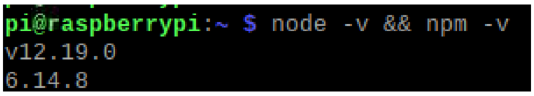
4.Widgets configuration

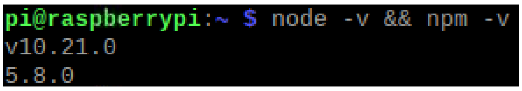
1. Install ipywidgets

1. Refer to `1.jupyter lab environment construction tutorial` to install Jupyter lab.
2. Check whether `node` and `nmp` are installed. If the version number is displayed, it means that it has been installed. You can continue the operation (the second part **Installing Node.js** can be skipped directly), otherwise you should jump first Go to the **Second, Install Node.js** section, install Node.js and then continue. (As shown in the figure, both versions can be used directly)

```
node -v && npm -v
```



```
pi@raspberrypi:~ $ node -v && npm -v
v12.19.0
6.14.8
```



```
pi@raspberrypi:~ $ node -v && npm -v
v10.21.0
5.8.0
```

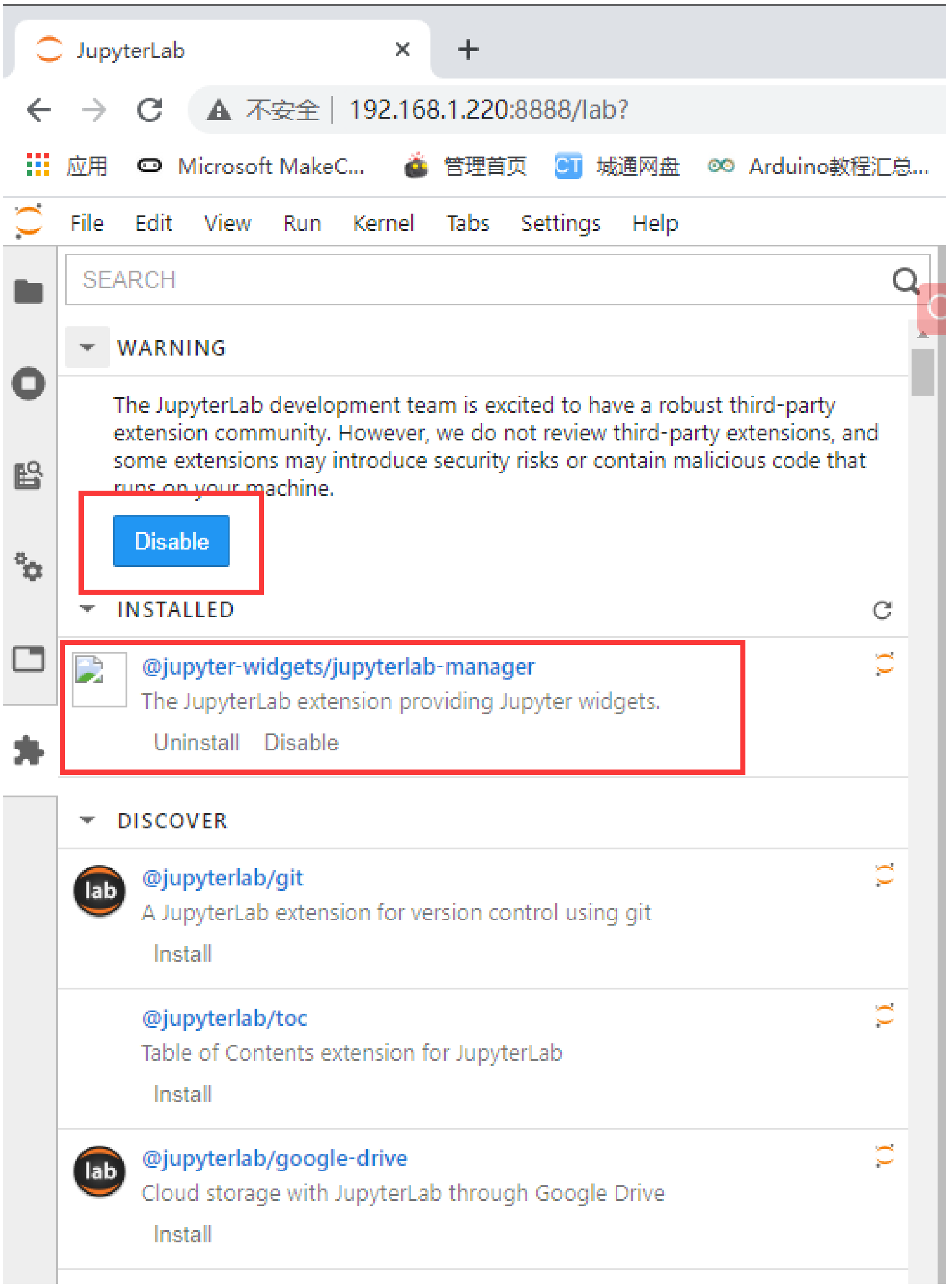
3. Install @jupyter-widgets/jupyterlab-manager (this step requires operating in jupyter lab)

