

CSCI262

LAB 1 - SETUP LAB ENVIRONMENT

Description

This document instructs you to set up the attacking lab environment on your computers based on VirtualBox's virtualisation software. We need to **download two files: VirtualBox and OVA file of a virtual machine (VM)**. After installing VirtualBox, we import the ova file of the virtual machine to VirtualBox (**name: cyberlab.ova**) to run the following labs. You will be familiar with docker technology.

Learning Outcomes

By doing this hands-on configuration, you should be able to:

1. Understand the virtualisation platform based on Virtual Box
2. Practice how to import a virtualisation system file (ova file) to build a virtual machine
3. Configure Network for VM, and check and debug network environment.
4. Get familiar with Docker technology and practice basic Docker commands.
5. Finish 3 tasks

Setup Lab Environment

Table of Contents

Description.....	1
Learning Outcomes.....	1
1. Download and Install VirtualBox.....	3
2. Download the Virtual Machine (OVA file)	3
3. Configuring Network Base	4
3.1. Network topology.....	4
3.2. Configure NAT Network on Virtual Box	5
4. Run CyberLab VM on Virtual Box	7
4.1. Import cyberlab VM and start the VM.....	7
Step 1: Import the VM	7
Step 2: Start the VM	11
TASK 1:.....	14
4.2 Checking the network from the VM to the Internet.....	14
TASK 2:.....	14
4.3 Working with container technique	15
TASK 3:.....	16
Appendix A: Start the VM and Take Snapshot	17
Appendix B: Stop the VM	17
Appendix C: Creating a Shared Folder	18

Login Information:

We will use the following account to log in to the virtual machine:

- ✓ Account ID: **seed**
- ✓ Password : **dees**

Setup Lab Environment

1. Download and Install VirtualBox

Please download a new version of Virtual Box from this official website link <https://www.virtualbox.org/wiki/Downloads>

Depending on your computer operating system (macOS, Windows, or Linux), you need to download an appropriate version.

Then, please follow this guide to install Virtual Box. This link instructs you on how to install the old Virtual Box version 6.1. However, it has been no change in installing process for the latest version.

<https://www.wikihow.com/Install-VirtualBox>

Attention to the hardware requirements: to run the virtual machine, your computer or laptop has at least 8GB of RAM, and the processor (CPU) may have two or four cores. For the disk, your computer's hard disk should have an available space of 15GB.

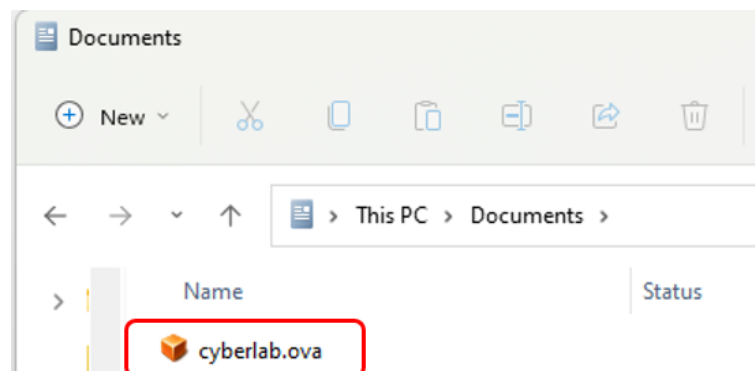
You can use a USB 3.0+ Flash drive or external SSD with at least 128GB if you want to practice this VM on School Lab or other computers.

2. Download the Virtual Machine (OVA file)

The ova file of the virtual machine (VM) is uploaded to the Cloudstor or Ondrive.

The link to download the OVA file is provided on the Moodle site.

So, now you have the ova file of the VM.



3. Configuring Network Base

3.1. Network topology

We all know that networking is critical to the IT and Cybersecurity system. It is like our transportation system. The road is like the link between two computers in the network system. It can be one way or two ways. It depends on the conditions and purposes of the designed Network. In our Network, we have three virtual machines (**VM**). These **VMs** must communicate with each other and access the Internet via a host computer. In some lab tasks, the **VM** needs to access the Internet to install several tools or get information about the target. Also, the lab simulates the attacking system among three pillars attacker, client, and server. So, they have to communicate with each other. (*Please see Figure 1. Lab Network Topology*)

We will configure the Network on Virtual Box named “**NAT Network**” or “**NAT service**” to set up this Network. We will configure the **NAT Network** with the range **10.0.2.0/24**. The virtual box engine in the host computer plays a role as a software-defined gateway (10.0.2.1). It will transfer packets between **VMs** and the **NAT Network** to the Internet.

To understand each kind of network setting, including **NAT**, **NAT Network**, **Host-Only**, and **Bridged**, you can refer to this link below.

