Tap (v1.0)

엔터프라이즈 시스템/네트워크 운영자 대상
(for IT Pros and System Administrators)

JS Lab

안종석
james@jslab.kr

2018년 12월

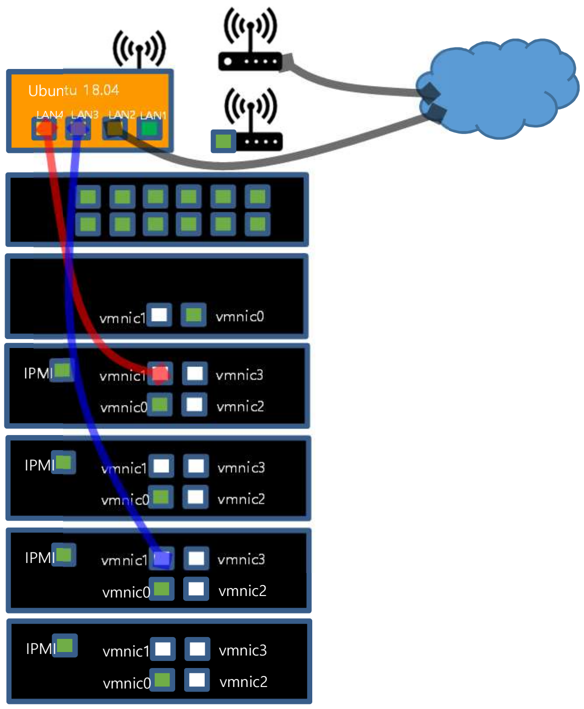
JS Lab

JS Lab

0. 환경

❖ 개요

1. **네트워크 감시를 위해 JS Lab 시험 적용 중**
2. **인라인(In-line) 적용, 미러 포트, 관리 유선/무선** (WiFi)
3. **Tap Hardware Appliance: Whitebox / 베어메탈**
4. **오픈소스 사용 (ubuntu or fedora or CentOS)**
5. **Production 용은 시스템 폴더의 RO(Read Only) 설정, 하드웨어의 Bypass 지원으로 안정성 강화 가능**
6. **하드웨어 성능 강화로 IDS등 기능 추가 가능**



메모:

❖ **범용 하드웨어를 이용한 Tap :** 상용과 동일한 가격 수준에서 성능 추가 가능하여 Tap 내에 추가 분석 기능을 같이 넣을 수 있으며, SDN 지원 OVS에 SDN 컨트롤러를 연결하여 L2/L3 매핑 정보에 대한 신뢰성을 기반으로 필요한 분석 기능을 추가 할 수 있다.

0. 환경

❖ Tap 하드웨어 구성

1. **Type 1** (100만원 이상)

1. CPU w/Passive CPU heat sink
   - Intel® Xeon® processor D-1528
   - FCBGA 1667
   - CPU TDP support 35W, 9MB, 6 Cores, 12 Threads, 1.9-2.2GHz
1. RAM (16GB /Max 128 GB)
2. IPMI 2.0
3. 10GbE 2포트, 1 GbE LAN 2포트, IPMI 2.0 전용 LAN
4. SR-IOV (Single-Root Virtualization)

2. **Type 2** (20만원 이상)

1. CPU: J1900 (Intel Celeron 4 Core, 2M Cache, 2.0GHz)
2. RAM: 4 GB (Max 8GB)
3. SSD: 32 GB
4. OS: Ubuntu 17.01
5. 이더넷: 1 GbE 4 포트
6. 무선랜 (선택, 관리용으로 사용하며 접속이 불안 할 수 있음)
7. 내장 가상 스위치 3 포트: LAN 2, LAN 3, LAN 4 (미러포트)

3. **Type 3** (약 20만원)

1. CPU: Intel NM70 2 Core, 2M Cache, 1.8 GHz)
2. RAM: 4 GB (Max 8GB)
3. SSD:
4. OS:
5. 이더넷: 1 GbE 6 포트 (Bypass 지원)
6. 무선랜 (선택)

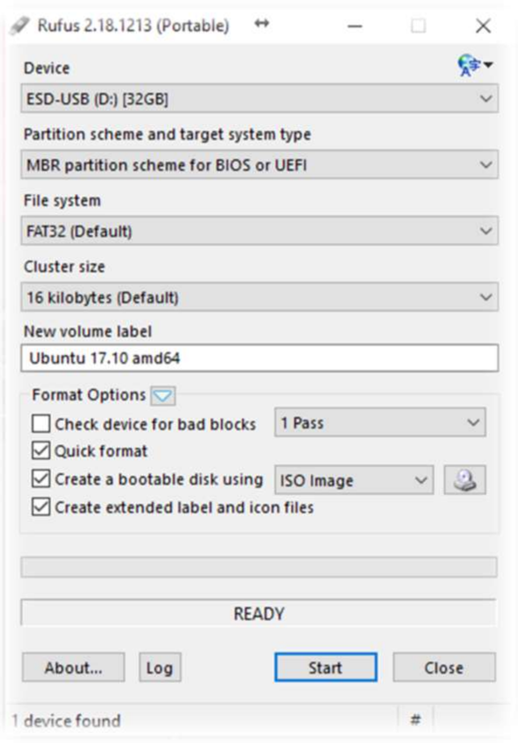
메모:

JS Lab

1. Ubuntu 도구 설치

❖ **Ubuntu 17.10** (Ubuntu 17.10)

- **Ubuntu 부팅 USB 메모리 준비**



Rufus 2.18.1213 (Portable)

