# Hướng dẫn cài đặt và cấu hình

**Percona XtraDB Cluster 5.7.23-31**

# Chuẩn bị trước khi cài đặt.

# Download Percona-XtraDB-Cluster-5.7.23-31.31-r436-el7-x86\_64-bundle.tar :

# <https://www.percona.com/downloads/Percona-XtraDB-Cluster-56/>

# Download Percona-XtraBackup-2.4.12-r170eb8c-el7-x86\_64-bundle.tar

# <https://www.percona.com/downloads/XtraBackup/>

# Download Percona GPG Key:

# <https://www.percona.com/downloads/RPM-GPG-KEY-percona>

# Download [percona-release-0.1-6.noarch.rpm](https://www.percona.com/redir/downloads/percona-release/redhat/percona-release-0.1-3.noarch.rpm)

# https://www.percona.com/redir/downloads/percona-release/redhat/0.1-6/percona-release-0.1-6.noarch.rpm

# Bắt đầu cài đặt.

# Mở port trong iptables 3306, 4444, 4567 and 4568 , disabled selinux.

# Dùng lệnh : # vi /etc/sysconfig/iptables

# Sau đó thêm đoạn sau :

# ########### Connect DB ##############################

# -A OUTPUT -m state --state NEW -d 192.168.0.0/24 -m tcp -p tcp --dport 3306 -j ACCEPT

# -A INPUT -m state --state NEW -s 192.168.0.0/24 -m tcp -p tcp --dport 3306 -j ACCEPT

# service iptables restart

# Cài đặt :

# Bước 1: Cài gói Percona-XtraBackup-2.4.12-r170eb8c-el7-x86\_64-bundle.tar

# Tiến hành giải nén :

# # tar -xvf Percona-XtraBackup-2.4.12-r170eb8c-el7-x86\_64-bundle.tar

# 

# Trước khi cài file này cần cài những packages đi kèm :

# A, Mount đến file iso ổ cài đặt Centos 7.5

# # mount /dev/dvdrom /mnt/dvdrom

# Vào đường dẫn # cd /mnt/dvdrom/Packages và tìm 2 gói

# perl-DBD-MySQL

# perl-DBI

# perl-Digest-MD5

# Để cài đặt được perl-DBI cần cài các gói đi kèm sau :

# perl-Compress-Raw-Bzip2-2.061-3.el7.x86\_64.rpm

# perl-Compress-Raw-Zlib-2.061-4.el7.x86\_64.rpm

# perl-IO-Compress-2.061-2.el7.noarch.rpm

# perl-Net-Daemon-0.48-5.el7.noarch.rpm

# perl-PlRPC-0.2020-14.el7.noarch.rpm

# Sửa repository trong file /etc/yum.repos.d/CentOS-Media.repo để trỏ đường dẫn về file iso setup CentOS như ở trên là /mnt/dvdrom dùng lệnh sau và tiến hành cài gói # vi /etc/yum.repos.d/CentOS-Media.repo

