Launch:

```
import os
from ament_index_python.packages import
get_package_share_directory
from launch import LaunchDescription
from launch.actions import IncludeLaunchDescription
from launch.launch_description_sources import
PythonLaunchDescriptionSource
from launch_ros.actions import Node
import xacro

def generate_launch_description():
   # This must match the robot name used in the Xacro file
   robot_name = 'differential_drive_robot'

   # Name of your ROS 2 package
   package_name = 'mobile_robot'

   # Relative paths to model and world files
   model_path_rel = 'model/robot.xacro'
   world_path_rel = 'model/empty_world.world'

   # Absolute paths
   model_path =
os.path.join(get_package_share_directory(package_name),
model_path_rel)
   world_path =
os.path.join(get_package_share_directory(package_name),
world_path_rel)

   # Parse the xacro file to get the robot description
   robot_description_config = xacro.process_file(model_path)
   robot_description = robot_description_config.toxml()

   # Launch Gazebo with the specified world
```

```
    gazebo_launch_file =
os.path.join(get_package_share_directory('gazebo_ros'),
'launch', 'gazebo.launch.py')
    gazebo = IncludeLaunchDescription(
        PythonLaunchDescriptionSource(gazebo_launch_file),
        launch_arguments={'world': world_path}.items()
    )

    # Node to publish the robot state to TF
    robot_state_publisher_node = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        parameters=[{'robot_description': robot_description,
'use_sim_time': True}],
        output='screen'
    )

    # Node to spawn the robot in Gazebo
    spawn_entity_node = Node(
        package='gazebo_ros',
        executable='spawn_entity.py',
        arguments=['-topic', 'robot_description', '-entity',
robot_name],
        output='screen'
    )

    # Final LaunchDescription
    return LaunchDescription([
        gazebo,
        robot_state_publisher_node,
        spawn_entity_node
    ])
```

Gazebo :

```
<?xml version="1.0"?>
<robot>

 <!-- Friction, material properties for links -->
 <gazebo reference="body_link">
   <mu1>0.2</mu1>
   <mu2>0.2</mu2>
   <material>Gazebo/Red</material>
 </gazebo>
 <gazebo reference="wheel1_link">
   <mu1>0.2</mu1>
   <mu2>0.2</mu2>
   <material>Gazebo/Yellow</material>
 </gazebo>
 <gazebo reference="wheel2_link">
   <mu1>0.2</mu1>
   <mu2>0.2</mu2>
   <material>Gazebo/Yellow</material>
 </gazebo>
 <gazebo reference="wheel3_link">
   <mu1>0.2</mu1>
   <mu2>0.2</mu2>
   <material>Gazebo/Yellow</material>
 </gazebo>
 <gazebo reference="wheel4_link">
   <mu1>0.2</mu1>
   <mu2>0.2</mu2>
   <material>Gazebo/Yellow</material>
 </gazebo>

 <!-- Differential / Skid-steer style plugin -->
 <gazebo>
```

```
   <plugin name="skid_steer_drive"
filename="libgazebo_ros_diff_drive.so">
     <ros>
       <!-- optional namespace or remapping -->
       <namespace>/</namespace>
       <!-- remappings if needed -->
       <!-- <remapping>cmd_vel:=cmd_drive</remapping> -->
       <!-- <remapping>odom:=odom</remapping> -->
     </ros>

     <!-- Number of wheel pairs -->
     <num_wheel_pairs>2</num_wheel_pairs>

     <!-- Wheel joints for each pair -->
     <!-- Pair 0 -->
     <left_joint>wheel4_joint</left_joint>
     <right_joint>wheel3_joint</right_joint>
     <!-- Pair 1 -->
     <left_joint>wheel2_joint</left_joint>
     <right_joint>wheel1_joint</right_joint>

     <!-- Kinematic parameters -->
     <wheel_separation>${body_link_y_dim +
wheel_link_length}</wheel_separation>
     <wheel_diameter>${2.0 *
wheel_link_radius}</wheel_diameter>

     <!-- Control limits -->
     <max_wheel_torque>1000</max_wheel_torque>
     <max_wheel_acceleration>5.0</max_wheel_acceleration>

     <!-- Output / odometry settings -->
     <publish_odom>true</publish_odom>
     <publish_odom_tf>true</publish_odom_tf>
     <publish_wheel_tf>true</publish_wheel_tf>

     <odometry_frame>odom</odometry_frame>
     <robot_base_frame>body_link</robot_base_frame>
   </plugin>
 </gazebo>

