



链滴

MySQL InnoDB Cluser 多主模式实现高可用负载均衡

作者: [zxniuniu](#)

原文链接: <https://ld246.com/article/1553131648906>

来源网站: 链滴

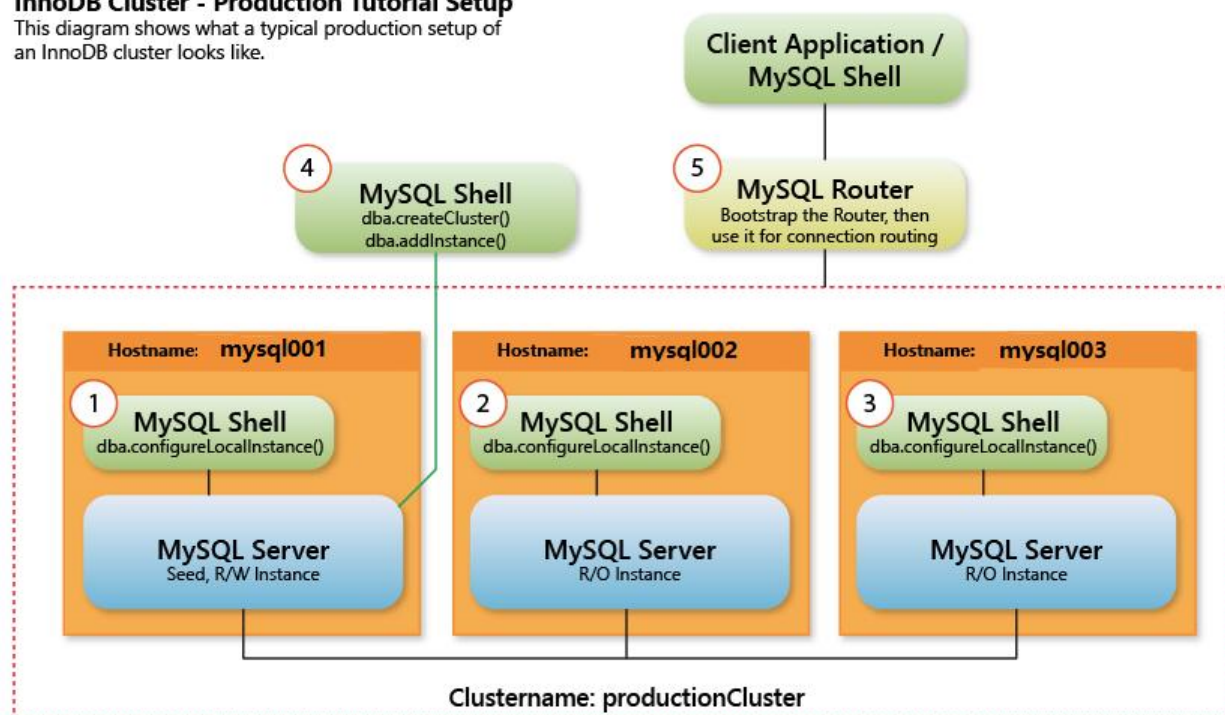
许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

核心架构

- MySQL 的 Group Replication 功能，可以在一组 MySQL 服务器之间实现自动主机选举，形成一多从结构。经过高级配置后，可以实现多主多从结构。
- MySQL Router 是一个轻量级透明中间件，可以自动获取上述集群的状态，规划 SQL 语句，分配合理的 MySQL 后端进行执行。
- MySQL Shell 是一个同时支持 JavaScript 和 SQL 的交互程序，可以快速配置 InnoDB Cluster。

InnoDB Cluster - Production Tutorial Setup

This diagram shows what a typical production setup of an InnoDB cluster looks like.

