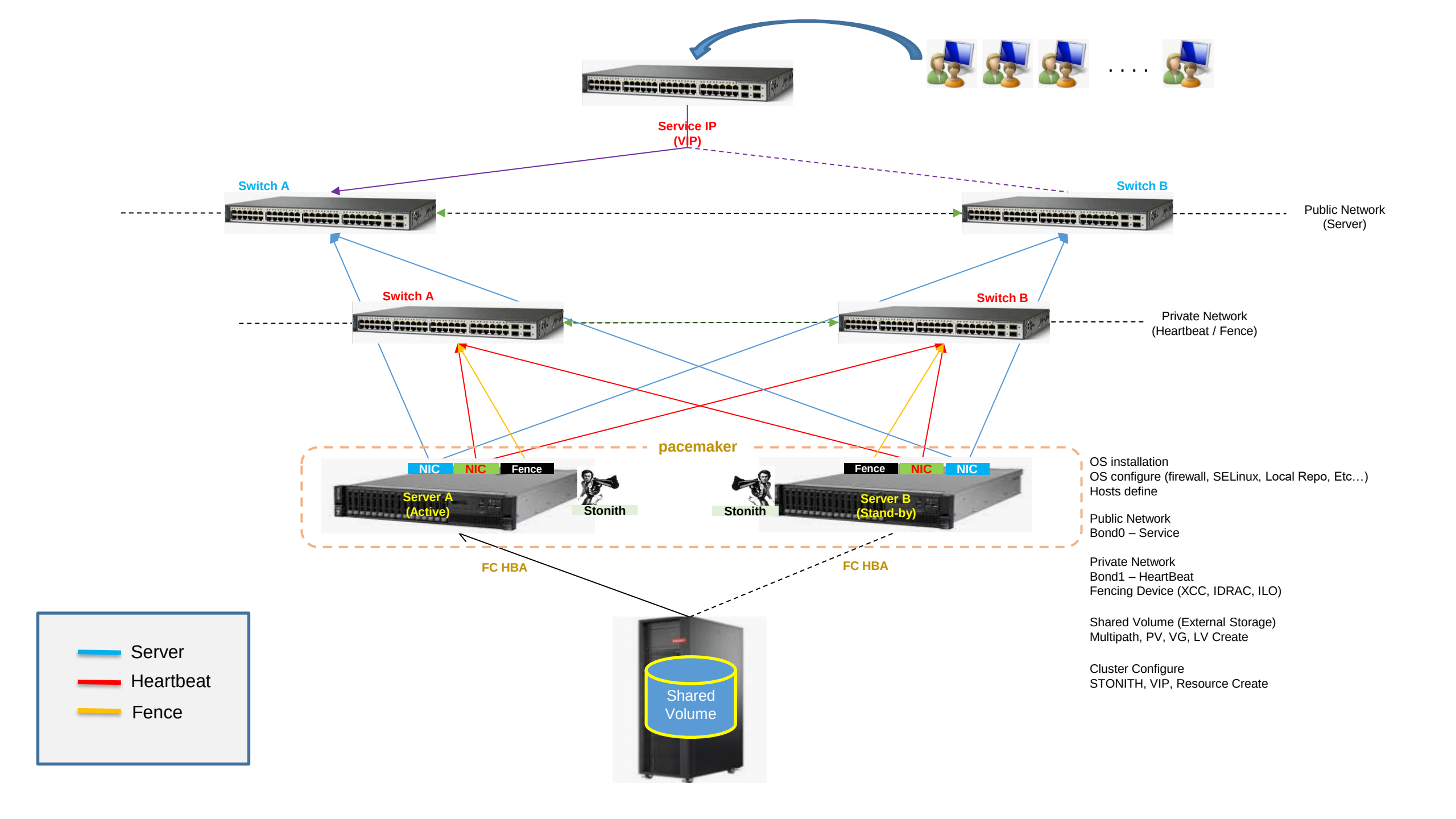
RedHat Linux 8
Pacemaker
(HA-LVM)


