Smart Mobility Engineering Lab (IGS3231)

Jump Together, Fly Farther!

인하대학교 국제학부

Week 2

ISE Department

Prof. Mehdi Pirahandeh

Content

인하대학교 INHA UNIVERSITY

- **Review the syllabus again!**
- **Introduction to Smart Mobility Engineering Lab**
- **Q & A**
- **VMWare Workstation Player Setup**
- **Configuring Ubuntu Virtual Machine**
- **ROS Installation**
- **Activity Session (OS & ROS 1 Test)**

Introduction

인하대학교 INHA UNIVERSITY

- This course consists of multiple guided tutorials and lab sessions with increasing difficulty levels when working with an autonomous robot and machine learning.

- You learn how to set up such a TurtleBot3 from scratch using ROS/ROS2, how to interface the individual sensors and actuators, and finally, how to implement the practical projects.

- This course provides students with hands-on lab experience applying Ai and robot programming techniques for implementing the final autonomous robot projects.

Course Goals

인하대학교 INHA UNIVERSITY

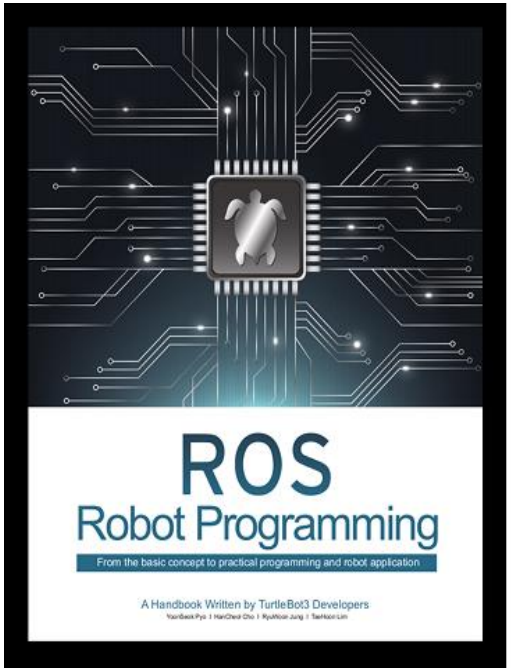
- By participating in this course attendees will learn how to:

  - Practice ROS/ROS2 python/C++ client library: Creating your own ROS/2 C++/Python programs
  - Practice the recent simulating tool with ROS: RVIZ, RQT, Gazebo simulator
  - Familiarize the student with the recent machine learning project with ROS/ROS2
  - Familiarize the student with the recent Autonomous Vehicle project with ROS/ROS2

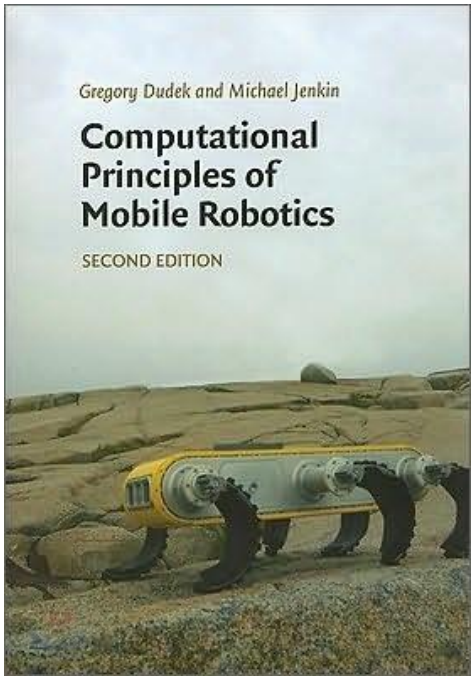
Course Oulines

- This course structure is outlined as follows,
  - I will notify the detailed class information in GitHub Repository, I-class system, and dedicated Kakao Group.
  - The student will present their projects and I will provide them with multiple feedback.
  - Blended Lecture and practice will be provided for students using a mix of offline classes, video content and online Zoom sessions.
  - There will be multiple tutoring sessions for helping students.

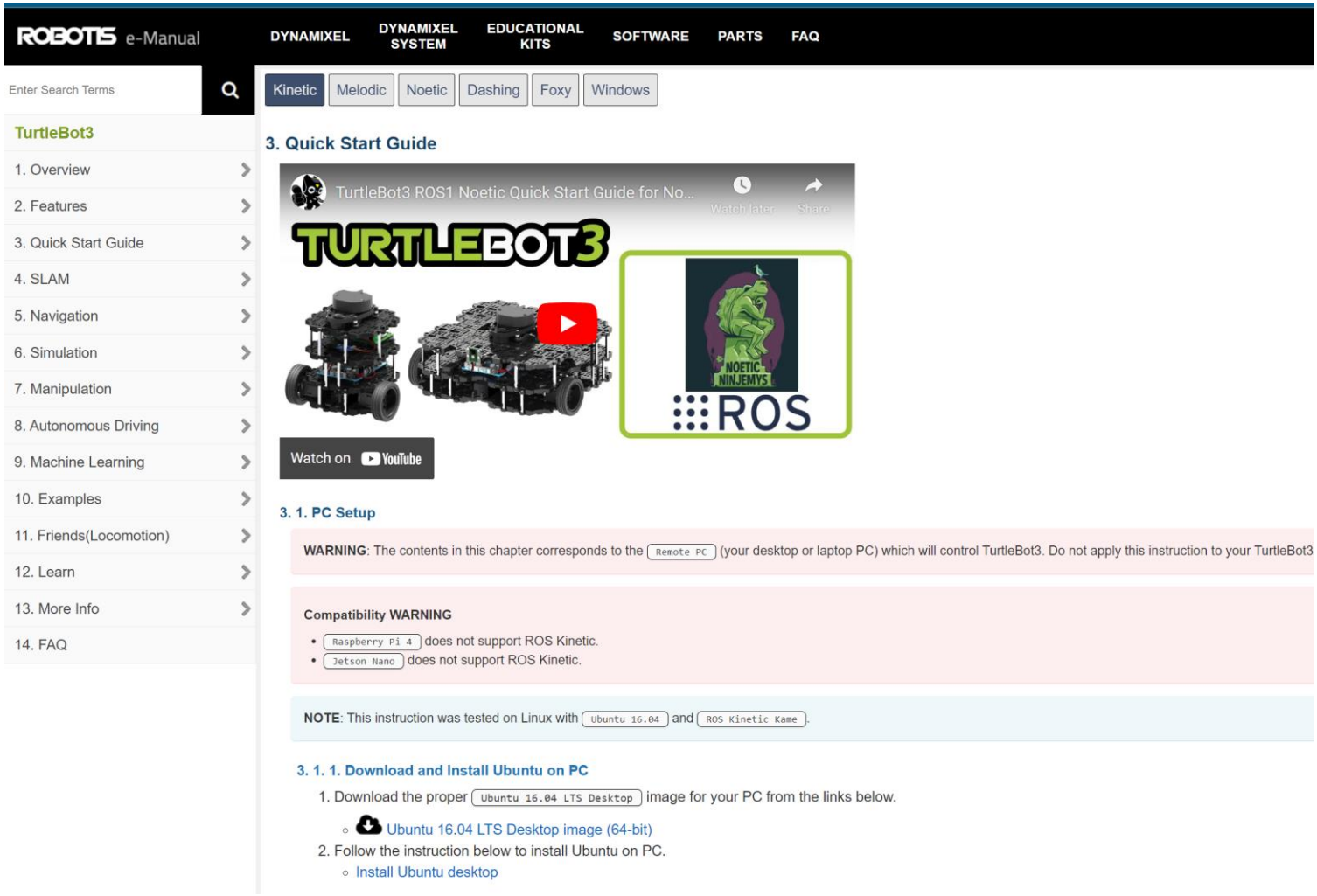
Books


ROS
Robot Programming
From the basic concept to practical programming and robot application
A Handbook Written by TurtleBot3 Developers


