

Jupyter Lab environment construction

Jupyter Lab environment construction

1. Install Jupyter Lab
2. Open Jupyter Lab
3. Set up LAN to access jupyter lab
4. Set up access to jupyter lab
5. Set jupyter lab to start automatically after booting

Jupyter Lab is a web-based interactive development environment that supports multiple programming languages; it provides a flexible workspace that can perform various data science tasks such as data cleaning, visualization, and machine learning modeling.

1. Install Jupyter Lab

- Check the system python version

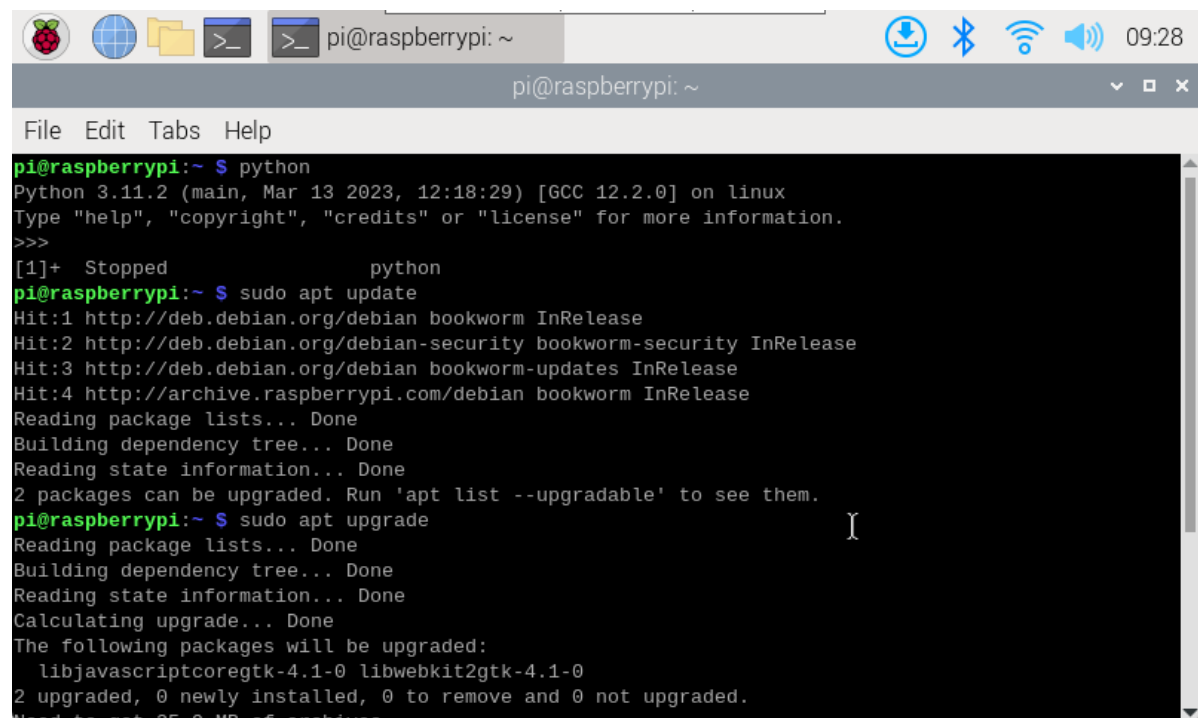
Enter the command in the terminal:

```
python
```

- Install Jupyter Lab

Update the repository list and software before installing the software:

```
sudo apt update  
sudo apt upgrade
```

A screenshot of a Raspberry Pi terminal window. The window title is 'pi@raspberrypi: ~'. The terminal shows the following commands and output:

```
pi@raspberrypi:~ $ python  
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
[1]+  Stopped                  python  
pi@raspberrypi:~ $ sudo apt update  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
2 packages can be upgraded. Run 'apt list --upgradable' to see them.  
pi@raspberrypi:~ $ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
The following packages will be upgraded:  
  libjavascriptcoregtk-4.1-0 libwebkit2gtk-4.1-0  
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
Need to get 25.0 MB of archives.
```

Install Jupyter Lab in a Python 3 environment and enter the command in the terminal:

```
sudo pip3 install jupyterlab
```

