

Laporan Praktikum Teknologi Cloud Computing - Minggu 4

Infrastructure as a Service dan SDN



Oleh :

Nama : Jauhari Ahmad
No. Mhs : 205411167
Jurusan : Teknik Informatika / S1

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AKAKOM
YOGYAKARTA
2020

PENDAHULUAN

A. TUJUAN

1. Mahasiswa memahami pengertian serta komponen-komponen dari IaaS.
2. Mahasiswa memahami dan mampu melakukan instalasi DevStack sebagai software yang menyediakan IaaS serta memahami arah dari pembelajaran IaaS menggunakan OpenStack / DevStack.
3. Mahasiswa memahami pengertian serta komponen-komponen dari SDN (Software-Defined Networking).
4. Mahasiswa memahami dasar-dasar cara kerja dari SDN menggunakan simulator Mininet.
5. Mahasiswa memahami keterkaitan antara IaaS dengan SDN.

B. DASAR TEORI

1. [Artikel dari IBM untuk memahami IaaS](#).
2. [Dokumentasi DevStack](#).
3. [Memahami SDN](#).
4. [The OpenDaylight Foundation](#), lihat juga di [overview](#).
5. [Mininet](#), lihat juga [dokumentasi Mininet](#).
6. [Keterkaitan IaaS dengan SDN - menggunakan SDN untuk optimasi IaaS](#).

PEMBAHASAN

A. LATIHAN

Install DevStack

1. Membuat user Stack pada Linux

Membuat user Stack dengan nama “stack”

```
jauhmad@jauhmad-VirtualBox:~$ sudo useradd -s /bin/bash -d /opt/stack -m stack
[sudo] password for jauhmad:
jauhmad@jauhmad-VirtualBox:~$
```

Perizinan sudo:

```
jauhmad@jauhmad-VirtualBox:~$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
[sudo] password for jauhmad:
stack ALL=(ALL) NOPASSWD: ALL
jauhmad@jauhmad-VirtualBox:~$ sudo su - stack
```

2. Download DevStack

Git clone

```
stack@jauhmad-VirtualBox:~$ sudo git clone https://opendev.org/openstack/devstack
Cloning into 'devstack'...
remote: Enumerating objects: 46212, done.
remote: Counting objects: 100% (46212/46212), done.
remote: Compressing objects: 100% (21142/21142), done.
remote: Total 46212 (delta 32661), reused 37469 (delta 24365)
Receiving objects: 100% (46212/46212), 9.52 MiB | 62.00 KiB/s, done.
Resolving deltas: 100% (32661/32661), done.
Checking connectivity... done.
```

3. Membuat local.conf

Nano local.conf

```
GNU nano 2.5.3

[[local|localrc]]
ADMIN_PASSWORD=123456
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

4. Mulai install

```
stack@jahmad-VirtualBox:~/devstack$ .stack.sh
+ unset GREP_OPTIONS
+ unset LANG
+ unset LANGUAGE
+ LC_ALL=en_US.utf8
+ export LC_ALL
++ cut -d = -f 1
++ grep -E '^OS_'
++ env
+ unset
+ unask 022
+ PATH=/opt/stack/bin:/opt/stack/./local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/sbin:/usr/sbin:/sbin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/opt/stack/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%s
+ DEVSTACK_START_TIME=1601552795
+ [[ -r /opt/stack/devstack/.stackenv ]]
+ FILES=/opt/stack/devstack/files
+ '[' '' '!' -d /opt/stack/devstack/files ']'
+ '[' '' '!' -d /opt/stack/devstack/lib ']'
+ '[' '' '!' -d /opt/stack/devstack/lib ']'
+ '[' '' == V ']'
+ [[ 1002 -eq 0 ]]
+ [[ -n '' ]]
+ [[ -e /opt/stack/.no-devstack ]]
+ LAST_SPINNER_PID=
+ source /opt/stack/devstack/functions
++ [[ -z '' ]]
++ declare -r -g DEVSTACK_FUNCTIONS=1
++++ dirname /opt/stack/devstack/functions
+++ cd /opt/stack/devstack
+++ pwd
++ FUNC_DIR=/opt/stack/devstack
++ source /opt/stack/devstack/functions-common
++++ grep Xtrace
++++ set +o
```

5.

B. TUGAS DAN PENYELESAIAN

Kerjakan [walkthrough](#) untuk Mininet, setelah itu semaksimal mungkin untuk bisa memahami SDN, kerjakan [tutorial OpenFlow](#). Lihat di sisi sebelah kanan untuk urut-urutannya.

B.1. Mininet

Native Installation from Source

- ## 1. Git clone repository mininet

