ONOS Installation Tutorial

James Won-Ki Hong, Jian Li, Seyeon Jeong

Dept. of Computer Science & Engineering

POSTECH

http://dpnm.postech.ac.kr/~jwkhong

jwkhong@postech.ac.kr

Outline

- W6-1: Preparation, Pre-installation
- W6-2: ONOS Single Instance Installation
- W6-3: ONOS Multiple Instance Installation

Preparation, Pre-installation

James Won-Ki Hong, Jian Li, Seyeon Jeong

Dept. of Computer Science & Engineering

POSTECH

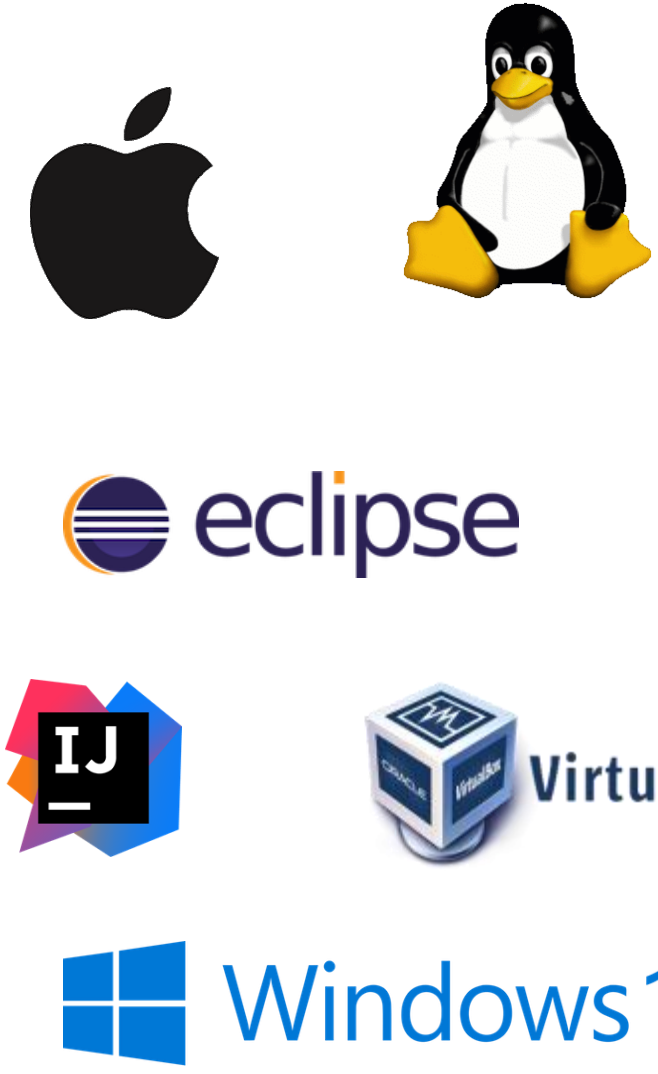
http://dpnm.postech.ac.kr/~jwkhong

jwkhong@postech.ac.kr

Prerequisites (1/2)

❖ Host Machine

- Purpose
  - Development
  - Spawn VM
- Hardware spec.
  - CPU: 8 cores+
  - RAM: 8 GB+
  - HDD: 100 GB+
- Software
  - OS
    - **MAC (recommended)**
    - Windows
    - Linux (any)
  - Applications
    - Java JDK + JRE
    - Java IDE
      - **IntelliJ (Recommended)**
      - Eclipse
    - Oracle Virtual Box



Prerequisites (2/2)

