

Depth Sensor Setup

Hyperlinks contain associated downloads

Creating VM in virtual box

Install the latest version of [Oracle VM Virtual Box](#) if you haven't already.

Download [Ubuntu Server 22.04.5 LTS ISO](#)

In virtual box, select Machine -> New

On the first screen of the new machine setup, give your VM any name and select the ISO image that you just downloaded.

Make sure that you check 'Skip unattended installation'

On the third screen, give the VM around 8gb of RAM and give it 2 CPUS.

On the fourth screen, create a hard disk that is 75gb and check "Pre-allocate Full Size".

Now, just click finish on the final page. The VM will automatically open once VirtualBox is finished building it

VM Setup

Once the VM opens, it will take a couple of seconds to load then present you with language options. Choose English with the Enter key

On the second screen, select **Continue without updating** this is very important!

On the keyboard config screen, don't change anything and just select Done.

On the type of installation screen, also don't change anything and select Done.

On the network config screen, also don't change anything and select Done.

On the proxy config screen, also don't change anything and select Done.

On the Ubuntu archive mirror configuration screen, wait for the mirror location tests to pass then select done

On the guided storage screen, don't change anything and select Done. You can move down the list with tab.

On the storage config screen, don't change anything and select Done.

On the "Confirm Destructive action" popup, select continue.

On the profile config screen, set a name and password that you can remember. I'm not sure if this matters, but I used the same username and password that I set when doing virtual box's new machine setup.

On the ubuntu Pro screen, don't change anything and continue.

In ssh config select **Install OpenSSH server** then Done.

On the Featured server snaps screen, don't select any apps and click done.

You will now be installing the VM, select reboot now once it is finished.

First boot

The first thing that you will need to do once the machine boots is login.

SSH into machine (optional but highly encouraged)

This section will teach you how to ssh into the VM that you just created. This will allow you to access your machine through your windows terminal, enabling you to copy and paste commands.

To start, in your VM window go to Devices -> Network -> Network Settings.

In the Network Settings window change "Attached to:" from NAT to Bridged Adapter, then click ok.

Now, go back to that Network Settings screen.

Click on "Advanced", and toggle the "Cable Connected" option off then back on, then click ok.

Go back to your VM terminal and type the command **ip a** to get your ip address.

```
matt@pass:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
    link/ether 08:00:27:9b:a1:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.22/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 43057sec preferred_lft 43057sec
    inet6 2603:6000:b3f0:1c10::10b5/128 scope global dynamic noprefixroute
        valid_lft 86255sec preferred_lft 86255sec
    inet6 fd00:87b:125d:54f9::10b5/128 scope global dynamic noprefixroute
        valid_lft 86255sec preferred_lft 86255sec
    inet6 2603:6000:b3f0:1c10:a00:27ff:fe9b:a1fa/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86310sec preferred_lft 86310sec
    inet6 fd00:87b:125d:54f9:a00:27ff:fe9b:a1fa/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86310sec preferred_lft 86310sec
    inet6 fe80::a00:27ff:fe9b:a1fa/64 scope link
        valid_lft forever preferred_lft forever
```

Now open a Windows command line terminal.

The command that you enter to SSH will be slightly different than mine. The format of the command is:
ssh username@your_ip.

For example, ssh command for this demo is **ssh matt @192.168.1.22**

Next, the terminal will ask "Are you sure you want to continue connecting". Type **yes** and hit enter

It will then ask you for your VM's login password. After entering that you should be able to control your VM through windows! During the rest of this document, you can copy and paste any commands you need to control the VM into this windows terminal.

^ END OF SSH SECTION

Run the commands:

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

To clone the Git repo run:

```
git clone https://github.com/aaron97neu/Discovery-World-Pong.git
```

We are now going to run the post install script. To do this run the commands:

```
cd Discovery-World-Pong/setup/
```

```
sudo ./post-install.sh
```

Now we need to shutoff the machine. To do this run the command:

```
sudo shutdown now
```

Second Boot

You may notice that you have a GUI now. That is completely normal. Go ahead and login. **CLICK DON'T UPGRADE ON ANY UPDATE POPUPS**

You can now SSH into the VM again since shutting it down would have closed the connection. We are going to build the docker containers. To do this run the commands:

```
cd Discovery-World-Pong/
```

```
sudo docker compose build
```

Once those build, we can install the intel realsense camera SDK into our VM. Run these commands to do so:

```
cd ~
```

```
sudo apt-get install curl sudo lsb-base lsb-release -y
```

```
sudo apt-get install ffmpeg libsm6 libxext6 -y
```

```
sudo apt install -f
```

```
sudo mkdir -p /etc/apt/keyrings
```

```
sudo curl -sSf https://librealsense.intel.com/Debian/librealsense.pgp | sudo tee /etc/apt/keyrings/librealsense.pgp > /dev/null
```

```
sudo apt-get install apt-transport-https -y
```

```
sudo echo "deb [signed-by=/etc/apt/keyrings/librealsense.gpg] https://librealsense.intel.com/Debian/apt-repo `lsb_release -cs`  
main" | sudo tee /etc/apt/sources.list.d/librealsense.list
```

sudo apt-get update

**sudo apt-get install -y librealsense2-dkms **

**librealsense2-utils **

**librealsense2-dev **

librealsense2-dbg

realsense-viewer

Now lets install a web browser to view our docker images once we compose them:

curl -fsS https://dl.brave.com/install.sh | sh

We now have all the components that we need to build and run the containers. We just need to shutdown one last time to pass through the USB device to the VM.

After shutdown

After shutting down, there should be no active VM windows running.

In the Virtual Box main menu, click on our VM and then click settings.

Click on the “USB” tab.

Now switch the USB controller to “USB 3.0 (xHCI) controller” and click ok.

Next, in your VM window, go to Devices -> USB -> and check the Intel RealSense Camera.

Now restart the VM

Third Boot

To check that the camera was passed through to the VM, we can add run this command:

realsense-viewer

If the GUI says “Intel realsense device detected” on the startup screen then the camera was passed through properly.

Let’s compose up the docker images now that the camera is set to be pass through to them. To do this run the commands:

cd Discovery-World-Pong/

sudo docker compose up -d

