :6033 | PostgreSQL :6133 | Admin :6032

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

DEPLOYMENT

Three ways to run it

MANAGED

We host it

The fastest start. Your clusters run in our cloud and you get one endpoint.

BYOC

Your cloud account

We provision and operate the fleet inside your account. The data stays in your VPC.

BYOK

Your Kubernetes

Bring an existing cluster. Install the agent by Helm chart and keep your own platform.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 01 · CONNECTION POOLING

Ten thousand clients, two hundred backend connections

10,000 → **200**

client connections at the proxy

backend connections in use

The database stops being sized for your app's connection count and starts being sized for its concurrency.

proxysql_client_connections_connected · proxysql_connpool_conns · illustrative ratio

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 02 · QUERY ROUTING

Read/write split in two rules

```
-- read/write split, in two rules
INSERT INTO mysql_query_rules
  (rule_id, match_digest, destination_hostgroup)
VALUES
  (10, '^SELECT.*FOR UPDATE', 10),
  (20, '^SELECT', 20);
-- hostgroup 10 = writer, 20 = readers
```

Ordered rules

Evaluated by rule_id. First match wins, so specific patterns sit above general ones.

No app change

The split lives in the proxy, not in your ORM or connection strings.

More than routing

The same table rewrites, caches, sets timeouts and blocks patterns.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 03 · FAILOVER

A backend fails and the endpoint stays put

01 · DETECT

Health checks

Ping, connect and replication-lag checks run continuously against every backend.

02 · SHUN

Take it out

The failing host leaves its hostgroup. Its connections are not reused.

03 · REROUTE

Same endpoint

Queries go to the remaining backends. No DNS change, no new connection string.

04 · RETURN

Back in service

Once checks pass again the host rejoins its hostgroup and takes traffic.

Failover becomes a proxy-side event instead of an application-side outage.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 04 · DATA BOUNDARY

What crosses the boundary, and what never does

LEAVES YOUR VPC

- Cluster desired state: replicas, versions, hostgroups, query rules
- Lifecycle status and phase from the operator
- Aggregated metrics and query digests, shapes not values
- Audit events: who changed what, and when

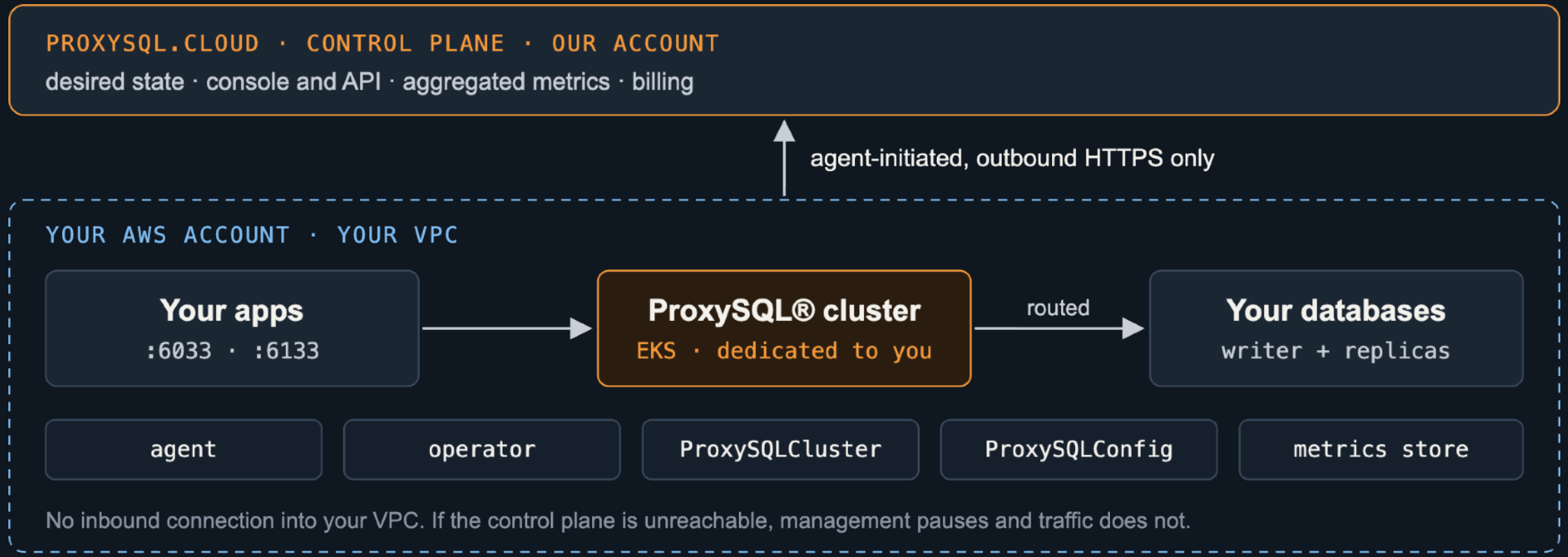
NEVER LEAVES YOUR VPC

- Row data. Every query is served inside your network
- Database credentials and application secrets
- Query text with literals in it
- Backups, written to your own S3 bucket

