Galera replication – how to recover a PXC cluster

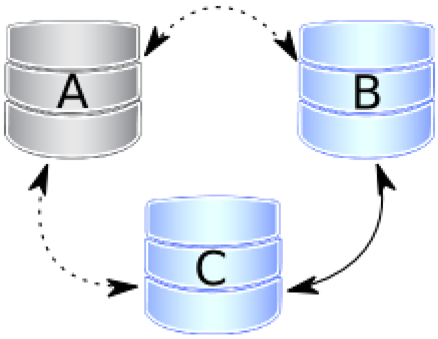
Galera replication for MySQL brings not only the new, great features to our ecosystem, but also introduces completely new maintenance techniques. Are you concerned about adding such new complexity to your MySQL environment? Perhaps that concern is unnecessarily.

I am going to present here some simple tips that hopefully will let fresh Galera users prevent headaches when there is the need to recover part or a whole cluster in certain situations. I used Percona XtraDB Cluster (https://www.percona.com/software/percona-xtradb-cluster) (project based on Percona Server and Galera library + MySQL extensions from Codership) to prepare this post, but most if not all of the scenarios should also apply to any solution based on MySQL+Galera tandem you actually chose, whether these are binaries from Codership (http://galeracluster.com/downloads/#downloads), MariaDB Galera Cluster or maybe your own builds.

Unlike standard MySQL replication, a PXC cluster acts like one logical entity, which takes care about each node status and consistency as well as cluster status as a whole. This allows to maintain much better data integrity then you may expect from traditional asynchronous replication while allowing safe writes on multiple nodes in the same time. This is though for the price of more possible scenarios where database service will be stopped with no node being able to serve requests.

Lets assume the simplest case cluster of nodes **A**, **B** and **C** and few possible scenarios where some or all nodes are out of service. What may happen and what we have to do, to bring them (or whole cluster) back up.

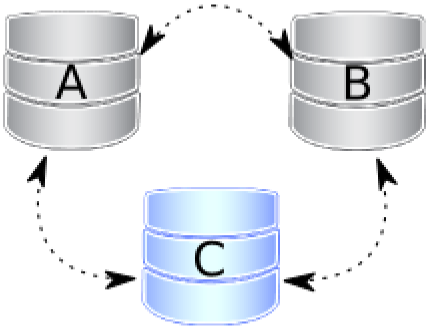
Scenario 1



(https://www.percona.com/blog/wp-content/uploads/2014/08/g1.png)Node A is gracefully stopped. Likely for the purpose of maintenance, configuration change, etc. In this case the other nodes receive “good bye” message from that node, hence the cluster size is reduced and some properties like quorum calculation or auto increment are automatically changed. Once we start the A node again, it will join the cluster based on it’s wsrep_cluster_address setting in my.cnf. This process is much different from normal replication – the joiner node won’t serve any requests until it is again fully synchronized with the cluster, so connecting to it’s peers isn’t enough, state transfer must succeed first. If the writeset cache (`gcache.size` (https://www.percona.com/doc/percona-xtradb-cluster/5.6/wsrep-provider-index.html#gcache.size)), on nodes B and/or C has still all the transactions there were executed during the time this node was down, joining will be possible via (usually fast and light) IST (http://galeracluster.com/documentation-webpages/statetransfer.html#incremental-state-transfer-ist).

Otherwise, full SST (https://www.percona.com/doc/percona-xtradb-cluster/5.6/manual/state_snapshot_transfer.html) will be needed, which in fact is full binary data snapshot copy. Hence it may be important here to determine the best donor, as shown in this article (https://www.percona.com/blog/2014/01/08/finding-good-ist-donor-percona-xtradb-cluster-5-6/). If IST is impossible due to missing transactions in donor's gcache, the fallback decision is made by the donor and SST is started automatically instead.

