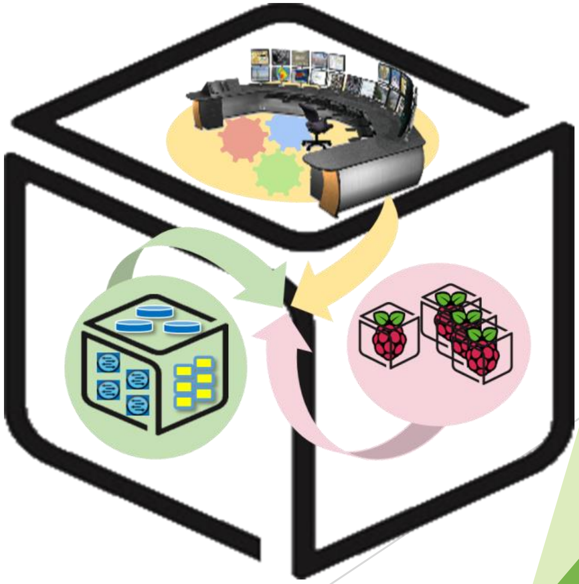
SmartX Labs for Computer Systems

Build Lab

(2016, Spring)

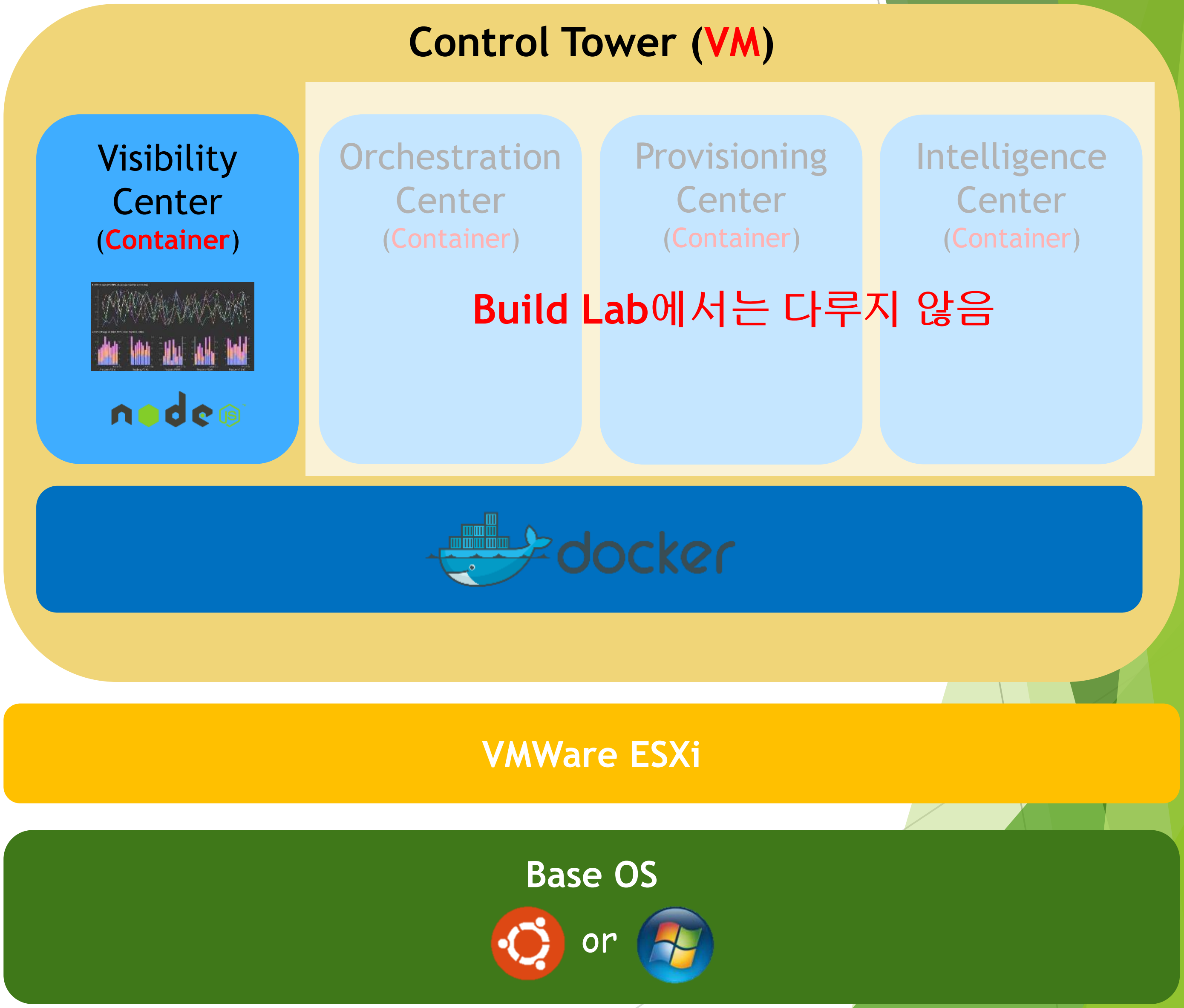
NetCS Lab



History and Contributor of Build Lab (2016. 05. 06.)

Version	Updated Date	Updated Contents	Contributor
-	2016/04	(구) Playground Lab 최종본 작성	김병돈
v1.0	2016/05	Build Lab 초안 작성 (Outline 및 Control Tower 추가)	김승룡
v1.1	2016/05	검수 및 제안 사항 반영 (apt-get source 변경 내용 외)	김승룡 김진우
V1.2	2016/05	NUC에 대한 모니터링 추가 순서 변경에 따른 내용 변경	김승룡

Build Lab: Design



Control Tower (VM)

Visibility Center (Container)

Orchestration Center (Container)

Provisioning Center (Container)

Intelligence Center (Container)

Build Lab에서는 다루지 않음

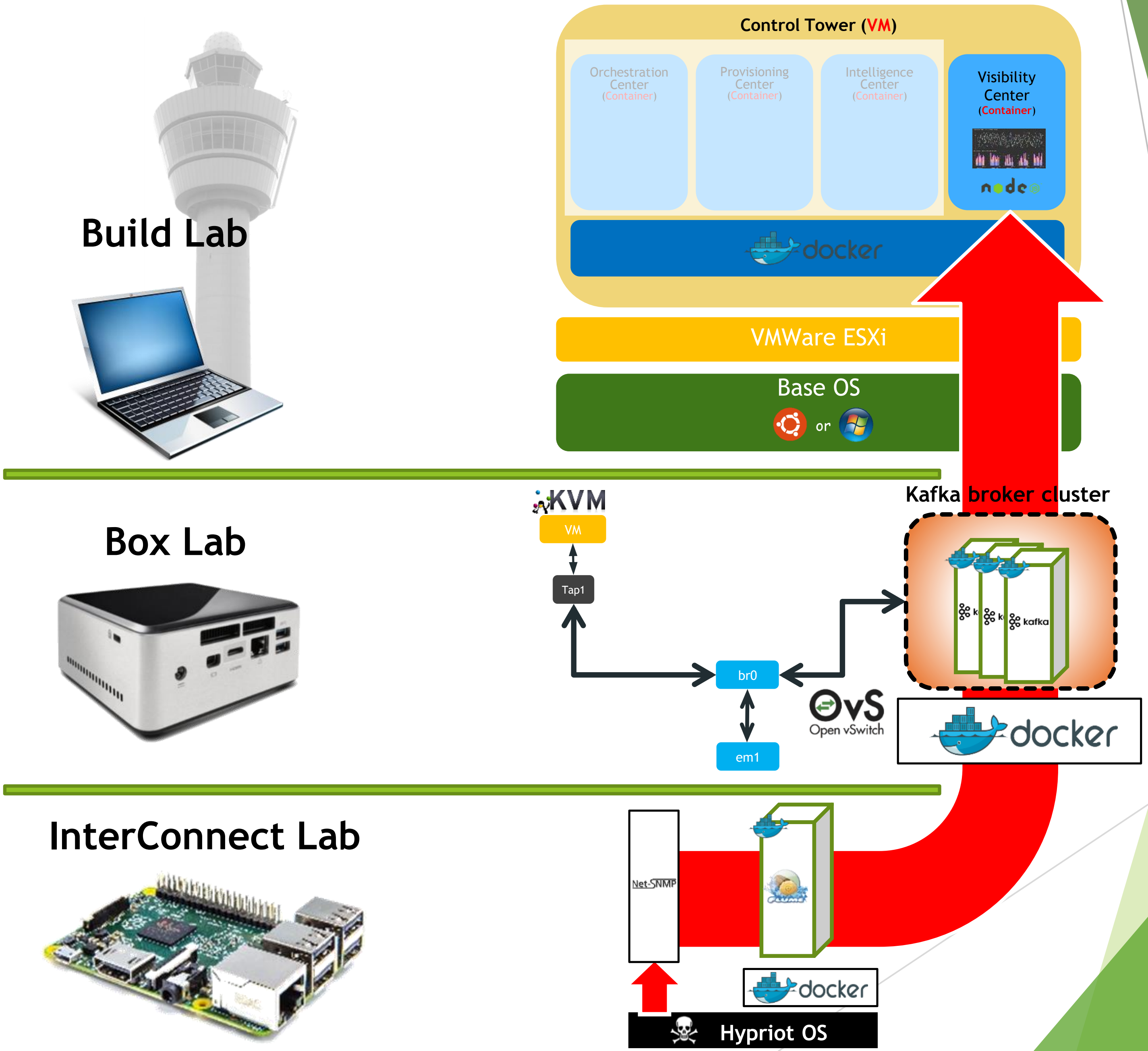
docker

VMWare ESXi

Base OS

or

Relation of SmartX Lab



Build Lab

Control Tower (VM)

