Dans les coulisses d’une infrastructure hautement disponible

Auteur

- Julien Riou
- DBA depuis 2012
- Tech lead dans l’équipe databases à OVH depuis 2015
- pgterminate @ github

Sommaire

- Contexte
- Haute disponibilité
- Sauvegardes et restaurations
- Business intelligence
- Mises à jour
- La suite

OVH

Innovation for Freedom

Produits

Cloud

Serveurs dédiés
VPS
Public cloud
Private cloud
Stockage

Platform

Kubernetes
Logs & Metrics Data Platforms
Databases
Big data
AI & Machine Learning

Web hosting

Noms de domaine
Hébergement web et sites
Solutions E-mail
SSL / CDN
Office & Solutions Microsoft

Télécom

Offres Internet
Téléphonie
SMS / Fax
Bureau virtuel
OverTheBox

Périmètre

Bases internes

60	Clusters
3000	Applications
700	Utilisateurs
400	Bases de données

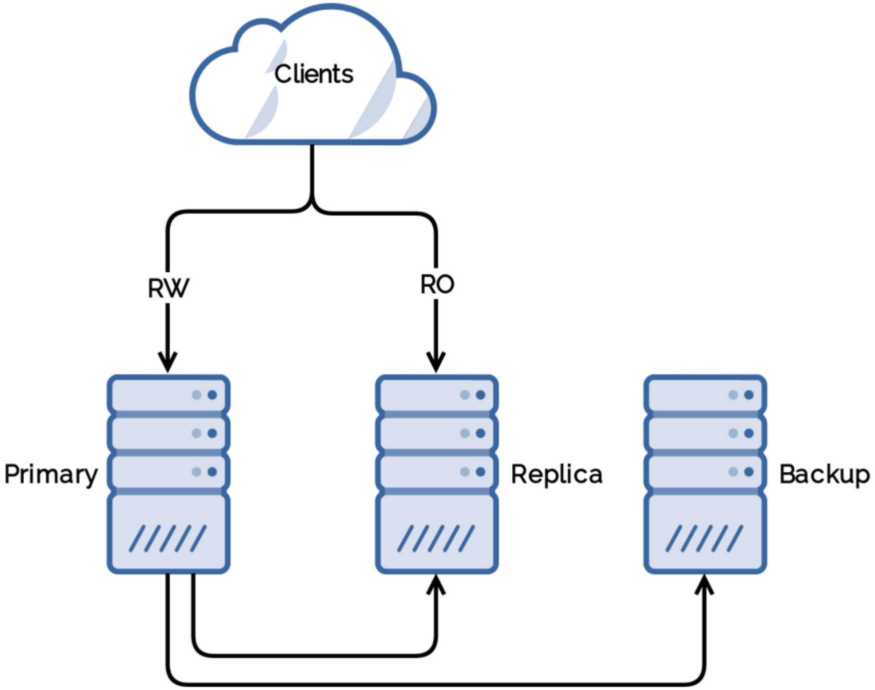
Réagir vite

En cas de panne

- Monitoring
- Développeurs
- Support
- Twitter

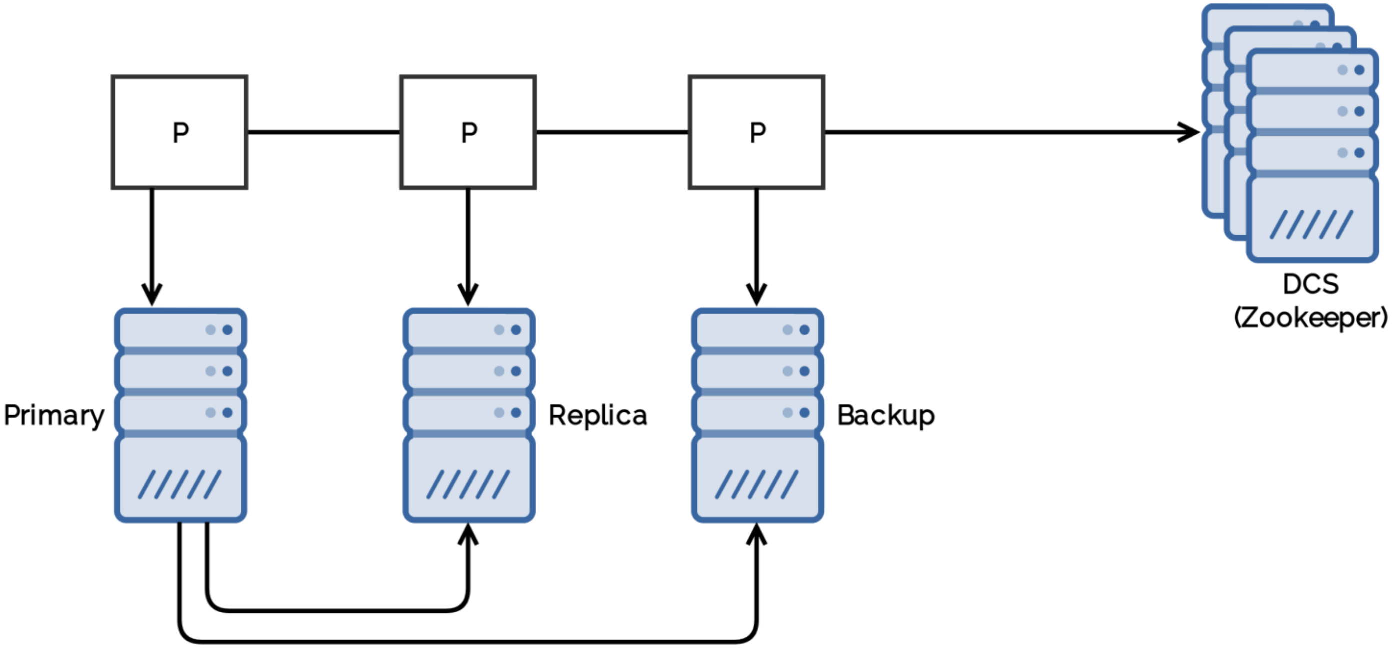
Cluster type

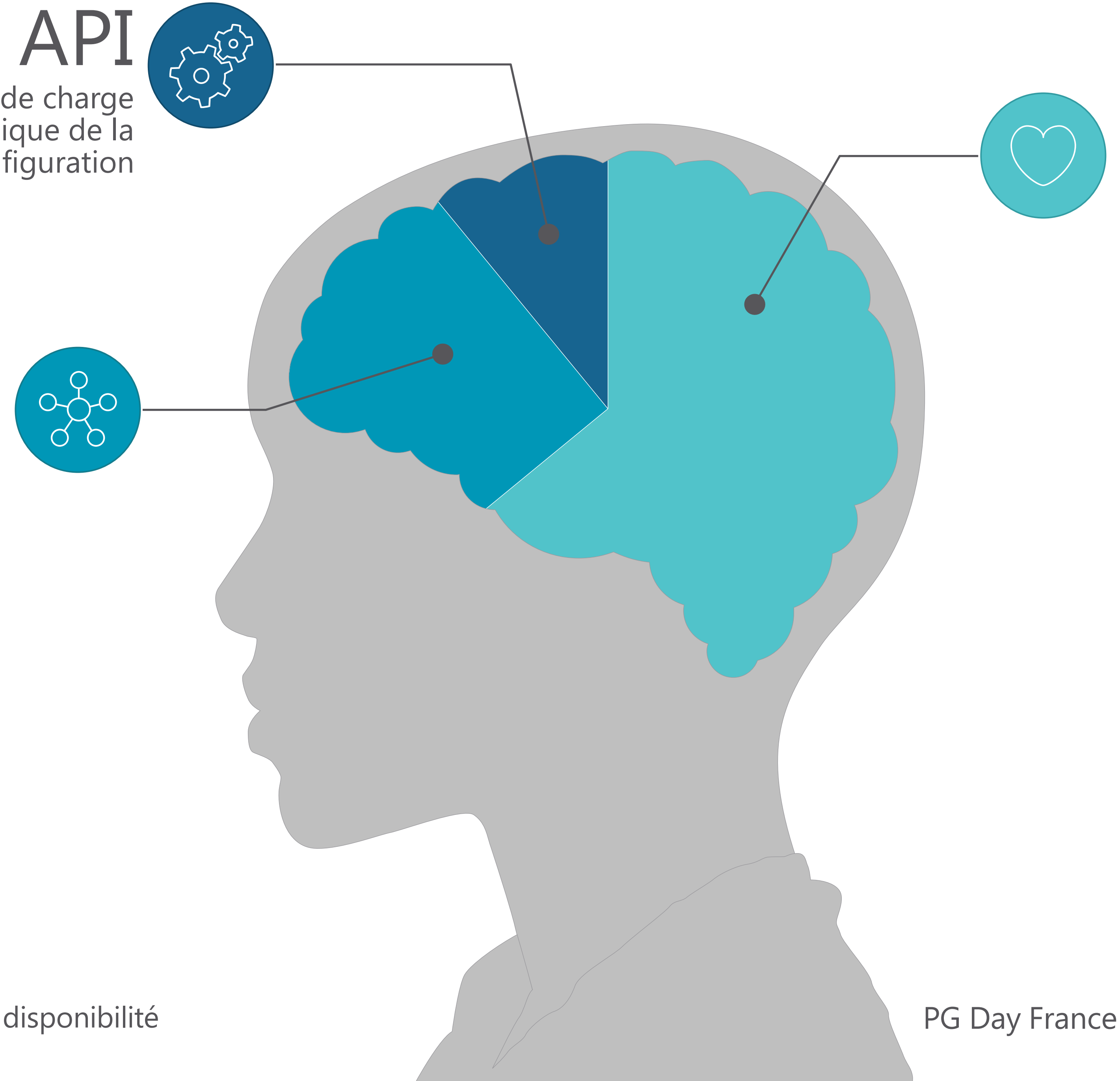
- MySQL
- PostgreSQL



Promotion automatique

- Patroni
