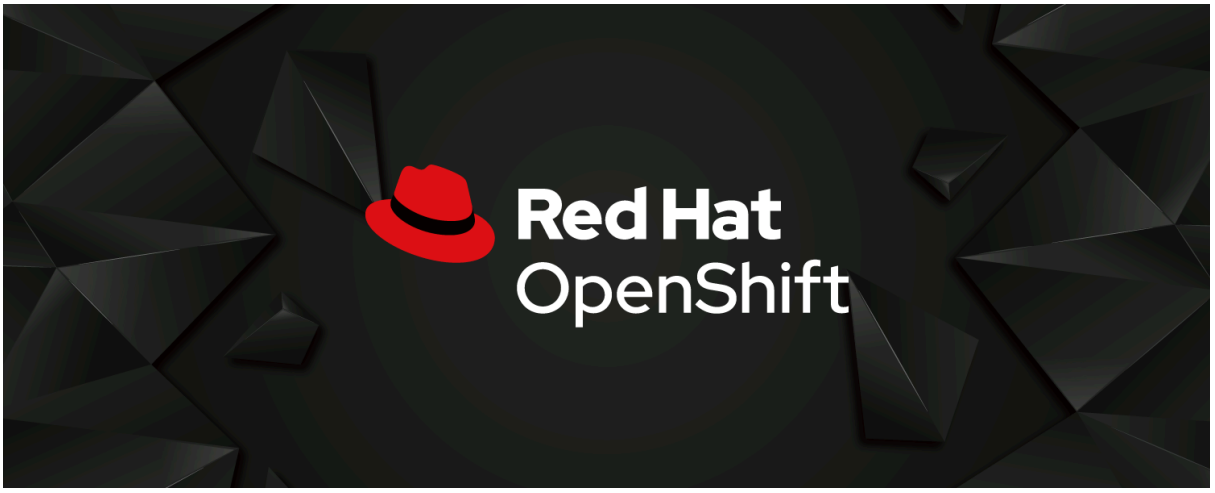

3. Bastion Server 구축 (DNS, KeepAlived, HAProxy)

> **References**
>
> - https://lilo.tistory.com/123
> - https://lilo.tistory.com/124

※ Bastion 서버의 역할

OpenShift는 일반적인 웹사이트와 달리, 내부 통신과 설치 과정에서 도메인 이름(FQDN)을 많이 사용

- **Local DNF Repository:** 외부 인터넷(Red Hat CDN)에 접속하지 않고, 로컬 서버(Bastion 등) 내부에 보관된 RPM 패키지 파일들을 사용하여 소프트웨어를 설치하고 업데이트할 수 있도록 만든 저장소
- **DNS 서버 대행:** 외부 DNS가 없다면, Bastion 서버에 CoreDNS나 Bind9 같은 DNS 서비스를 직접 올려서 클러스터 전용 DNS 서버 역할을 수행해야 함. Bastion이 없다면 노드들은 서로의 이름을 해석하지 못해 설치 불가능.
- **API 및 Ingress 접근:** `api.<cluster_name>.<base_domain>` 같은 주소가 있어야 노드끼리 서로를 찾고 설치를 진행가능.

→ 다음 가이드에서는 Local DNF Repository, DNS 서버, VIP & Proxy 역할을 모두 Bastion 서버에서 수행하도록 구성

