

Pacemaker on Azure CentOS

▼ 상태	3rd party
🕒 작성일시	@2022년 8월 31일 오후 3:50

OS : Cent OS 7 (ha 나 hpc 이미지를 권장하지만...)
On Azure

참고링크

<https://honglab.tistory.com/127>

<https://docs.microsoft.com/ko-kr/azure/virtual-machines/workloads/sap/high-availability-guide-rhel-pacemaker>

<https://docs.microsoft.com/ko-kr/azure/virtual-machines/workloads/sap/sap-hana-high-availability-rhel>

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deploying_red_hat_enterprise_linux_8_on_public_cloud_platforms/configuring_rhel-high-availability-on-azure_cloud-content

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_administration/ch-startup-haaa

<http://arajucse.blogspot.com/2016/08/apache-web-server-clustering-using.html>

Cent OS 서버 2개 준비

별도의 언급이 없다면 첫번째 서버에서만 실행하면 된다.



명령어는 root 기준



노드간 방화벽 등은 열려 있는것을 기준으로 한다
만약 필요하다면 하기 표시한 정보 참고

TCP : 2224, 3121, 5403, 21064, 9929
UDP : 5404, 5405, 9929

pacemaker 와 fence 에이전트 설치
관련 글들에 pacemaker와 corosync 등을 설치하라고 나오지만

7.9버전 기준으로 pcs 패키지만 설치하여도 3개 모두 설치된다.
resource-agents 도 있으면 넘어가고 없으면 설치한다.

```
yum install -y pcs fence-agents-azure-arm nmap-ncat
```

- Azure 위의 Redhat 관련 이미지는 fence-agent 로의 fencing 만 지원함.
- 두번째 서버도 설치한다.

/etc/hosts 파일 수정

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.6     vm-pcs
10.0.0.7     vm-pcs2
10.0.0.250   cluster
```

- 두번째 서버도 동일하게 수정한다

위의 pcs를 설치했다면 hacluster 유저가 생성되어있는 것을 볼 수 있다.
비밀번호를 변경한다.

```
passwd hacluster
```

- 기본 유저사용이 불편하다면 다른 유저를 생성해도 된다.
- 두번째 서버도 동일하게 변경한다

Azure Docs 에는 하기와 같은 내용이 있는데 안해도 된다.
만약 다른 사이트에 centos도 등록해서 쓰라는게 있으면 그거도 무시해도 된다.

1. [A] 등록 - 선택적 단계입니다. RHEL SAP HA-지원 이미지를 사용 중이라면 필요 없는 단계입니다.

가상 머신을 등록하고 RHEL 7의 리포지토리가 포함된 풀에 연결합니다.

복사

```
sudo subscription-manager register
# List the available pools
sudo subscription-manager list --available --matches '*SAP*'
sudo subscription-manager attach --pool=<pool id>
```

풀을 Azure Marketplace PAYG RHEL 이미지에 연결하면 RHEL 사용량에 대하여 사실상 이중 청구 (PAYG 이미지에 대하여 한 번, 연결한 풀의 RHEL 자격에 대하여 한 번)가 됩니다. 이제 Azure는 이와 같은 이중 청구를 완화하기 위해 BYOS RHEL 이미지를 제공합니다. 자세한 내용은 [Azure의 Red Hat Enterprise Linux Bring-Your-Own-Subscription](#) 이미지를 참조하세요.

2. [A] SAP 리포지토리에 RHEL을 사용하도록 설정 - 선택적 단계입니다. RHEL SAP HA-지원 이미지를 사용 중이라면 필요 없는 단계입니다.

RHEL 7 필수 패키지를 설치하려면 다음 리포지토리를 사용하도록 설정합니다.

복사

```
sudo subscription-manager repos --disable "*"
sudo subscription-manager repos --enable=rhel-7-server-rpms
sudo subscription-manager repos --enable=rhel-ha-for-rhel-7-server-rpms
sudo subscription-manager repos --enable=rhel-sap-for-rhel-7-server-rpms
sudo subscription-manager repos --enable=rhel-ha-for-rhel-7-server-eus-rpms
```

pcs 서비스 시작 및 자동시작 등록

```
systemctl start pcsd.service
systemctl enable pcsd.service
```

- 두번째 서버도 동일하게 설정한다.

각 호스트에 hacluster 로 접속확인

```
pcs host auth prod-cl1-0 prod-cl1-1 -u hacluster
```

cluster 생성 및 시작

```
pcs cluster setup --name clustername node1 node2 --token 30000
pcs cluster start --all
pcs cluster enable --all
```

- 이름들은 알맞잡하자

cluster 상태확인

```
pcs status
```

잠시 기다리면 아래와 같이 표시되는데, pcsd 만 enabled 면 괜찮다.

