EE211 24Fall Lab2

Content

1. **Prepare your OS**
2. **ROS2**

1 Prepare your OS - Local Machine

For better development experience,

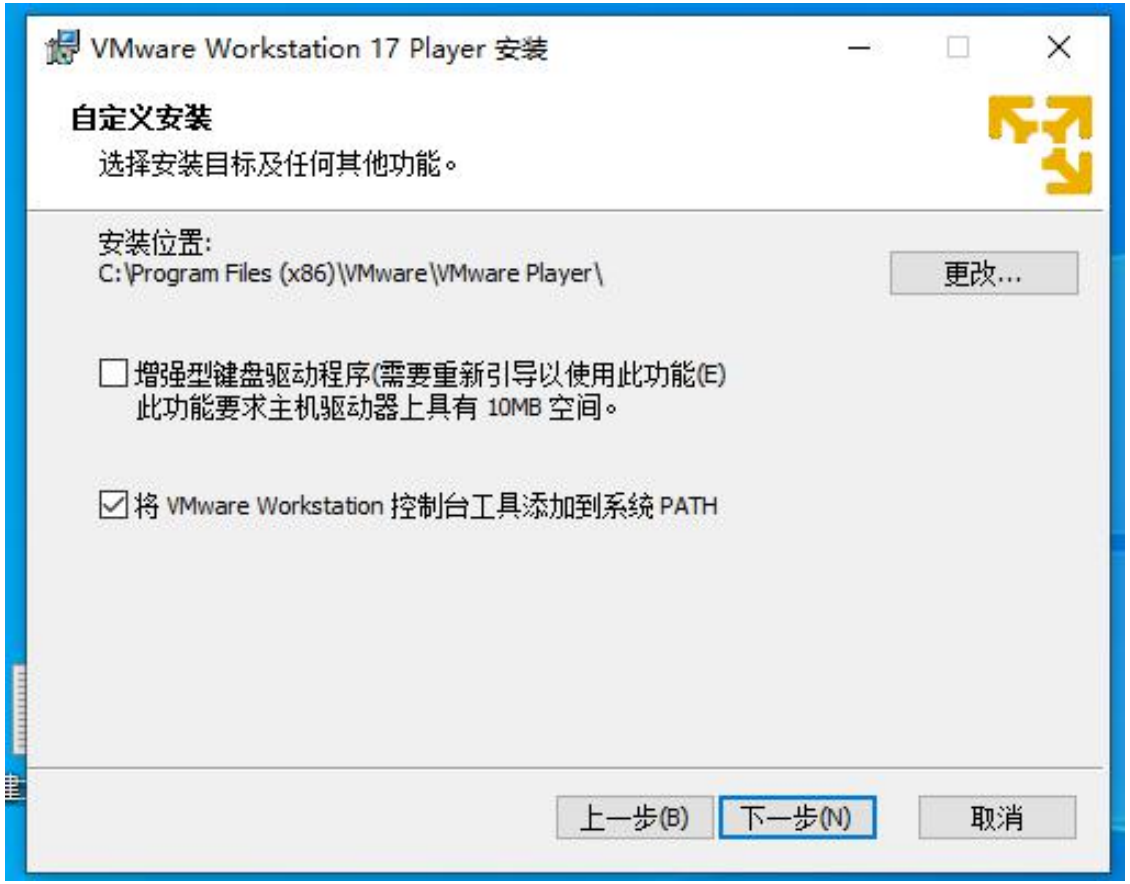
using a local ubuntu machine is recommended!

You may install a separate ubuntu os on your windows PC, so that you can using win and ubuntu on the same computer!

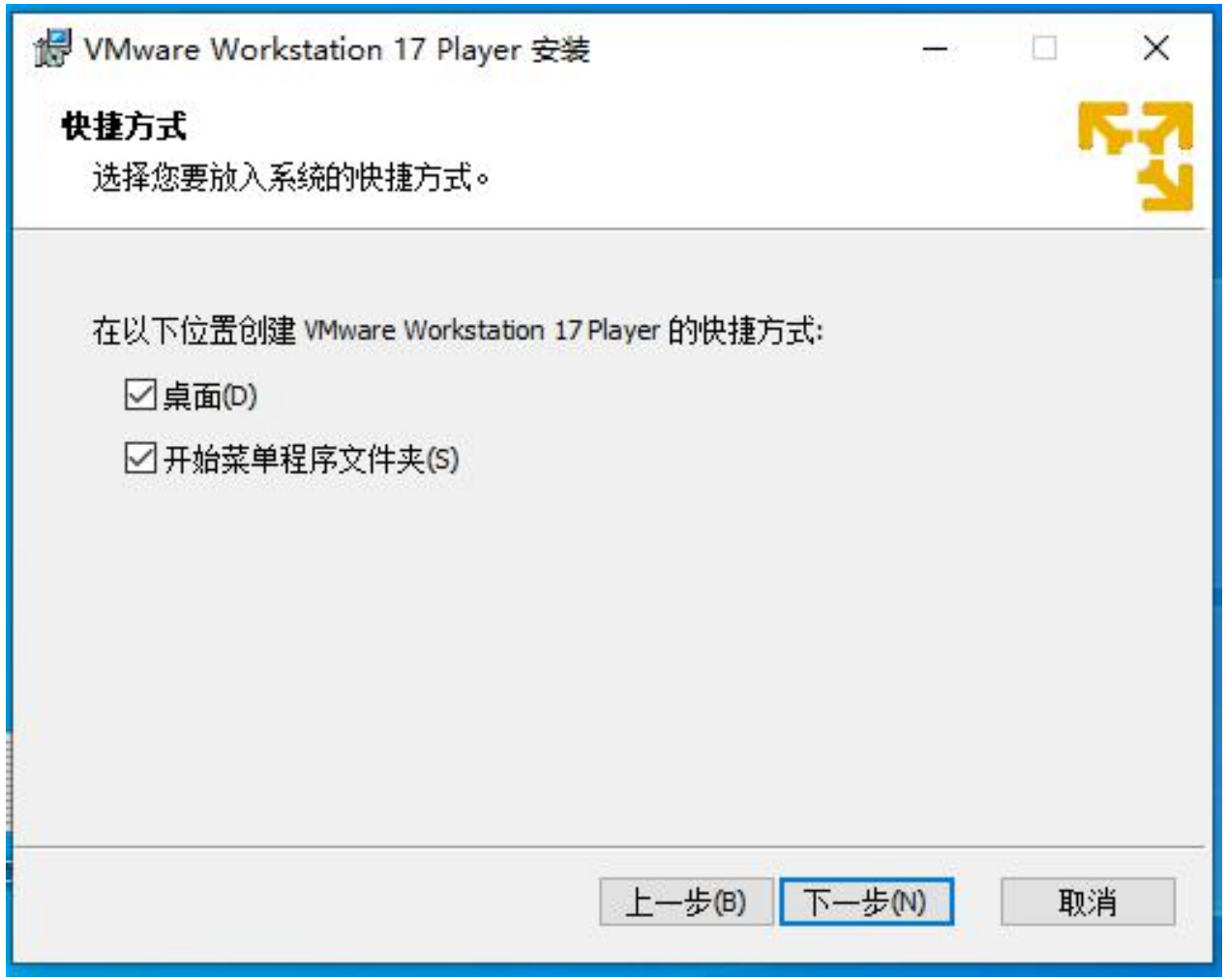
1 Prepare your OS - Virtual Machine

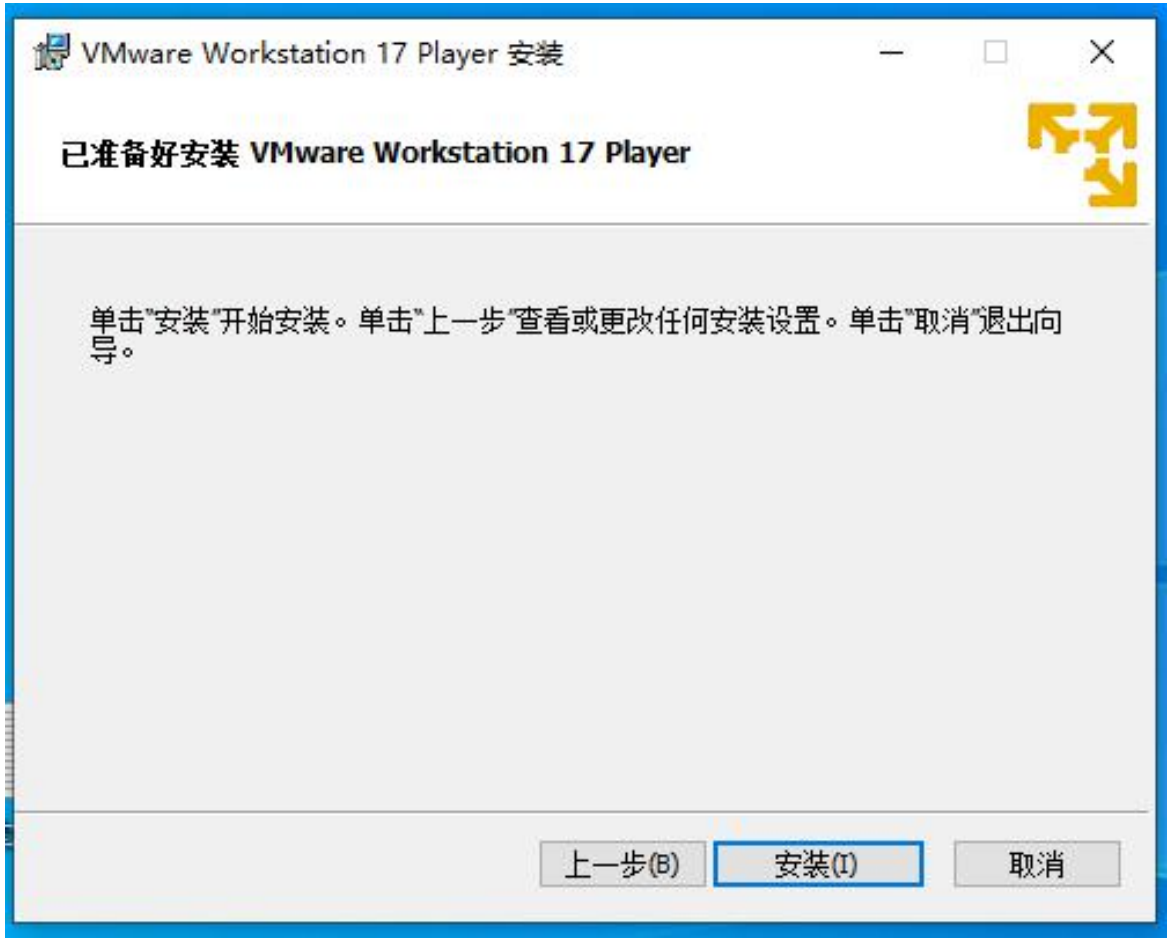
[VMware Workstation Player - VMware Customer Connect]

