**Step 1 and 2 is to install Ubuntu and ROS which is separately documented in this repo.**

**Step 3: WiFi setup using netplan**

Starting from Ubuntu 18.04 LTS, Ubuntu uses [Netplan](https://netplan.io/)to configure network interfaces by default. Netplan is a utility for configuring network interfaces on Linux. Netplan uses YAML files for configuring network interfaces. YAML configuration file format is really simple. It has clear and easy to understand syntax.

To be able to setup Wifi on Raspberry Pi, you first need to get the name of the wifi card by showing physical components using the following command:

sudo lshw

In my case it was **wlan0.** Then navigate to /etc/netplan/ using the cd command

cd /etc/netplan/

Edit the Netplan YAML configuration file /etc/netplan/50-cloud-init.yaml with the following command:

sudo nano 50-cloud-init.yaml

Add your WiFi access information. Make sure not to use tab for space, use the spacebar to create the blank.

# This file is generated from information provided by  
# the datasource. Changes to it will not persist across an instance.  
# To disable cloud-init's network configuration capabilities, write a file  
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:  
# network: {config: disabled}  
network:  
 version: 2  
 ethernets:  
 eth0:  
 optional: true  
 dhcp4: true  
 # add wifi setup information here ...  
 wifis:  
 wlan0:  
 optional: true  
 access-points:  
 "YOUR-SSID-NAME":  
 password: "YOUR-NETWORK-PASSWORD"  
 dhcp4: true

Change the **SSID-NAME** and the **YOUR-NETWORK-PASSWORD** with your information. Close and save the file using ctrl+x and press yes.

Now, check whether there’s any error in the configuration file with the following command:

sudo netplan –debug try

If any error encounters then you can check with this command for detailed error information.

sudo netplan --debug generate

Apply the configuration file with the following command:

sudo netplan --debug apply

Finally, reboot your PI

sudo reboot

**Step 4: Updating and upgrading software packages on your Pi**

To make sure all dependencies are up to date, run the following command

sudo apt-get update

If you want to get the latest versions of software you installed already, run

sudo apt-get upgrade

This command upgrades all the software on your Pi to the latest version. It can take a while to run, so you don’t need to do it often. You have to press Y and Enter to confirm.

**Step 5: Enabling SSH**

Get the current status of the ssh server.

sudo systemctl ssh status

If you see errors, then the ssh rsa key is invalid and you would have to regenerate the keys.

sudo ssh-keygen -A

then run:

sudo systemctl start ssh

Check the status again.

If ssh is active on your raspberry, you can log on to the target device from your computer via a terminal.

**Step 6: SSH into your Raspberry Pi**

Now that you have enabled SSH and found out your IP address you can go ahead and SSH into your Raspberry Pi from any other computer. You’ll also need the username and the password for the Raspberry Pi.

Default Username and Password is:

* username: ubuntu
* password: your\_password

Open a terminal (on Mac and Linux) on the computer from which you want to SSH into your Pi and type the command below. On Windows, you can use a SSH client like [Putty.](https://itsfoss.com/putty-linux/)

Linux and macOS user have an [SSH](https://linuxize.com/post/ssh-command-in-linux/)client installed by default, and can SSH into the Pi by typing:

ssh ubuntu@pi\_ip\_address

When you connect through SSH for the first time, you will be prompted to accept the RSA key fingerprint, Type “yes” to continue. You have learned how to enable SSH on Raspberry Pi.

**Step 7: Add Swap Space(optional)**

This step is not mandatory, however if you have either Raspberry Pi 4 Model B with RAM 2GB or 4GB proceed by running below commands to add additional swap memory.

You can verify that there is no active swap using the free utility:

free -h

Output:

total used free shared buff/cache available  
Mem: 3.7Gi 206Mi 2.5Gi 3.0Mi 1.1Gi 3.4Gi  
Swap: 0B 0B 0B

We will use the **fallocate** program to create a swap file.

sudo fallocate -l 2G /swapfile

Now check if the file was created.

ls -lh /swapfile

If it was created correctly, you should see something like:

-rw-r--r-- 1 root root 2.0G Oct 26 19:05 /swapfile

Make the file only accessible to root by typing:

sudo chmod 600 /swapfile

Verify the permissions change by typing:

ls -lh /swapfile

Output:

-rw------- 1 root root 2.0G Oct 26 19:05 /swapfile

As you can see, only the root user has the read and write flags enabled.

