Patroni

A real-life experience and migration path

Jan Tomsa

About me

- Jan Tomsa
- DevOps at Showmax
- jan.tomsa@showmax.com

What is showmax?

- VOD (Video On Demand) service
  - Microservice architecture (we love containers)
  - EU (datacenter) + Africa (customers)
  - Engineering in Prague & Beroun (CZ)
  - We love PostgreSQL :)
    - And other open-source technologies!

https://kinkyrhino.co.za/product/showmax-and-chill

Before the migration

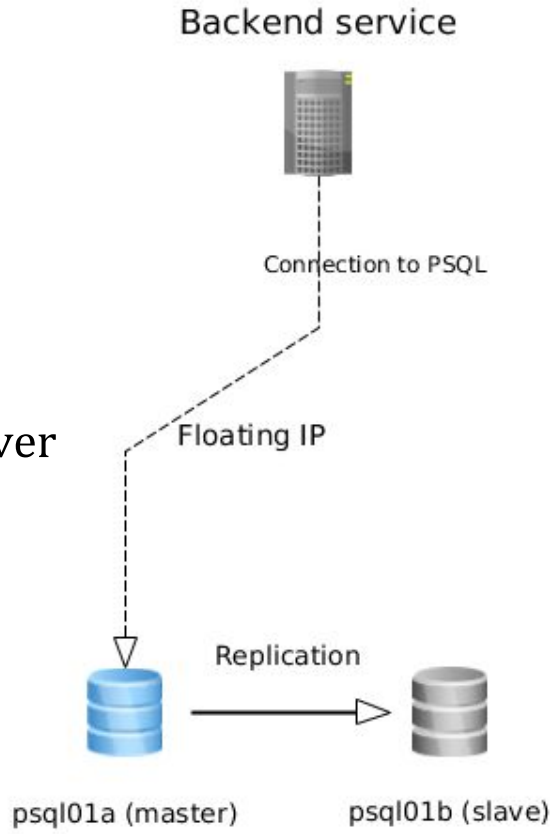
- 7 production (+1 stage) application PSQL clusters
  - ~600 GB
- 1 data warehouse
  - 2 TB
- 1 cluster for the analytics guys
  - 1 TB
- 1 support cluster for the infrastructure

Original topology

- Master -> slave replication (repmgr)
- “Floating IP”
  - Pointing to the current master
  - Not a DNS failover - an extra IP routable to any server

\+ It’s just a worker

\- Manual failover
  - Master <-> Slave
  - IP (using API call - datacenter management)



Unsustainable

- Slow failover
- Long downtime
- On-caller’s nightmare
- Only one replica

http://2.bp.blogspot.com/-nHW-TIx6TKk/U9Kw8Fi_tqI/AAAAAAAALFg/gfJGhbb1bQE/s1600/unbearable.jpg

We need an automated solution

- Automated solution
- Virtually no need for human interference
- Quick and easy (manual) failover

https://dynamicmedia.zuza.com/zz/m/original_/8/a/8a64a122-8ec3-43e3-9f51-93054360bcff/IYN4_SS__Super_Portrait.jpg

Candidates

- Two candidates
  - Similar properties and architecture
- Patroni (https://github.com/zalando/patroni)
  - python, HAProxy
- Stolon (https://github.com/sorintlab/stolon)
  - go, homemade proxy

Patroni

- Maintained by Zalando
- “Template for HA PostgreSQL”
  - Very adjustable
- Python “wrapper” around PostgreSQL
- Handles the lifecycle of PostgreSQL

Where to get it?

