

Cloud Management WEB UI

설치 가이드

목차

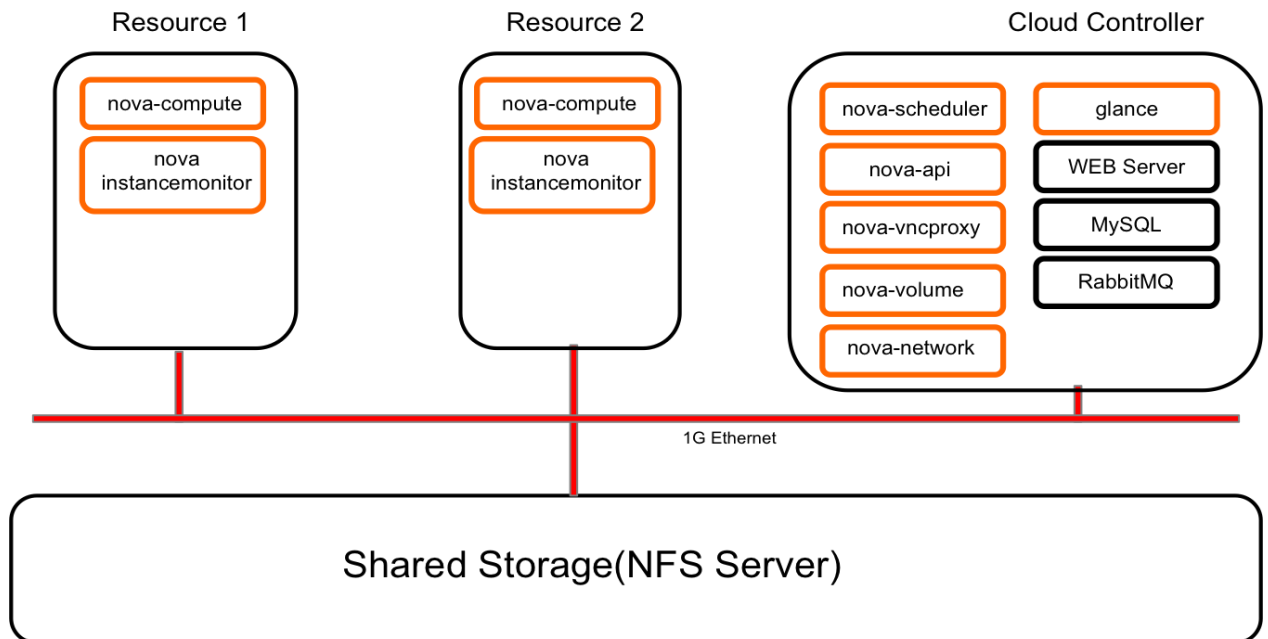
Cloud Management WEB UI 설치 가이드	1
1. 시스템구성	3
1.1 시스템구성도(H/W구성).....	3
1.2 소프트웨어구성	3
2. Cloud Management WEB UI 설치.....	4
2.1 Django와 Apache의 설치 및 연동.....	4
2.2 Cloud Management WEB UI 설정.....	5
2.3 Instance Monitor 설치.....	7
2.4 기타사항	7