Service IP
(VIP)
Switch A
Switch B
Public Network
(Server)
Switch A
Switch B
Private Network
(Heartbeat / Fence)
pacemaker
NIC
NIC
Fence
Server A
(Active)
Stonith
Fence
NIC
NIC
Server B
(Stand-by)
Stonith
FC HBA
FC HBA
Shared
Volume
Server
Heartbeat
Fence
OS installation
OS configure (firewall, SELinux, Local Repo, Etc…)
Hosts define
Public Network
Bond0 – Service
Private Network
Bond1 – HeartBeat
Fencing Device (XCC, IDRAC, ILO)
Shared Volume (External Storage)
Multipath, PV, VG, LV Create
Cluster Configure
STONITH, VIP, Resource Create

1. hosts파일 설정 (node all)

[root@node1] # vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.56.35.200 GSMDDB_vip --VIP

10.56.35.201 GSMDDB01 --node1 IP

10.56.35.202 GSMDDB02 --node2 IP

10.10.8.1 GSMDDB01_fd1 --Stonith IP

10.10.8.2 GSMDDB02_fd2 --Stonith IP

10.10.8.3 GSMDDB01_hb1 --heartbeat IP

10.10.8.4 GSMDDB02_hb2 --heartbeat IP

Important!!

1. fence device로 사용할 IMM(XCC)은 ipmi over lan (623port) 을 enable 상태인지 확인해야 함 : ping과 무관하게 확인 해야함

Tip. fence device는 아래 command를 통해 정상적인 output을 보고 정상여부를 판단할수 있음..

ex> fence_ipmilan -P -a 10.10.8.1 -o status -v -l USERID -p PASSW0RD

2. heartbeat IP의 경우 양 노드사이 switch를 두고 연결되어 있어야함.
: 그렇지 않고 직결할 경우 한쪽이 disconnect되도 어느 쪽인지 불확실한 상황이 됨. => split brain 야기

2. firewalld 및 selinux 비활성화 (node all)

[root@node1] # systemctl stop firewalld

[root@node1] # systemctl disable firewalld

[root@node1] # vi /etc/selinux/config

This file controls the state of SELinux on the system.

SELINUX= can take one of these three values:

enforcing - SELinux security policy is enforced.

permissive - SELinux prints warnings instead of enforcing.

disabled - No SELinux policy is loaded.

SELINUX=disabled

SELINUXTYPE= can take one of three two values:

targeted - Targeted processes are protected,

minimum - Modification of targeted policy. Only selected processes are protected.

mls - Multi Level Security protection.

SELINUXTYPE=targeted

!! 위 처럼 firewalld 전부를 stop 하지 않을 경우

아래와 같이 필요한 port만 allow 할수 있다. (TCP 2224,TCP 3121,TCP 5403,UDP 5404,UDP 5405,TCP 21064,TCP 9929, UDP 9929)

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

3. yum repository 구성 (node all)

[root@node1] # vi /etc/yum.repos.d/local.repo/

```
[local_repo1]
name=AppStream
baseurl=file:///mnt/rhel8.6/AppStream
gpgcheck=0
enabled=1

[local_repo1]
name=BaseOS
baseurl=file:///mnt/rhel8.6/BaseOS
gpgcheck=0
enabled=1

[HighAvailability]
name=HighAvailability
baseurl=file:///mnt/rhel8.6/HA
gpgcheck=0
Enabled=1

~~[ResilientStorage]~~
~~name=ResilientStorage~~
~~baseurl=file:///mnt/rhel7.4/addons/ResilientStorage~~
~~gpgcheck=0~~
~~enabled=1~~
```

8.x 버전으로 넘어오면서 iso 파일이

OS img 와 HighAvailability img 두개로 나뉘어 졌음

ResilientStorage 의 yum 구성 여부는 확인 필요

별도 img 없음

4. 볼륨 생성

[root@node1 ~]# pvcreate /dev/sdc

Physical volume "/dev/sdc" successfully created.

