openstack+openshift_자동화

Status

가상머신 디스크 이미지 수정

Master Node 3 / Worker Node 2

```
## root 패스워드 변경
# virt-customize -a rhel-server~.qcow2 --root-password password:redhat
## repo 추가
guestfish --rw -a rhel-server~.qcow2
# run
# list-filesystems
/dev/sda1: xfs # 마운트시킬 파일시스템이 출력됨
# mount /dev/sda1 / # 루트에 마운트시켜야 이미지 수정 가능
# touch /etc/yum.repos.d/ocp.repo # repo추가
================================
[rhel-7-server-rpms]
name=rhel-7-server-rpms
baseurl=http://172.16.4.100/rhel-7-server-rpms # 172.16.4.100 : bastion의 ip
enabled=1
.....
================================
## NTP 추가
# /etc/chrony.conf
================================
# Use public servers from the pool.ntp.org project.
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 172.16.4.100 iburst # 주석처리 후 bastion의 ip추가
================================
# exit
```

Bastion 노드 구성

REPO 구성

```
## subscription-manager 사용을 위해 dns 서버 등록
# /etc/resolv.conf
nameserver 8.8.8.8
## subscription-manager를 사용하여 필요한 RPM를 활성화시킵니다.
# subscription-manager register
# subscription-manager list --available --all
# subscription-manager attach --pool=<풀아이디등록>
# subscription-manager repos --enable=rhel-7-fast-datapath-rpms \
--enable=rhel-7-server-ansible-2.7-rpms --enable=rhel-7-server-ose-3.9-rpms \
--enable=rhel-7-server-extras-rpms --enable=rhel-7-server-rpms
## repo저장소 구성에 필요한 웹서비스 패키지 다운로드
# yum install -y httpd
# systemctl start httpd
# systemctl enable httpd
# yum install -y createrepo # repo저장소를 만드는 패키지
# yum repolist all
# cd /var/www/html
# reposync -n ./ # Redhat Repo Package를 로컬디렉토리에 다운로드
# createrepo . (각 rpm폴더) # 각 디렉토리를 repo로 생성함 (repo의 그룹 데이터에
접근가능하게함)
```

DNS 구성

```
# yum install -y bind bind-utils # DNS 서비스 패키지 설치
# vi /etc/named.conf # DNS 서버의 DB 및 zone 파일의 위치, 접근제어 등 주요 설정
===========================================================
options {
listen-on port 53 { any; }; # 127.0.0.1 -> any로 변경
listen-on-v6 port 53 { none; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";
recursing-file "/var/named/data/named.recursing";
secroots-file "/var/named/data/named.secroots";
allow-query { any; }; # localhost -> any로 변경
============================== 중략==========================

zone "." IN {
type hint;
file "named.ca";
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
# OCP 배포에 사용할 도메인 oneshot.co.kr zone 파일을 참고하도록 설정합니다.
# 아래 내용을 새로 추가합니다.
zone "oneshot.co.kr" IN {
type master;
file "oneshot.co.kr.zone";
allow-update{ none; };
};
===========================================================
## ZONE 파일을 생성 후 소유자를 named로 변경해줍니다.
# vi /var/named/oneshot.co.kr.zone
===========================================================
$TTL 3H
@ SOA @ root. (2 1D 1H 1W 1H)
IN NS bastion.oneshot.co.kr.
IN A 172.16.4.100
ns IN A 172.16.4.100
registry IN A 172.16.4.100
master1 IN A 172.16.4.51
master2 IN A 172.16.4.52
master3 IN A 172.16.4.53
infra1 IN A 172.16.4.54
infra2 IN A 172.16.4.55
service1 IN A 172.16.4.56
service2 IN A 172.16.4.57
console IN A 172.16.4.58
bastion IN A 172.16.4.100
*.app IN A 172.16.4.54
tower IN A 172.16.
===========================================================
# chown named:named /var/named/oneshot.co.kr.zone
## DNS 서비스를 재시작해줍니다.
# systemctl enable named
# systemctl start named
```

```
```

시간 동기화 서버 구성(chrony 사용)

```
# vi /etc/chrony.conf # chrony서비스 설정파일 수정
============================================================
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (<http://www.pool.ntp.org/join.html>).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 1.kr.pool.ntp.org iburst
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3
# Enable kernel synchronization of the real-time clock (RTC).
rtcsync
# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *
# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2
# Allow NTP client access from local network.
allow 172.16.3.0/24
# Serve time even if not synchronized to a time source.
#local stratum 10
# Specify file containing keys for NTP authentication.
#keyfile /etc/chrony.keys
# Specify directory for log files.
logdir /var/log/chrony
# Select which information is logged.
#log measurements statistics tracking
============================================================
## chrony 서비스 재시작 및 동기화 활성화
# systemctl restart ntp
# timedatectl set-ntp true
```

