

OpenStack Mitaka Step-by-Step 설치

4.4 컴퓨트(Compute) 서비스

4.4.1 컴퓨트 서비스 개요

OpenStack 컴퓨트(Compute)는 서비스 형 인프라 클라우드 컴퓨팅 시스템의 운영과 관리의 가장 중요한 부분을 담당하는 서비스로 주요 모듈은 파이썬으로 구현되어 있다.

OpenStack 컴퓨트 서비스는 인증을 위한 자격증명 서비스, 디스크와 서버 이미지를 위한 이미지 서비스, 사용자와 관리자 인터페이스를 위한 대시보드와 상호작용하여 동작한다. 이미지 접근은 프로젝트와 사용자 별로 제한되며, 인스턴스의 숫자와 같은 제한은 프로젝트 별로 이루어진다.

OpenStack 컴퓨트는 표준 하드웨어로 수평 확장이 가능하며 이미지를 다운로드하여 인스턴스를 구동한다.

OpenStack 컴퓨트는 다음의 영역과 구성 요소로 구성되어 있다.

- nova-api 서비스

최종 사용자의 컴퓨트 API 호출을 처리하며, OpenStack 컴퓨트 API, 아마존 EC2 API 및 권한을 가진 사용자의 관리 작업을 수행하기 위한 특정 API 를 지원한다. 인스턴스 실행과 같은 대부분의 orchestration 작업과 특정 정책을 적용한다.

- nova-api-metadata 서비스

인스턴스의 메타데이터 요청을 처리한다. nova-api-metadata 서비스는 nova-network 가 설치된 다중 호스트에서 실행될 때 사용된다. 자세한 내용은 OpenStack 관리자 가이드의 [메타데이터 서비스](#) 부분을 참조하기 바란다.

- nova-compute 서비스

하이퍼바이저 API 를 통해 가상 머신 인스턴스의 생성과 종료를 실행하는 데몬으로 다음과 같은 서비스가 해당된다.

- XenServer/XCP 를 위한 XenAPI
- KVM 또는 QEMU 를 위한 libvirt
- VMware 를 위한 VMwareAPI

실제 데몬의 동작은 상당히 복잡한 과정을 거치게 된다. 데몬은 큐에 저장된 작업을 받아들인 후에 KVM 인스턴스를 실행하고 상태를 데이터베이스에서 업데이트 하는 등의 일련의 시스템 명령을 수행한다.

- nova-scheduler 서비스

큐에서 가상 머신 인스턴스에 관련된 요청을 처리하고 어떤 컴퓨트 서버 호스트에서 실행할지를 결정한다.

- nova-conductor 모듈

OpenStack Mitaka Step-by-Step 설치

nova-compute 서비스와 데이터베이스 간의 상호작용을 중재하여 nova-compute 서비스가 클라우드 데이터베이스에 직접 접근하는 것을 방지한다. nova-conductor 모듈은 수평확장이 가능하나 nova-compute 서비스가 실행되는 노드에는 배포해서는 안된다. 더 자세한 정보는 [구성 레퍼런스 가이드](#)를 참조하기 바란다.

- nova-cert 모듈
X509 인증에 사용되는 Nova Cert 서비스를 실행하는 서버 데몬이다. euca-bundle-image 를 위한 인증서 생성에 사용되며, EC2 API 에만 필요하다.
- nova-network worker 데몬
nova-compute 서비스와 유사하게 큐로부터 네트워킹 작업을 받아들이고 통신을 처리한다. 브리지 인터페이스 구성이나 IPtable 규칙 변경 등의 작업을 수행한다.
- nova-consoleauth 데몬
콘솔 프록시가 제공하는 사용자용 토큰을 인증한다. nova-novncproxy 와 nova-xvncproxy 를 참조하기 바란다. 이 서비스는 콘솔 프록시 동작을 위해 필수적으로 실행되어야 한다. 클러스터 구성에서 단일한 nova-consoleauth 서비스에 대해서는 하나의 프록시 유형만 실행할 수 있다. [nova-consoleauth 에 관하여](#) 라는 부분을 참조하기 바란다.
- nova-novncproxy 데몬
VNC 연결을 통해 실행 중인 인스턴스 접근을 제공하는 프록시이다. 브라우저 기반의 novnc 클라이언트를 지원한다.
- nova-spicehtml5proxy 데몬
SPICE 연결을 통해 실행 중인 인스턴스 접근을 제공하는 프록시이다. 브라우저 기반의 HTML5 클라이언트를 지원한다.
- nova-xvncproxy 데몬
VNC 연결을 통해 실행 중인 인스턴스 접근을 제공하는 프록시이다. OpenStack 고유의 java 클라이언트를 지원한다.
- nova-cert 데몬
X509 인증서를 처리한다.
- nova 클라이언트
사용자가 테넌트 관리자 또는 최종 사용자로 명령어를 제출하도록 한다.
- 큐(queue)
데몬들 간의 메시지 전달을 위한 중앙 허브로, 주종 RabbitMQ 로 구현되어 있으나 ZeroMQ 와 같은 다른 유형의 AMQP 메시지 큐로 구현될 수 있다.
- SQL 데이터베이스
사용가능한 인스턴스 유형, 사용중인 인스턴스, 가용한 네트워크, 프로젝트 등의 클라우드 인프라 구성 시의 상태와 실행 시의 상태를 저장한다. OpenStack 컴퓨트에서는 이론적으로는 SQLAlchemy 를 지원하는 어떤 유형의 데이터베이스도 지원가능하지만,

