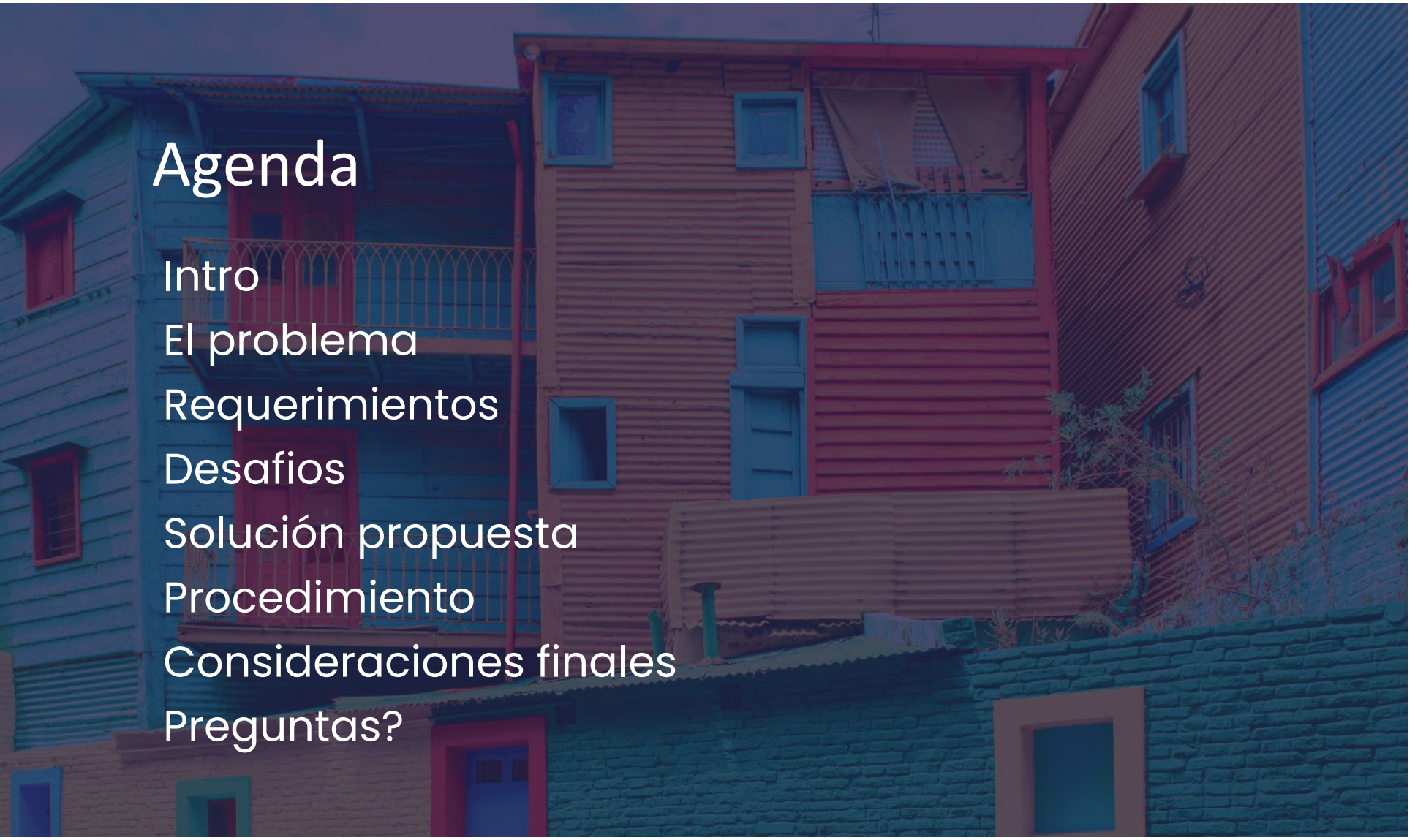
Un viaje desde un Single Point of Failure hacia una solución de alta disponibilidad usando InnoDB ClusterSet

Francisco Bordenave

Buenos Aires, Argentina 2023

PERCONA UNIVERSITY

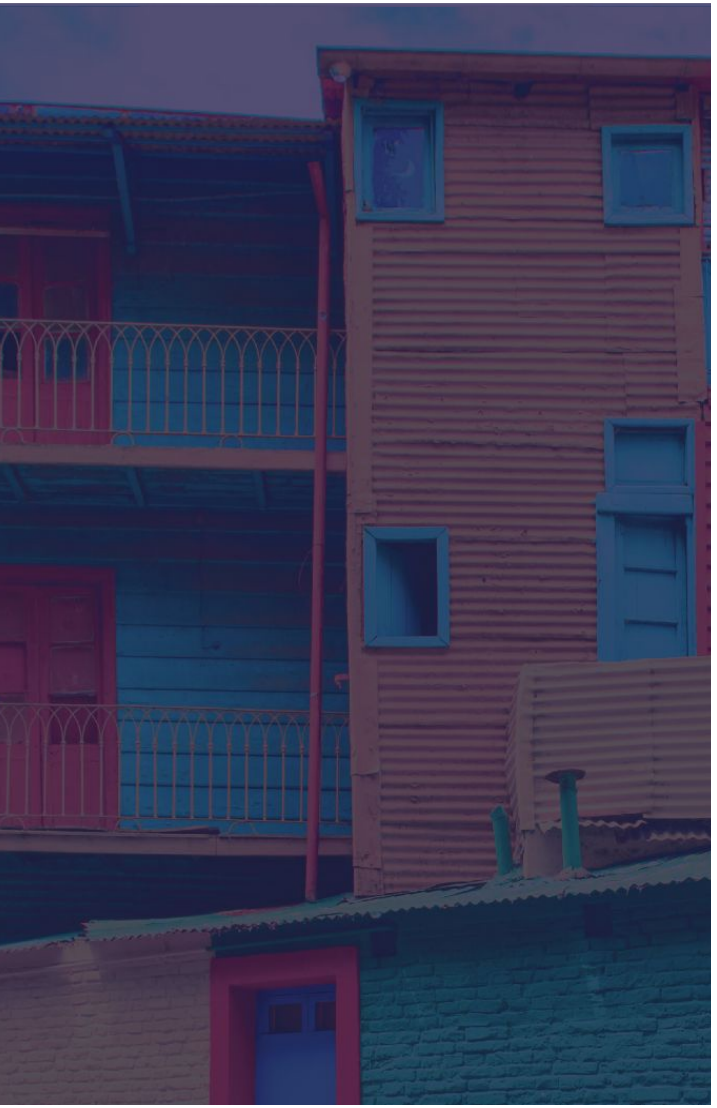
PERCONA



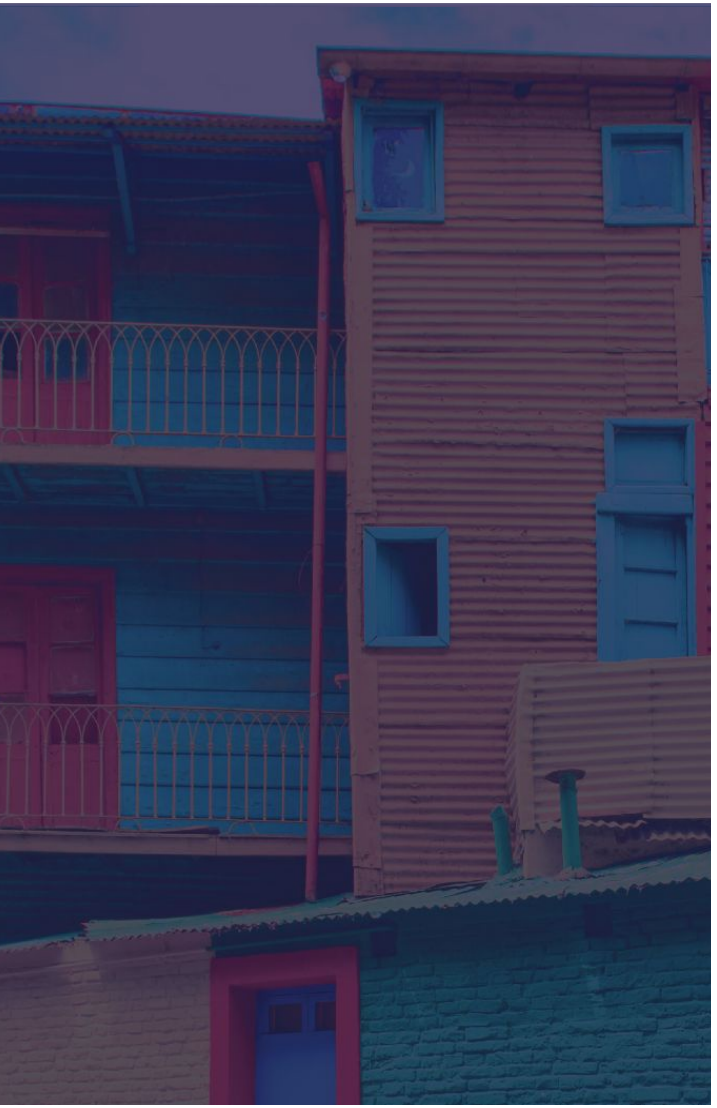
Agenda

Intro
El problema
Requerimientos
Desafios
Solución propuesta
Procedimiento
Consideraciones finales
Preguntas?