  - Zalando
  - Opensource
  - Python





API

Répartition de charge
Gestion dynamique de la configuration

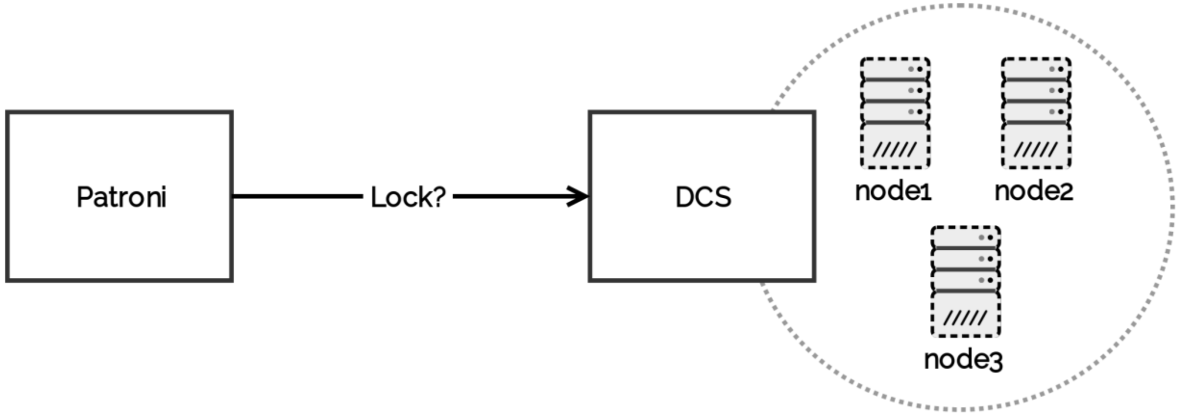
Promotion

Election de leader
Passage en mode sécurité

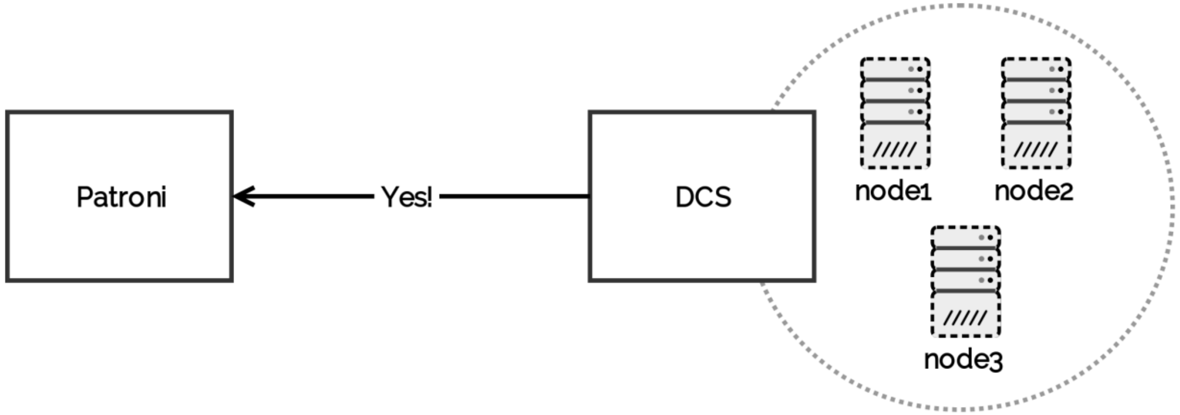
Réplication

Gestion automatique
Reprise après panne

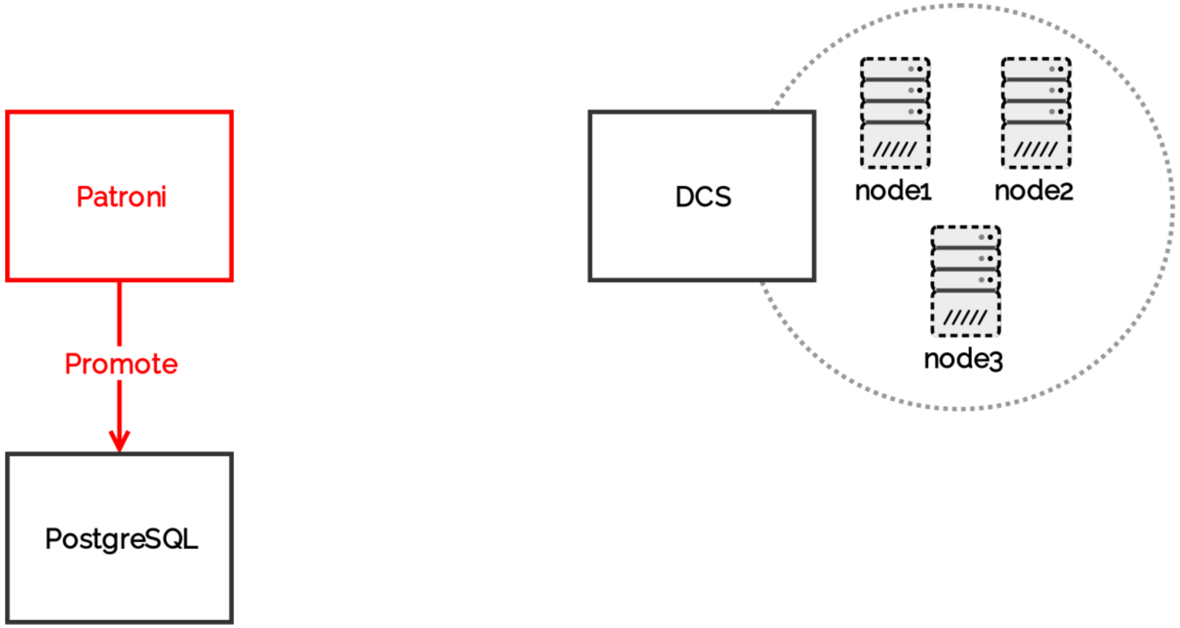
Promotion

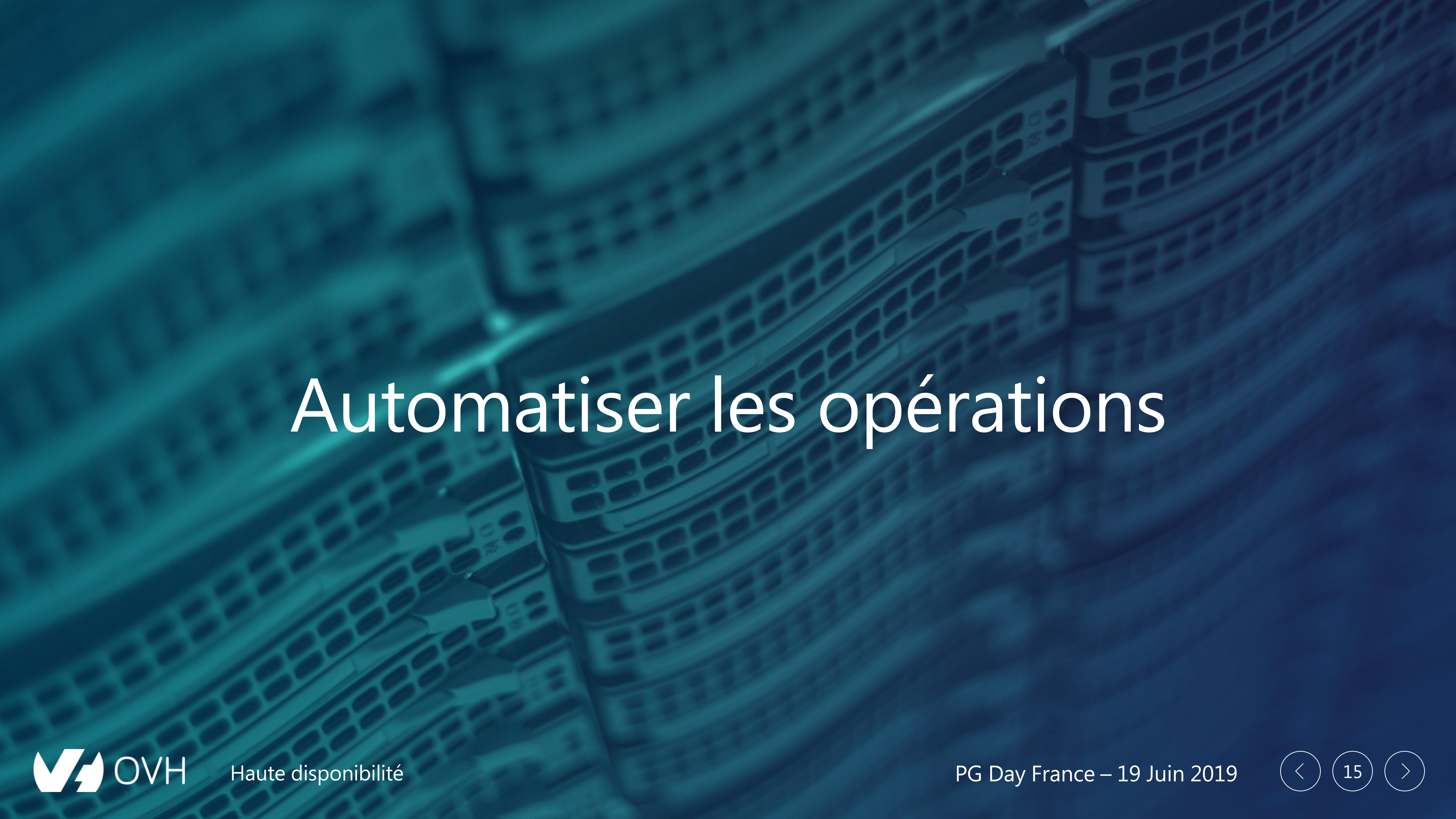


Promotion



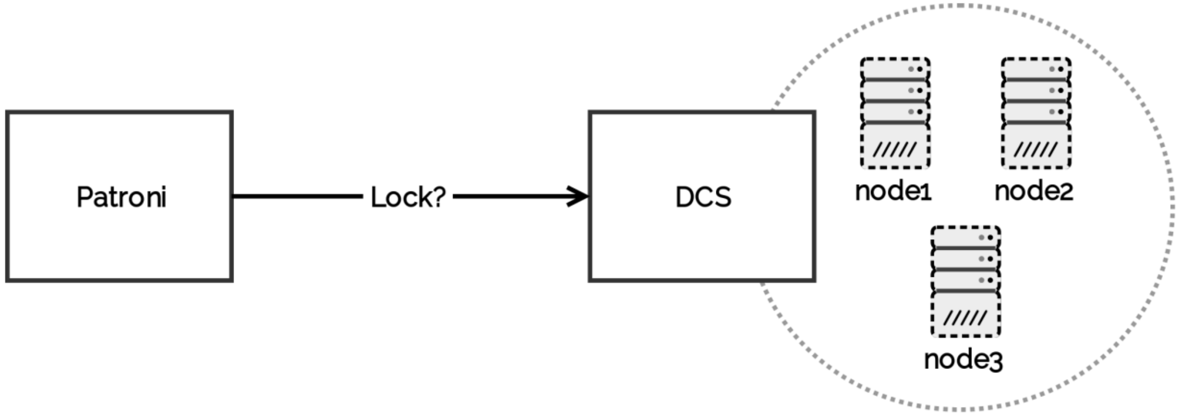
Promotion



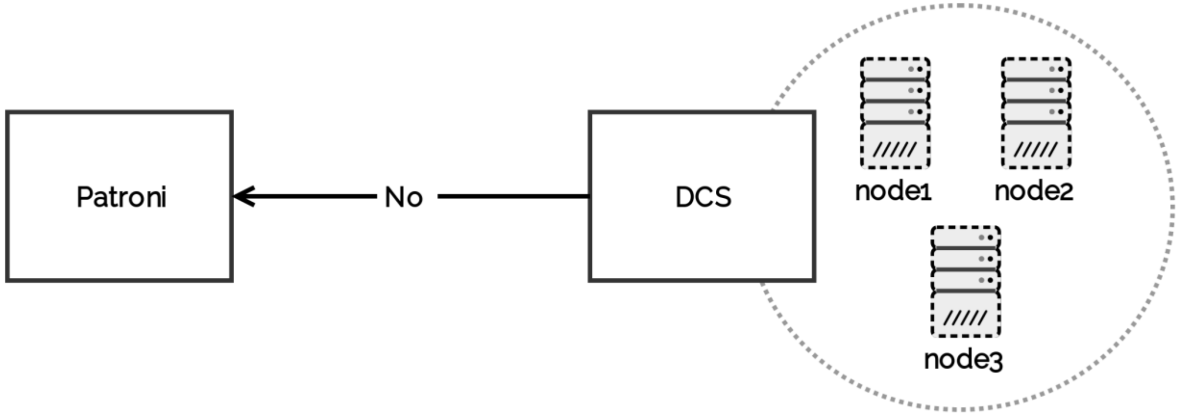


Automatiser les opérations

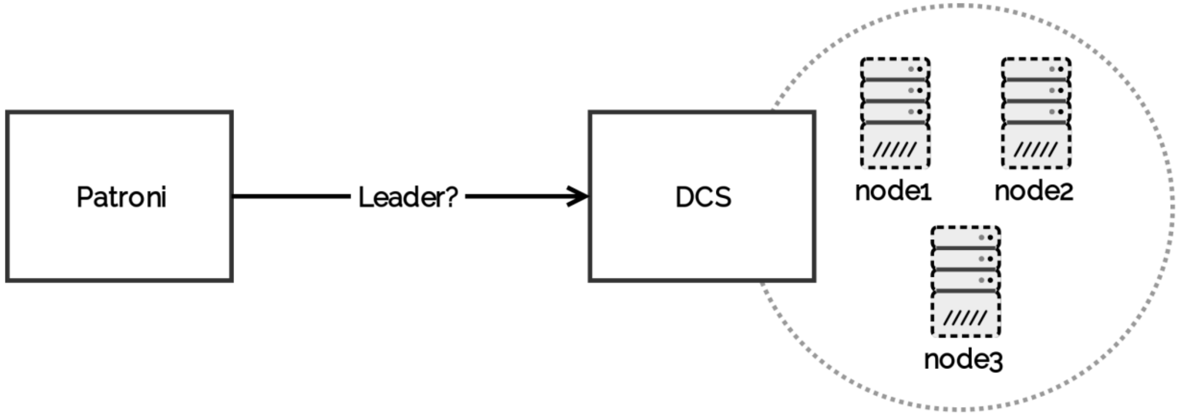
Réplication



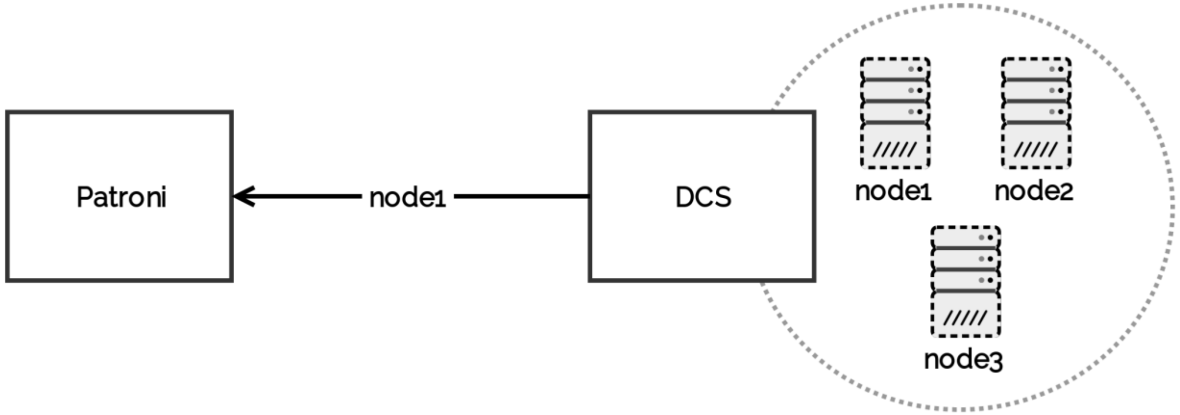
Réplication



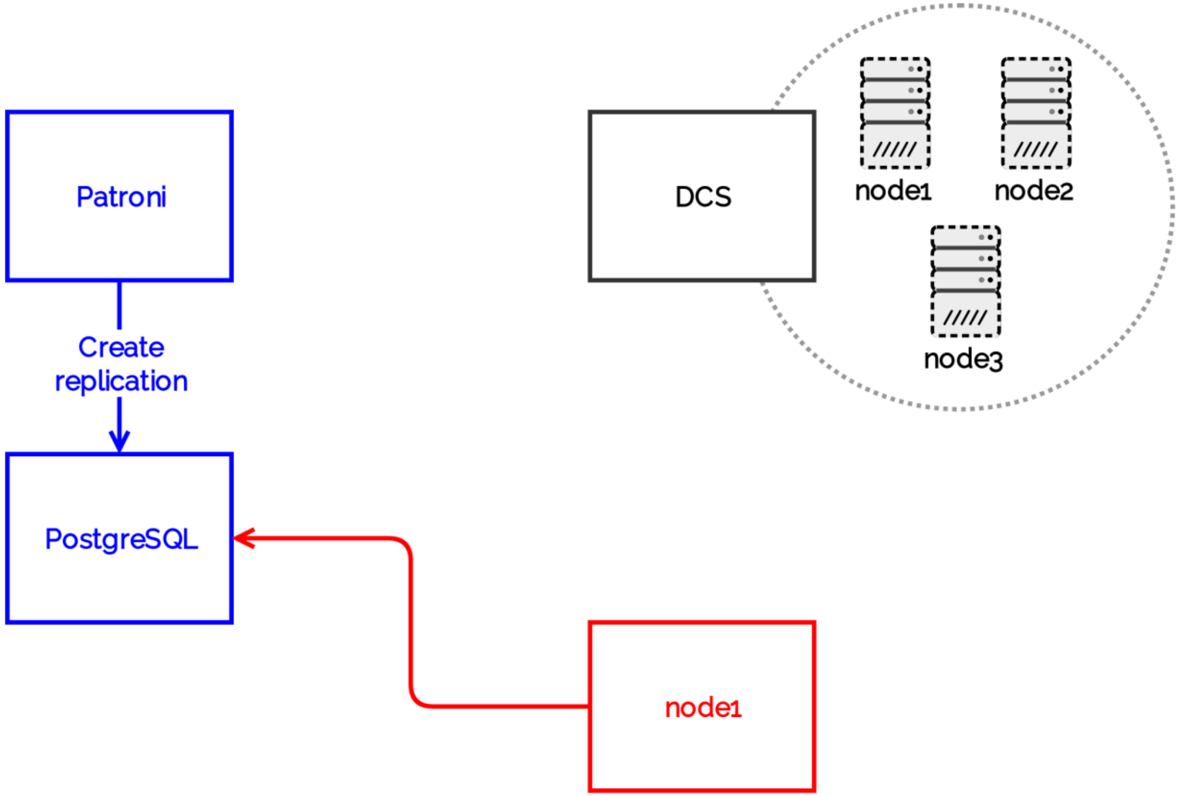
Réplication



Réplication



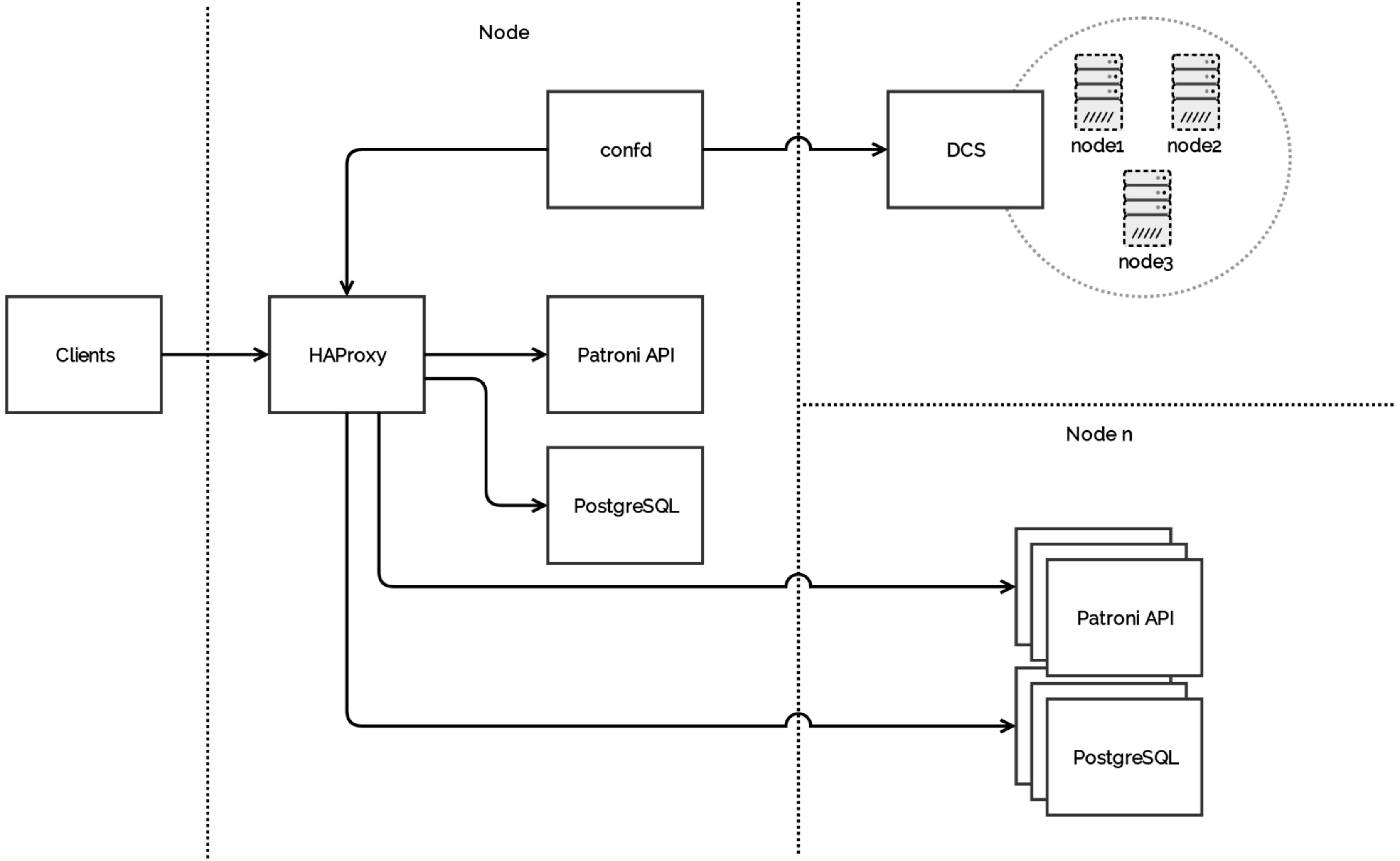
Réplication



Rediriger le trafic

Répartition de charge

- HAProxy
- Patroni API
- confd



HAProxy

- Mode TCP
- Probes HTTPS
- 2 pools de connexion (= 2 ports)
  - Lectures et écritures
  - Lectures seules
- Fichier d’état
  - server-state-base /var/lib/haproxy/state