OpenStack Mitaka Step-by-Step 설치

테스트나 개발 모드에서는 SQLite3 가 주로 사용되며, 일반적으로는 MySQL 과 PostgreSQL 이 사용된다.

4.4.2 컨트롤러 노드 설치 및 구성

이 섹션에서는 컨트롤러 노드에 컴퓨트 서비스인 Nova 를 설치하고 구성하는 과정에 대해서 설명한다.

- 사전 준비 사항

컴퓨트 서비스 설치와 구성에 앞서 데이터베이스, 서비스 인증 및 API 엔드포인트 생성을 수행한다.

1. 데이터베이스는 다음의 과정을 통해 생성한다.

a. 데이터베이스 접속 클라이언트를 사용하여 데이터베이스 서버에 root 로 로그인한다. 이 때 암호는 앞에서 설정한 *dbpassword* 이다.

```
# mysql -u root -p
Enter password:
```

b. nova_api 데이터베이스와 nova 데이터베이스를 생성한다.

```
MariaDB [(none)]> CREATE DATABASE nova_api;
MariaDB [(none)]> CREATE DATABASE nova;
```

c. 데이터베이스에 권한을 부여한다. 데이터베이스 암호는 앞에서 설정한 *novadbpass* 를 사용한다. 모든 명령은 줄 바꿈 없이 한 줄로 입력한다.

```
MariaDB [(none)]> GRANT ALL PRIVILEGES on nova_api.* TO 'nova'@'localhost'
IDENTIFIED BY 'novadbpass';
MariaDB [(none)]> GRANT ALL PRIVILEGES on nova_api.* TO 'nova'@'%' IDENTIFIED BY
'novadbpass';
MariaDB [(none)]> GRANT ALL PRIVILEGES on nova.* TO 'nova'@'localhost' IDENTIFIED
BY 'novadbpass';
MariaDB [(none)]> GRANT ALL PRIVILEGES on nova.* TO 'nova'@'%' IDENTIFIED BY
'novadbpass';
```

d. 데이터베이스 클라이언트를 종료한다.

2. 관리자 모드로 CLI 명령을 실행하기 위해 admin 인증 스크립트를 실행한다.

```
# source /root/admin-openrc
```

3. 다음의 과정으로 서비스 인증을 생성한다.

a. nova 사용자를 생성한다. 암호는 *novapass* 를 사용한다.

```
# openstack user create --domain default --password-prompt nova
```

OpenStack Mitaka Step-by-Step 설치

```
User Password:
Repeat User Password:

+-----+-----+
| Field   | Value           |
+-----+-----+
| domain_id | dfff203e4310474ab2ff3f8de8c63a98 |
| enabled   | True            |
| id        | f47bca88375b478b84e00a91658501fe |
| name      | nova            |
+-----+-----+
```

b. nova 사용자에게 admin 역할을 부여한다.(이 명령은 출력이 나타나지 않는다)

```
# openstack role add --project service --user nova admin
```

c. nova 서비스 엔티티를 생성한다.

```
# openstack service create --name nova --description "OpenStack Compute" compute

+-----+-----+
| Field   | Value           |
+-----+-----+
| description | OpenStack Compute |
| enabled   | True            |
| id        | 91a077571ad743329059538270cbc8d5 |
| name      | nova            |
| type      | compute         |
+-----+-----+
```