❖ VM for Build Machine

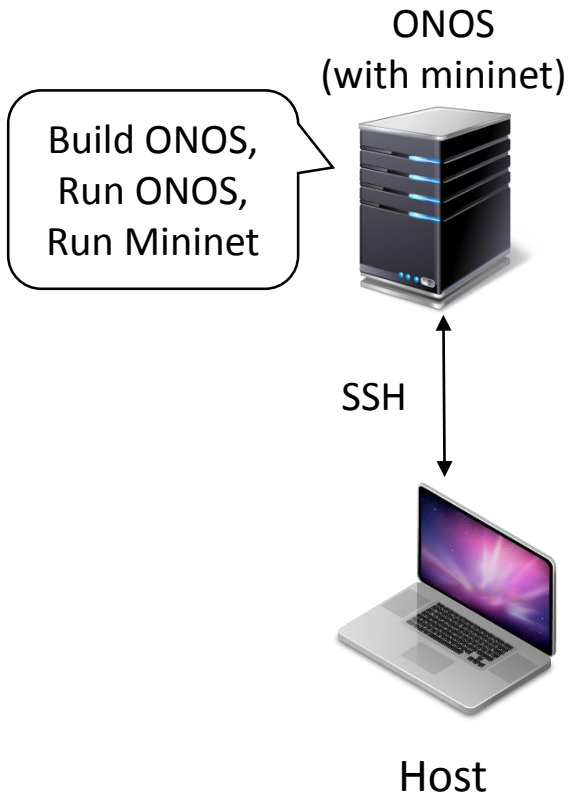
- Purpose
  - Build ONOS source
  - Run Mininet
- Hardware Spec.
  - CPU: 2 Core+
  - RAM: 2GB+
  - HDD: 8GB+
- Software
  - OS
    - Ubuntu 14.04 LTS 32/64bits (Recommended)
    - **CentOS 7.x 32/64bits**
  - Application
    - Oracle Java 8 JDK + JRE
    - Apache Maven (3.3.9)
    - Git (latest version)
    - Apache Karaf (3.0.8) (optional)

❖ VM for Running ONOS

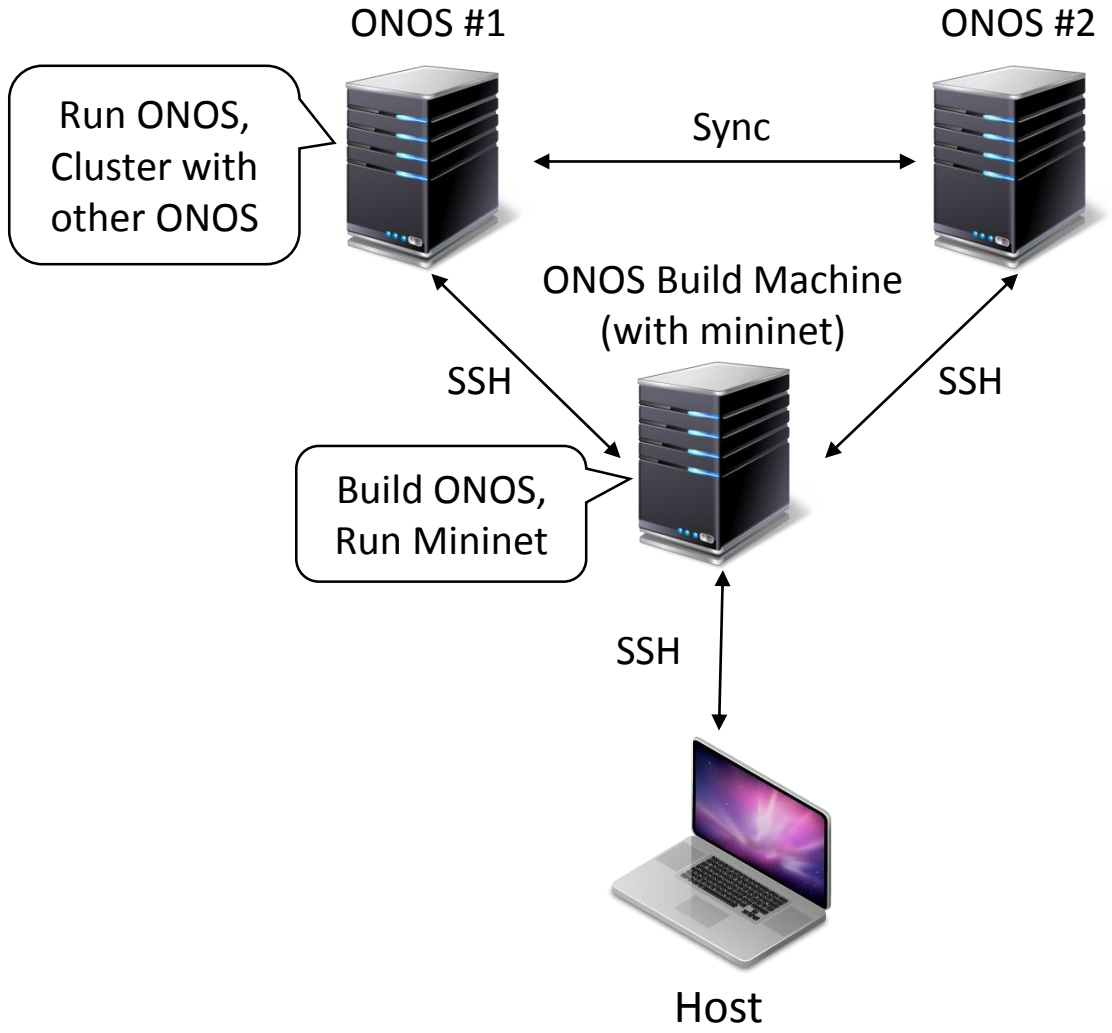
- Purpose
  - Run ONOS
- Hardware Spec.
  - CPU: 1 Cores+
  - RAM: 1GB+
  - HDD: 8GB+
- Software
  - OS
    - Ubuntu 14.04 LTS 32/64bits (Recommended)
    - **CentOS 7.x 32/64bits**
  - Applications
    - Oracle Java 8 JDK + JRE