Orchestration Center (Container)

Provisioning Center (Container)

Intelligence Center (Container)

Visibility Center (Container)

node

docker

VMWare ESXi

Base OS

or

Box Lab

KVM

VM

Tap1

br0

em1

Open vSwitch

Kafka broker cluster

kafka

docker

InterConnect Lab

Net-SNMP

docker

Hypriot OS

Index

- Control Tower 구축
  - Hypervisor 설치
  - Virtual Machine(VM) 생성
  - VM 환경 설정

- Visibility Center 구축
  - Docker Engine 설치
  - Docker Image 생성 및 Docker Container 실행

- Use Case: Operation Data Visibility Service
  - (InterConnect Lab에 이어)
  - NUC에서 Flume Container 구동
  - Visibility Center 환경 설정
  - Operation Data Visibility Service 실행

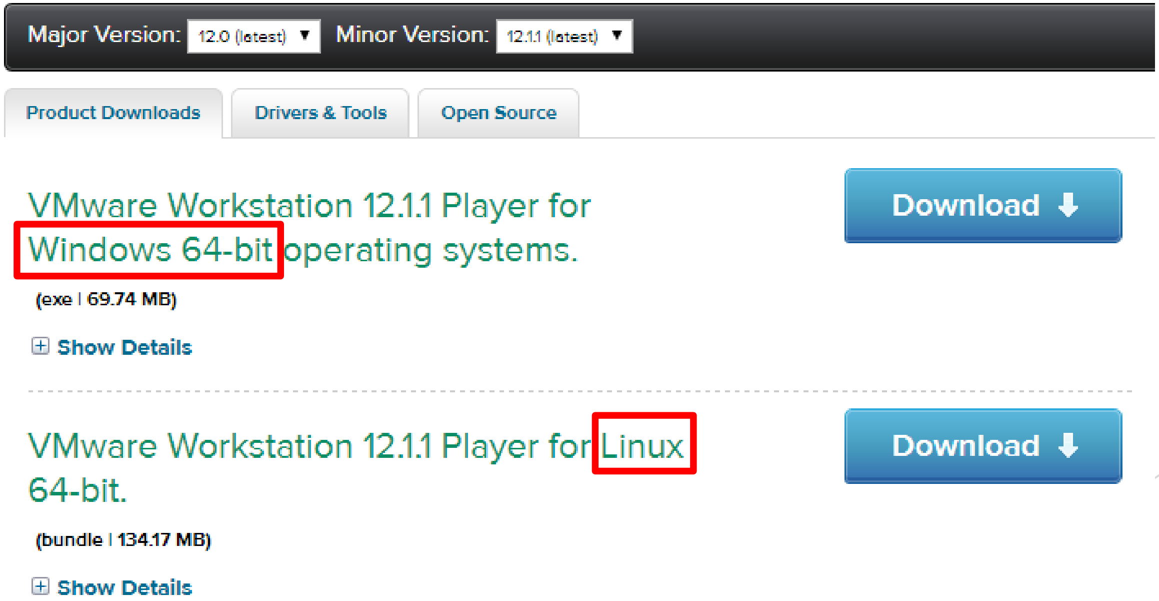
Hypervisor: VMWare ESXi

- Hypervisor
  - The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources.
- VMWare ESXi
  - VMware ESXi (formerly ESX) is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that one installs in an operating system; instead, it includes and integrates vital OS components, such as a kernel.

Hypervisor 설치

- VMWare Workstation Player Install
  - 노트북 혹은 NUC에 VM으로 Control Tower를 구축
  - 환경에 맞는 링크를 통해 VMWare Workstation Player 다운로드 https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0
  - 실행 후 옵션 변경 없이 모두 Next 클릭하여 설치 완료

64Bit OS


Major Version: 12.0 (latest) Minor Version: 12.1.1 (latest)

VMware Workstation 12.1.1 Player for Windows 64-bit operating systems.

(exe | 69.74 MB)

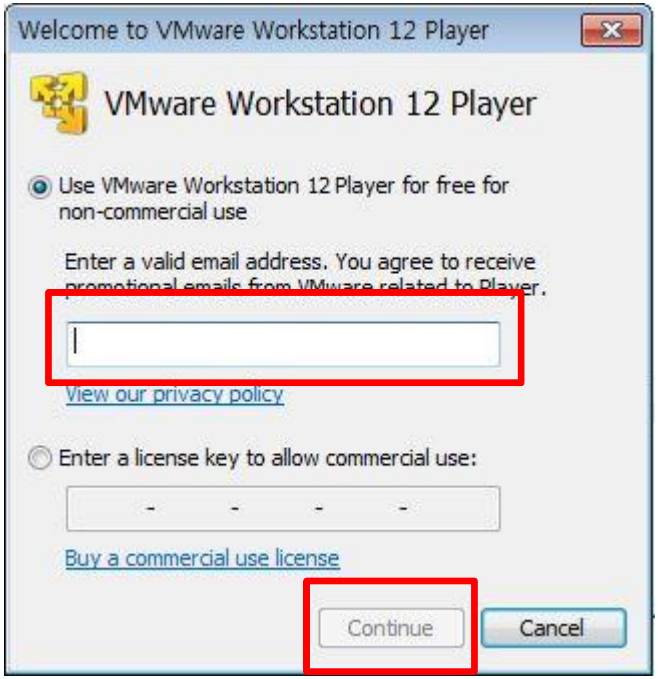
VMware Workstation 12.1.1 Player for Linux 64-bit.

(bundle | 134.17 MB)

Virtual Machine(VM) 생성 (1/4)

64Bit OS

- VMWare Workstation Player Start
  - 실습 목적의 무료 사용을 위해 유효한 메일 주소 입력 후 Continue
  - 메인 화면에서 새로운 가상 머신 추가 버튼을 클릭하여 가상 머신 생성



Welcome to VMware Workstation 12 Player

VMware Workstation 12 Player

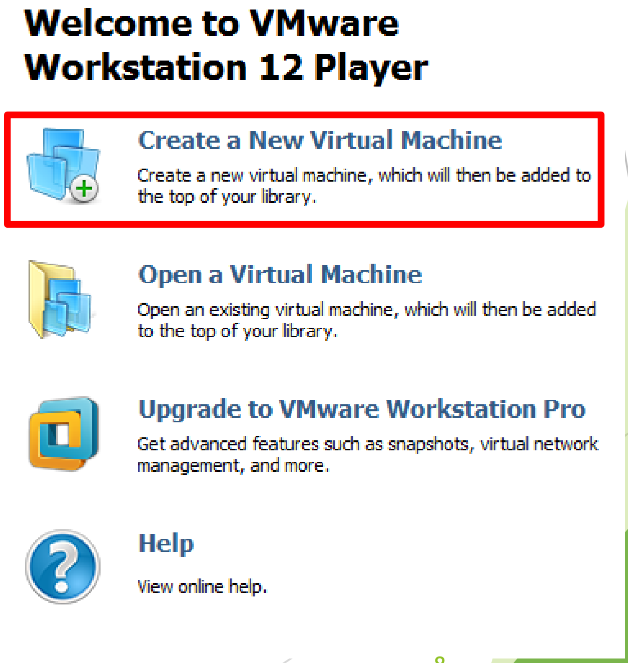
● Use VMware Workstation 12 Player for free for non-commercial use

Enter a valid email address. You agree to receive promotional emails from VMware related to Player.