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 05 · ARCHITECTURE

Two planes, one direction of travel



© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 06 · DAY 0

Four inputs, one manifest

01 Apply a CloudFormation stack that creates a scoped IAM role for the agent.

02 Choose the VPC, subnets and region the cluster runs in.

03 The agent installs into your Kubernetes cluster and registers itself.

04 Create the first cluster in the console: name, size, engine, backends.

```
# the cluster you asked for,
# as the operator sees it
kind: ProxySQLCluster
metadata:
  name: customer-acme-01
spec:
  replicas: 3
  engine: mysql
  version: "3.0"
status:
  phase: Running
```

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 07 · LIFECYCLE

One phase per cluster, straight from the operator

Pending	Creating	Running	Updating
Degraded	Drifted	Failed	Paused

Degraded is not down

Two of three replicas ready still serves traffic. The fleet is just thinner than you asked for.

Drifted is a warning about the future

Something changed the runtime out of band. It works now and gets reverted on the next reconcile.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 08 · RECONCILE LOOP

You write desired state and the loop closes it

CONSOLE

You declare it

A change in the console or API becomes a new desired-state version.

→

AGENT

It pulls it

Your in-cluster agent fetches the version over its outbound link.

→

K8S

CRDs written

State lands as ProxySQLCluster and ProxySQLConfig objects.

→

OPERATOR

Converges

Loads the config into each ProxySQL pod and keeps it there.

→

STATUS

Reports back

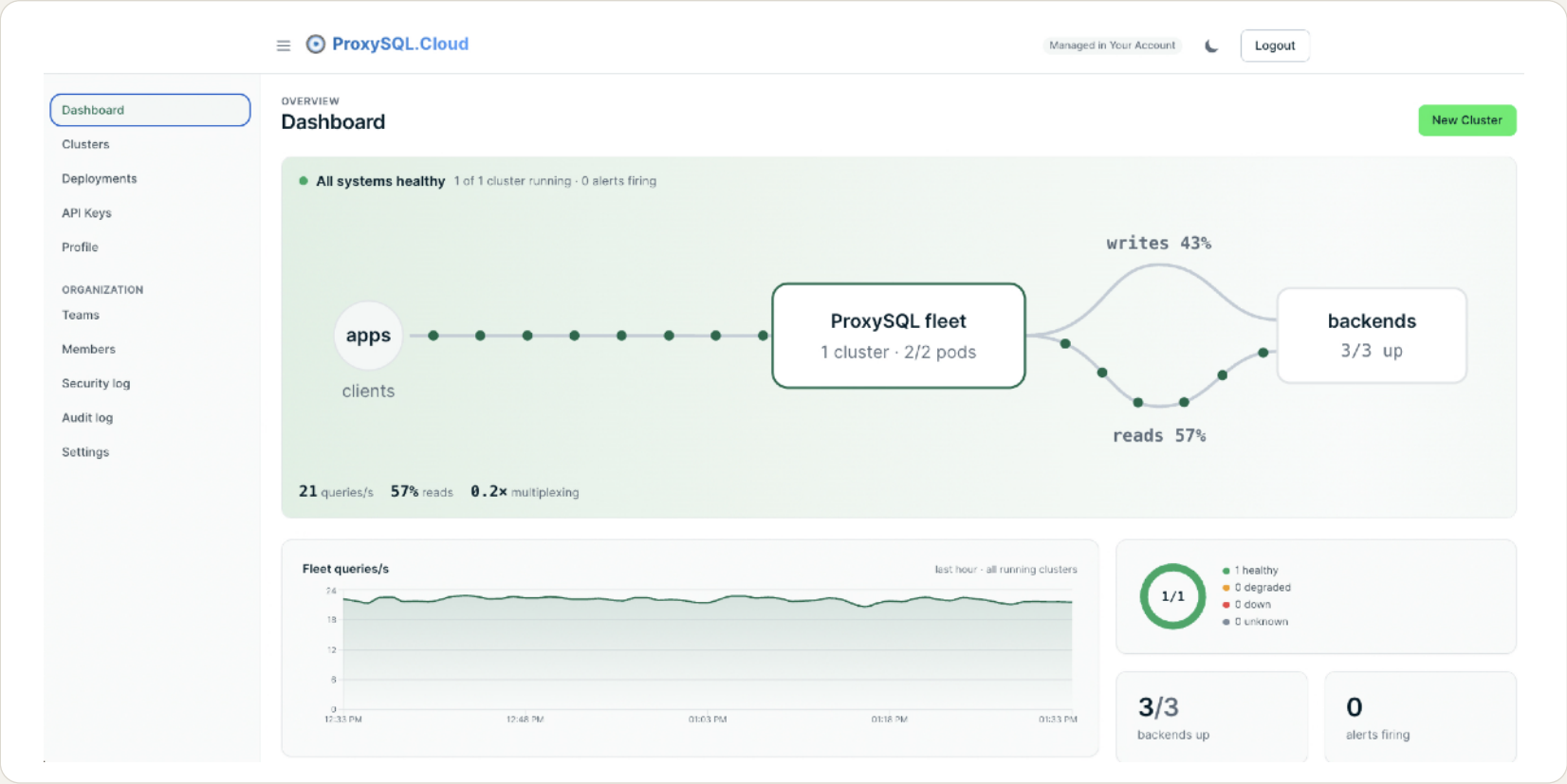
Applied version and per-replica sync state return to the console.

