Improve your SQL workload with observability

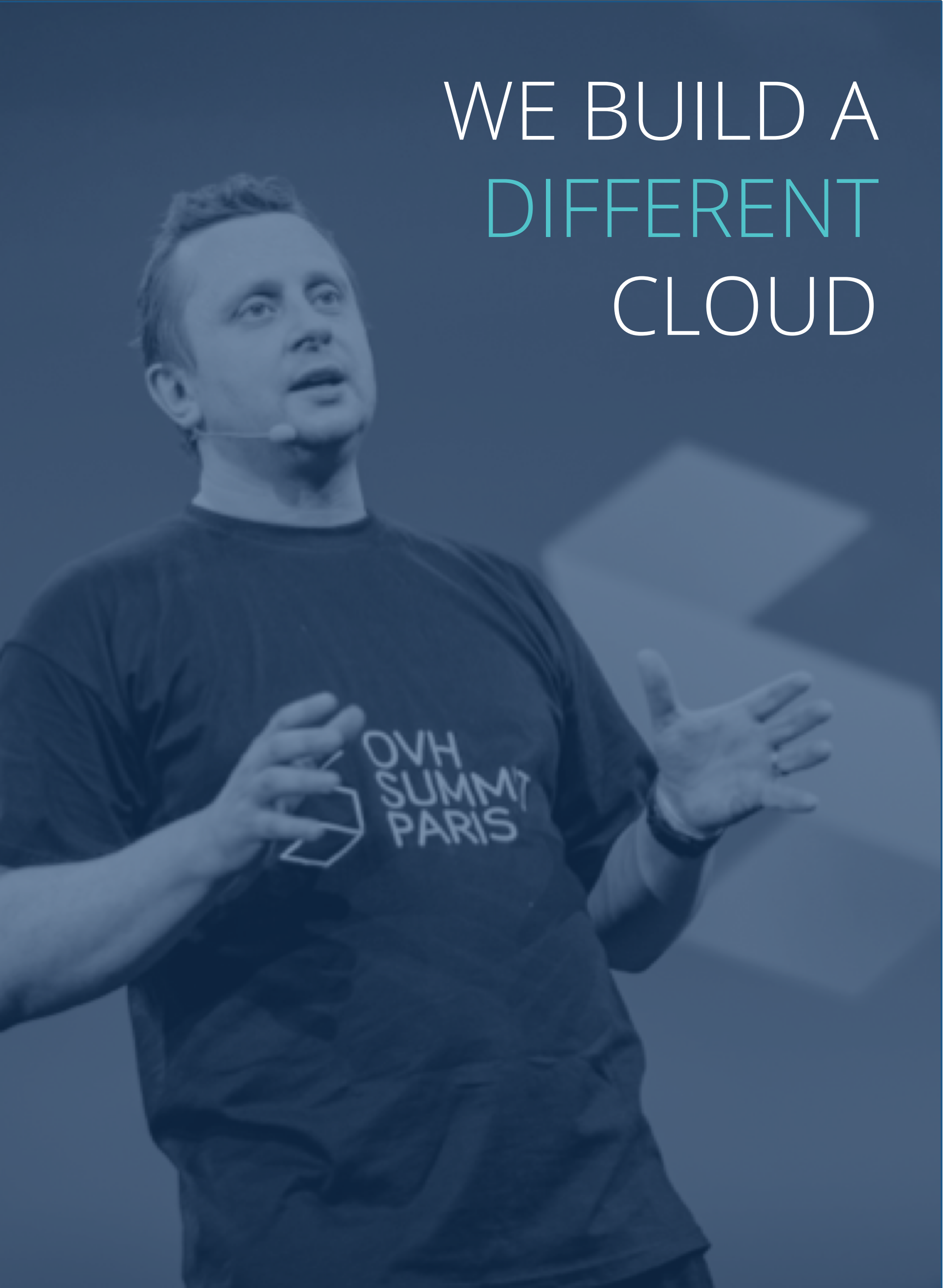
PostgresOpen 2019 @ Orlando

OVH

Wilfried ROSET, Engineering Manager

Speaker

- Wilfried ROSET
- wilfriedroset @ github
- Engineering Manager @ OVH



WE BUILD A DIFFERENT CLOUD

a truly SMART Cloud

SIMPLE AND QUICK TO SET UP

MULTI-LOCAL, CLOSE TO EACH PERSON THROUGHOUT THE WORLD

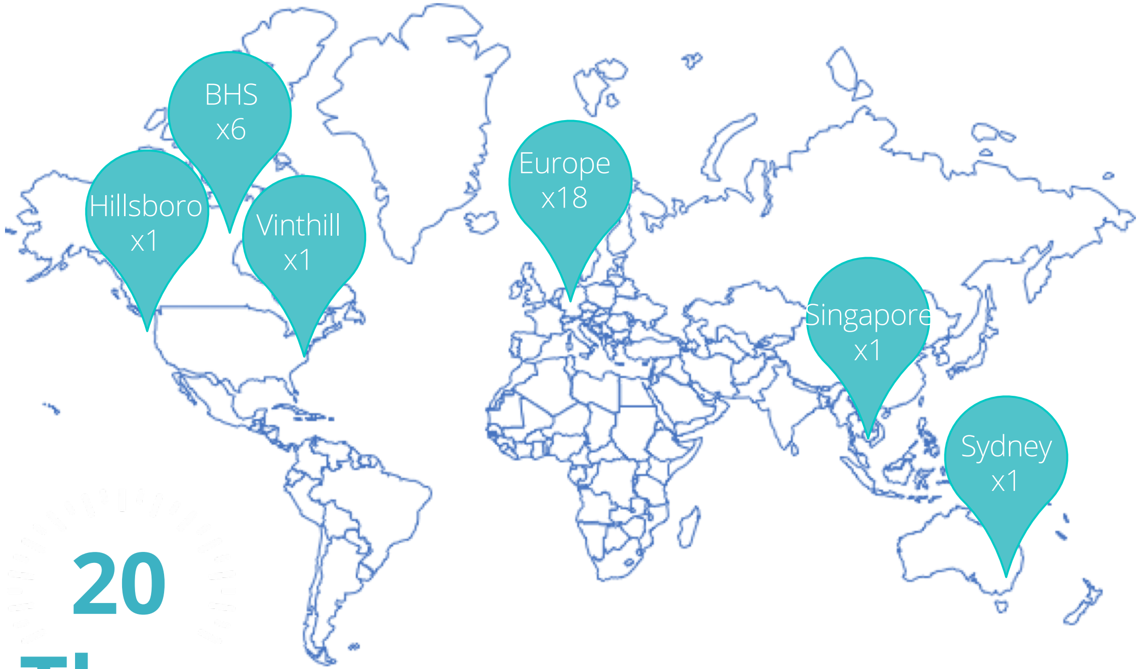
WHOSE COST IS **A**CCESSIBLE AND PREDICTABLE

REVERSIBLE, OPEN AND INTEROPERABLE

TRANSPARENT AND RESPONSIBLE

TO GROW AND SUCCEED TOGETHER

GLOBAL CLOUD PROVIDER



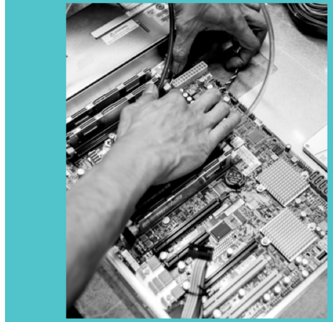
20 Tbps

34 Point of Presence
IN 3 CONTINENTS

OVH

2 200
EMPLOYEES

IN **18**
COUNTRIES



BUILDER OF ITS
OWN SERVERS
SINCE 2002

+ **1 million**
SERVERS built
since 1999

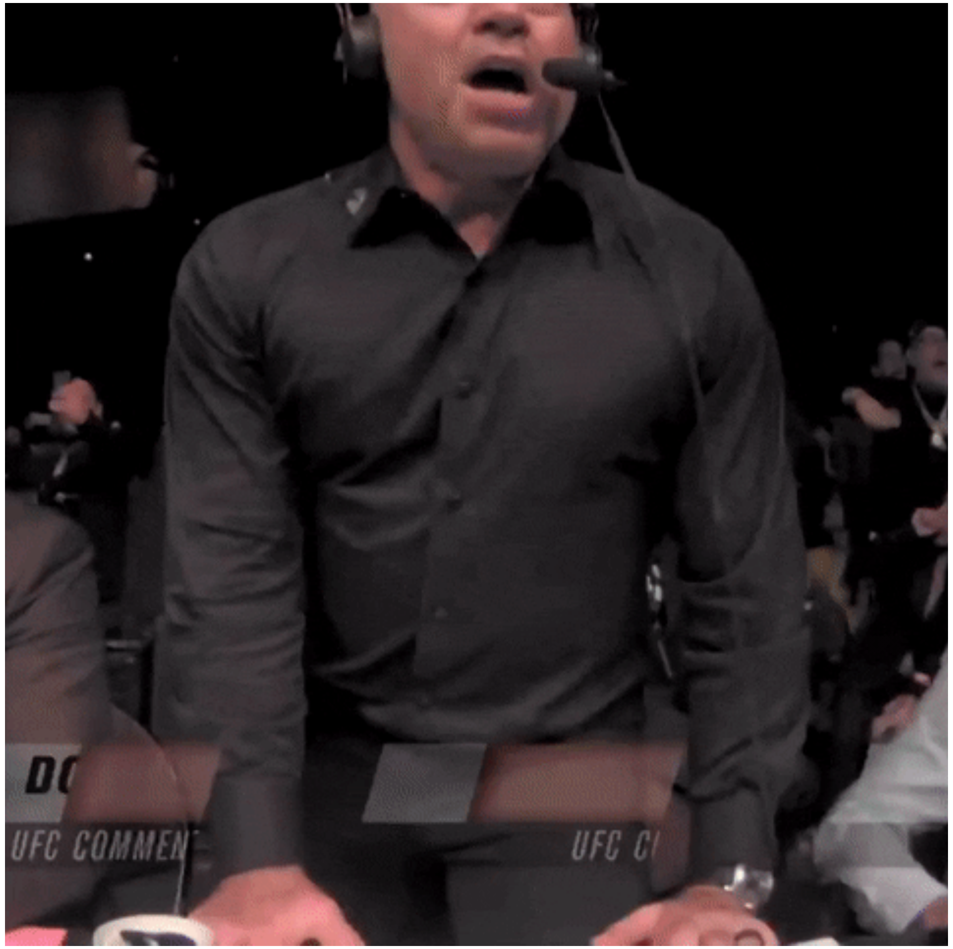
28
DATA CENTERS in
12 locations

EXISTING SINCE
1999

WE ARE
AN INDUSTRIAL PLAYER

Everyone uses SQL

- Directly or Indirectly:
  - CEO / CTO / CXO
  - Dev / DevOps / SRE
  - Support & Run



Is this happening to you?