Docker Private Registry 구성

```
# yum install -y docker docker-distribution
## docker 기본 설정파일 수정
# vi /etc/sysconfig/docker
## 아래와 같은 내용을 맨 밑에 추가합니다.
ADD_REGISTRY='--add-registry registry.oneshot.co.kr:5000'
BLOCK_REGISTRY='--block-registry docker.io'
INSECURE_REGISTRY='--insecure-registry registry.oneshot.co.kr:5000'
# vi /etc/docker-distribution/registry/config.yml
==========================================
version: 0.1
log:
fields:
service: registry
storage:
cache:
layerinfo: inmemory
filesystem:
rootdirectory: /docker-registry
http:
```

```
addr: :5000
headers:
X-Content-Type-Options: [nosniff] # 보안 관련 설정
## private registry의 작동을 확인하기 위한 health check 설정
health:
storagedriver:
enabled: true
interval: 60s
threshold: 3
==========================================
# systemctl start docker
# systemctl enable docker
# systemctl start docker-distribution
# systemctl enable docker-distribution
```

docker 이미지를 가져온 뒤 태그를 달아 push 해줍니다.

```
#!/bin/bash
docker pull registry.access.redhat.com/openshift3/ose-ansible:v3.9.99
docker pull registry.access.redhat.com/openshift3/ose-cluster-capacity:v3.9.99
docker pull registry.access.redhat.com/openshift3/ose-deployer:v3.9.99
```

```
#!/bin/bash
private_registry=registry.oneshot.co.kr:5000
docker_images=$(docker images | grep "registry.access.redhat.com" | awk '{print
$1":"$2}')
for beforeimage in $docker_images;
do
afterimage="$private_registry/$(echo $beforeimage | cut -d/ -f2,3)"
echo "docker tag $beforeimage $afterimage"
sudo docker tag $beforeimage $afterimage
done
```

```
#!/bin/bash
docker_images=$(sudo docker images|grep registry.oneshot.co.kr:5000 | awk '{print
$1":"$2}')
for push_image in $docker_images; do
echo "docker push $push_image"
sudo docker push $push_image
done
```

Gitlab 노드 구성

```
# curl -sS
https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo
bash
# yum update
# yum install -y gitlab-ce
# gitlab-ctl reconfigur
```

OCP 노드 구성

☐ Openstack Instance 생성

```
#!/bin/bash
for i in \\
master1.oneshot.co.kr \\
master2.oneshot.co.kr \\
master3.oneshot.co.kr \\
infra1.oneshot.co.kr \\
infra2.oneshot.co.kr \\
service1.oneshot.co.kr \\
service2.oneshot.co.kr \\
lb.oneshot.co.kr
do
ssh root@$i "hostname; yum -y install wget git net-tools bind-utils yum-utils
iptables-services bridge-utils bash-completion kexec-tools sos psacct docker;
systemctl start docker; systemctl enable docker;";
done
```

