Expecto Patroni – Protecting Postgres against outages

Michael Mühlbeyer
17 September 2021

trivadis
Part of Accenture

MICHAEL MÜHLBEYER

- since 10/2012 @Trivadis Stuttgart
- Postgres & Oracle Databases
- BigData & Streaming Architecture

- music, reading, beekeeping

3 AGENDA

- Introduction
- Patroni
- Installation & Configuration
- Use Cases

4 INTRODUCTION

- All changes in a Postgres Cluster are written in a transaction log, the so called Write-Ahead-Logs (WAL)
- Get to a consistent state with table files and the WAL files
- It’ possible to send WAL data to a remote server

INTRO
STREAMING REPLICATION

- Available since Postgres 9.0
  - Sync since Postgres 9.1
- WAL is being sent to a standby server (in addtion to local file system)
- Standby server applies the received data

INTRO
STREAMING REPLICATION

- Single-master/Multi-slave setup
- Only one primary server (read-write)
- Multiple standby servers

INTRO
STREAMING REPLICATION

- Backups from standby possible
- Promote standby if primary crashs

INTRO
STREAMING REPLICATION

- No automatic failover possible
- Split-brain in multi-standby environments
- Distributed consensus needed

PATRONI

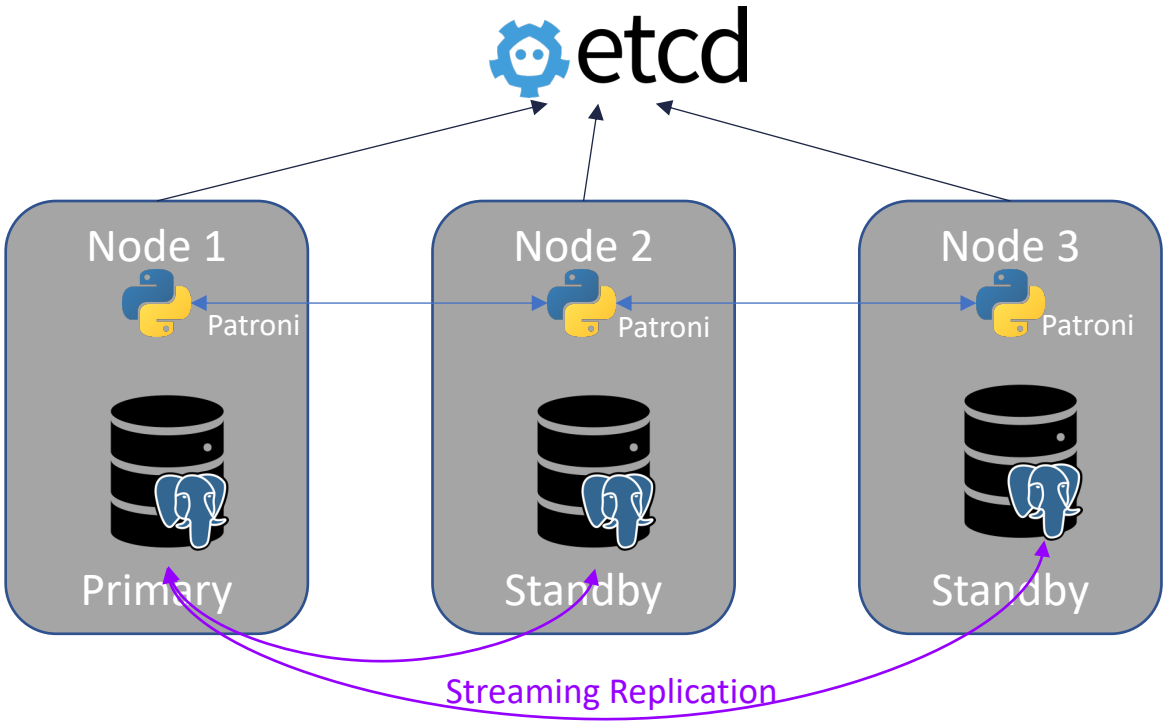
PATRONI

- A template to create your own High Availability solution
- Python based
- Uses a distributed configuration store(DCS)
  - etcd, Zookeeper, Consul