<https://www.virtualbox.org/manual/ch06.html>

Setup Lab Environment

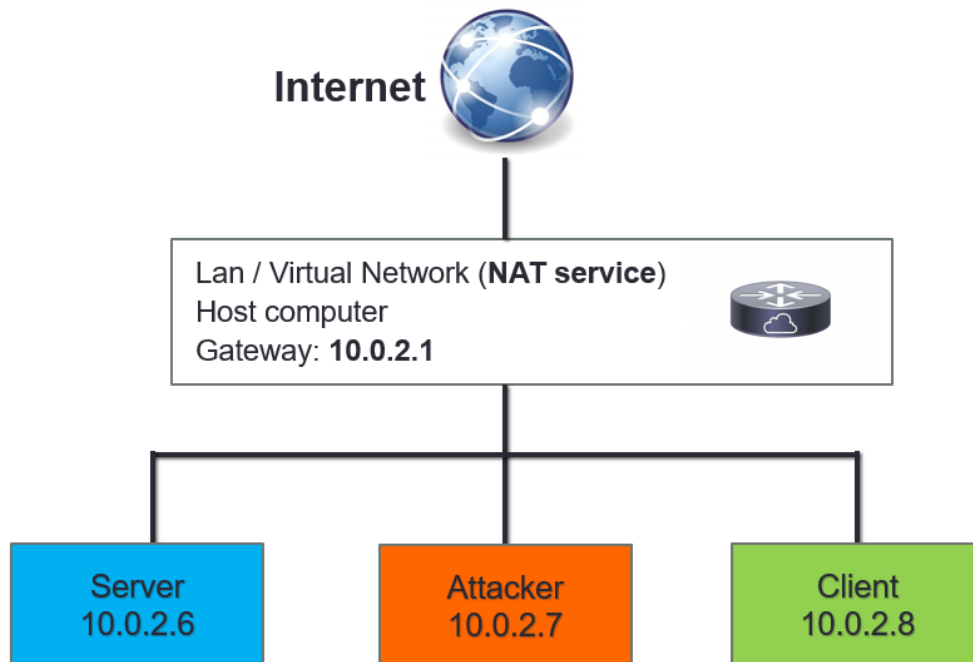


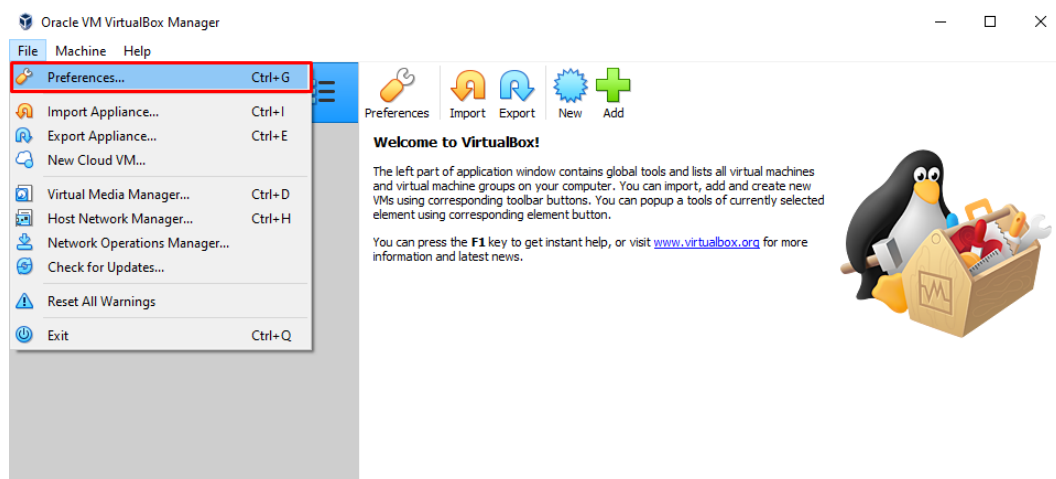
Figure 1. Lab Network Topology

Note: we will use the Docker technique so that the Network Topology may change with different IPs. However, the model will have three machines: Server, Attacker, and Client.

3.2. Configure NAT Network on Virtual Box

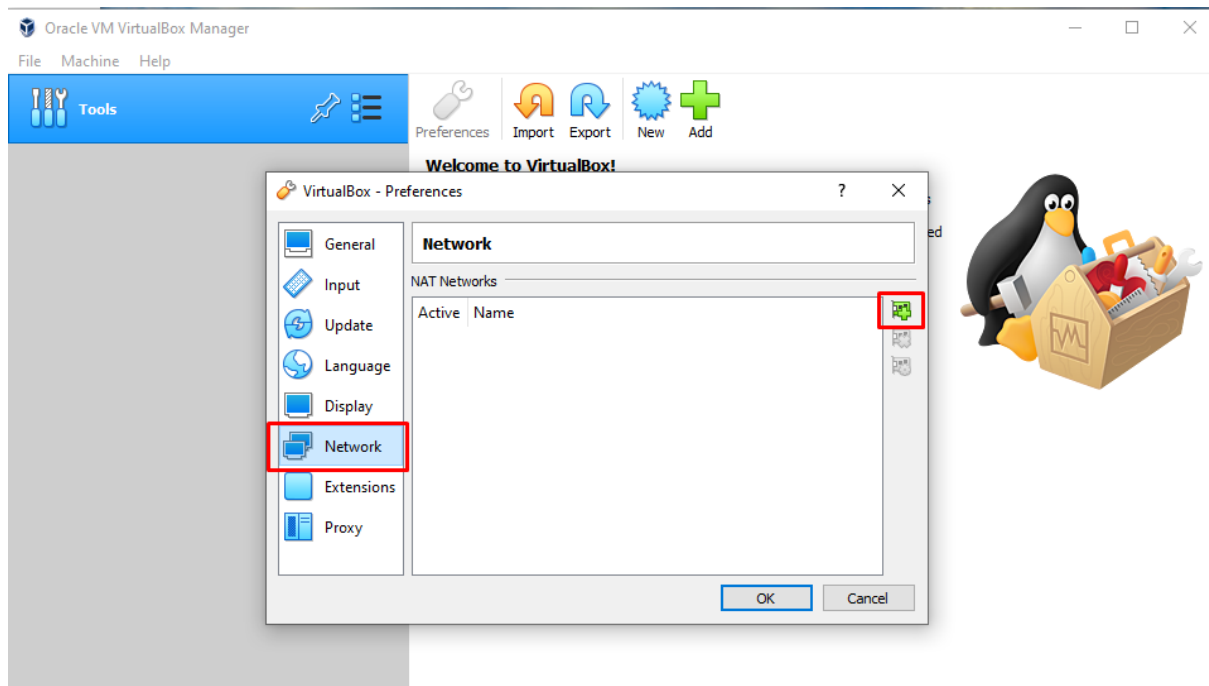
To configure NAT Network, please follow the steps:

- Open Virtual Box, click **File >> Preferences...**

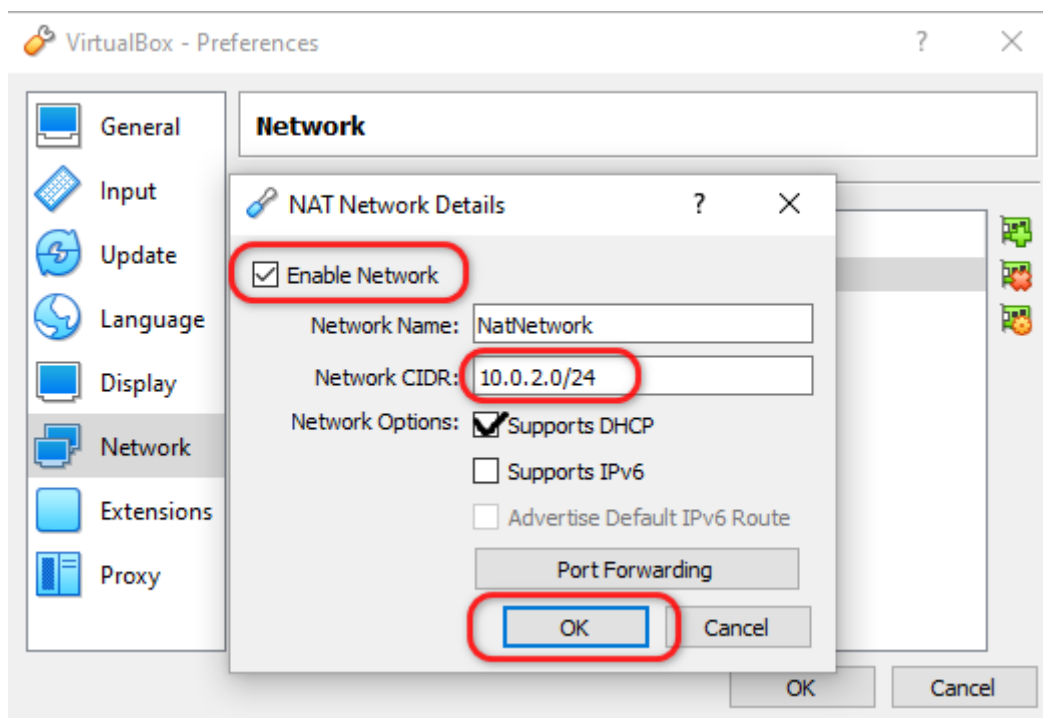


Setup Lab Environment

- Choose **Network** >> Click the plus symbol from **NAT Networks** panel

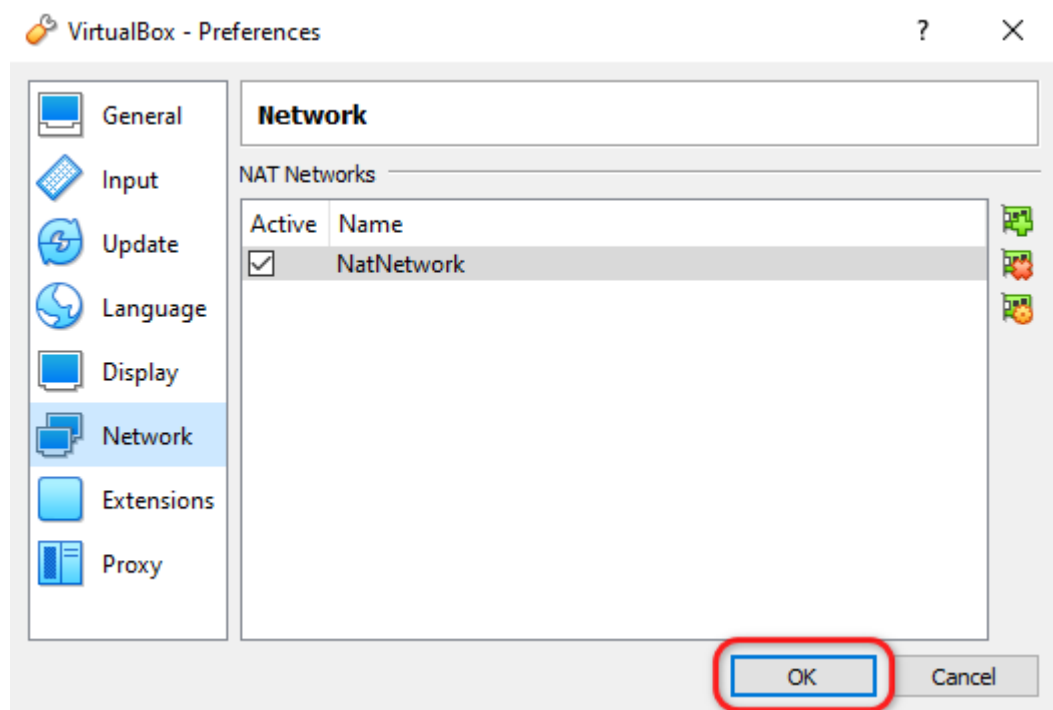


- Configure as shown below: Network Name: leave **NatNetwork** as default, **Network CIDR**: 10.0.2.0/24; **Enable supports DHCP** because we will use **dynamic IP**.



Setup Lab Environment

Then click “OK” to finish setting up **NAT Network** (or NAT Service) on Virtual Box.



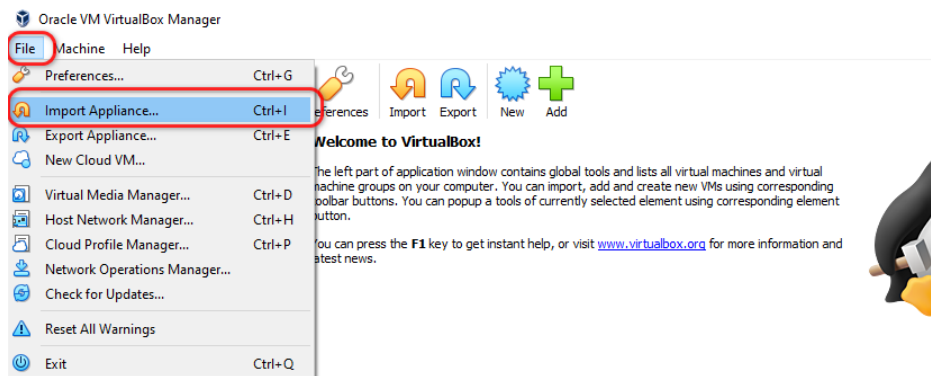
4. Run CyberLab VM on Virtual Box

4.1. Import cyberlab VM and start the VM

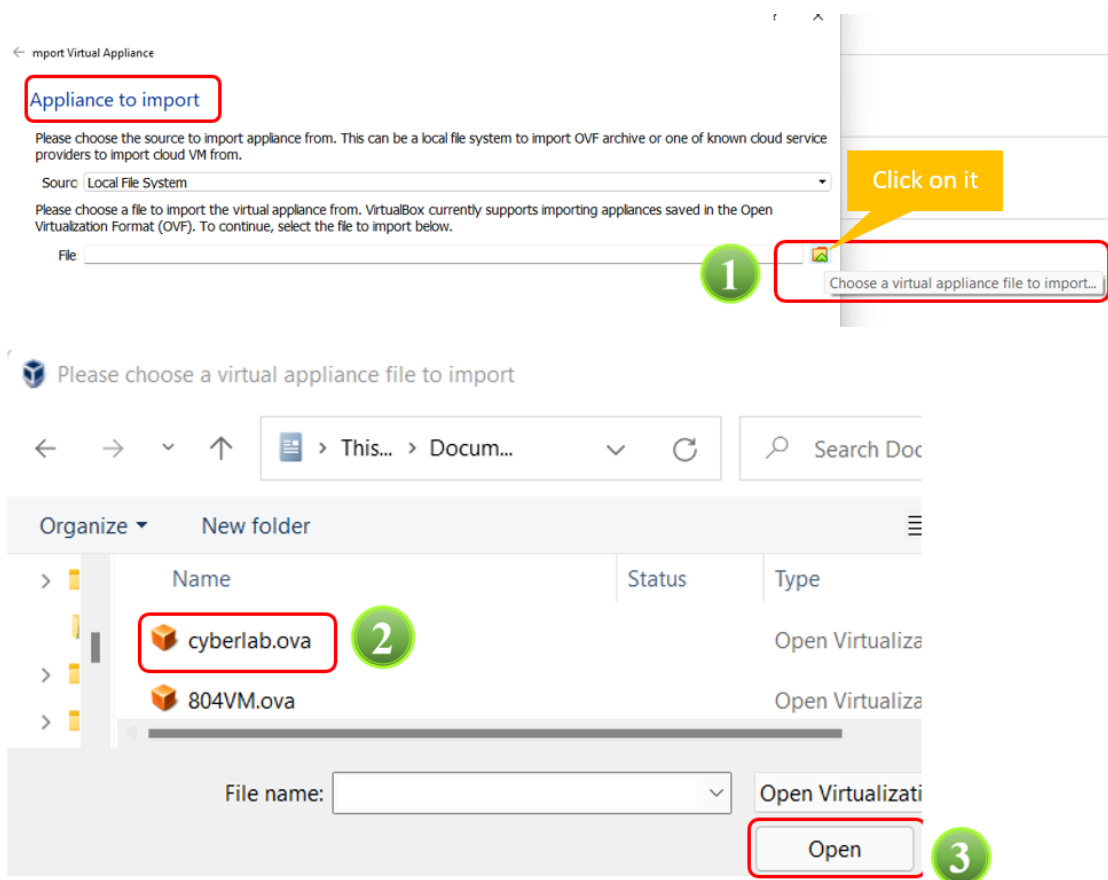
Step 1: Import the VM

To import the VM to VirtualBox, goes to VirtualBox and navigates to **File >> Import Appliance**

Setup Lab Environment



Then, you need to browse the location of the ova file of the VM.



