



PERCONA

*Implementando alta
disponibilidade e
continuidade de serviço no
MySQL nos dias atuais*

Fernando Laudares Camargos
Senior Architect @ Percona

MySQL BR Conf 2025

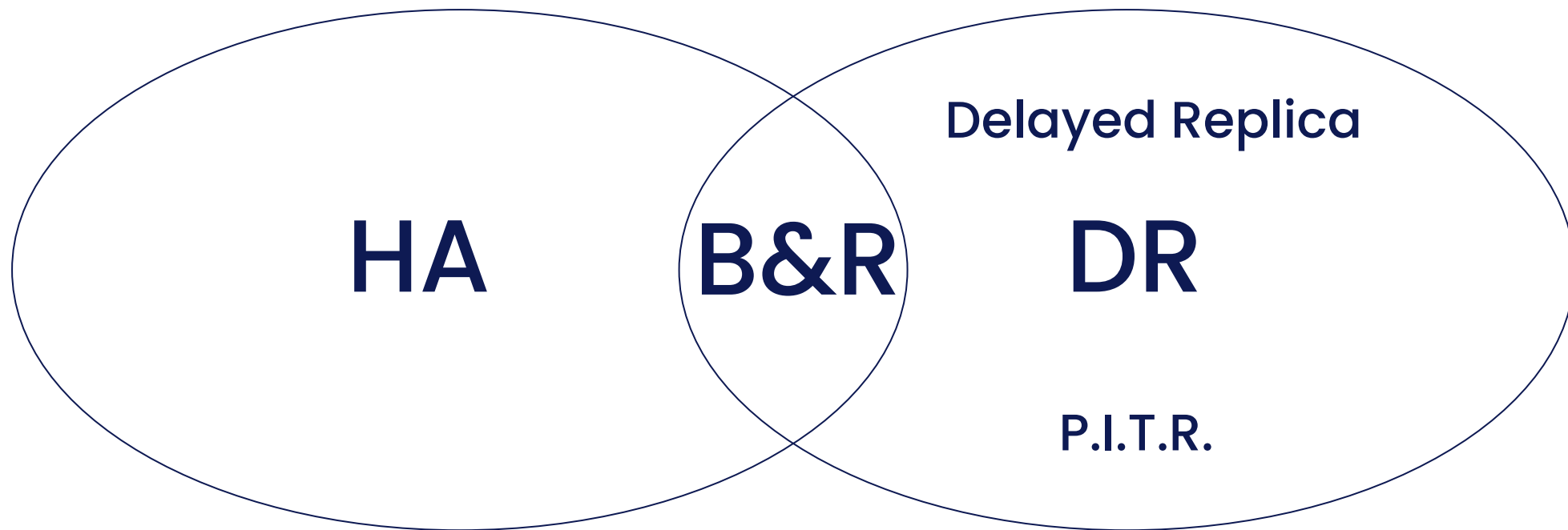
Definindo

Continuidade de serviço

- Alta disponibilidade ("HA")
- Disaster Recovery (DR)

Definindo

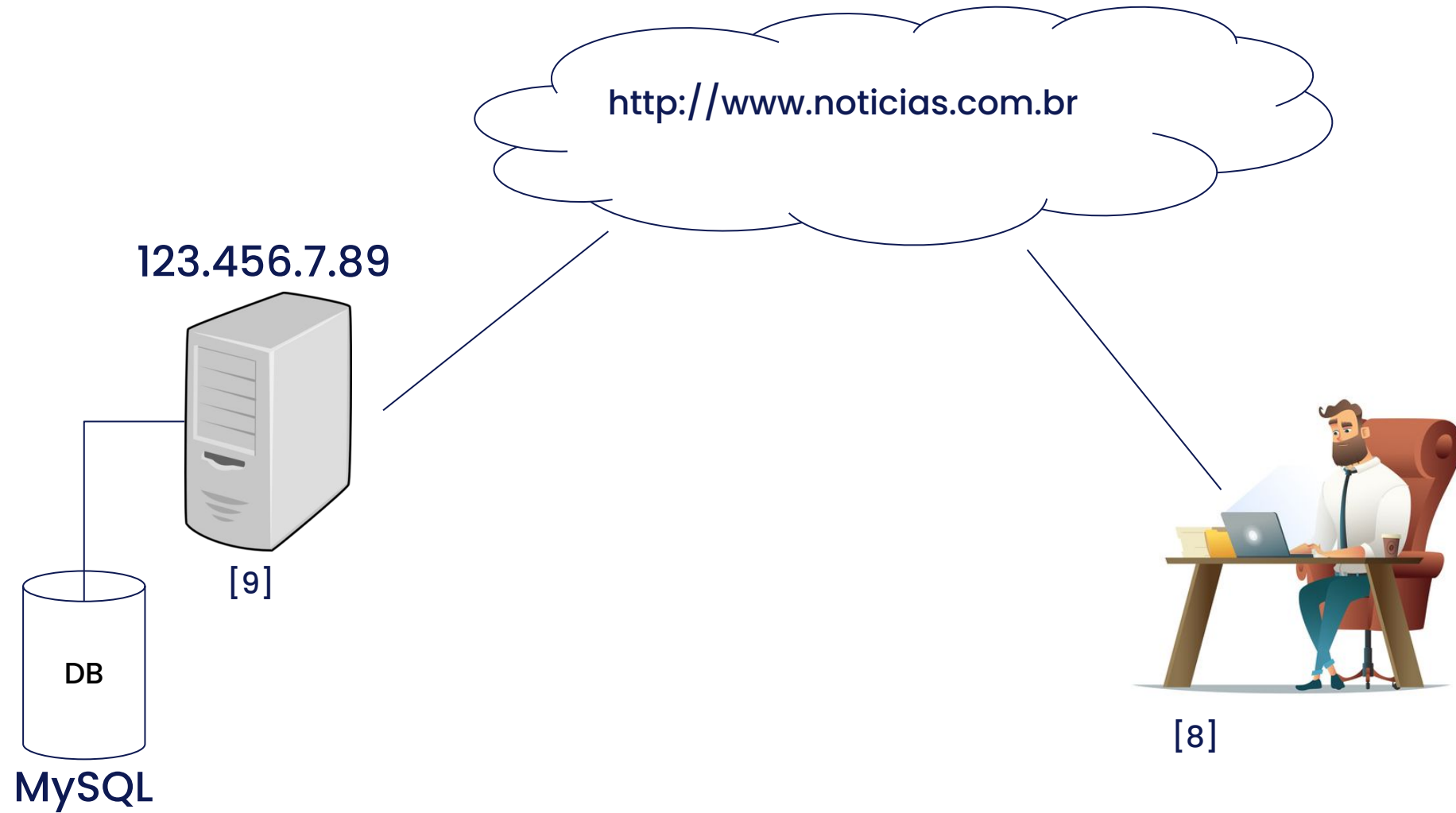
Continuidade de serviço



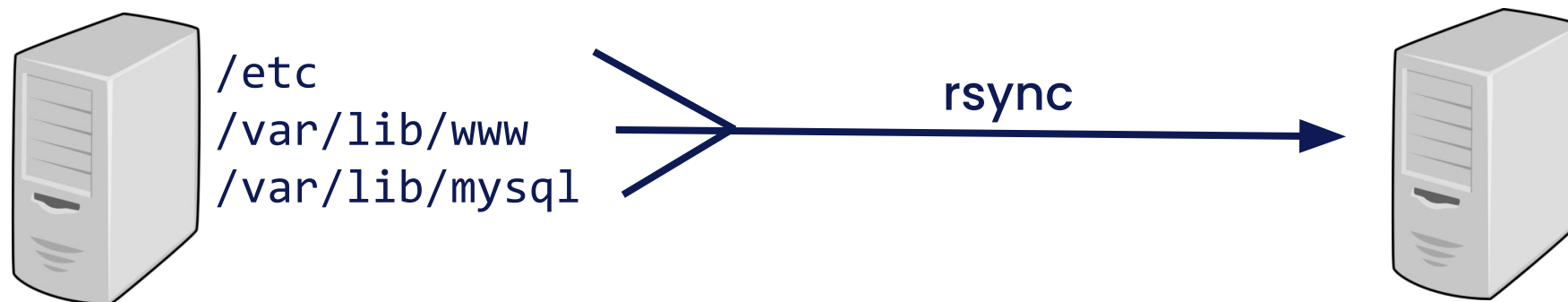


[1]

LAMP



Continuidade de serviço baseada em backups



Continuidade de serviço baseada em backups

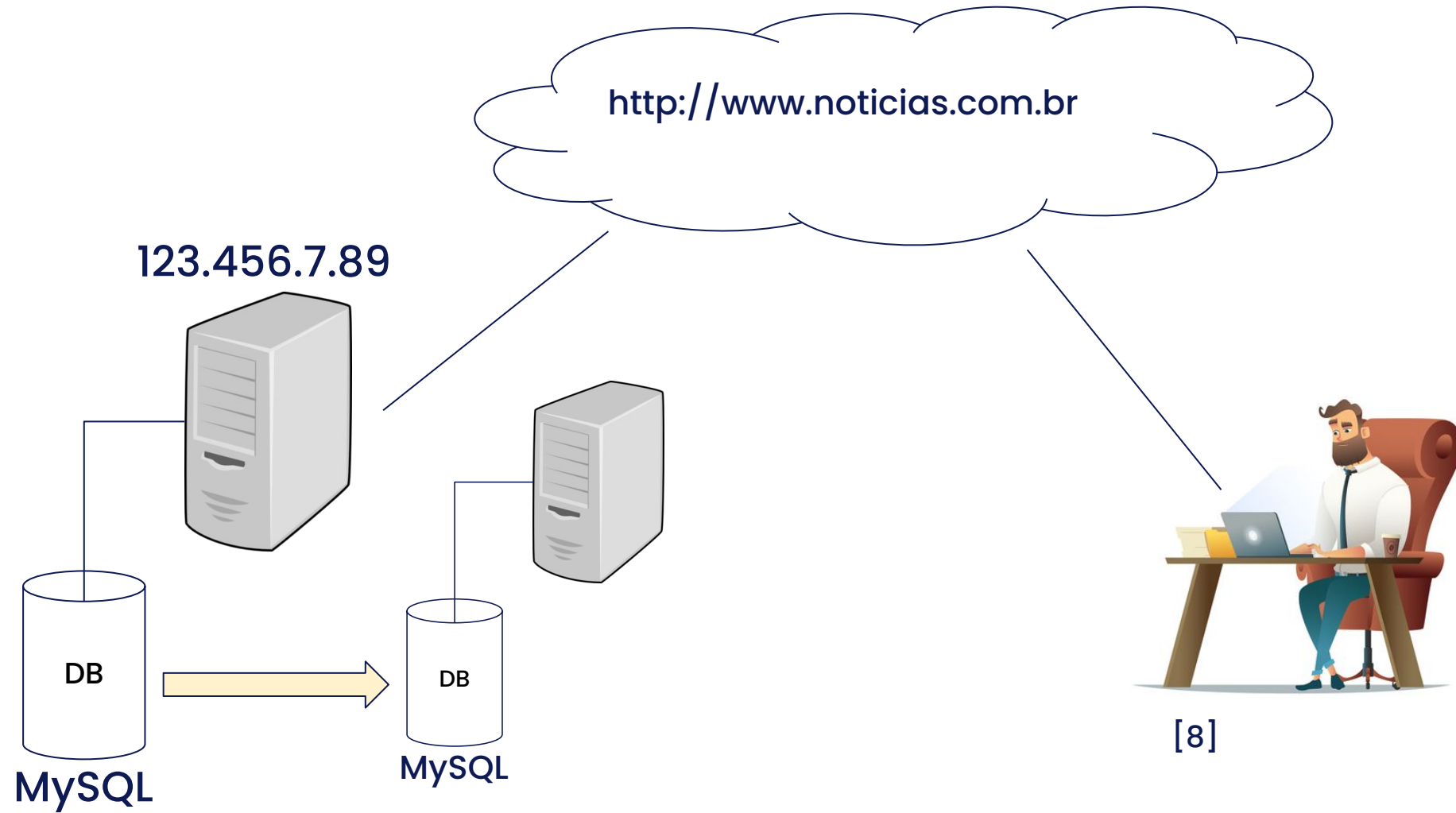


MyISAM

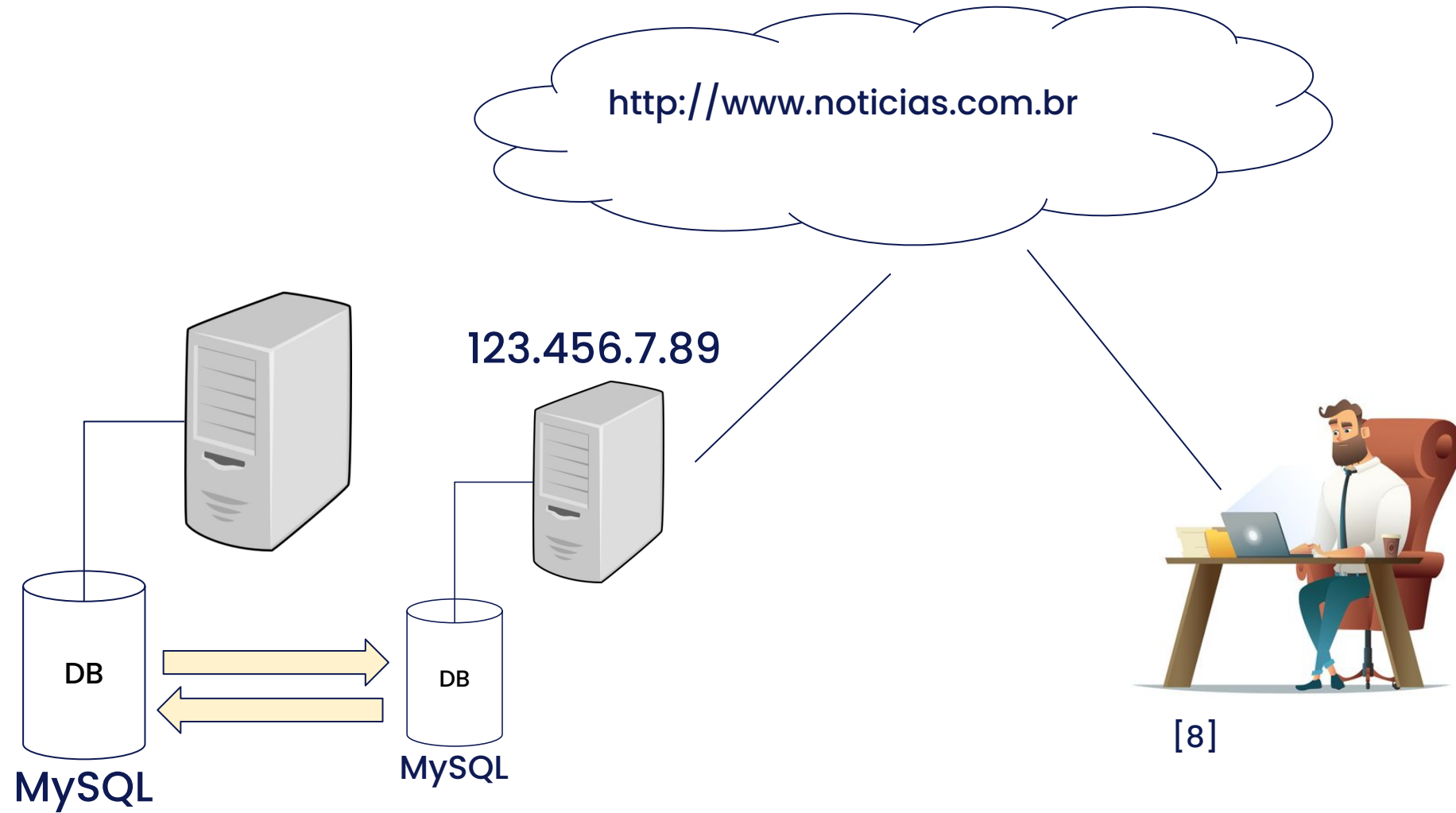


Replicação

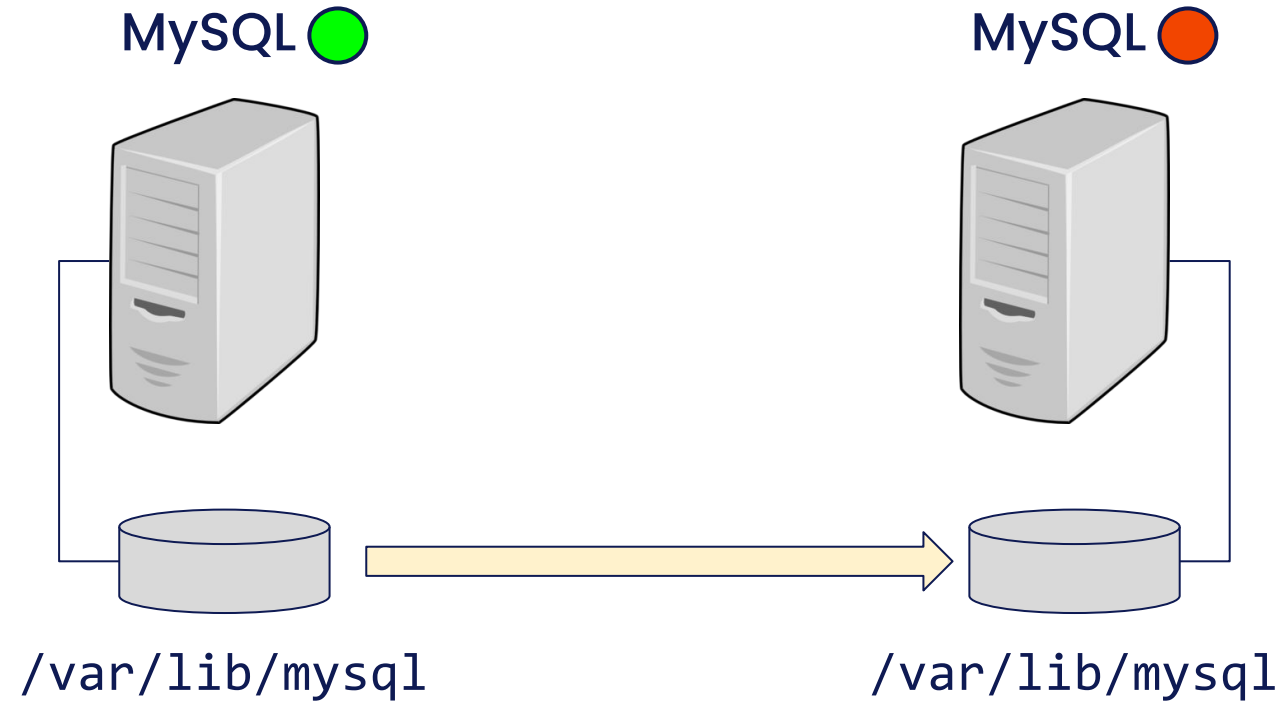
LAMP



LAMP



DRDB



Replicação no MySQL

- Introdução com SBR em 2000 (3.23)
- MBR and RBR em 2006 (5.1)
- GTIDs, MTS, crash-safe em 2013 (5.6)
- Depois:
 - Multi-source replication
 - Group replication (2015), InnoDB Cluster
- Galera Cluster em 2007

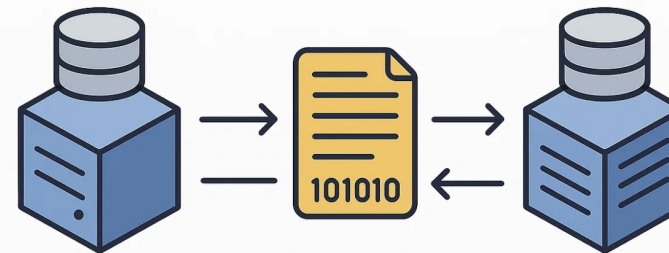
A Brief History of MySQL Replication

 Marcelo Altmann [Follow](#) 3 min read · May 13, 2025

 3  1

While preparing slides for a presentation, I found myself researching about the history of MySQL Replication and decided to write some key moments of this wonderful functionality of MySQL.