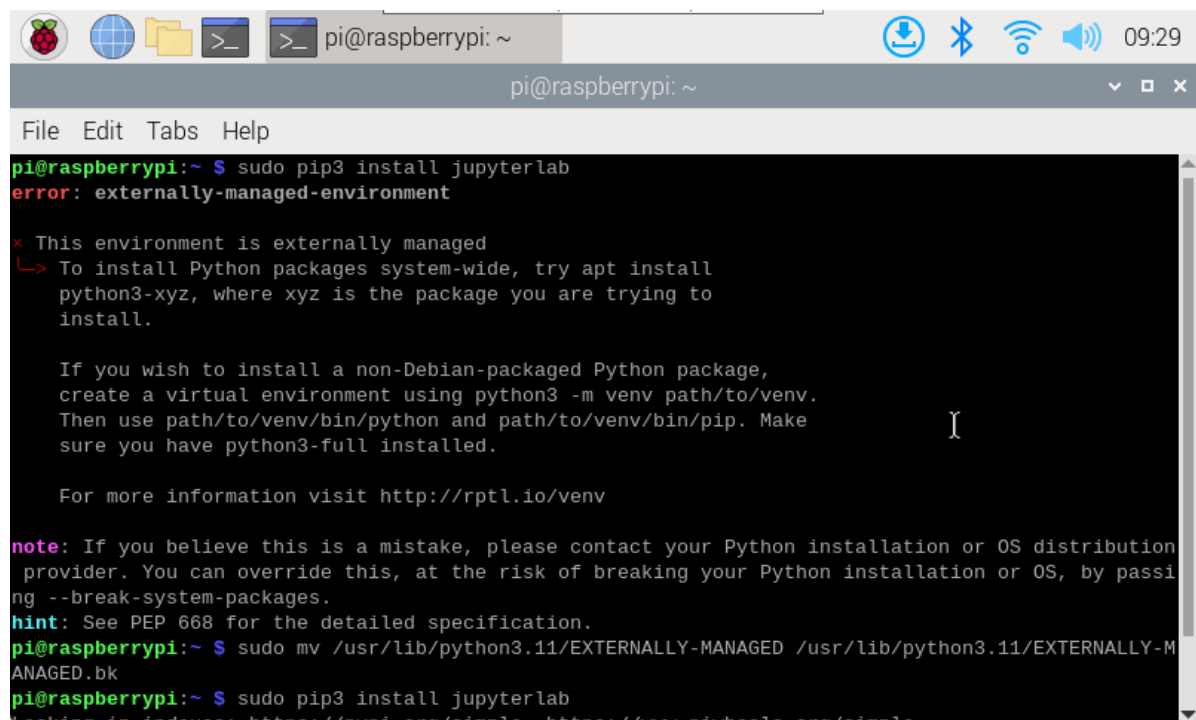
If the download fails multiple times, you can specify the Python software package mirror address of Tsinghua University to speed up domestic downloads:

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

- Error reporting and resolution

If you directly enter the command to install Jupyter Lab in the terminal, an "error: externally-managed-environment" error will appear. You can use the following command to solve it: The python version is modified according to the version of your own system. My current system version is 3.11.

```
sudo mv /usr/lib/python3.11/EXTERNALLY-MANAGED /usr/lib/python3.11/EXTERNALLY-MANAGED.bk
```



The screenshot shows a terminal window on a Raspberry Pi. The top bar displays the Raspberry Pi logo, network icons, and the time 09:29. The terminal title is 'pi@raspberrypi: ~'. The menu bar includes 'File', 'Edit', 'Tabs', and 'Help'. The terminal content shows the command 'sudo pip3 install jupyterlab' being executed, which results in an 'error: externally-managed-environment'. A detailed error message follows, explaining that the environment is externally managed and providing instructions on how to install Python packages system-wide or create a virtual environment. The user then runs the command 'sudo mv /usr/lib/python3.11/EXTERNALLY-MANAGED /usr/lib/python3.11/EXTERNALLY-MANAGED.bk' to resolve the issue. Finally, the user runs 'sudo pip3 install jupyterlab' again, and the installation is successful.

```
pi@raspberrypi:~ $ sudo pip3 install jupyterlab
error: externally-managed-environment

× This environment is externally managed
╔> To install Python packages system-wide, try apt install
python3-xyz, where xyz is the package you are trying to
install.

If you wish to install a non-Debian-packaged Python package,
create a virtual environment using python3 -m venv path/to/venv.
Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
sure you have python3-full installed.

For more information visit http://rptl.io/venv

note: If you believe this is a mistake, please contact your Python installation or OS distribution
provider. You can override this, at the risk of breaking your Python installation or OS, by passi
ng --break-system-packages.
hint: See PEP 668 for the detailed specification.
pi@raspberrypi:~ $ sudo mv /usr/lib/python3.11/EXTERNALLY-MANAGED /usr/lib/python3.11/EXTERNALLY-M
ANAGED.bk
pi@raspberrypi:~ $ sudo pip3 install jupyterlab
Looking in indexes: https://pypi.org/simple, https://pypi.tuna.tsinghua.edu.cn/simple
```

