



ROS-1 | ROS-2

1. 1. Instalando ROS-1 (Noetic)

1.1. Link: noetic/Installation/Ubuntu - ROS Wiki

Link: <http://wiki.ros.org/noetic/Installation/Ubuntu>

1.2. Passo 1: Configurar o sources.list para aceitar a instalação.

1.2.1. Comando: `sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'`

Link: <http://packages.ros.org/ros/ubuntu>

1.3. Passo 2: Configurando chave.

1.3.1. Usar APENAS um dos dois comandos a seguir.

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

1.3.3. Comando: `curl -sSL 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B4F42ED6FBAB17C654' | sudo apt-key add -`

1.3.3.1. Use APENAS se não tiver o pacote curl instalado.

1.3.3.2. Comando: `sudo apt install curl`

1.4. Passo 3: Atualizando pacotes do sistema.

1.4.1. Comando: `sudo apt update`

1.5. Passo 4: Instalando o ROS-1.

1.5.1. Usar APENAS um dos comando a seguir.

1.5.2. Instalando ROS Desktop-Full (Instalação Completa).

1.5.2.1. Comando: `sudo apt install ros-noetic-desktop-full`

1.5.3. Instalando ROS-Desktop.

1.5.3.1. Comando: `sudo apt install ros-noetic-desktop`

1.5.4. Instalando ROS-Base.

1.5.4.1. Comando: `sudo apt install ros-noetic-ros-base`

1.6. Passo 5: Instalando dependências para construção de pacotes.

1.6.1. Comando: `sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential`

1.7. Passo 6: Configurando Rosdep.

1.7.1. Passo 6.1: Inicializando Rosdep.

1.7.1.1. Comando: `sudo rosdep init`

1.7.2. Passo 6.2: Atualizando Rosdep.

1.7.2.1. Comando: `rosdep update`

1.8. Passo 7: Configurando o caminho do ROS-1.

1.8.1. Passo 7.1: Abrindo o arquivo '.bashrc'.

1.8.1.1. Comando: `sudo gedit .bashrc`

1.8.2. Passo 7.2: Criando o alias 'initros1'.

1.8.2.1. Caminho: `alias initros1="source /opt/ros/noetic/setup.bash"`

1.8.2.1.1. Após adicionar o alias salve e feche o arquivo '.bashrc'.

1.8.3. Passo 7.3: Atualizando '.bashrc'.

1.8.3.1. Comando: `source .bashrc`

2. * Verificando/Instalando Gazebo (gazebo_ros_pkg) (caso não instalado)

2.1. Link: Gazebo : Tutorial : Installing gazebo_ros_pkgs (ROS 1)

Link: http://gazebosim.org/tutorials?tut=ros_installing&cat=connect_ros

2.2. Passo 1: Formas de verificar se o Gazebo está instalado.

2.2.1. Passo 1.1: Verificar se tem o ícone do Gazebo no menu do Ubuntu, se ele estiver lá o Gazebo estará instalado.

2.2.2. Passo 1.2: Verificar no terminal.

2.2.2.1. Comando: `gazebo --version`

2.2.3. Passo 1.3: Verificar se as pastas '/gzserver' e '/gzclient' estão instaladas.

2.2.3.1. Comando: `which gzserver && which gzclient`

2.2.4. Passo 1.4: Tentar executar o comando para abrir o gazebo.

2.2.4.1. Comando: `gazebo`

2.3. Passo 2: Instalando o Gazebo (gazebo_ros_pkg).

2.3.1. Passo 2.1: Instalando Debians pré construídos.

2.3.1.1. Comando: `sudo apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control`

2.3.2. Passo 2.2: Instalando o Gazebo.

2.3.2.1. Comando: `sudo apt-get install -y libgazebo11-dev`

3. 2. Criando workspace 'catkin_ws'

3.1. Link: [pt_BR/ROS/Tutorials/catkin/BuildingPackages](http://wiki.ros.org/pt_BR/ROS/Tutorials/catkin/BuildingPackages) - ROS Wiki

