NAME

crm_startup_fencing – how the cluster should act on startup

DESCRIPTION

The crm basic parameter **startup-fencing**, together with others, defines how the cluster should act on startup. Corosync and pacemaker both have parameters for controlling the behaviour on cluster start.

For corosync the parameters are two_node and wait_for_all. For pacemaker the parameters are no-quorum-policy, startup-fencing, expected-quorum-votes, and dc-deadtime.

Of course, first of all it is important whether the cluster service is started on boot automatically or not.

On startup each cluster node tries to see all other known nodes.

TODO

OPTIONS

startup-fencing = true

If a node is allowed to act (by having quorum or by no-quorum-policy), it will fence all unseen known nodes after becoming DC. This is to solve potential split-brain in order to start resources. This is the default.

startup-fencing = false

If a node is allowed to act (by having quorum or by no-quorum-policy), it will ignore unseen known nodes and just start resources. The cluster does not protect resources in case of split-brain during cluster start. This is not recommended.

EXAMPLES

*** Two-node cluster, SBD STONITH**

Common configuration for the widely used Simple Stack, as well as for SAP Enqueue Replication, SAP HANA System Replication scale-up, and IBM DB2 HADR.

With SBD, startup-fencing has no immediate consequence if the fenced node is up without the cluster running. In this case a later start of the cluster on the fenced node will be blocked by SBD. So, the resources are save against split-brain.

TODO pacemaker crm:
startup-fencing=true
no-quorum-policy=ignore

corosync conf:
two_node: 1
wait_for_all: 0

*** Two-node cluster, remote management board STONITH**

Server hardware (e.g. iLO, IPMI, DRAC) falls in this category, as well as server management consoles (e.g. HMC), hypervisors, virtualisation management (e.g. vCenter, OpenStack), or public cloud management APIs.

With remote management board, startup-fencing will result in immediate reboot or shutdown of the fenced node.

From all possible combinations of options, two setups might make sense.

a. Cluster will wait for all nodes on startup

```
TODO pacemaker crm:
startup-fencing=true

corosync conf:
two_node: 1
wait_for_all: 1
```

b. Cluster will fence unseen node and start resources

```
TODO pacemaker crm:
startup-fencing=true
no-quorum-policy=ignore

corosync conf:
two_node: 1
wait_for_all: 0
```

FILES

/usr/share/ClusterTools2/samples/

samples of config files and scripts.

BUGS

Feedback is welcome, please use the project page at
https://build.opensuse.org/package/show?package=ClusterTools2&project=home%3Afmherschel

SEE ALSO

crm(8), **cs_prepare_crm_basics**(8), **stonith**(8), **stonith_sbd**(7), **sbd**(8), **crm_simulate**(8), **corosync-cfgtool**(8), **corosync.conf**(7), **ClusterTools2**(7), **crm_no_quorum_policy**(7),
http://www.suse.com/documentation/sle_ha/ ,
http://www.clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/ch05s03.html ,

AUTHORS

The content of this manual page was mostly derived from online documentation mentioned above.

COPYRIGHT