

Percona Cluster Setup and Configuration

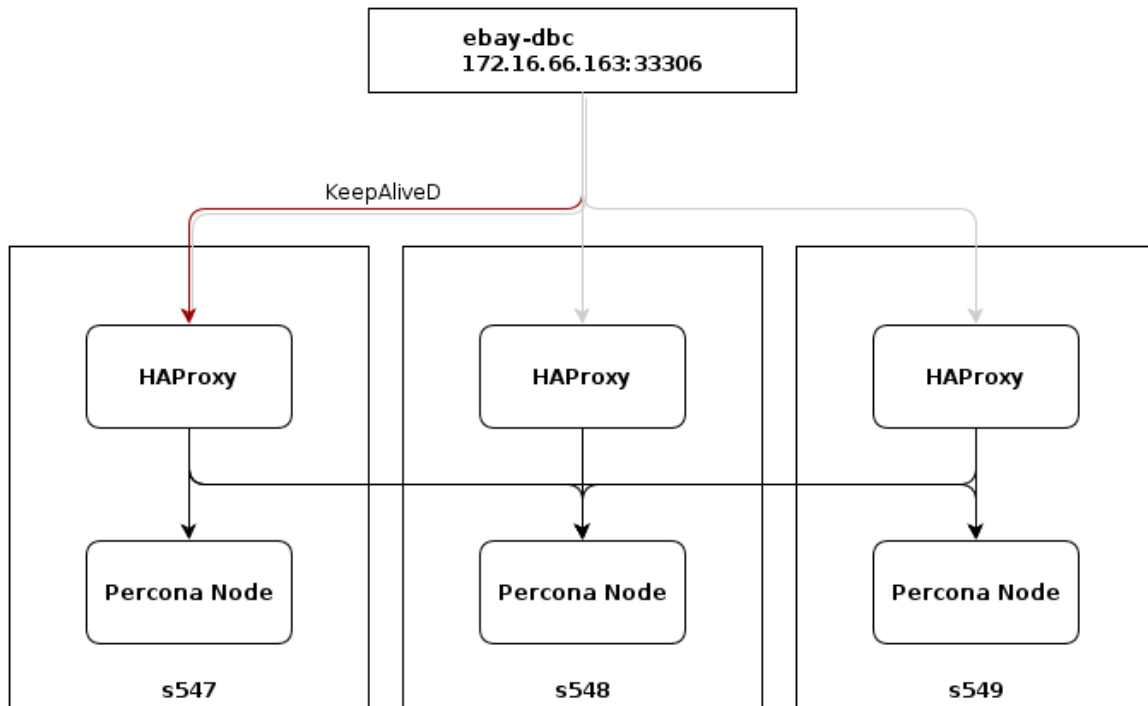
- Allgemeine Hinweise
 - Node resync sperrt den Master
 - Neuaufsetzen mit xtrabackup
 - Leistungsverlust durch Netzwerk-Übertragung
 - Maximalgröße für Transaktionen
 - Scheitern von Transaktionen
- Die ätzendsten Tücken
- Basic Installation
 - Adapt to idealo configuration
 - Now install percona server - with idealo my.cnf settings:
 - MySQL Commands for Testing
 - Diamond install
- HaProxy
 - Primitive Configuration => with alive-check only
 - Perfect Configuration => with sync-check of each node
- VRRP - Virtual IP - KeepaliveD
- Appendix and Known Problems

Verbindung über

```
mysql -h ebay-dbc -port=33306 -u xxx -psecret
```

oder - für renitente Anwendungen:

```
mysql -h ebay-dbc -port=33307 -u xxx -psecret
```



Handbuch

von Percona über Registrierung runtergeladen.

Hier als [Attachment](#).

Außerdem hier die [Percona Cluster FAQ](#).

Allgemeine Hinweise

Node resync sperrt den Master

Für *kurzzeitige* Ausfälle eines Node gibt es einen Puffer.

gcache.size = 128M

Aber sobald dieser aufgebraucht ist, gibt es volles Neuaufsetzen.

