# showHướng dẫn cài đặt và cấu hình

# Percona XtraDB Cluster 5.6

# Chuẩn bị trước khi cài đặt.

# Download Percona XtraDB Cluster 5.6:

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

# Download Percona XtraBackup 2.2.5:

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

# Download Percona GPG Key:

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

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

# <https://www.percona.com/downloads/percona-release/redhat/>

# 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 :

# -A INPUT -p udp -m state --state NEW --dport 3306 -j ACCEPT

# -A OUTPUT -p tcp -m state --state NEW --dport 3306 -j ACCEPT

# -A INPUT -p udp -m state --state NEW --dport 4567 -j ACCEPT

# -A OUTPUT -p tcp -m state --state NEW --dport 4567 -j ACCEPT

# -A INPUT -p udp -m state --state NEW --dport 4568 -j ACCEPT

# -A OUTPUT -p tcp -m state --state NEW --dport 4568 -j ACCEPT

# -A INPUT -p udp -m state --state NEW --dport 4444 -j ACCEPT

# -A OUTPUT -p tcp -m state --state NEW --dport 4444 -j ACCEPT

# Cài đặt :

# Giải nén file Percona XtraDB Cluster 5.6.tar

# Tìm file Percona-XtraDB-Cluster-server-56-5.6.30-25.16.3.el6.x86\_64.rpm

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

# A, Mount đến file iso ổ cài đặt Centos 6.7

# # mount /dev/cdrom /mnt/cdrom

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

# perl-DBD-MySQL

# perl-DBI

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

# <http://rpm.pbone.net/index.php3/stat/4/idpl/18151145/dir/redhat_el_6/com/socat-1.7.2.1-1.el6.rf.x86_64.rpm.html>

# C, Giải nén Percona XtraBackup sau đó down file perl(Time::Hires) từ link

# <http://rpm.pbone.net/index.php3/stat/4/idpl/16457442/dir/redhat_el_6/com/perl-Time-HiRes-1.9722-1.el6.rfx.x86_64.rpm.html>

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

# D, Cài đặt Percona-XtraDB-Cluster-client-56-5.6.30-25.16.3.el6.x86\_64.rpm

# E, Cài đặt Percona-XtraDB-Cluster-galera-3-3.16-3.rhel6.x86\_64.rpm

# F, Cài đặt Percona-XtraDB-Cluster-shared-56-5.6.30-25.16.3.el6.x86\_64.rpm

# G, Cuối cùng ta cài Percona-XtraDB-Cluster-server-56-5.6.30-25.16.3.el6.x86\_64.rpm

# Tiếp đó cài những file còn lại trong file Percona XtraDB Cluster 5.6.tar

# A, Cài đặt gói Percona-XtraDB-Cluster-garbd-3-3.16-3.rhel6.x86\_64.rpm cần download packages đi kèm là “nc” tại link

# <http://www.rpmfind.net/linux/RPM/ftp.scientificlinux.org/linux/scientific/6.4/x86_64/os/Packages/nc-1.84-22.el6.x86_64.html>