Link: http://wiki.ros.org/pt_BR/ROS/Tutorials/catkin/BuildingPackages

3.2. Passo 1: Executando o comando para atribuir a versão noetic.

3.2.1. Comando: `initros1`

3.3. Passo 2: Criando pasta que será o pacote para o ROS-1.

3.3.1. Comando: `mkdir -p ~/catkin_ws/src`

3.4. Passo 3: Entrando na pasta '/catkin_ws/src'.

3.4.1. Comando: `cd ~/catkin_ws/src`

3.5. Passo 4: Iniciando workspace.

3.5.1. Comando: `catkin_init_workspace`

3.6. Passo 5: Retornando para a pasta raiz do catkin_ws.

3.6.1. Comando: `cd ~/catkin_ws/`

3.6.2. Comando Alternativo: `cd ..`

3.7. Passo 6: Executando o comando que adiciona utilidades ao fluxo de trabalho do padrão catkin.

3.7.1. Comando: `catkin_make`

3.8. Passo 7: Adicionando o caminho do catkin ao alias 'initros1'.

3.8.1. Passo 7.1: Retornando a pasta raiz do Ubuntu.

3.8.1.1. Comando: `cd`

3.8.2. Passo 7.2: Abrindo o arquivo '.bashrc'.

3.8.2.1. Comando: `sudo gedit .bashrc`

3.8.3. Passo 7.3: Adicionando o seguinte caminho ao alias do 'initros1'.

3.8.3.1. Caminho: `source ~/catkin_ws/devel/setup.bash`

3.8.3.2. Assim ficando com o caminho: alias initros1="source /opt/ros/noetic/setup.bash && source ~/catkin_ws/devel/setup.bash"

3.8.3.2.1. Após adicionar o alias salve e feche o arquivo '.bashrc'.

3.8.4. Passo 7.4: Atualizando '.bashrc'.

3.8.4.1. Comando: source .bashrc

4. * MoveIt

4.1. Link: Getting Started — moveit_tutorials Noetic documentation

Link: https://ros-planning.github.io/moveit_tutorials/doc/getting_started/getting_started.html

4.2. Passo 1: Atualizando pacotes ROS.

4.2.1. Passo 1.1: Atualizando Rosdep.

4.2.1.1. rosdep update

4.2.2. Passo 1.2: Atualizando pacotes do sistema.

4.2.2.1. sudo apt update

4.2.3. Passo 1.3: Atualização geral.

4.2.3.1. sudo apt dist-upgrade

4.3. Passo 2: Instalando o sistema de compilação ROS (catkin).

4.3.1. Comando: `sudo apt install ros-noetic-catkin python3-catkin-tools`

4.4. Passo 3: Instalando o wstool.

4.4.1. Comando: `sudo apt install python3-wstool`

4.5. Passo 4: Criando pasta 'ws_moveit'.

4.5.1. Comando: `mkdir -p ~/ws_moveit/src`

4.6. Passo 5: Entrando na pasta 'ws_moveit'.

4.6.1. Comando: `cd ~/ws_moveit/src`

4.7. Passo 6: Baixando fonte do Moveit.

4.7.1. Passo 6.1: Inicializando wstool.

4.7.1.1. Comando: `wstool init .`

4.7.2. Passo 6.2: Fundindo wstool.

4.7.2.1. Comando: `wstool merge -t . https://raw.githubusercontent.com/ros-planning/moveit/master/moveit.rosinstall`

Link: <https://raw.githubusercontent.com/ros-planning/moveit/master/moveit.rosinstall>

4.7.3. Passo 6.3: Removendo moveit_tutorials de wstool.

4.7.3.1. Comando: `wstool remove moveit_tutorials`

4.7.4. Passo 6.4: Atualizando wstool.

4.7.4.1. Comando: `wstool update -t.`

4.8. Passo 7: Baixar código de exemplo.

4.8.1. Comando: `git clone ros-planning/moveit_tutorials -b master`

Link: https://github.com/ros-planning/moveit_tutorials.git

4.8.2. Comando: `git clone ros-planning/panda_moveit_config -b melodic-devel`

Link: https://github.com/ros-planning/panda_moveit_config.git

4.9. Passo 8: Build catkin workspace.

4.9.1. Comando: `rosdep install -y --from-paths . --ignore-src --rosdistro noetic`

4.9.2. Comando: `sudo sh -c 'echo "deb http://packages.ros.org/ros-testing/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'`

Link: <http://packages.ros.org/ros-testing/ubuntu>

4.9.3. Comando: `sudo apt update`

4.9.4. Comando: `cd ~/ws_moveit`

4.9.5. Comando: `catkin config --extend /opt/ros/noetic --cmake-args -DCMAKE_BUILD_TYPE=Release`

4.9.6. Comando: `catkin build`

5. 4. Criando workspace 'dev_ws'