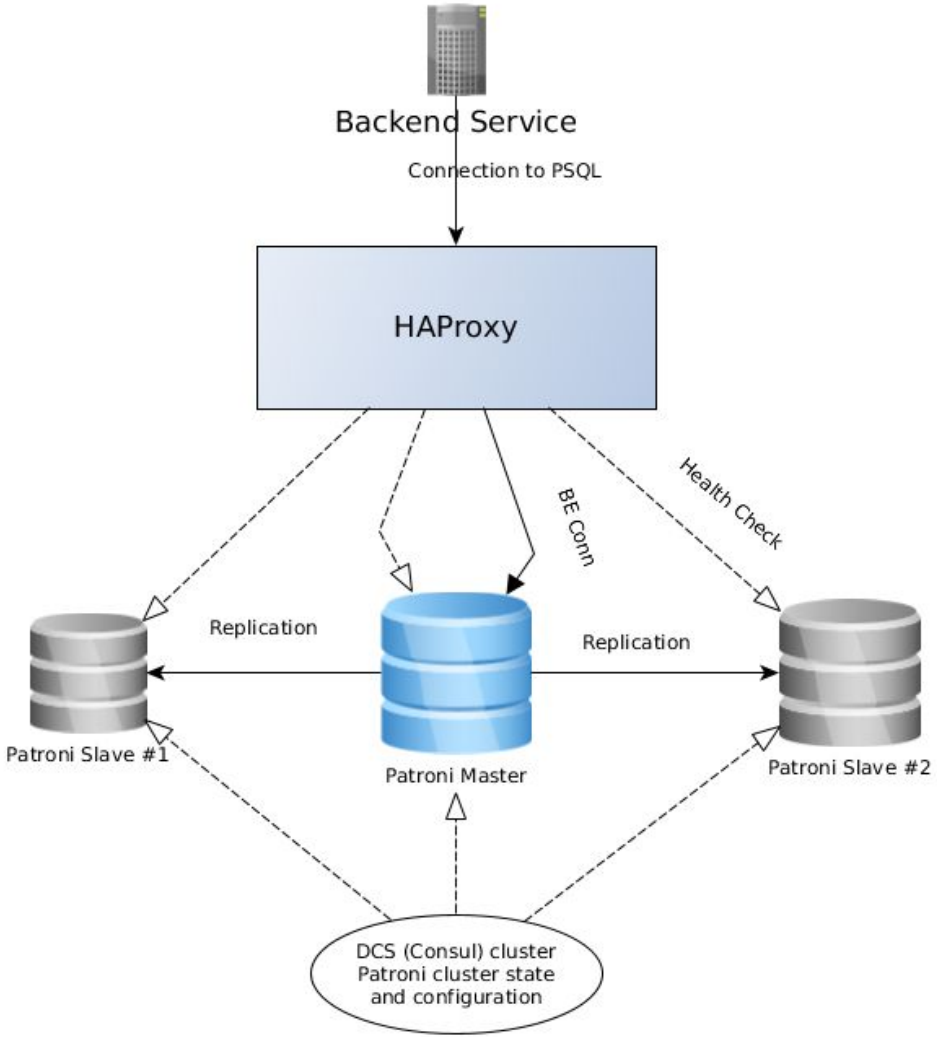
- Source code
  - GitHub - https://github.com/zalando/patroni
- Docs
  - https://patroni.readthedocs.io/en/latest/
- Prebuilt packages
  - PIP - https://pypi.org/project/patroni/
  - DEB - PostgreSQL repo https://www.postgresql.org/download/

First steps - Dockerized cluster for testing

- Clone the repo - https://github.com/zalando/patroni
  - Contains Dockerfile and Docker compose file
- docker-compose up
- Starts
  - ETCD
  - 3x Patroni with PostgreSQL
  - HAProxy

https://img.csfd.cz/files/images/film/photos/000/081/81627_08378a.jpg

Topology


Backend Service
Connection to PSQL
HAProxy
BE Conn
Health Check
Replication
Replication
Patroni Slave #1
Patroni Master
Patroni Slave #2
DCS (Consul) cluster
Patroni cluster state
and configuration

Tools - REST API

- Getting current state
  - Used by HAProxy for routing
    - API returns HTTP code 200 or 503

```
curl -v -XOPTIONS localhost:8008/master
curl -v -XOPTIONS localhost:8008/replica
```