Dieser Puffer entspricht der `binlog_size`. Bei `ebay-db/offer-db` wäre er schon nach wenigen Sekunden aufgebraucht. Wir erhöhen ihn daher:

```
wsrep_provider_options="gcache.size = 2G"
```

Details: <http://www.severalnines.com/blog/understanding-gcache-galera>

Neuaufsetzen mit xtrabackup

Zum Neuaufsetzen lässt sich auch xtrabackup konfigurieren. Dann müssen nur die *.frm Dateien kurz gelockt zu werden. Der Cluster bleibt voll schreibbar. Die Performance sinkt um ca. 30 Prozent.

In jedem Fall müssen alle Tabellen-Daten übers Netz gesendet werden.

Änderungen während des Neuaufsetzens werden über "`gcache_pages`" zusätzlich übermittelt. xtrabackup und laufende Änderungen werden jeweils über einen Netzwerkport gestreamt.

Damit xtrabackup Neuaufsetzen funktioniert, müssen redo log files und ibdata files direkt im datadir liegen. Bestimmte settings in der idealo `mysqld my.cnf` stören das Neuaufsetzen. Weiter unten ist eine reduzierte, funktionierende `my_reduced.cnf` eingebunden.

Gemessene Dauer für ein SST-rejoin mit xtrabackup:

20 Minuten je 100 GB Database size.

Leistungsverlust durch Netzwerk-Übertragung

Durch die Netzwerk-Roundtrips zwischen den Knoten sinkt die insert-Rate von 120.000 inserts pro Sekunde auf 55.000 inserts pro Sekunde.

Der Roundtrip vom HAProxy zum Node erhöht die Dauer noch ein Mal.

Maximalgröße für Transaktionen

Maximale Transaktionsgröße ist limitiert durch

```
show status variables '%wsrep_max%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_max_ws_rows | 131072 |
| wsrep_max_ws_size | 1073741824 |
+-----+-----+
```

Scheitern von Transaktionen

Beim Commit kann ein Lock-Fehler auftreten.

Wenn die Anwendungen das nicht verkraften, dann müssen alle Anwendungen per HAProxy auf einen einzelnen Node gelenkt werden, und die anderen als Backup konfiguriert werden.

Dazu verbinden sie sich an den HAProxy Host mit port 33307 statt 33306.

Die entsprechenden HAProxy Konfigurationen sind unten beschrieben: `frontend pxc-onenode-front` und `pxc-front`, described below.

Kein MyISAM

Nur Innodb ist verwendbar.

Schwächstes Glied

Der schwächste Knoten bestimmt die maximale Geschwindigkeit.

Arbiter

Handbuch Seite 28 erwähnt einen Arbiter. Der würde eine Riesenmaschine ersparen. Leider gibt's keine weiteren Infos.

Autoincrement-Lösung

Autoincrement Spalten zählen jetzt anders hoch.

Jedes INSERT erzeugt einen Wert Teilbar durch Nodeanzahl

Die ätzendsten Tücken

debian-sys-maint user

debian-sys-maint braucht natürlich wieder auf allen nodes das selbe Passwort.

Andernfalls kommt der node nach dem resync nicht wieder hoch - ohne jede Fehlermeldung.

```
sed -i -e "s/^password.*/password = secret/;" /etc/mysql/debian.cnf # each node

echo "SET PASSWORD FOR 'debian-sys-maint'@'localhost' = PASSWORD( 'secret' );" | mysql --
user=root --password=test # once

echo "SELECT NOW();" | mysql --user=debian-sys-maint --password=secret # check access
```

Kein include in der my.cnf

Die my.cnf darf nicht auf include settings aus dem conf.d Verzeichnis angewiesen sein.

Habe nicht genau herausgefunden, welcher Schritt des xtrabackup dadurch crashed, aber ohne include ging es.

Pfade fest verdrahtet

ibdataXXX und iblogdirXXX müssen im *selben* Verzeichnis liegen.

