

1. Install Jupyter Lab

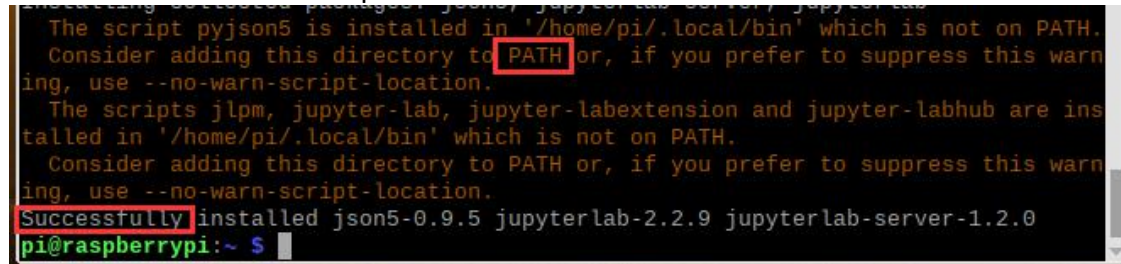
Input following command to install Jupyter Lab.

```
sudo pip3 install jupyterlab
```

2. Add Jupyter lab path

When the system prompts successfully, the installation of Jupyter Lab is completed.

If the warning message shown in the thumbnail as shown below appears, the path is not found. We need to add the path.

A terminal window showing the output of the pip3 install command. It displays two warnings about the installation path /home/pi/.local/bin not being on the PATH, suggesting to add it or use --no-warn-script-location. The final line shows a successful installation of json5, jupyterlab, and jupyterlab-server.

```
Installing collected 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 jlpmp, jupyterlab, jupyterlabextension and jupyterlabhub 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:~ $
```

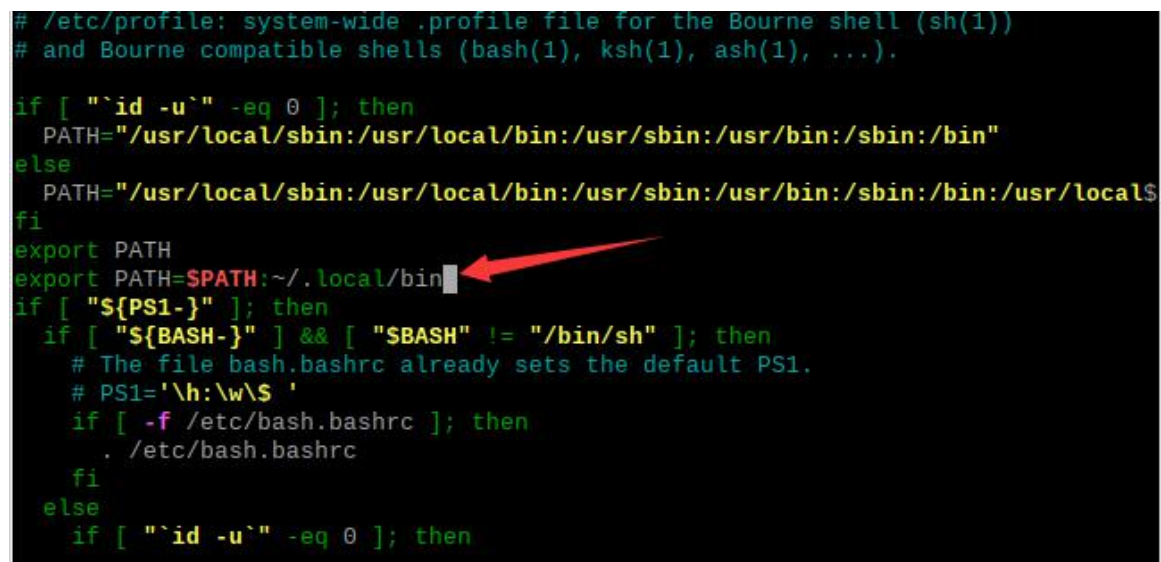
Modify profile file.

Input following command:

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

Add configuration instructions, as shown below.

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

A terminal window showing the contents of the /etc/profile file. The file contains shell configuration for Bourne, Bash, and Korn shells. A red arrow points to the line 'export PATH=\$PATH:~/.local/bin' which has been added to the file.

```
# /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/loca
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\S '
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "`id -u`" -eq 0 ]; then
```

After modify is complete, press "Ctrl + X", and press Y to save and exit.

Input following command start up Jupyter Lab.

```
source /etc/profile
```

The browser will automatically pop up the jupyterlab interface.

3. Set up the LAN to access Jupyter Lab

3.1 Input following command to create configuration file.(Please remember this path).

```
jupyter notebook --generate-config
```

3.2 modify configuration file

```
sudo nano /home/pi/.jupyter/jupyter_notebook_config.py
```

Write the following two pieces of code into the specified location.

```
c.NotebookApp.allow_origin = '*' #allow all origins
```

```
c.NotebookApp.ip = '0.0.0.0' # listen on all Ips
```

```
#c.NotebookApp.allow_credentials = False

## Set the Access-Control-Allow-Origin header
# Use '*' to allow any origin to access your server.
# Takes precedence over allow_origin_pat.
c.NotebookApp.allow_origin = ''
c.NotebookApp.allow_origin = '*' #allow all origins
c.NotebookApp.ip = '0.0.0.0' # listen on all Ips
## Use a regular expression for the Access-Control-Allow-Origin header
# Requests from an origin matching the expression will get replies with:
# Access-Control-Allow-Origin: origin
# where `origin` is the origin of the request.
# Ignored if allow_origin is set.
```

After modify is complete, press “Ctrl + X”, and press Y to save and exit.