View our privacy policy

○ Enter a license key to allow commercial use:

Buy a commercial use license



Welcome to VMware Workstation 12 Player

Create a New Virtual Machine
Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine
Open an existing virtual machine, which will then be added to the top of your library.

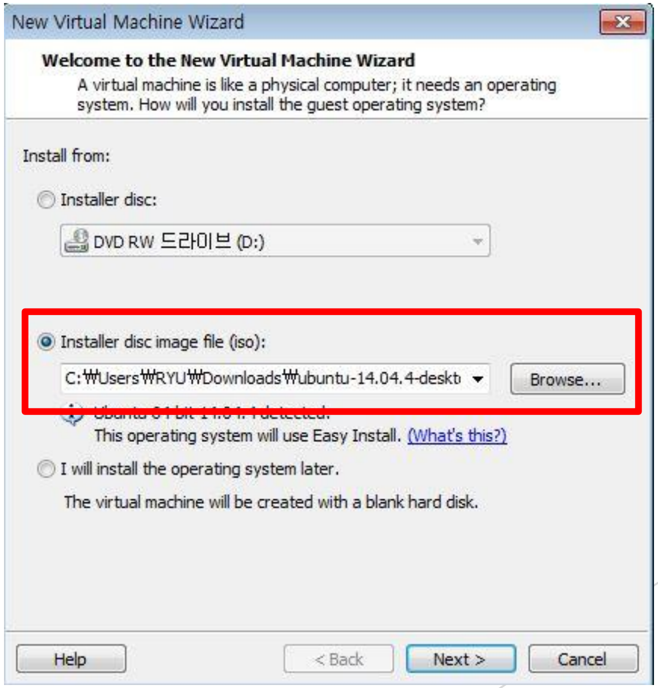
Upgrade to VMware Workstation Pro
Get advanced features such as snapshots, virtual network management, and more.

Help
View online help.

Virtual Machine(VM) 생성 (2/4)

64Bit OS

- Create a New Virtual Machine for Control Tower
  - Control Tower OS : Ubuntu Desktop 14.04.4 LTS(64bit)
  - Download Link : http://ftp.daumkakao.com/ubuntu-releases/14.04/ubuntu-14.04.4-desktop-amd64.iso
  - VMWare Workstation Player에서 Virtual Machine 생성에 사용할 Ubuntu Image 다운로드 한 뒤 해당 파일의 경로 지정



Welcome to the New Virtual Machine Wizard

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

○ Installer disc:

DVD RW 드라이브 (D:)

● Installer disc image file (iso):

C:₩Users₩RYU₩Downloads₩ubuntu-14.04.4-deskt

Ubuntu 64-bit 14.04.4 detected.

This operating system will use Easy Install. (What's this?)

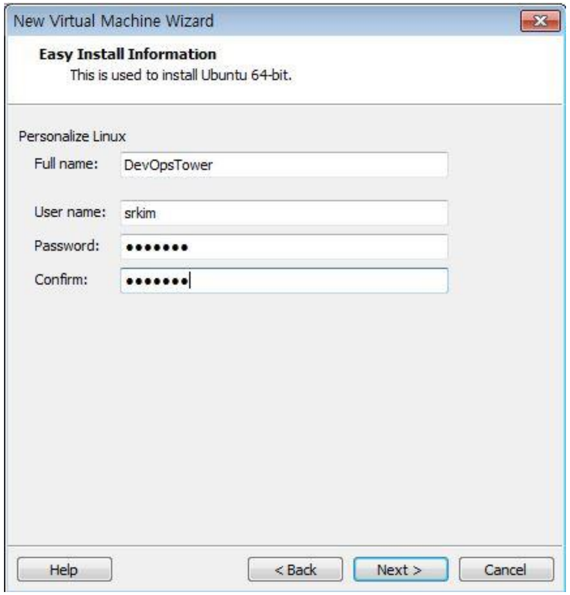
○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Virtual Machine(VM) 생성 (3/4)

64Bit OS

- Create a New Virtual Machine for Control Tower
  - 이후 계정 생성 및 저장 공간 및 위치 지정은 상황에 맞게 설정함으로써 가상 머신 생성



New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu 64-bit.

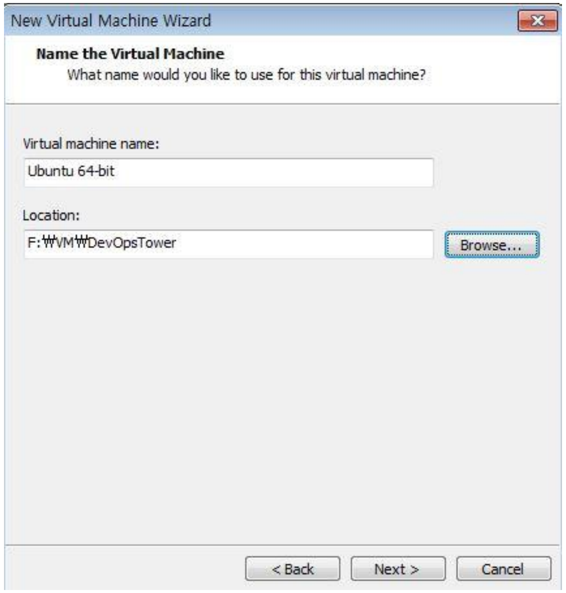
Personalize Linux

Full name: DevOpsTower

User name: srkim

Password: •••••••

Confirm: •••••••

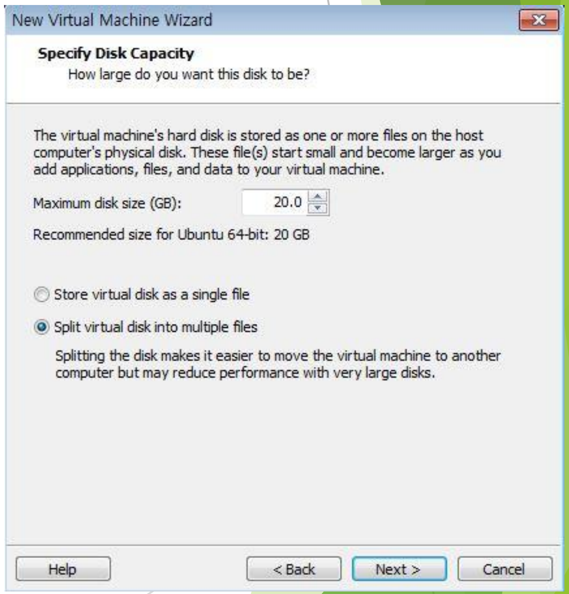


New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
Ubuntu 64-bit

Location:
F:₩VM₩DevOpsTower



New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 20.0

Recommended size for Ubuntu 64-bit: 20 GB

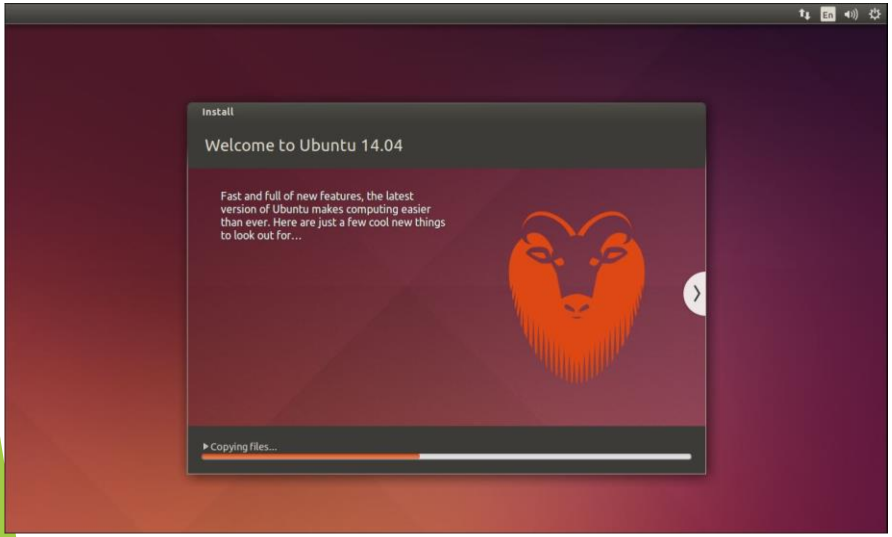
○ Store virtual disk as a single file

● Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Virtual Machine(VM) 생성 (4/4)

- Create a New Virtual Machine for Control Tower
  - OS 설치 작업이 끝나면 설정에서 지정한 계정 정보로 로그인하여 Control Tower를 활용