Die Datenbank-Tabellen-Dateien müssen ein Verzeichnis tiefer liegen.

Der join/rejoin Mechanismus kommt sonst nicht klar.

Ebenfalls darf mysqld_using_numa nicht wie bisher unter /data/mysql liegen.

Es wird nämlich beim rejoin gelöscht, und der mysqld startet nicht.

Zombie Threads verhindern den start/restart von mysqld

some thread did not exit - bedeutet,
dass noch ein hängengebliebenes xtrabackup läuft.

Benutze

```
ps aux | grep mysqld
ps aux | grep xtrabackup
```

zum Abschießen.

Zombie UUIDs

grastate.dat und auto.cnf enthalten die UUID des Node.

Beim Komplett-Neuaufsetzen sollte man diese nicht löschen,
andernfalls bleibt die "alte" UUID für immer im "view"

```
view(view_id(PRIM,d99f1441-d1ff-11e3-ac7f-e726af1414ff,11) memb {
```

```

d99f1441-d1ff-11e3-ac7f-e726af1414ff,0
dcb7278b-d04d-11e3-aaf7-0fccb9afad40,0
e1138f36-d04b-11e3-9f1a-bee03c410b90,0
} joined {
} left {
} partitioned {
    e1138f3... old UID ..f1a-bee03c410b90,0
}}

```

Dagegen hilft Runterfahren *aller* Knoten.

Neuaufsetzen ohne Zombie UUIDs

```

rm -rf /data/mysql_binlogs/*

rm /data/mysql/galera.cache

rm /data/mysql/ib_logfile*

rm /data/mysql/ibdata1*

rm /data/mysql/*.log

```

Keep /data/mysql/grastate.dat

Keep /data/mysql/auto.cnf

percona_node_restart # should prompt complete state transfer

Das rejoinen unter Last für eine 120 GB große Datenbank
sieht wie folgt aus:



Basic Installation

Percona Repository

There is a repo and Debian setup guide here:

http://www.percona.com/doc/percona-xtradb-cluster/5.6/installation/apt_repo.html

```

apt-key adv --keyserver keys.gnupg.net --recv-keys 1C4CBDCDCD2EFD2A
echo -e "\ndeb http://repo.percona.com/apt wheezy main \ndeb-src
http://repo.percona.com/apt wheezy main " >> /etc/apt/sources.list

```

Alternatively: Debian Packages

We might get debian packages directly from

http://www.percona.com/downloads/Percona-XtraDB-Cluster/LATEST/deb/wheezy/x86_64/

Init, Clean up

Loop over all servers

automate

```
for SERVERX in s547 s548 s549
  ssh $SERVERX " sudo apt-get update"
  ssh $SERVERX " sudo apt-get remove mysql-server mysql-client mysql-common ";
  ssh $SERVERX " sudo apt-get purge mysql-server mysql-client mysql-common ";
  ssh $SERVERX " sudo apt-get autoremove ";
  ssh $SERVERX " sudo apt-get remove --purge percona-xtradb-cluster-server-5.5
percona-xtradb-cluster-client-5.5 percona-xtradb-cluster-server-5.6 ";
  ssh $SERVERX " sudo apt-get autoremove ";
done
```

Adapt to idealo configuration

/var/lib/mysql as datadir has never enough space.

We need to remap the installation to /data/mysql.

During resync, /data/mysql gets **cometely cleansed**.

