

Jupyter Lab environment setup tutorial

Note: This development environment has been built in the factory image. You can skip this tutorial if you use the factory image.

Since Jetson Orin NX/Orin Nano uses Docker and the system environment is already available, this installation takes Jetson Nano as an example.

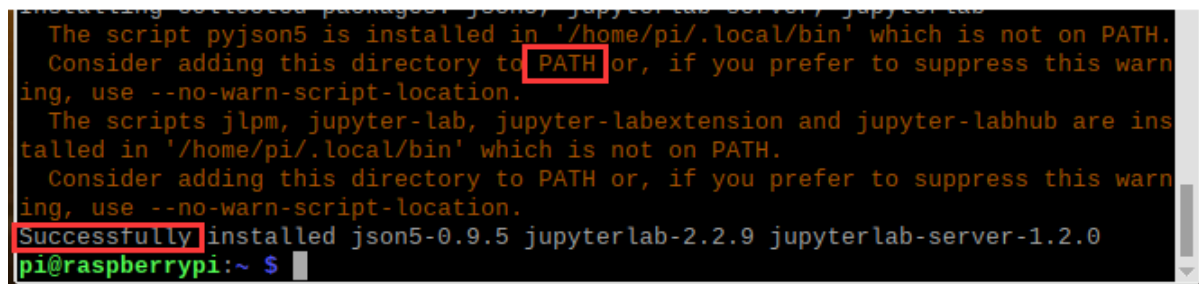
1. Jupyter Lab installation

First, install Jupyter Lab from Tsinghua source (the installation and download speed of this source in China is relatively fast at present, and other sources can also be used for installation). During the installation, some dependent packages need to be downloaded from the external network, and occasionally the connection will be disconnected, causing the installation to fail. You can execute the installation command again and continue the installation from the breakpoint.

```
sudo pip3 install jupyterlab -i https://pypi.tuna.tsinghua.edu.cn/simple
```

2. Add Jupyter Lab path

When successfully appears, it means that Jupyter Lab is installed, but it should be noted that the above alarm information does not find the path. If this problem is not solved, Jupyter Lab cannot be used.



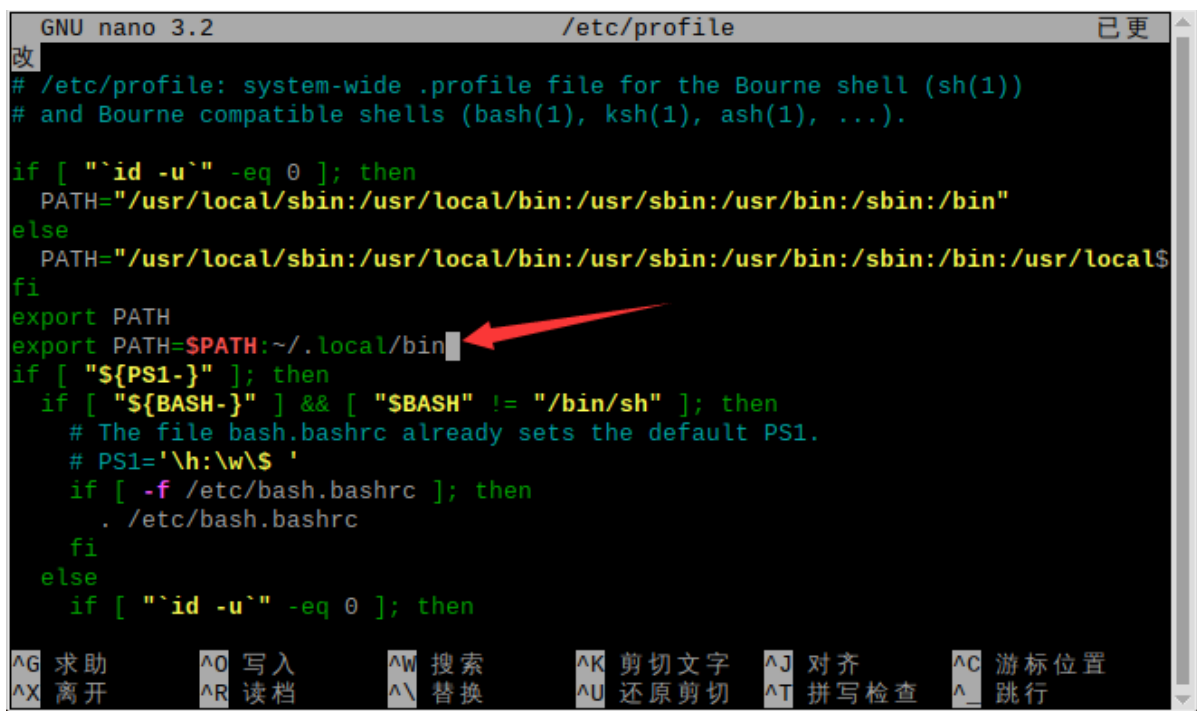
```
Installing selected packages: json5, jupyterlab-server, jupyterlab
The script pyjson5 is installed in '/home/pi/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warn
ing, use --no-warn-script-location.
The scripts jlpm, jupyter-lab, jupyter-labextension and jupyter-labhub are ins
talled in '/home/pi/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warn
ing, use --no-warn-script-location.
Successfully installed json5-0.9.5 jupyterlab-2.2.9 jupyterlab-server-1.2.0
pi@raspberrypi:~ $
```