```
jauhmad@jauhmad-VirtualBox:~$ git clone git://github.com/mininet/mininet
Cloning into 'mininet'...
remote: Enumerating objects: 9752, done.
remote: Total 9752 (delta 0), reused 0 (delta 0), pack-reused 9752
Receiving objects: 100% (9752/9752), 3.03 MiB | 1.66 MiB/s, done.
Resolving deltas: 100% (6472/6472), done.
Checking connectivity... done.
```

- ## 2. Install

```
jauhmad@jauhmad-VirtualBox:~$ sudo mininet/util/install.sh -a
Detected Linux distribution: Ubuntu 16.04 xenial amd64
sys.version info(major=2, minor=7, micro=12, releaselevel='final', serial=0)
Detected Python (python) version 2
Installing all packages except for -eix (doxypy, ivs, nox-classic)...
Install Mininet-compatible kernel if necessary
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:2 http://id.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://id.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [93,1 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [128 kB]
Get:7 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2.464 B]
Get:8 http://id.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [326 kB]
Get:9 http://id.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [279 kB]
Get:10 http://id.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11 Metadata [5.956 B]
Get:11 http://id.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Metadata [3.332 B]
```

Miniet terinstall di /home/jauhmad/mininet

- ### 3. Menjalankan Mininet

```
jauhmad@jauhmad-VirtualBox:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Walkthrough Mininet

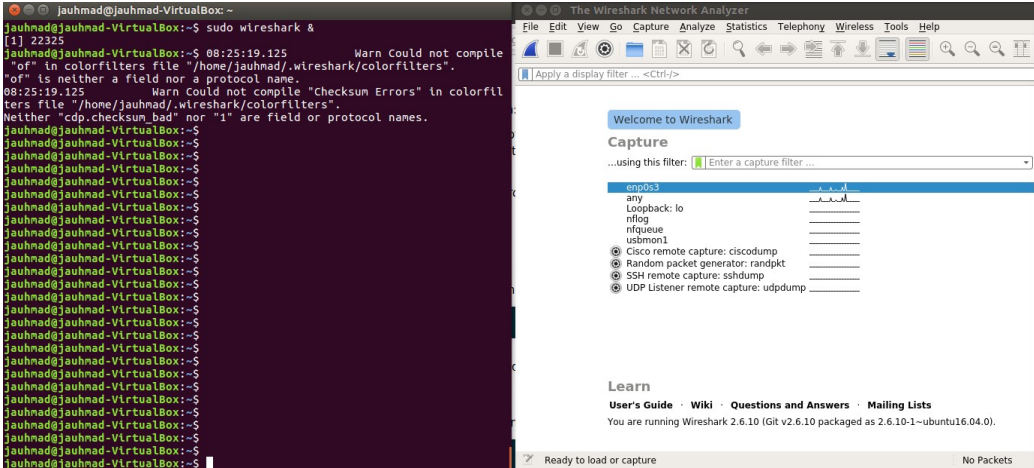
Display Startup Options

```
jauhmad@jauhmad-VirtualBox:~$ sudo mn -h
Usage: mn [options]
(type mn -h for details)

The mn utility creates Mininet network from the command line. It can create
parametrized topologies, invoke the Mininet CLI, and run tests.

Options:
  -h, --help                show this help message and exit
  --switch=SWITCH            default|ivs|lxb|ovs|ovsbr|ovsk|user[,param=value...]
                             ovs=OVSSwitch default=OVSSwitch ovsk=OVSSwitch
                             lxb=LinuxBridge user=UserSwitch ivs=IVSSwitch
                             ovsbr=OVSBridge
                             cfl=ovsctl[,param=value...]
  -b HOST                    host=HOST
```

Start Wireshark



In the Wireshark filter box, enter this filter, then click Apply:

KESMIPULAN

1. OpenStack dibangun dari seperangkat layanan microservice yang dapat digabungkan ke dalam berbagai pengaturan berdasarkan kebutuhan aktual. Layanan menyediakan REST API untuk penggunaannya. Ini dapat berupa operator cloud atau layanan lainnya. Untuk mempermudah penggunaan API, dimungkinkan untuk menggunakan Software Development Kits (SDK) yang juga dikembangkan sebagai proyek di dalam komunitas OpenStack.
 2. Mininet menciptakan jaringan virtual dengan memanfaatkan software real komponen jaringan berdasarkan kernel linux sehingga dapat digunakan dalam pengujian jaringan SDN.
-