Scenario 2

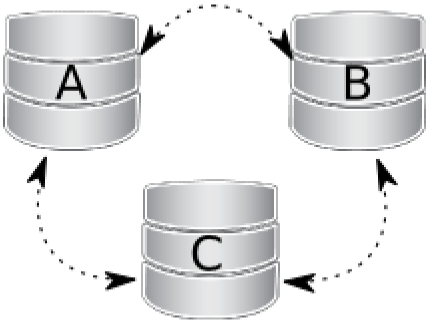


(https://www.percona.com/blog/wp-content/uploads/2014/08/g2.png)Nodes A and B are gracefully stopped. Similar to previous case, cluster size is reduced to 1, hence even the single remaining node C forms a primary component (http://galeracluster.com/documentation-webpages/weightedquorum.html#primary-component) and is serving client requests. To get the nodes back into the cluster, you just need to start them. However, the node C will be switched to "Donor/Desynced" state as it will have to provide state transfer to at least first joining node. It is still possible to read/write to it during that process, but it may be much slower, depending how large state transfers it needs to send. Also some load balancers may consider the donor node as not operational and remove it from the pool. So it is best to avoid situation when only one node is up.

Note though, if you restart A and then B in that order, you may want to make sure B won't use A as state transfer donor, as A may not have all the needed writesets in it's gcache. So just specify the C node as donor this way ("nodeC" name is the one you specify with wsrep_node_name variable):

```
service mysql start --wsrep_sst_donor=nodeC
```

Scenario 3



(https://www.percona.com/blog/wp-content/uploads/2014/08/g3.png)All three nodes are gracefully stopped. Cluster is deformed. In this case, the problem is how to initialize it again. Here, it is important to know, that during clean shutdown, a PXC node writes it's last executed position into the `grastate.dat` file. By comparing the seqno number inside, you will see which node is the most advanced one (most likely the last one stopped). Cluster must be bootstrapped using this node, otherwise nodes that had more advanced position will have to perform full SST to join cluster initialized from the less advanced one (and some transactions will be lost). To bootstrap the first node, invoke the startup script like this:

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

or

```
service mysql bootstrap-pxc
```

or

```
service mysql start --wsrep_new_cluster
```

or

```
service mysql start --wsrep-cluster-address="gcomm://"
```

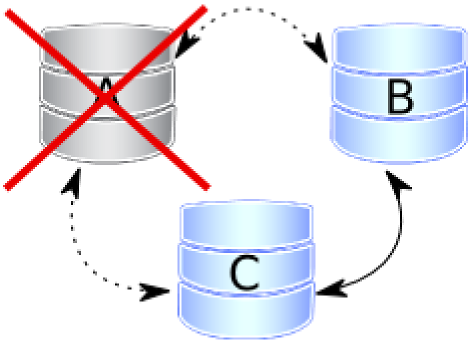
or in packages using systemd (http://www.freedesktop.org/wiki/Software/systemd/) service manager (Centos7 at the moment):

```
systemctl start mysql@bootstrap.service
```

In older PXC versions, to bootstrap cluster, you had to edit my.cnf and replace previous wsrep_cluster_address line with empty value like this: wsrep_cluster_address=gcomm:// and start mysql normally. More details to be found here (http://galeracluster.com/documentation-webpages/restartingcluster.html).

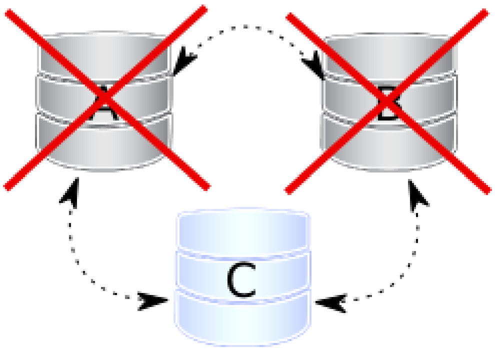
Please note that even if you bootstrap from the most advanced node, so the other nodes have lower sequence number, they will have to still join via full-SST because the Galera Cache is not retained on restart. For that reason, it is recommended to stop writes to the cluster *before* it's full shutdown, so that all nodes stop in the same position. Edit: This changes since Galera 3.19 thanks to gcache-recover (http://galeracluster.com/documentation-webpages/galeraparameters.html#gcache-recover) option.

(https://www.percona.com/blog/wp-content/uploads/2014/08/g4.png)Scenario 4



Node A disappears from the cluster. By disappear I mean power outage, hardware failure, kernel panic, mysqld crash, kill -9 on mysqld pid, OOMkiller, etc. Two remaining nodes notice the connection to A node is down and will be trying to re-connect to it. After some timeouts, both agree that node A is really down and remove it "officially" from the cluster. Quorum is saved (2 out of 3 nodes are up), so no service disruption happens. After restarting, A will join automatically the same way as in scenario 1.