Buenos Aires, Argentina 2023
PERCONA UNIVERSITY



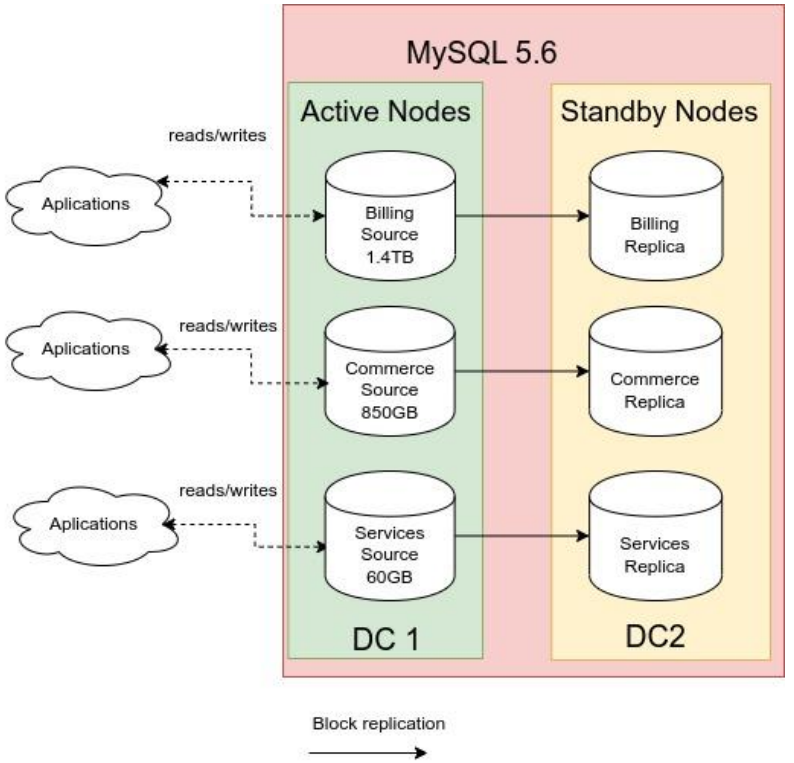
El problema



El problema
(basado en una historia real)