✓ Config applied. 3/3 replicas synced, 0 drifted.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · DASHBOARD

The fleet at a glance



Traffic as a flow

Apps into the fleet, then split to backends: 43% writes, 57% reads.

Health in one line

1 of 1 cluster running, 0 alerts firing, 3/3 backends up.

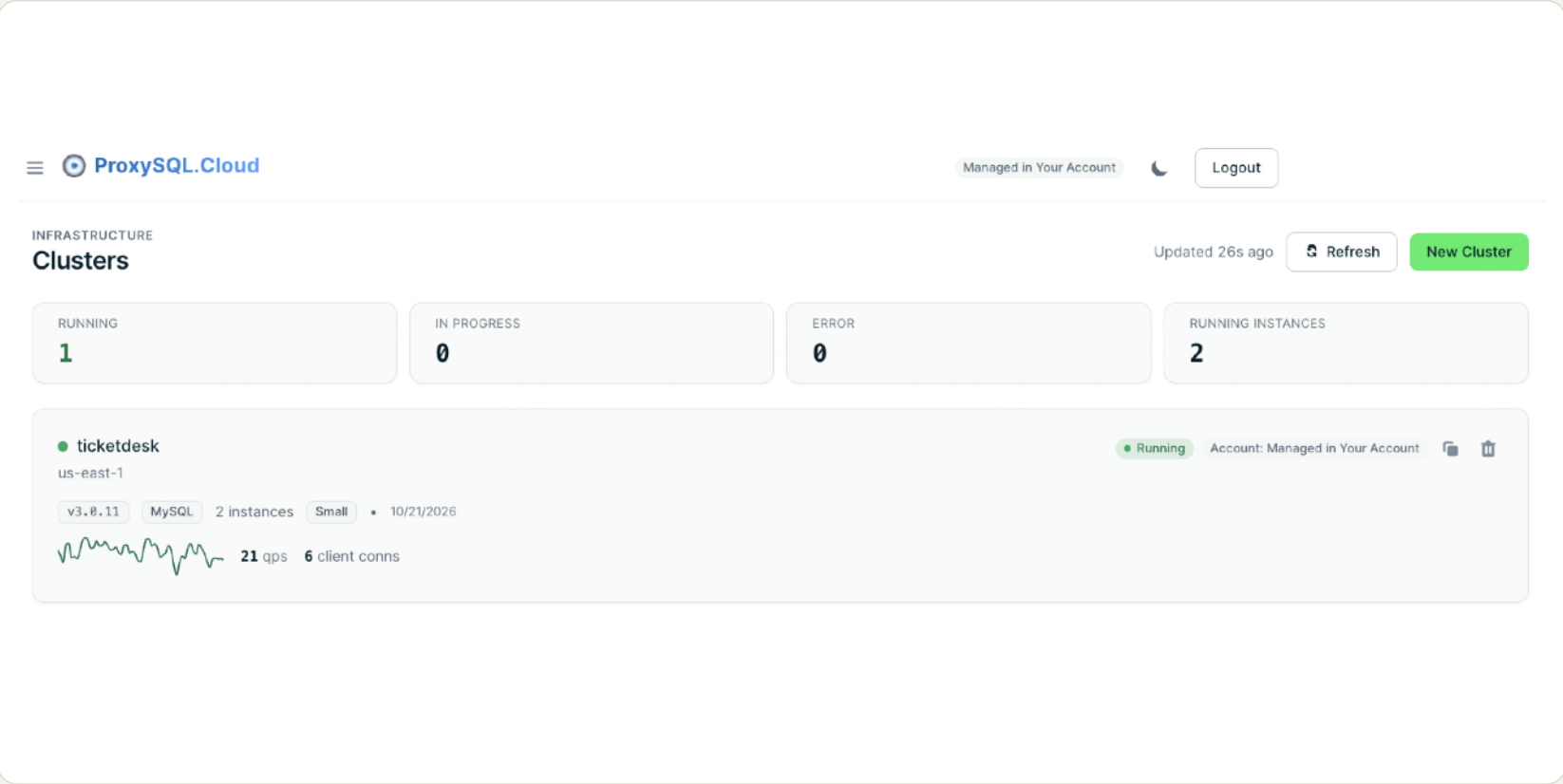
Fleet queries/s

The last hour across every running cluster.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · CLUSTERS