</robot>
```

Xacro:

```
<?xml version="1.0"?>
<robot name="differential_drive_robot"
xmlns:xacro="http://www.ros.org/wiki/xacro">

 <!-- body dimensions -->
 <xacro:property name="body_link_x_dim" value="1.0"/>
 <xacro:property name="body_link_y_dim" value="0.6"/>
 <xacro:property name="body_link_z_dim" value="0.3"/>
  <!-- Wheel dimensions-->
 <xacro:property name="wheel_link_radius" value="0.15"/>
 <xacro:property name="wheel_link_length" value="0.1"/>
 <xacro:property name="wheel_link_z_location" value="-0.1"/>

 <!--Material density-->
 <xacro:property name="body_density" value="2710.0"/>
 <xacro:property name="wheel_density" value="2710.0"/>
<!-- Pi constant -->
<xacro:property name="pi_const" value="3.14159265"/>
<!-- Robot body and wheel mass -->
<xacro:property name="body_mass"
value="${body_density*body_link_x_dim*body_link_y_dim*body_link_z_dim}"/>
<xacro:property name="wheel_mass"
value="${wheel_density*pi_const*wheel_link_radius*wheel_link_radius*wheel_link_length}"/>
<!-- Moments of inertia of the wheel -->
<xacro:property name="Iz_wheel"
value="${0.5*wheel_mass*wheel_link_radius*wheel_link_radius}" />
<xacro:property name="I_wheel"
value="${(1.0/12.0)*wheel_mass*(3.0*wheel_link_radius*wheel_link_radius+wheel_link_length*wheel_link_length)}" />
<!-- This macro defines the complete inertial section of the wheel
-->
<!-- It is used later in the code -->
```

```
<xacro:macro name="inertia_wheel">
    <inertial>
    <origin rpy="0 0 0" xyz="0 0 0" />
    <mass value="${wheel_mass}" />
    <inertia ixx="${I_wheel}" ixy="0.0" ixz="0.0" iyy="${I_wheel}"
iyz="0" izz="${Iz_wheel}" />
    </inertial>
</xacro:macro>
<!--Over here we include the file that defines extra Gazebo options
and notion control driver-->
<xacro:include filename="$(find mobile_robot)/model/robot.gazebo"/>
<!--###############################################################
############################-->
<!--FROM HERE WE DEFINE LINKS, JOINTS        -->
<!--###############################################################
############################-->
<!-- we need to have this dummy link otherwise gazebo will
compliain -->
<link name ="dummy"/>
<joint name="dummy_joint" type ="fixed">
 <parent link ="dummy"/>
 <child link="body_link"/>
 </joint>

<!--#########################################################-->
<!--START : Body link of the robot -->
<!--###############################################################
####-->
<link name="body_link">
   <visual>
      <geometry>
           <box size="${body_link_x_dim} ${body_link_y_dim}
${body_link_z_dim}"/>
      </geometry>
   <origin rpy="0 0 0" xyz="0 0 0"/>
   </visual>
   <collision>
    <geometry>
       <box size="${body_link_x_dim} ${body_link_y_dim}
${body_link_z_dim}"/>
     </geometry>
     <origin rpy="0 0 0" xyz="0 0 0"/>
   </collision>
   <inertial>
          <origin rpy="0 0 0" xyz="0 0 0"/>
          <mass value ="${body_mass}"/>
       <inertia


ixx="${(1/12)*body_mass*(body_link_y_dim*body_link_y_dim+body_link_
z_dim*body_link_z_dim)}"
           ixy="0"
           ixz="0"

iyy="${(1/12)*body_mass*(body_link_x_dim*body_link_x_dim+body_link_
z_dim*body_link_z_dim)}"
           iyz="0"

izz="${(1/12)*body_mass*(body_link_y_dim*body_link_y_dim+body_link_
x_dim*body_link_x_dim)}"/>
   </inertial>
</link>
<!--##############################################-->
<!--END: Body link of the robot -->
<!--####################################################-->
<!--START: back right wheel of the robot and the joint-->
<!--###############################################################
#-->
<joint name="wheel1_joint" type="continuous">
    <parent link="body_link"/>
    <child link="wheel1_link"/>
    <origin xyz="${-body_link_x_dim/2+1.2*wheel_link_radius}
${-body_link_y_dim/2-wheel_link_length/2} ${wheel_link_z_location}"
rpy="0 0 0"/>
    <axis xyz="0 1 0"/>
    <limit effort="1000" velocity="1000"/>
    <dynamics damping="1.0" friction="1.0"/>
</joint>

<link name ="wheel1_link">
  <visual>