Escenario

MySQL 5.6 EoL por años

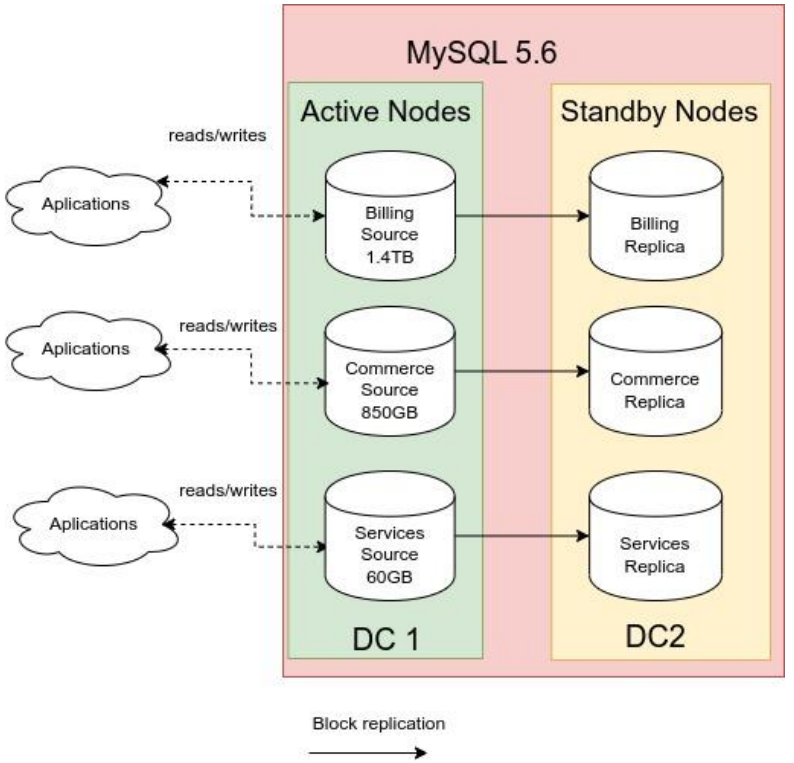


PERCONA

Escenario

MySQL 5.6 EoL por años

Single point of failure



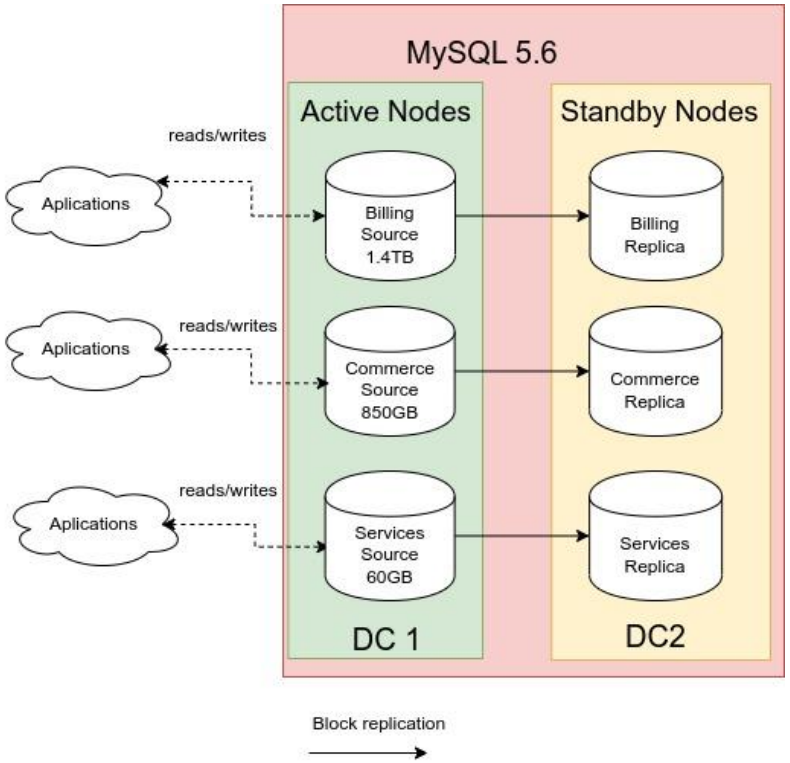
PERCONA

Escenario

MySQL 5.6 EoL por años

Single point of failure

Failover manual



PERCONA

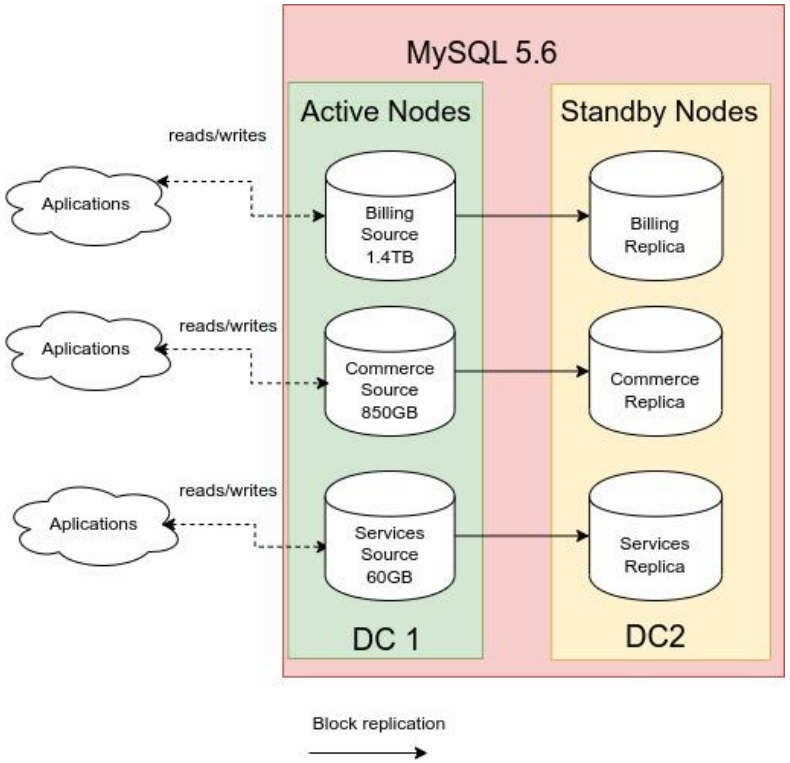
Escenario

MySQL 5.6 EoL por años

Single point of failure

Failover manual

Sin balanceo de carga



PERCONA

Escenario

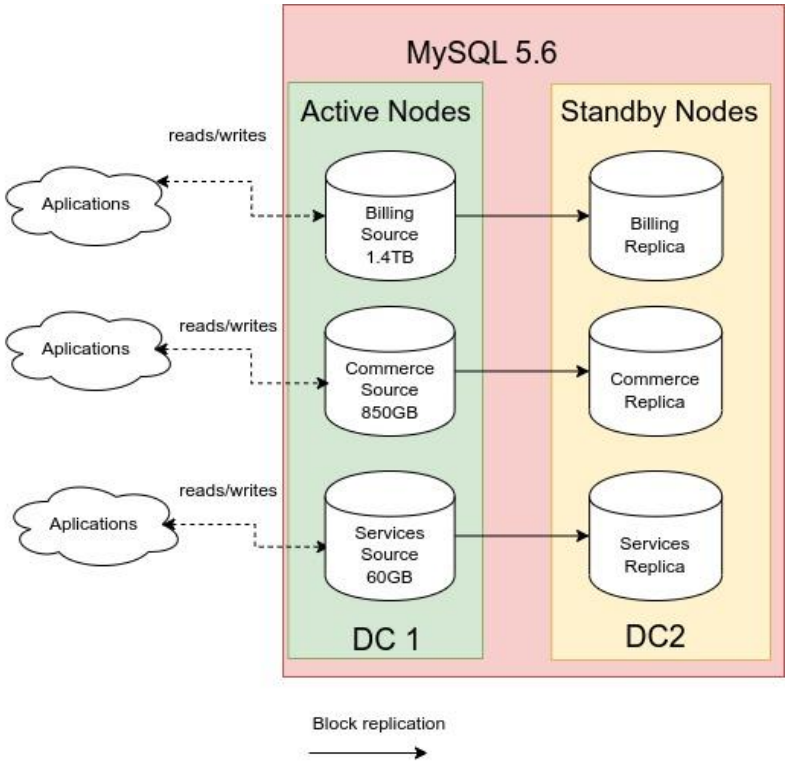
MySQL 5.6 EoL por años

Single point of failure

Failover manual

Sin balanceo de carga

Administración compleja



PERCONA

Escenario

MySQL 5.6 EoL por años

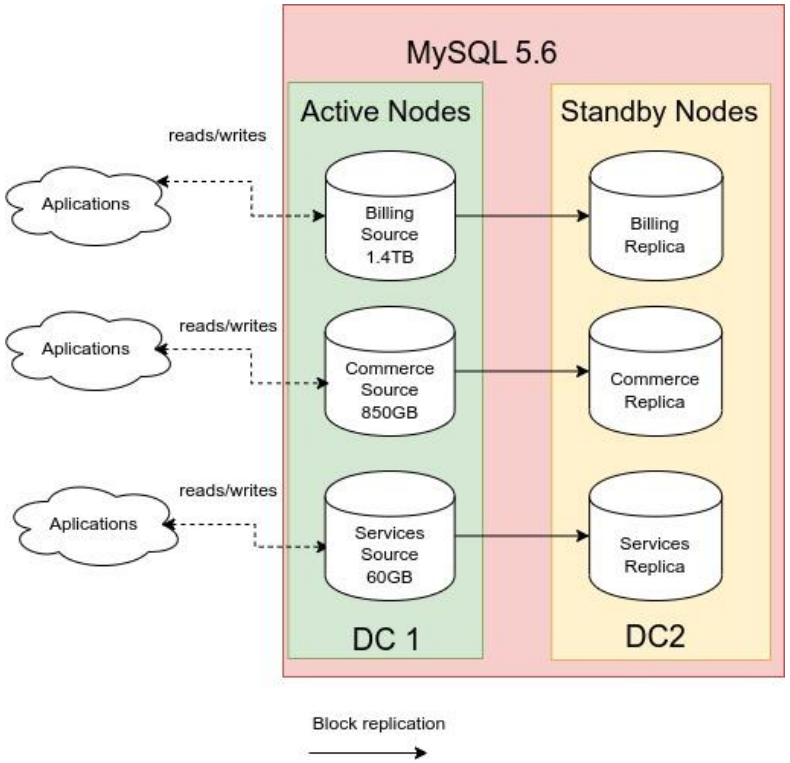
Single point of failure

Failover manual

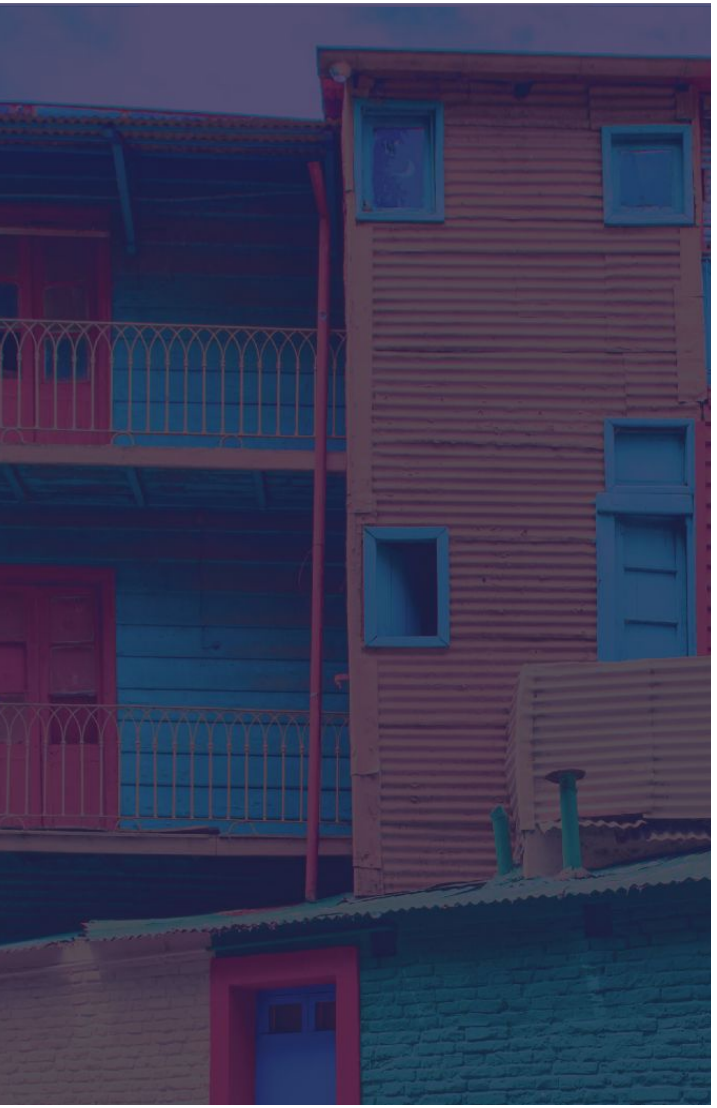
Sin balanceo de carga

Administración compleja

Mal manejo de DR



PERCONA



Requerimientos

Requerimientos

Actualizar a 8.0 - SLA

PERCONA

Requerimientos

Actualizar a 8.0 - SLA

Alta disponibilidad

PERCONA

Requerimientos

Actualizar a 8.0 - SLA

Alta disponibilidad

Tolerante a fallos de DC - Planificar DR

PERCONA

Requerimientos

Actualizar a 8.0 - SLA

Alta disponibilidad

Tolerante a fallos de DC - Planificar DR

0 Downtime durante el upgrade

Requerimientos

Actualizar a 8.0 - SLA

Alta disponibilidad

Tolerante a fallos de DC - Planificar DR

0 Downtime durante el upgrade

Fácil mantenimiento

Requerimientos

Actualizar a 8.0 - SLA

Alta disponibilidad

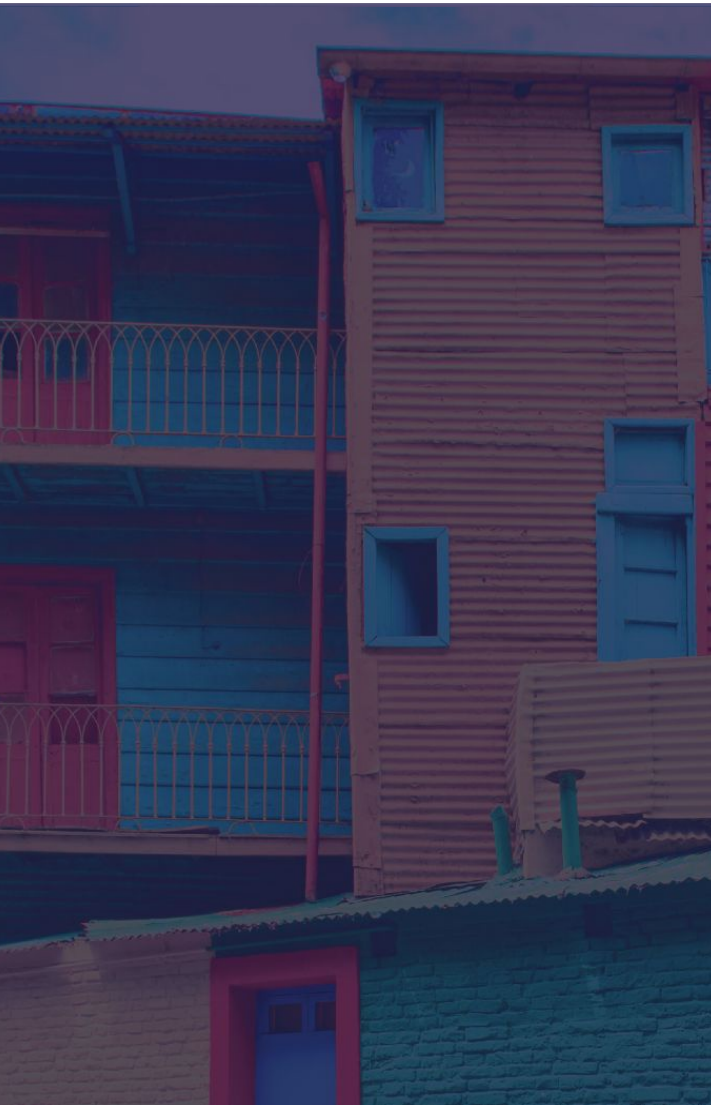
Tolerante a fallos de DC - Planificar DR