One real cluster: ticketdesk



What you picked

us-east-1, ProxySQL v3.0.11, MySQL, 2 instances, Small tier.

The operator's phase

Running, the same word example 07 described.

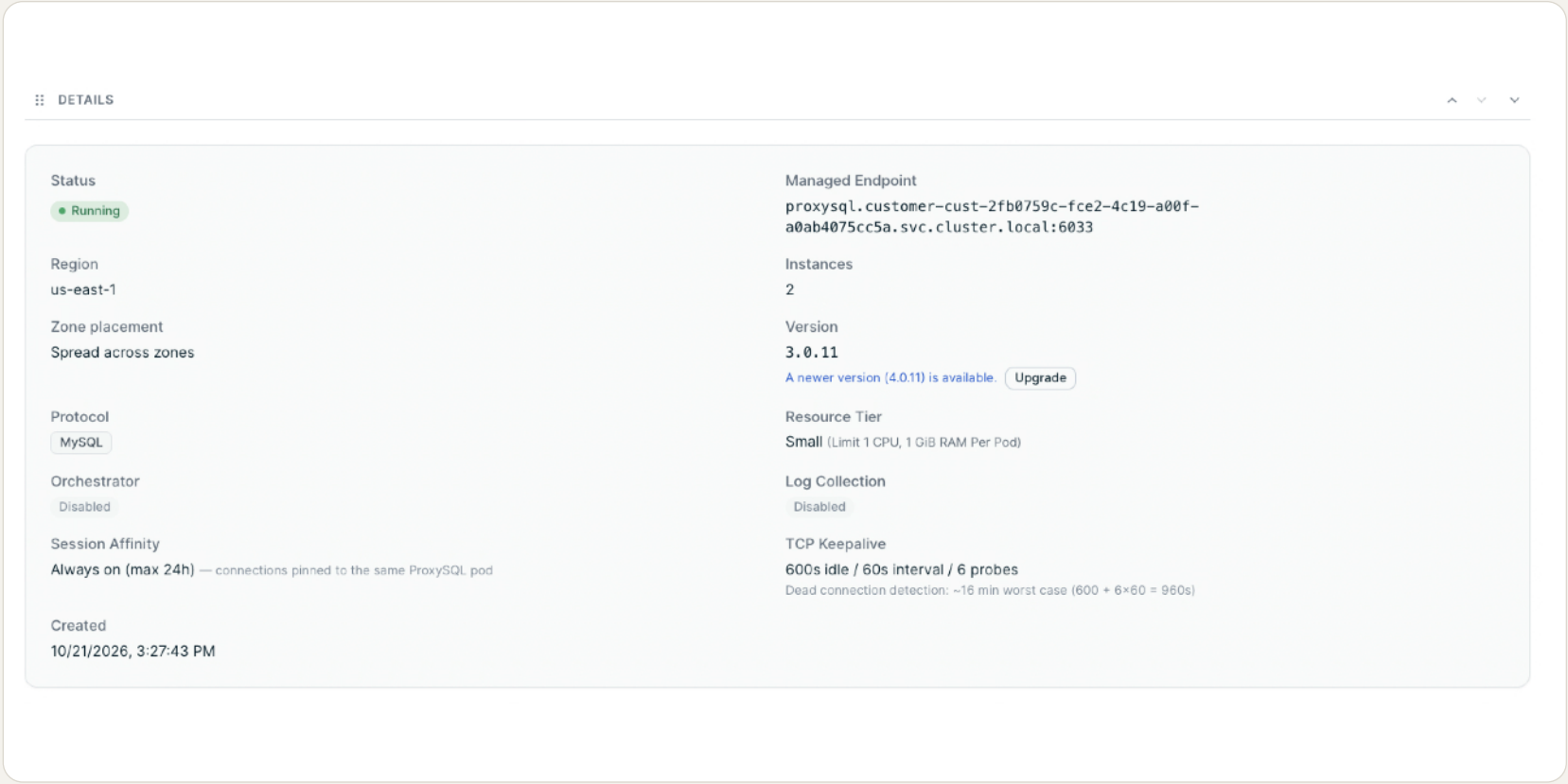
Live traffic

21 qps and 6 client connections, with a sparkline per cluster.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · CLUSTER DETAILS

Every setting, in plain view



In-cluster endpoint

A cluster.local service name on :6033. Apps never leave the VPC.