# Support Percona XtraDB Cluster 5.6 by buying support at http://www.percona.com/products/mysql-support

# WARNING: Default config file /etc/my.cnf exists on the system

# This file will be read by default by the MySQL server

# If you do not want to use this, either remove it, or use the

# --defaults-file argument to mysqld\_safe when starting the server

# Percona XtraDB Cluster is distributed with several useful UDFs from Percona Toolkit.

# Run the following commands to create these functions:

# mysql -e "CREATE FUNCTION fnv1a\_64 RETURNS INTEGER SONAME 'libfnv1a\_udf.so'"

# mysql -e "CREATE FUNCTION fnv\_64 RETURNS INTEGER SONAME 'libfnv\_udf.so'"

# mysql -e "CREATE FUNCTION murmur\_hash RETURNS INTEGER SONAME 'libmurmur\_udf.so'"

# See http://www.percona.com/doc/percona-server/5.5/management/udf\_percona\_toolkit.html for more details

# DISABLE SELINUX

# ##### Lưu ý trước khi cấu hình file /etc/my.cnf ############

# Chay lenh Service mysql start

# Kiểm tra log in password mysql root xem có được hay ko ( mặc đinh là password trống)

# Nếu vào được thì update lại password root để đổi pass theo cách này:

# $ mysql -u root -p

# Use mysql database (type command at mysql> prompt):

# mysql> use mysql;

# Change password for user nixcraft, enter:

# mysql> update user set password=PASSWORD("NEWPASSWORD") where User='nixcraft';

# Finally, reload the privileges:

# mysql> flush privileges;

# mysql> quit

# Hoặc có thể đổi pass bằng lệnh “# mysql\_secure\_installation”

# Nếu ko biết pass là gì thì làm theo cách sau để reset password mysql root

# -------Stop the MySQL service

# (Ubuntu and Debian) Run the following command:

# # sudo /etc/init.d/mysql stop

# (CentOS, Fedora, and Red Hat Enterprise Linux) Run the following command:

# # sudo /etc/init.d/mysqld stop

# ----- Start MySQL without a password

# Run the following command. The ampersand (&) at the end of the command is required.

# # sudo mysqld\_safe --skip-grant-tables &

# -----Connect to MySQL

# Run the following command:

# #mysql -uroot

# -----Set a new MySQL root password

# Run the following command:

# #mysql>use mysql;

# #mysql>update user set password=PASSWORD("mynewpassword") where User='root';

# #mysql>flush privileges;

# #mysql>quit

# ----Stop and start the MySQL service

# (Ubuntu and Debian) Run the following commands:

# #sudo /etc/init.d/mysql stop

# ...

# #sudo /etc/init.d/mysql start

# (CentOS, Fedora, and Red Hat Enterprise Linux) Run the following commands:

# #sudo /etc/init.d/mysqld stop

# ...

# #sudo /etc/init.d/mysqld start

# Log in to the database

# Test the new password by logging in to the database.

# #mysql -u root -p

# You are prompted for your new password.

# Installing Percona XtraDB Cluster on CentOS

This tutorial will show how to install the Percona XtraDB Cluster on three CentOS 6.3 servers, using the packages from Percona repositories.

This cluster will be assembled of three servers/nodes:

node #1

hostname: percona1

IP: 192.168.70.71

node #2

hostname: percona2

IP: 192.168.70.72

node #3

hostname: percona3

IP: 192.168.70.73

## **Prerequisites**

* All three nodes have a CentOS 6.3 installation.
* Firewall has been set up to allow connecting to ports 3306, 4444, 4567 and 4568
* SELinux is disabled

## **Installation**

Installation information can be found in the [Installing Percona XtraDB Cluster from Binaries](https://www.percona.com/doc/percona-xtradb-cluster/5.6/installation.html#installation) guide.

## **Configuring the nodes**

Individual nodes should be configured to be able to bootstrap the cluster. More details about bootstrapping the cluster can be found in the [Bootstrapping the cluster](https://www.percona.com/doc/percona-xtradb-cluster/5.6/manual/bootstrap.html#bootstrap) guide.

Configuration file /etc/my.cnf for the first node should look like:

[mysqld]

datadir=/var/lib/mysql

user=mysql

# Path to Galera library

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

# Cluster connection URL contains the IPs of node#1, node#2 and node#3

wsrep\_cluster\_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW

binlog\_format=ROW

# MyISAM storage engine has only experimental support

default\_storage\_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a requirement for Galera

innodb\_autoinc\_lock\_mode=2

# Node #1 address

wsrep\_node\_address=192.168.70.71

# SST method

wsrep\_sst\_method=xtrabackup-v2

# Cluster name

wsrep\_cluster\_name=my\_centos\_cluster

# Authentication for SST method

wsrep\_sst\_auth="sstuser:s3cret"

After this, first node can be started with the following command:

[root@percona1 ~]# /etc/init.d/mysql bootstrap-pxc

In case you’re running this tutorial on CentOS 7 server, systemd bootstrap service should be used instead:

[root@percona1 ~]# systemctl start mysql@bootstrap.service

This command will start the cluster with initial [wsrep\_cluster\_address](https://www.percona.com/doc/percona-xtradb-cluster/5.6/wsrep-system-index.html" \l "wsrep_cluster_address" \o "wsrep_cluster_address) set to gcomm://. This way the cluster will be bootstrapped and in case the node or MySQL have to be restarted later, there would be no need to change the configuration file.

After the first node has been started, cluster status can be checked by:

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)

This output shows that the cluster has been successfully bootstrapped.

