RHCS6: ’fencing’ basics

Article Number: 194 | Rating: Unrated | Last Updated: Sun, Jun 3, 2018 9:31 AM

RHCS: 'fencing' basics

```
# Tested on RHEL 6



# The act of 'fencing' is the process where one cluster node will be cut off from access to
# shared storage by the other cluster nodes. This can be done either at the power level or
# at the storage level (we'll see only power level fencing).

# This step is necessary in order to recover service in situations where a node becomes
# non-responsive. Although this seems aggressive it may be the only way to guarantee storage,
# and thus data, integrity.

# Power off fencing can be done using a network-controlled power strip or by using a remote
# management device like ILO or DRAC.

# When using power fencing we can choose between turning off the target server or turning it
# off and then on again. Usually we will turn it back so it can re-join the cluster in a clean
# status




# Fencing config for virtual servers on RHEL hypervisor with libvirt - Libvirt fencing
# -------------------------------------------------------------------
```

```
-----------------------

# 1.- On hypervisor, install required packages

   yum install fence-virtd
   yum install fence-virtd-libvirt
   yum install fence-virtd-multicast

# 2.- Create key file on hypervisor

   mkdir /etc/cluster
   dd if=/dev/urandom of=/dev/cluster/fence_xvm.key bs=1k count=4

# 3.- Configure fence_virt to listen on "private" network using
'multicast' and 'libvirt'
# backend

   fence_virt -c   # (accept -Enter- every default answer except
following ones)
      Interface [none]: choose private network
      Backend module [checkpoint]: libvirt

# 4.- Start and enable fence daemon on the hypervisor

   chkconfig fence_virtd on
   service fence_virtd start

# 5.- Copy key file to all virtual nodes forming the cluster (same
location, owner
#     and rights)

# 6.- Create a fence device via Luci; type "Fence Virt (Multicast
Mode)", this will add
# following lines to our cluster.conf file (modifications can be
written directly to
# configuration file; in this case don't forget to spread the
configuration to all nodes):
```

```
<?xml version="1.0"?>
<cluster config_version="12" name="mycluster">
   <clusternodes>
      <clusternode name="nodeA" nodeid="1"/>
      <clusternode name="nodeB" nodeid="2"/>
   </clusternodes>
   <cman expected_votes="1" two_node="1">
      <multicast addr="239.192.XX.XXX"/>
   </cman>
   <fencedevices>
      <fencedevice agent="fence_xvm" name="myfencedevice"/>
   </fencedevices>
   <rm log_level="7"/>
</cluster>

# 7.- On Luci, for every node, "Add Fence Method" and, then, "Add
Fence Instance", that
# will add the following to the config file (modifications can be
written directly to
# configuration file; in this case don't forget to spread the
configuration to all nodes):

<?xml version="1.0"?>
<cluster config_version="21" name="mycluster">
   <clusternodes>
      <clusternode name="nodeA" nodeid="1">
         <fence>
            <method name="myfencemethod">
               <device domain="nodeA" name="myfencedevice"/>
            </method>
         </fence>
      </clusternode>
      <clusternode name="nodeB" nodeid="2">
         <fence>
            <method name="myfencemethod">
               <device domain="nodeB" name="myfencedevice"/>
            </method>
         </fence>
      </clusternode>
   </clusternodes>
   <cman expected_votes="1" two_node="1">
      <multicast addr="239.192.XX.XXX"/>
   </cman>
   <fencedevices>
      <fencedevice agent="fence_xvm" name="myfencedevice"/>
   </fencedevices>
   <rm log_level="7"/>
</cluster>

# where "domain" is the name of the virtual machine in kvm, not the hostname or dns
# domain name of the cluster node




# Fencing config for IBM Blade servers
# ------------------------------------------------------------------------------------------

# Requirements:
#
# - IPs for Blade chassis holding the Blades servers forming our cluster
# - Port (slot) of each Blade server forming the cluster on the Blade chassis
# - User/password on Blade chassis with enough permissions to power-off Blade servers

   <clusternodes>
      <clusternode name="nodeA" nodeid="1" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeA" port="8"/>
            </method>
         </fence>
      </clusternode>
      <clusternode name="nodeB" nodeid="2" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeB" port="6"/>
            </method>
         </fence>
      </clusternode>
   </clusternodes>

   <fencedevices>
      <fencedevice agent="fence_bladecenter" ipaddr="XX.XXX.XXX.37"
login="FenceUser" name="fence_nodeA" passwd="FenceUser_pwd"/>
      <fencedevice agent="fence_bladecenter" ipaddr="XX.XXX.XXX.58"
login="FenceUser" name="fence_nodeB" passwd="FenceUser_pwd"/>
   </fencedevices>




# Fencing config for HP Proliant servers
# -------------------------------------------------------------------
-----------------------

# Requirements:
#
# - iLO IPs of servers
# - User/password on servers with enough permissions to do a power-
off

