Mininet
Installation & How to use

Shin Jae Min

shin.jaemin@csi.skku.edu

What is Mininet?

- “Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.” – Mininet overview

- You can create various network situation by manipulating link rate, link delay, queue length, loss rate, topology etc.

Installation and Setting

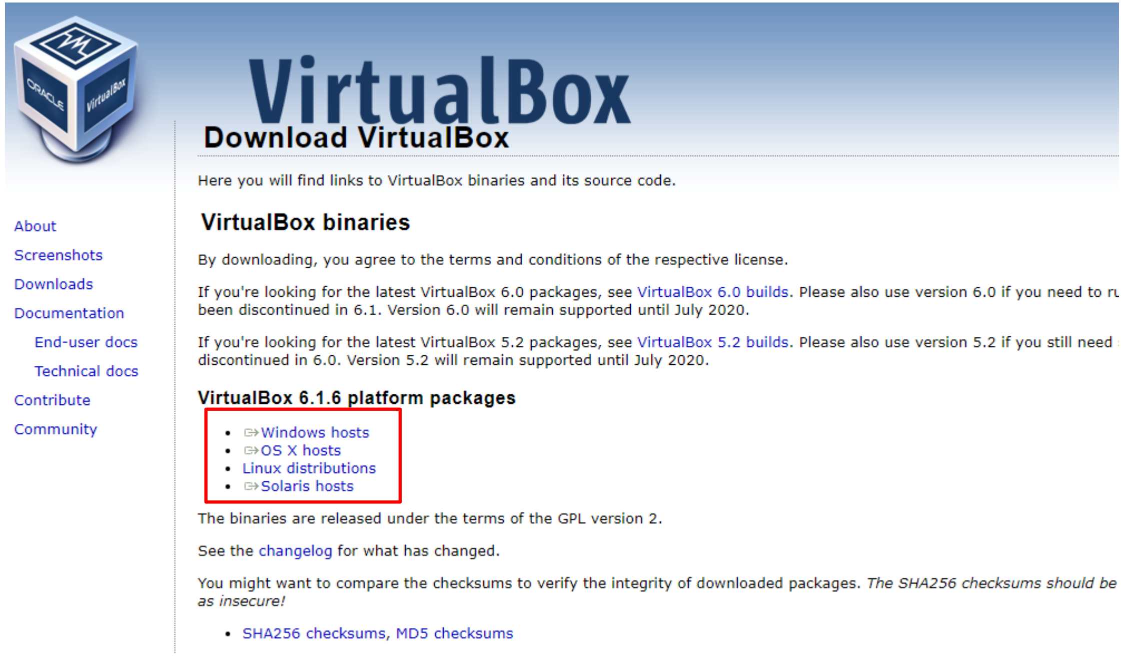
1. Install Virutal Machine

- We recommend to use VirtualBox.

- Download package (https://www.virtualbox.org/wiki/Downloads)

- Select according to you OS

- Install the package



Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see VirtualBox 6.0 builds. Please also use version 6.0 if you need to ru
been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds. Please also use version 5.2 if you still need
discontinued in 6.0. Version 5.2 will remain supported until July 2020.

VirtualBox 6.1.6 platform packages

- ⇨Windows hosts
- ⇨OS X hosts
- Linux distributions
- ⇨Solaris hosts

The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be*
as insecure!

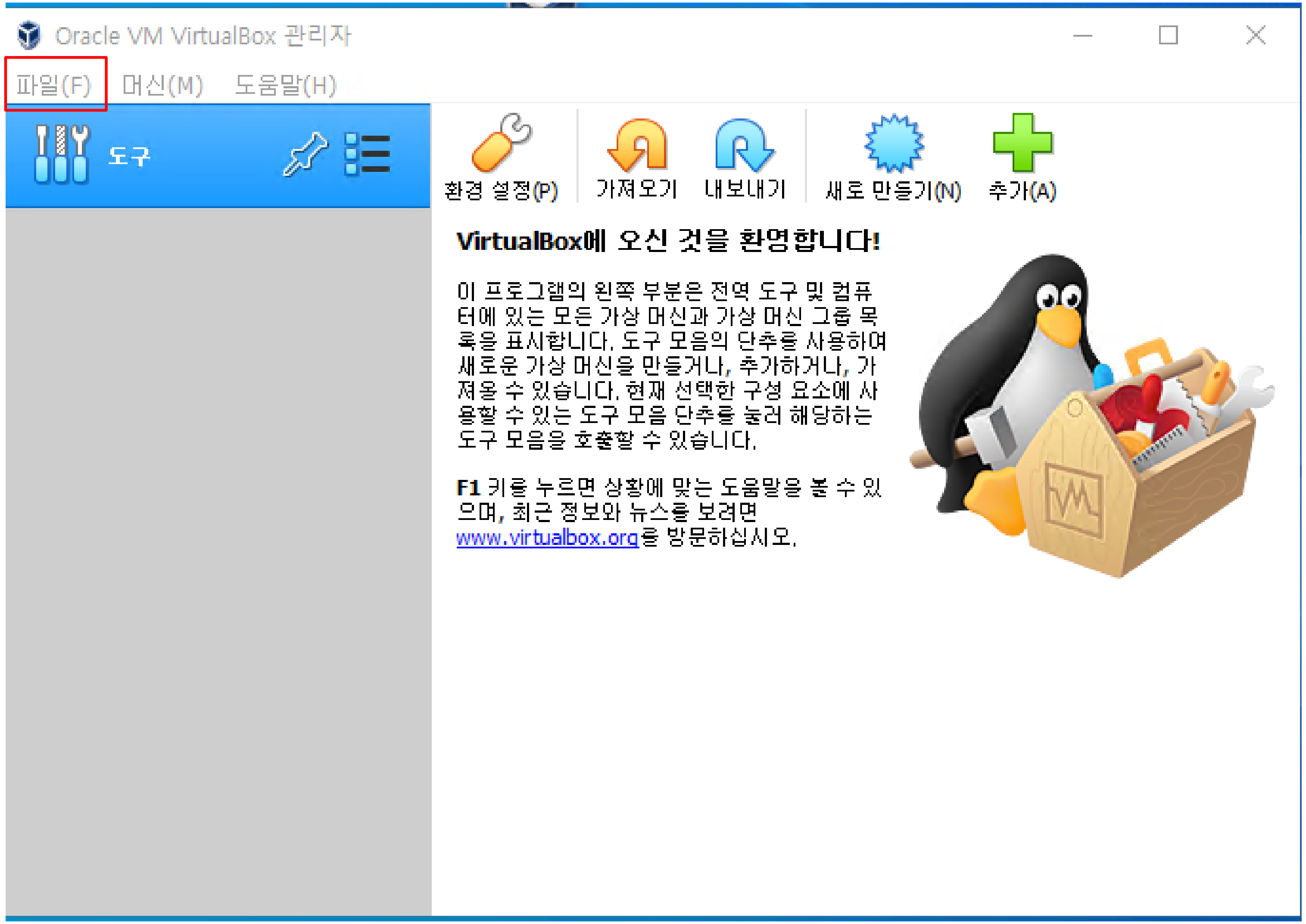
- SHA256 checksums, MD5 checksums

2. Download Mininet image

- Mininet image with provided codes is available at following link. (Recommended) https://drive.google.com/file/d/1LRzzqqEkv498i6TGI-dpnPDCyZUdr2b-/view?usp=sharing

- If you want to download original Mininet image, it is available at following link. (https://github.com/mininet/mininet/releases)