=> For reconfiguration of file system or directories - shutdown **entire Cluster**

move installation to /data/mysql

```
apt-get install numactl
apt-get install libtcmalloc-minimal4
apt-get install sysbench
apt-get install socat

/etc/init.d/mysql stop

mkdir -p /data/mysql
mkdir -p /data/mysql_helper
mkdir -p /data/mysql_tmp
mkdir -p /data/mysql_binlogs

echo '#!/bin/sh ' > /data/mysql_helper/mysqld_using_numa
echo -e "\n#work around to startup mysqld using numactl \nnumactl=/usr/bin/numactl
# Adjust the path as needed \nmysqld=/usr/sbin/mysqld          # Adjust the path as
needed \n\n# use exec to avoid having an extra shell around." >>
/data/mysql_helper/mysqld_using_numa
echo 'exec $numactl --interleave all $mysqld "$@" ' >>
/data/mysql_helper/mysqld_using_numa

chmod 755 /data/mysql_helper/mysqld_using_numa
chown -R mysql:mysql /data/mysql /data/mysql_helper /data/mysql_tmp
/data/mysql_binlogs

# if install was with default directories, we have to move the installation data...
mv /var/lib/mysql/* /data/mysql/galera/
# symlinking
rmdir /var/lib/mysql/
rm /var/lib/mysql # pref symlink
ln -s /data/mysql/galera /var/lib/mysql
```

Idealo specific mysql settings are saved in

[/etc/mysql/my_reduced.cnf](#)

Here is an [extended version](#), that still causes trouble during resync.

Here is the [full version](#). of the ebay my.cnf for comparison of the changes.

Now we put only the percona cluster settings into my.cnf.

host-specific my.cnf

```
MYIP="$(ifconfig | grep -A 1 'eth0' | tail -1 | cut -d ':' -f 2 | cut -d ' ' -f 1)"

echo "$MYIP" # check

/etc/init.d/mysql stop
# cleanup
echo "" > /data/mysql/$HOSTNAME.err
echo "" > /etc/mysql/my.cnf

echo -e "[mysqld]
wsrep_provider=/usr/lib/libgalera_smm.so
wsrep_cluster_name=xdm
wsrep_node_name=pxc_$HOSTNAME
wsrep_node_address=$MYIP
wsrep_cluster_address=gcomm://172.16.64.30,172.16.64.47,172.16.64.82
wsrep_provider_options="gcache.size = 2G"
wsrep_slave_threads=4
wsrep_sst_method=xtrabackup          # [rsync | mysqldump | xtrabackup]
wsrep_sst_auth=root:test            # only for wsrep_sst_method <> rsync
binlog_format=ROW
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
innodb_locks_unsafe_for_binlog=1
" >> /etc/mysql/my.cnf

cat /etc/mysql/my_reduced.cnf >> /etc/mysql/my.cnf

echo -e '!includedir /etc/mysql/conf.d/' >> /etc/mysql/my.cnf
# cat /etc/mysql/my.cnf # check
```

The entire start/restart procedure is put into this [script](#).

Now install percona server - with idealo my.cnf settings:

```
sudo apt-get install percona-xtradb-cluster-56
```

```
root password is test
```

Ignore start failure of installation.

Also check out the problem solutions at the end.

Then start the service manually:

```
/etc/init.d/mysql start
```

OR

```
/etc/init.d/mysql bootstrap-pxc
```

```
tail -n 100 -f /data/mysql/$HOSTNAME.err
```

Synchronize debian-sys-maint accounts

```
sed -i -e "s/^password.*/password = secret/;" /etc/mysql/debian.cnf # each node
```

```
echo "SET PASSWORD FOR 'debian-sys-maint'@'localhost' = PASSWORD( 'secret' );" | mysql --
user=root --password=test # once
```

```
echo "SELECT NOW(); " | mysql --user=debian-sys-maint --password=secret # check access
```