```
Cluster name: hant

WARNINGS:
No stonith devices and stonith-enabled is not false

Stack: corosync
Current DC: vm-pcs (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Wed Aug 31 06:24:00 2022
Last change: Wed Aug 31 06:22:41 2022 by hacluster via crmd on vm-pcs

2 nodes configured
0 resource instances configured

Online: [ vm-pcs vm-pcs2 ]

No resources

Daemon Status:
corosync: active/disabled
pacemaker: active/disabled
pcsd: active/enabled
```

쿼럼 투표 설정

```
# Check the quorum votes
pcs quorum status
# If the quorum votes are not set to 2, execute the next command
pcs quorum expected-votes 2
```

- 노드가 세 개 이상인 다중 노드 클러스터를 빌드하는 경우, 투표를 2로 설정하지 마세요.

Fence 동시 작업 허용

```
pcs property set concurrent-fencing=true
```

STONITH 디바이스 만들기

Fence Agent 용 Azure SPN을 생성하고 필요한 권한(VM start / poweroff / read)을 지닌 역

할을 구독 혹은 리소스 그룹에 부여한다. 자세한 과정은 생략한다.

아래에는 필요한 정보를 나열한다.

애플리케이션 ID

디렉터리 ID

애플리케이션 비밀

VM의 리소스그룹

구독 ID

VM 이름 (이부분은 이후 다시 설명한다.)

```
pcs property set stonith-timeout=900
```

- 두번째 서버에도 적용한다.

아래는 기본형태

```
sudo pcs stonith create rsc_st_azure fence_azure_arm login="App ID(Client ID)" passwd="password" resourceGroup="resource group" tenantId="tenant ID" subscriptionId="subscription id" pcmk_host_map="prod-cl1-0:prod-cl1-0-vm-name;prod-cl1-1:prod-cl1-1-vm-name" \
power_timeout=240 pcmk_reboot_timeout=900 pcmk_monitor_timeout=120 pcmk_monitor_retries=4 pcmk_action_limit=3 pcmk_delay_max=15 \
op monitor interval=3600
```

실제 입력의 예

```
sudo pcs stonith create rsc_st_azure fence_azure_arm login="f8d68314-894f-4c7a-b200-7c13cc067cf5" passwd="o.w8Q~IILBfNzrY2ykgWLBa32vyjY2lTb_zfbbgh" resourceGroup="vm-pcs_group" tenantId="9ab80bfb-f6a5-4b78-9a6f-dd037dc0e745" subscriptionId="9578bc4a-f4f0-4810-b715-78bbcd7e36ec" pcmk_host_map="vm-pcs:vm-pcs;vm-pcs2:vm-pcs2" \
power_timeout=240 pcmk_reboot_timeout=900 pcmk_monitor_timeout=120 pcmk_monitor_retries=4 pcmk_action_limit=3 pcmk_delay_max=15 \
op monitor interval=3600
```

- 만약 Azure VM 리소스의 이름과 VM OS 내에서의 hostname 이 다르다면 pcmk_host_map 옵션을 통해 지정해야한다.
- 형식은 OShostname:AzureVMname 이다.
- 즉, 동일하다면 빼도 되는 옵션.

STONITH 디바이스를 사용하도록 설정

```
pcs property set stonith-enabled=true
```

- 두번째 서버에도 적용한다.

fence 테스트용 명령어

```
pcs stonith fence vm-pcs2
```

VIP를 만든다

필요하다면 --group apachegroup 등을 활용하여 리소스 그룹으로 묶을 수 있다.
아이피는 처음 입력했던 vip

```
pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=10.0.0.250 \  
cidr_netmask=24
```

서비스 리소스가 있다면 묶을 수 있다.

INFINITY 는 리소스를 강제로 옮기기 위한 옵션

```
pcs constraint colocation add 리소스이름 VirtualIP INFINITY
```

Stonith 리소스가 있어서 묶어봤다 (기존에는 서로 다른 노드에서 잡고 있었다.)