We can now mark the file as swap space by typing:

sudo mkswap /swapfile

If it was successful, you should see something like

Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)  
no label, UUID=f9dfa448-79af-4dc5-a49c-892fd73ca3ba

Finally we will tell the system to start using our new swap file,

sudo swapon /swapfile

To verify that the swap is now available type:

sudo swapon --show

Result:

NAME TYPE SIZE USED PRIO  
/swapfile file 2G 0B -2

This swap will only last until next reboot. In order to make it permanent, we will add it to the **/etc/fstab** file.

echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab

Reboot you Raspberry PI.

After reboot, verify it using **free -h** command.

total used free shared buff/cache available  
Mem: 3.7Gi 191Mi 3.2Gi 3.0Mi 307Mi 3.5Gi  
Swap: 2.0Gi 0B 2.0Gi

**Step 8: Enable X11 forwarding**

X11 Forwarding allows us to run software on Linux/Unix server with a Windows style GUI (Graphical User Interface). When you run a SSH, TELNET or RLOGIN/RSH session you will be able to display your remote applications directly on your local Windows PC or Mac Os.

* For Mac, use XQuartz. You can install XQuartz using homebrew with **brew cask install xquartz** or directly from the website [here](https://www.xquartz.org/)
* For Windows, use PuTTY and VcXsrv.

Open up a terminal window and launch **XQuartz.**

open -a XQuartz

Allow connections from remote clients

* xQuartz’s Preference → Security → Allow connections from network clients

You can also execute the following command to disable the access control, by which you can allow clients to connect from any host. Open up a terminal window and run the following at the command line:

xhost +

You should see a response like this:

access control disabled, clients can connect from any host

You will see XQuartz server in the Dock.**Leave this terminal window runningwithout closing.**

Open up a second terminal and run the following command in order to connect to your Raspberry Pi, remember to use the **-X** flag to enable X11 window forwarding:

ssh username@hostname -X

You should let your Raspberry Pi forward x11 display. Update **/etc/ssh/sshd\_config**and set **X11 Forwarding**yes, then restart your Raspberry PI server.

Then, install **x11-apps**:

sudo apt install x11-apps

Test X11 by running **xeyes** or **xclock** or any another GUI application you wish. The syntax is as follows on your remote server:

xeyes

If you still get the “cannot open display” error, set the DISPLAY variable as shown below.

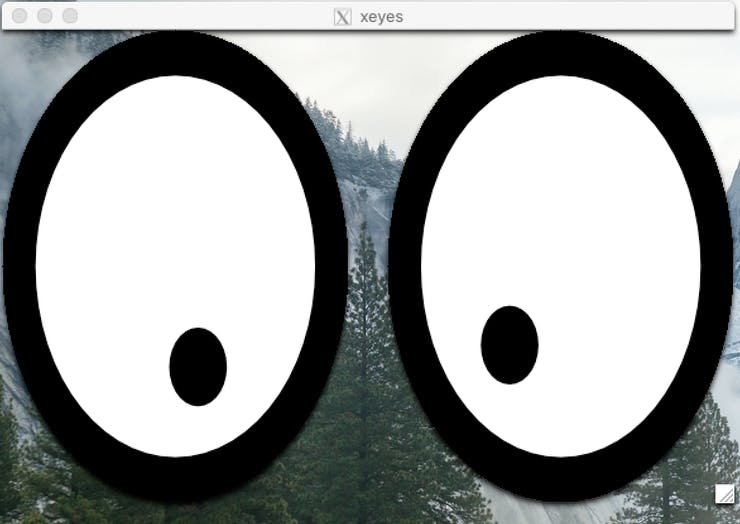
export DISPLAY='<ip\_address\_of\_your\_PC>:0.0'

IP is the local workstation’s IP where you want the GUI application to be displayed.

Run again:

xeyes

If everything is correct, we will get the following screen:



Now you can configure and login using X-servers to X Forward.

**Step 9: Setup your sources.list**

Setup your raspberry pi 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'

If the command succeeds, you won’t see any output message.

**Step 10: Set up your keys**

By running the following command, we will download the key from Ubuntu’s keyserver into the trusted set of keys:

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

Output:

Executing: /tmp/apt-key-gpghome.DFEscOhTF9/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654  
gpg: key F42ED6FBAB17C654: "Open Robotics <info@osrfoundation.org>" not changed  
gpg: Total number processed: 1  
gpg: unchanged: 1

**Step 11: Installation of ROS**

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

sudo apt update

Now pick how much of ROS you would like to install.

* **Desktop-Full Install:** : Everything in **Desktop** plus 2D/3D simulators and 2D/3D perception packages
* **Desktop Install:**Everything in **ROS-Base** plus tools like [rqt](http://wiki.ros.org/rqt)and [rviz](http://wiki.ros.org/rviz)
* **ROS-Base: (Bare Bones)** ROS packaging, build, and communication libraries. No GUI tools.