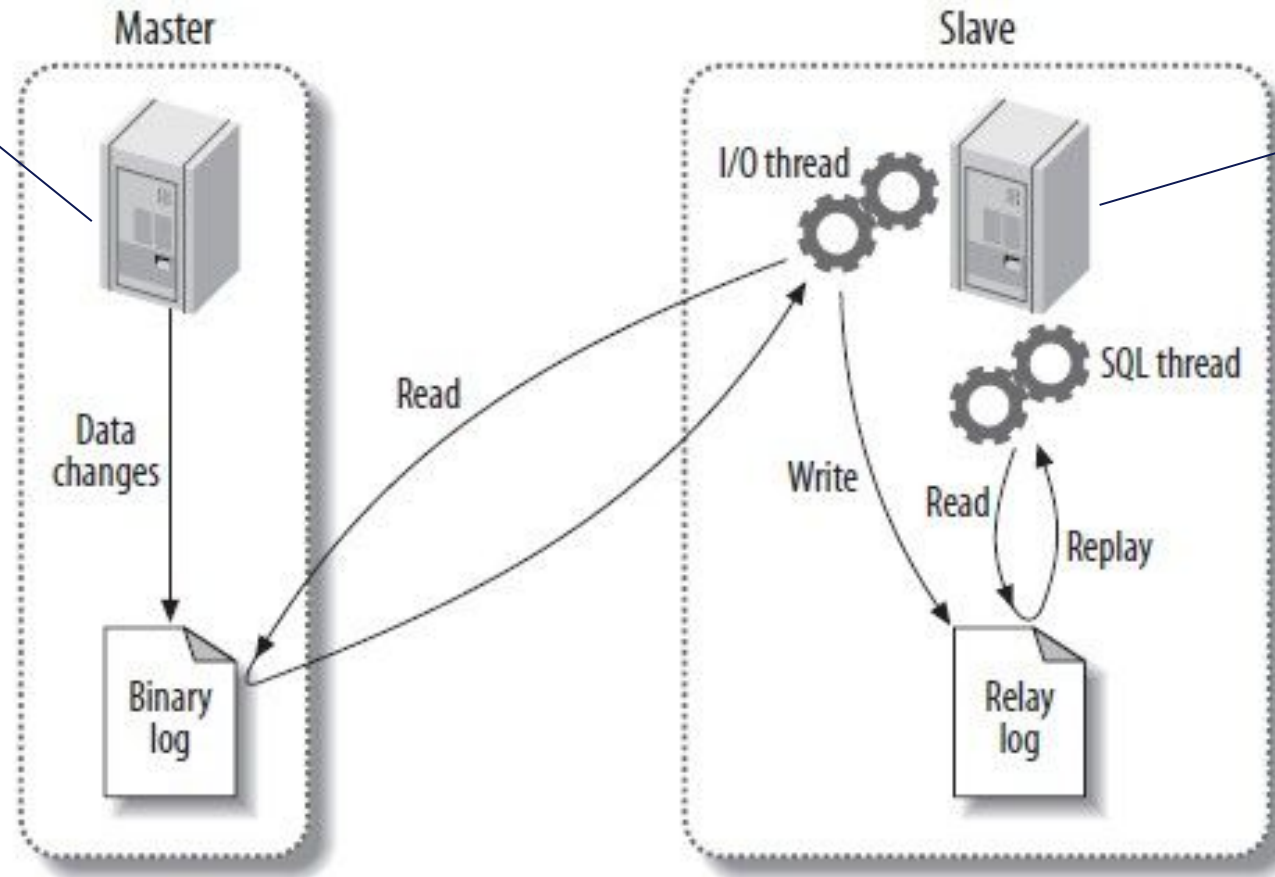
MySQL
Replication

<https://altmannmarcelo.medium.com/a-brief-history-of-mysql-replication-85f057922800>

Replicação assíncrona

numbers

ID	NAME	Time
----	------	------



numbers

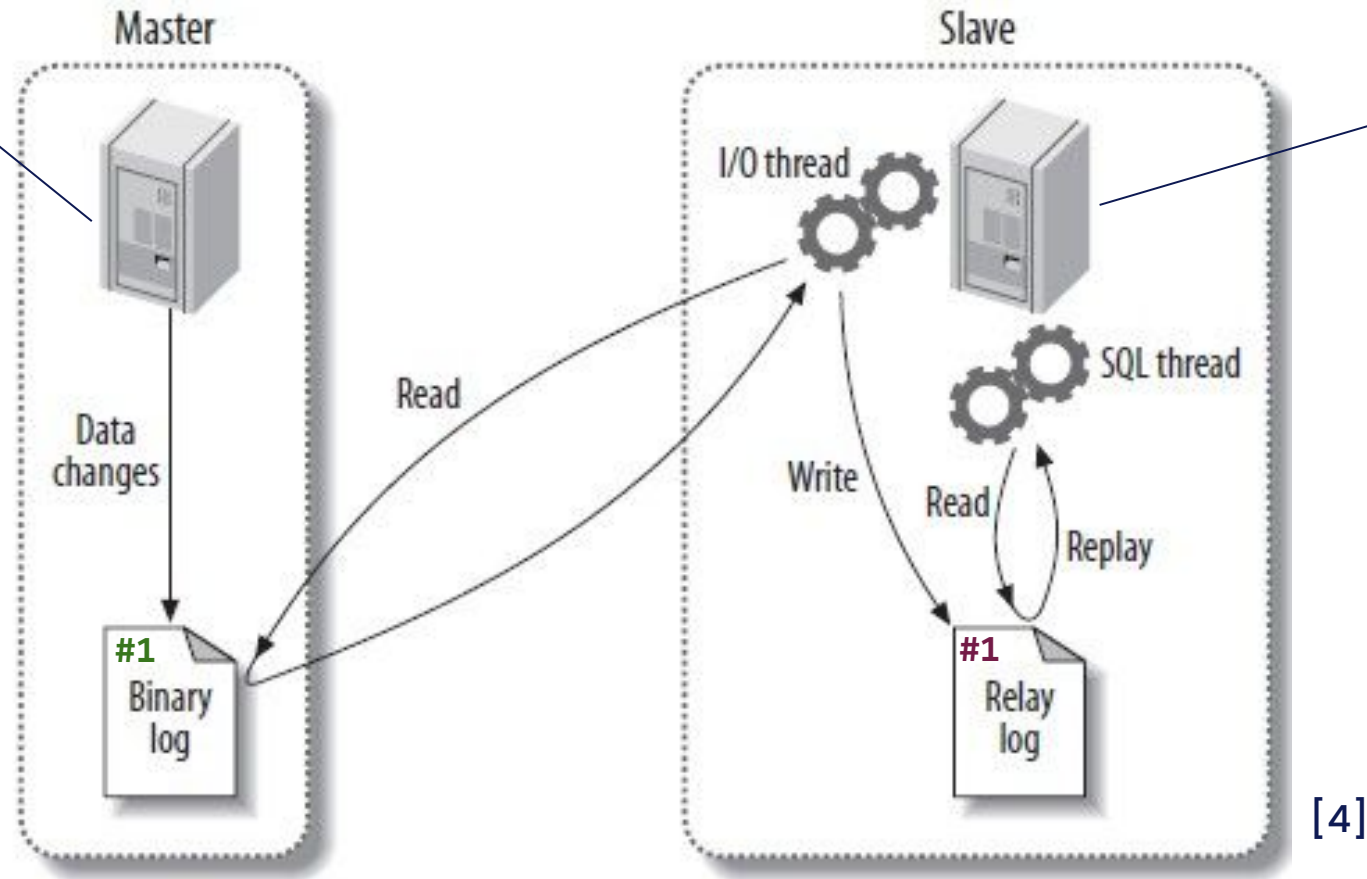
ID	NAME	Time
----	------	------

[4]

Replicação assíncrona

numbers

ID	NAME	Time
1	One	10:30



numbers

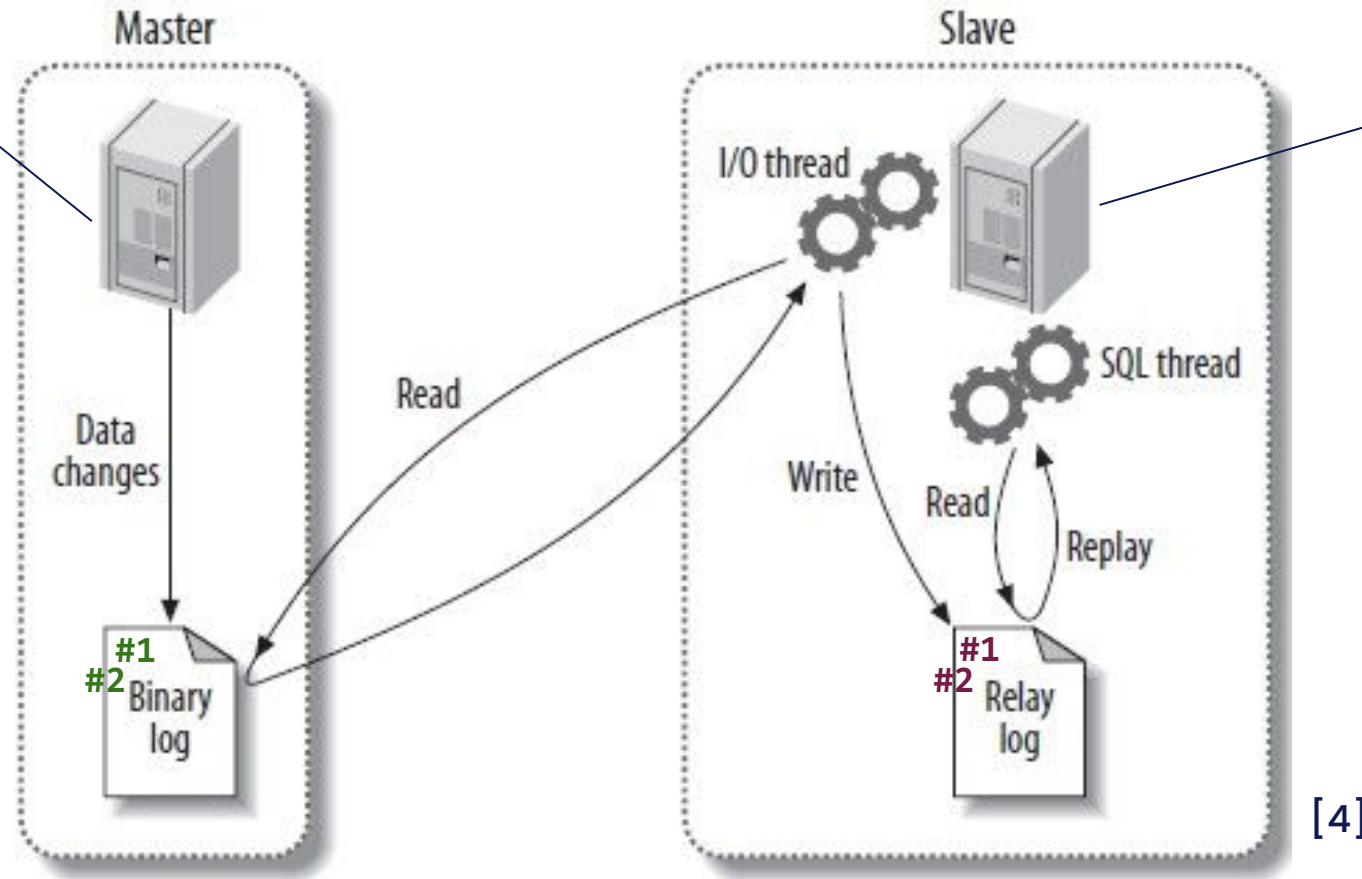
ID	NAME	Time
1	One	10:30



Replicação assíncrona

numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31



numbers

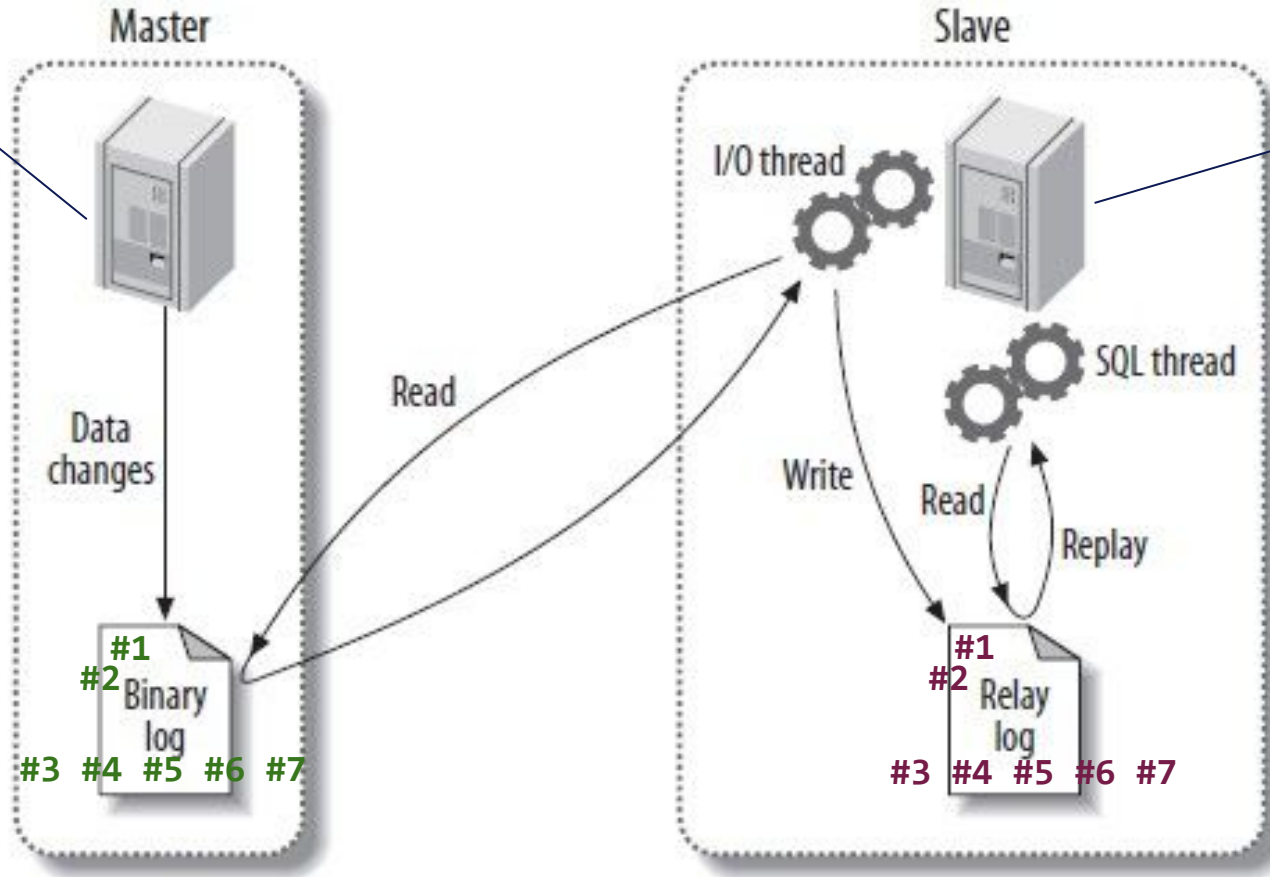
ID	NAME	Time
1	One	10:30
2	Two	10:31



Replicação assíncrona

numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32



numbers

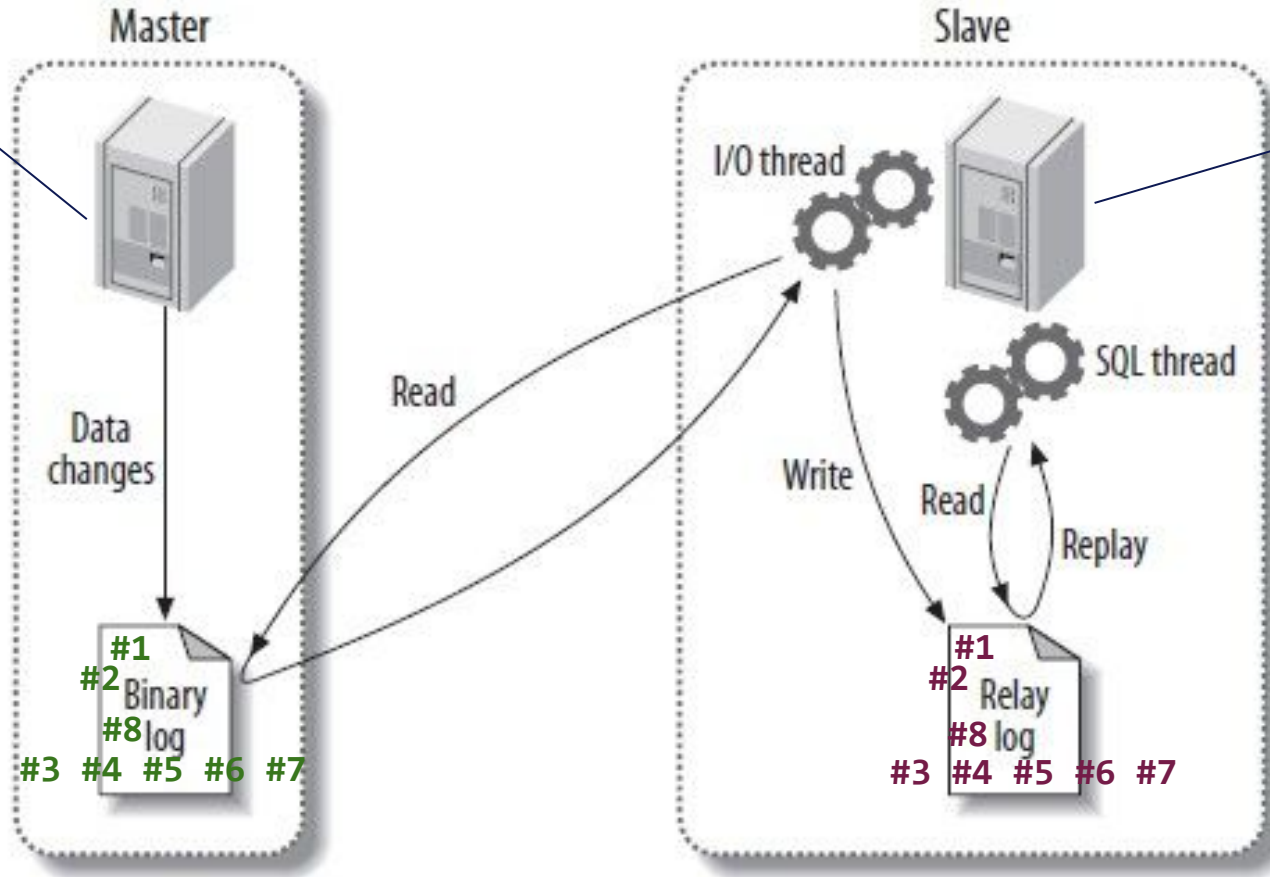
ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32