3. Import Virtual System

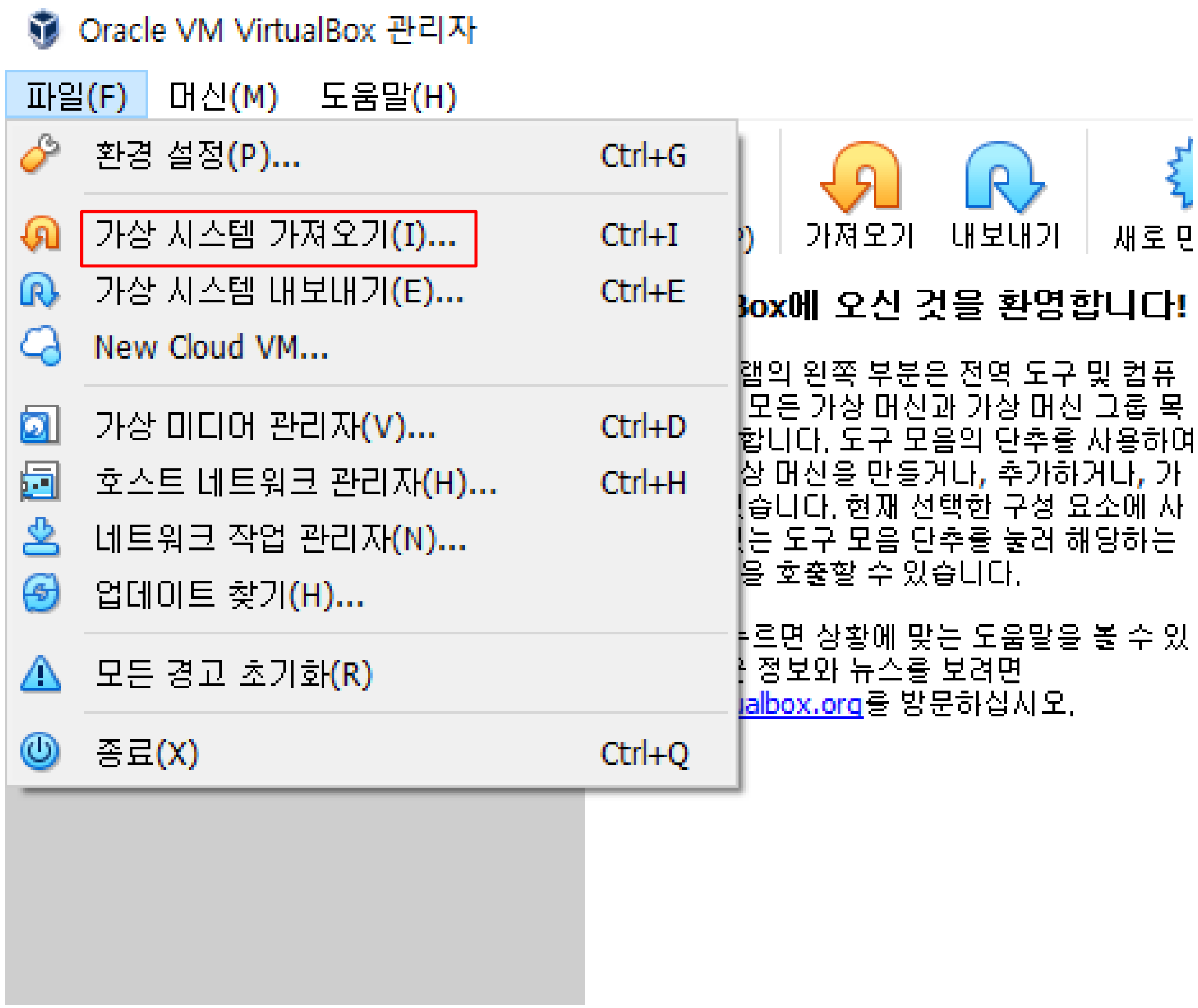


VirtualBox에 오신 것을 환영합니다!

이 프로그램의 왼쪽 부분은 전역 도구 및 컴퓨터에 있는 모든 가상 머신과 가상 머신 그룹 목록을 표시합니다. 도구 모음의 단추를 사용하여 새로운 가상 머신을 만들거나, 추가하거나, 가져올 수 있습니다. 현재 선택한 구성 요소에 사용할 수 있는 도구 모음 단추를 눌러 해당하는 도구 모음을 호출할 수 있습니다.

F1 키를 누르면 상황에 맞는 도움말을 볼 수 있으며, 최근 정보와 뉴스를 보려면 www.virtualbox.org을 방문하십시오.

3. Import Virtual System


Oracle VM VirtualBox 관리자
파일(F)
머신(M)
도움말(H)
환경 설정(P)... Ctrl+G
가상 시스템 가져오기(I)... Ctrl+I
가상 시스템 내보내기(E)... Ctrl+E
New Cloud VM...
가상 미디어 관리자(V)... Ctrl+D
호스트 네트워크 관리자(H)... Ctrl+H
네트워크 작업 관리자(N)...
업데이트 찾기(H)...
모든 경고 초기화(R)
종료(X) Ctrl+Q
)
가져오기
내보내기
새로 만

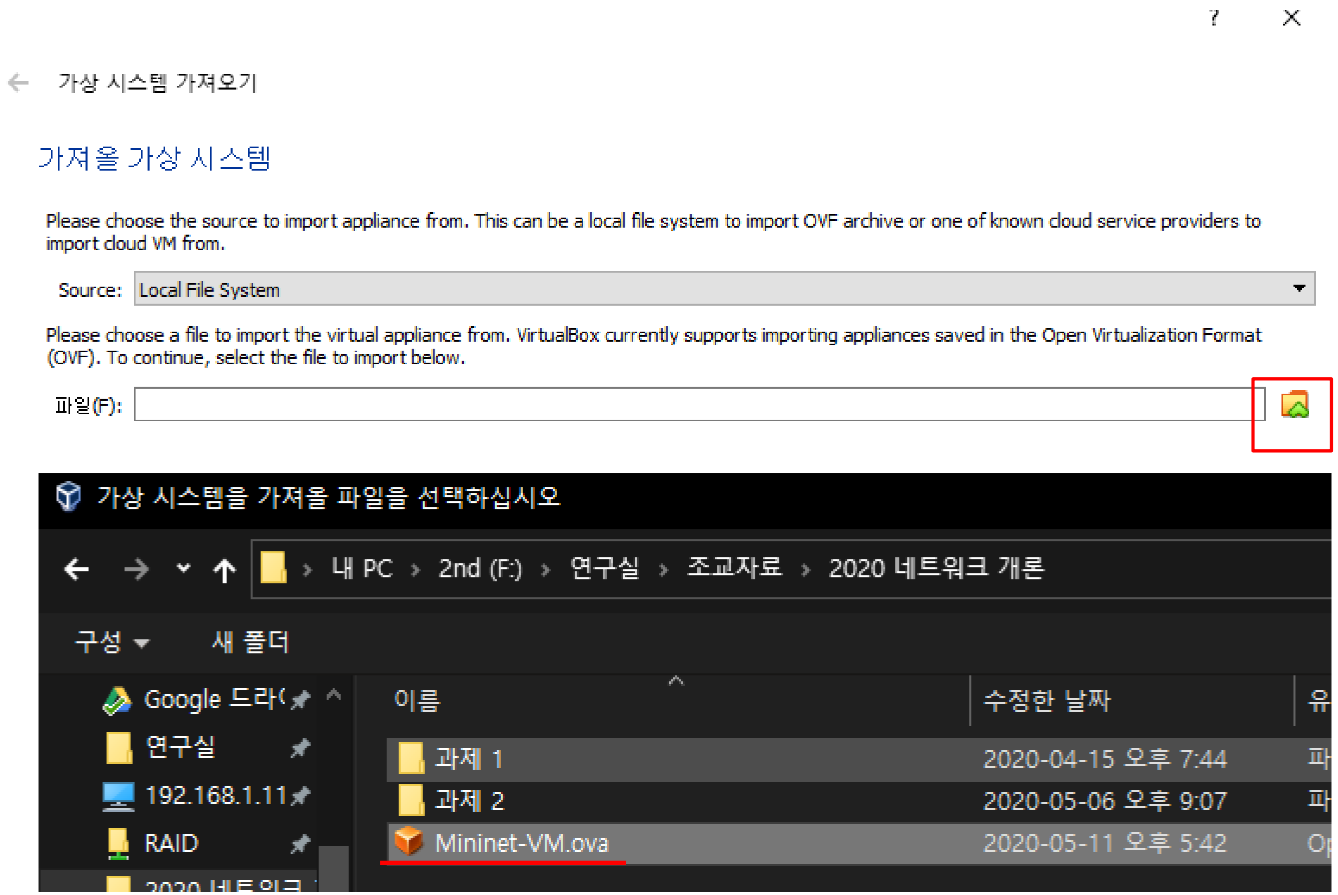
3ox에 오신 것을 환영합니다!

램의 왼쪽 부분은 전역 도구 및 컴퓨
모든 가상 머신과 가상 머신 그룹 목
합니다. 도구 모음의 단추를 사용하여
상 머신을 만들거나, 추가하거나, 가
습니다. 현재 선택한 구성 요소에 사
는 도구 모음 단추를 눌러 해당하는
을 호출할 수 있습니다.

르면 상황에 맞는 도움말을 볼 수 있
정보와 뉴스를 보려면
ualbox.org를 방문하십시오.

