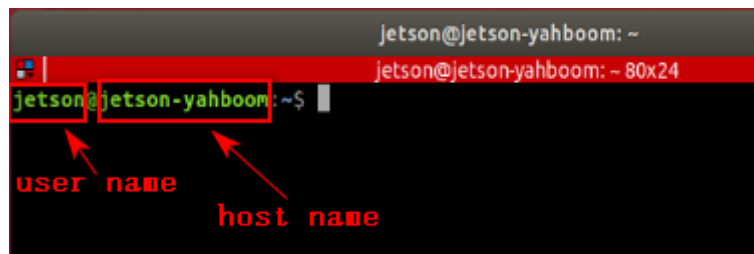


4、 Multi-machine communication

- 4、 Multi-machine communication
 - 4.1、 Login remotely
 - 4.1.1、 SSH
 - 4.1.2、 jupyter lab
 - 4.1.3、 VNC
 - 4.2、 ROS multi-machine communication
 - 4.2.1、 Universal version
 - 4.2.2、 Smiple version
 - 4.3、 Webpage real-time monitoring

Note: You must know the IP of the robot before remote login. You can view the current IP address of the robot through an external display or OLED

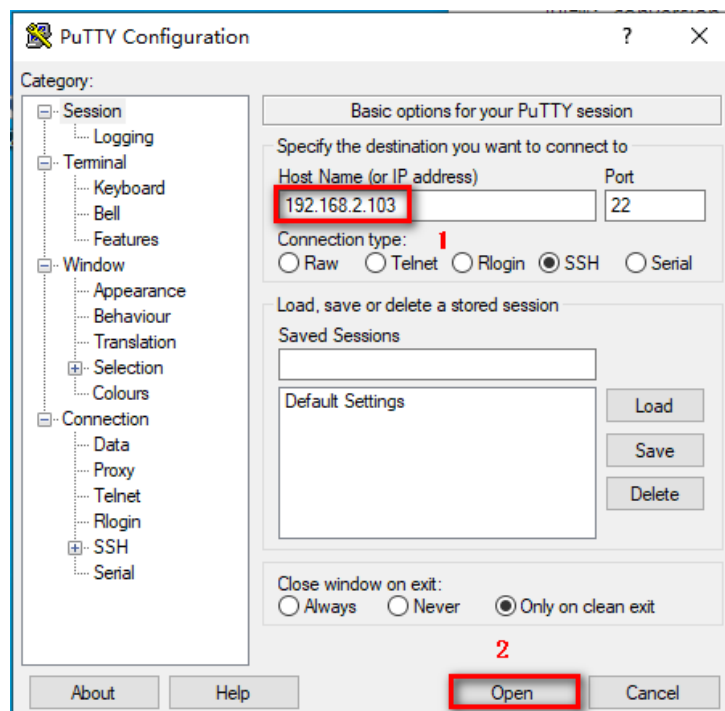


4.1、 Login remotely

4.1.1、 SSH

Note: The system graphical interface cannot be displayed in this way.

- PuTTY Login



Input IP address, then click 【open】 ;

```
jetson@jetson-yahboom: ~  
login as: jetson  
jetson@192.168.2.103's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.9.201-tegra aarch64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
0 updates can be applied immediately.  
  
*** System restart required ***  
Last login: Tue Aug 31 15:17:45 2021 from 192.168.2.90
```

Input user name and password.

For Yahoom image, user name is **jetson** password is **yahboom**

- ubuntu system

```
jetson@jetson-yahboom: ~  
jetson@jetson-yahboom: ~ 80x24  
yahboom@Yahboom:~$ ssh jetson@192.168.2.103  
The authenticity of host '192.168.2.103 (192.168.2.103)' can't be established.  
ECDSA key fingerprint is SHA256:9UYkwisqeLwDgJTC+dzH//Ey8AtRTa4jgzBsMbXChE.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.2.103' (ECDSA) to the list of known hosts.  
jetson@192.168.2.103's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.9.201-tegra aarch64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
0 updates can be applied immediately.  
  
*** System restart required ***  
Last login: Tue Aug 31 15:33:40 2021 from 192.168.2.106
```