```
pcs constraint colocation add rsc_st_azure VirtualIP INFINITY
```

```

Online: [ vm-pcs vm-pcs2 ]

Full list of resources:

rsc_st_azure    (stonith:fence_azure_arm):      Started vm-pcs
VirtualIP       (ocf::heartbeat:IPaddr2):         Started vm-pcs2

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@vm-pcs ~]# pcs constraint colocation add rsc_st_azure VirtualIP 1
[root@vm-pcs ~]#
[root@vm-pcs ~]#
[root@vm-pcs ~]# pcs status
Cluster name: hant
Stack: corosync
Current DC: vm-pcs (version 1.1.23-1.el7_9.1-9acf116022) - partition with
Last updated: Thu Sep  1 05:12:45 2022
Last change: Thu Sep  1 05:12:43 2022 by root via cibadmin on vm-pcs

2 nodes configured
2 resource instances configured

Online: [ vm-pcs vm-pcs2 ]

Full list of resources:

rsc_st_azure    (stonith:fence_azure_arm):      Starting vm-pcs2
VirtualIP       (ocf::heartbeat:IPaddr2):         Started vm-pcs2

Daemon Status:

```

- 옮겨진 것을 확인할 수 있다.

위의 create 명령 중 ocf:heartbeat:IPaddr2 부분은 각각
스탠다드 : 프로바이더 : 스크립트
순으로 보면된다.

아래의 명령어로 사용가능한 목록을 볼 수 있다.

IPaddr2 의 경우는 ocf:heartbeat 에서 사용 가능한 VIP를 만드는 스크립트라고 보면된다.

```

pcs resource standards
pcs resource providers
pcs resource agents ocf:heartbeat

```



기본적인 세팅은 끝입니다
이후에는 서비스별 연동 방법이 다릅니다.

다음 글 부터는 필요한 시나리오별 모듈처럼 진행됩니다.

공유디스크 + LVM

Azure 공유 디스크 생성 및 attach 방법은 생략합니다.

사용 가능한 디스크 확인 및 예제 출력
테스트용으로 디스크를 1개만 붙였다.

```
ls /dev/disk/azure/scsi1/lun*  
  
/dev/disk/azure/scsi1/lun0
```

피지컬 볼륨 생성

```
pvcreate /dev/disk/azure/scsi1/lun0
```

볼륨 그룹 생성
띄어쓰기로 구분하여 더 넣으면 더 들어감

```
vgcreate vg_pcs_disk /dev/disk/azure/scsi1/lun0
```

로지컬 볼륨 생성 (디스크는 4GB다 '-L 4GB' 옵션을 사용해도 된다.)

```
lvcreate -l 100%FREE -n lv_pcs_disk vg_pcs_disk
```

논리적 볼륨 초기화
형식은 /dev/VG이름/LV이름

```
mkfs.ext4 /dev/vg_pcs_disk/lv_pcs_disk
```



아래의 Mount 의 경우는 시나리오 상 필요할 수도 필요 없을 수도 있습니다.
안하는것을 전제로 진행합니다.

LV Mount

이후 편의를 위해 fstab 파일에 넣어도 된다.

/dev/vg_pcs_disk/lv_pcs_disk /data ext4 defaults 0 0

```
mkdir /data
mount /dev/vg_pcs_disk/lv_pcs_disk /data
```

다른 저장소는 이후 추가

Apache with Azure LB



하기 내용 부터는 별도로 표시 하지 않았을 경우, 개별 노드에 각각 실행한다.

패키지는 httpd

/etc/httpd/conf/httpd.conf 에 하기 라인들을 추가한다.