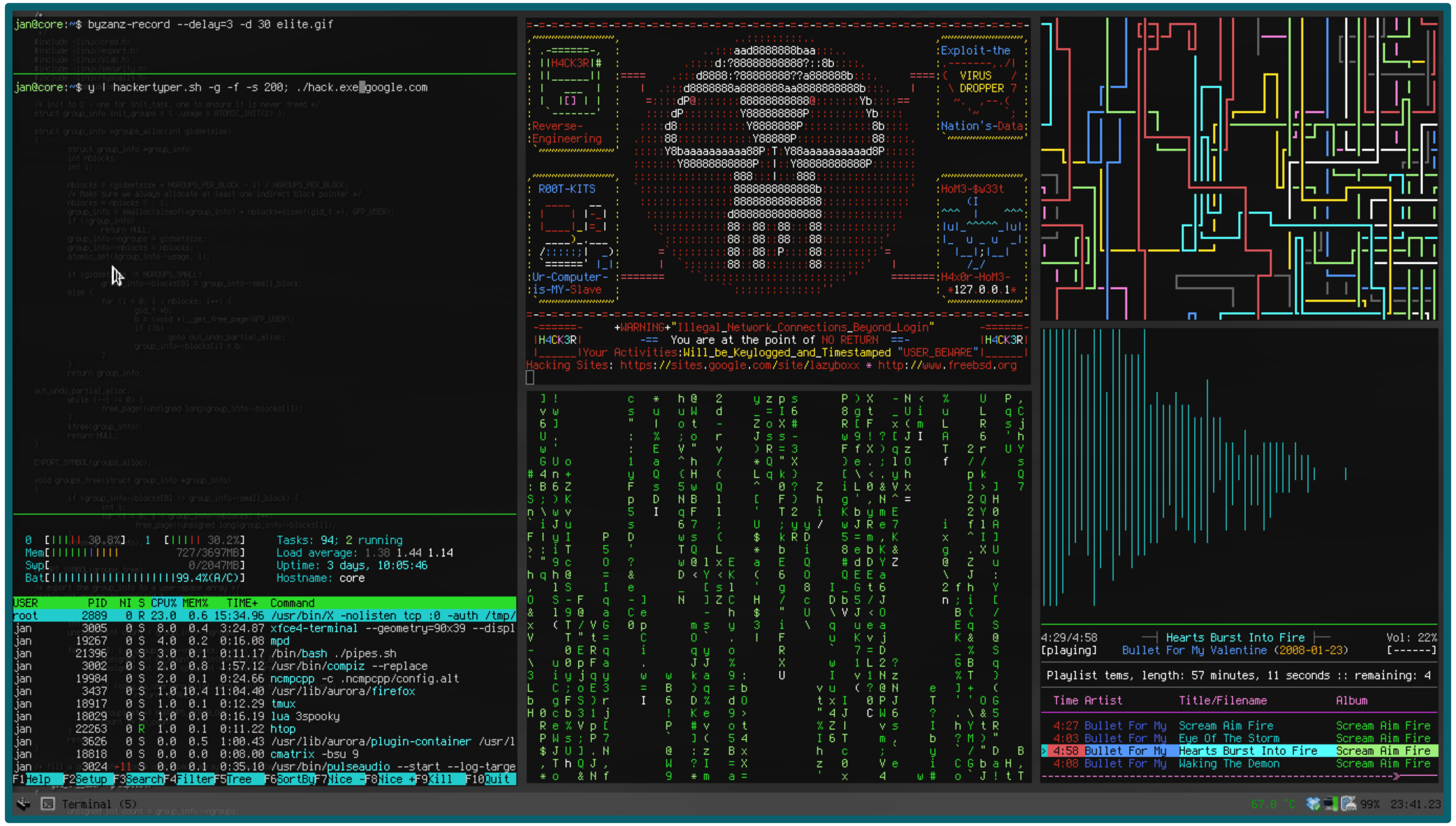
« Our API is slow because of the DB! »

« We need a bigger DB! »

OVH

No Observability

Ugly Observability



```
jan@core:~$ byzanz-record --delay=3 -d 30 elite.gif
```

```
jan@core:~$ y | hackertyper.sh -g -f -s 200; ./hack.exe google.com
```

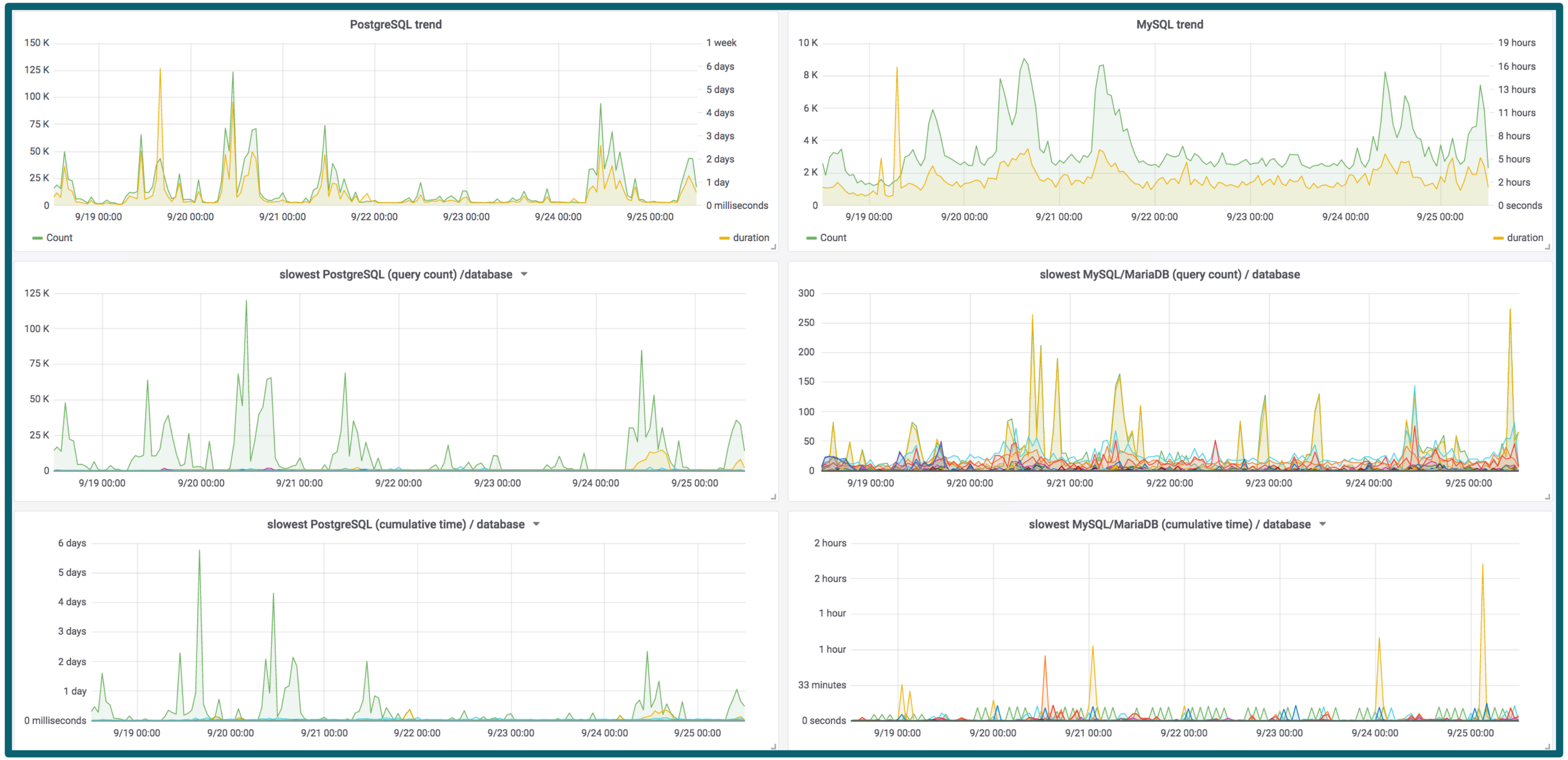
```
  0  [|||||  30.8%]   1  [|||||  30.2%]     Tasks: 94; 2 running
  Mem[|||||||||||        727/3697MB]     Load average: 1.38 1.44 1.14
  Swp[                     0/2047MB]     Uptime: 3 days, 10:05:46
  Bat[||||||||||||||||||||99.4%(A/C)]     Hostname: core

USER       PID  NI S CPU% MEM%   TIME+  Command
root      2889   0 R 23.0  0.6 15:34.96 /usr/bin/X -nolisten tcp :0 -auth /tmp/
jan       3005   0 S  8.0  0.4  3:24.87 xfce4-terminal --geometry=90x39 --displ
jan      19267   0 S  4.0  0.2  0:16.08 mpd
jan      21396   0 S  3.0  0.1  0:11.17 /bin/bash ./pipes.sh
jan       3002   0 S  2.0  0.8  1:57.12 /usr/bin/compiz --replace
jan      19984   0 S  2.0  0.1  0:24.66 ncmpcpp -c .ncmpcpp/config.alt
jan       3437   0 S  1.0 10.4 11:04.40 /usr/lib/aurora/firefox
jan      18917   0 S  1.0  0.1  0:12.29 tmux
jan      18029   0 S  1.0  0.0  0:16.19 lua 3spooky
jan      22263   0 R  1.0  0.1  0:11.22 htop
jan       3626   0 S  0.0  0.5  1:00.43 /usr/lib/aurora/plugin-container /usr/l
jan      18818   0 S  0.0  0.0  0:08.00 cmatrix -bsu 9
jan       3024 -11 S  0.0  0.1  0:35.10 /usr/bin/pulseaudio --start --log-targe
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

```
 -======-      +WARNING+"Illegal_Network_Connections_Beyond_Login"       -======-
 |H4CK3R|          -==  You are at the point of NO RETURN  ==-            |H4CK3R|
 |______|Your Activities:Will_be_Keylogged_and_Timestamped "USER_BEWARE"|______|
Hacking Sites: https://sites.google.com/site/lazyboxx * http://www.freebsd.org
```

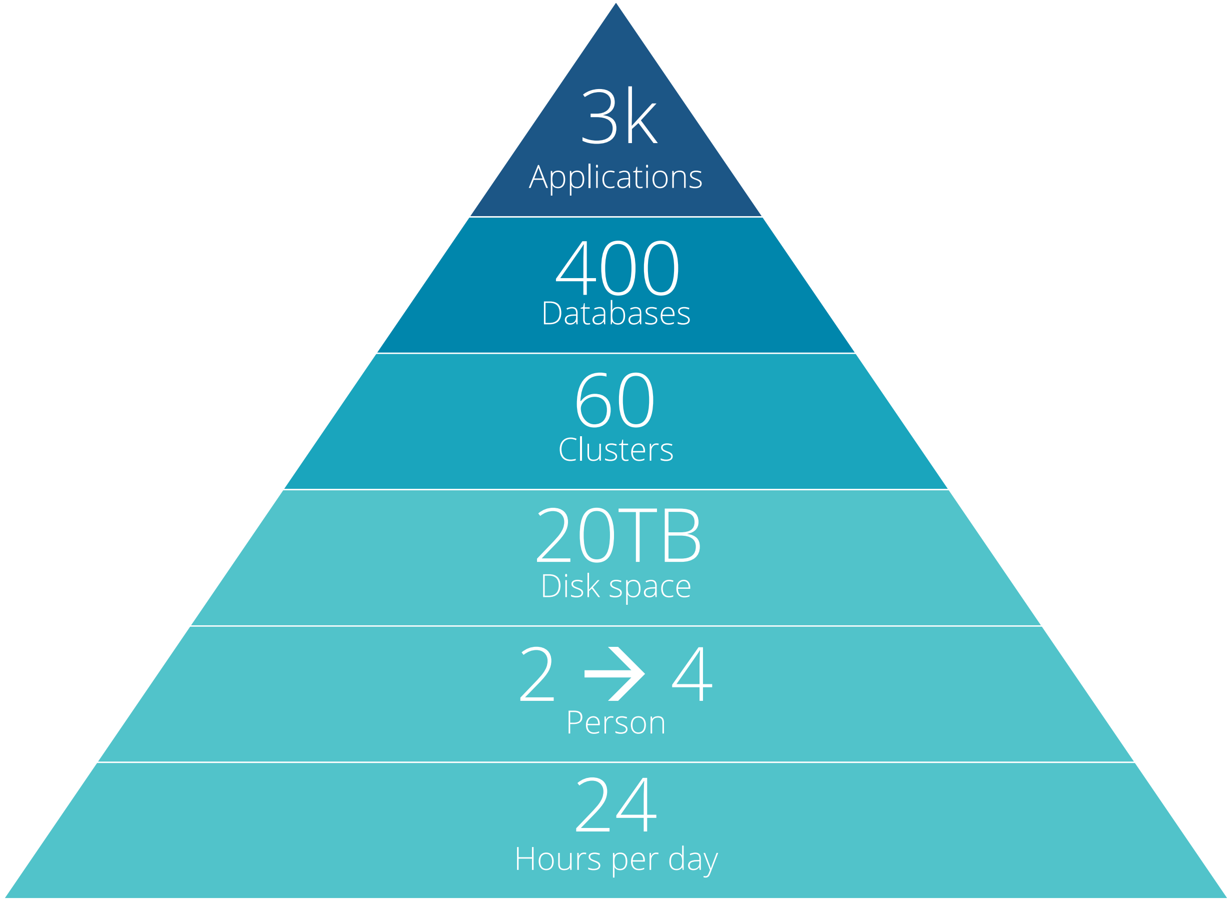
```
4:29/4:58        ─┤ Hearts Burst Into Fire ├─              Vol: 22%
[playing]    Bullet For My Valentine (2008-01-23)         [------]

