

- How to configure the Fencing on Pacemaker? .....** 3
- (            ) .....
- fence\_xvm                    KVM .....
- .....
- STONITH .....
- .....



# How to configure the Fencing on Pacemaker?

: <https://www.unixarena.com/2016/01/rhel-7-configure-fencing-pacemaker.html/>

(STONITH)

가 가

“ ” ( : , ) 가

가

pacemaker

- (Resource Level Fencing)
- (Node Level Fencing)

Resource level fencing 가

.Node level fencing 가

가

“STONITH”(Shoot The Other Node In The Head;

. Pacemaker / corosync )

[clusterlabs.org](http://clusterlabs.org)

```
[root@Node1-LAB ~]# pcs status
Cluster name: GFSCCLUS
Last updated: Wed Jan 20 12:43:36 2016
Last change: Wed Jan 20 09:57:06 2016 via cibadmin on Node1
Stack: corosync
Current DC: Node1 (1) - partition with quorum
Version: 1.1.10-29.el7-368c726
2 Nodes configured
2 Resources configured
Online: [ Node1 Node2 ]
PCSD Status:
Node1: Online
Node2: Online
Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
[root@Node1-LAB ~]#
```

KVM

fence\_xvm

stonith/fencing

STONITH / FENCING

( )

- node1 node2 - pacemaker / corosync
- UNIXKB-CP - Node1 Node2 KVM

## fence\_xvm KVM

1. KVM . 2. 가 .

```
[root@UNIXKB-CP ~]# virsh list
Id      Name           State
-----
6       Node1          running
7       Node2          running
[root@UNIXKB-CP ~]#
```

3. KVM ( ) .

```
[root@UNIXKB-CP ~]# yum install fence-virt fence-virtfd fence-virtfd-libvirt
fence-virtfd-multicast fence-virtfd-serial
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can
use subscription-manager to register.
Package fence-virt-0.3.0-16.el7.x86_64 already installed and latest version
Package fence-virtfd-0.3.0-16.el7.x86_64 already installed and latest version
Package fence-virtfd-libvirt-0.3.0-16.el7.x86_64 already installed and latest
version
Package fence-virtfd-multicast-0.3.0-16.el7.x86_64 already installed and
latest version
Package fence-virtfd-serial-0.3.0-16.el7.x86_64 already installed and latest
version
Nothing to do
[root@UNIXKB-CP ~]#
```

4. . .

```
[root@UNIXKB-CP ~]# mkdir -p /etc/cluster
[root@UNIXKB-CP ~]# cd /etc/cluster/
[root@UNIXKB-CP cluster]# dd if=/dev/urandom of=/etc/cluster/fence_xvm.key
bs=4k count=1
1+0 records in
1+0 records out
4096 bytes (4.1 kB) copied, 0.000506736 s, 8.1 MB/s
[root@UNIXKB-CP cluster]#
```

5. `scp -r /etc/cluster/fence_xvm.key root@Node1:/etc/cluster/fence_xvm.key`

```
[root@UNIXKB-CP cluster]# scp -r /etc/cluster/fence_xvm.key
root@Node1:/etc/cluster/fence_xvm.key
root@node1's password:
fence_xvm.key
100% 4096    4.0KB/s   00:00
[root@UNIXKB-CP cluster]# scp -r /etc/cluster/fence_xvm.key
root@Node2:/etc/cluster/fence_xvm.key
root@node2's password:
fence_xvm.key
100% 4096    4.0KB/s   00:00
[root@UNIXKB-CP cluster]#
```



: xvm

/etc/cluster

6. `fence_virttd -c /etc/fence_virt.conf`

```
[root@UNIXKB-CP ~]# fence_virttd -c
Module search path [/usr/lib64/fence-virt]:
Available backends:
libvirt 0.1
Available listeners:
multicast 1.2
Listener modules are responsible for accepting requests
from fencing clients.
Listener module [multicast]:
The multicast listener module is designed for use environments
where the guests and hosts may communicate over a network using
multicast.
The multicast address is the address that a client will use to
send fencing requests to fence_virttd.
Multicast IP Address [225.0.0.12]:
Using ipv4 as family.
Multicast IP Port [1229]:
Setting a preferred interface causes fence_virttd to listen only
on that interface. Normally, it listens on all interfaces.
In environments where the virtual machines are using the host
machine as a gateway, this *must* be set (typically to virbr0).
Set to 'none' for no interface.
Interface [virbr0]: br0:1
The key file is the shared key information which is used to
authenticate fencing requests. The contents of this file must
be distributed to each physical host and virtual machine within
```

```
a cluster.  
Key File [/etc/cluster/fence_xvm.key]:  
Backend modules are responsible for routing requests to  
the appropriate hypervisor or management layer.  
Backend module [libvirt]:  
Configuration complete.  
=== Begin Configuration ===  
backends {  
  libvirt {  
    uri = "qemu:///system";  
  }  
}  
listeners {  
  multicast {  
    port = "1229";  
    family = "ipv4";  
    interface = "br0:1";  
    address = "225.0.0.12";  
    key_file = "/etc/cluster/fence_xvm.key";  
  }  
}  
fence_virt {  
  module_path = "/usr/lib64/fence-virt";  
  backend = "libvirt";  
  listener = "multicast";  
}  
=== End Configuration ===  
Replace /etc/fence_virt.conf with the above [y/N]? y  
[root@UNIXKB-CP ~]#
```