1 Prepare your OS - Virtual Machine

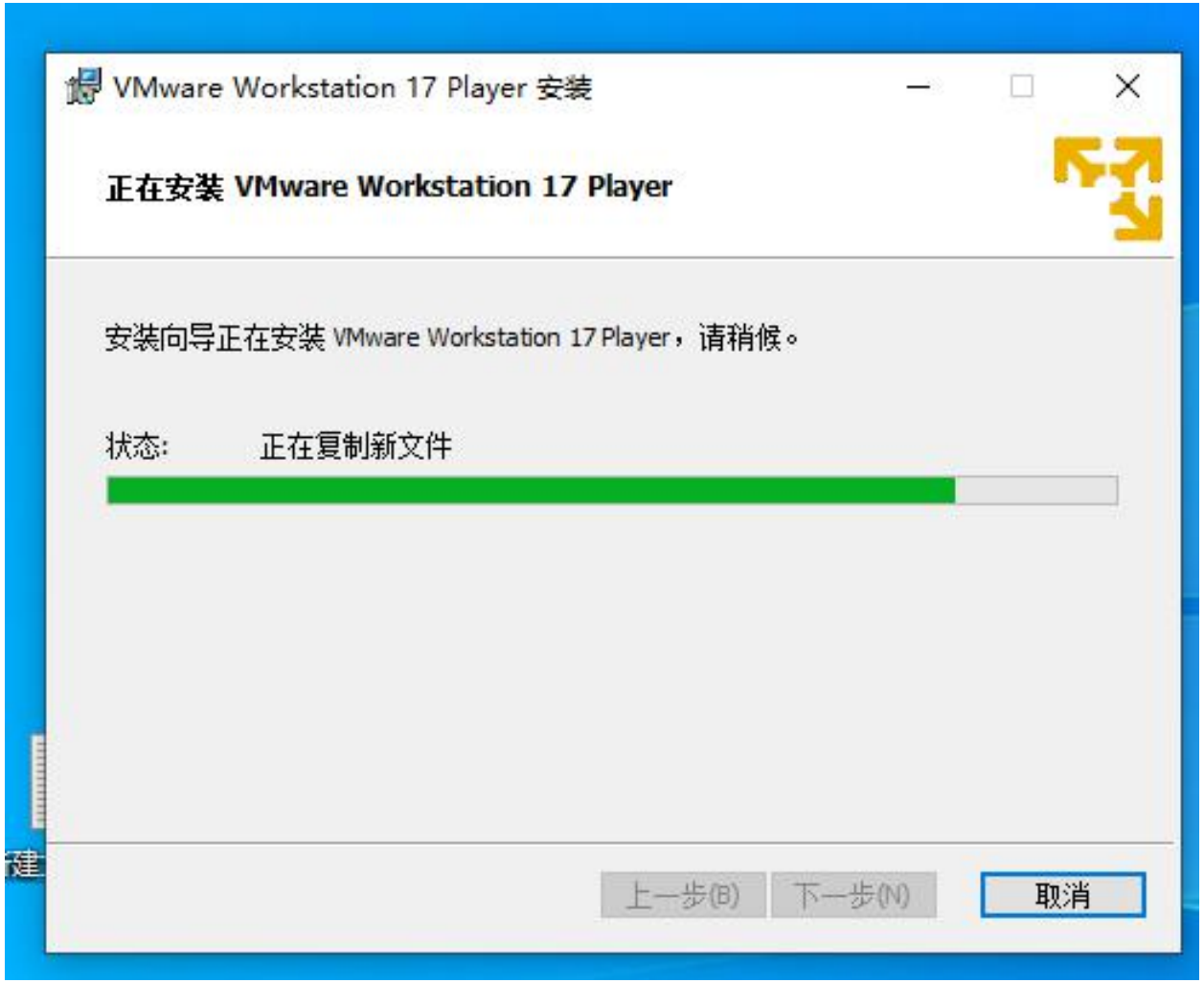


1 Prepare your OS - Virtual Machine



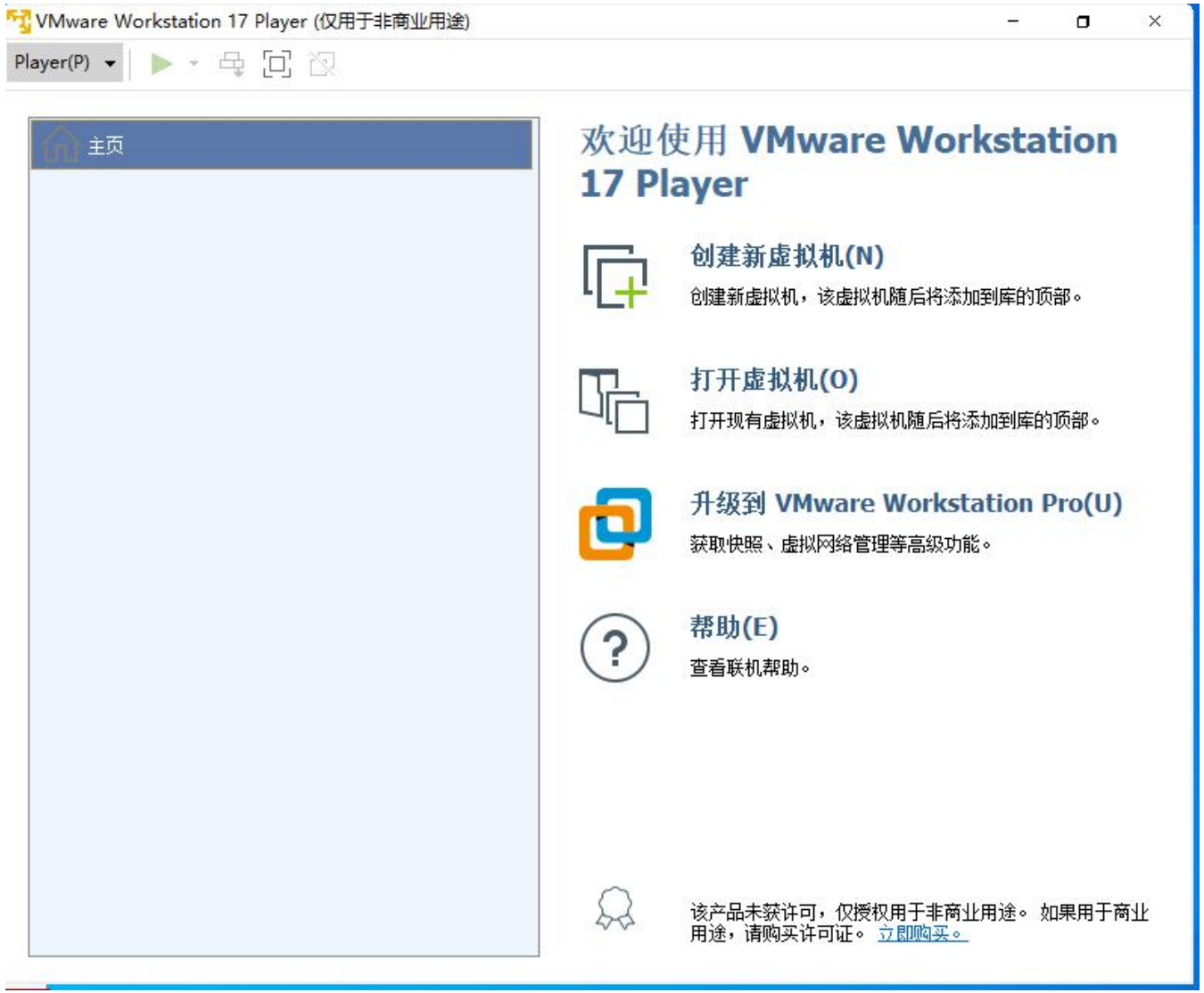


1 Prepare your OS - Virtual Machine

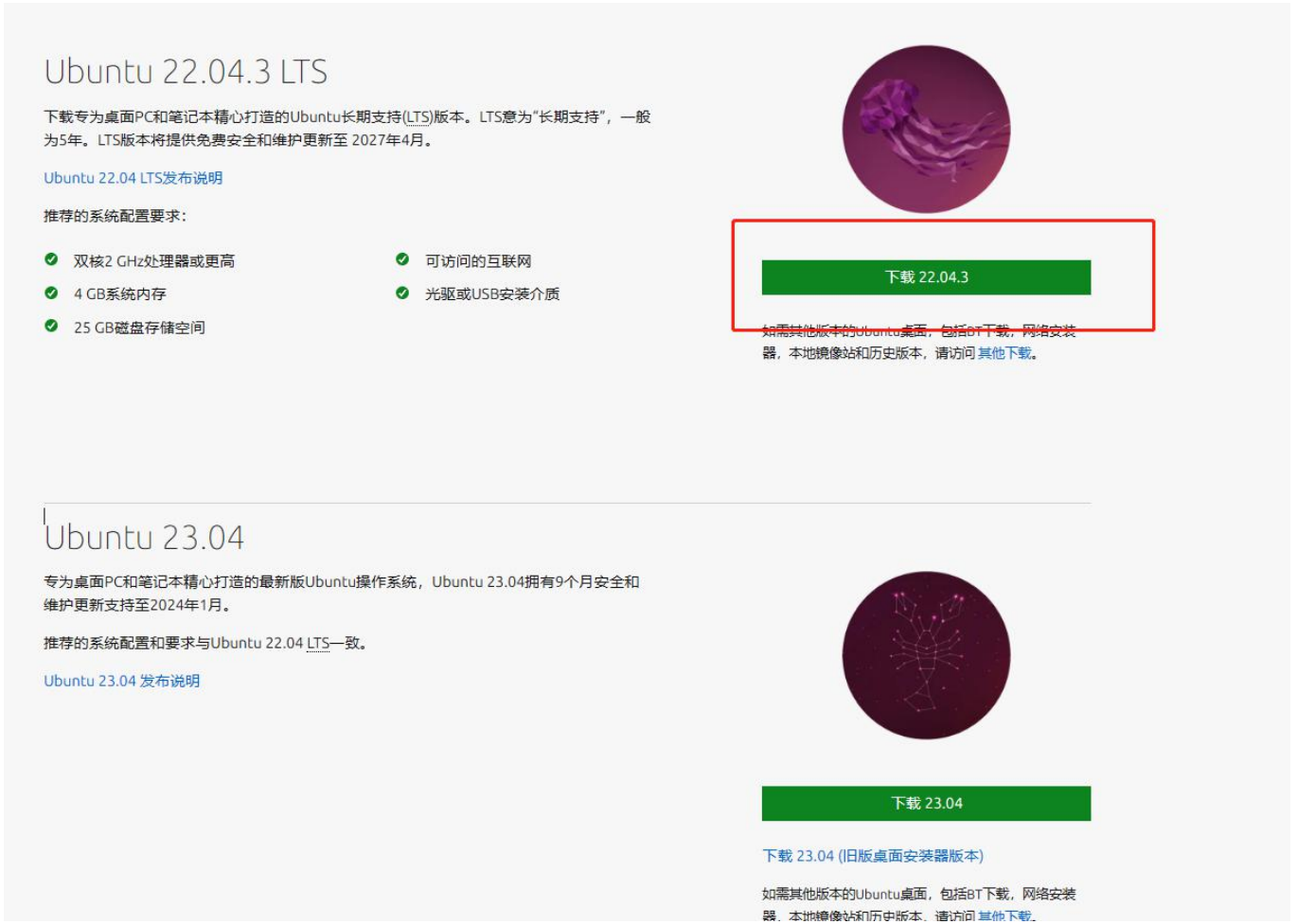