Upgrade on offer

3.0.11 today, 4.0.11 one button away.

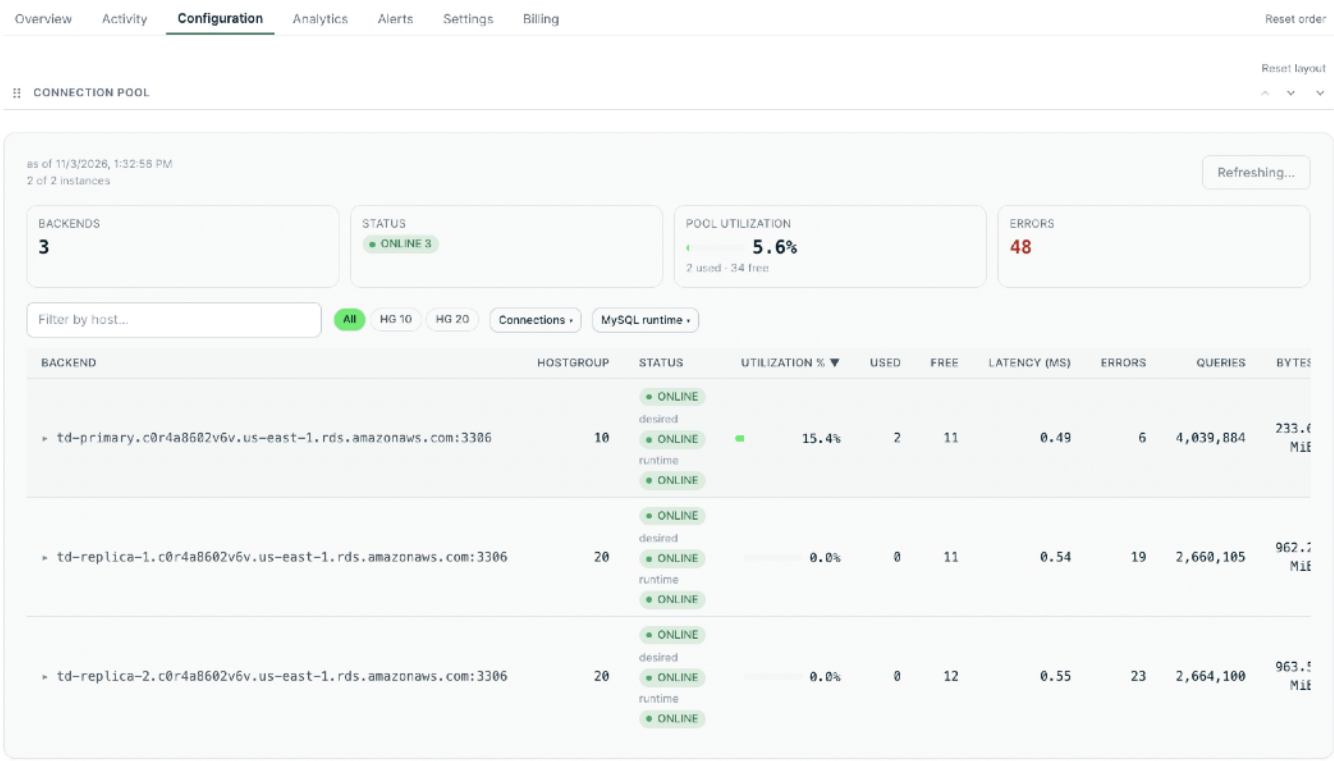
Keepalive, explained

600 + 6×60 = 960s worst case to detect a dead connection.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · CONNECTION POOL

Desired and runtime, per backend



The hostgroups from example 02

td-primary in HG 10, two RDS replicas in HG 20.

Drift, made visible

Each row shows desired and runtime status side by side.