3. Import Virtual System

Select Mininet image



가상 시스템 가져오기

가져올 가상 시스템

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

파일(F):

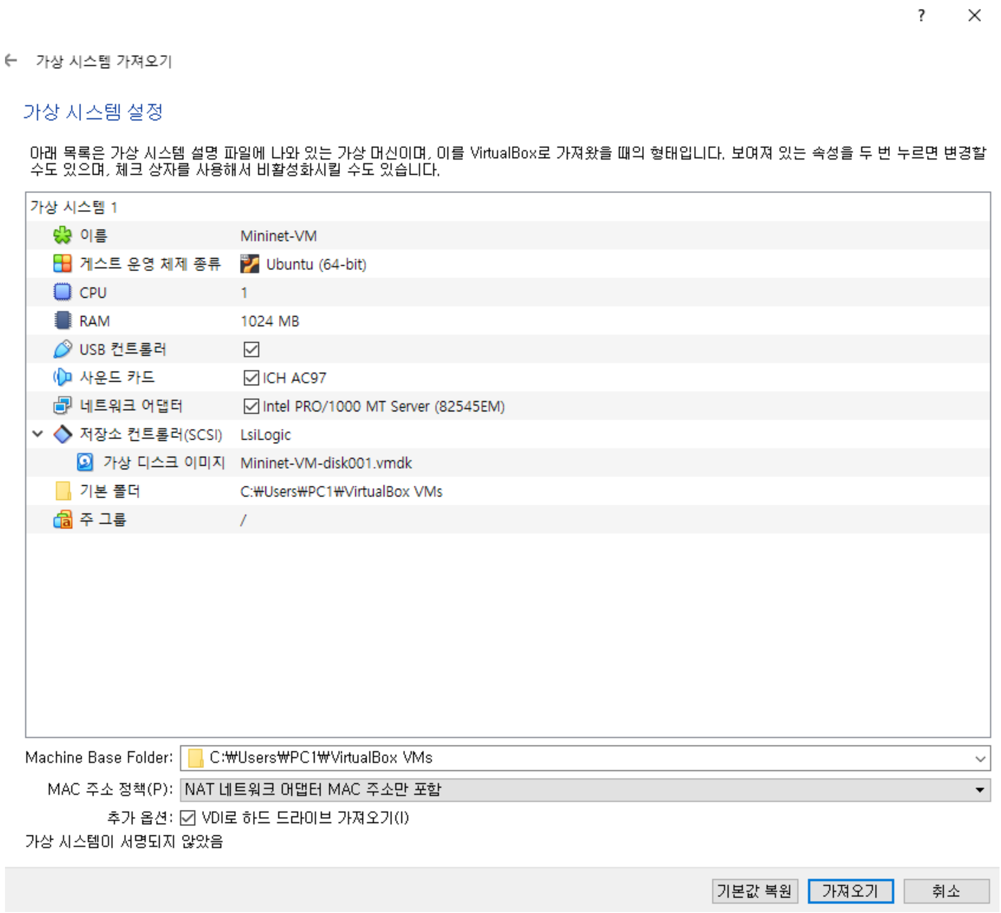
가상 시스템을 가져올 파일을 선택하십시오

내 PC › 2nd (F:) › 연구실 › 조교자료 › 2020 네트워크 개론

이름	수정한 날짜	유
과제 1	2020-04-15 오후 7:44	파
과제 2	2020-05-06 오후 9:07	파
Mininet-VM.ova	2020-05-11 오후 5:42	O

3. Import Virtual System

Continue to create virtual machine



가상 시스템 설정

아래 목록은 가상 시스템 설명 파일에 나와 있는 가상 머신이며, 이를 VirtualBox로 가져왔을 때의 형태입니다. 보여져 있는 속성을 두 번 누르면 변경할 수도 있으며, 체크 상자를 사용해서 비활성화시킬 수도 있습니다.

가상 시스템 1

이름	Mininet-VM
게스트 운영 체제 종류	Ubuntu (64-bit)
CPU	1
RAM	1024 MB
USB 컨트롤러	☑
사운드 카드	☑ ICH AC97
네트워크 어댑터	☑ Intel PRO/1000 MT Server (82545EM)
저장소 컨트롤러(SCSI)	LsiLogic
가상 디스크 이미지	Mininet-VM-disk001.vmdk
기본 폴더	C:₩Users₩PC1₩VirtualBox VMs
주 그룹	/

Machine Base Folder: C:₩Users₩PC1₩VirtualBox VMs

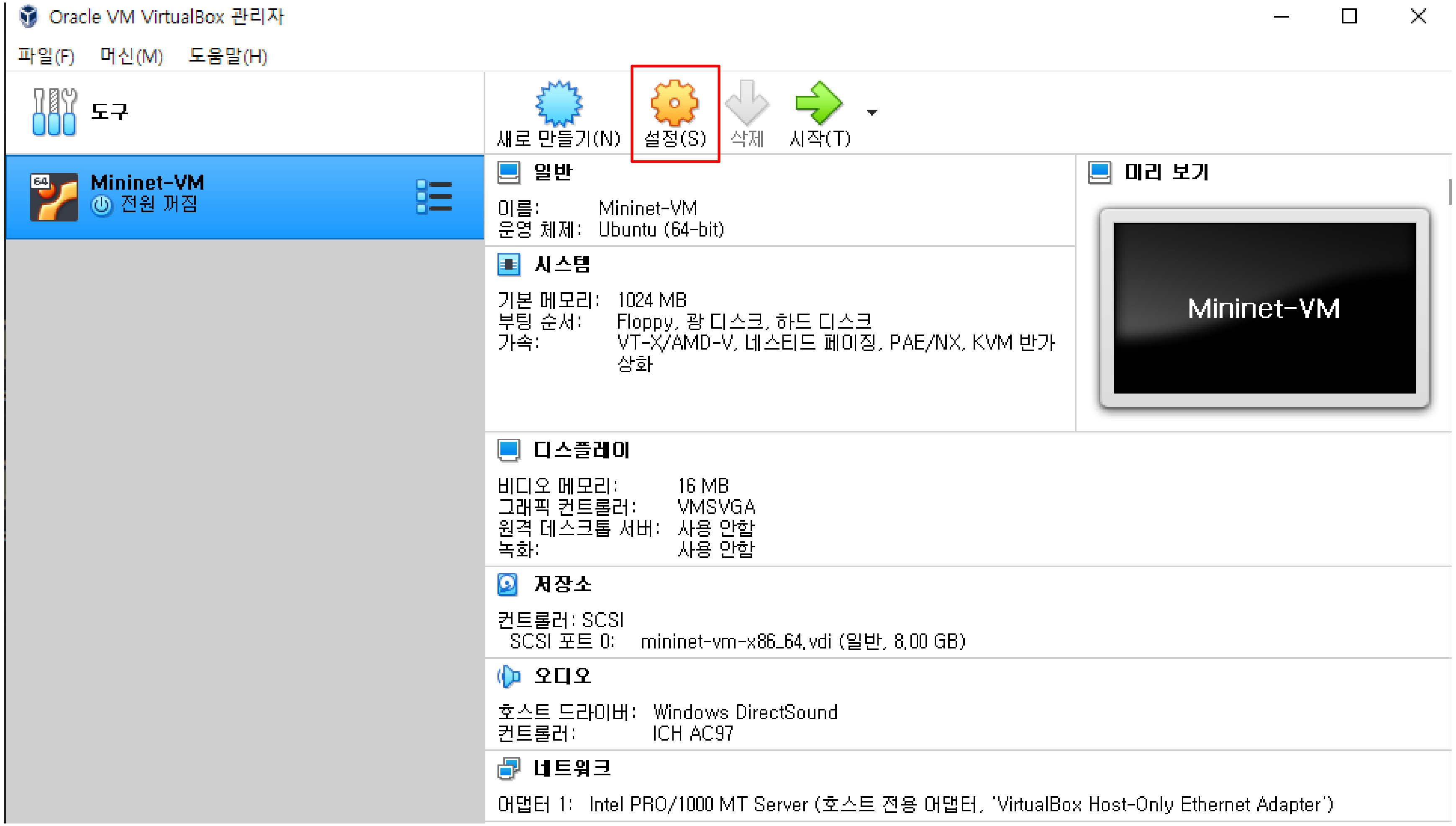
MAC 주소 정책(P): NAT 네트워크 어댑터 MAC 주소만 포함

추가 옵션: ☑ VDI로 하드 드라이브 가져오기(I)

가상 시스템이 서명되지 않았음

4. Setting Virtual Machine

Click “setting”



일반

이름: Mininet-VM
운영 체제: Ubuntu (64-bit)

시스템

기본 메모리: 1024 MB
부팅 순서: Floppy, 광 디스크, 하드 디스크
가속: VT-X/AMD-V, 네스티드 페이징, PAE/NX, KVM 반가상화

디스플레이

비디오 메모리: 16 MB
그래픽 컨트롤러: VMSVGA
원격 데스크톱 서버: 사용 안함
녹화: 사용 안함

저장소

컨트롤러: SCSI
SCSI 포트 0: mininet-vm-x86_64.vdi (일반, 8.00 GB)