1 Prepare your OS - Virtual Machine

1 Prepare your OS - Virtual Machine

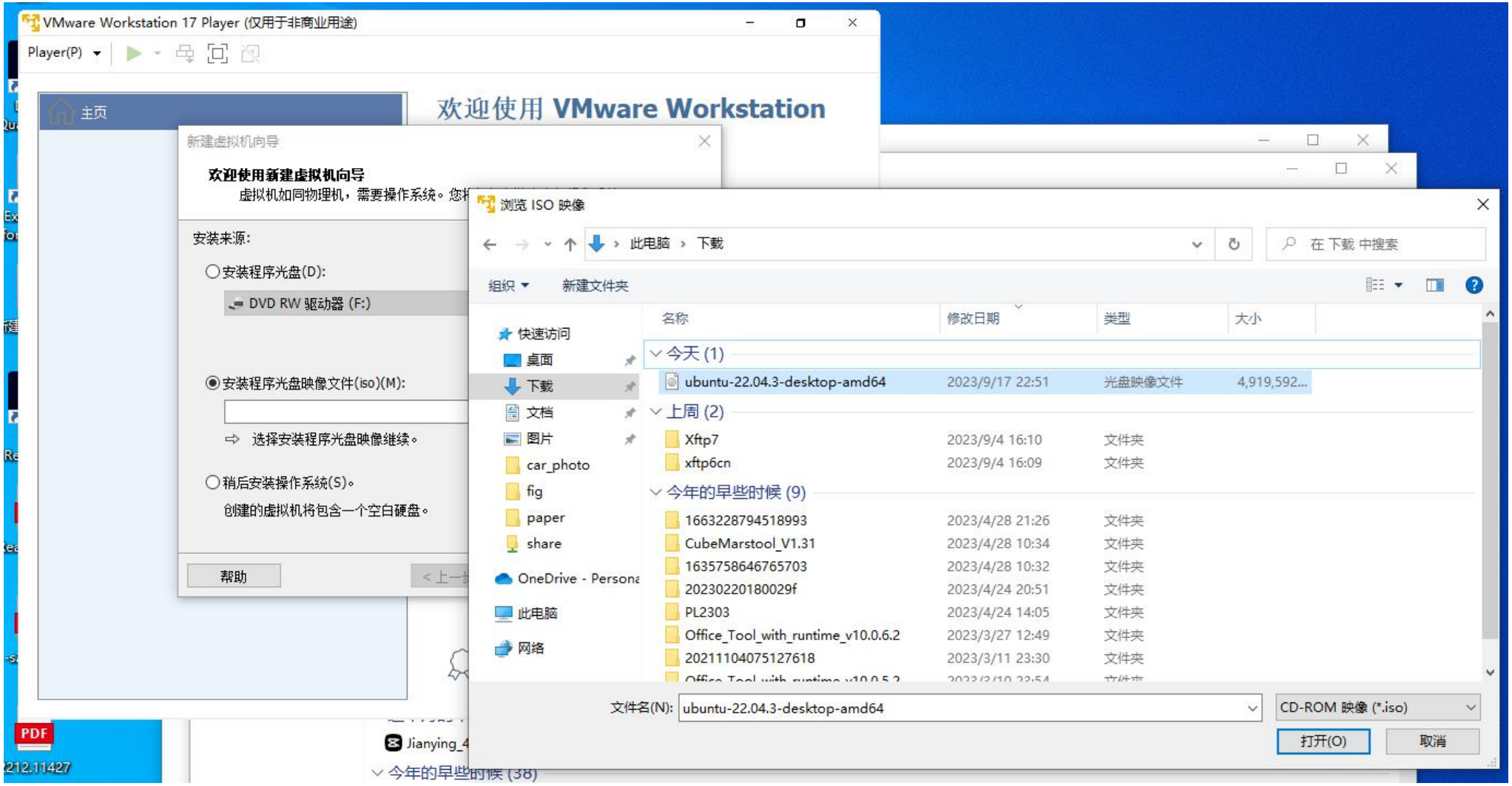


1 Prepare your OS - Virtual Machine

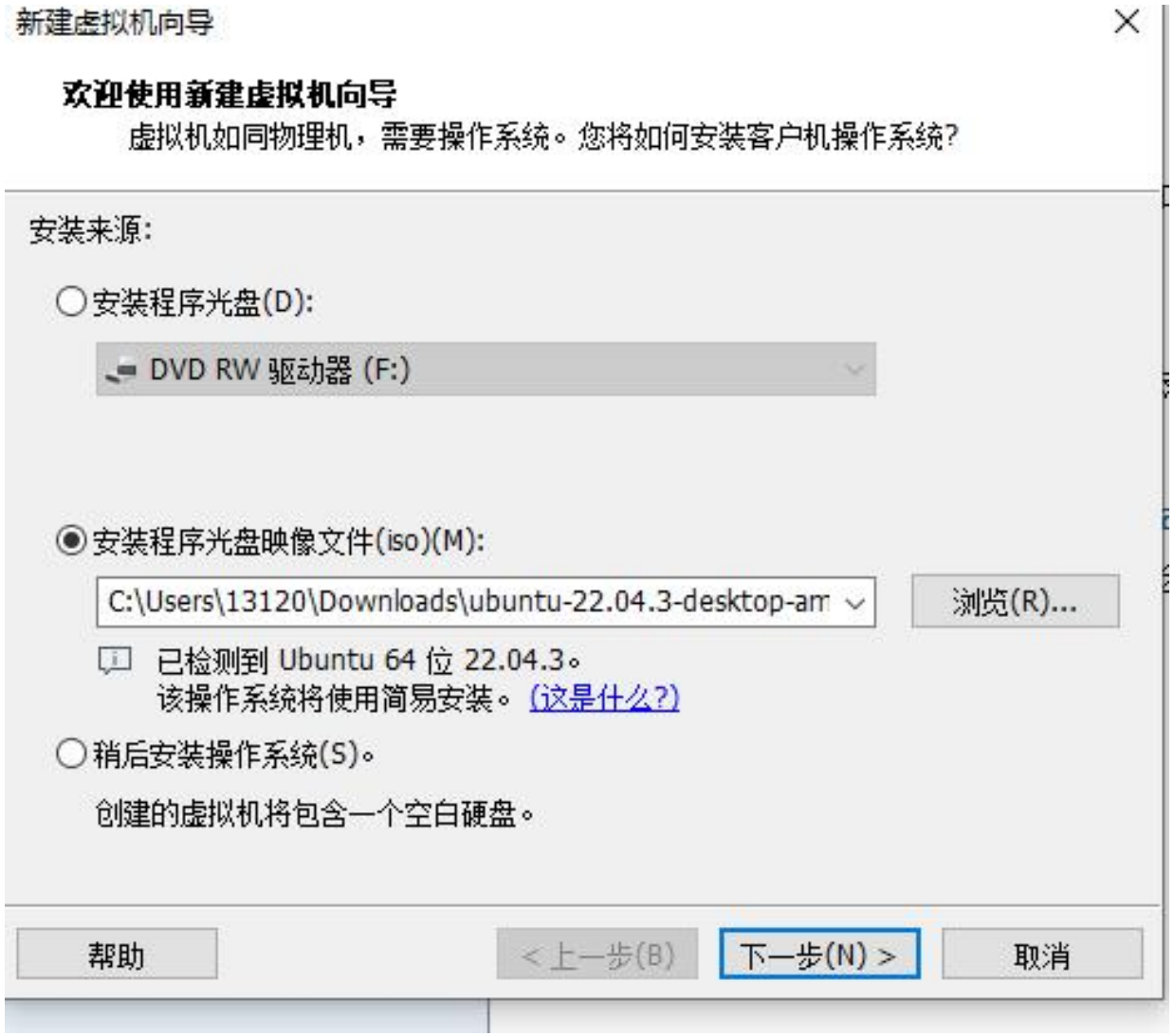


[下载Ubuntu桌面系统 | Ubuntu]

