

# Openstack 을 이용한 kubernetes 환경 구축

김명수

# Openstack - 설치 과정

## 1. cpu 가상화 지원 확인

- `cat /proc/cpuinfo | egrep ' (svm|vmx|lm) '`

## 2. SELINUX 비활성화

- `vi /etc/selinux/config`

- `SELINUX=disabled`

## 3. NetworkManager 비활성화

- `systemctl stop NetworkManager`

## 4. 방화벽 비활성화

- `systemctl stop firewalld`

## 5. 호스트명 변경

- `hostnamectl set-hostname desktop.example.com`

# Openstack - 설치 과정

## 6. centos-release-openstack 설치

- `yum install centos-release-openstack-stein`

## 7. yum update

- `yum update -y`

## 8. openstack-packstack 설치

- `yum install openstack-packstack`

## 9. 재부팅

- `systemctl reboot`

## 10. packstack 설치 실행

- `packstack --allinone --provision-demo=n --os-neutron-ovs-bridge-mappings=extnet:br-ex --os-neutron-ovs-bridge-interfaces=br-ex:enp2s0 --os-neutron-ml2-type-drivers=vlan,flat,local`

# Openstack - 설치 과정

## 11. bridge network 구성

### - ifcfg-enp2s0

```
DEVICE=enp2s0  
NAME=enp2sp  
DEVICETYPE=ovs  
TYPE=OVSPort  
OVS_BRIDGE=br-ex  
ONBOOT=yes  
BOOTPROTO=none  
~
```

### - ifcfg-br-ex

```
DEVICE=br-ex  
DEVICETYPE=ovs  
TYPE=OVSBridge  
BOOTPROTO=static  
IPADDR=172.16.83.2  
NETMASK=255.255.255.0  
GATEWAY=172.16.83.254  
DNS1=8.8.8.8  
ONBOOT=yes  
~
```

## 12. 네트워크 재시작

### - systemctl restart network

# Openstack - 설치 과정

## 13. openstack 설치 완료

172.16.83.2/dashboard/auth/login/?next=/dashboard/project/



openstack®

로그인

사용자 이름

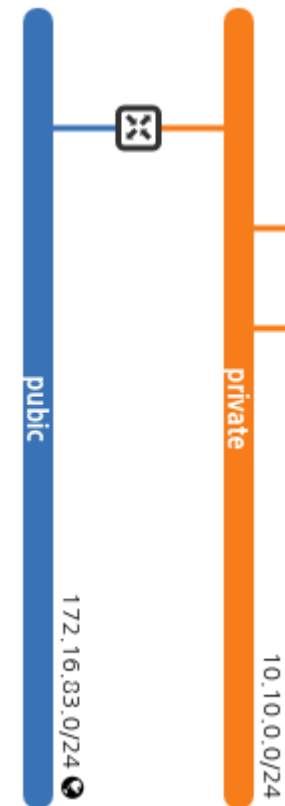
암호

로그인

# Openstack - 네트워크 설정

## 1. Network Topology 구성

- source keystonerc\_admin
- neutron net-create public --router:external=True
- neutron subnet-create --name public\_subnet --enable\_dhcp=False --allocation\_pool start=172.16.83.21,end=172.16.83.29 -gateway=172.16.83.254 public 172.16.83.0/24
- neutron net-create private\_network
- neutron subnet-create private\_network 10.10.0.0/24 --name private
- neutron router-create router
- neutron router-gateway-set router public
- neutron router-interface-add router private



# Openstack - 인스턴스 구성하기

## 1. 이미지 생성

이미지 생성

이미지 상세 정보

메타데이터

이미지 상세 정보

이미지 서비스에 업로드할 이미지를 지정합니다.

이미지 이름

centos7

이미지 설명

이미지 소스

파일\*

검색... CentOS-7-x86\_64-Generic

포맷\*

QCOW2 - QEMU 에뮬레이터 (Emu)

이미지 요구 사항

커널 (Kernel)

이미지 선택

램디스크 (Ramdisk)

이미지 선택

아키텍처

최소 디스크 (GB)

0

최소 RAM (MB)

0

이미지 공유

가시성

사설 공유 커뮤니티 공용

보호됨

예 아니오

취소

뒤로 다음

이미지 생성

프로젝트

API 액세스

Compute

개요

인스턴스

이미지

필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.