TURTLEBOT3
Burger


Gregory Dudek and Michael Jenkin
Computational Principles of Mobile Robotics
SECOND EDITION

Introduction to Course



https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start

Introduction to Course

인하대학교 INHA UNIVERSITY

ROS 2 Documentation: Foxy

Search docs

- Installation
- Distributions
- Tutorials
- How-to Guides
- Concepts
- Contact
- The ROS 2 Project
- Related Projects
- Glossary
- Citations

» ROS 2 Documentation Edit on GitHub

You're reading the documentation for an older, but still supported, version of ROS 2. For information on the latest version, please have a look at Humble.

ROS 2 Documentation

The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. From drivers and state-of-the-art algorithms to powerful developer tools, ROS has the open source tools you need for your next robotics project.

Since ROS was started in 2007, a lot has changed in the robotics and ROS community. The goal of the ROS 2 project is to adapt to these changes, leveraging what is great about ROS 1 and improving what isn't.

This site contains the documentation for ROS 2. If you are looking for ROS 1 documentation, check out the ROS wiki.

If you use ROS 2 in your work, please see Citations to cite ROS 2.

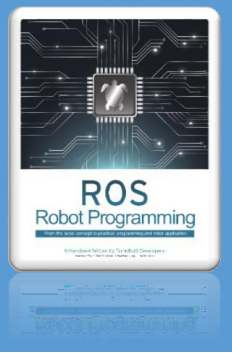
Getting started

- Installation
  - Instructions to set up ROS 2 for the first time
- Tutorials
  - The best place to start for new users!
  - Hands-on sample projects that help you build a progression of necessary skills
- How-to Guides