Scenario 5



(https://www.percona.com/blog/wp-content/uploads/2014/08/g51.png)Nodes A and B disappear. The node C is not able to form the quorum alone, so the cluster is switching into a non-primary mode, in which MySQL refuses to serve any SQL query. In this state, mysqld process on C will be still running, you can connect to it, but any statement related to data fails with:

```
 mysql> select * from test.t1;
ERROR 1047 (08S01): Unknown command
```

Actually reads will be possible for a moment until C decides that it cannot reach A and B, but immediately no new writes will be allowed thanks to the certification based replication (http://galeracluster.com/documentation-webpages/certificationbasedreplication.html) in Galera. This is what we are going to see in the remaining node's log:

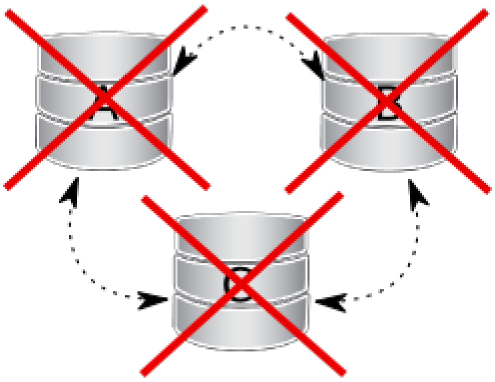
```
 140814 0:42:13 [Note] WSREP: commit failed for reason: 3
140814 0:42:13 [Note] WSREP: conflict state: 0
140814 0:42:13 [Note] WSREP: cluster conflict due to certification failure for
threads:
140814 0:42:13 [Note] WSREP: Victim thread:
THD: 7, mode: local, state: executing, conflict: cert failure, seqno: -1
SQL: insert into t values (1)
```

The single node C is then waiting for it's peers to show up again, and in some cases if that happens, like when there was network outage and those nodes were up all the time, the cluster will be formed again automatically. Also if the nodes B and C were just network-severed from the first node, but they can still reach each other, they will keep functioning as they still form the quorum. If A and B were crashed (due to data inconsistency, bug, etc.) or off due to power outage, you need to do manual action to enable primary component (http://galeracluster.com/documentation-webpages/weightedquorum.html#primary-component) on the C node, before you can bring A and B back. This way, we tell the C node “Hey, you can now form a new cluster alone, forget A and B!”. The command to do this is:

```
SET GLOBAL wsrep_provider_options='pc.bootstrap=true';
```

However, you should double check in order to **be very sure the other nodes are really down** before doing that! Otherwise, you will most likely end up with two clusters having different data.

Scenario 6



(https://www.percona.com/blog/wp-content/uploads/2014/08/g6.png)All nodes went down without proper shutdown procedure. Such situation may happen in case of datacenter power failure, hitting some MySQL or Galera bug leading to crash on all nodes, but also as a result of data consistency being compromised where cluster detects that each node has different data. In each of those cases, the grastate.dat file is not updated and does not contain valid sequence number (seqno). It may look like this:

```
 cat /var/lib/mysql/grastate.dat
# GALERA saved state
version: 2.1
uuid: 220dcdcb-1629-11e4-add3-aec059ad3734
seqno: -1
cert_index:
```

In this case, we are not sure if all nodes were consistent with each other, hence it is crucial to find the most advanced one in order to boostrap the cluster using it. Before starting mysql daemon on any node, you have to extract the last sequence number by checking it’s transactional state. You can do it this way:

```
 [root@percona3 ~]# mysqld_safe --wsrep-recover
140821 15:57:15 mysqld_safe Logging to '/var/lib/mysql/percona3_error.log'.
140821 15:57:15 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
140821 15:57:15 mysqld_safe WSREP: Running position recovery with --
log_error='/var/lib/mysql/wsrep_recovery.6bUIqM' --pid-file='/var/lib/mysql/percona3-
recover.pid'
140821 15:57:17 mysqld_safe WSREP: Recovered position 4b83bbe6-28bb-11e4-a885-
4fc539d5eb6a:2
140821 15:57:19 mysqld_safe mysqld from pid file /var/lib/mysql/percona3.pid ended
```

So the last committed transaction sequence number on this node was 2. Now you just need to bootstrap from the latest node first and then start the others.

