

ECE 650 Project 5 Local VM Installation and Usage Guide

Introduction

This guide will enable you to configure and use a local virtual machine (VM) configured for completing Project 5. The VM's OS is Ubuntu 20.04 running the Linux 5.4.0-104-generic kernel. Designed to provide a consistent and controlled environment, it allows you to focus on the project without worrying about system compatibility or configuration issues, and allows you to easily back up the VM environment to avoid unnecessary time spent reconfiguring a consistent VM environment when the kernel is contaminated by your code.

*****Important Warning Before You Start Your Code:**

Before you run your Project 5 code, please be sure to **back up your code locally, not in the VM**, and make sure that you have **a clean backup VM that has not run your Project 5 code at all!** If you do not do this, you may not be able to start your VM normally after the kernel is damaged or contaminated by your code. This will cause you to **permanently lose your completed code** and will require you to **delete the damaged VM then import it**.

Please do not take any chances. Once your code causes a crash or any other strange situation such as a restart, it will most likely cause irreversible damage to the kernel of your VM. If there are no observable problems with your VM, you can continue to try, but after your attempt is successful, please be sure to test again using the non-contaminated VM that was backed up at the beginning. Otherwise, when we test your code, if there are any problems that should not occur, such as rebooting the VM after finishing the program, we will have to deduct points. If you are worried about forgetting whether your code has caused kernel problems, you can also shut down your VM and use your backup to recover after each problem occurs.

Prerequisites

Before you begin the installation process, ensure that your computer meets the following requirements:

Operating System:

Windows 10/11, macOS (Sierra or later), or a Linux distribution capable of running the required VM software.

Hardware:

At least 8GB of RAM (16GB recommended);
20GB of free disk space (28-32GB recommended), 6.04G for ova image file, 13.8G for the expanded ontology of vm; and
a CPU that supports virtualization.

Software:

Latest version of **VirtualBox**.

Tips:

1. For your convenience, instead of repeatedly deleting and re-backing up VMs, we recommend reserving an additional 13.8G of space for physical backup of clean and pollution-free original VMs. You can also delete the ova image file after import and backup is complete to save disk space.
2. You can also use VMware or other available virtualization software, but for the convenience of setup and use, we will only provide a guide and demo here for VirtualBox.

Set Up

Step 1: Download and Installation

Download the latest version by visiting the Oracle VM VirtualBox official website:
<https://www.virtualbox.org/>

If you are using an M1/M2/M3 processor, please look down at the macOS section.

Non-M1/M2/M3 processors:

clicking the “Download VirtualBox” button,



then choosing the version suitable to your system.

VirtualBox 7.0.14 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

You can install it anywhere (path) you like; there are no restrictions.

M1/M2/M3 processors macOS:

If you use the M1/M2/M3 processors, you should use this link:

https://www.virtualbox.org/wiki/Download_Old_Builds_7_0

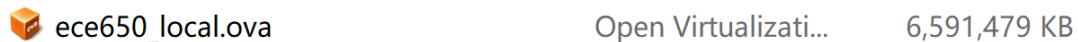
To download the Ver. 7.0.8:

- [VirtualBox 7.0.8 \(released April 18 2023\)](#)
 - [Windows hosts](#)
 - [macOS / Intel hosts](#)
 - [Developer preview for macOS / Arm64 \(M1/M2\) hosts](#) ←
 - [Solaris hosts](#)
 - [Solaris 11 IPS hosts](#)
 - [Linux Hosts:](#)
 - [Oracle Linux 9 / Red Hat Enterprise Linux 9](#)
 - [Oracle Linux 8 / Red Hat Enterprise Linux 8](#)
 - [Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7](#)
 - [Ubuntu 22.04](#)

Step 2: Downloading The VM Image

Download the VM image file: ece650_local.ova to a **known** location on your computer using the following link:

<https://drive.google.com/file/d/1INI4LWicjiIop6z4zigZhDX37eQThKhI/view?usp=sharing>



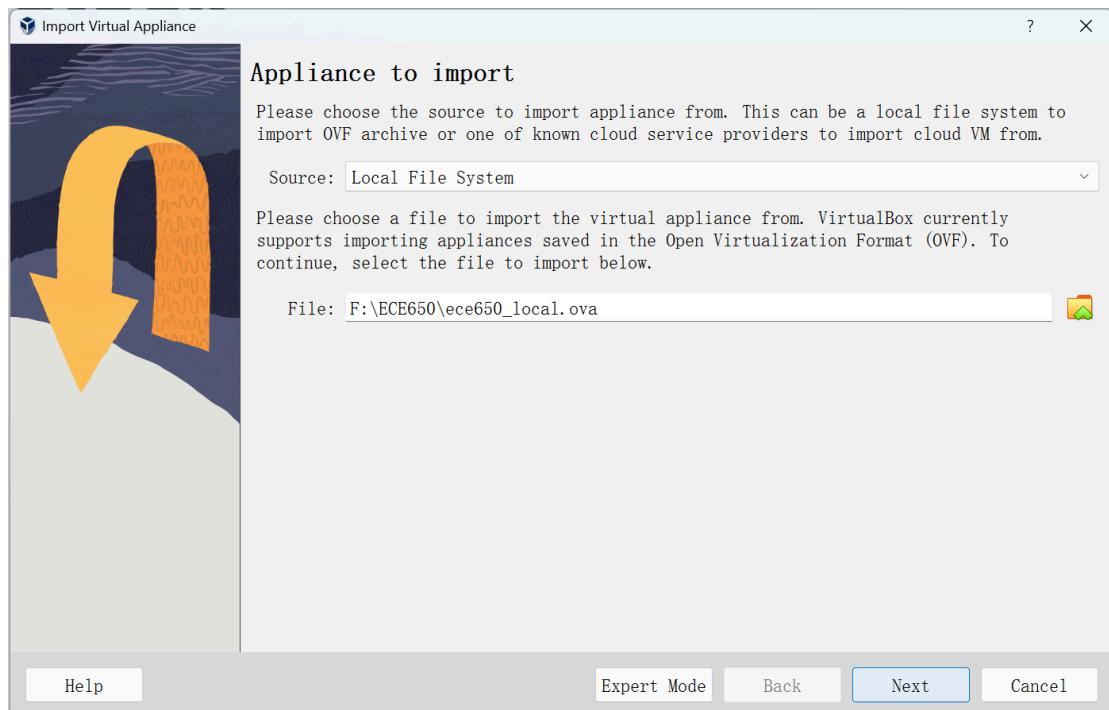
Step 3: Importing the VM into Your Virtualization Software

For VirtualBox:

Open VirtualBox and go to "**File**" > "**Import Appliance**", or you can click the big "**Import**" button on the homepage.



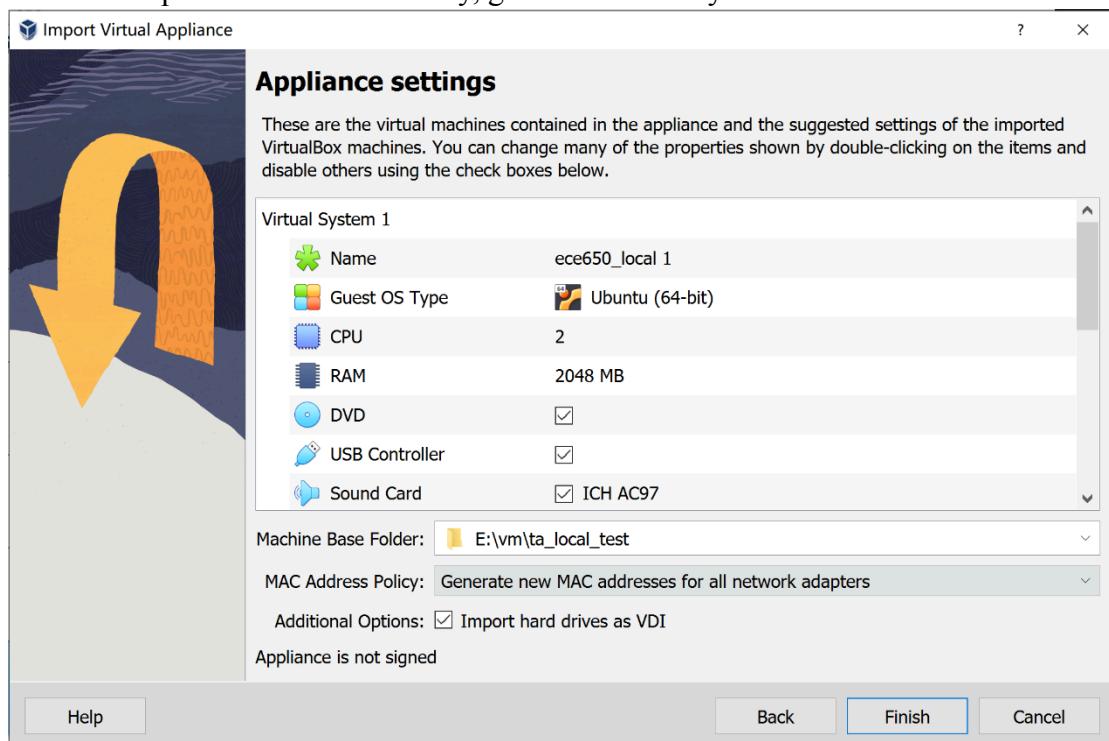
Select the downloaded .ova file and follow the on-screen instructions to import.

