ROS

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

```
ros2 <command>
```

Some useful commands:

1. action:

    1.1. info: shows info about an action

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 action info send_goal
Action: send_goal
Action clients: 0
Action servers: 0
```

    1.2. list: output a list of action names
    1.3. send_goal: send an action goal

2. list : shows all the available packages

3. executables [package name] : shows all the available executables

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 pkg executables turtlesim
turtlesim draw_square
turtlesim mimic
turtlesim turtle_teleop_key
turtlesim turtlesim_node
```

4. run <package> <executable>

Eg: ros2 run turtlesim turtlesim_node: will open the turtlesim gui

5. node

    5.1. list: shows all the nodes that are currently there
    5.2. info: shows subscribers, publishers, service servers, service clients, action servers, and action clients

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 node info /turtlesim –
/turtlesim
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /turtle1/cmd_vel: geometry_msgs/msg/Twist
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /turtle1/color_sensor: turtlesim/msg/Color
    /turtle1/pose: turtlesim/msg/Pose
  Service Servers:
    /clear: std_srvs/srv/Empty
    /kill: turtlesim/srv/Kill
    /reset: std_srvs/srv/Empty
    /spawn: turtlesim/srv/Spawn
    …
    /turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:
    /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
  Action Clients:
```

6. interface proto <type>: shows prototype

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 interface proto turtlesim/srv/Spawn
"x: 0.0
y: 0.0
theta: 0.0
name: ''
"
```

7. topic pub [-t times] <topic> <message>

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 topic pub -1 /turtle1/cmd_vel
geometry_msgs/msg/Twist "linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 2.0
"
publisher: beginning loop
publishing #1: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0,
z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=2.0))
```

8. service
    8.1. list -t: shows the available services and their type

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 service list -t
/clear [std_srvs/srv/Empty]
/kill [turtlesim/srv/Kill]
/reset [std_srvs/srv/Empty]
```

```
/spawn [turtlesim/srv/Spawn]
…
/turtlesim/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
```

8.2. call <service name> <service type> [values]

```
Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 call /spawn turtlesim/srv/Spawn
"x: 0.0
y: 0.0
theta: 0.0
name: "turty"
"
```

9. param: parameters for nodes

10. launch <package> [launch file]

Publisher and Subscriber using C++:

```
cubesat@cubesat-OptiPlex-7460-AIO:~$ cd ros2_ws/src/
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws/src$ ros2 pkg create --build-type ament_cmake
cpp_topic
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws/src$ cd..
cd..: command not found
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws/src$ cd ..
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ colcon build
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ ros2 run cpp_topic
cpp_topic_publisher_spiral
[INFO] [1720506984.563733873] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '2.500000'
[INFO] [1720506985.063744280] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '2.600000'
[INFO] [1720506985.563968460] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '2.700000'
[INFO] [1720506986.063875405] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '2.800000'
[INFO] [1720506986.563874948] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '2.900000'
[INFO] [1720506987.063751099] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '3.000000'
[INFO] [1720506987.563706104] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '3.100000'
[INFO] [1720506988.063900609] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '3.200000'
```

```
[INFO] [1720506988.563896863] [cpp_topic_publisher_spiral]: Sending - Linear Velocity :
'4.000000', Angular Velocity : '3.300000'
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ ros2 run cpp_topic
cpp_topic_subscriber_spiral
[INFO] [1720506986.564407289] [cpp_topic_subscriber_spiral]: Recieved - Linear Velocity :
'4.000000', Angular Velocity : '2.900000'
[INFO] [1720506987.064263391] [cpp_topic_subscriber_spiral]: Recieved - Linear Velocity :
'4.000000', Angular Velocity : '3.000000'
[INFO] [1720506987.564207517] [cpp_topic_subscriber_spiral]: Recieved - Linear Velocity :
'4.000000', Angular Velocity : '3.100000'
[INFO] [1720506988.064409149] [cpp_topic_subscriber_spiral]: Recieved - Linear Velocity :
'4.000000', Angular Velocity : '3.200000'
[INFO] [1720506988.564411002] [cpp_topic_subscriber_spiral]: Recieved - Linear Velocity :
'4.000000', Angular Velocity : '3.300000'
```

