OCP 설치-ansible(X)

참고 = 오픈마루 - 제품(PRODUCT) : 207. OCP 4.10.9 설치방법 (ansible x)

구성 환경

IP대역 : 192.168.200.0~255

dns : 192.168.23.2 ⇒ 192.168.200.1

gateway : 1922.168.0.1

구분	hostname	ip	os	gateway	dns	사양
bastion	bastion.ocp.beomzh.com	192.168.200.1	RHEL8.10	192.168.0.1	192.168.23.2	4Core / 16G
bootstrap	bootstrap.ocp.beomzh.com	192.168.200.10	rhcos-4.10	192.168.0.1	192.168.200.1	4/16
master-1	master-1.ocp.beomzh.com	192.168.200.11	rhcos-4.10	192.168.0.1	192.168.200.1	4/16
master-2	master-2.ocp.beomzh.com	192.168.200.12	rhcos-4.10	192.168.0.1	192.168.200.1	4/16
master-3	master-3.ocp.beomzh.com	192.168.200.12	rhcos-4.10	192.168.0.1	192.168.200.1	4/16
worker-1	worker-1.ocp.beomzh.com	192.168.200.21	rhcos-4.10	192.168.0.1	192.168.200.1	4/16
worker-2	worker-2.ocp.beomzh.com	192.168.200.22	rhcos-4.10	192.168.0.1	192.168.200.1	4/16

Bastion 구성

- Bastion OCP Client 및 운영에 필요로 하는 유틸리티 서버 역활

1) Subscription 등록 (local.repo 를 등록시에 안해도 무관)

- subscription-manager란 Red Hat의 서비스 및 리포지토리에 접근할 수 있도록 구독(subscription)을 관리하는 역할

```
[root@localhost /]# subscription-manager register --user beomzh@opennaru.com --password #####
```

- ※ error 발생

```
#login 정보가 맞지 않을시

등록 대상: [subscription.rhsm.redhat.com:443/subscription](http://subscription.rhsm.redhat.com
```

```
Invalid username or password. To create a login, please visit https://www.redhat.com/wapps/ug

# 해당에러는 문의 -> local.repo 등록시 해당 절차 pass

등록 대상: subscription.rhsm.redhat.com:443/subscription
시스템은 ID로 등록되어 있습니다: e910c12a-813e-4a82-b853-dc5b49cd964e
등록된 시스템 이름: localhost.localdomain
자동 첨부 요청을 무시합니다. 콘텐츠 액세스 모드 설정으로 인해 "7198929" 조직에 대해 비활성화되어 있습니다.
```

1) local.repo 등록

```
# repo 등록
# vi /etc/yum.repos.d/local.repo
[local_BaseOS]
name=local_BaseOS
baseurl=file:///mnt/BaseOS/
enabled=1
gpgcheck=0
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[local_AppStream]
name=local_AppStreammo
baseurl=file:///mnt/AppStream
enabled=1
gpgcheck=0
#gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# 마운트
# disk 빠지면 마운트 적용안되니 확인필요
mount /dev/sr0 /mnt
```

2) SELinux 및 방화벽 종료

- SELinux 란 시스템의 애플리케이션, 프로세스, 파일에 대한 액세스 제어를 정의

```
# 방화벽 종료
systemctl  disable --now firewalld
setenforce 0
getenforce

# SELinux 종료
vi /etc/sysconfig/selinux
# 7번 line 변경
SELINUX=enforcing -> disabled

reboot
```

	의미	설정 명령어
enforce	기본값. 보안 정책에 맞지 않는 요청은 거부된다.	setenforce 1
permissive	경고만 한다.	setenforce 0
disable	SELinux를 영구적으로 비활성화한다.	grubby --update-kernel ALL --args selinux=0

3) 패키지 설치

```
yum update -y
yum install -y unzip bash-completion httpd bind bind-utils haproxy nfs-utils jq podman
mkdir -pv /opt/registry/{auth,certs,data}
```

4) HTTP 구축

```
[root@localhost ~]# vi /etc/httpd/conf/httpd.conf
# line 45
Listen 80 -> 8080
# ---

mkdir -pv /var/www/html/ocp
chmod -R 755 /var/www/html/ocp/
systemctl enable --now httpd
```

5) NFS 구축

```
# network file system 설치
systemctl enable --now nfs-server
mkdir -pv /var/nfsshare
chown -R nobody:nobody /var/nfsshare/  #RHEL7 경우 nfsnobody:nfsnobody
echo '/var/nfsshare 192.168.200.1/16(rw,sync,root_squash)' >> /etc/exports
exportfs -r
systemctl restart nfs-server

# 정상 확인
systemctl status nfs-server
```