- Configuration
  - Getting and setting a dynamic configuration

Tools - PATRONICTL

- Command line cluster management tool
- failover/switchover
- Replica reinitialization
- Cluster state monitoring (pretty, json, yaml, …)

```
 root@patroni01a:~# patronictl -c /etc/patroni/patroni.yml list
+---------------+------------+--------------------------+--------+---------+-----------+-----------------+
|    Cluster    |   Member   |           Host           |  Role  |  State  | Lag in MB | Pending restart |
+---------------+------------+--------------------------+--------+---------+-----------+-----------------+
| patroni01-p11 | patroni01a | patroni01a               | Leader | running |       0.0 |        *        |
| patroni01-p11 | patroni01b | patroni01b               |        | running |       0.0 |        *        |
| patroni01-p11 | patroni01c | patroni01c               |        | running |       0.0 |        *        |
+---------------+------------+--------------------------+--------+---------+-----------+-----------------+
```

Configuration approaches

Three main types

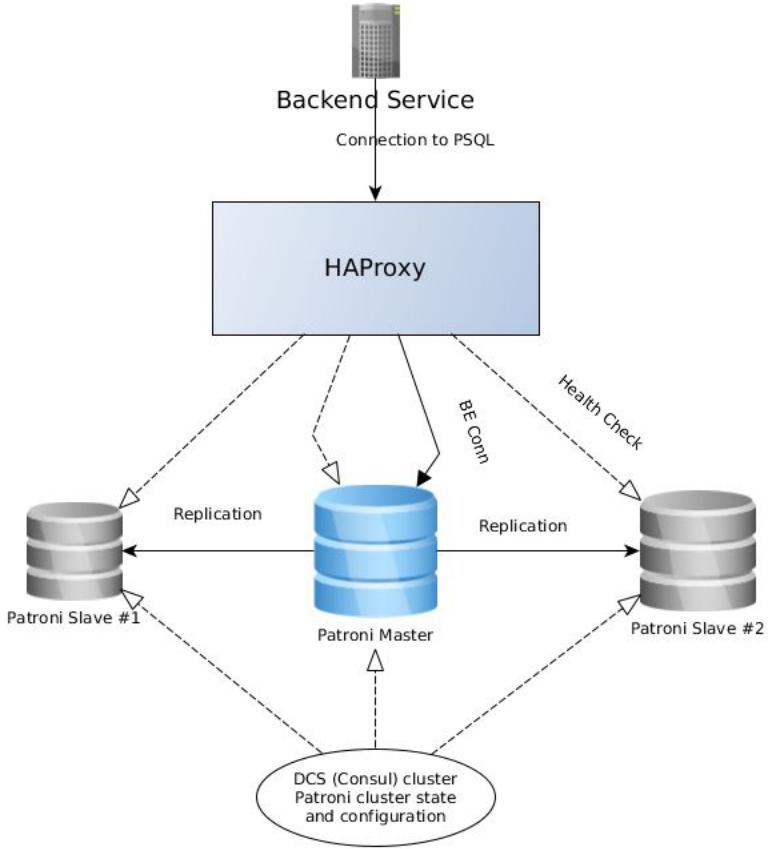
- Dynamic configuration
  - Stored within the DCS (consul, ETCD, …)
- Local configuration
- Environment variable configuration

Configuration options

- Patroni.yml
  - The main config file
  - Setting up both dynamic and local config
  - Cluster parameter setup (timeouts, …)
  - Bootstrap (initdb parameters, …)
  - PostgreSQL
    - HBA, accesses (superuser, …), PostgreSQL parameters