가 . My setup br0:1 가 KVM

## 7. fence\_virt

```
[root@UNIXKB-CP ~]# systemctl enable fence_virt.service  
[root@UNIXKB-CP ~]# systemctl start fence_virt.service  
[root@UNIXKB-CP ~]# systemctl status fence_virt.service  
fence_virt.service - Fence-Virt system host daemon  
Loaded: loaded (/usr/lib/systemd/system/fence_virt.service; enabled)  
Active: active (running) since Wed 2016-01-20 23:36:14 IST; 1s ago  
Process: 3530 ExecStart=/usr/sbin/fence_virt $FENCE_VIRT_ARGS  
(code=exited, status=0/SUCCESS)  
Main PID: 3531 (fence_virt)  
CGroup: /system.slice/fence_virt.service  
└─3531 /usr/sbin/fence_virt -w  
Jan 20 23:36:14 UNIXKB-CP systemd[1]: Starting Fence-Virt system host  
daemon...  
Jan 20 23:36:14 UNIXKB-CP systemd[1]: Started Fence-Virt system host daemon.
```

```
Jan 20 23:36:14 UNIXKB-CP fence_virt[3531]: fence_virt starting.
Listener: libvirt Backend: multicast
[root@UNIXKB-CP ~]#
```

1. `rpm -qa fence-virt` 가

```
[root@Node1-LAB ~]# rpm -qa fence-virt
fence-virt-0.3.0-16.el7.x86_64
[root@Node1-LAB ~]#
```

3.

```
[root@Node1-LAB ~]# fence_xvm -o list
Node1          6daac670-c494-4e02-8d90-96cf900f2be9 on
Node2          17707dcb-7bcc-4b36-9498-a5963d86dc2f on
[root@Node1-LAB ~]#
```

4. `cat /etc/hosts` .

```
[root@Node1-LAB ~]# cat /etc/hosts |grep Node
192.168.2.10    Node1-LAB Node1
192.168.2.11    Node2-LAB Node2
[root@Node1-LAB ~]#
```

5. pacemaker cluster `fence_xvm fence agent` .

```
[root@Node1-LAB ~]# pcs stonith create xvmfence fence_xvm
key_file=/etc/cluster/fence_xvm.key
[root@Node1-LAB ~]#
[root@Node1-LAB ~]# pcs stonith
xvmfence      (stonith:fence_xvm):      Started
[root@Node1-LAB ~]#
[root@Node1-LAB ~]# pcs stonith --full
Resource: xvmfence (class=stonith type=fence_xvm)
Attributes: key_file=/etc/cluster/fence_xvm.key
Operations: monitor interval=60s (xvmfence-monitor-interval-60s)
[root@Node1-LAB ~]#
```

RHEL7 - Pacemaker/Corosync  
KVM

).

## STONITH

stonith ? 가 .

1. . 2. .

```
[root@Node1-LAB ~]# pcs stonith fence Node2
Node: Node2 fenced
[root@Node1-LAB ~]#
```

Node2가 . 가 .

```
[root@Node1-LAB ~]# pcs property --all |grep stonith-action
stonith-action: reboot
[root@Node1-LAB ~]#
```

Stonith pcs ON / OFF .

```
[root@Node1-LAB ~]# pcs property --all |grep stonith-enabled
stonith-enabled: true
[root@Node1-LAB ~]#
```

- <https://www.unixarena.com/2016/01/rhel-7-configure-fencing-pacemaker.html/>

From:  
<https://at1.kr/dokuwiki/> - **AllThatLinux!**

Permanent link:  
[https://at1.kr/dokuwiki/doku.php/how\\_to\\_configure\\_the\\_fencing\\_on\\_pacemaker](https://at1.kr/dokuwiki/doku.php/how_to_configure_the_fencing_on_pacemaker)

Last update: **2018/09/11 05:21**