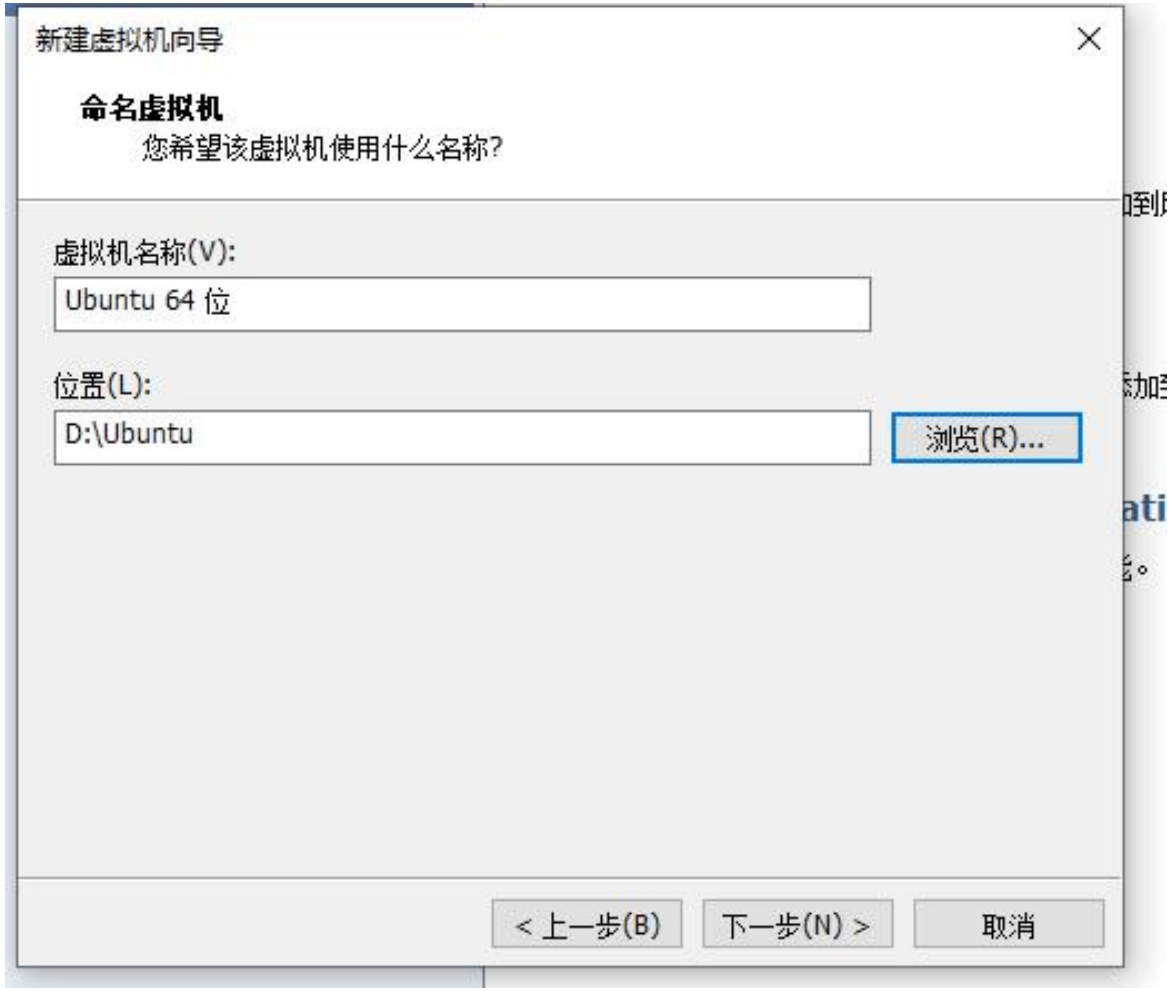
1 Prepare your OS - Virtual Machine

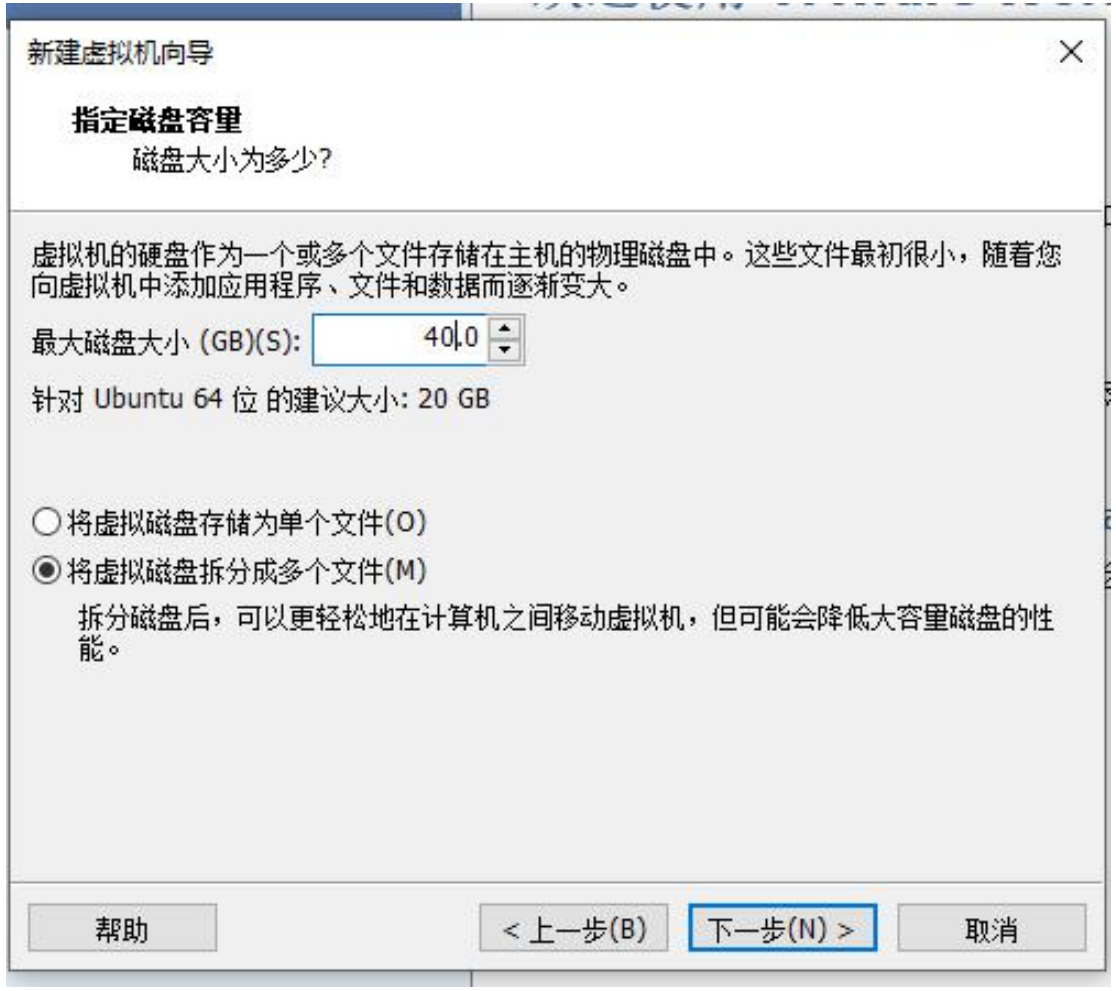


1 Prepare your OS - Virtual Machine

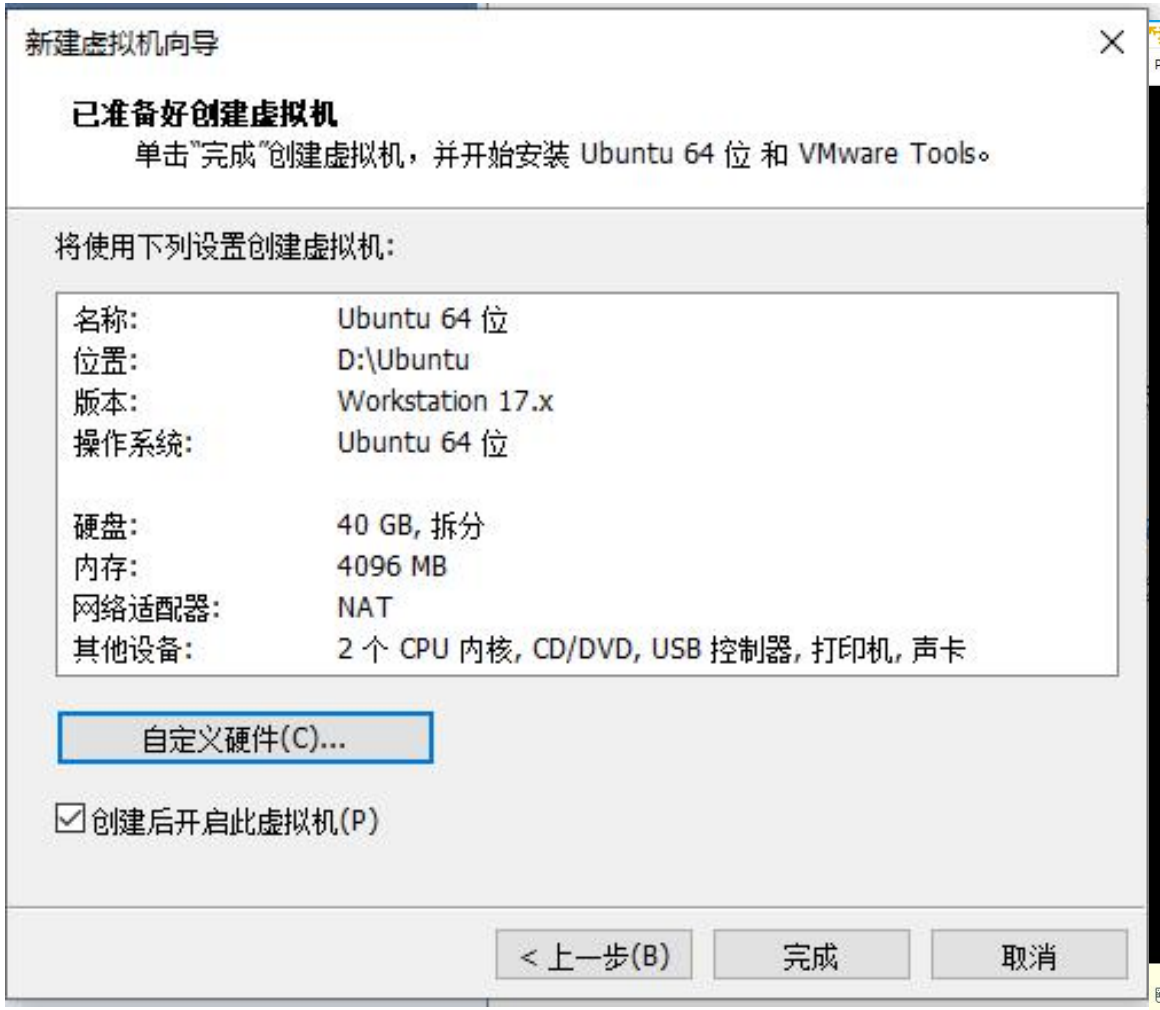


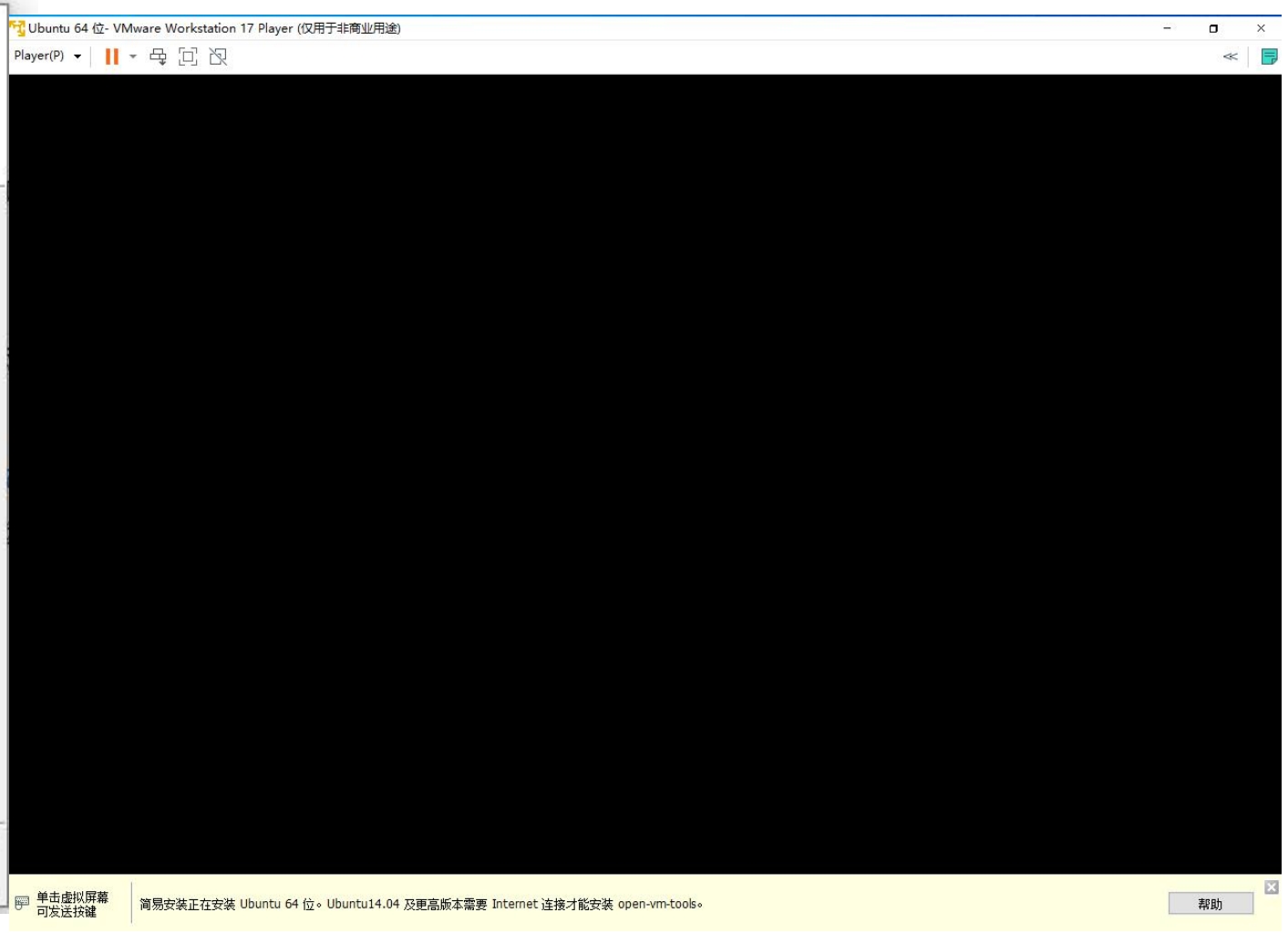
1 Prepare your OS - Virtual Machine



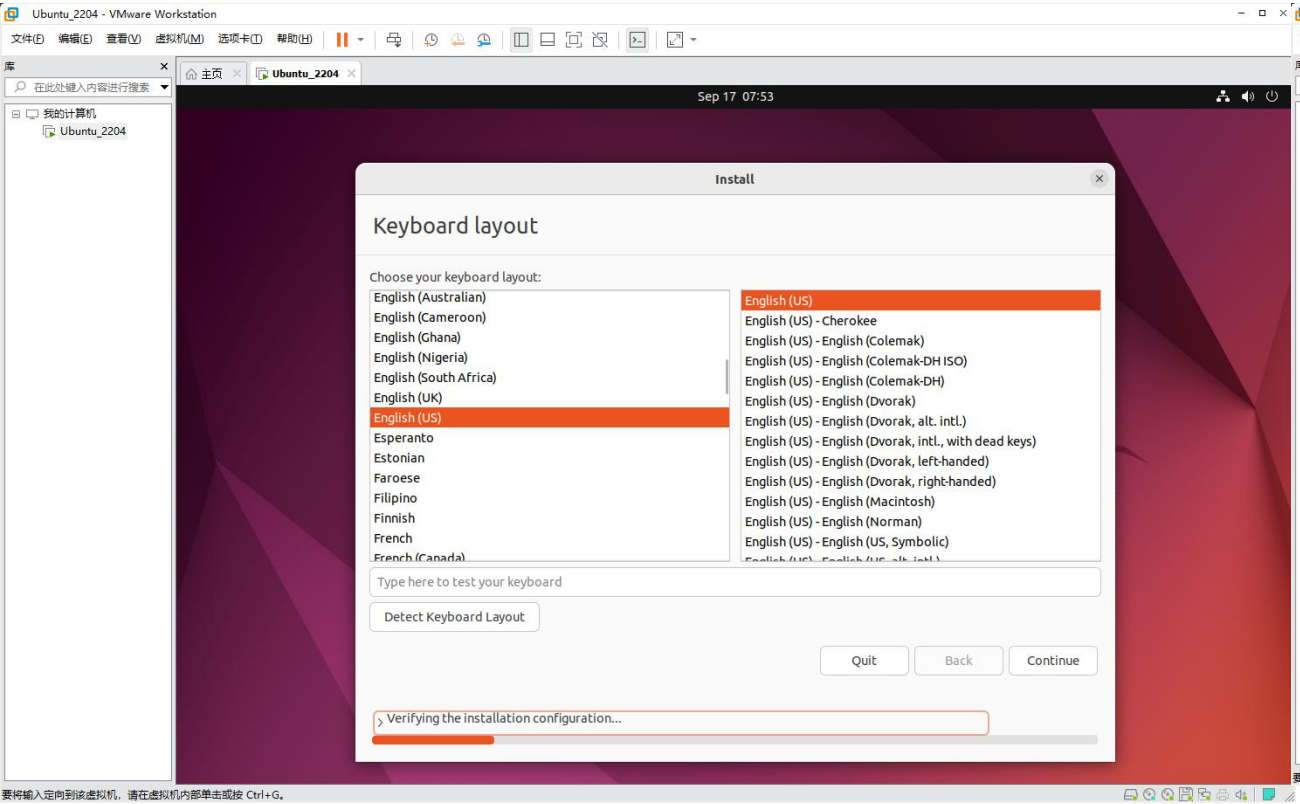


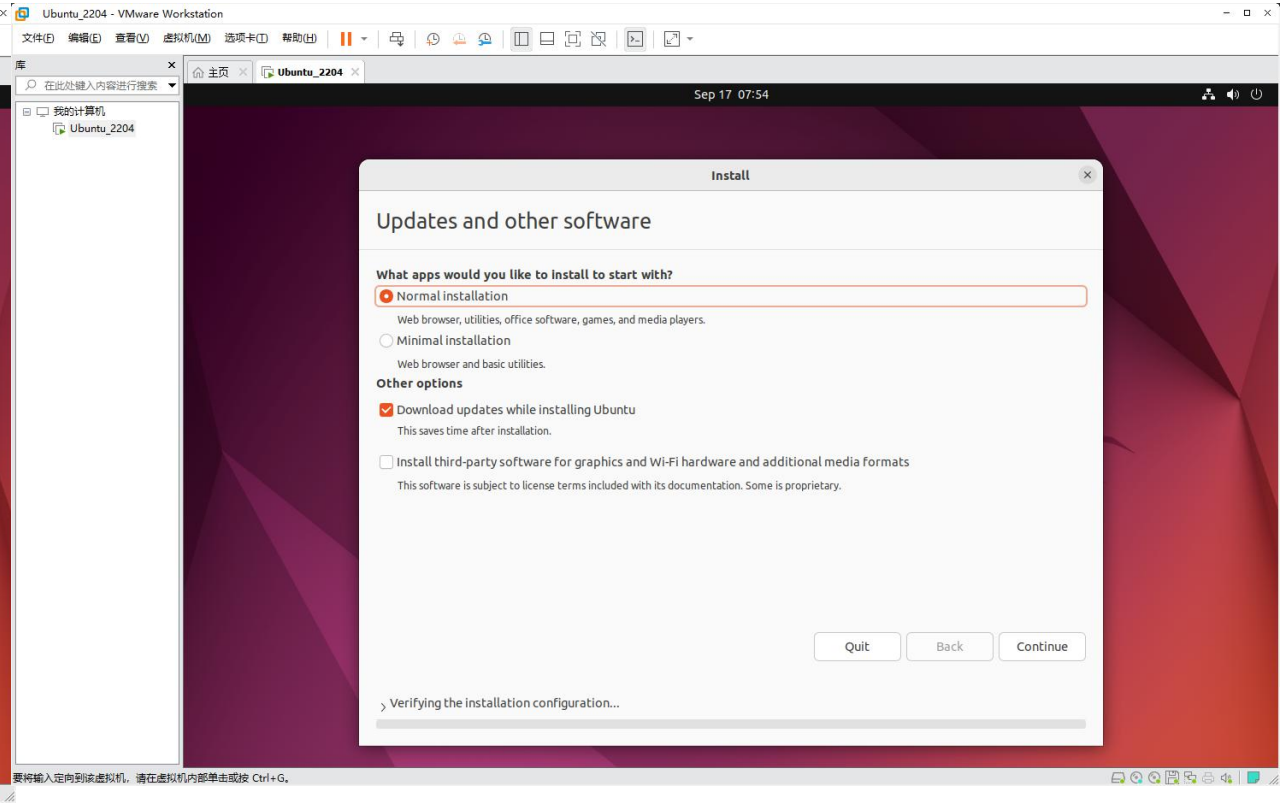
1 Prepare your OS - Virtual Machine



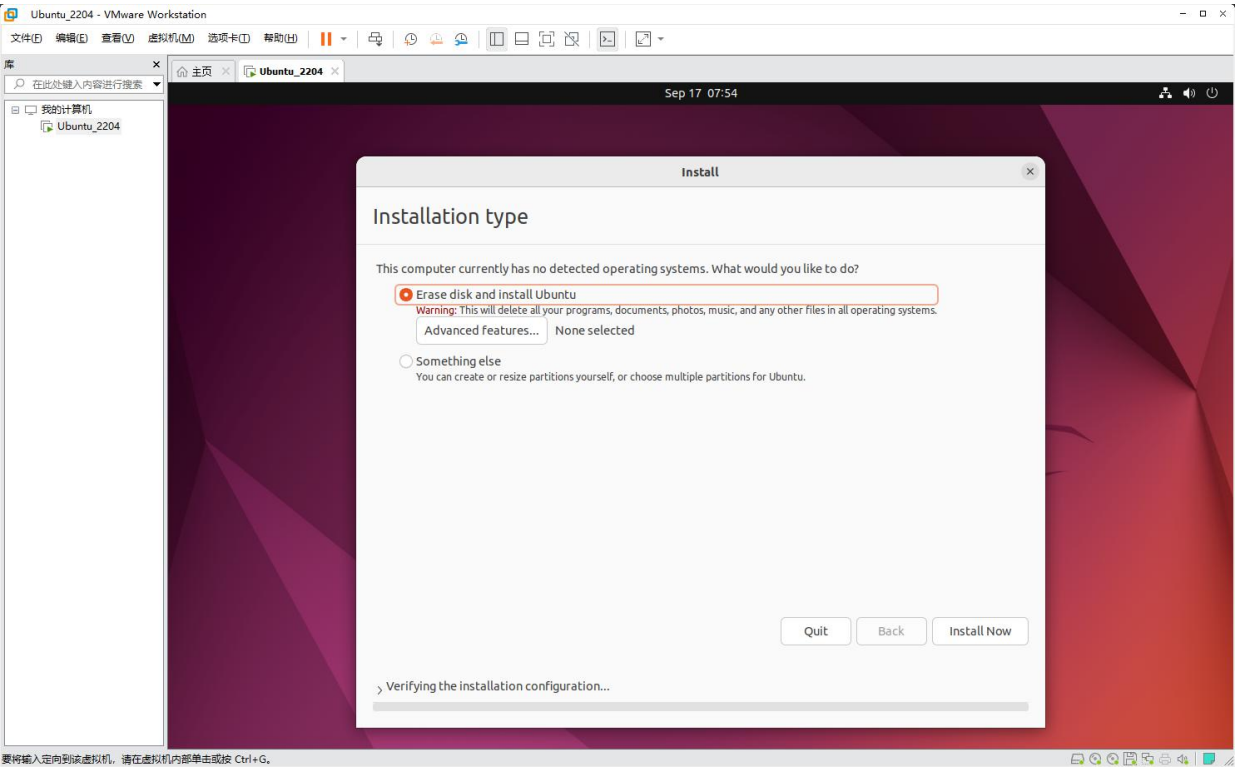


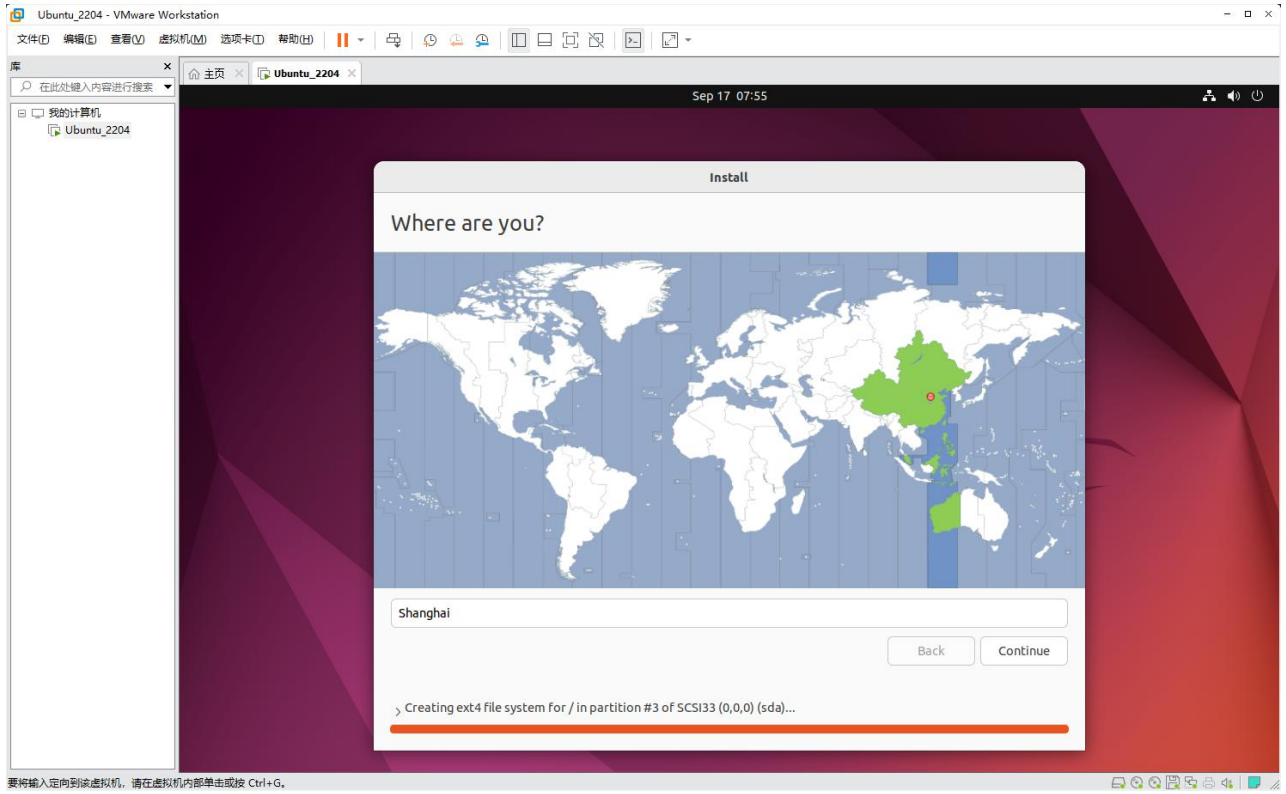
1 Prepare your OS - Virtual Machine



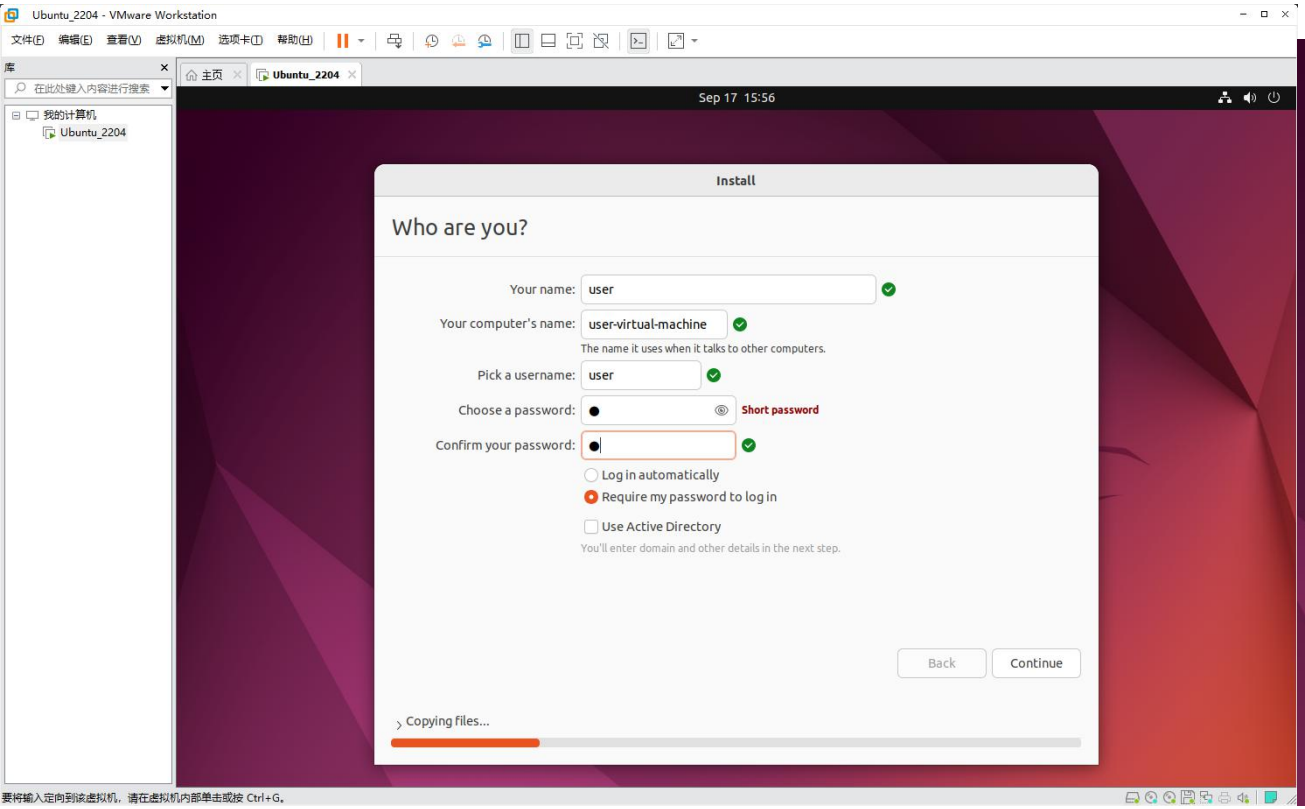