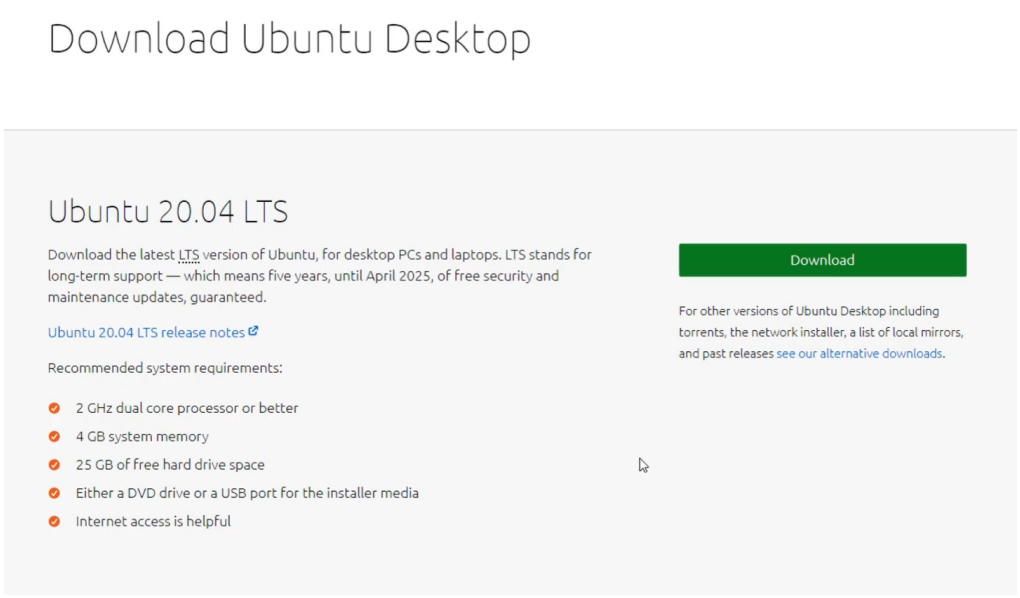
http://docs.ros.org/en/foxy/index.html

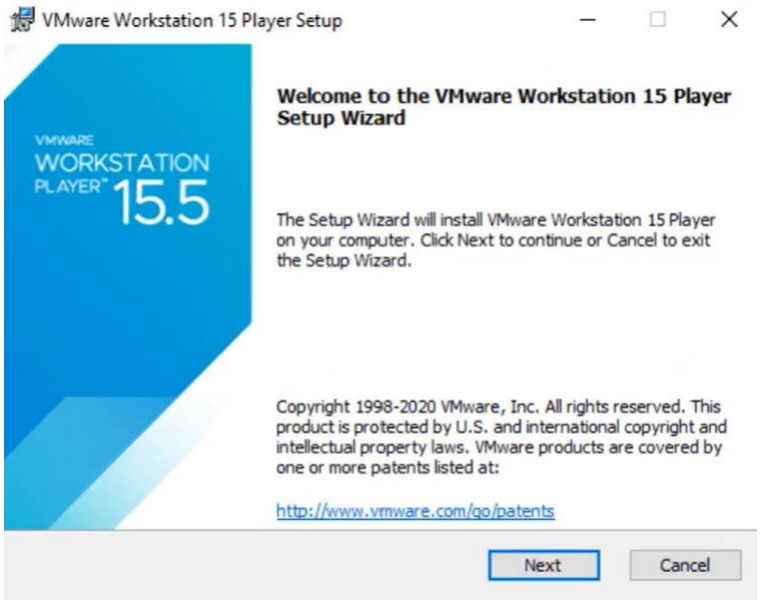
Robot Software Platform

ROBOTIS

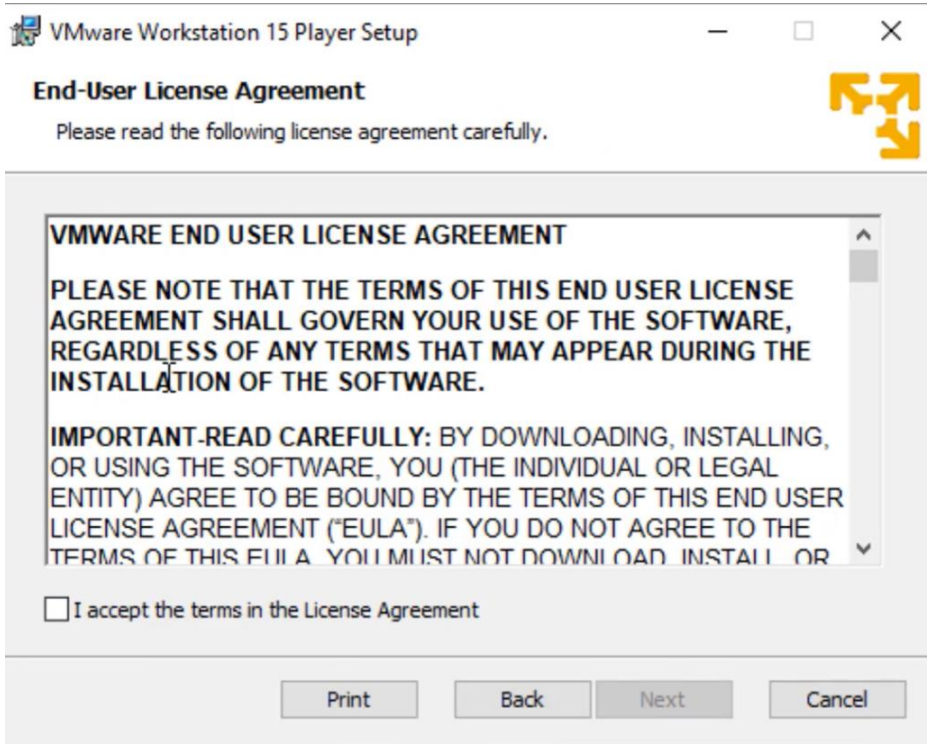


VMWare Workstation Player Setup

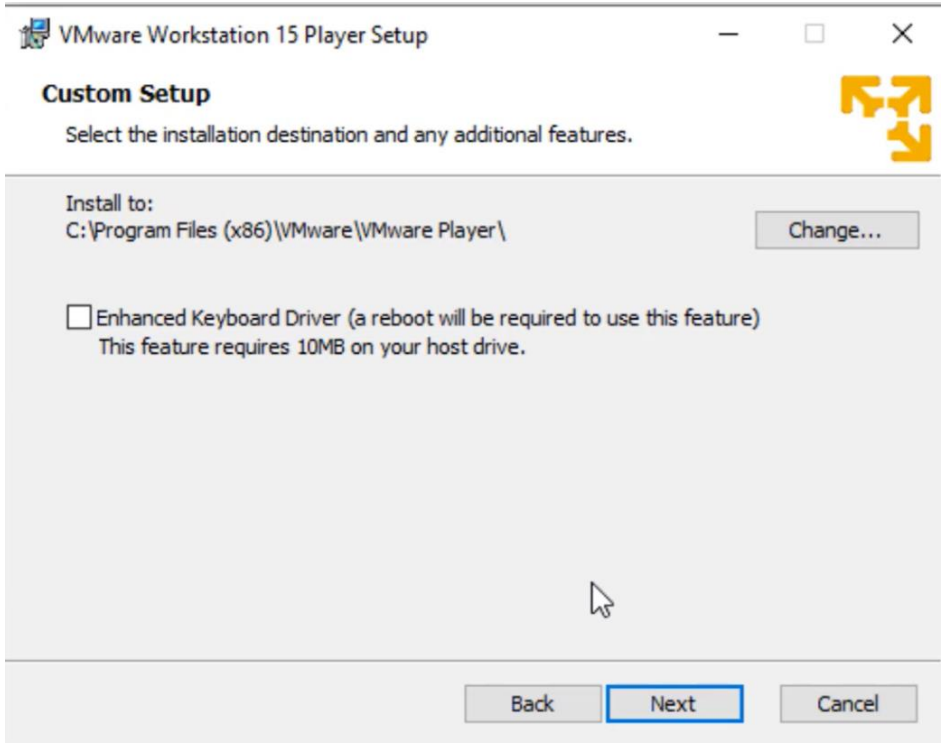


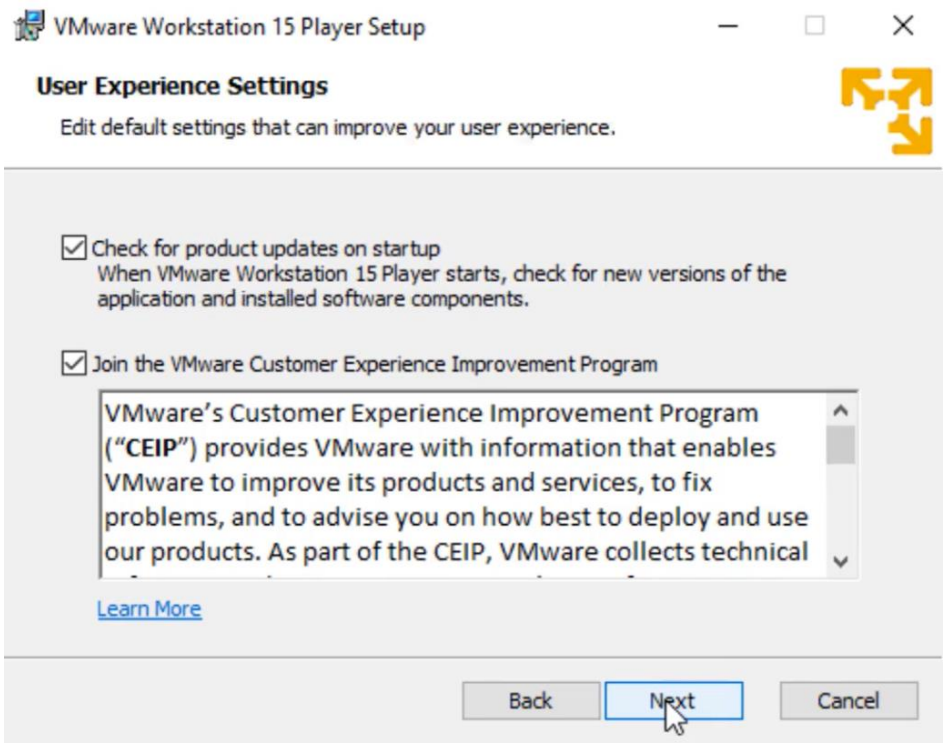


VMWare Workstation Player Setup

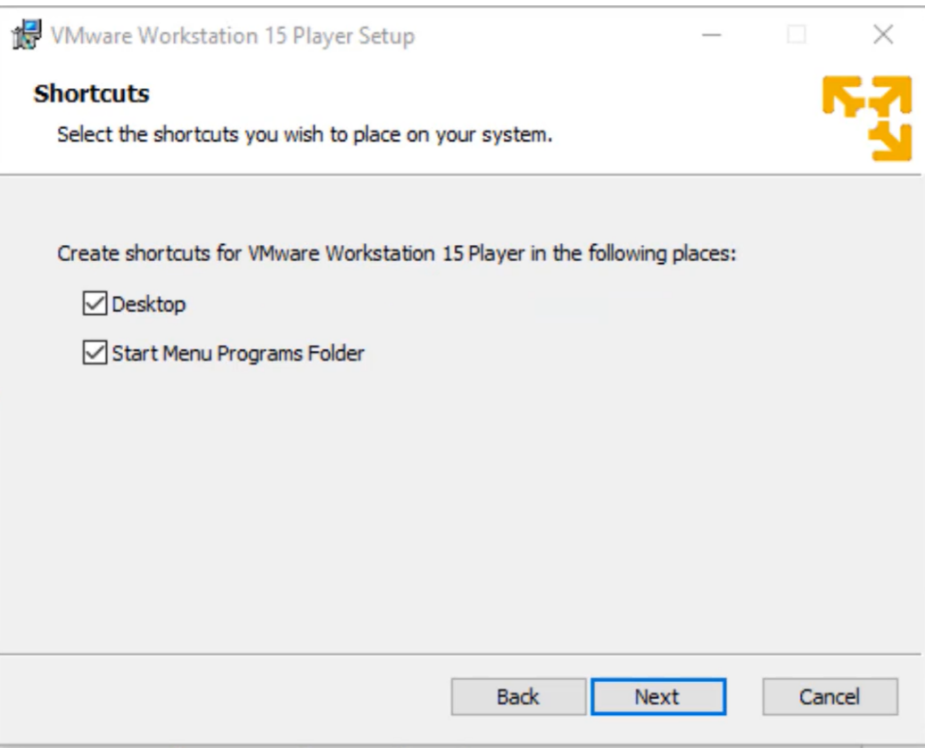


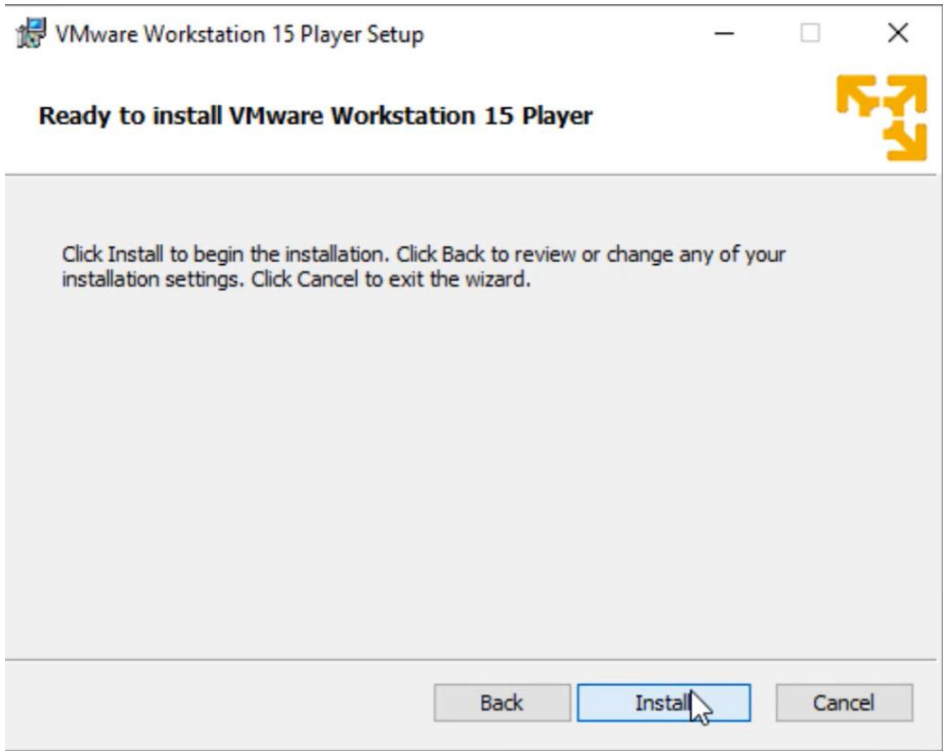
VMWare Workstation Player Setup



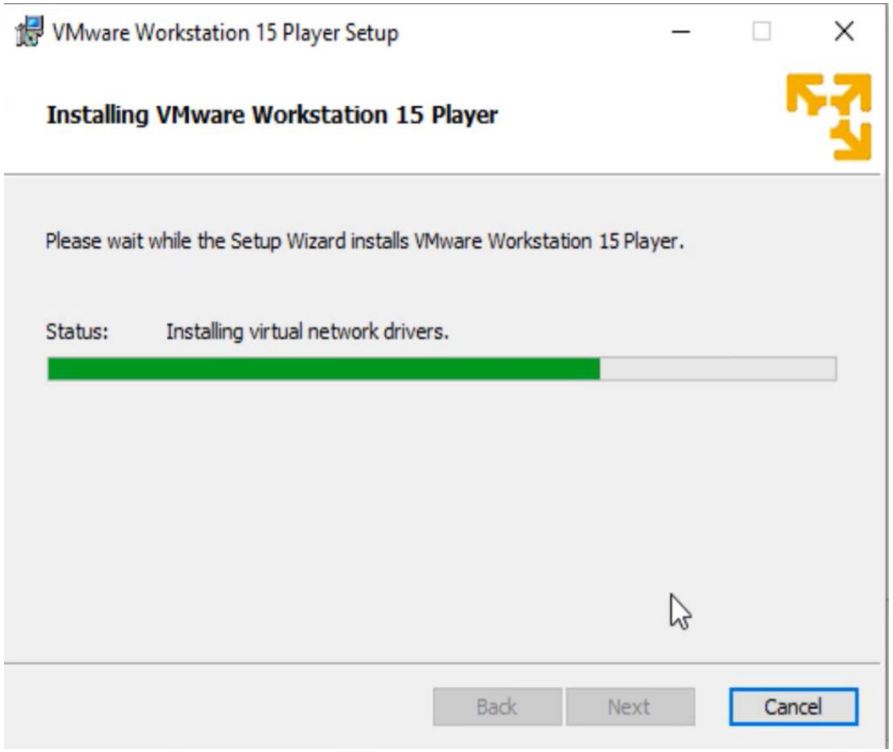


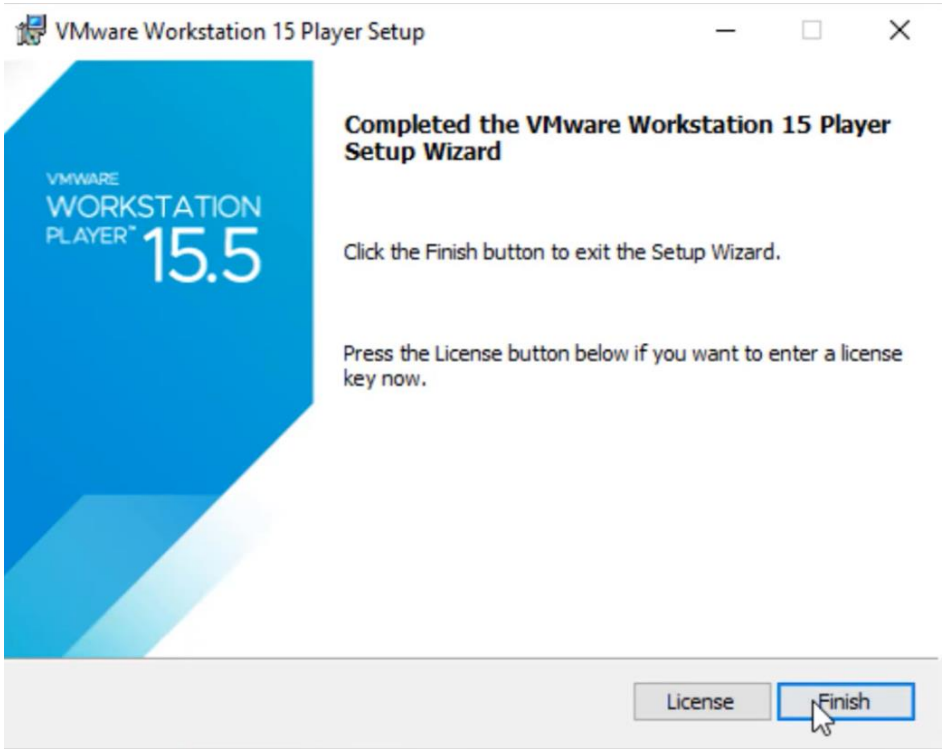
VMWare Workstation Player Setup