Device: ESD-USB (D:) [32GB]

Partition scheme and target system type: MBR partition scheme for BIOS or UEFI

File system: FAT32 (Default)

Cluster size: 16 kilobytes (Default)

New volume label: Ubuntu 17.10 amd64

Format Options

- [ ] Check device for bad blocks — 1 Pass
- [x] Quick format
- [x] Create a bootable disk using — ISO Image
- [x] Create extended label and icon files

READY

1 device found

- **sudo apt-get update**
- **sudo apt install net-tools (ifconfig)**

메모:

1. Ubuntu 도구 설치

❖ **SSH 서버, netdata, ntopng, net-tools** (Ubuntu 18.04)

- **SSH server**
  - ✓ **sudo apt-get update**
  - ✓ **sudo apt install openssh-server**
  - ✓ **sudo sshd**
- **netdata**
  - ✓ **sudo apt install curl**
  - ✓ **bash <(curl -Ss https://my-netdata.io/kickstart.sh)**
  - ✓ **http://127.0.0.1:19999/**
- **ntopng**
  - ✓ **sudo apt install ntopng**
  - ✓ **sudo systemctl enable ntopng**
  - ✓ **sudo ntopng (sudo systemctl start ntopng)**
  - ✓ **http://127.0.0.1:3000/ (admn/admin)**
- **Net tools for 'ifconfig'**
  - ✓ **sudo apt install net-tools**

메모:

❖ systemctl stop netdata
❖ systemctl start netdata
❖ 팬리스(Fanless) 하드웨어를 위한 센서 드라이버 설치: sudo apt install lm-sensors (sensors)

1. Ubuntu 도구 설치

❖ **Static IP for WiFi** (Ubuntu 18.04)

- **WiFi 설정**

1. **ip link show**

```
james@ubuntu18:/etc/netplan$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP mode DEFAULT group default qlen 1000
    link/ether 00:aa:2a:e8:34:21 brd ff:ff:ff:ff:ff:ff
4: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP mode DEFAULT group default qlen 1000
    link/ether 00:aa:2a:e8:34:22 brd ff:ff:ff:ff:ff:ff
5: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:aa:2a:e8:34:23 brd ff:ff:ff:ff:ff:ff
7: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 96:be:89:0f:df:b5 brd ff:ff:ff:ff:ff:ff
8: ovs1qotom: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:aa:2a:e8:34:20 brd ff:ff:ff:ff:ff:ff
9: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:ee:0f:69:c6 brd ff:ff:ff:ff:ff:ff
10: wlx742f68923076: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 74:2f:68:92:30:76 brd ff:ff:ff:ff:ff:ff
12: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel master ovs-system state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:aa:2a:e8:34:20 brd ff:ff:ff:ff:ff:ff
james@ubuntu18:/etc/netplan$
```

2. **cd /etc/netplan**
3. **sudo nano 01-network-manager-all.yaml**

```
network:
  version: 2
  renderer: networkd
  wifis:
      wlx742f68923076:
            dhcp4: no
            dhcp6: no
            addresses: [192.168.0.18/24, ]
            gateway4: 192.168.0.1
            nameservers:
                search: [vsphere.local]
                addresses: [192.168.0.1, 8.8.8.8]
            access-points:
                Tech-Support:
                   password: 12345*****
```

4. **sudo netplan generate**
5. **sudo netplan apply**

메모:

❖ https://www.tecmint.com/configure-network-static-ip-address-in-ubuntu/

1. Ubuntu 도구 설치

❖ **OVS (Open vSwitch)** (Ubuntu 18.04)

- **OVS (Open vSwitch) Mirroring (2.8.0)**

1. **sudo apt-get install openvswitch-switch**
2. **sudo apt-get install openvswitch-common bridge-utils**

3. **sudo ovs-vsctl show**
4. **sudo ovs-vsctl add-br ovsbr0**
5. **sudo ovs-vsctl add-port ovsbr0 enp2s0**
6. **sudo ovs-vsctl add-port ovsbr0 enp3s0**
7. **sudo ovs-vsctl add-port ovsbr0 enp4s0** # Optional for tap monitoring

8. **sudo ovs-vsctl add-port ovsbr0 enp4s0 ₩**
-- --id=@p get port enp4s0 ₩
-- --id=@m create mirror name=m0 select-all=true output-port=@p ₩
-- set bridge ovsbr0 mirrors=@m

# example: sudo ovs-vsctl add-port ovsbr0 enp4s0 -- --id=@p get port enp4s0 -- --id=@m create mirror name=m0 select-all=true output-port=@p -- set bridge ovsbr0 mirrors=@m

9. **sudo ovs-vsctl clear bridge ovsbr0 mirrors** # To later disable mirroring

10. **docker run -d --name onos -p 8181:8181 -p 6653:6653 onosproject/onos**
11. **ovs-vsctl set-controller ovsbr0 tcp:172.17.0.2:6653**

https://stackoverflow.com/questions/29996213/openvswitch-mirroring-only-layer2-traffic

메모:

❖ Host Rebooting후 정상 동작 확인

❖ 강제 전원 종료는 복구 불능 Panic 발생 가능

❖ 미러포트 설정:

- sudo ovs-vsctl add-port ovsbr0 enp4s0 -- --id=@p get port enp4s0 -- --id=@m create mirror name=m0 select-all=true output-port=@p-- set bridge ovsbr0 mirrors=@m

1. Ubuntu 도구 설치

❖ **OVS (Open vSwitch)** (Ubuntu 18.04)

