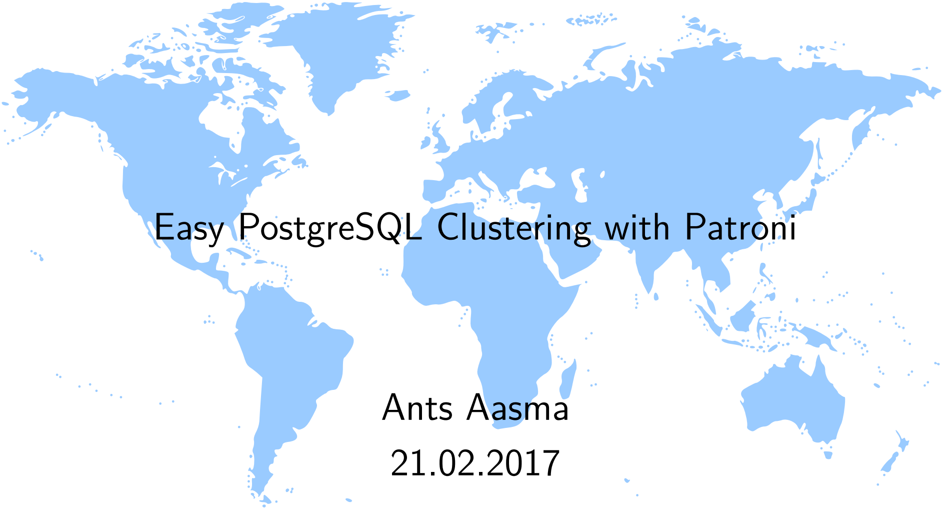


Easy PostgreSQL Clustering with Patroni

Ants Aasma

21.02.2017

Introduction

About me

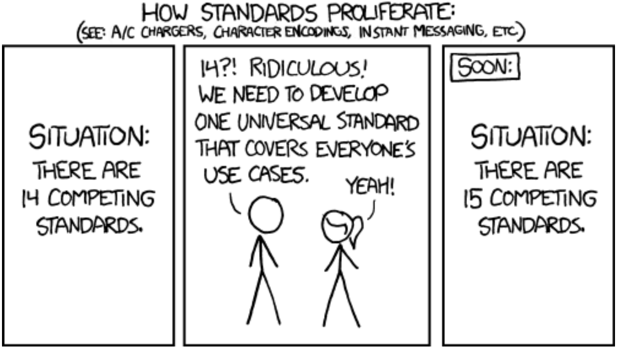
- Support engineer at Cybertec
- Helping others run PostgreSQL for 5 years.
- Helping myself run PostgreSQL since 7.4 days.

What are we going to talk about

- Patroni - a tool to build high availability clusters with PostgreSQL
    - https://github.com/zalando/patroni
- Questions welcome during the talk

PostgreSQL clustering state of the art

- pgpool2, Pacemaker, repmgr, . . .


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

What Patroni does

Parts of the HA problem

- Detect failure
- Promote new master
- Route clients to the correct master

Why is HA tricky

- PostgreSQL provides single-master replication
- Having more than one master is worse than having none.
- Need to agree on who gets to be master
  - When servers fail
  - When networks fail
  - When things almost work but not quite

Distributed databases to the rescue

- Distributed consensus algorithms were inveneted to solve this. Paxos, Raft
- Many existing distirbuted consensus databases:
  - etcd
  - Consul
  - Zookeeper

How Patroni solves the HA problem

- Each node runs a Patroni agent
- Patroni agent runs a constant loop to check
- health of local PostgreSQL
- health of cluster
- fix things when they are not ideal
- A distributed consensus store is used to pick a leader

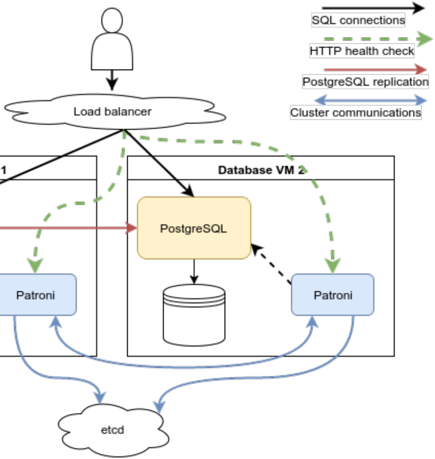
How Patroni works

- Picks one node to initialize the database
- Clones new nodes joining the cluster using `pg_basebackup`
- Monitors health of PostgreSQL
- Promotes a new master if existing master fails
- Sets `primary_conninfo` on other standbys
- Rejoins old master using `pg_rewind`