PATRONI

- Developed by Zalando
- OpenSource (MIT license)
- Originally forked from governor

12 PATRONI


etcd
Node 1
Patroni
Primary
Node 2
Patroni
Standby
Node 3
Patroni
Standby
Streaming Replication

ETCD

- A consistent distributed key-value store
- Raft algorithm
  - Leader-based
- Great for config & metadata

ETCD ALTERNATIVES

- Zookeeper
- Consul
- Exhibitor

PATRONI

- Monitors and manages the Postgres process
- Current state is written to DCS
- Promotes or degrades standby or primary

PATRONI

- Adapts postgresql.conf
- Config done via YAML file
- Config per patroni instance
- Can be used with an existing cluster

INSTALL & CONFIG

PATRONI INSTALLATION

- Standard setup
  - 2-3 Postgres nodes
  - 3 etcd nodes
  - 1-2 HA Proxy nodes (optional)

ETCD (DCS) PART

- Quorum of 3 servers needed
- Setup etcd according to your Linux distro
- Adapt configuration
- Create a systemd service

20 ETCD CONFIG

- Create etcd config file

```
$ cat /etc/etcd.conf
ETCD_LISTEN_PEER_URLS="http://192.168.36.131:2380,http://localhost:2380"
ETCD_LISTEN_CLIENT_URLS="http:////192.168.36.131:2379,http://localhost:2379"
[Clustering]
ETCD_INITIAL_ADVERTISE_PEER_URLS="http:////192.168.36.131:2380"
ETCD_ADVERTISE_CLIENT_URLS="http:////192.168.36.131:2379"
ETCD_INITIAL_CLUSTER="default=http:////192.168.36.131:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER_STATE="new"
```

21 ETCD CONFIG

- Create etcd data dir

```
$ mkdir -p /var/lib/etcd/
$ sudo useradd -s /sbin/nologin --system -g etcd etcd
$ sudo groupadd --system etcd
$ sudo chown -R etcd:etcd /var/lib/etcd/
```

22 ETCD CONFIG

- Create etcd system file dir

```
$ cat /etc/systemd/system/etcd3.service


[Unit]
Description=etcd key-value store
Documentation=https://github.com/etcd-io/etcd
After=network-online.target local-fs.target remote-fs.target time-sync.target
Wants=network-online.target local-fs.target remote-fs.target time-sync.target
```

ETCD CONFIG

- Create etcd system file dir (contd)

```
[Service]
User=etcd
Type=notify
Environment=ETCD_DATA_DIR=/var/lib/etcd
Environment=ETCD_NAME=%m
Environment=ETCD_ENABLE_V2
ExecStart=/usr/local/bin/etcd
Restart=always
RestartSec=10s
LimitNOFILE=40000
[Install]
WantedBy=multi-user.target
```

24 ETCD CONFIGS

- Start etcd

```
$ sudo systemctl daemon-reload
$ systemctl start etcd3
```

PATRONI CONFIGURATION

- Install Patroni binary
- Included in the Postgres Common Repos
- Use pip or paket manager for easy installation of dependencies

```
$ dnf install patroni
$ pip3 install patroni [etcd3]
```

PATRONI CONFIGURATION

- Default config file located at /etc/patroni/patroni.yml
- Adapt config location if needed
- Adapt config file to your needs

27 PATRONI CONFIGURATION

```
scope: pgcluster
name: pg1
restapi:
listen: 127.0.0.1:8008
connect_address: 127.0.0.1:8008
etcd:
host: 127.0.0.1:2379
dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
    postgresql:
      use_pg_rewind: true
  initdb:
  - encoding: UTF8
  - data-checksums
  pg_hba:
  - host replication replicator
127.0.0.1/32 md5
  - host all all 0.0.0.0/0 md5
```

```
postgresql:
  listen: 127.0.0.1:5432
  connect_address: 127.0.0.1:5432
  data_dir: data/postgresql0
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: replicator
      password: welcome1
    superuser:
      username: postgres
      password: welcome1
    rewind:
      username: rewind_user
      password: wecome1
```