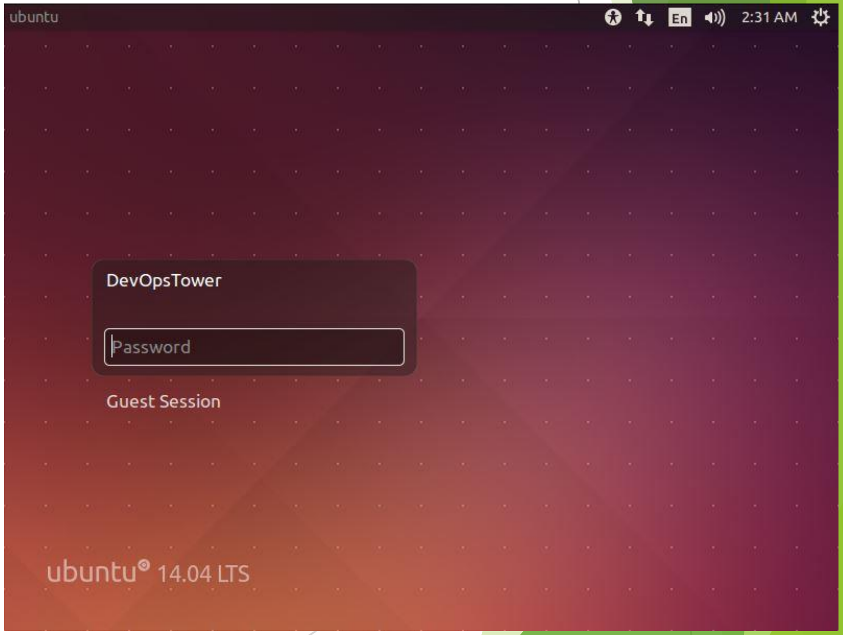


Install

Welcome to Ubuntu 14.04

Fast and full of new features, the latest version of Ubuntu makes computing easier than ever. Here are just a few cool new things to look out for...

▸ Copying files...



DevOpsTower

Password

Guest Session

VM 환경 설정

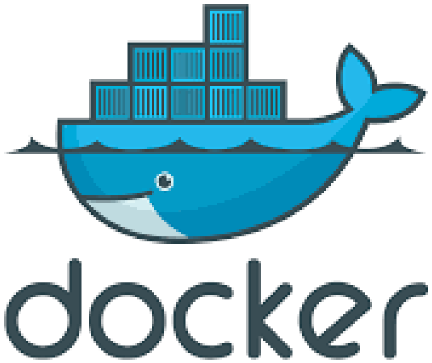
- Change apt-get sources

```
sudo sed –i 's/us.archive.ubuntu.com/ftp.daum.net/g' /etc/apt/sources.list
sudo sed –i 's/kr.archive.ubuntu.com/ftp.daum.net/g' /etc/apt/sources.list
```

- 현재 설정된 아카이브에 따라 바꿔야 할 내용이 다르지만 둘 다 입력해도 무방

```
sudo apt-get update
sudo apt-get upgrade
```

What is Docker?



- Docker is an open platform for building, shipping and running distributed applications. It gives programmers, development teams and operations engineers the common toolbox they need to take advantage of the distributed and networked nature of modern applications.

Docker Engine 설치

- Install Docker
  - curl -fsSL https://get.docker.com/ | sh
  - sudo usermod -aG docker [username]
  - sudo service docker restart
  - 이후 터미널 재시작 후에는 root 권한 없이 바로 docker 커맨드 사용 가능

Docker Image 생성 및 Docker Container 실행

- Download files to make docker image from Github
  - git clone https://github.com/SmartXBox/SmartX-mini.git
  - cd ~/SmartX-mini/ubuntu-nodejs

```
git clone https://github.com/SmartXBox/SmartX-mini.git
cd ~/SmartX-mini/ubuntu-nodejs
```

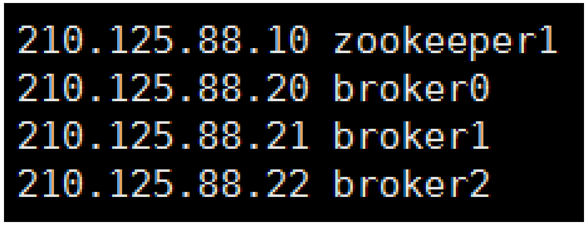
- Build Dockerfile ※ It takes long time.
  - docker build --tag ubuntu-nodejs .

- Run Docker Container
  - docker run -it --net=host --name [container name] ubuntu-nodejs

Flume Container 구동
(1) Edit /etc/hosts

- Raspberry Pi2에 Flume Container를 구동시켰던 InterConnect Lab과 같은 방식으로 진행
- NUC에서 동작하는 모든 컨테이너들의 IP 주소와 호스트 이름을 기록한다.
  - sudo vi /etc/hosts

  - 예)



```
210.125.88.10 zookeeper1
210.125.88.20 broker0
210.125.88.21 broker1
210.125.88.22 broker2
```

Flume Container 구동
(2) Install Net-SNMP

```
sudo apt-get update
```

- Download Net-SNMP

```
apt-get install –y snmp snmpd
```

- Download and apply mibs

```
apt-get install –y snmp-mibs-downloader
download-mibs
```

- Modify configuration file

```
vi /etc/snmp/snmpd.conf
```

#rocommunity public localhost -> Delete #

```
/etc/init.d/snmpd restart
```

Flume Container 구동 (3) Container 실행 및 설정 수정

1) Build Dockerfile

※ It takes long time

```
cd SmartX-mini/ubuntu-flume
docker build --tag ubuntu-flume .
docker run –it –net=host –name ubuntu-flume ubuntu-flume
```

2) Modify configuration file

```
vi conf/flume-conf.properties
```

We modify this.

```
agent.sinks.sink1.topic=[topic_name]
agent.sinks.sink1.brokerList=[broker_ipaddress:port]
```

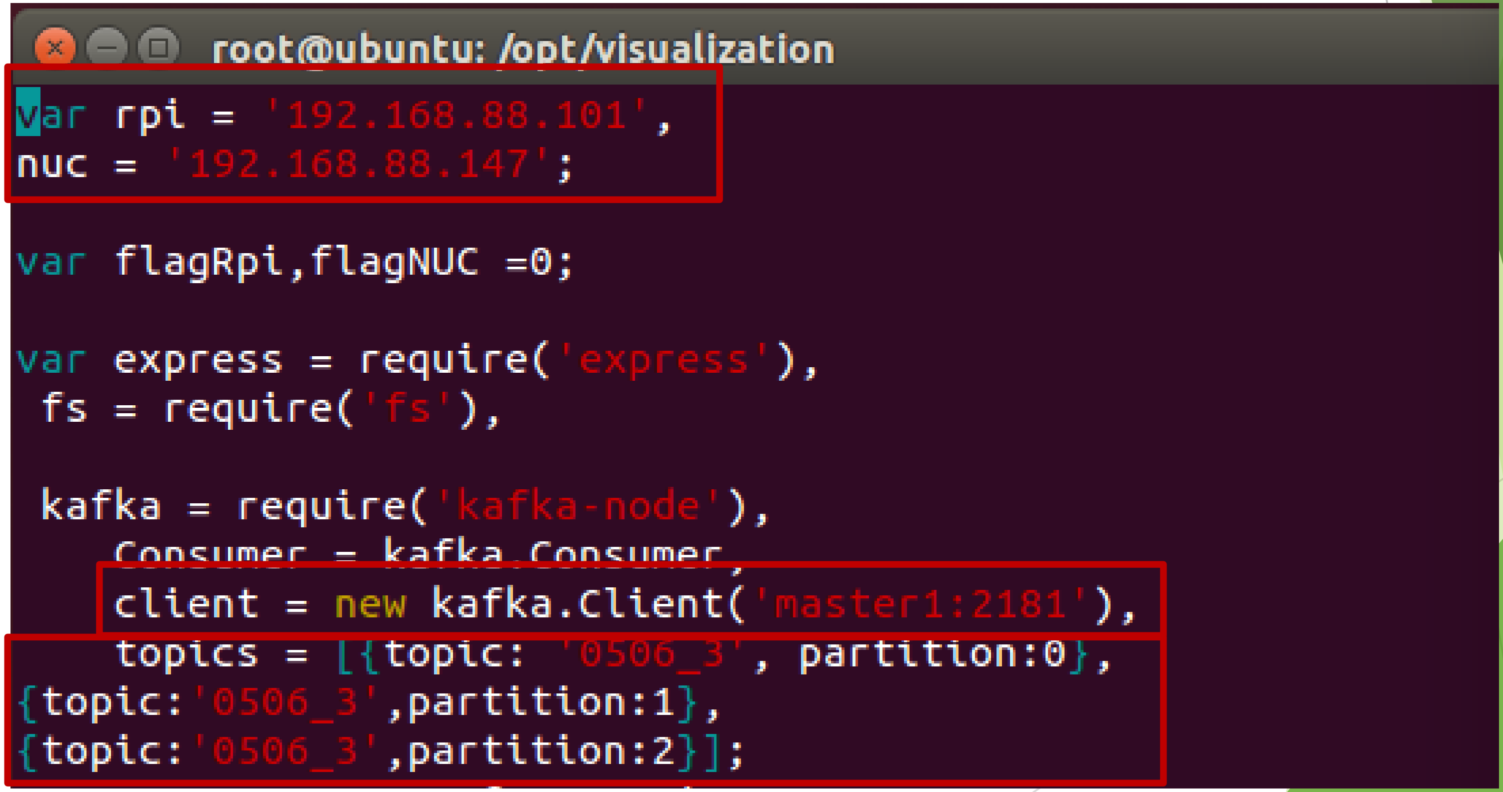
Flume Container 구동
(4) Flume Agent 실행

- Run Flume on NUC

```
bin/flume-ng agent --conf conf --conf-file conf/flume-conf.properties --name agent
-Dflume.root.logger=INFO,console
```