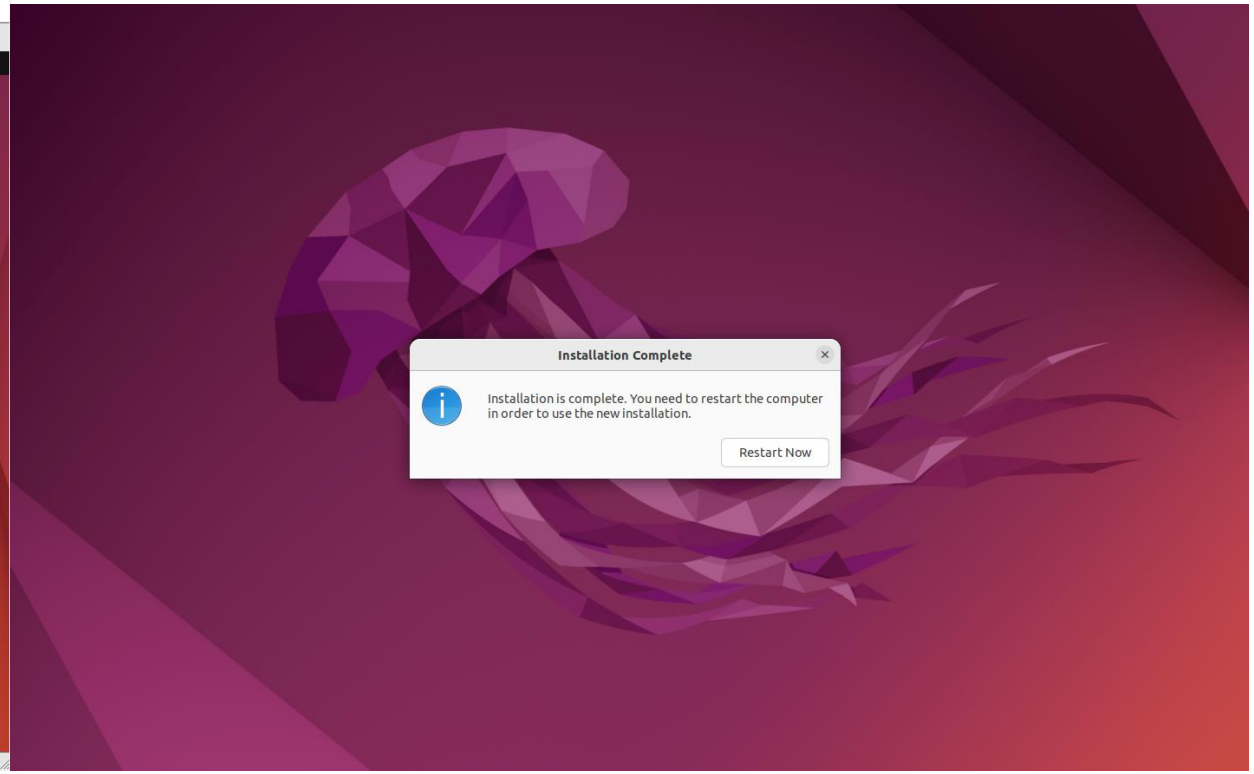
1 Prepare your OS - Virtual Machine



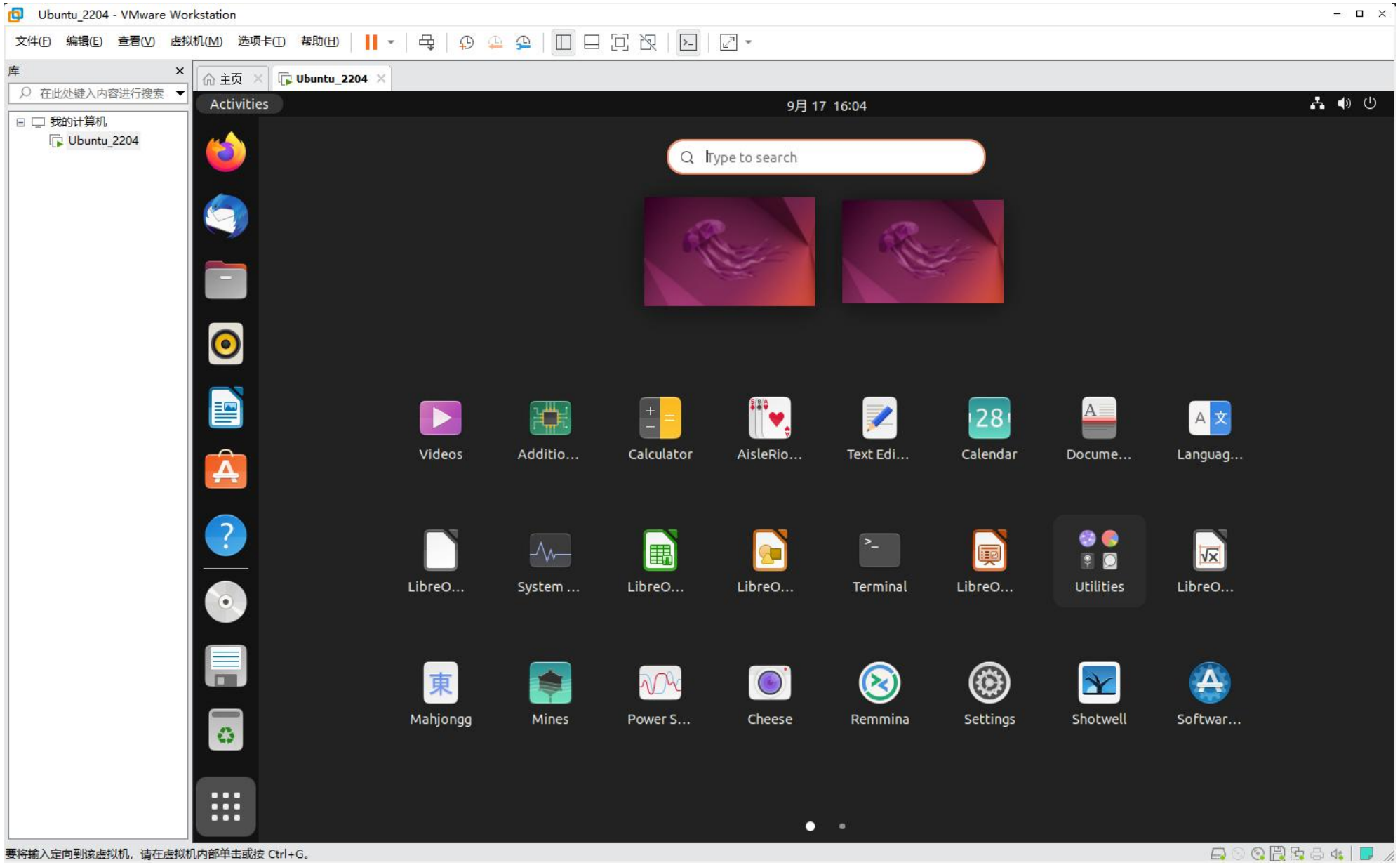


1 Prepare your OS - Virtual Machine





1 Prepare your OS - Virtual Machine



1. Setting up Encoding

```
$ sudo apt update && sudo apt install locales
$ sudo locale-gen en_US en_US.UTF-8
$ sudo update-locale LC_ALL=en_US.UTF-8LANG=en_US.UTF-8
$ export LANG=en_US.UTF-8
```

2. Setting apt source (https://mirror.tuna.tsinghua.edu.cn/help/ros2/)

```
$ sudo apt install curl gnupg2
$ sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key  -o
/usr/share/keyrings/ros-archive-keyring.gpg
$ sudo echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/ros-archive-keyring.gpg]
https://mirrors.tuna.tsinghua.edu.cn/ros2/ubuntu jammy main" | sudo tee
/etc/apt/sources.list.d/ros2.list > /dev/null
$sudo apt update
$sudo apt upgrade
```

1. Install

```
$ sudo apt install ros-humble-desktop
$ sudo apt install python3-colcon-common-extensions python3-argcomplete python3-rosdep
$ sudo apt install git
$ echo " source /opt/ros/humble/setup.bash" >> ~/.bashrc
```