1) Input following command in terminal

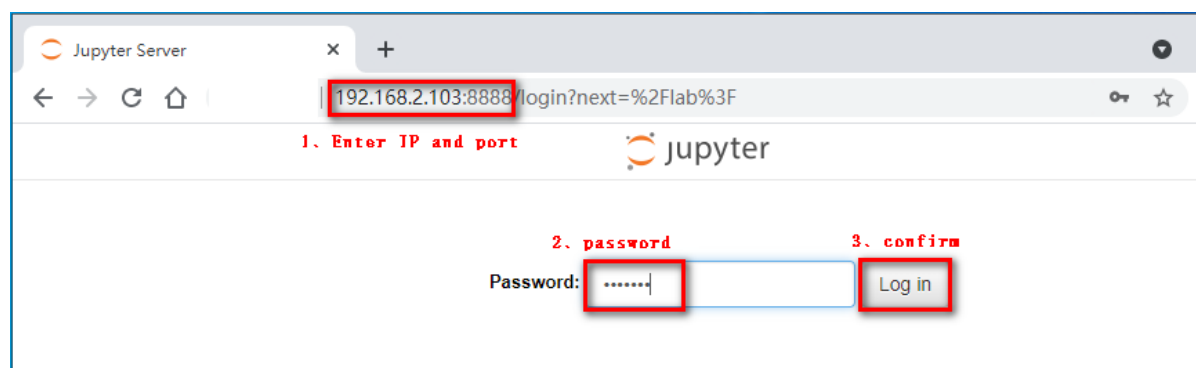
```
ssh jetson@192.168.2.103
```

2) Then, input **yes**

3) Next, input **yahboom**

4.1.2. jupyter lab

Note: The system graphical interface cannot be displayed in this way.



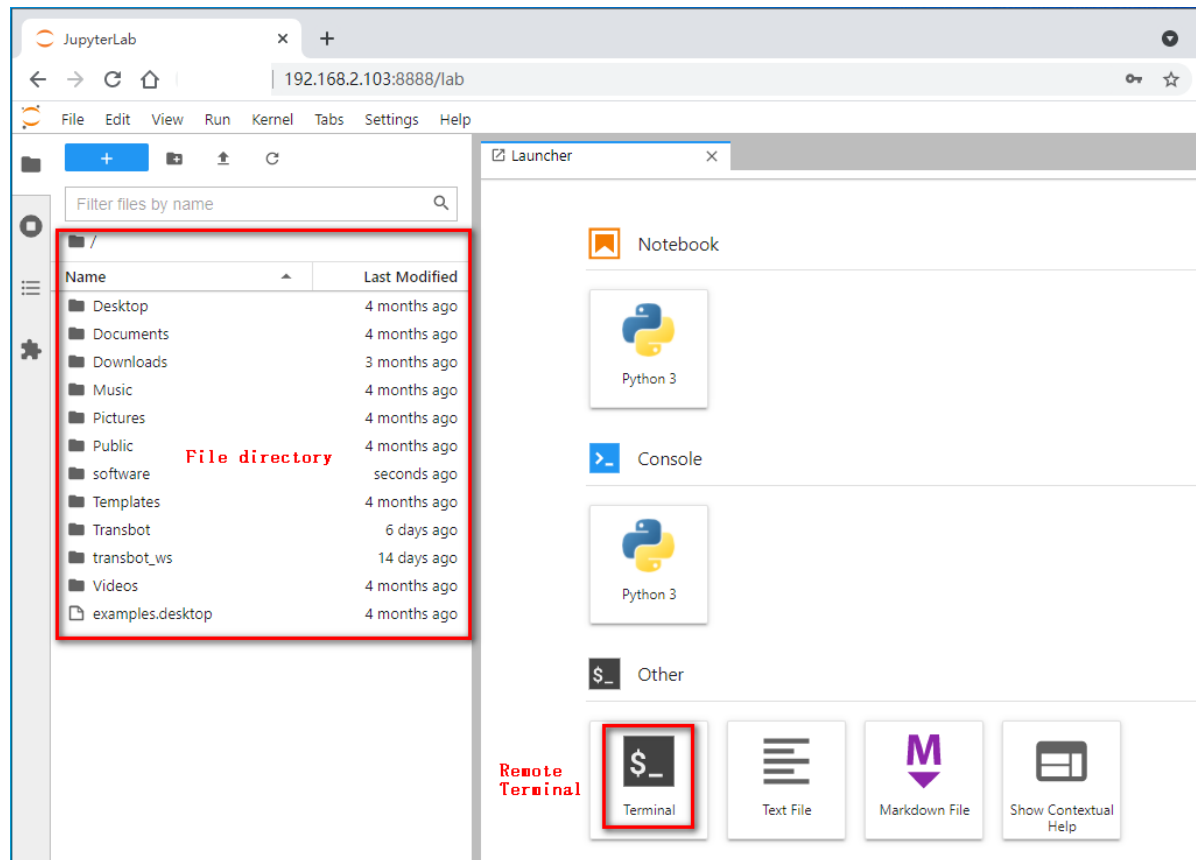
Input following URL on browser, pree 【Enter】