I’ll go with installing Desktop Install here.

sudo apt install ros-noetic-desktop

It takes around 1 hour for downloading and installation processes.

Finally run,

sudo apt-get install build-essential

**Step 12: Verify Noetic installation on your Raspberry Pi 4**

Let’s try some ROS commands to make sure the installation has finished successfully. A simple way to check the functionality of ROS is to use the **turtlesim** simulator that is part of the ROS installation.

Open up a terminal window and run the following at the command line:

xhost +

**Leave this terminal window running.**

Open a new terminal, connect to your Raspberry Pi via ssh and run the following commands:

You must source in every **bash** terminal you use ROS.

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

Then run,

roscore

**Leave this terminal window running as well.**

Because **roscore** is running in the foreground, you need to open up another terminal window to connect via ssh using **ssh username@hostname -X** to your Raspberry Pi

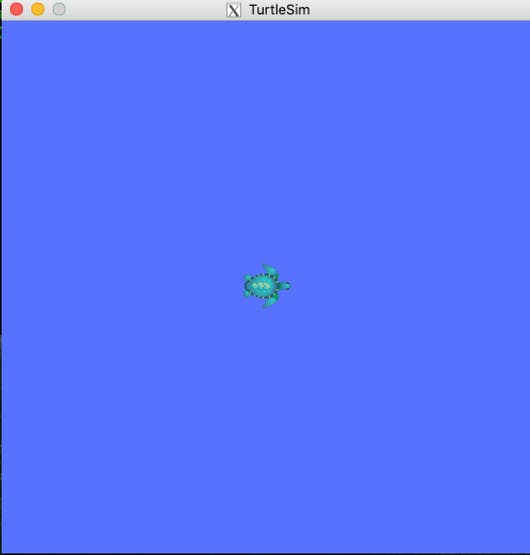
Source it again. Then run:

export DISPLAY='<ip\_address\_of\_your\_PC>:0.0'

and finally run the simulation command.

rosrun turtlesim turtlesim\_node

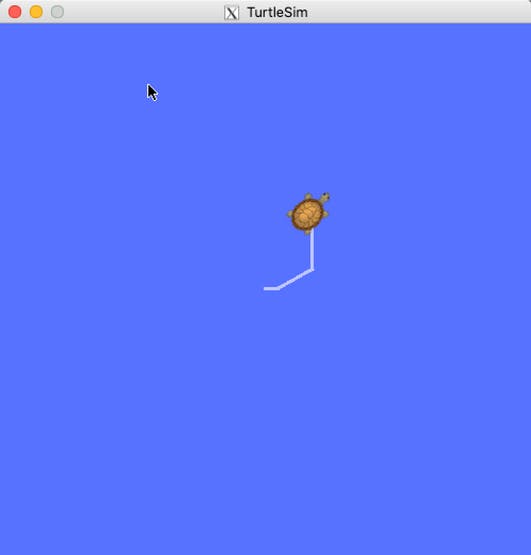
If everything is correct, we will get the following screen:



* Open up yet another terminal window. Run the following:

rosrun turtlesim turtle\_teleop\_key

* Click the mouse in the last container window you created so that it has focus. Use the arrow keys to move the turtle around the screen.
* If everything goes right, you will obtain the following result on current terminal:



Press Ctrl-C to stop **roscore**, etc.

Congratulations! We are done with the ROS installation.

**Step 13: Connect your RPLiDAR to Raspberry Pi**

Connect your RPLiDAR to Raspberry Pi 4 Model B using Micro USB Cable. Flashing green light indicates normal activity of sensor.

Once you have connected the RPLiDAR to your Raspberry Pi, type the following command line to check the permissions:

Open your terminal and run the following command.

ls -l /dev | grep ttyUSB

Output of the following command must be:

crw-rw---- 1 root dialout 188, 0 Jan 3 14:59 ttyUSB

Run below command to change permission:

sudo chmod 666 /dev/ttyUSB0

Once the permissions are configured, you have to download and install the RPLIDAR ROS packages.

**Step 14: Install RPLIDAR ROS Package**

Let’s create a separate workspace for other packages, that are not part of core ROS.

Create the catkin root and source folders:

mkdir -p ~/catkin\_ws/src  
cd ~/catkin\_ws/

Configure the catkin workspace

catkin\_init\_workspace

and source it to bashrc:

echo "source $HOME/catkin\_ws/devel/setup.bash" >> ~/.bashrc

Okay, we’re ready to start installing RPLIDAR ROS package. Go to the source folder of the catkin workspace that you just created:

cd src

Clone the ROS node for the Lidar in the catkin workspace.

sudo git clone https://github.com/Slamtec/rplidar\_ros.git

After that build with catkin.

cd ~/catkin\_ws/

Run **catkin\_make** to compile your catkin workspace.

catkin\_make

Then run to source the environment with your current terminal.**Don't close the terminal.**

source devel/setup.bash

and launch RPILIDAR launch file

roslaunch rplidar\_ros rplidar.launch

If you can see scan messages on terminal windows, everything works as it supposed to work.

