

# **教育部補助「5G 行動寬頻跨校教學聯盟計畫」**

**下世代 Network Slicing 模組設：  
開源軟體實現 NFV 以及虛擬資源分配**

**實驗單元：Mininet 搭配 Controller 使  
用 OpenFlow 模擬 SDN 網路環境**

授課教師：李宗南

教材編撰：陳 陞

## 目錄

一、	課程目標.....	3
二、	實驗設備.....	3
三、	軟體介紹.....	4
3.1、	OpenStack 介紹 .....	4
3.2、	OpenStack Tacker 介紹 .....	4
四、	安裝以及執行步驟教學.....	6
4.1、	安裝及執行 OpenStack 步驟.....	6
4.2、	安裝及執行 Tacker 步驟.....	9
五、	實驗要求.....	12
六、	參考資料.....	13

## 一、 課程目標

1. 課程目標 1：修課同學能夠實際了解如何建置雲端環境。
2. 課程目標 2：修課同學學習如何透過 OpenStack 實作雲端虛擬資源分配。
3. 課程目標 3：學習如何撰寫 config 檔並且透過 Tacker 來實現存取 OpenStack 虛擬資源以及實現 Network Function。

## 二、 實驗設備

硬體：

- 電腦：Centos7 作業系統
- 版本：(CentOS-7-x86\_64-DVD-1810)
- 需求：至少 12G 記憶體以及 64G 的儲存空間

軟體：

- OpenStack Pike
- Tacker
- gcc 編譯器
- Python 2.072
- vi/vim 文字編輯器

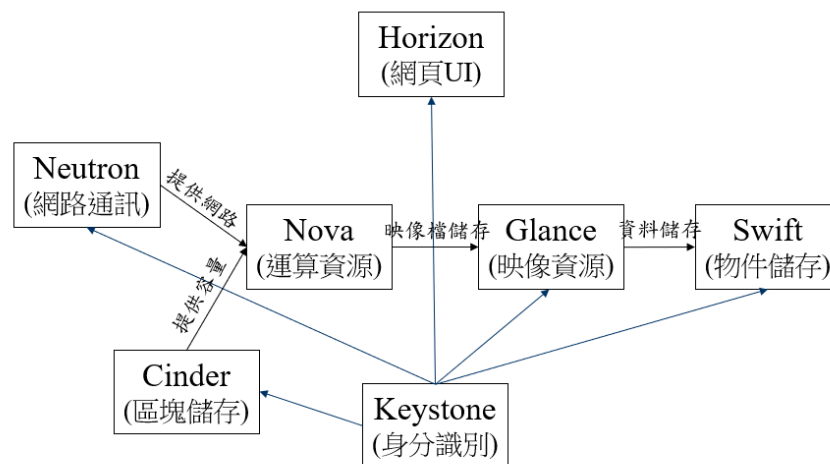
### 三、 軟體介紹

#### 3.1、 OpenStack 介紹

OpenStack 是美國航空暨太空總署和 Rackspace 合作研發的雲端運算軟體，並且以 Apache 授權條款授權，並且是一個 open source。

OpenStack 擁有三大模組運算模組(Nova)、網路通訊模組(Neutron)和儲存模組(Cinder)，加上一套集中式管理的儀表板模組，來組合成一套 OpenStack 共享服務提供虛擬機方式，對外帶來運算資源，以便利彈性擴充或調度。