※ Local DNF Repository 구성

→ RedHat 에서 RPM 을 다운로드 하여 내부 폐쇄망 환경에 설치 할 RPM 제공하는 Repository 구성

→ 구성 시간이 오래걸려 외부망이 연결 된 환경에서는 Skip 권장

```
## 최신버전 RPM 미러링 (버전 주의 RHEL 버전, RHCOS 버전 확인 필요)
# 외부망 접근이 가능한 PC 에서 실행시키는 명령어 (Bastion 에서 해도 무방)
reposync -p /home/ocp/repo/ --download-metadata --repoid=rhel-9-for-x86_64-baseos-rpms
reposync -p /home/ocp/repo/ --download-metadata --repoid=rhel-9-for-x86_64-appstream-rp
ms
reposync -p /home/ocp/repo/ --download-metadata --repoid=rhocp-4.20-for-rhel-9-x86_64-r
pms
reposync -p /home/ocp/repo/ --download-metadata --repoid=fast-datapath-for-rhel-9-x86_64
-rpms

## 외부에서 RPM 미러링 후 Repository Server(Bastion 서버)로 이동시키는 명령어
rsync -az  /home/ocp/repo/*  <계정>@<Repository 서버 IP>:/data/repo/ocp

## Bastion 에서 직접 RPM 미러링 시
mkdir -p /data/repo/ocp
rsync -az /home/ocp/repo/* root@localhost:/data/repo/ocp

## Apache 설치 후 8080 으로 포트 변경
dnf install httpd
sed -i "s/Listen\ 80/Listen\ 8080/g"  /etc/httpd/conf/httpd.conf
systemctl enable --now httpd

curl -IHEAD localhost:8080/repo/ocp


## DNF Repository 설정 {BastionIP} 는 고정 IP 입력
cat << EOF > /etc/yum.repos.d/ocp.repo
[rhel-9-for-x86_64-baseos-rpms]
name=rhel-9-for-x86_64-baseos-rpms
baseurl=http://{BastionIP}:8080/repo/rhel-9-for-x86_64-baseos-rpms
gpgcheck=0
enabled=1

[rhel-9-for-x86_64-appstream-rpms]
name=rhel-9-for-x86_64-appstream-rpms
baseurl=http://{BastionIP}:8080/repo/rhel-9-for-x86_64-appstream-rpms
gpgcheck=0
enabled=1

[rhocp-4.20-for-rhel-9-x86_64-rpms]
name=rhocp-4.20-for-rhel-9-x86_64-rpms
baseurl=http://{BastionIP}:8080/repo/rhocp-4.20-for-rhel-9-x86_64-rpms
```

```
gpgcheck=0
enabled=1

[fast-datapath-for-rhel-9-x86_64-rpms]
name=fast-datapath-for-rhel-9-x86_64-rpms
baseurl=http://{BastionIP}:8080/repo/fast-datapath-for-rhel-9-x86_64-rpms
gpgcheck=0
enabled=1
EOF

dnf repolist -vv |grep -i repo-id
```

※ DNS 서버 구축 (BIND)