● **OVS (Open vSwitch) Mirroring for Virtual Port (2.8.0)**

1. sudo apt-get install openvswitch-switch
2. sudo apt-get install openvswitch-common bridge-utils

3. sudo ovs-vsctl show
4. ovs-vsctl add-br ovsbr0
5. ovs-vsctl add-port ovsbr0 enp2s0
6. ovs-vsctl add-port ovsbr0 enp3s0
7. ovs-vsctl add-port ovsbr0 enp4s0 # Optional for tap monitoring

8. **sudo ip tuntap add mode tap tap**
9. **sudo ip link set tap up**
10. **sudo ovs-vsctl add-port ovsbr0 tap**

11. **sudo ovs-vsctl show**

http://abregman.com/2016/10/18/open-vswitch-introduction-part-1/

메모:

❖ 미러포트 설정:

① sudo ovs-vsctl add-port ovsbr0 enp4s0 -- --id=@p get port enp4s0 -- --id=@m create mirror name=m0 select-all=true output-port=@p-- set bridge ovsbr0 mirrors=@m

1. Ubuntu 도구 설치

❖ **Docker/Swarm 설치** (Ubuntu 18.04)

1. **sudo apt-get update**
2. **sudo apt install curl**
3. **sudo curl -fsSL https://get.docker.com/ | sh**
4. **sudo usermod -aG docker** james
5. **sudo systemctl enable docker   ### Set Docker to auto-start**
6. **sudo systemctl start docker**
7. **sudo docker swarm init --listen-addr 192.168.0.xx**
8. sudo docker service create ₩   # ONOS Service Install
9. --name onos ₩
10. --publish 8383:8181/tcp ₩
11. --publish 6653:6653/tcp ₩
12. --constraint node.role==manager ₩
13. --mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock ₩
14. onosproject/onos:latest
15. **sudo docker service create **   # Visualizer Service Install
16. **--name viz **
17. **--publish 8282:8080/tcp **
18. **--constraint node.role==manager **
19. **--mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock **
20. **alexellis2/visualizer-arm:latest**

메모:

❖ sudo docker run -d --name onos -p 8181:8181 -p 6653:6653 onosproject/onos

❖ http://127.0.0.1:8181/onos/ui (ID/Password: karaf/karaf)

JS Lab

2. ONOS-OVS 연결

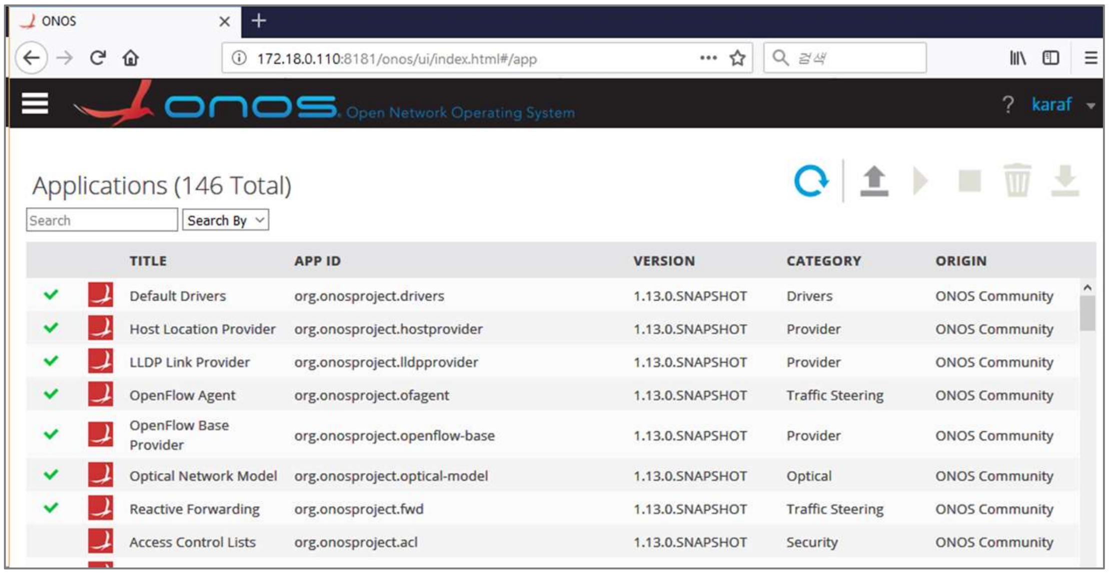
❖ ONOS / OVS 실행

- **ONOS 실행과 OVS 접속**

1. sudo docker run -d --name onos -p 8181:8181 -p 6653:6653 onosproject/onos
2. Ifconfig
3. sudo network docker inspect bridge
4. sudo docker service ls
5. ovs-vsctl set-controller ovsbr0 tcp:172.17.0.2:6653

- **ONOS 컨테이너 실행 호스트 IP주소로 외부 접속 가능**

① http://172.18.0.110:8181/onos/ui (ID/Password:


ONOS
172.18.0.110:8181/onos/ui/index.html#/app
검색
onos Open Network Operating System
karaf
Search
Search By

Applications (146 Total)