0 Downtime durante el upgrade

Fácil mantenimiento

…. y más

PERCONA



Desafíos

Desafíos

Actualización intermedia a 5.7

PERCONA

Desafíos

Actualización intermedia a 5.7

Consolidar servicios para reducir complejidad

PERCONA

Desafíos

Actualización intermedia a 5.7

Consolidar servicios para reducir complejidad

Asegurar compatibilidad

PERCONA

Desafíos

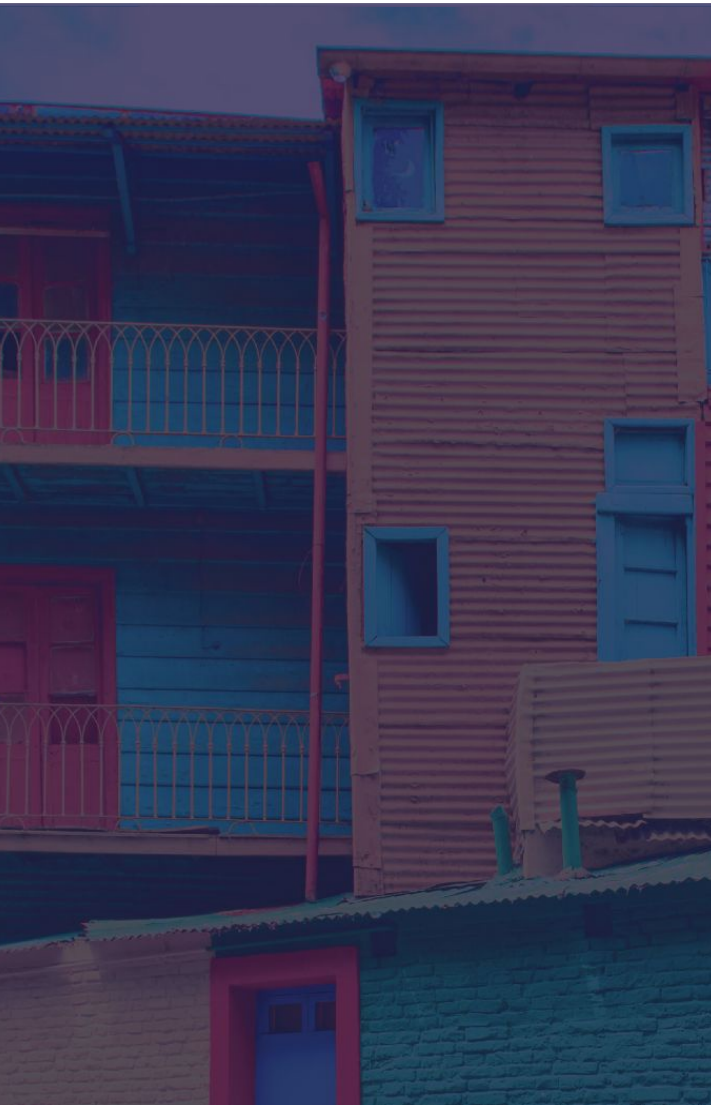
Actualización intermedia a 5.7

Consolidar servicios para reducir complejidad

Asegurar compatibilidad

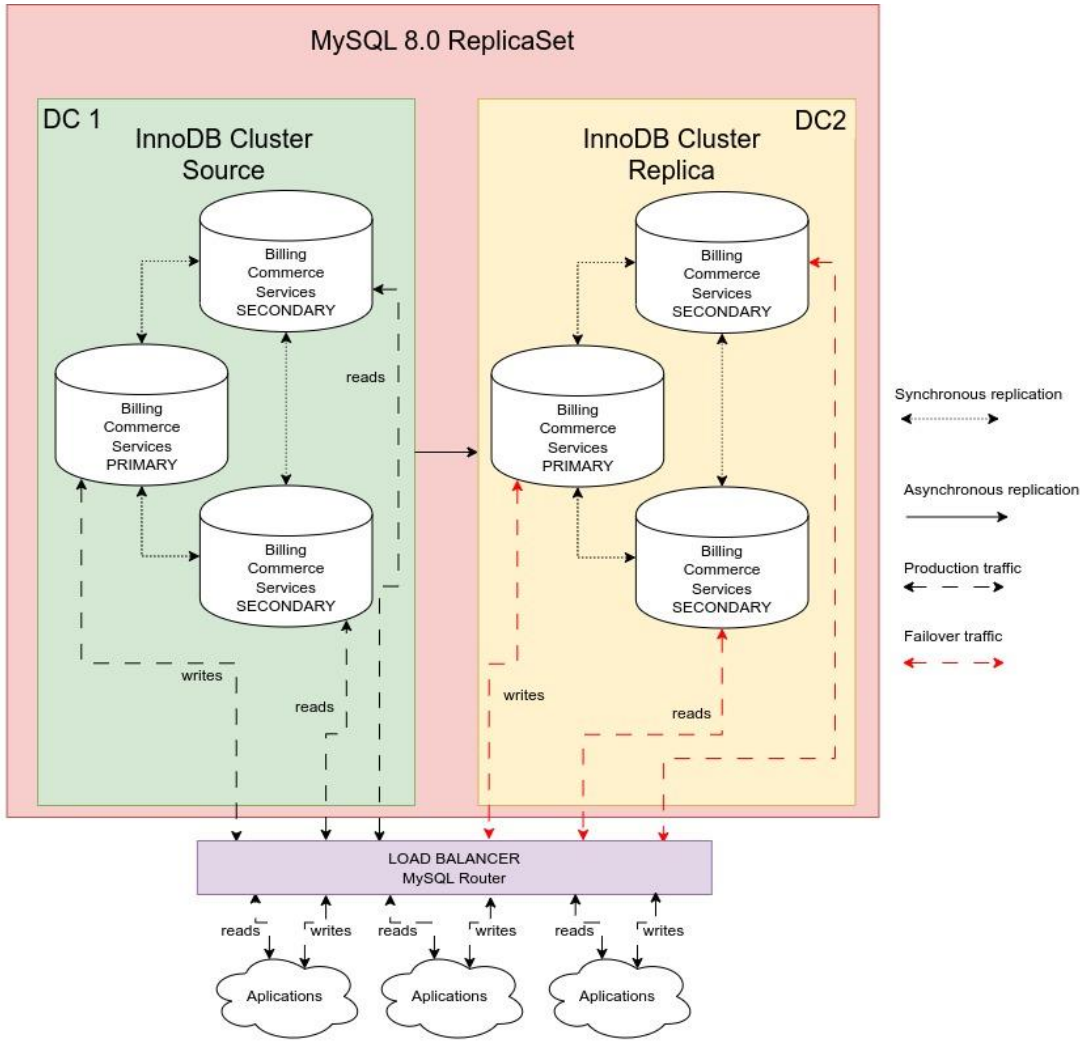
Mantener consistencia de datos durante la migración