Modify the profile configuration file

```
sudo nano /etc/profile
```

Add configuration instructions, as shown in the figure

```
export PATH=$PATH:~/.local/bin
```



```
GNU nano 3.2 /etc/profile 已更
改
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "`id -u`" -eq 0 ]; then
    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local$
fi
export PATH
export PATH=$PATH:~/.local/bin
if [ "${PS1-}" ]; then
    if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
        # The file bash.bashrc already sets the default PS1.
        # PS1='\h:\w\$ '
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "`id -u`" -eq 0 ]; then
            . /etc/bash.bashrc
        fi
    fi
fi

^G 求助      ^O 写入      ^W 搜索      ^K 剪切文字  ^J 对齐      ^C 光标位置
^X 离开      ^R 读档      ^\ 替换      ^U 还原剪切  ^T 拼写检查  ^_ 跳行
```

After writing, press Ctrl+S to save, and then press Ctrl+X to exit.

Then execute

```
source /etc/profile
```

Start jupyterlab. After starting, the browser will automatically pop up the jupyterlab interface.

```
jupyter lab
```

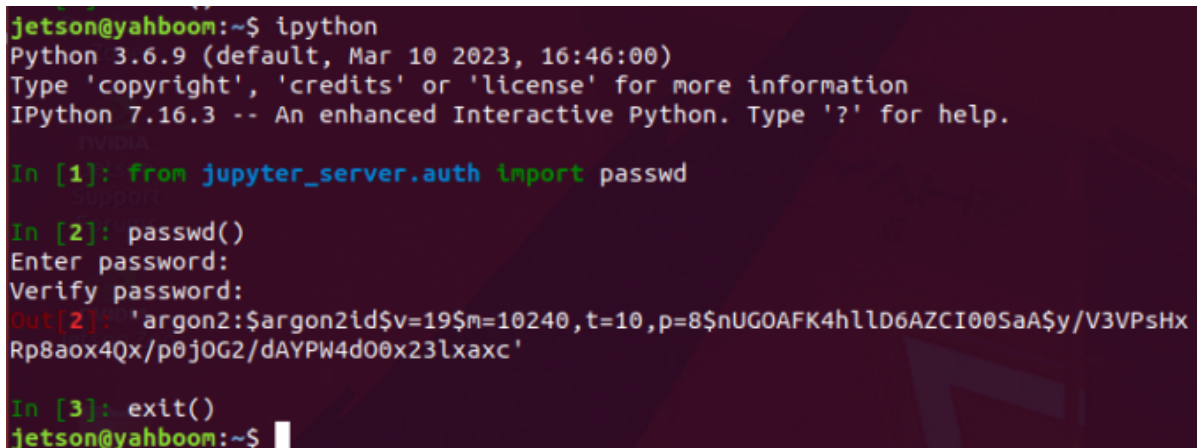
3. Set up LAN to access Jupyter Lab

If you want to run it on other devices in the same LAN, you need to follow the following process

1. Generate a key

Open the terminal and enter the following content in sequence, and then enter the password twice according to the prompt. It is recommended to be consistent with the user password.

```
ipython
from jupyter_server.auth import passwd
passwd()
```



```
jetson@yahboom:~$ ipython
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.3 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from jupyter_server.auth import passwd
In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$nUGOAFK4h1lD6AZCI00SaA$y/V3VPsHx
Rp8aox4Qx/p0jOG2/dAYPW4d00x23lxaxc'

In [3]: exit()
jetson@yahboom:~$
```

A string of keys will be generated after the password confirmation is completed. Copy this key and enter the following command to exit.

```
exit()
```

2. Create a configuration file

After creation, the detailed location of the file will be output, please remember this location.