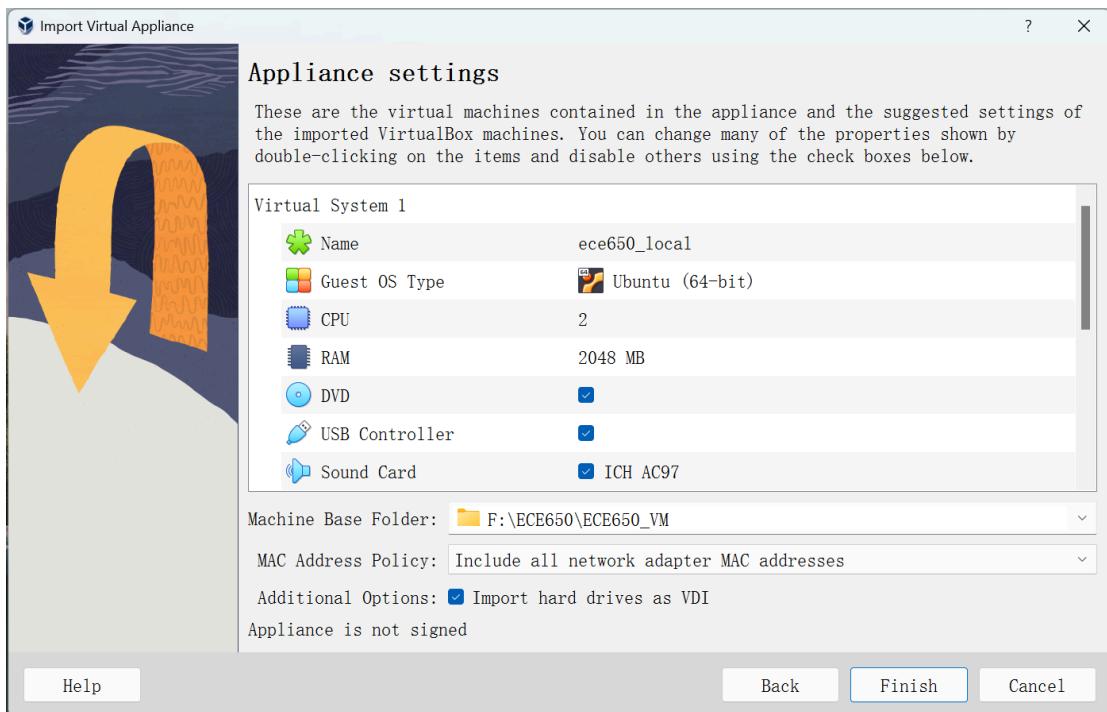


Please choose a path that has enough disk space (at least 14G) to store the expanded VM after importing. We recommend using an empty directory.

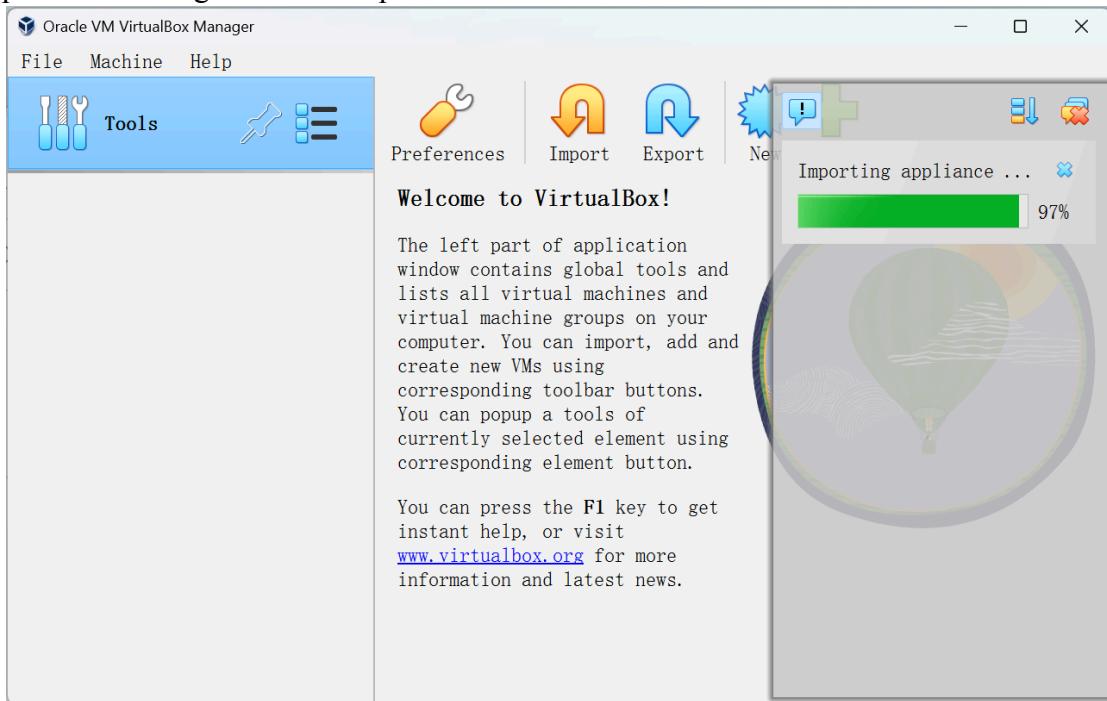
Please review the MAC Address Policy: If there is an option to "Generate new MAC addresses for all network adapters", select it, otherwise, select "Include all network adapters MAC addresses".

If you choose other options, after completing the import process, check the network column in the setting of the imported VM to see if the MAC addresses of all the network adapters exists. If necessary, generate manually.





After you check that everything is correct and click the Finish button, the import process will begin. Please be patient and wait for it to finish.



Once imported, you can adjust the VM settings (e.g., allocated memory or CPUs) based on your system capabilities and on the parameters and performance of your hardware.

Note that you do not have to make any changes.

When the VM is successfully imported, you will see the VM in the VirtualBox's VM list.

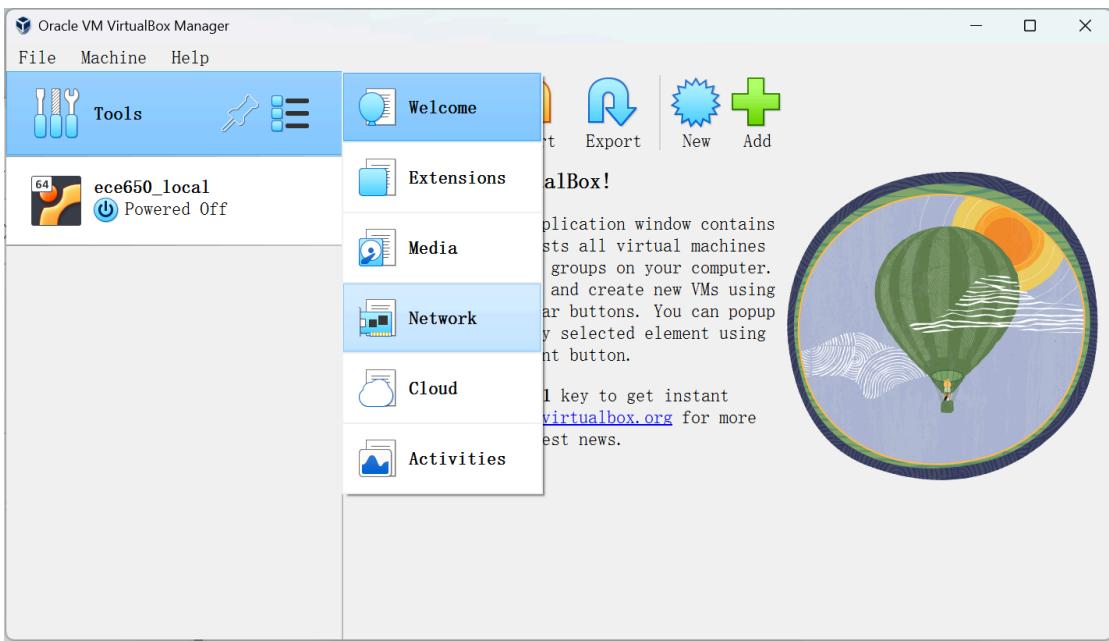
Settings

Next are some preliminary settings for your convenience.

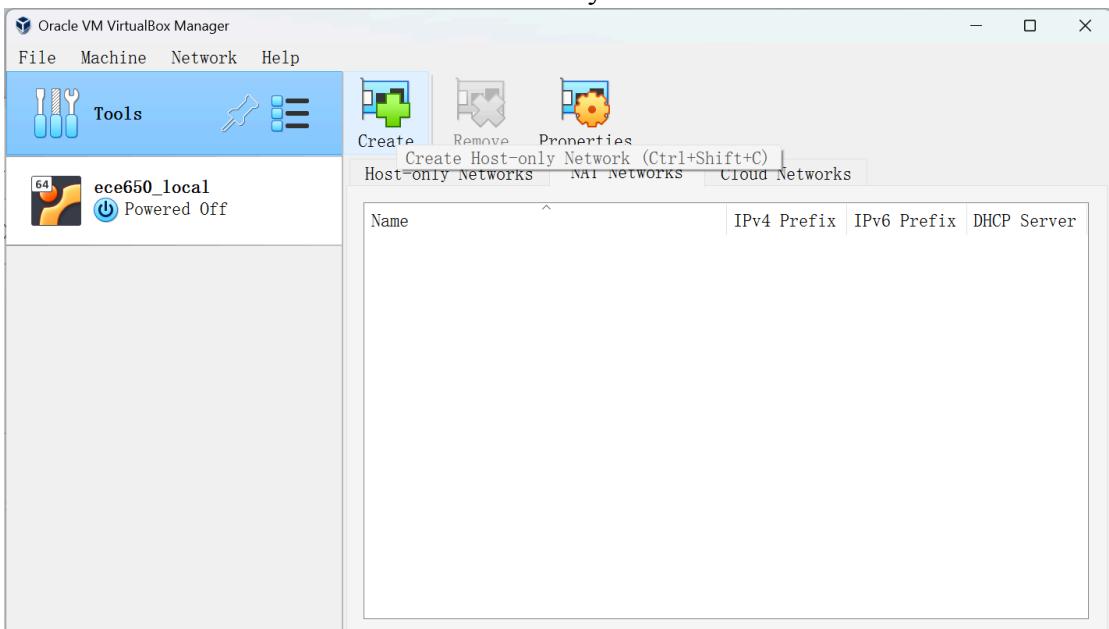
Step1:

Check and create the NAT Network to allow your VM to serve as a slave of your host and connect to the external network at the same time.