Open the file **cyberlab.ova**, then click **Next**

Setup Lab Environment

← Import Virtual Appliance

Appliance to import

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.

cyberlab.ova

Expert Mode Next Cancel

Note: Before clicking Import

If you use your own flash drive (USB 3.0 recommended), or the portable SSD drive, you will browse the Base Folder to your flash drive. It is for the case that you practice the lab in School Laboratory. Or for the possibility that you can use your VM on other computers.

After defining your base folder, click **Import**

Import appliance

← Import Virtual Appliance

Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Virtual System 1	
Name	cyberlab
Guest OS Type	Ubuntu (64-bit)
CPU	1
RAM	2048 MB
DVD	<input checked="" type="checkbox"/>
USB Controller	<input checked="" type="checkbox"/>
Sound Card	<input checked="" type="checkbox"/> ICH AC97
Network Adapter	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4
Storage Controller (IDE)	PIIX4
Storage Controller (SATA)	AHCI
Virtual Disk Image	cyberlab-disk001.vmdk
Base Folder	VirtualBox VMs
Primary Group	/

hine Base Folder: VirtualBox VMs

Address Policy: Include only NAT network adapter MAC addresses

Additional Options: ☒ Import hard drives as VDI

Appliance is not signed

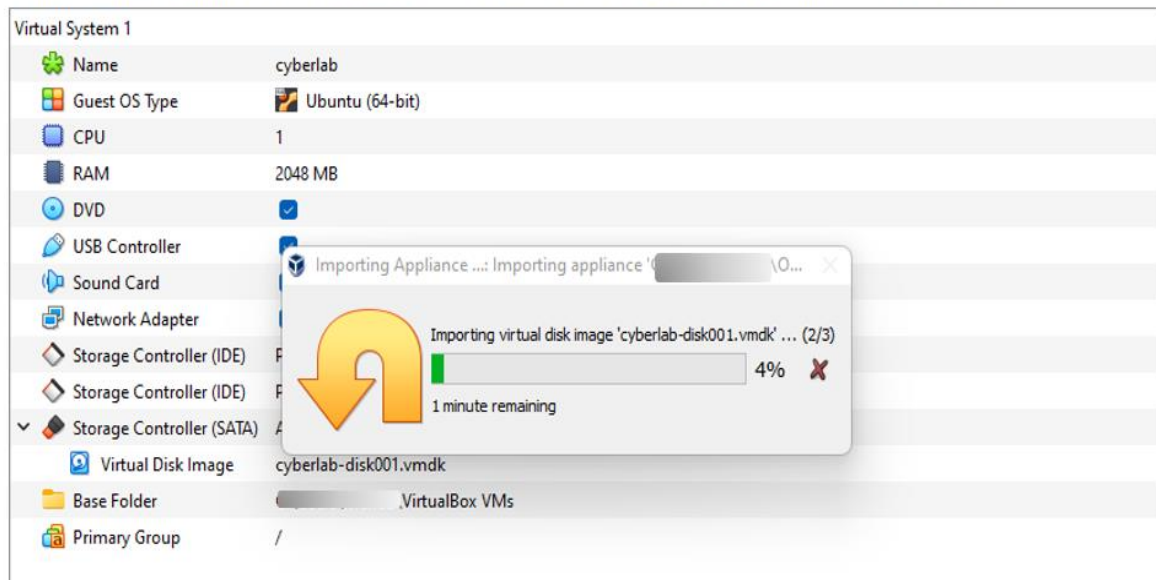
Restore Defaults Import Cancel

Setup Lab Environment

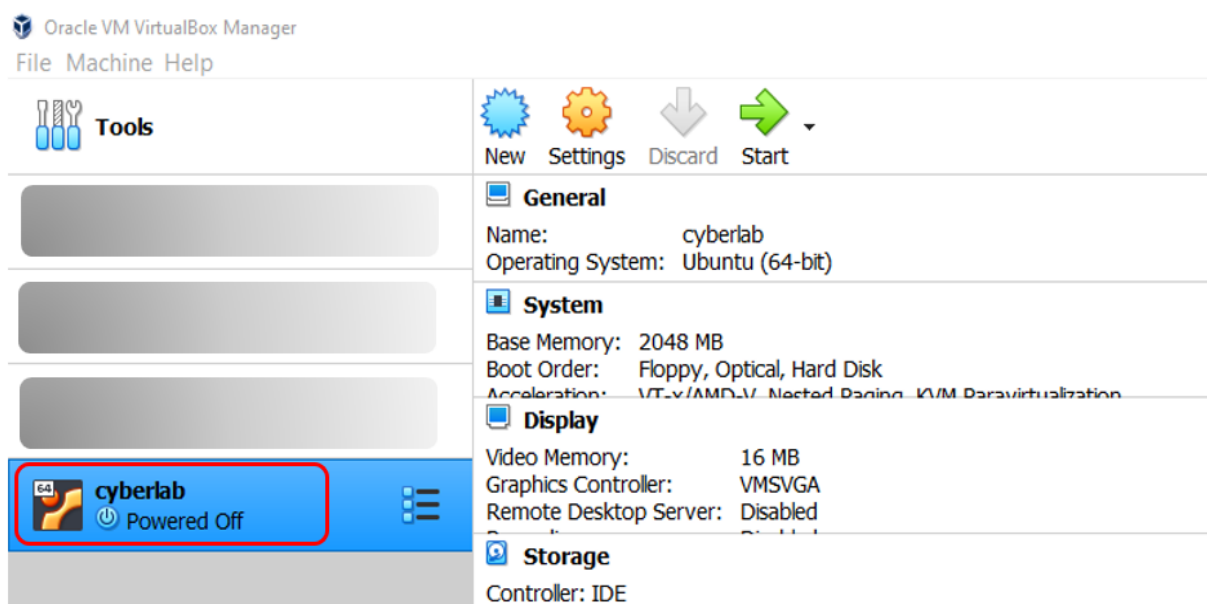
You need to wait for several minutes to import the system file

Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.



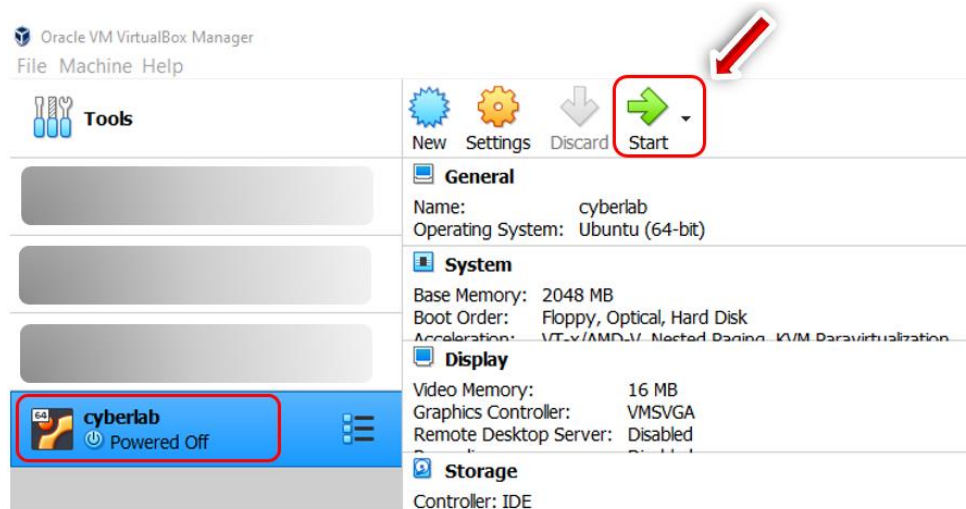
Finally, the VM is successfully imported to the VirtualBox



Setup Lab Environment

Step 2: Start the VM

After importing, you can see the VM in the list with the name **cyberlab**. Please click the **start** button to run the VM.