Overall Structure

❖ Single Instance Scenario



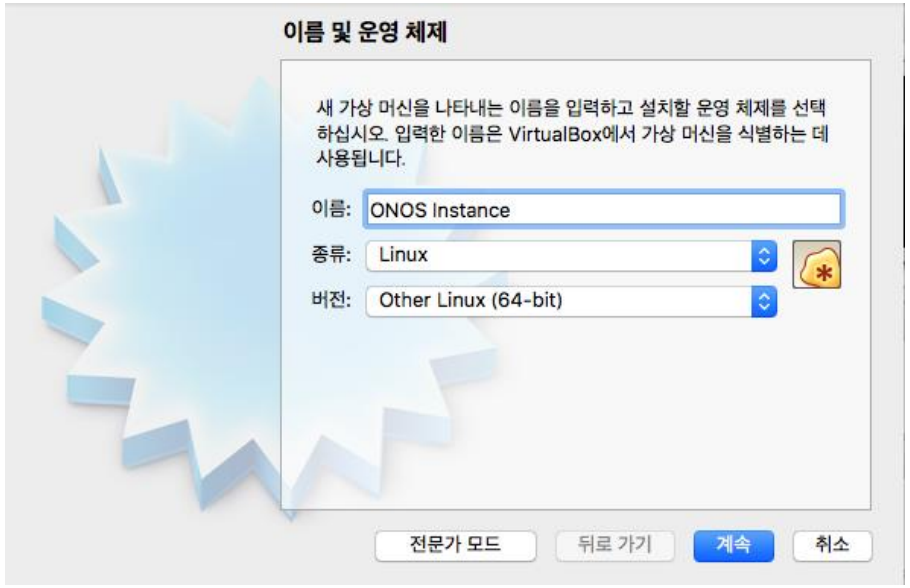
❖ Multi-Instance Scenario

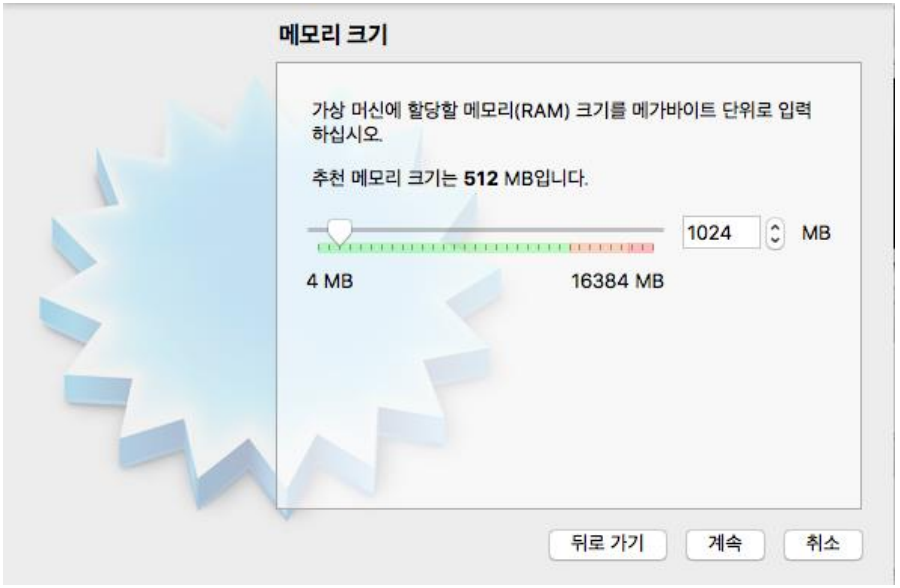


Host (1/5)

❖ Virtual Box Installation

- Download and install Virtual Box 5.x (e.g., 5.1.20)
  - https://www.virtualbox.org/wiki/Downloads
- Spawn VM and install CentOS 7.x (e.g., 7.0 minimal)