However, the above procedure won't be needed in the recent Galera versions (3.6+?), available since PXC 5.6.19 (https://www.percona.com/doc/percona-xtradb-cluster/5.6/release-notes/Percona-XtraDB-Cluster-5.6.19-25.6.html). There is a new option – `pc.recovery` **(https://www.percona.com/doc/percona-xtradb-cluster/5.6/wsrep-provider-index.html#pc.recovery)** (enabled by default), which saves the cluster state into a file named `gvwstate.dat` on each member node. As the variable name says (pc – primary component), it saves only a cluster being in PRIMARY state. An example content of that file may look like this:

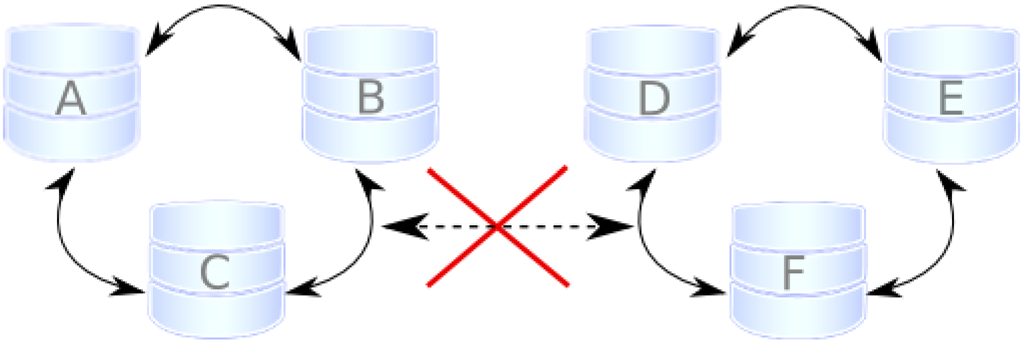
```
 cat /var/lib/mysql/gvwstate.dat
my_uuid: 76de8ad9-2aac-11e4-8089-d27fd06893b9
#vwbeg
view_id: 3 6c821ecc-2aac-11e4-85a5-56fe513c651f 3
bootstrap: 0
member: 6c821ecc-2aac-11e4-85a5-56fe513c651f 0
member: 6d80ec1b-2aac-11e4-8d1e-b2b2f6caf018 0
member: 76de8ad9-2aac-11e4-8089-d27fd06893b9 0
#vwend
```

We can see three node cluster above with all members being up. Thanks to this new feature, in the case of power outage in our datacenter, after power is back, the nodes will read the last state on startup and will try to restore primary component once all the members again start to see each other. This makes the PXC cluster to automatically recover from being powered down without any manual intervention! In the logs we will see:

```
 140823 15:28:55 [Note] WSREP: restore pc from disk successfully
(...)
140823 15:29:59 [Note] WSREP: declaring 6c821ecc at tcp://192.168.90.3:4567 stable
140823 15:29:59 [Note] WSREP: declaring 6d80ec1b at tcp://192.168.90.4:4567 stable
140823 15:29:59 [Warning] WSREP: no nodes coming from prim view, prim not possible
140823 15:29:59 [Note] WSREP: New COMPONENT: primary = no, bootstrap = no, my_idx = 2, memb_num = 3
140823 15:29:59 [Note] WSREP: Flow-control interval: [28, 28]
140823 15:29:59 [Note] WSREP: Received NON-PRIMARY.
140823 15:29:59 [Note] WSREP: New cluster view: global state: 4b83bbe6-28bb-11e4-a885-4fc539d5eb6a:11, view# -1: non-Primary, number of nodes: 3, my index: 2, protocol version -1
140823 15:29:59 [Note] WSREP: wsrep_notify_cmd is not defined, skipping notification.
140823 15:29:59 [Note] WSREP: promote to primary component
140823 15:29:59 [Note] WSREP: save pc into disk
140823 15:29:59 [Note] WSREP: New COMPONENT: primary = yes, bootstrap = yes, my_idx = 2, memb_num = 3
140823 15:29:59 [Note] WSREP: STATE EXCHANGE: Waiting for state UUID.
140823 15:29:59 [Note] WSREP: clear restored view
(...)
140823 15:29:59 [Note] WSREP: Bootstrapped primary 00000000-0000-0000-0000-000000000000 found: 3.
140823 15:29:59 [Note] WSREP: Quorum results:
version = 3,
component = PRIMARY,
conf_id = -1,
members = 3/3 (joined/total),
act_id = 11,
last_appl. = -1,
protocols = 0/6/2 (gcs/repl/appl),
group UUID = 4b83bbe6-28bb-11e4-a885-4fc539d5eb6a
140823 15:29:59 [Note] WSREP: Flow-control interval: [28, 28]
140823 15:29:59 [Note] WSREP: Restored state OPEN -> JOINED (11)
140823 15:29:59 [Note] WSREP: New cluster view: global state: 4b83bbe6-28bb-11e4-a885-4fc539d5eb6a:11, view# 0: Primary, number of nodes: 3, my index: 2, protocol version 2
140823 15:29:59 [Note] WSREP: wsrep_notify_cmd is not defined, skipping notification.
140823 15:29:59 [Note] WSREP: REPL Protocols: 6 (3, 2)
```

```
140823 15:29:59 [Note] WSREP: Service thread queue flushed.
140823 15:29:59 [Note] WSREP: Assign initial position for certification: 11, protocol
version: 3
140823 15:29:59 [Note] WSREP: Service thread queue flushed.
140823 15:29:59 [Note] WSREP: Member 1.0 (percona3) synced with group.
140823 15:29:59 [Note] WSREP: Member 2.0 (percona1) synced with group.
140823 15:29:59 [Note] WSREP: Shifting JOINED -> SYNCED (TO: 11)
140823 15:29:59 [Note] WSREP: Synchronized with group, ready for connections
```