First enable the third-party extension community

Then install (install) @jupyter-widgets/jupyterlab-manager management plug-in

After the installation is complete, the plug-in will appear in the **INSTALLED** (installed) area, as shown below.

Generally, after the installation is completed, you will be prompted to **Build JupyterLab** again. This process will take a long time, and there will be no prompt after success. It is recommended to wait for about 2-3 minutes before re-entering jupyterlab. If there is no prompt **Build JupyterLab**, it means Build was successful.



JupyterLab
不安全 | 192.168.1.220:8888/lab?
应用 Microsoft MakeC... 管理首页 城通网盘 Arduino教程汇总...
File Edit View Run Kernel Tabs Settings Help
SEARCH
WARNING

The JupyterLab development team is excited to have a robust third-party extension community. However, we do not review third-party extensions, and some extensions may introduce security risks or contain malicious code that runs on your machine.

@jupyter-widgets/jupyterlab-manager
The JupyterLab extension providing Jupyter widgets.

Uninstall Disable
DISCOVER

@jupyterlab/git
A JupyterLab extension for version control using git

@jupyterlab/toc
Table of Contents extension for JupyterLab

@jupyterlab/google-drive
Cloud storage with JupyterLab through Google Drive

3.Install `ipywidgets`

```
pip3 install ipywidgets
```

4.Start up `widgetsnbextension`

```
jupyter nbextension enable --py widgetsnbextension
```

5.Delete temporary and static directories

```
jupyter lab clean
jupyter lab path
```

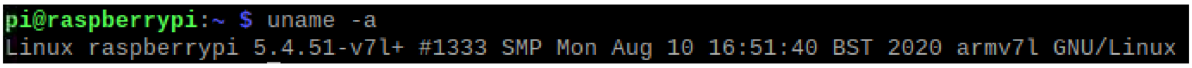
6.Restart Raspberry Pi

```
sudo reboot
```