Then, input password **yahboom**

Click 【Log in】

`http://192.168.2.103:8888`

After login, you can see following interface.

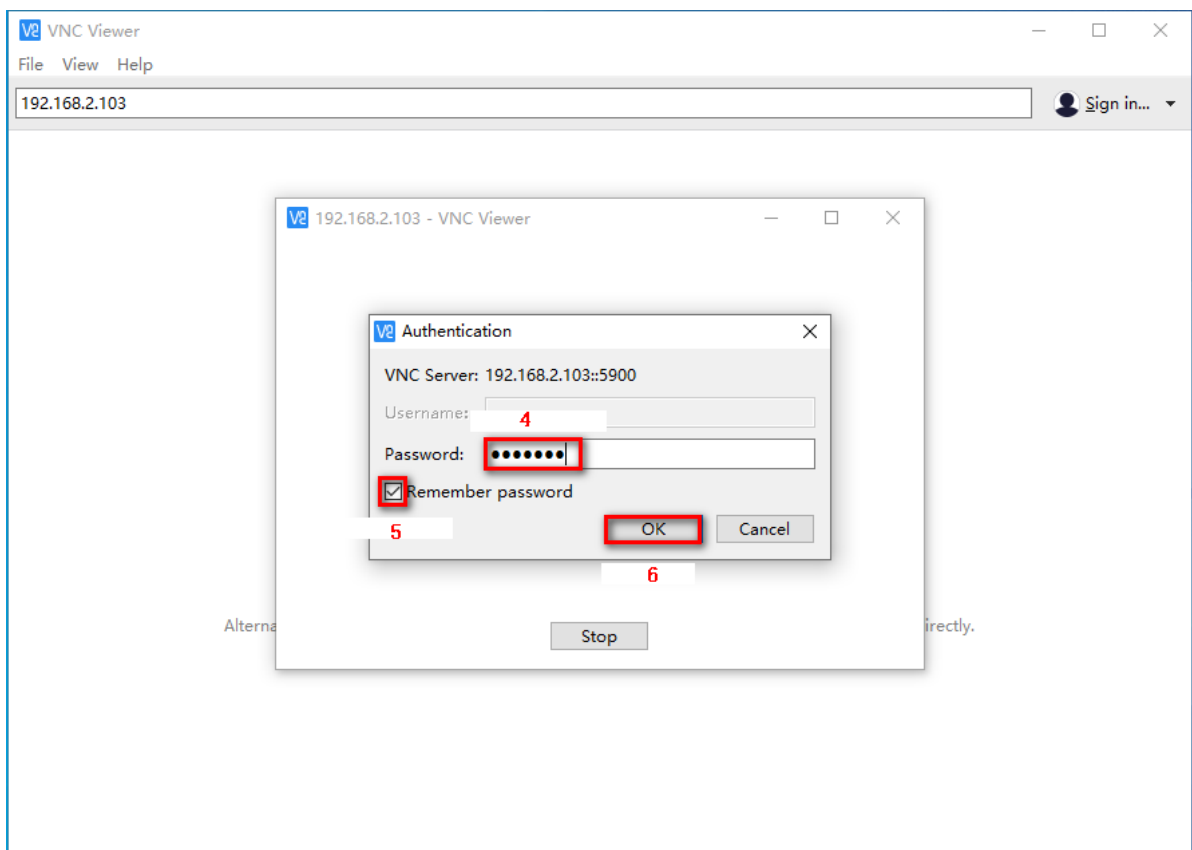
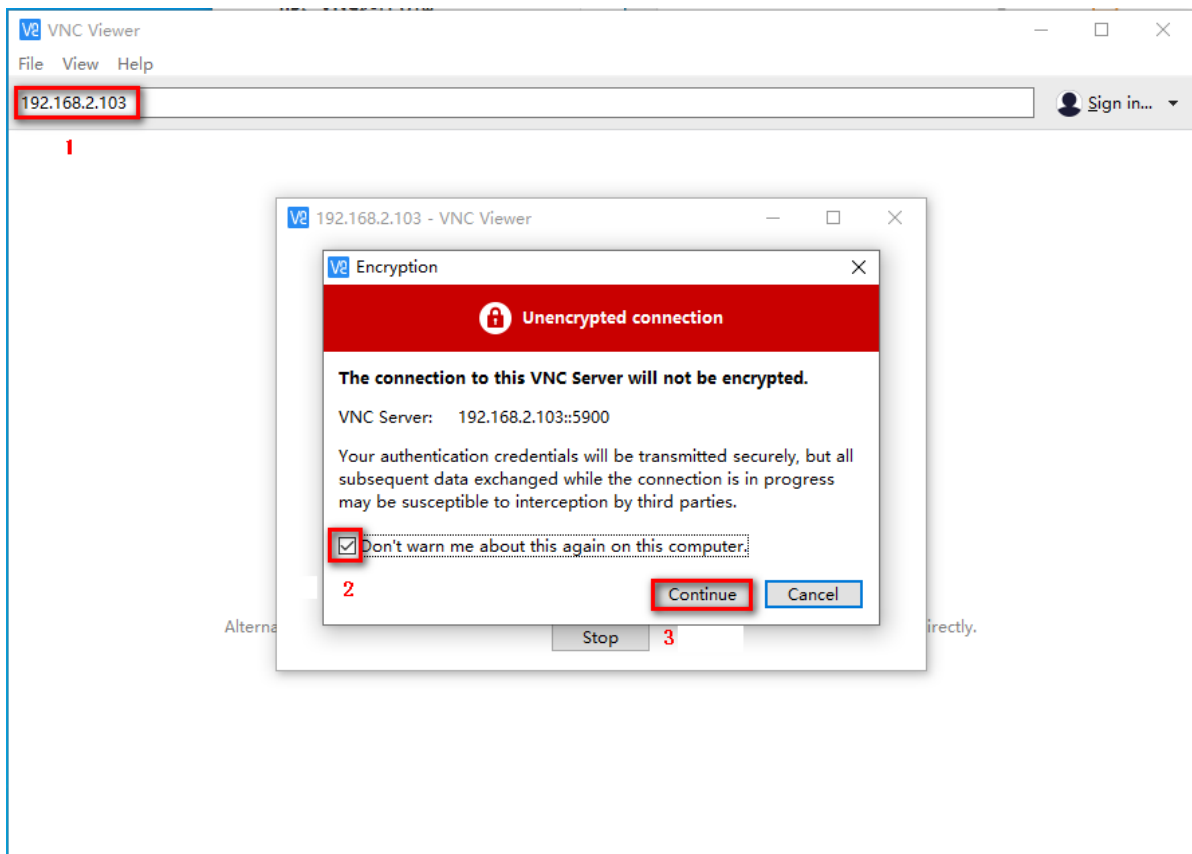