2. Uninstall

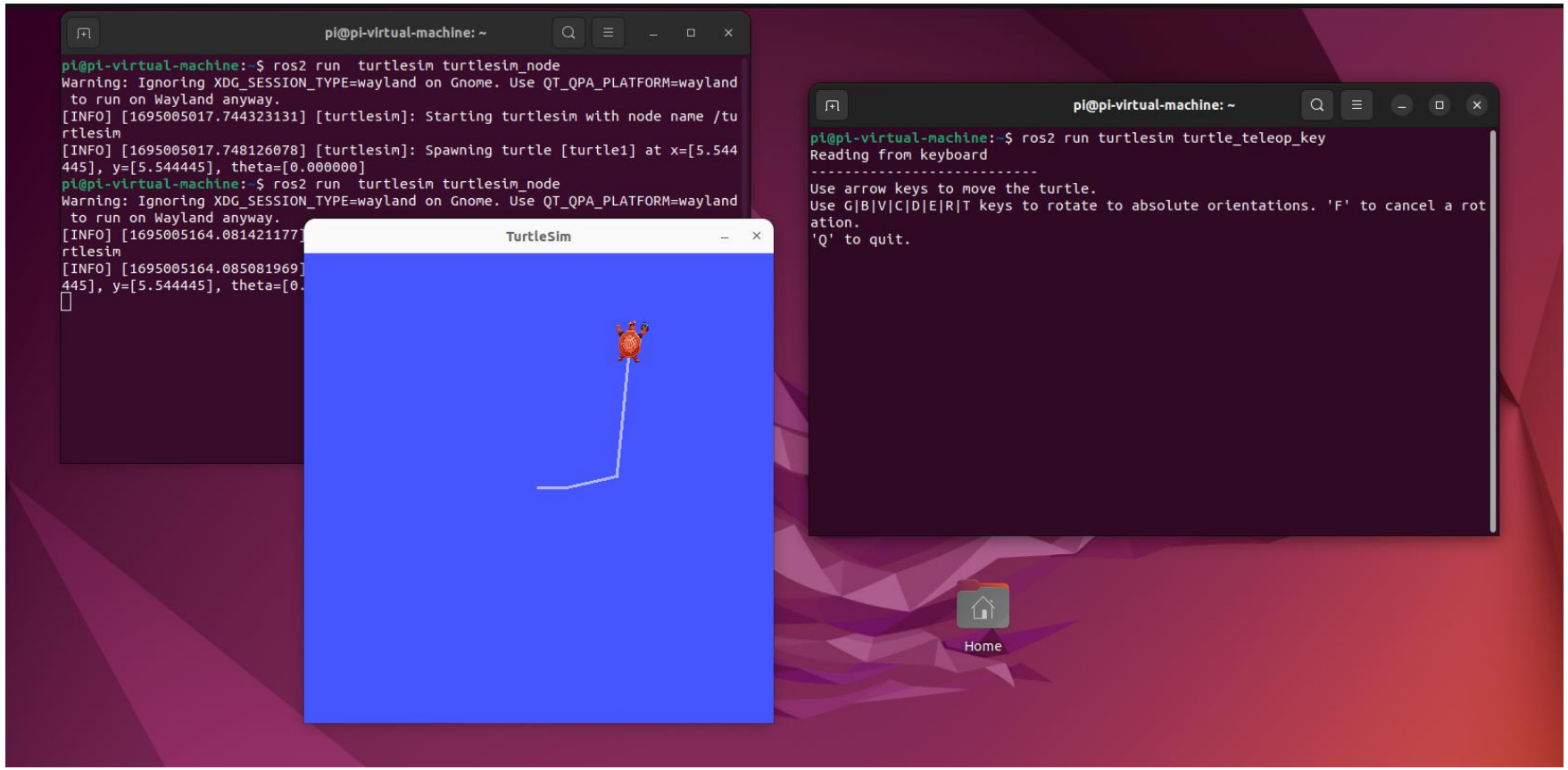
```
$ sudo apt remove ros-humble-*
```

*

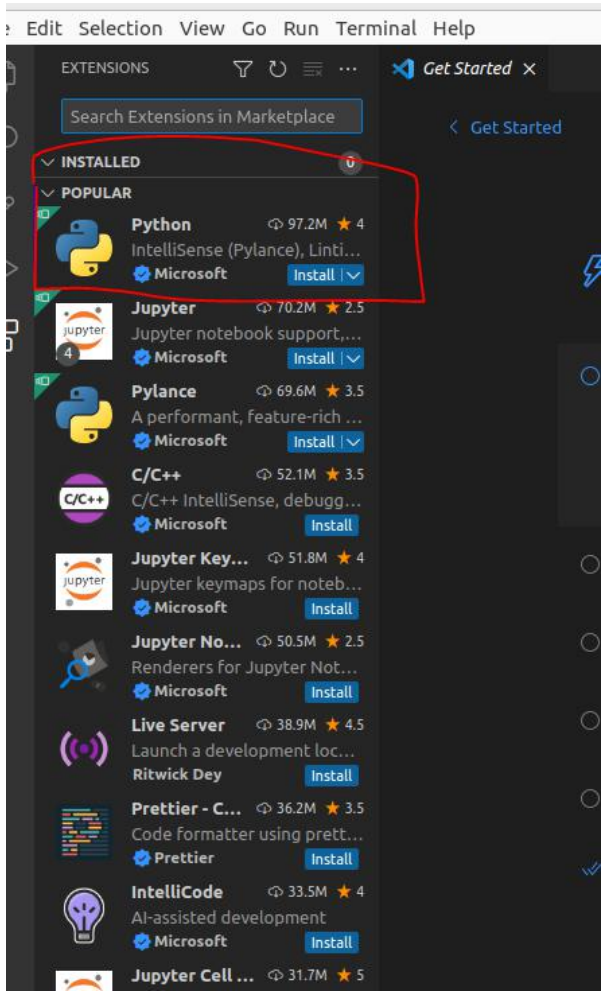
```
$ wget http://fishros.com/install -O fishros && . fishros
```

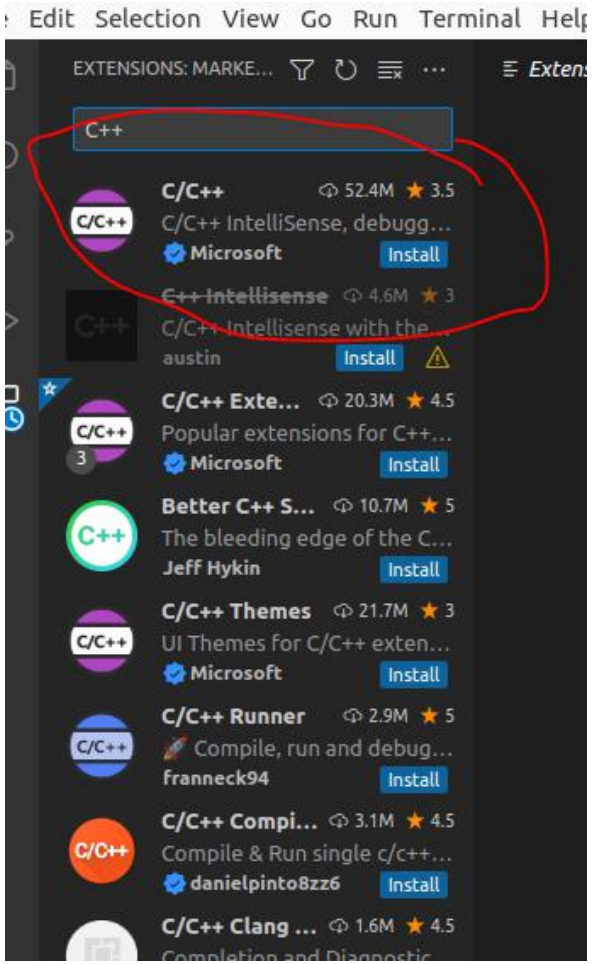
1. Run first ROS2 node

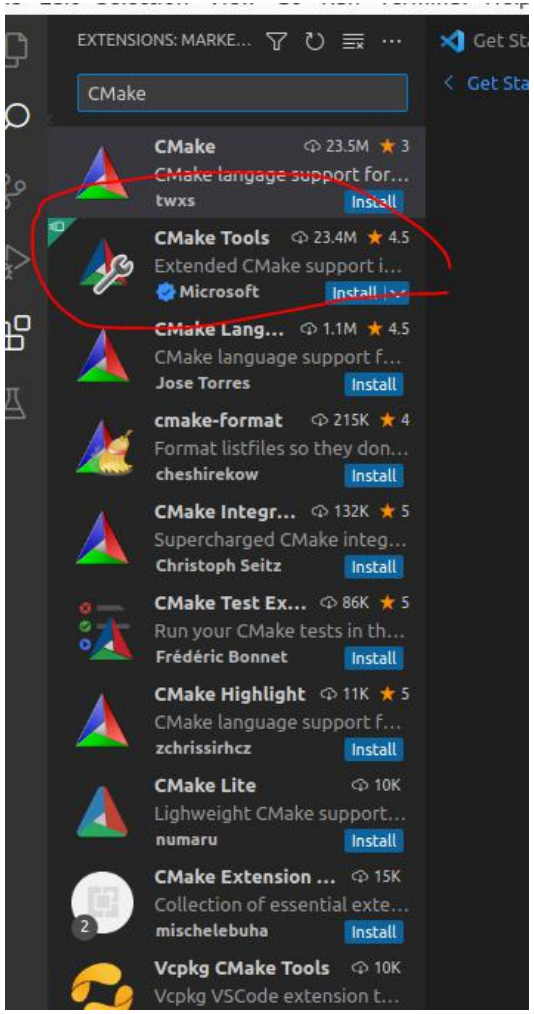
```
$ ros2 run  turtlesim turtlesim_node
$ ros2 run turtlesim turtle_teleop_key
```



VScode Setting







Common command operations

cd <dir>

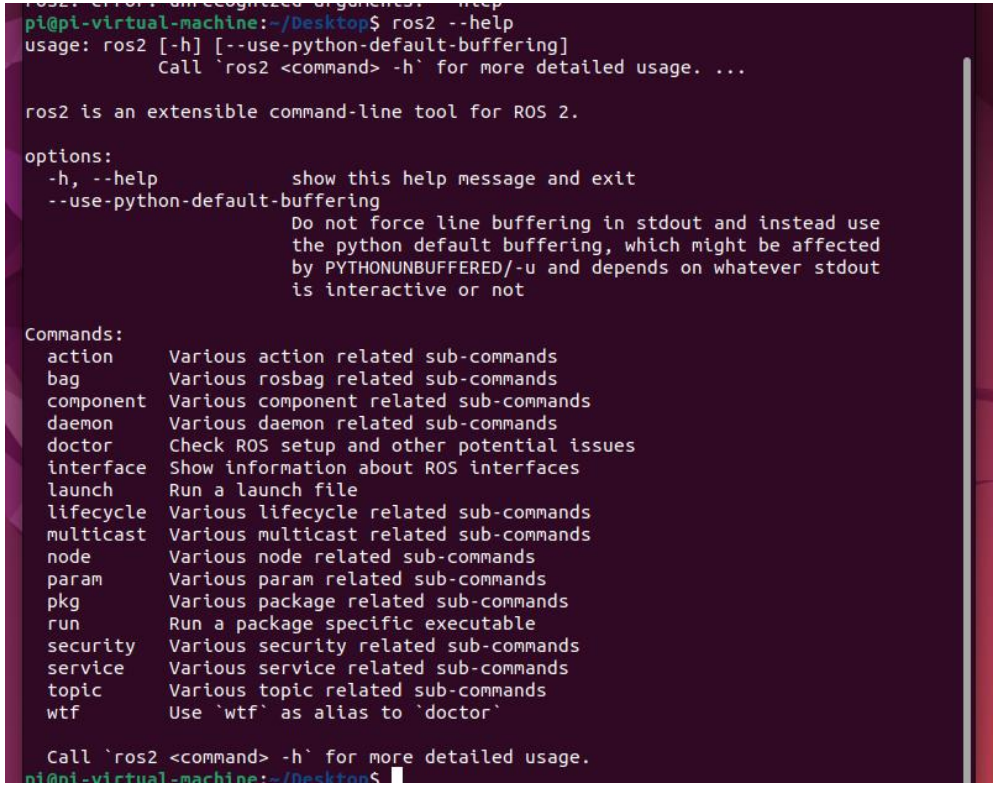
mkdir [option] <dir>

cp [option] <source> <target>

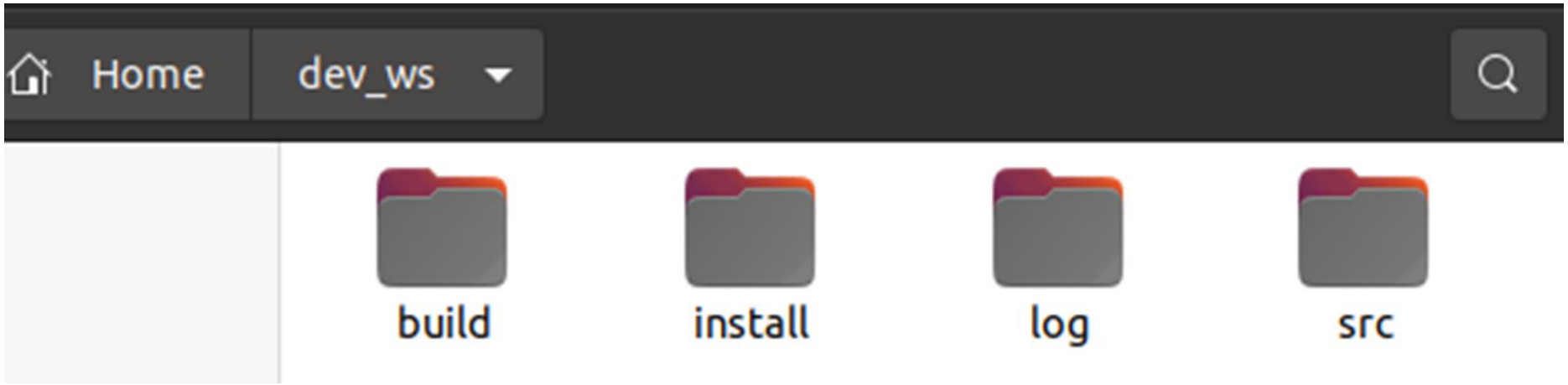
gedit <file>

mv [option] <source> <target>

rm [option] <file/dir…>



WorkSpace



Create WorkSpace

```
$ mkdir -p ~/dev_ws/src
$ cd ~/dev_ws/src
$ git clone https://gitee.com/guyuehome/ros2_21_tutorials.git
```

1.Installing dependencies

```
$ sudo apt install -y python3-pip
$ sudo pip3 install rosdepc
$ sudo rosdepc init
$ rosdepc update
$ cd ..
$ rosdepc install -i --from-path src --rosdistro humble -y
```

2.Building

```
$ sudo apt install python3-colcon-ros
$ cd ~/dev_ws/
$ colcon build
```