OpenStack 架構：

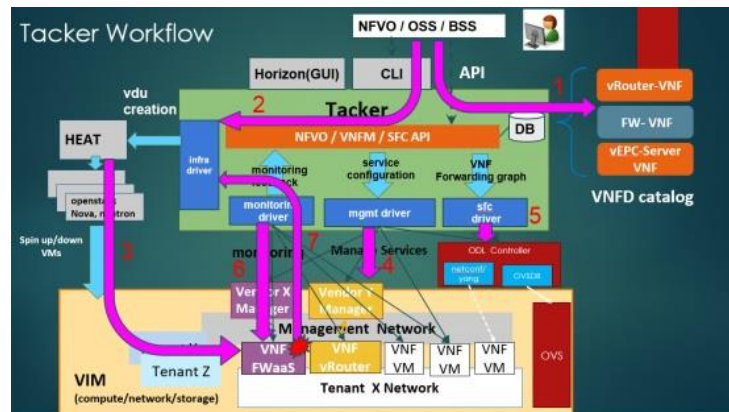


#### 3.2、 OpenStack Tacker 介紹

OpenStack Tacker 是一個在 OpenStack 內部孵化的項目，他的作用是 NVF 管理器，用於管理 NVF 的生命週期。Tacker 的重點是配置 VNF，並監視他們。如果需要，還可重啓和/或擴展（自動修復）NVF。整個進程貫穿

ETSIMANO 所描述的整個生命週期。

OpenStack Tacker 架構介紹：



第一步：Tacker 根據需求從服務目錄選出相應的服務項目，例如：vrouter

第二步：Tacker 把具體的 VNFD 推送給 OpenStack Heat 來生成 VDU  
(Virtual Deployment Unit，對應含 VNF 要求的 VM 部署單元)。

第三步：用 Heat 來啟動生成具體的 VM 實例，如圖下方的 VNF FWaaS，  
VNF vRouter 等。

第四步：透過 mgmt driver 來配置 vm。

第五步：實現 sfc，透過 odl 實現通過描述數據面的 Header 來沿着網絡服務  
路徑 (Service Path) 承載網絡服務信息，意在實現與傳輸獨立的“服務面”  
(Service Plane)。

第六、七步：監控 VNF 健康/可用性 availability 狀況，出現問題是自動治  
癒回覆 (重新生成 VNF，保證業務連續性)。

#### 四、 安裝以及執行步驟教學

##### 4.1、 安裝及執行 OpenStack 步驟

```
$ yum update -y && reboot
```

```
$ sudo yum install -y centos-release-openstack-pike
```

```
$ yum update -y
```

```
$ sudo yum install -y openstack-packstack
```

```
$ yum update -y
```

```
$ sudo packstack --gen-answer-file answer.txt
```

```
$ vim answer.txt(進入編輯 answer.txt)
```

編輯 answer.txt：

```
CONFIG_DEFAULT_PASSWORD= "PW" (PW 修改成你要設定的密碼。)
```

```
CONFIG_NTP_SERVERS=time.windows.com( 要改成離我們最近的 NTP->  
clock.stdtime.gov.tw)
```

```
CONFIG_CONTROLLER_HOST=Deploy IP(他自己會預設，不用理他。)
```

```
CONFIG_COMPUTE_HOSTS=Computes IP(他自己會預設，不用理他)
```

```
CONFIG_NETWORK_HOSTS=Network IP(他自己會預設，不用理他)
```

```
CONFIG_KEYSTONE_ADMIN_PW=admin 登入密碼
```

```
CONFIG_HEAT_INSTALL=y
```

```
CONFIG_PROVISION_DEMO=n
```

提醒：修改檔案的指令，英文大小寫有區別。

\$ sudo packstack --answer-file answer.txt(此行指令為 OpenStack 基本設定檔，如果失敗需要重新執行，執行時間較長。)

\$ nmcli c modify NIC ipv4.addresses 172.24.0.IP/19(NIC 為網卡名稱，可以用 ifconfig 查看，預設應該都是 ens33，後面的 IP 填自己虛擬機器的。)

\$ nmcli c modify NIC ipv4.gateway 172.24.31.254(此 IP 可以用 \$tracert 8.8.8.8 命令看到 gateway。)