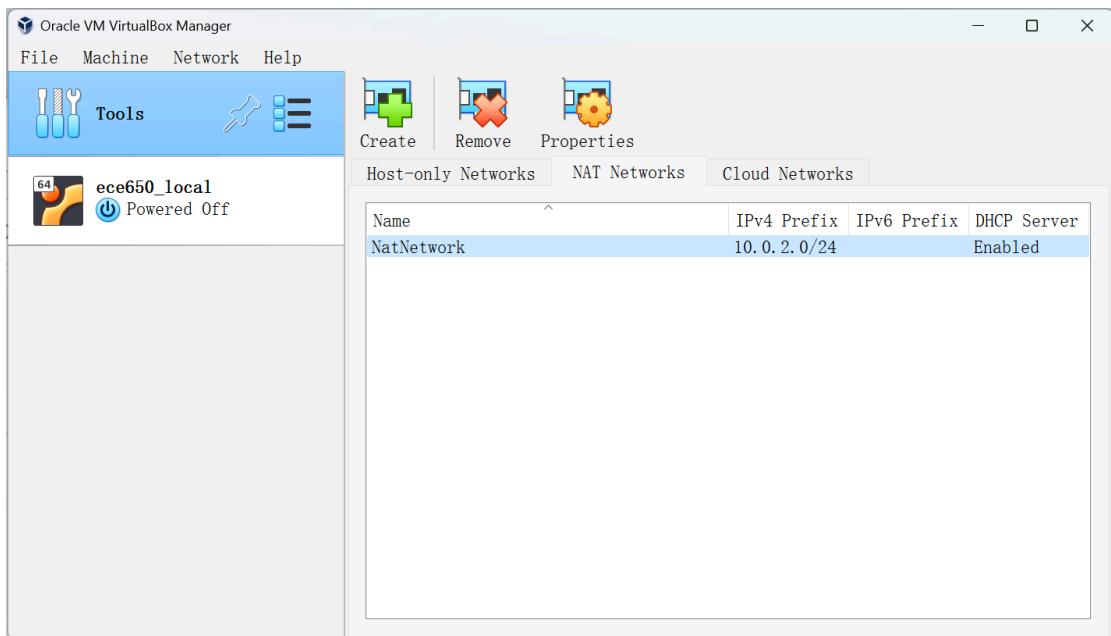
Click  then click the Network.



Select the NAT Network column. If it is empty like in the picture, click **Create** to create a new one. You do not need to make any modifications to it.

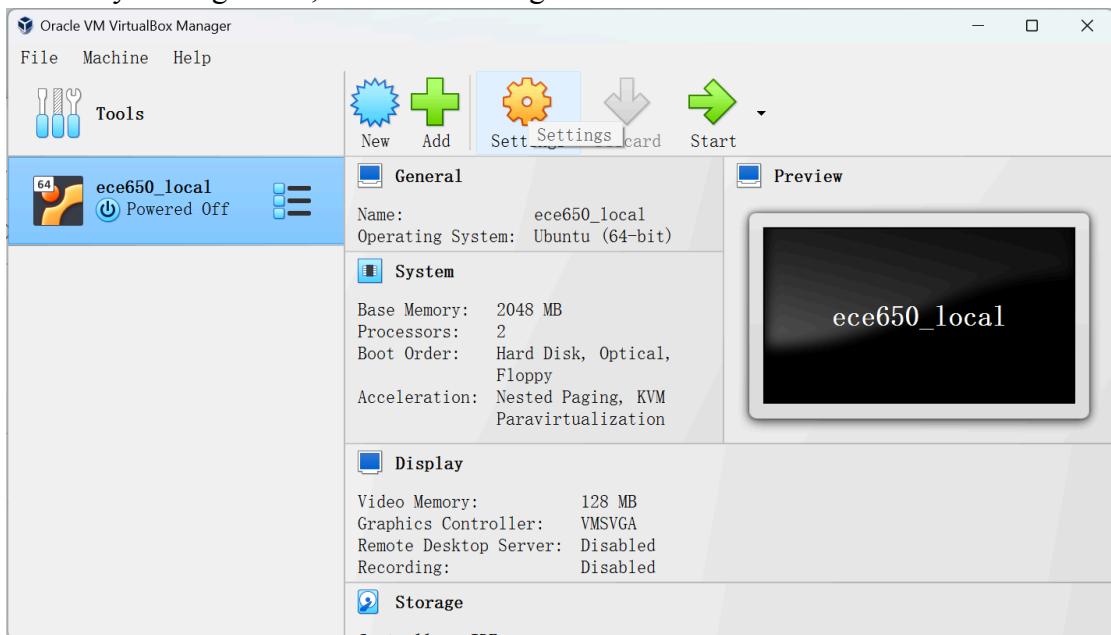


If your list is not empty or you have finished creating it, its status should look like the image below:



Step2:

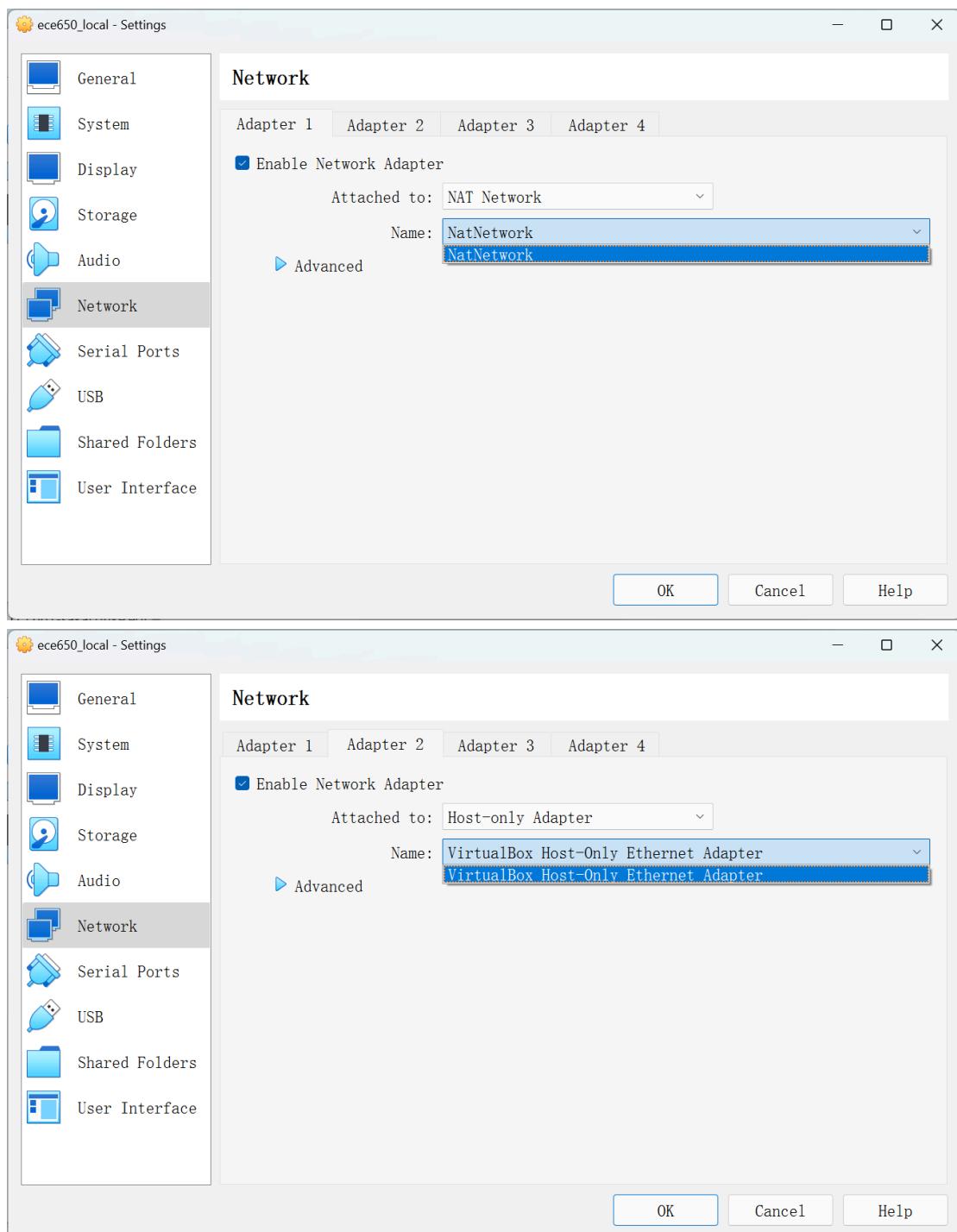
After adding the NAT Network to your Virtual Box environment, we need to add the newly created NatNetwork to your VM so that it can connect to the external network. Click to your target VM, then click setting:



Click to Network menu.