      <origin rpy="1.570795 0 0" xyz="0 0 0"/>
      <geometry>
           <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
      </geometry>
  </visual>

  <collision>
     <origin rpy="1.570795 0 0" xyz="0 0 0"/>
     <geometry>
          <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
     </geometry>
  </collision>
  <xacro:inertia_wheel />
</link>




<!--############################################-->
<!--END: Back right wheel of the robot and the joint-->
<!--####################################################-->

<!--#########################################-->
<!--START: Back left wheel of the robot and the joint-->
<!--####################################-->

<joint name="wheel2_joint" type="continuous">
     <parent link="body_link"/>
     <child link="wheel2_link"/>
     <origin xyz="${-body_link_x_dim/2+1.2*wheel_link_radius}
${body_link_y_dim/2+wheel_link_length/2} ${wheel_link_z_location}"
rpy="0 0 0" />
     <axis xyz="0 1 0"/>
     <limit effort = "1000" velocity="1000"/>
     <dynamics daping="1.0" friction="1.0"/>
</joint>

<link name="wheel2_link">
   <visual>
        <origin rpy="1.570795 0 0" xyz="0 0 0"/>
        <geometry>
            <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
        </geometry>
   </visual>

   <collision>
      <origin rpy="1.570795 0 0" xyz="0 0 0"/>
      <geometry>
          <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
      </geometry>
   </collision>
   <xacro:inertia_wheel />
</link>


<!--###############################################################
##########-->
<!--END: Back left wheel of the robot and the joint-->
<!--###############################################################
###-->
<!--START: Front  right wheel of the robot and the joint-->
<!--################################################-->
<joint name="wheel3_joint" type="continuous">
   <parent link="body_link"/>
   <child link="wheel3_link"/>
   <origin xyz="${body_link_x_dim/2-1.2*wheel_link_radius}
${-body_link_y_dim/2-wheel_link_length/2} ${wheel_link_z_location}"
rpy="0 0 0"/>
   <axis xyz="0 1 0"/>
   <limit effort ="1000" velocity="1000"/>
   <dynamics damping="1.0" friiction="1.0"/>
</joint>

<link name="wheel3_link">
   <visual>
        <origin rpy ="1.570795 0 0" xyz="0 0 0"/>
        <geometry>
            <cylinder length = "${wheel_link_length}"
radius="${wheel_link_radius}"/>
        </geometry>
   </visual>
   <collision>
        <origin rpy="1.570795 0 0" xyz="0 0 0"/>
           <geometry>
              <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
           </geometry>
   </collision>
   <xacro:inertia_wheel />
</link>
<!--###############################################################
###################-->
<!--END: Front right wheel of the robot and the joint-->
<!--###############################################################
#####################-->
<!--START: Front left wheel of the robot and the joint-->
<!--###############################################-->

<joint name="wheel4_joint" type="continuous">
  <parent link="body_link"/>
  <child link="wheel4_link"/>
  <origin xyz="${body_link_x_dim/2-1.2*wheel_link_radius}
${body_link_y_dim/2+wheel_link_length/2} ${wheel_link_z_location}"
rpy="0 0 0"/>
  <axis xyz ="0 1 0 "/>
  <limit effort ="1000" velocity="1000"/>
  <dynamics damping="1.0" friction="1.0"/>
</joint>

<link name="wheel4_link">
  <visual>
      <origin rpy="1.570795 0 0" xyz="0 0 0 "/>
      <geometry>
          <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
       </geometry>
   </visual>

   <collision>
       <origin rpy="1.570795 0 0" xyz="0 0 0"/>
       <geometry>
          <cylinder length="${wheel_link_length}"
radius="${wheel_link_radius}"/>
       </geometry>
   </collision>
   <xacro:inertia_wheel />
</link>
<!--#############################################################--
>
<!--END: Front left wheel of the robot and joint-->
<!--###########################################################-->

</robot>
```