  - socat ${sock} - <<< "show servers state ${backend}" > /var/lib/haproxy/state/${backend}

Patroni API

```
$ curl -i -X OPTIONS https://localhost:443/primary
HTTP/1.0 200 OK
$ curl -i -X OPTIONS https://localhost:443/replica
HTTP/1.0 503 Service Unavailable
```

confd

01 Découvre les nodes

02 Crée la configuration HAProxy

03 Vérifie la configuration HAProxy

04 Reload le service HAProxy

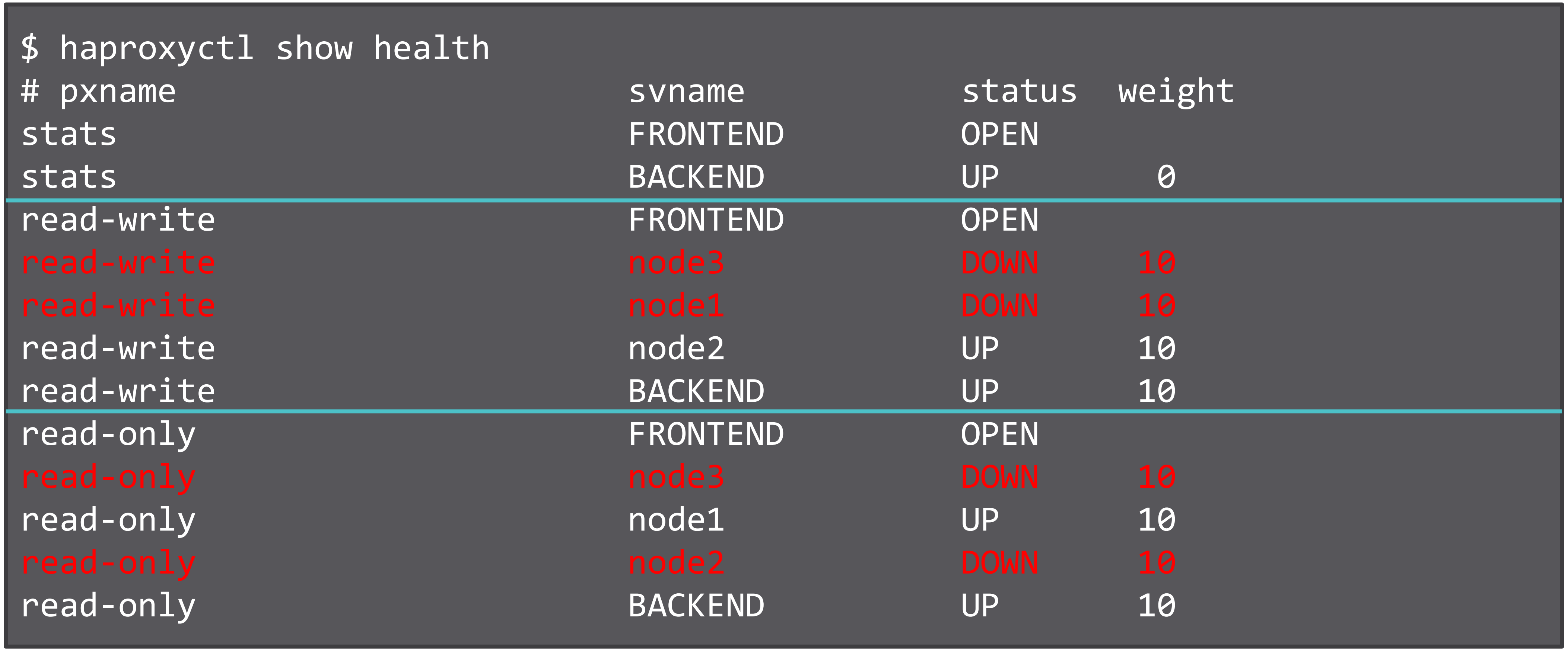
Répartition de charge

```
$ haproxyctl show health
# pxname                       svname           status  weight
stats                          FRONTEND         OPEN
stats                          BACKEND          UP        0
read-write                     FRONTEND         OPEN
read-write                     node3            DOWN     10
read-write                     node1            DOWN     10
read-write                     node2            UP       10
read-write                     BACKEND          UP       10
read-only                      FRONTEND         OPEN
read-only                      node3            DOWN     10