5.1. Passo 1: Executando o comando para atribuir a versão foxy.

5.1.1. Comando: `initros2`

5.2. Passo 2: Instalando compilador Colcon.

5.2.1. Comando: `sudo apt install python3-colcon-common-extensions`

5.3. Passo 3: Criando pasta que será o pacote ROS-2.

5.3.1. Comando: `mkdir -p ~/dev_ws/src`

5.4. Passo 4: Entrando na pasta '/dev_ws/'.

5.4.1. Comando: `cd ~/dev_ws`

5.5. Passo 5: Verificando dependências.

5.5.1. Comando: `rosdep install -i --from-path src --rosdistro foxy -y`

5.6. Passo 6: Criando espaço de trabalho.

5.6.1. Comando: `colcon build --symlink-install`

5.7. Passo 7: Compilando.

5.7.1. Comando: `colcon build`

5.8. Passo 8: Adicionando caminho da pasta dev_ws ao alias 'initros2'.

5.8.1. Passo 8.1: Retornando a pasta raiz do Ubuntu.

5.8.1.1. Comando: `cd`

5.8.2. Passo 8.2: Abrindo o arquivo '.bashrc'.

5.8.2.1. Comando: `sudo gedit .bashrc`

5.8.3. Passo 8.3: Adicionando o seguinte caminho ao alias do 'initros2'.

5.8.3.1. Caminho: `source ~/dev_ws/install/local_setup.bash`

5.8.3.2. Assim ficando com o caminho: `alias initros2="source /opt/ros/ros2-linux/local_setup.bash && source ~/dev_ws/install/local_setup.bash"`

5.8.3.2.1. Após adicionar o alias salve e feche o arquivo '.bashrc'.

5.8.4. Passo 8.4: Atualizando .bashrc.

5.8.4.1. Comando: `source .bashrc`

6. 3. Instalando ROS-2 (Foxy)

6.1. Link: Installing ROS 2 on Linux

Link: <https://index.ros.org/doc/ros2/Installation/Foxy/Linux-Install-Binary/>

6.2. Passo 1: Instando pacotes necessários (curl, gnupg2, lsb-release).

6.2.1. Comando: `sudo apt update && sudo apt install curl gnupg2 lsb-release`

6.3. Passo 2: Configurando chave.

6.3.1. Comando: `curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -`

6.4. Passo 3: Adicionando repositório a lista de fontes.

6.4.1. Comando: `sudo sh -c 'echo "deb [arch=$(dpkg --print-architecture)]
http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/ros2-latest.list'`

Link: <http://packages.ros.org/ros2/ubuntu>

6.5. Passo 4: Download ROS-2 (Foxy), baixar a versão compatível com o seu sistema operacional.

6.5.1. Link: <https://github.com/ros2/ros2/releases/tag/release-foxy-20201211>

Link: <https://github.com/ros2/ros2/releases/tag/release-foxy-20201211>

6.6. Passo 5: Extraíndo instalador.

6.6.1. Antes de extrair, mover o arquivo compactado para '/opt/ros'.

6.6.1.1. Comando: `sudo mv nome_do_arquivo /caminho_da_pasta`

6.6.1.2. Exemplo: `sudo mv ros2-foxy-20201211-linux-focal-amd64.tar.bz2 /opt/ros`

6.6.2. Comando: `sudo tar xf nome_do_arquivo`

6.6.3. Exemplo: `sudo tar xf ros2-foxy-20201211-linux-focal-amd64.tar.bz2`

6.7. Passo 6: Configurando o caminho do ROS-2.

6.7.1. Passo 6.1: Abrindo o arquivo '.bashrc'.

6.7.1.1. Comando: `sudo gedit .bashrc`

6.7.2. Passo 6.2: Criando alias 'initros2'.

6.7.2.1. Caminho: alias initros2="source /opt/ros/ros2-linux/local_setup.bash"

6.7.2.1.1. Após adicionar o alias salve e feche o arquivo '.bashrc'.

6.7.3. Passo 6.3: Atualizando '.bashrc'.

6.7.3.1. Comando: source .bashrc

6.8. Passo 7: Atualizando pacotes.

6.8.1. Comando: sudo apt update

6.9. Passo 8: Instalando pacote 'python3-rosdep'.

6.9.1. Comando: sudo apt install -y python3-rosdep

6.10. Passo 9: Configurando rosdep.

6.10.1. Passo 9.1: Inicializando Rosdep.

6.10.1.1. Comando para iniciar: sudo rosdep init

6.10.2. Passo 9.2: Atualizando Rosdep.

6.10.2.1. Comando para atualizar: rosdep update

6.11. Passo 10: Instalando dependências ausentes.

6.11.1. AVISO: O caminho '/opt/ros/ros2-linux/share' varia dependendo de onde se põe o instalador do ROS-2.

6.11.2. Comando: `rosdep install --from-paths /opt/ros/ros2-linux/share --ignore-src --rosdistro foxy -y --skip-keys "console_bridge fastcdr fastrtps osrf_testing_tools_cpp poco_vendor rmw_connext_cpp rosidl_typesupport_connext_c rosidl_typesupport_connext_cpp rti-connext-dds-5.3.1 tinyclang_vendor tinyclang2_vendor urdfdom urdfdom_headers"`

6.12. Passo 11: Instalando bibliotecas do python3.

6.12.1. Comando: `sudo apt install -y libpython3-dev python3-pip`

6.12.2. Comando: `pip3 install -U argcomplete`