+ 이미지 생성

- 이미지 삭제

2 항목 표시

키 페어	소유자	이름	유형	상태	가시성	보호됨
볼륨						
네트워크						
오브젝트 스토리지						
관리						
인증						

이름	상태	가시성	보호됨	실행
admin centos7	Active	공유	아니오	실행
admin cirros	Active	공유	아니오	실행

2 항목 표시

# Openstack- 인스턴스 구성하기

## 2. 인스턴스 생성

새부 정보

인스턴스 초기 호스트 이름, 배포할 가용 구역, 인스턴스 수를 입력하십시오. 동일한 설정으로 여러 인스턴스를 생성하려는 수를 입력하면 됩니다.

인스턴스 이름 \*

master

설명

가용 구역

nova

계수 \*

1

인스턴스 총계 (10 Max)

30%

2 현재 사용량

1 추가됨

7 기억하기

소스 \*

Flavor \*

네트워크 \*

네트워크 포트

보안 그룹

키 페어

구성

서버 그룹

스케줄러 힌트

메타데이터

새부 정보

Flavor는 인스턴스의 컴퓨터, 메모리 및 스토리지 용량에 대한 크기를 관리합니다.

할당됨

이름	VCPUS	RAM	디스크 총계	공용
m1.medium	2	4 GB	40 GB	예

사용 가능 1

하나 선택

필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.

이름	VCPUS	RAM	디스크 총계	공용
m1.tiny	1	512 MB	1 GB	예
m1.small	1	2 GB	20 GB	예
master	4	4 GB	50 GB	예
m1.large	4	8 GB	80 GB	예
m1.xlarge	8	16 GB	160 GB	예

네트워크 포트

보안 그룹

키 페어

구성

서버 그룹

스케줄러 힌트

메타데이터

Flavor

할당됨

1 항목 표시

이름

key

1 항목 표시

사용 가능 1

하나 선택

필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.

0 항목 표시

이름

표시할 항목이 없습니다.

0 항목 표시

키 페어

구성

서버 그룹

스케줄러 힌트

메타데이터

새부 정보

인스턴스 소스는 인스턴스를 생성할 때 사용되는 템플릿입니다. 인스턴스의 스냅샷 이미지를 사용할 수 있습니다. (이미지 스냅샷, 볼륨 또는 볼륨스냅샷 (활성화된 경우). 또한 새 볼륨을 생성할 때 영구적인 스토리지를 사용하도록 선택할 수 있습니다.

부팅 소스 선택

이미지

새로운 볼륨 생성

예

아니오

할당됨

이름	업데이트 완료	크기	유형	가시성
centos7	7/17/20 1:56 PM	819.00 MB	qcow2	공용

사용 가능 1

하나 선택

필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.

이름	업데이트 완료	크기	유형	가시성
centos7	7/17/20 1:56 PM	819.00 MB	qcow2	공용

소스 \*

Flavor \*

네트워크 \*

네트워크 포트

보안 그룹

키 페어

구성

서버 그룹

스케줄러 힌트

메타데이터

새부 정보

네트워크는 클라우드 내 인스턴스에 대한 통신 채널을 제공합니다.

할당됨 1

아래의 목록에서 네트워크를 선택합니다.

네트워크	공용	관리자 상태	상태
private	아니오	Up	Active

사용 가능 1

적어도 하나 이상 네트워크를 선택합니다.

필터 또는 전체 텍스트 검색은 여기를 클릭하십시오.