Visibility Center 환경설정

- node.js 파일을 환경에 맞게 수정
  - vi node.js

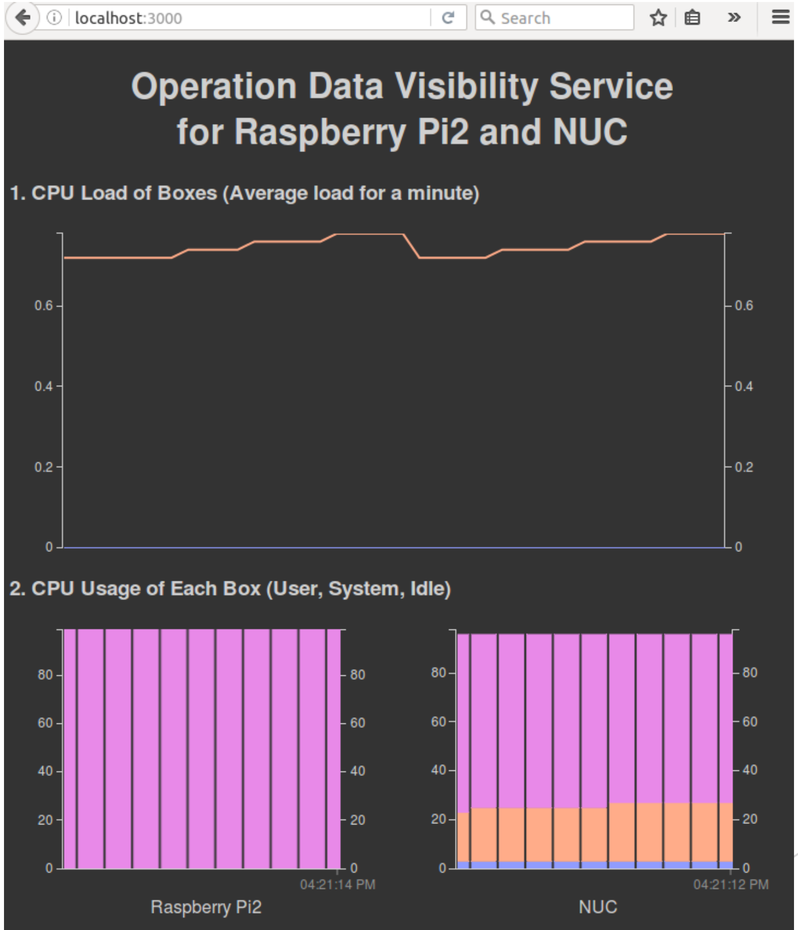


root@ubuntu: /opt/visualization

```
var rpi = '192.168.88.101',
nuc = '192.168.88.147';

var flagRpi,flagNUC =0;

var express = require('express'),
 fs = require('fs'),

 kafka = require('kafka-node'),
    Consumer = kafka.Consumer,
    client = new kafka.Client('master1:2181'),
    topics = [{topic: '0506_3', partition:0},
{topic:'0506_3',partition:1},
{topic:'0506_3',partition:2}];
```

Visibility Service 실행

- Run Visibility Service
  - node node.js
  - 이 후 VM 상에서 웹 브라우저 실행 후 http://localhost:3000/ 접속



localhost:3000

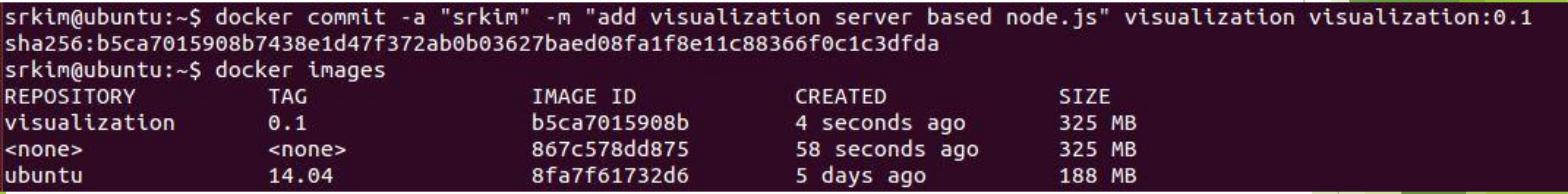
Operation Data Visibility Service for Raspberry Pi2 and NUC

1. CPU Load of Boxes (Average load for a minute)

2. CPU Usage of Each Box (User, System, Idle)

(참고) Container 변경사항 저장 및 재시작

- Commit Container
  - 컨테이너 내의 변경사항을 반영하여 새로운 컨테이너 이미지 작성
  - Ctrl+P+Q
  - docker commit -a “[username]” -m "add visualization server based node.js" visualization visualization:0.1



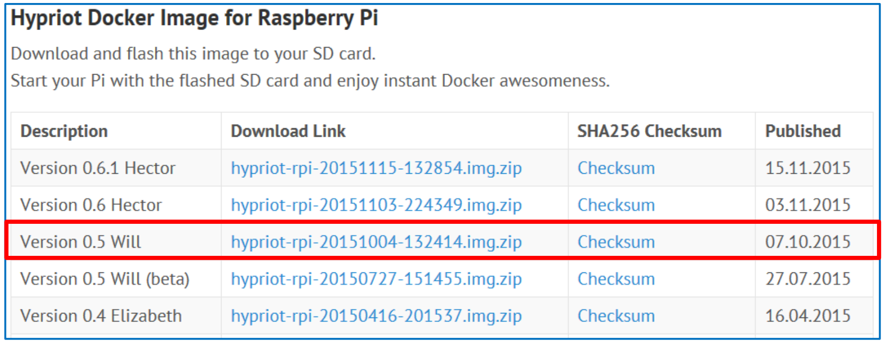
```
srkim@ubuntu:~$ docker commit -a "srkim" -m "add visualization server based node.js" visualization visualization:0.1
sha256:b5ca7015908b7438e1d47f372ab0b03627baed08fa1f8e11c88366f0c1c3dfda
srkim@ubuntu:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
visualization       0.1                 b5ca7015908b        4 seconds ago       325 MB
<none>              <none>              867c578dd875        58 seconds ago      325 MB
ubuntu              14.04               8fa7f61732d6        5 days ago          188 MB
```

- Restart Container
  - Stop했던 컨테이너를 Restart하면 이전 작업 내용을 유지한 채로 다시 컨테이너를 시작할 수 있다.
  - docker stop visualization
  - docker restart visualization

Appendix

Raspberry Pi2 OS Setting

- OS : Hypriot(Version : 0.5 Will, 07.10.2015 published)
  - Download Site : http://blog.hypriot.com/downloads/

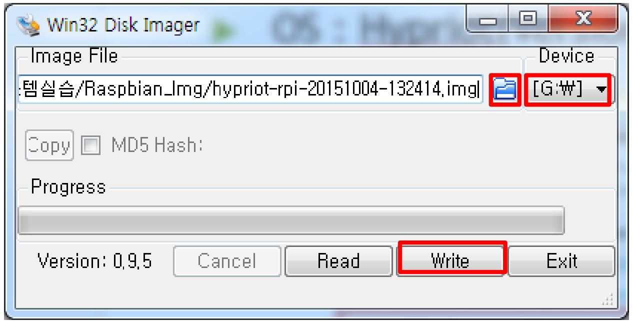


Hypriot Docker Image for Raspberry Pi

Download and flash this image to your SD card.
Start your Pi with the flashed SD card and enjoy instant Docker awesomeness.

