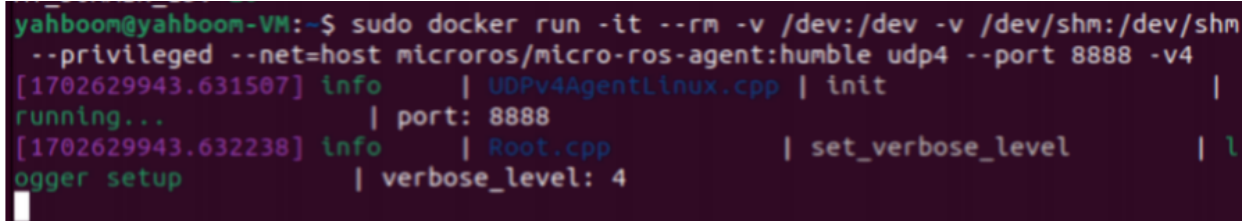


Enter the robot's docker container

Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be consistent. You can check [Must read before use] to set the IP and ROS_DOMAIN_ID on the board.

Taking the supporting virtual machine as an example, enter the following command to enter the docker container:

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host  
microros/micro-ros-agent:humble udp4 --port 8888 -v4
```



```
yahboom@yahboom-VM:~$ sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm  
--privileged --net=host microros/micro-ros-agent:humble udp4 --port 8888 -v4  
[1702629943.631507] info      | UDPv4AgentLinux.cpp | init      |  
running...                | port: 8888          |  
[1702629943.632238] info      | Root.cpp            | set_verbose_level | l  
ogger setup                | verbose_level: 4    |
```

After starting the container, the agent will be turned on, turn on the car switch, and wait for the car to connect to the agent. The connection is successful, as shown in the figure below.

```

[1702630014.015846] info | ProxyClient.cpp | create_participant | participant created | client_key: 0x0862A009, participant_id: 0x000(1)
[1702630014.135363] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x000(2), participant_id: 0x000(1)
[1702630014.223609] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0862A009, publisher_id: 0x000(3), participant_id: 0x000(1)
[1702630014.415510] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0862A009, datawriter_id: 0x000(5), publisher_id: 0x000(3)
[1702630014.428530] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x001(2), participant_id: 0x000(1)
[1702630014.527190] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0862A009, publisher_id: 0x001(3), participant_id: 0x000(1)
[1702630014.543889] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0862A009, datawriter_id: 0x001(5), publisher_id: 0x001(3)
[1702630014.554490] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x002(2), participant_id: 0x000(1)
[1702630014.737059] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0862A009, publisher_id: 0x002(3), participant_id: 0x000(1)
[1702630014.755072] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0862A009, datawriter_id: 0x002(5), publisher_id: 0x002(3)
[1702630014.818985] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x003(2), participant_id: 0x000(1)
[1702630014.840001] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0862A009, subscriber_id: 0x000(4), participant_id: 0x000(1)
[1702630014.864010] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0862A009, datareader_id: 0x000(6), subscriber_id: 0x000(4)
[1702630014.959908] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x004(2), participant_id: 0x000(1)
[1702630015.033537] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0862A009, subscriber_id: 0x001(4), participant_id: 0x000(1)
[1702630015.140350] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0862A009, datareader_id: 0x001(6), subscriber_id: 0x001(4)
[1702630015.158510] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0862A009, topic_id: 0x005(2), participant_id: 0x000(1)
[1702630015.241039] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0862A009, subscriber_id: 0x002(4), participant_id: 0x000(1)
[1702630015.347393] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0862A009, datareader_id: 0x002(6), subscriber_id: 0x002(4)

```

After waiting for the car to connect, a node named /YB_Car_Node will be started. In the terminal of the supporting virtual machine, enter the following command to query:

```
ros2 node list
```

```

yahboom@yahboom-VM:~$ ros2 node list
/YB_Car_Node

```