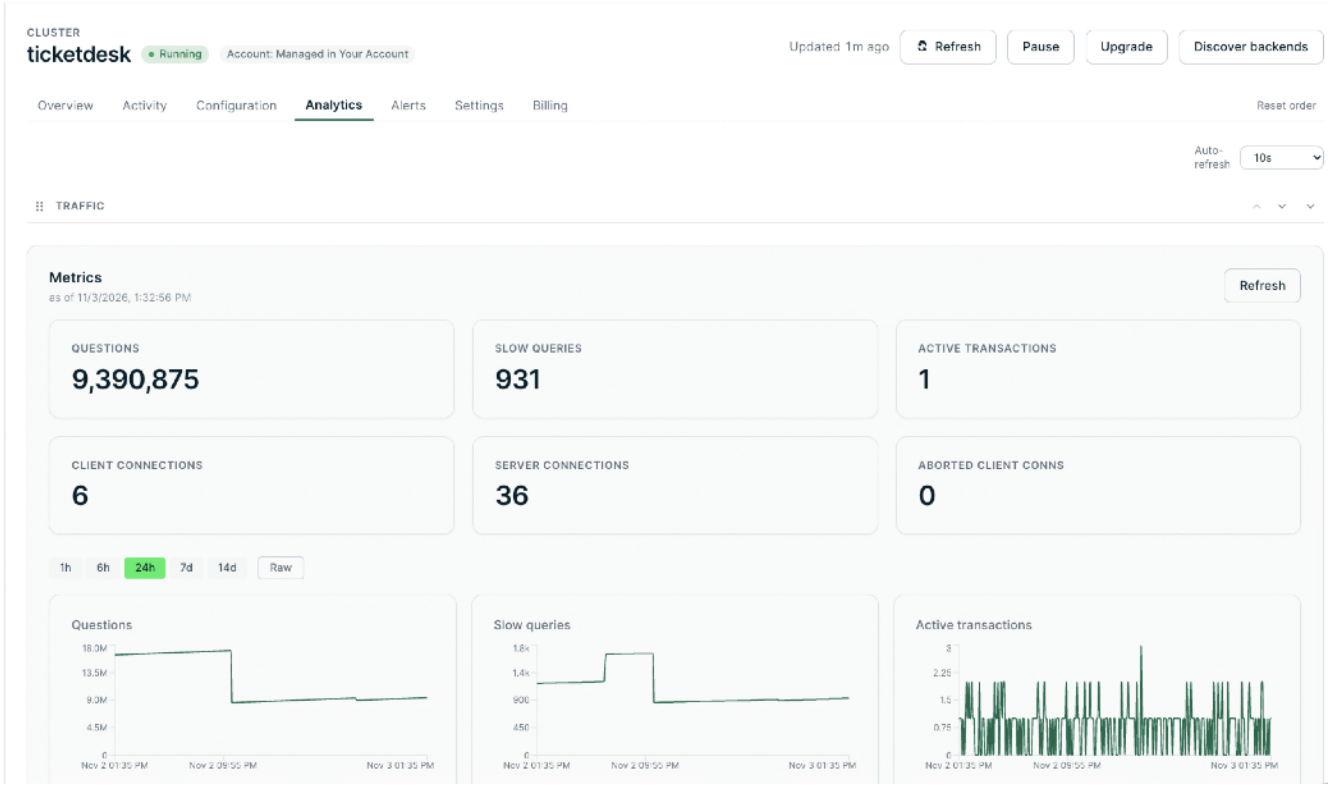
Pool headroom

5.6% utilization: 2 used, 34 free.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · ANALYTICS

Traffic metrics with history



Counters first

9.39M questions, 931 slow queries, 0 aborted client conns.

Then the trend

Ranges from 1h to 14d, or raw samples.