[root@node1 ~]# vgcreate vgcluster /dev/sdc

Volume group "vgcluster" successfully created

[root@node1 ~]# lvcreate -L 400M -n lvcluster vgcluster

Logical volume "lvcluster" created.

[root@node1 ~]# mkfs.xfs /dev/vgcluster/lvcluster

```
meta-data=/dev/vgcluster/lvcluster isize=512    agcount=4, agsize=25600 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=102400, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5. 볼륨 구성 확인 (node all)

[root@node1 ~]# lvs

```
  LV       VG       Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root     rootvg   -wi-ao---- <15.78g
  swap     rootvg   -wi-ao---- 760.00m
  lvdata1  datavg   -wi-a------ 400.00m
```

[root@node1 ~]# vgs

```
  VG       #PV #LV #SN Attr   VSize   VFree
  rootvg     2   2   0 wz--n- <17.52g 1020.00m
  datavg     1   1   0 wz--n-  <8.00g   <7.61g
```

6. lvm2-lvmetad service및 socket 비활성화 (node all)

[root@node1 ~]# lvmconf --enable-halvm --services --startstopservices

Warning: Stopping lvm2-lvmetad.service, but it can still be activated by:

lvm2-lvmetad.socket

Removed symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmetad.socket.

[root@node1 ~]# systemctl mask lvm2-lvmetad.socket

Created symlink from /etc/systemd/system/lvm2-l

Important!!

/etc/lvm c

data

0 으로 설정하는 부분으로, 내용은 lvm meta

게 함으로 양 노드가 동일한 meta data를 가지게

게 되면 양 노드에 즉각적인 메타데이터 반영

적용방법 : 작업하지 않은 반대 노드에서 수행!

아서 import 되지 않았다는 메세지가 뜨나,

데이트 됨.

8.X 버전 에서 제거됨

NOTE:

lvm2-2.02.118-2보다 오래된 lvm2 rpm인 [redacted] 명령을 사용

```
# lvmconf --enable-halvm
# systemctl disable lvm2-lvmetad.service    ## RHEL 7
# systemctl disable lvm2-lvmetad.socket     ## RHEL 7
# systemctl stop lvm2-lvmetad.service       ## RHEL 7
# systemctl stop lvm2-lvmetad.socket        ## RHEL 7
# chkconfig lvm2-lvmetad off                ## RHEL 6
# service lvm2-lvmetad stop                 ## RHEL 6
```

7. 로컬 볼륨 그룹 volume_list 등록 (node all)

rootvg는 공유 볼륨 그룹이 아닌 로컬 그룹 이므로 lvm.conf 의 volume_list에 대해 rootvg를 등록

[root@node1 ~]# vgs