2. Install Node.js

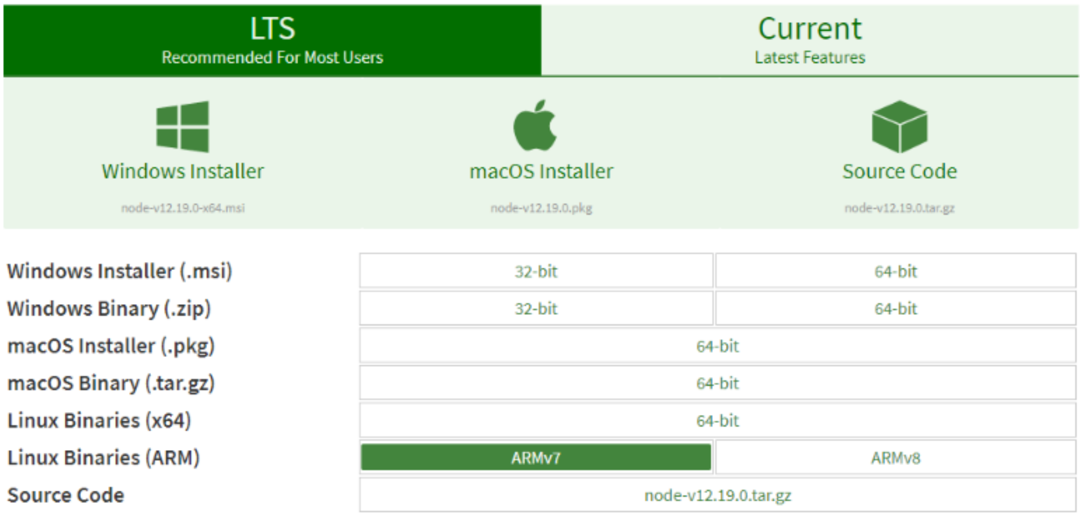
1.Looking at the architecture of the Raspberry Pi, you can see that my Raspberry Pi 4B is armv7.

```
uname -a
```



```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 5.4.51-v7l+ #1333 SMP Mon Aug 10 16:51:40 BST 2020 armv7l GNU/Linux
```

2.Enter the Nood.js official website [download page] (https://nodejs.org/en/download/) to find the corresponding version and download it.



Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.19.0.tar.gz	

3.Unzip the downloaded compressed file (the current Nood.js official website provides version 12.19.0, if subsequent versions change, please refer to the actual situation)

```
xz -d node-v12.19.0-linux-armv7l.tar.xz
tar -xavf node-v12.19.0-linux-armv7l.tar
```

4.将Delete the `/usr/bin.node` that originally existed in the system

```
sudo rm -rf /usr/bin/node
```

5.Move the binary package to `/usr/local/node`

```
sudo mv ./node-v12.19.0-linux-armv7l /usr/local/node
```

6.Create a soft connection between `node` and `npm`

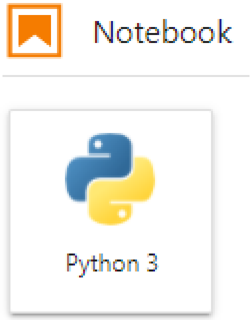
```
sudo ln -s /usr/local/node/bin/node /usr/bin/node
sudo ln -s /usr/local/node/bin/npm /usr/bin/npm
```

3. Check

1.Enter jupyter lab

```
jupyter lab
```

2.Enter Notebook



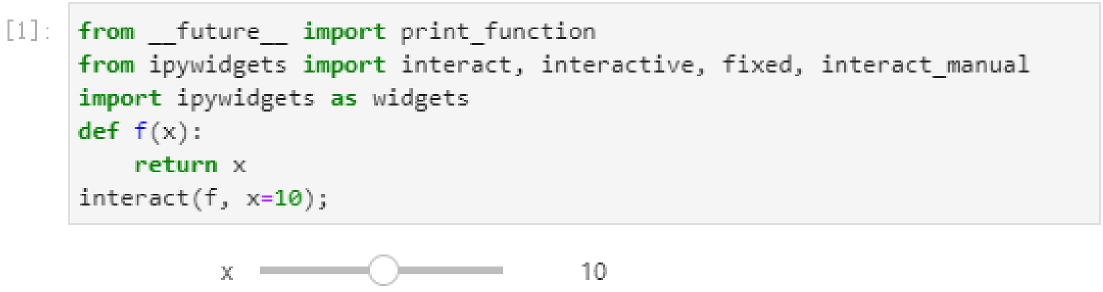
3.Copy and paste the following sample code

```
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets
def f(x):
    return x
interact(f, x=10);
```

4.Use the following shortcut key to start running.

If the result is as follows, the Jupyter Widgets configuration is complete.

Enter key + Shift key



```
[1]: from __future__ import print_function
     from ipywidgets import interact, interactive, fixed, interact_manual
     import ipywidgets as widgets
     def f(x):
         return x
     interact(f, x=10);
```

x 10