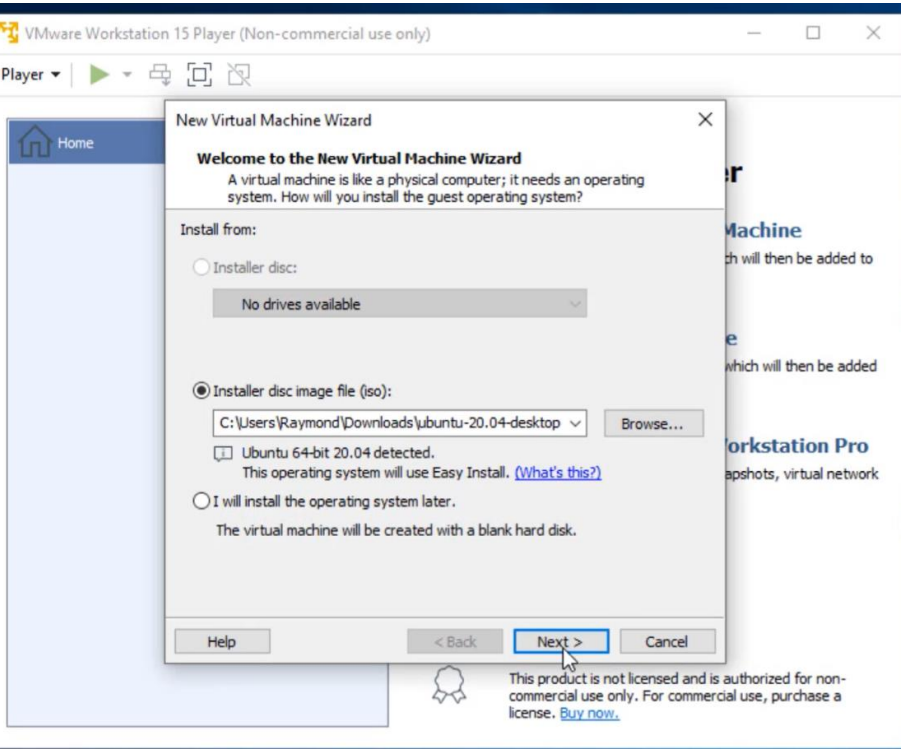
VMWare Workstation Player Setup

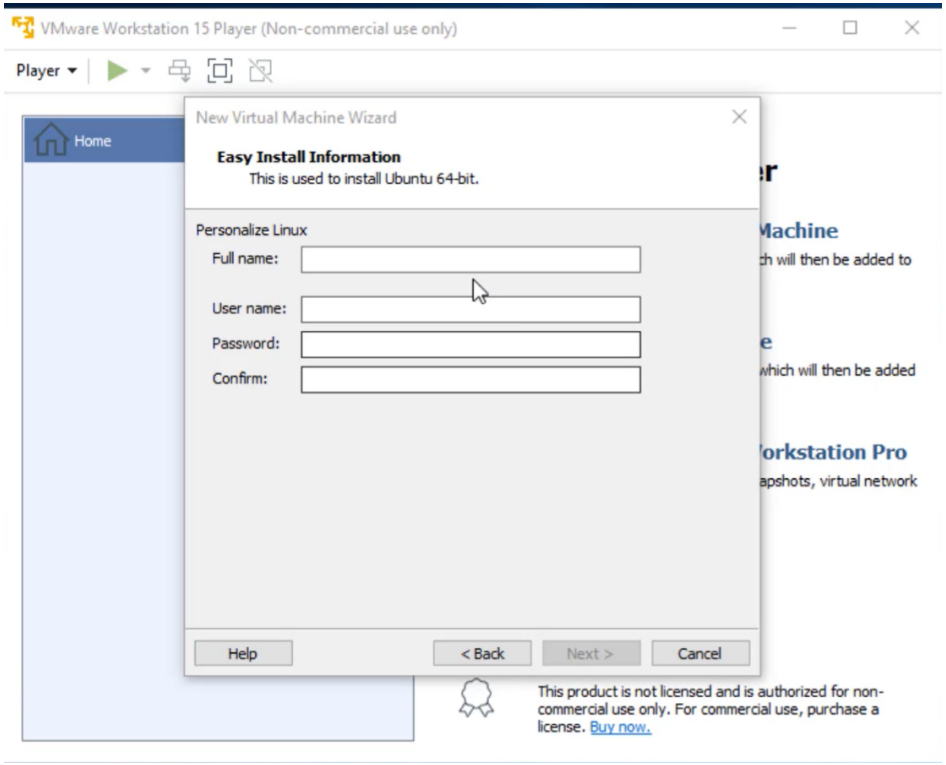




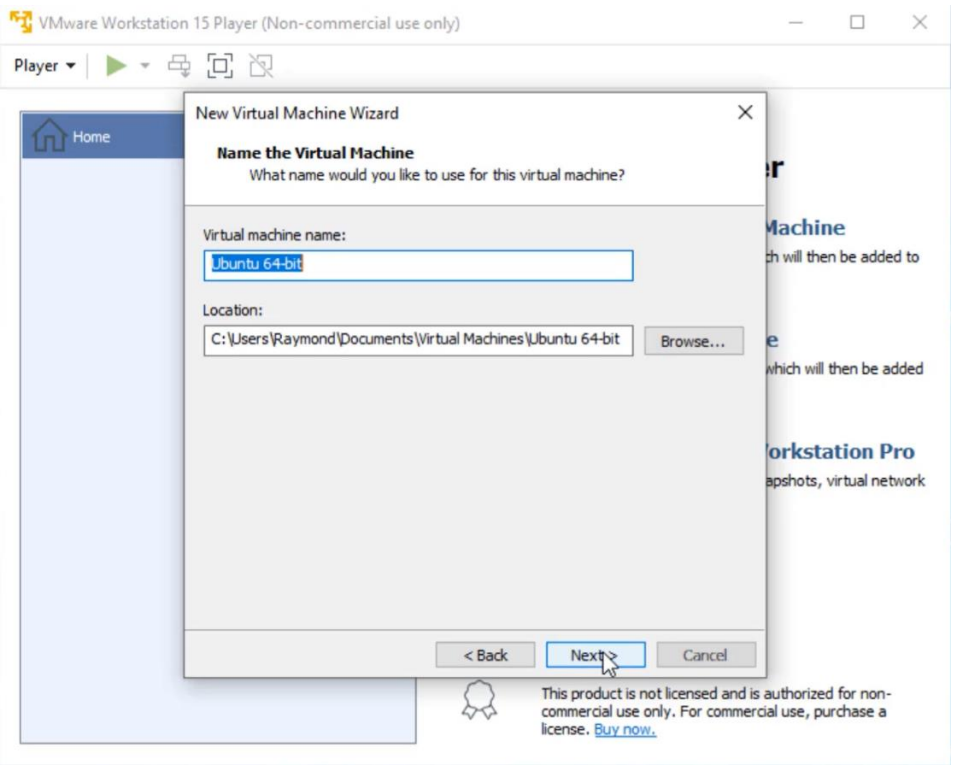
Configuring Ubuntu Virtual Machine

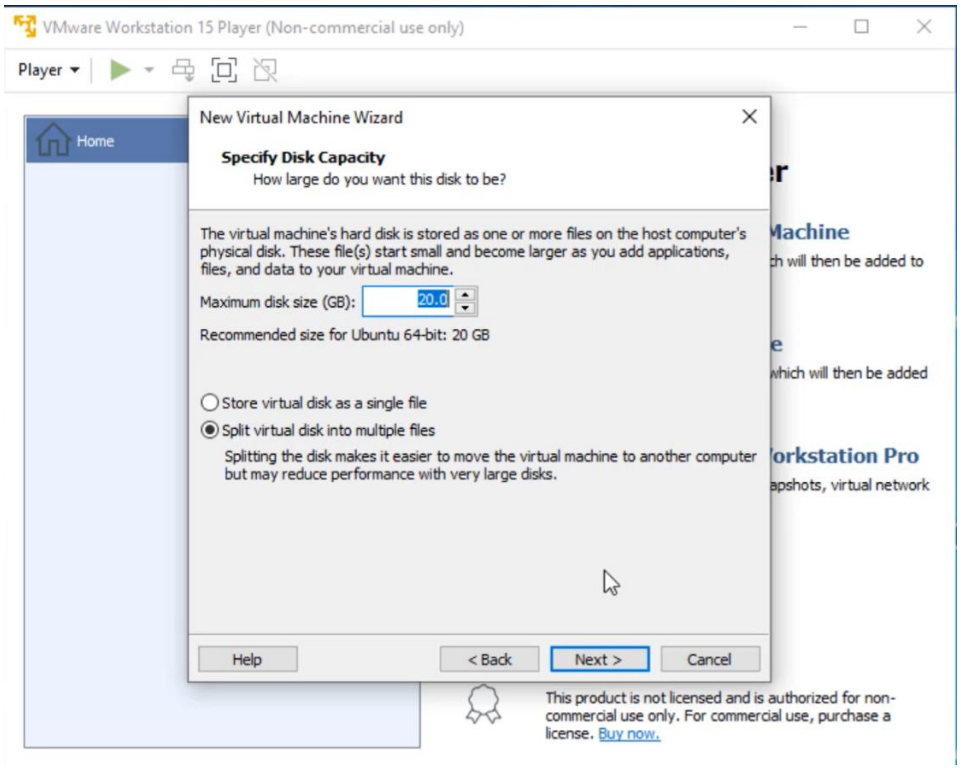
Configuring Ubuntu Virtual Machine



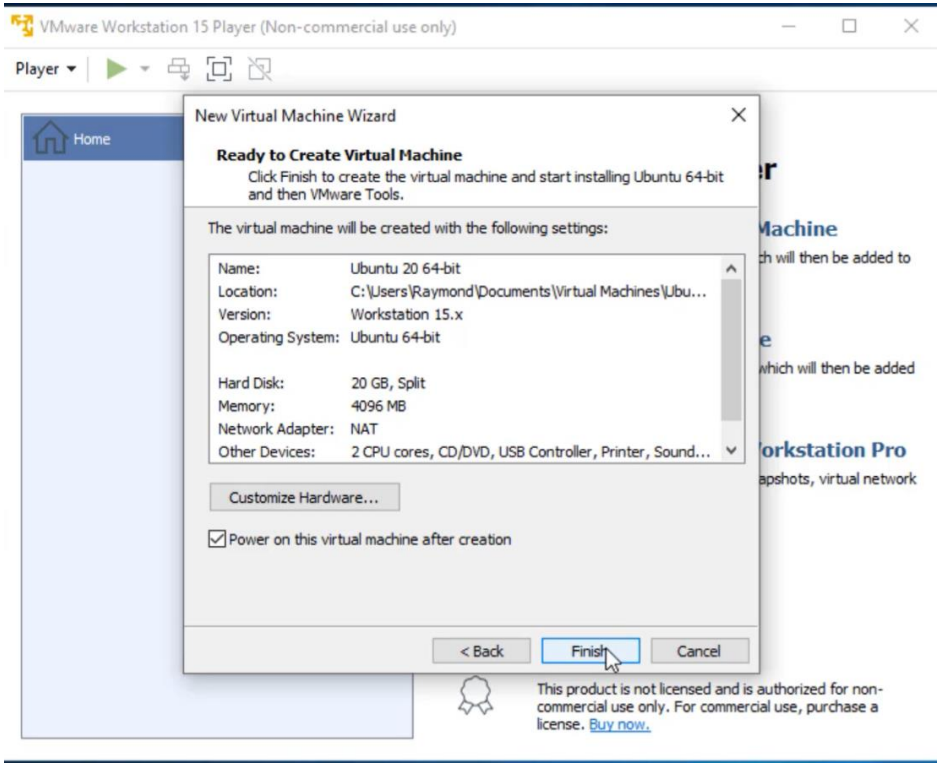


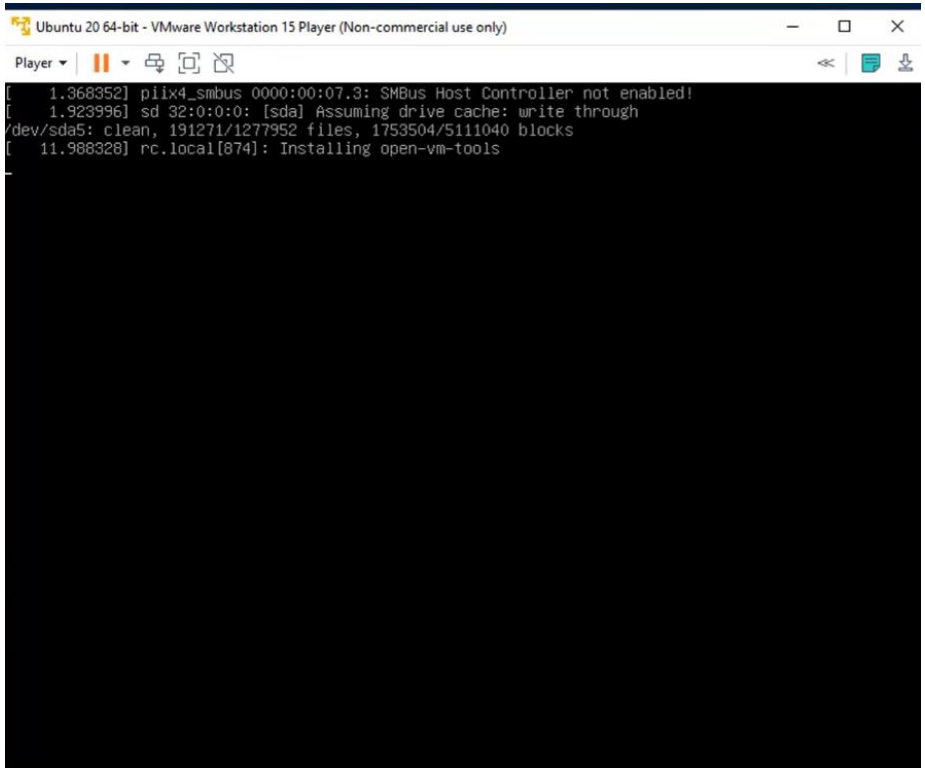
Configuring Ubuntu Virtual Machine



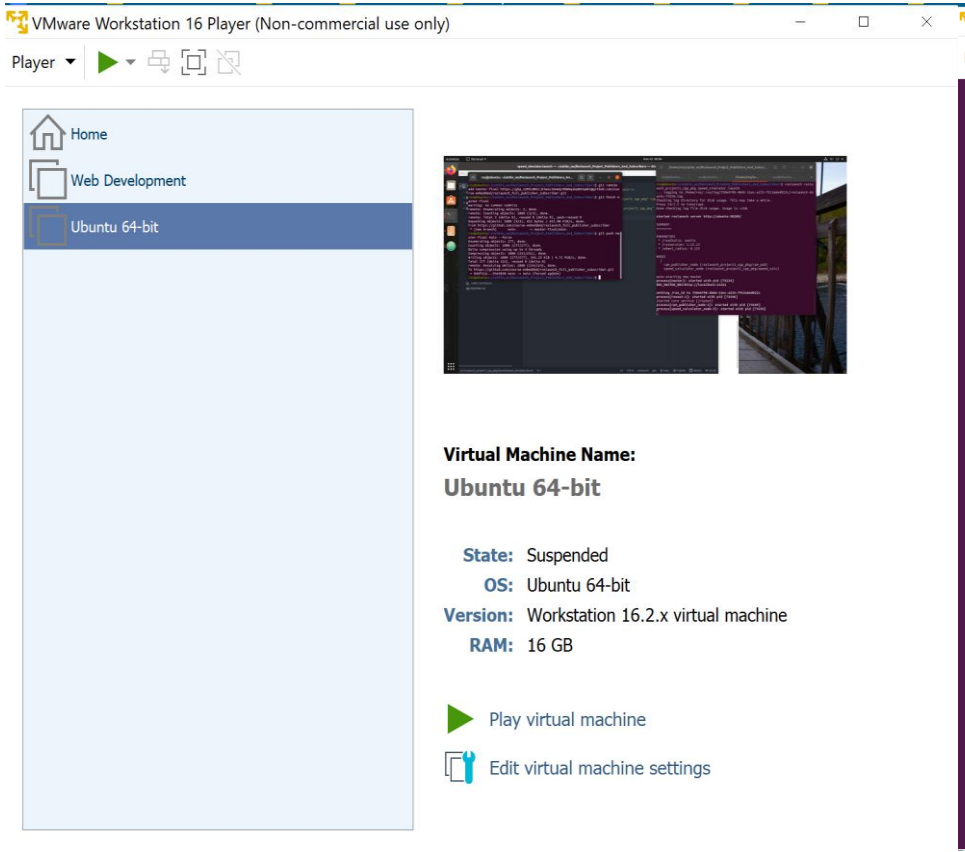


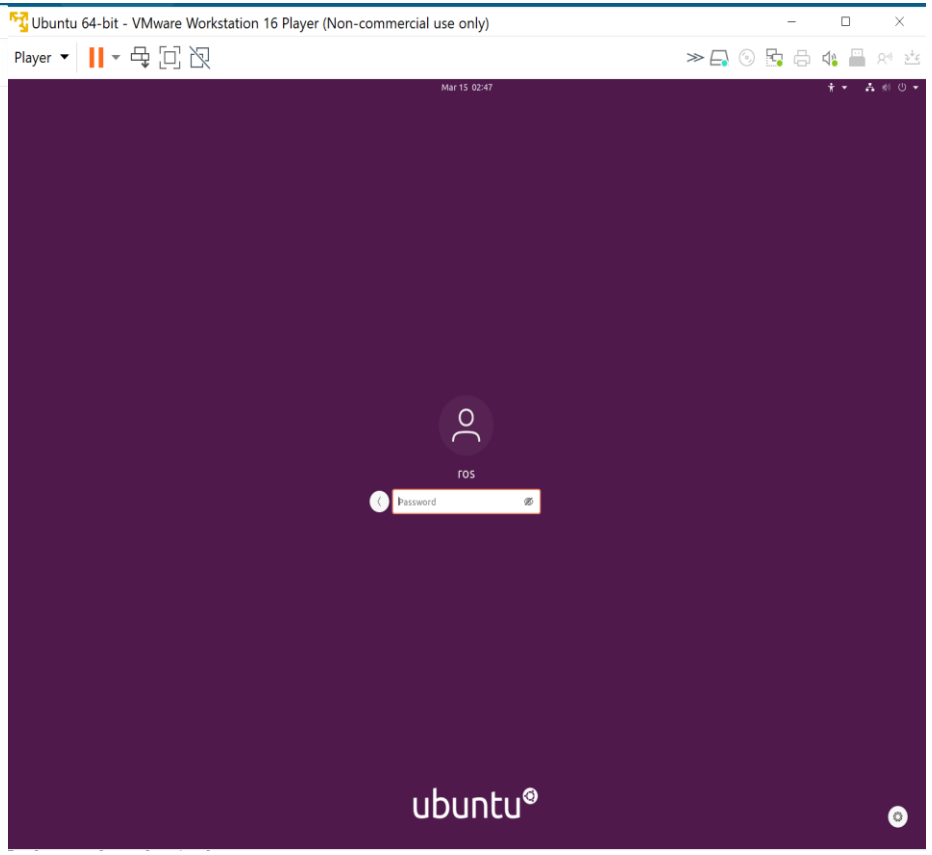
Configuring Ubuntu Virtual Machine



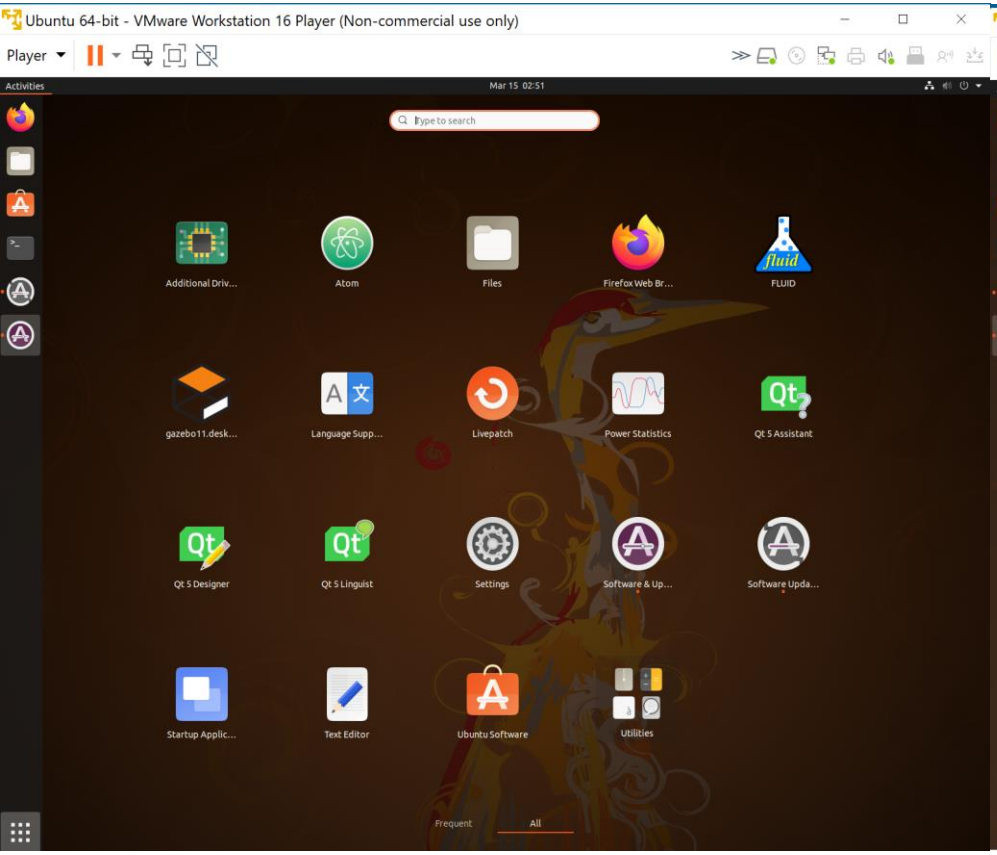