개요

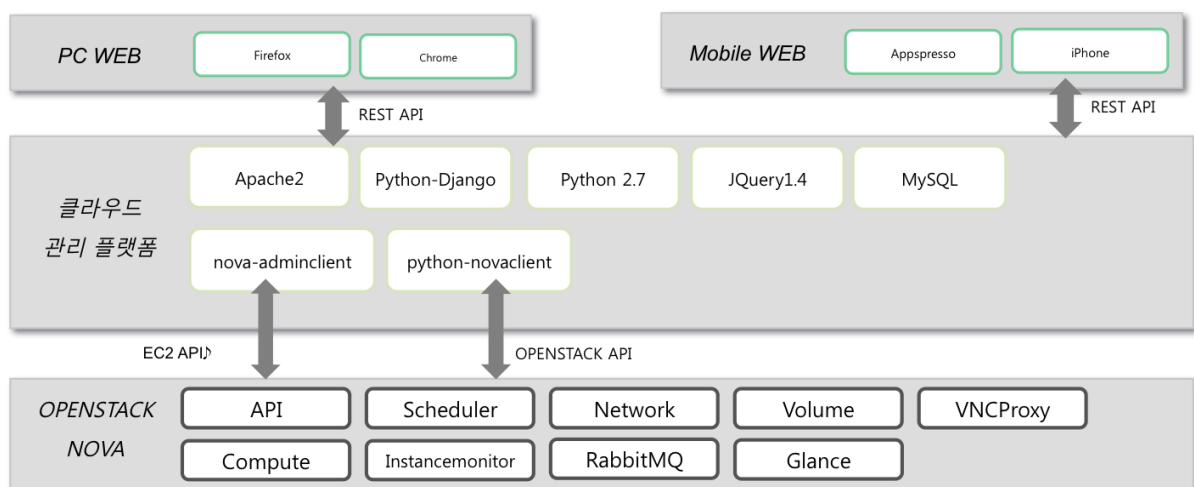
본 가이드는 오픈소스 기반 클라우드 관리 솔루션의 결과물로서, Cloud Management WEB UI의 설치를 위한 가이드를 제공한다.

1. 시스템구성

1.1 시스템구성도(H/W구성)



1.2 소프트웨어구성



2. Cloud Management WEB UI 설치

Cloud Management WEB UI는 python기반의 웹 개발 프레임워크인 Django를 사용하여 개발 되었으며, OpenStack NOVA 를 기반으로 한다.

Management WEB UI를 설치를 위해 Django및 Apache의 설치 및 연동, UI Web application설정, Instance Monitor설치, 기타사항 가이드하고 있으며 본 가이드는 Ubuntu 10.10(Maverick Meerkat) 환경을 기준으로 한다.

2.1 Django와 Apache의 설치 및 연동

- 1) Django 패키지를 설치한다.

```
# apt-get install apache2 libapache2-mod-wsgi python-pip
# pip install django
```

- 2) 소스코드를 다운로드 받는다.

SourceForge(<https://sourceforge.net/projects/ktopenstack/>)에서 소스코드(osscloud.tar)를 다운로드 받아 임의의 위치에 디렉토리는 생성 후 압축을 해제한다.

```
#mkdir -p /openstack/web
#cd /openstack/web
#tar -xvf osscloud.tar
```

- 3) 소스코드의 장고 디렉토리(django_osscloud)안에 django.wsgi 파일을 작성한다.

```
import os
import sys
sys.path.append('/openstack/web/osscloud/src')
os.environ['DJANGO_SETTINGS_MODULE'] = 'django_osscloud.settings'

import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

- 4) Apache2설정을 위해 /etc/apache2/sites-available/django_osscloud 파일을 작성한다.

```
NameVirtualHost Django설치 IP
<VirtualHost Django설치 IP >
    ServerName Django설치 IP
    WSGIScriptAlias / /openstack/web/osscloud/src/django_osscloud/django.wsgi
</VirtualHost>
```

- 5) 설정한 VirtualHost를 적용시키고 Apache를 재가동 시킨다.

```
# a2ensite django_osscloud
# service apache2 restart
```

2.2 Cloud Management WEB UI 설정

Cloud Management WEB UI를 설정하기 위해서는 /openstack/web/osscloud/src/django_osscloud 아래에 있는 settings.py 파일을 열어 아래 가이드대로 해당 값들을 설정해야 한다.

DB는 postgresql, mysql, sqlite3, oracle 등 여러 가지를 사용할 수 있지만 제공하는 소스패키지는 mysql을 사용해서 만들어졌다. mysql을 설치하고 설정 값을 작성하며, Nova 설치 시 사용한 Database를 사용해야 한다.