28 PATRONI COMMANDLINE

```
$ patronictl -c patroni_0.yml  list
+--------+---------------------+---------+---------+----+-----------+
| Member | Host                | Role    | State   | TL | Lag in MB |
+ Cluster: patroni_cluster_1 (7001807246973739139) +----+-----------+
| pg_1   | 192.168.36.131:5432 | Leader  | running |  5 |           |
| pg_2   | 192.168.36.131:5433 | Replica | running |  4 |         0 |
+--------+---------------------+---------+---------+----+-----------+

$ patronictl -c patroni_0.yml list
+--------+---------------------+--------------+---------+----+-----------+
| Member | Host                | Role         | State   | TL | Lag in MB |
+ Cluster: mm (7002579841390271362) ----------+---------+----+-----------+
| pg1    | 192.168.36.131:5432 | Sync Standby | running |  7 |         0 |
| pg2    | 192.168.36.131:5433 | Leader       | running |  7 |           |
+--------+---------------------+--------------+---------+----+-----------+
```

29 PATRONI SWITCHOVER

```
$ patronictl -c patroni_0.yml switchover
Master [pg1]:
Candidate ['pg2'] []:
When should the switchover take place (e.g. 2021-09-14T11:34 )  [now]:
Current cluster topology
+--------+---------------------+--------------+---------+----+-----------+
| Member | Host                | Role         | State   | TL | Lag in MB |
+ Cluster: mm (7002579841390271362) ----------+---------+----+-----------+
| pg1    | 192.168.36.131:5432 | Leader       | running |  6 |           |
| pg2    | 192.168.36.131:5433 | Sync Standby | running |  6 |         0 |
+--------+---------------------+--------------+---------+----+-----------+
Are you sure you want to switchover cluster mm, demoting current master pg1? [y/N]: y
2021-09-14 10:34:09.05604 Successfully switched over to "pg2"
```

30 PATRONI SWITCHOVER

```
+--------+---------------------+---------+---------+----+-----------+
| Member | Host                | Role    | State   | TL | Lag in MB |
+ Cluster: mm (7002579841390271362) -----+---------+----+-----------+
| pg1    | 192.168.36.131:5432 | Replica | stopped |    |   unknown |
| pg2    | 192.168.36.131:5433 | Leader  | running |  6 |           |
+--------+---------------------+---------+---------+----+-----------+

$ patronictl -c patroni_0.yml list
+--------+---------------------+--------------+---------+----+-----------+
| Member | Host                | Role         | State   | TL | Lag in MB |
+ Cluster: mm (7002579841390271362) ----------+---------+----+-----------+
| pg1    | 192.168.36.131:5432 | Sync Standby | running |  7 |         0 |
| pg2    | 192.168.36.131:5433 | Leader       | running |  7 |           |
+--------+---------------------+--------------+---------+----+-----------+
```

HAPROXY

- Add HAProxy to route traffic to the correct nodes
- Read-only traffic to Standby
- Read-write traffic to Primary

HAPROXY
READ WRITE POLICY

```
listen pg_read_write
    bind *:5000
    option httpchk OPTIONS/master
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server postgresql_127.0.0.1_5432 127.0.0.1:5432 maxconn 100 check port 8008
    server postgresql_127.0.0.1_5433 127.0.0.1:5433 maxconn 100 check port 8009
```

HAPROXY
READ ONLY POLICY

```
listen pg_read_only
    bind *:5001
    option httpchk OPTIONS/replica
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server postgresql_127.0.0.1_5432 127.0.0.1:5432 maxconn 100 check port 8008
    server postgresql_127.0.0.1_5433 127.0.0.1:5433 maxconn 100 check port 8009
```