\$ nmcli connection modify NIC ipv4.method manual

\$ nmcli connection modify NIC ipv4.dns 8.8.8.8

接下來修改網卡 scripts，需要修改 ifcfg-NIC 以及新建一個 br-ex。

以下為範例，固網以及區網設定略有不同，可以上網查看教學。

\$ cat /etc/sysconfig/network-scripts/ifcfg-NIC

DEVICE=eth0

TYPE=OVSPort

DEVICETYPE=ovs

OVS\_BRIDGE=br-ex

ONBOOT=yes

\$ vim ifcfg-br-ex

ONBOOT=yes

IPADDR=192.168.122.31

PREFIX=24

GATEWAY=192.168.122.1

DNS1=8.8.8.8

DEVICE=br-ex

DEVICETYPE=ovs

TYPE=OVSBridge

BOOTPROTO=static

\$ systemctl restart network(重新啟動網路，啟動剛剛修改的設定，如果上面網路設定有錯，重啟網路會有問題。)

Bridge 調整完畢後，我們可以下幾個指令檢查一下 eth0 是否存在。

\$ ovs-vsctl show

\$ ovs-vsctl list-ports br-ex

eth0

phy-br-ex

\$ systemctl stop NetworkManager; systemctl disable NetworkManager; systemctl restart network ; systemctl start NetworkManager; systemctl enable NetworkManager; systemctl restart network(重開網路)

\$ vim /etc/openstack-dashboard/local\_settings(編輯 local\_settings)

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True(#拿掉改為 True)

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = 'Default'(#拿掉改為 Default)

\$ systemctl restart httpd

安裝 centos 的 gnome，才會有圖示點選瀏覽器至登入介面：

\$ yum grouplist

\$ yum groups install "GNOME Desktop"

\$ systemctl start graphical.target

裝完之後就可以在瀏覽器上輸入自己的虛擬機 IP 後就會出現了。

實驗 1：

任務 1：將安裝成功的 openstack dashboard 登入成功畫面，結果以截圖方式顯示。



任務 2：使用 img 檔建立一個 instance 以及 instance 所需的一個網段，結果以截圖方式顯示。

任務 3：透過 ssh 連線進入 instance，透過 ping 指令以及 iperf 指令，測試 instance 能連通外網以及測速，結果以截圖方式顯示

## 4.2、 安裝及執行 Tacker 步驟

安裝及建立 tacker 數據庫

```
$ mysql -uroot -pcreate database
```

```
grant all privileges on tacker.* to 'tacker'@'%' identified by 'tackerdb';
```

```
grant all privileges on tacker.* to 'tacker'@'127.0.0.1' identified by 'tackerdb';flush
privileges;
```

創建用戶角色

```
$ source /root/keystone_admin
```

```
$openstack user create --domain default --password tacker tacker
```

```
$openstack role add --project services --user tacker admin
```

創建 service

```
openstack service create --name tacker \
```

```
--description "Tacker Project" nfvo-orchestration
```

創建 Endpoint

```
$ openstack endpoint create --region RegionOne nfvo-orchestration \      public
http://127.0.0.1:9890/
```

```
$ openstack endpoint create --region RegionOne nfvo-orchestration \      internal
http://127.0.0.1:9890/
```

```
$ openstack endpoint create --region RegionOne nfv-orchestration \      admin
http://127.0.0.1:9890/
```

安裝 Tacker 資源包

```
yum install -y openstack-tacker openstack-tacker-common \
puppet-tacker python-tacker python2-tackerclient
```

修改 Tacker.conf

```
[DEFAULT]
```

```
auth_strategy = keystone
```

```
policy_file = /etc/tacker/policy.json
```

```
debug = True
```

```
use_syslog = False
```

```
bind_host = 0.0.0.0
```

```
bind_port = 9890
```

```
service_plugins = nfvo,vnfm
```

```
state_path = /var/lib/tacker
```

```
[agent]
```

```
root_helper=sudo/usr/bin/tackerrootwrap/etc/tacker/rootwrap.conf
```

```
[database]
```

```
connection=mysql://tacker:tackerdb@127.0.0.1:3306/tacker?charset=utf8
```

```
[tacker]
```

```
monitor_driver = ping,http_ping
```

```
$/usr/bin/tacker-db-manage --config-file /etc/tacker/tacker.conf upgrade head
```

下載 Tacker horizon : <https://github.com/openstack/tacker-horizon/tree/stable/pike>

