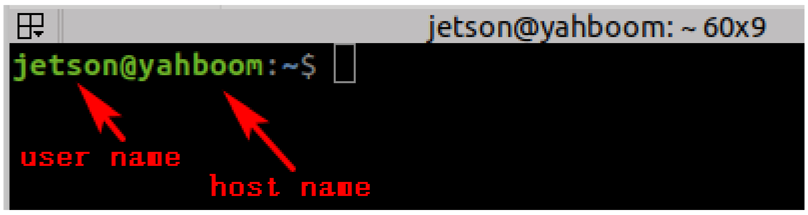
4. Multi-machine communication configuration

Note: Before logging in remotely, you must know the IP of the robot, which can be displayed on an external monitor or OLED.

For example, the following figure: Username [jetson], hostname [yahboom].



```
jetson@yahboom: ~ 60x9
jetson@yahboom:~$
```

The operating environment and software and hardware reference configurations are as follows:

- Reference model： ROSMASTER X3
- Robot hardware configuration: Arm series master, Slan A1 LiDAR, AstraPro Plus depth camera
- Robot system: Ubuntu (version not required) + docker (version 20.10.21 and above)
- PC VM: Ubuntu (20.04) + ROS2 (Foxy)
- Usage scenario: Used on a relatively clean 2D plane

4.1. concept

Multi-computer communication, or distributed communication, is a communication strategy that can realize data interaction between different hosts through the network.

ROS2 itself is a distributed communication framework, which can conveniently realize the communication between different devices. The middleware based on ROS2 is DDS. When in the same network, distributed communication can be realized through the ROS DOMAIN ID mechanism of DDS. The general process is as follows: Before starting the node, you can set the value of the domain ID. If the domain ID of different nodes is the same, they can discover and communicate freely. Otherwise, if the domain ID value is different, it cannot be realized. By default, all nodes are started with a domain ID of 0, in other words, as long as you ensure that you are on the same network, you do not need to do any configuration, different nodes on different ROS2 devices can achieve distributed communication.

The application scenarios of distributed communication are more extensive, such as unmanned vehicle formation, drone formation, remote control, etc. The interaction of these data all rely on distributed communication.

4.2. achieve

4.2.1. Default implementation

You only need to put the host and slave (can have multiple)in the same network, you have achieved distributed communication. For example, the host and slave are connected to the same WiFi or the same router.

In Windows, if the network of the VM is set to Bridge mode, the VM and the host are on the same network

TEST:

1. Host side [robot] execution:

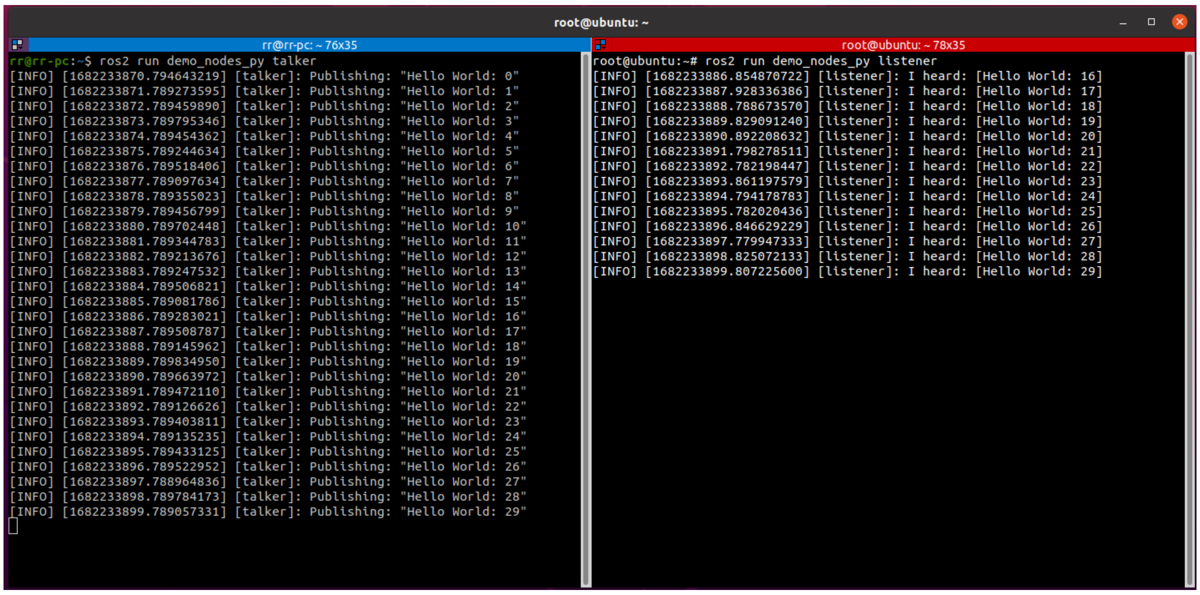
What is demonstrated here is that the robot is in docker, and the network mode used by docker is host mode, which simply means that the host mode shares a network with the robot, so it is no different from the execution on the robot.

```
ros2 run demo_nodes_py talker
```

