
# Lab 0 - Setting up Your Virtual Machine

| | |
|---|---|
| Type | Lab |
| Status | Ready |
| Created By | Tiffany Cappellari |

**Note**: You must complete this lab **before** the start of Lab 1 as you will be unable to begin work on Lab 1 if your VM is not yet properly set up. Lab 0 should only take about an hour of your time.

### Relevant Tutorials And Documentation

- ROS Wiki: http://wiki.ros.org/
- Baxter SDK: https://github.com/RethinkRobotics/sdk-docs/wiki/API-Reference
- Sawyer SDK: http://sdk.rethinkrobotics.com/intera/API_Reference
- Gazebo and ROS: http://gazebosim.org/tutorials?tut=ros_overview

## Table of Contents

Relevant Tutorials And Documentation

Table of Contents

Introduction

Download Your Virtual Machine

Installing Software and Packages

- Installing ROS
- Installing Gazebo
- Installing MoveIt!
- Installing Sawyer SDK and Simulation Packages
- Installing Baxter SDK and Simulation Packages

Common Issues and Solutions

# Introduction

For all remote labs this semester we will be using a virtual machine (VM) because Robotic Operating System (ROS) only works on Linux machines. A *Virtual Machine* allows you to take a small chunk of your computer's processing power and memory to emulate a separate machine, isolated from the rest of the processes on your computer. This new *virtual* machine can then be running any operating system and processes you want, without interacting with the rest of your computer. Here, we will use it to run an instance of Ubuntu 16.04 so that you can use ROS even if you don't have a Linux computer.

For this class we will be using VMware to create our VM. VMware takes "images" of a virtual machine and creates an instance of the machine for you to use. When you create a virtual machine, you can stop and start the same VM with the same files still saved on it; however, the files you create and save on it will not be shared across multiple instances of the VM. Please be aware that if you delete a VM, you also delete any files you have saved on it. **We highly recommend you use some sort of version control such as Github just in case you accidentally delete your VM (just make sure your repo is private!**).

# Download Your Virtual Machine

1. Go to https://releases.ubuntu.com/16.04/ and download the 64-bit PC (AMD64) desktop image. You should now have a .iso file.
2. Go to https://software.berkeley.edu/vmware and acquire a suitable version of VMWare using the Berkeley institutional license. For MacOS you can use VMWare Fusion 11.

3. Install VMWare, and launch a new virtual machine from the .iso file you downloaded in step 1. You can use the "easy install" set up with the default settings. Boot the virtual machine and follow the set up instructions. Create a user account.
4. Open up a terminal window in your VM (right click anywhere → Open Terminal) and in it run
   `echo "export SVGA_VGPU10=0" >> ~/.profile`
5. Now shut down the virtual machine (shut down, not pause. Please be sure you are aware of the difference). Next we want to allocate an appropriate amount of RAM to the VM. With the machine shut down, from the menu go to Virtual Machine → Settings → Processors & Memory. Use the slider to allocate at least 5GB of RAM. On a 16GB machine, you should be able to easily allocate over 8GB, but 5GB should be sufficient for our purposes, and you should be able to allocate that much RAM even on 8GB systems. You will not be able to edit this if your VM is not shut down.

# Installing Software and Packages

Now we can begin installing the necessary packages for our labs. Make sure to first start up your VM and be logged in. Now you can open up a terminal window.

## Installing ROS

In this class, we will be using Robotic Operating System (ROS), version Kinetic. You will learn more about ROS in the next lab.

First, set up 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'
```

Next, let's set up your keys

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

Now we can begin the ROS installation

```
sudo apt-get update sudo apt-get install ros-kinetic-desktop-full
```

Make sure your environment variables get correctly setup every time a new shell is launched by adding to your `.bashrc` file. Run the following commands

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

Finally, add dependencies for building packages

```
sudo apt install python-rosdep python-rosinstall python-rosinstall-generator
python-wstool build-essential sudo rosdep init rosdep update
```

## Installing Gazebo

Gazebo is the simulation environment we will be using in place of our normal robots this semester. In order to install it, run the following lines:

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable
`lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list' wget
http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add - sudo
apt-get update sudo apt-get install gazebo7
```

**Note for Gazebo troubleshooting**: If you get the error that process `[gazebo_gui-3]` has died with `exit code 139` whenever you try to relaunch gazebo after the first time, run `killall gzserver` before relaunching Gazebo.

## Installing MoveIt!

MoveIt! is one of the most popular packages for manipulation (and mobile manipulation) in ROS. You will learn more about it and how to use it later in the semester. In order to install it please run

```
sudo apt-get install ros-kinetic-moveit
```

## Installing Sawyer SDK and Simulation Packages

For some of the labs this semester you will be using a simulation of a Sawyer robot. You can learn more about Sawyers at http://www.rethinkrobotics.com/sawyer/.