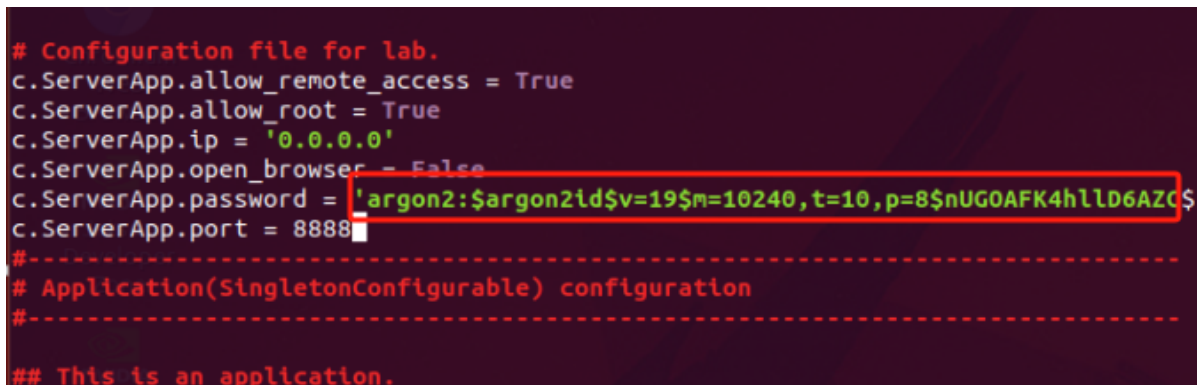
```
jupyter lab --generate-config
```

3. Modify the configuration file

```
nano ~/.jupyter/jupyter_lab_config.py
```

Add the following content, where the password item needs to be replaced with the key generated above.

```
c.ServerApp.allow_remote_access = True
c.ServerApp.allow_root = True
c.ServerApp.ip = '0.0.0.0'
c.ServerApp.open_browser = False
c.ServerApp.password =
'argon2:$argon2id$v=19$m=10240,t=10,p=8$nUG0AFK4h1lD6AZCI00SaA$y/V3VPsHxRp8aox4Qx
/p0jOG2/dAYPW4d00x23lxaxc'
c.ServerApp.port = 8888
```



```
# Configuration file for lab.
c.ServerApp.allow_remote_access = True
c.ServerApp.allow_root = True
c.ServerApp.ip = '0.0.0.0'
c.ServerApp.open_browser = False
c.ServerApp.password = 'argon2:$argon2id$v=19$m=10240,t=10,p=8$nUG0AFK4h1lD6AZCI00SaA$y/V3VPsHxRp8aox4Qx/p0jOG2/dAYPW4d00x23lxaxc'
c.ServerApp.port = 8888
#-----
# Application(SingletonConfigurable) configuration
#-----
## This is an application.
```

After writing, press Ctrl+S to save, and then press Ctrl+X to exit.