2. This command is executed [secondary VM]:

```
ros2 run demo_nodes_py listener
```

If the following information is displayed: A topic published on the host can be subscribed to from the host in time, multi-host communication has been implemented



```
rr@rr-pc:~$ ros2 run demo_nodes_py talker
[INFO] [1682233870.794643219] [talker]: Publishing: "Hello World: 0"
[INFO] [1682233871.789273595] [talker]: Publishing: "Hello World: 1"
[INFO] [1682233872.789459890] [talker]: Publishing: "Hello World: 2"
[INFO] [1682233873.789795346] [talker]: Publishing: "Hello World: 3"
[INFO] [1682233874.789454362] [talker]: Publishing: "Hello World: 4"
[INFO] [1682233875.789244634] [talker]: Publishing: "Hello World: 5"
[INFO] [1682233876.789518406] [talker]: Publishing: "Hello World: 6"
[INFO] [1682233877.789097634] [talker]: Publishing: "Hello World: 7"
[INFO] [1682233878.789355023] [talker]: Publishing: "Hello World: 8"
[INFO] [1682233879.789456799] [talker]: Publishing: "Hello World: 9"
[INFO] [1682233880.789702448] [talker]: Publishing: "Hello World: 10"
[INFO] [1682233881.789344783] [talker]: Publishing: "Hello World: 11"
[INFO] [1682233882.789213676] [talker]: Publishing: "Hello World: 12"
[INFO] [1682233883.789247532] [talker]: Publishing: "Hello World: 13"
[INFO] [1682233884.789506821] [talker]: Publishing: "Hello World: 14"
[INFO] [1682233885.789081786] [talker]: Publishing: "Hello World: 15"
[INFO] [1682233886.789283021] [talker]: Publishing: "Hello World: 16"
[INFO] [1682233887.789508787] [talker]: Publishing: "Hello World: 17"
[INFO] [1682233888.789145962] [talker]: Publishing: "Hello World: 18"
[INFO] [1682233889.789834950] [talker]: Publishing: "Hello World: 19"
[INFO] [1682233890.789663972] [talker]: Publishing: "Hello World: 20"
[INFO] [1682233891.789472110] [talker]: Publishing: "Hello World: 21"
[INFO] [1682233892.789126626] [talker]: Publishing: "Hello World: 22"
[INFO] [1682233893.789403811] [talker]: Publishing: "Hello World: 23"
[INFO] [1682233894.789135235] [talker]: Publishing: "Hello World: 24"
[INFO] [1682233895.789433125] [talker]: Publishing: "Hello World: 25"
[INFO] [1682233896.789522952] [talker]: Publishing: "Hello World: 26"
[INFO] [1682233897.788964836] [talker]: Publishing: "Hello World: 27"
[INFO] [1682233898.789784173] [talker]: Publishing: "Hello World: 28"
[INFO] [1682233899.789057331] [talker]: Publishing: "Hello World: 29"
```

```
root@ubuntu:~# ros2 run demo_nodes_py listener
[INFO] [1682233886.854870722] [listener]: I heard: [Hello World: 16]
[INFO] [1682233887.928336386] [listener]: I heard: [Hello World: 17]
[INFO] [1682233888.788673570] [listener]: I heard: [Hello World: 18]
[INFO] [1682233889.829091240] [listener]: I heard: [Hello World: 19]
[INFO] [1682233890.892208632] [listener]: I heard: [Hello World: 20]
[INFO] [1682233891.798278511] [listener]: I heard: [Hello World: 21]
[INFO] [1682233892.782198447] [listener]: I heard: [Hello World: 22]
[INFO] [1682233893.861197579] [listener]: I heard: [Hello World: 23]
[INFO] [1682233894.794178783] [listener]: I heard: [Hello World: 24]
[INFO] [1682233895.782020436] [listener]: I heard: [Hello World: 25]
[INFO] [1682233896.846629229] [listener]: I heard: [Hello World: 26]
[INFO] [1682233897.779947333] [listener]: I heard: [Hello World: 27]
[INFO] [1682233898.825072133] [listener]: I heard: [Hello World: 28]
[INFO] [1682233899.807225600] [listener]: I heard: [Hello World: 29]
```

4.2.2. Distributed network packet

If you are in a network with other robots in use, you can also set up a group for your robots in order not to be disturbed by other robots.