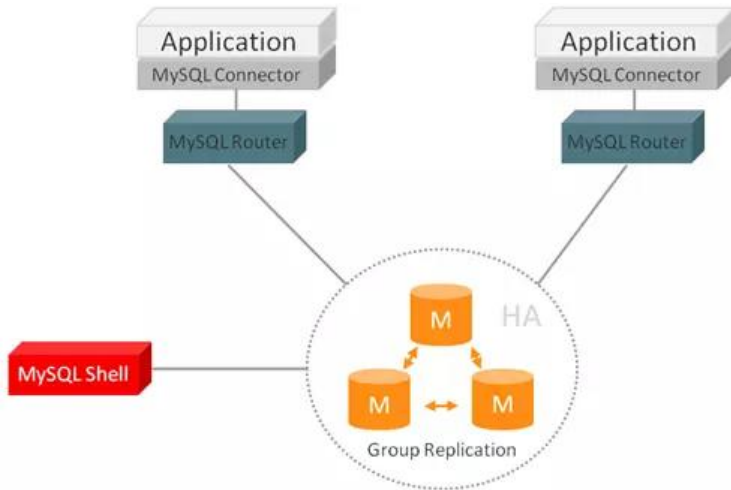


准备工作

te

- 本次共5台机器，服务器均采用Windows Server 2012，服务器地址如下：

172.16.220.10
172.16.220.11
172.16.220.12
172.16.220.13
172.16.220.14



下载MySQL、Shell、Router软件

<https://cdn.mysql.com/Downloads/MySQL-8.0/mysql-8.0.15-winx64.zip>

<https://cdn.mysql.com/Downloads/MySQL-Router/mysql-router-8.0.15-winx64.zip>

<https://cdn.mysql.com/Downloads/MySQL-Shell/mysql-shell-8.0.15-windows-x86-64bit.zip>

安装MySQL

首先配置my.ini，以下my.ini为172.16.2201.10上的配置：

```
# Other default tuning values
```

```
# MySQL Server Instance Configuration File
```

```
# -----
```

```
# Generated by the MySQL Server Instance Configuration Wizard
```

```
#
```

```
#
```

```
# Installation Instructions
```

```
# -----
```

```
#
```

```
# On Linux you can copy this file to /etc/my.cnf to set global options,
```

```
# mysql-data-dir/my.cnf to set server-specific options
```

```
# (@localstatedir@ for this installation) or to
```

```
# ~/.my.cnf to set user-specific options.
```

```
#
```

```
# On Windows you should keep this file in the installation directory
```

```
# of your server (e.g. C:\Program Files\MySQL\MySQL Server X.Y). To
```

```
# make sure the server reads the config file use the startup option
```

```
# "--defaults-file".
```

```
#
```

```
# To run the server from the command line, execute this in a
```

```
# command line shell, e.g.
```

```
# mysqld --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.ini"
```

```
#
```

```
# To install the server as a Windows service manually, execute this in a
```

```
# command line shell, e.g.
```

```
# mysqld --install MySQLXY --defaults-file="C:\Program Files\MySQL\MySQL Server X.Y\my.in
```

```
"
```

```

#
# And then execute this in a command line shell to start the server, e.g.
# net start MySQLXY
#
#
# Guidelines for editing this file
# -----
#
# In this file, you can use all long options that the program supports.
# If you want to know the options a program supports, start the program
# with the "--help" option.
#
# More detailed information about the individual options can also be
# found in the manual.
#
# For advice on how to change settings please see
# https://dev.mysql.com/doc/refman/8.0/en/server-configuration-defaults.html
#
#
# CLIENT SECTION
# -----
#
# The following options will be read by MySQL client applications.
# Note that only client applications shipped by MySQL are guaranteed
# to read this section. If you want your own MySQL client program to
# honor these values, you need to specify it as an option during the
# MySQL client library initialization.
#
[client]

# pipe=

socket=MYSQL

port=3346

# -----
# 1、替换PCSS为PCSS10, PCSS11, PCSS12, PCSS13, PCSS14等
# 2、server-id 为指定值, 需要不同
# 3、report-host 为指定值, 需要不同
# -----
[mysql]
no-beep

default-character-set=UTF-8

# SERVER SECTION
# -----
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
# server_type=1

```

[mysqld]

The next three options are mutually exclusive to SERVER_PORT below.

skip-networking
enable-named-pipe
shared-memory

shared-memory-base-name=MYSQL

The Pipe the MySQL Server will use
socket=MYSQL

The TCP/IP Port the MySQL Server will listen on
port=3346

Path to installation directory. All paths are usually resolved relative to this.
basedir="C:/Program Files/MySQL/MySQL Server 8.0/"