34 HAPROXY

pg_read_write

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN
postgresql_127.0.0.1_5432	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	7s UP
postgresql_127.0.0.1_5433	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	6s DOWN
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	7s UP

pg_read_only

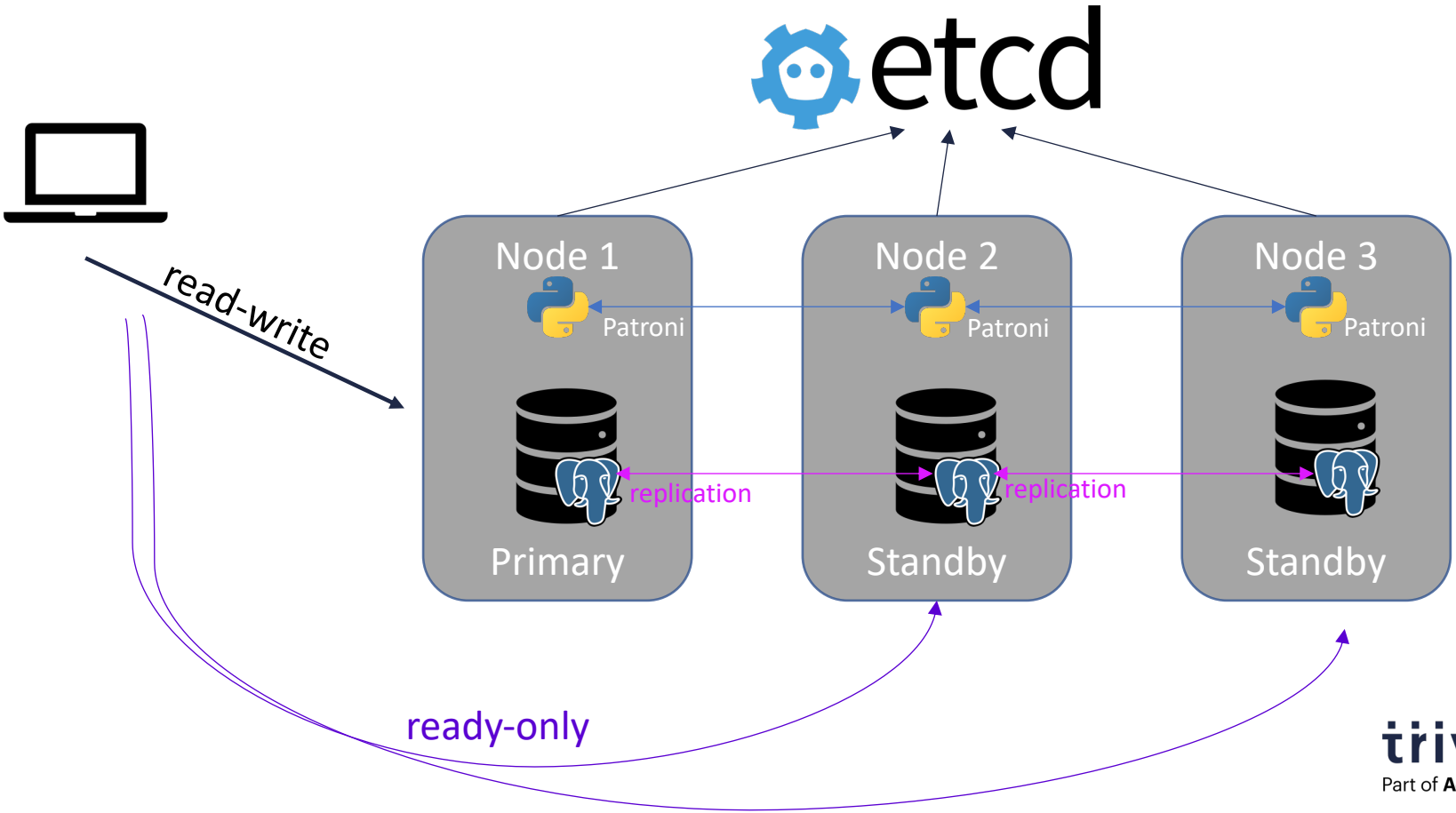
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN
postgresql_127.0.0.1_5432	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	5s DOWN
postgresql_127.0.0.1_5433	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	7s UP
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	7s UP

