POSTGRESQL MONITORING IN ZALANDO

Helsinki PostgreSQL Meetup

October 2019

POSTGRESQL IN ZALANDO

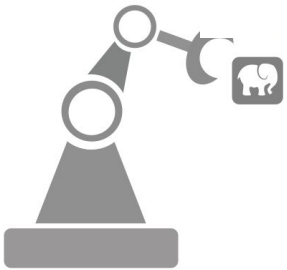
POSTGRESQL IN ZALANDO

Zalando works with PostgreSQL since approx. 2010 (running in DC)

2015 - migration to AWS (using RDS);
Patroni and Spilo have been started.

2016-2017 - using provided PostgreSQL clusters
(based on Spilo/STUPS);

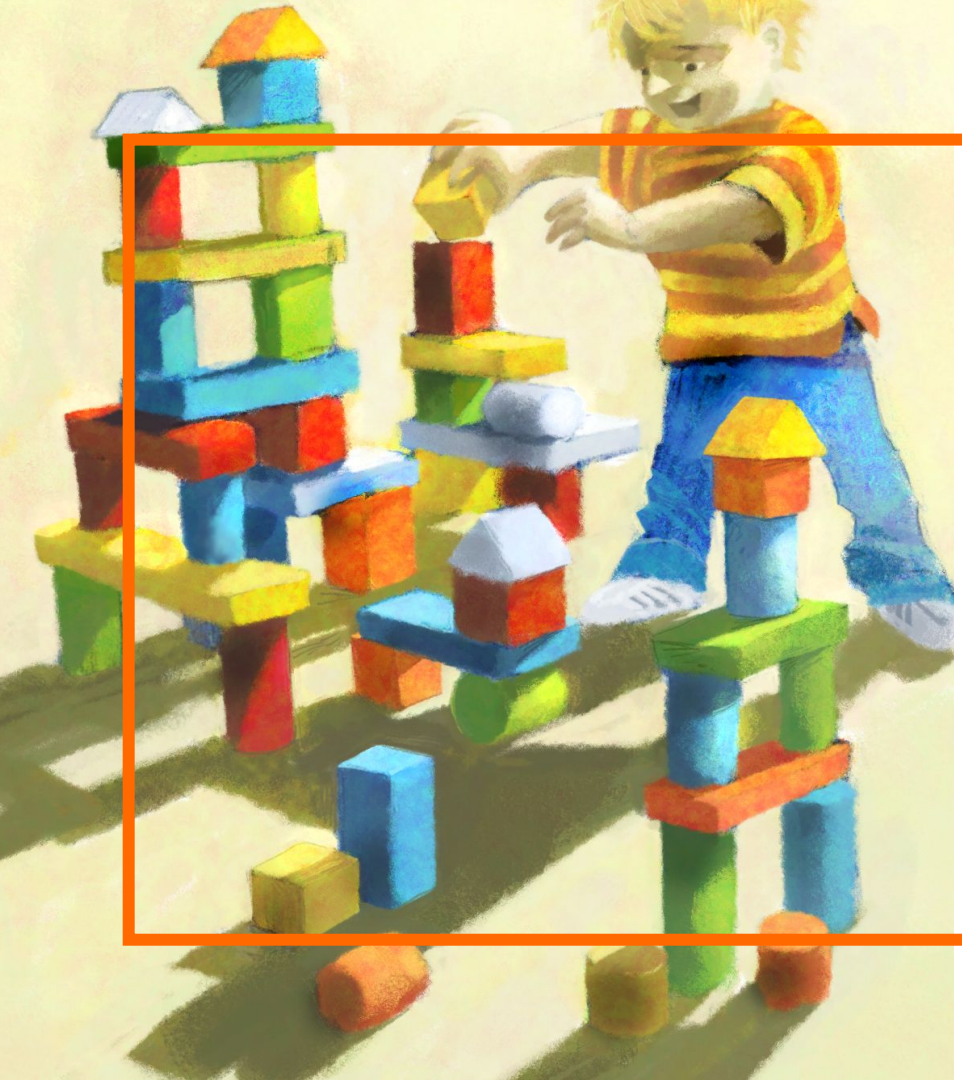
2018-... - using PostgreSQL operator.