**Step 15: Data display with Rviz using X11 forwarding**

RVIZ is a ROS graphical interface that allows you to visualize a lot of information, using plugins for many kinds of available topics.

Open a terminal, connect to your Raspberry Pi via ssh and run the following commands:

source /opt/ros/noetic/setup.bash

Then run roscore:

roscore

Open a new terminal, connect to your Raspberry Pi via ssh and run the following commands:

cd ~/catkin\_ws/

Then run to source the environment with your current terminal.

source devel/setup.bash

Then we launch the Rplidar launch file in the package.

roslaunch rplidar\_ros rplidar.launch

Finally open last terminal window and connect to your Raspberry Pi via ssh using **ssh username@hostname -X** command

Source it:

source /opt/ros/noetic/setup.bash

Then run,

export DISPLAY='<ip\_address\_of\_your\_PC>:0.0'

And finally run

rviz

You will obtain the following result on current terminal:

[ERROR] [1604066165.661029569]: Unable to create the rendering window after 100 tries.  
[ INFO] [1604066165.661137306]: Stereo is NOT SUPPORTED  
terminate called after throwing an instance of 'std::logic\_error'  
 what(): basic\_string::\_M\_construct null not valid  
Aborted (core dumped)

The X protocol is based on remote procedure calls to the X server. Unfortunately, this doesn't really work for OpenGL calls so you cannot run programs that require 3D acceleration using X forwarding.

You can check running **glxgears** command**.** It is a test program to make sure your graphics accelerations are working. You can try **glxgears** as well to prove your OpenGL rendering hardware/software pipeline is working.

glxgears

If glxgears doesn't work then rviz probably won't either. **Rviz** needs OpenGL which doesn't work via X forwarding.

Have a look to [VirtualGL](http://en.wikipedia.org/wiki/VirtualGL)as alternative. VirtualGL is an open source toolkit that gives any Unix or Linux remote display software the ability to run OpenGL applications with full 3D hardware acceleration. Normal X-forwarding sends all X rendering instructions and the client does all the rendering on a CPU. VirtualGL overrides this protocol and renders 3D graphics at the host’s GPU and then send only the rendered 2D images to the client.

So, we are going to install minimal ubuntu desktop on raspberry pi and run **rviz** locally.

**Step 16: Install the desktop components and VNC server**

Once you've installed Ubuntu Server you will need a mechanism to be able to launch GUI application. As a result many people actually install Ubuntu Desktop instead of installing Ubuntu Server since the desktop version of Ubuntu has the GUI built in. There is a better way however, and that is to install VNC. VNC provides a **virtual desktop** so it is more lightweight than a full blown desktop installation.

Virtual Network Computing (VNC) allows you to use GUI applications over a remote connection. Virtual Network Computing is a lot like X11 forwarding on steroids. For the server, the easiest for Ubuntu would be **X11VNC**. It is very simple to install and to use. This program is not only free of charge, open source, but also supports OpenGL programs. For example, rviz and other programs can also open normally.

There are plenty of desktop flavours but [Xfce](https://www.xfce.org/) is perhaps one of the lighter and more popular flavours so we will go for that one. [Xfce](https://www.xfce.org/) is a lightweight desktop environment for UNIX-like operating systems. It aims to be fast and low on system resources, while still being visually appealing and user friendly.

So type the following command:

sudo apt-get install xfce4-goodies xfce4

when prompted type **Y** and then press [Enter]. This will install the basic xfce desktop components. We have three display managers – **LightDM**, **GDM** and **SDDMC.** Choose the display manager over to **lightdm**.