6-1) DNS 설정

```
vi /etc/named.conf

# line 11,19 내용 변경
127.0.0.1 -> any
localhost -> any
```

```
10 options {
11         listen-on port 53 { any; };
12         listen-on-v6 port 53 { ::1; };
13         directory       "/var/named";
14         dump-file       "/var/named/data/cache_dump.db";
15         statistics-file "/var/named/data/named_stats.txt";
16         memstatistics-file "/var/named/data/named_mem_stats.txt";
17         secroots-file   "/var/named/data/named.secroots";
18         recursing-file  "/var/named/data/named.recursing";
19         allow-query     { any; };
```

6-2) zone 추가

```
vi /etc/named.rfc1912.zones
zone "ocp.beomzh.com" IN {
    type master;
    file "ocp.beomzh.com.zone";
    allow-update { none; };
```

```
};

zone "200.168.192.in-addr.arpa" IN {
    type master;
    file "ocp.beomzh.com.rev";
    allow-update { none; };
};
# 192.168.200.7 ip대역 사용중이므로 200.168.192.in-addr.arpa 설정
# reverse 쪽 IP는 c/b/a 클래스 순으로 입력해야한다.
```

6-3) zone 생성

```
vi /var/named/ocp.beomzh.com.zone
$TTL 1D
@       IN SOA  @ bastion.ocp.beomzh.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum

; name servers - NS records
        NS     bastion.ocp.beomzh.com.

; OpenShift Container Platform Cluster - A records
master-1        IN      A       192.168.200.11
master-2        IN      A       192.168.200.12
master-3        IN      A       192.168.200.13
worker-1        IN      A       192.168.200.21
worker-2        IN      A       192.168.200.22
infra-1         IN      A       192.168.200.31
bootstrap       IN      A       192.168.200.10
bastion         IN      A       192.168.200.1

; OpenShift internal cluster IPs - A records
api             IN      A    192.168.200.1
api-int         IN      A    192.168.200.1
*.apps          IN      A    192.168.200.1
```

6-4) rev 생성

```
vi /var/named/ocp.beomzh.com.rev
$TTL 1D
@       IN SOA  @ bastion.ocp.beomzh.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
; name servers - NS records
        NS     bastion.ocp.beomzh.com.

; OpenShift Container Platform Cluster - PTR records
10     IN      PTR   bootstrap.ocp.beomzh.com.
11     IN      PTR   master-1.ocp.beomzh.com.
```

```
12      IN      PTR   master-2.ocp.beomzh.com.
13      IN      PTR   master-3.ocp.beomzh.com.
21      IN      PTR   worker-1.ocp.beomzh.com.
22      IN      PTR   worker-2.ocp.beomzh.com.
31      IN      PTR   infra-1.ocp.beomzh.com.
1       IN      PTR  api.ocp.beomzh.com.
1       IN      PTR  api-int.ocp.beomzh.com.
```

6-5) zone 체크

```
chmod 644 /var/named/ocp.beomzh.com.*
named-checkconf /etc/named.conf
named-checkconf /etc/named.rfc1912.zones
named-checkzone ocp.beomzh.com /var/named/ocp.beomzh.com.zone

systemctl enable --now named
nmcli # dns server에 현재 작업중인 ip 확인

[root@localhost named]# nmcli connection show --active
NAME    UUID                                  TYPE      DEVICE
ens192  7eb70e56-caf4-49b0-99f0-271c7efb11e2  ethernet  ens192
# name 값을 가져온다.

nmcli connection modify ens192 ipv4.dns "192.168.200.1,192.168.23.2"
nmcli con up ens192
```

7) HAproxy(LB) 구축

```
# haproxy 파일 백업생성
cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg_bak
vi /etc/haproxy/haproxy.cfg

global
    maxconn     20000
    log         /dev/log local0 info
    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option forwardfor       except 127.0.0.0/8
```

```
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          300s
    timeout server          300s
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 20000

frontend openshift-api-server
    bind *:6443
    default_backend openshift-api-server
    mode tcp
    option tcplog

backend openshift-api-server
    balance source
    mode tcp
    server bootstrap 192.168.200.10:6443 check fall 3 rise 2
    server master-1 192.168.200.11:6443 check fall 3 rise 2
    server master-2 192.168.200.12:6443 check fall 3 rise 2
    server master-3 192.168.200.13:6443 check fall 3 rise 2


frontend machine-config-server
    bind *:22623
    default_backend machine-config-server
    mode tcp
    option tcplog

backend machine-config-server
    balance source
    mode tcp
    server bootstrap 192.168.200.10:22623 check fall 3 rise 2
    server master-1 192.168.200.11:22623 check fall 3 rise 2
    server master-2 192.168.200.12:22623 check fall 3 rise 2
    server master-3 192.168.200.13:22623 check fall 3 rise 2

frontend ingress-http
    bind *:80
    default_backend ingress-http
    mode tcp
    option tcplog

backend ingress-http
    balance source
    mode tcp
    server worker-1 192.168.200.21:80 check fall 3 rise 2
    server worker-2 192.168.200.22:80 check fall 3 rise 2
    server infra-1 192.168.200.31:80 check fall 3 rise 2


frontend ingress-https
    bind *:443
    default_backend ingress-https
    mode tcp
```

```
    option tcplog

backend ingress-https
    balance source
    mode tcp
    server worker-1 192.168.200.21:443 check fall 3 rise 2
    server worker-2 192.168.200.22:443 check fall 3 rise 2
    server infra-1 192.168.200.31:443 check fall 3 rise 2
```

```
# haproxy 실행
sudo systemctl enable --now haproxy

# validation
sudo systemctl status haproxy.service
```