```
<Location /server-status>
    SetHandler server-status
    Require local
</Location>
```

아래와 같이하라는 것이 최신인데 확인해보자

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```

로그 관련 설정. 갑자기 이부분이 사라져서 일단 생략한다.

/etc/logrotate.d/httpd

```
/bin/systemctl reload httpd.service > /dev/null 2>/dev/null || true
```

이 부분을

```
/usr/bin/test -f /run/httpd.pid >/dev/null 2>/dev/null &&
/usr/bin/ps -q $(/usr/bin/cat /run/httpd.pid) >/dev/null 2>/dev/null &&
/usr/sbin/httpd -f /etc/httpd/conf/httpd.conf -c "PidFile /run/httpd.pid" -k graceful > /dev/null 2>/dev/null || true
```

요것으로 교체한다.(3줄입니다.)

웹서버 기본 설정

```
mount /dev/vg_pcs_disk/lv_pcs_disk /var/www/
mkdir /var/www/html
mkdir /var/www/cgi-bin
mkdir /var/www/error
restorecon -R /var/www
cat <<-END >/var/www/html/index.html
<html>
<body>Hello</body>
</html>
END
umount /var/www
```

- 한쪽에서만 해도 된다.

LVM config 수정

```
lvmconf --enable-halvm --services --startstopservices
```

vg 리스트 확인 (vg_name 변경없이 입력)

```
vgs --noheadings -o vg_name
```

/etc/lvm/lvm.conf 에서 클러스터용 vg 를 제외한 다른 vg 리스트를 적는다.

```
volume_list = []
```

- 약 1250 번 줄 근처에 있다.

파티션? 디스크? 를 부트이미지가 아닌 클러스터가 잡도록 설정
약 1분 소요 > 이후 재부팅

```
dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

- 만약 OS 업데이트나 커스텀 OS 업데이트의 경우 다시 적용해야할 가능성 있음.

로드 밸런서 생성 [표준 / 내부 / 서비스할 포트(80) / 나는 HA로함]
상태프로브 TCP 61000 포트

(과정 생략)

보통 외부의 노출되는 아파치서버는 클러스터링 보단 서버를 2개 두고 쓰는것이 낫지만 이번에는 톰켓이나 sap 등의 다른 서비스를 위한 용도라고 보고 내부LB 로 사용한다.

로드밸런서 PCS 상태 체크용 리소스 생성

```
pcs resource create az-apache-lb azure-lb port=61000 --group az-apache-cluster
```

가시성이나 관리를 위해 필요하다면 기존리소스를 리소스 그룹에 넣는다.

```
pcs resource group add az-apache-cluster VirtualIP
```

LVM 및 파일시스템 리소스 생성

```
pcs resource create lv_pcs_disk LVM volgrpname=vg_pcs_disk \
exclusive=true --group az-apache-cluster
```

```
pcs resource create apache_fs Filesystem \
device="/dev/vg_pcs_disk/lv_pcs_disk" directory="/var/www" fstype="ext4" --group \
az-apache-cluster
```

서비스 생성

```
pcs resource create Website apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://127.0.0.1/server-status" --group az-apache-cluster
```

이렇게 생성된 서비스는 systemctl로는 확인이 안된다.

```
[root@vm-pcs ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
```

apache 의 경우는 아래와 같이 정리되었다.

```

[root@vm-pcs ~]# pcs status
Cluster name: hant
Stack: corosync
Current DC: vm-pcs (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Mon Sep  5 08:30:57 2022
Last change: Mon Sep  5 08:15:00 2022 by root via cibadmin on vm-pcs2

2 nodes configured
6 resource instances configured

Online: [ vm-pcs vm-pcs2 ]

Full list of resources:

Resource Group: az-apache-cluster
  VirtualIP (ocf::heartbeat:IPaddr2):      Started vm-pcs
  az-apache-lb (ocf::heartbeat:azure-lb):    Started vm-pcs
  lv_pcs_disk (ocf::heartbeat:LVM):         Started vm-pcs
  apache_fs (ocf::heartbeat:Filesystem):     Started vm-pcs
  Website (ocf::heartbeat:apache):          Started vm-pcs
Resource Group: management
  rsc_st_azure (stonith:fence_azure_arm):    Started vm-pcs

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

```

VirtualIP
 az-apache-lb
 lv_pcs_disk
 apache_fs
 Website

- pcs constraint colocation add 명령어를 이용하여 필요한 리소스들을 묶어줘도 된다.

내부 IP를 쓴 탓에 내부에 VM을 만든 후 접근테스트 하면된다.