read-only                      node1            UP       10
read-only                      node2            DOWN     10
read-only                      BACKEND          UP       10
```

Répartition de charge

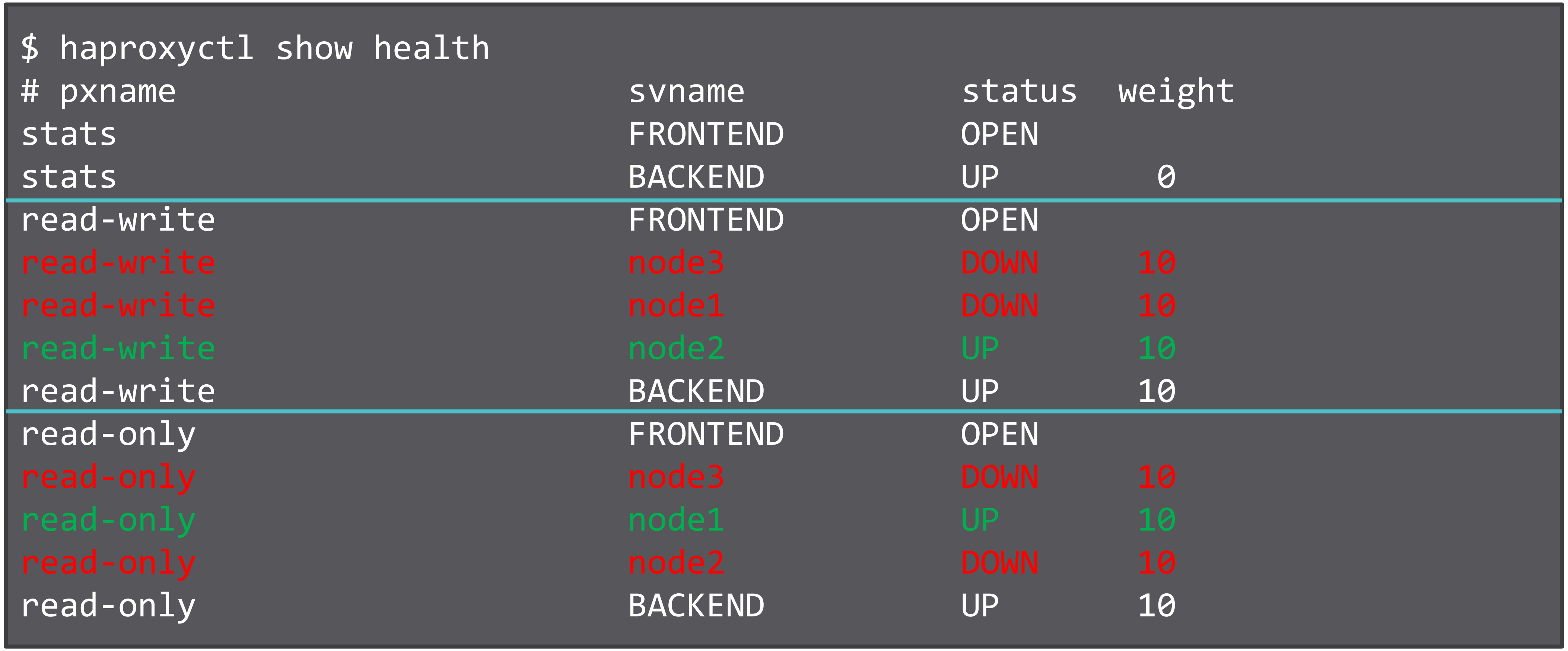
```
$ haproxyctl show health
# pxname                       svname          status  weight
stats                          FRONTEND        OPEN
stats                          BACKEND         UP          0
read-write                     FRONTEND        OPEN
read-write                     node3           DOWN       10
read-write                     node1           DOWN       10
read-write                     node2           UP         10
read-write                     BACKEND         UP         10
read-only                      FRONTEND        OPEN
read-only                      node3           DOWN       10
read-only                      node1           UP         10
read-only                      node2           DOWN       10
read-only                      BACKEND         UP         10
```