將檔案進行解壓縮： zip 檔名

```
$ cd /tacker-horizon-stable-pike
```

新建 PKG-INFO，內容為：

Metadata-Version: 0.8.0

Name: tacker-horizon

Version: 0.8.0

Summary: Tacker project for OpenStack

Home-page: <http://docs.openstack.org/developer/tacker/>

Author: OpenStack

Author-email: [openstack-dev@lists.openstack.org](mailto:openstack-dev@lists.openstack.org)

License: UNKNOWN

Description: =====

Platform: UNKNOWN

在/tacker-horizon-stable-pike 底下

```
$ python setup.py install
```

在 dashboard 啟動 Tacker

```
$ cp tacker_horizon/enabled/_80_nfv.py /usr/share/openstack-  
dashboard/openstack_dashboard/enabled/
```

重啟介面

```
$ systemctl restart httpd
```

重啟 Tacker

```
$ systemctl restart openstack-tacker-server.service
```

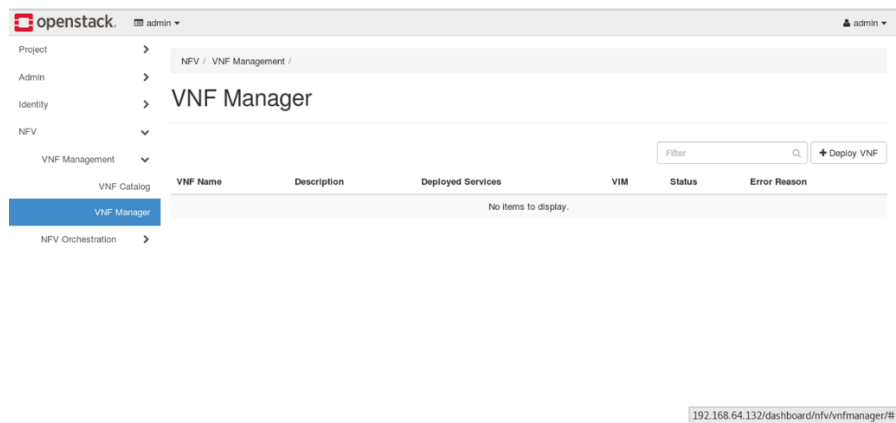
```
$ systemctl enable openstack-tacker-server.service
```

安裝完成之後，需要設置權限：

```
mkdir -p /etc/tacker/vim/fernet_keys
```

```
chown tacker:tacker /etc/tacker/* -R
```

登陸 OpenStack，安裝成功 dashboard 會增加 NFV 管理功能。



實驗 2：

任務 1：透過 dashboard 建立一個 VIM 以及 VNFD，結果以截圖方式呈現。

任務 2：撰寫 config.yaml 檔案建立 VIM，檔案撰寫可以參考官方文件說明，結果以截圖方式呈現。

## 五、實驗要求

實驗 1：必須能夠成功依照教學文檔安裝 Openstack，並且將安裝成功截圖繳交，需針對 Openstack 內部元件相關使用有一定了解，分別將建立 instance、創立網段、上傳 img 檔等完成成功截圖繳交。

1. 安裝 Openstack
2. 建立網段

3. 上傳 img 檔
4. 創立 instance

實驗 2：必須在實驗 1 完成之後，透過修改設定檔完成 Tacker 安裝，並且將安裝成功截圖繳交，並且利用 Tacker 建立 vim 以及學習透過 config 檔呼叫 Tacker 向 Openstack 取得資源，並將實驗結果截圖繳交，並且附上 config 檔。

1. 安裝 Tacker
2. 透過 dashboard 成功創立 vim
3. 透過撰寫 config 檔成功創立 vim

## 六、 參考資料

- [1]<https://zh.wikipedia.org/wiki/OpenStack>
- [2]<https://blog.csdn.net/linshenyuan1213/article/details/78120623>
- [3]<https://blog.csdn.net/chenhaifeng2016/article/details/78924857>
- [4]<https://blog.csdn.net/linshenyuan1213/article/details/78193886>
- [5]<https://www.twblogs.net/a/5c0a2b55bd9eee6fb213148a>