MySQL Commands for Testing

```
cat /data/mysql/$HOSTNAME.err
```

```
mysql -u root -ptest
```

```
SHOW STATUS LIKE 'wsrep%';
use test;
show tables;

CREATE TABLE `test`.`sbt` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `k` int(10) unsigned NOT NULL DEFAULT '0',
  `c` char(120) NOT NULL DEFAULT '',
  `pad` char(60) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `k` (`k`)
);
insert into test.sbt (k,c,pad) values (4, now(), concat_ws( ' - ', now(), user() )
);
select * from test.sbt ;

show engine innodb status\G

SELECT
      A.variable_name ,
      A.variable_value
FROM      information_schema.global_status  A
WHERE      1=1
      AND (
            A.variable_name in
('wsrep_cluster_size','wsrep_cluster_status','wsrep_connected','wsrep_ready')
            OR A.variable_name in
('wsrep_local_cert_failures','wsrep_local_bf_aborts')
            OR A.variable_name in
('wsrep_local_recv_queue','wsrep_local_send_queue')
            OR A.variable_name in
('wsrep_flow_control_sent','wsrep_flow_control_recv')
            OR A.variable_name in ('wsrep_local_state')
            OR A.variable_name LIKE 'wsrep_replicated%'
            OR A.variable_name LIKE 'wsrep_received%'
      )
ORDER BY      A.variable_name;
```

Diamond install


```

apt-get install diamond
apt-get install python-mysqldb

echo "enabled = true
hosts = diamond:diamond@localhost:3306/mysql,
innodb = true
master = true
slave = true
" >> /etc/diamond/collectors/MySQLCollector.conf
echo "[[NetworkCollector]]
enabled=true
" >> /etc/diamond/diamond.conf
/etc/init.d/diamond restart

# on any host - only once
echo "CREATE USER 'diamond'@'localhost' ; " | mysql --user=root --password=test
echo "SET PASSWORD FOR 'diamond'@'localhost' = PASSWORD( 'diamond' ); " | mysql
--user=root --password=test
echo "GRANT SELECT ON *.* TO 'diamond'@'localhost' ; " | mysql --user=root
--password=test
echo "FLUSH PRIVILEGES; " | mysql --user=root --password=test

```

HaProxy

```

apt-get install xinetd
apt-get install socat
apt-get install haproxy

cp -n /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy_orig.cfg

echo "" > /etc/haproxy/haproxy.cfg

vim /etc/haproxy/haproxy.cfg

```

Primitive Configuration => with alive-check only

```

GRANT USAGE ON *.* TO 'user_mysql_check'@'localhost' IDENTIFIED BY PASSWORD '';

GRANT USAGE ON *.* TO 'user_mysql_check'@'%' IDENTIFIED BY PASSWORD '';

```

```

# this config needs haproxy-1.4.20
global
    log 127.0.0.1 local0
    log 127.0.0.1 local1 notice
    maxconn 4096
    #chroot /usr/share/haproxy
    #user haproxy
    #group haproxy
    #uid 99
    #gid 99
    #daemon
    #debug
    #quiet

defaults
    log global
    mode http
    option tcplog
    option dontlognull
    retries 3
    option redispatch
    maxconn 2000
    timeout 5000
    clitimeout 50000
    srvtimeout 50000

listen mysql-cluster 0.0.0.0:3306
    mode tcp
    balance roundrobin
    option mysql-check user_mysql_check root

server s547 172.16.64.30:3306 check
server s548 172.16.64.47:3306 check
server s549 172.16.64.82:3306 check

```

```

#/etc/init.d/haproxy restart
screen -S 'haproxy' -d -m /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg

screen -r haproxy

```

Perfect Configuration => with sync-check of each node

Siehe Handbuch S. 46 - besonders ab Seite 53.

HAProxy muss die Cluster-Nodes über http überwachen können.
 Hierzu hat jeder Cluster-Node ein lokales Script clustercheck

```

apt-get install xinetd
apt-get install socat
apt-get install haproxy

```

Prepare Script for Node Check

```

GRANT USAGE ON *.* TO 'clustercheck'@'localhost' IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB99005ADC
A2EC9D1E19';

```

```

GRANT USAGE ON *.* TO 'clustercheck'@'%' IDENTIFIED BY PASSWORD '*2470C0C06DEE42FD1618BB99005ADC
A2EC9D1E19';

```

then

```
/usr/bin/clustercheck [mysql_user] [the_password] 0
```

```
/usr/bin/clustercheck clustercheck password 0
```

Prüfe

```
/usr/bin/clustercheck clustercheck password 0
```

```
$ ...  
$ Percona XtraDB Cluster Node is synced.
```