 Playlist tems, length: 57 minutes, 11 seconds :: remaining: 4

  Time Artist        Title/Filename          Album

  4:27 Bullet For My Scream Aim Fire          Scream Aim Fire
  4:03 Bullet For My Eye Of The Storm         Scream Aim Fire
> 4:58 Bullet For My Hearts Burst Into Fire   Scream Aim Fire
  4:08 Bullet For My Waking The Demon         Scream Aim Fire
```

OVH

Good Observability



Our internal Databases perimeter



3k
Applications

400
Databases

60
Clusters

20TB
Disk space

2 → 4
Person

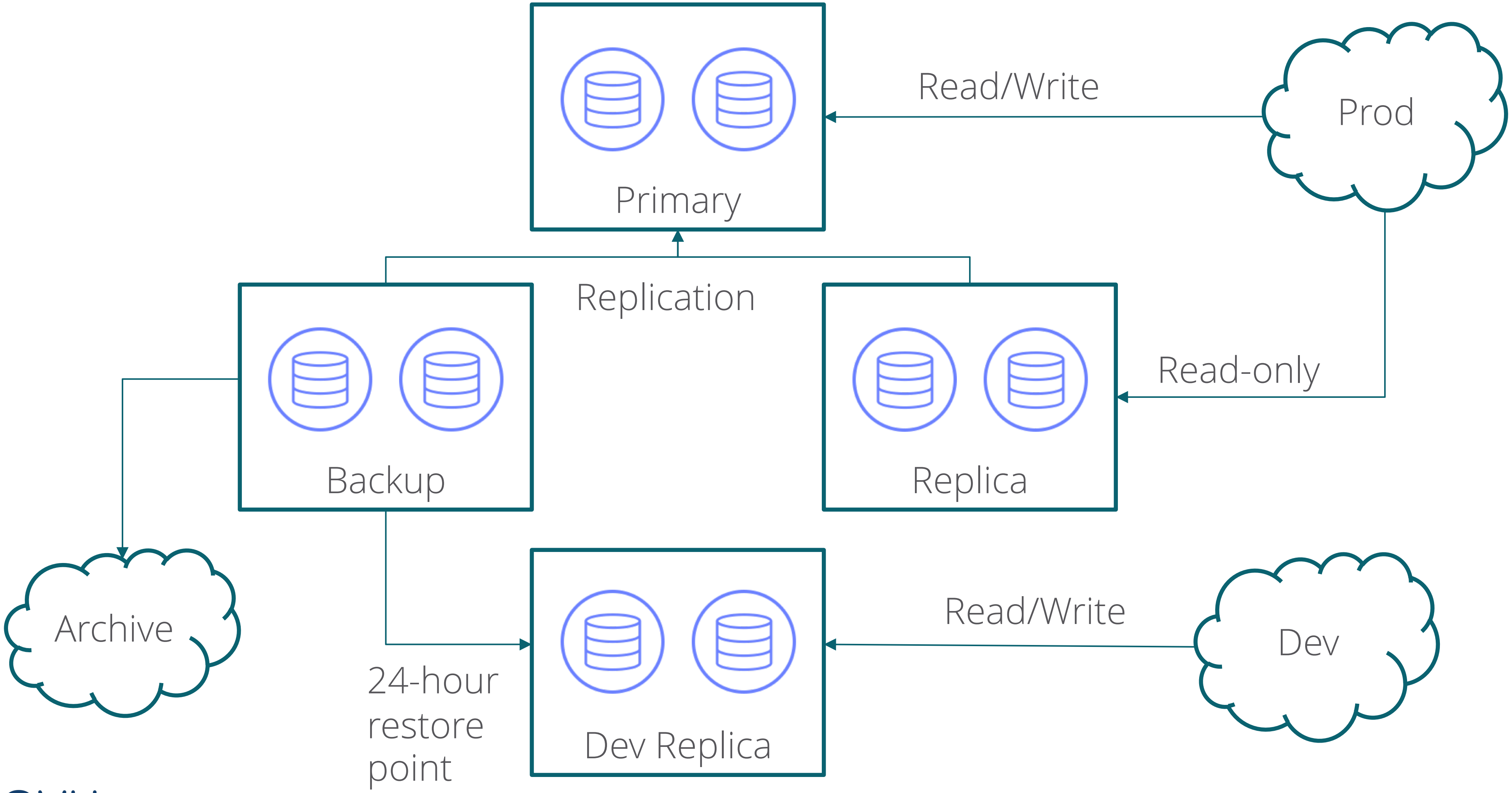
24
Hours per day

How to isolate a SLOW query

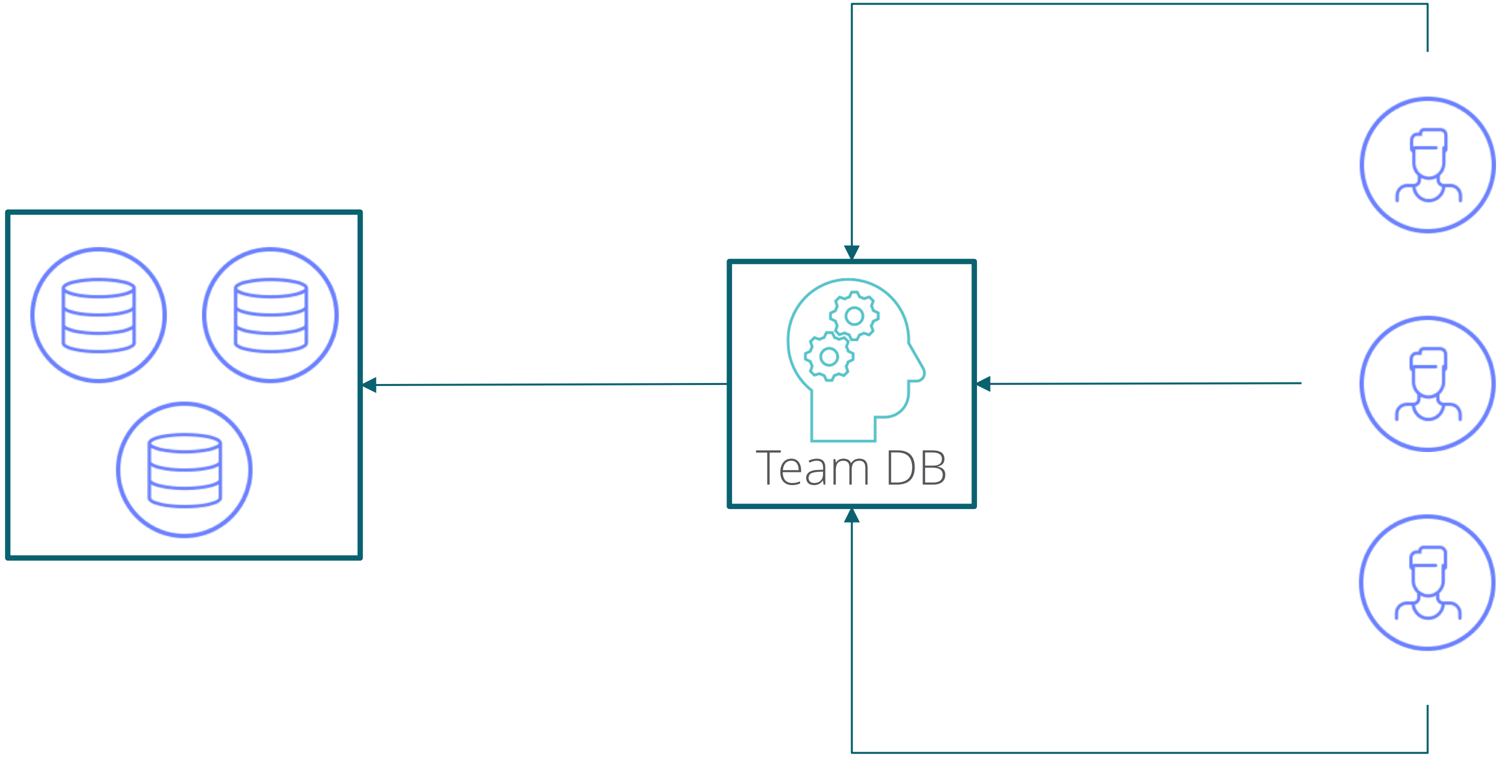
```
db# SELECT username
          FROM customers
          WHERE username LIKE ‘_wilfried%’;