You can enter these folder, and modify content in file, click 【Terminal】 to enter command terminal.

4.1.3、VNC

Note: The system graphical interface can be displayed in this way.

Login steps, as shown below.



4.2、ROS multi-machine communication

4.2.1、Universal version

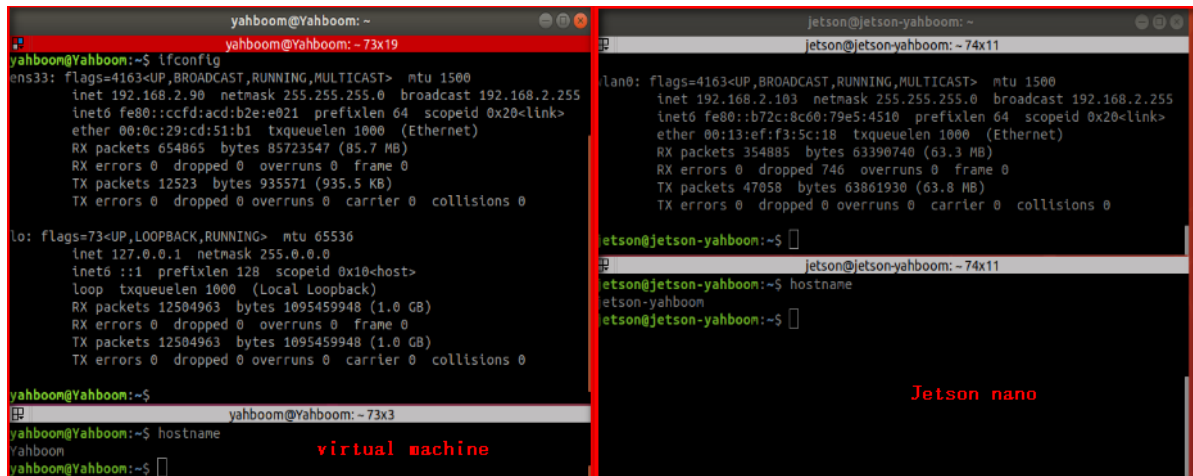
All ros masters are under the same network; if multiple devices have similar processes, choose one as the master, and the others are all slaves; the following two devices are taken as an example.

Install the ssh server on two devices; install the chrony package for synchronization:

```
sudo apt-get install chrony openssh-server
```

Input following command to view the IP information and host name of the two devices:

```
Command:ifconfig and hostname
```



```

yahboom@Yahboom: ~
yahboom@Yahboom: ~ 73x19
yahboom@Yahboom:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.90 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::ccfd:acd:b2e:e021 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:cd:51:b1 txqueuelen 1000 (Ethernet)
    RX packets 654865 bytes 85723547 (85.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12523 bytes 935571 (935.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12504963 bytes 1095459948 (1.0 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12504963 bytes 1095459948 (1.0 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

yahboom@Yahboom:~$
yahboom@Yahboom: ~ 73x3
yahboom@Yahboom:~$ hostname
Yahboom
yahboom@Yahboom:~$

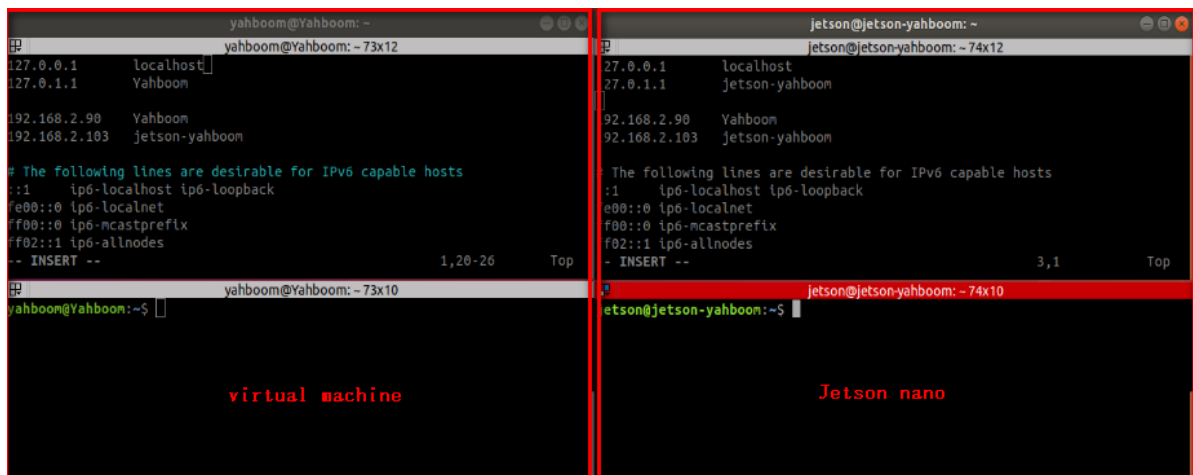
jetson@jetson-yahboom: ~
jetson@jetson-yahboom: ~ 74x11
lan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.103 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::b72c:8c60:79e5:4510 prefixlen 64 scopeid 0x20<link>
    ether 00:13:ef:f3:5c:18 txqueuelen 1000 (Ethernet)
    RX packets 354885 bytes 63390740 (63.3 MB)
    RX errors 0 dropped 746 overruns 0 frame 0
    TX packets 47058 bytes 63861930 (63.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jetson@jetson-yahboom:~$
jetson@jetson-yahboom: ~ 74x11
jetson@jetson-yahboom:~$ hostname
jetson-yahboom
jetson@jetson-yahboom:~$
Jetson nano
```