네트워크	공용	관리자 상태	상태
public	아니오	Up	Active

네트워크 \*

네트워크 포트

보안 그룹

키 페어

구성

서버 그룹

스케줄러 힌트

메타데이터



# Openstack - 인스턴스 구성하기

## 2. 인스턴스 생성

개요

인스턴스

이미지

키 페어

서버 그룹

볼륨

네트워크

오브젝트 스토리지

관리

인증

Instance ID =

필터

인스턴스 시작

인스턴스 삭제

기타 작업 ▼

2 항목 표시

	인스턴스 이름	이미지 이름	IP 주소	Flavor	키 페어	Status	가용 구역	작업	전원 상태	기간 (Age)	작업
<input type="checkbox"/>	node1	centos7	10.10.0.34, 172.16.83.21	m1.medium	key	Shutoff	nova	None	Shut Down	17시간, 30분	인스턴스 시작 ▼
<input type="checkbox"/>	master	centos7	10.10.0.74, 172.16.83.26	m1.medium	key	Shutoff	nova	None	Shut Down	17시간, 30분	인스턴스 시작 ▼

2 항목 표시

# Openstack - kubernetes 환경 구축

## Kubernetes 자동 구축 start.sh

3 - ansible 환경 구축

4,5 - ansible 을 이용한 쿠버네티스 환경 구축

7 - kubernetes 에서 wordpress 을 생성하기

위한 쉘스크립트

9~15 - wordpress service 활성화 확인후

Openstack 보안정책에 포트 추가

```
1 #!/bin/bash
2
3 bash ./init/node_init.sh
4 cd /root/master_work/kubernetes-ansible/centos
5 ansible-playbook pull_setting.yml
6
7 cat ~/master_work/wordpress | ssh master.example.com sh
8
9 port=$(ssh master.example.com "kubectl get service" | awk '/wordpress/{print $
5}' | cut -d: -f 2 | cut -d/ -f 1)
10
11 if [ -n "$port" ]
12 then
13
14     source ~/keystonerc_admin
15     neutron security-group-rule-create --protocol tcp --port-range-min $
{port} --port-range-max ${port} --direction ingress b528842a-d70d-43c2-8b7a-
80e737e400bb
16 fi
~
--
```

# Openstack - start.sh 분석

## 1. ansible 환경 구축

### init\_host.sh

3 - known\_hosts 를 초기화

5~10 - ssh 끊김 방지 및 fingerprint 묻기 생략

코드가 없을 시 .ssh/config 파일에 추가

12~13 - 인스턴트에 마스터로 접속하기 위한  
설정

15~16 - 호스트네임 변경

```
3 bash ~/master_work/init/init_host.sh
4
5 grep ServerAlive ~/.ssh/config > /dev/null
6
7 if [ $? -eq 1 ]
8 then
9     cat ~/master_work/init/config_setting.txt >> ~/.ssh/config
10 fi
11
12 cat ~/master_work/init/delauth.txt | ssh -i ~/.ssh/key.pem centos@node1.example.com sh
13 cat ~/master_work/init/delauth.txt | ssh -i ~/.ssh/key.pem centos@master.example.com sh
14
15 ssh master.example.com 'hostnamectl set-hostname master.example.com'
16 ssh node1.example.com 'hostnamectl set-hostname node1.example.com'
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

- prerequisites.yml
  - swap 비활성화
- setting\_up
  - docker 및 kubernetes 설치
- configure
  - kubernetes 환경 설

```
--  
- import_playbook: playbooks/prerequisites.yml  
- import_playbook: playbooks/setting_up_master.yml  
- import_playbook: playbooks/setting_up_nodes.yml  
- import_playbook: playbooks/configure_master_node.yml  
- import_playbook: playbooks/configure_worker_nodes.yml
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

prerequisites.yml

```
--  
- hosts: all  
  become: yes  
  vars_files:  
  - env_variables  
  tasks:  
  - name: Disabling Swap on all nodes  
    shell: swapoff -a  
  
  - name: Commenting Swap entries in /etc/fstab  
    replace:  
      path: /etc/fstab  
      regexp: '(^/. *swap*)'  
      replace: '# W'
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

setting\_up\*.yaml

```
--
- hosts: kubernetes-master-nodes
  become: yes
  vars_files:
    - env_variables
  tasks:
    - name: Creating a repository file for Kubernetes
      file:
        path: /etc/yum/repos.d/kubernetes.repo
        state: touch