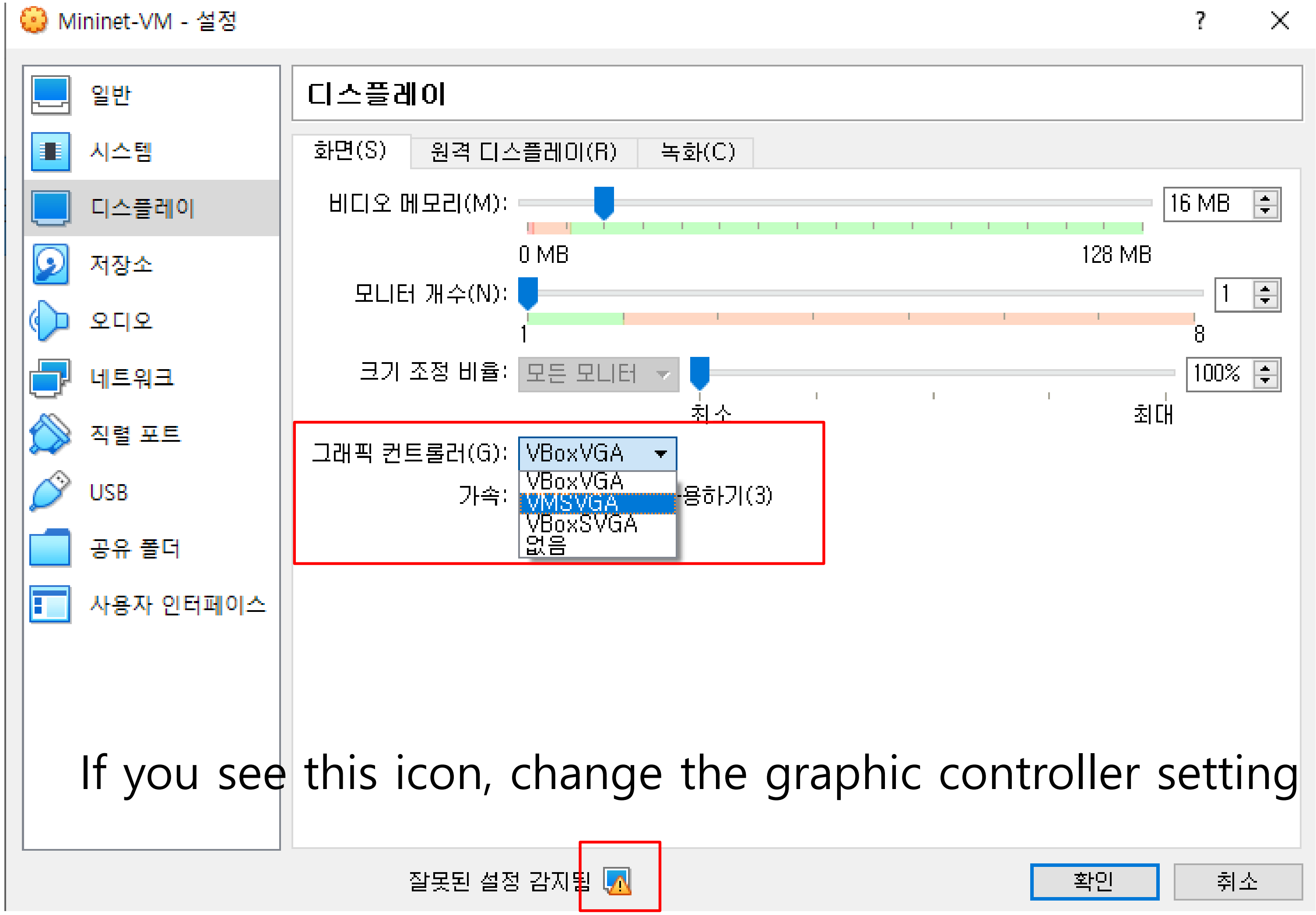
오디오

호스트 드라이버: Windows DirectSound
컨트롤러: ICH AC97

네트워크

어댑터 1: Intel PRO/1000 MT Server (호스트 전용 어댑터, 'VirtualBox Host-Only Ethernet Adapter')

4. Setting Virtual Machine



비디오 메모리(M): 16 MB (0 MB – 128 MB)

모니터 개수(N): 1 (1 – 8)

크기 조정 비율: 모든 모니터 100% (최소 – 최대)

그래픽 컨트롤러(G): VBoxVGA

- VBoxVGA
- VMSVGA
- VBoxSVGA
- 없음

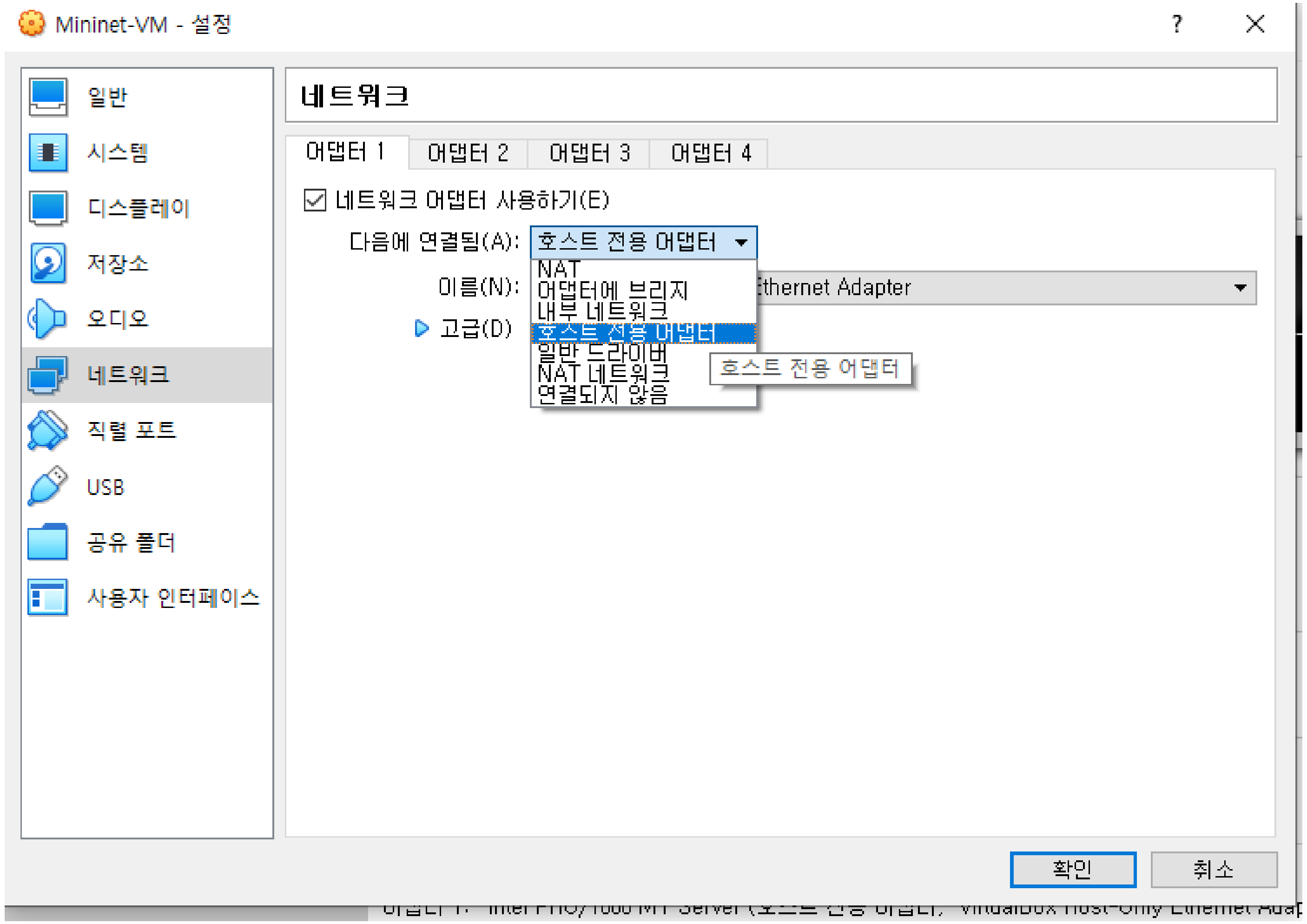
가속: 용하기(3)

잘못된 설정 감지됨 ⚠

If you see this icon, change the graphic controller setting to VMSVGA

4. Setting Virtual Machine

Change network from NAT(by default) to Host-only-adapter



Mininet-VM - 설정

네트워크

어댑터 1

☑ 네트워크 어댑터 사용하기(E)