USE CASES

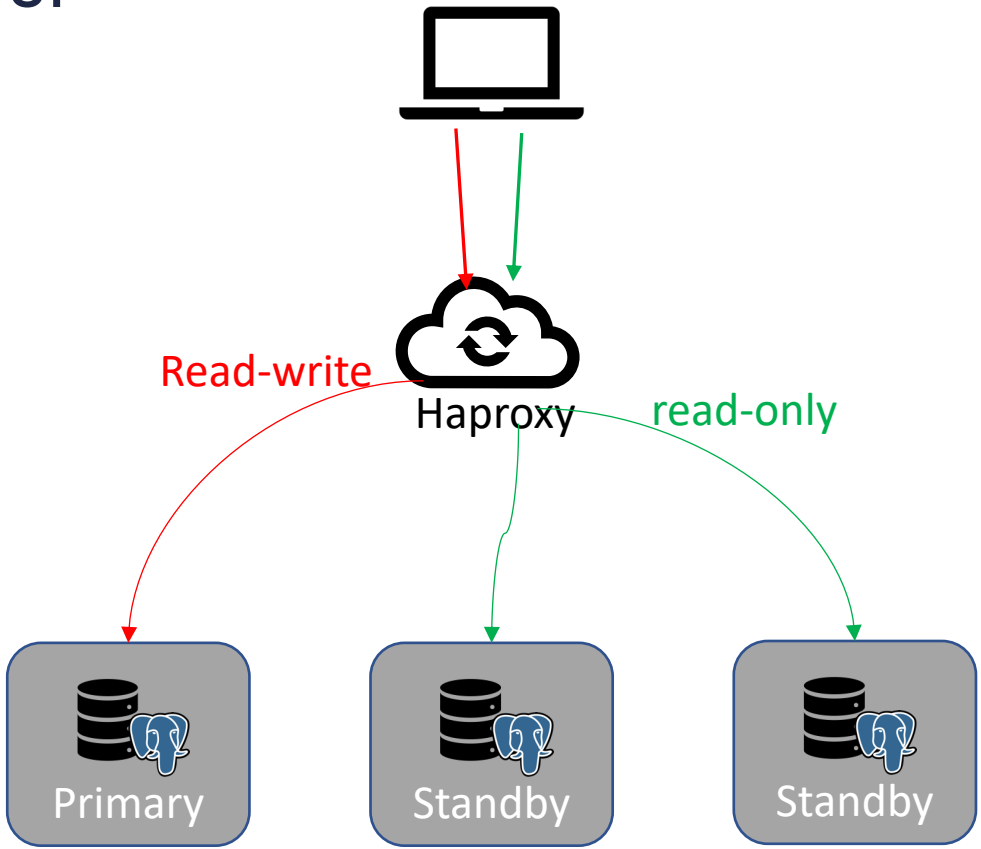
BASIC SETUP

- Patroni with one or more standby servers
- Automatic failover
- Read from all, write to primary
- Default commit settings

37 BASIC SETUP



BASIC SETUP



GUARANTEED COMMIT

- Patroni with one or more standbys
- Guarantee that data is written to all standby servers
- Wait for standby to write data

GUARANTEED COMMIT

- Enable synchronous_mode for (all) standbys
- Client watis until all standbys send ack