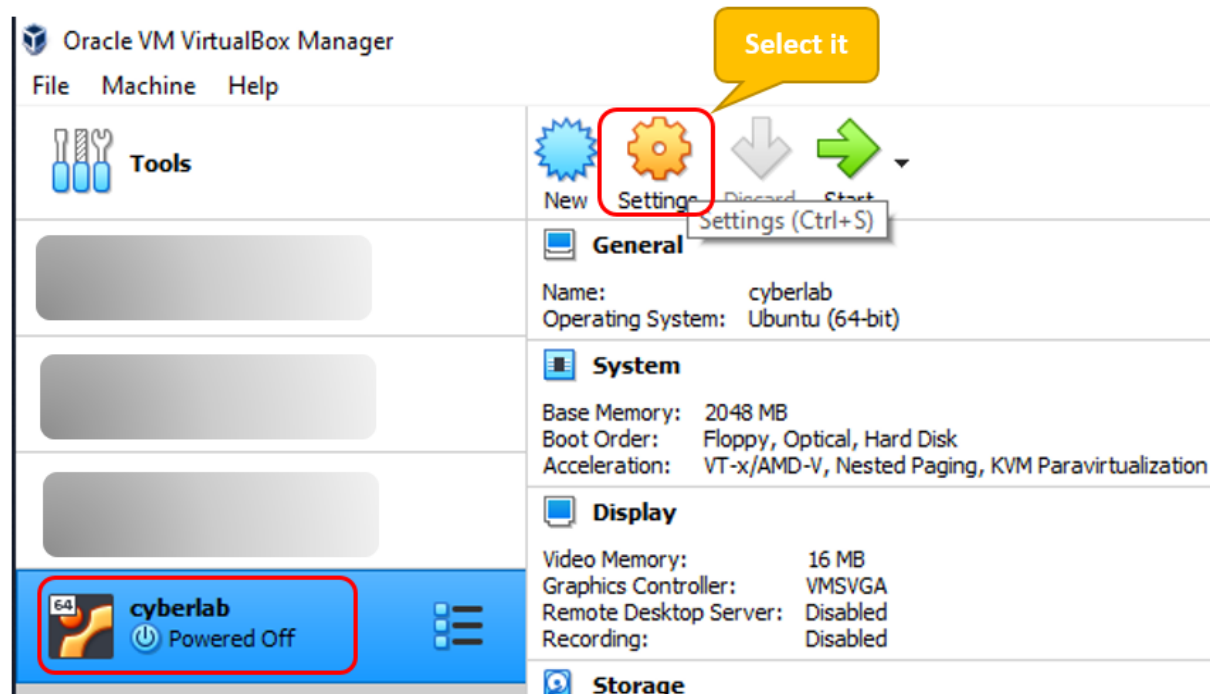


Note: While the VM starts, please click OK if there is an error as below.

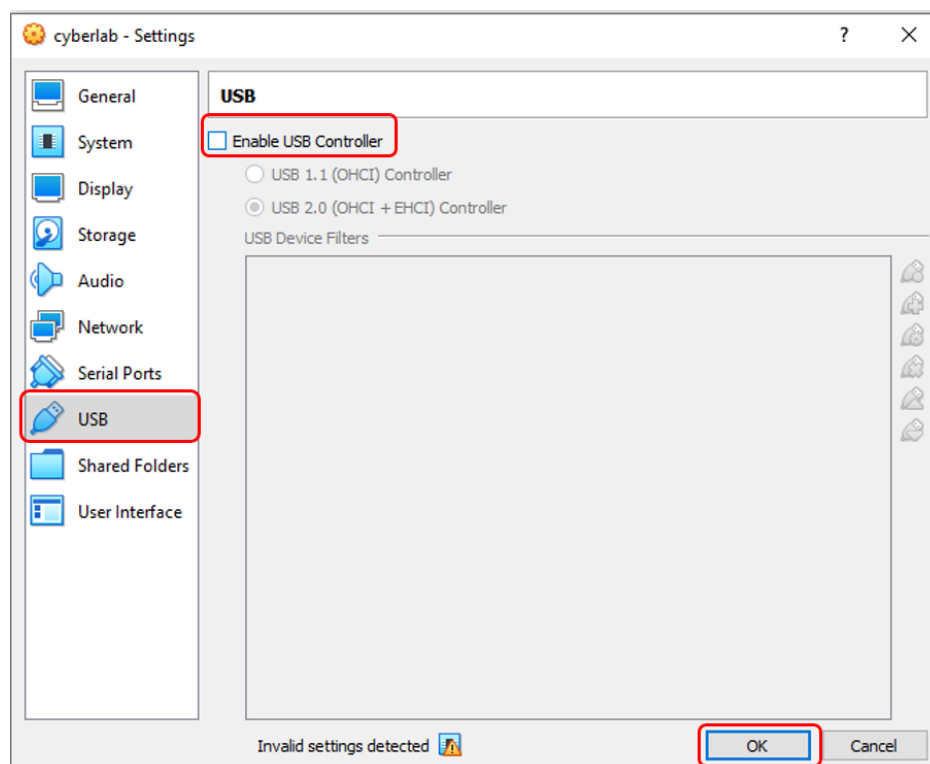


Setup Lab Environment

Select the **cyberlab** VM >> **Settings**

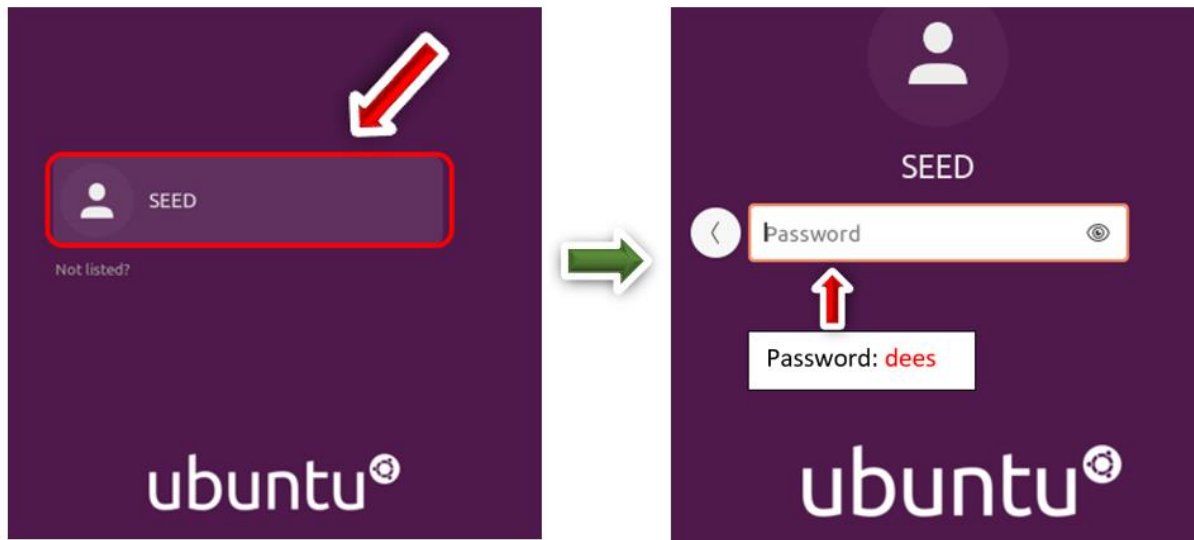


Then select **USB** >> **untick** the box **Enable USB Controller** >> click **OK**