Replicação assíncrona

numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32
8	Eight	10:33



numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32
8	Eight	10:33

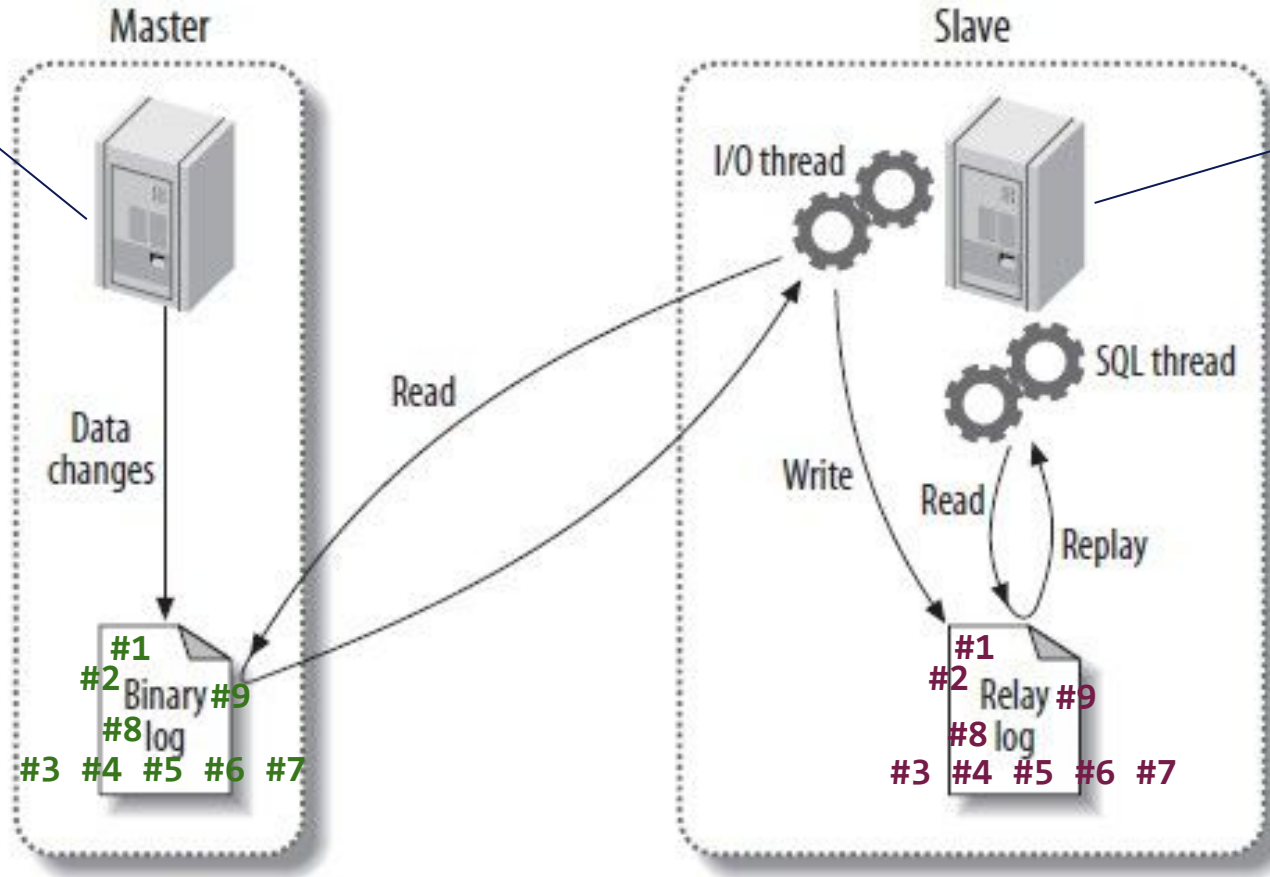
[4]



Replicação assíncrona

numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32
8	Eight	10:33
9	Nine	10:34

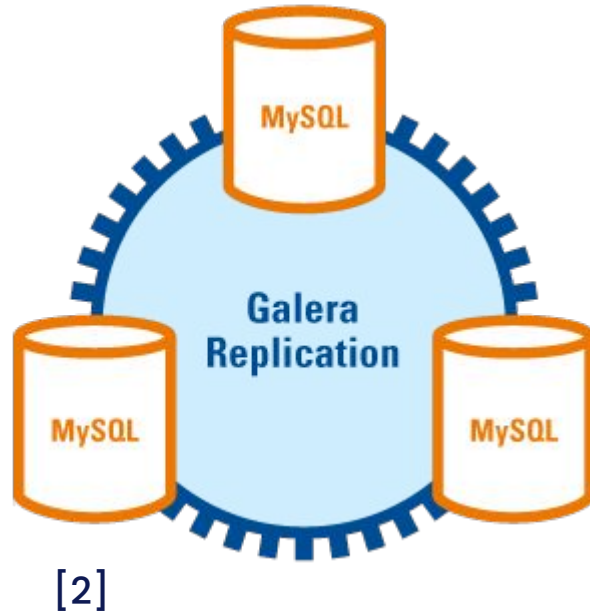


numbers

ID	NAME	Time
1	One	10:30
2	Two	10:31
3	Three	10:32
4	Four	10:32
5	Five	10:32
6	Six	10:32
7	Seven	10:32
8	Eight	10:33
9	Nine	10:34



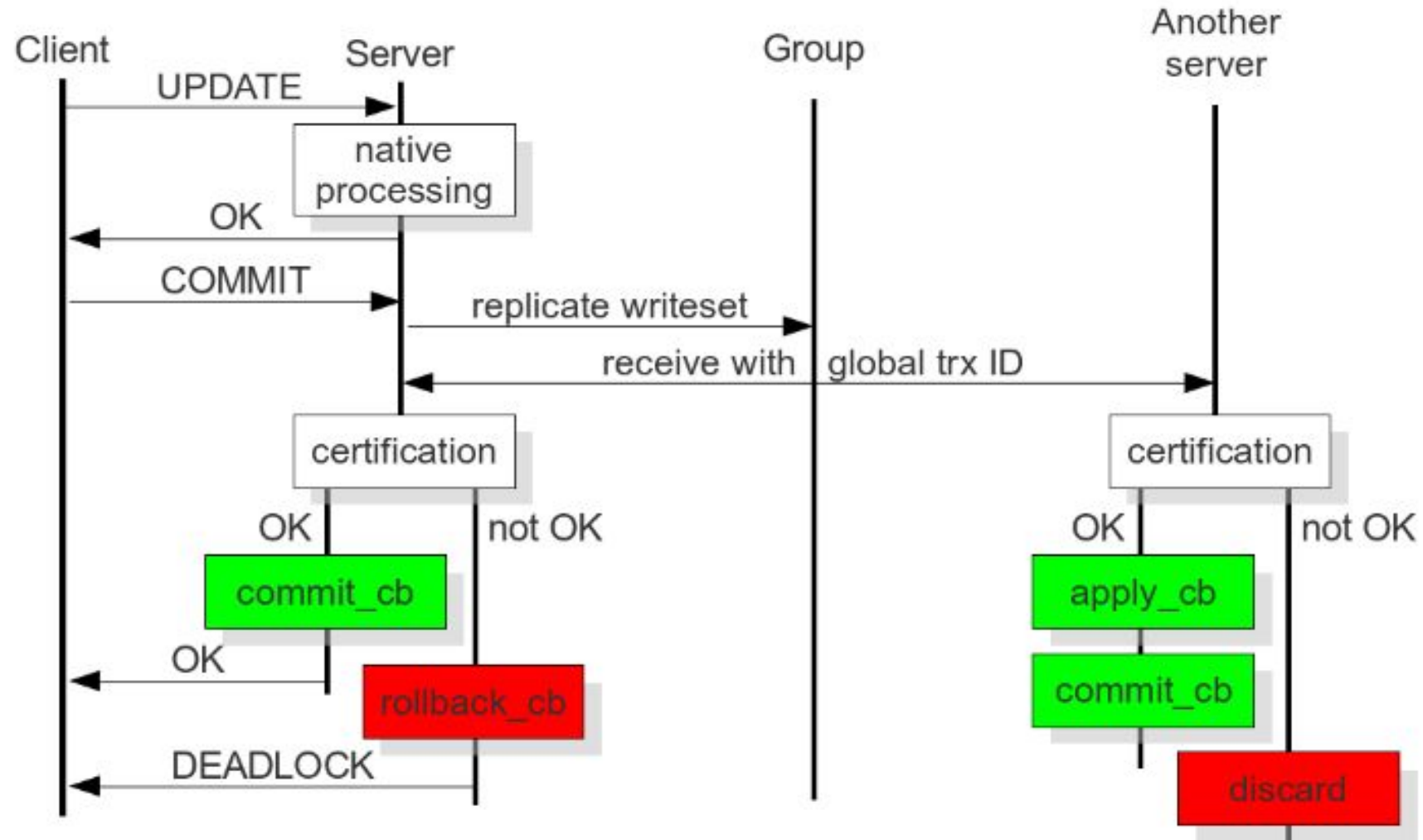
Replicação semi-síncrona



- Multi-master (active-active)
 - No failover required
- "Controlled" replication lag
- Automatic node provisioning



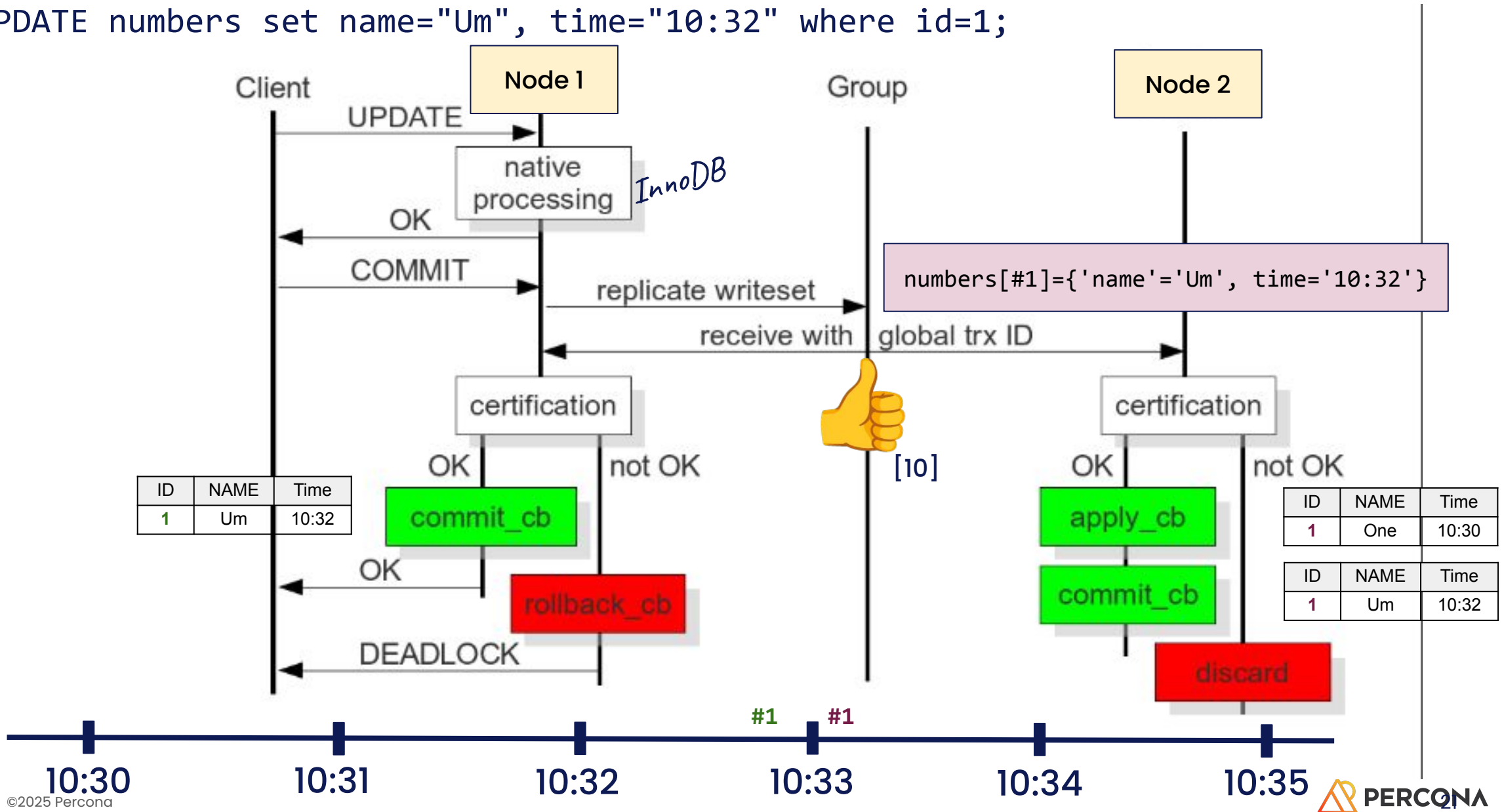
Replicação semi-síncrona



<https://mariadb.com/docs/galera-cluster/galera-architecture/certification-based-replication>

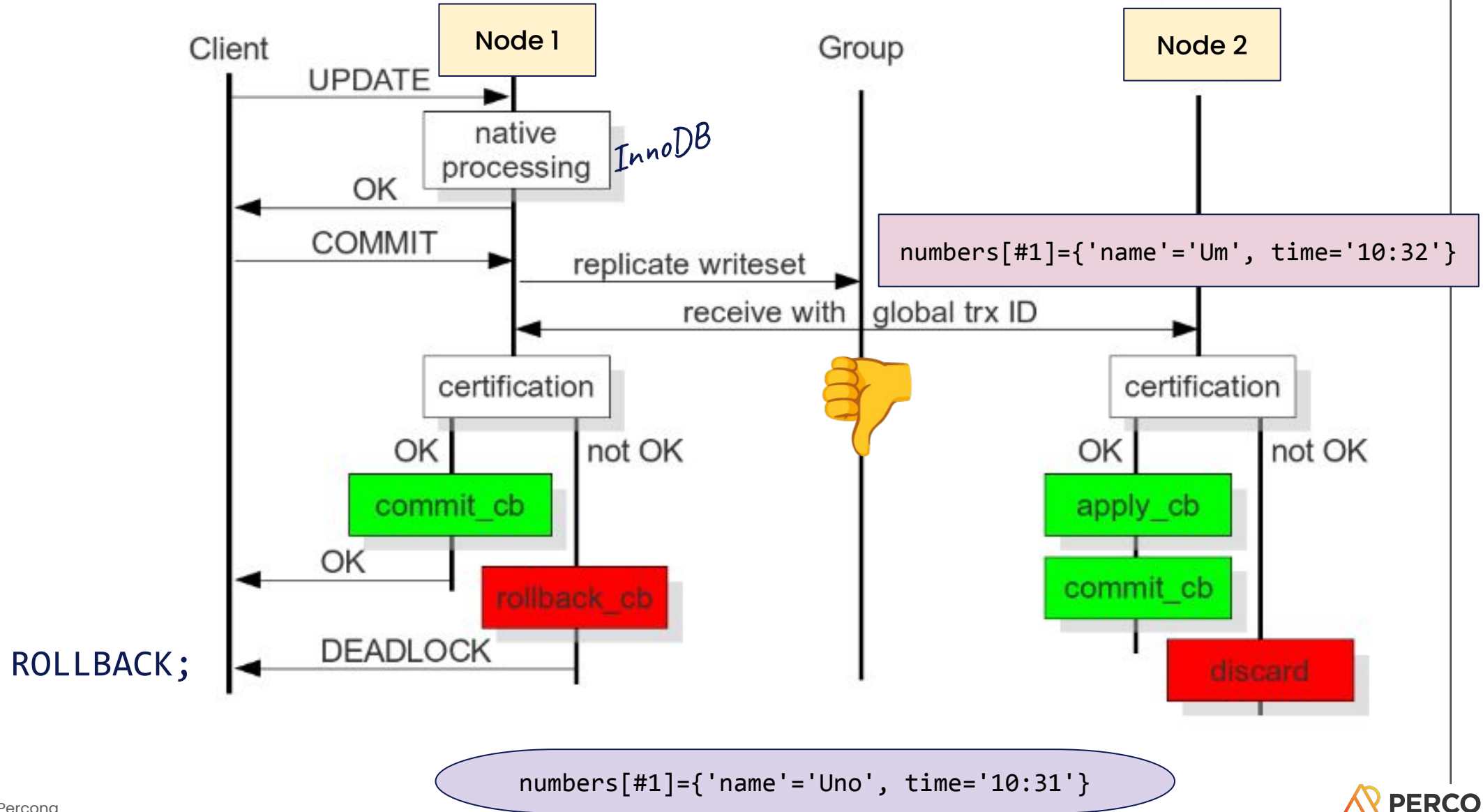
Replicação semi-síncrona

UPDATE numbers set name="Um", time="10:32" where id=1;

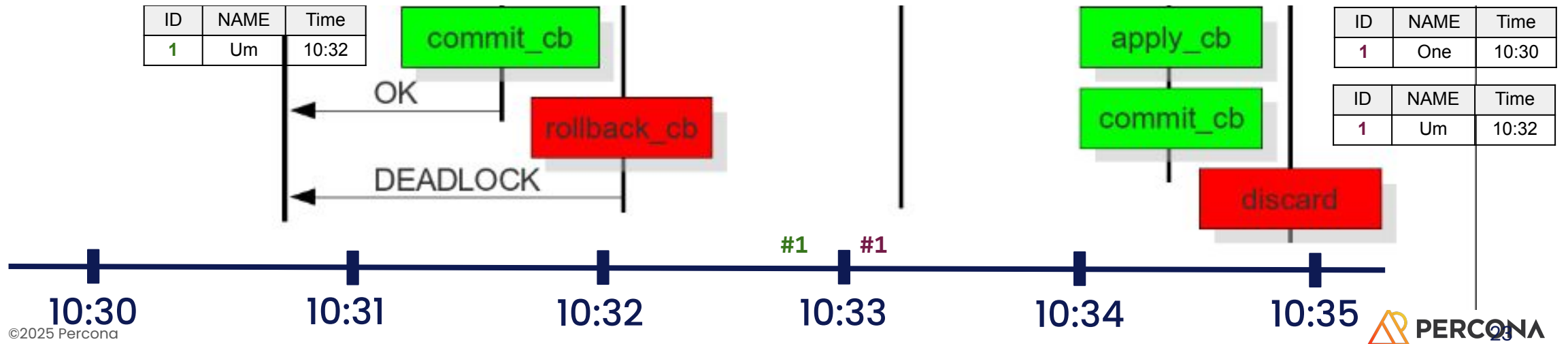


Replicação semi-síncrona

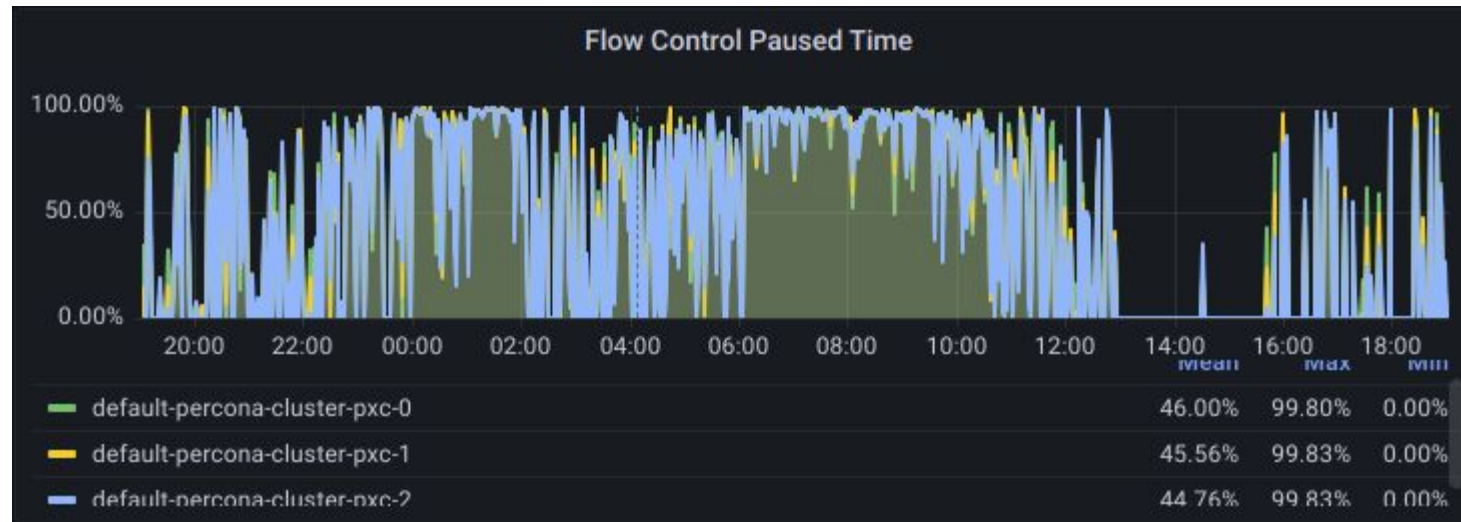
UPDATE numbers set name="Um", time="10:32" where id=1;