Then reboot your raspberry Pi.

sudo reboot

Connect to your raspberry PI and open a terminal and issue the following

cat /etc/X11/default-display-manager

This will return :

/usr/sbin/lightdm

Open a terminal and run the following commands to update default repositories and install required packages.

sudo apt-get install x11vnc -y

Now create a password to connect using vnc viewer from the client system. This will not require any username to connect vnc.

x11vnc -storepasswd /etc/x11vnc.pwd

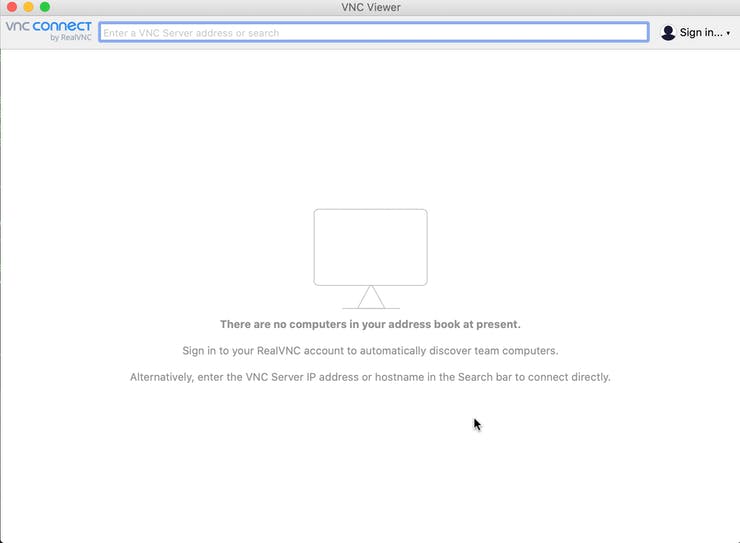
After the successful installation of the x11vnc server on your system. Let’s start it using the following command. Change the parameters as per your setup.

sudo x11vnc -auth guess -capslock -forever -loop -noxdamage -repeat -rfbauth /etc/x11vnc.pass -rfbport 5900 -shared

The VNC server will start on default port 5900.

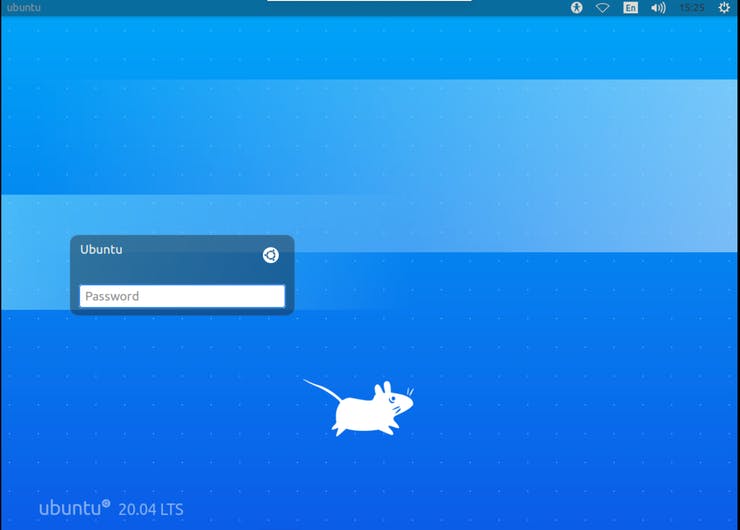
To connect to the server using VNC, you need a VNC client on your system. VNC viewer clients are available for Windows, Linux and Macintosh workstations.

For this tutorial, I am using [RealVNC viewer](https://www.realvnc.com/download/viewer/)on our system. You can use any other client of your choice.



Start your VNC viewer and open a new connection using ip address of your Raspberry Pi.

Your VNC desktop session appears. Now you have connected to **Ubuntu Server**, and you can use graphical user interface to work with it.



You can successfully configured VNC server on your Ubuntu system.

**Step 17: Running rviz with VNC on a remote computer**

We assume that you've already done the installation step.

Open a terminal windows and run the following command:

source /opt/ros/noetic/setup.bash

Then run roscore:

roscore

Without roscore running, the various nodes within ROS can’t talk to each other. So you need to leave it running in the background.

Open a new terminal and run the following commands:

cd ~/catkin\_ws/

Then run to source the environment with your current terminal.

source devel/setup.bash

and launch RPILIDAR launch file

roslaunch rplidar\_ros rplidar.launch

To check if the node we just launched really publishing something, we open another terminal windown and type:

rostopic echo -n1 /scan

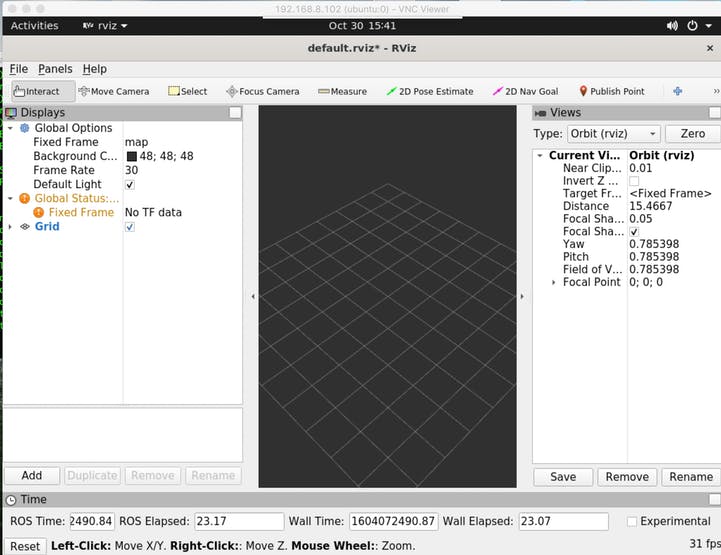
If everything goes right, you will obtain the following result on current terminal:

header:  
seq: 5032  
stamp:  
secs: 1604073153  
nsecs: 896877623  
frame\_id: "laser"  
angle\_min: -3.1241390705108643  
angle\_max: 3.1415927410125732  
angle\_increment: 0.01745329238474369  
time\_increment: 0.00037511246046051383  
scan\_time: 0.1346653699874878  
range\_min: 0.15000000596046448  
range\_max: 12.0  
ranges: [0.32100000977516174, 0.32100000977516174, 0.3240000009536743, 0.3269999921321869, 0.33000001311302185, 0.3319999873638153, 0.3330000042915344, 0.3370000123977661, 0.33899998664855957, 0.34299999475479126, 0.3449999988079071, 0.34700000286102295, 0.35199999809265137, 0.3540000021457672, 0.3580000102519989, 0.3630000054836273, 0.36899998784065247, 0.3709999918937683, 0.37700000405311584, 0.38100001215934753, 0.3889999985694885, 0.3919999897480011, 0.3959999978542328, inf, 0.2160000056028366, 0.2160000056028366, inf, inf, inf, 5.068999767303467, inf, inf, inf, inf, inf, 4.505000114440918, 4.375, 4.3429999351501465, 4.244999885559082, inf, inf, 0.33399999141693115, 0.32600000500679016, 0.3190000057220459, 0.3149999976158142, 0.3059999942779541, 0.3019999861717224, 0.3009999990463257, 0.29899999499320984, 0.3019999861717224, 0.31200000643730164, 0.33000001311302185, inf, inf, inf, inf, 3.38700008392334, 3.367000102996826, 3.3239998817443848, 3.308000087738037, 3.2699999809265137, 3.256999969482422, 3.239000082015991, 3.2339999675750732, 3.2090001106262207, 3.196000099182129, 3.1689999103546143, 3.1579999923706055, 3.1429998874664307, 3.132999897003174, 3.11899995803833, 3.115000009536743, 3.1080000400543213, 3.0829999446868896, 3.0880000591278076, 3.0799999237060547, 3.072000026702881, 3.049999952316284, 3.0480000972747803, 3.0429999828338623, 3.0460000038146973, 3.0409998893737793, 3.0420000553131104, 3.0429999828338623, 3.0360000133514404, 2.51200008392334, 2.1019999980926514, 1.9160000085830688, 1.593999981880188, 1.3669999837875366, 1.2120000123977661, 1.1540000438690186, 1.0420000553131104, 0.949999988079071, 0.9120000004768372, 0.8379999995231628, 0.7710000276565552, 0.7160000205039978, 0.6899999976158142, 0.6449999809265137, 0.6079999804496765, 0.5899999737739563, 0.5609999895095825, 0.5360000133514404, 0.5109999775886536, 0.5, 0.47999998927116394, 0.460999995470047, 0.4519999921321869, 0.4339999854564667, 0.4180000126361847, 0.41100001335144043, 0.38999998569488525, 0.382999986410141, 0.3700000047683716, 0.35899999737739563, 0.3529999852180481, 0.3440000116825104, 0.33799999952316284, 0.32899999618530273, 0.3199999928474426, 0.3160000145435333, inf, 0.3050000071525574, 0.2939999997615814, 0.29100000858306885, inf, 0.27799999713897705, 0.27000001072883606, 0.27300000190734863, 0.26600000262260437, 0.257999986410141, 0.25600001215934753, 0.2540000081062317, 0.25, 0.24799999594688416, 0.23999999463558197, 0.24400000274181366, 0.23899999260902405, 0.2370000034570694, 0.23000000417232513, 0.2329999953508377, 0.2280000001192093, 0.2240000069141388, 0.22300000488758087, 0.22100000083446503, inf, 0.2199999988079071, 0.2150000035762787, 0.21400000154972076, 0.21199999749660492, 0.21299999952316284, 0.20999999344348907, 0.20900000631809235, 0.20800000429153442, 0.2070000022649765, 0.20600000023841858, 0.20499999821186066, 0.20399999618530273, 0.20200000703334808, 0.2029999941587448, 0.20100000500679016, 0.20000000298023224, 0.19900000095367432, 0.19900000095367432, 0.1979999989271164, 0.1979999989271164, 0.1979999989271164, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.19699999690055847, 0.1979999989271164, 0.1979999989271164, 0.19900000095367432, 0.1979999989271164, 0.19900000095367432, 0.20000000298023224, 0.20000000298023224, 0.20100000500679016, 