Path to the database root
basedir=D:/PhpWamp/phpwamp/server/mysql-8.1
datadir=D:/PhpWamp/phpwamp/server/mysql-8.1/data

The default character set that will be used when a new schema or table is
created and no character set is defined
character-set-server=

The default authentication plugin to be used when connecting to the server
default_authentication_plugin=mysql_native_password

The default storage engine that will be used when create new tables when
default-storage-engine=INNODB

Set the SQL mode to strict
sql-mode="STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION"

General and Slow logging.
log-output=FILE
general-log=0
general_log_file="PCSS.log"
slow-query-log=1
slow_query_log_file="PCSS-slow.log"
long_query_time=5

report-host=172.16.220.10
report-password=123456
report-port=3346
report-user=root

relay-log
relay-log=PCSS-relay-bin

Binary Logging.
log-bin="PCSS-bin"

```
# Error Logging.
log-error="PCSS.err"

# Server Id.
server-id=310

# Indicates how table and database names are stored on disk and used in MySQL.
# Value = 0: Table and database names are stored on disk using the lettercase specified in the
# CREATE TABLE or CREATE DATABASE statement. Name comparisons are case sensitive.
# You should not set this variable to 0 if you are running MySQL on a system that has
# case-insensitive file names (such as Windows or macOS).
# Value = 1: Table names are stored in lowercase on disk and name comparisons are not
# case-sensitive. MySQL converts all table names to lowercase on storage and lookup.
# This behavior also applies to database names and table aliases.
# Value = 3, Table and database names are stored on disk using the lettercase specified in the
# CREATE TABLE or CREATE DATABASE statement, but MySQL converts them to lowercase on
# lookup. Name comparisons are not case sensitive. This works only on file systems
# that are not case-sensitive! InnoDB table names and view names are stored in
# lowercase, as for Value = 1.
# NOTE: lower_case_table_names can only be configured when initializing the server.
# Changing the lower_case_table_names setting after the server is initialized is prohibited.
lower_case_table_names=1

# Secure File Priv.
secure-file-priv="D:/PhpWamp/phpwamp/server/mysql-8.1/uploads"

# The maximum amount of concurrent sessions the MySQL server will
# allow. One of these connections will be reserved for a user with
# SUPER privileges to allow the administrator to login even if the
# connection limit has been reached.
max_connections=2000

# The number of open tables for all threads. Increasing this value
# increases the number of file descriptors that mysqld requires.
# Therefore you have to make sure to set the amount of open files
# allowed to at least 4096 in the variable "open-files-limit" in
# section [mysqld_safe]
table_open_cache=200000

# Maximum size for internal (in-memory) temporary tables. If a table
# grows larger than this value, it is automatically converted to disk
# based table This limitation is for a single table. There can be many
# of them.
tmp_table_size=2048M

# How many threads we should keep in a cache for reuse. When a client
# disconnects, the client's threads are put in the cache if there aren't
# more than thread_cache_size threads from before. This greatly reduces
# the amount of thread creations needed if you have a lot of new
# connections. (Normally this doesn't give a notable performance
# improvement if you have a good thread implementation.)
```

```
thread_cache_size=1000
```

```
**** MyISAM Specific options
```

```
# The maximum size of the temporary file MySQL is allowed to use while  
# recreating the index (during REPAIR, ALTER TABLE or LOAD DATA INFILE.  
# If the file-size would be bigger than this, the index will be created  
# through the key cache (which is slower).
```

```
mysam_max_sort_file_size=100G
```

```
# If the temporary file used for fast index creation would be bigger  
# than using the key cache by the amount specified here, then prefer the  
# key cache method. This is mainly used to force long character keys in  
# large tables to use the slower key cache method to create the index.
```

```
mysam_sort_buffer_size=512M
```

```