POSTGRESQL OPERATOR

When a new postgresql custom resource appears, the operator creates:

1. **StatefulSet** for PostgreSQL/Patroni cluster
2. **Service** for master node (ClusterIP or LB)
3. **Service** for replica nodes (ClusterIP or LB)
4. Extra DNS names for the services *if needed*



ZMON - ZALANDO MONITORING SYSTEM

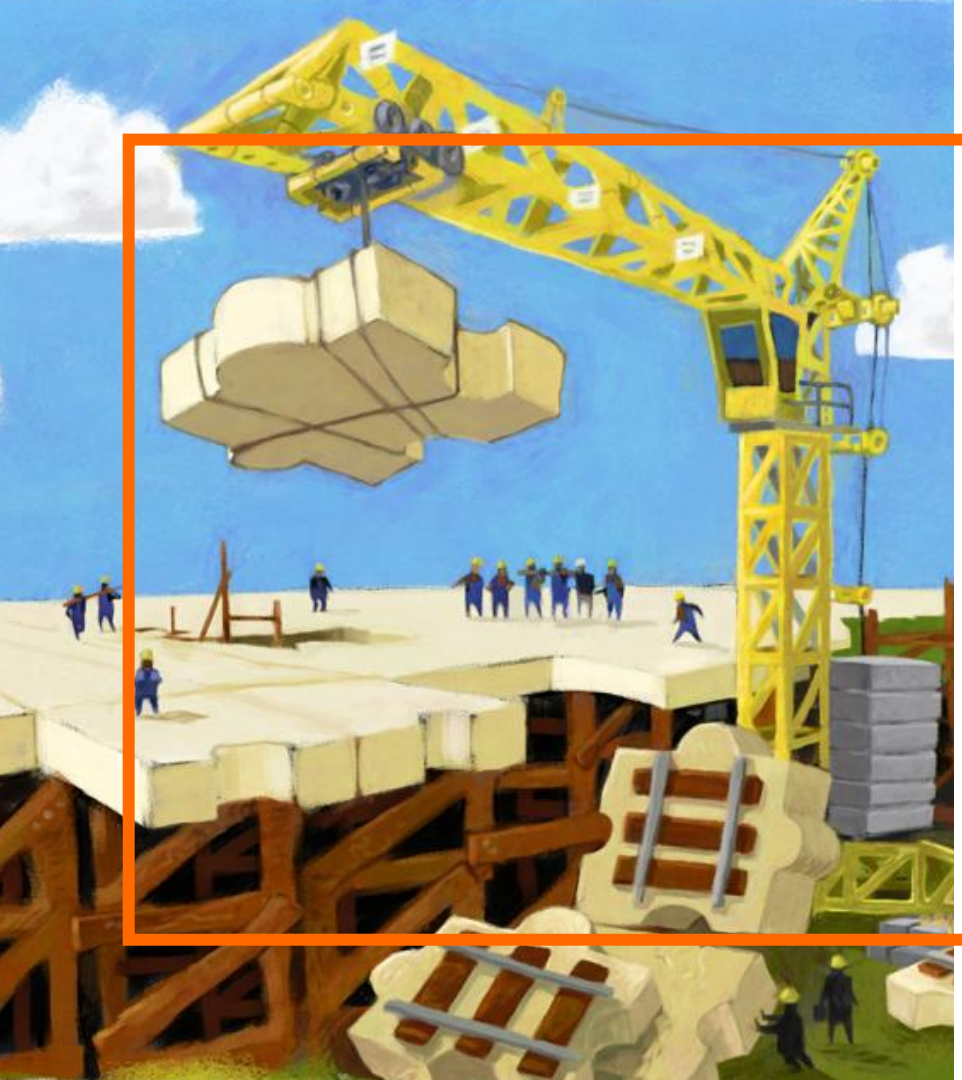
zalando

Few facts about ZMON

- Created in 2013 during a HackWeek, in production since 2014;
- Optimized for our use-case: autonomous teams, multiple Kubernetes clusters, common central storage;
- Open-sourced (APL2.0) and available from GitHub;
- Stores time-series data in KairosDB (based on top of Cassandra);
- Stores infrastructure data in PostgreSQL.

What's important for us?

- ZMON uses the "pull" model, its workers *fetch* the metrics
- ZMON is a *distributed* monitoring system, the workers are running in every K8s cluster;
- Autodiscover results in the most up-to-date view of our apps;
- Checks are centralized and may be shared between the teams;
- PostgreSQL Patroni/Spilo clusters are fully supported.