		TITLE	APP ID	VERSION	CATEGORY	ORIGIN
✓		Default Drivers	org.onosproject.drivers	1.13.0.SNAPSHOT	Drivers	ONOS Community
✓		Host Location Provider	org.onosproject.hostprovider	1.13.0.SNAPSHOT	Provider	ONOS Community
✓		LLDP Link Provider	org.onosproject.lldpprovider	1.13.0.SNAPSHOT	Provider	ONOS Community
✓		OpenFlow Agent	org.onosproject.ofagent	1.13.0.SNAPSHOT	Traffic Steering	ONOS Community
✓		OpenFlow Base Provider	org.onosproject.openflow-base	1.13.0.SNAPSHOT	Provider	ONOS Community
✓		Optical Network Model	org.onosproject.optical-model	1.13.0.SNAPSHOT	Optical	ONOS Community
✓		Reactive Forwarding	org.onosproject.fwd	1.13.0.SNAPSHOT	Traffic Steering	ONOS Community
		Access Control Lists	org.onosproject.acl	1.13.0.SNAPSHOT	Security	ONOS Community

메모:

- ❖ ONOS 다운로드 주소: https://wiki.onosproject.org/display/ONOS/Downloads
- ❖ ONOS Applications: Default Drivers, Host Location Provider, OpenFlow Agent, OpenFlow Base Provider, Reactive Forwarding, LLDP Link Provider

2. ONOS-OVS 연결

❖ OVS (Open vSwitch) DPDK Mirroring

- OVS (Open vSwitch) DPDK Mirroring (2.9.0)

1. ovs-vsctl add-br ovsbr0
2. ovs-vsctl add-port ovsbr0 myportname -- set Interface myportname ₩
3. type=dpdk options:dpdk-devargs=0000:06:00.0

   # configure a DPDK port as an access port
4. ovs-vsctl add-br ovsbr0
5. ovs-vsctl add-port ovsbr0 eth0
6. ovs-vsctl add-port ovsbr0 tap0 tag=10
7. ovs-vsctl ₩

-- --id=@m create mirror name=m0 select-all=true select-vlan=10 ₩
output-vlan=15 ₩
-- set bridge br0 mirrors=@m # a VLAN as an RSPAN VLAN

4. $ ovs-vsctl clear bridge br0 mirrors # To later disable mirroring

메모:

2. ONOS-OVS 연결

❖ OVS (Open vSwitch) Mirroring for GRE Tunnel

- **OVS (Open vSwitch) Mirroring (2.9.0) for GRE tunnel**

```
1. ovs-vsctl add-br br0
2. ovs-vsctl add-port br0 eth0
3. ovs-vsctl add-port br0 tap0
4. ovs-vsctl add-port br0 gre0 ₩
-- set interface gre0 type=gre options:remote_ip=192.168.1.10 ₩
-- --id=@p get port gre0 ₩
-- --id=@m create mirror name=m0 select-all=true output-port=@p ₩
-- set bridge br0 mirrors=@m  # Optional(an already added port as an access port)

5. $ ovs-vsctl clear bridge br0 mirrors # To later disable mirroring
6. $ ovs-vsctl del-port br0 gre0 # To later disable mirroring

7. ovs-vsctl add-br br0
8. ovs-vsctl add-port br0 eth0
9. ovs-vsctl add-port br0 tap0
10. ovs-vsctl add-br br1
11. ovs-vsctl add-port br1 tap1
12. ovs-vsctl ₩
-- add-port br0 patch0 ₩
-- set interface patch0 type=patch options:peer=patch1 ₩
-- add-port br1 patch1 ₩
-- set interface patch1 type=patch options:peer=patch0
                                                    # connect two bridges
```

메모:

- ❖ http://docs.openvswitch.org/en/latest/faq/configuration/
- ❖ 예: sudo ovs-vsctl add-port ovsbr0 enp4s0 -- --id=@p get port enp4s0 -- --id=@m create mirror name=m0 select-all=true output-port=@p -- set bridge ovsbr0 mirrors=@m

JS Lab

❖ 3. 네트워크 도구 (Network Tools)

❖ WireShark 설치

- WireShark 설치

✓ sudo dpkg --configure -a

✓ sudo apt install wireshark-qt

메모:

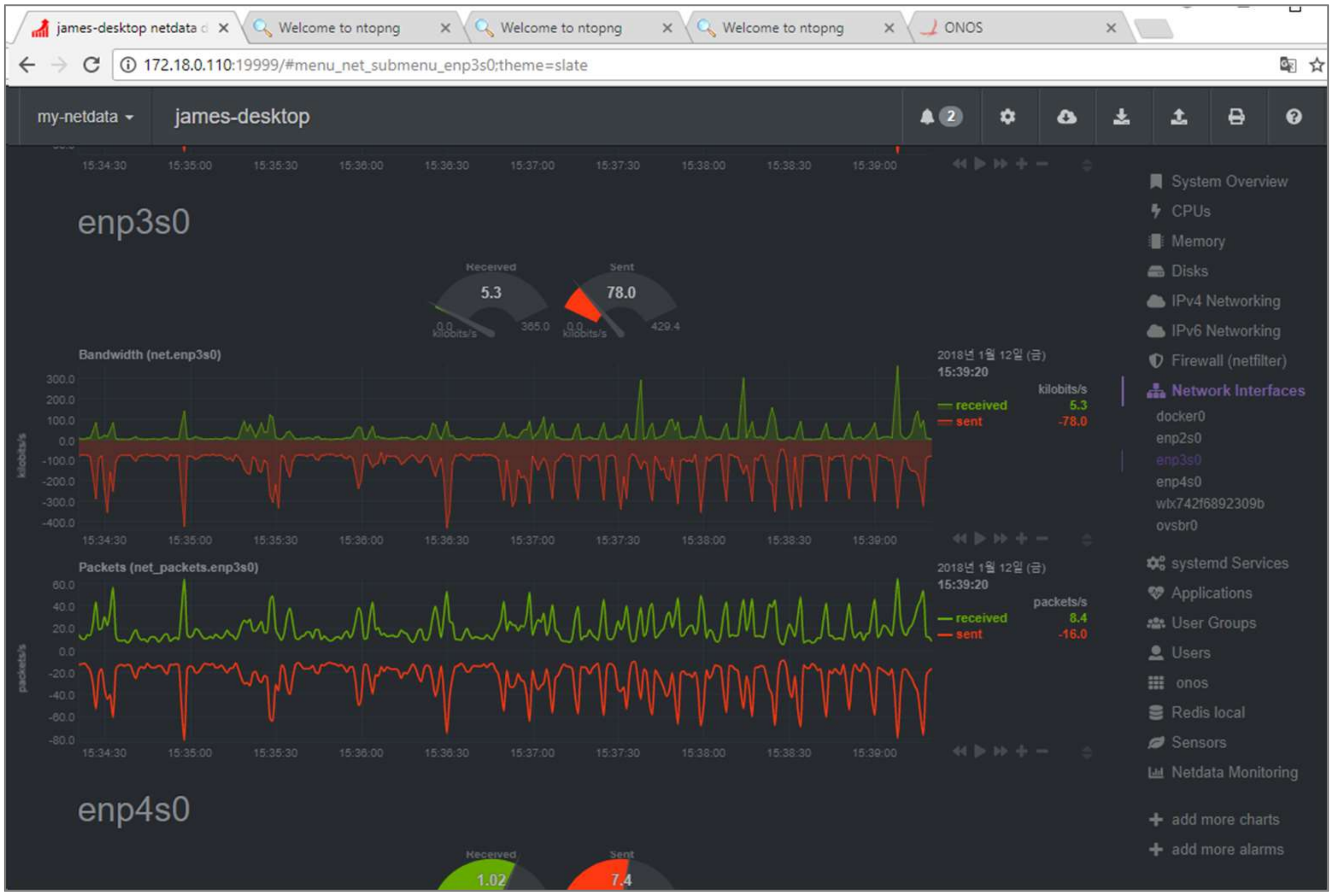
❖ X11 forwarding을 사용하는 SSH 원격 접속 시 권장하지 않음

❖ 소프트웨어 메뉴에서 WireShark (GTK+) 설치 가능

3. 네트워크 도구 (Network Tools)

❖ netdata 설치

1. bash <(curl -Ss https://my-netdata.io/kickstart.sh)
2. http://127.0.0.1:19999/



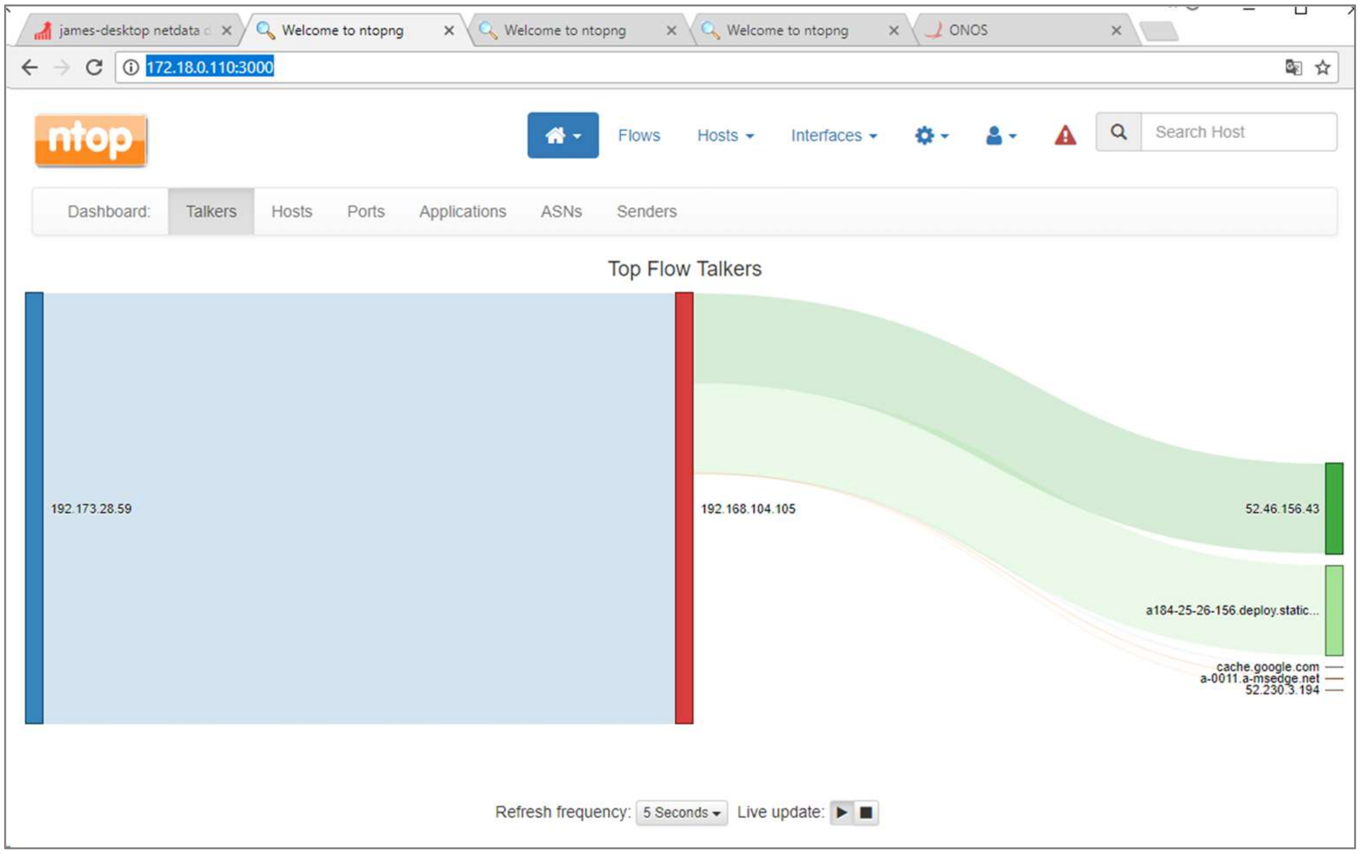
메모:

3. 네트워크 도구 (Network Tools)

❖ **ntopng**

❖ **대쉬보드**

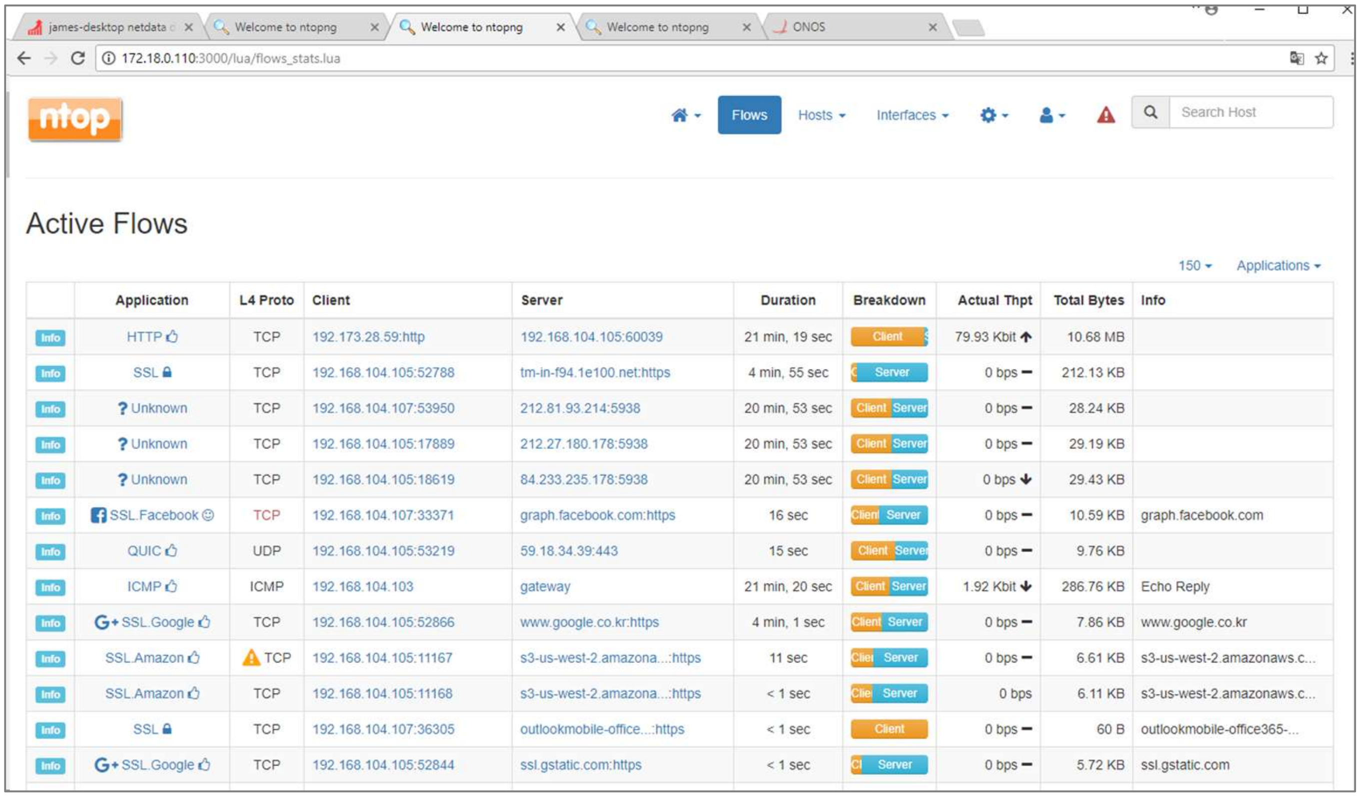
❖ **http://172.18.0.110:3000 (외부접속 IP가 172.18.0.110인 경우)**



메모:

3. 네트워크 도구 (Network Tools)