Restart the system

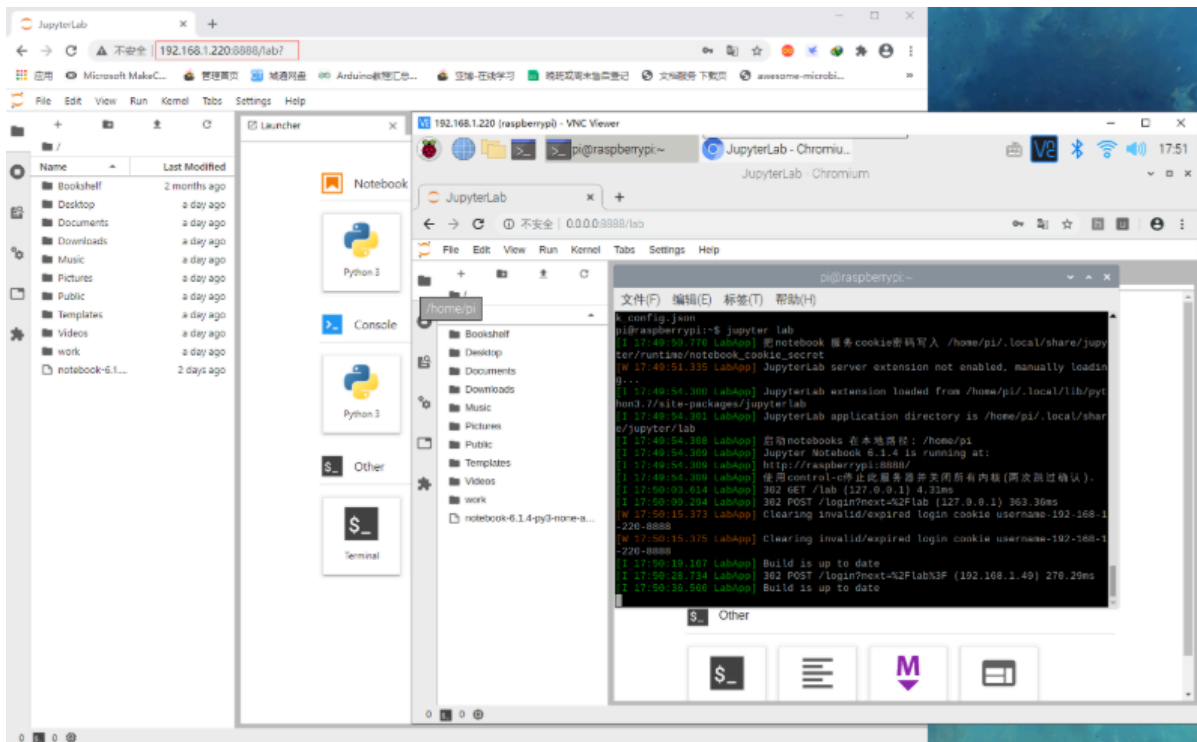
```
sudo reboot
```

Step 5: Run the test

```
jupyter lab
```

You can access <http://127.0.0.1:8888> in your local browser or <http://8888> on a machine in the LAN, then enter the password you set, and you can operate your robot on the web page.

For example: If the robot's address is 192.168.1.220, then the machine in the LAN should access <http://192.168.1.220:8888> as shown below:



4. Configure Jupyter Lab's startup program

After the installation is complete, under normal circumstances, each time you use JupyterLab, you need to enter the command `jupyter lab` to start. For convenience, we configure the startup program. The operation is as follows:

Step 1: Create a startup file

```
sudo vim /etc/systemd/system/yahboom_jupyterlab.service
```

Step 2: Copy the code to this file, save and exit

jetson nano system service file

```
[Unit]
Description=Jupyter Lab Service

[Service]
Type=simple
User=jetson
Environment="OPENBLAS_CORETYPE=ARMV8"
ExecStart=/bin/sh -c "jupyter lab --ip=0.0.0.0 --no-browser"
WorkingDirectory=/home/jetson
Restart=always

[Install]
WantedBy=multi-user.target
```

Raspberry Pi system service file

```
[Unit]
Description=Jupyter Lab Service

[Service]
Type=simple
User=pi
Environment="OPENBLAS_CORETYPE=ARMV8"
ExecStart=/bin/sh -c "jupyter lab --ip=0.0.0.0 --no-browser"
WorkingDirectory=/home/pi
Restart=always

[Install]
WantedBy=multi-user.target
```

After writing, press Ctrl+S to save, and then press Ctrl+X to exit.

Step 3: Enable the service to start automatically at boot

```
sudo systemctl daemon-reload
sudo systemctl enable yahboom_jupyterlab.service
sudo systemctl restart yahboom_jupyterlab.service
```

Step 4: Restart the system

```
sudo reboot
```

Try to access <http://127.0.0.1:8888> locally

Or

Access `http://《your-ip-address》:8888` on a machine in the LAN, such as: <http://192.168.1.220:8888>

If the access is normal, the Jupyter Lab environment is successfully built!