  - Name: ONOS Instance
  - Version: Other Linux (64-bit)
  - RAM: 2GB
  - CPU cores: 2

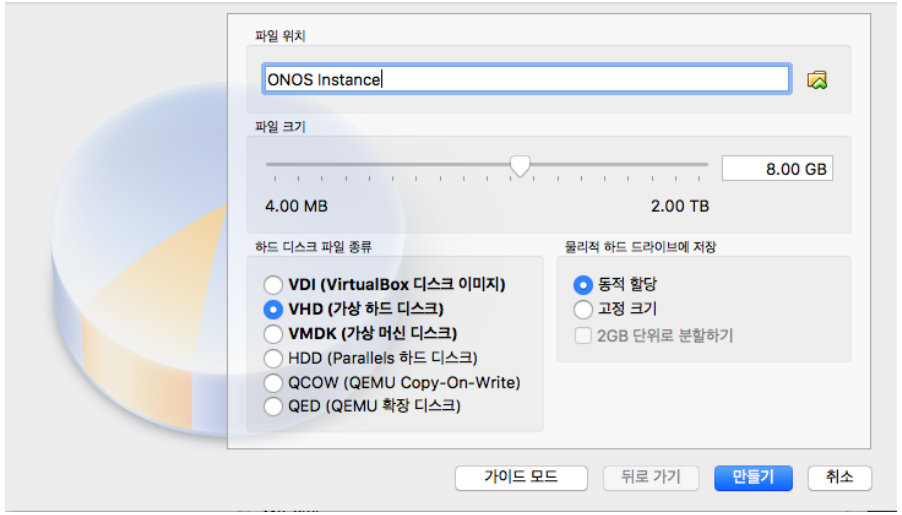




Host (2/5)

❖ Virtual Box Installation

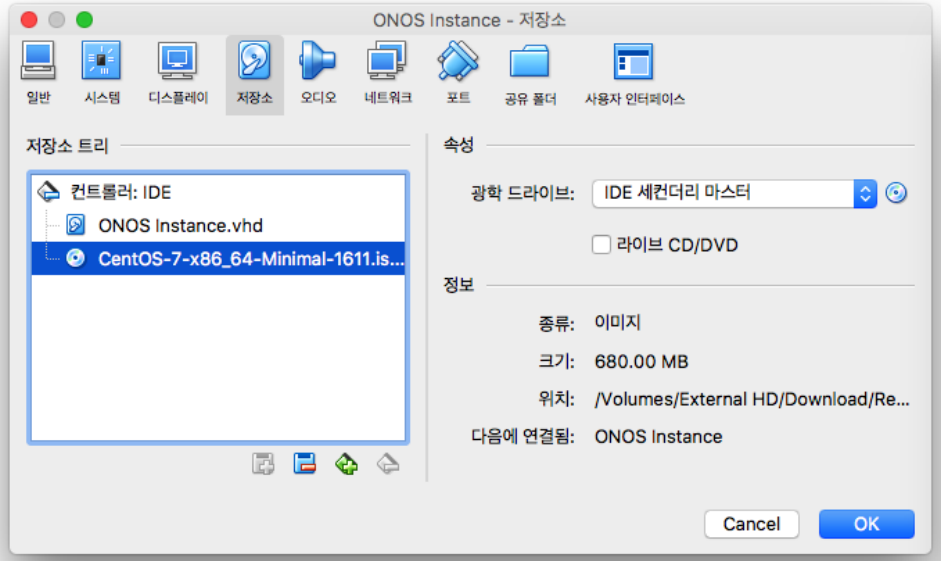
- Spawn VM and install CentOS 7.x
  - HDD size: 8GB+
  - HDD type: VHD
  - HDD allocation type: dynamic