Flow Control



Flow Control



Flow Control



Galera cluster

- multi-master
- "Controlled" replication lag
- Automatic node provisioning
 - SST with XtraBackup
 - IST with Gcache
- Implemented through a special library/plugin:

```
wsrep_provider=/usr/lib/galera4/libgalera_smm.so
wsrep_cluster_name=pxc-cluster
wsrep_cluster_address=gcomm://192.168.70.61,
192.168.70.62,192.168.70.63
```

Group Replication

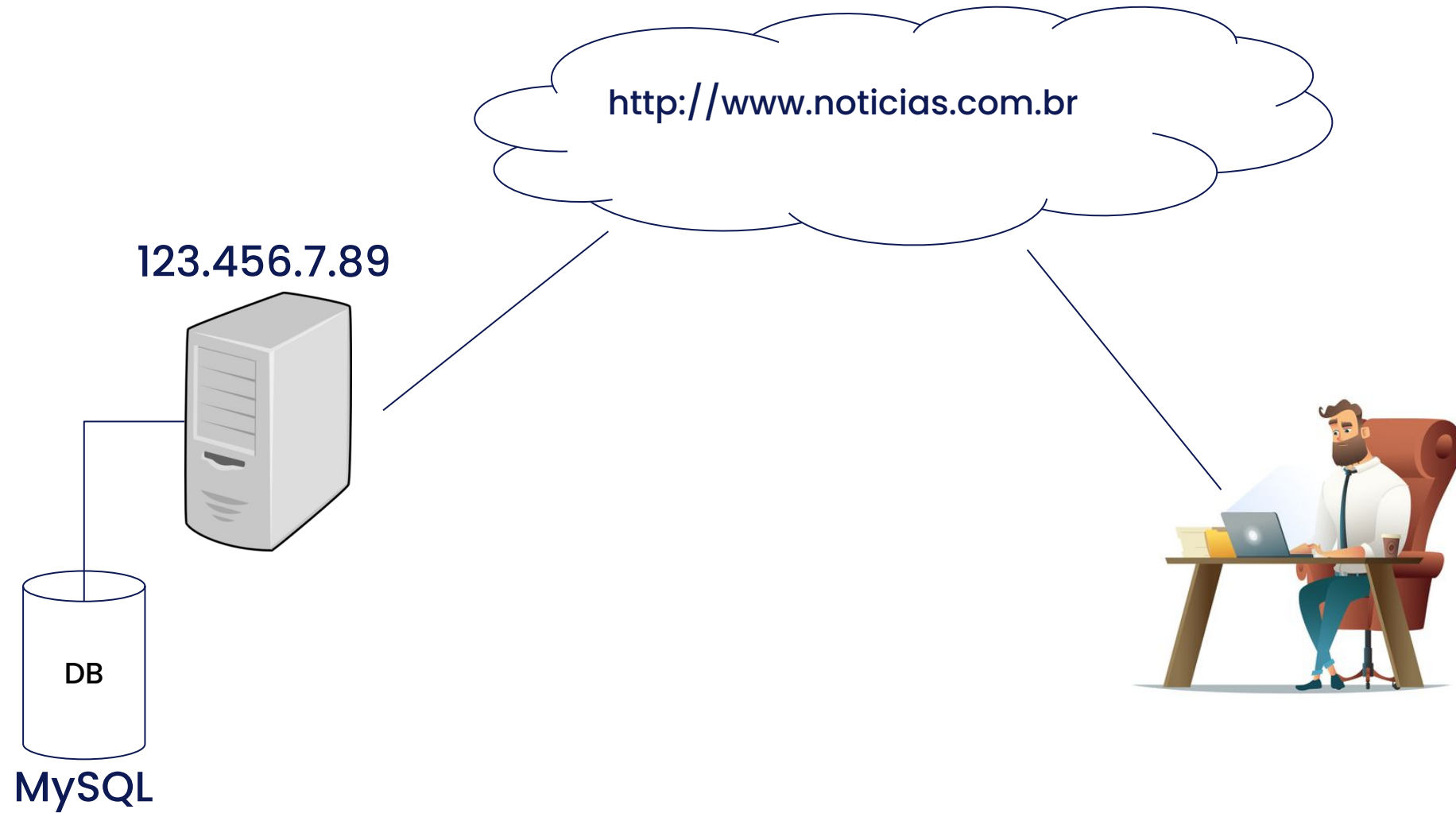
- single-master (by default)
- "Controlled" replication lag
- Automatic node provisioning
 - binary logs
 - clone plugin
- Implemented through a special library/plugin:

```
plugin_load_add='group_replication.so'
group_replication_group_name="25627f0f(...)"
group_replication_group_seeds=
"192.168.70.61:33061,192.168.70.62:33061,192.16
8.70.63:33061"
plugin_load_add='mysql_clone.so'
```