```
  VG       #PV #LV #SN Attr   VSize    VFree
  datavg     3   3   0 wz--n- <29.99g     0
  rootvg     1   3   0 wz--n- <277.27g    0          <- 로컬 볼륨
```

[root@node1 ~]# vi /etc/lvm/lvm.conf

...

locking_type = 1

....

.....

auto_activation_volume_list = ["rootvg"] <- node2 에도 적용 " volume_list = ["rootvg"] "

.....

....

Important!!
양 노드 전부 Cluster에서 관리 되지 않는 vg를 기입해야 하며,
pcs가 온라인 상황에서 **lvm작업이 필요한 경우** 주석처리를 해야 작업이 가능함!
(lvm.conf edit하는 경우 fence가 되기 때문에 작업이 필요한 경우 maintenance-mode를 true로 변경이 필요하다.
ex> pcs property set maintenance-mode=true

7.x버전의 tagging 방식에 대해 향후 제거할 계획은 없음. (Redhat 공식)

=> '글로벌 엔지니어를 통해 확인한 결과, 특별히 특정 방식을 제거할 계획은 없음.'
또한 7버전에서 사용하던 locking_type, lvm_used_metad 에서도 문의한 결과,
기존 RHEL7 에서 사용되던 **use_lvmetad 설정의 경우 RHEL8 에서 lvmetad 데몬과 함께 제거되었음**

=>locking_type 설정의 경우 RHEL8 에도 존재하지만,
디폴트 값인 1을 사용하기 때문에 구성 시 특별히 변경할 필요는 없음

=>8.x 버전에는 lvm.conf 에 system_id 값에 tagging 하는 방식으로 변경됨

다만, 기존 7.x 버전에서 사용하던 volume_list tagging 방식이
8.X 버전으로 바뀌면서 auto_activation_volume_list로 변경되었고 이를 병행 지원함

8. Initrd 장치 업데이트 (node all)

[root@node1 ~]#cp /boot/initramfs-3.10.0-693.el7.x86_64.img **/boot/**initramfs-3.10.0-693.el7.x86_64_**20210113backup**.img ==>업데이트 전에 날짜를 붙여 백업

[root@node1 ~]# dracut -f -v # dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)

[root@node1 ~]# ls -l /boot/

total 105440

-rw-r--r--. 1 root root 140894 Jul 6 2017 config-3.10.0-693.el7.x86_64

drwx------ 3 root root 16384 Dec 31 1969 efi

drwx------. 2 root root 21 Dec 30 23:28 grub2

-rw-------. 1 root root 51327078 Dec 30 23:27 initramfs-0-rescue-c8af0a0bde254094845d7e67a759a6e7.img

-rw-------. 1 root root 21318807 Dec 30 23:28 initramfs-3.10.0-693.el7.x86_64.img ← 변경된 날짜 & 시간 확인

-rw-------. 1 root root 19261780 Dec 30 23:34 initramfs-3.10.0-693.el7.x86_64kdump.img

-rw-r--r--. 1 root root 610412 Dec 30 23:26 initrd-plymouth.img

-rw-r--r--. 1 root root 293027 Jul 6 2017 symvers-3.10.0-693.el7.x86_64.gz

-rw-------. 1 root root 3228420 Jul 6 2017 System.map-3.10.0-693.el7.x86_64

-rwxr-xr-x. 1 root root 5875184 Dec 30 23:27 vmlinuz-0-rescue-c8af0a0bde254094845d7e67a759a6e7

-rwxr-xr-x. 1 root root 5875184 Jul 6 2017 vmlinuz-3.10.0-693.el7.x86_64

재 부팅후 문제 없이 적용되었는지 확인을 위해 reboot

(dracut 을 수행하는 이유는 부팅 이미지를 새로 빌드함으로써 부팅시에 LVM 설정이 올바르게 로드되는 것을 보장하도록 하기 위함이며, 다음 번 부팅시를 위해 수행해야 하는 작업입니다. =〉 즉 필수는 아님)

[root@node1 ~]# shutdown –r now (reboot)

9. HA package 설치 및 hacluster 계정 설정 (node all)

[root@node1] # yum -y install pcs pacemaker fence-agents-all

클러스터 노드 간 통신인증을 위한 비밀번호를 설정한다.
클러스터 노드간 'hacluster'라는 유저를 통해서 노드 인증이 이루어지며 각 계정별로 패스워드는 동일하게 설정한다.

[root@node1] # passwd hacluster

[root@node1] # chage –l hacluster

[root@node1] # systemctl start pcsd.service

[root@node1] # systemctl enable pcsd.service

10. Cluster 생성

PCS cluster 멤버 인증

[root@node1] # pcs host auth *node1 node2* -u hacluster