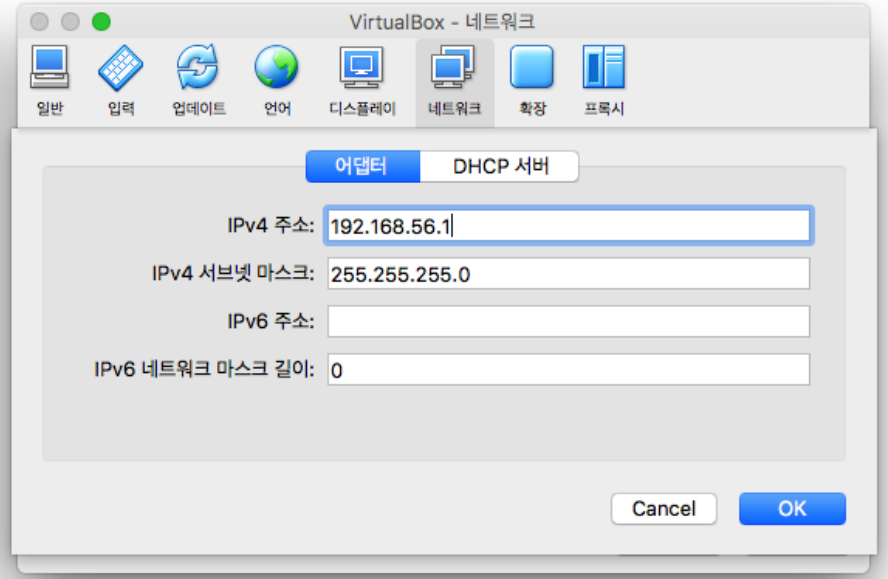


Host (3/5)

❖ Virtual Box Installation

- Spawn VM and install CentOS 7.x
  - Instance → Setting → Storage → Specify installation image file (.ISO)
    - E.g.,: CentOS-7-x86_64-Minimal-1611.iso
  - Preference → Network → Host-only Network → Add an adapter
    - Create a new adapter if no adapter exists
    - Gateway → 192.168.56.1
    - Submask → 255.255.255.0