Image Mirroring

1) 파일 준비

https://access.redhat.com/downloads/content/290/ver=4.10/rhel---8/4.10.9/x86_64/product-software

OpenShift v4.10 Linux Client

OpenShift v4.10 Linux Installer

```
# scp 또는 sftp 사용 후 압축풀기

chmod -R 755 /var/www/html/ocp/
tar zxvf openshift-install-linux-4.10.9.tar.gz
tar zxvf oc-4.10.9-linux.tar.gz
cp oc kubectl openshift-install /usr/local/bin/
cp oc kubectl openshift-install /usr/local/sbin/

# cp도 무관
# cp oc kubectl openshift-install /usr/local/bin/

# validation 원하는 걸로
openshift-install version
oc version
/usr/local/bin/openshift-install version
```

2) 인증서 적용

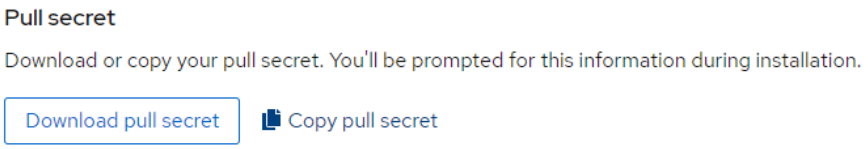
```
cd /opt/registry/certs
# RHEL7의 경우 RHEL8 서버에서 crt파일 생성 후 copy
openssl req -addext "subjectAltName=DNS:bastion.ocp.beomzh.com" -subj "/C=KO/ST=Seoul/L=Seoul
cp domain.crt /etc/pki/ca-trust/source/anchors/.
update-ca-trust
```

3) Pull Secret 파일 생성

https://console.redhat.com/openshift/install/metal/user-provisioned

https://cloud.redhat.com/openshift/install/metal/user-provisioned

경로 이동 후 Pull Secret 복사




```
mkdir /root/ocp && cd $_

# 내용 paste
vi /root/ocp/pull-secret

# harbor 계정 정보를 환경변수에 저장
REG_SECRET_INT=`echo -n 'admin:beomzh' | base64 -w0`
# harbor 정보를 추가해 pull-secret.json 파일 생성
cat /root/ocp/pull-secret | jq '.auths += {"bastion.ocp.beomzh.com:5000": {"auth": "REG_SECRE
```

4) Image Mirroring 환경변수 설정

```
vi /root/ocp/ocp_env
export OCP_RELEASE=4.16.10  #oc 및 openshift-install 버전과 동일
export LOCAL_REGISTRY='bastion.ocp.beomzh.com:5000'
export LOCAL_REPOSITORY='ocp/ocp4'
export PRODUCT_REPO='openshift-release-dev'
export LOCAL_SECRET_JSON='/root/ocp/pull-secret.json'
export RELEASE_NAME='ocp-release'
export ARCHITECTURE=x86_64

# 적용
source /root/ocp/ocp_env
```

5) Image Mirrroring 실행

```
cd /opt/registry/auth/
htpasswd -cBb ./htpasswd admin beomzh

podman run --name mirror-registry -p 5000:5000 -v /opt/registry/data:/var/lib/registry:z -v /

curl -k -u admin:beomzh https://bastion.ocp.beomzh.com:5000/v2/_catalog
podman login -u admin -p beomzh bastion.ocp.beomzh.com:5000
# Login succeeded

# 접속 확인 후
export GODEBUG=x509ignoreCN=0;
oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:
```

- error 내용 (인증서 관련) - 인증서 적용을 제대로 하지 않으면 발생`-

```
# 위의 curl 명령어 사용시 해당 에러 발생
curl: (60) SSL certificate problem: self signed certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
```

```
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

- 인증서 관련 legacy 에러 발생 시 다음 명령어로 인증서 생성

```
oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:
```

- 환경변수 누락시 에러 발생

```
oc adm release mirror 명령어 입력시 아래 에러발생
error: must specify a release image with --from
env에 환경변수가 제대로 설정되었는지 다시 체크
```

```
# result
Success
Update image:  bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9:4.10.9-x86_64
Mirror prefix: bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
Mirror prefix: bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9:4.10.9-x86_64

