DRBD - Maintenance and Resolve Split Brain or Node Errors

Caution

This is an advanced topic. Use at your own risk and **ALWAYS** backup your data before.

Useful Commands

View DRBD Status - DRBD 7

```
cat /proc/drbd
```

View DRBD Status - DRBD 9

```
drbdadm status
```

Reload all parameters

```
drbdadm adjust jtelshared
```

Disconnect the share (useful for planned maintenance)

```
drbdadm disconnect jtelshared
```

Down the share (useful for planned maintenance)

```
drbdadm down jtelshared
```

Up the share

```
drbdadm up jtelshared
```

Set the node to primary

```
drbdadm primary jtelshared
```

Connect the share

```
drbdadm connect jtelshared
```

PCS Cluster Commands (CentOS 8)

```
pcs cluster stop acd-store2
pcs cluster start acd-store2

pcs node standby acd-store2
pcs node unstandby acd-store2
```

Split Brain

Background

See also:

https://docs.linbit.com/doc/users-guide-84/s-resolve-split-brain/

Symptoms - CentOS 7 and earlier

cat /proc/drbd

```
cat /proc/drbd

-->

GIT-hash: a4d5de01fffd7e4cde48a080e2c686f9e8cebf4c build by mockbuild@, 2017-09-15 14:23:22
1: cs:StandAlone ro:Primary/Unknown ds:UpToDate/DUnknown r-----
 ns:0 nr:119823323 dw:119823323 dr:2128 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

cs:StandAlone means the node is not connected.

This should be visible on both sides.

Symptoms - CentOS 8 and earlier

drbdadm status

```
drbdadm status

-->

jtelshared role:Primary
  disk:UpToDate
  acd-store1 connection:Connecting

drbdadm status

-->

# No currently configured DRBD found.
```

The first command shows that DRBD is active on the first node, but not active on the second node.

Note: this can be due to the second node being stopped or in standby.

Find out which node is active in the PCS cluster - CentOS 7

pcs status

```
pcs status

-->

Cluster name: portal

Stack: corosync
Current DC: acd-store1 (version 1.1.16-12.el7_4.7-94ff4df) - partition with quorum
Last updated: Sun Mar 18 18:05:32 2018
Last change: Fri Feb 16 00:07:51 2018 by root via cibadmin on acd-store2
2 nodes configured
3 resources configured
Node acd-store1: standby
Online: [ acd-store2 ]
Full list of resources:
Resource Group: haproxy_group
 ClusterDataJTELSharedMount (ocf::heartbeat:Filesystem): Started acd-store2
 ClusterIP (ocf::heartbeat:IPaddr2): Started acd-store2
 samba (systemd:smb): Started acd-store2
Daemon Status:
 corosync: active/enabled
 pacemaker: active/enabled
 pcsd: active/enabled
```

In the example above, the first node is in standby. The most important thing to check, is on which server the resources are started.

In this case, the resources are started on acd-store2.

This will therefore be defined as the NON BROKEN node.

Find out which node is active in the PCS cluster - CentOS 8

pcs status

```
pcs status

-->

Cluster name: jtel_cluster
Cluster Summary:
  * Stack: corosync
  * Current DC: acd-lb1 (version 2.0.3-5.el8_2.1-4b1f869f0f) - partition with quorum
  * Last updated: Sat Oct  3 12:39:22 2020
  * Last change:  Sat Oct  3 12:31:22 2020 by root via cibadmin on acd-lb2
  * 2 nodes configured
  * 5 resource instances configured

Node List:
  * Online: [ acd-lb1 ]
  * OFFLINE: [ acd-lb2 ]

Full List of Resources:
  * Clone Set: DRBDClusterMount-clone [DRBDClusterMount] (promotable):
    * Masters: [ acd-lb1 ]
    * Stopped: [ acd-lb2 ]
  * DRBDClusterFilesystem       (ocf::heartbeat:Filesystem):    Started acd-lb1
  * Samba       (systemd:smb):  Started acd-lb1
  * ClusterIP   (ocf::heartbeat:IPaddr2):       Started acd-lb1

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

In the example above, the second node is offline. The most important thing to check, is on which server the resources are started.

In this case, the resources are started on acd-lb1.

This will therefore be defined as the NON BROKEN node.

Standby the broken node in the PCS cluster (if necessary)

This command can be run on either machine.

CentOS 7

Standby broken node

```
pcs cluster standby acd-lb-broken

--> Verify this with

pcs status
```

CentOS 8

Standby broken node

```
pcs node standby acd-lb-broken

--> Verify this with

pcs status
```

On broken node

Note: the first command will probably throw an error. Also, the share may not be mounted. This is OK.

drbd on broken node

```
umount /srv/jtel/shared
drbdadm disconnect jtelshared
drbdadm secondary jtelshared
drbdadm connect --discard-my-data jtelshared
```

On the healthy node

drbd on healthy node

```
drbdadm primary jtelshared
drbdadm connect jtelshared
```

Check re-sync activity

The re-sync might take a long time.