- Installation success prompt

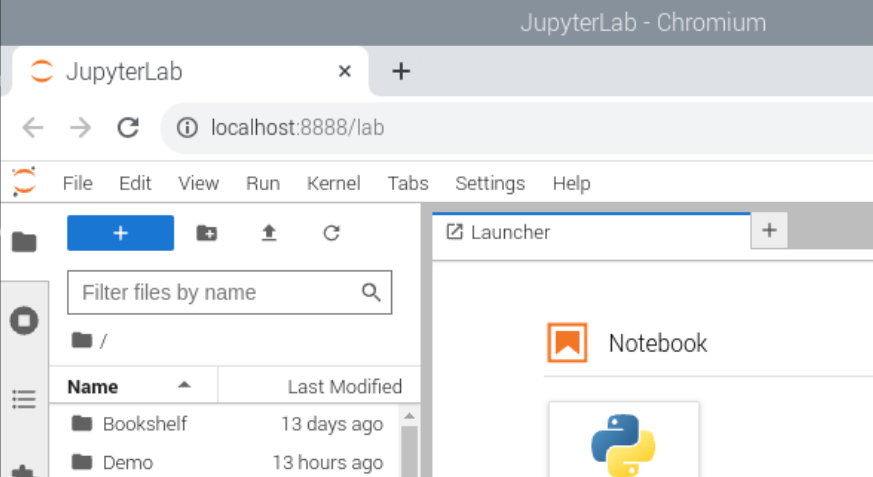
The following prompt appears indicating that the installation is successful.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
Can't uninstall 'Send2Trash'. No files were found to uninstall.  
Attempting uninstall: requests  
Found existing installation: requests 2.28.1  
Not uninstalling requests at /usr/lib/python3/dist-packages, outside environment /usr  
Can't uninstall 'requests'. No files were found to uninstall.  
Successfully installed anyio-4.2.0 argon2-cffi-23.1.0 argon2-cffi-bindings-21.2.0 arrow-1.3.0 asyn  
c-lru-2.0.4 attrs-23.1.0 bleach-6.1.0 cffi-1.16.0 comm-0.2.0 debugpy-1.8.0 decorator-5.1.1 defused  
xml-0.7.1 executing-2.0.1 fastjsonschema-2.19.0 fqdn-1.5.1 ipykernel-6.27.1 ipython-8.18.1 isodura  
tion-20.11.0 json5-0.9.14 jsonpointer-2.4 jsonschema-4.20.0 jsonschema-specifications-2023.11.2 ju  
pyter-client-8.6.0 jupyter-core-5.5.1 jupyter-events-0.9.0 jupyter-lsp-2.2.1 jupyter-server-2.12.1  
jupyter-server-terminals-0.5.0 jupyterlab-4.0.9 jupyterlab-pygments-0.3.0 jupyterlab-server-2.25.  
2 matplotlib-inline-0.1.6 mistune-3.0.2 nbclient-0.9.0 nbconvert-7.13.0 nbformat-5.9.2 nest-asyncio-1.5.8 notebook-shim-0.2.3 overrides-7.4.0 packaging-23.2 pandocfilters-1.5.0 prometheus-client-0  
.19.0 prompt-toolkit-3.0.43 pure-eval-0.2.2 pycparser-2.21 python-dateutil-2.8.2 python-json-logge  
r-2.0.7 pyyaml-6.0.1 pyzmq-25.1.2 referencing-0.32.0 requests-2.31.0 rfc3339-validator-0.1.4 rfc39  
86-validator-0.1.1 rpds-py-0.15.2 send2trash-1.8.2 sniffio-1.3.0 stack-data-0.6.3 terminado-0.18.0  
tinycss2-1.2.1 tornado-6.4 traitlets-5.14.0 types-python-dateutil-2.8.19.14 uri-template-1.3.0 wc  
width-0.2.12 webcolors-1.13 websocket-client-1.7.0  
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour  
with the system package manager. It is recommended to use a virtual environment instead: https://  
pip.pypa.io/warnings/venv  
pi@raspberrypi:~$
```

2. Open Jupyter Lab

Just enter jupyter lab in the terminal. If you need a password, you can follow the fourth step of the tutorial to set the password before using it!