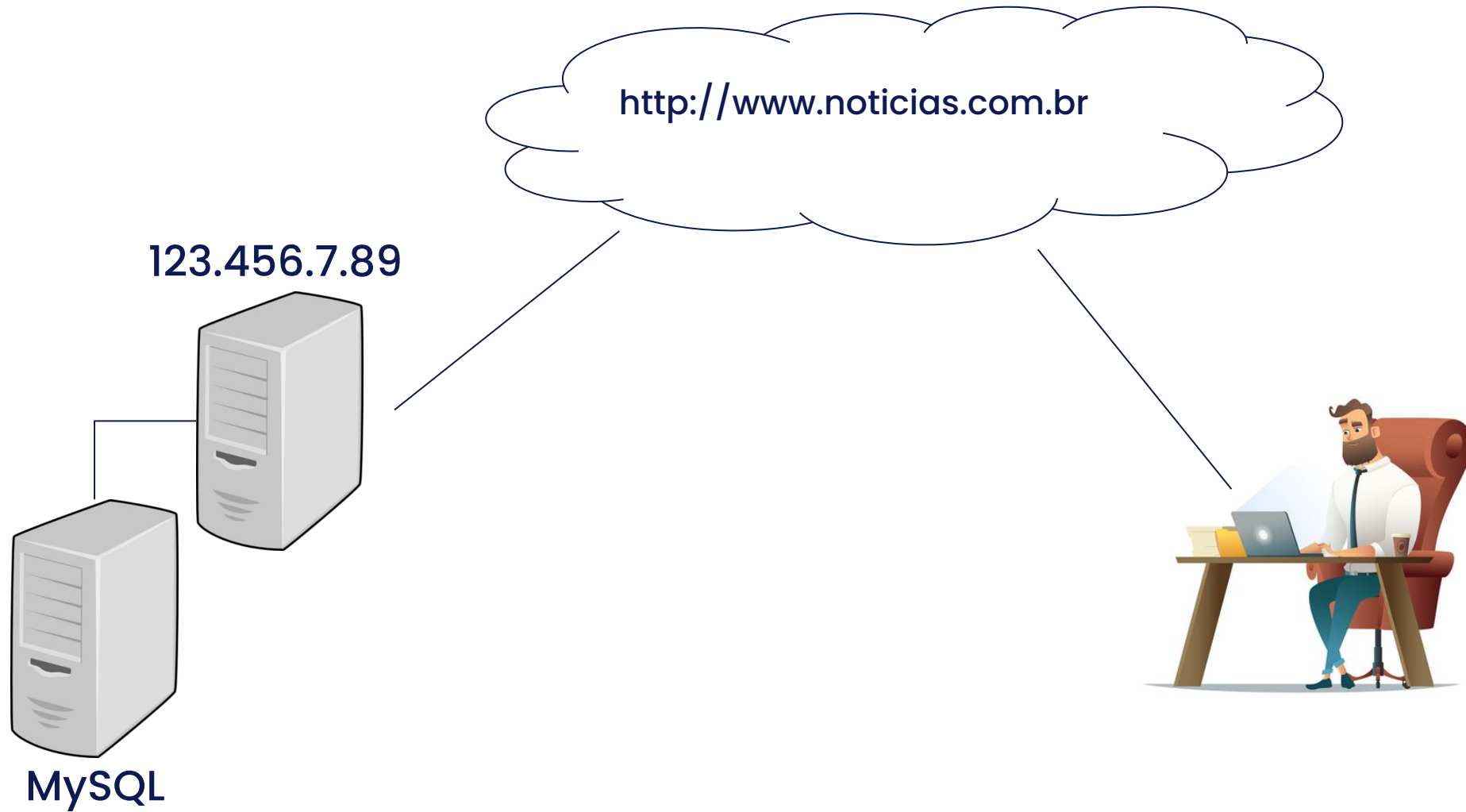


Alta disponibilidade

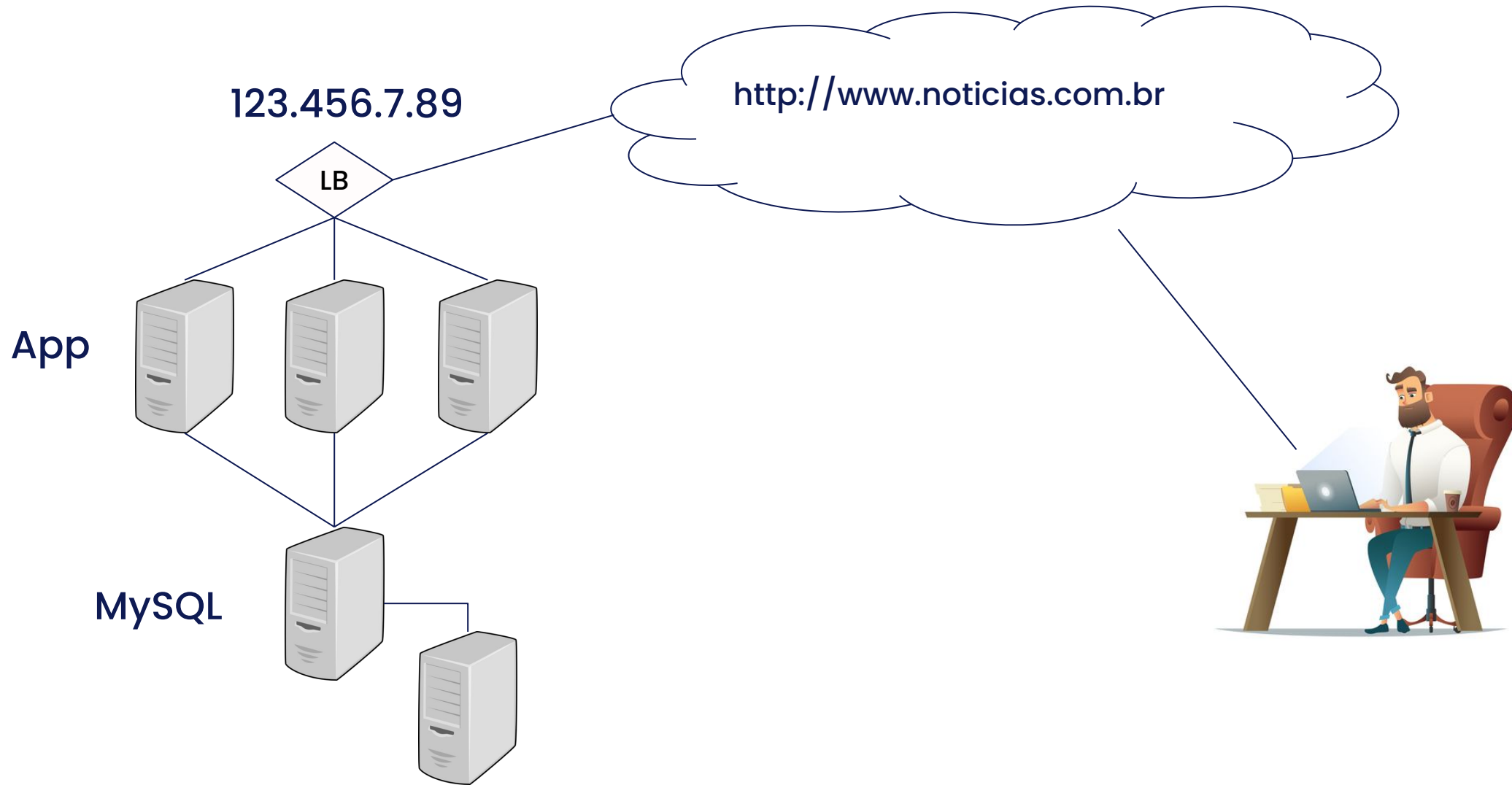
LAMP



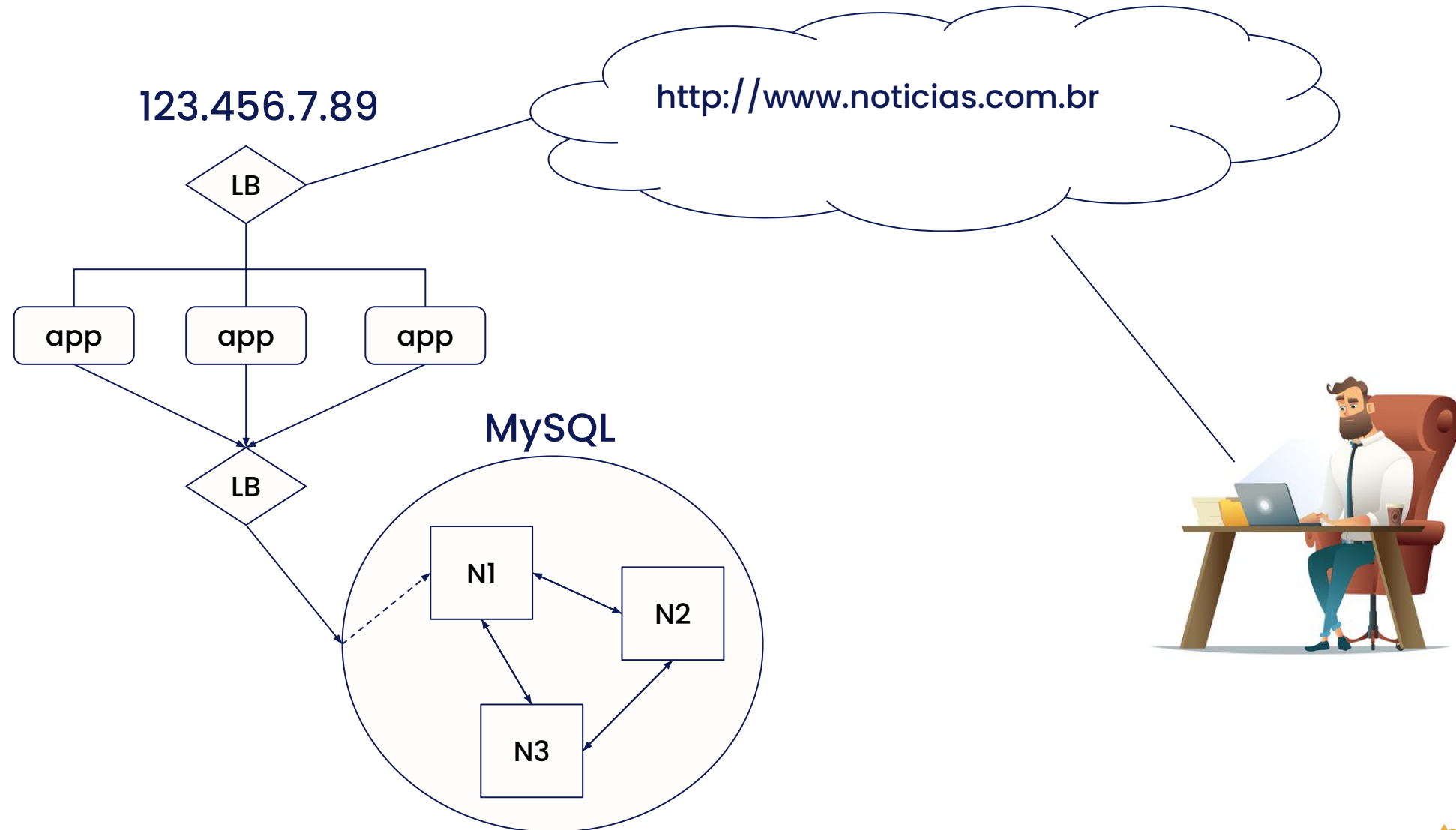
Decoupled



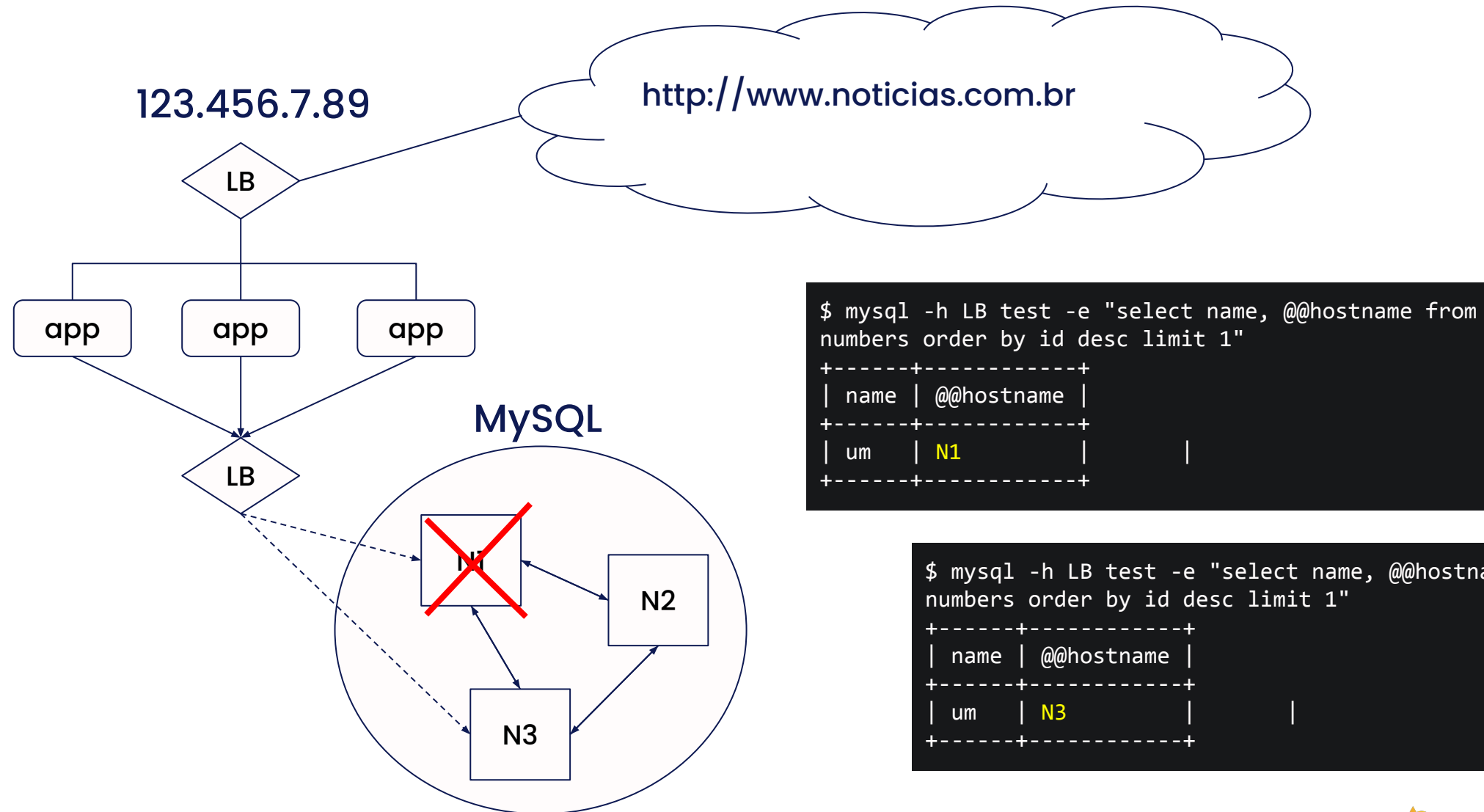
Alta disponibilidade



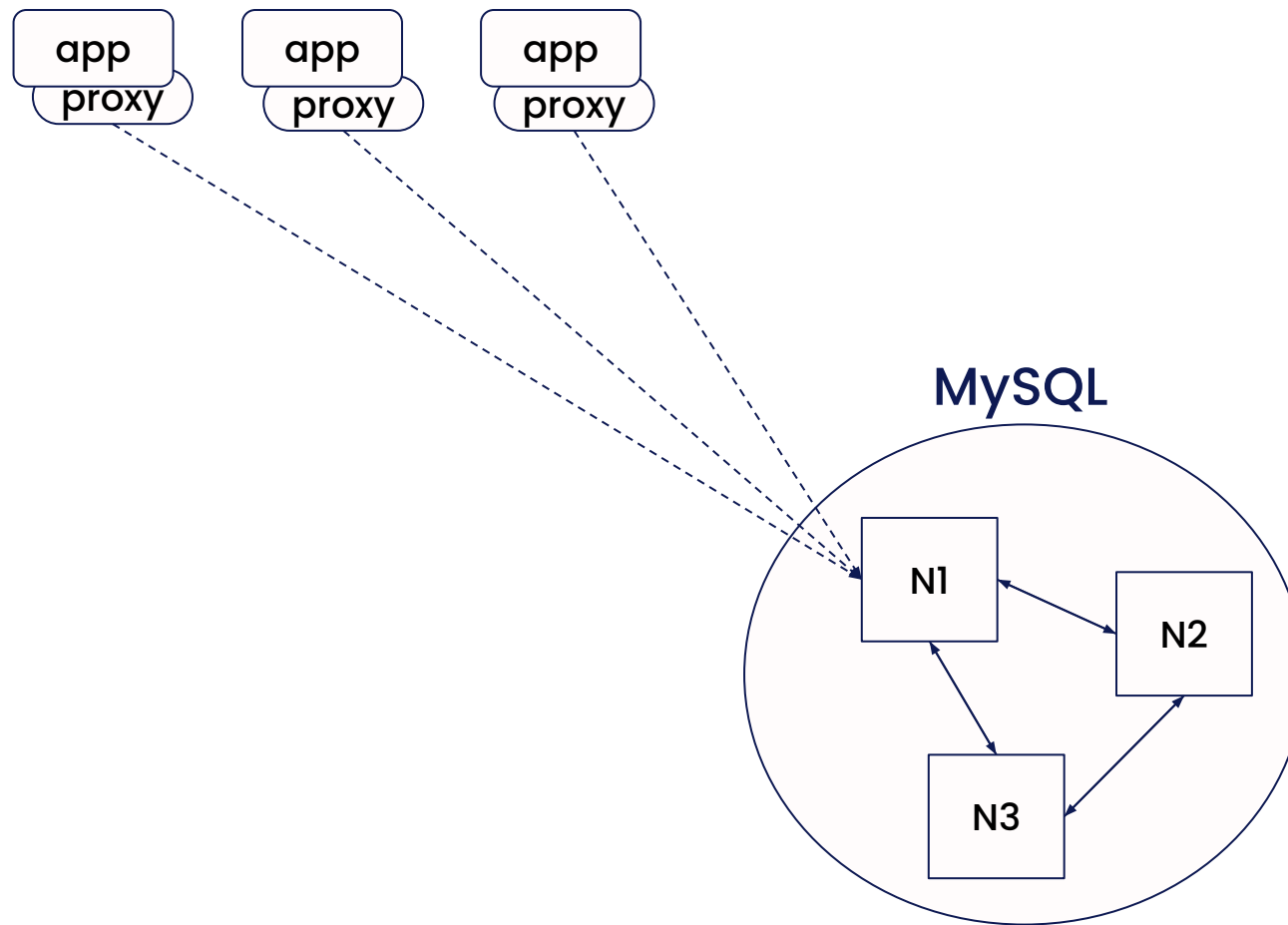
Alta disponibilidade



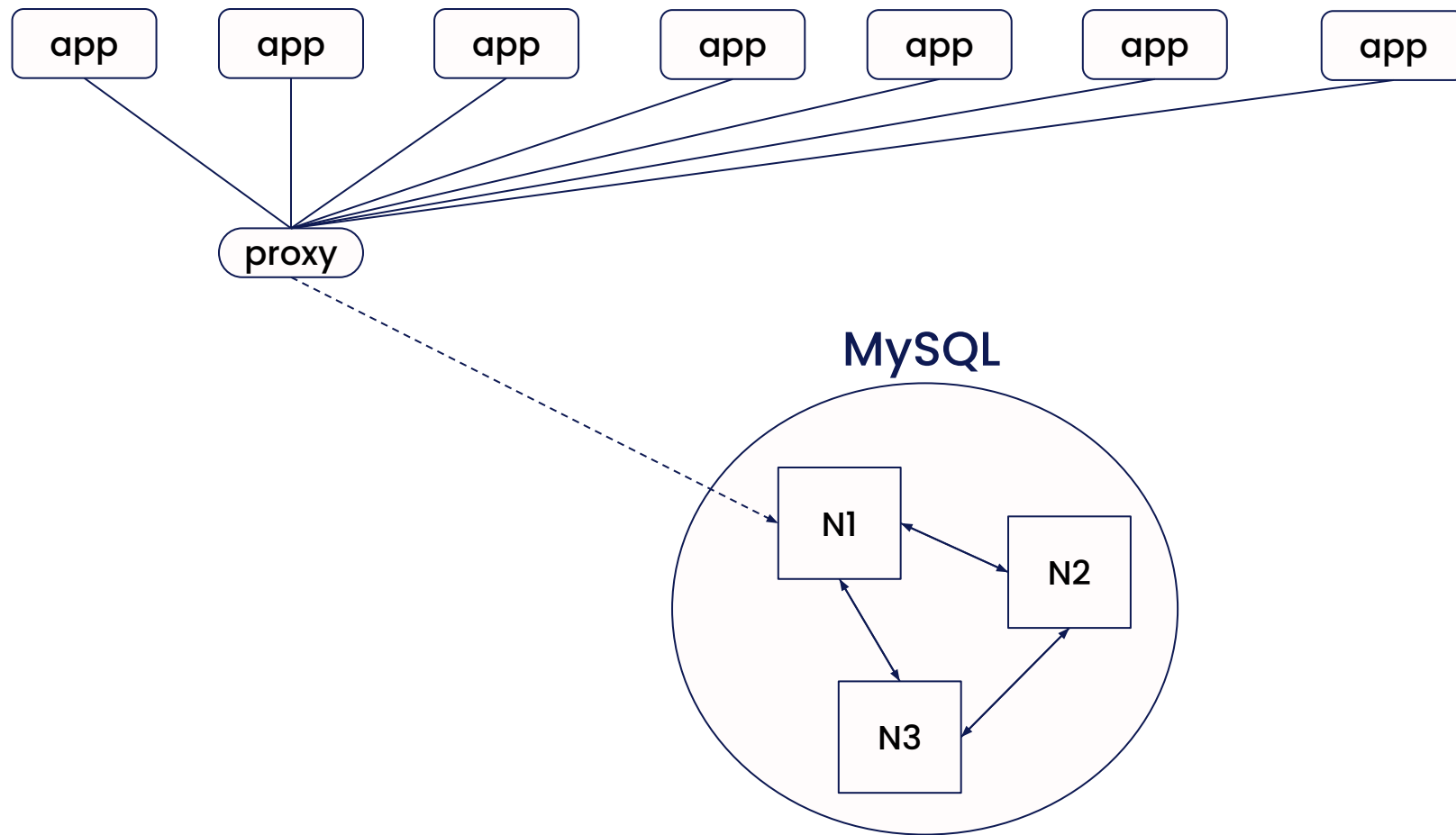
Alta disponibilidade



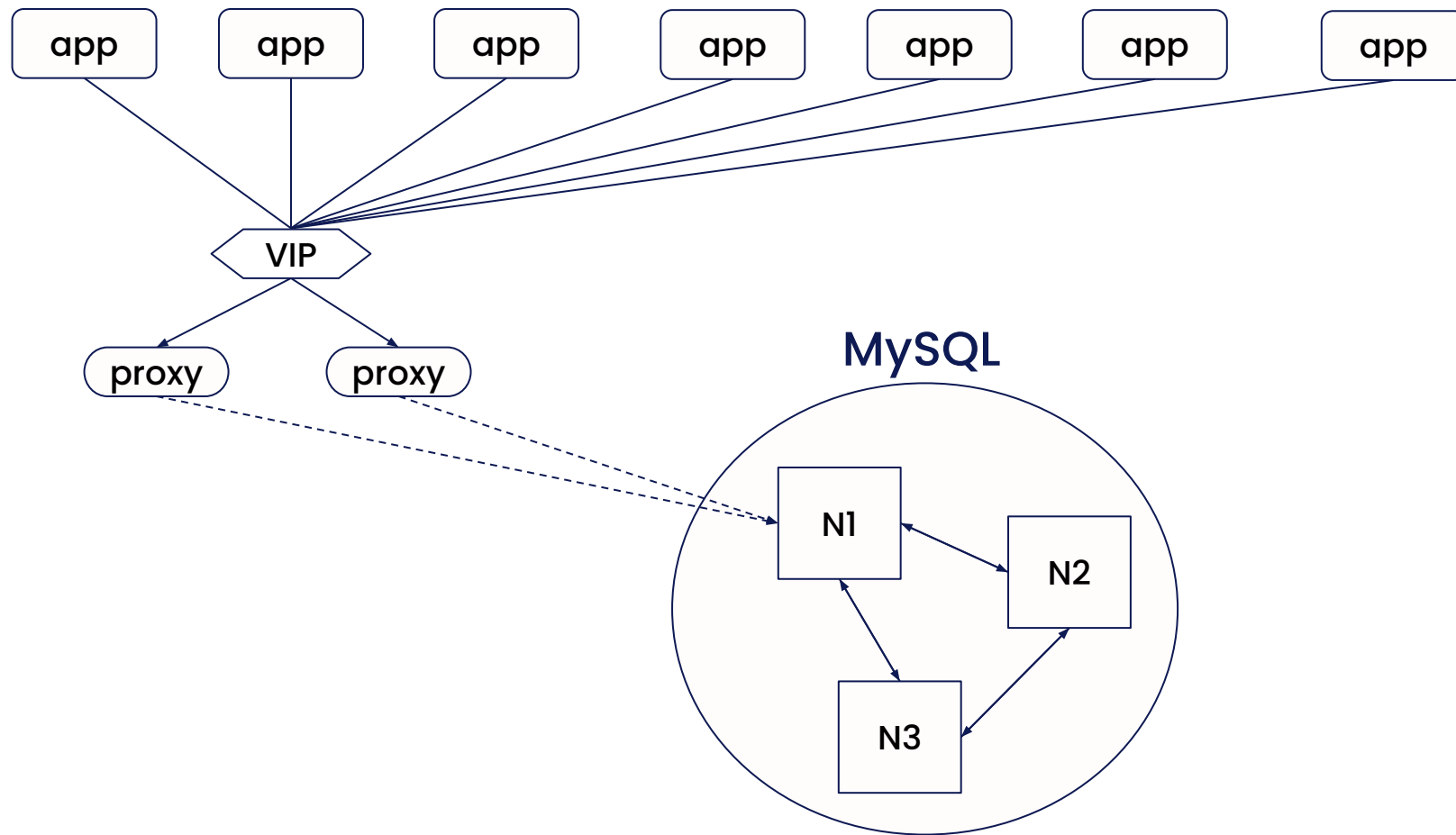
Alta disponibilidade



Alta disponibilidade



Alta disponibilidade



Alta disponibilidade

proxy

- HAProxy
- ProxySQL
- MySQL Router

HAProxy

/etc/haproxy/haproxy.cnf

```
(...)  
listen mycluster  
  bind 0.0.0.0:3307  
  mode tcp  
  balance roundrobin  
  option httpchk  
  
  server node1 192.168.0.1:3306 check port 9200 inter 12000 rise 3 fall 3  
  server node2 192.168.0.2:3306 check port 9200 inter 12000 rise 3 fall 3  
  server node3 192.168.0.3:3306 check port 9200 inter 12000 rise 3 fall 3
```

HTTP code:

- 200 → OK
- 500 → NOT OK



ProxySQL

runtime_mysql_galera_hostgroups

writer_hostgroup	backup_writer_hostgroup	reader_hostgroup	offline_hostgroup	active	max_writers	writer_is_also_reader	max_transactions_behind	comment
10	12	11	13	1	1	1	100	NULL

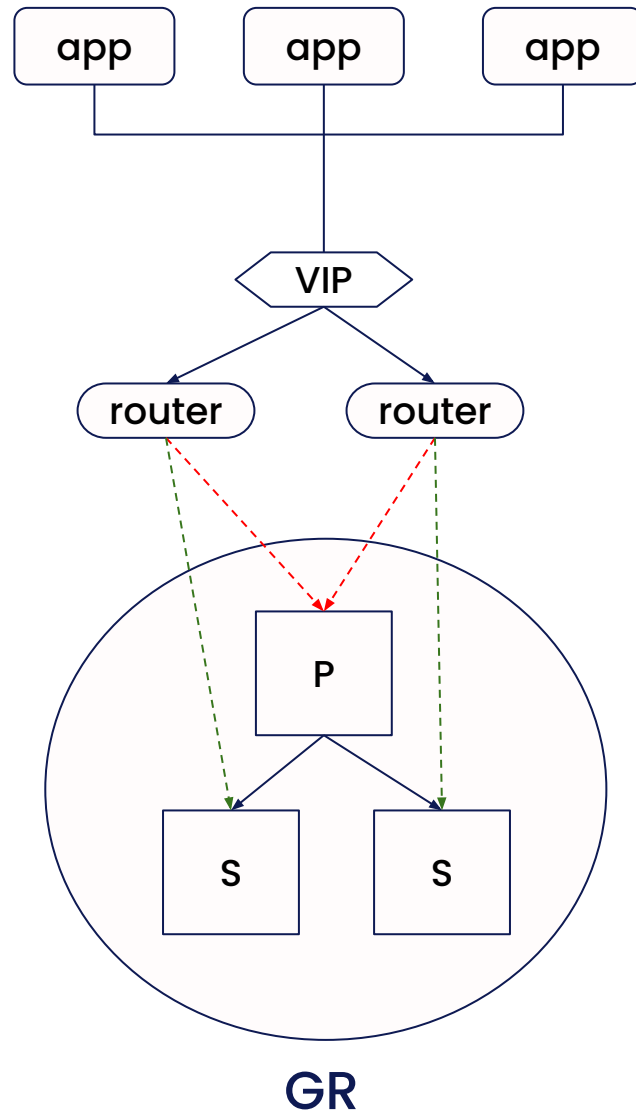
runtime_mysql_galera_hostgroups

hostgroup_id	hostname	port	gtid_port	status	weight
10	10.251.10.31	3306	0	SHUNNED	1000
10	10.251.10.32	3306	0	SHUNNED	1000
10	10.251.10.33	3306	0	ONLINE	1000
11	10.251.10.31	3306	0	ONLINE	1000
11	10.251.10.32	3306	0	ONLINE	1000
11	10.251.10.33	3306	0	ONLINE	1000
12	10.251.10.31	3306	0	ONLINE	1000
12	10.251.10.32	3306	0	ONLINE	1000

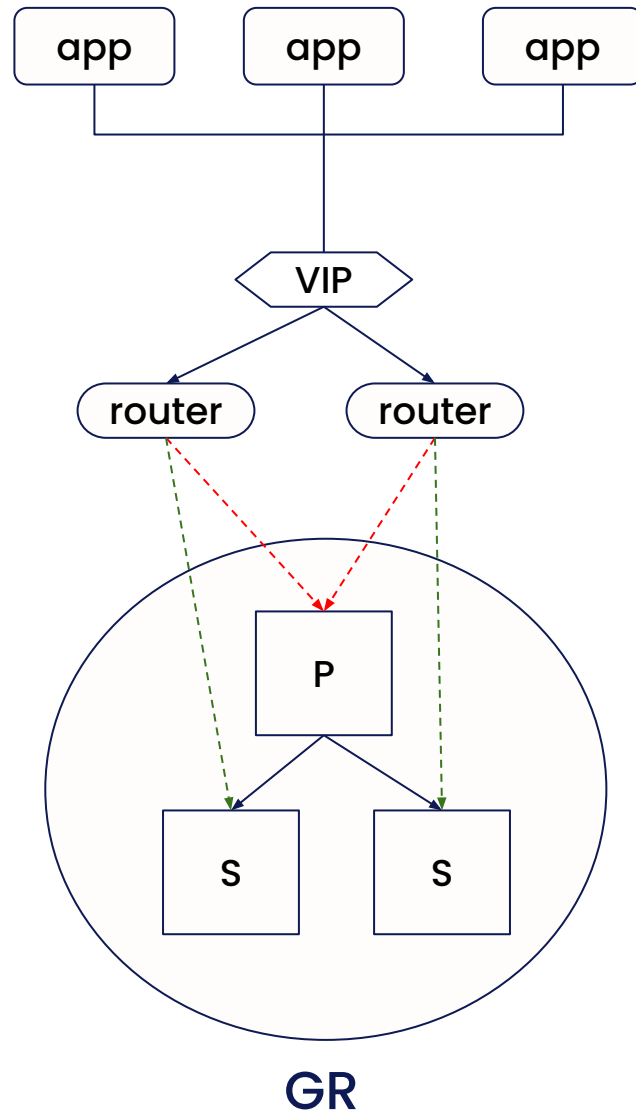
runtime_mysql_query_rules

***** 1. row *****
rule_id: 1
active: 1
match_digest: ^SELECT.*FOR UPDATE\$
destination_hostgroup: 10
(...)
***** 2. row *****
rule_id: 2
active: 1
match_digest: ^SELECT
destination_hostgroup: 11
(...)