First we need to set up a ROS workspace for the packages to live. This will also hold all of our Baxter packages (more on that in the next section):

```
mkdir -p ~/rethink_ws/src cd ~/rethink_ws/src catkin_init_workspace cd
~/rethink_ws catkin_make echo "source ~/rethink_ws/devel/setup.bash" >>
~/.bashrc
```

Sawyer packages and dependencies are labelled `Intera`. Let's install the Intera dependencies

```
sudo apt-get update sudo apt-get install git-core python-argparse python-
wstool python-vcstools python-rosdep ros-kinetic-control-msgs ros-kinetic-
joystick-drivers ros-kinetic-xacro ros-kinetic-tf2-ros ros-kinetic-rviz ros-
kinetic-cv-bridge ros-kinetic-actionlib ros-kinetic-actionlib-msgs ros-
kinetic-dynamic-reconfigure ros-kinetic-trajectory-msgs ros-kinetic-rospy-
message-converter sudo apt-get install libignition-math2-dev
```

and install the Intera robot SDK before finally building the workspace with `catkin_make`.

```
cd ~/rethink_ws/src wstool init . git clone
https://github.com/RethinkRobotics/sawyer_robot.git wstool merge
sawyer_robot/sawyer_robot.rosinstall wstool update source ~/.bashrc cd
~/rethink_ws catkin_make
```

We now also want to create a symlink between the `intera.sh` script and the top of the workspace.

```
ln -s ~/rethink_ws/src/intera_sdk/intera.sh ~/rethink_ws/intera.sh
```

Finally, open up `intera.sh` with the text editor of your choice (we recommend vim or Sublime Text but you will need to install it in the VM yourself) and find the variable `ros_version` and set it to `kinetic` (the default value should be `indigo`).

Now let's install the simulation packages needed for Sawyer as well.

```
sudo apt-get install ros-kinetic-qt-build ros-kinetic-gazebo-ros-control ros-kinetic-gazebo-ros-pkgs ros-kinetic-ros-control ros-kinetic-control-toolbox ros-kinetic-realtime-tools ros-kinetic-ros-controllers ros-kinetic-xacro python-wstool ros-kinetic-tf-conversions ros-kinetic-kdl-parser ros-kinetic-sns-ik-lib cd ~/rethink_ws/src git clone https://github.com/RethinkRobotics/sawyer_simulator.git wstool merge sawyer_simulator/sawyer_simulator.rosinstall wstool update cd ~/rethink_ws catkin_make
```

Check that Sawyer simulation works. Use `./intera.sh sim` whenever you want to run anything "on the robot".

```
./intera.sh sim cc
```

Gazebo should launch without any errors and a simulation of a Sawyer robot should appear on screen.

## Installing Baxter SDK and Simulation Packages

Baxter robots are another type of robot we will be using in our labs. Unlike Sawyer, Baxter has two arms. You can learn more about Baxter robots at https://en.wikipedia.org/wiki/Baxter_(robot). In order to use Baxter simulations, we will need to install packages similar to Sawyer.

First, install Baxter's SDK packages:

```
cd ~/rethink_ws/src wstool merge https://raw.githubusercontent.com/RethinkRobotics/baxter/master/baxter_sdk.rosi
wstool update cd ~/rethink_ws catkin_make
```

Now we will symlink it the same way we did for the Sawyer packages:

```
ln -s ~/rethink_ws/src/baxter/baxter.sh ~/rethink_ws/baxter.sh
```

and again open up `baxter.sh` with your preferred text editor and find and change the variable `ros_version` to `kinetic`.

Now we can install the Baxter simulation packages:

```
cd ~/rethink_ws/src wstool merge
https://raw.githubusercontent.com/RethinkRobotics/baxter_simulator/kinetic-
devel/baxter_simulator.rosinstall wstool update cd ~/rethink_ws catkin_make
```

Check that Baxter simulation works. Use `./baxter.sh sim` whenever you want to run anything "on the robot".

```
./baxter.sh sim roslaunch baxter_gazebo baxter_world.launch
```

Gazebo should launch without any errors and a simulation of a Baxter robot should appear on screen.

# Common Issues and Solutions

- If you encounter the error:
  `VMware Cannot Connect the Virtual Device sata0:1`
  Go to your VM settings, then in Hardware tab choose CD/DVD, then in Device Status, uncheck the option ''Connect at power on''.
- If you encounter the error:
  `This file can't be downloaded securely`
  when downloading the VMware software go to download details and press "keep anyway." This will allow for the download to be completed.
- (For Mac computers) When booting your virtual machine for the first time if you get the error
  `Could not open /dev/vmmon: Broken pipe.`
  on your computer (not in your VM) go to Systems Preferences → Security & Privacy → General. Under the allow apps downloaded from section press allow for VMware. Then restart VMware.
- If Gazebo is repeatedly crashing and shutting down, try shutting down the VM and with the VM still shut down, go to Virtual Machine → Settings → Display and uncheck ''Accelerate 3D Graphics''. This may cause Gazebo to have more lag but it should also stop it from crashing.
- If you cannot connect to the internet, click the wifi icon at the top right and make sure that ''Enable Networking'' is checked. It if it, try unchecking and rechecking it. When you are connected, the VM should recognize your connection as an ethernet connection and the wifi icon should be replaced with two arrows pointing in opposite directions (one up, one down). If it still will not connect, try restarting your VM.