To use the new mirrored repository to install, add the following section to the install-confi

imageContentSources:
- mirrors:
  - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev


To use the new mirrored repository for upgrades, use the following to create an ImageContentS

apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: example
spec:
  repositoryDigestMirrors:
  - mirrors:
    - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
    source: quay.io/openshift-release-dev/ocp-release
  - mirrors:
    - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
    source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

5) mirror-registry(podman) systemd 등록

```
# 재기동 시 container 자동 실행 파일 생성
podman generate systemd --restart-policy=always -f -t 1 --name mirror-registry
mv container-mirror-registry.service /etc/systemd/system
```

Ignition File 생성 및 Install-config.yaml 작성

1) SSH 키 생성

```
# /root/.ssh/id_rsa.pub. 경로에 ssh 키 생성
ssh-keygen -t rsa -b 4096 -N ''
eval "$(ssh-agent -s)"
# Agent pid 73524
ssh-add /root/.ssh/id_rsa
# Identity added: /root/.ssh/id_rsa (root@bastion.ocp.beomzh.com)
```

2) install-config.yaml 작성

```
# vi /root/ocp/install-config.yaml
apiVersion: v1
baseDomain: beomzh.com
metadata:
  name: ocp

compute:
- hyperthreading: Enabled
  name: worker
  replicas: 2

controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3

networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16

platform:
  none: {}

fips: false

# pull-secret.json에 있는 private image registry 정보 넣기
pullSecret: '{"auths":{"bastion.ocp.beomzh.com:5000":{"auth":"YWRtaW46b3Blbm5hcnU=","email":'

# ssh키 넣기. cat /root/.ssh/id_rsa.pub
sshKey: 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDS3DXFpnKmt9AblfAqoWQPWxvqNgUo9GXJLn1rj/wVGBSA

# private image registry 인증서
# cat /opt/registry/certs/domain.crt
# x.509 인증서값 앞에 2칸 들여쓰기
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  MIIGNDCCBBygAwIBAgIUCU9Gu/LvCI6Kh33XX3yU3AFPLVUwDQYJKoZIhvcNAQEL
  BQAwgZcxCzAJBgNVBAYTAktPMQ4wDAYDVQQIDAVTZW91bDEOMAwGA1UEBwwFU2Vv
  dWwxETAPBgNVBAoMCE9wZW5uYXJ1MRAwDgYDVQQLDAdzdXBwb3J0MR8wHQYDVQQD
  DBZiYXN0aW9uLm9jcC5iZW9temguY29tMSIwIAYJKoZIhvcNAQkBFhNiZW9temhA
  b3Blbm5hcnUuY29tMB4XDTI0MDcyMjA4NTg1M1oXDTI1MDcyMjA4NTg1M1owgZcx
  CzAJBgNVBAYTAktPMQ4wDAYDVQQIDAVTZW91bDEOMAwGA1UEBwwFU2VvdWwxETAP
  BgNVBAoMCE9wZW5uYXJ1MRAwDgYDVQQLDAdzdXBwb3J0MR8wHQYDVQQDDBZiYXN0
  aW9uLm9jcC5iZW9temguY29tMSIwIAYJKoZIhvcNAQkBFhNiZW9temhAb3Blbm5h
```

```
  cnUuY29tMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAp6A35vLlJfpZ
  ZmRzTMnYm9qRV6326u9MnzOeq1YolkcALHPWDmnGl/wGFN53XzGx8X14zkNuidv2
  sRv+PSrvWlUHQW/0xOKoYthQ7yHqR9DNtYukjBRt1UjFZkcx1d9y0PCIHAQ8xl6q