Watch the status of this using:

`cat /proc/drbd`

Example output:

cat /proc/drbd

```
[root@storage01 ~]# cat /proc/drbd
version: 8.4.10-1 (api:1/proto:86-101)
GIT-hash: a4d5de01fffd7e4cde48a080e2c686f9e8cebf4c build by mockbuild@, 2017-09-15 14:23:22
1: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
 ns:0 nr:1411538 dw:121234862 dr:2128 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:184698664
 [>....................] sync'ed: 0.8% (180368/181744)M
 finish: 26:12:15 speed: 1,940 (2,760) want: 2,120 K/sec
```

Tune the transfer (Second Node) - Only CentOS 7.x

Currently there is no procedure for tuning the transfer on CentOS 8.x

If the transfer is going to take ages, then tune it on the broken node:

drbd Transfer Tuning (on broken node)

```
drbdadm disk-options --c-plan-ahead=0 --resync-rate=110M jtelshared
```

Put broken node back to primary - **CentOS 7.x ONLY**

Do not do this on CentOS 8.x installations! Here DRBD is managed by the cluster.

Unstandby broken node

```
drbdadm primary jtelshared

--> Verify this with

cat /proc/drbd
```

Restart PCS node

CentOS 7.x

Unstandby broken node

```
pcs cluster unstandby acd-lb-broken

--> Verify this with

pcs status
```

CentOS 8.x

Unstandby broken node

```
pcs cluster start acd-lb-broken
pcs node unstandby acd-lb-broken

--> Verify this with

pcs status
```

Untune the transfer (Second Node) - CentOS 7x only

If the transfer was tuned, then untune it (on the broken node).

Note: it won't hurt to run this command anyway.

drbd - Untune Transfer

```
drbdadm adjust jtelshared
```

Check everything

Check everything

```
pcs status
# CentOS 7.x
cat /proc/drbd
# CentOS 8.x
drbdadm status
# On some other linux machines
ls /home/jtel/shared
# Windows
dir //acd-store/shared
```

File System Corrupt

Sometimes, when DRBD fails, the file system will also become corrupt.

In this case both nodes might be primary, however neither will have the share mounted.

The command **mount /srv/jtel/shared** will fail.

In this case, it may be necessary to repair the file system.

Symptoms

```
[17354513.483526] XFS (drbd1): log mount/recovery failed: error -22
[17354513.483569] XFS (drbd1): log mount failed
[17355040.104433] XFS (drbd1): Mounting V5 Filesystem
[17355040.122234] XFS (drbd1): Corruption warning: Metadata has LSN (56:112832) ahead of current LSN (56:112733). Please unmount and run xfs_repair (>= v4.3) to
resolve.
[17355040.122239] XFS (drbd1): log mount/recovery failed: error -22
[17355040.122322] XFS (drbd1): log mount failed
```

Repairing

One one of the nodes (need to choose one to become primary):

```
xfs_repair /dev/drbd/by-res/jtelshared/0
pcs resource cleanup
```

This should then mount and start the resources on that node.

Then proceed with the other node as "broken" in the split brain situation.

Stalled Resync

If the DRBD resync stalls - the output will be "stalled" when **cat /proc/drbd** is executed - then it may be necessary to restart the machine.

This has been observed once, and restarting resolved the situation. However not much more is known about this state, or the cause, at this time.

Failed Connect (Unrelated data, aborting)

When the secondary has been told to discard it's data, and all of the commands to start the sync have been entered on both the healthy and the broken node, sometimes **cat /proc/drbd** will not report a connection.

Check **/var/log/messages**

If you can see output like this:

```
kernel: block drbd0: uuid_compare()=-1000 by rule 100
kernel: block drbd0: Unrelated data, aborting!
```

Then the metadata has become corrupt.

This requires that the metadata be completely reconstructed on the bad node.

Use the following commands to recreate the data on the broken node:

```
drbdadm down jtelshared
drbdadm wipe-md jtelshared
drbdadm create-md jtelshared
```

Then proceed with the re-sync as above (start with the part "On the broken node" with the commands to place this in secondary and discard data.

Failed Connect (Unknown Connection)

This produces errors something like this:

```
??: Failure: (162) Invalid configuration request
additional info from kernel:
unknown connection
```

In this case, drbd might not be loaded and enabled.

Execute the following code on the broken node and then proceed as above:

```
modprobe drbd
systemctl enable drbd
systemctl start drbd
```