Time: 9433.400 ms (00:09.433)
```

Tips: it’s not a missing index ;)

Internal DBs Infra



Old request process



"Why is my DB slow?"

"Can I have an extract from my logs?"

Easier, Better, Faster, Stronger

- Self-service for complete autonomy
- Meaningful KPIs
- Capacity planning
- Ownership and proactivity

Prerequisites

- As-a-service
  - OVH Logs
  - OVH Metrics
  - UI
- Open source
  - Data collection

Logs

First things first

Observability is not about

how to collect data

but

what to do with it

DBMS Configuration

- PostgreSQL
  - Everything happens in postgresql.conf
  - Format log to produce report
    - log_line_prefix = '%t [%p]: [%l-1] db=%d,user=%u,app=%a,client=%h '
  - Log slow queries
    - log_min_duration_statement = '1000'

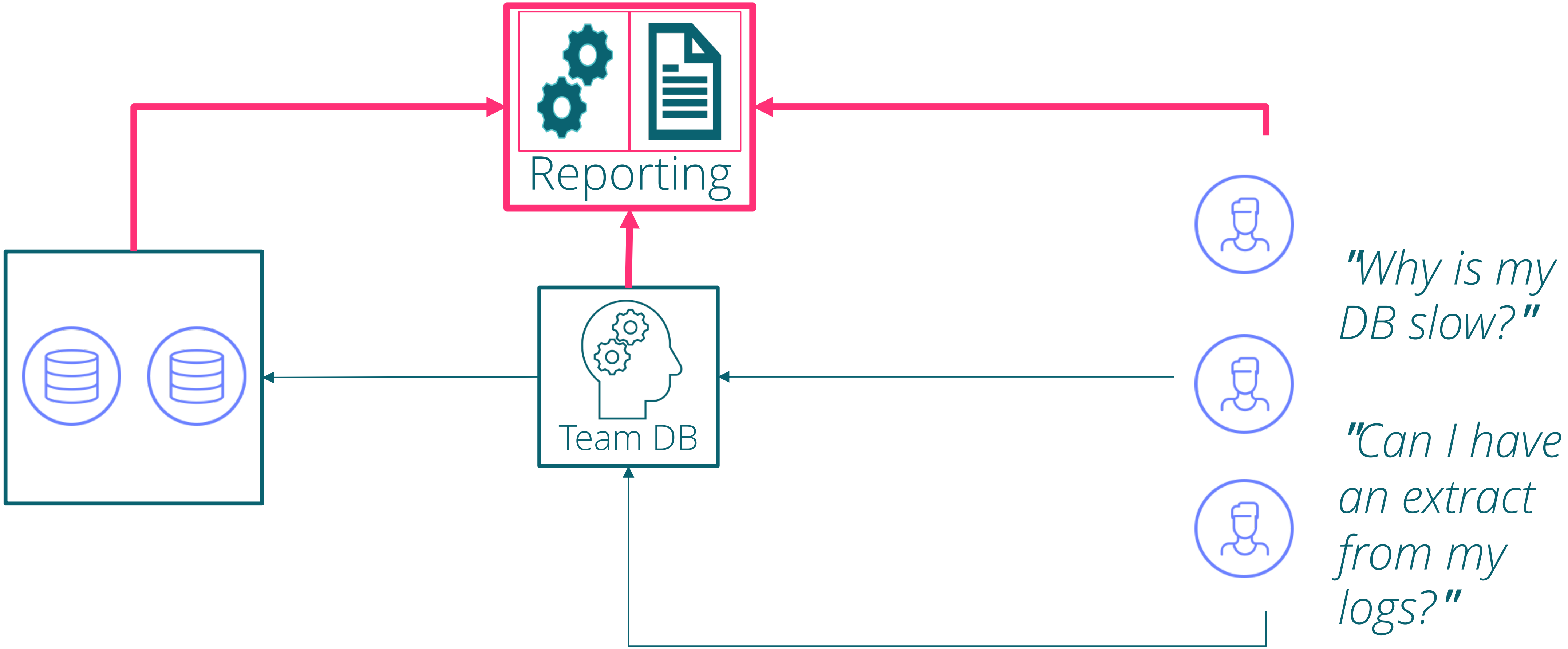
DBMS Configuration

- MySQL / MariaDB
  - Everything happens in my.cnf
  - Performance Schema
  - Log slow queries
    - slow_query_log = 1
    - slow_query_log_file = /var/log/mysql/slow.log

Give meaning to logs

- Process log files and produce fully readable reports
  - pgbadger for PostgreSQL
  - pt-query-digest for MySQL/MariaDB

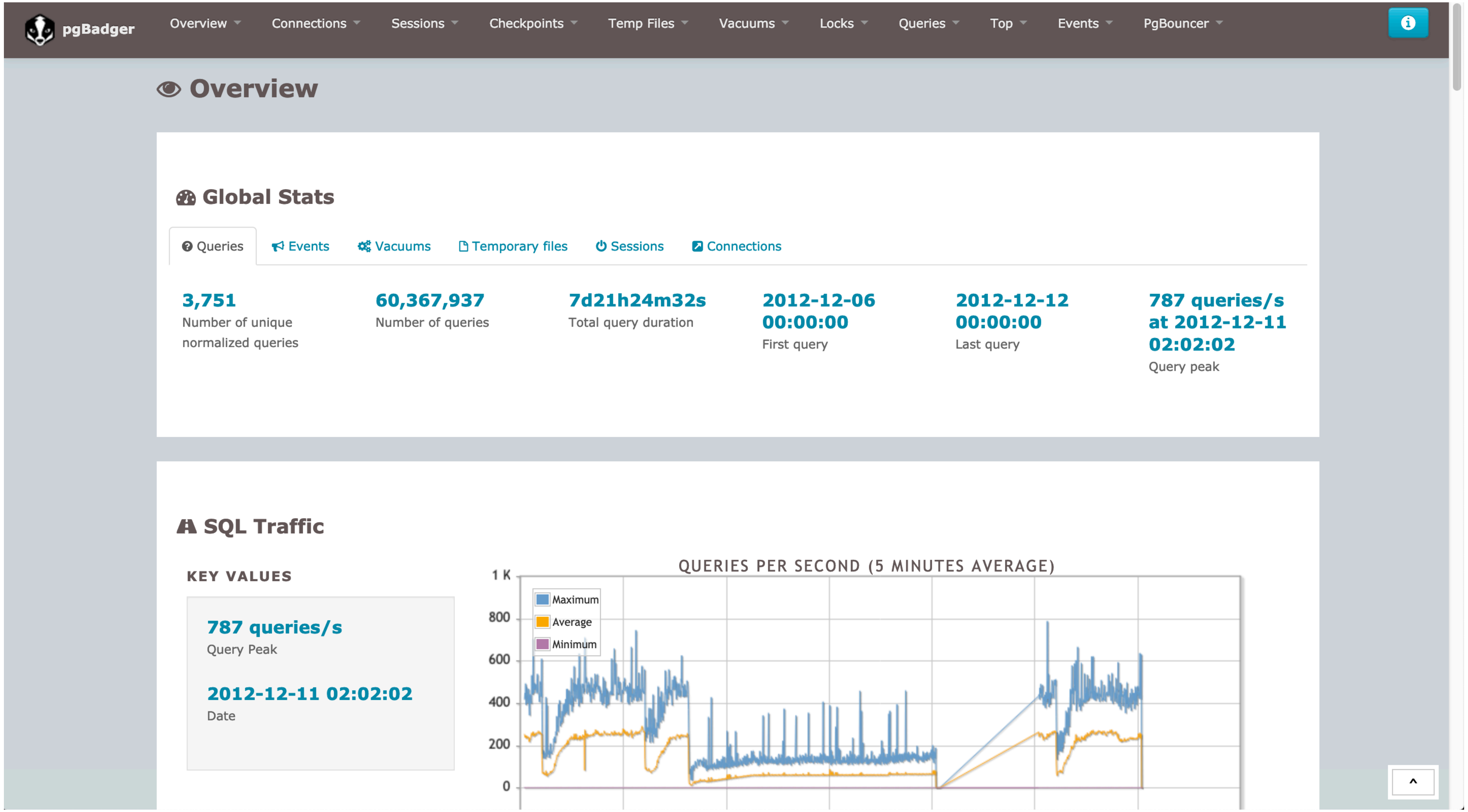
Observability: step 1/5



"Why is my DB slow?"

"Can I have an extract from my logs?"

Output Sample



Overview

Global Stats

3,751
Number of unique normalized queries

60,367,937
Number of queries

7d21h24m32s
Total query duration

2012-12-06 00:00:00
First query

2012-12-12 00:00:00
Last query

787 queries/s at 2012-12-11 02:02:02
Query peak

SQL Traffic

KEY VALUES

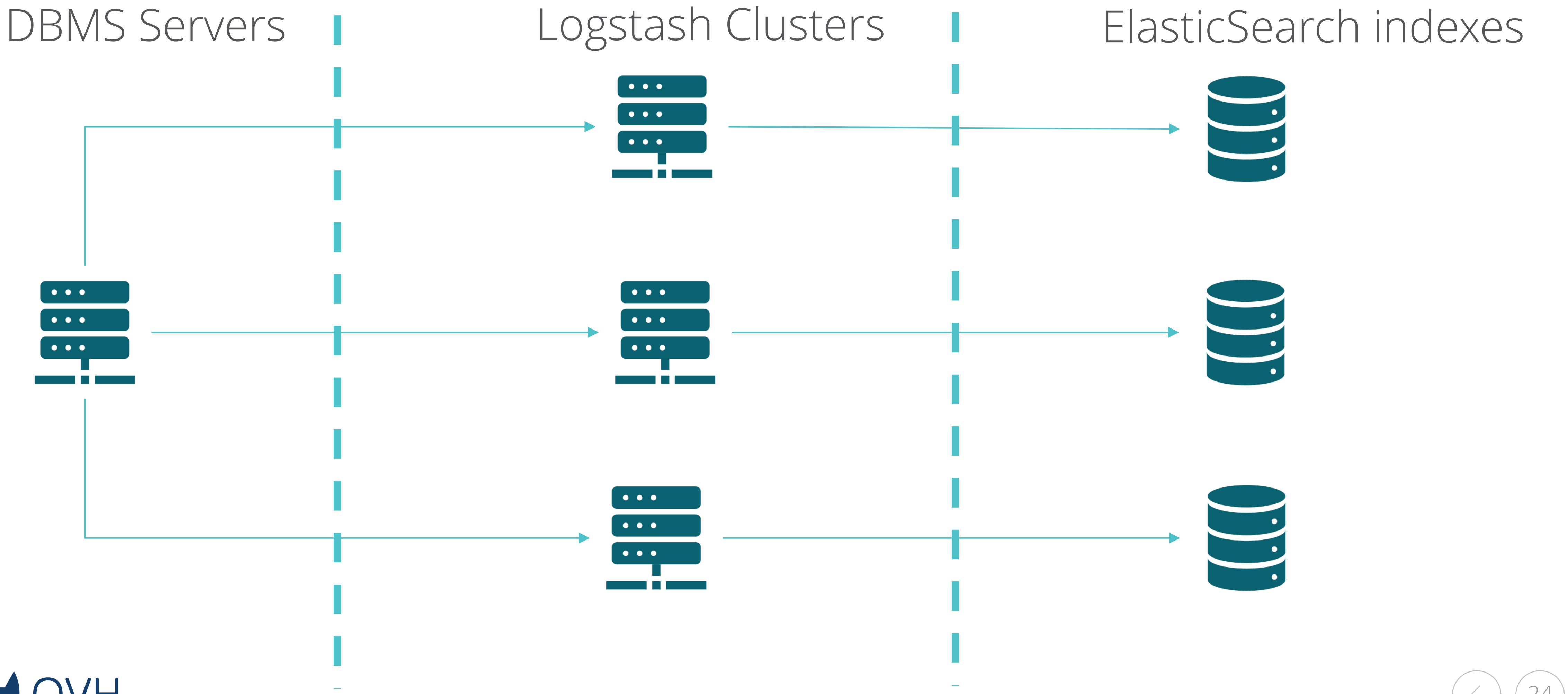
787 queries/s
Query Peak

2012-12-11 02:02:02
Date

Logs, Logs, Logs...

- Logs are useful only if they are usable
  - <3 grep, less, ...
  - Avoid ssh

Logs pipeline



Let’s grok

```
User@Host: app_user @ [1.2.3.4]
```

```
^# User@Host: %{WORD:query_user}\s*@\s*\[%{IP:query_ip}?\].*
```

```
{
    "query_user": "app_user",
    "query_ip": "1.2.3.4"
}
```

Sending Data

- Use whatever is Efficient && Easy to setup for your case
  - Filebeat
  - Syslog

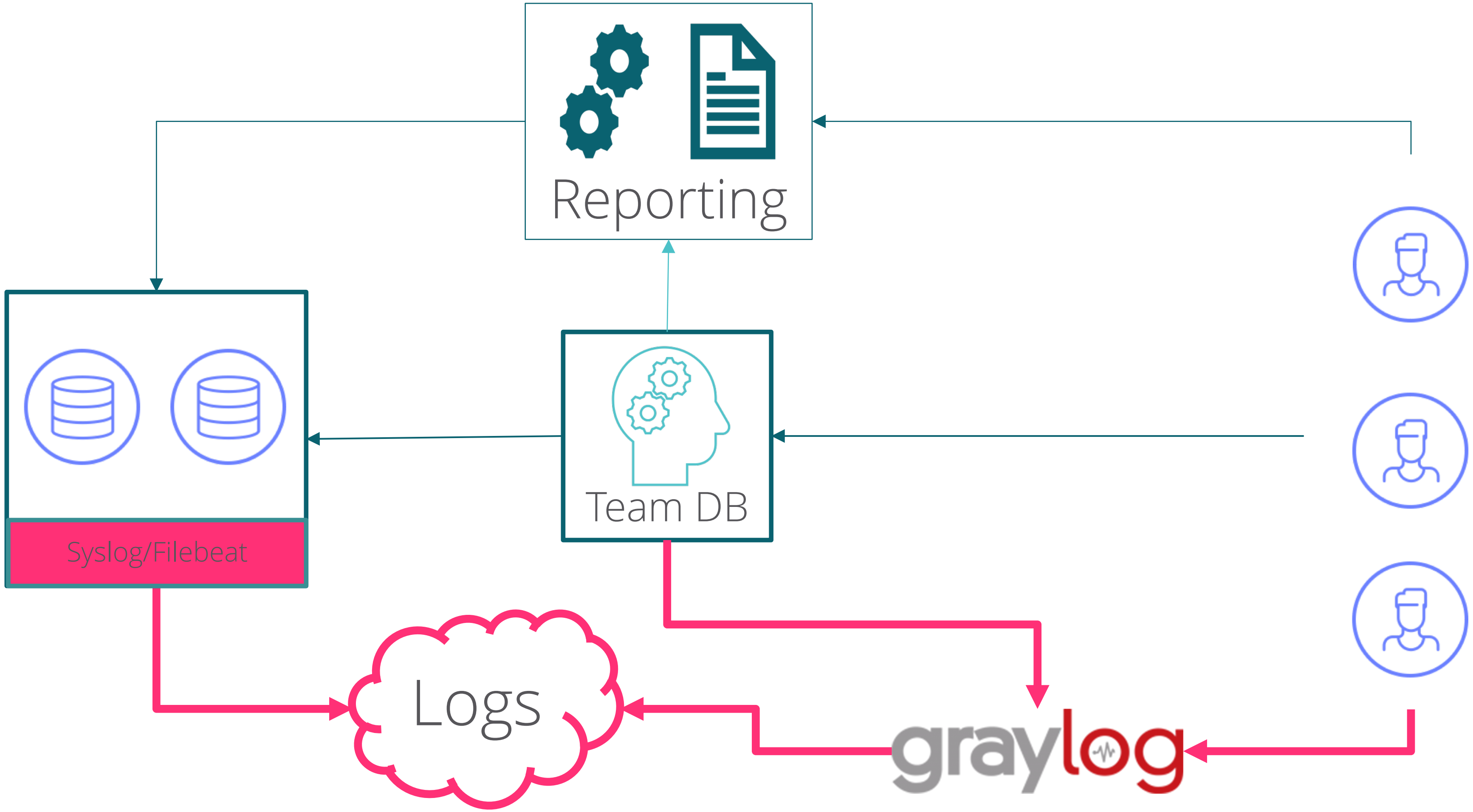
Filebeat

```
# /etc/filebeat /filebeat.yml
output.logstash:
  hosts: ["graX.logs.ovh.com:6514"]
  ssl.certificate_authorities: ["/etc/filebeat/cert.pem"]

filebeat.prospectors:
- type: log
  enabled: true
  paths:
    - /var/log/postgresql/postgresql.log
```

Syslog