  zGkwLWR1jsiATK49DYPV/2GF0Jv0HYkzBXTLntTAonE85crB0wfpkRyf/VCdUd1X
  zlnTjK/HTnxX5mPj99bzhCfaO89MwFp5zSI71eI9DLGsLaIJmBXe4PUNWZ7vqgIq
  JKSKS31EiZLkzDrnSHSoe0lQ/DwBpZUczOrvZFe+LrW/QVKz42wkDwYqEYW1t266
  4fJEQXbEXnK6bazNa9eqpLA2rFXOxmyV59uGQTQ3Ar9/77Mauq7J2RYwLmgY1OrT
  PNmQ0AppJBhRTbd61QwDLWM94dlgiYJUBbZtyUXGUc68NK6XJQbcpcFEXI/KFNpR
  rPts48TSPT4wxmvkkAwFKZv2ap8OGbJLCWLkRaI9kCA7DvwJ1MiAQsPovtGQgtah
  9y6BsbcLhUVj54eKMOuGZua556Hym23Pk0UU4HSB5FEyo58THKXPhwML9qfEr+10
  b8VBg6TxshaNZiLTVRCTHpn8Tvotg2cDF8fR+f3LWTeZKMwS2w+sGKL8mLtmvR48
  DC5GZ/SI3J8DBQfImvkKoJTeaKjlUL0CAwEAAaN2MHQwHQYDVR0OBBYEFFJfW0Wi
  RP4pju4rpdOvLyPw+3RxMB8GA1UdIwQYMBaAFFJfW0WiRP4pju4rpdOvLyPw+3Rx
  MA8GA1UdEwEB/wQFMAMBAf8wIQYDVR0RBBowGIIWYmFzdGlvbi5vY3AuYmVvbXpo
  LmNvbTANBgkqhkiG9w0BAQsFAAOCAgEAfg7Ryhol7KOk0Zb7XQ6NqTAp03aOPZrc
  zNnsmWyON+evMyVFW7yeyPYUnoPyYOmfdLTkqyBhKbgfwsN4JOtEKNbO98SAWihd
  g5X7FE/AoDOSe1TWemykbF1lNJp9hMJc8PDlMqUTzvxBke1zdAobg8T0er9cFdN5
  NG9YES1USPLtUDPuOvj/jtS78F+0Lo7iWVbg+kHI/1PWPDa0fJZKEMAFx4rE/ghz
  vWezE1nj1fC+fiBW6ZKkWB5Y3aOO61WuD3d6pvIbAlr6OOHwSBkNyWGlej0Uw7Q0
  Mo25lEC81zQJDpD0efW/FJKippbRb9fj6zKqy6LoiGIRjw/zHW/9ikt/Crwfxe8/
  42Mo1gJBAI4SVWcdK9LgL3ZxcuGRUjyE0fTqHn396TLrLlFo+qCzIX47e4anv7es
  a0BE59MkKmFabRkQh8DnALGalGDuZ2YSdA4T6+8QnY6n5qSIJWCMSHh/QTUJJGLh
  AlBRNxyMJ0rtoCqZYMtQ/EDcwTJ/pe/hkrPEeyj9/kaXbTAHIclJJIGBKTRlBvh4
  XnC3CrCq9FOd0NFTfy8doust7CwIVvbJ2tuT9uK+qkKsCNZO9tyyt34ZZHIoueE0
  1zbvvAss+E8RYzKPDcYnMuXRGRdUFIBVJxtvNBytuRQnaazN+MKvVtyzYLHqy+TD
  ArTbhTEO18s=
  -----END CERTIFICATE-----

# image mirroring 성공 후 출력된 문구 입력
imageContentSources:
- mirrors:
  - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - bastion.ocp.beomzh.com:5000/ocp/ocp4.10.9
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

3) Ignition File 생성

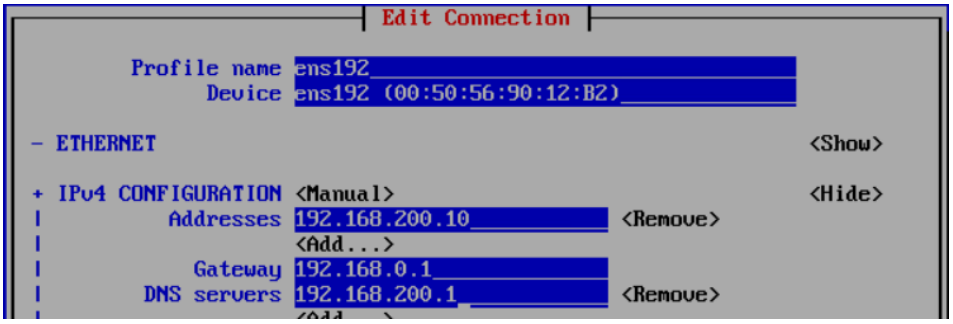
```
# dir 생성 후 config.yaml 파일 복사
# ingnition file 생성시 기존의 install-config.yaml이 삭제 되므로 에러추적을 위해 복사
mkdir install_dir && cd $_
cp ../install-config.yaml ./


# master 노드에서 worker 역할을 하지 않도록 하려면 mastersSchedulable을 false로 변경한다.
# line 7
vi /root/ocp/install_dir/manifests/cluster-scheduler-02-config.yml
mastersSchedulable: true -> false

# Ignition 생성
openshift-install create ignition-configs --dir=/root/ocp/install_dir
cp -vrp /root/ocp/install_dir/*.ign /var/www/html/ocp/.
cp -vrp /root/ocp/install_dir/metadata.json /var/www/html/ocp/.
chmod -R 755 /var/www/html/
```

```
# 메타데이터가 리턴되는지 테스트
curl localhost:8080/ocp/metadata.json
{"clusterName":"ocp","clusterID":"7b419995-70a7-4f04-acb6-9d2d5b4be16c","infraID":"ocp-dwff2'
```

Node 구축

- redhatLinux 8
- rhcos 4.10.x 으로 설치
- 작업: 1.ssh 접속 허용 2.passwd 변경 3.network 설정