Scenario 7



(https://www.percona.com/blog/wp-content/uploads/2014/08/g7.png)Cluster lost it’s primary state due to **split brain situation**. For the purpose of this example, let’s assume we have the cluster formed from even number of nodes – six and three of them are in one location while another three in second location (datacenter) and network connectivity is broken between them. Of course the best practice is to avoid such topology: if you can’t have odd number of real nodes, at least you can use an additional arbitrator (http://galeracluster.com/documentation-webpages/arbitrator.html) (garbd) node or set higher `pc.weight` (https://www.percona.com/doc/percona-xtradb-cluster/5.6/wsrep-provider-index.html#pc.weight) to some nodes. But when split brain happens any way, so none of the separated groups can maintain the quorum – all nodes must stop serving requests and both parts of the cluster are just continuously trying to re-connect. If you want to restore the service even before the network link is restored, you can make one of the groups primary again using the same command like in scenario 5:

```
SET GLOBAL wsrep_provider_options='pc.bootstrap=true';
```

After that, you are able to work on the manually restored part of the cluster, and the second half should be able to automatically re-join using incremental state transfer (http://galeracluster.com/documentation-webpages/statetransfer.html#incremental-state-transfer-ist) (IST) once the network link is restored. **But beware:** if you set the bootstrap option on both the separated parts, you will end up with two living cluster instances, with data likely diverging away from each other. Restoring network link in that case won’t make them to re-join until nodes are restarted and try to re-connect to members specified in configuration file. Then, as Galera replication model truly cares about data consistency – once the inconsistency will be detected, nodes that cannot execute row change statement due to different data – will perform emergency shutdown and the only way to bring them back to the cluster will be via full SST.

I hope I covered most of the possible failure scenarios of Galera-based clusters, and made the recovery procedures bit more clear.

Related

My Sessions at Percona Live MySQL Conference and Expo 2013 (https://www.percona.com/blo… janssens-sessions-at-percona-live-mysql-conference-and-expo-2013/)
April 1, 2013
In "Percona Live"

Better high availability: MySQL and Percona XtraDB Cluster with good application design (https://www.percona.com/blo… and-percona-xtradb-cluster-even-higher-availability-with-correct-application-design/)
December 21, 2015
In "High-availability"

State of the art: Galera - synchronous replication for InnoDB (https://www.percona.com/blo… of-the-art-galera-synchronous-replication-for-innodb/)
October 27, 2009
In "Insight for DBAs"

Przemysław Malkowski (/blog/author/przemek-malkowski/)

Przemek joined Support Team at Percona in August 2012. Before that he spent over five years working for Wikia.com (Quantcast Top 50) as System Administrator where he was a key person responsible for seamless building up MySQL powered database infrastructure. Besides MySQL he worked on maintaining all other parts of LAMP stack, with main focus on automation, monitoring and backups.