ROS2 provides a DOMAIN mechanism, just like grouping, computers in the same DOMAIN can communicate. We can add such a configuration in the .bashrc of the host side [robot] and the slave side [VM] to assign the two to a group:

```
$ export ROS_DOMAIN_ID=<your_domain_id>
```

If the ID assigned by the host [car] and the slave [virtual machine] is different, the two cannot communicate and achieve the purpose of grouping.

TEST:

1. Host side [robot] execution:

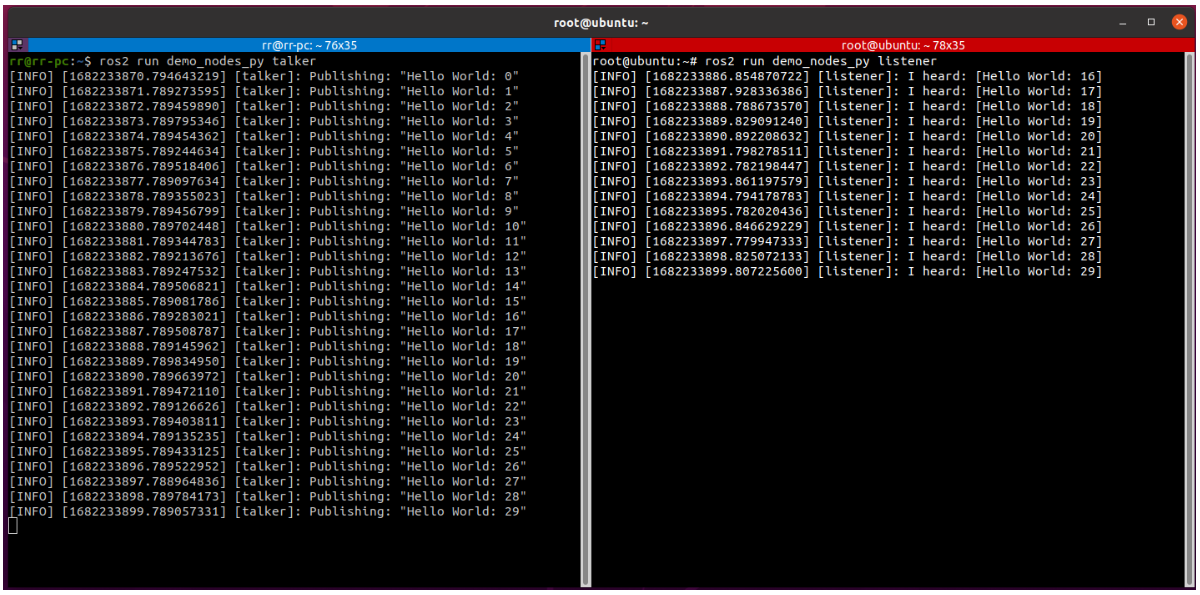
What is demonstrated here is that the robot is in docker, and the network mode used by docker is host mode, which simply means that the host mode shares a network with the robot, so it is no different from the execution on the robot.

```
ros2 run demo_nodes_py talker
```

2. This command is executed [secondary VM]:

```
ros2 run demo_nodes_py listener
```

If the following information is displayed: A topic published on the host can be subscribed to from the host in time, multi-host communication has been implemented


