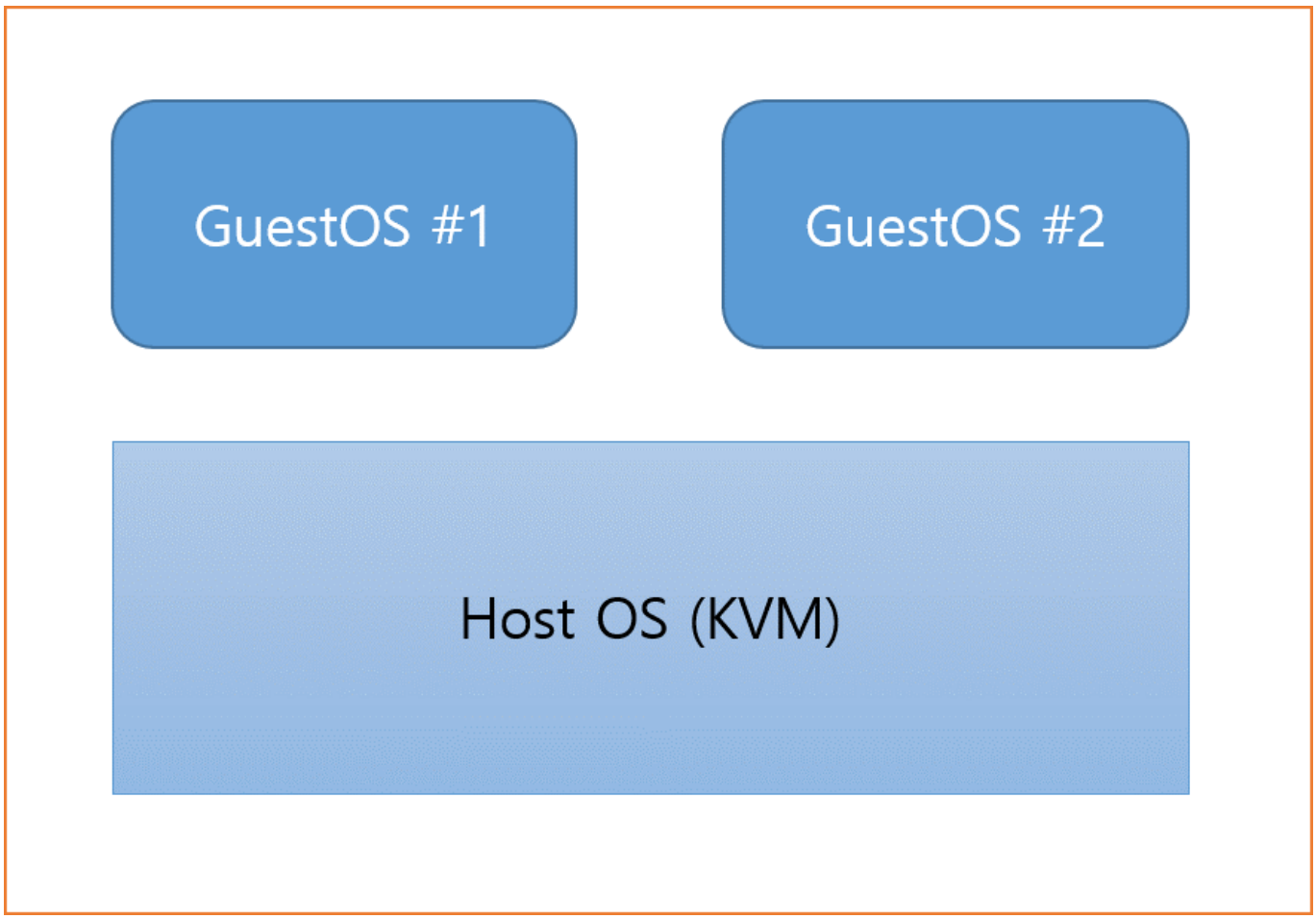
목차

KVM기반 fence_virtd 설정하기 3
기본 구조 3
Fence 설정 3
필요한 패키지 설치 3
키 생성 4
펜싱 데몬 구성 4
fence_virtd 시작 5
Guest OS 설정 5
cluster.conf 설정 6
펜싱 테스트 7
참조링크 7

KVM기반 fence_virtd 설정하기

기본 구조



기본 구성은 위와 같이 1대의 물리(가상) 호스트 머신위에 KVM으로 2대의 Guest OS를 생성하고 이 GuestOS를 HostOS의 fence_virtd를 통해 제어하도록 한다.

Fence 설정

클러스터가 구성되는데 있어 fence(이하 펜싱)설정은 매우 중요합니다. 펜싱 설정이 되어있지 않은 클러스터는 절대 권장되지 않습니다. 반드시 구성하여야 합니다.