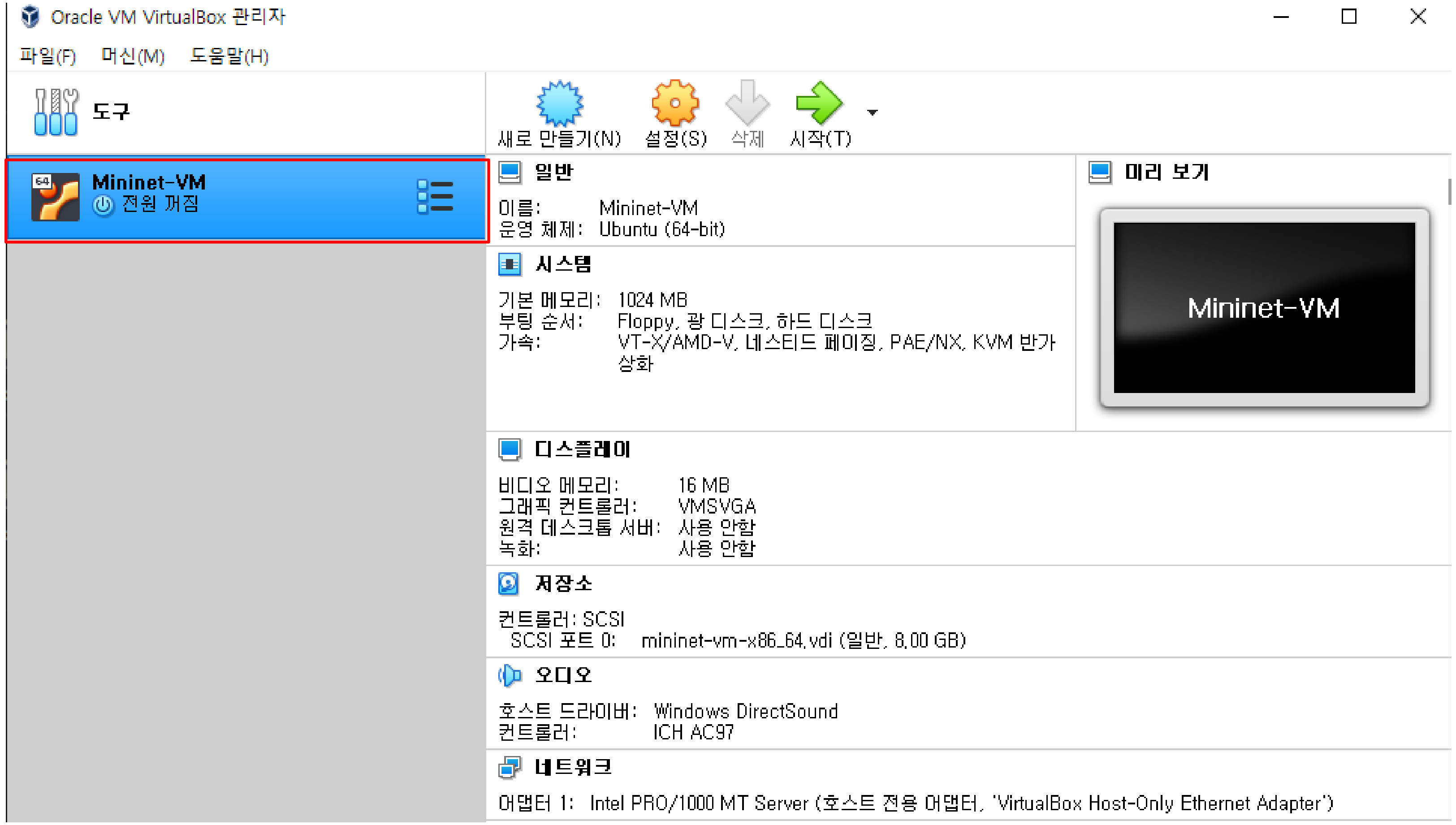
다음에 연결됨(A): 호스트 전용 어댑터

이름(N): Ethernet Adapter

어댑터 1: Intel PRO/1000 MT Server (호스트 전용 어댑터, VirtualBox Host-Only Ethernet Adap

5. Start Mininet-VM

Start VM by double clicking it



일반

이름: Mininet-VM
운영 체제: Ubuntu (64-bit)

시스템

기본 메모리: 1024 MB
부팅 순서: Floppy, 광 디스크, 하드 디스크
가속: VT-X/AMD-V, 네스티드 페이징, PAE/NX, KVM 반가상화

디스플레이

비디오 메모리: 16 MB
그래픽 컨트롤러: VMSVGA
원격 데스크톱 서버: 사용 안함
녹화: 사용 안함

저장소

컨트롤러: SCSI
SCSI 포트 0: mininet-vm-x86_64.vdi (일반, 8.00 GB)

오디오

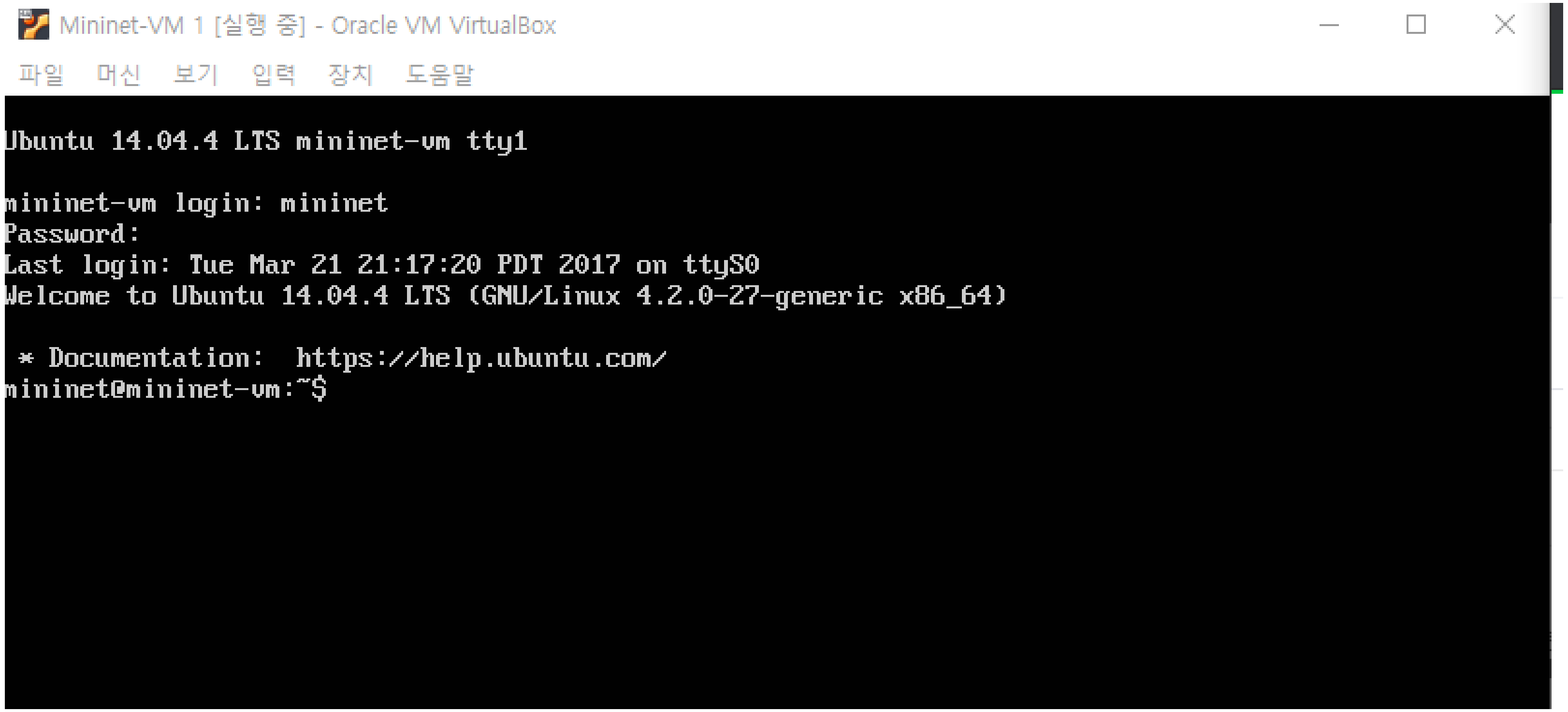
호스트 드라이버: Windows DirectSound
컨트롤러: ICH AC97

네트워크

어댑터 1: Intel PRO/1000 MT Server (호스트 전용 어댑터, 'VirtualBox Host-Only Ethernet Adapter')

6. Login

Both ID and password is ‘mininet’


Mininet-VM 1 [실행 중] - Oracle VM VirtualBox
파일 머신 보기 입력 장치 도움말

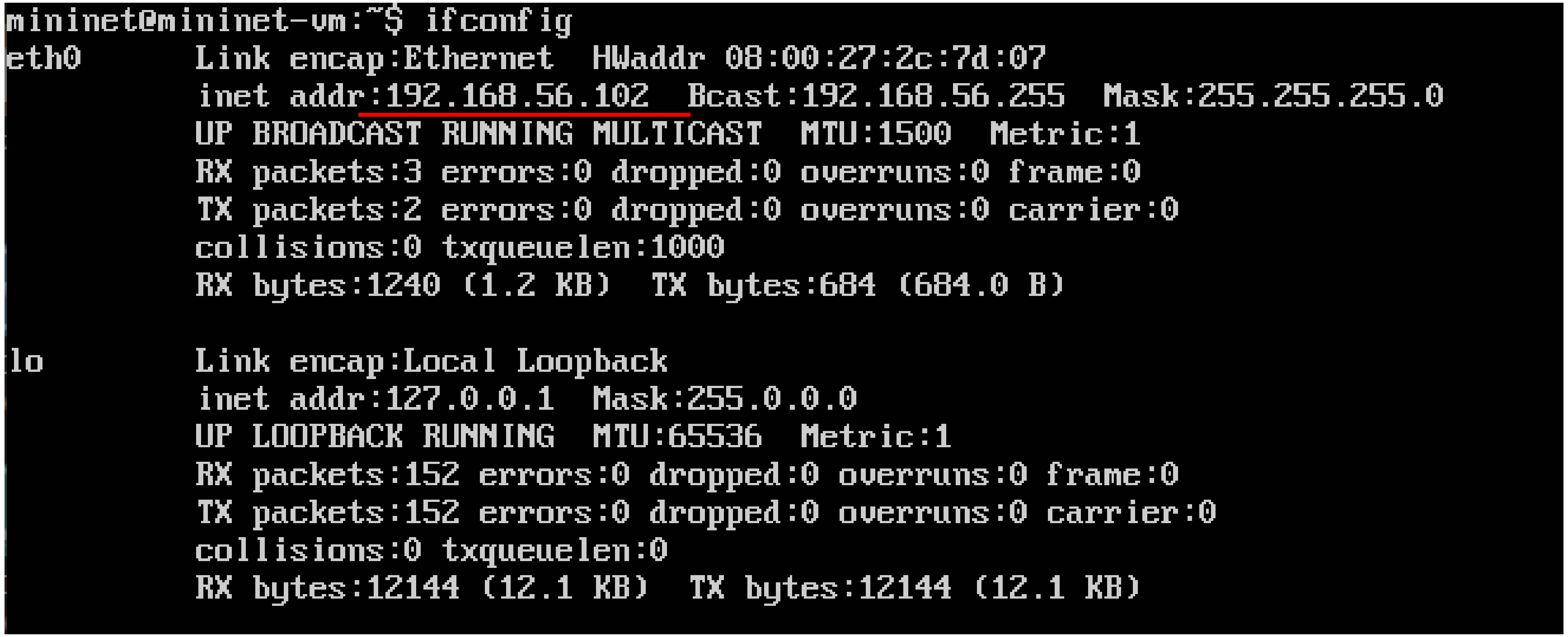
```
Ubuntu 14.04.4 LTS mininet-vm tty1

mininet-vm login: mininet
Password:
Last login: Tue Mar 21 21:17:20 PDT 2017 on ttyS0
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$
```