View the settings of Adapter1 and Adapter2 (please note: only these two Adapters are enabled).

Please make sure your settings are consistent with the image below. You do not need to pay attention to the **advanced** functions unless there is a problem with your MAC addresses.

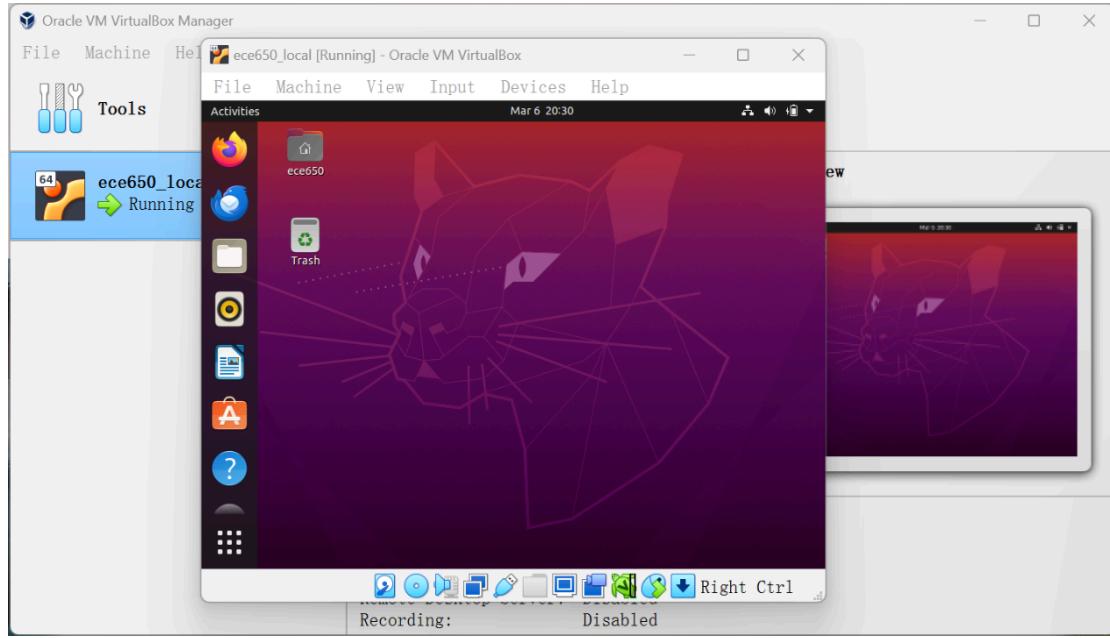


After you checked that the Adapters settings are correct and make appropriate changes based on your preferences and hardware capabilities, click the **OK** button to save and complete the VM settings.

Now you can turn on the VM you have set up.

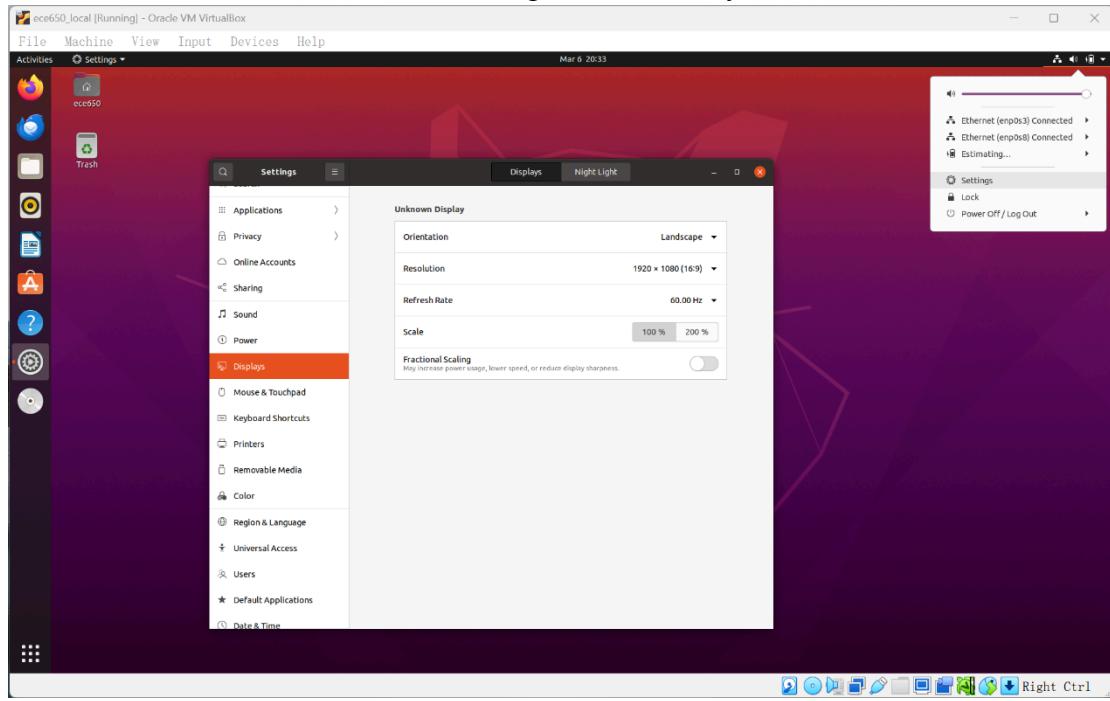
Username: eve650

Password: 123456987

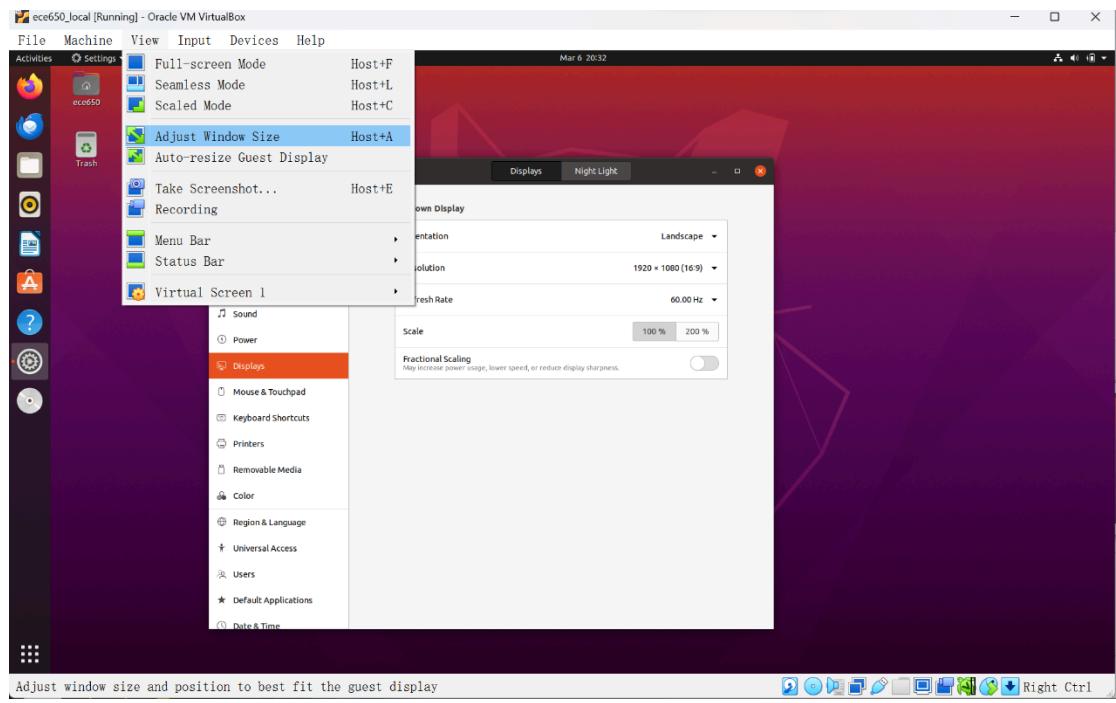


Step3:

If you find that the menu of VM cannot be scrolled, or you are not satisfied with the current resolution and window size, you can click the menu in the upper right corner of VM. Then select setting. Find display in the menu on the left in the setting window, and then select the resolution and other options that suit you, as shown below:



Then click "**View**" > "**Adjust Window Size**" in the function bar of the VM window, so that your window size will change with the actual size of the VM. Like the picture below:



Congratulations, the initial setup of your VM is completed!

Backup Settings and Notes

After completing the initialization settings of the virtual machine, you can back it up immediately. This will ensure that there will be no problems with the backup files used later.

You can also back up immediately after adding SSH password-free login and creating your project directory. This will make it easier to log in and use during subsequent recovery, and you can continue your project more quickly. Time wasted entering passwords and creating project paths. For specific details, please refer to the guide in the subsequent Usage section. (Recommend)

But remember: Be sure to back up your VM before you run your code for the first time!

Note: If your disk cannot bear two full VMs of 13.8G (total of about 28G), but you want to use this VM, please do not use the following backup method which is not for you. You still have to back up your code locally before running it. If your code causes kernel contamination or damage, you need to delete the original VM and re-import the VM.