- ntopng
  - Active Flows
  - http://172.18.0.110:3000



Active Flows

150 ▾ Applications ▾

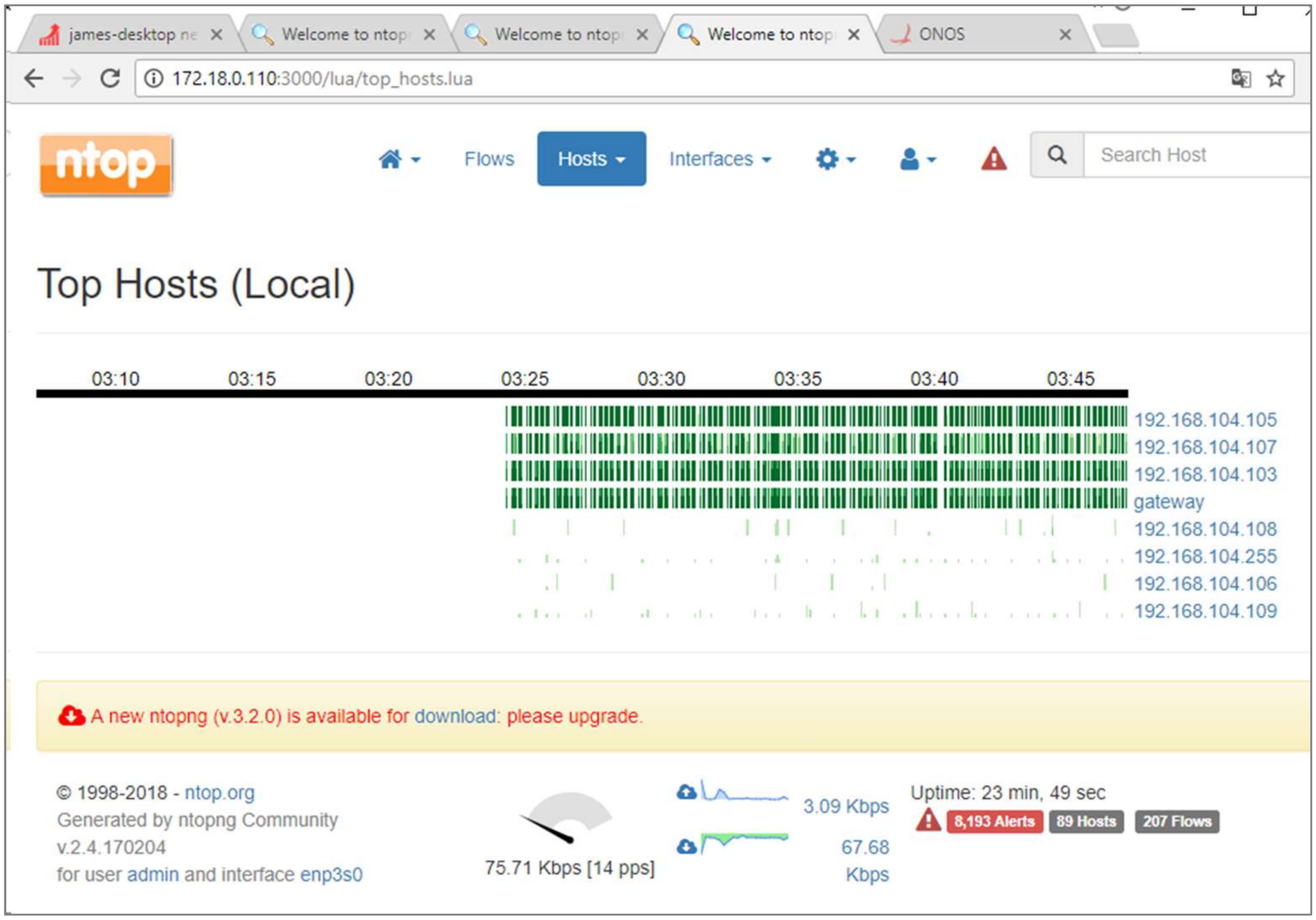
	Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
Info	HTTP	TCP	192.173.28.59:http	192.168.104.105:60039	21 min, 19 sec	Client	79.93 Kbit ↑	10.68 MB	
Info	SSL	TCP	192.168.104.105:52788	tm-in-f94.1e100.net:https	4 min, 55 sec	Server	0 bps −	212.13 KB	
Info	? Unknown	TCP	192.168.104.107:53950	212.81.93.214:5938	20 min, 53 sec	Client Server	0 bps −	28.24 KB	
Info	? Unknown	TCP	192.168.104.105:17889	212.27.180.178:5938	20 min, 53 sec	Client Server	0 bps −	29.19 KB	
Info	? Unknown	TCP	192.168.104.105:18619	84.233.235.178:5938	20 min, 53 sec	Client Server	0 bps ↓	29.43 KB	
Info	SSL.Facebook	TCP	192.168.104.107:33371	graph.facebook.com:https	16 sec	Client Server	0 bps −	10.59 KB	graph.facebook.com
Info	QUIC	UDP	192.168.104.105:53219	59.18.34.39:443	15 sec	Client Server	0 bps −	9.76 KB	
Info	ICMP	ICMP	192.168.104.103	gateway	21 min, 20 sec	Client Server	1.92 Kbit ↓	286.76 KB	Echo Reply
Info	G+ SSL.Google	TCP	192.168.104.105:52866	www.google.co.kr:https	4 min, 1 sec	Client Server	0 bps −	7.86 KB	www.google.co.kr
Info	SSL.Amazon	⚠ TCP	192.168.104.105:11167	s3-us-west-2.amazona...:https	11 sec	Clie Server	0 bps −	6.61 KB	s3-us-west-2.amazonaws.c...
Info	SSL.Amazon	TCP	192.168.104.105:11168	s3-us-west-2.amazona...:https	< 1 sec	Clie Server	0 bps	6.11 KB	s3-us-west-2.amazonaws.c...
Info	SSL	TCP	192.168.104.107:36305	outlookmobile-office...:https	< 1 sec	Client	0 bps −	60 B	outlookmobile-office365-...
Info	G+ SSL.Google	TCP	192.168.104.105:52844	ssl.gstatic.com:https	< 1 sec	Cl Server	0 bps −	5.72 KB	ssl.gstatic.com

메모:

3. 네트워크 도구 (Network Tools)

❖ **Top Hosts**

❖ **Active Flows**
❖ http://172.18.0.110:3000



Top Hosts (Local)

A new ntopng (v.3.2.0) is available for download: please upgrade.