Service Server and Client using Python:

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws/src$ ros2 pkg create --build-type ament_python
py_service
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws/src$ cd ..
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ colcon build
```

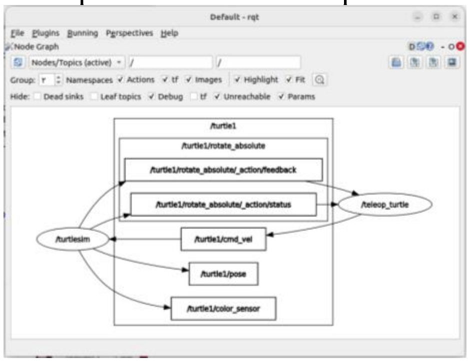
```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ ros2 run py_service py_service_server_polar
[INFO] [1720507959.416145756] [py_service_server_polar]: Response - Polar Radial
Coordinate : 7.071068, Polar Angular Coordinate : 45.000000
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2_ws$ ros2 run py_service py_service_client_polar
[INFO] [1720507959.404140904] [py_service_client_polar]: Sending - X Coordinate :
5.000000, Y Coordinate : 5.000000
[INFO] [1720507959.416642845] [py_service_client_polar]: Success
```

RQt

Plugins:

1. Introspection -> Node Graph

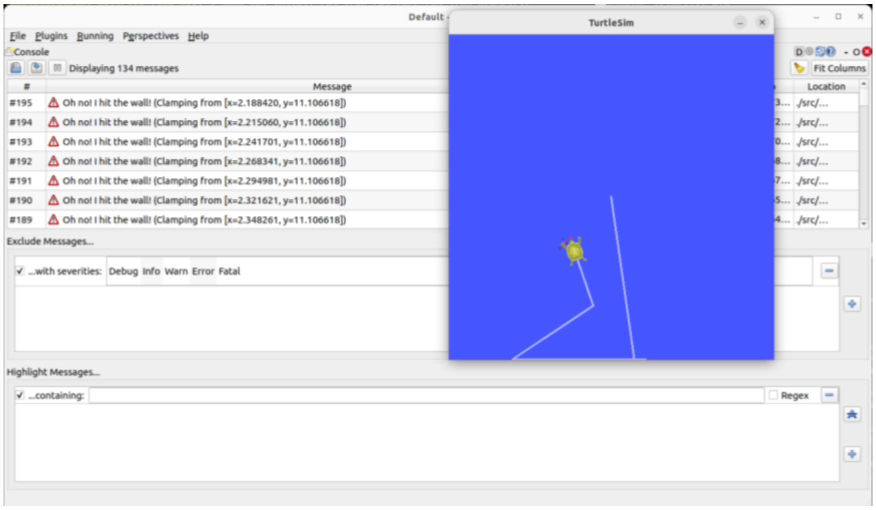


Nodes/Topics (active) / /

Group: ☑ Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit

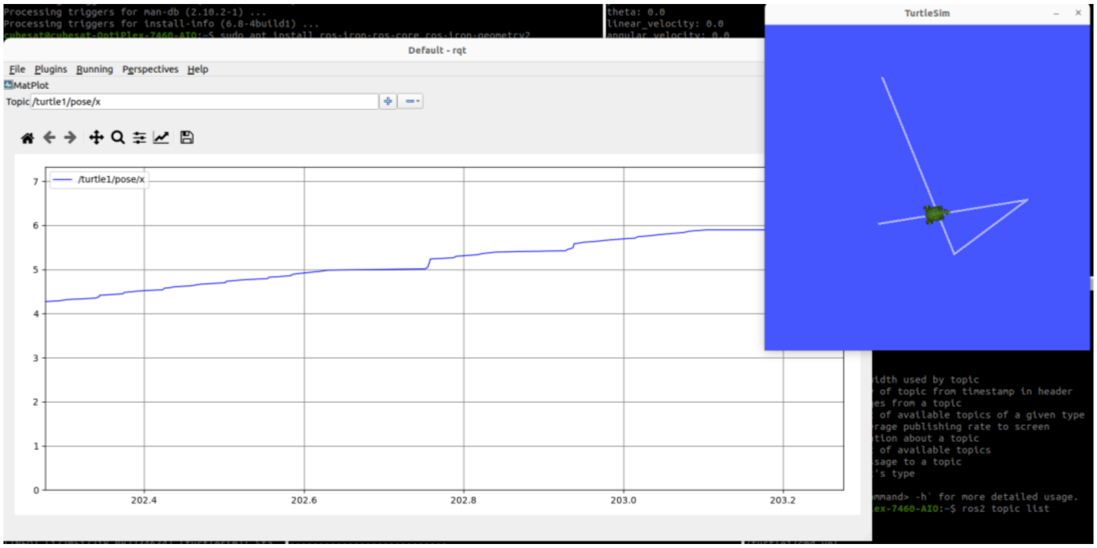
Hide: □ Dead sinks □ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params

2. Logging -> Console