```
destination d_pg_ldp {
    tcp( "graX.logs.ovh.com",
        port(6514),
        ts_format("iso"),
        keep-alive(yes),
        so_keepalive(yes),
        log-fifo-size(10000),
    );
};
```

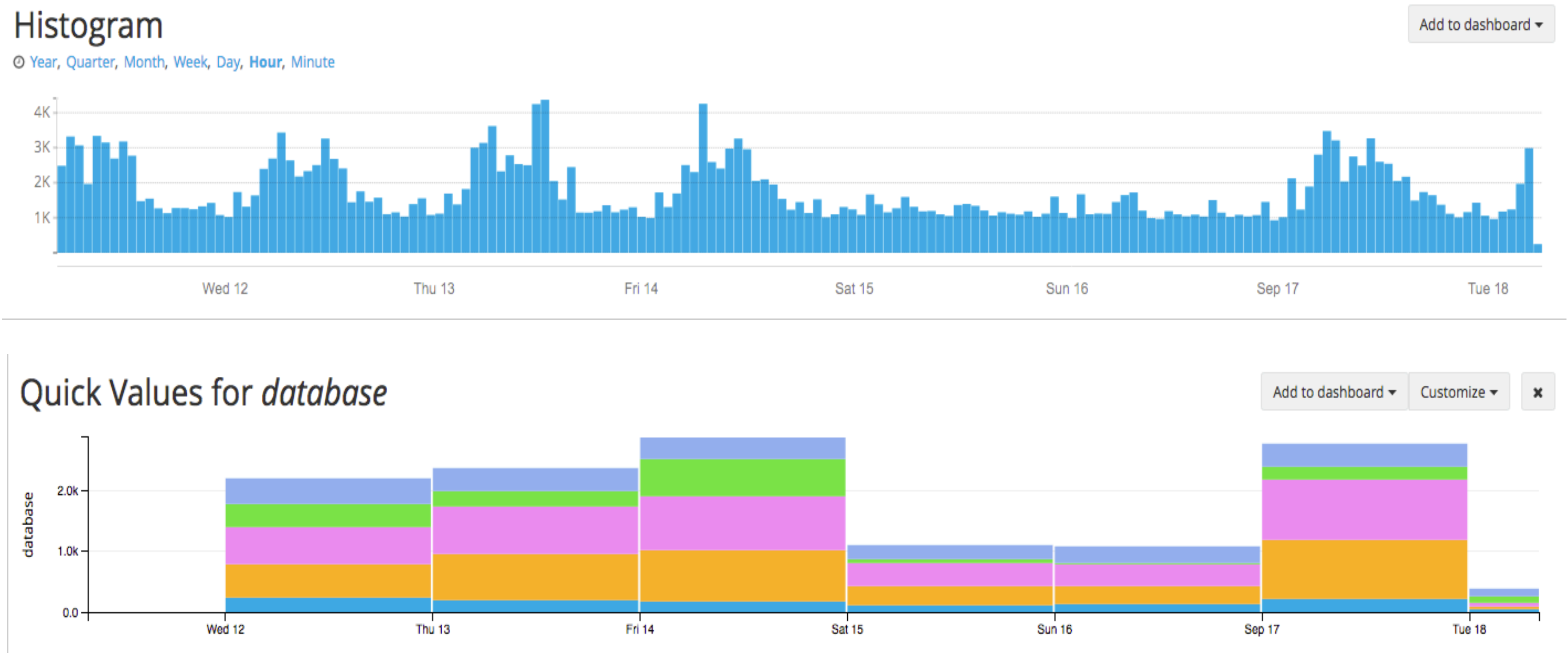
Observability: step 2/5



"Why is my DB slow?"

"Can I have an extract from my logs?"

Graylog



Histogram

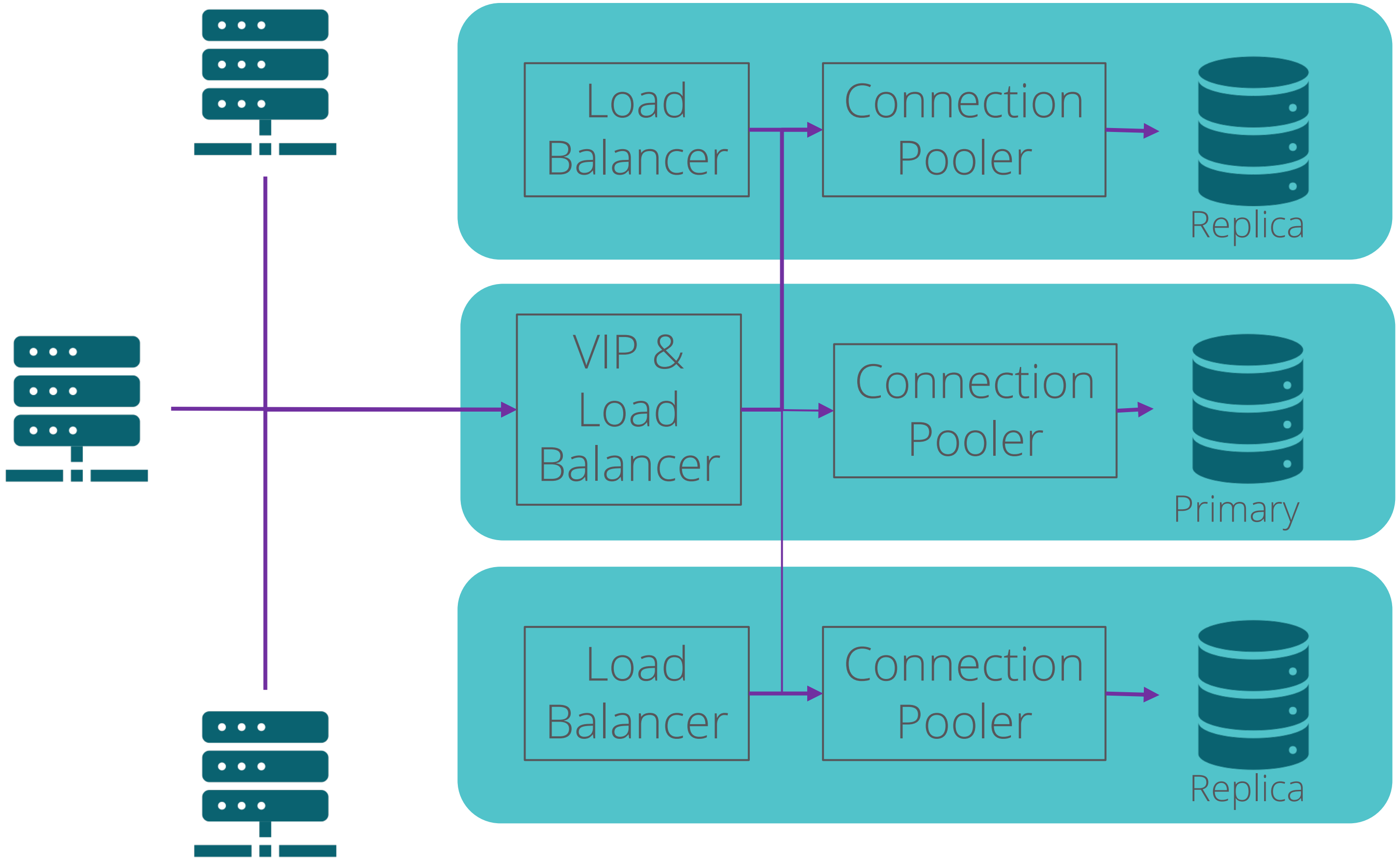
Year, Quarter, Month, Week, Day, **Hour**, Minute

Quick Values for *database*

Oops, Proxy and Connection Pooler...

With our setup we lost Source IP:

- Less usefull pg_hba
- Harder to track bad queries
- Security audit



… Proxy Protocol

What about Proxy Protocol?

We want to help:

- Proxy Protocol support @ PostgreSQL Hackers → https://bit.ly/2MN2H8U
- PR#390 @ pgbouncer → https://github.com/pgbouncer/pgbouncer/pull/390

Metrics

It's not about how

Metrics

is not about which tools

to use But

what to do with your *data*

Collect your metrics

- There are several options for doing this:
  - Homemade → please don’t do that
  - Collectl, Collectd, Statsd…
  - Telegraf, Prometheus…

Telegraf’s plug-ins

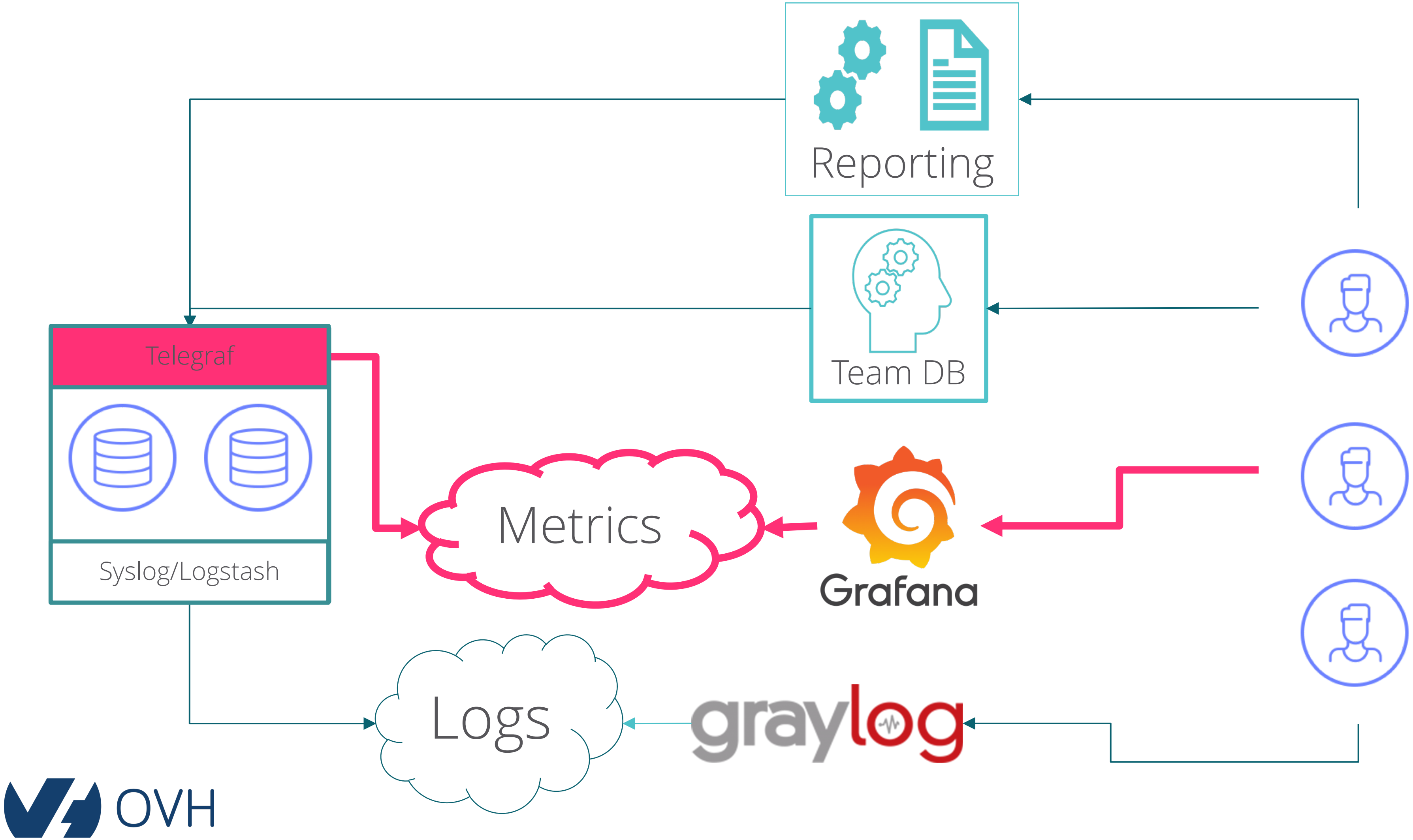
- Plug-ins:
  - System-related
    - cpu, disk, diskio, system, network …
  - DBMS
    - PostgreSQL, MySQL
  - Homemade
    - Exec

- There is a lot more plug-ins in telegraf

Let’s push some metrics