4. public, internal, admin 세 가지의 서비스 API 엔드포인트를 생성한다. 모든 명령어는 줄바꿈 없이 한 줄에 입력한다. 결과 값에서 id는 다른 값을 가질 수 있다.

```
# openstack endpoint create --region RegionOne compute public
http://controller:8774/v2.1/%(tenant_id)s

+-----+-----+
| Field   | Value           |
+-----+-----+
| enabled   | True            |
| id        | bee61345f1d94efbbbe2758ee65e734d |
| interface | public          |
| region    | RegionOne       |
| region_id | RegionOne       |
```

OpenStack Mitaka Step-by-Step 설치

```
| service_id | 91a077571ad743329059538270cbc8d5 |
| service_name | nova |
| service_type | compute |
| url | http://controller:8774/v2.1/%(tenant_id)s |
+-----+-----+

# openstack endpoint create --region RegionOne compute internal
http://controller:8774/v2.1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | d48d0375668e426e9cb69278afa16e9f |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 91a077571ad743329059538270cbc8d5 |
| service_name | nova |
| service_type | compute |
| url | http://controller:8774/v2.1/%(tenant_id)s |
+-----+-----+

# openstack endpoint create --region RegionOne compute admin
http://controller:8774/v2.1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | f420aacb0b524f688e35e574599b3689 |
| interface | admin |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 91a077571ad743329059538270cbc8d5 |
| service_name | nova |
| service_type | compute |
| url | http://controller:8774/v2.1/%(tenant_id)s |
```


OpenStack Mitaka Step-by-Step 설치

d. [DEFAULT] 와 [keystone_authstoken] 섹션에서 자격증명 서비스 접속을 설정한다. nova 사용자 암호로 앞에서 설정한 *novapass* 를 사용한다. [keystone_authstoken] 섹션의 다른 옵션은 주석처리 하거나 삭제한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
auth_strategy = keystone

[keystone_authstoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = novapass
```

e. [DEFAULT] 섹션에서 *my_ip* 옵션을 컨트롤러의 관리 네트워크 인터페이스 주소인 *10.0.0.11* 로 설정한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
my_ip = 10.0.0.11
```

f. [DEFAULT] 섹션에서 네트워킹 서비스를 활성화한다. 기본적으로 컴퓨트 서비스는 내부 방화벽 드라이버를 사용한다. 네트워킹 서비스에 방화벽 드라이버가 포함되어 있으므로, 방화벽 드라이버로 *nova.virt.firewall.NoopFirewallDriver* 를 사용하여 컴퓨트의 방화벽 드라이버를 비활성화 해야 한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

g. [vnc] 섹션에서 컨트롤러의 관리 네트워크 인터페이스 주소를 사용하여 VNC 프록시를 구성한다.

```
# vi /etc/nova/nova.conf
```

OpenStack Mitaka Step-by-Step 설치

```
[vnc]
...
vncserver_listen = $my_ip
vncserver_proxycient_address = $my_ip
```

h. [glance] 섹션에서 이미지 서비스 API의 위치를 설정한다.

```
# vi /etc/nova/nova.conf
[glance]
...
api_servers = http://controller:9292
```

i. [oslo_concurrency] 섹션에서 lock_path를 설정한다.

```
# vi /etc/nova/nova.conf
[oslo_concurrency]
...
lock_path = /var/lib/nova/tmp
```

4. 컴퓨트 서비스 데이터베이스를 채워준다. (deprecation 메시지는 무시한다.)

```
# su -s /bin/sh -c "nova-manage api_db sync" nova
# su -s /bin/sh -c "nova-manage db sync" nova
```

● 설치 완료

1. 컴퓨트 서비스를 시작하고, 시스템 시작시에 자동으로 실행되도록 등록한다.

```
# systemctl enable openstack-nova-api.service openstack-nova-consoleauth.service
openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-
novncproxy.service
# systemctl start openstack-nova-api.service openstack-nova-consoleauth.service
openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-
novncproxy.service
```

4.4.3 컴퓨트 노드 설치 및 구성

이 섹션에서는 컴퓨트 노드에 컴퓨트 서비스인 Nova를 설치하고 구성하는 과정에 대해서 설명한다. 이 서비스는 인스턴스 또는 가상 머신을 배포하는데 사용되는 하이퍼바이저 몇 종류를 지원하는데, 여기에서는 간략하게 가상 머신을 위한 하드웨어 가속을 지원하는 컴퓨트 노드에서는 KVM 확장기능을 갖는 QEMU 하이퍼바이저를 사용하여 구성한다. 구형 하드웨어에서는 일반적인 QEMU 하이퍼바이저를 사용한다. 여기서 설명하는 명령어를 수정하여 복수의 컴퓨트 노드로 수평 확장이 가능하다.