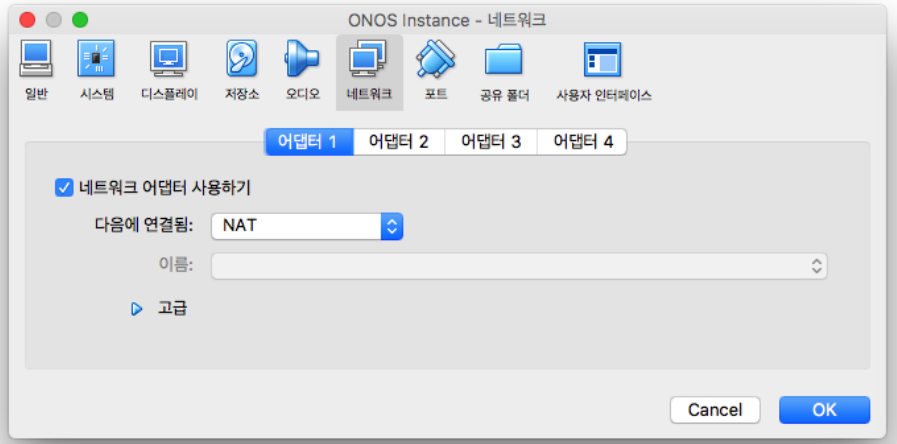


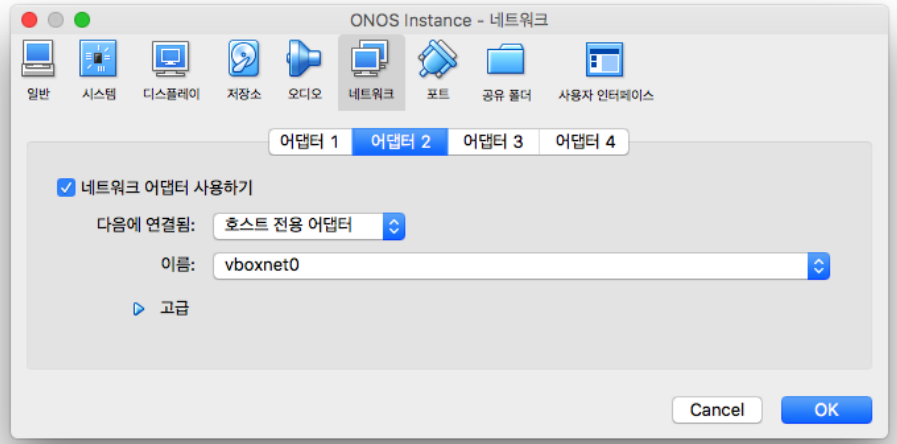


Host (4/5)

❖ Virtual Box Installation

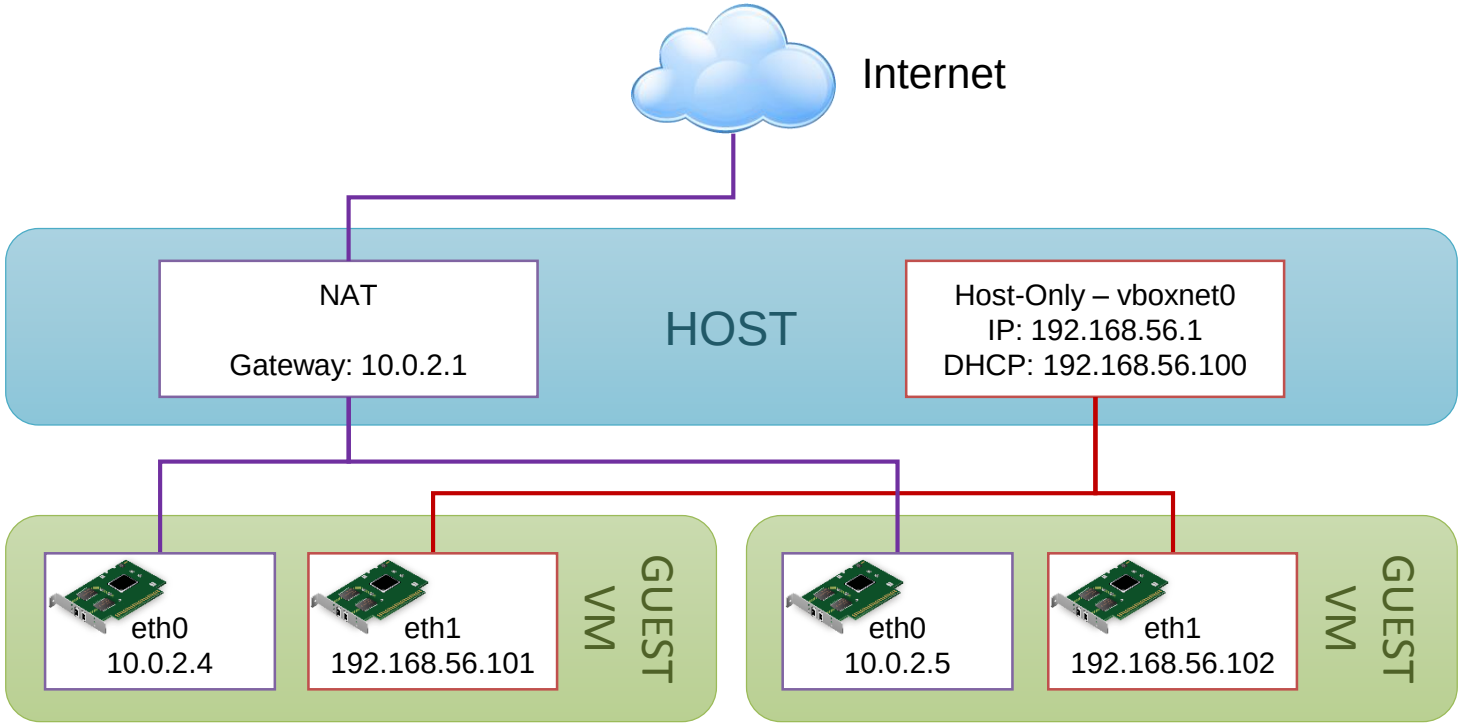
- Spawn VM and install CentOS 7.x
  - Instance → Setting → Network → Adapter #1 → NAT
    - Assign dynamic IP address using DHCP (connects to Internet)
  - Instance → Setting → Network → Adapter #2 → Host only Adapter
    - Assign static IP address in each VM (private networking)
  - Launch an instance