```
#195  Oh no! I hit the wall! (Clamping from [x=2.188420, y=11.106618])  3...  ./src/...
#194  Oh no! I hit the wall! (Clamping from [x=2.215060, y=11.106618])  2...  ./src/...
#193  Oh no! I hit the wall! (Clamping from [x=2.241701, y=11.106618])  0...  ./src/...
#192  Oh no! I hit the wall! (Clamping from [x=2.268341, y=11.106618])  8...  ./src/...
#191  Oh no! I hit the wall! (Clamping from [x=2.294981, y=11.106618])  7...  ./src/...
#190  Oh no! I hit the wall! (Clamping from [x=2.321621, y=11.106618])  5...  ./src/...
#189  Oh no! I hit the wall! (Clamping from [x=2.348261, y=11.106618])  4...  ./src/...
```

3. Plots



```
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for install-info (6.8-4build1) ...
cubesat@cubesat-OptiPlex-7460-AIO:~$ sudo apt install ros-iron-ros-core ros-iron-geometry2
```

```
theta: 0.0
linear_velocity: 0.0
angular_velocity: 0.0
```

```
idth used by topic
 of topic from timestamp in header
es from a topic
 of available topics of a given type
rage publishing rate to screen
tion about a topic
 of available topics
sage to a topic
's type

mmand> -h` for more detailed usage.
ex-7460-AIO:~$ ros2 topic list
```

Bag

Records data published on a topic and saves it. The data can be used to reproduce the result of our testcases.

```
cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 bag record /turtle1/cmd_vel
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/rosbag2_2024_07_09-13_38_45$ ros2 bag info rosbag2_2024_07_09-13_38_45_0.mcap

Files:          rosbag2_2024_07_09-13_38_45_0.mcap
Bag size:         8.3 KiB
```

```
Storage id:        mcap
Duration:          9.537s
Start:             Jul  9 2024 13:38:56.747 (1720517936.747)
End:               Jul  9 2024 13:39:06.284 (1720517946.284)
Messages:          61
Topic information: Topic: /turtle1/cmd_vel | Type: geometry_msgs/msg/Twist | Count: 61 |
Serialization Format: cdr
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/rosbag2_2024_07_09-13_38_45$ ros2 bag play
rosbag2_2024_07_09-13_38_45_0.mcap
```

The turtle will move according to the data stored.

Gazebo

```
Sudo apt install ros-iron-gazebo-ros-pkgs
```

Running a demo simulation:

```
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo_plugins/worlds$ gazebo --
verbose /opt/ros/iron/share/gazebo_plugins/worlds/gazebo_ros_diff_drive_demo.world
```

To move the robot forward:

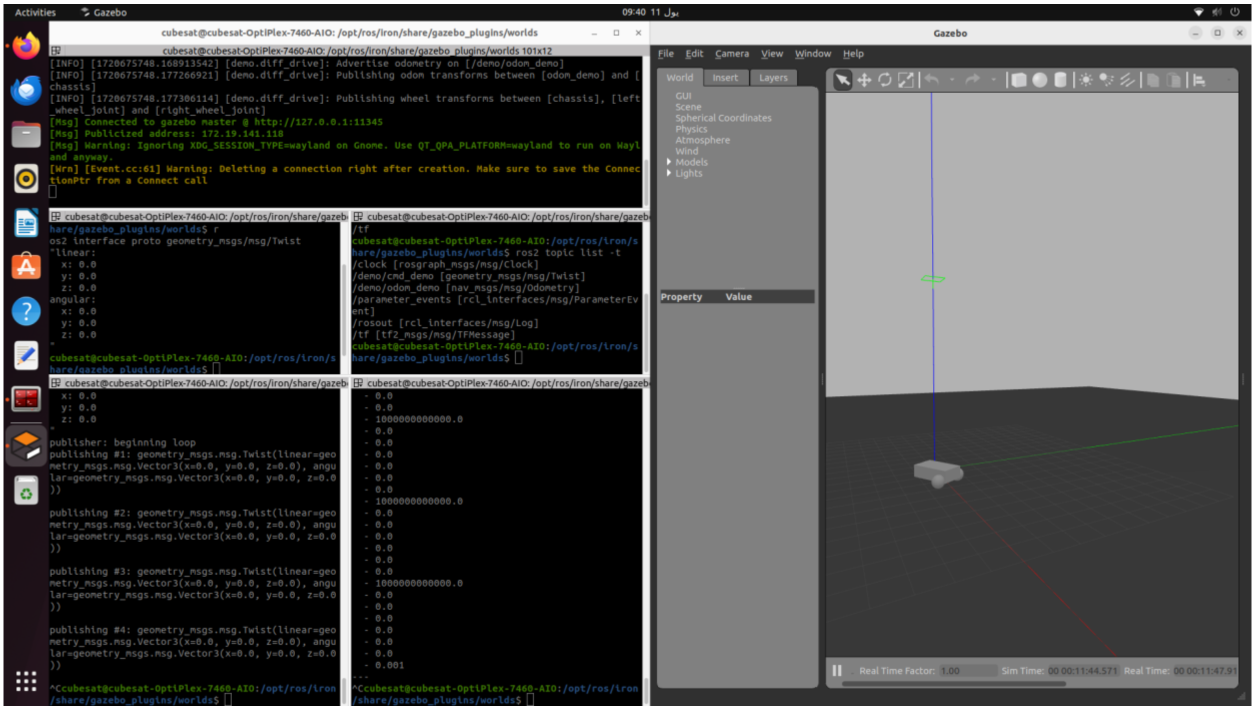
```
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo_plugins/worlds$ ros2 topic pub
/demo/cmd_demo geometry_msgs/msg/Twist "linear:
  x: 1.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
"
```

To stop the movement:

```
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo_plugins/worlds$ ros2 topic pub
/demo/cmd_demo geometry_msgs/msg/Twist "linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
"
```

To see odometry:

```
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo_plugins/worlds$ ros2 topic echo /demo/odom_demo
```


Activities
Gazebo
09:40 11 يول
cubesat@cubesat-OptiPlex-7460-AIO: /opt/ros/iron/share/gazebo_plugins/worlds
cubesat@cubesat-OptiPlex-7460-AIO: /opt/ros/iron/share/gazebo_plugins/worlds 101x12
cubesat@cubesat-OptiPlex-7460-AIO: /opt/ros/iron/share/gazeb
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value

```
[INFO] [1720675748.168913542] [demo.diff_drive]: Advertise odometry on [/demo/odom_demo]
[INFO] [1720675748.177266921] [demo.diff_drive]: Publishing odom transforms between [odom_demo] and [
chassis]
[INFO] [1720675748.177306114] [demo.diff_drive]: Publishing wheel transforms between [chassis], [left
_wheel_joint] and [right_wheel_joint]
[Msg] Connected to gazebo master @ http://127.0.0.1:11345
[Msg] Publicized address: 172.19.141.118
[Msg] Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayl
and anyway.
[Wrn] [Event.cc:61] Warning: Deleting a connection right after creation. Make sure to save the Connec
tionPtr from a Connect call
```