# 

# # yum --disablerepo=\\* --enablerepo=c7-media install perl-DBI perl-DBD-MySQL

# 

# #yum --disablerepo=\\* --enablerepo=c7-media install perl-Digest-MD5

# 

# # yum --disablerepo=\\* --enablerepo=c7-media install perl-Test-Simple perl-Env-1.04

# 

# + cài đặt packages libev-4.15-3.el7.x86\_64.rpm cung cấp thư viện libev.so.4()(64bit)

# <https://downloads.mariadb.com/staging/Tools/rhel/7/x86_64/rpms/libev-4.15-6.el7.x86_64.rpm>

# 

# Cuối cùng cài đặt các gói Percona-XtraBackup-2.4.12 :

# # rpm -ivh percona-xtrabackup-24-2.4.12-1.el7.x86\_64.rpm

# # rpm -ivh percona-xtrabackup-24-debuginfo-2.4.12-1.el7.x86\_64.rpm

# 

# Bước 2 : Cài đặt gói Percona-XtraDB-Cluster-5.7.23-31.31-r436-el7-x86\_64-bundle.tar

# Giải nén gói ra

# # tar –xvf Percona-XtraDB-Cluster-5.7.23-31.31-r436-el7-x86\_64-bundle.tar

# 

# B, Download file socat.rpm về cài theo đường dẫn

# <http://mirror.centos.org/centos/7/os/x86_64/Packages/socat-1.7.3.2-2.el7.x86_64.rpm>

# 

# C, Download qpress-11-1.el7.x86\_64.rpm cho Percona-XtraDB-server

# <https://downloads.mariadb.com/staging/Tools/rhel/7/x86_64/rpms/qpress-11-1.el7.x86_64.rpm>

# 

# Cài đặt [percona-release-0.1-6.noarch.rpm](https://www.percona.com/redir/downloads/percona-release/redhat/percona-release-0.1-3.noarch.rpm)

# 

# Remove mariadb-libs có sẵn trên CentOS 7.5

# 

# D, Cài đặt Percona-XtraDB-Cluster-client-57-5.7.23-31.31.1.el7.x86\_64.rpm

# 

# F, Cài đặt Percona-XtraDB-Cluster-shared-57-5.7.23-31.31.1.el7.x86\_64.rpm

# 

# G, Cài đặt Percona-XtraDB-Cluster-server-57-5.7.23-31.31.1.el7.x86\_64.rpm

# 

# Tiếp đó cài những file còn lại trong file Percona XtraDB Cluster 5.7.tar

$ rpm -ivh Percona-XtraDB-Cluster-57-5.7.23-31.31.1.el7.x86\_64.rpm

$ rpm -ivh Percona-XtraDB-Cluster-57-debuginfo-5.7.23-31.31.1.el7.x86\_64.rpm

$ rpm -ivh Percona-XtraDB-Cluster-devel-57-5.7.23-31.31.1.el7.x86\_64.rpm

$ rpm -ivh Percona-XtraDB-Cluster-garbd-57-5.7.23-31.31.1.el7.x86\_64.rpm

$ rpm -ivh Percona-XtraDB-Cluster-shared-compat-57-5.7.23-31.31.1.el7.x86\_64.rpm

* Sau khi cài xong start mysql lên để đổi pass trước khi config

1. Start the Percona XtraDB Cluster server:

$ systemctl start mysql.service

1. Copy the automatically generated temporary password for the superuser account:

$ sudo grep 'temporary password' /var/log/mysqld.log

1. Use this password to log in as root:

$ mysql -u root -p

1. Change the password for the superuser account and log out. For example:

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'TestRoot123';

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

1. Stop the mysql service:

$ systemctl stop mysql.service

If you are running Red Hat or CentOS, add the following configuration variables to /etc/my.cnf on the first node:

#

# The Percona XtraDB Cluster 5.7 configuration file.

#

#

# \* IMPORTANT: Additional settings that can override those from this file!

# The files must end with '.cnf', otherwise they'll be ignored.

# Please make any edits and changes to the appropriate sectional files

# included below.

#

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

user=mysql

wsrep\_provider=/usr/lib64/galera3/libgalera\_smm.so

wsrep\_cluster\_name=TESTDB

wsrep\_cluster\_address=gcomm://192.168.0.34,192.168.0.33,192.168.0.32

wsrep\_node\_name=thuantv # tự định danh, nếu ko có sẽ tự động lấy hostname

wsrep\_node\_address=192.168.0.34

wsrep\_sst\_method=xtrabackup-v2

wsrep\_sst\_auth=sstuser:passw0rd

pxc\_strict\_mode=ENFORCING

binlog\_format=ROW

default\_storage\_engine=InnoDB

innodb\_autoinc\_lock\_mode=2

#### Tunning MySQL ##############

innodb\_buffer\_pool\_size = 2048M

innodb\_log\_file\_size = 512M

max\_connections=500

innodb\_file\_per\_table= ON

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_flush\_method=O\_DIRECT

innodb\_log\_buffer\_size = 24M

innodb\_read\_io\_threads=8

innodb\_write\_io\_threads=8

innodb\_buffer\_pool\_instances=8

query\_cache\_size = 100M

slave-skip-errors = 1062

#slave-skip-errors=all

slave-skip-errors=1054

slave-skip-errors=1051

slave-skip-errors=1146

slave-skip-errors=1452

#####\_\_Chong mat dong bo DB\_\_\_

sync\_binlog = 0

##############################

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

[xtrabackup]