3.3 Set password.

Input following command.

```
jupyter notebook password
```

The system will ask you to enter the password twice.

!Tip: In the state of entering the password, it will no display characters.

3.4 Input following command to restart Raspberry Pi.

```
sudo reboot
```

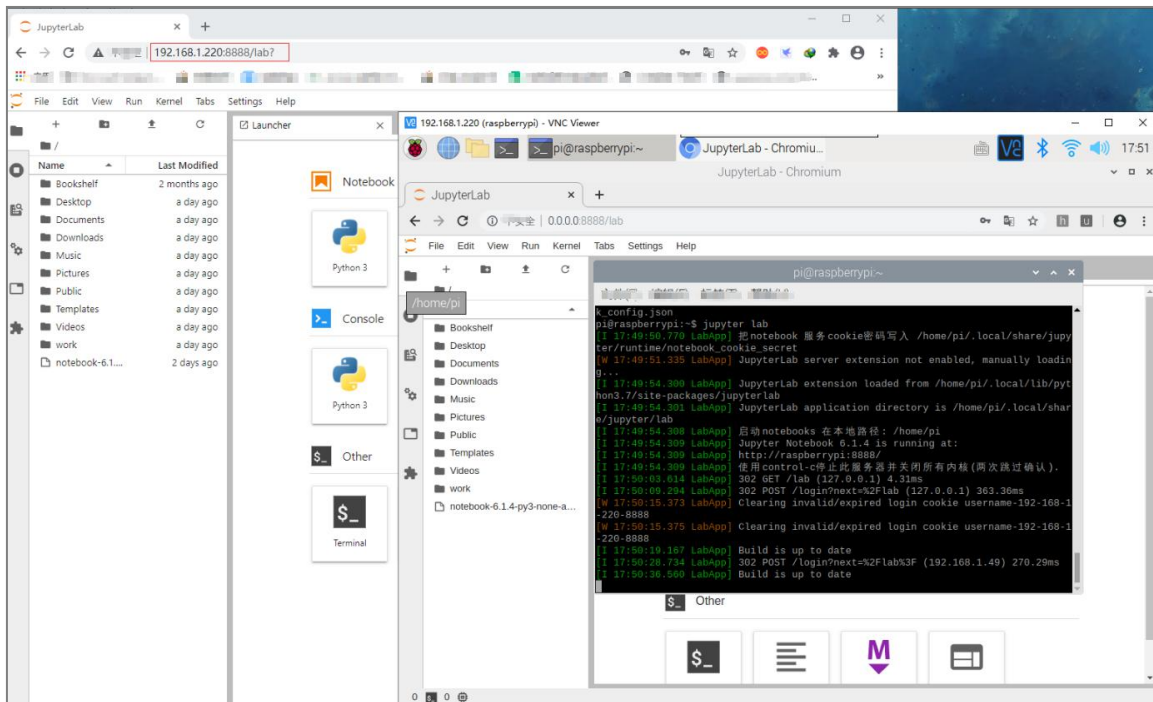
3.5 Test

```
jupyter lab
```

Next, we can enter <http://127.0.0.1:8888> locally on the Raspberry Pi to access Jupyter Lab. Or enter <http://<your-ip-address>:8888> on a machine in the local area network.

You can enter the password you set. Finally, you can operate your program on Webpage by JupyterLab.

As shown below.



4. Configure Jupyter Lab to start automatically

After the installation is complete. In normally, when use JupyterLab every time, you need to enter the command to start jupyter lab.

For convenience, we can make JupyterLab start the program automatically.

The operation is as follows:

4.1 Input following command to create a startup file.

```
nano /home/pi/yahboomlabboot.sh
```

4.2 Copy the code to the file, save and exit

```
# shell script to set path and run jupyter notebook server at boot
export PATH="$PATH:/home/pi/.local/bin/"
jupyter lab
```

After modify is complete, press "Ctrl + X", and press Y to save and exit.

4.3 Add the file to the startup item

```
sudo nano /etc/rc.local
```

Copy the following code to the previous line of exit 0 in the file.

```
su pi -c 'bash /home/pi/yahboomlabboot.sh'
```

```
GNU nano 3.2 rc.local Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
su pi -c 'bash /home/pi/yahboomlabboot.sh'
exit 0
```

After modify is complete, press “Ctrl + X”, and press Y to save and exit.

4.4 Input following command to restart Raspberry Pi.

sudo reboot

4.5 Test

You can try to access <http://127.0.0.1:8888> locally

or

access <http://<your-ip-address>:8888> on the device in the same LAN, such as:

<http://192.168.1.220:8888>

If it can be accessed normally, it means Jupyter Lab is successfully installed.