9 Comments

Martin

September 11, 2014 at 8:01 am (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-8716353)

Is it possbile to enable automatic recovery on MariaDB Galera Cluster 10.0.13 when all nodes go down without proper shutdown procedure (Scenario 6)?

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=8716353#respond)

Przemysław Malkowski

September 15, 2014 at 12:00 pm (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-8783226)

Martin,

I just checked with MariaDB Cluster 10.0.13 (latest available via MariaDB apt repository) and the pc.recovery option is not present:

```
root@node1:~# mysql -e “show variables like ‘wsrep_provider_options’\G”|sed ‘s/; /\n/g’|grep ‘pc.’
pc.announce_timeout = PT3S
pc.checksum = false
pc.ignore_quorum = false
pc.ignore_sb = false
pc.linger = PT20S
pc.npvo = false
pc.version = 0
pc.wait_prim = true
pc.wait_prim_timeout = P30S
pc.weight = 1
```

```
show status like ‘wsrep_provider_version’
+—————————————+——————————–+
| Variable_name | Value |
+—————————————+——————————–+
| wsrep_provider_version | 25.3.5-wheezy(rXXXX) |
+—————————————+——————————–+
```

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=8783226#respond)

Aray

November 14, 2014 at 11:08 am (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10012703)

Hi,

I have a 3 node PXC. Yesterday, one node of it crashed. Now it was running on two nodes. I tried to recover the crashed node but nothing helped so I decided to build it from scratch. I emptied the mysql data directory and started mysql on this crashed node. Doing this made the other two nodes in cluster unreachable and were shown as down. clustercheck was showing 503 n these servers. I stopped mysql on these two nodes but now I can only bootstrap one node and rest two are down. I know this is all messed up. Can I build the other two nodes by copying data from thrid running(bootstrapped) node? and will other two nodes start fine after data is copied?

Appreciate your help please.

Thanks,
Aray

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10012703#respond)

Nick (http://skylineservers.com)

November 19, 2014 at 5:48 pm (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10125920)

Thank you for this easy to follow and comprehensive post. Demystification goes such a long way towards making clustering more approachable. Just imagine what we'll need to demystify once clustering becomes the norm.

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10125920#respond)

Paweł

January 12, 2015 at 4:30 pm (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10372876)

What would happen to cluster if network was down between A-B, but A-C and B-C was fully functioning?

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10372876#respond)

Przemysław Malkowski

January 30, 2015 at 3:13 am (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10462184)

Paweł, Galera has a special ability to use alternative path for communication called message relaying. In this case, as both A and B can communicate with C, they will use C as a relay node and the cluster should continue functioning.

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10462184#respond)

George

February 13, 2015 at 12:09 pm (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10525631)

Hi,

thanks for your instructions on all these cases. I had a failure in two percona cluster nodes (crashed after power failure), and i managed to start only one of them using /etc/init.d/mysql start –wsrep_cluster_address=gcomm://

The other one cannot start saying
Starting MySQL (Percona XtraDB Cluster).The server quit wit[FAILED]ating PID file
(/percona/var/lib/mysql/san1.uk.arithon.com.pid).

MySQL (Percona XtraDB Cluster) server startup failed!

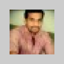
Is there anything else i can try ?

thanks,

George

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10525631#respond)

Sibin John



November 30, 2015 at 8:19 am (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10965639)

Hi ,
I configured 3 node percona cluster for cloudstack deployment and as per the above description the cluster will have to start automatically even if there is a power failure occur . But in my case after a down or failure of all the three nodes ,the cluster is not coming up .While booting the machine it is trying to make the cluster but failing ..showing as cluster failed to start , how can I resolve it . It creates a lot of problems for my environment ,since it is going make accessible by public . Expecting your quick and relevant solution …

Thanks and regards,
SIbin John

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10965639#respond)

Przemysław Malkowski

December 16, 2015 at 7:00 pm (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/#comment-10965701)

Blog comments section is not suitable for certain cases trouble shooting. I'd suggest using our forums: https://www.percona.com/forums/questions-discussions/percona-xtradb-cluster

Reply (https://www.percona.com/blog/2014/09/01/galera-replication-how-to-recover-a-pxc-cluster/?replytocom=10965701#respond)

Leave a Reply

Enter your comment here...