   <clusternodes>
      <clusternode name="nodeA" nodeid="1" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeA" port="nodeA"/>
            </method>
         </fence>
      </clusternode>
      <clusternode name="nodeB" nodeid="2" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeB" port="nodeB"/>
            </method>
         </fence>
      </clusternode>
   </clusternodes>

   <fencedevices>
      <fencedevice agent="fence_ipmilan" ipaddr="XX.XXX.XXX.37"
lanplus="1" login="FenceUser" name="fence_nodeA"
passwd="FenceUser_pwd"/>
      <fencedevice agent="fence_ipmilan" ipaddr="XX.XXX.XXX.58"
lanplus="1" login="FenceUser" name="fence_nodeB"
passwd="FenceUser_pwd"/>
   </fencedevices>


# Trick: to pre-check connection to fencing device on HP Proliant
servers we can use following command

ipmitool -H <iLO_IP> -I lanplus -U <FenceUser> -P <FenceUser_pwd>
chassis power status




# Fencing config for VMWare virtual servers
# -------------------------------------------------------------------
-----------------------

# Requirements:
#
# - IPs of vCenters hosting our virtual servers
# - User/password on vCenters enough permissions to do a power-off of
virtual servers
# - Names of datacenters virtual servers belong to on its
```

```
corresponding vCenter server

   <clusternodes>
      <clusternode name="nodeA" nodeid="1" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeA" port="nodeA" ssl="1"/>
            </method>
         </fence>
      </clusternode>
      <clusternode name="nodeB" nodeid="2" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeB" port="nodeB" ssl="1"/>
            </method>
         </fence>
      </clusternode>
      <clusternode name="nodeC" nodeid="3" votes="1">
         <fence>
            <method name="myfencemethod">
               <device name="fence_nodeC" port="nodeC" ssl="1"/>
            </method>
         </fence>
      </clusternode>
   </clusternodes>

   <fencedevices>
      <fencedevice agent="fence_vmware"
ipaddr="vcenter1.mydomain.com" login="FenceUser" name="fence_nodeA"
passwd="FenceUser_pwd" wmware_datacenter="Datacenter1"/>
      <fencedevice agent="fence_vmware"
ipaddr="vcenter1.mydomain.com" login="FenceUser" name="fence_nodeB"
passwd="FenceUser_pwd" vmware_datacenter="Datacenter2"/>
      <fencedevice agent="fence_vmware"
ipaddr="vcenter2.mydomain.com" login="FenceUser" name="fence_nodeC"
passwd="FenceUser_pwd" vmware_datacenter="Datacenter1"/>
   </fencedevices>
```

```
# Show fencing configuration
# ---------------------------------------------------------------------
-----------------------

ccs -h nodeA -p myriccipasswd --lsfenceinst
   nodeA
  fence_nodeA
  myfencedevice: domain=nodeA
   nodeB
  fence_nodeB
  myfencedevice: domain=nodeB




# Testing fencing
# ---------------------------------------------------------------------
-----------------------

# We can test fencing by either stopping all network interfaces on a
node

service network stop

# or running fence_node command from another node

fence_node <nodeB>

# If you only want to do a connection test, you can run on of
following commands
# (depending on HW):


fence_bladecenter -o status -a <IP> -l <FenceUser> -p <FenceUser_pwd>
-n <port_nmb>
```

```
fence_ilo -o status -a <IP> -l <FenceUser> -p <FenceUser_pwd>
fence_vmware -o status -a <IP> -l <FenceUser> -p <FenceUser_pwd> -n
<node_name>




# Two important options when configuring fencing on a Red Hat cluster
are 'post_fail_delay'
# and 'post_join_delay':

# 'post_join_delay': Number of seconds fenced will delay before
fencing any victims
# after nodes join the domain. This delay gives nodes that have been
tagged for fencing
# a chance to join the cluster and avoid being fenced. A delay of -1
here will cause the
# daemon to wait indefinitely for all nodes to join the cluster and
no nodes will actually
# be fenced on startup. This attribute only applies when a node is
joining a cluster,
# existing cluster members will not trigger the post_join_delay
timer.

# 'post_fail_delay': Number of seconds fenced will delay before
fencing a domain member
# that has failed. A cluster node will not be fenced if it tries to
rejoin the cluster
# before post_fail_delay completes, if it is joining after a reboot
or restarting cman.
# The post_fail_delay is 0 by default to minimize the time that other
systems are blocked
# from fencing.
# All cluster operations such as fencing a cluster node, handing out
new locks for GFS or
# GFS2, and relocating services will be blocked until the
post_fail_delay timer has
# completed. There is no risk for GFS or GFS2 corruption since new
locks will not be
```

```
# granted until fencing is complete which occurs after
post_fail_delay timer has completed.


# To set these parameters, execute:

ccs -h nodeA -p myriccipasswd --setfencedaemon post_join_delay=300
ccs -h nodeA -p myriccipasswd --setfencedaemon post_fail_delay=15
```

Posted - Sun, Jun 3, 2018 9:31 AM. This article has been viewed 2164 times.

Online URL: http://kb.ictbanking.net/article.php?id=194