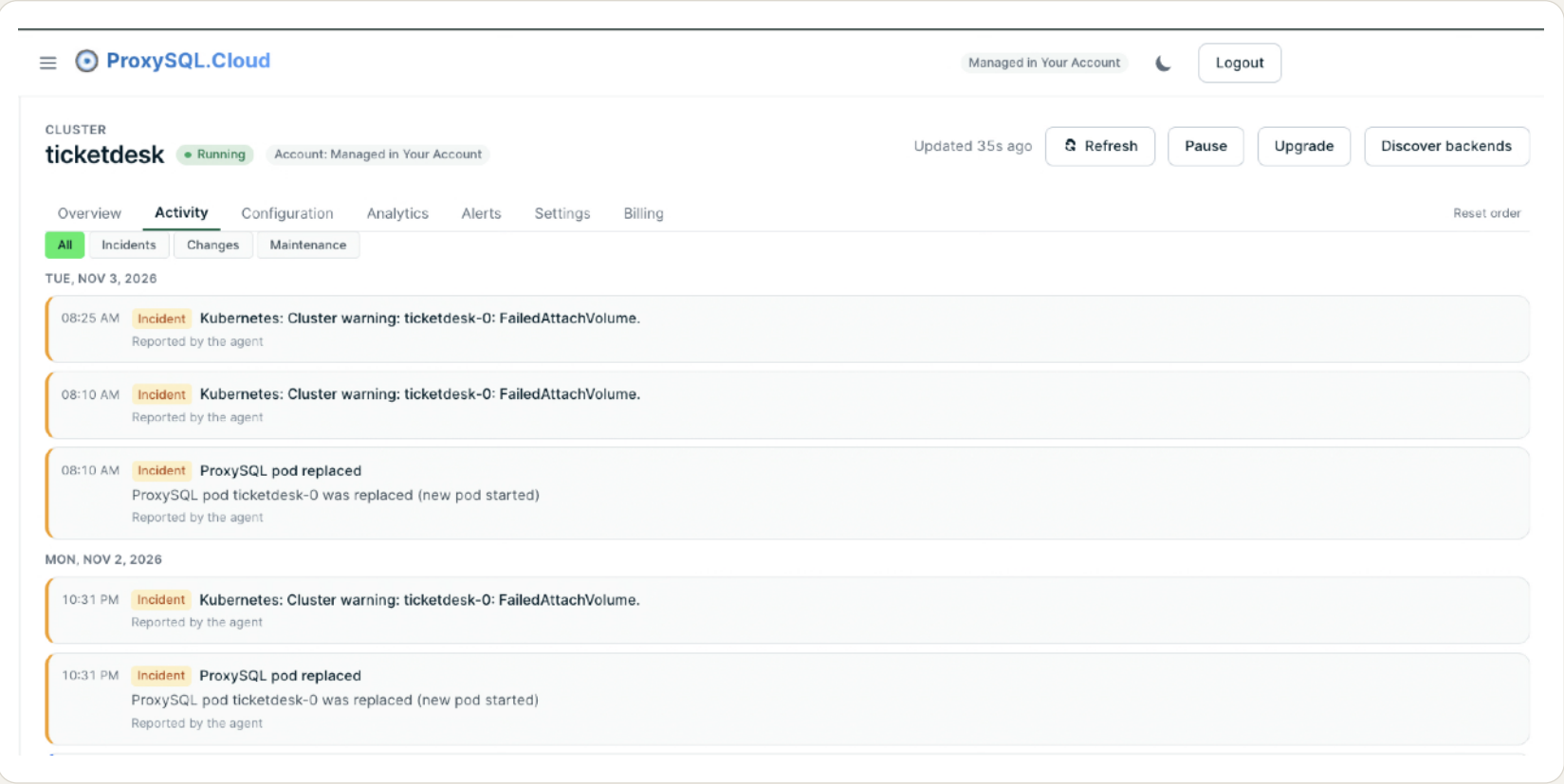
Always current

Auto-refresh every 10s by default.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

IN THE CONSOLE · ACTIVITY

Incidents, reported by the agent



A pod was replaced

ticketdesk-0 came back as a new pod. The endpoint never changed.

Kubernetes, surfaced

FailedAttachVolume events arrive without kubectl.

Filtered by kind

Incidents, changes and maintenance in one timeline.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 09 · MAINTENANCE

Drain, pause, and maintenance windows

01

Drain

The backend goes OFFLINE_SOFT. Open transactions finish, no new ones arrive. Then it is safe to touch.

02

Pause and resume

Reconciliation stops and the phase says so. Exactly what you want during a planned database migration.