Backup:

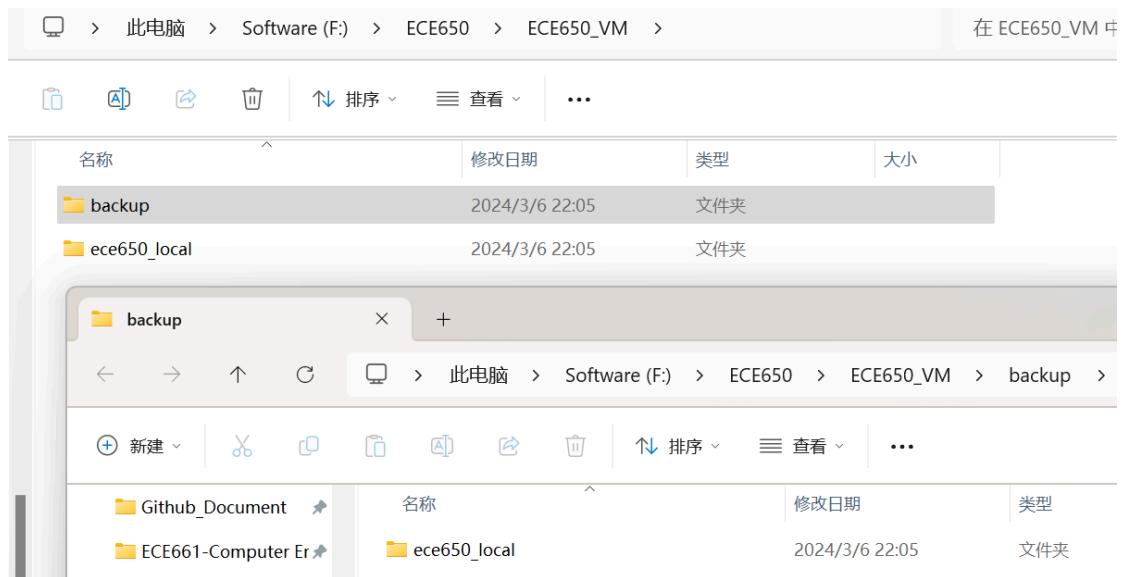
Find the directory where you store the VM. There is no need to enter the real root directory of the VM.

NOT HERE:

脑 > 本地磁盘 (E:) > vm > ece650ta > ece650_local

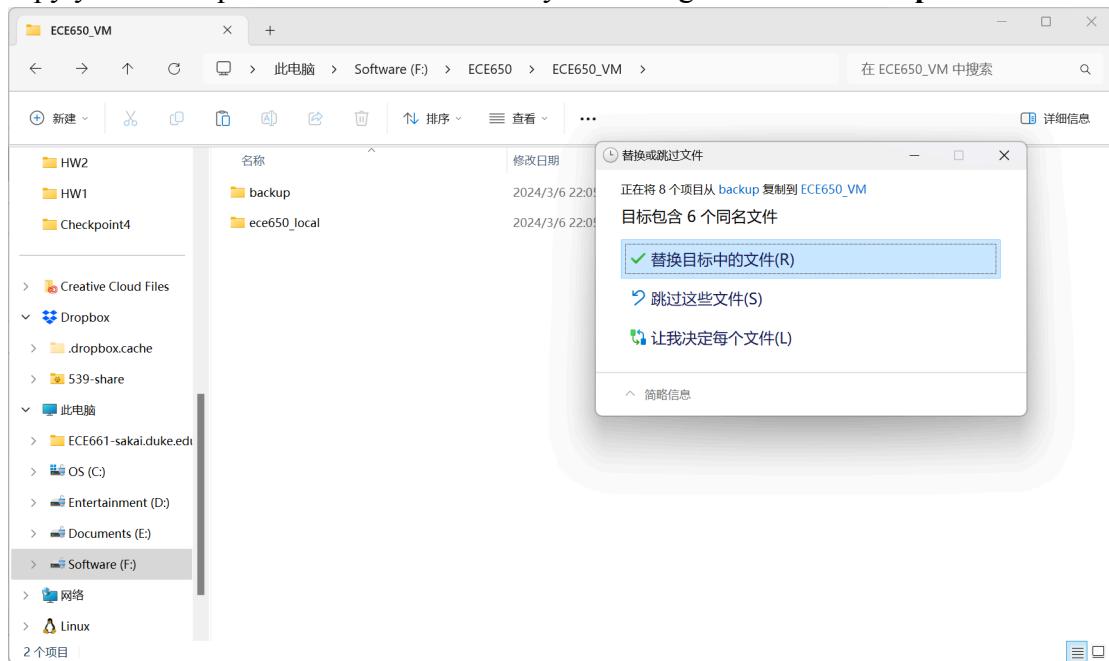
名称	修改日期	类型	大小
Logs	2024/3/7 0:51	文件夹	
ece650_local.vbox	2024/3/7 0:54	VirtualBox Machi...	5 KB
ece650_local.vbox-prev	2024/3/7 0:54	VBOX-PREV 文件	6 KB
ece650_local.vdi	2024/3/7 0:54	Virtual Disk Image	14,542,84...
Unattended-fd3f3039-590b-470d-a0...	2024/2/9 12:15	VISO 文件	1 KB
Unattended-fd3f3039-590b-470d-a0...	2024/2/9 12:15	Configuration 源...	1 KB
Unattended-fd3f3039-590b-470d-a0...	2024/2/9 12:15	Configuration 源...	3 KB
Unattended-fd3f3039-590b-470d-a0...	2024/2/9 12:15	Configuration 源...	4 KB
Unattended-fd3f3039-590b-470d-a0...	2024/2/9 12:15	SH 源文件	8 KB

You need to physically copy the **ece650_local** directory (VM directory) to another safe place. Here, for the purpose of demonstration, we created a backup directory in the same directory to store the backup files. You can see this in the figure below:



The two ece650_local directories are exactly the same and are copied and pasted into the backup directory.

When you need to use the backup file for recovery, please shut down the VM, then copy your backup file and overwrite it on your damaged VM. Select **replace all files**.



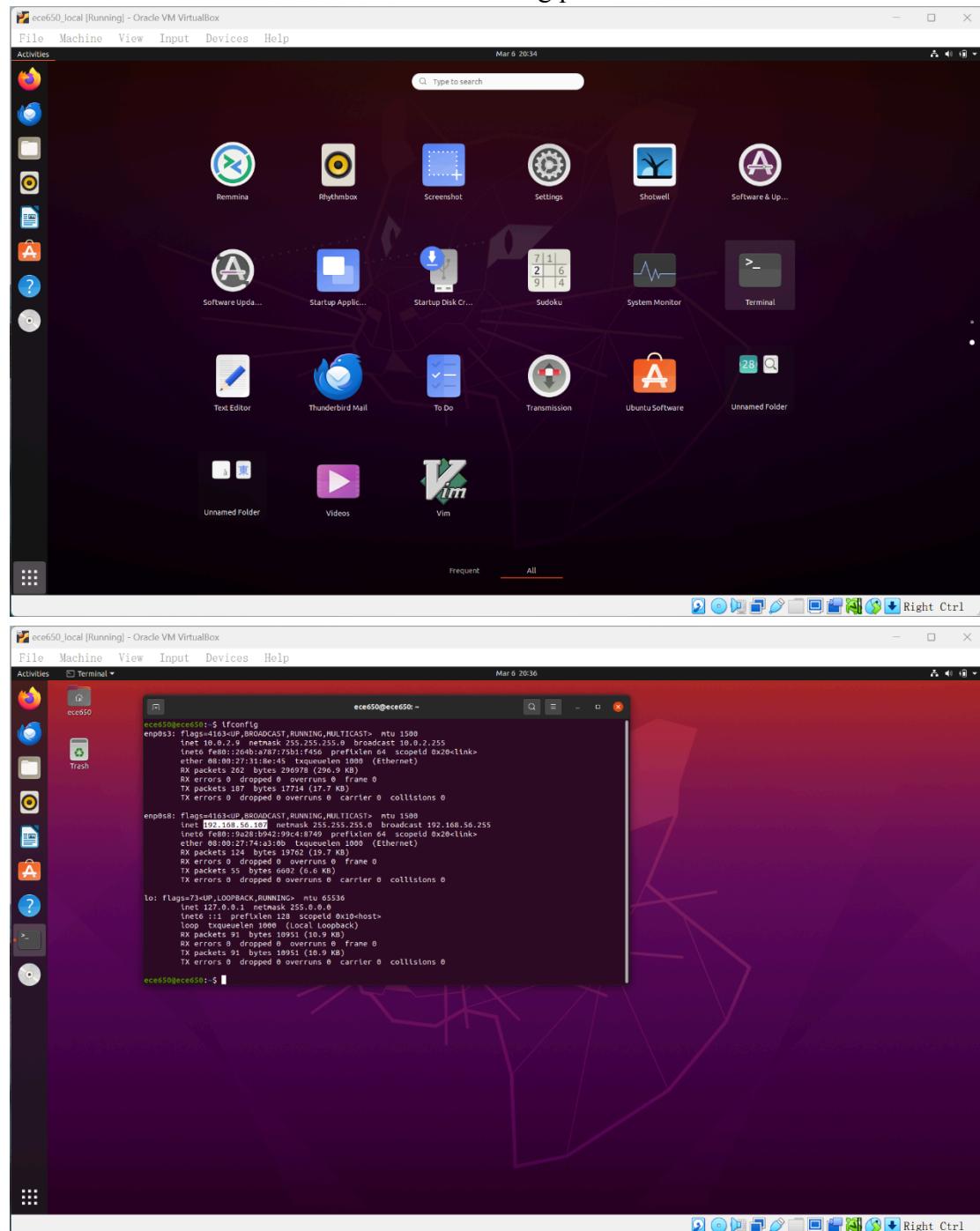
Then you will get an initial clean VM again. All its settings will not change. You just need to start it again and upload the locally backed up code files, and you can continue your work.