© 1998-2018 - ntop.org
Generated by ntopng Community
v.2.4.170204
for user admin and interface enp3s0

Uptime: 23 min, 49 sec

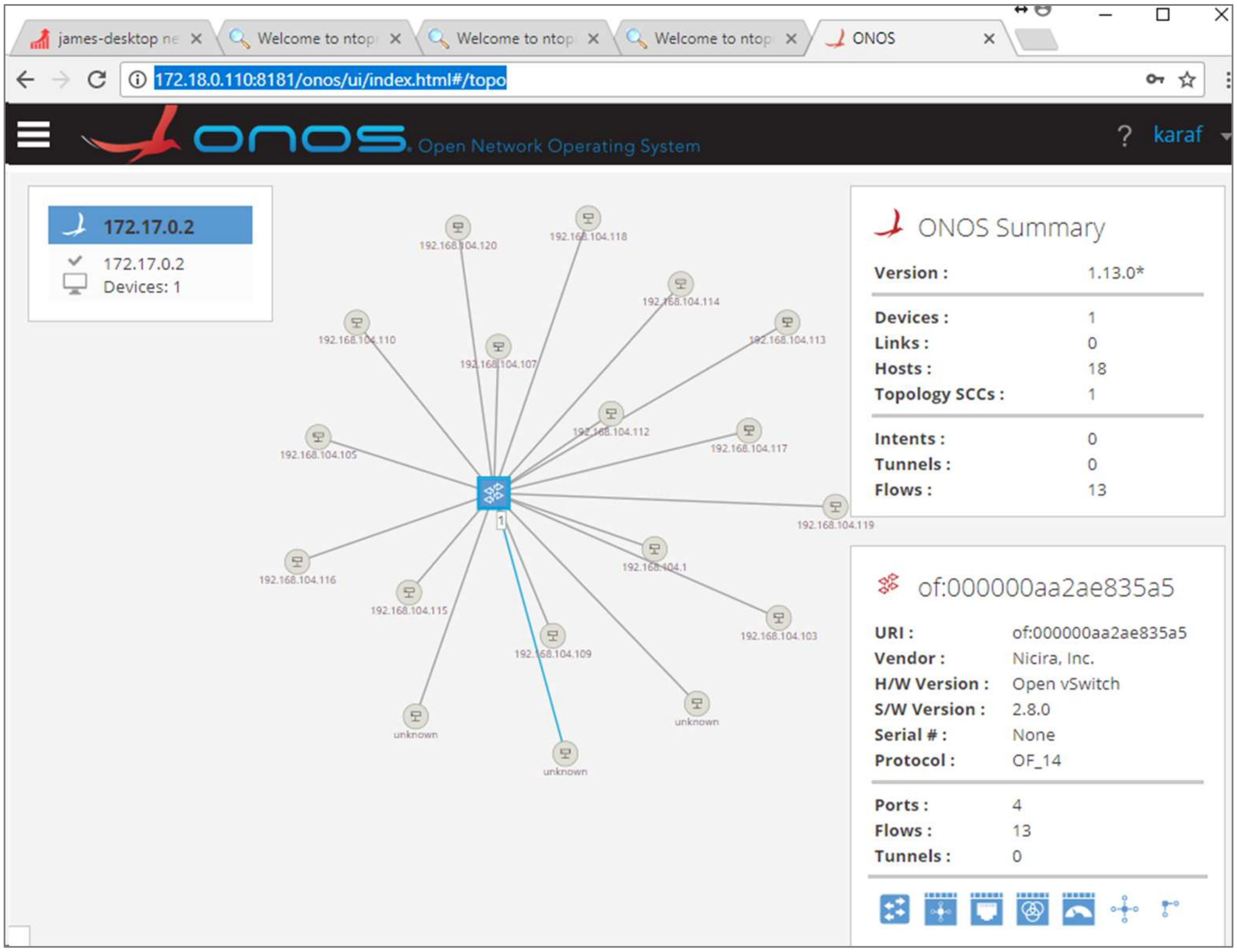
메모:

3. 네트워크 도구 (Network Tools)

❖ **ONOS**

❖ **ONOS 웹 접속**

❖ http://172.18.0.110:8181/onos/ui/



ONOS Summary

Version :	1.13.0*
Devices :	1
Links :	0
Hosts :	18
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	13

of:000000aa2ae835a5

URI :	of:000000aa2ae835a5
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.8.0
Serial # :	None
Protocol :	OF_14
Ports :	4
Flows :	13
Tunnels :	0

메모:

❖ 호스트의 L2/L3 매핑정보와 물리적 위치 정보 제공

3. 네트워크 도구 (Network Tools)

❖ **Side-Kick**

① **sudo docker run -t -i -d -p 3331:3000 --name ntopng1 lucaderi/ntopng-docker**

② **sudo docker run -t -i -d --net=host --name ntopng2 lucaderi/ntopng-docker**

메모:

- ❖ 네트워크 어플라이언스의 경우 일반적인 컨테이너 설정 권장과 달리 네트워크를 호스트모드로 설정하는 것이 더 많은 인터페이스를 모니터링 가능
- ❖ 노출 포트 3331 적용시 http://192.168.1.xx:3331
- ❖ 네트워크가 호스트모드의 경우 http://192.168.0.xx:3000