```
hare/gazebo_plugins/worlds$ r
os2 interface proto geometry_msgs/msg/Twist
"linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
"
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/s
hare/gazebo_plugins/worlds$
```

```
/tf
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/s
hare/gazebo_plugins/worlds$ ros2 topic list -t
/clock [rosgraph_msgs/msg/Clock]
/demo/cmd_demo [geometry_msgs/msg/Twist]
/demo/odom_demo [nav_msgs/msg/Odometry]
/parameter_events [rcl_interfaces/msg/ParameterEv
ent]
/rosout [rcl_interfaces/msg/Log]
/tf [tf2_msgs/msg/TFMessage]
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/s
hare/gazebo_plugins/worlds$
```

```
  x: 0.0
  y: 0.0
  z: 0.0
"
publisher: beginning loop
publishing #1: geometry_msgs.msg.Twist(linear=geo
metry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0), angu
lar=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0
))

publishing #2: geometry_msgs.msg.Twist(linear=geo
metry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0), angu
lar=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0
))

publishing #3: geometry_msgs.msg.Twist(linear=geo
metry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0), angu
lar=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0
))

publishing #4: geometry_msgs.msg.Twist(linear=geo
metry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0), angu
lar=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0
))

^Ccubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron
/share/gazebo_plugins/worlds$
```

```
  - 0.0
  - 0.0
  - 1000000000000.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 1000000000000.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 1000000000000.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.0
  - 0.001
---
^Ccubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron
/share/gazebo_plugins/worlds$
```

Real Time Factor: 1.00 Sim Time: 00 00:11:44.571 Real Time: 00 00:11:47.91

Making a robot and an environment:

1. Make a package ws_gazebo

```
cubesat@cubesat-OptiPlex-7460-AIO:/$ mkdir -p ~/ws_gazebo/src
cubesat@cubesat-OptiPlex-7460-AIO:/$ cd ~/ws_gazebo/
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ colcon build
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ cd src/
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src$ ros2 pkg create --build-type ament_cmake gazebo_test
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src$ cd gazebo_test/

cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src/gazebo_test$ ls -l
total 16
-rw-rw-r-- 1 cubesat cubesat  904 09:53 11 يول CMakeLists.txt
drwxrwxr-x 3 cubesat cubesat 4096 09:53 11 يول include
-rw-rw-r-- 1 cubesat cubesat  601 09:53 11 يول package.xml
drwxrwxr-x 2 cubesat cubesat 4096 09:53 11 يول src
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src/gazebo_test$ mkdir launch model
```

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src/gazebo_test$ ls -l
total 24
-rw-rw-r-- 1 cubesat cubesat  904 09:53 11 يول CMakeLists.txt
drwxrwxr-x 3 cubesat cubesat 4096 09:53 11 يول include
drwxrwxr-x 2 cubesat cubesat 4096 09:54 11 يول launch
drwxrwxr-x 2 cubesat cubesat 4096 09:54 11 يول model
-rw-rw-r-- 1 cubesat cubesat  601 09:53 11 يول package.xml
drwxrwxr-x 2 cubesat cubesat 4096 09:53 11 يول src

cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo/src/gazebo_test$ cd ~/ws_gazebo/
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ colcon build
```

2. Make xacro and gazebo files for the robot

robot.xacro

robot.gazebo

3. Make the file for the world

empty_world.world

4. Make the launch file

gazebo_model.launch.py

5. Edit package.xml and CMakeLists.txt to include all dependencies

6. Run on ros

```
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ colcon build
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ source ~/ws_gazebo/install/setup.bash
cubesat@cubesat-OptiPlex-7460-AIO:~/ws_gazebo$ ros2 launch
gazebo_test gazebo_model.launch.py
```