Host (5/5)

❖ Networking of Host and Virtual Machine



❖ Shell Tool Installation

- E.g., putty (Windows), iTerm 2 (Mac)

❖ IDE Installation

- E.g., IntelliJ or Eclipse

Virtual Machine (1/3)

❖ Install CentOS 7.x from ISO Image (Option #1)

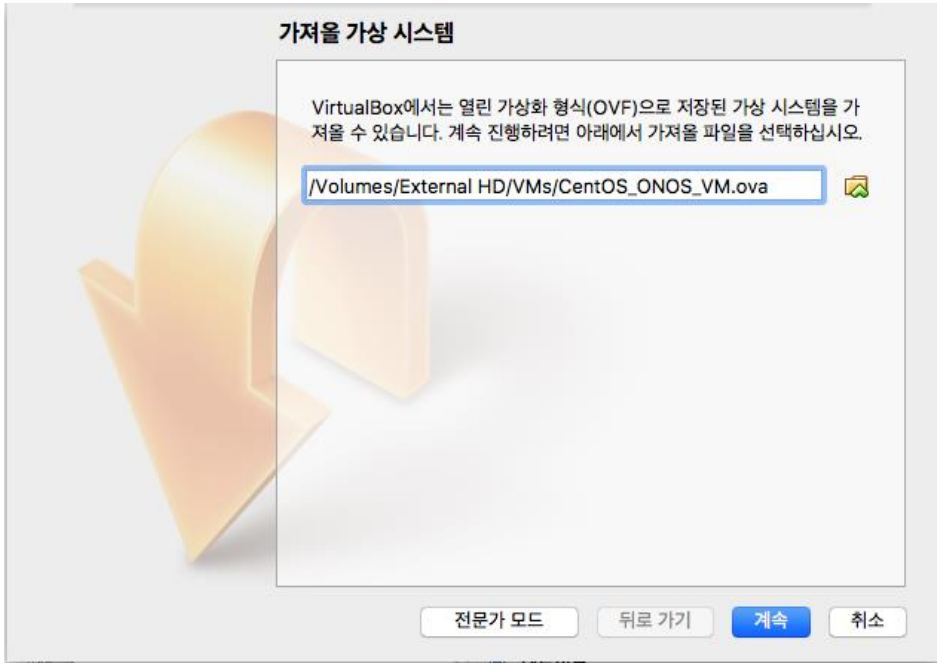
- Minimal install
- Network
  - ifcfg-enp0s3
    - Located under /etc/sysconfig/network-scripts/ifcfg-enp0s3
    - Bound to vNIC #1
    - Need to configure ”ONBOOT” to “yes”
  - ifcfg-enp0s8
    - Located under /etc/sysconfig/network-scripts/ifcfg-enp0s8
    - Bound to vNIC #2