ex) pcs cluster auth GSMDDB01_hb1 GSMDDB02_hb2 -u hacluster

PCS cluster 생성

[root@node1] # pcs cluster setup --start --name *cluster_name node1 node2*

pcs cluster setup --start --name GSMDDB_clu GSMDDB01_hb1 GSMDDB02_hb2

PCS cluster 서비스 시작

[root@node1] # pcs cluster status

[root@node1] # pcs status

<Web Browser>

https://localhost:2224

ID : hacluster

PW : hacluster

11. STONITH 구성 (fencing)

[root@node1] # pcs stonith describe fence_ipmilan

[root@node1] # pcs stonith list

아래 주목할 부분은 양 노드가 delay=x 다르다는 점이다. 혹시 split brain 현상이 발생해도 동시에 fence 하지 못하도록함.

[root@node1] # pcs stonith create *IMM1* fence_ipmilan pcmk_host_list="*node1*" ipaddr="*node1_fence_ip*" login="*USERID*" passwd="*PASSW0RD*" lanplus=on auth=password **delay=5** op monitor interval=30s

ex) pcs stonith create GSMDDB01_fd1 fence_ipmilan pcmk_host_list="GSMDDB01_hb1" ipaddr="10.10.8.1" login="USERID" passwd="PASSW0RD" lanplus=on auth=password delay=5 op monitor interval=30s

[root@node1] # pcs stonith create *IMM2* fence_ipmilan pcmk_host_list="*node2*" ipaddr="*node2_fence_ip*" login="*USERID*" passwd="*PASSW0RD*" lanplus=on auth=password **delay=10** op monitor interval=30s

ex) pcs stonith create GSMDDB02_fd2 fence_ipmilan pcmk_host_list="GSMDDB02_hb2" ipaddr="10.10.8.2" login="USERID" passwd="PASSW0RD" lanplus=on auth=password delay=10 op monitor interval=30s

[root@node1] # pcs constraint location *IMM1* prefers *node2 => prefer!! 오작동 없이 상대노드만을 fence 하도록 유도함. (필수는 아님)*

ex) pcs constraint location GSMDDB01_fd1 prefers GSMDDB02_hb2

[root@node1] # pcs constraint location *IMM2* prefers *node1 => prefer!! 오작동 없이 상대노드만을 fence 하도록 유도함. (필수는 아님)*

ex) pcs constraint location GSMDDB02_fd2 prefers GSMDDB01_hb1

Stonith Failed 방지를 위한 Option 설정 (시스템 운영 중 Stonith Failed가 발생했을 때 적용한다)

pcs stonith update GSMDDB01_fd1 fence_ipmilan op monitor interval=1800s

pcs stonith update GSMDDB02_fd2 fence_ipmilan op monitor interval=1800s

pcs stonith update GSMDDB01_fd1 pcmk_monitor_timeout=240s pcmk_monitor_retries=10 power_timeout=90 op stop on-fail=ignore

pcs stonith update GSMDDB02_fd2 pcmk_monitor_timeout=240s pcmk_monitor_retries=10 power_timeout=90 op stop on-fail=ignore

12. Resource생성 (vip)

vip 리소스 등록 및 리소스 그룹에 등록

[root@node1] # pcs resource create GSMDDB_vip ocf:heartbeat:IPaddr2 ip=10.56.35.200 cidr_netmask=24 --group clustergroup op monitor timeout=200s interval=30s (interval - heartbeat 상태 체크 간격 초 - 리소스 상태 응답 없을 경우 failover)

추가 Heartbeat-IP 모니터링 정책을 등록! : 아래 a,b 제한요소를 통해 모니터링 및 그에 대한 동작을 구현

ex> team0 or bond0 =>GSMDDB01_hb --heartbeat IP

a. [root@node1] # pcs resource create *team0-monitor* ocf:heartbeat:ethmonitor interface=*team0* --clone op monitor timeout=200s

(bond0-monitor) *(bond0)*

>> 추가 모니터링 정책을 선언 하는것으로 ,내용은 :

ethmonitor라는 agent를 사용하여 모니터링 하겠다는 뜻으로 해당 인터페이스는 team0을 대상으로 하며 모니터 주기는 200초로 한다는 것입니다.