```
# /etc/telegraf/telegraf.conf
[agent]
  interval = "30s"
  flush_interval = "30s"
[[outputs.influxdb]]
  urls = ["https://influxdb.graXXX.metrics.ovh.net"]
  timeout = "15s"
  username = "telegraf"
  password =  "write.token.from.metrics.manager"
```

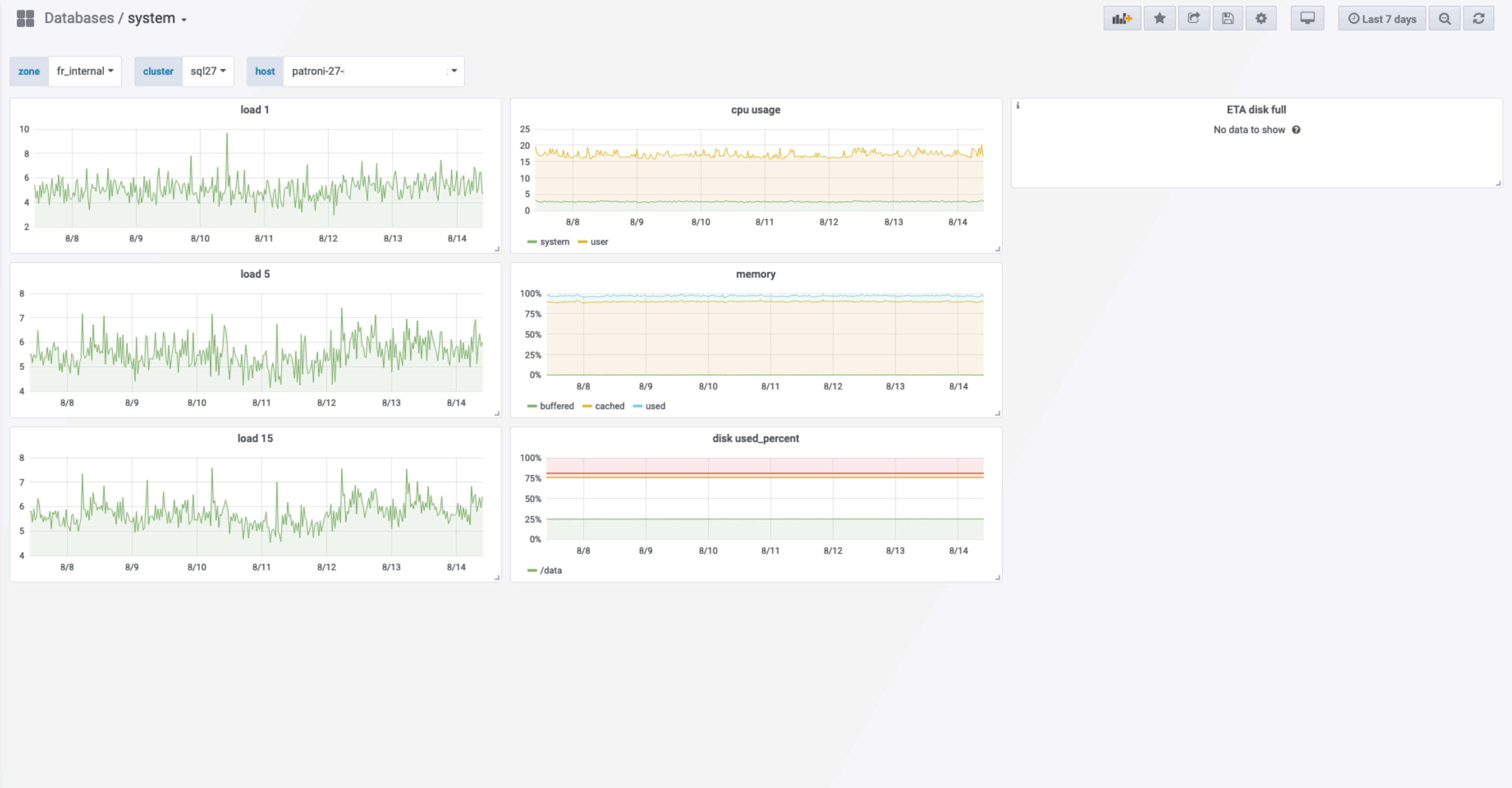
Observability: step 3/5



"Why is my DB slow?"

"Can I have an extract from my logs?"

System dashboard



Databases / system

zone fr_internal

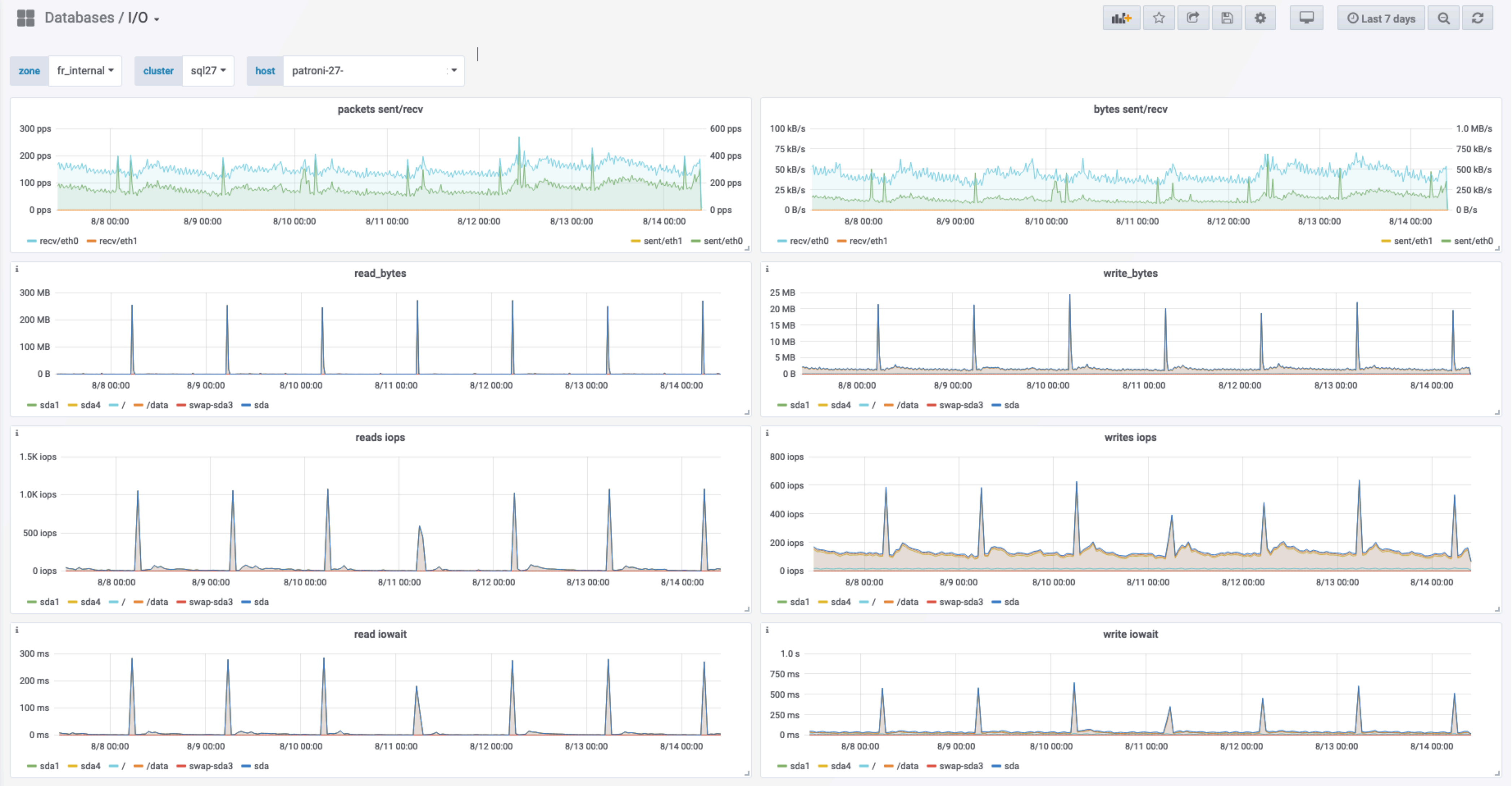
cluster sql27

host patroni-27-

ETA disk full

No data to show

I/O dashboard

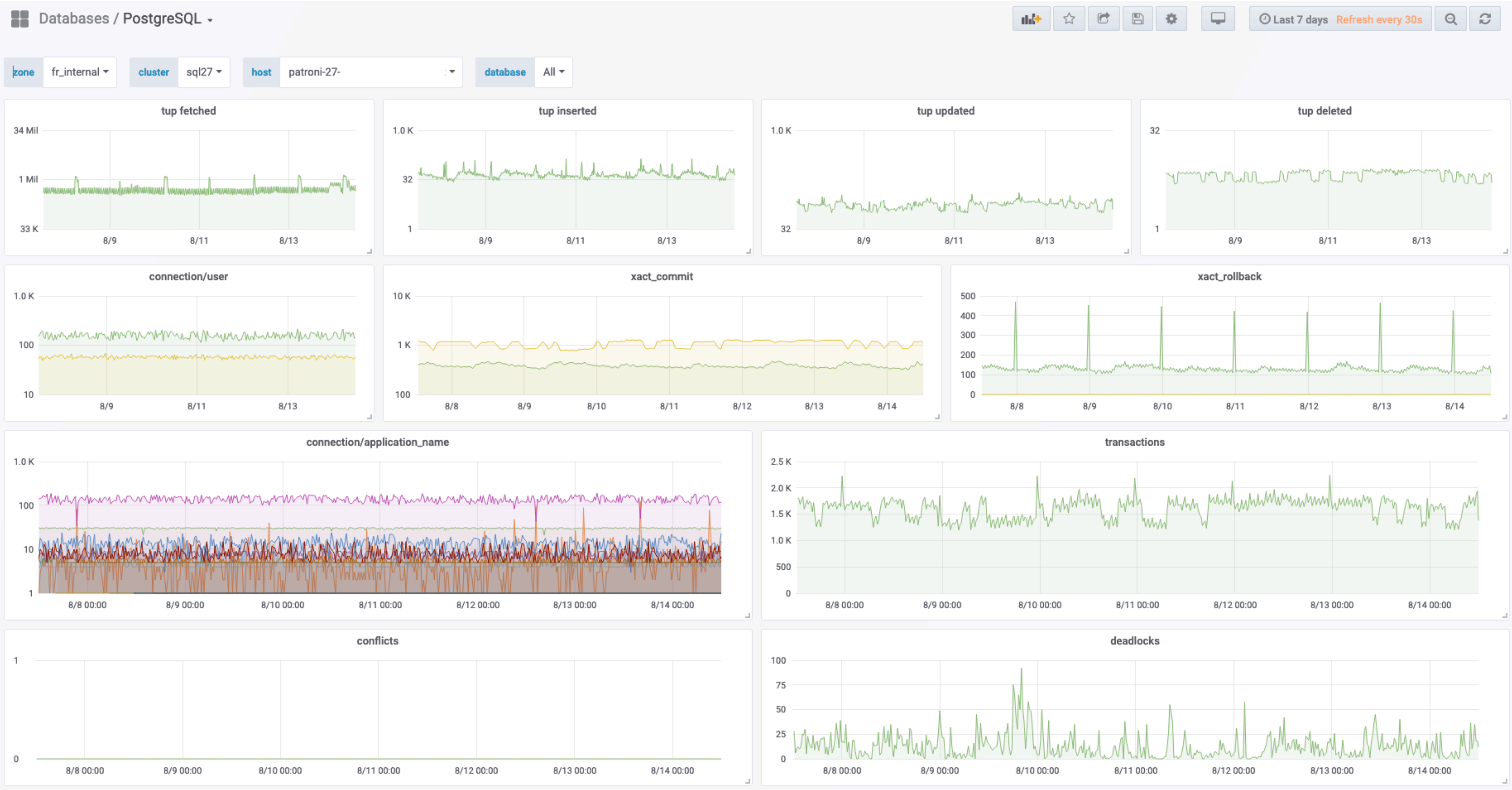


Databases / I/O

zone fr_internal · cluster sql27 · host patroni-27-

OVH

DBMS dashboard

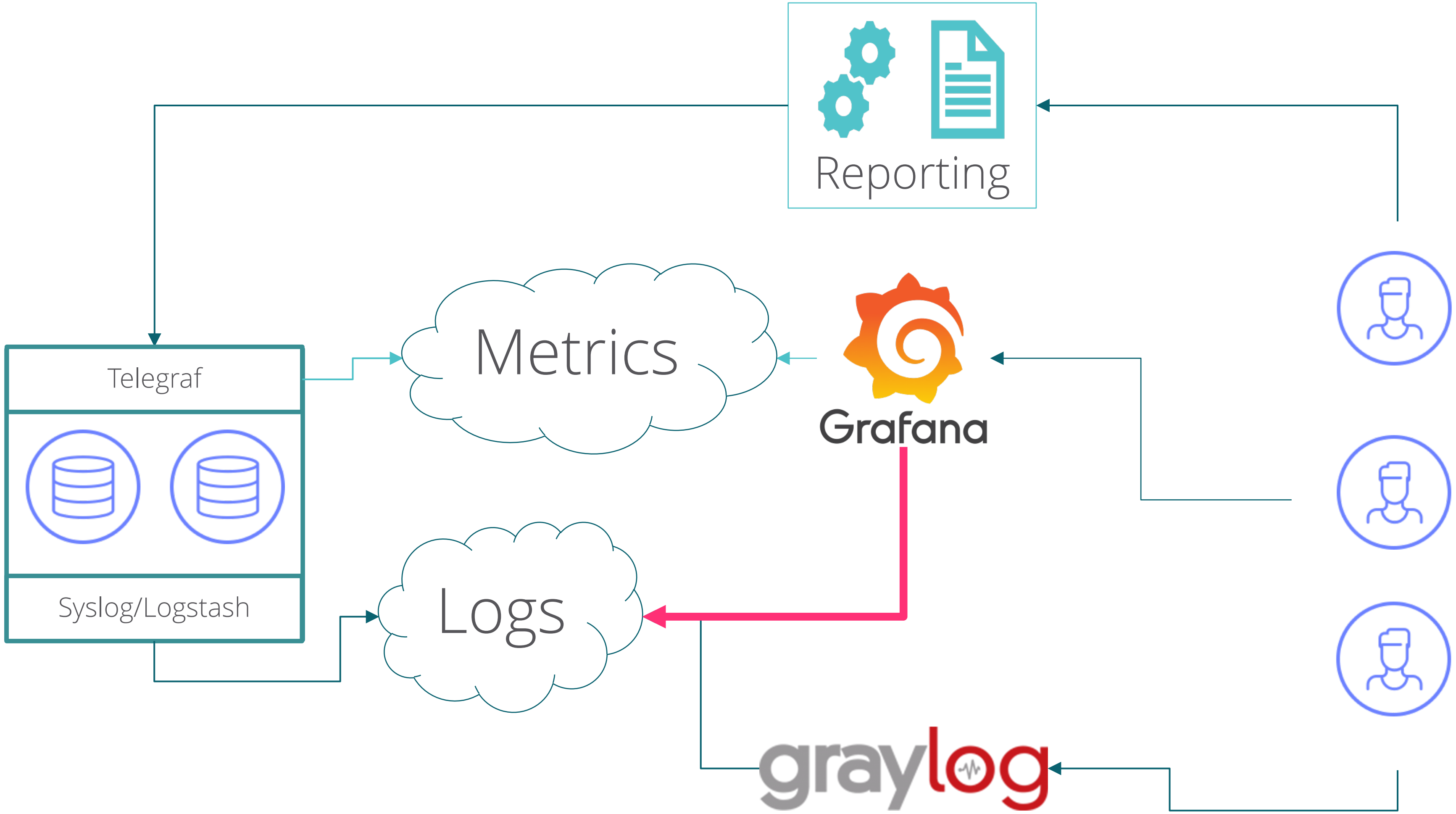


Databases / PostgreSQL

zone fr_internal · cluster sql27 · host patroni-27- · database All

Grafana is beautiful!

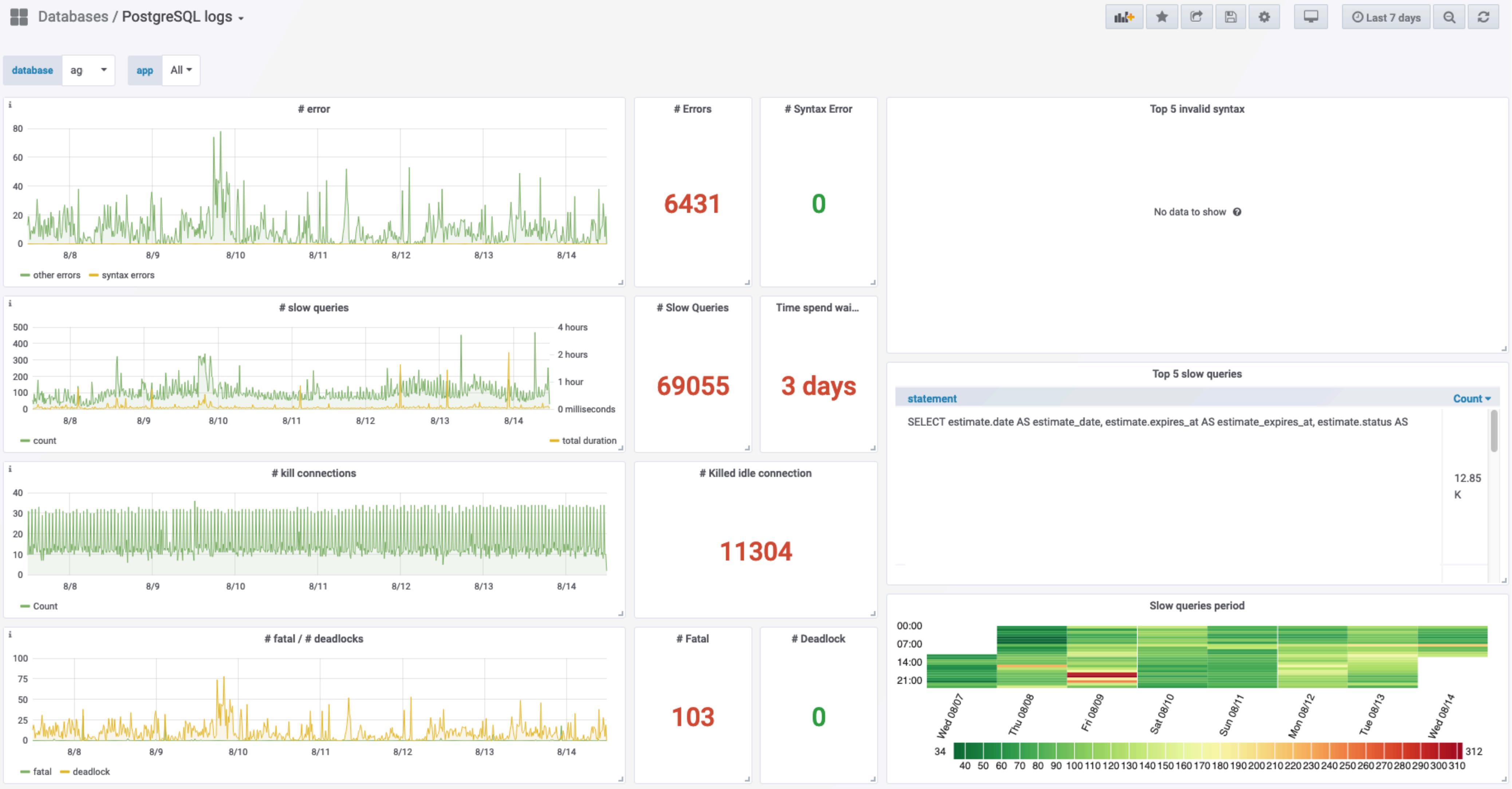
Observability: step 4/5



"Why is my DB slow?"

"Can I have an extract from my logs?"

Slow queries count/cumul. time



# Errors: 6431

# Syntax Error: 0

Top 5 invalid syntax: No data to show

# Slow Queries: 69055

Time spend wai...: 3 days

Top 5 slow queries

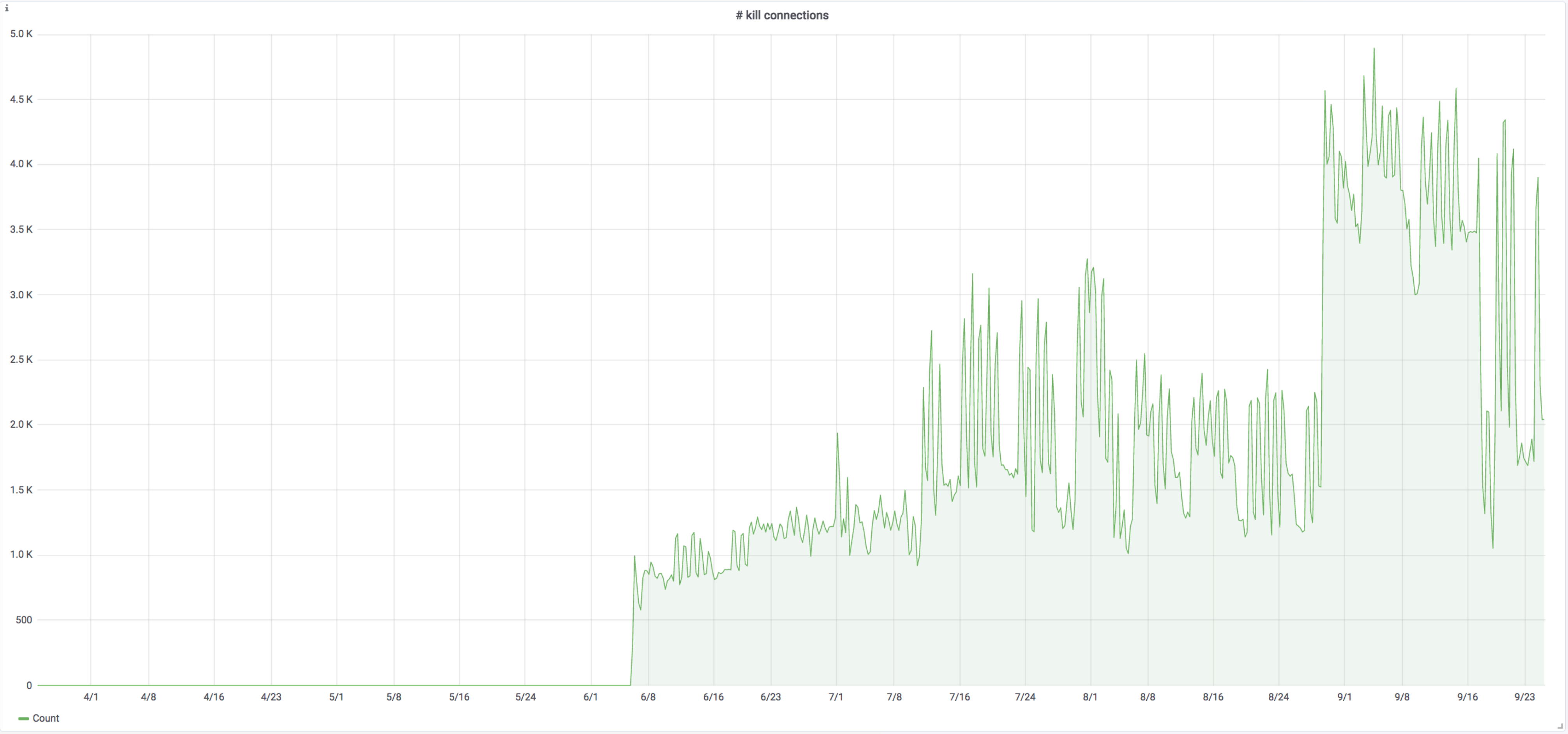
statement	Count
SELECT estimate.date AS estimate_date, estimate.expires_at AS estimate_expires_at, estimate.status AS	12.85 K

# Killed idle connection: 11304

# Fatal: 103

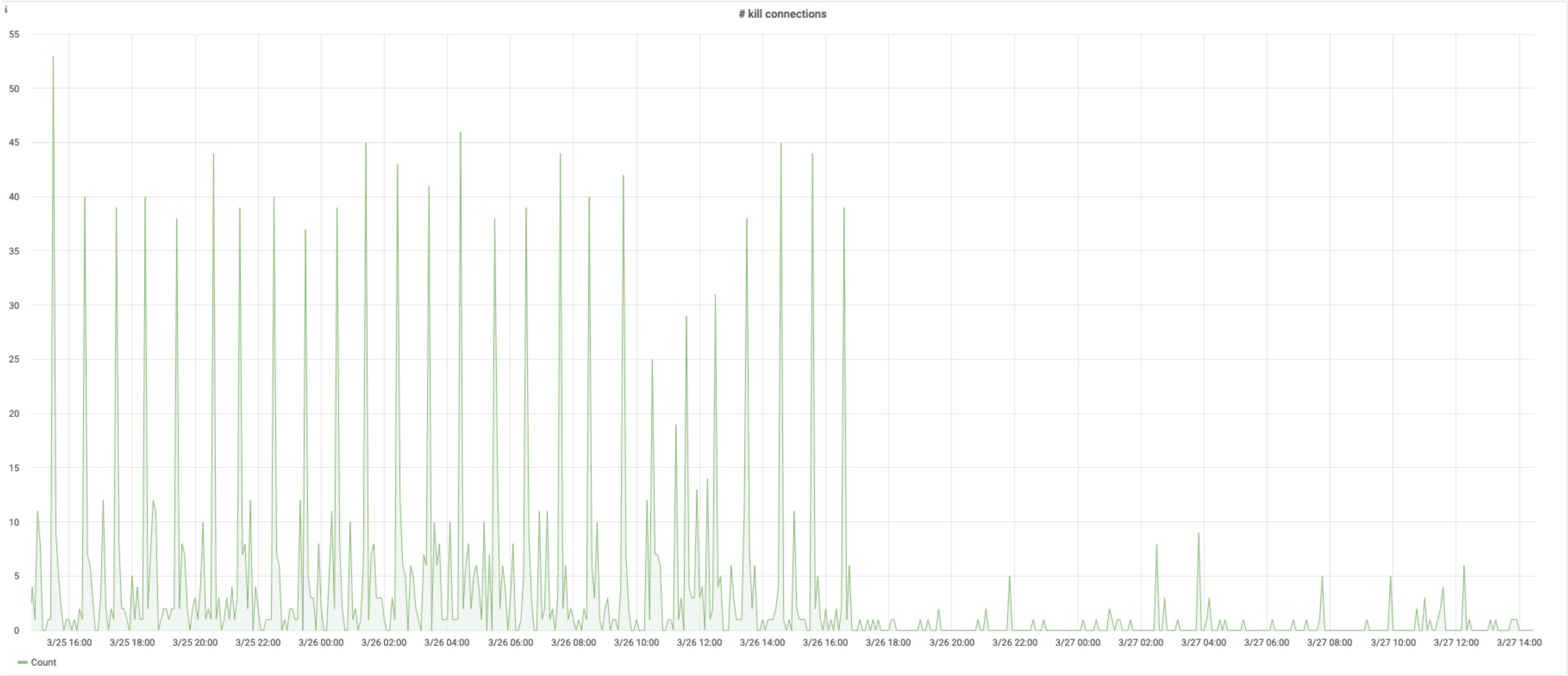
# Deadlock: 0

Spot bugs



OVH

Spot fixes



OVH

We already knew that!

- This is well known methods
  - Utilization Saturation Errors