# Size of the Key Buffer, used to cache index blocks for MyISAM tables.  
# Do not set it larger than 30% of your available memory, as some memory  
# is also required by the OS to cache rows. Even if you're not using  
# MyISAM tables, you should still set it to 8-64M as it will also be  
# used for internal temporary disk tables.
```

```
key_buffer_size=1024M
```

```
# Size of the buffer used for doing full table scans of MyISAM tables.  
# Allocated per thread, if a full scan is needed.
```

```
read_buffer_size=512M
```

```
read_rnd_buffer_size=256M
```

```
**** INNODB Specific options ****
```

```
# innodb_data_home_dir=
```

```
# Use this option if you have a MySQL server with InnoDB support enabled  
# but you do not plan to use it. This will save memory and disk space  
# and speed up some things.
```

```
# skip-innodb
```

```
# If set to 1, InnoDB will flush (fsync) the transaction logs to the  
# disk at each commit, which offers full ACID behavior. If you are  
# willing to compromise this safety, and you are running small  
# transactions, you may set this to 0 or 2 to reduce disk I/O to the  
# logs. Value 0 means that the log is only written to the log file and  
# the log file flushed to disk approximately once per second. Value 2  
# means the log is written to the log file at each commit, but the log  
# file is only flushed to disk approximately once per second.
```

```
innodb_flush_log_at_trx_commit=1
```

```
# The size of the buffer InnoDB uses for buffering log data. As soon as  
# it is full, InnoDB will have to flush it to disk. As it is flushed  
# once per second anyway, it does not make sense to have it very large  
# (even with long transactions).
```

```
innodb_log_buffer_size=1024M
```

```
# InnoDB, unlike MyISAM, uses a buffer pool to cache both indexes and  
# row data. The bigger you set this the less disk I/O is needed to
```

```

# access data in tables. On a dedicated database server you may set this
# parameter up to 80% of the machine physical memory size. Do not set it
# too large, though, because competition of the physical memory may
# cause paging in the operating system. Note that on 32bit systems you
# might be limited to 2-3.5G of user level memory per process, so do not
# set it too high.
innodb_buffer_pool_size=1024M

# Size of each log file in a log group. You should set the combined size
# of log files to about 25%-100% of your buffer pool size to avoid
# unneeded buffer pool flush activity on log file overwrite. However,
# note that a larger logfile size will increase the time needed for the
# recovery process.
innodb_log_file_size=48M

# Number of threads allowed inside the InnoDB kernel. The optimal value
# depends highly on the application, hardware as well as the OS
# scheduler properties. A too high value may lead to thread thrashing.
innodb_thread_concurrency=17

# The increment size (in MB) for extending the size of an auto-extend InnoDB system tablesp
ce file when it becomes full.
innodb_autoextend_increment=64

# The number of regions that the InnoDB buffer pool is divided into.
# For systems with buffer pools in the multi-gigabyte range, dividing the buffer pool into sep
arate instances can improve concurrency,
# by reducing contention as different threads read and write to cached pages.
innodb_buffer_pool_instances=8

# Determines the number of threads that can enter InnoDB concurrently.
innodb_concurrency_tickets=500000

# Specifies how long in milliseconds (ms) a block inserted into the old sublist must stay there
fter its first access before
# it can be moved to the new sublist.
innodb_old_blocks_time=1000

# It specifies the maximum number of .ibd files that MySQL can keep open at one time. The m
nimum value is 10.
innodb_open_files=30000

# When this variable is enabled, InnoDB updates statistics during metadata statements.
innodb_stats_on_metadata=0

# When innodb_file_per_table is enabled (the default in 5.6.6 and higher), InnoDB stores the d
ta and indexes for each newly created table
# in a separate .ibd file, rather than in the system tablespace.
innodb_file_per_table=1

# Use the following list of values: 0 for crc32, 1 for strict_crc32, 2 for innodb, 3 for strict_innod
, 4 for none, 5 for strict_none.
innodb_checksum_algorithm=0