Répartition de charge



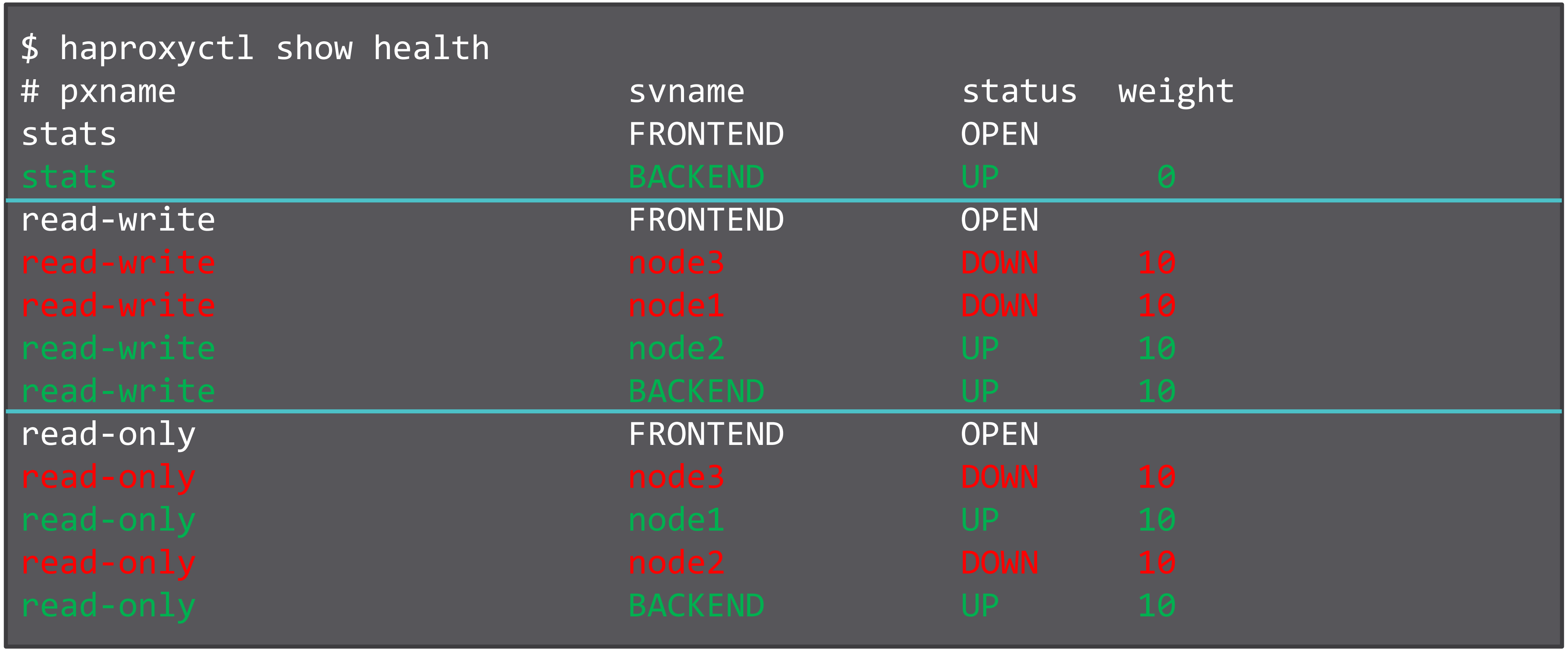
```
$ haproxyctl show health
# pxname                       svname           status  weight
stats                          FRONTEND         OPEN
stats                          BACKEND          UP        0
read-write                     FRONTEND         OPEN
read-write                     node3            DOWN     10
read-write                     node1            DOWN     10
read-write                     node2            UP       10
read-write                     BACKEND          UP       10
read-only                      FRONTEND         OPEN
read-only                      node3            DOWN     10
read-only                      node1            UP       10
read-only                      node2            DOWN     10
read-only                      BACKEND          UP       10
```

Répartition de charge



```
$ haproxyctl show health
# pxname                       svname           status  weight
stats                          FRONTEND         OPEN
stats                          BACKEND          UP        0
read-write                     FRONTEND         OPEN
read-write                     node3            DOWN     10
read-write                     node1            DOWN     10
read-write                     node2            UP       10
read-write                     BACKEND          UP       10
read-only                      FRONTEND         OPEN
read-only                      node3            DOWN     10
read-only                      node1            UP       10
read-only                      node2            DOWN     10
read-only                      BACKEND          UP       10
```

Répartition de charge



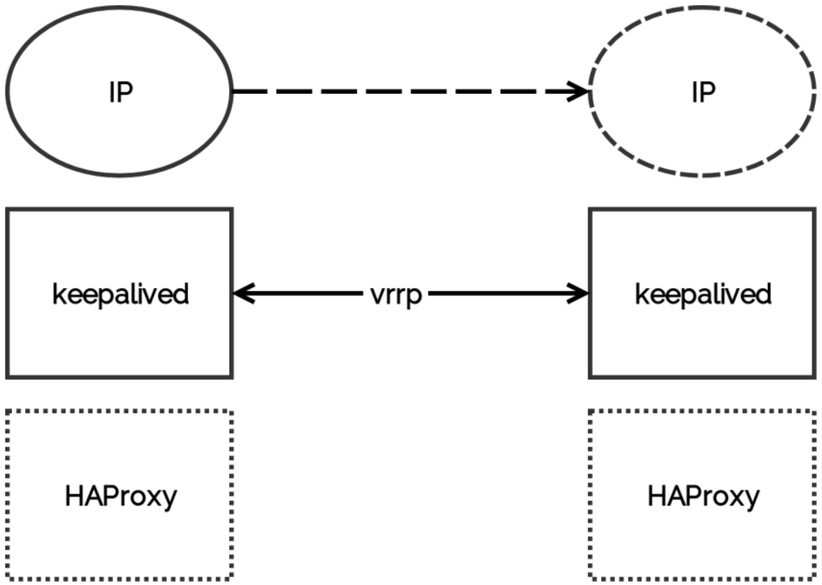
```
$ haproxyctl show health
# pxname                        svname           status  weight
stats                           FRONTEND         OPEN
stats                           BACKEND          UP        0
read-write                      FRONTEND         OPEN
read-write                      node3            DOWN     10
read-write                      node1            DOWN     10
read-write                      node2            UP       10
read-write                      BACKEND          UP       10
read-only                       FRONTEND         OPEN
read-only                       node3            DOWN     10
read-only                       node1            UP       10
read-only                       node2            DOWN     10
read-only                       BACKEND          UP       10
```

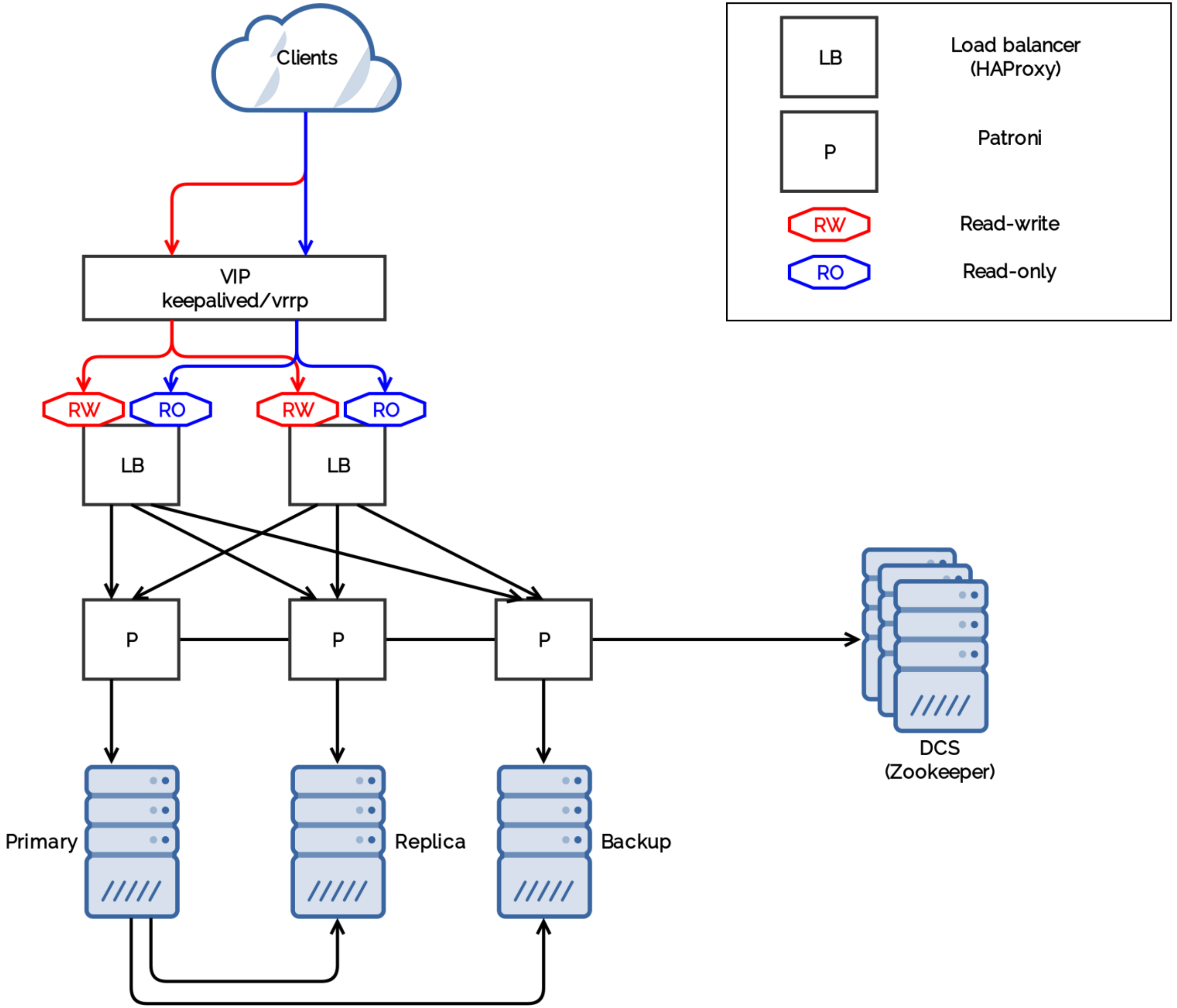
Répartition de charge

```
$ haproxyctl show backends
stats                          BACKEND                        UP         0
read-write                     BACKEND                        UP        10
read-only                      BACKEND                        UP        10
```