```
DATABASES = {
    'DEFAULT': {
        'ENGINE': 'DJANGO.DB.BACKENDS.MYSQL', #DB 엔진타입 설정
        'NAME': 'nova', #DB DATABASE
        'USER': 'root', #접속 계정 아이디
        'PASSWORD': 'nova', #접속 계정 패스워드
        'HOST': 'xxx.xxx.xxx.xxx', #DB서버 호스트
        'PORT': '3306', #DB 서버 포트
    }
}
```

- 1) 로그 파일 경로를 설정한다.

로그 파일 위치는 실제 존재해야 하며 apache2의 실행계정(예:www-data) 소유의 권한으로 되어 있어야 한다. (예: chown -R www-data /openstack/logs/web)

```
LOG_ROOT = os.path.join('/', 'openstack', 'logs', 'web')
```

- 2) NOVA API값을 설정한다.

현재, nova-manage 를 통해 계정 생성시 access key와 secret key를 지정하지 않고 생성하게 되면 novarc의 값과 DB의 값이 불일치 되는 버그로 인해 UI접속이 되지 않는다. 따라서, 초기 계정 생성시 access key 와 secret key 를 지정해서 생성한다.

```
ex) nova-manage user admin cloudadm access secret (O)
ex) nova-manage user admin admin (X)
export EC2_ACCESS_KEY=access
export EC2_SECRET_KEY=secret
```

- 3) NOVA API 및 NOVA와 연동하기 위해 필요한 설정 값을 작성한다.(novarc값 참조)

```
NOVA_DEFAULT_ENDPOINT = 'http://xxx.xxx.xxx.xxx:8773/services/Cloud'
NOVA_DEFAULT_REGION = 'nova'
NOVA_ACCESS_KEY = 'access'
NOVA_SECRET_KEY = 'secret'
NOVA_API_KEY='cloudadm'
NOVA_USERNAME='cloudadm'
NOVA_PROJECT_ID='cloudprj'
```

```
NOVA_URL='http://xxx.xxx.xxx.xxx:8774/v1.1'
NOVA_MANAGE_PATH="/openstack/nova/bin/nova-manage "
NOVA_DB_IP="xxx.xxx.xxx.xxx"
```

4) Key Pair 저장 경로를 설정한다.

```
KEY_PAIR_PATH='/root/django_osscloud/key_pair/'
```

5) VPN 설정시 필요한 VPN 이미지 아이디값을 설정한다.

```
VPN_IMAGE_ID=12 # Image Service의 아이디 값
```

6) 모니터링 이미지 저장 경로 값을 설정한다.

```
MONITORING_IMAGE_PATH='/openstack/monitoring/path'
```

7) 메일서버를 설정한다.

계정 생성시 사용하게 될 메일서버를 설정하며 메일 서버가 없을 경우 구글에서 제공하는 메일 서버를 사용할 수 있다. 아래 샘플은 구글 메일 서버를 사용한 예이다.

```
EMAIL_HOST="smtp.gmail.com" #구글 메일 서버
EMAIL_PORT="587" #구글 메일서버 포트
EMAIL_HOST_USER = "sample@gmail.com" #구글 이메일 계정
EMAIL_HOST_PASSWORD = "sample" #구글 이메일 계정 비밀번호
```

8) 볼륨 목록의 갱신 주기를 설정 한다.

```
VOLUME_LOAD_INTERVAL=10*1000
```

9) 인스턴스 목록의 갱신 주기를 설정한다.

```
INSTANCE_LOAD_INTERVAL=10*1000
```

10) 회원 가입 약관 파일 경로 설정

```
AGREE_FILE_PATH= os.path.join(os.path.join(ROOT_PATH, 'agree.txt'))
```

11) settings.py의 설정이 완료 됐으면 마지막으로 DB sync를 한다.

```
#cd /openstack/web/osscloud/src/django_osscloud
#python manage.py syncdb
```

12) UI의 admin 계정을 등록합니다.