```
sudo passwd core
nmtui # 이미지 참고 interface / ip / dns /gateway 수정

# nmcli connection up [interface명]
nmcli connection up ens192
hostname -I # 변경 IP 확인

# ssh 설정
# line 70
sudo vi /etc/ssh/sshd_config
PasswordAuthentication no -> yes

sudo systemctl restart sshd

# ssh tool로 ssh 확인
```




```
# bootstrap
sudo coreos-installer install --copy-network --ignition-url=http://bastion.ocp.beomzh.com:808
# master
sudo coreos-installer install --copy-network --ignition-url=http://bastion.ocp.beomzh.com:808
# worker
sudo coreos-installer install --copy-network --ignition-url=http://bastion.ocp.beomzh.com:808

# 작업 정상 완료 후
reboot

# Node가 bastion을 바라보기까지 시간이 좀 소비된다.
```

- zone 설정의 문제로 에러 발생
- dns / rev 둘중 하나 설정 상 오류

- dns 서버 설정의 오류
- cat /etc/resolv.conf 를 확인하면 namesever ip가 나오는데 bastion을 바라봐야 한다.
- 현재 이미지의 DNS 주소를 보면 192.168.23.2 로 설정했는데 192.168.200.7 로 변경하여 bastion을 바라보게 설정해 주면 해결된다.
- zone 설정 중 rev IP를 정상 입력하지 않으면 reboot 시 연결을 실패해 boot 되지 않음
  - sudo systemctl status haproxy → LB 상태 체크 후 다운 상태면 기동

```
Error: downloading source Ignition config http://bastion.ocp.beomzh.com:8080/ocp/bootstrap.ig

Caused by:
    0: fetching 'http://bastion.ocp.beomzh.com:8080/ocp/bootstrap.ign'
    1: error sending request for url (http://bastion.ocp.beomzh.com:8080/ocp/bootstrap.ign):
    2: error trying to connect: dns error: failed to lookup address information: Name or serv
    3: dns error: failed to lookup address information: Name or service not known
    4: failed to lookup address information: Name or service not known
```

Bootstrap 로그 확인

```
# bastion에서 ssh로 접근
journalctl -b -f -u release-image.service -u bootkube.service
# 99? 같은로그가 찍혀야 정상이라는데...
```

node Status 확인

```
mkdir ~/.kube
cp /root/ocp/install_dir/auth/kubeconfig ~/.kube/config
oc completion bash > oc_bash_completion
cp oc_bash_completion /etc/bash_completion.d/
source /etc/bash_completion.d/oc_bash_completion
```

CSR(Certificate Signing Requests) 승인

```
# 개별 승인을 하고자 하는 경우 해당 명령어로 이름 확인후 승인
oc get csr | grep -v NAME | awk '{print $1}'

# 해당 명령어로 전체 노드 승인
oc adm certificate approve $(oc get csr | grep -v NAME | awk '{print $1}')

oc get nodes
```

설치 확인

```
watch -n1 -d "oc get no,co,csr"
```

```
# 생각보다 오래걸림 = 잘못되어서 그런가보다...
# 나의 경우 up to 40m 나옴...
openshift-install --dir=/root/ocp/install_dir wait-for install-complete
```

```
[root@bastion install_dir]# openshift-install --dir=/root/ocp/install_dir wait-for install-complete
INFO Waiting up to 40m0s (until 1:26AM) for the cluster at https://api.ocp.beomzh.com:6443 to initialize...
INFO Waiting up to 10m0s (until 12:56AM) for the openshift-console route to be created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/root/ocp/install_dir/auth/kubecon
fig'
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp.beomzh.com
INFO Login to the console with user: "kubeadmin", and password: "zNRLf-Ct8cv-tLio4-Vt8kb"
INFO Time elapsed: 0s
```

- oc get co 명령어로 cluster operator를 조회하였을때 이러한 현상이 발생
  - install-config에 설정된 master/worker의 적정 수량을 맞춰줘야 정상 기동 되는걸로 판단
  - 현재 master:1 / worker:1 → 3/2 로 스펙 업 완료
  - master node spec = 4/16/30
  - worker node spec = 4/16/30
  - 아래 1번 image → 2번 image롤 변경되는 시간이 생각보다 길다...
  -

1번 image