```



```
# The number of outstanding connection requests MySQL can have.
# This option is useful when the main MySQL thread gets many connection requests in a very short time.
# It then takes some time (although very little) for the main thread to check the connection and start a new thread.
# The back_log value indicates how many requests can be stacked during this short time before MySQL momentarily
# stops answering new requests.
# You need to increase this only if you expect a large number of connections in a short period of time.
back_log=80

# If this is set to a nonzero value, all tables are closed every flush_time seconds to free up resources and
# synchronize unflushed data to disk.
# This option is best used only on systems with minimal resources.
flush_time=0

# The minimum size of the buffer that is used for plain index scans, range index scans, and joins that do not use
# indexes and thus perform full table scans.
join_buffer_size=256K

# The maximum size of one packet or any generated or intermediate string, or any parameter sent by the
# mysql_stmt_send_long_data() C API function.
max_allowed_packet=1024M

# If more than this many successive connection requests from a host are interrupted without successful connection,
# the server blocks that host from performing further connections.
max_connect_errors=10000

# Changes the number of file descriptors available to mysqld.
# You should try increasing the value of this option if mysqld gives you the error "Too many open files".
open_files_limit=41610

# If you see many sort_merge_passes per second in SHOW GLOBAL STATUS output, you can consider increasing the
# sort_buffer_size value to speed up ORDER BY or GROUP BY operations that cannot be improved with query optimization
# or improved indexing.
sort_buffer_size=256M

# The number of table definitions (from .frm files) that can be stored in the definition cache.
# If you use a large number of tables, you can create a large table definition cache to speed up opening of tables.
# The table definition cache takes less space and does not use file descriptors, unlike the normal table cache.
# The minimum and default values are both 400.
table_definition_cache=1400

# Specify the maximum size of a row-based binary log event, in bytes.
```

```

# Rows are grouped into events smaller than this size if possible. The value should be a multiple of 256.
binlog_row_event_max_size=8M

# If the value of this variable is greater than 0, a replication slave synchronizes its master.info file to disk.
# (using fdatsync()) after every sync_master_info events.
sync_master_info=10000

# If the value of this variable is greater than 0, the MySQL server synchronizes its relay log to disk.
# (using fdatsync()) after every sync_relay_log writes to the relay log.
sync_relay_log=10000

# If the value of this variable is greater than 0, a replication slave synchronizes its relay-log.info file to disk.
# (using fdatsync()) after every sync_relay_log_info transactions.
sync_relay_log_info=10000

# Load mysql plugins at start."plugin_x ; plugin_y".
# plugin_load

# The TCP/IP Port the MySQL Server X Protocol will listen on.
loose_mysqlx_port=33460

# https://dev.mysql.com/doc/refman/8.0/en/group-replication-configuring-instances.html
gtid_mode=ON
enforce_gtid_consistency=ON
binlog_checksum=NONE

```

其它电脑上的配置，需要修改`server-id`，`report-host`，其它参数与上面相同。

配置MySQL数据库

```

D:/PhpWamp/phpwamp/server/mysql-8.1/bin/mysql.exe --defaults-file=D:/PhpWamp/phpwamp/server/mysql-8.1/my.ini --initialize-insecure --user=root --basedir=D:/PhpWamp/phpwamp/server/mysql-8.1 --datadir=D:/PhpWamp/phpwamp/server/mysql-8.1/data

```

进入MySQL后，修改密码及配置权限：

```

RENAME USER `root`@`localhost` TO `root`@`%`;

ALTER USER 'root'@'%' IDENTIFIED BY '123456';

SET SQL_LOG_BIN=0;
CREATE USER rpl_user@'%' IDENTIFIED BY '123456';
GRANT REPLICATION SLAVE ON *.* TO rpl_user@'%';
FLUSH PRIVILEGES;
SET SQL_LOG_BIN=1;

CHANGE MASTER TO MASTER_USER='rpl_user', MASTER_PASSWORD='123456' FOR CHANNEL 'group_replication_recovery';

```

安装配置MySQL服务

```
D:/PhpWamp/phpwamp/server/mysql-8.1/bin/mysqlD.exe --install .MySQL.InnoDB.Cluster --
efaults-file=D:/PhpWamp/phpwamp/server/mysql-8.1/my.ini
```