0.20200000703334808, 0.20200000703334808, 0.2029999941587448, 0.20399999618530273, 0.20499999821186066, 0.2070000022649765, 0.20800000429153442, 0.20999999344348907, 0.20999999344348907, 0.21299999952316284, 0.21400000154972076, 0.2160000056028366, 0.21899999678134918, 0.2199999988079071, 0.22200000286102295, 0.22499999403953552, 0.22599999606609344, 0.2280000001192093, 0.23199999332427979, 0.2370000034570694, 0.23499999940395355, inf, 0.23899999260902405, inf, 0.24400000274181366, 0.2529999911785126, 0.257999986410141, 0.25600001215934753, 0.25999999046325684, 0.2619999945163727, 0.27000001072883606, 0.2669999897480011, 0.2759999930858612, 0.2809999883174896, 0.28999999165534973, 0.29899999499320984, 0.30399999022483826, 0.3089999854564667, 0.3140000104904175, 0.3190000057220459, 0.3310000002384186, inf, 0.34299999475479126, 0.3580000102519989, 0.36500000953674316, 0.3790000081062317, 0.38999998569488525, 0.4090000092983246, 0.4180000126361847, 0.42899999022483826, 0.44999998807907104, 0.46299999952316284, 0.47699999809265137, 0.5049999952316284, 0.5210000276565552, 0.5389999747276306, 0.5789999961853027, 0.6039999723434448, 0.6269999742507935, 0.6959999799728394, inf, 0.3179999887943268, 0.8270000219345093, 0.8640000224113464, inf, inf, inf, inf, 2.0380001068115234, 2.0829999446868896, inf, inf, 0.5830000042915344, 2.578000068664551, inf, 2.628999948501587, 2.63100004196167, 2.628999948501587, 2.627000093460083, 2.631999969482422, 2.63100004196167, 2.63100004196167, 2.634999990463257, 2.63700008392334, 2.635999917984009, 2.6410000324249268, 2.6449999809265137, 2.6510000228881836, inf, inf, inf, 0.6600000262260437, inf, 0.6489999890327454, 0.6349999904632568, 0.628000020980835, 0.6150000095367432, inf, inf, inf, 2.819999933242798, 2.8410000801086426, 2.864000082015991, 2.880000114440918, inf, 2.450000047683716, 2.3450000286102295, 2.2929999828338623, 2.24399995803833, 2.1579999923706055, 2.1480000019073486, 2.1630001068115234, 2.197999954223633, 2.2109999656677246, 2.25, 2.2739999294281006, 2.318000078201294, 2.3329999446868896, 2.374000072479248, 2.4040000438690186, 2.444999933242798, 2.4749999046325684, 2.5350000858306885, 2.5510001182556152, 2.6070001125335693, 2.6470000743865967, 2.9010000228881836, 2.937000036239624, 3.0220000743865967, 3.061000108718872, 3.1440000534057617, 3.194999933242798, 4.936999797821045, 5.168000221252441, 4.9720001220703125, 3.5, 4.695000171661377, 4.638000011444092, inf, 3.996000051498413, 4.1539998054504395, 4.26800012588501, 4.584000110626221, 4.554999828338623, 4.502999782562256, 4.505000114440918, inf, 0.7960000038146973, 0.7879999876022339, 0.7770000100135803, 0.7710000276565552, 0.7699999809265137, 0.7720000147819519, 0.7699999809265137, 0.8100000023841858, inf, inf, 0.3100000023841858, 0.3089999854564667, 0.3070000112056732, 0.3070000112056732, 0.3070000112056732, 0.3059999942779541, 0.3070000112056732, 0.3070000112056732, 0.30799999833106995, 0.30799999833106995, 0.3089999854564667, 0.3100000023841858, inf, 0.3109999895095825, 0.31299999356269836, 0.3149999976158142, 0.3160000145435333, 0.31700000166893005, 0.3179999887943268]  
intensities: [47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 0.0, 0.0, 0.0, 47.0, 0.0, 0.0, 0.0, 0.0, 0.0, 47.0, 47.0, 47.0, 47.0, 0.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 0.0, 0.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 0.0, 0.0, 0.0, 0.0, 47.0, 47.0, 0.0, 0.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 0.0, 0.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 0.0, 0.0, 0.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0, 0.0, 47.0, 47.0, 47.0, 47.0, 47.0, 47.0]  
---