- Bastion 1 (Master) / Bastion 2 (Slave) 구조로 이중화 DNS 서버를 구축

```
### Bastion 1 서버에서 실행
## Master 서버 DNS 구성
dnf install -y bind bind-utils
cp /etc/named.conf /etc/named.conf_bak
cp /etc/sysconfig/named /etc/sysconfig/named_bak

# IPv6 비활성화
sed -i "s/listen-on\ port\ 53\ { 127.0.0.1; };/listen-on\ port\ 53\ { any; };/g" /etc/named.conf
sed -i "s/listen-on-v6\ port\ 53\ { ::1; };/listen-on-v6\ port\ 53\ { none; };/g" /etc/named.conf
sed -i "s/allow-query\ \ \ \ { localhost; };/allow-query\ \ \ \ { any; };/g" /etc/named.conf
echo 'include "/etc/tb_ocp.named.rfc1912.zones";' >> /etc/named.conf
echo 'OPTIONS="-4"' >> /etc/sysconfig/named

# DNS Zone 설정 - {Bastion Master Node IP} 수정 필요
cat << EOF > /etc/tb_ocp.named.rfc1912.zones
//RHOCP DNS Forward Zone
zone "example.com" IN {
        type master;
        file "example.com.zone";
        allow-update { {Bastion Master Node IP}; };
};

//RHOCP DNS Reverse Zone
zone "101.168.192.in-addr.arpa" IN {
        type master;
        file "101.168.192.in-addr.rev";
```

```
        allow-update { {Bastion Master Node IP}; };
};
EOF

## 정방향 Zone 파일 생성
cat << EOF > /var/named/example.com.zone
\$TTL 1W
@       IN      SOA     ns1.example.com. root.example.com. (
                        2025111101 ; serial
                        3H         ; refresh
                        30M        ; retry
                        2W         ; expiry
                        1W )       ; minimum (1 week)

; Name servers
        IN      NS      ns1.example.com.
        IN      NS      ns2.example.com.

; DNS servers
ns1.example.com.    IN  A   {Bastion Node 1 IP}
ns2.example.com.    IN  A   {Bastion Node 2 IP}

; Bastion
bastion1.ocp4       IN  A   {Bastion Node 1 IP}
bastion2.ocp4       IN  A   {Bastion Node 2 IP}

; Bootstrap
bootstrap.ocp4      IN  A   {Bootstrap Node IP}

; Control Plane
master1.ocp4        IN  A   {Master Node 1 IP}
master2.ocp4        IN  A   {Master Node 2 IP}
master3.ocp4        IN  A   {Master Node 3 IP}

; Compute Nodes
worker1.ocp4        IN  A   {Worker Node 1 IP}
worker2.ocp4        IN  A   {Worker Node 2 IP}

; API / Ingress
api.ocp4            IN  A   {VIP}
api-int.ocp4        IN  A   {VIP}
*.apps.ocp4         IN  A   {VIP}

; Registry
harbor.example.com.  IN  A   {Harbor Registry IP}
EOF

## 역방향 Zone 파일 생성 -- HostID 변경 필요 (50, 116 등)
```

```
cat << EOF > /var/named/101.168.192.in-addr.rev
\$TTL 1W
@   IN  SOA ns1.example.com. root.example.com. (
        2025111001
        3H
        30M
        2W
        1W )

; NS
    IN  NS  ns1.example.com.
    IN  NS  ns2.example.com.

; Bastion
174      IN  PTR bastion1.ocp4.example.com.
175     IN  PTR bastion2.ocp4.example.com.

; Bootstrap
1     IN  PTR bootstrap.ocp4.example.com.

; Control Plane
2      IN  PTR master1.ocp4.example.com.
3      IN  PTR master2.ocp4.example.com.
4      IN  PTR master3.ocp4.example.com.

; Worker
5      IN  PTR worker1.ocp4.example.com.
6      IN  PTR worker2.ocp4.example.com.

; API / Ingress
30      IN  PTR api.ocp4.example.com.
30      IN  PTR api-int.ocp4.example.com.
30      IN  PTR apps.ocp4.example.com.

; Registry (Harbor는 별도 zone)
83      IN  PTR harbor.example.com.
EOF

## DNS 서버 시작
systemctl enable --now named
systemctl restart named
systemctl status named

## Zone 파일 scp (Bastion 1 → Bastion2)
scp /etc/tb_ocp.named.rfc1912.zones root@192.168.101.175:/etc/
scp /var/named/example.com.zone root@192.168.101.175:/var/named/
scp /var/named/101.168.192.in-addr.rev root@192.168.101.175:/var/named/
```

```
### Bastion 2 서버에서 실행
## Slave 서버 DNS 구성
dnf install -y bind bind-utils
cp /etc/named.conf /etc/named.conf_bak
cp /etc/sysconfig/named /etc/sysconfig/named_bak
sed -i "s/listen-on\ port\ 53\ { 127.0.0.1; };/listen-on\ port\ 53\ { any; };/g" /etc/named.conf
sed -i "s/listen-on-v6\ port\ 53\ { ::1; };/listen-on-v6\ port\ 53\ { none; };/g" /etc/named.conf
sed -i "s/allow-query\ \ \ \ { localhost; };/allow-query\ \ \ \ { any; };/g" /etc/named.conf
echo 'include "/etc/tb_ocp.named.rfc1912.zones";' >> /etc/named.conf

vi /etc/named.conf

options {
...
                masterfile-format text; ## 설정 추가
...
}

echo 'OPTIONS="-4"' >> /etc/sysconfig/named

## DNS 서버 시작
systemctl enable --now named
systemctl restart named
systemctl status named
```