필요한 패키지 설치

HostOS 에서 아래의 패키지를 설치하도록 한다.

```
# yum install fence-virt fence-virtd fence-virtd-libvirt fence-virtd-multicast fence-virtd-serial
```

설치가 완료되면 `/etc/cluster`디렉토리가 존재하는지 확인하고 없으면 아래의 명령어로 디렉토리를 생성한다.

```
# mkdir -p /etc/cluster
```

키 생성

모든 Host OS별로 다른 키를 가져야 함

펜싱 통신에 사용할 키를 생성하도록 한다. 이 키는 `HostOS`별로 생성하도록 한다. 펜싱 패킷을 날릴때 이 key를 기반으로 멀티캐스트에 응답하여 동작하는 방식이므로 같은 키를 다수의 Host OS에서 사용시 오동작 할 수 있습니다.

```
# dd if=/dev/urandom of=/etc/cluster/fence_xvm.key bs=4k count=1
# scp /etc/cluster/fence_xvm.key root@guest:/etc/cluster/fence_xvm_host1.key
```

HostOS에서 생성한 키를 모든 GuestOS에도 복사해준다.

펜싱 데몬 구성

Host OS에서 펜싱 데몬을 구성하도록 한다.

```
# fence_virtd -c
```

위의 명령어를 입력시 아래와같은 항목을 입력하게 된다.

- 디폴트 검색 경로를 받아 들인다.
- 멀티 캐스트를 기본값으로 허용
- 디폴트 멀티 캐스트 주소를 받아 들인다.
- 기본 멀티 캐스트 포트를 허용한다.
- 인터페이스를 br0으로 설정합니다 (브리지 이름을 호스트에 구성된 이름으로 바꿉니다)
- default fence_xvm.key 경로를 받아 들인다.
- 백엔드 모듈을 libvirt로 설정
- 기본 URI 수락
- 구성을 쓰려면 “y”를 입력하십시오.

이렇게 하면 아래와 비슷한 내용의 설정파일`/etc/fence_virt.conf`이 생성됩니다.

```
fence_virtd {
    module_path = "/usr/lib64/fence-virt";
    listener = "multicast";
```

```
    backend = "libvirt";
}

listeners {
    multicast {
        key_file = "/etc/cluster/fence_xvm.key";
        interface = "br0";
        port = "1229";
        address = "225.0.0.12";
        family = "ipv4";
    }
}

backends {
    libvirt {
        uri = "qemu:///system";
    }
}
```

Host OS 별로 multicast address는 다르게 설정해야함. 여러대의 Host OS인경우 아래와같이 설정하도록 한다.

```
    (on host1)
    address = "225.0.1.12";

    (on host2)
    address = "225.0.2.12";
```

fence_virtd 시작

```
# service fence_virtd start
# chkconfig fence_virtd on
```

Guest OS 설정

```
# yum install fence-virt
```

Guest OS에 패키지 설치가 되어있다면 아래의 명령어로 실제로 펜싱이 동작하는지 확인해 볼 수 있습니다.

```
[on guest1]$ fence_xvm -a 225.0.2.12 -k /etc/cluster/fence_xvm-host2.key -H
guest2 -o status
[on guest2]$ fence_xvm -a 225.0.1.12 -k /etc/cluster/fence_xvm_host1.key -H
guest1 -o status
```

실제로 펜싱 명령을 수행해보려면 아래와 같이 실행하면 된다.

```
[on guest2]# fence_xvm -o reboot -a 225.0.1.12 -k
/etc/cluster/fence_xvm_host1.key -H guest1
```

위의 `-H`옵션 뒤에 붙은 Guest OS의 이름은 KVM내에서 생성된 가상머신의 이름을 사용하여야 합니다.

HostOS에서 `virsh list`명령어로 가상머신의 이름을 확인할 수 있습니다.

```
[on host1]# virsh list
 Id Name                 State
----------------------------------
 1  guest1               running
```

cluster.conf 설정

펜싱 장치가 설정된 cluster.conf 파일예제는 아래와 같습니다.

```
[root@node2 ~]# cat /etc/cluster/cluster.conf
<?xml version="1.0"?>
<cluster config_version="1" name="kvm_cluster">
  <clusternodes>
    <clusternode name="node1.example.com" nodeid="1">
      <fence>
    <method name="1">
      <device domain="node1" name="virtfence1"/>
    </method>
      </fence>
    </clusternode>
    <clusternode name="node2.example.com" nodeid="2">
      <fence>
    <method name="1">
      <device domain="node2" name="virtfence2"/>
    </method>
      </fence>
    </clusternode>
  </clusternodes>
  <cman expected_votes="1" two_node="1"/>
  <fencedevices>
    <fencedevice agent="fence_xvm" name="virtfence1"
key_file="/etc/cluster/fence_xvm_host1.key" multicast_address="225.0.1.12"/>
    <fencedevice agent="fence_xvm" name="virtfence2"
key_file="/etc/cluster/fence_xvm_host2.key" multicast_address="225.0.2.12"/>
  </fencedevices>
  <rm>
    <failoverdomains/>
```

```
    <resources/>
  </rm>
</cluster>
```

펜싱 테스트

```
[on guest2]$ fence_node guest1
fence guest1 success
```

참조링크

- https://access.redhat.com/solutions/293183

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/kvm_%EA%B8%B0%EB%B0%98_fence_virtd_%EC%84%A4%EC%A0%95%ED%95%98%EA%B8%B0?rev=1514705863

Last update: **2017/12/31 07:37**