Modify the hosts file in the /etc folder:

```
sudo chmod a+w /etc/hosts
sudo vim /etc/hosts
```

Add the IP and host names of the two devices to the hosts file of the two devices to bind users; the IP in the front and the name in the back.



```

yahboom@Yahboom: ~
yahboom@Yahboom: ~ 73x12
127.0.0.1    localhost
127.0.1.1    Yahboom

192.168.2.90  Yahboom
192.168.2.103 jetson-yahboom

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
-- INSERT --
1,20-26    Top

yahboom@Yahboom:~$
yahboom@Yahboom: ~ 73x10
yahboom@Yahboom:~$
virtual machine

jetson@jetson-yahboom: ~
jetson@jetson-yahboom: ~ 74x12
27.0.0.1    localhost
27.0.1.1    jetson-yahboom

92.168.2.90  Yahboom
92.168.2.103 jetson-yahboom

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
-- INSERT --
3,1        Top

jetson@jetson-yahboom:~$
jetson@jetson-yahboom: ~ 74x10
jetson@jetson-yahboom:~$
Jetson nano
```

After the modification, enter the following commands on both devices to restart the following network to realize the communication between the two devices:

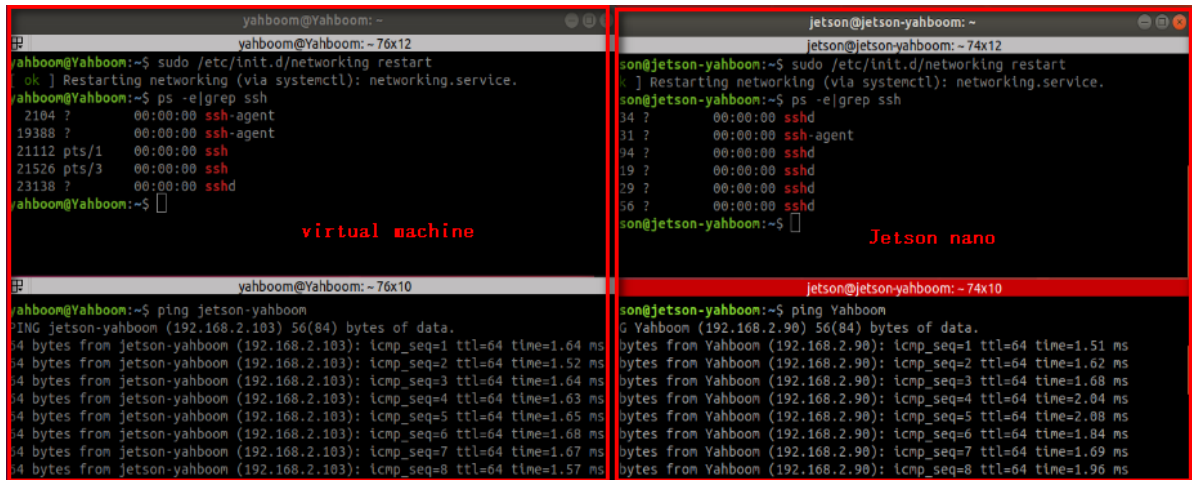
```
sudo /etc/init.d/networking restart
```

After installation, input following command confirm whether the server has been started:

```
ps -e|grep ssh
```

Input following command to check if the communication is normal:

```
ping hostname of the device
```



The image shows two terminal windows side-by-side. The left window is titled 'yahboom@Yahboom: ~' and shows the command 'sudo /etc/init.d/networking restart' being executed, followed by 'ps -e|grep ssh'. The output shows several 'ssh-agent' and 'sshd' processes. Below this, a 'ping' command is run to 'jetson-yahboom', showing successful connectivity with various TTL and time values. The right window is titled 'jetson@jetson-yahboom: ~' and shows the same 'sudo /etc/init.d/networking restart' and 'ps -e|grep ssh' commands. The output shows 'sshd' and 'ssh-agent' processes. Below this, a 'ping' command is run to 'Yahboom', also showing successful connectivity with various TTL and time values. Both windows have a red border and a red title bar.