Jetzt soll man dieses script clustercheck via http verfügbar machen,
und zwar über ein **xinetd script**:

```
echo -e "service mysqlchk  
{  
    disable = no  
    flags = REUSE  
    socket_type = stream  
    port = 9200  
    wait = no  
    user = nobody  
    server = /usr/bin/clustercheck  
    server_args = clustercheck password 1  
    log_on_failure += USERID  
    only_from = 0.0.0.0/0  
    per_source = UNLIMITED  
}" > /etc/xinetd.d/mysqlchk
```

Port und Dienstname für xinetd eintragen:

```
echo "mysqlchk 9200/tcp # mysqlchk" >> /etc/services
```

Restart xinetd.

```
/etc/init.d/xinetd restart
```

Und nun soll clustercheck über http port 9200 von remote abfragbar sein:

```
nc localhost 9200
```

```
nc s547 9200
```

```
$ Percona XtraDB Cluster Node is synced.
```

Im Handbuch steht die HAProxy Konfig mit *.[Port]

Ich habe *.80, *.330 ersetzt durch 0.0.0.0:[port]

Finale HAProxy config:

```
echo "# this config needs haproxy-1.4.20"
global
    log 127.0.0.1 local0
    log 127.0.0.1 local1 notice
    user      haproxy
    group     haproxy
    #daemon
    #debug
    #quiet
    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats mode 777 level admin

defaults
    log global
    mode http
    option tcplog
    option dontlognull
    retries 3
    option redispatch
    maxconn 2000
    contimeout 5000
    clitimeout 50000
    srvtimeout 50000

frontend pxc-front
    bind 0.0.0.0:33306
    mode tcp
    default_backend pxc-back

frontend stats-front
    bind 0.0.0.0:80
    mode http
    default_backend stats-back

frontend pxc-onenode-front
    bind 0.0.0.0:33307
    mode tcp
    default_backend pxc-onenode-back

backend pxc-back
    mode tcp
    balance leastconn
    option httpchk
    server c1 172.16.64.30:3306 check port 9200 inter 12000 rise 3 fall 3
    server c2 172.16.64.47:3306 check port 9200 inter 12000 rise 3 fall 3
    server c3 172.16.64.82:3306 check port 9200 inter 12000 rise 3 fall 3

backend stats-back
    mode http
    balance roundrobin
    stats uri /haproxy/stats
    stats auth pxcstats:secret

backend pxc-onenode-back
    mode tcp
    balance leastconn
    option httpchk
    server c1 172.16.64.30:3306 check port 9200 inter 12000 rise 3 fall 3
    server c2 172.16.64.47:3306 check port 9200 inter 12000 rise 3 fall 3 backup
    server c3 172.16.64.82:3306 check port 9200 inter 12000 rise 3 fall 3 backup

" > /etc/haproxy/haproxy.cfg
```

Wobei eine Verbindung zu Port 33307 einen Node zum alleinigen Ansprechpartner macht, während die mysql-Verbindung zu Port 33306 alle Nodes nach roundrobin anspricht.

Hierbei können jedoch commit-abort Fehler auftreten.

Start haproxy in screen:

```
screen -S 'haproxy' -d -m /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg
screen -r haproxy
```

Check it:

```
echo "show stat" | socat stdio /var/lib/haproxy/stats
echo "show stat" | socat stdio /var/lib/haproxy/stats | cut -d, -f1,2,18
echo "disable server pxc-onenode-back/c3" | socat stdio /var/lib/haproxy/stats
```

Or use the http frontend:

<http://s547/haproxy/stats>

VRRP - Virtual IP - Keepalived

Based on <http://support.severalnines.com/entries/23612682-Install-HAProxy-and-Keepalived-Virtual-IP->

```
apt-get install keepalived
```

```
##etc/default/ipv6adm
```

Tell kernel to allow binding non-local IP into the hosts and apply the changes:

```
echo "net.ipv4.ip_nonlocal_bind = 1" >> /etc/sysctl.conf
sysctl -p
```

Our virtual IP address is **172.16.66.163**.