  - Rate Errors Durations
  - Four Golden Signals

80/20

- Don’t try to fix the world…
- Focus where it matters
  - Start by your Top producers and iterate

Talk...

Hey folks, I've open 60 tickets for your database slow queries and errors.

Talk, again...

Folks, come on!
What the F***

OVH

Always talk ~~(louder)~~ …

That’s it folks! I’m done with you. No more prod for you until you fixe everything.

… But do it gently

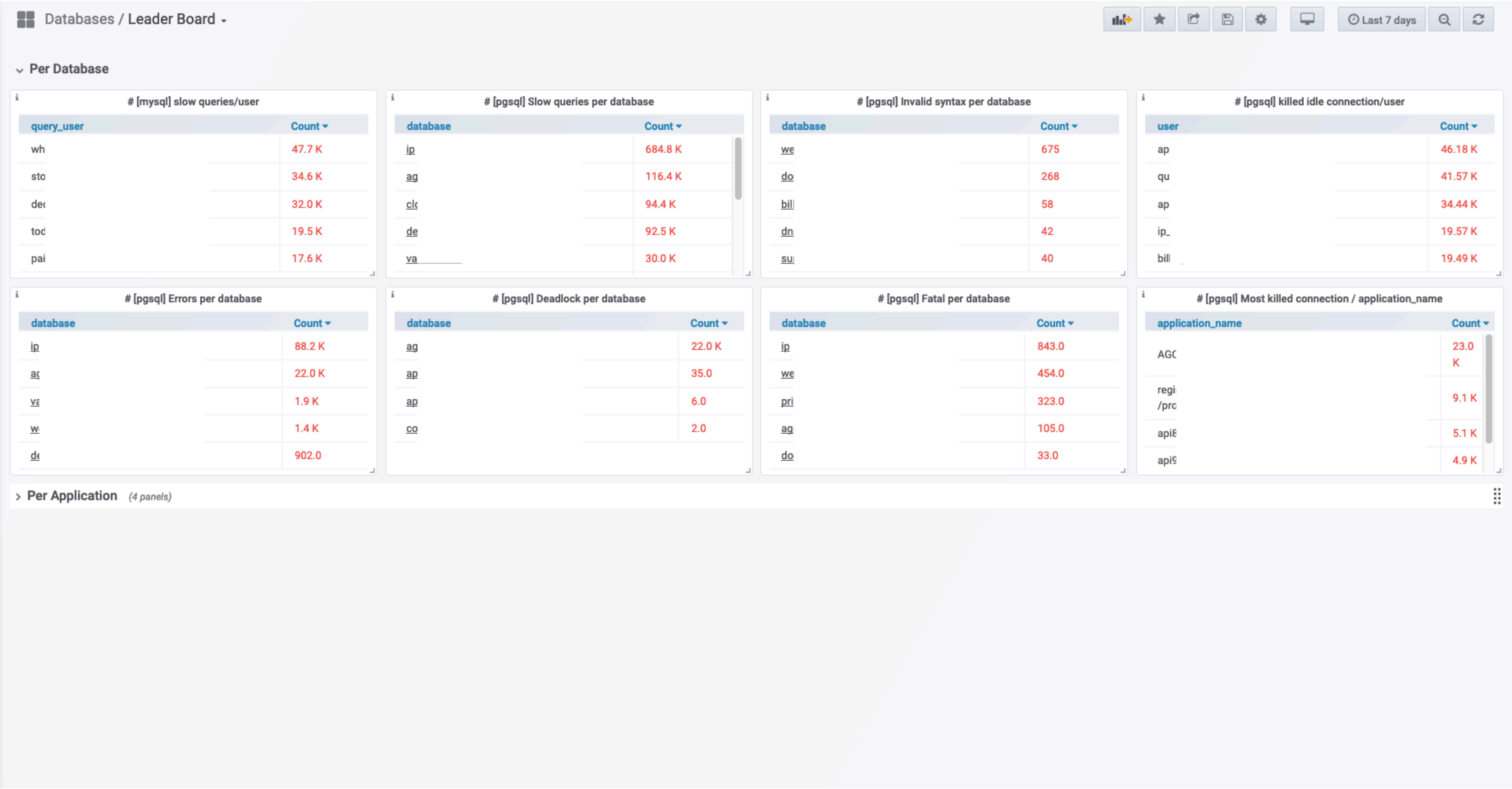
Ok folks, my bad. Let's try again.

Gamification

- Make it fun
- Make it interesting
- Turn it to a game

- Each game needs a ....

Leader Board



Per Database

[mysql] slow queries/user

query_user	Count
wh	47.7 K
sto	34.6 K
de	32.0 K
tod	19.5 K
pai	17.6 K

[pgsql] Slow queries per database

database	Count
ip	684.8 K
ag	116.4 K
clc	94.4 K
de	92.5 K
va	30.0 K

[pgsql] Invalid syntax per database

database	Count
we	675
do	268
bill	58
dn	42
su	40

[pgsql] killed idle connection/user

user	Count
ap	46.18 K
qu	41.57 K
ap	34.44 K
ip_	19.57 K
bill	19.49 K

[pgsql] Errors per database

database	Count
ip	88.2 K
ag	22.0 K
va	1.9 K
w	1.4 K
de	902.0

[pgsql] Deadlock per database

database	Count
ag	22.0 K
ap	35.0
ap	6.0
co	2.0

[pgsql] Fatal per database

database	Count
ip	843.0
we	454.0
pri	323.0
ag	105.0
do	33.0

[pgsql] Most killed connection / application_name

application_name	Count
AGC	23.0 K
regi /pro	9.1 K
api8	5.1 K
api9	4.9 K

Per Application (4 panels)

Leader Board Weekly Mail

Hello,

You will find information which can help you identify your queries in our welcome guide:

< insert documentation link >

Tldr:

- Database 33 did great because of blablabla
- Database 94 is our #1 producer of slow queries
- Blablabla

…

Engage regularly

What have we learned?

- Observability can be set up in **less than a week**
- Choose the right tools for the job
  - Grafana for dashboard (mix and match sources)
  - Graylog for search
  - Reporting tools are not used by developers
- 1 year down the road, we can expect…
  - A widely-adopted gamechanger
  - Inspire others to do the same kind of report <3
  - **x4 less slow queries**
- KPIs

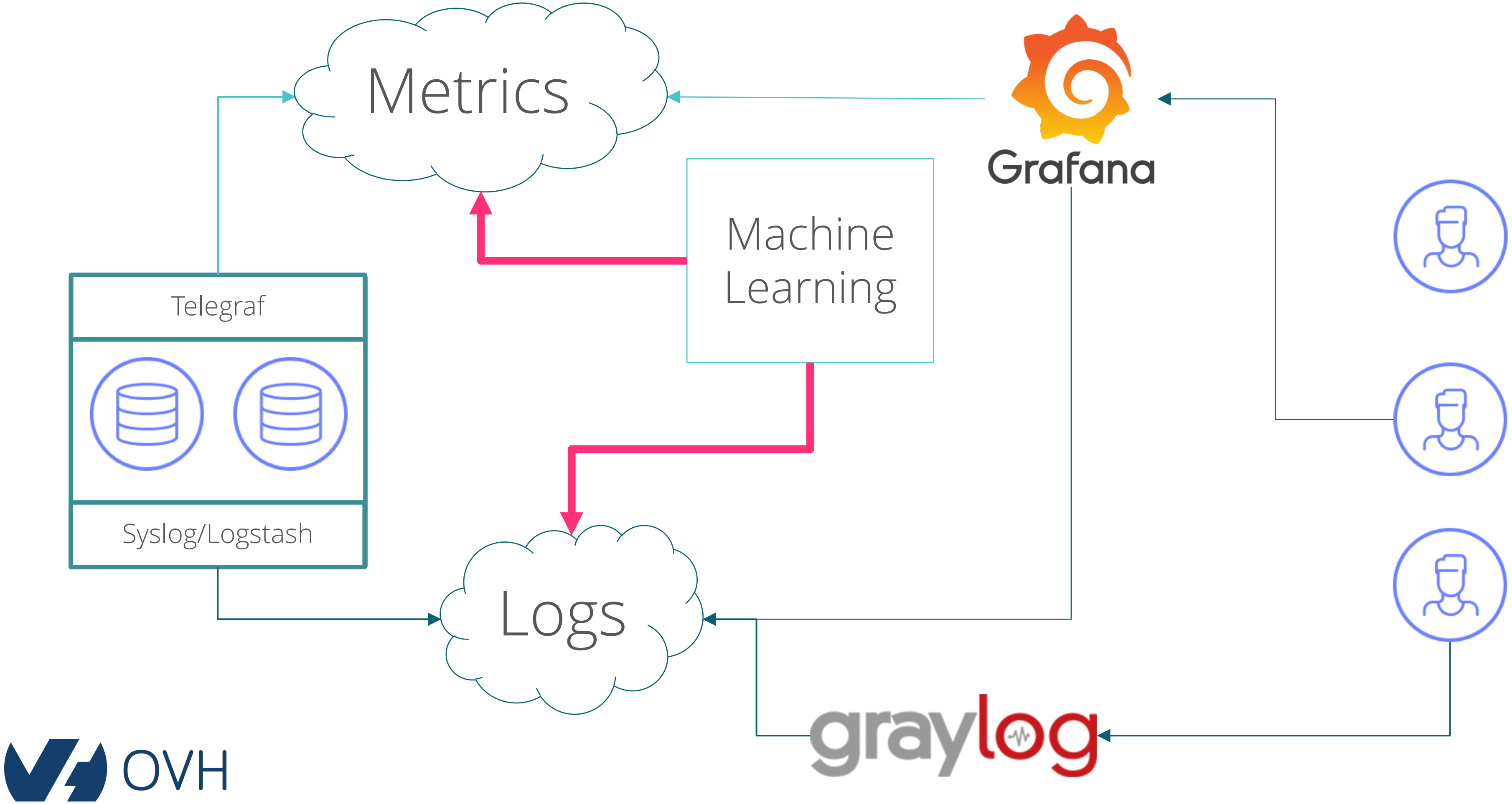
What’s next?

- The easy part is done, we need to dig deeper
  - Engage at higher level
  - Point everyone in the right direction… Broadcast the KPI
  - Help Developper refactor app & schema

- Improve monitoring
- Feed the data to Machine Learning

- Automatic indexes recommendations
  - Thanks to Percona blog
  - Based on pg_qualstats & hypopg

Observability: step 5/5



"Why is my DB slow?"

"Can I have an extract from my logs?"

Remember this SLOW query

```
db# SELECT username
        FROM customers
        WHERE username LIKE ‘_wilfried%’;

Time: 9433.400 ms (00:09.433)
```

Tips: it’s not a missing index ;)

PS: We are hiring!

- Opensource database expert
- Site Reliability Engineers (Private cloud, Openstack, DNS, Deploy, Observability)
- Software engineers (containers, baremetal, webhosting)
- Back-end developers (go, python)
- Engineering manager webhosting
- … And a lot more

Questions?