OpenStack Mitaka Step-by-Step 설치

이 섹션에서 설명하는 과정은 첫 번째 컴퓨터 노드에서 수행하는 것을 가정한다. 컴퓨터 노드를 추가하고자 할 경우는 Figure 3의 구성도에 나타나 있는 첫 번째 컴퓨터 노드와 유사한 방식으로 추가 컴퓨터 노드를 준비하고, 추가 노드에 고유한 IP 어드레스를 할당한다.

- 사전 준비 사항

1. CentOS에서는 보안정책 관리를 위해 SELinux가 enforcing 모드로 기본으로 활성화되어 있는데 이를 permissive 모드로 변경하고, 시스템 시작시에 적용되도록 /etc/sysconfig/selinux 파일을 수정한다.(enforcing 모드일 경우 추가 디스크를 마운트 할 때, 쓰기가 허용되지 않는다.)

```
# setenforce 0
# vi /etc/sysconfig/selinux
...
SELINUX=permissive
```

2. VMware Player 구성에서 생성한 Nova 인스턴스 저장용 100 GB의 가상 디스크를 마운트에서 사용한다. 그 과정은 다음과 같다.

a. /dev/sdb 디스크 파티션을 생성한다.

```
# fdisk /dev/sdb
...
Command(m for help): n
Partition type:
  p   primary(0 primary, 0 extended, 4 free)
  e   extended
Select(default p): p
Partition number(1-4, default 1): 1
First sector(2048-104857599, default 2048): 2048
Last sector, +sectors or +size{K,M,G}(2048-209715199, default 209715199):
Using default value 209715199
Partition 1 of type Linux and of size 100 GiB is set

Command(m for help): p

Disk /dev/sdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size(logical/physical): 512 bytes / 512 bytes
I/O size(minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
```

OpenStack Mitaka Step-by-Step 설치

```
Disk identifier: 0xa19d83e2

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1                2048    209715199    104856576   83  Linux

Command(m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

b. 파티션을 ext4 로 포맷한다.

```
# mkfs.ext4 /dev/sdb1
...
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872
...
```

● 구성요소 설치 및 설정

기본 설정 파일에서 주석 처리되어 있는 항목 그대로 두고 바로 아래에 설정 내용을 추가할 것을 권장한다. 설정 파일에서(...) 부분은 기존 설정파일에서 유지되어야 할 디폴트 옵션을 나타낸다. 다음의 모든 과정은 컴퓨터 노드에서 진행한다.

1. Nova 패키지를 설치한다.

```
# yum -y install openstack-nova-compute
```

2. /etc/nova/nova.conf 파일을 다음과 같이 수정한다.

a. [DEFAULT] 와 [oslo_messaging_rabbit] 섹션에서 RabbitMQ 메시지 큐 접속을 설정한다. rabbit_password 에는 *rabbitpass* 를 사용한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
rpc_backend = rabbit

[oslo_messaging_rabbit]
...
```

OpenStack Mitaka Step-by-Step 설치

```
rabbit_host = controller
rabbit_userid = openstack
rabbit_password = rabbitpass
```

b. [DEFAULT] 와 [keystone_authtoken] 섹션에서 자격증명 서비스 접속을 설정한다. nova 사용자 암호로 앞에서 설정한 *novapass* 를 사용한다. [keystone_authtoken] 섹션의 다른 옵션은 주석처리 하거나 삭제한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
auth_strategy = keystone

[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = novapass
```

c. [DEFAULT] 섹션에서 *my_ip* 옵션을 컴퓨터 노드의 관리 네트워크 인터페이스 주소인 10.0.0.31 로 설정한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
my_ip = 10.0.0.31
```

d. [DEFAULT] 섹션에서 네트워킹 서비스를 활성화한다. 기본적으로 컴퓨터 서비스는 내부 방화벽 드라이버를 사용한다. 네트워킹 서비스에 방화벽 드라이버가 포함되어 있으므로, 방화벽 드라이버로 *nova.virt.firewall.NoopFirewallDriver* 를 사용하여 컴퓨터의 방화벽 드라이버를 비활성화 해야 한다.

```
# vi /etc/nova/nova.conf
[DEFAULT]
...
use_neutron = True
```

OpenStack Mitaka Step-by-Step 설치

```
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

e. [vnc] 섹션에서 원격 콘솔 접속을 활성화하고 설정한다. vnc 서버는 모든 IP 주소로부터의 접속을 대기하지만, 프록시 구성요소는 컴퓨트 노드의 관리 인터페이스 IP 주소에서의 접속만 대기한다. 기본 URL 은 웹 브라우저로 이 컴퓨트 노드에서 구동되는 인스턴스에 원격 접속이 가능한 주소를 가리킨다. 만약 controller 라는 호스트 이름을 인식하지 못하는 외부 컴퓨터에서의 웹 브라우저로 원격 콘솔에 접속하고자 할 경우에는 controller 에 해당하는 부분을 관리 네트워크 인터페이스의 주소로 변경해야 한다.