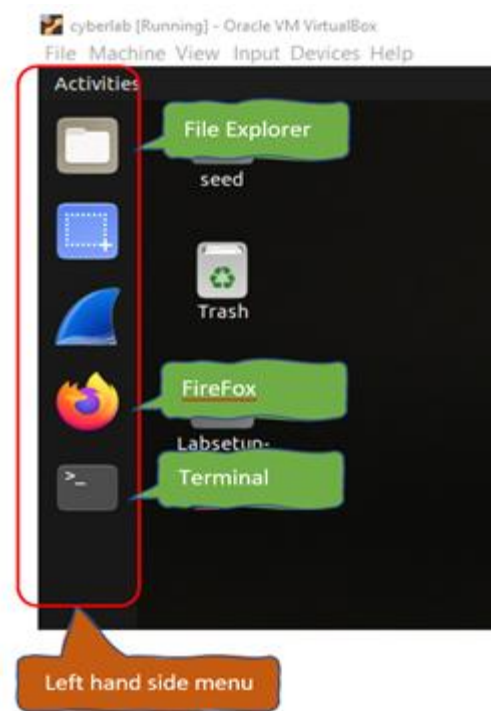


Setup Lab Environment

Now, start the VM again. Then, you will see the login interface for the VM. To log in, please type the password **dees**

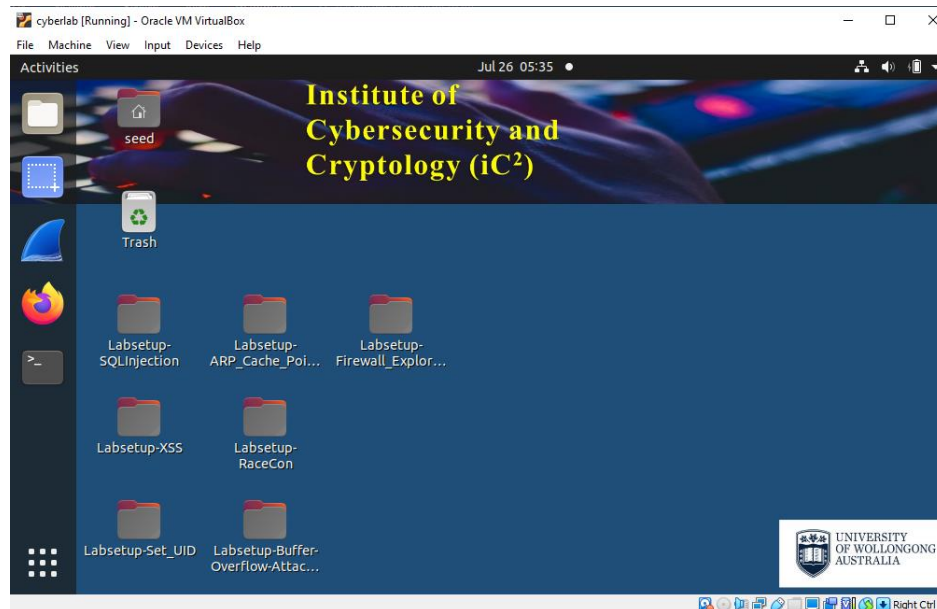


When the VM starts, you will see the view of the Desktop as below. The following image briefly describes icons on the left-hand side menu.



Setup Lab Environment

Other icons on the Desktop that we need to notice are Labsetup-XSS and Labsetup-SQLInjection and other folders to prepare for the later labs



TASK 1:

- Take your screenshot of your VM to make sure your VM is working well.

Hint: you did all the above steps correctly; your VM should work.

4.2 Checking the Network from the VM to the Internet

We use NATNetwork for the VM. As explained before, this VM should have to connect to the Host machine (your computer) and the Internet in case your host machine is connecting to the Internet.

TASK 2:

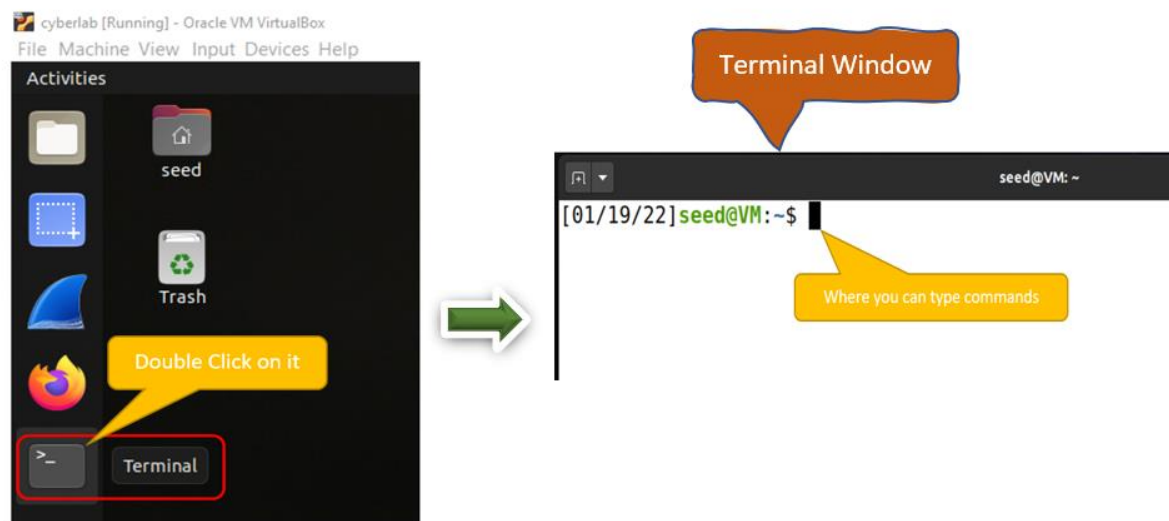
- What is the IP of your VM? Take the screenshot to show that it works
- Does your machine connect to uow.edu.au and google.com or not? Take the screenshot to show that it works.

Hint: use ip a and ping command to do this task.

4.3 Working with container technique

Firstly, we need to open the terminal to run commands.

Please double-click on the **terminal** icon on the left-hand side menu. Then a terminal window pops up. At the cursor, you can type commands to be executed.