```
jupyterlab  
  
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ jupyter lab  
[W 2023-12-19 09:32:14.012 ServerApp] A `_jupyter_server_extension_points` function was not found  
in jupyter_lsp. Instead, a `_jupyter_server_extension_paths` function was found and will be used f  
or now. This function name  
[W 2023-12-19 09:32:14.019  
in notebook_shim. Instead,  
for now. This function na  
[I 2023-12-19 09:32:14.019  
[I 2023-12-19 09:32:14.023  
[I 2023-12-19 09:32:14.027  
[I 2023-12-19 09:32:14.029  
e/jupyter/runtime/jupyter_  
[I 2023-12-19 09:32:14.316  
[I 2023-12-19 09:32:14.330  
[I 2023-12-19 09:32:14.332  
[I 2023-12-19 09:32:14.332  
[I 2023-12-19 09:32:14.333  
-packages/jupyterlab  
[I 2023-12-19 09:32:14.333  
b  
[I 2023-12-19 09:32:14.334  
b
```



Before installing Jupyter Lab, select the system default browser, otherwise Jupyter Lab will not be started directly from the browser; use the sudo command to install jupyter lab. The warning message that appears can be ignored.

3. Set up LAN to access jupyter lab

- Create configuration file

The generated configuration file path is the path to the file that will be modified later.

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

```
pi@raspberrypi:~ $ jupyter lab --generate-config  
Writing default config to: /home/pi/.jupyter/jupyter_lab_config.py
```

- Modify configuration file

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

Uncomment the content of the file and modify it to the following content: You can use the Ctrl+W shortcut key to search for keywords in the nano editor

```
c.ServerApp.allow_origin = '*'  
c.ServerApp.ip = '0.0.0.0'
```

Press Ctrl+X, enter Y, then press Enter to save and exit editing!

```
GNU nano 7.2 /home/pi/.jupyter/jupyter_lab_config.py *
# Default: 0.0
# c.ServerApp.iopub_data_rate_limit = 0.0

## DEPRECATED. Use ZMQChannelsWebsocketConnection.iopub_msg_rate_limit
# Default: 0.0
# c.ServerApp.iopub_msg_rate_limit = 0.0

## The IP address the Jupyter server will listen on.
# Default: 'localhost'
c.ServerApp.ip = '0.0.0.0'

## Supply extra arguments that will be passed to Jinja environment.
# Default: {}
# c.ServerApp.jinja_environment_options = {}

## Extra variables to supply to jinja templates when rendering.
# Default: {}
# c.ServerApp.jinja_template_vars = {}

AG Help      AO Write Out  AW Where Is  AK Cut       AT Execute   AC Location
AX Exit      AR Read File  AL Replace  AU Paste     AJ Justify   AN Go To Line

GNU nano 7.2 /home/pi/.jupyter/jupyter_lab_config.py *
## Whether or not to allow external kernels, whose connection files are placed in
# external_connection_dir.
# Default: False
# c.ServerApp.allow_external_kernels = False

## Set the Access-Control-Allow-Origin header
#
# Use '*' to allow any origin to access your server.
#
# Takes precedence over allow_origin_pat.
# Default: ''
c.ServerApp.allow_origin = '*'

## 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
[ Search Wrapped ]

AG Help      AO Write Out  AW Where Is  AK Cut       AT Execute   AC Location
AX Exit      AR Read File  AL Replace  AU Paste     AJ Justify   AN Go To Line
```

4. Set up access to jupyter lab

Enter the command to set the password on the terminal. You need to enter it twice. Entering the password will not display the input content.