- patroni.yml

```
synchronous_mode: true
synchronous_node_count: 2
```

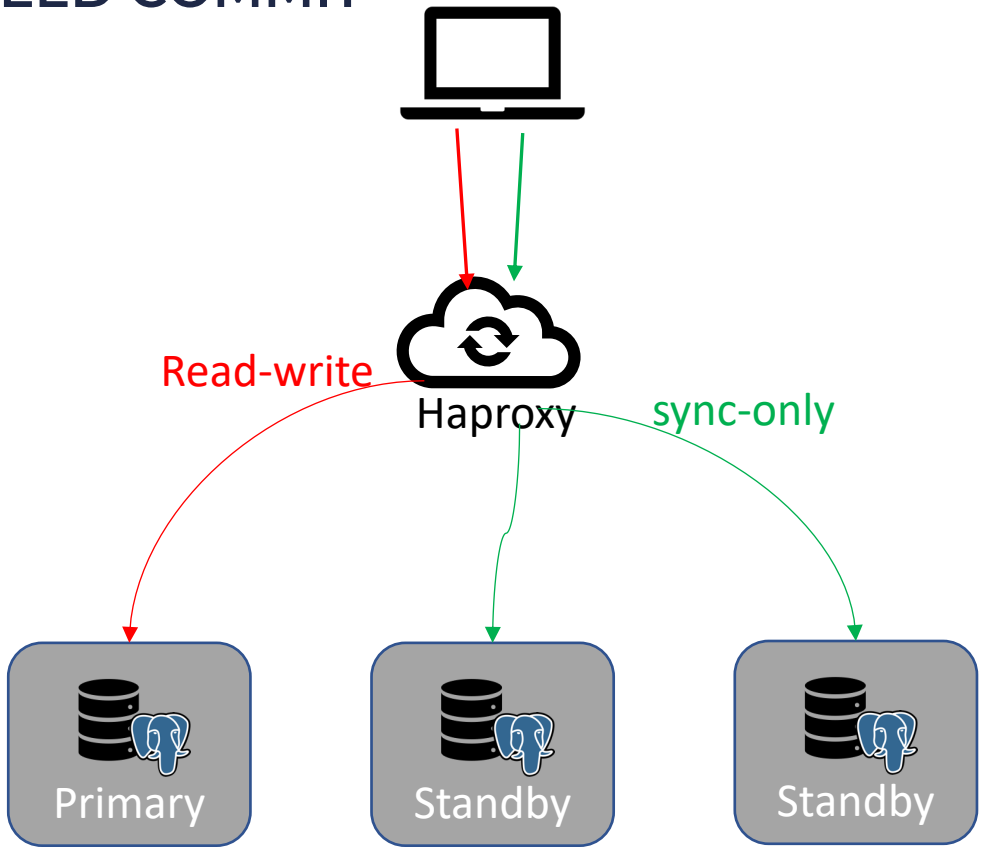
GUARANTEED COMMIT

```
$ patronictl -c patroni_0.yml list
+--------+---------------------+-------------+---------+----+-----------+
| Member | Host                | Role        | State   | TL | Lag in MB |
+ Cluster: mm (7002579841390271362) ----------+---------+----+-----------+
| pg1    | 192.168.36.131:5432 | Leader      | running |  9 |           |
| pg2    | 192.168.36.131:5433 | Sync Standby | running |  9 |         0 |
| pg3    | 192.168.36.131:5434 | Sync Standby | running |  9 |         0 |
+--------+---------------------+-------------+---------+----+-----------+
```