```
[root@bastion ocp]# oc get co
NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
authentication                             4.10.9    False       False         True       57m     APIServicesAvailable: Preco
baremetal                                  4.10.9    True        False         False      56m
cloud-controller-manager                   4.10.9    True        False         False      63m
cloud-credential                                     True        False         False      163m
cluster-autoscaler                         4.10.9    True        False         False      56m
config-operator                            4.10.9    True        False         False      57m
console
csi-snapshot-controller                    4.10.9    False       True          False      22m     CSISnapshotControllerAvaila
dns                                        4.10.9    True        False         False      55m
etcd                                                 False       True          True       56m     StaticPodsAvailable: 0 node
image-registry
ingress                                              False       True          True       55m     The "default" ingress contr
: The deployment has Available status condition set to False (reason: MinimumReplicasUnavailable) with message: Deployment do
insights                                   4.10.9    True        False         False      50m
kube-apiserver                                       False       True          True       57m     StaticPodsAvailable: 0 node
kube-controller-manager                    4.10.9    True        True          True       53m     InstallerPodContainerWaitin
se ContainerCreating...
kube-scheduler                             4.10.9    False       True          False      57m     StaticPodsAvailable: 0 node
kube-storage-version-migrator              4.10.9    True        False         False      44m
machine-api                                4.10.9    True        False         False      56m
machine-approver                           4.10.9    True        False         False      55m
machine-config                             4.10.9    True        False         False      56m
marketplace                                4.10.9    True        False         False      55m
monitoring                                           False       True          True       42m     Rollout of the monitoring s
network                                    4.10.9    True        True          True       58m     DaemonSet "openshift-ovn-ku
node-tuning                                4.10.9    True        False         False      56m
openshift-apiserver                        4.10.9    False       False         True       57m     APIServicesAvailable: Preco
openshift-controller-manager               4.10.9    True        False         False      44m
openshift-samples
operator-lifecycle-manager                 4.10.9    True        False         False      56m
operator-lifecycle-manager-catalog         4.10.9    True        False         False      56m
operator-lifecycle-manager-packageserver             False       False         False      56m     ClusterServiceVersion opens
dy before timeout: deployment "packageserver" exceeded its progress deadline
service-ca                                 4.10.9    True        False         False      57m
storage                                    4.10.9    True        False         False      57m
[root@bastion ocp]#
```

2번 image

```
[root@bastion install_dir]# oc get co
NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
authentication                             4.10.9    True        False         False      15m
baremetal                                  4.10.9    True        False         False      3h45m
cloud-controller-manager                   4.10.9    True        False         False      3h50m
cloud-credential                           4.10.9    True        False         False      4h6m
cluster-autoscaler                         4.10.9    True        False         False      3h44m
config-operator                            4.10.9    True        False         False      3h46m
console                                    4.10.9    True        False         False      47m
csi-snapshot-controller                    4.10.9    True        False         False      3h46m
dns                                        4.10.9    True        False         False      3h44m
etcd                                       4.10.9    True        False         False      3h43m
image-registry                             4.10.9    True        False         False      3h37m
ingress                                    4.10.9    True        False         False      50m
insights                                   4.10.9    True        False         False      3h39m
kube-apiserver                             4.10.9    True        False         False      3h34m
kube-controller-manager                    4.10.9    True        False         False      3h43m
kube-scheduler                             4.10.9    True        False         False      3h40m
kube-storage-version-migrator              4.10.9    True        False         False      3h46m
machine-api                                4.10.9    True        False         False      3h45m
machine-approver                           4.10.9    True        False         False      3h45m
machine-config                             4.10.9    True        False         False      3h45m
marketplace                                4.10.9    True        False         False      3h45m
monitoring                                 4.10.9    True        False         False      47m
network                                    4.10.9    True        False         False      3h47m
node-tuning                                4.10.9    True        False         False      50m
openshift-apiserver                        4.10.9    True        False         False      3h34m
openshift-controller-manager               4.10.9    True        False         False      3h45m
openshift-samples                          4.10.9    True        False         False      3h37m
operator-lifecycle-manager                 4.10.9    True        False         False      3h45m
operator-lifecycle-manager-catalog         4.10.9    True        False         False      3h45m
operator-lifecycle-manager-packageserver   4.10.9    True        False         False      3h37m
service-ca                                 4.10.9    True        False         False      3h46m
storage                                    4.10.9    True        False         False      3h46m
[root@bastion install_dir]#
```

HAproxy 수정 - bootstrap 제외

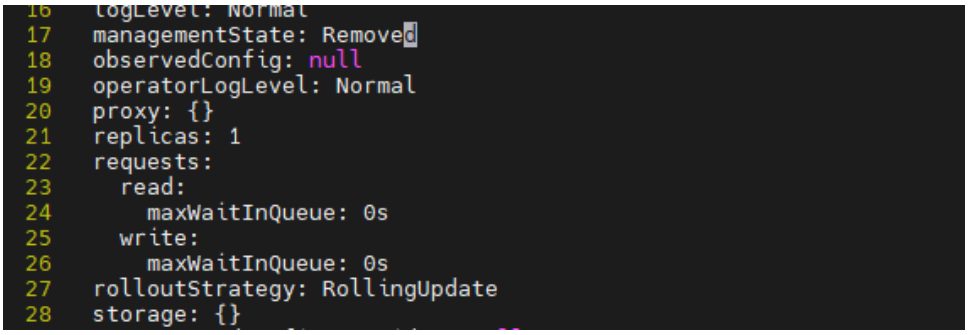
```
vi /etc/haproxy/haproxy.cfg
# bootstrap 정보 주석처리