IP virtuelle

- keepalived/vrrp
- iputils-arping
- notify_master /usr/bin/arping -U -c 4 $IP




Clients
VIP
keepalived/vrrp
RW
RO
RW
RO
LB
LB
P
P
P
Primary
Replica
Backup
DCS
(Zookeeper)
LB
Load balancer
(HAProxy)
P
Patroni
RW
Read-write
RO
Read-only

Promotion automatique

Crédits : https://github.com/googlei18n/noto-emoji/blob/master/svg/emoji_u1f60c.svg

Promotion automatique

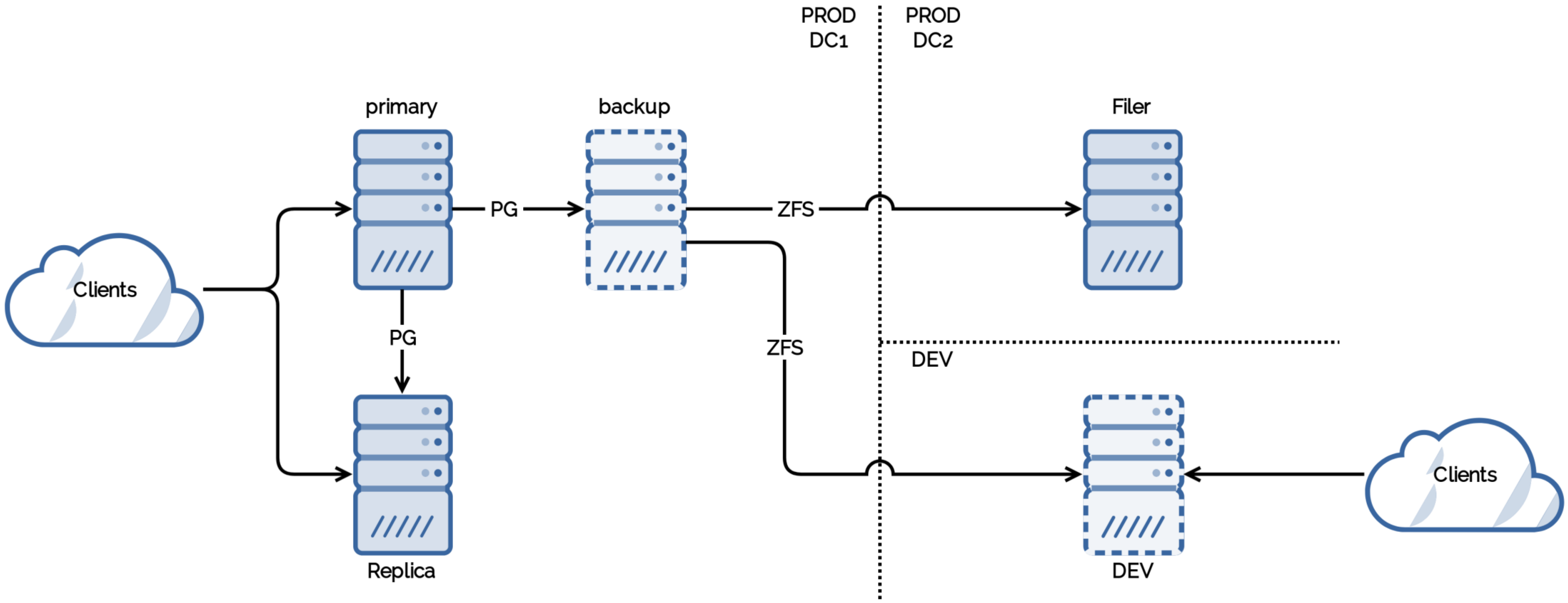
- Proxy non transparent
  - pg_hba.conf
  - Logs
- Protocole PROXY
- application_name

(Re)Trouver ses données

Sauvegardes et restaurations

- Accident logique
  - DROP <objet> (DATABASE, ROLE, TABLE, …)
- Accident physique (panne hardware)
- Sauvegardes impactante (I/O, locks)
- Compatibilité avec le reste de l’infrastructure
- Pas de sauvegarde sans test de restauration

Sauvegardes et restaurations



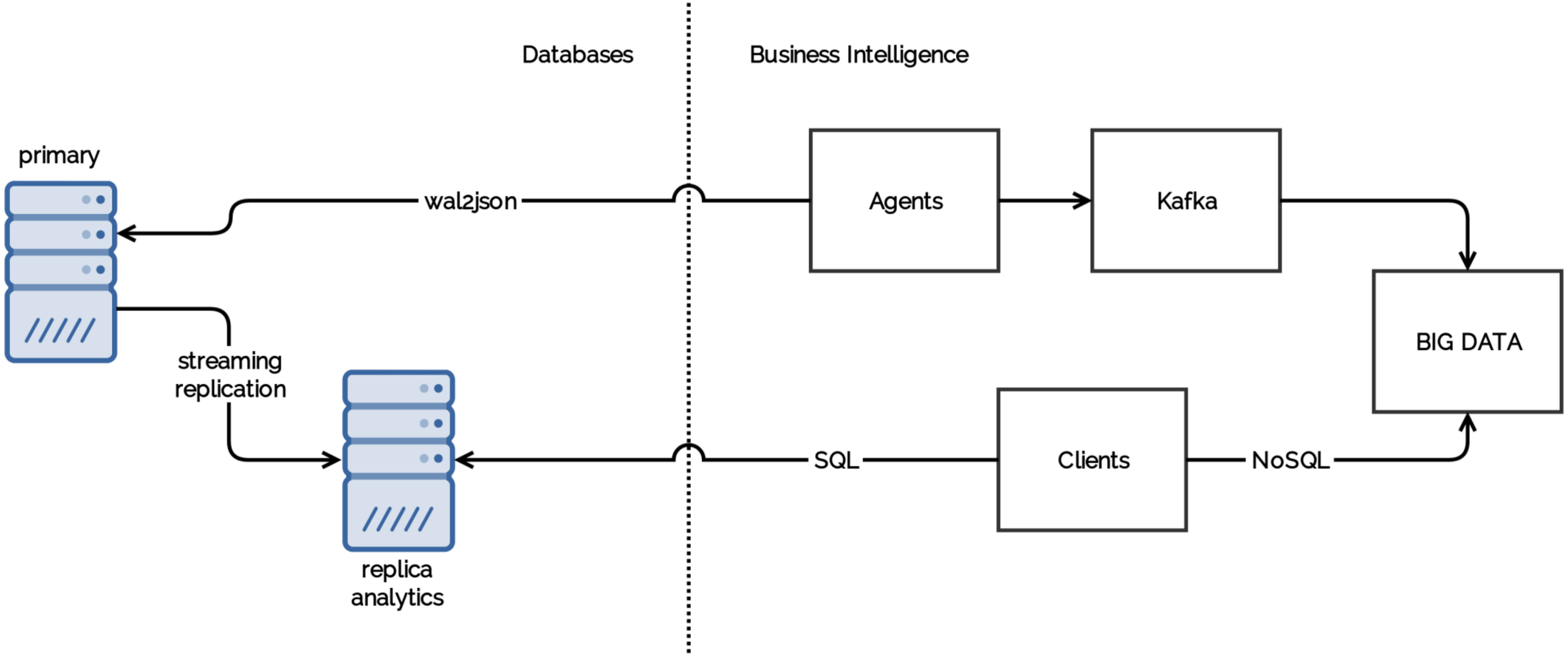
Faciliter l’analyse

Business intelligence

- OLTP vs OLAP
  - OLTP (On-line Transaction Processing)
  - OLAP (On-line Analytical Processing)

- Les deux charges ne vont pas ensemble
- Comment ne pas bloquer le système operationel ?

Business intelligence



Avoir un système à jour

Mises à jour mineures

- clustershell

```
$ clush -bw @patroni
$ clush -bw @patroni\&@cluster:69
$ clush –bw node1,node2,node3
Enter 'quit' to leave this interactive mode
Working with nodes: node[1-3]
clush> psql -c 'show server_version;'
---------------
node[1-3] (3)
---------------
server_version
----------------
 9.6.11
(1 row)
```

Mises à jour mineures

- clustershell

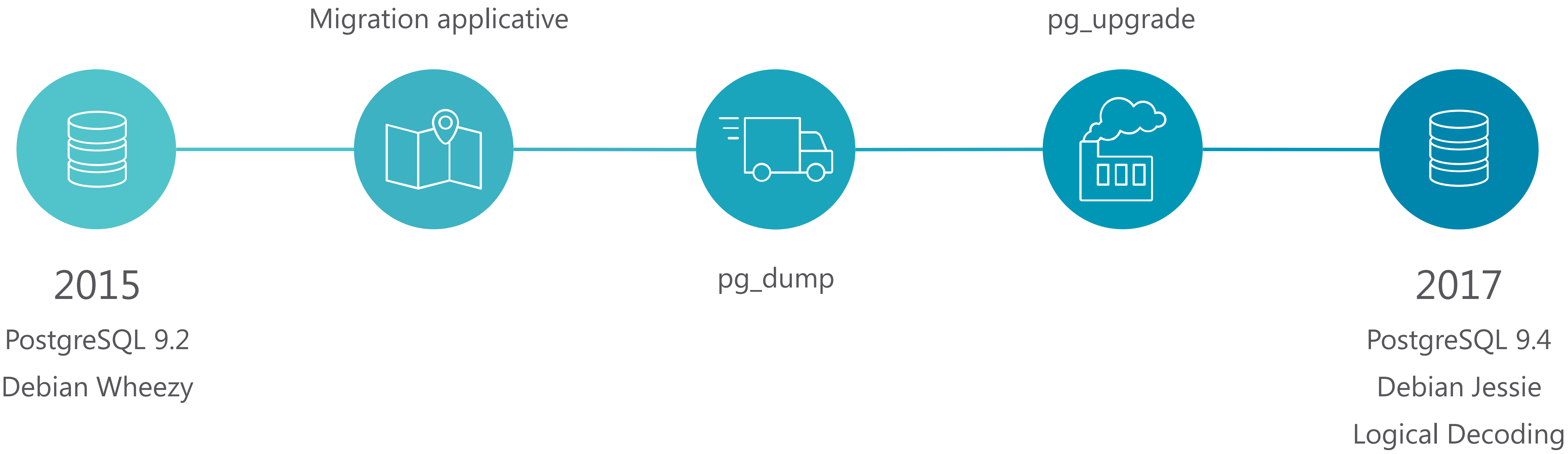
```
$ clush -f 1 -bw node1,node2,node3
Enter 'quit' to leave this interactive mode
Working with nodes: node[1-3]
clush> apt-get upgrade -y
```