- **postgres.conf**

```
[…]
synchronous_standby_names = '2 (pg2,pg3)'
[…]
```

42 GUARANTEED COMMIT



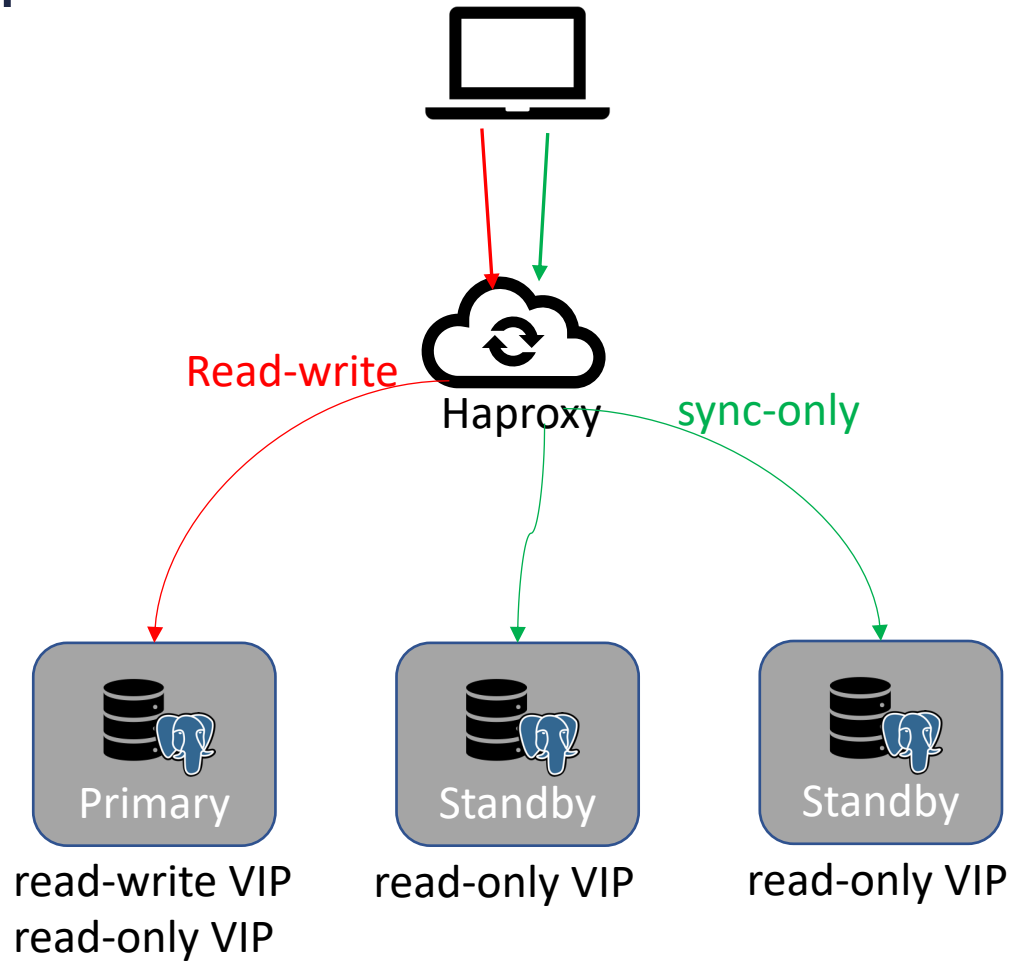
HAPROXY
SYNC ONLY

```
listen pg_sync_only
    bind *:5001
    option httpchk OPTIONS/sync
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server postgresql_127.0.0.1_5432 127.0.0.1:5432 maxconn 100 check port 8008
    server postgresql_127.0.0.1_5433 127.0.0.1:5433 maxconn 100 check port 8009
```