```
jupyter lab password
```

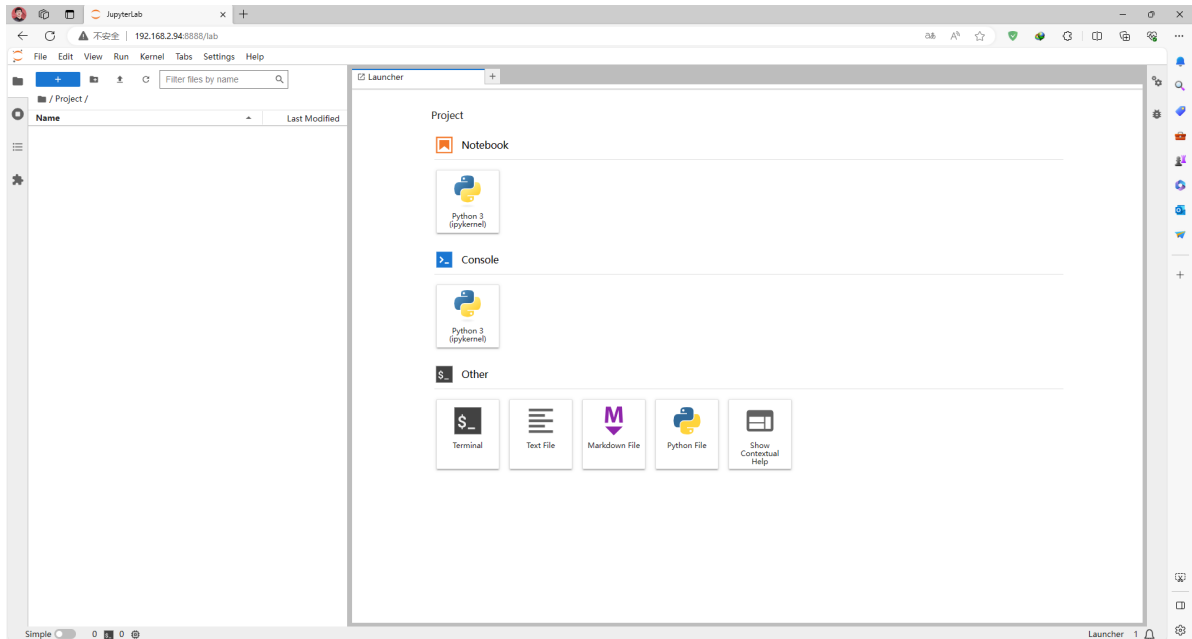
```
pi@raspberrypi:~$ jupyter lab password
Enter password:
Verify password:
[JupyterPasswordApp] Wrote hashed password to /home/pi/.jupyter/jupyter_server_config.json
```

Restart the Raspberry Pi after setting the password!

- verify

Devices on the same LAN can enter IP:8888 in the browser to access!

The password is the password set before: yahboom



5. Set jupyter lab to start automatically after booting

After completing the above steps, you need to enter a command in the terminal every time you use jupyter lab; for more convenient use, we can configure jupyter lab to start automatically at boot.

- Configure startup items

Enter the following command in the terminal:

```
sudo nano /etc/systemd/system/jupyter.service
```

Add the following content to the file:

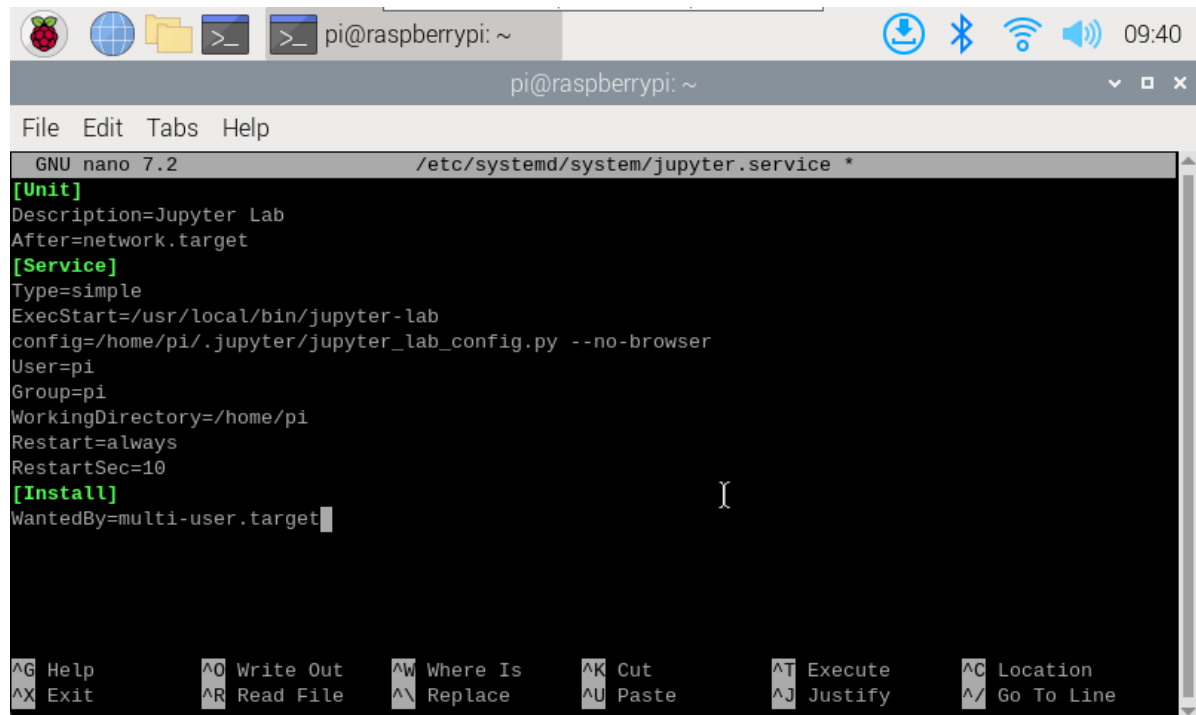
```
[Unit]
Description=Jupyter Lab
After=network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/jupyter-lab
config=/home/pi/.jupyter/jupyter_lab_config.py --no-browser
User=pi
Group=pi
workingDirectory=/home/pi
Restart=always
RestartSec=10
[Install]
wantedBy=multi-user.target
```

pi: my current system username

ExecStart: The command to start Jupyter lab, change it to the installation path and configuration file path of JupyterLab (if the steps are all according to our operations, then enter the same path)

Check the Jupyter-lab installation path: which jupyter-lab
The configuration file path refers to the path to the configuration file generated above.

WorkingDirectory: The working directory of Jupyter-lab, which can be changed by yourself



The screenshot shows a terminal window on a Raspberry Pi. The top bar indicates the user is 'pi@raspberrypi' and the time is 09:40. The terminal window title is 'pi@raspberrypi: ~'. The nano editor is open, editing the file '/etc/systemd/system/jupyter.service'. The content of the file is as follows:

```
GNU nano 7.2 /etc/systemd/system/jupyter.service *
[Unit]
Description=Jupyter Lab
After=network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/jupyter-lab
config=/home/pi/.jupyter/jupyter_lab_config.py --no-browser
User=pi
Group=pi
WorkingDirectory=/home/pi
Restart=always
RestartSec=10
[Install]
WantedBy=multi-user.target
```

The bottom of the terminal shows the nano editor's command shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, ^_ Go To Line.

- jupyter.service service

Enable auto-start at boot

```
sudo systemctl enable jupyter
```

Disable auto-start at boot

```
sudo systemctl disable jupyter
```

Start service

```
sudo systemctl start jupyter
```

Out of service

```
sudo systemctl stop jupyter
```

Check service status

```
sudo systemctl status jupyter
```

Enter the enable jupyter.service service self-start and start service commands in the terminal and then restart the Raspberry Pi system.

```
pi@raspberrypi:~$ sudo nano /etc/systemd/system/jupyter.service
pi@raspberrypi:~$ sudo systemctl enable jupyter
Created symlink /etc/systemd/system/multi-user.target.wants/jupyter.service → /etc/systemd/system/jupyter.service.
pi@raspberrypi:~$ sudo systemctl start jupyter
```

After completing the above steps, you can access the LAN without typing jupyter lab in the terminal!