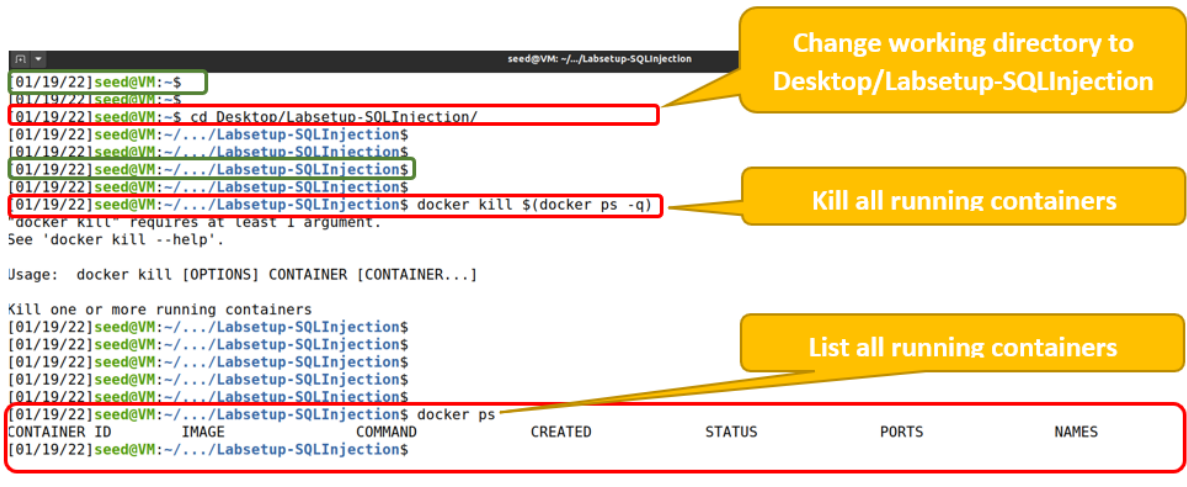


Secondly, we navigate to the target directory, where we run commands for stopping all running containers related to SQLInjection and XSS tasks.

To stop all running containers of **SQLInjection** tasks:

- Navigate to **Desktop/Labsetup-SQLInjection** by using the command:
`cd Desktop/Labsetup-SQLInjection`
- Running a command to stop all running containers:
`docker kill $(docker ps -q)`
- List all running containers
`docker ps`

Setup Lab Environment



```

[01/19/22]seed@VM:~$
[01/19/22]seed@VM:~$ cd Desktop/Labsetup-SQLInjection/
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$ docker kill $(docker ps -q)
docker kill requires at least 1 argument.
See 'docker kill --help'.

Usage: docker kill [OPTIONS] CONTAINER [CONTAINER...]

Kill one or more running containers
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$

```

In this case, there is no containers working at all.

That is correct to start any labs later.

Notice: we can only run Docker services for SQL Injection Lab or other labs at one time. If we run docker files of both SQL Injection or SetUID tasks, for example, at the same time, there will be an error as below:

```

[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$ dcup
Starting 63b074f5cbd6_www-10.9.0.5 ... error
Starting 93e46667f003_mysql-10.9.0.6 ...

ERROR: for 63b074f5cbd6_www-10.9.0.5 Cannot start service www: Address already Starting 93e46667f003_mysql-10.9.0.6 ... error

ERROR: for 93e46667f003_mysql-10.9.0.6 Cannot start service mysql: Address already in use

ERROR: for www Cannot start service www: Address already in use

ERROR: for mysql Cannot start service mysql: Address already in use
ERROR: Encountered errors while bringing up the project.
[01/19/22]seed@VM:~/Desktop/Labsetup-SQLInjection$

```

To **avoid the error**, before any lab, please stop all running containers (as shown in section 4.2) before doing a lab for the coming weeks.

TASK 3:

- What is the version of Docker? Take the screenshot to your report.
- Use the above commands to kill all containers? Take the screenshot.

Hint: from the terminal, use the command `docker --version`

You can refer more to the below link to learn how to check the version of Docker

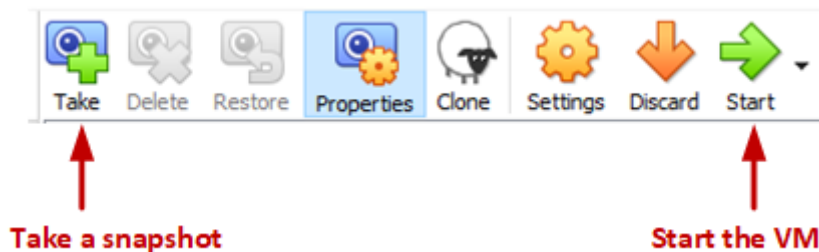
<https://hyperledger-fabric.readthedocs.io/en/release-2.2/prereqs.html>

That is the end of Lab 1!

Below, I introduce more knowledge (3 Appendix) for you to work with the VM more convenient. It is not compulsory, but it is interesting for students who finish the lab and are curious about Virtual Box.

Appendix A: Start the VM and Take Snapshot

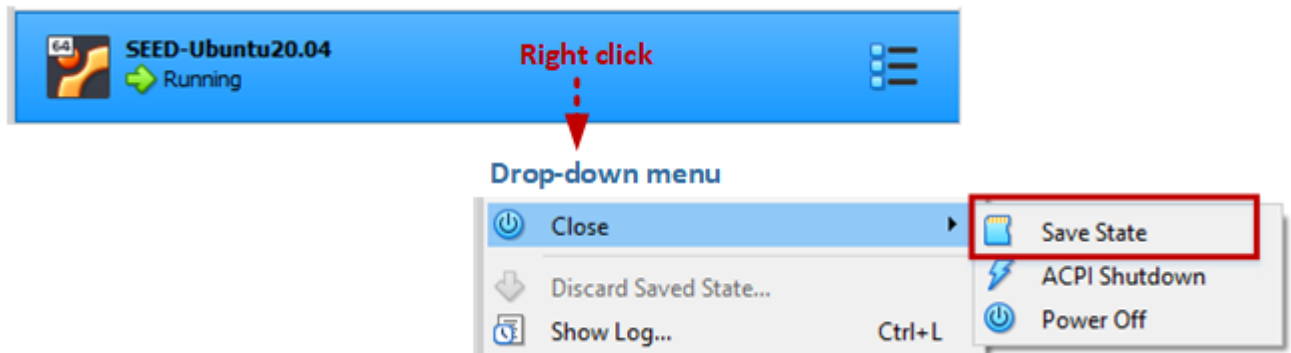
In case you want to take the Snapshot of the VM, you can use the **Take** button to take snapshot of your VM. This way, if something goes wrong, you can roll back the state of your VM using the saved snapshots.



Appendix B: Stop the VM

There are many ways to stop the VM. The best way is to use the **Save State**. This is different from shutting down the VM. It saves the current VM state, so the next time when you restart the VM, the state will be recovered. Moreover, the speed is also faster than booting up a VM.