Description	Download Link	SHA256 Checksum	Published
Version 0.6.1 Hector	hypriot-rpi-20151115-132854.img.zip	Checksum	15.11.2015
Version 0.6 Hector	hypriot-rpi-20151103-224349.img.zip	Checksum	03.11.2015
Version 0.5 Will	hypriot-rpi-20151004-132414.img.zip	Checksum	07.10.2015
Version 0.5 Will (beta)	hypriot-rpi-20150727-151455.img.zip	Checksum	27.07.2015
Version 0.4 Elizabeth	hypriot-rpi-20150416-201537.img.zip	Checksum	16.04.2015

  - 압축을 푼 후 파일(hypriot-rpi-20151004-132414.img, 1.39Gb)은 SD Writer 를 이용하여 Write.
  - SD Writer Download : https://sourceforge.net/projects/win32diskimager/files/latest/download?source=navbar



Win32 Disk Imager

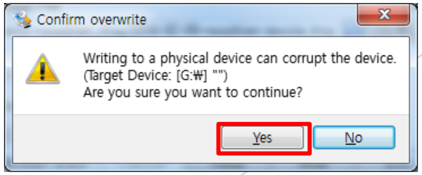
Image File: 템실습/Raspbian_Img/hypriot-rpi-20151004-132414.img

Device: [G:₩]

Copy □ MD5 Hash:

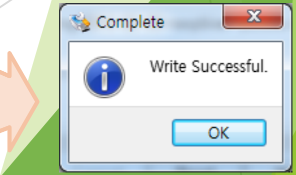
Progress

Version: 0.9.5



Confirm overwrite

Writing to a physical device can corrupt the device.
(Target Device: [G:₩] "")
Are you sure you want to continue?

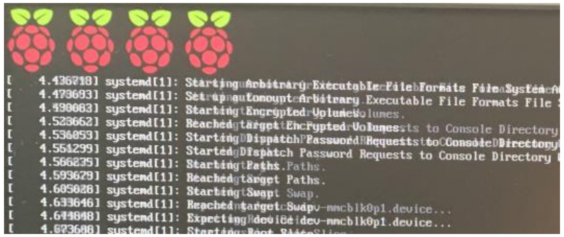


Complete

Write Successful.

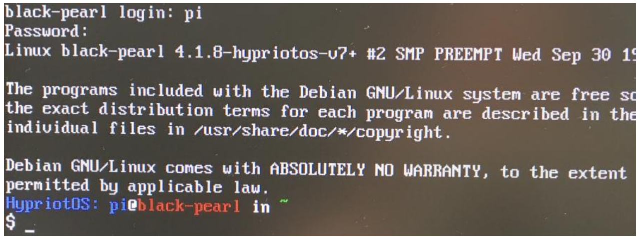
Raspberry Pi2 Environment Setting

- Micro SD 카드에 Hypriot 설치가 완료되면 Pi2에 삽입 후 부팅



```
[    4.436718] systemd[1]: Starting Arbitrary Executable File Formats File System A
[    4.473693] systemd[1]: Set up automount Arbitrary Executable File Formats File S
[    4.490083] systemd[1]: Starting Encrypted Volumes.
[    4.523662] systemd[1]: Reached target Encrypted Volumes.
[    4.536053] systemd[1]: Starting Dispatch Password Requests to Console Directory
[    4.551299] systemd[1]: Started Dispatch Password Requests to Console Directory
[    4.566235] systemd[1]: Starting Paths.
[    4.593629] systemd[1]: Reached target Paths.
[    4.605028] systemd[1]: Starting Swap.
[    4.633646] systemd[1]: Reached target Swap.
[    4.644048] systemd[1]: Expecting device dev-mmcblk0p1.device...
[    4.673688] systemd[1]: Starting Root Slice.
```

loading 화면

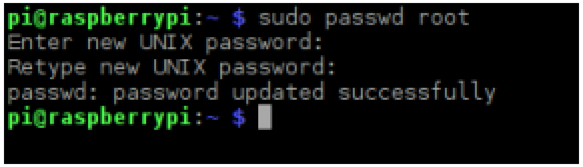


```
black-pearl login: pi
Password:
Linux black-pearl 4.1.8-hypriotos-v7+ #2 SMP PREEMPT Wed Sep 30 19

The programs included with the Debian GNU/Linux system are free so
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
HypriotOS: pi@black-pearl in ~
$ _
```

CUI 환경으로 부팅되면 성공
Login ID : pi
Password : raspberry

- Root password 변경
  - package 설치, RPM upgrade, 시스템 관리를 위해 필요

$sudo passwd root



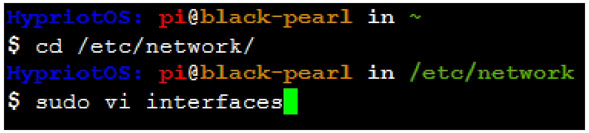
```
pi@raspberrypi:~ $ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $ █
```

- password 입력시 * 표시가 나오지 않지만 입력되고 있으니 걱정하지 말 것
- root password는 꼭 기억할 것!!!
- login ID인 pi 계정의 password 도 변경하도록 하자. **$passwd**

NUC & Pi2 IP address Setting

- Pi2 IP address 설정을 위해 필요한 파일(편집은 root 만 가능)
  - /etc/network/interfaces

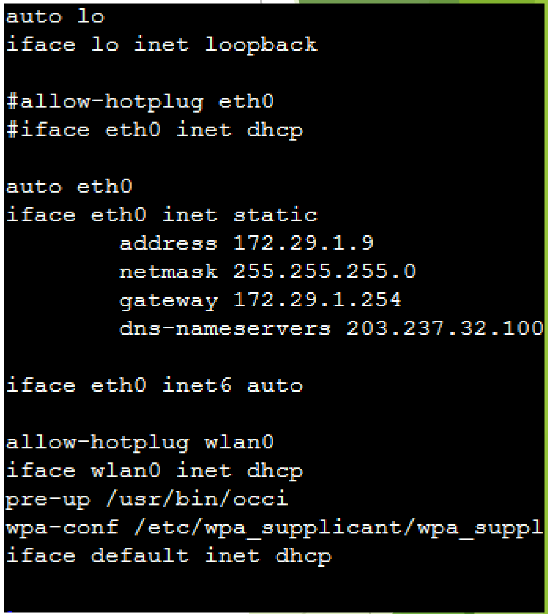
$cd /etc/network
$sudo vi interfaces



```
HypriotOS: pi@black-pearl in ~
$ cd /etc/network/
HypriotOS: pi@black-pearl in /etc/network
$ sudo vi interfaces
```

```
#iface eth0 inet dhcp ← # 은 주석
auto eth0
iface eth0 inet static
address 172.29.0.X ← ip address
netmask 255.255.255.0 ← subnet mask
gateway 172.29.0.254 ← Gateway
dns-nameservers 203.237.32.100 203.237.32.101
```

입력 예



```
auto lo
iface lo inet loopback

#allow-hotplug eth0
#iface eth0 inet dhcp

auto eth0
iface eth0 inet static
        address 172.29.1.9
        netmask 255.255.255.0
        gateway 172.29.1.254
        dns-nameservers 203.237.32.100

iface eth0 inet6 auto

allow-hotplug wlan0
iface wlan0 inet dhcp
pre-up /usr/bin/occi
wpa-conf /etc/wpa_supplicant/wpa_suppl
iface default inet dhcp
```

- 일반적으로 dns-nameservers 를 입력하면 9 page 는 필요 없으나, Hypriot OS 는 삽입되지 않으므로 resolv.conf 파일에 직접 nameserver를 입력해야 함!