target\_dir = /opt/backupdb/mysql

!includedir /etc/my.cnf.d/

!includedir /etc/percona-xtradb-cluster.conf.d/

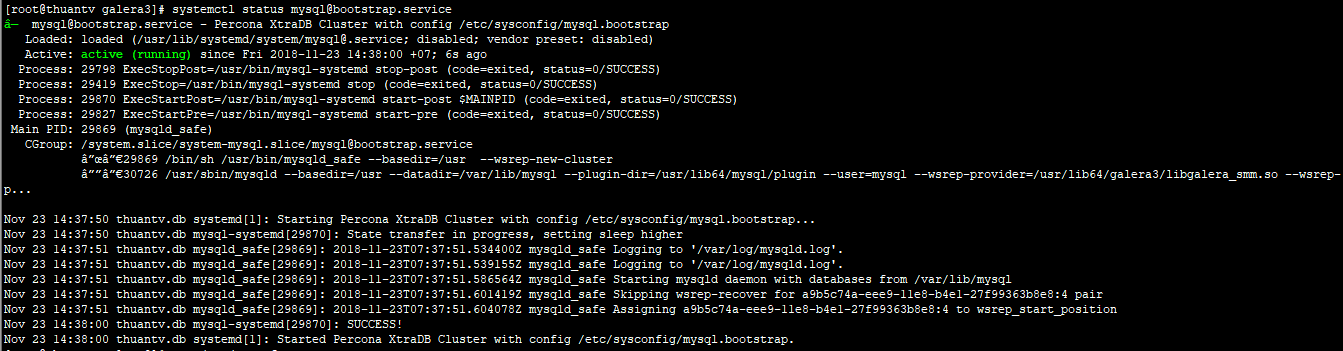
Use the same configuration for the second and third nodes, except the wsrep\_node\_name and wsrep\_node\_address variables:

* For the second node:
* wsrep\_node\_name=node2
* wsrep\_node\_address=192.168.0.33
* For the third node:
* wsrep\_node\_name=node3

wsrep\_node\_address=192.168.0.32

On RHEL or CentOS 7, use the following bootstrap command:

[root@thuantv ~]# systemctl start mysql@bootstrap.service



When you start the node using the previous command, it runs in bootstrap mode with wsrep\_cluster\_address=gcomm://. This tells the node to initialize the cluster with [wsrep\_cluster\_conf\_id](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/wsrep-status-index.html" \l "wsrep_cluster_conf_id" \o "wsrep_cluster_conf_id) set to 1. After you [add other nodes](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/add-node.html#add-node) to the cluster, you can then restart this node as normal, and it will use standard configuration again.

To make sure that the cluster has been initialized, run the following:

[root@thuantv ~]# mysql –u root –p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 12

Server version: 5.7.23-23-57-log Percona XtraDB Cluster (GPL), Release rel23, Revision 969c8b5, WSREP version 31.31, wsrep\_31.31

Copyright (c) 2009-2018 Percona LLC and/or its affiliates

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql > show status like 'wsrep%';

+----------------------------+--------------------------------------+

| Variable\_name | Value |

+----------------------------+--------------------------------------+

| wsrep\_local\_state\_uuid | c2883338-834d-11e2-0800-03c9c68e41ec |

| ... | ... |

| wsrep\_local\_state | 4 |

| wsrep\_local\_state\_comment | Synced |

| ... | ... |

| wsrep\_cluster\_size | 1 |

| wsrep\_cluster\_status | Primary |

| wsrep\_connected | ON |

| ... | ... |

| wsrep\_ready | ON |

+----------------------------+--------------------------------------+

40 rows in set (0.01 sec)

The previous output shows that the cluster size is 1 node, it is the primary component, the node is in Synced state, it is fully connected and ready for write-set replication.