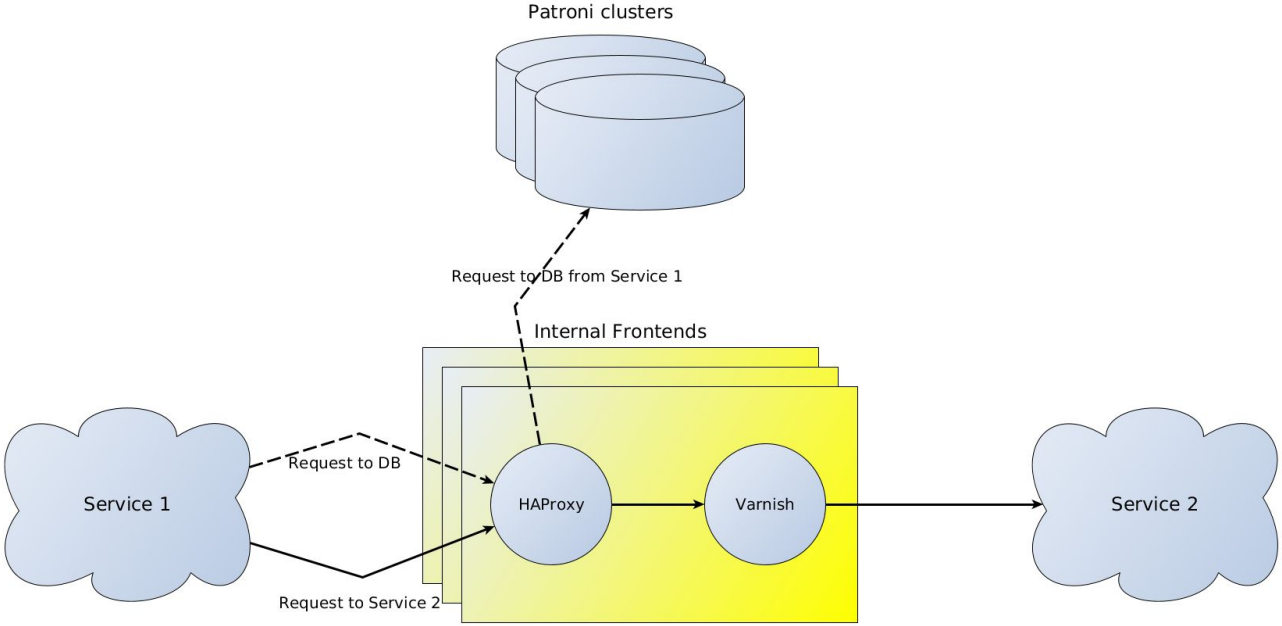
Patroni @ Showmax

- 6 production clusters
  out of 9 total
- Cluster: 3 dedicated servers
- DCS: consul (5 node cluster)
- Proxy: HAProxy
- Backups: PgDump + Barman
- Monitoring: Icinga + Prometheus



Patroni @ Showmax - HAProxy

- Microservices use HAProxy and Varnish as a communication mediator



Patroni @ Showmax - HAProxy

```
frontend patroni01_master
    mode tcp
    timeout client          40m
    bind <bind_ip>:5000
    default_backend bk_patroni01_master

backend bk_patroni01_master
    mode tcp
    timeout server          40m
    timeout connect         500
    option httpchk OPTIONS /master
    server <hostname1> <ip1>:5432 maxconn 500 check port 8008 inter 1000
    server <hostname2> <ip2>:5432 maxconn 500 check port 8008 inter 1000
    server <hostname3> <ip3>:5432 maxconn 500 check port 8008 inter 1000
```

Patroni @ Showmax - Backup - pgdump

- Regularly; every night
- Pgdump connects to the DB via HAProxy

https://commons.wikimedia.org/wiki/File:Dump_Truck_Dumping_Toxic_Medical_Waste.png

Patroni @ Showmax - Backup - Barman

- Barman https://www.pgbarman.org/
  - “administration tool for disaster recovery of PostgreSQL”
  - WAL streaming (archive_command)
  - Regularly doing base_backup
  - The possibility of PIT recovery
- Patroni adaptation:
  1. Barman configuration for all nodes in the cluster
  2. Go wrapper triggering the Barman backup against the current Patroni master