MySQL Router



InnoDB Cluster



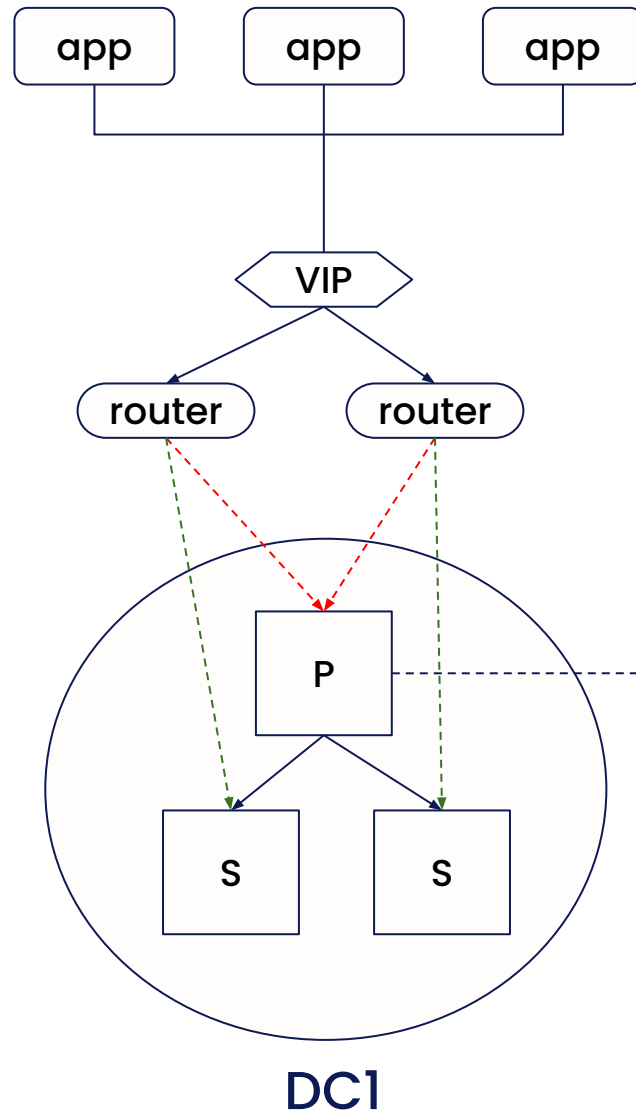
= Group Replication
+
MySQL Router
+
MySQL Shell

```
$ mysqlsh --js --socket=/var/run/mysqld/mysqld.sock --user=root  
mysqlsh> dba.configureInstance()  
mysqlsh> var cluster = dba.createCluster('mycluster')  
mysqlsh> cluster.addInstance('N2')  
mysqlsh> cluster.addInstance('N3')
```

```
$ mysqlrouter --bootstrap root@node1 --account router  
--account-host 192.168.0.% --account-create always --user router
```

```
mysqlsh> cluster.listRouters()
```


InnoDB Cluster :



ClusterSet

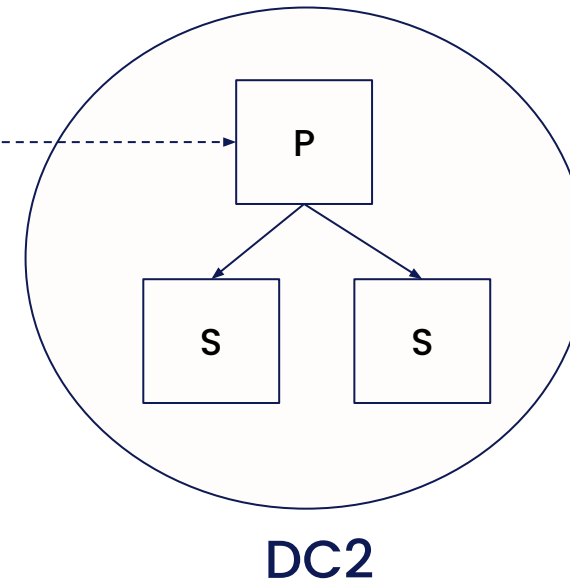
```
mysqlsh> cluster = dba.getCluster()
```

```
mysqlsh> clusterset = cluster.createClusterSet('myclusterset')
```

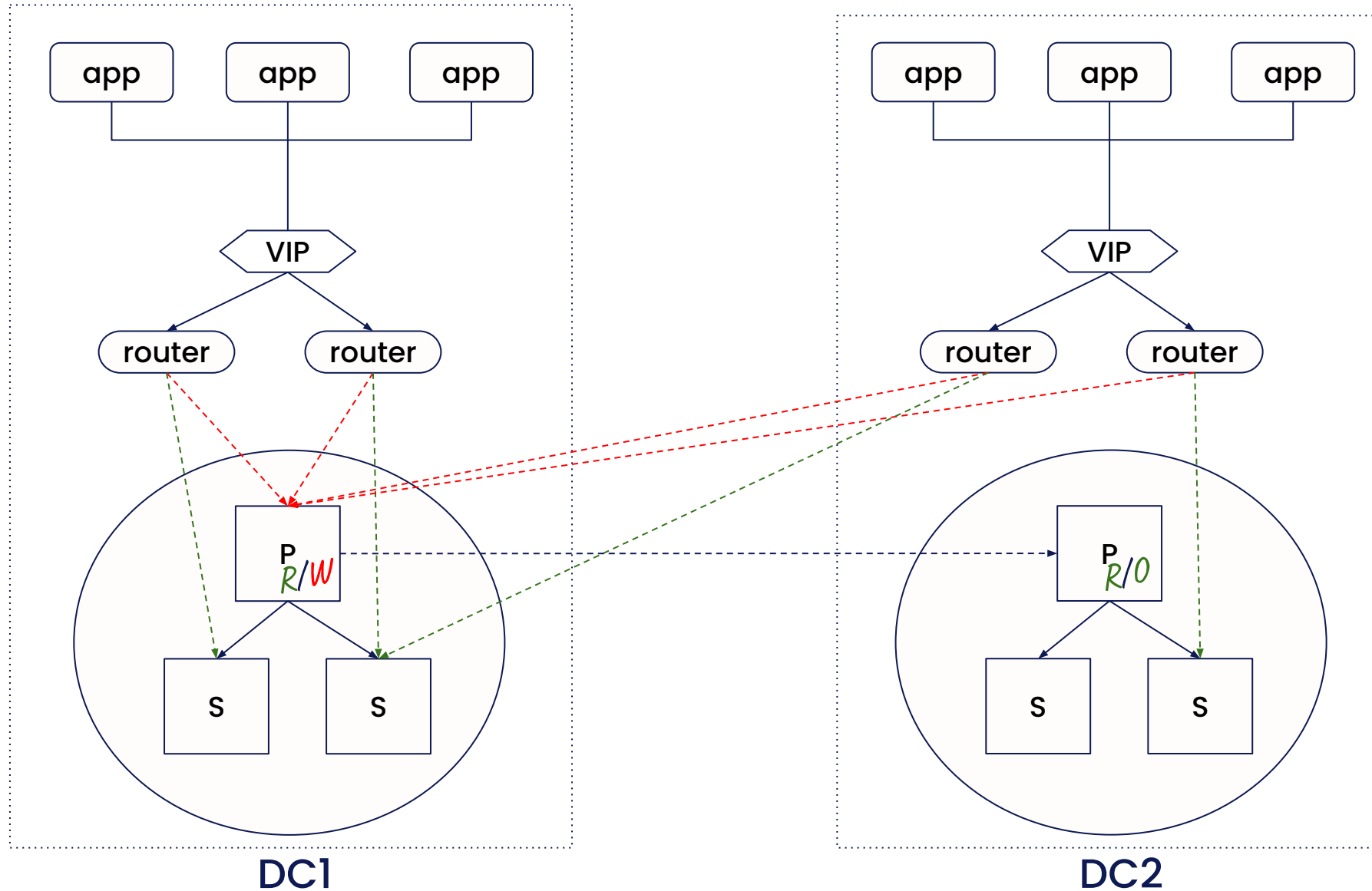
```
mysqlsh> cluster2 = clusterset.createReplicaCluster("N4", "DC2")
```

```
mysqlsh> cluster2.addInstance("N5")
```

```
mysqlsh> cluster2.addInstance("N6")
```



InnoDB Cluster



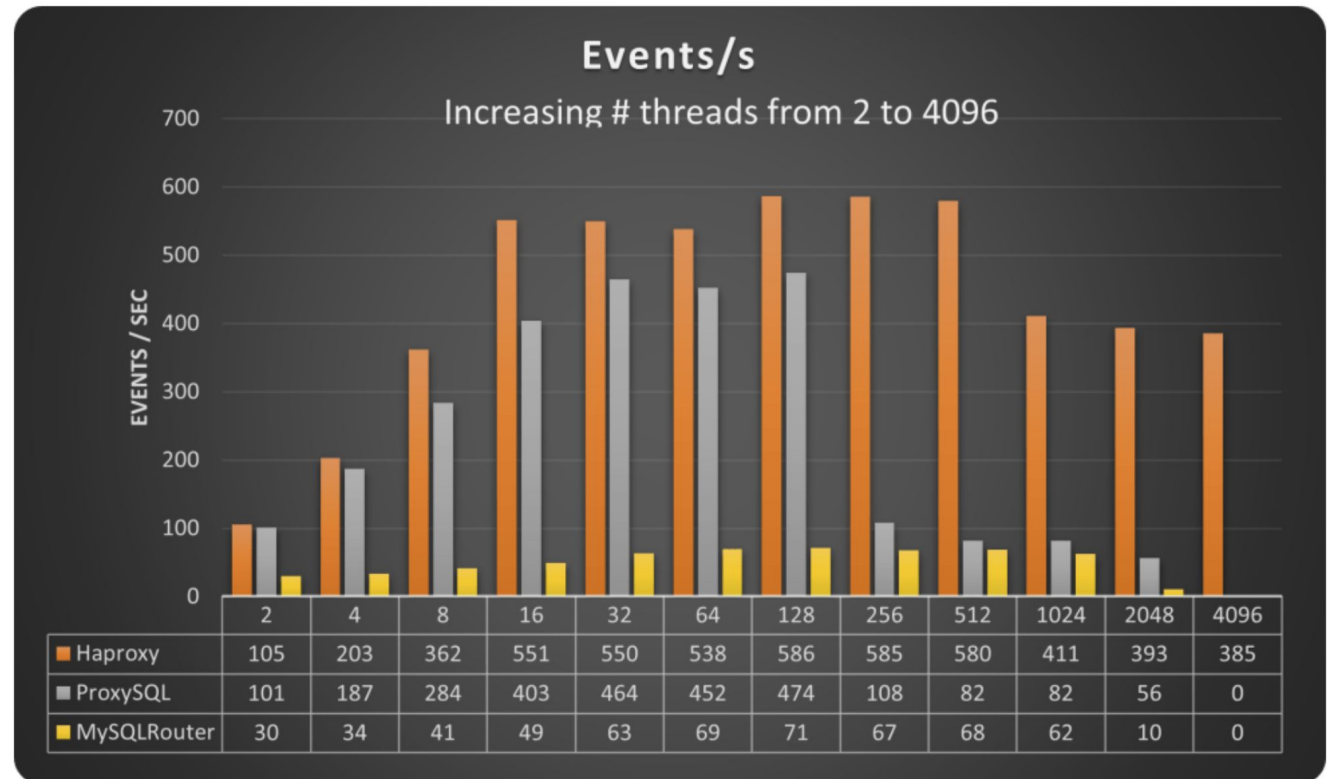
Comparisons of Proxies for MySQL

<https://www.percona.com/blog/comparisons-of-proxies-for-mysql/>

March 20, 2023

Marco Tusa

```
$ sysbench
./src/lua/windmills/oltp_read.lua
--mysql-host=<host> --mysql-port=<port>
--mysql-user=<user>
--mysql-password=<pw> --mysql-db=<schema>
--db-driver=mysql --tables=200
--table_size=1000000 --rand-type=zipfian
--rand-zipfian-exp=0 --skip_trx=true
--report-interval=1
--mysql-ignore-errors=all
--mysql_storage_engine=innodb
--auto_inc=off --histogram
--table_name=<tablename>
--stats_format=csv --db-ps-mode=disable
--point-selects=50
--reconnect=10 --range-selects=true
--threads=<#Threads from 2 to 4096>
--time=1200 run
```



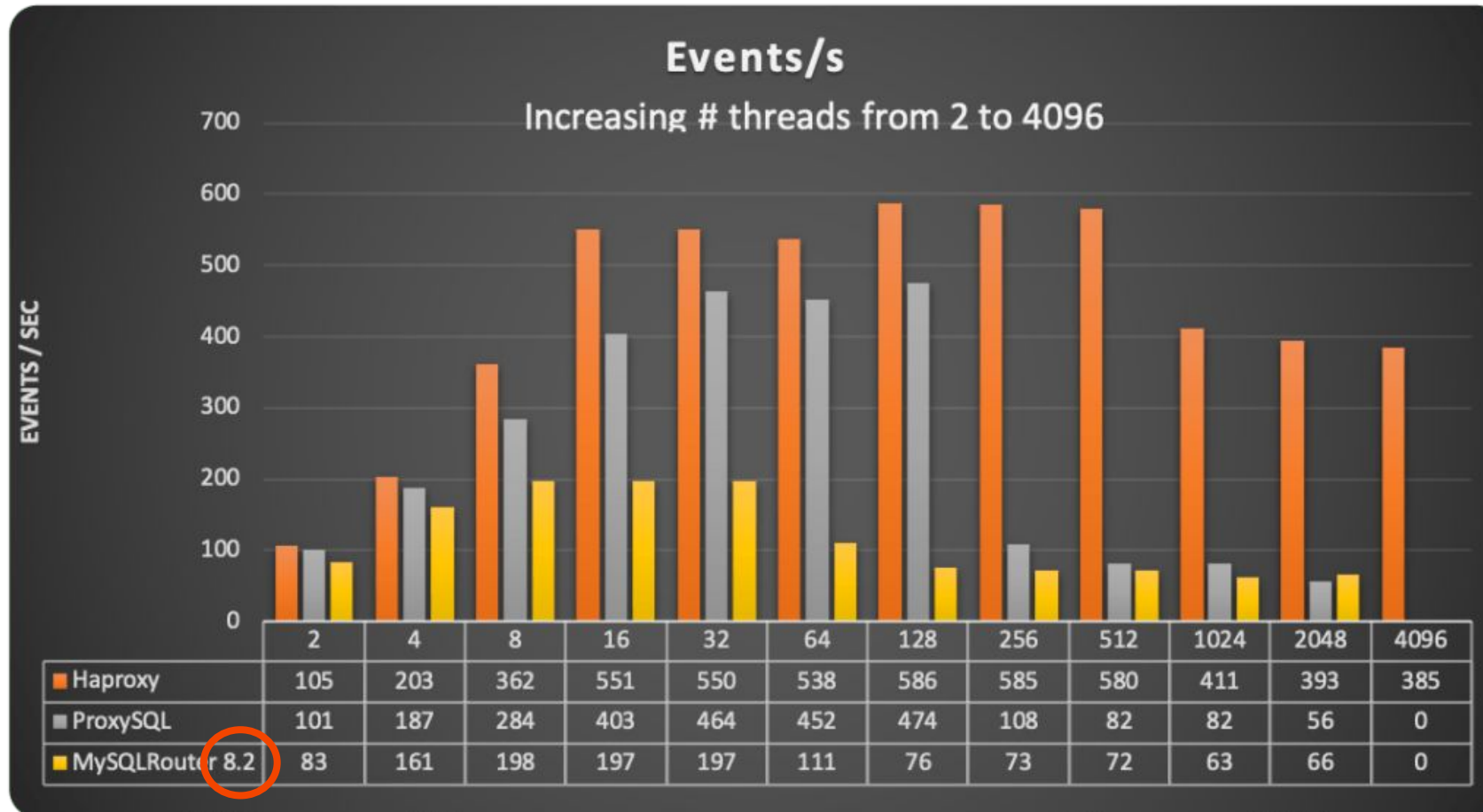
The scenario with load changes drastically. We can see how HAProxy can serve the connection and allow the execution of more operations for the whole test. ProxySQL is immediately after it and behaves quite well, up to 128 threads, then it just collapses.

Is MySQL Router 8.2 Any Better?

