Install and start the Microros agent

Note: You can select any one of the three startup methods.

1. Docker starts the microros agent

Docker starts the serial port proxy

```
docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble serial --dev /dev/myserial -b 2000000 -v4
```

Among them, --dev /dev/myserial is the bound serial port device, which can also be changed to /dev/ttyUSB0, etc.

-b 2000000 is the baud rate.

To exit the proxy, press Ctrl+C in the terminal.

Note that you cannot close the terminal directly, otherwise Docker will continue to run in the background.

If the MCU starts the reconnection proxy multiple times, causing ROS2 to search for multiple identical nodes, it will not actually affect the use. Press Ctrl+C to end the proxy and then reset the MCU to reconnect to the proxy.

Agent startup failure

The microROS agent can only be started in one terminal. If a terminal has already started the microROS agent in the background, an error will be reported when starting the agent again. Please press Ctrl+C to exit the agent in the original agent terminal before running the agent again.

If the next time you start the agent, it fails because you directly closed the terminal, you can restart the virtual machine/computer to solve it, or manually end Docker to solve it.

How to manually end docker:

Please first query the current Docker process number and end the current proxy Docker process.

```
docker ps -a | grep microros/micro-ros-agent
docker stop xxxxxxxxxx
```

2. Start microros agent from source code

Install tinyxml2 dependencies

Enter the following command in the terminal to install tinyxml2

```
cd ~/
git clone https://github.com/leethomason/tinyxml2.git
cd tinyxml2
mkdir build
cd build
sudo cmake ..
sudo make
sudo make install
```

Install the python3-rosdep tool

Enter the following command in the terminal to install the rosdep tool. If it has already been installed, you can skip this step.

```
sudo apt install python3-rosdep
```

Compile the micro_ros_setup environment

Activate the ROS2 environment variables. Here we take the humble version as an example. If it has already been activated, you can skip the activation step.

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

Create and enter the workspace uros_ws in the user directory

```
mkdir ~/uros_ws && cd ~/uros_ws
mkdir src
```

Download the micro_ros_setup file to the src folder

```
git clone -b $ROS_DISTRO https://github.com/micro-ROS/micro_ros_setup.git
src/micro_ros_setup
```

Initialize rosdep

```
sudo rosdep init
```

If there is a network problem, please add the -E parameter

```
sudo -E rosdep init
```

```
:~/uros_ws$ sudo rosdep init
ERROR: cannot download default sources list from:
https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/sources.list.d/20-
default.list
Website may be down.
:~/uros_ws$ sudo -E rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update
```

If all the above steps return an error and rosdep still cannot be initialized, you can create a new 20-default.list file in the /etc/ros/rosdep/sources.list.d/ directory and add the following content, then proceed to the next step.

```
# os-specific listings first
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-
homebrew.yaml osx

# generic
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
gbpdistro
https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
fuerte

# newer distributions (Groovy, Hydro, ...) must not be listed anymore, they are
being fetched from the rosdistro index.yaml instead
```

Update rosdep and install related driver packages

```
rosdep update && rosdep install --from-paths src --ignore-src -y
```

```
:~/uros_ws$ rosdep update && rosdep install --from-paths src --igno
re-src -y
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
```

Compile Workspace

```
colcon build
```

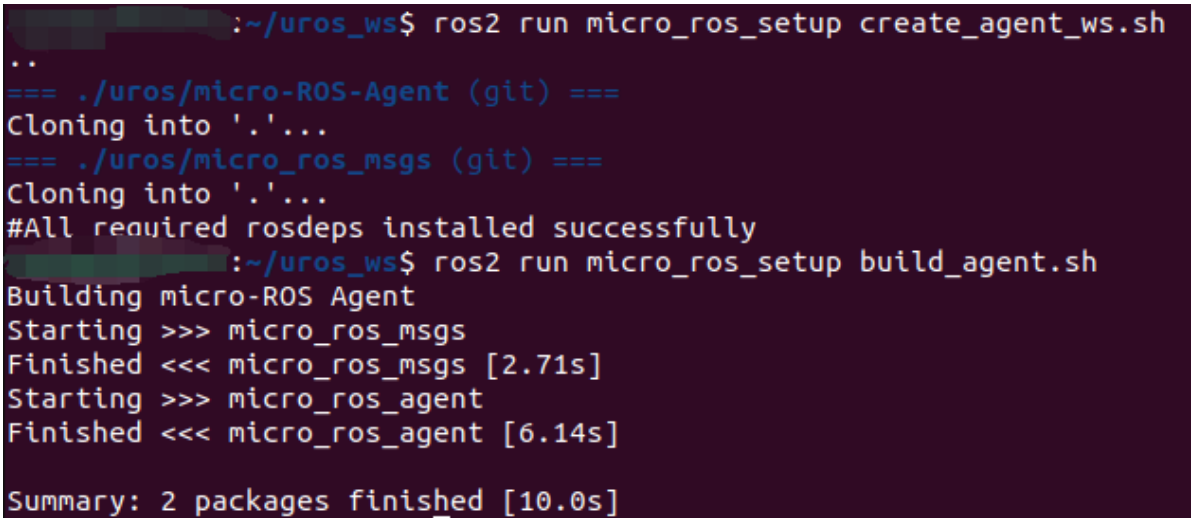
```
:~/uros_ws$ colcon build
Starting >>> micro_ros_setup
Finished <<< micro_ros_setup [1.17s]

Summary: 1 package finished [2.23s]
:~/uros_ws$ source install/local_setup.bash
```

Activate the micro_ros_setup environment

```
source install/local_setup.bash
```

Compile the micro_ros_agent environment

```
ros2 run micro_ros_setup create_agent_ws.sh
ros2 run micro_ros_setup build_agent.sh
```



If an error occurs when executing build_agent.sh, compile again.

Start the microros serial port proxy from source code

Activate the micro_ros_agent agent environment

```
source ~/uros_ws/install/local_setup.sh
```

Enter the following command to start the serial port proxy from the ROS2 environment

```
ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/myserial -b  2000000
-v4
```

Among them, --dev /dev/myserial is the bound serial port device, which can also be changed to /dev/ttyUSB0, etc.

-b 2000000 is the baud rate.

To exit the proxy, press Ctrl+C in the terminal.

3. Start the microros agent with the factory image

The factory image system has integrated the proxy function and made it into a script. You only need to run the following command to enable the proxy.

```
sh ~/start_agent.sh
```