Patroni @ Showmax - monitoring

- Icinga
  - Active checks - db connection test
  - HAProxy backends state monitoring
- Prometheus
  - Clusters stats ingestion and alerting
    - Master switching
    - Replica lags
    - Current connection count

Patroni troubles @ Showmax
Or how to shoot yourself in the foot

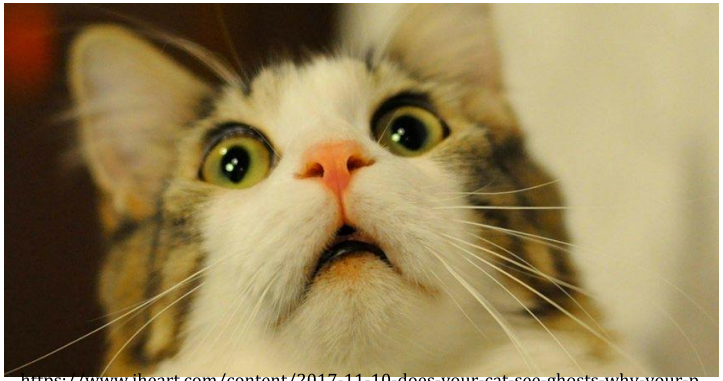
https://geekandsundry.com/the-wait-is-over-discover-your-patronus-with-pottermores-new-quiz.jpg

https://pixabay.com/en/patrons-bullets-sleeves-ammo-1493479.png

The amplification disaster

- After a migration to different HW
- Night backup - the master stopped responding
  - Patroni worked! A replica was promoted
- A few nights later…
  - 2 of 3 nodes died
- Investigation
  - It was kernel panic
  - -> Kernel upgrade
  - -> Catching traces using netconsole