Mises à jour mineures

- clustershell

```
$ clush -bw node1,node2,node3
Enter 'quit' to leave this interactive mode
Working with nodes: node[1-3]
clush> psql -c 'show server_version;'
---------------
node[1-3] (3)
---------------
server_version
----------------
 9.6.12
(1 row)
```

Mises à jour majeures



Migration applicative

pg_upgrade

2015

PostgreSQL 9.2

Debian Wheezy

pg_dump

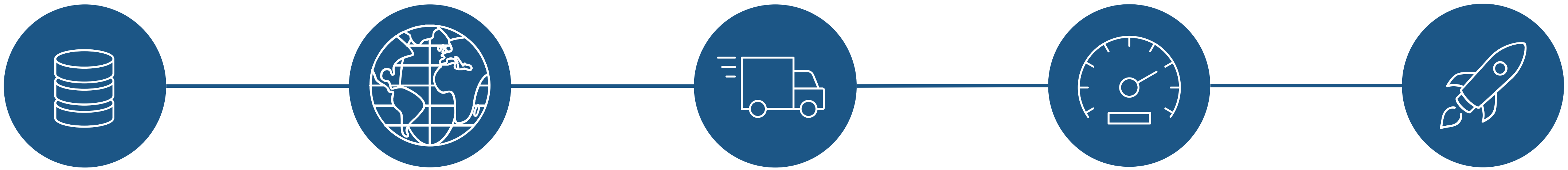
2017

PostgreSQL 9.4

Debian Jessie

Logical Decoding

Mises à jour majeures



pglogical

pg_upgrade

2017

PostgreSQL 9.4

Debian Jessie

Standalone

pg_dump

2019

PostgreSQL 9.6

Debian Stretch

Patroni

pglogical

- deadlocks

```
ERROR:  deadlock detected at character 237
DETAIL:  Process 16477 waits for AccessShareLock on relation 17241 of database 17032;
blocked by process 17333.
         Process 17333 waits for AccessExclusiveLock on relation 4920800 of database
17032; blocked by process 16477.
         Process 16477: <application query>
         Process 17333: SELECT pglogical.replication_set_add_all_tables('default',
ARRAY['public']);
HINT:  See server log for query details.
STATEMENT:  <application query>
```

pglogical

- Charset

```
ERROR:  encoding conversion for binary datum not supported yet
DETAIL:  expected_encoding UTF8 must be unset or match server_encoding SQL_ASCII
CONTEXT:  slot "pgl_<slotname>", output plugin "pglogical_output", in the startup
callback
LOG:  could not receive data from client: Connection reset by peer
```

- Non supporté (doc)

> ***4.13 Database encoding differences***
> *PGLogical does not support replication between databases with different encoding. We recommend using UTF-8 encoding in all replicated databases.*

pglogical

- Séquences

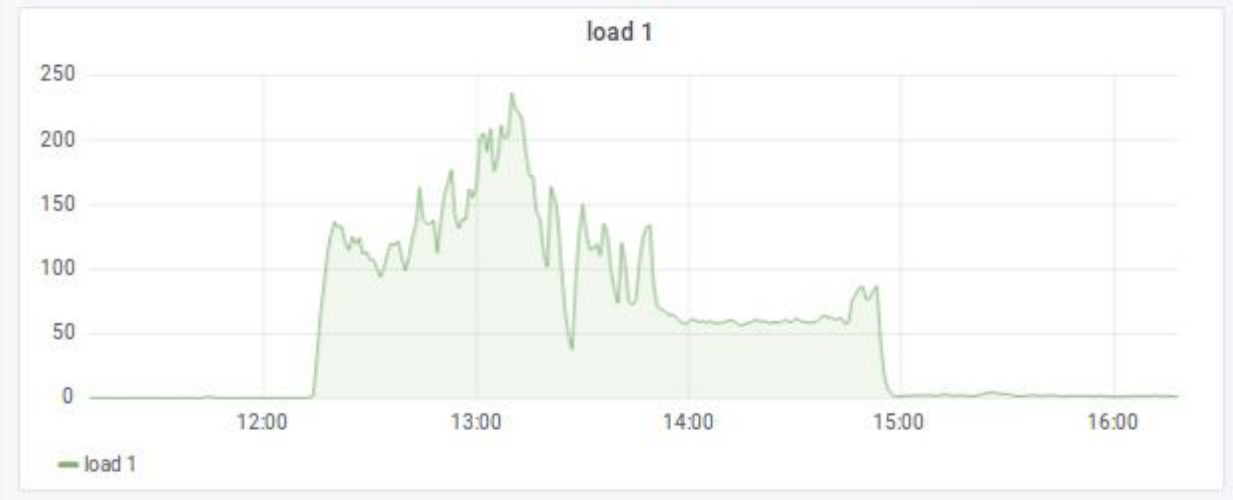
```
ERROR:  duplicate key value violates unique constraint "table_pkey"
```

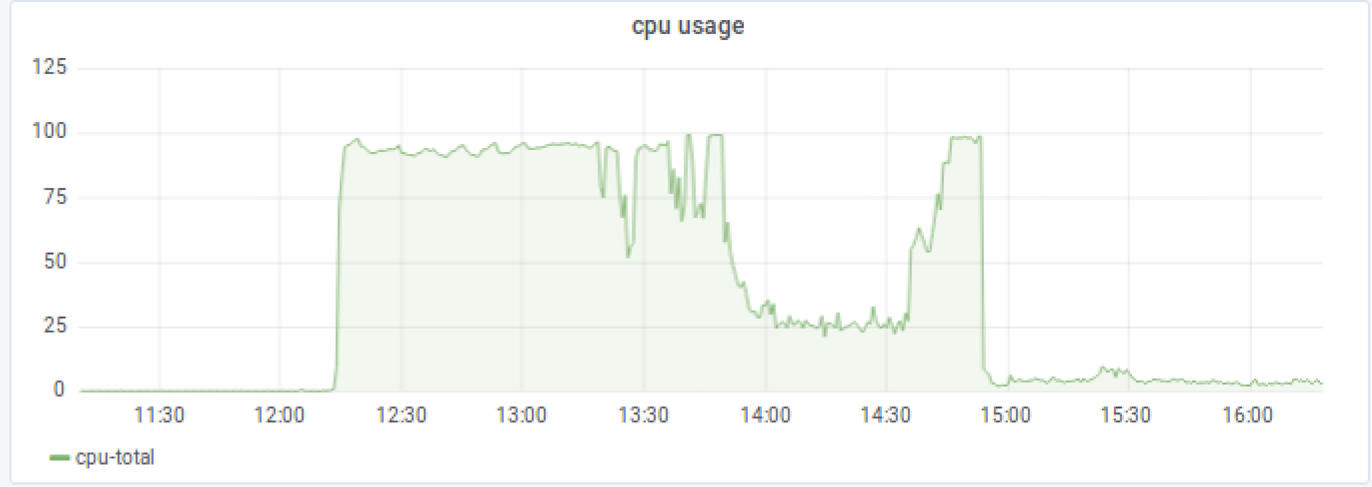
pg_upgrade

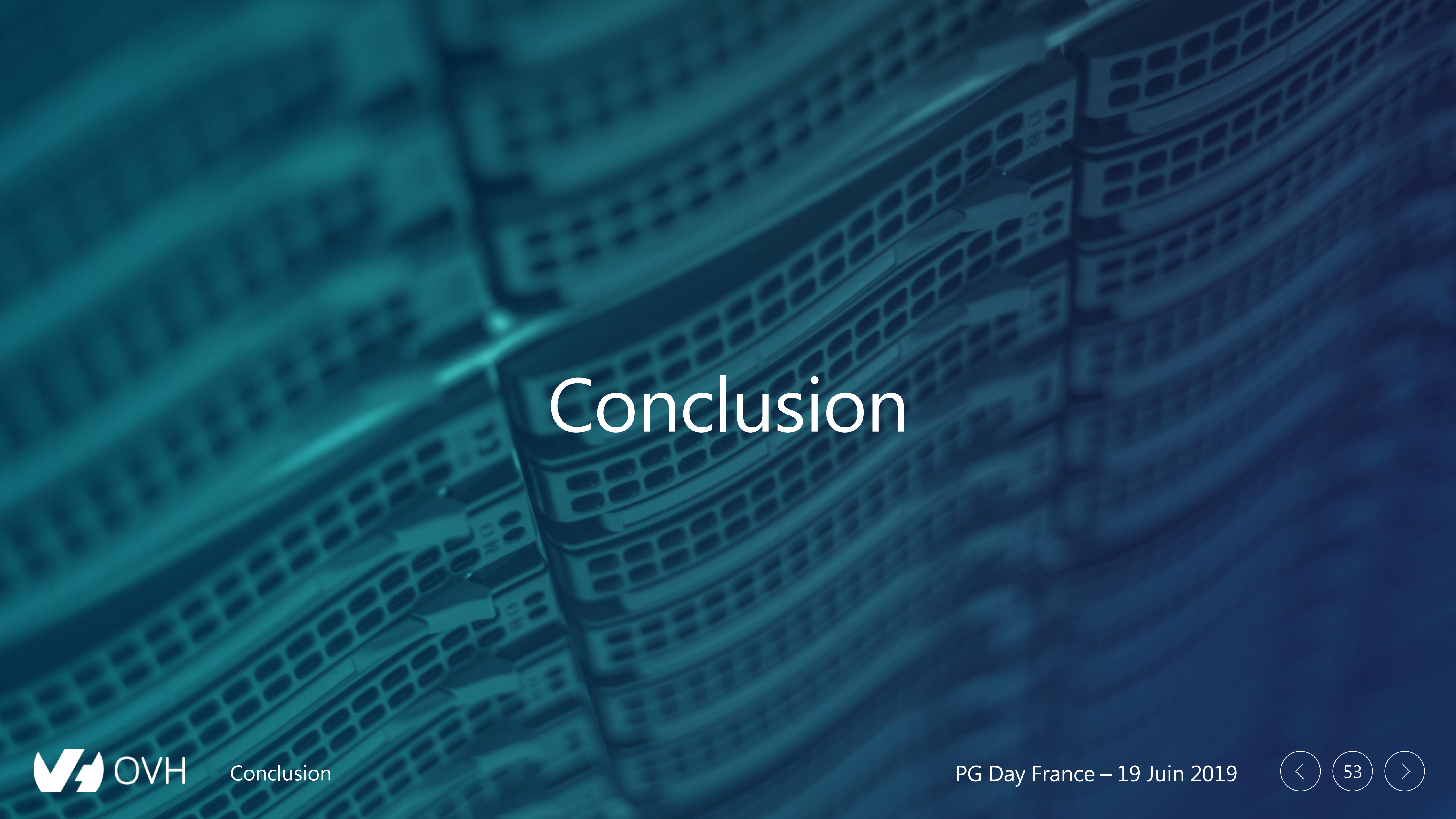
- Hardlinks (-k)
- Statistiques

```
$ vacuumdb --all --analyze-in-stages -j 10
```