PERCONA

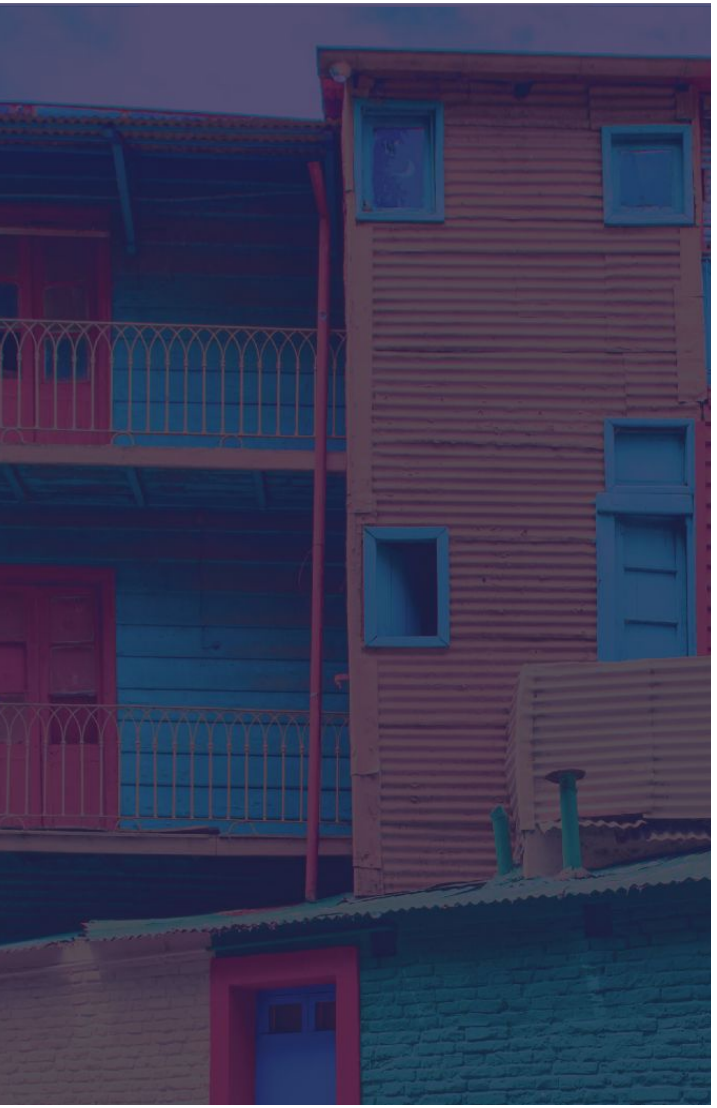


Solución propuesta

Innodb ClusterSet



PERCONA

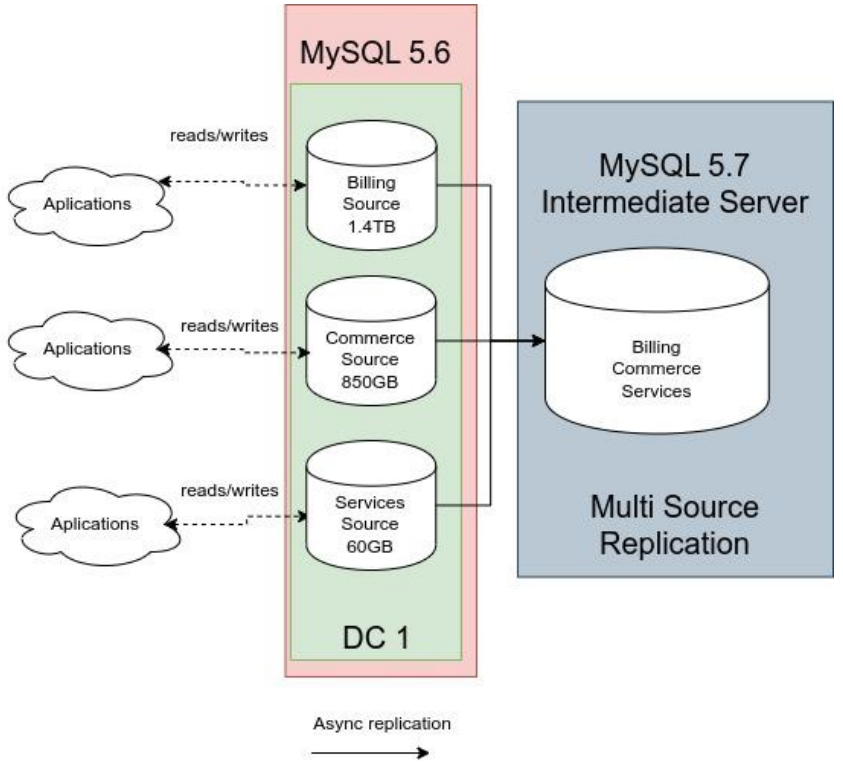


Procedimiento

Procedimiento

Crear un servidor intermedio

Testear upgrade a 5.7



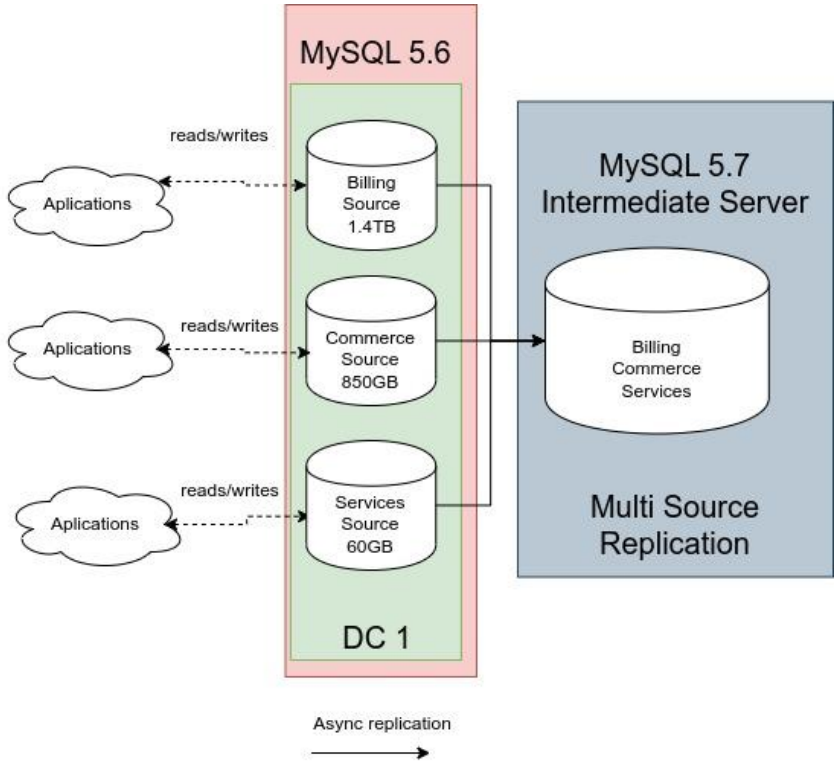
PERCONA

Procedimiento

Crear un servidor intermedio

Testear upgrade a 5.7

Testear compatibilidad con 5.6



PERCONA

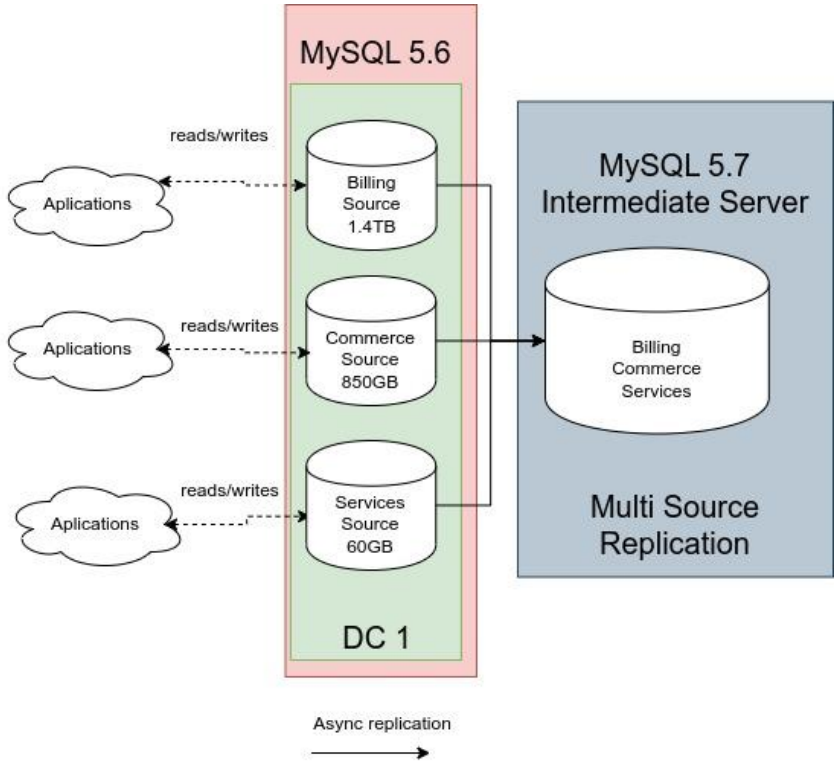
Procedimiento

Crear un servidor intermedio

Testear upgrade a 5.7

Testear compatibilidad con 5.6

Consolidar bases de datos para reducir complejidad



PERCONA

Procedimiento

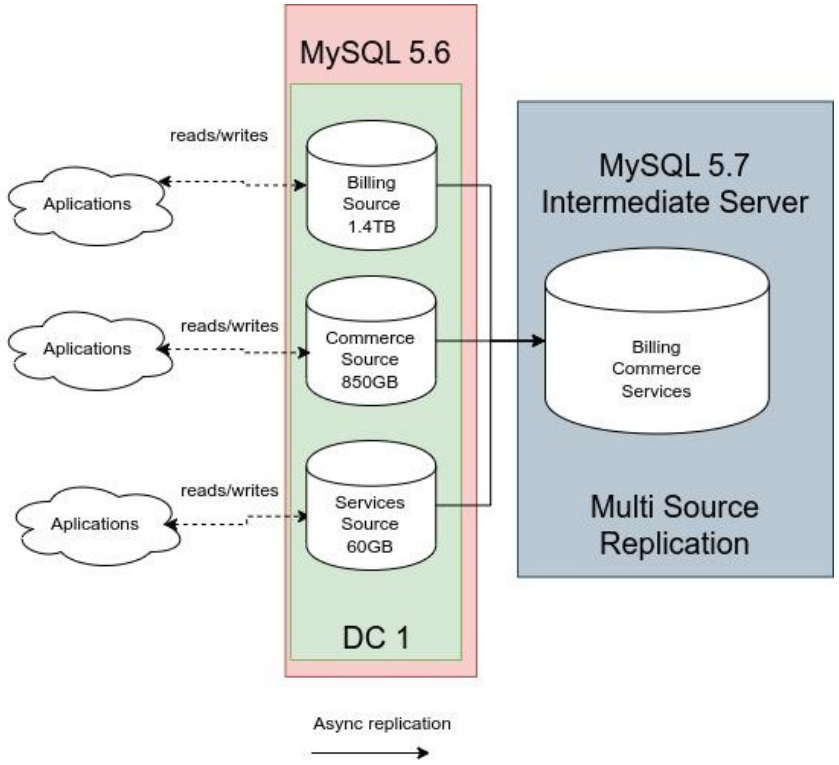
Crear un servidor intermedio

Testear upgrade a 5.7

Testear compatibilidad con 5.6

Consolidar bases de datos para reducir complejidad

No se pueden usar los secundarios como backup



PERCONA

Procedimiento

Crear un servidor intermedio

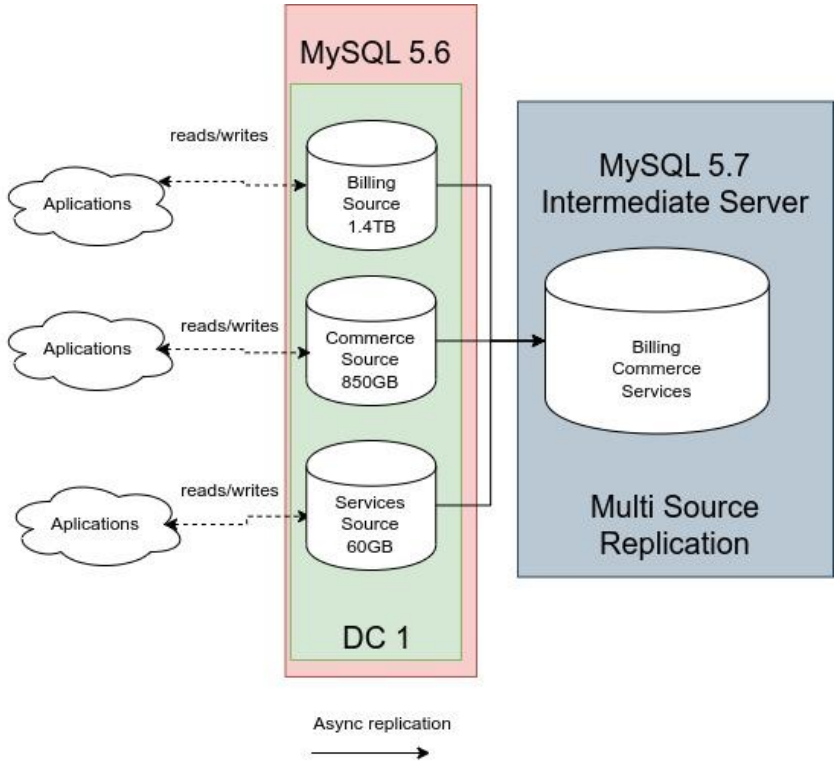
Testear upgrade a 5.7

Testear compatibilidad con 5.6

Consolidar bases de datos para reducir complejidad

No se pueden usar los secundarios como backup

Usar XtraBackup para la DB más grande



PERCONA

Procedimiento

Crear un servidor intermedio

Testear upgrade a 5.7

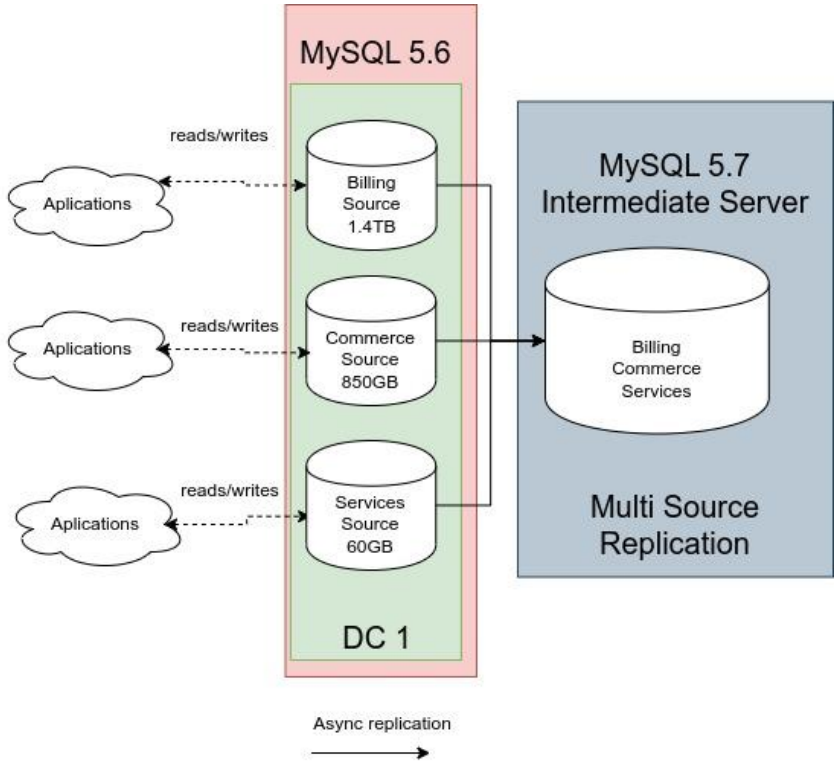
Testear compatibilidad con 5.6

Consolidar bases de datos para reducir complejidad

No se pueden usar los secundarios como backup

Usar XtraBackup para la DB más grande

Usar mydumper/myloader para las DBs restantes

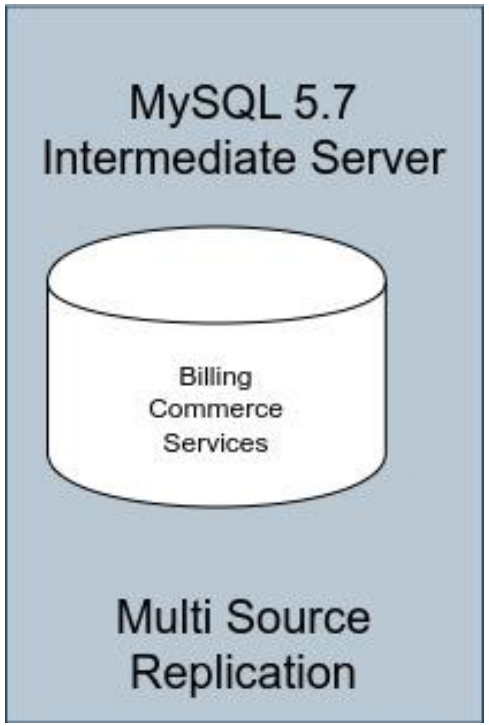


PERCONA

Procedimiento

Creación del servidor intermedio

```
my.cnf
log-bin
log-slave-updates
binlog-format=row
```



Procedimiento

Creación del servidor intermedio

Origen - billing

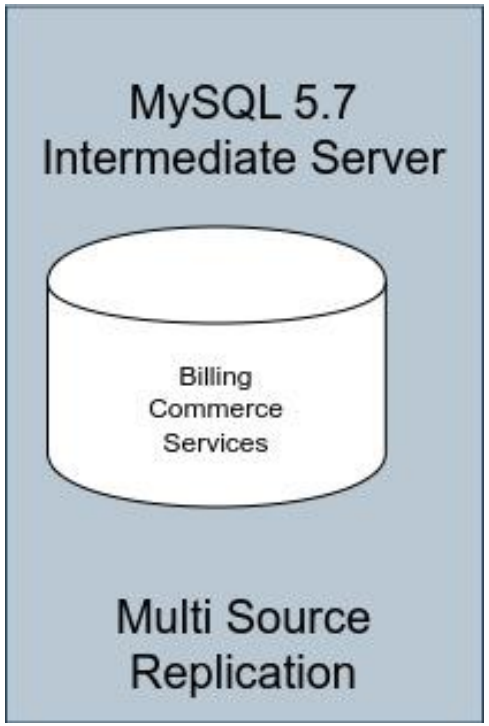
```
xtrabackup --user='bkpuser' --password='s3cret' --backup
--target-dir=./ --stream=xbstream |nc 10.50.30.250 3306
```

Servidor intermedio

```
nc -l 9999 xbstream -x -C /var/lib/mysql/
```

```
xtrabackup --prepare --target-dir=/data/backups/
```

```
mysql_upgrade -s
```



PERCONA

Procedimiento

Creación del servidor intermedio

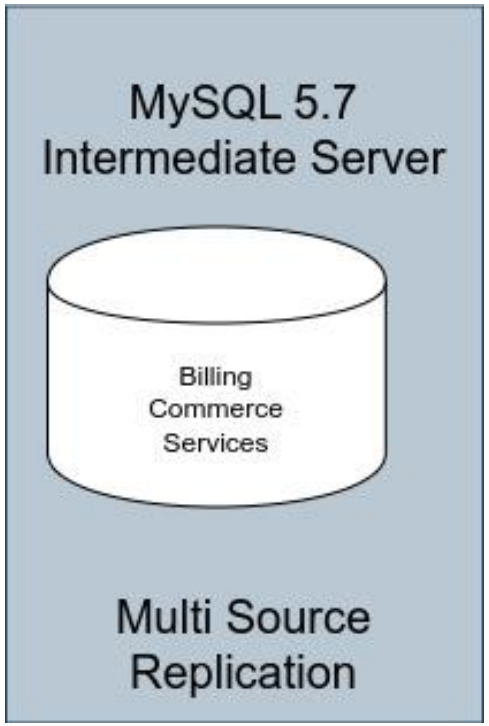
Origen - resto de las DBs

```
mydumper -v 2 -t 4 -G -E -R -o /var/lib/mysql/backup --regex
'^(audit_service_production\.|backend_service_production\.|bco
mmerce_service_production\.)'
```

```
mydumper -v 2 -t 4 -G -E -R -o /var/lib/mysql/backup --regex
'^(loyalty_service_production\.|support_service_production\.|v
pos_service_production\.)'
```

Servidor intermedio

```
myloader --host=127.0.0.1 --directory=/var/lib/mysql/backup
--threads=8
```



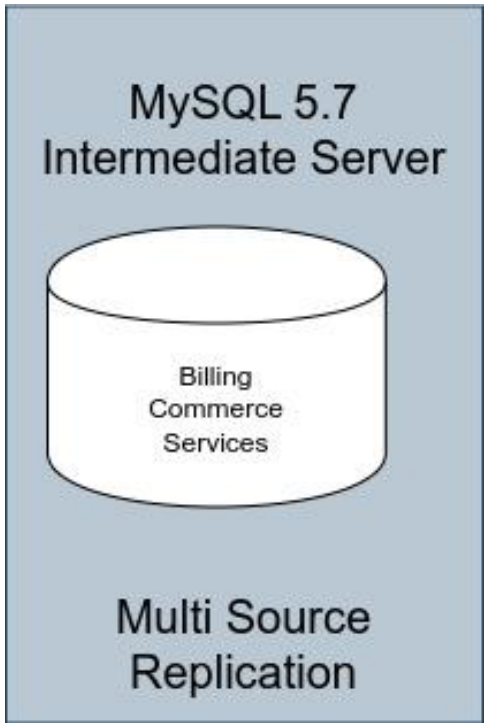
PERCONA

Procedimiento

Creación del servidor intermedio

Servidor intermedio

```
CHANGE MASTER TO master_host='10.50.30.11',
master_user='replication', master_password='r3pl1c4t10n',
master_log_file='mysql-bin.002546', master_log_pos=570428664
for channel 'commerce';

CHANGE MASTER TO master_host='10.50.30.21',
master_user='replication', master_password='r3pl1c4t10n',