The amplification disaster

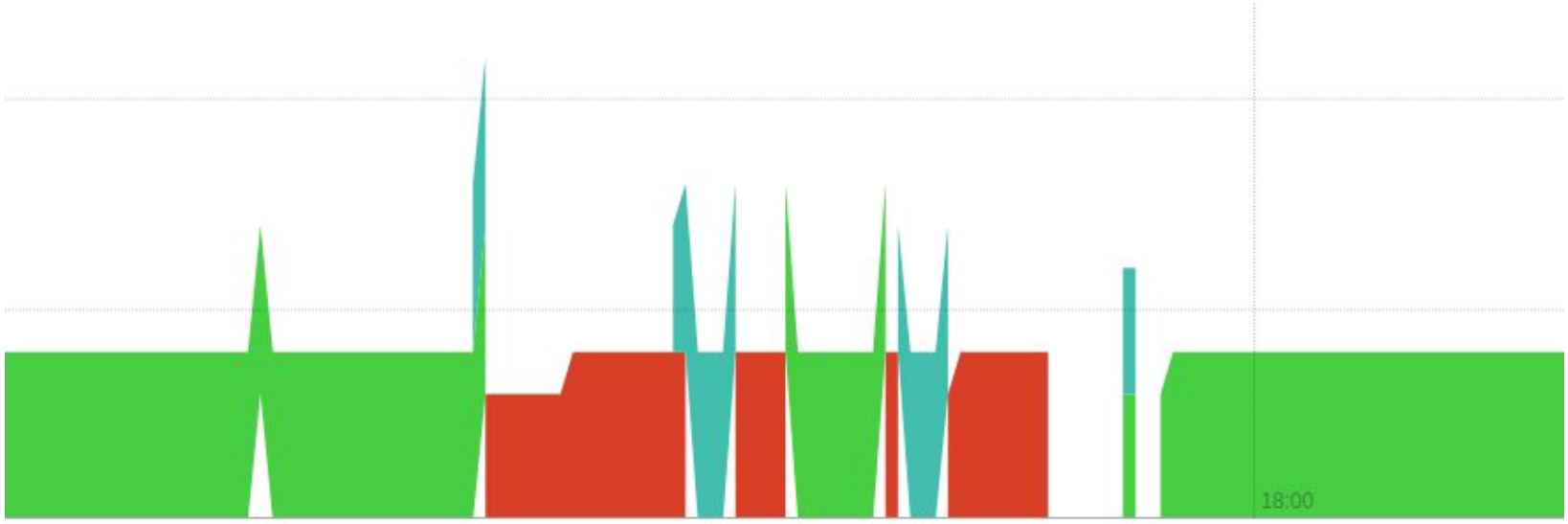


https://www.iheart.com/content/2017-11-10-does-your-cat-see-ghosts-why-your-pet-stares-into-space/.jpeg

- Early evening peak
  - Servers disappearing from patronictl
  - Sometimes only one server alive
  - Patroni complains it cannot communicate with DCS (consul)
- Very scary situation
  - We were pretty busy keeping at least one server alive
  - Thankfully we were able to reinitialize at least some slaves before the current master died
  - User requests were failing - services were constantly reconnecting to different masters

The amplification disaster

- Patroni master switching visualization



What actually happened?

- Poor quality network card (Intel i219LM, driver e1000)
  - Unstable under load with offloading turned on
  - Causing NIC restarts - “Reset adapter unexpectedly” in dmesg
  - Sometimes even Kernel Panic
- From the outside it looks like a mild packet loss

What actually happened?

Network restarts caused:

- Local Consul agent disconnection (from the main cluster)
  - Patroni could not connect to the DCS and get cluster state
  - It caused frequent master changes

The ultimate reason for the problems was that Patroni actually worked exactly as expected! But, the rapid switching taken together with machines dying caused additional problems

The fix

- Temporarily turn on maintenance mode
  - Patroni stops managing postgres
  - Postgres is able to handle the network outages
- Turn off offloading
  - Tcp-segmentation-offload
  - Generic-segmentation-offload
  - generic-receive-offload
  - ethtool -K eth0 gso off gro off tso off
  - It immediately stabilized the situation

It broke an index!

- Hung SELECT queries
  that could not be killed (without -9)
- Broken data or broken index?

https://pngtree.com/freepng/the-owl-staring-in-front_3324955.html.jpg

```
user_lists=# reindex database user_lists;
ERROR:  could not create unique index
"index_user_list_items_on_user_list_id_and_asset_id"
DETAIL:  Key (user_list_id, asset_id)=(ab22fcff-0d5f-476b-902c-3c29d3221228,
cfd2b50e-152d-2278-88db-a72b4b1bf22d) is duplicated.
```

- Duplicate records despite unique index
  - Manual delete and reindex eventually helped

Notes & advice

- When using Consul
  - Let Patroni connect to the main Consul cluster directly
  - Our consul setup was a bit unfortunate during this outage
    - Main cluster
    - Local client on the host
    - Patroni connecting to the local client
- Verify the quality and speed of you network card

Notes & advice

- Maintenance mode is your friend
  - `patronictl pause` / `patronictl resume`
  - Patroni stops managing Postgres
  - Possibility to make manual postgres interventions and/or upgrade patroni, etc
  - It helped us stabilize the situation

Notes & advice

- Some PostgreSQL parameters are passed as a command line args to pgctl start
  - For example max_connections or wal_keep_segments
  - The same for the whole cluster
  - patroni.yml - place into to the `bootstrap::dcs::postgresql` section
  - https://patroni.readthedocs.io/en/latest/dynamic_configuration.html

Notes & advice

- Correct HAProxy timeouts
  - Might cut off your connections

Summary

- Patroni is great!
- It just works

https://forum.webflow.com/t/hurray-for-ix-2-0-beta/48446

- Nothing is perfect - even Patroni can cause you trouble
  - But only if the Gods are against you
- Don’t underestimate testing and correct setup. Patroni is quite complex

Come and join us!

We’re looking for new colleagues

tech.showmax.com

Thanks!

Questions?