Let’s check it further with Rviz by typing.

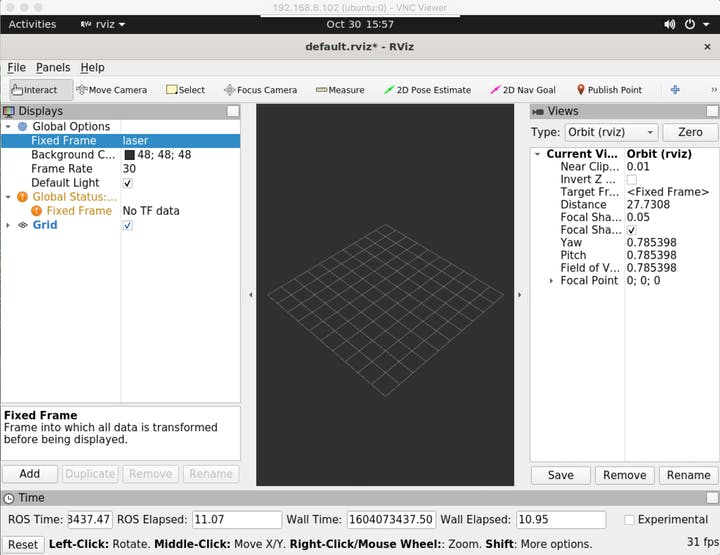
rviz

You'll see the rviz window pops out.

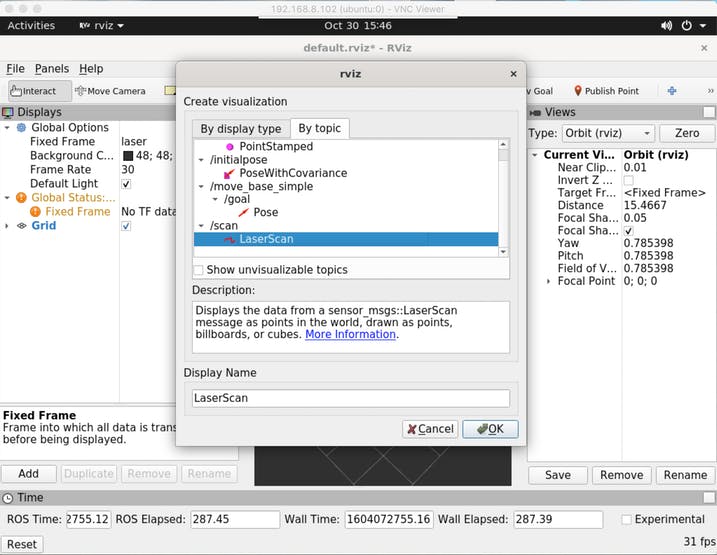


The visualization is not showing due to some problems with the frame.

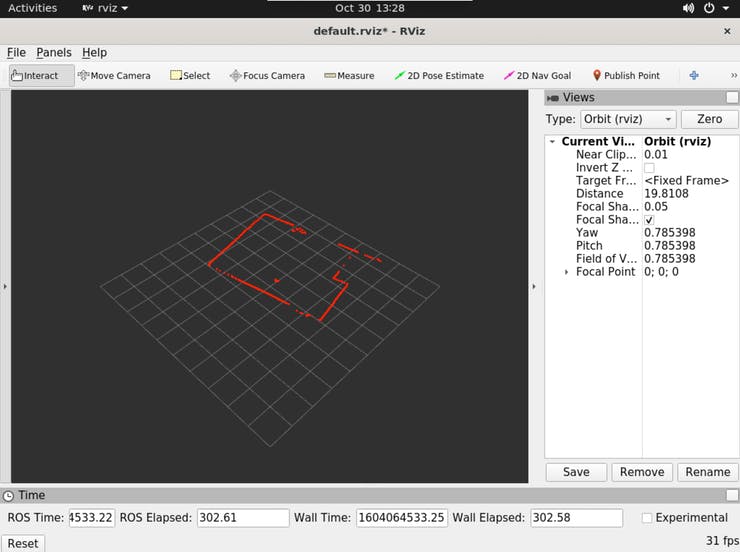
We now need to tell rviz which fixed frame we want to use. Change fixed frame to **laser.**



Now add a LaserScan using Add button.



Now everything looks fine.



**Rviz** will open with a map of the RPLIDAR’s surroundings.

The view from the laser scanner will be marked in red. The laser scanner is positioned at the center of the grid, it has a range of roughly 15cm to 6 meters, so you’ll be able to see everything around it on its scanning plane within that range.