```
<?xml version="1.0"?>
<package format="2">

<launch>

 <!-- Transformation Configuration ... Setting Up the Relationships Between Coordinate Frames -->
 <node pkg="tf" type="static_transform_publisher" name="base_link_to_laser" args="0.06 0 0.08 0 0 0 base_link laser 30" />
 <node pkg="tf" type="static_transform_publisher" name="imu_broadcaster" args="0 0.06 0.02 0 0 0 base_link imu 30" />
 <node pkg="tf" type="static_transform_publisher" name="base_link_broadcaster" args="0 0 0.09 0 0 0 base_footprint base_link 30" />
 <!-- odom to base_footprint transform will be provided by the robot_pose_ekf node -->
 <!-- map to odom will be provided by the AMCL -->
 <node pkg="tf" type="static_transform_publisher" name="map_to_odom" args="0 0 0 0 0 0 map odom 30" />

 <!-- Wheel Encoder Tick Publisher and Base Controller Using Arduino -->
 <!-- motor_controller_diff_drive_2.ino is the Arduino sketch -->
 <!-- Subscribe: /cmd_vel -->
 <!-- Publish: /right_ticks, /left_ticks -->
<node pkg="rosserial_python" type="serial_node.py" name="serial_node">
  <param name="port" value="/dev/ttyACM0"/>
  <param name="baud" value="115200"/>
</node>

 <!-- Wheel Odometry Publisher -->
 <!-- Subscribe: /right_ticks, /left_ticks, /initial_2d -->
 <!-- Publish: /odom_data_euler, /odom_data_quat -->
<node pkg="localization_data_pub" type="ekf_odom_pub" name="ekf_odom_pub">
</node>

 <!-- IMU Data Publisher Using the BNO055 IMU Sensor -->
 <!-- Publish: /imu/data -->
<node ns="imu" name="imu_node" pkg="imu_bno055" type="bno055_i2c_node" respawn="true" respawn_delay="2">
  <param name="device" type="string" value="/dev/i2c-1"/>
  <param name="address" type="int" value="40"/> <!-- 0x28 == 40 is the default for BNO055 -->
  <param name="frame_id" type="string" value="imu"/>
</node>

 <!-- Extended Kalman Filter from robot_pose_ekf Node-->
 <!-- Subscribe: /odom, /imu_data, /vo -->
 <!-- Publish: /robot_pose_ekf/odom_combined -->
 <remap from="odom" to="odom_data_quat" />
 <remap from="imu_data" to="imu/data" />
 <node pkg="robot_pose_ekf" type="robot_pose_ekf" name="robot_pose_ekf">
  <param name="output_frame" value="odom"/>
  <param name="base_footprint_frame" value="base_footprint"/>
  <param name="freq" value="30.0"/>
  <param name="sensor_timeout" value="1.0"/>
  <param name="odom_used" value="true"/>
  <param name="imu_used" value="true"/>
  <param name="vo_used" value="false"/>
  <param name="gps_used" value="false"/>
  <param name="debug" value="false"/>
```

```
  <param name="self_diagnose" value="false"/>
 </node>

 <!-- Initial Pose and Goal Publisher -->
 <!-- Publish: /initialpose, /move_base_simple/goal -->
 <node pkg="rviz" type="rviz" name="rviz" args="-d
/home/automaticaddison/catkin_ws/src/jetson_nano_bot/navigation_data_pub/maps/floorplan4.rviz">
 </node>

 <!-- Subscribe: /initialpose, /move_base_simple/goal -->
 <!-- Publish: /initial_2d, /goal_2d -->
 <node pkg="localization_data_pub" type="rviz_click_to_2d" name="rviz_click_to_2d">
 </node>

 <!-- Lidar Data Publisher Using RPLIDAR from Slamtec -->
 <!-- Used for obstacle avoidance and can be used for mapping -->
 <!-- Publish: /scan -->
<node name="rplidarNode"          pkg="rplidar_ros"  type="rplidarNode" output="screen">
  <param name="serial_port"         type="string" value="/dev/ttyUSB0"/>
  <param name="serial_baudrate"     type="int"    value="115200"/><!--A1/A2 -->
  <!--param name="serial_baudrate"  type="int"    value="256000"--><!--A3 -->
  <param name="frame_id"            type="string" value="laser"/>
  <param name="inverted"            type="bool"   value="false"/>
  <param name="angle_compensate"    type="bool"   value="true"/>
</node>

 <!-- Map File -->
<arg name="map_file" default="$(find navigation_data_pub)/maps/floorplan4.yaml"/>

 <!-- Map Server -->
 <!-- Publish: /map, /map_metadata -->
<node pkg="map_server" name="map_server" type="map_server" args="$(arg map_file)" />

 <!-- Add AMCL example for differential drive robots for Localization -->
 <!-- Subscribe: /scan, /tf, /initialpose, /map -->
  <!-- Publish: /amcl_pose, /particlecloud, /tf -->
  <include file="$(find amcl)/examples/amcl_diff.launch"/>

 <!-- Move Base Node -->
  <!-- Subscribe: /move_base_simple/goal -->
  <!-- Publish: /cmd_vel -->
  <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
   <rosparam file="$(find navstack_pub)/param/costmap_common_params.yaml" command="load" ns="global_costmap" />
   <rosparam file="$(find navstack_pub)/param/costmap_common_params.yaml" command="load" ns="local_costmap" />
   <rosparam file="$(find navstack_pub)/param/local_costmap_params.yaml" command="load" ns="local_costmap" />
   <rosparam file="$(find navstack_pub)/param/global_costmap_params.yaml" command="load" ns="global_costmap" />
   <rosparam file="$(find navstack_pub)/param/base_local_planner_params.yaml" command="load" />
  </node>

</launch>
```