INFRASTRUCTURE MONITORING

What are the metrics we need?

- Kubernetes
  - Pods CPU
  - Network I/O
  - Free space in Persistent Volumes
  - Open TCP connections for PostgreSQL processes
- AWS
  - EBS I/O
  - ELB throughput (rarely)
  - Backup S3 bucket

Amazon Cloudwatch

Prometheus

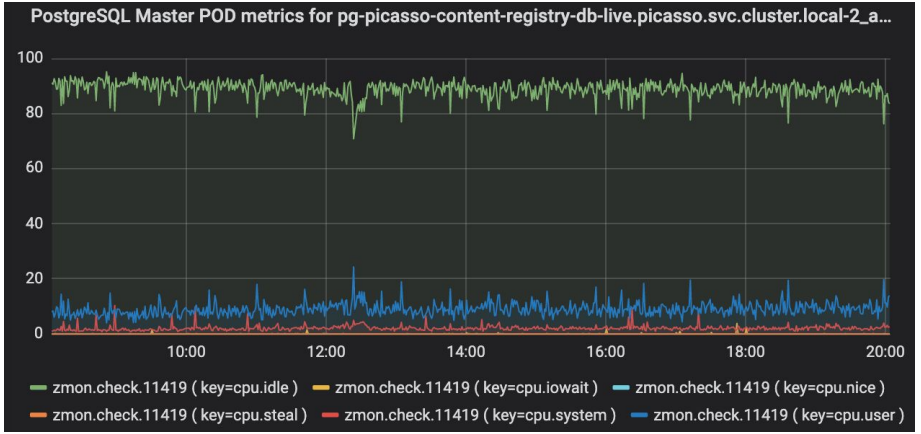
Description

Monitor PostgreSQL cluster masters

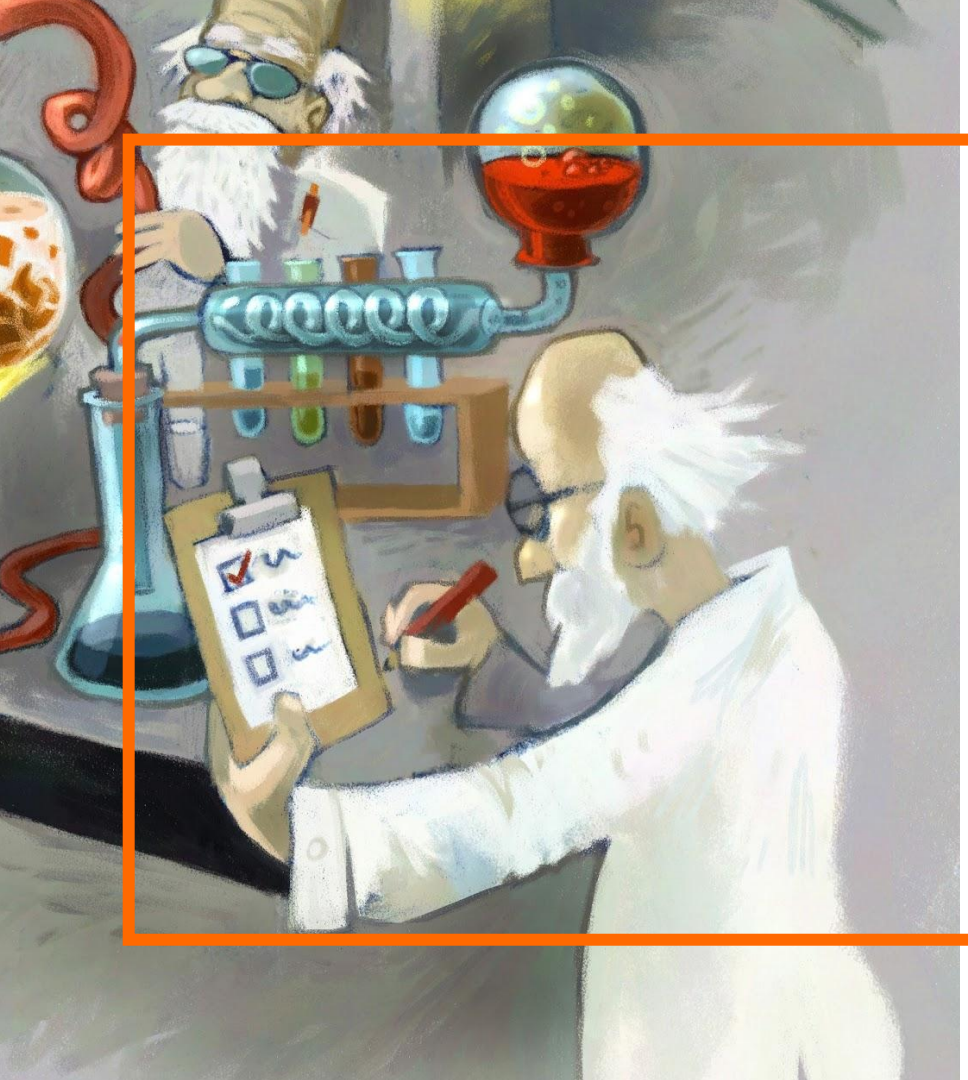
Command

```
def check():
    try:
      d = http("http://" + entity["ip"] + ':8080').json()
    except:
      d = http("http://" + entity["ip"] + ':8080').json()

    r = {}
    r["load1"] = d["system_stats"]["load_average"][0]
    r["load5"] = d["system_stats"]["load_average"][1]
    r["connections"] = {
        "total": d["postgresql"]["connections"]["total"],
        "active": d["postgresql"]["connections"]["active"],
    }
    if 'cgroup' in d:
        r["memory"] = d["cgroup"]["memory"]
    else:
        r["memory"] = d["system_stats"]["memory"]["cgroup"]
    r["cpu"] = d["system_stats"]["cpu"]

    return r
```