Change priority **102,101,100** on each node.

```
echo -e "vrrp_script chk_haproxy {
    script \"killall -0 haproxy\" # verify the pid existence
    interval 2 # check every 2 seconds
    weight 2 # add 2 points of prio if OK
}

vrrp_instance VI_1 {
    interface eth0 # interface to monitor
    state MASTER
    virtual_router_id 51 # Assign one ID for this route
    priority 102 # 101 on master, 100 on backup
    virtual_ipaddress {
        172.16.66.163 # the virtual IP
    }
    track_script {
        chk_haproxy
    }
}" > /etc/keepalived/keepalived.conf

/etc/init.d/keepalived stop
```

```
/etc/init.d/keepalived start
```

Verify the Keepalived status

[node_1] should hold the VIP and the MASTER state
while [node_2] should run as BACKUP state without VIP.

[node_1]

```
$ ip a | grep -e inet.*eth0
inet 172.16.66.30/24 brd 192.168.10.255 scope global eth0
inet 172.16.66.163/32 scope global eth0

$ cat /var/log/messages | grep VRRP_Instance
[datestamp] [node_1] Keepalived_vrrp[6146]: VRRP_Instance(VI_1) Transition to MASTER STATE
[datestamp] [node_1] Keepalived_vrrp[6146]: VRRP_Instance(VI_1) Entering MASTER STATE
```

[node_2]

```
$ ip a | grep -e inet.*eth0
inet 172.16.66.47/24 brd 192.168.10.255 scope global eth0

$ cat /var/log/messages | grep VRRP_Instance
[datestamp] [node_2] Keepalived_vrrp[6146]: VRRP_Instance(VI_1) Transition to MASTER STATE
[datestamp] [node_2] Keepalived_vrrp[6146]: VRRP_Instance(VI_1) Received higher prio advert
[datestamp] [node_2] Keepalived_vrrp[6146]: VRRP_Instance(VI_1) Entering BACKUP STATE
```

[node_3]

Wie node2

Appendix and Known Problems

Old way of bootstrapping

```
# old way of bootstrapping
#echo $LEADERIP
#/etc/init.d/mysql start --wsrep-cluster-address="gcomm://$LEADERIP" # node1 WITHOUT a leader,
node2/node3 with node1 as leader
# later
# sed -i -e "s/#wsrep_cluster_address/wsrep_cluster_address/" /etc/mysql/my.cnf
```

Possible problem: *.ibd does not exist, though *.frm exists

This happens, when ibdata files have different/changed directory positions.

Make sure ibdata files remain exactly where they have been created.

Make sure the directory structure remains the same for an xtrabackup.

Possible problem: Unable to set password

Unable to set password for the Percona Server "root" user

=> purge previous mysql files

Possible problem: Query cache not allowed

```
sed -i -e "s/query_cache_limit/#query_cache_limit/" /etc/mysql/my.cnf
sed -i -e "s/query_cache_size/#query_cache_size/" /etc/mysql/my.cnf
```

Possible problem: mysql user database not available

```
/usr/sbin/mysqld --skip-grant &  
/usr/bin/mysql -u root mysql  
show tables
```

Possible problem: sock or pid error

check:

```
chown mysql:root /var/run/mysqld/  
ls -lAh /var/run/mysqld/
```

Problem - wsrep_sst_method=xtrabackup on rejoining

mysql.err says something like

```
could not copy host:/var/lib/mysql/**/*
```

=> the datadir has problems with symlinks.

Problem - if ports are not open across the cluster

required ports: 3306, 4444, 4567, 4568

```
lsof -ni tcp:4444  
netstat -nat | grep :4444
```

```
lsof -ni tcp:4567  
netstat -nat | grep :4567
```

```
nc -vz s547 3306  
nc -vz s547 4444  
nc -vz s547 4566  
nc -vz s547 4567
```