master_log_file='mysql-bin.000152', master_log_pos=36271765
for channel billing;

CHANGE MASTER TO master_host='10.50.30.7',
master_user='replication', master_password='r3pl1c4t10n',
master_log_file='mysql-bin.001611', master_log_pos=936303055
for channel 'services';
```



Procedimiento

El servidor 5.7 está listo - Y ahora?

PERCONA

Procedimiento

El servidor 5.7 está listo - Y ahora?

Nuevos desafíos

PERCONA

Procedimiento

El servidor 5.7 está listo - Y ahora?

Nuevos desafíos

Testear compatibilidad con 8.0

PERCONA

Procedimiento

El servidor 5.7 está listo - Y ahora?

Nuevos desafíos

Testear compatibilidad con 8.0

```
mysqlsh -- util checkForServerUpgrade
```


PERCONA

Procedimiento

El servidor 5.7 está listo - Y ahora?

Nuevos desafíos

Testear compatibilidad con 8.0

```
mysqlsh -- util checkForServerUpgrade
```

Group Replication requiere GTID

PERCONA

Procedimiento

El servidor 5.7 está listo - Y ahora?

Nuevos desafíos

Testear compatibilidad con 8.0

```
mysqlsh -- util checkForServerUpgrade
```

Group Replication requiere GTID

```
ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS
```


PERCONA

Preparando la instancia

```
mysqlsh -- util checkForServerUpgrade user@localhost:3306 --output-format=JSON
The MySQL server at example.com:3306, version
5.7.33-enterprise-commercial-advanced - MySQL Enterprise Server - Advanced Edition (Commercial),
will now be checked for compatibility issues for upgrade to MySQL 8.0.29...
1) Usage of old temporal type
  No issues found
2) Usage of db objects with names conflicting with new reserved keywords
  Warning: The following objects have names that conflict with new reserved keywords.
  Ensure queries sent by your applications use `quotes` when referring to them or they will result in errors.
  More information: https://dev.mysql.com/doc/refman/en/keywords.html
  dbtest.System - Table name
  dbtest.System.JSON_TABLE - Column name
  dbtest.System.cube - Column name
```


PERCONA

Configuración del server 8.0

```
[mysqld]
bind-address = 0.0.0.0
report-host = 10.50.30.200 #change this value on each 8.0 instance
server_id = 1001 #change this value on each 8.0 instance
gtid_mode = ON
enforce_gtid_consistency = ON
log-replica-updates = ON
group_replication_start_on_boot=OFF
binlog-checksum = NONE
binlog_transaction_dependency_tracking = WRITESET
replica_preserve_commit_order = 1
replica_parallel_type = LOGICAL_CLOCK
transaction-write-set-extraction = XXHASH64
```


PERCONA

Creando el InnoDB ClusterSet - DC1

```
CREATE USER 'clusteradm'@'10.%';

GRANT ALL PRIVILEGES ON *.* TO 'clusteradm'@'10.%';

GRANT CLONE_ADMIN, CONNECTION_ADMIN, CREATE USER, EXECUTE, FILE, GROUP_REPLICATION_ADMIN,
PERSIST_RO_VARIABLES_ADMIN, PROCESS, RELOAD, REPLICATION CLIENT, REPLICATION SLAVE,
REPLICATION_APPLIER, REPLICATION_SLAVE_ADMIN, ROLE_ADMIN, SELECT, SHUTDOWN,
SYSTEM_VARIABLES_ADMIN ON *.* TO 'clusteradm'@'10.%'  WITH GRANT OPTION;
```