Setup Lab Environment



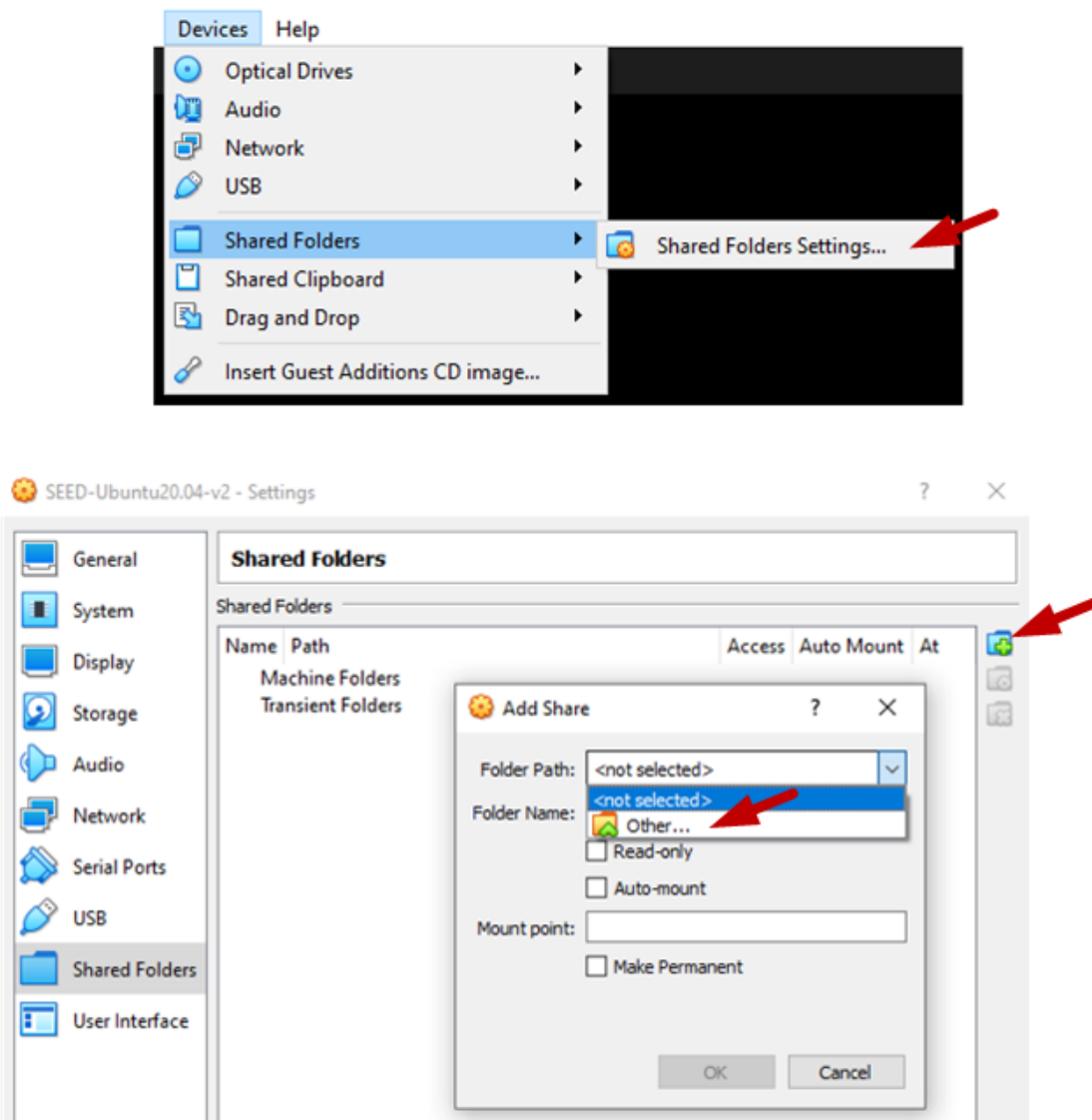
Appendix C: Creating a Shared Folder

Sometimes, we need to copy files between the host machine and the VM. You can see our cloud VM manual for instructions if you are using the VM from the cloud. Or, you can just use a cloud storage service, such as Dropbox and Google Drive to share files between your VM and host machine.

If you run the VM on your local computer, you can create a shared folder between your computer and the VM.

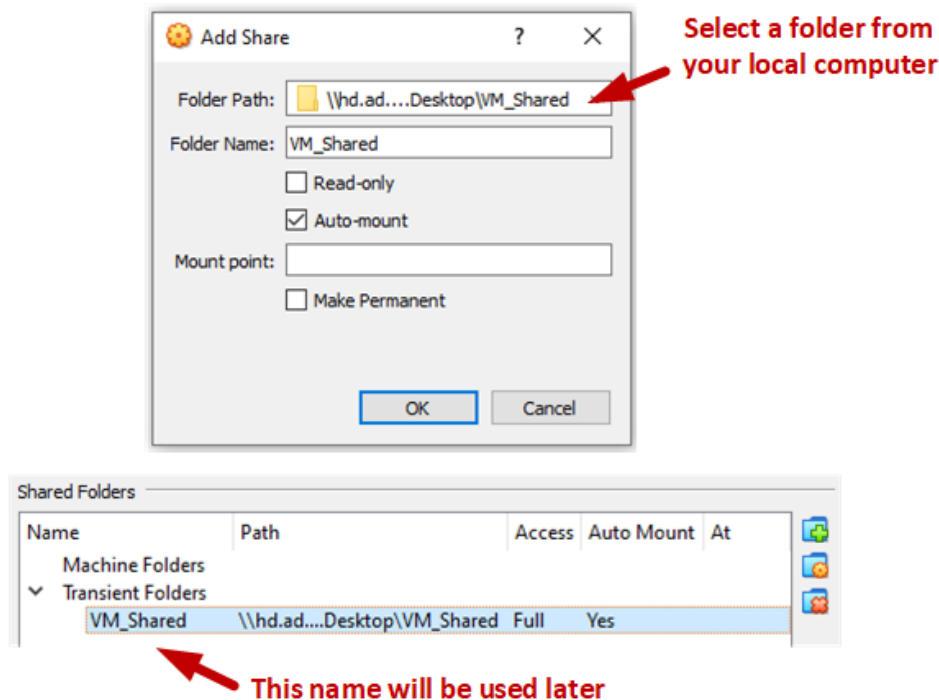
Step A. First, you must create a folder on your local computer (or using an existing folder). We will let the VirtualBox know that this folder should be shared with the VM. Go to the following menus:

Setup Lab Environment



Once you see a **Add Share** popup window, select the folder that you want to share, click OK, and you will see that the folder is now made available for sharing.

Setup Lab Environment



Step B. Inside the VM, we need to mount the shared folder somewhere. Let's mount it to the home directory as a folder `Share`. We will create a folder called `Share` in the home directory and then mount the shared folder `VM_Shared` to this `Share` folder using the following command. After that, you can access the shared folder from `~/Share`.

```
$ mkdir -p ~/Share
```

```
$ sudo mount -t vboxsf VM_Shared ~/Share
```

Important Note. Please only use the shared folder to copy files between the VM and the host machine, and never use it as your working folder. Working from the shared folder has caused many problems, especially with the permissions of the files created inside the shared folder. For example, if we unzip the `Labsetup.zip` file inside the shared folder, the permissions of the unzipped files will be different from those on the original files. Some labs and containers are very sensitive to those permissions.

Enjoy the lab, thank you!