Usage

Due to various reasons, such as lag, it is not very convenient to directly use the Ubuntu desktop visualization version as a local VM, so we use other IDEs or terminals or shells to connect to our VM, just like systems without visualization. The adapters have been set up in the previous steps. If the settings are not completed, your VM will not be able to use ssh to connect. If you want to use ssh to connect in all network environments, please return to the previous steps to set up the adapters.

Step1:

Find terminal in the menu, open it and enter the command `ifconfig` to obtain network information. As shown in the following pictures:



We found enp0s8, which is the highlighted IP in the above picture. The network

allocation of each person's VM can be different. Please find your own slave machine IP address based on the results of ifconfig command.

Please remember this IP, you can also copy it. This IP address is to connect to your VM by ssh.

Step2:

You can use ssh to connect your VM to develop your project.

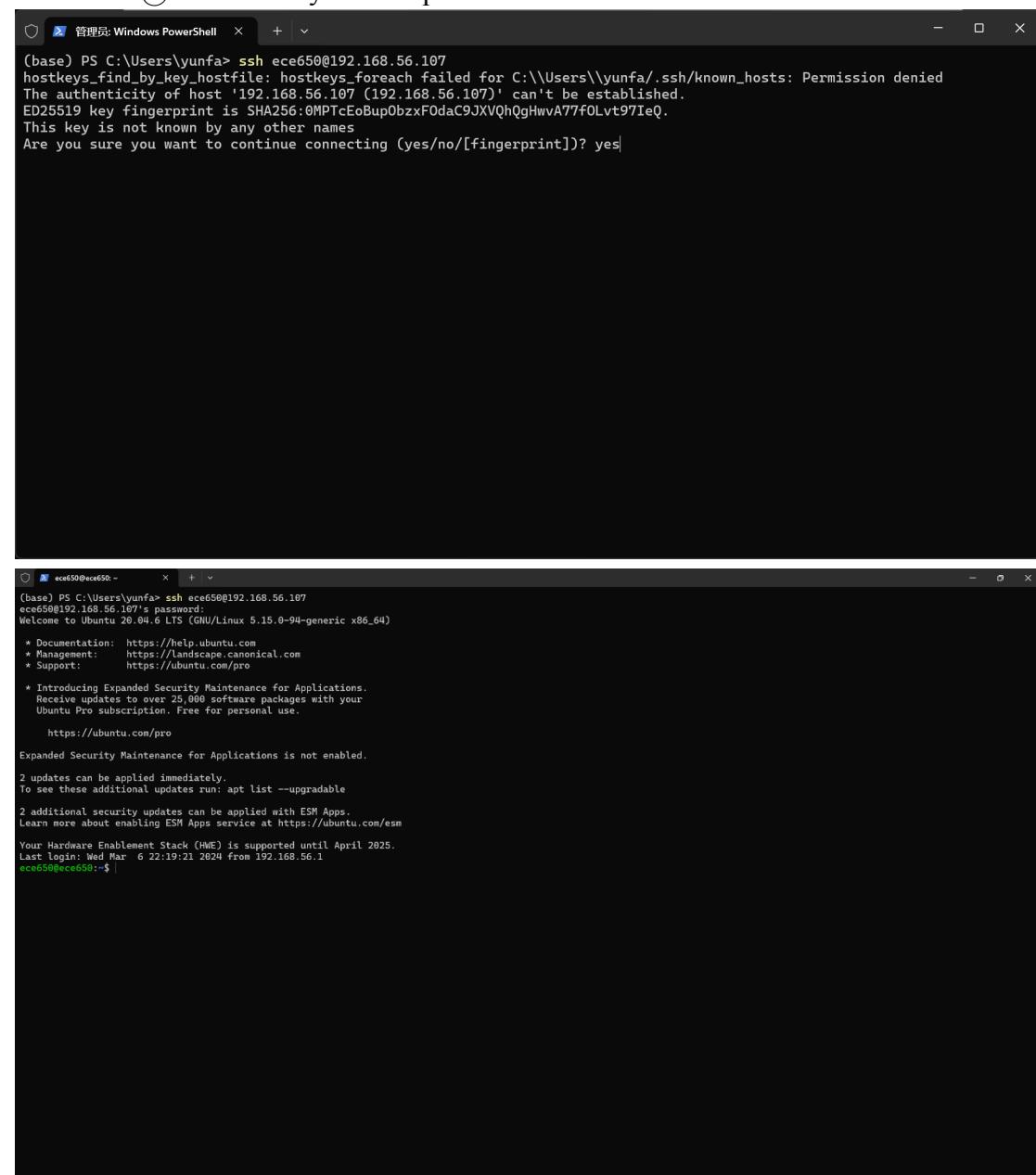
Username: eve650

Password: 123456987

1. Terminal:

You can use the most traditional terminal or shell (admin) to connect to your VM using ssh:

ssh ece650@192.168.56.yourownipnumber

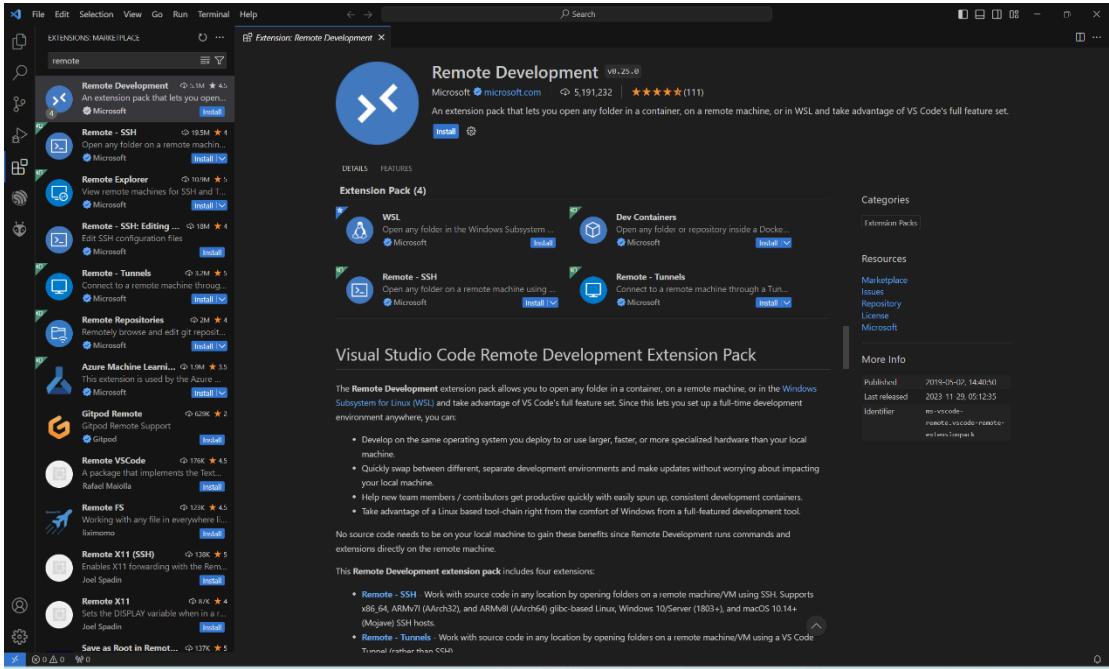


The image shows two terminal windows side-by-side. The left window is a Windows PowerShell session titled '管理员: Windows PowerShell'. It displays the command 'ssh ece650@192.168.56.107' and its output, which includes a host key fingerprint and a prompt asking if the user wants to continue connecting (yes/no/[fingerprint]). The right window is a Linux terminal session titled 'ece650@ece650: ~'. It shows the user has logged in successfully to an Ubuntu 20.04.6 LTS system (GNU/Linux 5.15.0-94-generic x86_64). The terminal displays standard Ubuntu welcome messages, software updates information, and a prompt at the end.

2. IDEs(VS Code here)

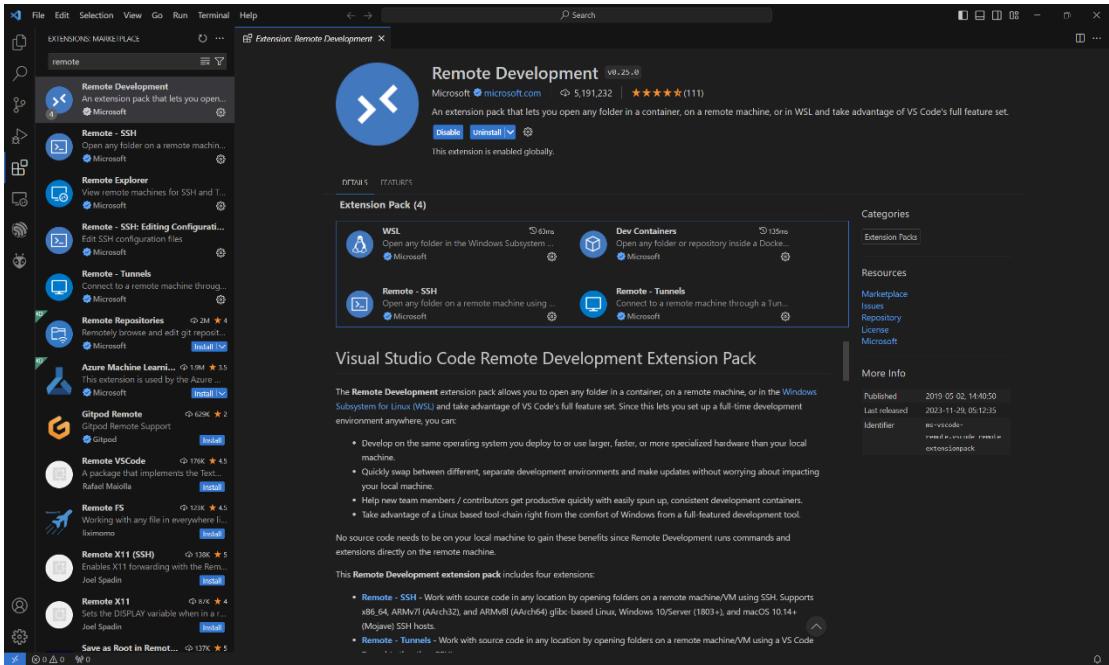
You can also use any IDE you are comfortable with to connect to your VM using ssh.
Here, we will use vscode to demonstrate.