```
#cd /openstack/web/osscloud/src/django_osscloud
# python manage.py shell
>>> from django_osscloud.common.util import *
>>> from django_osscloud.osscloud_models.models import *
>>> password = encrypt('password')
```

```
>>> user = User(user_id='id', name='admin',password=password,
email='admin@admin.com', is_active=1, active_code=")
>>> user.save()
```

13) Nova-api가 위치한 서버의 nova.conf에 아래 값을 추가하고 nova를 재시작한다.

```
--allow_ec2_admin_api
--allow_admin_api
```

2.3 Instance Monitor 설치

Diablo 릴리즈부터 빠진 nova-instancemonitor 기능을 위해 Cactus버전의 nova-instancemonitor를 일부 보완하였으며 아래의 방법을 통해 설치한다.

1) Compute 노드에 nova-instancemonitor를 설치한다.

```
#cd /openstack/web/osscloud/src/monitor
#vi trans.sh (nova 설치 위치를 수정)
# sh trans.sh
```

2) compute 노드의 nova.conf에 아래 부분을 추가한다.

monitoring_rrd_path는 모니터링 데이터를 생성하기 위한 임시 디렉토리로 현재 서버에 실제 디렉토리가 존재하여야 동작한다. aws_access_key_id와 aws_secret_access_key는 novarc의 access_key와 secret_key값을 넣어준다.

```
--monitoring_instances_delay=60
--monitoring_instances_step=300
--monitoring_rrd_path=/openstack/monitor/instances
--aws_access_key_id="access"
--aws_secret_access_key="secret"
```

2.4 기타사항

1) Nova Databases 를 UTF-8로 생성해야 한다. (생성 방법은 DB 클라이언트 프로그램을 이용하면 쉽게 하실 수 있을 것입니다.)

```
#mysql -p
> CREATE DATABASE nova CHARACTER SET utf8;
```

2) Nova DB를 UTF-8로 지정하고 Table 생성시 한 테이블에 Index를 타는 column의 바이트 수가 1000바이트를 넘게 되면 오류가 발생한다. 이를 해결하기 위해 Nova 에서 제공하는 소스를 수정해야 한다.

nova/db/sqlalchemy/migrate_repo/versions/001_austin.py

위의 파일의 user_id, project_id, role 컬럼이 String 255로 되어 있는데 이것을 String 100으로 변경한다. 변경한 후에 nova-manage db sync 하면 db가 잘 만들어진다.

- 3) Web 화면에서 알림메세지를 구현하기 위해서는 아래와 같이 nova에서 제공하는 notifier를 수정하여 사용할 수 있다.(DB 접속하는 부분은 알맞게 수정)

```
#vi /openstack/nova/nova/notifier/no_op_notifier.py
import MySQLdb
def notify(message):
    """Notifies the recipient of the desired event given the model"""
    db = None
    cursor = None
    try:
        db = MySQLdb.Connect(db="",user="",passwd="", host="",charset='utf8', use_unicode=True)
        cursor = db.cursor()
        user_id=message['payload']['user_id']
        type=message['priority']
        content = message['event_type']
        data = (user_id,type,content)
        cursor.execute("""insert into osscloud_log (user_id,type,content,deleted,is_view,created_at)
                        value('%s','%s','%s',0,0,now())"" % data )
        db.commit()
    except:
        db.rollback()
    finally:
        if cursor:
            cursor.close()
        if db:
            db.close()
```

- 4) Web VNC를 적용하기

A. NoVNC 다운로드 한다. 다운받은후 noVNC를 /var/lib/nova/ 폴더 밑으로 이동한다

```
# apt-get install git
# git clone git://github.com/openstack/noVNC.git
# mv noVNC /var/lib/nova/
```

B. nova.conf의 설정 (nova-api 가 있는 host의 설정 변경)

```
--vnc_enabled=True
--vncproxy_host=121.166.195.59
--vncproxy_port=80
```

C. nova를 실행할 때 nova-vncproxy 도 같이 실행하여야 한다.