<https://www.percona.com/blog/is-mysql-router-8-2-any-better/>

January 11, 2024

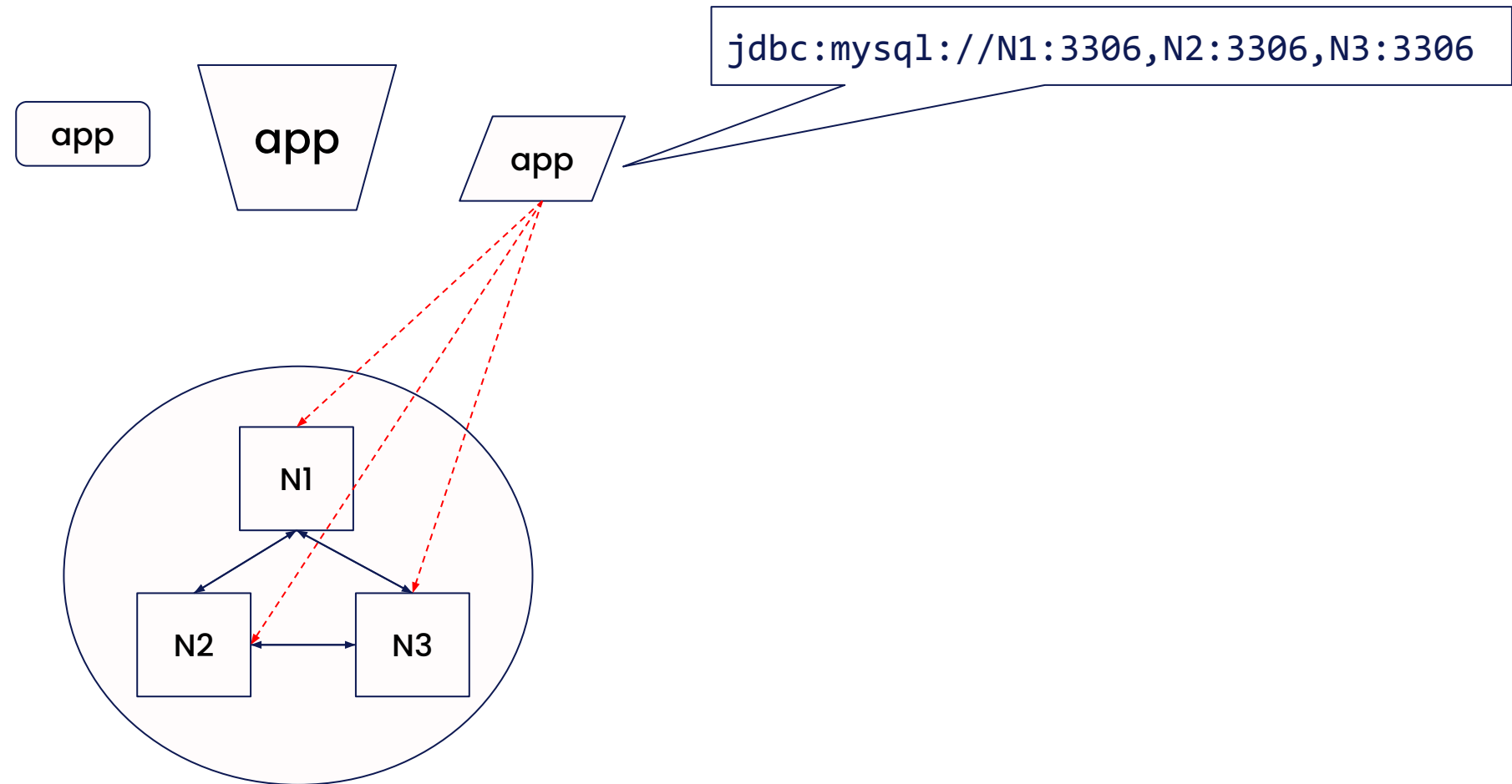
Marco Tusa



(...) we can answer a small (very small) yes

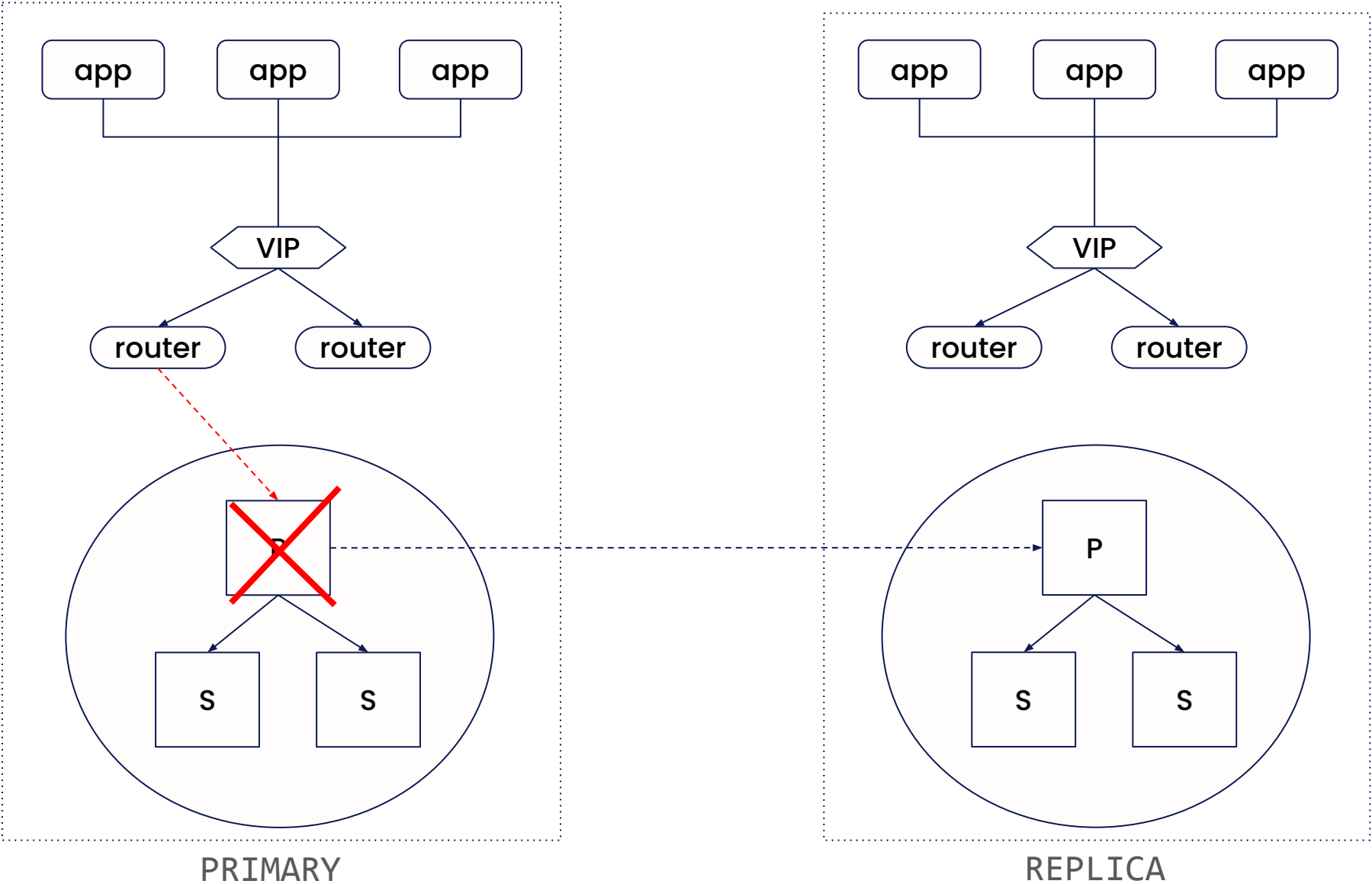
Exemplo 1

PXC

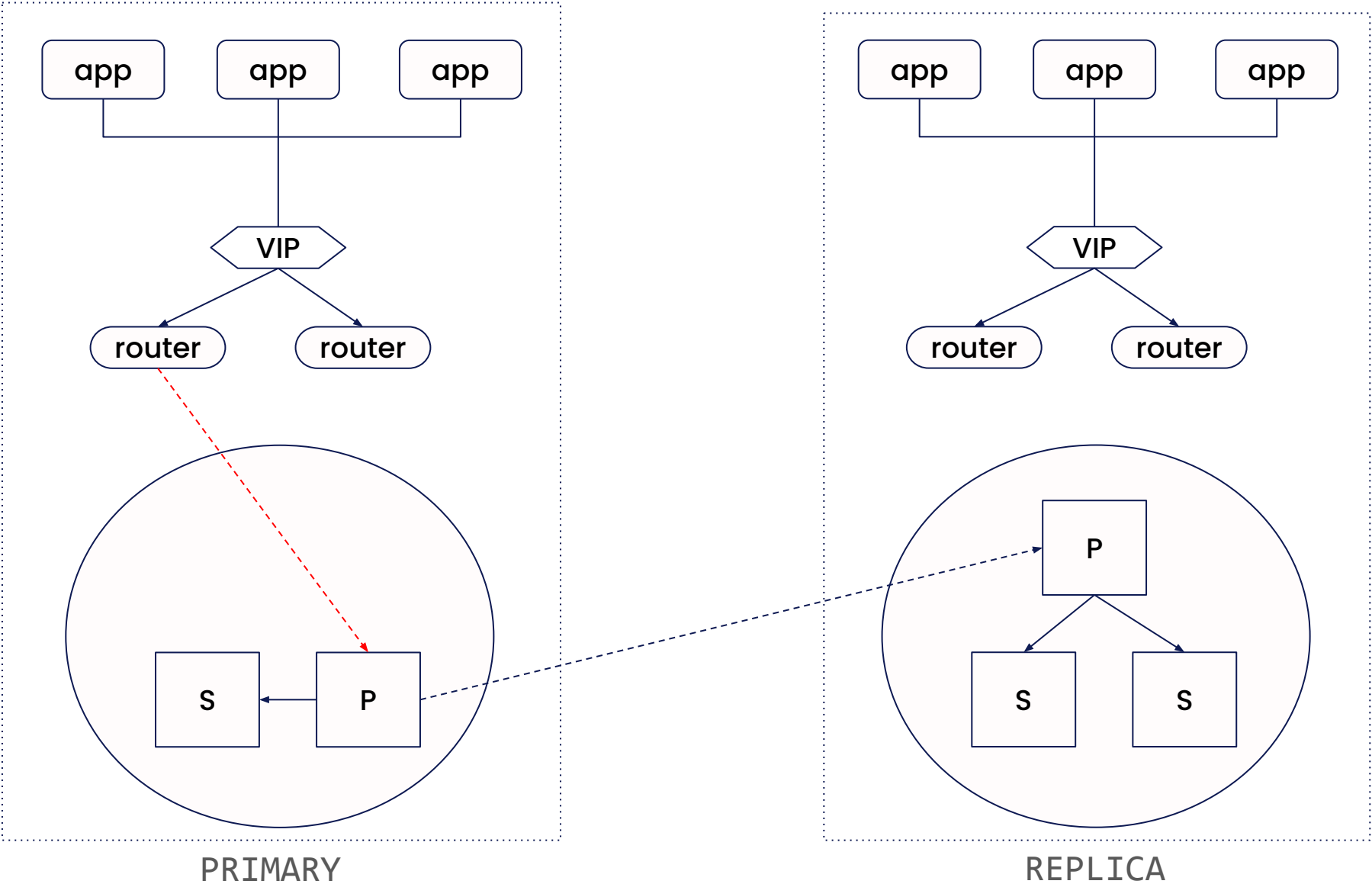


<https://dev.mysql.com/doc/connector-j/en/connector-j-config-failover.html>

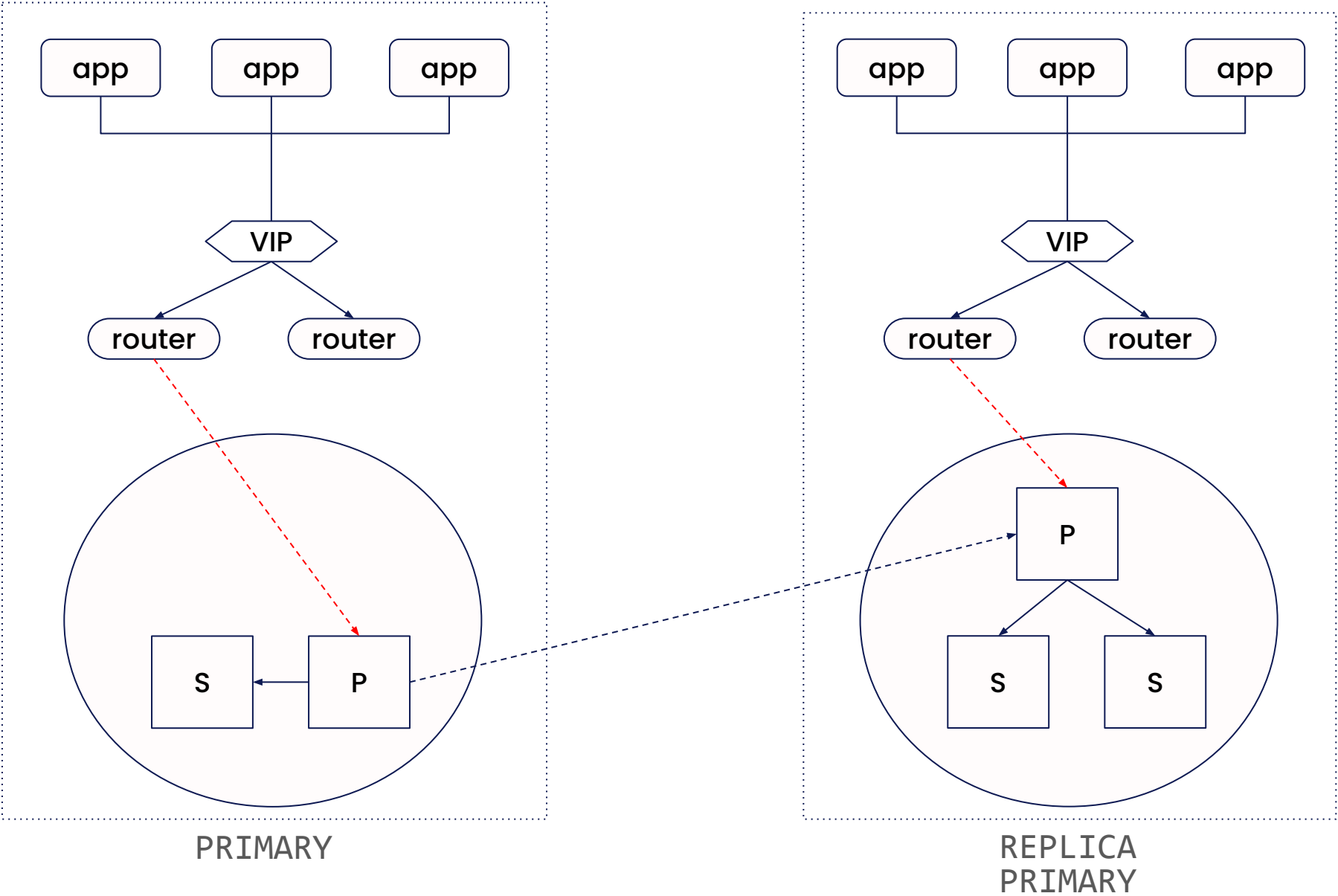
Exemplo 2



Exemplo 2

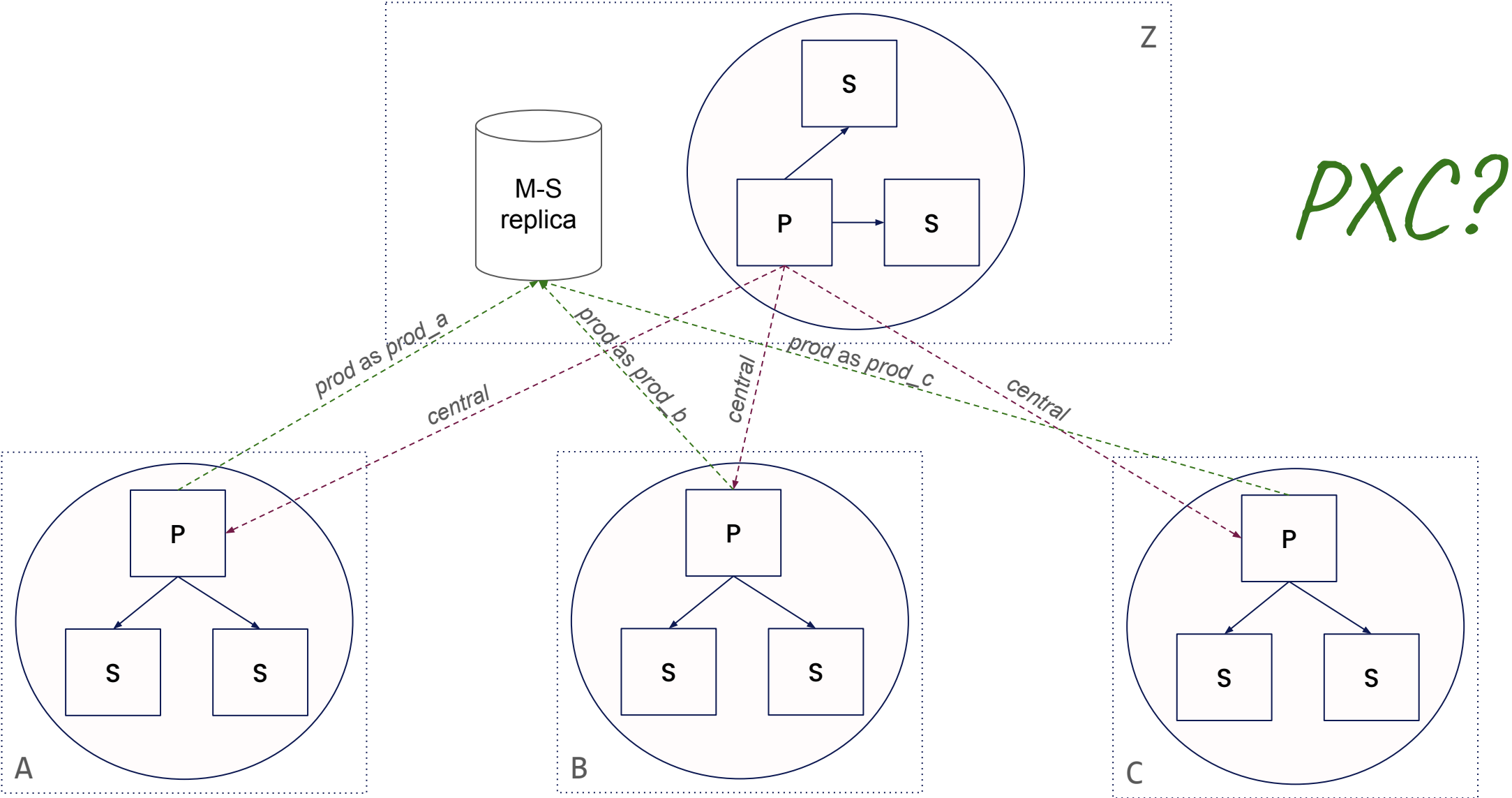


Exemplo 2 InnoDB Cluster with ClusterSet

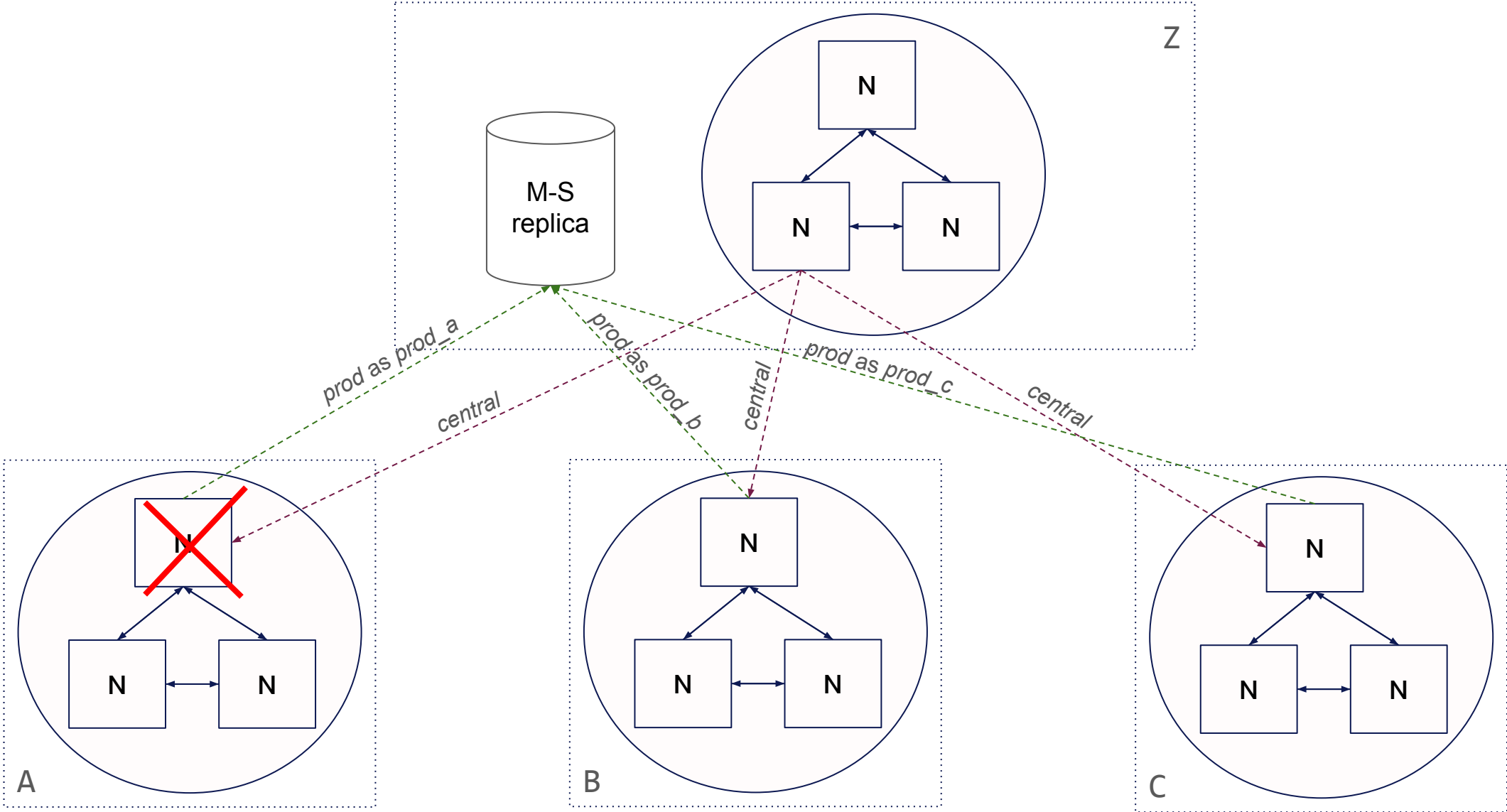


Exemplo 3

Group Replication with rep. filters

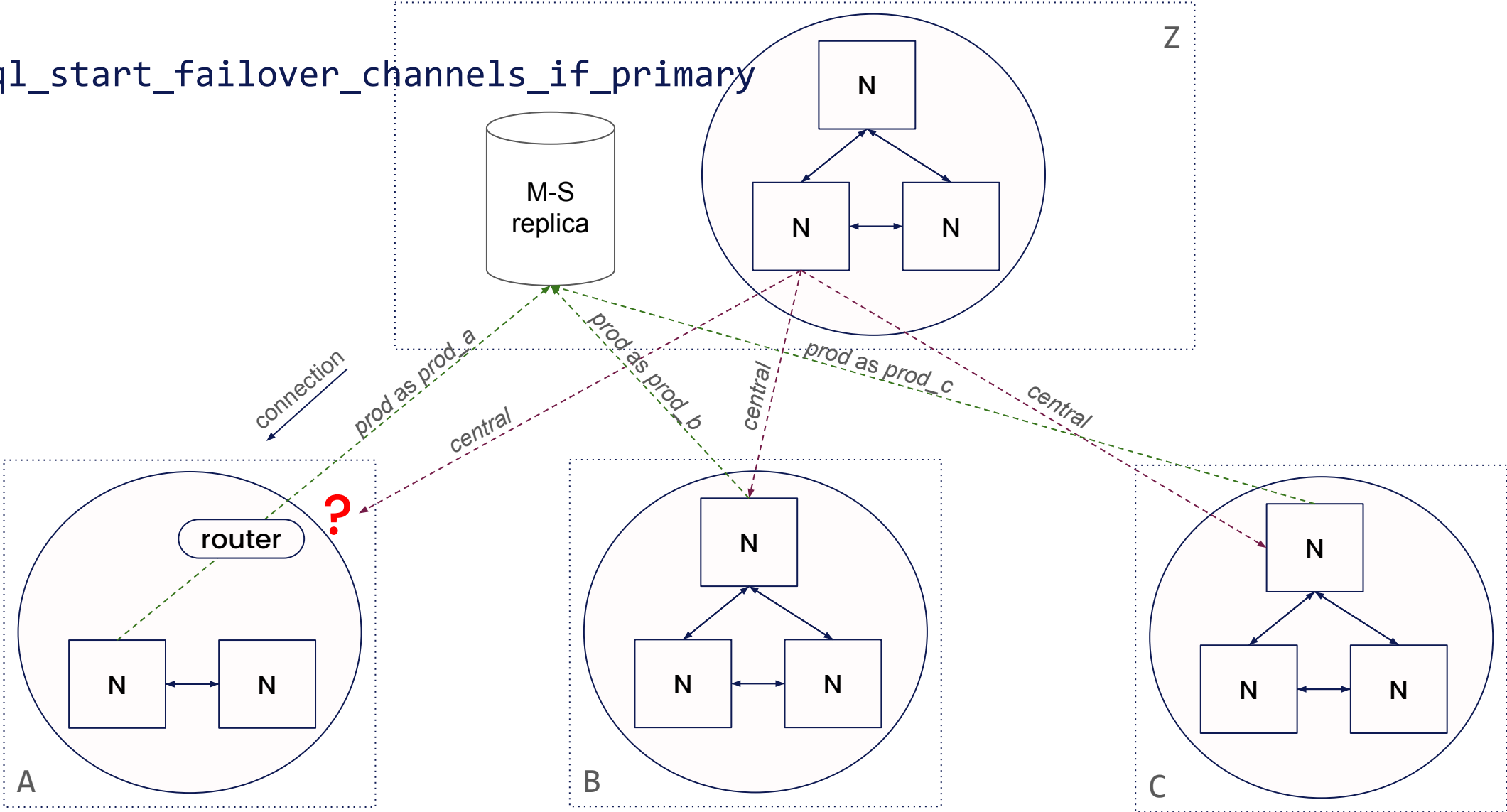


Exemplo 3 ~~PXC~~ Group Replication with rep. filters

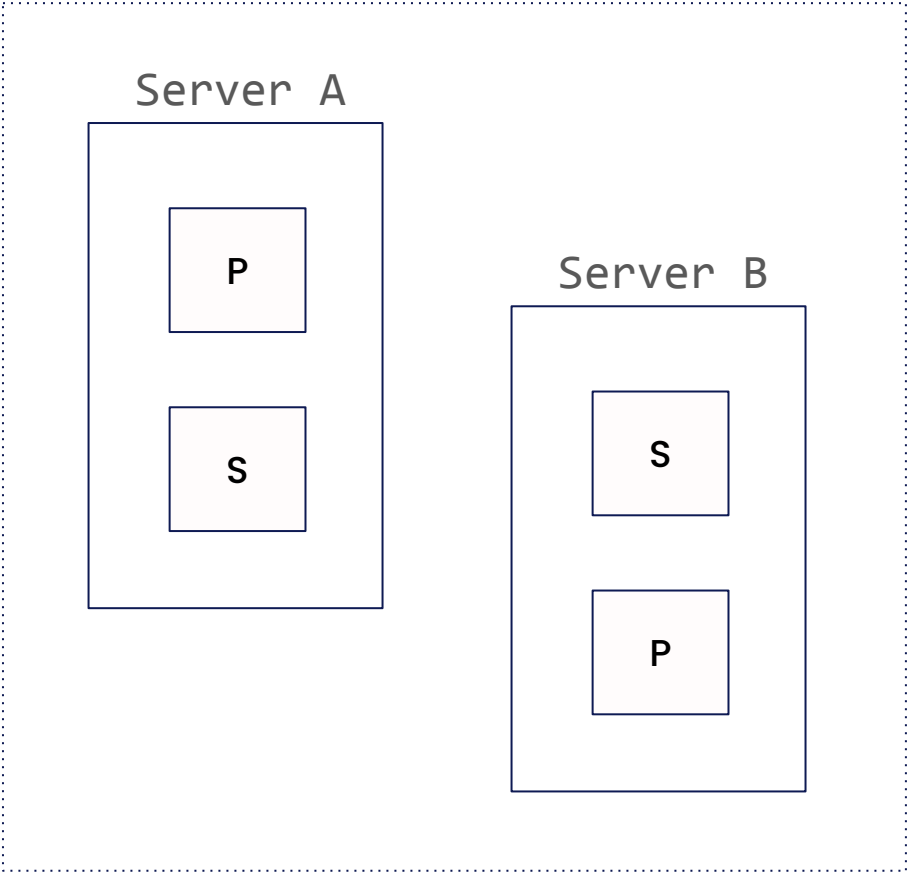


Exemplo 3 ~~PXC~~ Group Replication with rep. filters

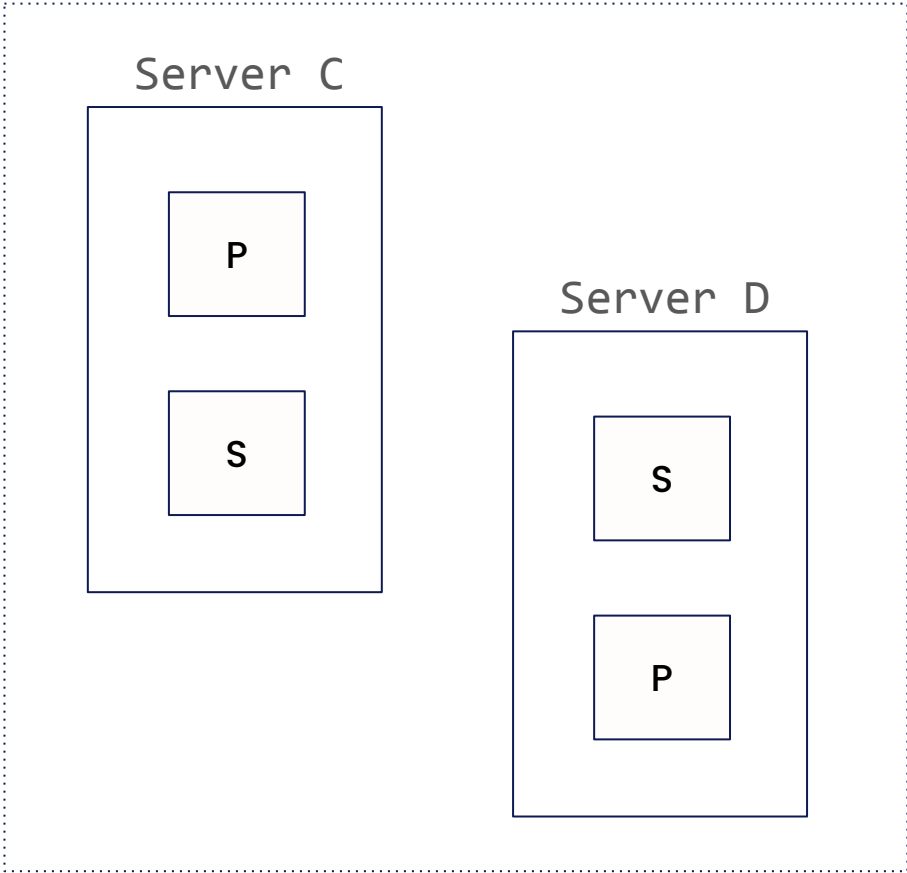
```
mysql_start_failover_channels_if_primary
```



Exemplo 4

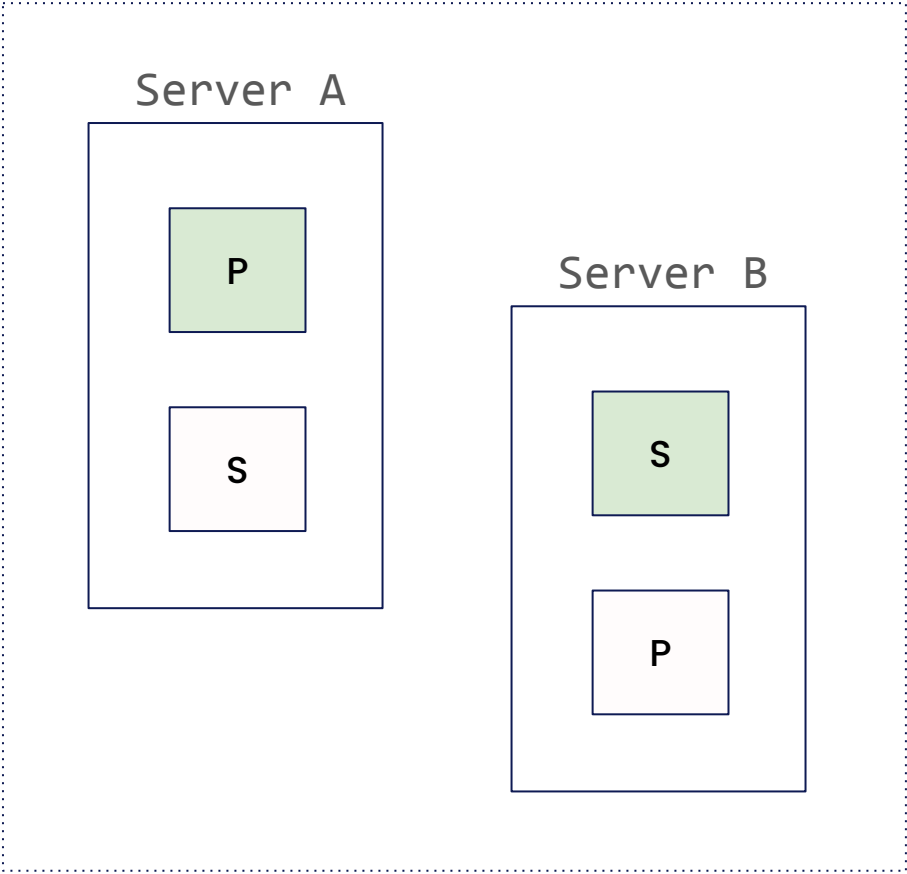


DC1

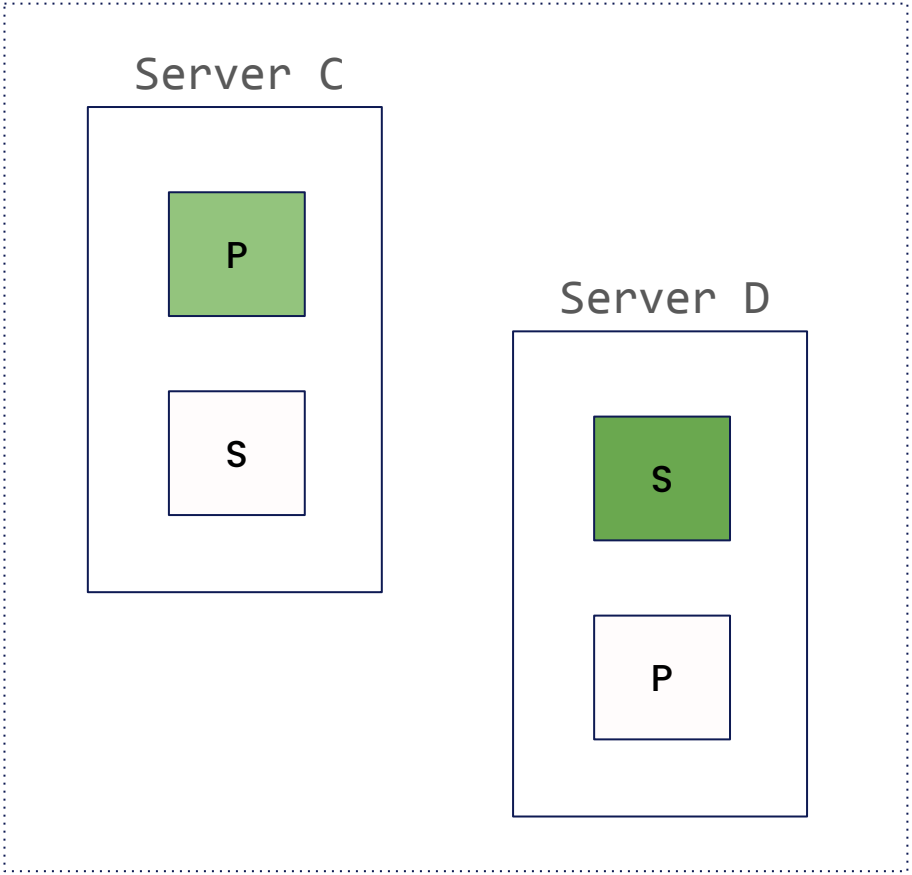


DC2

Exemplo 4



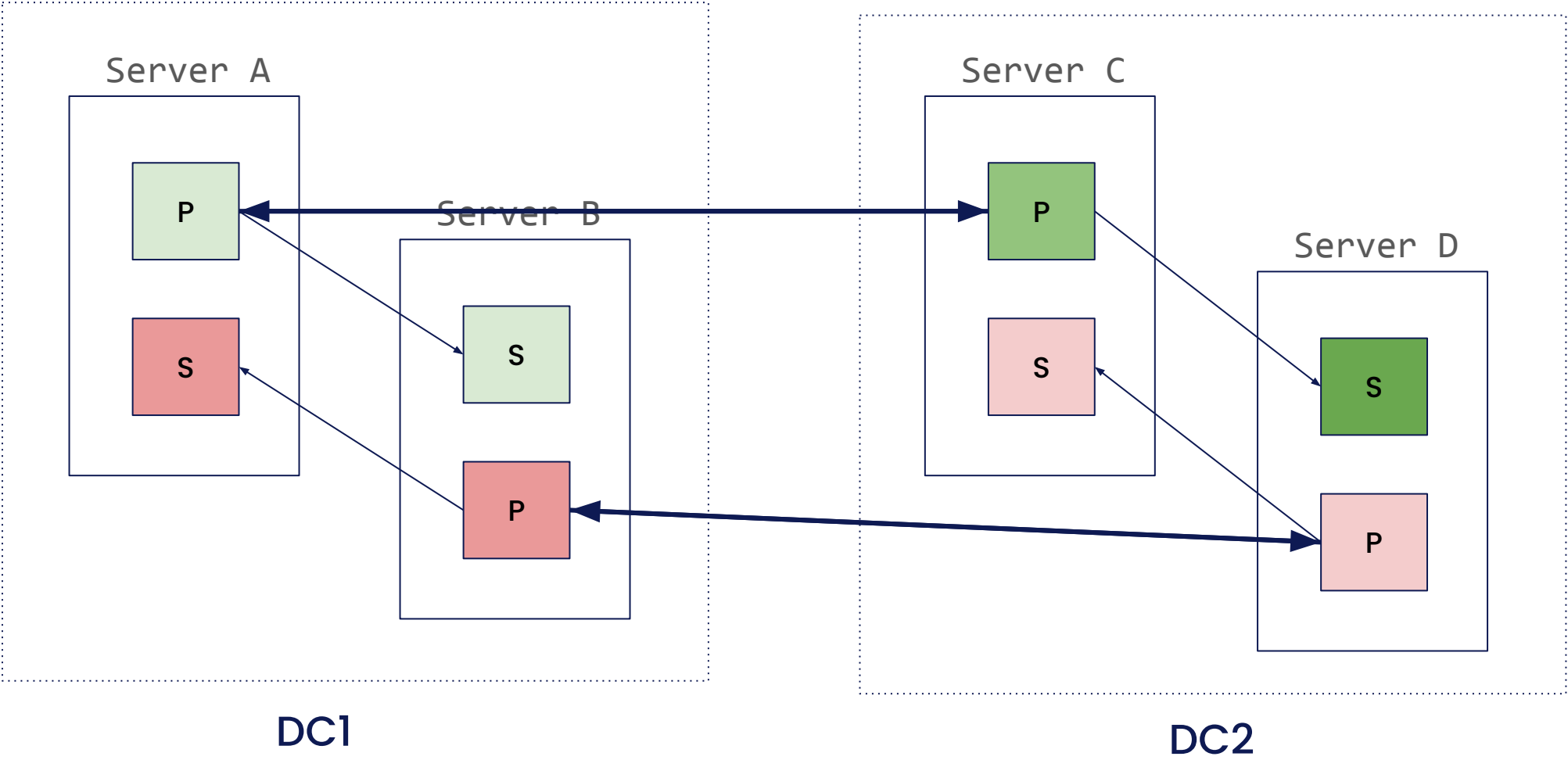
DC1



DC2

Exemplo 4