- **Modify ~/.bashrc file**

```
sudo vim ~/.bashrc
```

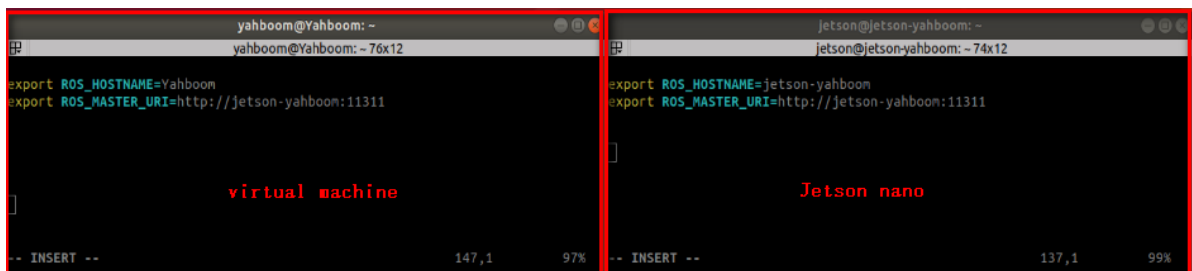
Add following content to ~/.bashrc file of master

```
export ROS_HOSTNAME=Local user name
export ROS_MASTER_URI=http://Host username:11311
```

Add following content to ~/.bashrc file of slave

```
export ROS_HOSTNAME=Local user name
export ROS_MASTER_URI=http://Host username:11311
```

For example, jetson nano as a master



The image shows two terminal windows side-by-side. The left window is titled 'yahboom@Yahboom: ~' and shows the commands 'export ROS_HOSTNAME=Yahboom' and 'export ROS_MASTER_URI=http://jetson-yahboom:11311'. The right window is titled 'jetson@jetson-yahboom: ~' and shows the commands 'export ROS_HOSTNAME=jetson-yahboom' and 'export ROS_MASTER_URI=http://jetson-yahboom:11311'. Both windows have a red border and a red title bar.

After setting the IP, we need to refresh it, and then we can communicate.

```
source ~/.bashrc
```

- **Phenomenon show**

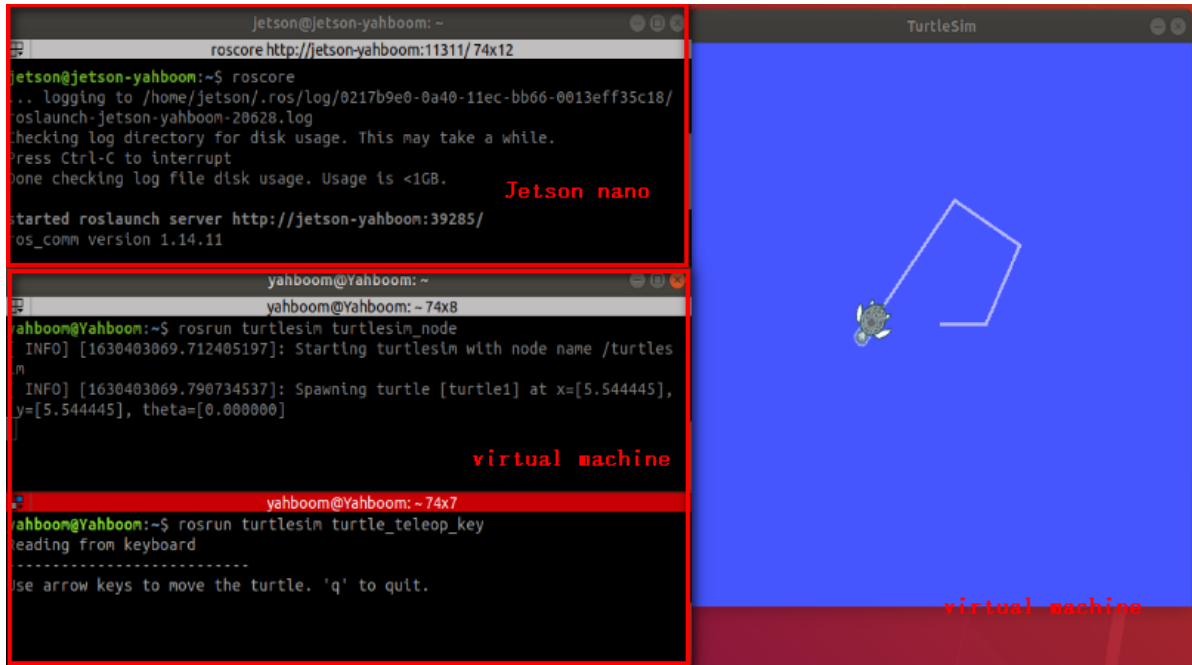
Note: we need to start up ROS Master on ROS Master