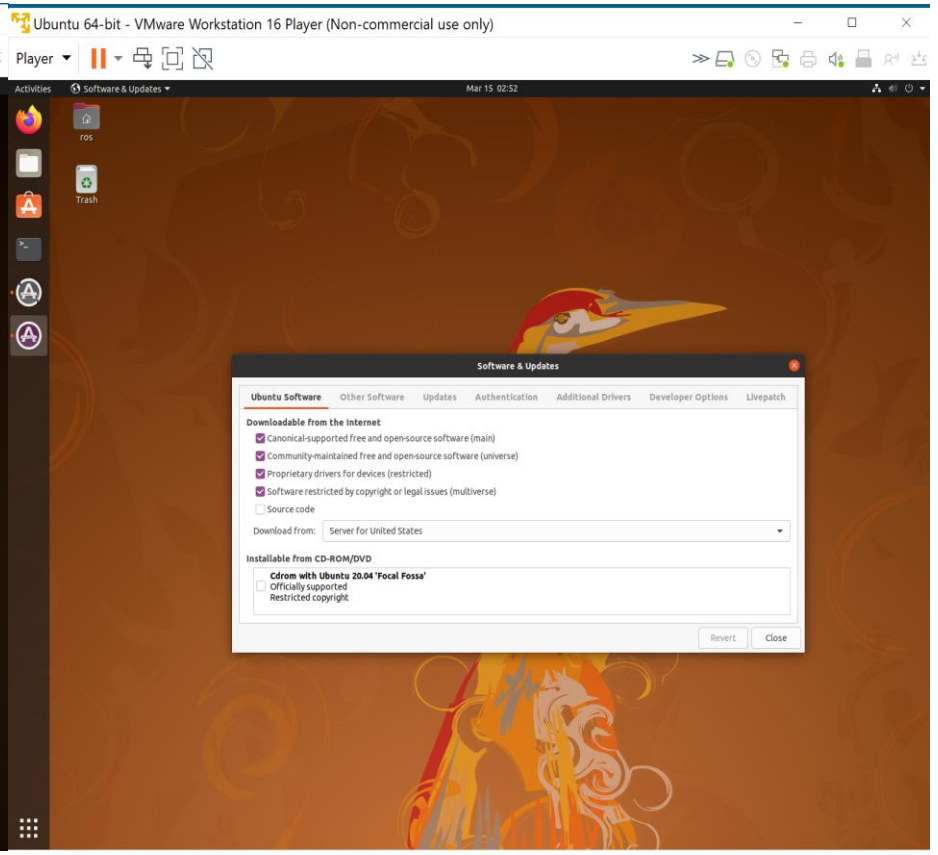
Configuring Ubuntu Virtual Machine





Configuring Ubuntu Virtual Machine





Activity Session

(OS & ROS 1 Test)

ROS Installation

ROS Installation

인하대학교 INHA UNIVERSITY

1. **Configure your Ubuntu repositories**

   Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse." You can follow the Ubuntu guide for instructions on doing this.

2. **Setup your sources.list**

   Setup your computer to accept software from packages.ros.org

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

3. **Set up your keys**

   Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse." You can follow the Ubuntu guide for instructions on doing this.

```
sudo apt install curl # if you haven't already installed curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

3. Installation

make sure your Debian package index is up-to-date:

```
sudo apt update -y
```

4. Desktop-Full Install: (Recommended) :

Everything in Desktop plus 2D/3D simulators and 2D/3D perception packages

```
sudo apt install ros-noetic-desktop-full -y
```

5. Ubuntu Environment Setup

You must source this script in every bash terminal you use ROS in.

```
source /opt/ros/noetic/setup.bash
```

It can be convenient to automatically source this script every time a new shell is launched. These commands will do that for you.

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

6. Testing ROS Installation

roscore is a collection of nodes and programs that are pre-requisites of a ROS -based system. You must have a *roscore* running in order for ROS nodes to communicate. It is launched using the *roscore* command.

```
roscore
```

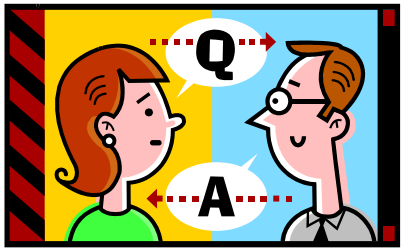
rostopic contains the *rostopic* command-line tool for displaying debug information about ROS Topics, including publishers, subscribers, publishing rate, and ROS Messages. It also contains an experimental Python library for getting information about and interacting with topics dynamically. This library is for internal-use only as the code API may change, though it does provide examples of how to implement dynamic subscription and publication behaviors in ROS.

```
rostopic list
```

Brief break (if on schedule)

인하대학교 INHA UNIVERSITY

Q&A?



Prof. Mehdi Pirahandeh
E-mail : mehdi at inha dot ac dot kr