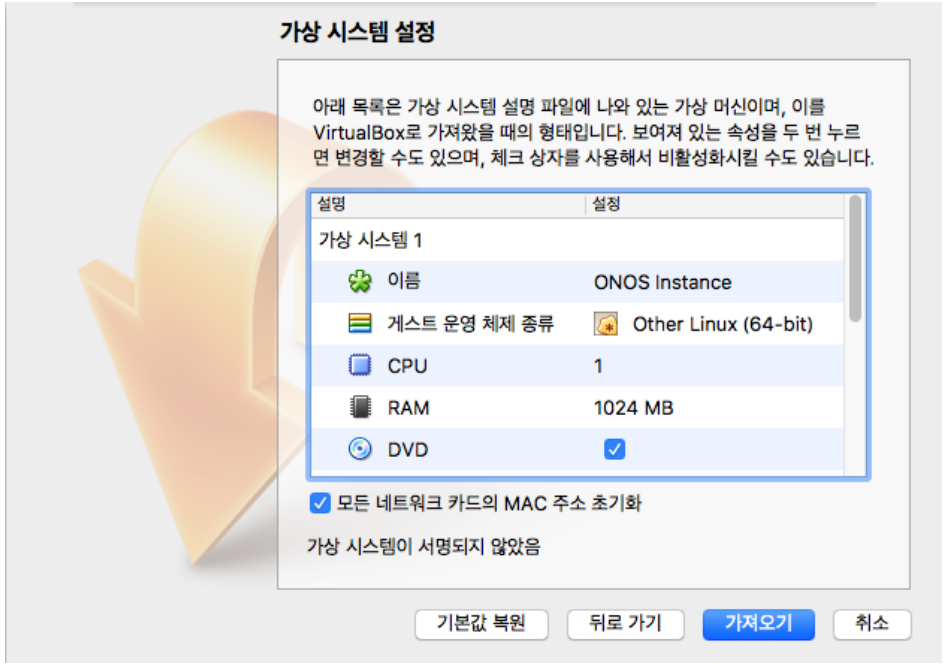
```
TYPE=Ethernet
BOOTPROTO=static
IPADDR=192.168.56.101
NETMASK=255.255.255.0
NM_CONTROLLED=no
...
DEVICE=enp0s8
ONBOOT=yes
```

Virtual Machine (2/3)

❖ Import CentOS 7.x from VM Image (Option #2)

- Checkout the CentOS 7.x pre-installed VM image
  - FileName: CentOS_ONOS_VM.ova
- Import a VM image
  - File → Import a VM image
  - ID: root, PW: onosproject





Virtual Machine (3/3)

❖ Check Network Configurations

- Two NICs should be assigned IP addresses

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:1a:ab:7f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86229sec preferred_lft 86229sec
    inet6 fe80::f776:35c1:d012:3022/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:a2:a9:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea2:a96c/64 scope link
       valid_lft forever preferred_lft forever
```

Pre-Installation (1/3)

❖ Basic Setup

- Install networking related utilities and setup tools

```
# yum update
# yum install -y net-tools wget setuptool ntsysv perl
```

- Deactivate SELinux
  - It is recommended to set SELinux to ***Permissive*** mode

```
# setenforce 0
# vi /etc/sysconfig/selinux
...
SELINUX=permissive
...
```

- Add a new user
  - ID: sdn, PW: sdn

```
# adduser sdn
# passwd sdn
# su - sdn
$
```

Pre-Installation (2/3)

❖ Basic Setup

- Generate RSA key for sdn user
  - Objective
    - Allow ONOS instance logins to other ONOS instances without prompting password dialog
  - Method
    - Generate a asymmetric RSA key

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sdn/.ssh/id_rsa):
Created directory '/home/sdn/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sdn/.ssh/id_rsa.
Your public key has been saved in /home/sdn/.ssh/id_rsa.pub.
The key fingerprint is:
51:3d:b2:13:99:e2:6e:26:dd:36:b3:6b:88:e9:e8:3d sdn@localhost.localdomain
The key's randomart image is:
...
```

Pre-Installation (3/3)

❖ Basic Setup

- In order to allow RSA key based authentication, we need to generate authorized_keys file under .ssh directory
- Easily realizable using following commands
  - Note: need to type-in password first time

```
$ ssh-copy-id localhost
...
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

$ ssh sdn@localhost
```

- Assign root privilege to sdn user without any password

```
# vi /etc/sudoers
sdn             ALL=(ALL)           NOPASSWD:ALL
```

DEMO