配置Shell

打开mysqlsh.exe，运行以下命令，5台服务器均需要运行，注意修改连接地址：

```
# 检查mysql 配置文件（5台主机都要操作此步骤）
```

```
dba.checkInstanceConfiguration('root@172.16.220.10:3346')
```

```
+-----+-----+-----+-----+
-----+
| Variable                | Current Value | Required Value | Note                                |
+-----+-----+-----+-----+
| binlog_checksum        | CRC32         | NONE           | Update the server variable or restar
the server |
| binlog_format          | MIXED        | ROW           | Update the server variable or restart t
e server |
| enforce_gtid_consistency | OFF          | ON            | Restart the server
|
| gtid_mode              | OFF          | ON            | Restart the server                    |
| log_slave_updates      | 0            | ON            | Restart the server                    |
| master_info_repository | FILE         | TABLE        | Restart the server                    |
| relay_log_info_repository | FILE        | TABLE        | Restart the server
|
| transaction_write_set_extraction | OFF          | XXHASH64      | Restart the server
|
+-----+-----+-----+-----+
-----+
```

```
# 修复mysql 配置文件，必须用 root(5台主机都要操作此步骤)
```

```
dba.configureLocalInstance('root@172.16.220.10:3346')
```

```
Please provide the password for 'root@172.16.220.10:3346':
```

```
Detecting the configuration file...
```

```
Found configuration file at standard location: /etc/my.cnf
```

```
Do you want to modify this file? [Y|n]: [Y|n]: Y
```

```
# 重启mysql
```

```
# 重新检查（5台主机都要操作此步骤）
```

```
dba.checkInstanceConfiguration('root@db1:3306')
```

```
Please provide the password for 'root@db1:3306':
```

```
Validating instance...
```

```
The instance 'db1:3306' is valid for Cluster usage
```

```
{
  "status": "ok"
}
```

其中一台电脑上运行：

```
# 连接MySQL
shell.connect('root@172.16.220.10:3346')
```

```
# 创建集群
# var cluster = dba.createCluster('mycluster')
var cluster = dba.createCluster('mycluster', {multiPrimary: true, force: true})
```

A new InnoDB cluster will be created on instance 'root@172.16.220.10:3346'.

```
Creating InnoDB cluster 'main' on 'root@172.16.220.10:3346'...
Adding Seed Instance...
```

Cluster successfully created. Use Cluster.addInstance() to add MySQL instances.
At least 3 instances are needed for the cluster to be able to withstand up to one server failure.

```
# 添加子节点
cluster.addInstance('root@172.16.220.11:3346')
cluster.addInstance('root@172.16.220.12:3346')
```

```
cluster.addInstance('root@172.16.220.13:3346')
cluster.addInstance('root@172.16.220.14:3346')
```

```
# 查看节点信息
cluster.status()
```

```
# 查看基本信息
cluster.describe();
```

```
# 退出之后, 再查看节点信息
var cluster = dba.getCluster();
cluster.status();
```

```
MySQL [172.16.220.10 ssl] JS> cluster.status()
{
  "clusterName": "mycluster",
  "defaultReplicaSet": {
    "name": "default",
    "ssl": "REQUIRED",
    "status": "OK",
    "statusText": "Cluster is ONLINE and can tolerate up to 2 failures.",
    "topology": {
      "172.16.220.10:3346": {
        "address": "172.16.220.10:3346",
        "mode": "R/W",
        "readReplicas": {},
        "role": "HA",
        "status": "ONLINE"
      },
      "172.16.220.11:3346": {
        "address": "172.16.220.11:3346",
        "mode": "R/W",
        "readReplicas": {},
        "role": "HA",
```

```

    "status": "ONLINE"
  },
  "172.16.220.12:3346": {
    "address": "172.16.220.12:3346",
    "mode": "R/W",
    "readReplicas": {},
    "role": "HA",
    "status": "ONLINE"
  },
  "172.16.220.13:3346": {
    "address": "172.16.220.13:3346",
    "mode": "R/W",
    "readReplicas": {},
    "role": "HA",
    "status": "ONLINE"
  },
  "172.16.220.14:3346": {
    "address": "172.16.220.14:3346",
    "mode": "R/W",
    "readReplicas": {},
    "role": "HA",
    "status": "ONLINE"
  }
},
"topologyMode": "Multi-Primary"
},
"groupInformationSourceMember": "172.16.220.10:3346"
}