7. Check IP

Use “ifconfig” to check ip address of Mininet machine



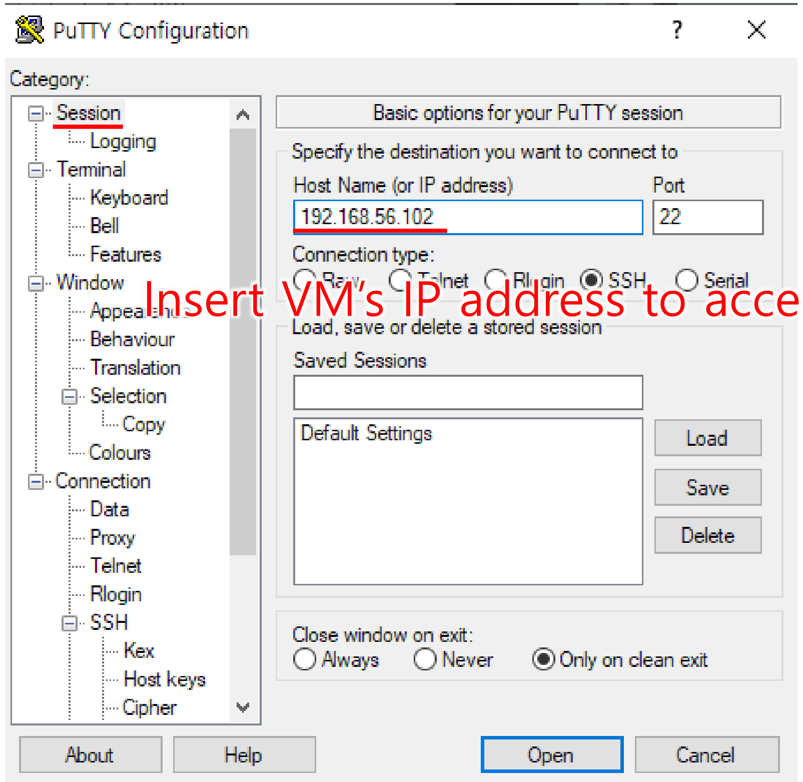
```
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:2c:7d:07
          inet addr:192.168.56.102  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1240 (1.2 KB)  TX bytes:684 (684.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12144 (12.1 KB)  TX bytes:12144 (12.1 KB)
```

How to use Mininet

Use SSH and SFTP

- SSH is for access terminal and GUI platform(wireshark etc).
- We recommend to use “Putty” and “Xming” to use SSH.
- Turn on the the Xming.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.56.102

Port: 22

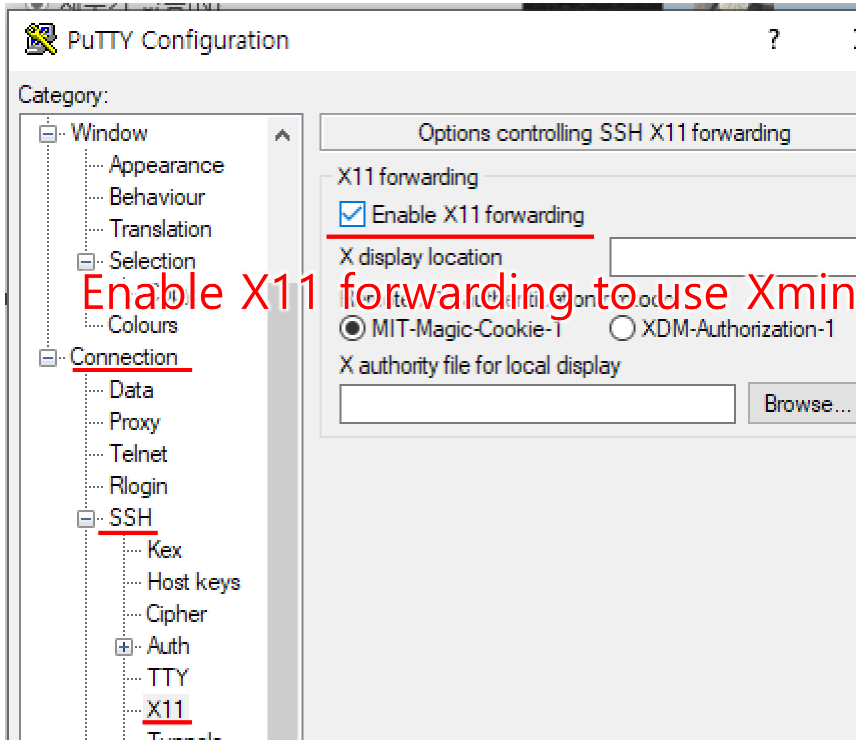
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Insert VM’s IP address to access



PuTTY Configuration

Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location:

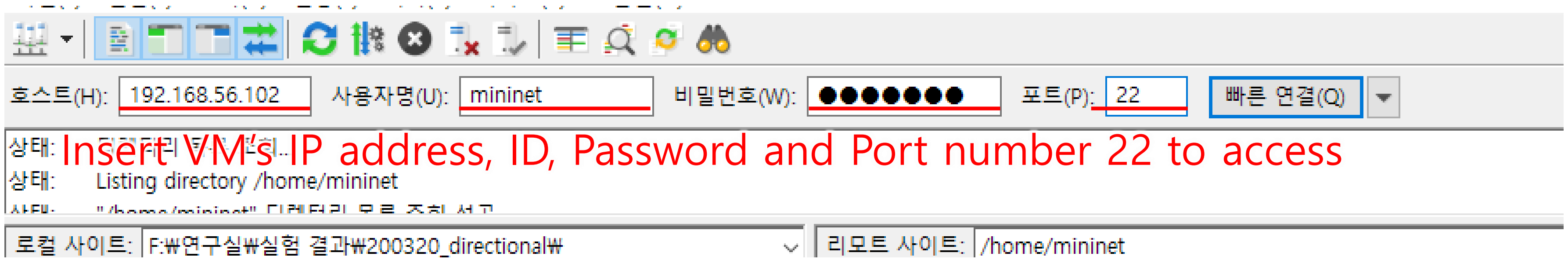
Remote X11 authentication protocol: ● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display:

Enable X11 forwarding to use Xming for GUI

Use SSH and SFTP

- SFTP is for access file system via SSH protocol
- We recommend to use “Filezilla” to use SFTP



호스트(H): 192.168.56.102 사용자명(U): mininet 비밀번호(W): ●●●●●●●● 포트(P): 22 빠른 연결(Q)

```
상태:    디렉터리 목록 조회...
상태:    Listing directory /home/mininet
상태:    "/home/mininet" 디렉터리 목록 조회 성공
```

로컬 사이트: F:₩연구실₩실험 결과₩200320_directional₩

리모트 사이트: /home/mininet

Insert VM’s IP address, ID, Password and Port number 22 to access

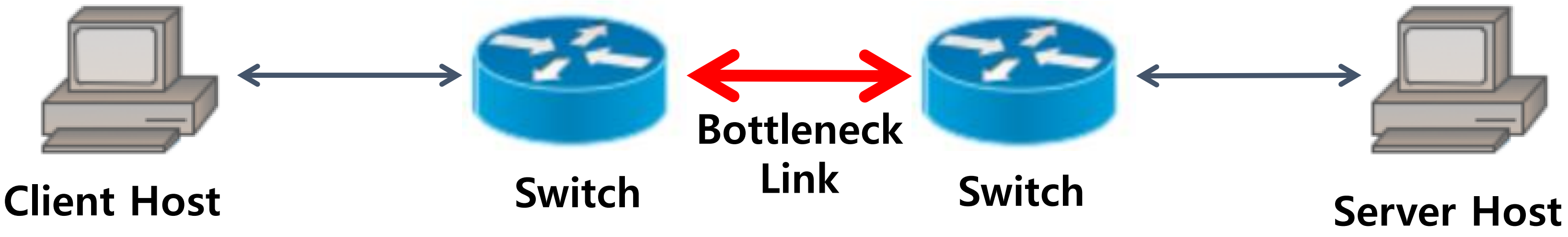
Mininet with python API

- We will give you the Basic python code to use the Mininet.