PERCONA

Creando el InnoDB ClusterSet - DC1

```
mysqlsh -uclusteradm -h10.50.30.101 -ppass
dba.configureInstance()
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...
This instance reports its own address as 10.50.30.101:3306
applierWorkerThreads will be set to the default value of 4.
The instance '10.50.30.101:3306' is valid to be used in an InnoDB cluster.
The instance '10.50.30.101:3306' is already ready to be used in an InnoDB cluster.
Successfully enabled parallel appliers.

dba.checkInstanceConfiguration()
Validating local MySQL instance listening at port 3306 for use in an InnoDB cluster...
…
The instance '10.50.30.101:3306' is valid to be used in an InnoDB cluster.
{
    "status": "ok"
```

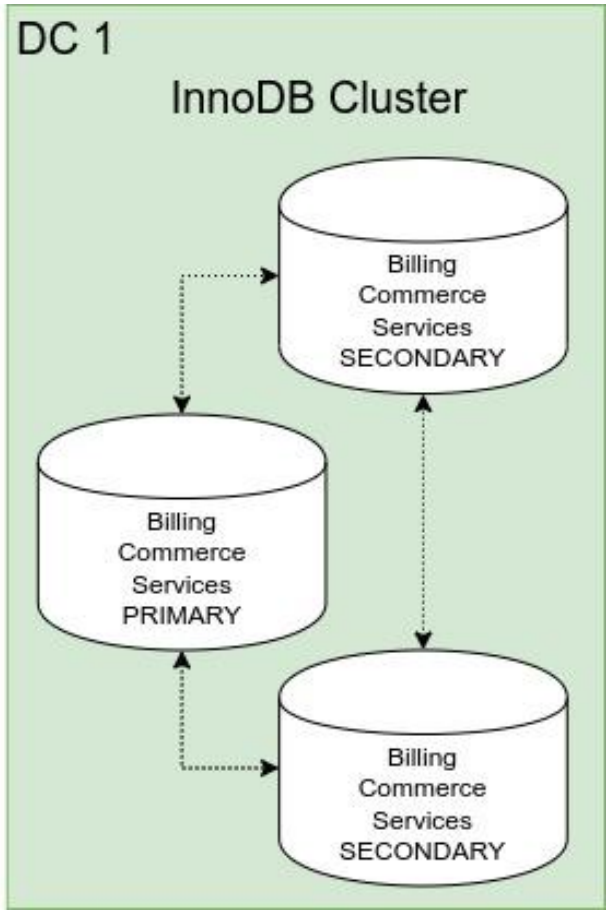

PERCONA

Creando el InnoDB ClusterSet - DC1

```
mysqlsh -uclusteradm -h10.50.30.101 -ppass

var cluster = dba.createCluster('GR_DC1');

cluster.addInstance({user:'clusteradm', host:'10.50.30.103', password:'pass'});

cluster.addInstance('clusteradm@10.50.30.105');
cluster.status()
{
    "clusterName": "GR_DC1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "10.50.30.101:3306",
        "status": "OK",
        "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
        "topology": {
            "10.50.30.101:3306": {
                "address": "10.50.30.101:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",...
            },
            "10.50.30.103:3306": {
                "address": "10.50.30.103:3306",
                "memberRole": "SECONDARY",
…
}
```


PERCONA

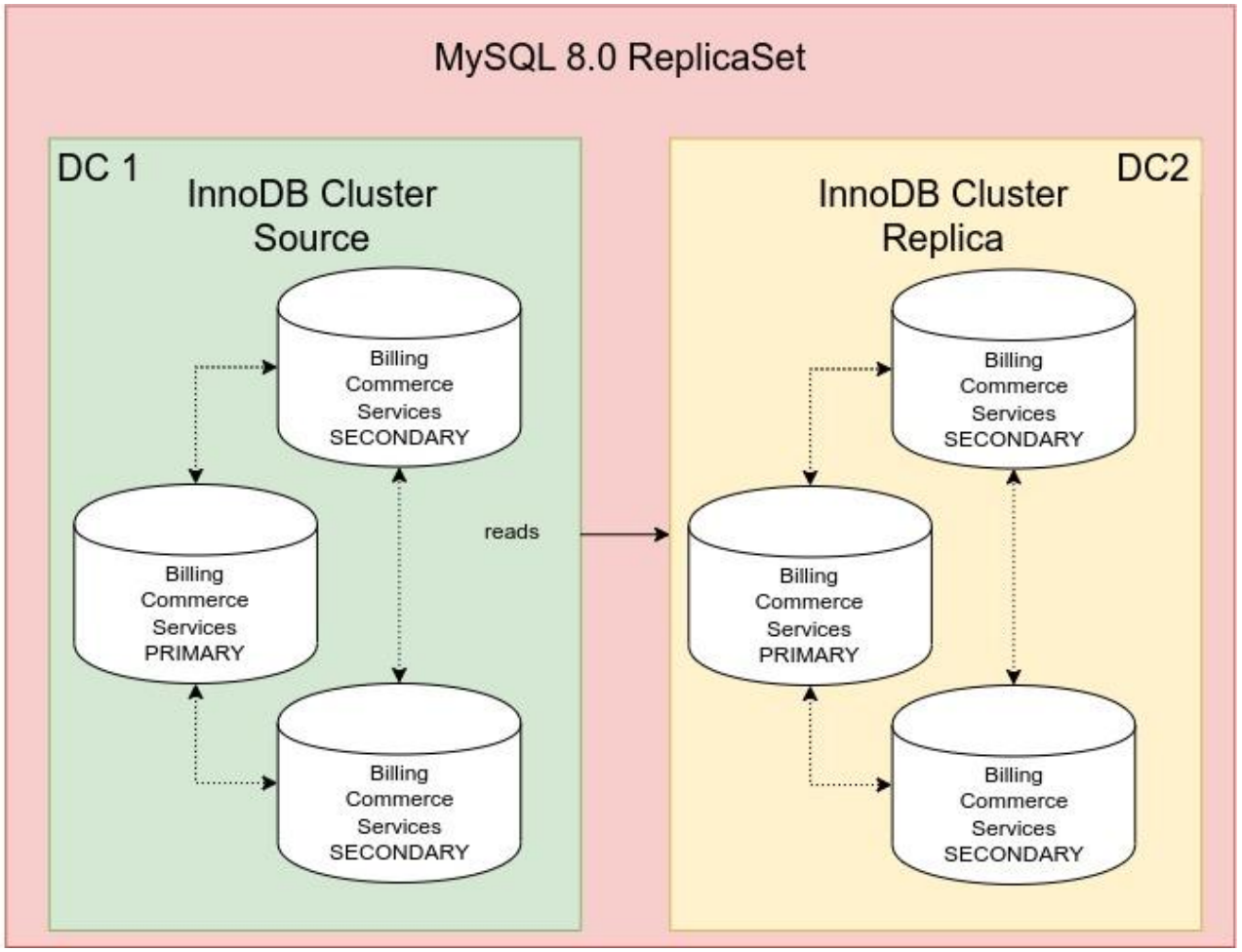
Creando el InnoDB ClusterSet - DC1



Creando el InnoDB ClusterSet - DC2 (Replica Cluster)

```
mysqlsh> var clusterset = cluster.createClusterSet("GR_DC2");
mysqlsh> var clusterDC2 = clusterset.createReplicaCluster("clusteradm@10.50.30.102", "GR_DC2");
mysqlsh> clusterDC2.addInstance('clusteradm@10.50.30.112');
mysqlsh> clusterDC2.addInstance('clusteradm@10.50.30.114');
mysqlsh> clusterDC2.status()

mysqlsh> clusterset.status();
{
    "clusters": {
        "GR_DC1": {
            "clusterRole": "PRIMARY",
            "globalStatus": "OK",
            "primary": "10.50.30.101:3306"
        },
        "GR_DC2": {
            "clusterRole": "REPLICA",
            "clusterSetReplicationStatus": "OK",
            "globalStatus": "OK"
        }
    },
    "domainName": "GR_DC2",
    "globalPrimaryInstance": "10.50.30.101:3306",
    "primaryCluster": "GR_DC1",
    "status": "HEALTHY",
    "statusText": "All Clusters available."
}
```

Creando el InnoDB ClusterSet - DC2 (Replica Cluster)



PERCONA

Creando el InnoDB ClusterSet - MySQL Router

```
mysqlrouter --bootstrap root@10.50.30.101 --account=clusteradm --name='GR_Router' --force
--directory=/home/mysql_router/mysql_router --conf-use-gr-notifications
Please enter MySQL password for root:
# Bootstrapping MySQL Router instance at '/home/mysql_router/mysql_router'...

Please enter MySQL password for clusteradm:
Fetching Cluster Members
trying to connect to mysql-server at 10.50.30.101:3306
- Creating account(s) (only those that are needed, if any)
- Verifying account (using it to run SQL queries that would be run by Router)
- Storing account in keyring
- Adjusting permissions of generated files
- Creating configuration /home/mysql_router/mysql_router/mysqlrouter.conf

# MySQL Router 'GR_Router' configured for the InnoDB Cluster 'gr_MULTI_DC'

After this MySQL Router has been started with the generated configuration
    $ mysqlrouter -c /home/mysql_router/mysql_router/mysqlrouter.conf

InnoDB Cluster 'gr_MULTI_DC' can be reached by connecting to:
## MySQL Classic protocol
- Read/Write Connections: localhost:6446
- Read/Only Connections:  localhost:6447

## MySQL X protocol
- Read/Write Connections: localhost:6448
- Read/Only Connections:  localhost:6449
```