```

查看最终状态

```

mysql> SELECT * FROM performance_schema.replication_group_members;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| CHANNEL_NAME          | MEMBER_ID          | MEMBER_HOST | MEMBER_PORT |
| MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| group_replication_applier | 6446e1da-4632-11e9-bc0a-4ccc6a123cdf | 172.16.220.10 | 3
46 | ONLINE      | PRIMARY      | 8.0.15      |
| group_replication_applier | 87112871-4613-11e9-a64e-4ccc6a123c66 | 172.16.220.12 | 3
46 | ONLINE      | PRIMARY      | 8.0.15      |
| group_replication_applier | 9020fd22-4612-11e9-bcbb-4ccc6a124138 | 172.16.220.13 | 3
46 | ONLINE      | PRIMARY      | 8.0.15      |
| group_replication_applier | a7785098-4633-11e9-8af6-4ccc6a123be0 | 172.16.220.11 | 3
46 | ONLINE      | PRIMARY      | 8.0.15      |
| group_replication_applier | f1883c54-4606-11e9-8e4a-4ccc6a1242e6 | 172.16.220.14 | 3
46 | ONLINE      | PRIMARY      | 8.0.15      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)

```

配置Router

```
D:/PhpWamp/phpwamp/server/mysql-8.1/bin/mysqlrouter.exe --bootstrap root@172.16.220.0:3346
```

Please enter MySQL password for root:

WARNING: The MySQL server does not have SSL configured and metadata used by the router may be transmitted unencrypted.

Bootstrapping system MySQL Router instance...
MySQL Router has now been configured for the InnoDB cluster 'main'.

The following connection information can be used to connect to the cluster.

Classic MySQL protocol connections to cluster 'main':

- Read/Write Connections: localhost:6446 读写
- Read/Only Connections: localhost:6447 只读

X protocol connections to cluster 'main':

- Read/Write Connections: localhost:64460
- Read/Only Connections: localhost:64470

Existing configurations backed up to /etc/mysqlrouter/mysqlrouter.conf.bak

```
# File automatically generated during MySQL Router bootstrap
```

```
[DEFAULT]
```

```
name=system
```

```
keyring_path=D:/PhpWamp/phpwamp/server/mysql-8.1/var/lib/mysqlrouter/keyring
```

```
master_key_path=D:/PhpWamp/phpwamp/server/mysql-8.1/mysqlrouter.key
```

```
connect_timeout=15
```

```
read_timeout=30
```

```
dynamic_state=D:/PhpWamp/phpwamp/server/mysql-8.1/var/lib/mysqlrouter/state.json
```

```
[logger]
```

```
level = INFO
```

```
[metadata_cache:mycluster]
```

```
router_id=6
```

```
user=mysql_router6_g10rrazm7cya
```

```
metadata_cluster=mycluster
```

```
ttl=0.5
```

```
[routing:mycluster_default_rw]
```

```
bind_address=0.0.0.0
```

```
bind_port=6446
```

```
destinations=metadata-cache://mycluster/default?role=PRIMARY
```

```
routing_strategy=round-robin
```

```
protocol=classic
```

```
[routing:mycluster_default_x_rw]
```

```
bind_address=0.0.0.0
```

```
bind_port=64460
```

```
destinations=metadata-cache://mycluster/default?role=PRIMARY
```

```
routing_strategy=round-robin
```

protocol=x

故障模拟

在Java中的连接配置

```
# 多主模式: loadbalance
jdbc.url=jdbc:mysql:loadbalance://172.16.220.10:3346,172.16.220.11:3346,172.16.220.12:3346,172.16.220.13:3346,172.16.220.14:3346/eis?loadBalanceBlacklistTimeout=5000&useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true
```

```
# 主从模式:replication
jdbc.url=jdbc:mysql:replication://172.16.220.10:3346,172.16.220.11:3346,172.16.220.12:3346,172.16.220.13:3346,172.16.220.14:3346/eis?useUnicode=true&characterEncoding=utf-8&allowMultiQueries=true
```