(기본 60초) 참고로 ethmonitor agent는 해당 eth의 상태 up/down등의 상태를 체크하며, 다른 agent로는 ping(이 agent는 특정구간으로 ping이 정상적인지를 체크합니다.)이 있습니다.

.

a. [root@node1] # pcs constraint location GSMDDB_vip rule score=-INFINITY ethmonitor-team0 ne 1

(bond0)

>> 이것은 위에서 선언된 정책에 어느 조건에서 trigging 되면 어떤 action을 해야 할지 기술한것입니다. :

team0에서 1이 아닌 상태가 되면 (ne 1) trigging 되며 그 action 으로는 해당 ip를 다른 노드로 넘기라는 뜻입니다.

(score=-INFINITY 는 스코어 값을 마이너스 최대치로 두어 현재 노드에서 밀어내는것을 말합니다.)

.

12. Resource생성 (vip)_continue

해당 정책이 적용되는 것은 **crm_mon -A**로 확인할수 있으며,
ex> node: pcs1-1, pcs1-2 만들어진 node의 예입니다.
Stack: corosync
Current DC: pcs1-2 (version 1.1.18-11.el7.centos.p-2b07d5c5a9) - partition with quorum
Last updated: Sun Sep 13 05:13:18 2020
Last change: Sun Sep 13 05:11:51 2020 by root via cibadmin on pcs1-1
2 nodes configured
12 resources configured
Online: [pcs1-1 pcs1-2]
Active resources:
Resource Group: TESTRG1
pcs1-1_ser1 (ocf::heartbeat:IPaddr2): Started pcs1-2
halvm (ocf::heartbeat:LVM): Started pcs1-2
test1_fs (ocf::heartbeat:Filesystem): Started pcs1-2
test3_fs (ocf::heartbeat:Filesystem): Started pcs1-2
test8_fs (ocf::heartbeat:Filesystem): Started pcs1-2
service_sc (lsb:test.sh): Started pcs1-2
lpar_fence1 (stonith:fence_lpar): Started pcs1-2
lpar_fence2 (stonith:fence_lpar): Started pcs1-2
Clone Set: eth1-mon-clone [eth1-mon]
Started: [pcs1-1 pcs1-2]
Clone Set: eth0-mon-clone [eth0-mon]
Started: [pcs1-1 pcs1-2]
Node Attributes:
*** Node pcs1-1:**
> ethmonitor-eth1 :0-> **>> 노드 pcs1-1에 eth1을 다운 시키면 1이 아닌상태가 되며, 해당 액션이 시작되게 됩니다.**

*** Node pcs1-2:**
> ethmonitor-eth1 : 1>

13. 클러스터 노드 HA-LVM 구성 및 리소스 생성 (HA-LVM & FileSystem)

[root@node1] # pcs resource create *halvm* LVM volgrpname=*datavg* exclusive=true --group clustergroup

Assumed agent name 'ocf:heartbeat:LVM' (deduced from 'LVM')

[root@node1] # pcs resource create *filesystem_name* ocf:heartbeat:Filesystem device*="/dev/mapper/datavg-lv_oracle*" directory="*/oracle*" fstype="*ext4*" --group clustergroup op monitor timeout=200s interval=30s

ex)

pcs resource create pgsmd ocf:heartbeat:Filesystem device="/dev/mapper/DBVG-lvpgsmd" directory="/pgsmd" fstype="ext4" --group clustergroup op monitor timeout=200s interval=30s

pcs resource create pgsmdwork ocf:heartbeat:Filesystem device="/dev/mapper/DBVG-lvpgswork" directory="/pgsmdwork" fstype="ext4" --group clustergroup op monitor timeout=200s interval=30s

pcs resource create pgsmdins ocf:heartbeat:Filesystem device="/dev/mapper/DBVG-lvpgsmdins" directory="/pgsmdins" fstype="ext4" --group clustergroup op monitor timeout=200s interval=30s

14. Resource생성 (Service Scripts)

Service script 파일을 /etc/init.d/ 위치에 넣고 web에서 resource 항목에서 lsb -> 스크립트 파일 존재 여부 확인

[root@node1] # pcs resource create *service_scripts* lsb:oracle.sh --group clustergroup

ex) pcs resource create orace_service lsb:oracle_start.sh --group clustergroup

적용 후 아래 작업 수행