☐ OpenShift 설치

```
# yum install -y atomic-openshift-utils
# vi /etc/ansible/hosts
==================================================
[OSEv3:children]
masters
nodes
etcd
lb
[OSEv3:vars]
ansible_ssh_user=root
deployment_type=openshift-enterprise
openshift_release="3.9"
openshift_image_tag="v3.9.99"
openshift_pkg_version=-3.9.99
openshift_clock_enabled=true
os_firewall_use_firewalld=false
openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind':
'HTPasswdPasswordIdentityProvider', 'filename': '/etc/origin/htpasswd'}]
openshift_master_cluster_hostname=lb.oneshot.co.kr
openshift_master_cluster_public_hostname=console.oneshot.co.kr
openshift_master_cluster_method=native
#if docker image error
openshift_disable_check=docker_image_availability
# Configuring Master API and Console Ports
openshift_master_api_port=8443
openshift_master_console_port=8443
os_sdn_network_plugin_name='redhat/openshift-ovs-multitenant'
osm_cluster_network_cidr=10.128.0.0/14
osm_host_subnet_length=9
openshift_portal_net=172.30.0.0/16
openshift_master_default_subdomain=app.oneshot.co.kr
osm_default_node_selector='region=nodes'
# ADD web_console_prefix
openshift_web_console_prefix=registry.oneshot.co.kr:5000/openshift3/ose-
# Configuring a Registry Location
oreg_url=registry.oneshot.co.kr:5000/openshift3/ose-${component}:${version}
openshift_docker_additional_registries=registry.oneshot.co.kr:5000
openshift_docker_insecure_registries=registry.oneshot.co.kr:5000
```

```
openshift_docker_blocked_registries=docker.io
openshift_docker_options="--selinux-enabled --insecure-registry 172.30.0.0/16 -l warn --log-driver
--log-opt max-size=100M --log-opt max-file=10 "
# openshift imagesstream
openshift_examples_modify_imagestreams=true
openshift_install_examples=true
# Configuring Dedicated Infrastructure Nodes
openshift_router_selector='region=infra,zone=default'
openshift_hosted_router_replicas=2
# openshift registry
openshift_registry_selector='region=infra,zone=default'
openshift_hosted_registry_replicas=2
openshift_enable_unsupported_configurations=True
#cockpit
openshift_cockpit_deployer_prefix=registry.oneshot.co.kr:5000/openshift3
openshift_cockpit_deployer_version=v.3.9.99
#catalog
openshift_enable_service_catalog=false
#service broker template
template_service_broker_install=false
#ansible service broker
ansible_service_broker_install=false
#certification
openshift_hosted_registry_cert_expire_days=3650
openshift_ca_cert_expire_days=3650
openshift_node_cert_expire_days=3650
openshift_master_cert_expire_days=3650
etcd_ca_default_days=3650
#host group for masters
[masters]
master1.oneshot.co.kr
master2.oneshot.co.kr
master3.oneshot.co.kr
# host group for etcd
[etcd]
master1.oneshot.co.kr
master2.oneshot.co.kr
master3.oneshot.co.kr
[lb]
lb.oneshot.co.kr
# host group for nodes, includes region info
[nodes]
master1.oneshot.co.kr openshift_node_labels="{'region': 'master','zone': 'default'}"
master2.oneshot.co.kr openshift_node_labels="{'region': 'master','zone': 'default'}"
master3.oneshot.co.kr openshift_node_labels="{'region': 'master','zone': 'default'}"
infra1.oneshot.co.kr openshift_node_labels="{'region': 'infra','zone': 'default'}"
infra2.oneshot.co.kr openshift_node_labels="{'region': 'infra','zone': 'default'}"
service1.oneshot.co.kr openshift_node_labels="{'region': 'nodes','zone': 'default'}"
service2.oneshot.co.kr openshift_node_labels="{'region': 'nodes','zone': 'default'}" json-file
```

- [ ] OpenShift 설치 (ansible - host 확인)

```
ansible-playbook -i /etc/ansible/hosts
/usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml
ansible-playbook -i /etc/ansible/hosts
/usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml
```

Open Stack 구성

☐ Open Stack 설치 (ansible -인스턴스 생성)

```
---
- name: Create instance for bastion
 hosts: localhost
 vars_files: vars/secret.yaml
 tasks:
 - name: create volume
 os_volume:
 state: present
 size: 150
 display_name: vol_bastion
 image: image_default
 timeout: 1800
 - name: create instance
 os_server:
 state: present
 key_name: "bastion-key"
 flavor: "{{ Bastion_set['flavor'] }}"
 boot_volume: vol_bastion
 security_groups: "{{ Bastion_set['security'] }}"
 floating_ips:
 - "{{ Bastion_set['flip'] }}"
 nics:
 - { net-id: "{{ net_internal_id }}", v4-fixed-ip: "{{ Bastion_set['fxip'] }}" }
 name: "{{ Bastion_set['name'] }}"
 wait: yes
 meta:
 group: ocp
 userdata: |
 #cloud-config
 disable_root: false
 ssh_pwauth: true
 hostname: "bastion.oneshot.co.kr"
```

☐ tasks/dns.yaml

```
- name: Install DNS packages
 yum:
 name:
 - bind
 - bind-utils
 state: latest
- name: Copy zone file
 copy:
 src: files/oneshot.co.kr.zone
 dest: /var/named/oneshot.co.kr.zone
 owner: named
 group: named
- name: Copy named.conf file
 copy:
 src: files/named.conf
 dest: /etc/named.conf
 owner: named
 group: named
 mode: u=rw,g=r
- name: Start DNS server
 systemd:
 name: named
 state: started
 enabled: yes
```

☐ tasks/chrony.yaml

```
- name: Copy chrony.conf file
 copy:
 src: files/chrony_server.conf
 dest: /etc/chrony.conf
 mode: 0644
- name: Retart Chronyd
 systemd:
 name: chronyd
 state: restarted
 enabled: yes
- shell: timedatectl set-ntp true
```

☐ repo.yaml

```
---
- name: Create Repo
 hosts: bastion
 vars_files:
 - vars/repos_list.yaml
 tasks:
 - name: Install required packages for mirroring repo
 yum:
 name:
 - httpd
 - createrepo
 state: latest
 - name: Start and Enable httpd
 systemd:
 name: httpd
 state: started
 enabled: yes
 - name: Download Redhat Repo Packages to local directory
 shell: "reposync -n -p /var/www/html/"
 register: repo_result
 until: repo_result is succeeded
 delay: 10
 retries: 50
 - name: Download ose rpms all to local directory
 shell: "reposync -r rhel-7-server-ose-3.9-rpms -p /var/www/html/"
 register: repo_result
 until: repo_result is succeeded
 delay: 10
 retries: 50
 - name: Create Repo based on local directory
 shell: createrepo /var/www/html/{{ item }}
 loop: "{{ rhsm_repos }}"
 - name: Restorecon /var/www/html
 shell: "restorecon -RFv /var/www/html"
```