systemctl restart haproxy
```

```
    mode tcp
  # server bootstrap 192.168.200.10:6443 check fall 3 rise 2
    server master 192.168.200.11:6443 check fall 3 rise 2
    server master-1 192.168.200.12:6447 check fall 3 rise 2

rontend machine-config-server
    bind *:22623
    default_backend machine-config-server
    mode tcp
    option tcplog

ackend machine-config-server
    balance source
    mode tcp
  # server bootstrap 192.168.200.10:22623 check fall 3 rise 2
```

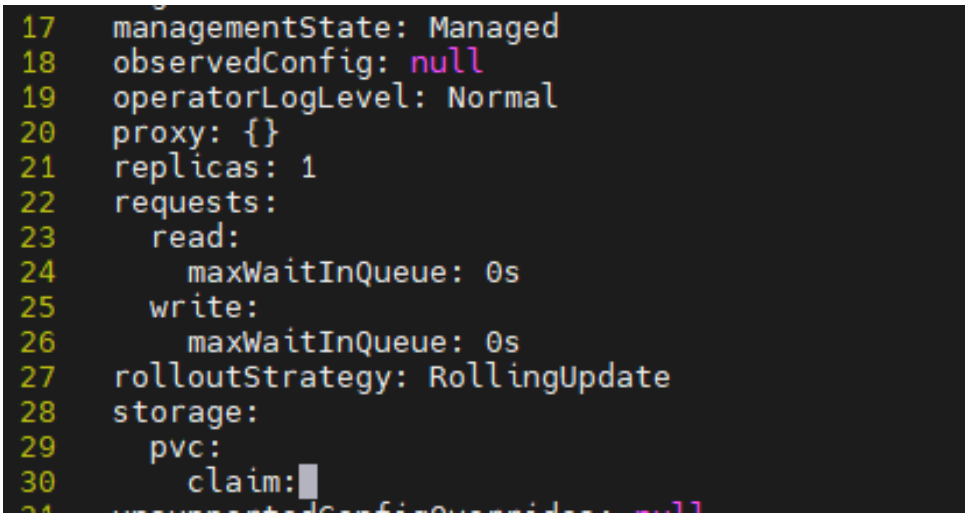
Image Registry Storage(PV) 구성

```
# vi /root/ocp/registry_pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: image-registry-volume
spec:
  capacity:
    storage: 100Gi
  nfs:
    server: 192.168.200.1   #nfs server IP = bastion IP
    path: /DATA
  storageClassName: ""
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
```

```
# yaml 파일 생성 및 점검
oc create -f /root/ocp/registry_pv.yaml
oc get pv
oc edit configs.imageregistry.operator.openshift.io

# 아래 image 참고 17 , 29 변경
# Removed -> Managed
# storage: {} -> 아래와 같이
# storage:
#   pvc:
#     claim:
```



위에서 아래..



정상 완료시 STATUS가 Bound

kubeconfig 복사

```
cp /root/ocp/install_dir/auth/kubeconfig /root/.kube/config
echo "export KUBECONFIG=/root/ocp/install_dir/auth/kubeconfig" >> ~/.bash_profile
source ~/.bash_profile
```

argument 값으로 로그인

```
oc login --kubeconfig=/root/ocp/install_dir/auth/kubeconfig
# 정상처리
# You must obtain an API token by visiting https://oauth-openshift.apps.ocp.beomzh.com/oauth/
```

kubeadmin 외 계정 만들기

```
# 계정 생성
htpasswd -c -B -b /root/ocp/admin.htpasswd admin opennaru

# yaml파일과 매칭 시킬 secret 생성
oc create secret generic htpasswd-secret \
--from-file=htpasswd=/root/ocp/admin.htpasswd -n openshift-config

oc adm policy add-cluster-role-to-user cluster-admin admin

# vi admin_htpasswd.yaml
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
  - name: htpasswd
    mappingMethod: claim
    type: HTPasswd
    htpasswd:
      fileData:
        name: htpass-secret

# yaml 작성 완료
oc apply -f admin_htpasswd.yaml
```

web-console

```
 oc whoami --show-console
 # https://console-openshift-console.apps.ocp.beomzh.com
```

window에서 설정 (C:\Windows\System32\drivers\etc/hosts) 파일에 입

```
192.168.200.1    console-openshift-console.apps.ocp.beomzh.com  oauth-openshift.apps.ocp.beom
```

접속 확인

```
# console 주소 확인
oc whoami --show-server
# https://api.ocp.beomzh.com:6443

oc login -u admin -p opennaru https://api.ocp.beomzh.com:6443
```