03

Maintenance windows

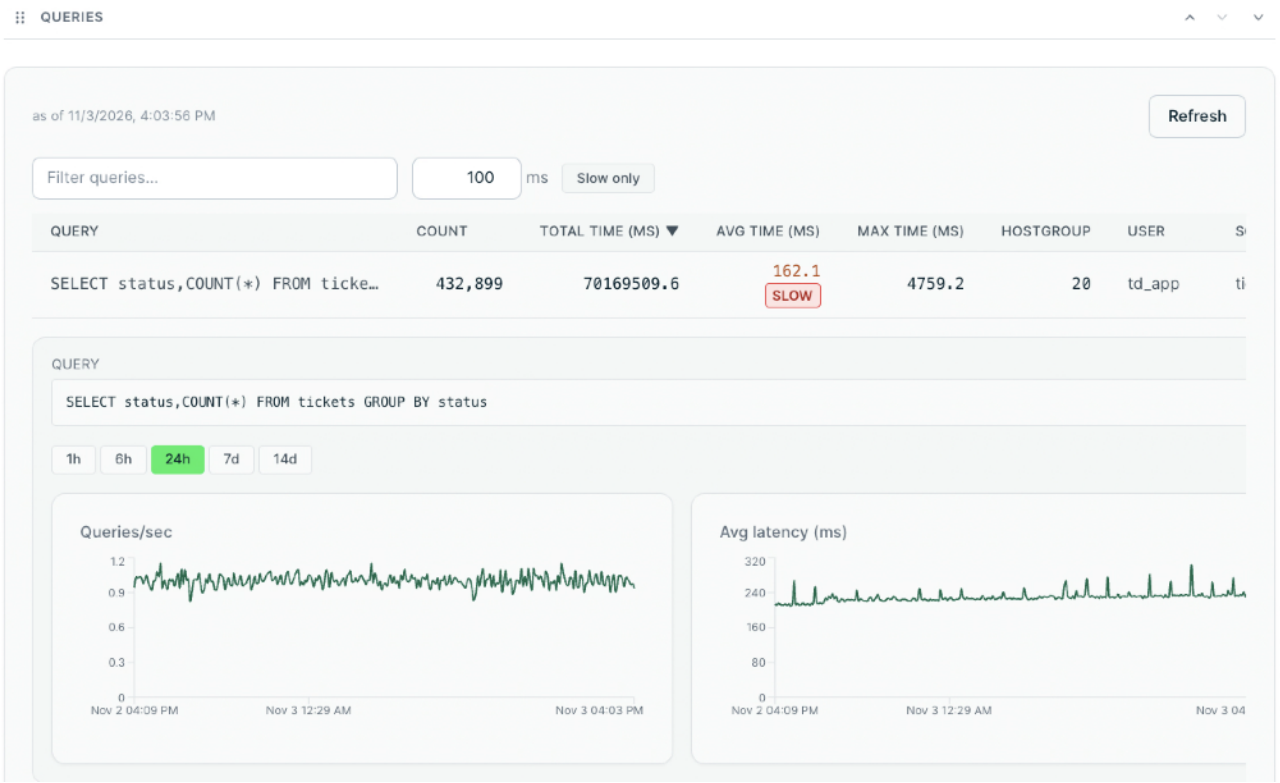
Recurring, in your timezone. The drain runs and the backend comes back afterwards.

```
18:00:00  window opened  hg 20 · db-r2 OFFLINE_SOFT
18:14:37  window closed  hg 20 · db-r2 ONLINE
```

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 10 · QUERY ANALYTICS

Which query is doing this to us



The slow one, found

GROUP BY status on tickets: 432,899 calls, 162.1 ms avg, flagged SLOW.

Ranked by total time

4.76 s max, on hostgroup 20 as td_app.

Shapes, not values

Digests with literals stripped, each with its own qps and latency history.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 11 · UPGRADES

Zero-downtime upgrades, blue/green

3.0.11 → 4.0.11

ZERO-DOWNTIME

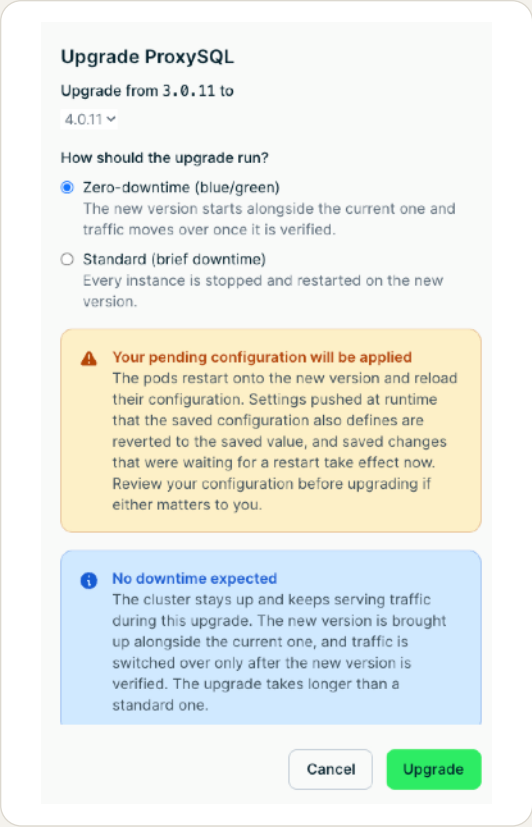
Blue/green

The new version starts alongside the current one. Traffic moves only once it is verified.

STANDARD

Restart in place

Every instance stops and restarts on the new version. Faster, with a brief downtime.



Pending config is applied

Runtime-only changes revert to the saved value, and changes waiting for a restart take effect. The dialog says so before you click.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

EXAMPLE 12 · ROLLOUT

A Helm chart, a token, one service at a time

01

Connect and discover

Install the agent, point discovery at one backend, review the proposed config. Nothing serves traffic yet.

02

Shadow one service

Move a single read-heavy service to the proxy endpoint and compare latency and errors with the direct path.

03

Cut over, service by service

Then turn on pooling limits, routing rules and autoscaling once traffic settles.

At every step the rollback is a connection string.

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.

PROXYSQL.CLOUD®

Ask us anything about your data tier.

We will run a discovery pass against a staging replica and show you the configuration it proposes.

Managed · BYOC · BYOK

proxysql.com/saas

© 2012–2026 ProxySQL. ProxySQL® is a registered trademark of ProxySQL.