- Put the code file in Mininet VM and execute it with python.

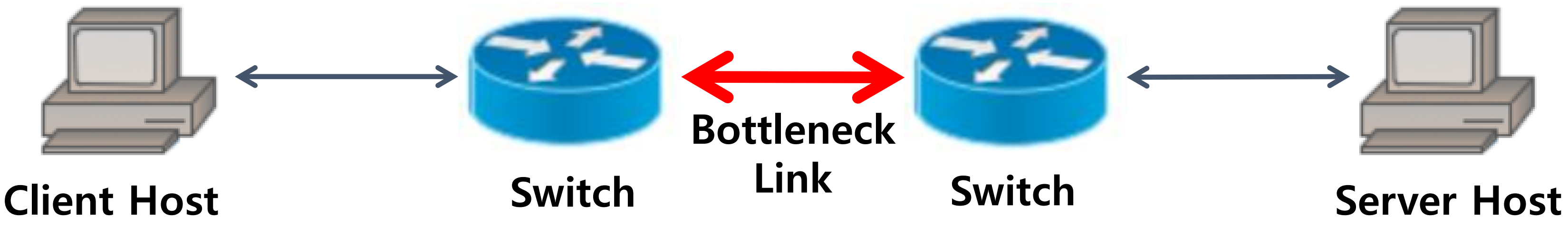
"sudo python execute_mn.py <window Size, source Filename, Destination Filename>"

- This will create 2 Hosts, 2 Switches and 3 Link. The link in the middle of Switches is bottleneck link that we are going to manipulate Bandwidth, Delay, Loss rate.



- Please read the comment in code first. Modify it as you want.

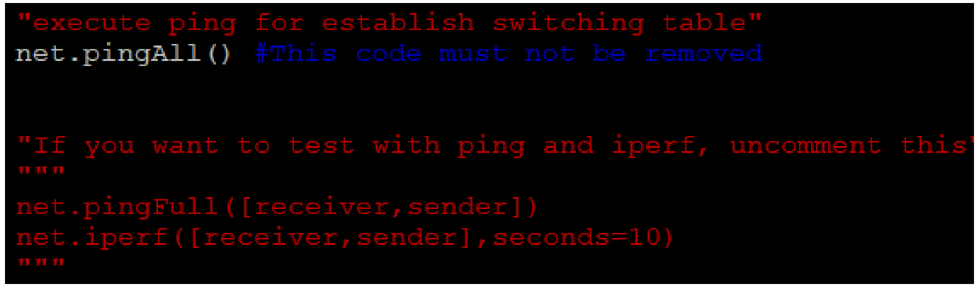
Bottleneck Link



- Bottleneck link is the link in between two switches which are connected to each host.
- You can change three properties of the link described below
  - Bandwidth : How many bits can be transferred through link
  - Delay : How much time will be delayed until the packet start to transfer (one-way propagation delay)
  - Loss Rate : Probability of the random packet loss

Ping, Iperf test

- You can check the connectivity and network performance by ping and iperf. In “execute_my.py” you can use these by uncomment the code below.



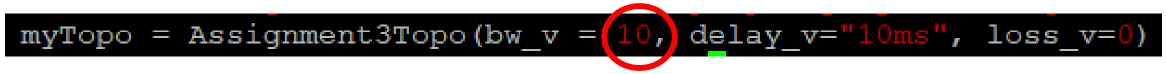
```
"execute ping for establish switching table"
net.pingAll() #This code must not be removed


"If you want to test with ping and iperf, uncomment this"
"""
net.pingFull([receiver,sender])
net.iperf([receiver,sender],seconds=10)
"""
```

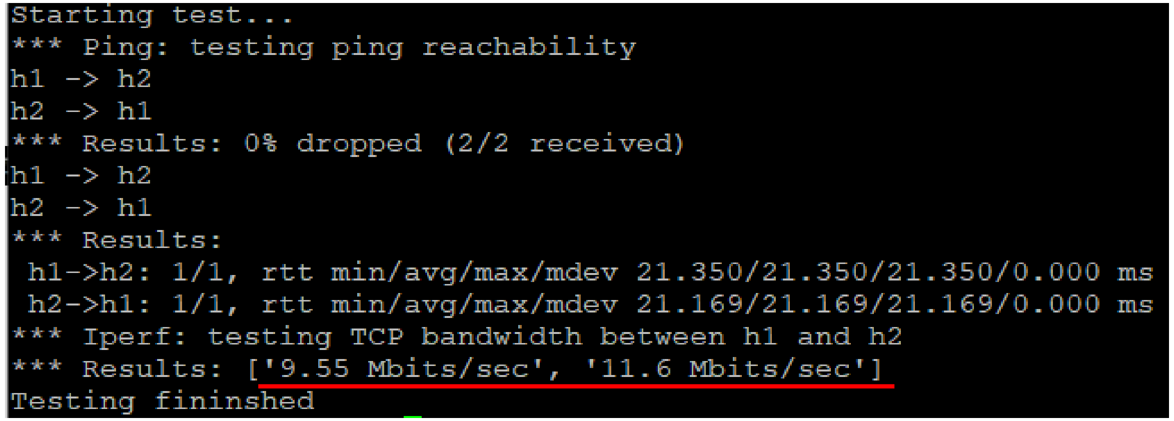
- **“Ping”** is used to test reachability of a host on an IP network. The result of Ping shows packet loss and RTT.
- **“Iperf”** is used to test the performance of network between two hosts. The result of iperf shows bandwidth between two hosts using TCP.
- **Caution:** “net.pingAll()” must not be removed because of switching table initialize.

Bandwidth test

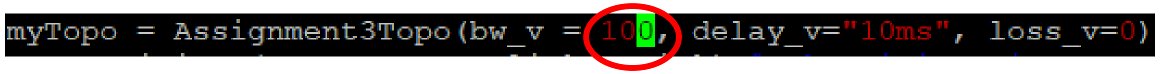
- You can check the bandwidth by the iperf result.



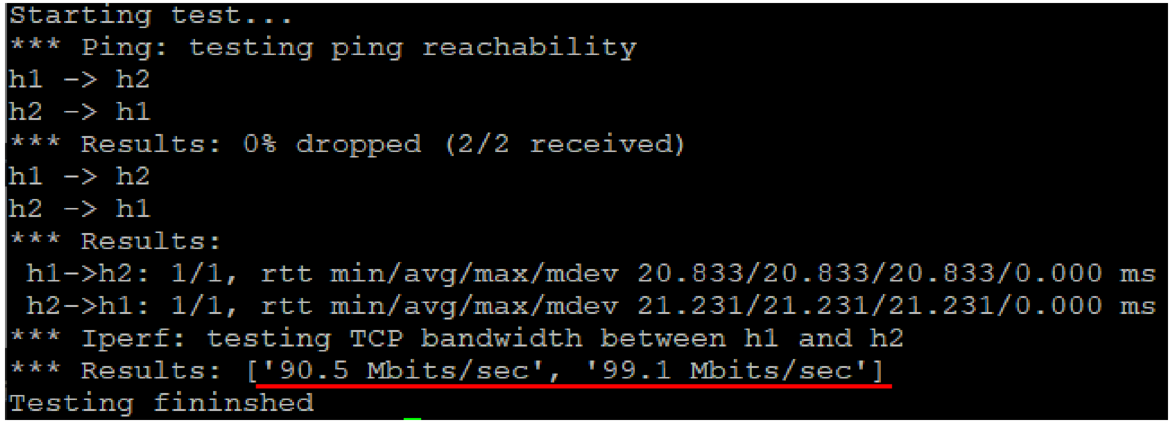
```
myTopo = Assignment3Topo(bw_v = 10, delay_v="10ms", loss_v=0)
```



```
Starting test...
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
h1 -> h2
h2 -> h1
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 21.350/21.350/21.350/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 21.169/21.169/21.169/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.55 Mbits/sec', '11.6 Mbits/sec']
Testing fininshed
```



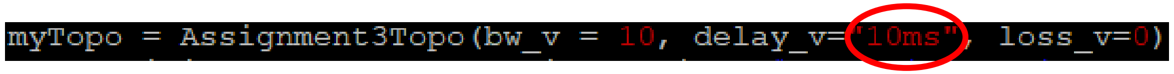
```
myTopo = Assignment3Topo(bw_v = 100, delay_v="10ms", loss_v=0)
```