3.Source

```
$ source install/local_setup.sh
```

1.Create your first package

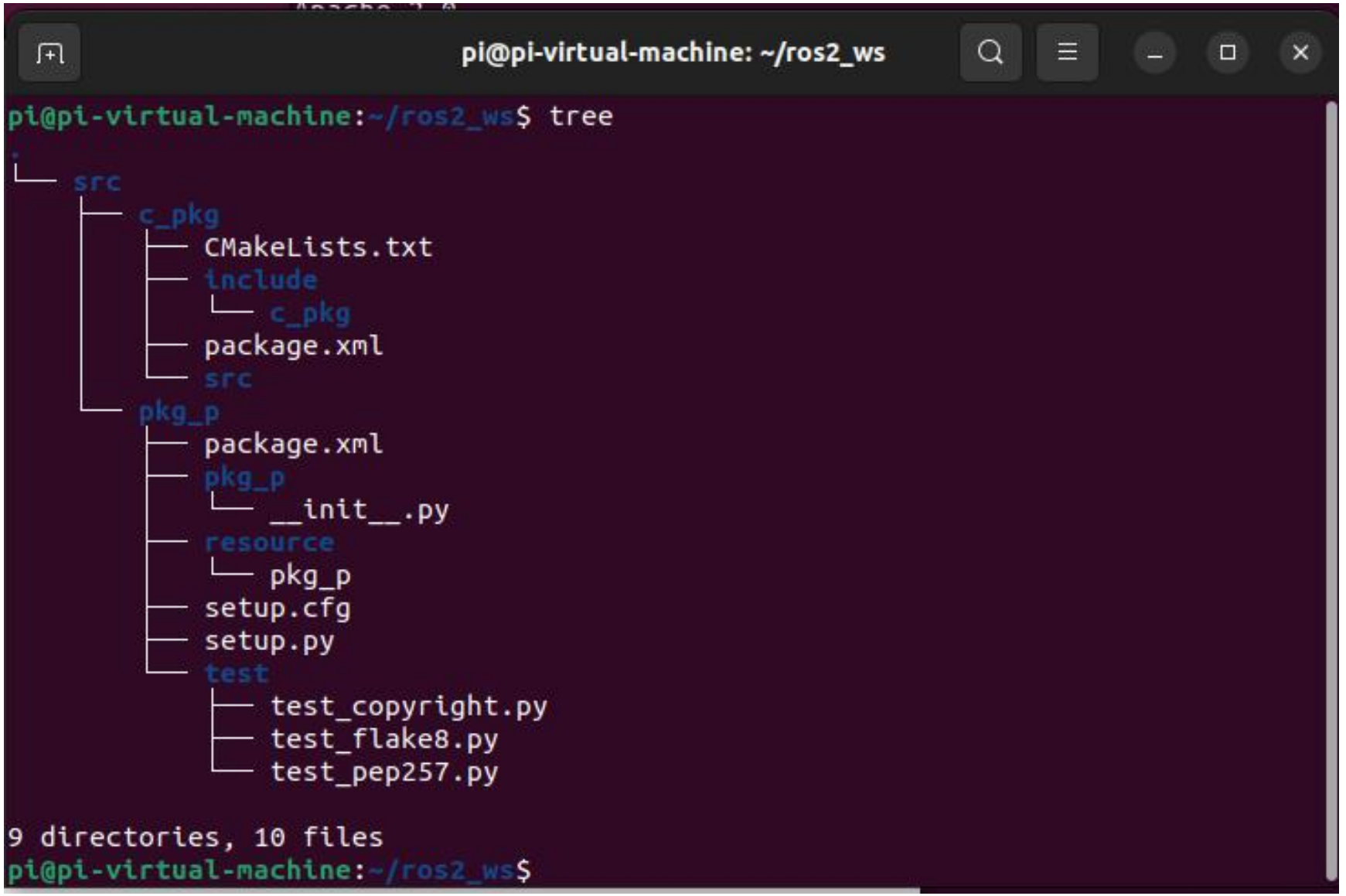
```
$ ros2 pkg create --build-type <build-type>  <package_name>
```

<build-type>

C/C++: ament_cmake

Python: ament_python

```
$ cd ~/dev_ws/src
$ ros2 pkg create --build-type ament_cmake c_pkg --dependencies rclcpp std_msgs
$ ros2 pkg create --build-type ament_python pkg_p --dependencies rclpy std_msgs
```


pi@pi-virtual-machine: ~/ros2_ws
pi@pi-virtual-machine:~/ros2_ws$ tree
.
└── src
    ├── c_pkg
    │   ├── CMakeLists.txt
    │   ├── include
    │   │   └── c_pkg
    │   ├── package.xml
    │   └── src
    └── pkg_p
        ├── package.xml
        ├── pkg_p
        │   └── __init__.py
        ├── resource
        │   └── pkg_p
        ├── setup.cfg
        ├── setup.py
        └── test
            ├── test_copyright.py
            ├── test_flake8.py
            └── test_pep257.py
9 directories, 10 files
pi@pi-virtual-machine:~/ros2_ws$

Open | CMakeLists.txt ~/ros2_ws/src/c_pkg | Save

```
cmake_minimum_required(VERSION 3.8)
project(c_pkg)

if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
  add_compile_options(-Wall -Wextra -Wpedantic)
endif()

# find dependencies
find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(std_msgs REQUIRED)

if(BUILD_TESTING)
  find_package(ament_lint_auto REQUIRED)
  # the following line skips the linter which checks for copyrights
  # comment the line when a copyright and license is added to all source files
  set(ament_cmake_copyright_FOUND TRUE)
  # the following line skips cpplint (only works in a git repo)
  # comment the line when this package is in a git repo and when
  # a copyright and license is added to all source files
  set(ament_cmake_cpplint_FOUND TRUE)
  ament_lint_auto_find_test_dependencies()
endif()

ament_package()
```

CMake | Tab Width: 8 | Ln 1, Col 1 | INS

Open | package.xml ~/ros2_ws/src/c_pkg | Save

```
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>c_pkg</name>
  <version>0.0.0</version>
  <description>TODO: Package description</description>
  <maintainer email="pi@todo.todo">pi</maintainer>
  <license>TODO: License declaration</license>

  <buildtool_depend>ament_cmake</buildtool_depend>

  <depend>rclcpp</depend>
  <depend>std_msgs</depend>

  <test_depend>ament_lint_auto</test_depend>
  <test_depend>ament_lint_common</test_depend>

  <export>
    <build_type>ament_cmake</build_type>
  </export>
</package>
```

XML | Tab Width: 8 | Ln 1, Col 1 | INS

Open Save **setup.py** ~/ros2_ws/src/pkg_p

```python
from setuptools import find_packages, setup

package_name = 'pkg_p'

setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='pi',
    maintainer_email='pi@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
        ],
    },
)
```

Python 2 Tab Width: 8 Ln 1, Col 1 INS

Open Save **package.xml** ~/ros2_ws/src/pkg_p

```xml
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>pkg_p</name>
  <version>0.0.0</version>
  <description>TODO: Package description</description>
  <maintainer email="pi@todo.todo">pi</maintainer>
  <license>TODO: License declaration</license>

  <depend>rclpy</depend>
  <depend>std_msgs</depend>

  <test_depend>ament_copyright</test_depend>
  <test_depend>ament_flake8</test_depend>
  <test_depend>ament_pep257</test_depend>
  <test_depend>python3-pytest</test_depend>

  <export>
    <build_type>ament_python</build_type>
  </export>
</package>
```

XML Tab Width: 8 Ln 1, Col 1 INS

1.Create first node (C++)

```
$ cd ~/ros2_ws/src/c_pkg/src/
$ touch talker.cpp
```

```
#include "rclcpp/rclcpp.hpp"

int main(int argc,char **argv)
{
    rclcpp::init(argc,argv);
    auto node = std::make_shared<rclcpp::Node>("node1");
    RCLCPP_INFO(node->get_logger(),"node1 is start!");
    rclcpp::spin(node);
    rclcpp::shutdown();
    return 0;
}
```

talker.cpp 2 | CMakeLists.txt

src > c_pkg > CMakeLists.txt

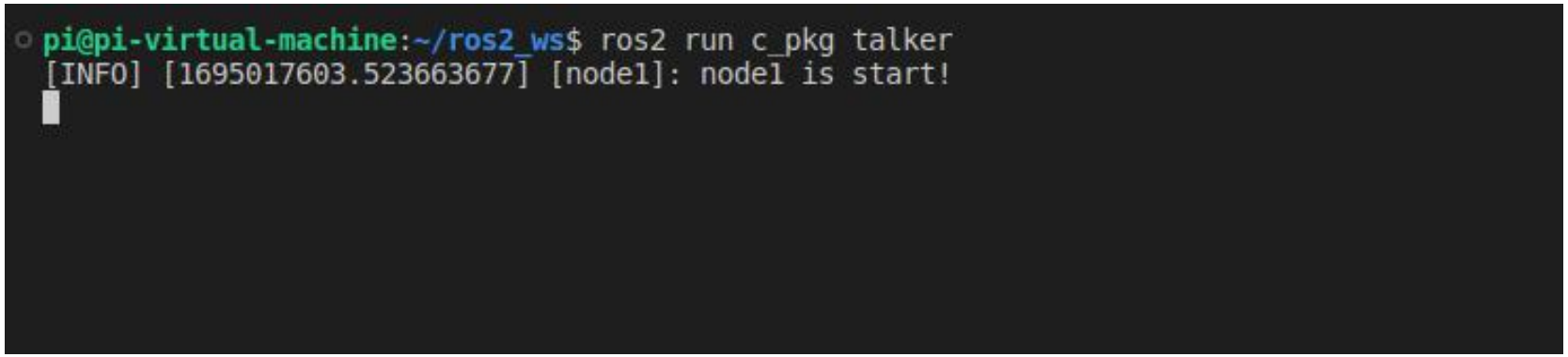
```
1  cmake_minimum_required(VERSION 3.8)
2  project(c_pkg)
3
4  if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5    add_compile_options(-Wall -Wextra -Wpedantic)
6  endif()
7
8  # find dependencies
9  find_package(ament_cmake REQUIRED)
10 find_package(rclcpp REQUIRED)
11 find_package(std_msgs REQUIRED)
12
13
14 add_executable(talker src/talker.cpp)
15 ament_target_dependencies(talker rclcpp)
16
17
18 install(TARGETS
19   talker
20   DESTINATION lib/${PROJECT_NAME}
21 )
22
23 if(BUILD_TESTING)
24   find_package(ament_lint_auto REQUIRED)
25   # the following line skips the linter which checks for copyrights
26   # comment the line when a copyright and license is added to all source files
27   set(ament_cmake_copyright_FOUND TRUE)
28   # the following line skips cpplint (only works in a git repo)
29   # comment the line when this package is in a git repo and when
30   # a copyright and license is added to all source files
31   set(ament_cmake_cpplint_FOUND TRUE)
32   ament_lint_auto_find_test_dependencies()
33 endif()
34
35 ament_package()
36
```

add_executable(talker src/talker.cpp)
ament_target_dependencies(talker rclcpp)

install(TARGETS
  talker
  DESTINATION lib/${PROJECT_NAME}
)

```
$ cd ~/ros2_ws/
$ colcon build
$ source install/setup.bash
```

```
$ros2 run  c_pkg talker
```



1.Create first package(python)

```
$ cd ~/ros2_ws/src/pkg_p/pkg_p/
$ touch listener.py
```

```
import rclpy
from rclpy.node import Node
import sys
def main(args=Node):
    rclpy.init(args=sys.argv)
    node = Node("node2")
    node.get_logger().info("node2 is start!")
    rclpy.spin(node=node)
    rclpy.shutdown()
```

pkg_p > setup.py > ...

```python
from setuptools import find_packages, setup

package_name = 'pkg_p'

setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='pi',
    maintainer_email='pi@todo.todo',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            "listener = pkg_p.listener:main"
        ],
    },
)

```

"listener = pkg_p.listener:main"

```
pi@pi-virtual-machine:~/ros2_ws$ ros2 run pkg_p listener
[INFO] [1695018532.918658796] [node2]: node2 is start!
```


南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY
Thanks for Listening