Interval 1m



zalando



POSTGRESQL INTERNAL METRICS

How to collect internal metrics

- ZMON worker is running in the same K8s cluster as the PostgreSQL nodes;
- It supports running SQL queries natively;
- We need only to figure out the proper tables/views to select from.

What are the metrics we need?

- Server
  - idle transactions — `pg_stat_activity`
  - failed login attempts
- Tables — `pg_stat_user_tables`
  - size / seq_scans / inserts / updates / deletes
- Indexes
  - size / scans — `pg_stat_user_indexes`
- Backups
  - WAL archiver status — `pg_stat_archiver`
  - age of last backup — ***S3 bucket check***

Idle transactions

Description

List the pids of the transactions which where idle since 15 or more minutes.

Command

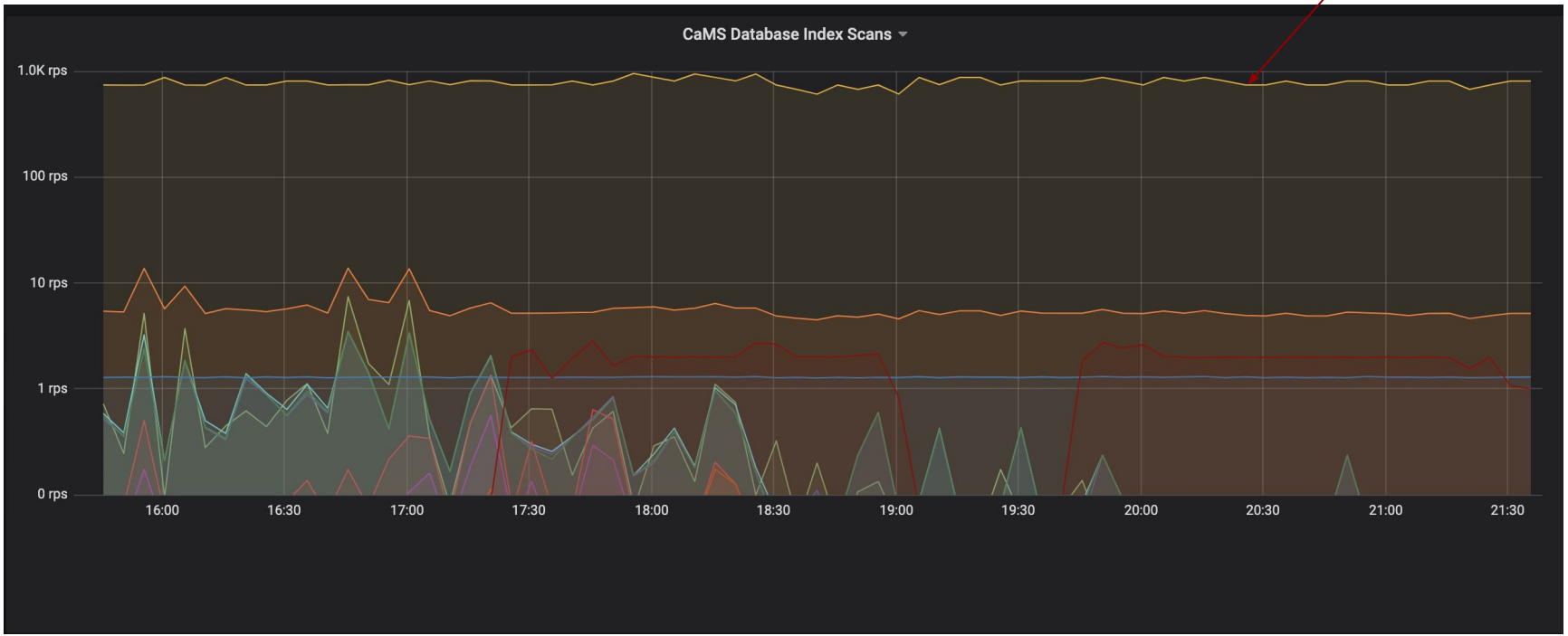
```
def check():
    query = """
              SELECT *
              FROM   pg_stat_activity
              WHERE state in ('idle in transaction', 'idle in transaction (aborted)')
              AND current_timestamp - state_change >= INTERVAL '15' MINUTE
              AND  pid <> pg_backend_pid();
            """
    d1 = {}
    for row in sql(shard='postgres').execute(query).results():
        d1[row['pid']] = {}
        d1[row['pid']]['datname'] = row['datname']
        d1[row['pid']]['usename'] = row['usename']
        d1[row['pid']]['state_change'] = row['state_change']
        d1[row['pid']]['application_name'] = row['application_name']
        d1[row['pid']]['state'] = row['state']
    return d1
```

Interval 15m

Entities environment = production **AND** type = postgresql_cluster

Index scans

It looks like a problem!



zalando

But what's about authentication?

- ZMON worker uses its own credentials to connect to all the databases in the cluster. The credentials are *separate* from the application credentials;
- ZMON worker user is unique in every K8s cluster: **`robot_cluster_name`**;