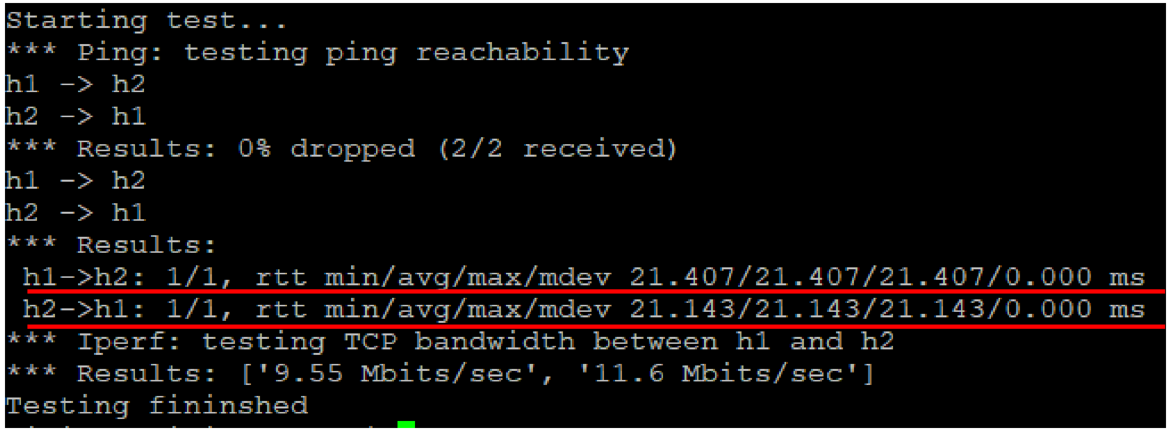
```
Starting test...
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
h1 -> h2
h2 -> h1
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 20.833/20.833/20.833/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 21.231/21.231/21.231/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['90.5 Mbits/sec', '99.1 Mbits/sec']
Testing fininshed
```

Delay test

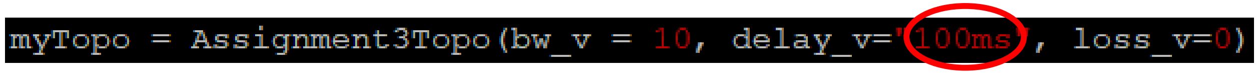
- You can check the delay by the ping result.



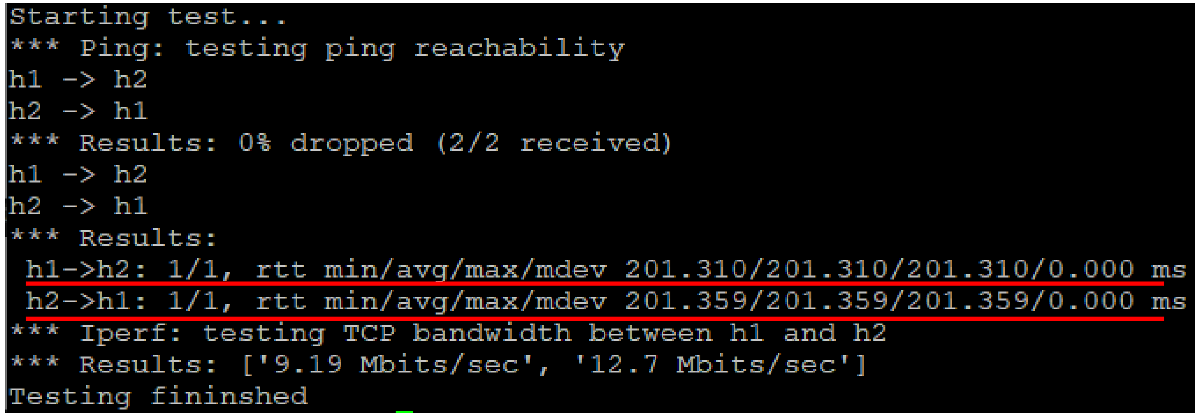
```
myTopo = Assignment3Topo(bw_v = 10, delay_v="10ms", loss_v=0)
```



```
Starting test...
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
h1 -> h2
h2 -> h1
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 21.407/21.407/21.407/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 21.143/21.143/21.143/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.55 Mbits/sec', '11.6 Mbits/sec']
Testing finished
```



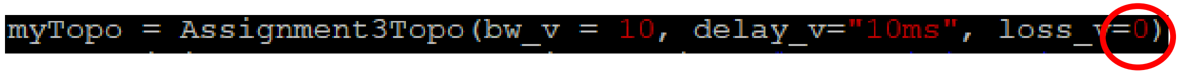
```
myTopo = Assignment3Topo(bw_v = 10, delay_v='100ms', loss_v=0)
```



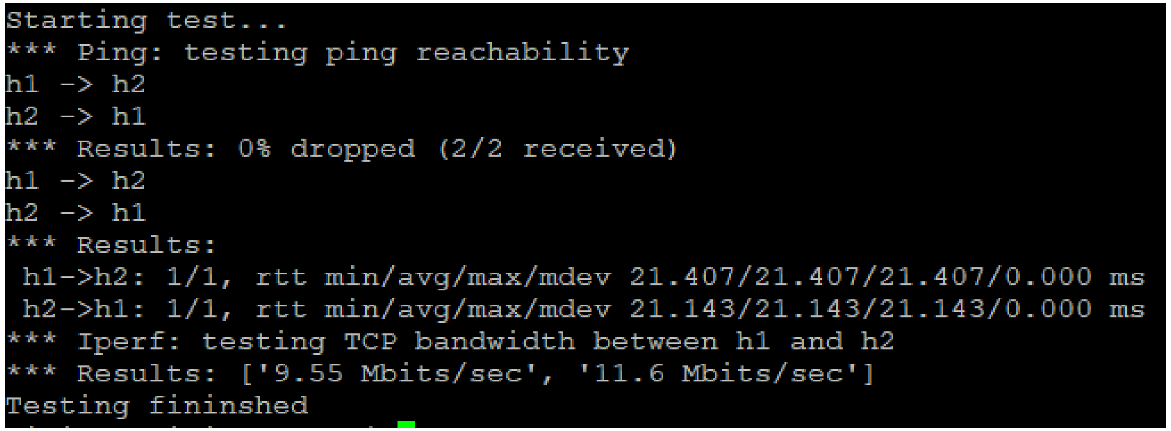
```
Starting test...
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
h1 -> h2
h2 -> h1
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 201.310/201.310/201.310/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 201.359/201.359/201.359/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.19 Mbits/sec', '12.7 Mbits/sec']
Testing finished
```

Loss Rate test

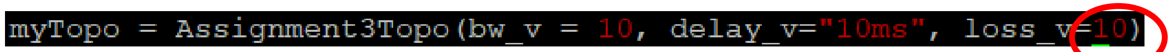
- You can check the bandwidth by the ping result.



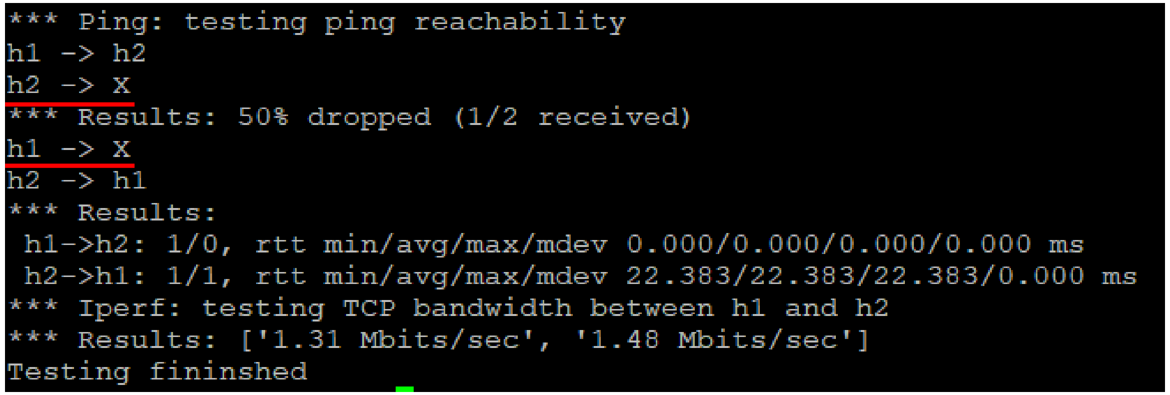
```
myTopo = Assignment3Topo(bw_v = 10, delay_v="10ms", loss_v=0))
```



```
Starting test...
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
h1 -> h2
h2 -> h1
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 21.407/21.407/21.407/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 21.143/21.143/21.143/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.55 Mbits/sec', '11.6 Mbits/sec']
Testing finished
```



```
myTopo = Assignment3Topo(bw_v = 10, delay_v="10ms", loss_v=10)
```



```
*** Ping: testing ping reachability
h1 -> h2
h2 -> X
*** Results: 50% dropped (1/2 received)
h1 -> X
h2 -> h1
*** Results:
 h1->h2: 1/0, rtt min/avg/max/mdev 0.000/0.000/0.000/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 22.383/22.383/22.383/0.000 ms
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['1.31 Mbits/sec', '1.48 Mbits/sec']
Testing finished
```

Caution: Ping only sent one packet. So packet loss may not be occurred depend on the probability.

Clearing Mininet

- If the program was not terminated successfully(maybe because of error or something) please type following order first.

  ```
  “sudo mn –c”
  ```

  This will clear the Mininet.

Q&A

- If you want to know more about the Mininet, use Mininet official website (http://mininet.org/)

- If you have any question, please use google sheet.