목차

- undercloud node 에서 stack 사용자를 생성

```
[root@osp-director ~]# useradd stack
[root@osp-director ~]# passwd stack
[root@osp-director ~]# echo "stack ALL=(root) NOPASSWD:ALL" | tee -a
/etc/sudoers.d/stack
stack ALL=(root) NOPASSWD:ALL
[root@osp-director ~]# chmod 0440 /etc/sudoers.d/stack
```

- undercloud node 에 FQDN으로 호스트명 설정

```
[root@osp-director ~]# hostnamectl set-hostname osp-director.local.com
[root@osp-director ~]# hostnamectl
   Static hostname: osp-director.local.com
         Icon name: computer-vm
           Chassis: vm
        Machine ID: dd2b112292874ace925a58b7a83c4254
           Boot ID: 15ba823406d448ab9278803e2498bbba
    Virtualization: kvm
  Operating System: Red Hat Enterprise Linux Server 7.7 (Maipo)
       CPE OS Name: cpe:/o:redhat:enterprise_linux:7.7:GA:server
            Kernel: Linux 3.10.0-1062.7.1.el7.x86_64
      Architecture: x86-64
```

- 아래부터는 stack 사용자로 전환하여 작업을 수행 합니다.

```
[root@osp-director ~]# su - stack
[stack@osp-director ~]$
```

- python-tripleoclient 패키지 설치

```
[stack@osp-director ~]$ sudo yum install -y python-tripleoclient
```

- 위 패키지 파일 설치가 완료 되었다면, undercloud.conf 구성파일을 준비해야 함.
- 샘플 파일을 복사하여 수동으로 편집

```
[stack@osp-director ~]$ cp /usr/share/instack-
undercloud/undercloud.conf.sample ~/undercloud.conf
[stack@osp-director ~]$ vim undercloud.conf
```

undercloud.conf

```
[DEFAULT]
image_path = /home/stack/images
## undercloud 서버 FQDN 작성
undercloud_hostname = osp-director.local.com
```

```
## SSL/TLS 디렉터의 공개 API IP 로 SSL/TLS 앤드포인트에 엑세스 해야 하는 IP 작성
undercloud_public_host = 10.10.10.9

## SSL/TLS 사용시 디렉터의 관리 API IP
undercloud_admin_host = 10.10.10.8

## 날짜 및 시간 동기화 하는 서버 IP
undercloud_ntp_servers = 10.10.10.1

## undercloud 프로비저닝 IP, PXE 부팅 서비스에 사용하는IP, eth0번 IP주소
local_ip = 10.10.10.10/24

## undercloud 프로비저닝 NIC 이름
local_interface = eth0

## local_interface 에서 사용할 MTU 값 작성
local_mtu = 1500

## undercloud 에서 introspection에 사용하는 브리지 명, 기본인 br-ctlplane으로 두면 됨
inspection_interface = br-ctlplane

overcloud_domain_name = local.com
generate_service_certificate = false
enable_ui = true

[auth]
undercloud_admin_password = redhat
undercloud_haproxy_stats_password = redhat

[ctlplane-subnet]
cidr = 10.10.10.10/24
dhcp_start = 10.10.10.5
dhcp_end = 10.10.10.50
inspection_iprange = 10.10.10.100,10.10.10.120
gateway = 10.10.10.1
masquerade = true
```

- undercloud 설치

```
[stack@osp-director ~]$ openstack undercloud install -y
....
019-12-03 14:37:12,270 INFO: Created flavor "baremetal" with profile "None"
2019-12-03 14:37:12,341 INFO: Created flavor "control" with profile
"control"
2019-12-03 14:37:15,998 INFO: Created flavor "compute" with profile
"compute"
2019-12-03 14:37:16,050 INFO: Created flavor "ceph-storage" with profile
"ceph-storage"
2019-12-03 14:37:16,100 INFO: Created flavor "block-storage" with profile
"block-storage"
```

```
2019-12-03 14:37:16,152 INFO: Created flavor "swift-storage" with profile
"swift-storage"
2019-12-03 14:37:16,152 INFO: Configuring Mistral workbooks
2019-12-03 14:37:29,212 INFO: Mistral workbooks configured successfully
2019-12-03 14:38:25,268 INFO: Configuring an hourly cron trigger for
tripleo-ui logging
2019-12-03 14:38:26,300 INFO:
#############################################################################
#
Undercloud install complete.


The file containing this installation's passwords is at
/home/stack/undercloud-passwords.conf.


There is also a stackrc file at /home/stack/stackrc.


These files are needed to interact with the OpenStack services, and should
be
secured.


#############################################################################
#
```

- undercloud 패키지 설치 확인

```
[stack@osp-director ~]$ ls
stackrc  undercloud.conf  undercloud.conf.bak  undercloud-passwords.conf
```

- undercloud 설치 확인

```
[stack@osp-director ~]$ sudo systemctl list-units openstack-*
UNIT                                        LOAD   ACTIVE SUB
DESCRIPTION
openstack-glance-api.service                loaded active running OpenStack
Image Service (code-named Glance) API server
openstack-heat-engine.service               loaded active running Openstack
Heat Engine Service
openstack-ironic-conductor.service          loaded active running OpenStack
Ironic Conductor service
openstack-ironic-inspector-dnsmasq.service  loaded active running PXE boot
dnsmasq service for Ironic Inspector
openstack-ironic-inspector.service          loaded active running Hardware
introspection service for OpenStack Ironic
openstack-mistral-api.service               loaded active running Mistral
API Server
openstack-mistral-engine.service            loaded active running Mistral
```

```
Engine Server
openstack-mistral-executor.service           loaded active running Mistral
Executor Server
openstack-nova-api.service                   loaded active running OpenStack
Nova API Server
openstack-nova-compute.service               loaded active running OpenStack
Nova Compute Server
openstack-nova-conductor.service             loaded active running OpenStack
Nova Conductor Server
openstack-nova-scheduler.service             loaded active running OpenStack
Nova Scheduler Server
openstack-swift-account-reaper.service       loaded active running OpenStack
Object Storage (swift) - Account Reaper
openstack-swift-account.service              loaded active running OpenStack
Object Storage (swift) - Account Server
openstack-swift-container-sync.service       loaded active running OpenStack
Object Storage (swift) - Container Sync
openstack-swift-container-updater.service    loaded active running OpenStack
Object Storage (swift) - Container Updater
openstack-swift-container.service            loaded active running OpenStack
Object Storage (swift) - Container Server
openstack-swift-object-expirer.service       loaded active running OpenStack
Object Storage (swift) - Object Expirer
openstack-swift-object-reconstructor.service loaded active running OpenStack
Object Storage (swift) - Object Reconstructor
openstack-swift-object-updater.service       loaded active running OpenStack
Object Storage (swift) - Object Updater
openstack-swift-object.service               loaded active running OpenStack
Object Storage (swift) - Object Server
openstack-swift-proxy.service                loaded active running OpenStack
Object Storage (swift) - Proxy Server
openstack-zaqar@1.service                    loaded active running OpenStack
Message Queuing Service (code-named Zaqar) Server Instance 1

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

23 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/openstack_undercloud

Last update: **2020/05/08 05:21**