- PostgreSQL role **`robot_zmon`** is created during the database setup to authorize the ZMON user to access the database objects;
- ZMON may also be authorized to run queries against the application tables or views, but the permissions for the ZMON role should be granted explicitly in DB:

```
GRANT SELECT ON ALL TABLES

IN SCHEMA public TO robot_zmon;
```

Application metrics

Description

Retrieves the list of campaignss in error

Command

```
def check():
    result = {}
    dbResults = sql().execute("""
        SELECT cmp.id, cmp.source_group_id, cmp.destination_id, cmp.runtime_start, cmp.runtime_end,
cmp.country, cmp.gender, cmp.platform_type, cmp.campaign_type, io.name as io_name, io.destination_id
as io_destination_id, io.source_id as io_source_id, io.source as io_source
        FROM campaigns.campaign_registry cmp, campaigns.insertion_order_registry io
            WHERE cmp.insertion_order_id=io.id
        AND cmp.is_exported = false
        AND cmp.is_error = true
        AND cmp.runtime_end > now();
    """).results(max_results=10000)

    result['total_count'] = len(dbResults)
    result['items'] = dbResults

    return result
```

Interval 2m

Key Takeaways

- Get useful metrics from your infrastructure (K8s, AWS, …)
- Collect PostgreSQL metrics from the `pg_stat_…` views
- Get your application metrics by querying your tables or views directly (but beware of the performance impact)
- Treat your monitoring tool as another application, not as the superuser

Feel free to reach me:

Uri Savelchev
<uri.savelchev@zalando.fi>

Find out more about our Culture, People & Jobs:

- **ON SOCIAL MEDIA:**

  Linkedin **@Zalando SE**
  Facebook **@ Inside Zalando**
  Instagram **@insidezalando**
  Twitter **@ZalandoTech**

- **CAREER WEBSITE:** jobs.zalando.com
- **CORPORATE WEBSITE:** corporate.zalando.com

zalando