NUC & Pi2 IP address Setting

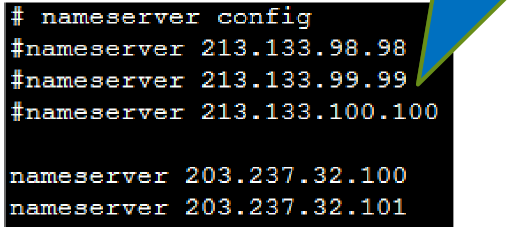
- Pi2 IP address 설정을 위해 필요한 파일
  - /etc/resolv.conf

$cd /etc/

$sudo vi resolv.conf

```
# nameserver config

nameserver 203.237.32.100

nameserver 203.237.32.101
```

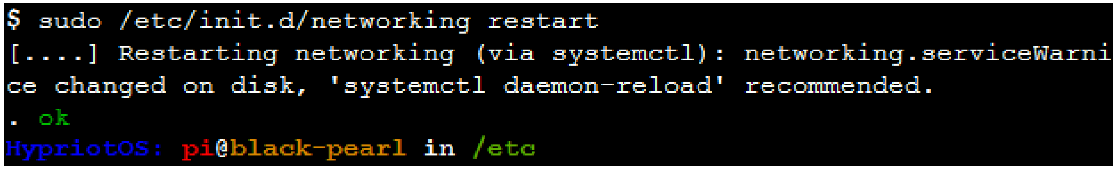
기존의 nameserver
는 #을 추가하여
주석처리



```
# nameserver config
#nameserver 213.133.98.98
#nameserver 213.133.99.99
#nameserver 213.133.100.100

nameserver 203.237.32.100
nameserver 203.237.32.101
```

$sudo /etc/init.d/networking restart 입력 또는 rebooting 후 network 확인

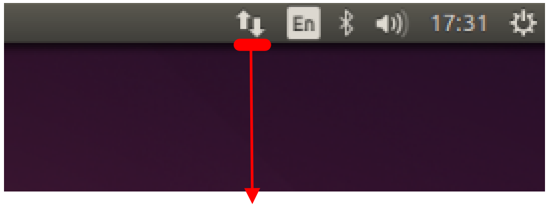
$sudo reboot (rebooting command)



```
$ sudo /etc/init.d/networking restart
[....] Restarting networking (via systemctl): networking.serviceWarni
ce changed on disk, 'systemctl daemon-reload' recommended.
. ok
HypriotOS: pi@black-pearl in /etc
```

NUC & Pi IP address Setting

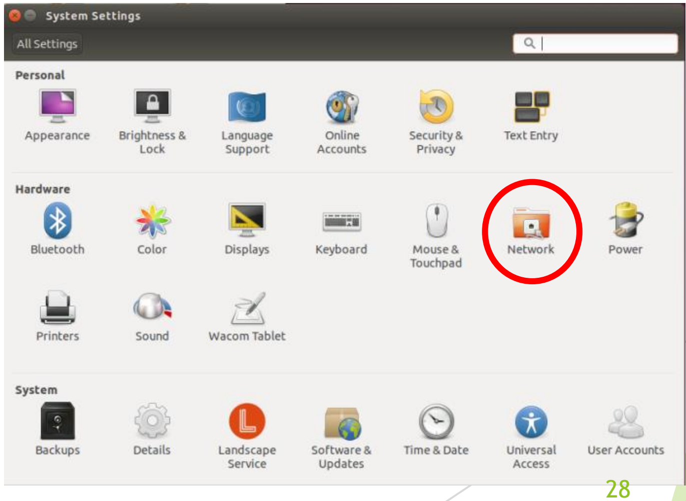
- NUC IP address Setting
  1. Pi와 동일하게 진행
  2. GUI 환경에서 setting



Network icon

Edit Connection.. 선택

System Setting icon



Linux Setting(NUC & Pi)

- Ubuntu & Hypriot OS : Debian 계열 linux
  - Package 관리 관련 명령어(apt-get : Advanced Packaging Tool)

    $sudo apt-get update

    - Update package index 갱신(/etc/apt/sources.list)

    $sudo apt-get upgrade

    - Package 목록을 비교하여 package upgrade 실행

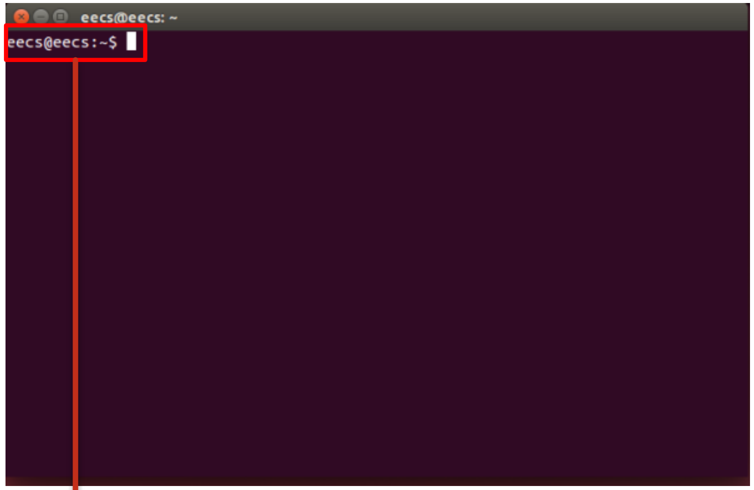
    $sudo apt-get dist-upgrade

    - Package 간의 의존성 검사를 하며 upgrade(optional)
  - Package 설치(/var/cache/apt/archive/에 설치)

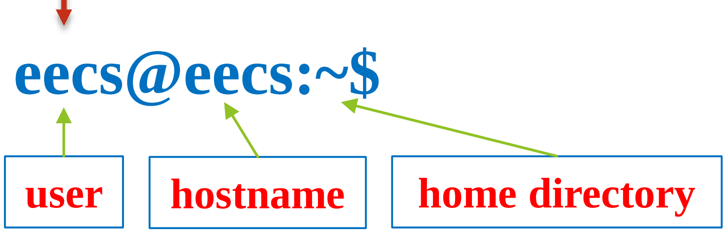
    $sudo apt-get install <package_name>
  - 자신에 맞는 기본적인 package 설치
    - Kernel Lab(Pi2)을 위한 package : gcc-4.9, make
    - Editor : vim(vi iMproved), emacs, gedit, etc.
    - SSH(Secure Shell) : openssh

Linux Beginner Guide

- Terminal 창



```
eecs@eecs: ~
eecs@eecs:~$ 
```



eecs@eecs:~$

user

hostname

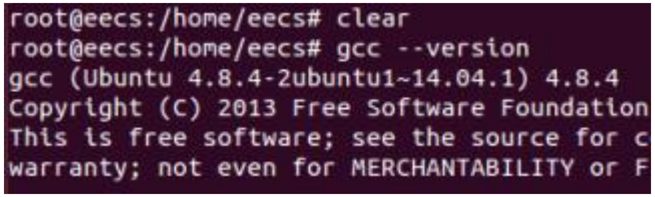
home directory

< 간단한 명령어 모음 >

$pwd	: 현재위치
$cd	: Change directory
$ls	: directory 보기
$mkdir	: directory 생성
$su	: root 계정으로 이용
$apt-get	: package 설치 및 삭제
$poweroff	
$shutdown –h now	

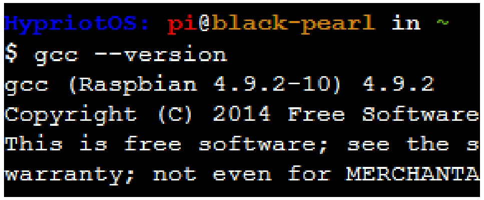
Linux Beginner Guide

- gcc-4.9 설치(NUC & Pi2)
  - gcc 설치 **$sudo apt-get install gcc-4.9**
  - gcc version 확인 : **$gcc --version**



```
root@eecs:/home/eecs# clear
root@eecs:/home/eecs# gcc --version
gcc (Ubuntu 4.8.4-2ubuntu1~14.04.1) 4.8.4
Copyright (C) 2013 Free Software Foundation
This is free software; see the source for c
warranty; not even for MERCHANTABILITY or F
```

< NUC >

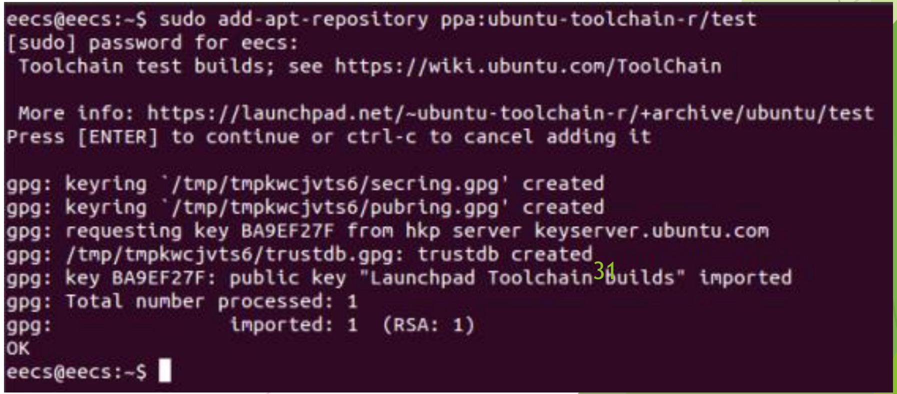


```
HypriotOS: pi@black-pearl in ~
$ gcc --version
gcc (Raspbian 4.9.2-10) 4.9.2
Copyright (C) 2014 Free Software
This is free software; see the s
warranty; not even for MERCHANTA
```