```
# vi /etc/nova/nova.conf
[vnc]
...
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://controller:6080/vnc_auto.html
```

f. [glance] 섹션에서 이미지 서비스 API 의 위치를 설정한다.

```
# vi /etc/nova/nova.conf
[glance]
...
api_servers = http://controller:9292
```

g. [oslo_concurrency] 섹션에서 lock_path 를 설정한다.

```
# vi /etc/nova/nova.conf
[oslo_concurrency]
...
lock_path = /var/lib/nova/tmp
```

● 설치 완료

1. 다음의 명령어로 컴퓨트 노드가 가상 머신을 위한 하드웨어 가속을 지원하는지 확인한다. 명령어에서 1 또는 그 이상의 숫자가 반환되면, 컴퓨트 노드가 하드웨어 가속을 지원하며, 별도의 설정이 필요없다는 것을 의미한다.(VMware Player 에서 컴퓨트 노드를 생성할 때, Virtualize Intel VT-x/EPT or AMD-V/RVI 를 활성화하면 하드웨어 가속을 지원한다)

```
# egrep -c '(vmx|svm)' /proc/cpuinfo
4
```

만약 0 을 반환하면, 컴퓨트 노드가 하드웨어 가속을 지원하지 않으므로 KVM 대신에 QEMU 를 사용하도록 다음과 같이 libvirt 를 설정해야만 한다.

```
# vi /etc/nova/nova.conf
[libvirt]
```

OpenStack Mitaka Step-by-Step 설치

```
...  
virt_type = qemu
```

2. 컴퓨터 서비스를 시작하고, 시스템 시작시에 자동으로 실행되도록 등록한다.

```
# systemctl enable libvirtd.service openstack-nova-compute.service  
# systemctl start libvirtd.service openstack-nova-compute.service
```

3. Nova 인스턴스 저장용 디스크를 `/var/lib/nova/instances/` 에 마운트하고, 시스템 시작시에 자동으로 마운트 되도록 `/etc/fstab` 에 등록한다.(별도의 디스크를 사용하지 않을 경우 이 과정은 생략한다)

- a. 다음의 명령어로 `/dev/sdb1`을 마운트하고 소유권을 변경한다.

```
# mount /dev/sdb1 /var/lib/nova/instances  
# chown -R nova.nova /var/lib/nova/instances
```

- b. `/etc/fstab` 파일을 다음과 같이 수정한다.

```
# vi /etc/fstab  
...  
/dev/sdb1          /var/lib/nova/instances    ext4    defaults    0 0
```

4.4.4 동작 확인

컴퓨터 서비스가 정상적으로 동작하는지를 확인한다. 다음은 컨트롤러 노드에서 수행한다.

1. 관리자 모드로 CLI 명령을 실행하기 위해 `admin` 인증 스크립트를 실행한다.

```
# source /root/admin-openrc
```

2. 각 프로세스가 정상적으로 실행되고 등록되어 있는지를 서비스 구성 요소 리스트를 통해 확인한다. 세 개의 서비스는 컨트롤러 노드에서 실행되어야 하고, `nova-compute` 는 컴퓨터 노드에서 실행되어야 한다.

```
# openstack compute service list  
+-----+-----+-----+-----+-----+-----+-----+  
| Id | Binary          | Host       | Zone   | Status | State | Updated At  
+-----+-----+-----+-----+-----+-----+-----+  
| 1 | nova-scheduler  | controller | internal | enabled | up    | 2016-08-12T12:1  
| 2 | nova-consoleauth | controller | internal | enabled | up    | 2016-08-12T12:1  
| 3 | nova-conductor   | controller | internal | enabled | up    | 2016-08-12T12:1  
| 7 | nova-compute     | compute1  | nova   | enabled | up    | 2016-08-12T12:1  
+-----+-----+-----+-----+-----+-----+-----+
```