pg_upgrade


load 1
250
200
150
100
50
0
12:00
13:00
14:00
15:00
16:00
load 1


cpu usage
125
100
75
50
25
0
11:30
12:00
12:30
13:00
13:30
14:00
14:30
15:00
15:30
16:00
cpu-total

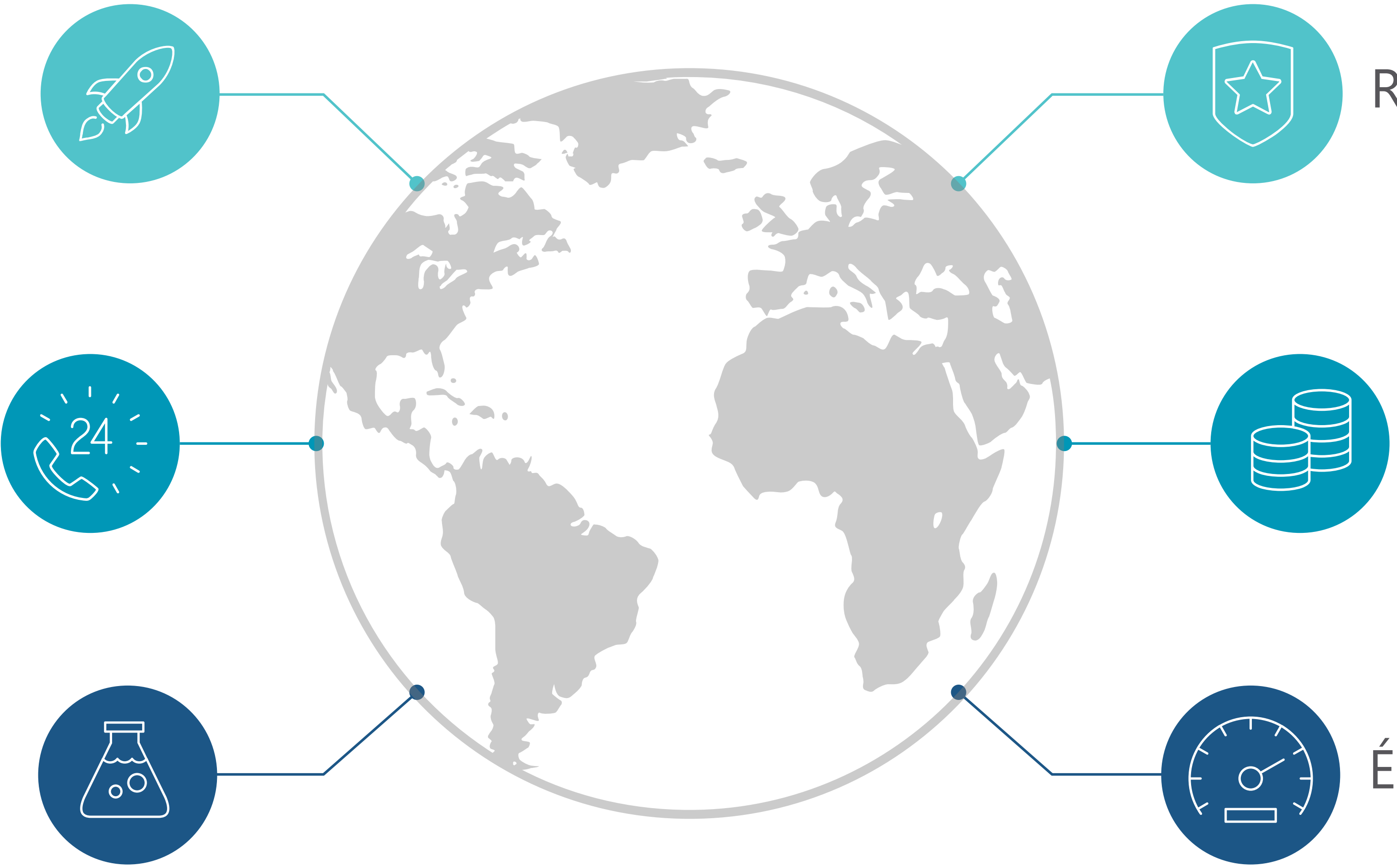


Conclusion

Conclusion

- Meilleure disponibilité
- Meilleure stabilité
- Moins d’alertes
- Moins d’administration

Bases de données externes



Performance

Managed

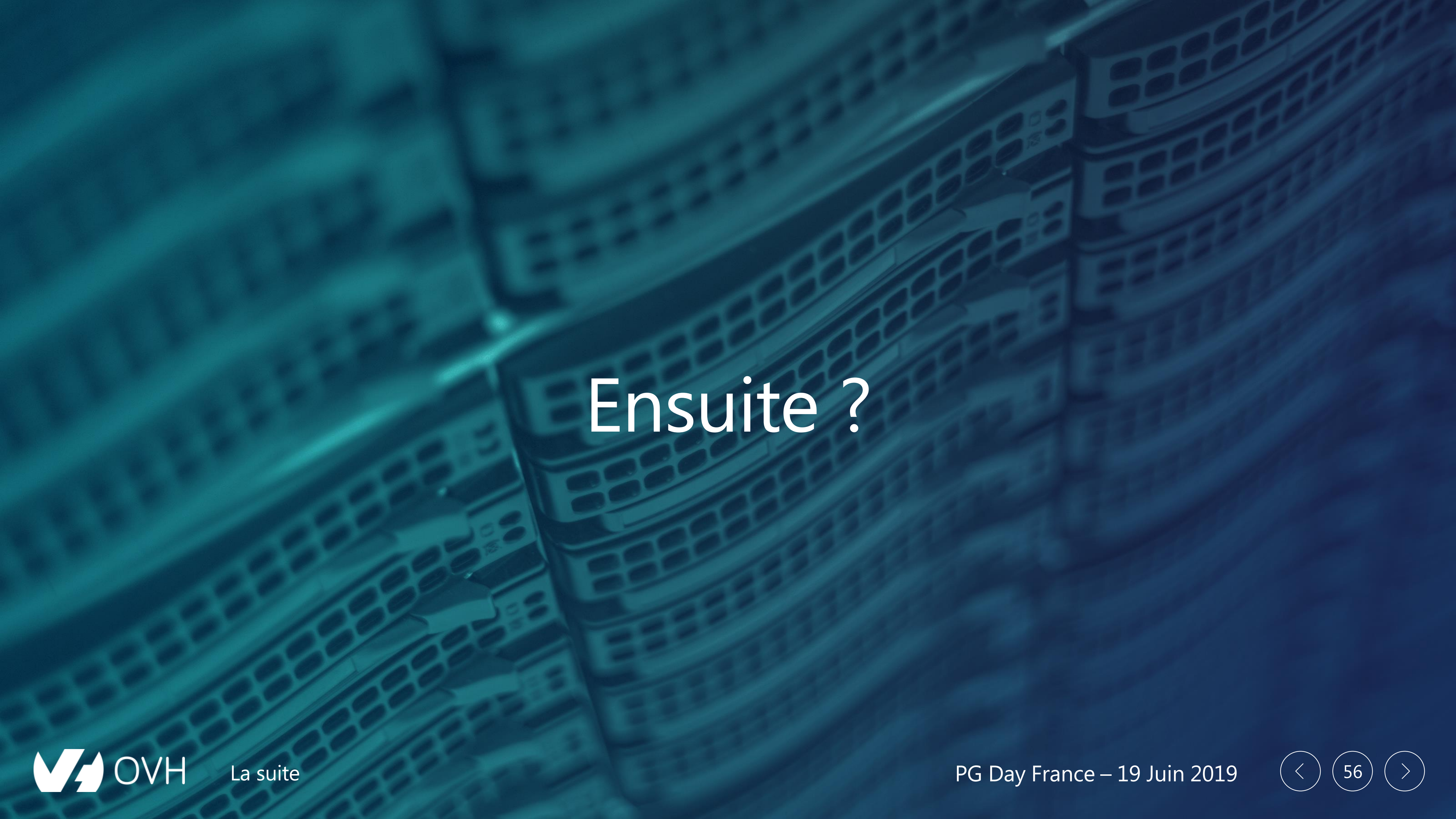
Eat your own food

Réversibilité

Haute disponibilité

Élasticité

Ensuite ?



La suite

- Mise à jour vers PostgreSQL 12
- Mise à jour vers Debian 10
- Migration de MySQL vers PostgreSQL
- Automatisation, automatisation, automatisation !

On recrute

- Opensource Database Engineers
- Site Reliability Engineers (Private Cloud, Openstack, DNS, Deploy, Observability)
- Software Engineers (containers, baremetal, web hosting)
- Backend Developpers (Python, Go)
- Et plus !

Questions