root@ubuntu: ~

rr@rr-pc: ~ 76x35

```
rr@rr-pc:~$ ros2 run demo_nodes_py talker
[INFO] [1682233870.794643219] [talker]: Publishing: "Hello World: 0"
[INFO] [1682233871.789273595] [talker]: Publishing: "Hello World: 1"
[INFO] [1682233872.789459890] [talker]: Publishing: "Hello World: 2"
[INFO] [1682233873.789795346] [talker]: Publishing: "Hello World: 3"
[INFO] [1682233874.789454362] [talker]: Publishing: "Hello World: 4"
[INFO] [1682233875.789244634] [talker]: Publishing: "Hello World: 5"
[INFO] [1682233876.789518406] [talker]: Publishing: "Hello World: 6"
[INFO] [1682233877.789097634] [talker]: Publishing: "Hello World: 7"
[INFO] [1682233878.789355023] [talker]: Publishing: "Hello World: 8"
[INFO] [1682233879.789456799] [talker]: Publishing: "Hello World: 9"
[INFO] [1682233880.789702448] [talker]: Publishing: "Hello World: 10"
[INFO] [1682233881.789344783] [talker]: Publishing: "Hello World: 11"
[INFO] [1682233882.789213676] [talker]: Publishing: "Hello World: 12"
[INFO] [1682233883.789247532] [talker]: Publishing: "Hello World: 13"
[INFO] [1682233884.789506821] [talker]: Publishing: "Hello World: 14"
[INFO] [1682233885.789081786] [talker]: Publishing: "Hello World: 15"
[INFO] [1682233886.789283021] [talker]: Publishing: "Hello World: 16"
[INFO] [1682233887.789508787] [talker]: Publishing: "Hello World: 17"
[INFO] [1682233888.789145962] [talker]: Publishing: "Hello World: 18"
[INFO] [1682233889.789834950] [talker]: Publishing: "Hello World: 19"
[INFO] [1682233890.789663972] [talker]: Publishing: "Hello World: 20"
[INFO] [1682233891.789472110] [talker]: Publishing: "Hello World: 21"
[INFO] [1682233892.789126626] [talker]: Publishing: "Hello World: 22"
[INFO] [1682233893.789403811] [talker]: Publishing: "Hello World: 23"
[INFO] [1682233894.789135235] [talker]: Publishing: "Hello World: 24"
[INFO] [1682233895.789433125] [talker]: Publishing: "Hello World: 25"
[INFO] [1682233896.789522952] [talker]: Publishing: "Hello World: 26"
[INFO] [1682233897.788964836] [talker]: Publishing: "Hello World: 27"
[INFO] [1682233898.789784173] [talker]: Publishing: "Hello World: 28"
[INFO] [1682233899.789057331] [talker]: Publishing: "Hello World: 29"
```

root@ubuntu: ~ 78x35

```
root@ubuntu:~# ros2 run demo_nodes_py listener
[INFO] [1682233886.854870722] [listener]: I heard: [Hello World: 16]
[INFO] [1682233887.928336386] [listener]: I heard: [Hello World: 17]
[INFO] [1682233888.788673570] [listener]: I heard: [Hello World: 18]
[INFO] [1682233889.829091240] [listener]: I heard: [Hello World: 19]
[INFO] [1682233890.892208632] [listener]: I heard: [Hello World: 20]
[INFO] [1682233891.798278511] [listener]: I heard: [Hello World: 21]
[INFO] [1682233892.782198447] [listener]: I heard: [Hello World: 22]
[INFO] [1682233893.861197579] [listener]: I heard: [Hello World: 23]
[INFO] [1682233894.794178783] [listener]: I heard: [Hello World: 24]
[INFO] [1682233895.782020436] [listener]: I heard: [Hello World: 25]
[INFO] [1682233896.846629229] [listener]: I heard: [Hello World: 26]
[INFO] [1682233897.779947333] [listener]: I heard: [Hello World: 27]
[INFO] [1682233898.825072133] [listener]: I heard: [Hello World: 28]
[INFO] [1682233899.807225600] [listener]: I heard: [Hello World: 29]
```

4.3. Attention

Setting the ROS DOMAIN ID value is not arbitrary, and there are certain constraints:

1. It is recommended that the ROS DOMAIN ID be between 0 and 101, including 0 and 101.
2. The total number of nodes in each domain ID is limited and must be less than or equal to 120.
3. If the domain ID is 101, the total number of nodes in the domain must be less than or equal to 54.

4.4. DDS Rules for calculating domain ids

The rules for calculating domain ID values are as follows:

1. DDS is based on the TCP/IP or UDP/IP network communication protocol. During network communication, you need to specify a port number. The port number is represented by an unsigned integer of 2 bytes, whose value ranges from 0 to 65535.
2. Port number allocation also has its rules, not can be used arbitrarily, according to the DDS protocol to 7400 as the starting port, you can also use the port is [7400,65535], and it is known that according to the DDS protocol by default, each domain ID occupies 250 ports, then the number of domain ids is: (65,535-7400)/250 = 232(each), the corresponding value range is [0,231].
3. The operating system also sets some reserved ports, and when using ports in DDS, you need to avoid these reserved ports, so as to avoid conflicts in use. Different operating systems have different reserved ports, and the final result is that under Linux, the available domain ids are [0,101] and [215-231]. The available domain ID on Windows and Mac is [0,166]. To be compatible with multiple platforms, it is recommended that the domain ID be in the range of [0,101].
4. By default, each domain ID occupies 250 ports, and each ROS2 node needs to occupy two ports. In addition, according to the DDS protocol, the first and second ports in the port segment of each domain ID are Discovery Multicast ports and User Multicast ports. Start with ports 11 and 12, which is the first node Discovery Unicast port and User Unicast port, Ports occupied by subsequent nodes are extended successively. The maximum number of nodes in a domain ID is : (250-10)/2 = 120.
5. Special case: If the domain ID is 101, the latter half of the domain port is reserved for the operating system, and the maximum number of nodes is 54.

Notes : The above calculation rules can be just simple understood.