First install the Remote Development plug-in in your vscode:



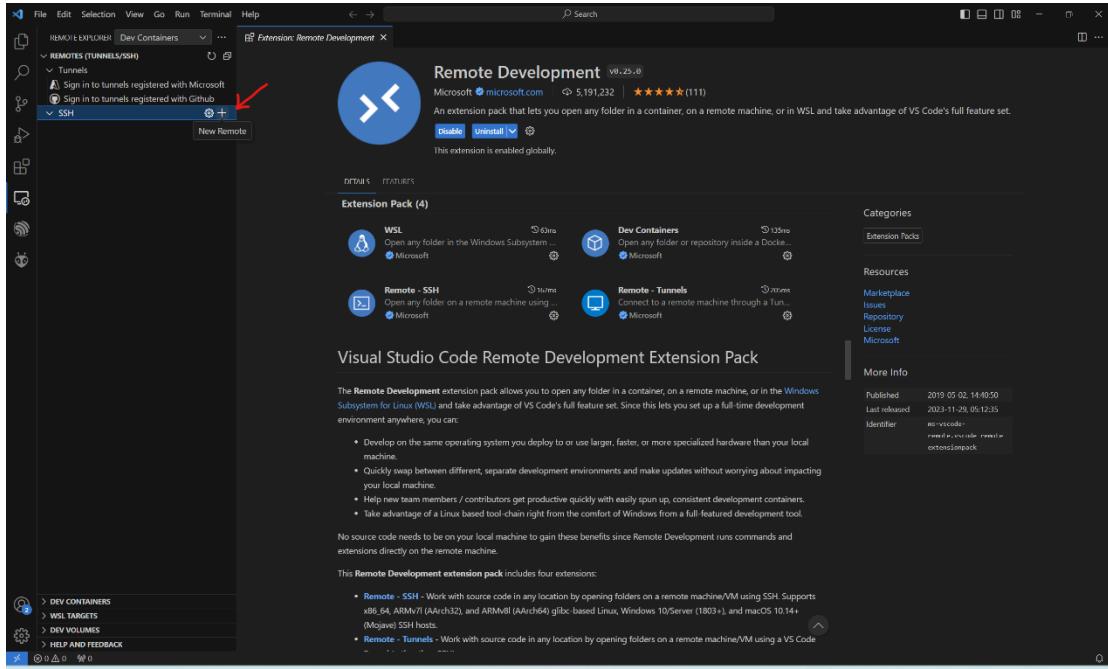
The remote development suite facilitates the connection and management of all your remote-connected devices or accounts.

The status after installation is shown in the picture below:

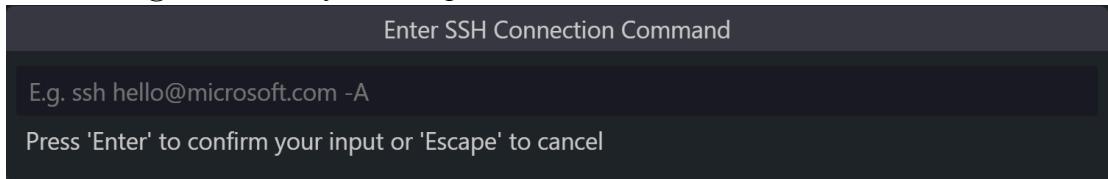


After the installation is complete, there will be a remote function logo in the toolbar on the left side of your vscode.

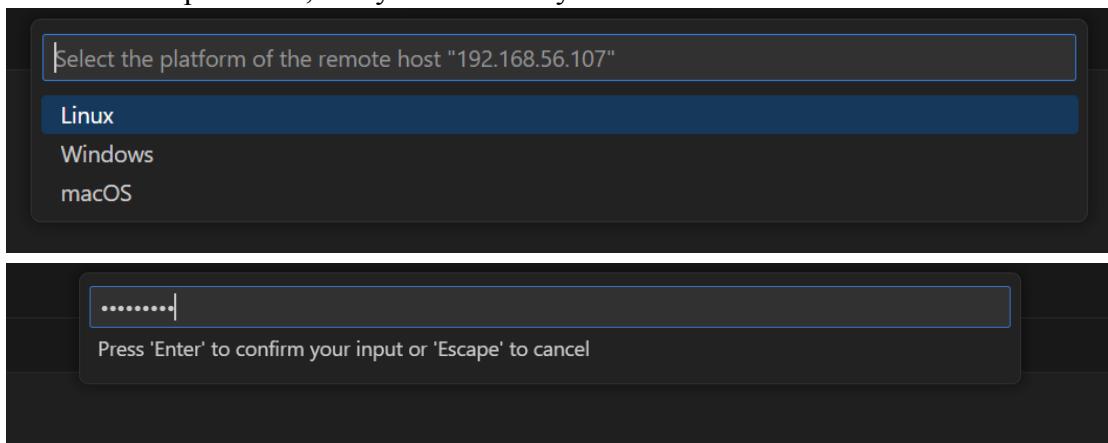
After clicking the remote logo button, the remote column will appear. After clicking, the remote column will appear, as shown below.



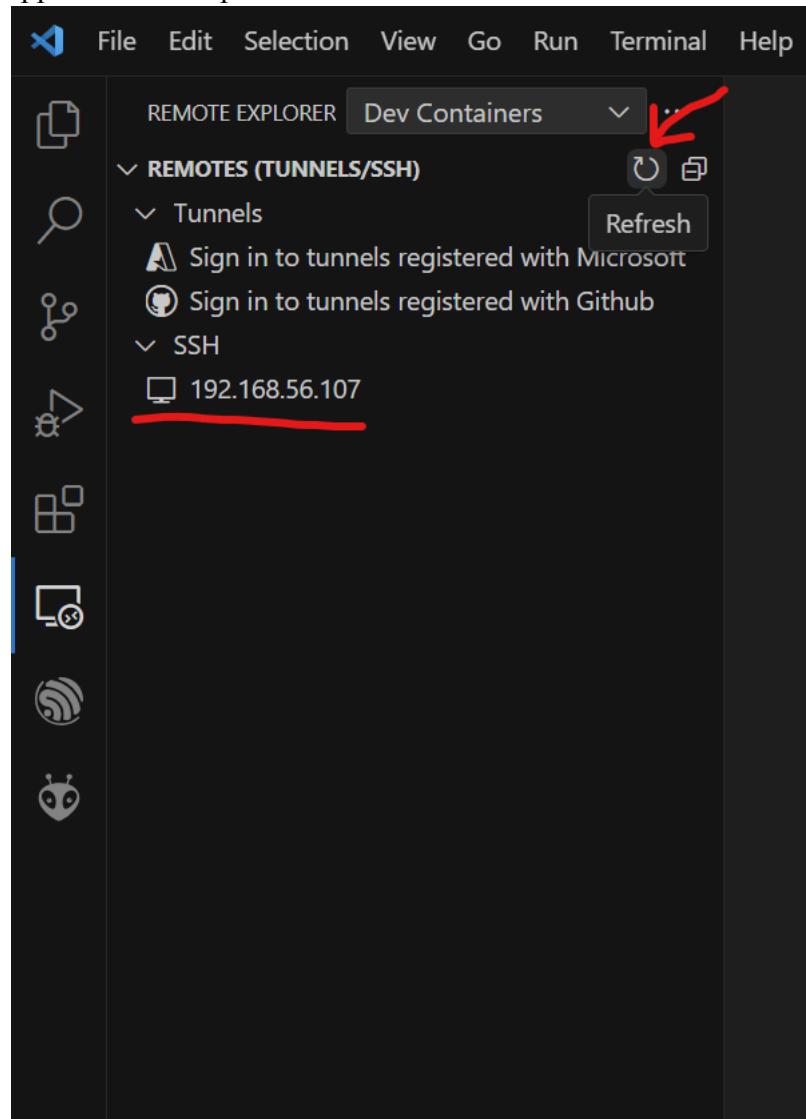
Click the plus button pointed by the red arrow in the picture above, the following block will appear, enter:
ssh ece560@192.168.56.yourownipnumber



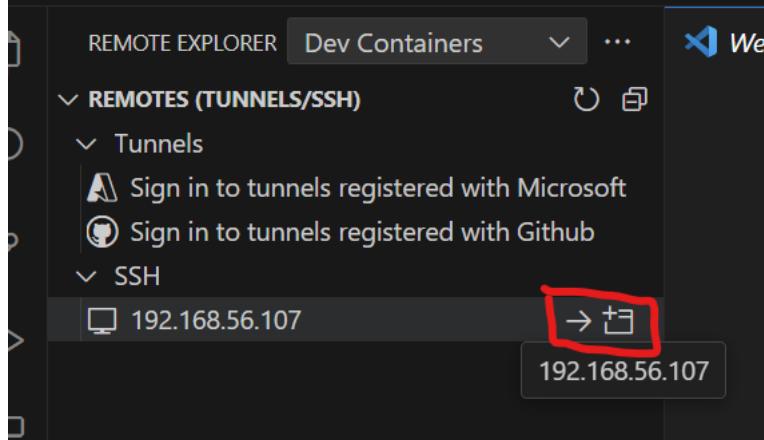
Then follow the prompts to perform subsequent operations: select the Linux system, then enter the password, and you can enter your VM in vscode.