    - name: Adding repository details in Kubernetes repo file.
      blockinfile:
        path: /etc/yum/repos.d/kubernetes.repo
        block: |
          [kubernetes]
          name=Kubernetes
          baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
          enabled=1
          gpgcheck=1
          repo_gpgcheck=1
          gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

```
- name: setenforce 0
  lineinfile:
    path: /etc/selinux/config
    regexp: 'SELINUX=en'
    line: 'SELINUX=permissive'

- name: Creating a iptables
  file:
    path: /etc/sysctl.d/k8s.conf
    state: touch

- name: Adding repository details in Kubernetes repo file.
  blockinfile:
    path: /etc/sysctl.d/k8s.conf
    block: |
      net.bridge.bridge-nf-call-ip6tables = 1
      net.bridge.bridge-nf-call-iptables = 1

- name: sysctl --system
  shell: "sysctl --system"

- name: Installing Docker
  yum:
    name: "{{ item }}"
    state: latest
  loop:
    - yum-utils
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

setting\_up\*.yaml

```

- name: docker repo
  yum:
    - device-mapper-persistent-data
    - lvm2

- name: docker repo
  shell: "yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo"

- name: Installing Docker and firewall
  yum:
    name: docker-ce
    state: latest

- name: create directory
  file:
    path: /etc/docker
    state: directory
    mode: '0755'

- name: copy daemon
  copy:
    src: /root/master_work/kubernetes-ansible/centos/playbooks/daemon.json
    dest: /etc/docker/daemon.json
```

```

- name: copy hosts
  copy:
    src: /root/master_work/kubernetes-ansible/centos/playbooks/hosts
    dest: /home/centos/hosts

- name: add hosts
  shell: "cat /home/centos/hosts >> /etc/hosts"

- name: copy kubernetes yaml file
  copy:
    src: /root/master_work/kubernetes-ansible/kube-practice-yaml.tar
    dest: /root/kube-practice-yaml.tar

- name: install unzip
  yum:
    name: "{{ item }}"
    state: present
  loop:
    - unzip
    - rsync

- name: unarchive latest.tar.gz
  unarchive:
    src: /root/kube-practice-yaml.tar
    dest: /root/
    remote_src: yes
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

setting\_up\*.yaml

```
- name: create directory
  file:
    path: /etc/systemd/system/docker.service.d
    state: directory
    mode: '0755'

- name: Installing required packages
  yum:
    name: "{{ item }}"
    state: present
    disable_excludes: kubernetes
  with_items: "{{ packages }}"

- name: Starting and Enabling the required services
  service:
    name: "{{ item }}"
    state: started
    enabled: yes
  with_items: "{{ services }}"
```



# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

### configure\_master\_node.yml

```
...
- hosts: kubernetes-master-nodes
  become: yes
  vars_files:
  - env_variables
  tasks:
  - name: Pulling images required for setting up a Kubernetes cluster
    shell: kubeadm config images pull

  - name: Initializing Kubernetes cluster
    shell: "kubeadm init --apiserver-advertise-address {{ad_addr}} --pod-network-cidr={{cidr_v}}"
    register: output

  - name: "Storing Logs and Generated token for future purpose,"
    local_action: copy content={{ output.stdout }} dest={{ token_file }}

  - name: Copying required files
    shell: |
      mkdir -p $HOME/.kube
      sudo cp -f /etc/kubernetes/admin.conf $HOME/.kube/config
      sudo chown $(id -u):$(id -g) $HOME/.kube/config

  - name: install bash-completion
    yum:
      name: bash-completion
      state: latest

  - name: Copying required files
    command: echo "source <(<kubect< completion bash)>" >> ~/.bashrc; source ~/.bashrc

  - name: Install calico pod network-1
    command: curl -O https://docs.projectcalico.org/manifests/calico.yaml

  - name: Install calico pod network-2
    command: sed s/192.168.0.0/16/20.96.0.0/12/g -i calico.yaml

  - name: Install calico pod network-3
    command: kubectl apply -f calico.yaml
~
~
```