Before [adding other nodes](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/add-node.html#add-node) to your new cluster, create a user for [SST](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/glossary.html#term-sst) and provide necessary privileges for it. The credentials must match those specified when [Configuring Nodes for Write-Set Replication](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/configure.html#configure).

mysql> CREATE USER 'sstuser'@'localhost' IDENTIFIED BY 'passw0rd';

mysql> GRANT RELOAD, LOCK TABLES, PROCESS, REPLICATION CLIENT ON \*.\* TO 'sstuser'@'localhost';

mysql> FLUSH PRIVILEGES;

## **Starting the Second Node**

Start the second node using the following command:

[root@node2 ~]# systemctl start mysql.service

After the server starts, it should receive [SST](https://www.percona.com/doc/percona-xtradb-cluster/LATEST/glossary.html#term-sst) automatically.

To check the status of the second node, run the following:

mysql@pxc2> show status like 'wsrep%';

+----------------------------+--------------------------------------+

| Variable\_name | Value |

+----------------------------+--------------------------------------+

| wsrep\_local\_state\_uuid | c2883338-834d-11e2-0800-03c9c68e41ec |

| ... | ... |

| wsrep\_local\_state | 4 |

| wsrep\_local\_state\_comment | Synced |

| ... | ... |

| wsrep\_cluster\_size | 2 |

| wsrep\_cluster\_status | Primary |

| wsrep\_connected | ON |

| ... | ... |

| wsrep\_ready | ON |

+----------------------------+--------------------------------------+

40 rows in set (0.01 sec)

Previous output shows that the new node has been successfully added to the cluster. Cluster size is now 2 nodes, it is the primary component, and it is fully connected and ready to receive write-set replication.

If the state of the second node is Synced as in the previous example, then the node received full SST, is synchronized with the cluster, and you can proceed to add the next node.

Note

If the state of the node is Joiner, it means that SST hasn’t finished. Do not add new nodes until all others are in Synced state.

## **Starting the Third Node**

To add the third node, start it as usual:

[root@node3 ~]# systemctl start mysql.service

To check the status of the third node, run the following:

mysql@pxc3> show status like 'wsrep%';

+----------------------------+--------------------------------------+

| Variable\_name | Value |

+----------------------------+--------------------------------------+

| wsrep\_local\_state\_uuid | c2883338-834d-11e2-0800-03c9c68e41ec |

| ... | ... |

| wsrep\_local\_state | 4 |

| wsrep\_local\_state\_comment | Synced |

| ... | ... |

| wsrep\_cluster\_size | 3 |

| wsrep\_cluster\_status | Primary |

| wsrep\_connected | ON |

| ... | ... |

| wsrep\_ready | ON |

+----------------------------+--------------------------------------+

40 rows in set (0.01 sec)

Previous output shows that the new node has been successfully added to the cluster. Cluster size is now 3 nodes, it is the primary component, and it is fully connected and ready to receive write-set replication.

# Verifying Replication

Use the following procedure to verify replication by creating a new database on the second node, creating a table for that database on the third node, and adding some records to the table on the first node.

1. Create a new database on the second node:

mysql@node2> CREATE DATABASE percona;

Query OK, 1 row affected (0.01 sec)

1. Create a table on the third node:

mysql@node3> USE percona;

Database changed

mysql@node3> CREATE TABLE example (node\_id INT PRIMARY KEY, node\_name VARCHAR(30));

Query OK, 0 rows affected (0.05 sec)

1. Insert records on the first node:

mysql@node1> INSERT INTO percona.example VALUES (1, 'percona1');

Query OK, 1 row affected (0.02 sec)

1. Retrieve rows from that table on the second node:

mysql@node2> SELECT \* FROM percona.example;

1. +---------+-----------+
2. | node\_id | node\_name |
3. +---------+-----------+
4. | 1 | percona1 |
5. +---------+-----------+

1 row in set (0.00 sec)