☐ docker_install.yaml

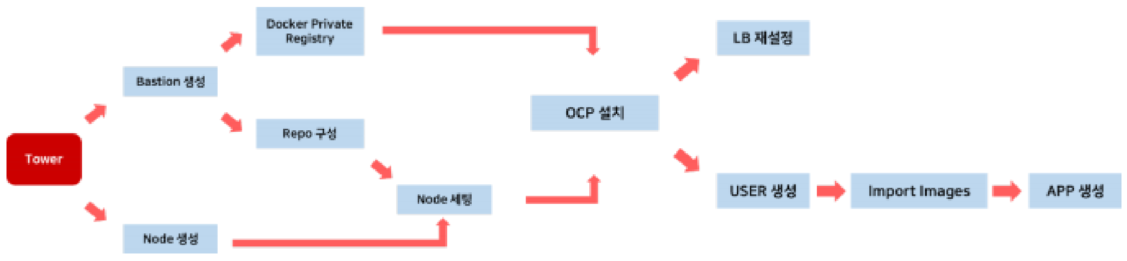
```
---
- name: Set Docker private registry
 hosts: bastion
 vars_files: vars/image_list.yaml
```

```
tasks:
- name: Install packages for docker private registry
yum:
name:
- docker
- docker-distribution
state: latest
- name: Copy docker config file
copy:
src: files/docker
dest: /etc/sysconfig/docker
mode: 644
owner: root
group: root
- name: Copy docker private registry config file
copy:
src: files/docker_registry.yml
dest: /etc/docker-distribution/registry/config.yml
mode: 644
owner: root
group: root
- name: systemd daemon reload
systemd:
daemon_reload: yes
- name: Start docker
systemd:
name: docker
state: started
enabled: yes
- name: Start docker-distribution
systemd:
name: docker-distribution
state: started
enabled: yes
- name: wait
wait_for:
timeout: 300
- name: Pull images
docker_image:
name: "registry.access.redhat.com{{ item }}"
source: pull
docker_host: tcp://172.16.4.100:4243
register: pulling until: pulling is succeeded
delay: 3
loop: "{{ private_images }}"
delegate_to: 127.0.0.1
- name: Restart docker
systemd:
name: docker
state: restarted
- name: Restart docker-distribution
systemd:
name: docker-distribution
state: restarted
- name: Tag and push to local registry
docker_image:
name: "registry.access.redhat.com{{ item }}"
repository: "registry.oneshot.co.kr:5000{{ item }}"
tag: latest
push: yes
source: local
docker_host: tcp://172.16.4.100:4243
loop: "{{ private_images }}"
delegate_to: 127.0.0.1
```

도커 프라이빗 레지스트리를 사용하기 위한 설정파일도 변경

```
# /etc/sysconfig/docker
# Modify these options if you want to change the way the docker daemon runs
OPTIONS='--selinux-enabled --log-driver=journald --signature-verification=false -H
unix:///var/run/docker.sock -H tcp://0.0.0.0:4243'
if [ -z "${DOCKER_CERT_PATH}" ]; then
 DOCKER_CERT_PATH=/etc/docker
fi
# docker-latest daemon can be used by starting the docker-latest unitfile.
# To use docker-latest client, uncomment below lines
#DOCKERBINARY=/usr/bin/docker-latest
#DOCKERDBINARY=/usr/bin/dockerd-latest
#DOCKER_CONTAINERD_BINARY=/usr/bin/docker-containerd-latest
#DOCKER_CONTAINERD_SHIM_BINARY=/usr/bin/docker-containerd-shim-latest
ADD_REGISTRY='--add-registry registry.oneshot.co.kr:5000'
BLOCK_REGISTRY='--block-registry docker.io'
INSECURE_REGISTRY='--insecure-registry registry.oneshot.co.kr:5000'# /etc/docker-distribution/registry/config.yml
version: 0.1
log:
 fields:
 service: registry
storage:
 cache:
 layerinfo: inmemory
 filesystem:
 rootdirectory: /docker-registry
http:
 addr: :5000
 headers:
 X-Content-Type-Options: [nosniff]
health:
 storagedriver:
 enabled: true
 interval: 60s
 threshold: 3
```

Openshift 자동화 설계도


Tower
Bastion 생성
Node 생성
Docker Private Registry
Repo 구성
Node 세팅
OCP 설치
LB 재설정
USER 생성
Import Images
APP 생성