# Openstack - start.sh 분석

## 2. ansible 을 이용한 쿠버네티스 환경 구축

configure\_worker\_nodes.yml

```
---
- hosts: kubernetes-worker-nodes
  become: yes
  vars_files:
    - env_variables
  tasks:
    - name: Copying token to worker nodes
      copy: src={{ token_file }} dest=join_token

    - name: Joining worker nodes with kubernetes master
      shell: |
        cat join_token | tail -2 > out.sh
        sh out.sh
  ~
  ~
```

env\_variables

---

```
ad_addr: 10.10.0.74
cidr_v: 20.96.0.0/12

packages:
- kubeadm
- kubelet
- kubectl

services:
- docker
- kubelet

token_file: join_token
~
```

# Openstack - start.sh 분석

## 3. kubernetes 에서 wordpress 을 생성하기 위한 셸스크립트

### Wordpress.sh

```
#!/bin/bash
```

```
kubectl apply -f ~/test/pv1.yaml
kubectl apply -f ~/test/pv2.yaml
kubectl apply -f ~/test/httpd-pvc.yaml
```

```
echo -n "Running ."
while [ true ]
do
    con=`kubectl get pvc | awk /mysql/ | awk /Bound/`
    if [ -n "$con" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done
```

```
kubectl apply -f ~/test/wp-pvc.yaml
echo -n "Running ."
while [ true ]
do
    con=`kubectl get pvc | awk /wp/ | awk /Bound/`
    if [ -n "$con" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done
```

```
kubectl create secret generic mysql-password --from-literal=password=mypass
kubectl describe secret mysql-password
```

```
kubectl apply -f ~/test/mysql.yaml
echo -n "Running ."
while [ true ]
do
    con=`kubectl get pods | awk /mysql/ | awk /running/`
    if [ -n "$con" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done
```

```
kubectl apply -f ~/test/wp.yaml
echo -n "Running ."
while [ true ]
do
    con=`kubectl get pods | awk /wp/ | awk /running/`
    if [ -n "$con" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done
```

# Openstack - start.sh 분석

## 3. kubernetes 에서 wordpress 을 생성하기 위한 쉘스크립트

### Wordpress.sh

```
kubectl apply -f ~/test/mysql-service.yaml
echo -n "Running ."
while [ true ]
do
    con=`kubectl get service | awk /mysql/`
    if [ -n "$con" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done

kubectl apply -f ~/test/wp-service.yaml
echo -n "Running ."
while [ true ]
do
    port=`kubectl get service | awk /wordpress/'{print $5}' | cut -d: -f 2 | cut
    -d/ -f 1`
    if [ -n "$port" ]
    then
        break
    fi
    echo -n "."
    sleep 1
done

echo "finish!!!!!!"
```

# Openstack - start.sh 분석

## 4. 보안 정책의 포트 추가

- ssh 를 통해 포트값 받아오기
- port 변수에 값이 있으면 보안 정책에 포트 추가

```
port=$(ssh master.example.com "kubectl get service" | awk /wordpress/'{print $5}'  
| cut -d: -f 2 | cut -d/ -f 1)  
  
if [ -n "$port" ]  
then  
  
    source ~/keystonerc_admin  
    neutron security-group-rule-create --protocol tcp --port-range-min ${port}  
--port-range-max ${port} --direction ingress b528842a-d70d-43c2-8b7a-80e737e4  
00bb  
fi
```

# Openstack - start.sh 분석

## 4. 보안 정책의 포트 추가

<input type="checkbox"/>	Direction	Ether 타입	IP 프로토콜	포트 범위	원격 IP 접두사	원격 보안 그룹	Description	작업
<input type="checkbox"/>	내보냄	IPv4	전체	전체	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	내보냄	IPv6	전체	전체	:::/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	전체	전체	-	default	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	ICMP	전체	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	TCP	3306 (MYSQL)	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv4	TCP	31463	0.0.0.0/0	-	-	규칙 삭제
<input type="checkbox"/>	들어옴	IPv6	전체	전체	-	default	-	규칙 삭제

# Openstack - wordpress 설치 완료

## Wordpress 구축 완료