备注信息

```
var cluster = dba.rebootClusterFromCompleteOutage(); # 如果需要重启集群
```

```
dba.dropMetadataSchema(); # 删除schema
```

```
cluster.rejoinInstance("root@172.16.220.10:3346"); # 重新加入节点, 我本地测试的时候发现rejoin一直无效
```

```
cluster.dissolve({force:true}); # 解散集群
```

```
cluster.addInstance("root@172.16.220.10:3346"); # 增加节点
```

```
cluster.removeInstance("root@172.16.220.10:3346"); # 删除节点
```

```
cluster.removeInstance('root@172.16.220.10:3346',{force:true}); # 强制删除节点
```

```
cluster.describe(); # 集群描述
```

```
# 集群节点状态
```

```
* ONLINE: The instance is online and participating in the cluster.
```

```
* OFFLINE: The instance has lost connection to the other instances.
```

```
* RECOVERING: The instance is attempting to synchronize with the cluster by retrieving transactions it needs before it can become an ONLINE member.
```

```
* UNREACHABLE: The instance has lost communication with the cluster.
```

```
* ERROR: The instance has encountered an error during the recovery phase or while applying transaction.
```

```
# 集群有哪些状态
```

```
* OK – 所有节点处于online状态, 有冗余节点。
```

```
* OK_PARTIAL – 有节点不可用, 但仍有冗余节点。
```

```
* OK_NO_TOLERANCE – 有足够的online节点, 但没有冗余, 例如: 两个节点的Cluster, 其中一个了, 集群就不可用了。
```

```
* NO_QUORUM – 有节点处于online状态, 但达不到法定节点数, 此状态下Cluster无法写入, 只能取。
```

```
* UNKNOWN – 不是online或recovering状态, 尝试连接其他实例查看状态。
```

* UNAVAILABLE – 组内节点全是offline状态, 但实例在运行, 可能实例刚重启还没加入Cluster

如果失败原因如下:

```
ERROR: Group Replication join failed.
```

```
ERROR: Error joining instance to cluster: '172.16.220.10:3346' - Query failed. 3092 (HY000): The server is not configured properly to be an active member of the group. Please see more details on error log.. Query: START group_replication (RuntimeError)
after I run cluster.checkInstanceState('root@172.16.220.10:3346');
```

The instance '172.16.220.10:3346' is invalid for the cluster.
The instance contains additional transactions in relation to the cluster.

```
{
  "reason": "diverged",
  "state": "error"
}
```

I know it is because the inconsistent state in different db. But I searched the internet but there is no documentation about how to resolve conflict in innodb cluster. Any advice would be helpful!

I ended up partially wipe the database and resync it again by `reset master` and `drop database mysql_innodb_cluster_metadata`.

所有节点都重启了, 重新加入

```
mysqlsh --uri root@172.16.220.10:3346
mysql-js> var cluster = dba.rebootClusterFromCompleteOutage();
```

Reconfiguring the default cluster from complete outage...

The instance '172.16.220.10:3346' was part of the cluster configuration.
Would you like to rejoin it to the cluster? [y|N]: y

The instance '172.16.220.10:3346' was part of the cluster configuration.
Would you like to rejoin it to the cluster? [y|N]: y

The cluster was successfully rebooted.

#重启节点后, 需要手动加入

```
cluster.rejoinInstance('root@172.16.220.10:3346')
```

```
"172.16.220.10:3346": {
  "address": "172.16.220.10:3346",
  "mode": "R/O",
  "readReplicas": {},
  "role": "HA",
  "status": "(MISSING)"
}
```

The instance '172.16.220.10:3346' was successfully added to the MySQL Cluster.

如果出现了 "status": "NO_QUORUM" 执行修复, 重新加入

```
cluster.forceQuorumUsingPartitionOf("172.16.220.10:3346")
```

```
mysql-js> cluster.rejoinInstance('root@172.16.220.10:3346')
```



```
# 如何重置实验环境，主节点：  
dba.dropMetadataSchema() mysqlshell 清空集群  
mysql> stop group_replication;  
mysql> reset master;          (清空日志，确保和从库的表没有冲突奥，)  
mysql> reset slave  
  
# 其他节点，主要清理和主库的主从信息，确保主库和从库的表没有冲突奥  
mysql> stop group_replication;  
mysql> reset master;  
mysql> reset slave
```