[root@node1] # pcs resource update *service_scripts* op stop timeout=200s

[root@node1] # pcs resource update *service_scripts* op start timeout=200s

[root@node1] # pcs resource update *service_scripts* op monitor timeout=200s

15. Cluster Property 설정

pcs property set stonith-enabled=false

(펜싱 비활성화) 리소스 추가하기 위한 유지보수 단계이므로 stonith를 사용하지 않는다..

pcs property set stonith-action=reboot

stonith 장치 액션 밸류값 기입 , stonith device로 수행 명령을 내린다 value값= (reboot | off)

pcs property set stonith-timeout=120s

stonith fencing 수행이 완료되기까지 대기하는 시간

~~# pcs property set default-resource-stickiness=1000 ⇒~~ **현재 변경된 옵션 (deprecated 된 속성)=〉변경 # pcs resource defaults resource-stickiness=1000**

장애복구되면 auto fail-backed 하지 않는다. 기본값=100 ,현재 서비스되고 있는 노드 우선

오토패일백시 기존 노드로 서비스를 재 이관하려는 마이그래이션 타임(순단)이 생기므로 시간적 손해가 발생한다.

리소스 고정(고착), auto-failbacked와 동일한 개념. 기본값 100에서 크게 벗어나지 않아도 된다.

pcs property set maintenance-mode=true

특정 리소스를 작업할 경우 다른 리소스에도 다 영향을 미칠 수 있으므로,

클러스터의 리소스를 클러스터에서 관리하지 않도록 만든 후 리소스에 대한 작업이 이루어져야 한다.

pcs resource defaults migration-threshold=1 (1~ 10)

클러스터 장애발생시 리소스 이관 옵션 기입 active 노드에서 리소스 서비스가 1회 실패할 경우 상대 노드로 리소스가 이동됨.

pcs property set default-action-timeout=60s

stonith 장치(개별적인)의 시간 제한 변경

별첨. Resource 관리

클러스터에 등록된 리소스는 클러스터 데몬에 의해 제어가 이루어진다. 리소스를 추가, 수정, 삭제 시 리소스에도 영향을 미칠 수 있으므로,
클러스터의 리소스를 클러스터에서 관리하지 않도록 만든 후 리소스에 대한 작업이 이루어져야 한다.

<Property 설정 및 삭제>

[root@node1] # pcs property set maintenance-mode=true

[root@node1] # pcs property set stonith-enabled=false

<리소스 수정/제거>

pcs resource update pgsmdlog op monitor on-fail=standby timeout=200s interval=30s --group clustergroup

pcs resource delete pgsmdlog

<싱글노드 서비스>

pcs quorum unblock

[root@node1] # pcs property set maintenance-mode=false (작업시 : true)

[root@node1] # pcs property set stonith-enabled=true (작업시 : false)

<Cluster 삭제 (node all)>

[root@node1] # pcs cluster destroy

[root@node2] # pcs cluster destroy

별첨. 클러스터 take-over 테스트

[root@node1] # pcs resource move clustergroup *node_name* <- 상대편 노드

ex) pcs resource move clustergroup GSMDDB01_hb02

[root@node1] # pcs status

[root@node1 ~]# pcs constraint

Location Constraints:

  Resource: NODE_FD01

    Enabled on: NODE_HB01 (score:INFINITY)

  Resource: NODE_FD02

    Enabled on: NODE_HB02 (score:INFINITY)

  Resource: clustergroup

    Enabled on: TEST_HB01 (score:INFINITY) (role: Started)

Ordering Constraints:

Colocation Constraints:

Ticket Constraints:

클러스터 서비스 중 강제로 리소스를 move시키면 위와 같이 Move된 노드에 INFINITY 값이 생긴다, ban발생

아래 명령어로 꼭 Clear를 해준다

[root@node1] # pcs resource clear clustergroup

별첨. Pacemaker 성능관리

1. FileSystem

신규 리소스 파일시스템에 반영시...
[root@node1] # pcs resource create oradata .... options=noatime,errors=remount-ro
[root@node1] # cat /proc/mounts

기존 생성된 리소스 파일시스템에 반영시...
[root@node1] # pcs resource update oradata .... options=noatime,errors=remount-ro
[root@node1] # cat /proc/mounts