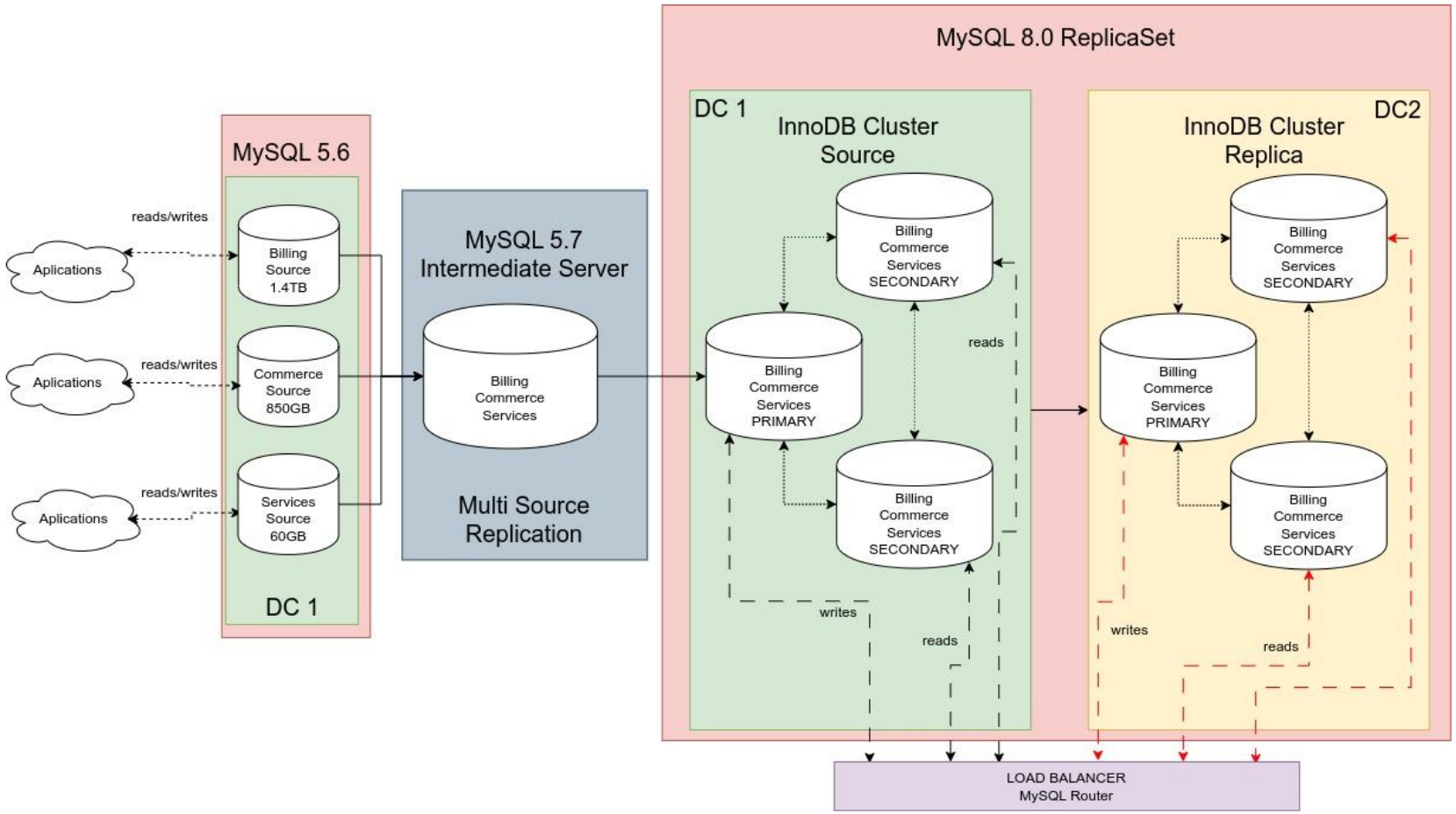

PERCONA

Setear replicación de 5.7 a 8.0

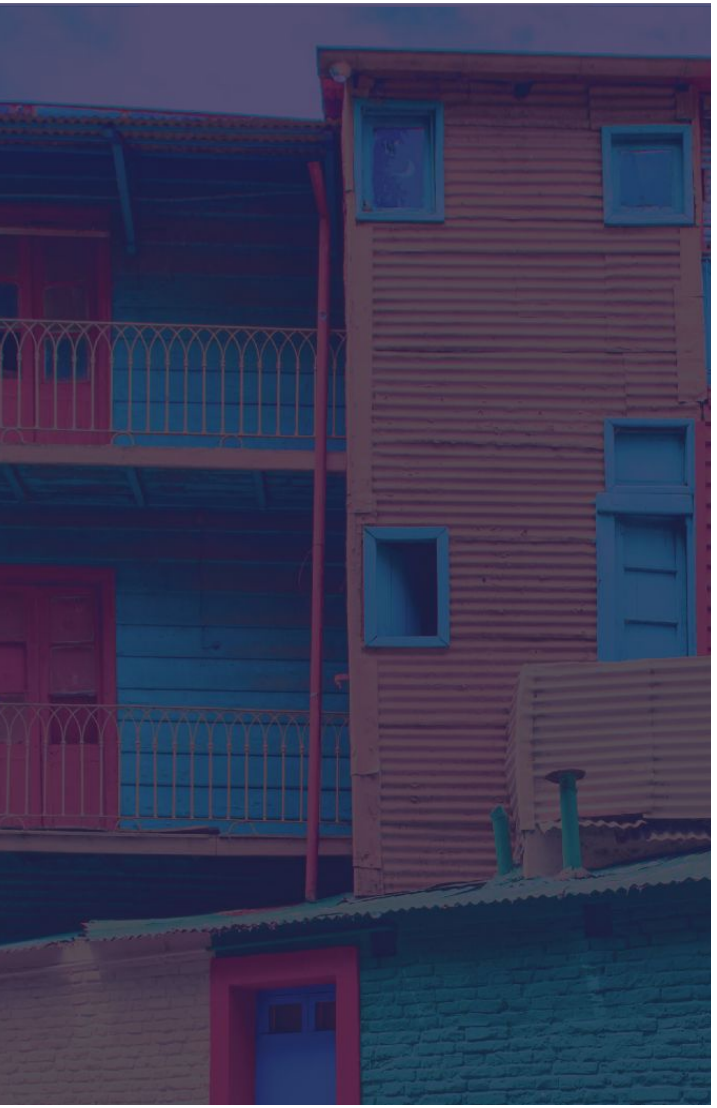
```
CHANGE REPLICATION SOURCE TO source_host='10.50.30.250', source_user='replication',
source_password='r3pl1c4t10n', source_log_file='mysql-bin.000181', source_log_pos=185086690,
ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS='LOCAL' for channel 'production';

START REPLICA for channel 'production';
```


PERCONA

Escenario previo a la migración



PERCONA



Consideraciones finales

Podemos mejorar este setup?

Por supuesto que podemos!

PERCONA

Podemos mejorar este setup?

Por supuesto que podemos!

Agregar un 2do MySQL router

PERCONA

Podemos mejorar este setup?

Por supuesto que podemos!

Agregar un 2do MySQL router
Usar ProxySQL para balanceo de carga

Podemos mejorar este setup?

Por supuesto que podemos!

Agregar un 2do MySQL router
Usar ProxySQL para balanceo de carga
Incrementar redundancia creando un cluster de ProxySQL

Podemos mejorar este setup?

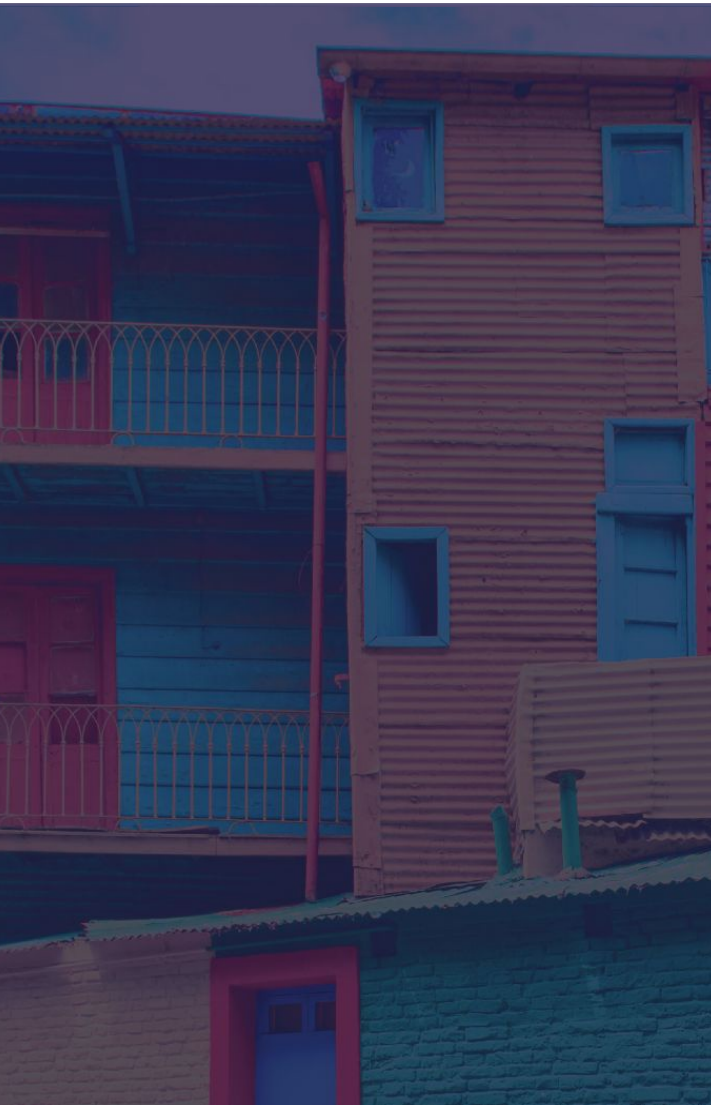
Por supuesto que podemos!

Agregar un 2do MySQL router
Usar ProxySQL para balanceo de carga
Incrementar redundancia creando un cluster de ProxySQL
Agregar una VIP por encima de la capa de conexión y administrarla con keepalived

Podemos mejorar este setup?

Por supuesto que podemos!

Agregar un 2do MySQL router
Usar ProxySQL para balanceo de carga
Incrementar redundancia creando un cluster de ProxySQL
Agregar una VIP por encima de la capa de conexión y administrarla con keepalived
Sean creativos!



Preguntas?

PERCONA