It’s recommended not to leave the empty password for the root account. Password can be changed with:

mysql@percona1> UPDATE mysql.user SET password=PASSWORD("Passw0rd") where user='root';

mysql@percona1> FLUSH PRIVILEGES;

In order to perform successful [State Snapshot Transfer](https://www.percona.com/doc/percona-xtradb-cluster/5.6/manual/state_snapshot_transfer.html#state-snapshot-transfer) using XtraBackup new user needs to be set up with proper[privileges](http://www.percona.com/doc/percona-xtrabackup/innobackupex/privileges.html#permissions-and-privileges-needed):

mysql@percona1> CREATE USER 'sstuser'@'localhost' IDENTIFIED BY 's3cret';

mysql@percona1> GRANT PROCESS, RELOAD, LOCK TABLES, REPLICATION CLIENT ON \*.\* TO 'sstuser'@'localhost';

mysql@percona1> FLUSH PRIVILEGES;

**Note**

MySQL root account can also be used for setting up the SST with Percona XtraBackup, but it’s recommended to use a different (non-root) user for this.

Configuration file /etc/my.cnf on the second node (percona2) should look like this:

[mysqld]

datadir=/var/lib/mysql

user=mysql

# Path to Galera library

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

# Cluster connection URL contains IPs of node#1, node#2 and node#3

wsrep\_cluster\_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW

binlog\_format=ROW

# MyISAM storage engine has only experimental support

default\_storage\_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a requirement for Galera

innodb\_autoinc\_lock\_mode=2

# Node #2 address

wsrep\_node\_address=192.168.70.72

# Cluster name

wsrep\_cluster\_name=my\_centos\_cluster

# SST method

wsrep\_sst\_method=xtrabackup-v2

#Authentication for SST method

wsrep\_sst\_auth="sstuser:s3cret"

Second node can be started with the following command:

[root@percona2 ~]# /etc/init.d/mysql start

After the server has been started it should receive the state snapshot transfer automatically. This means that the second node won’t have the empty root password anymore. In order to connect to the cluster and check the status changed root password from the first node should be used. Cluster status can now be checked on both nodes. This is the example from the second node (percona2):

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 | 2 |

| wsrep\_cluster\_status | Primary |

| wsrep\_connected | ON |

...

| wsrep\_ready | ON |

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

40 rows in set (0.01 sec)

This output shows that the new node has been successfully added to the cluster.

MySQL configuration file /etc/my.cnf on the third node (percona3) should look like this:

[mysqld]

datadir=/var/lib/mysql

user=mysql

# Path to Galera library

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

# Cluster connection URL contains IPs of node#1, node#2 and node#3

wsrep\_cluster\_address=gcomm://192.168.70.71,192.168.70.72,192.168.70.73

# In order for Galera to work correctly binlog format should be ROW

binlog\_format=ROW

# MyISAM storage engine has only experimental support

default\_storage\_engine=InnoDB

# This changes how InnoDB autoincrement locks are managed and is a requirement for Galera

innodb\_autoinc\_lock\_mode=2

# Node #3 address

wsrep\_node\_address=192.168.70.73

# Cluster name

wsrep\_cluster\_name=my\_centos\_cluster

# SST method

wsrep\_sst\_method=xtrabackup-v2

#Authentication for SST method

wsrep\_sst\_auth="sstuser:s3cret"

Third node can now be started with the following command:

[root@percona3 ~]# /etc/init.d/mysql start

After the server has been started it should receive the SST same as the second node. Cluster status can now be checked on both nodes. This is the example from the third node (percona3):

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 | 3 |

| wsrep\_cluster\_status | Primary |

| wsrep\_connected | ON |

...

| wsrep\_ready | ON |

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

40 rows in set (0.01 sec)

This output confirms that the third node has joined the cluster.

## **Testing the replication**

Although the password change from the first node has replicated successfully, this example will show that writing on any node will replicate to the whole cluster. In order to check this, new database will be created on second node and table for that database will be created on the third node.

Creating the new database on the second node:

mysql@percona2> CREATE DATABASE percona;

Query OK, 1 row affected (0.01 sec)

Creating the example table on the third node:

mysql@percona3> USE percona;

Database changed

mysql@percona3> CREATE TABLE example (node\_id INT PRIMARY KEY, node\_name VARCHAR(30));

Query OK, 0 rows affected (0.05 sec)

Inserting records on the first node:

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

Query OK, 1 row affected (0.02 sec)

Retrieving all the rows from that table on the second node:

mysql@percona2> SELECT \* FROM percona.example;

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

| node\_id | node\_name |

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

| 1 | percona1 |

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

1 row in set (0.00 sec)

This small example shows that all nodes in the cluster are synchronized and working as intended.

# 