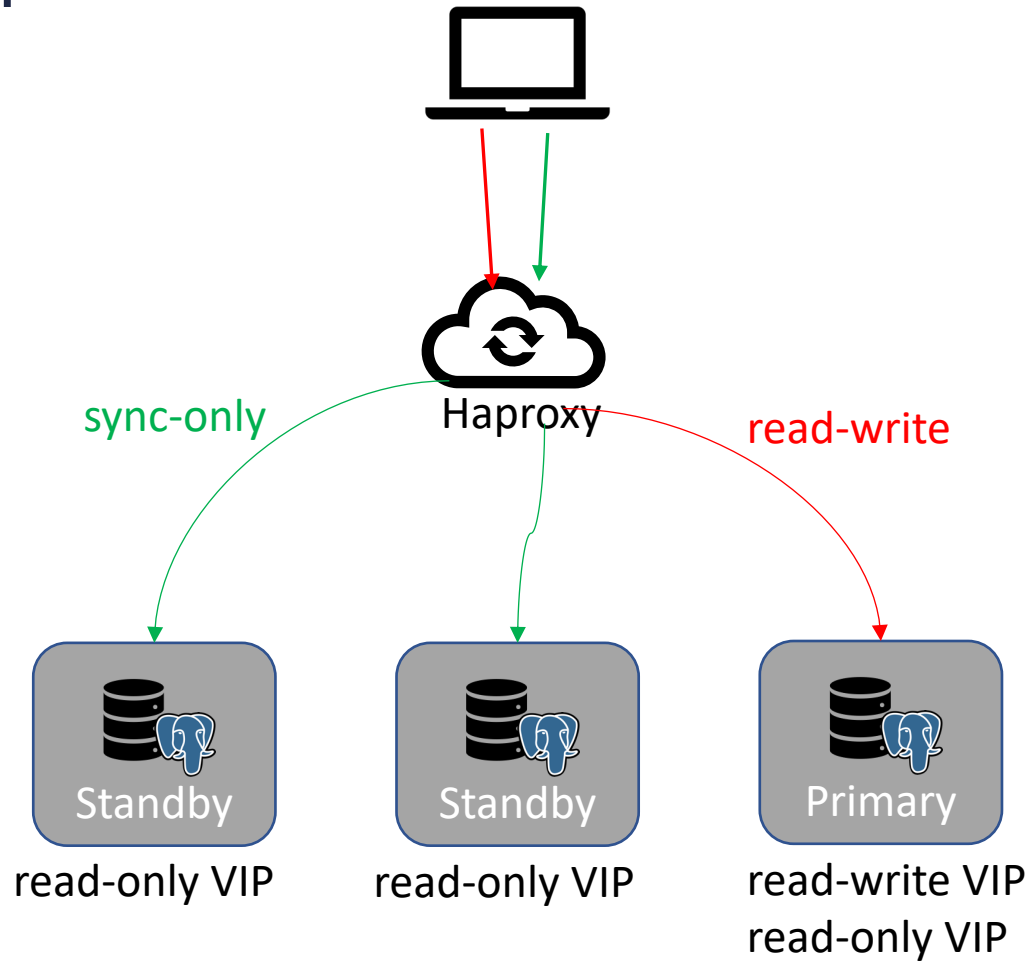
44 VIP SETUP

- Read from all (primary and standby(s))
- Start VIP depending on role
- Use callback script

45 VIP SETUP



46 VIP SETUP



VIP SETUP
CONFIG

```
postgresql:
  listen: 192.168.36.131:5432
  connect_address: 192.168.36.131:5432
  data_dir: /var/lib/pgsql/pg1/data
  bin_dir: /usr/pgsql-13/bin
[…]
  callbacks:
    on_role_change: /var/lib/pgsql/patroni/patroni_callback.sh
[…]
```

48 VIP SETUP
CALLBACK

```
readonly cb_name=$1
readonly role=$2
readonly scope=$3
d=$(date +%Y-%m-%d-%R)

function usage() {
    echo "Usage: $0 <on_start|on_stop|on_role_change> <role> <scope>";
    exit 1;
}
echo "this is patroni callback $cb_name $role $scope"
case $cb_name in
    on_stop)
        echo "stop node at $d" >> /tmp/patroni_status.log
        sudo ip addr del 192.168.36.10/24 dev ens192
        sudo arping -q -A -c 1 -I ens192 192.168.36.10
        ;;

[…]
```

VIP SETUP
CALLBACK

```
[…]
on_role_change)
        if [[ $role == 'master' ]]; then
            echo "becoming master at $d" >> /tmp/patroni_status.log
            sudo ip addr add 192.168.36.10/24 dev ens192
            sudo arping -q -A -c 1 -I ens192 192.168.36.10
        elif [[ $role == 'slave' ]]||[[ $role == 'replica' ]]||[[ $role == 'logical' ]];
then
            echo "becoming slave" at $d >> /tmp/patroni_status.log
            sudo ip addr del 192.168.36.10/24 dev ens192
            sudo arping -q -A -c 1 -I ens192 192.168.36.10
        fi
        ;;
    *)
        usage
        ;;
esac
```

CONCLUSION

51 CONCLUSION

- Patroni is a pretty cool tool
- Easily setup high available Postgres environments
- Possible to solve quite complex environments and demands
  - Can get very complicated

CONCLUSION

- Plan you setup properly
  - All components need to be high available
  - Don’t forget HAProxy and DCS (etcd, consul,...)
  - Test!

FURTHER READING

- https://patroni.readthedocs.io
- https://github.com/zalando/patroni
- https://github.com/zalando/spilo

trivadis