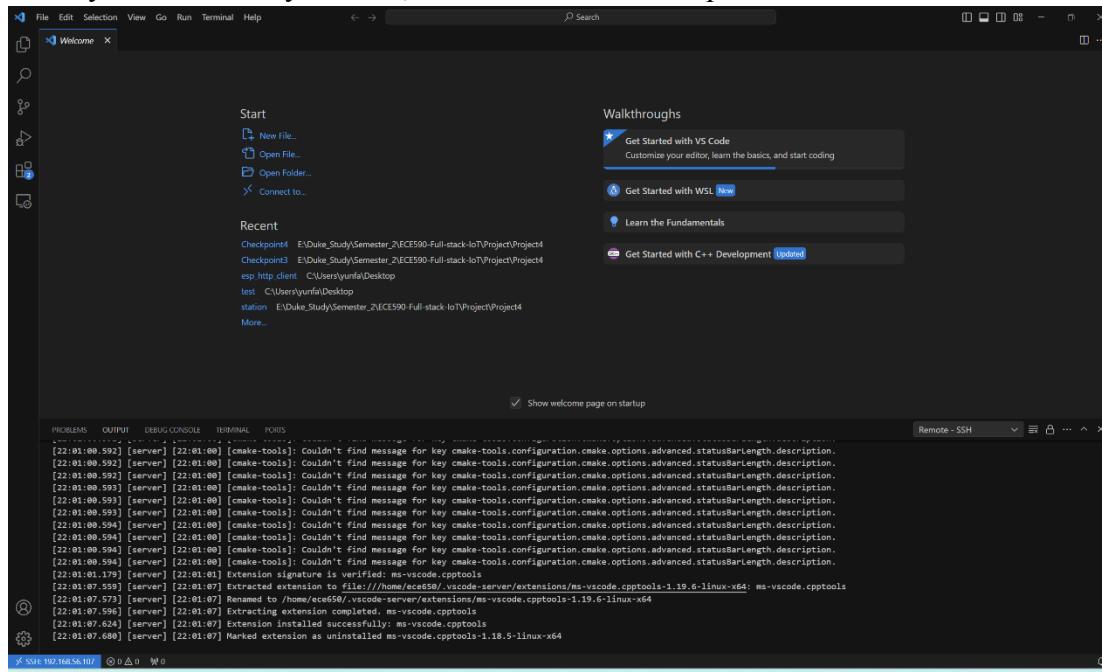
After you enter the VM once, you can see the remote device you have connected to in the remote window. If you don't see them, click the refresh button below and they will appear. Like the picture below:



When you want to connect to your VM, you can select either of the two icons marked in red on the right in the picture below:

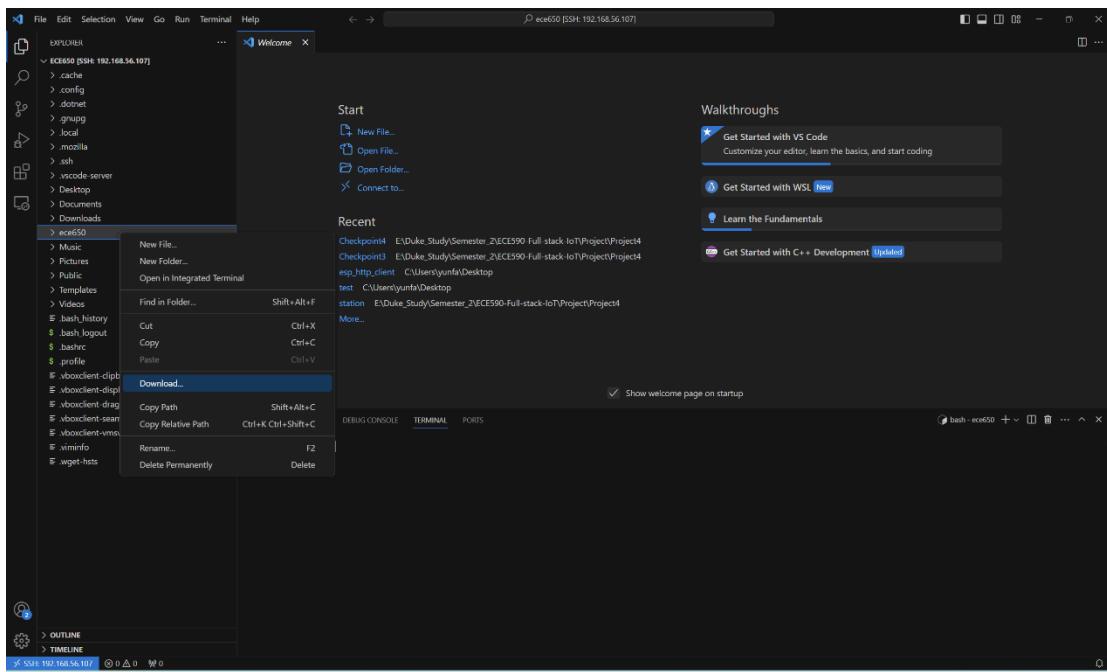


When you connect to your VM, it should look like the picture below:



You can enter the directory you want to go to and create the files and directories you want. Here, we take the ece560 directory as an example.

We have completed the creation here. Assuming that the code has been completed and you want to try to run it, before running it, you must right-click the file or directory to be backed up and select download to back up everything locally. Like the picture below:



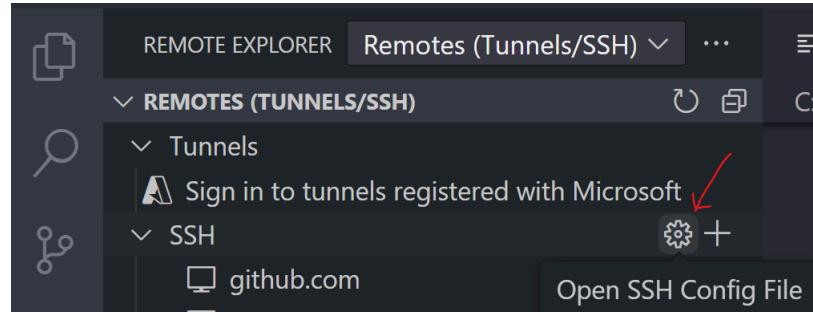
Password-free access:

Create your public and private keys in the C:\Users\yourusername\.ssh\ directory on your local computer or use existing public and private keys such as id_ed25519 and id_rsa. Copy all the contents of the public key of the key you want to use and paste it into the ~/.ssh/authorized_keys file in the VM. If there are multiple keys, each line is one key, like the picture below:

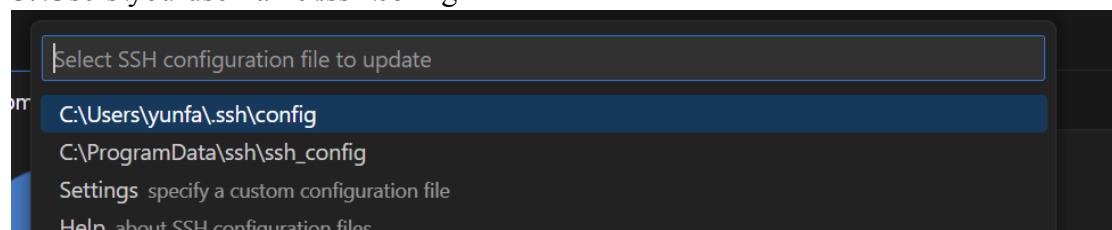
```
ssh-rsa AAAAB3NzaC1y3F5AAAQABgQDLwpixd6QK3D9q15mpcU2TFkdEo3HILXo5HCMjH5x8wPUjCAz1FgkTMyoVtTlYoxwAyyGdV7RFQ$8vz8eDor8Q
```

The screenshot shows the VS Code interface with the 'authorized_keys' file open in the terminal tab. The file contains a single line of SSH public key data. The terminal output shows the server starting and listening on port 2198. The bottom status bar shows 'Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Plain Text'.

After you added the public key to your VM `~/.ssh/authorized_keys`, click the **gear settings button** on the right side of the ssh selection bar in the remote development interface (pointed by the red arrow) as shown below:



The following menu will appear at the top of the window, select:
`C:\Users\yourusername\.ssh\config`



Find your specified VM and add the **IdentifyFile** line at the end, the following value is the path of the private key file corresponding to your public key.

When you use the default keys `id_ed25519` and `id_rsa`, please remember **not to add the **IdentifyFile** line.

```
config
C: > Users > xu729 > .ssh > config
12  # 650 Local
13  Host 650 local
14    HostName 192.168.56.107
15    User ece650
16    IdentityFile C:\Users\yourusername\.ssh\yourprivatekeyfile
17
```

After completing the above steps, you can now focus on your project5!