Load balancer

- Patroni does not do client routing or virtual IP movement
- libpq expects to see a single host to connect to.
  - This will be fixed in PostgreSQL 10
  - JDBC already supports this directly
- Use your favourite load balancer to perform the connection forwarding
  - HAProxy/nginx + VIP failover/run on app servers and connect to localhost
  - F5 BigIP, etc.
  - Your cloud providers load balancer
  - Customize your connection pooler

Patroni architecture



Demo time

Local etcd for testing

```
# Download
curl -sL https://github.com/coreos/etcd/releases/download/\
v3.1.1/etcd-v3.1.1-linux-amd64.tar.gz   | tar xz

# And run
etcd-v3.1.1-linux-amd64/etcd
```

Setting up Patroni

```
# Install Patroni
virtualenv --quiet patroni-venv && source patroni-venv/bin/activate
pip install patroni

# Sample config
wget https://github.com/zalando/patroni/raw/master/postgres0.yml
vim postgres0.yml

# Ready to go
patroni postgres0.yml
```

Second node

```
# Change node name, datadir name and ports
cat postgres0.yml |\
    sed s/postgresql0/postgresql1/ |\
    sed s/:5432/:5433/ |\
    sed s/:8008/:8009/ > postgres1.yml

patroni postgres1.yml
```

Setting up a load balancer

```
# Get the system HAProxy package installed
sudo apt-get/yum/... install haproxy

# Get confd
curl -O https://github.com/kelseyhightower/confd/releases/download/\
v0.11.0/confd-0.11.0-linux-amd64
chmod a+x confd-0.11.0-linux-amd64 && ln -s confd-0.11.0-linux-amd64 confd

# Get Patroni confd config examples
curl -sL https://github.com/zalando/patroni/archive/v1.2.3.tar.gz | tar xz
# Adjust HAProxy to run locally
sed -i 's#/etc/haproxy'         patroni-1.2.3/extras/confd/conf.d/haproxy.toml
sed -i 's#/var/run/#haproxy/#' patroni-1.2.3/extras/confd/conf.d/haproxy.toml
```

Setting up a load balancer continued

```
# Run confd
./confd -prefix=/service/batman -backend etcd \
   -node http://localhost:2379/ \
   -interval 10 \
   -confdir patroni-1.2.3/extras/confd/
```

Wrapping up

How we avoid split brain

- All nodes try to acquire leader key in DCS.
- Leader key has a timeout, the master runs a loop that keeps updating the leader key.
  - If DCS gives an error - PostgreSQL gets demoted
  - If DCS access times out - PostgreSQL gets demoted
  - If we discover leader key was timed out - PostgreSQL gets demoted
- When there is no leader other nodes try to contact the previous leader.
- If Patroni is not responding, load balancer removes that node from rotation.
- Future versions will have kernel watchdog support
  - If Patroni does not get to run, the whole OS gets rebooted.

Leader elections

- If there is no leader key in the cluster, the remaining nodes
  - Check if the old leader is still responding
  - Contact all other members, the ones with most xlog get to participate in the leader race
  - Check if they are too far behind from last known master xlog position.

That’s all

Thank you

- Questions?
- If you need professional support, contact us info@cybertec.at

Extra content

Configuration

- Configuration is merged from cluster configuration stored in etcd and local configuration given as a parameter.
- Patroni does configuration management for PostgreSQL
- You can update Patroni configuration through the REST API

```
curl -XPATCH \
    -d '{"postgresql": {"parameters": {"work_mem": "32MB"}}}' \
    http://localhost:8008/config
```

- Patroni updates PostgreSQL configuration on all nodes and issues SIGHUP

Other nifty features

- Schedule a node restart in the future
  - Optionally, only if there are config changes that require a restart
  - Optionally, only if still running a specified version
- Voluntary restart runs a checkpoint before shutdown
- Run a manual failover
  - Schedule a failover in the future

Nifty features, continued.

- Clone new nodes from a special node instead of master.
- Clone new nodes using a backup
- Turn off cluster management to do whatever.