Group Replication active-active





Continuidade
de serviço: DR

- DC failover
- Backups
 - Binary
 - Logical
- Binlog streaming
- Delayed replica

Imagens

1. <https://medium.com/tensult/configuring-lamp-linux-apache-mysql-php-web-server-on-an-amazon-ec2-linux-instance-2ad01bee1158>
2. <https://galeracluster.com/products/>
3. <https://mariadb.com/docs/galera-cluster/galera-architecture/certification-based-replication/>
4. High Performance MySQL, O'Reilly, 3rd Edition, page 450
5. https://www.youtube.com/channel/UCHq4vL5mUYtA2_5ykOQhQQw
6. <https://www.percona.com/blog/percona-xtradb-cluster-5-6-45-28-36-is-now-available/>
7. <https://www.josedomingo.org/pledin/2022/02/galera-mariadb/>
8. <https://freedesignfile.com/upload/2019/08/Business-people-office-at-computer-vector-illustration.jpg>
9. i2clipart.com
10. <https://emojiterrra.com/thumbs-up/>



Laila Purina
Senior Global Talent Acquisition
Percona

**Conectando Seu Talento ao
Mundo: Carreira Global em
Tecnologia**
15:40 - sala 708

Obrigado!

fernando.laudares@percona.com

www.percona.com/careers