jetson nano side

```
roscore
```

ubuntu (virtual machine)

```
roslaunch turtlesim turtlesim_node
roslaunch turtlesim turtle_teleop_key
```



4.2.2. Simple version

If jetson nano is the host and the IP address is known, you only need to modify the .bashrc file of the slave.

```
sudo vim ~/.bashrc
```

Add following content file at the bottom.

```
export ROS_MASTER_URI=http://master username IP:11311
```

4.3. Webpage real-time monitoring

Environment setup

```
sudo apt-get install ros-melodic-async-web-server-cpp ros-melodic-web-video-server ros-melodic-usb-cam
```

Ensure that the USB camera be insert correctly, input following command to check if the USB device exists (it is video0)

```
ls /dev
```

```

drwxr-xr-x  2 root  root    60 Jan  1  1970 vfi0/
crw-----  1 root  root    10, 137 Jul 26 23:31 vhci
crwxrwxrwx+ 1 root  video   81,   0 Aug 30 11:48 video0
crw-----  1 root  root    10, 130 Aug 30 11:48 watchdog
crw-----  1 root  root   244,   0 Aug 30 11:48 watchdog0
crw-rw-rw-  1 root  root     1,   5 Aug 30 11:48 zero
brw-rw----  1 root  disk   252,   0 Aug 30 11:48 zram0
brw-rw----  1 root  disk   252,   1 Aug 30 11:48 zram1
brw-rw----  1 root  disk   252,   2 Aug 30 11:48 zram2
brw-rw----  1 root  disk   252,   3 Aug 30 11:48 zram3
jetson@jetson-yahboom:~$

```

If the system prompts that the execution authority is not enough, you need to input the following command to add the execution authority.

```
sudo chmod 777 /dev/video*
```

Modify usb_cam-test.launch file

```
sudo vim /opt/ros/melodic/share/usb_cam/launch/usb_cam-test.launch
```

```

<launch>
  <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="yuyv" />
    <param name="camera_frame_id" value="usb_cam" />
    <param name="io_method" value="mmap"/>
  </node>
  <node name="image_view" pkg="image_view" type="image_view" respawn="false"
output="screen">
    <remap from="image" to="/usb_cam/image_raw"/>
    <param name="autosize" value="true" />
  </node>
</launch>

```

change

```

<launch>
  <arg name="open_view" default="false"/>
  <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen">
    <param name="video_device" value="/dev/video0"/>
    <param name="image_width" value="640"/>
    <param name="image_height" value="480"/>
    <param name="pixel_format" value="yuyv"/>
    <param name="camera_frame_id" value="usb_cam"/>
    <param name="io_method" value="mmap"/>
  </node>
  <!-- 启动web_video_server -->
  <node pkg="web_video_server" type="web_video_server" name="web_video_server"
output="screen"/>
  <!-- 是否启动image_view -->
  <group if="$(arg open_view)">
    <node name="image_view" pkg="image_view" type="image_view"
respawn="false" output="screen">

```



```
<remap from="image" to="/usb_cam/image_raw"/>
  <param name="autosize" value="true"/>
</node>
</group>
</launch>
```

Open terminal, input following command to start it.

```
roslaunch usb_cam usb_cam-test.launch
```

View pictures

```
view on local web browser
http://localhost:8080/
If you want to view by other devices, you must ensure they at the same local area
network
http://192.168.2.103:8080/
(192.168.2.103 is the IP address of the master)
Note: It is recommended to use Google browser, other browsers may not be able to
open the image
```