※ HAProxy + KeepAlived (VIP) 이중화 구성

```
## Master / Slave 동시 실행

## DNS 임시 연결
vi /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.101.174 ## 임시 Bastion1 서버로 연결 (Master DNS)
nameserver 168.126.63.1
nameserver 8.8.8.8

## 방화벽 Off
systemctl disable firewalld && systemctl stop firewalld

## SELinux Disabled
setenforce 0
```

```
vi /etc/selinux/config

...
#    grubby --update-kernel ALL --remove-args selinux
#
SELINUX=disabled ## 수정
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected
...


### HAProxy 설치 및 구성

dnf install -y haproxy

cat << EOF > /etc/haproxy/haproxy.cfg
global
  log         127.0.0.1 local2
  pidfile     /var/run/haproxy.pid
  maxconn     4000
  daemon

defaults
  mode                    http
  log                     global
  option                  dontlognull
  option http-server-close
  option                  redispatch
  retries                 3
  timeout http-request    10s
  timeout queue           1m
  timeout connect         10s
  timeout client          1m
  timeout server          1m
  timeout http-keep-alive 10s
  timeout check           10s
  maxconn                 3000

# K8s API Server
listen api-server-6443
    bind *:6443
    mode tcp
    server bootstrap bootstrap.ocp4.example.com:6443 check  ## Master node 설치 완료 후 주석
교체
    #server master1 master1.ocp4.example.com:6443 check inter 1s
    #server master2 master2.ocp4.example.com:6443 check inter 1s
    #server master3 master3.ocp4.example.com:6443 check inter 1s
```

```
# RHOCP MC Server
listen machine-config-server-22623
    bind *:22623
    mode tcp
    server bootstrap bootstrap.ocp4.example.com:22623 check  ## Master node 설치 완료 후 주석 교체
    #server master1 master1.ocp4.example.com:22623 check inter 1s
    #server master2 master2.ocp4.example.com:22623 check inter 1s
    #server master3 master3.ocp4.example.com:22623 check inter 1s

# RHOCP Ingress Router for 443 port
listen ingress-router-443
    bind *:443
    mode tcp
    balance source
    server worker1 worker1.ocp4.example.com:443 check inter 1s
    server worker2 worker2.ocp4.example.com:443 check inter 1s

# RHOCP Ingress Router for 80 port
listen ingress-router-80
    bind *:80
    mode tcp
    balance source
    server worker1 worker1.ocp4.example.com:80 check inter 1s
    server worker2 worker2.ocp4.example.com:80 check inter 1s
EOF

systemctl enable --now haproxy
systemctl status haproxy
systemctl restart haproxy
```

```
## KeepAlived 구성
dnf install -y keepalived

cp /etc/keepalived/keepalived.conf  /etc/keepalived/keepalived.conf_bak

cat << EOF > /etc/keepalived/keepalived.conf
vrrp_script check_haproxy
{
    script "/usr/bin/systemctl is-active --quiet haproxy"
    interval 5
    fall 2
    rise 2
}

vrrp_instance OCP {
```

```
    state BACKUP
    interface enX0 #### 사용할 Network Interface 명 설정 필요
    virtual_router_id 50
    priority 200
    advert_int 5
    nopreempt
    authentication {
        auth_type PASS
        auth_pass passwd123
    }
    virtual_ipaddress {
    192.168.101.30/24 #### VIP 설정 필요
    }
    track_script
    {
        check_haproxy
    }

    notify_master /etc/keepalived/start.sh
    notify_backup /etc/keepalived/start.sh
    notify_fault /etc/keepalived/start.sh
}
EOF

## KeepAlived 기동
systemctl enable --now keepalived
systemctl status keepalived
systemctl restart keepalived

## Active 된 서버에서 VIP 확인 가능
ip a |grep inet |grep enX0
    inet 192.168.101.175/24 brd 192.168.101.255 scope global dynamic noprefixroute enX0
    inet 192.168.101.30/24 scope global secondary enX0

## 각 Bastion 서버에 모두 설정
# 본인에게 없는 IP(VIP)라도 서비스를 바인딩할 수 있게 허용
echo "net.ipv4.ip_nonlocal_bind = 1" >> /etc/sysctl.conf
sysctl -p

## 각 서버에 DNS 설정 (/etc/resolv.conf)

search ocp4.example.com example.com
nameserver 192.168.101.30   # VIP (Primary)
...

systemctl restart named
```