< pi2 >

- NUC 에 gcc 설치를 위해 아래의 command 실행

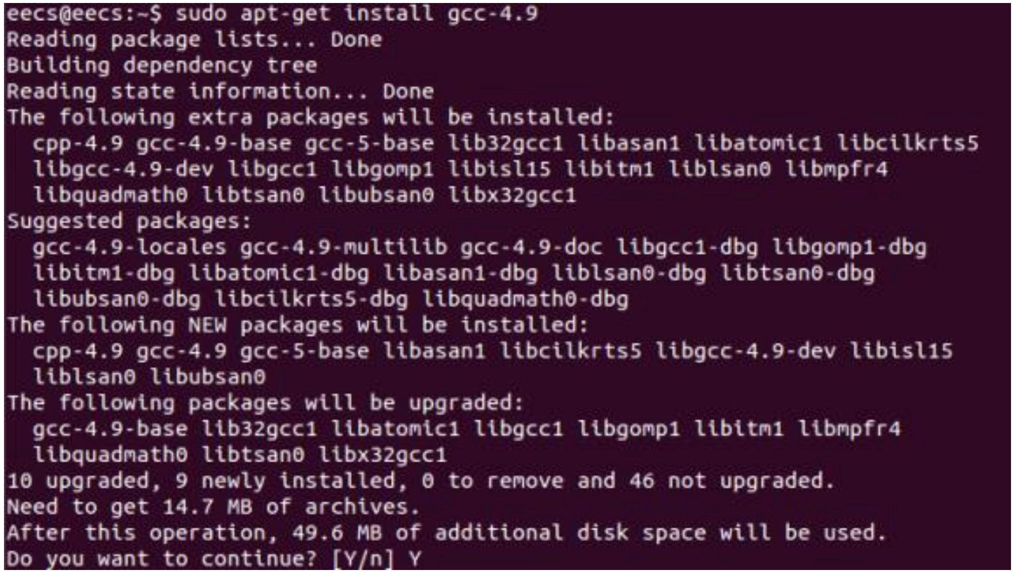
$sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$sudo apt-get update
$sudo apt-get install gcc-4.9



```
eecs@eecs:~$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
[sudo] password for eecs:
 Toolchain test builds; see https://wiki.ubuntu.com/ToolChain

 More info: https://launchpad.net/~ubuntu-toolchain-r/+archive/ubuntu/test
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmpkwcjvts6/secring.gpg' created
gpg: keyring `/tmp/tmpkwcjvts6/pubring.gpg' created
gpg: requesting key BA9EF27F from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpkwcjvts6/trustdb.gpg: trustdb created
gpg: key BA9EF27F: public key "Launchpad Toolchain builds" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
OK
eecs@eecs:~$ █
```

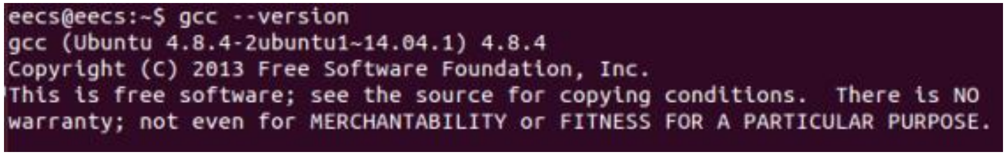
Linux Beginner Guide

- gcc-4.9 설치(NUC & Pi2)



```
eecs@eecs:~$ sudo apt-get install gcc-4.9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  cpp-4.9 gcc-4.9-base gcc-5-base lib32gcc1 libasan1 libatomic1 libcilkrts5
  libgcc-4.9-dev libgcc1 libgomp1 libisl15 libitm1 liblsan0 libmpfr4
  libquadmath0 libtsan0 libubsan0 libx32gcc1
Suggested packages:
  gcc-4.9-locales gcc-4.9-multilib gcc-4.9-doc libgcc1-dbg libgomp1-dbg
  libitm1-dbg libatomic1-dbg libasan1-dbg liblsan0-dbg libtsan0-dbg
  libubsan0-dbg libcilkrts5-dbg libquadmath0-dbg
The following NEW packages will be installed:
  cpp-4.9 gcc-4.9 gcc-5-base libasan1 libcilkrts5 libgcc-4.9-dev libisl15
  liblsan0 libubsan0
The following packages will be upgraded:
  gcc-4.9-base lib32gcc1 libatomic1 libgcc1 libgomp1 libitm1 libmpfr4
  libquadmath0 libtsan0 libx32gcc1
10 upgraded, 9 newly installed, 0 to remove and 46 not upgraded.
Need to get 14.7 MB of archives.
After this operation, 49.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

- version 확인



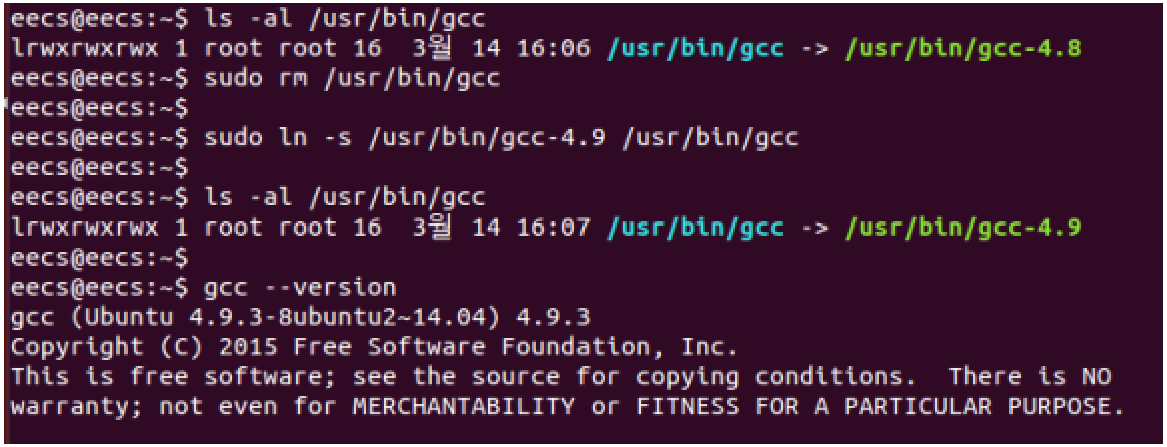
```
eecs@eecs:~$ gcc --version
gcc (Ubuntu 4.8.4-2ubuntu1~14.04.1) 4.8.4
Copyright (C) 2013 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- link 설정을 해주면 된다.

Linux Beginner Guide

- gcc-4.9 link 설정
  - 현재 gcc-4.8 version 이 link 되어 있음
  - 현재의 link 를 지우고 4.9 version으로 재설정

$sudo rm /usr/bin/gcc : link 삭제
$sudo ln –s /usr/bin/gcc-4.9 /usr/bin/gcc : link 설정



```
eecs@eecs:~$ ls -al /usr/bin/gcc
lrwxrwxrwx 1 root root 16  3월  14 16:06 /usr/bin/gcc -> /usr/bin/gcc-4.8
eecs@eecs:~$ sudo rm /usr/bin/gcc
eecs@eecs:~$
eecs@eecs:~$ sudo ln -s /usr/bin/gcc-4.9 /usr/bin/gcc
eecs@eecs:~$
eecs@eecs:~$ ls -al /usr/bin/gcc
lrwxrwxrwx 1 root root 16  3월  14 16:07 /usr/bin/gcc -> /usr/bin/gcc-4.9
eecs@eecs:~$
eecs@eecs:~$ gcc --version
gcc (Ubuntu 4.9.3-8ubuntu2~14.04) 4.9.3
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Vi/vim editor 명령어 모음

- 입력 모드 : 원하는 글자를 입력
  - **a** : 현재 위치의 다음부터 입력 시작
  - **i** : 현재 위치의 앞에서부터 입력 시작
- 명령 모드 : 문서 편집을 할 수 있으며, 입력 모드 상태에서 ESC키를 누르면 명령모드로 전환됨
  - **x** : 커서가 있는 문자 삭제
  - **dd** : 현재 줄 전체 삭제
- 라인 모드 : ESC 키를 누른 후 colon(:) prompt 에서 명령을 입력하며 저장, 편집, 검색 기능 제공
  - **:q** : 그대로 종료하기
  - **:q!** : 변경된 내용을 저장하지 않고 종료하기
  - **:wq** : 변경된 내용을 저장하고 종료하기
- 보다 다양한 명령어가 있으며, 직접 실습을 하는 것이 좋은 방법