Install JupyterLab

1.Introduction to JupyterLab

JupyterLab is an interactive development environment and the next generation product of Jupyter Notebook. It integrates more features, supports plugin extensions, and can be run and operated through web pages. It is simple, convenient, and powerful, making it a very worthwhile code editing tool.

2.Switch to root user

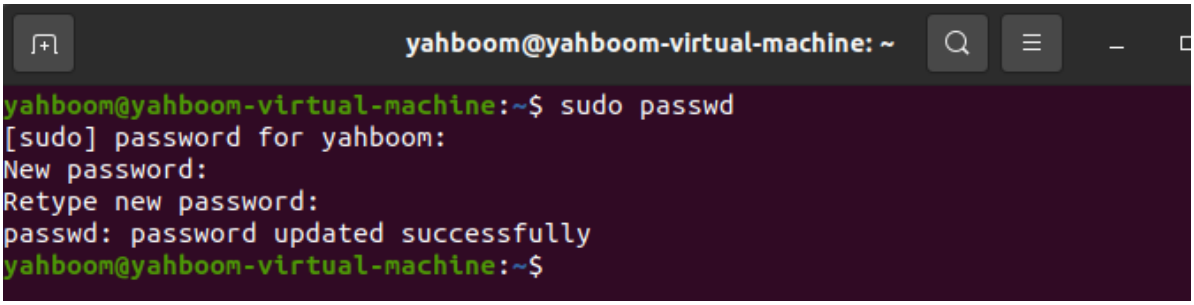
Installing jupyterlab requires the use of a root user, and the root user of the system does not have a password by default and cannot be switched.

Therefore, a password needs to be set for the root account before it can be used.

Note: The password set for root here must be remembered, preferably consistent with the user's password, so as not to be easily forgotten

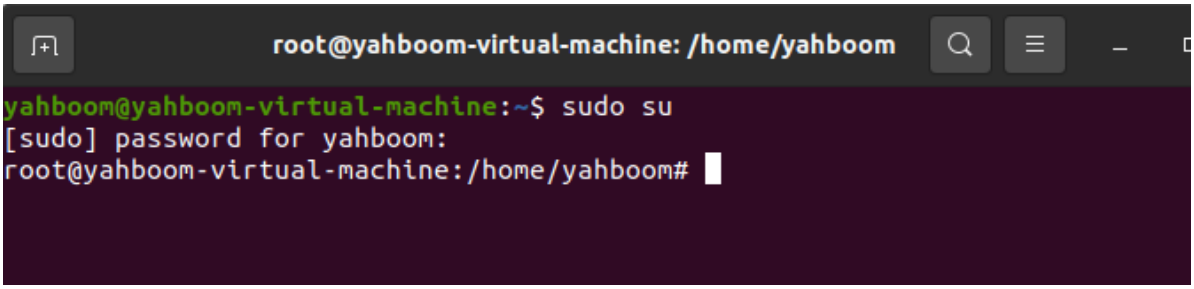
1. Enter the following command to set the password for root, and then enter the same password twice to confirm.

```
sudo passwd
```



2.Switch to root user

```
sudo su
```



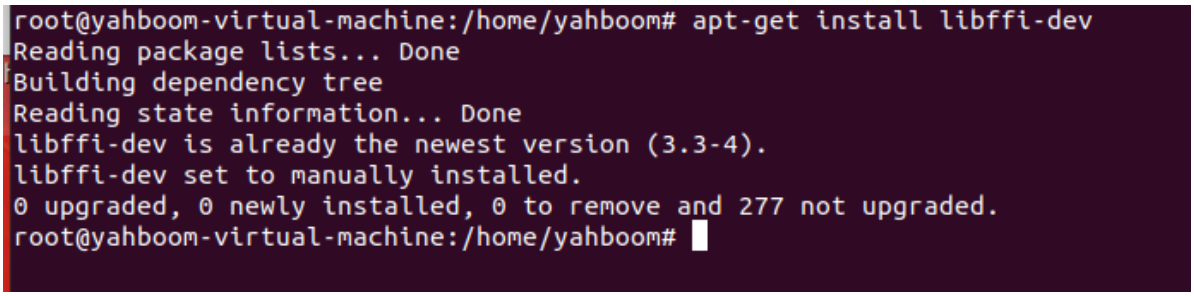
From the above figure, it can be seen that the user has switched to root.

And the $symbol before the editing command has changed to a # symbol.

3. Install JupyterLab

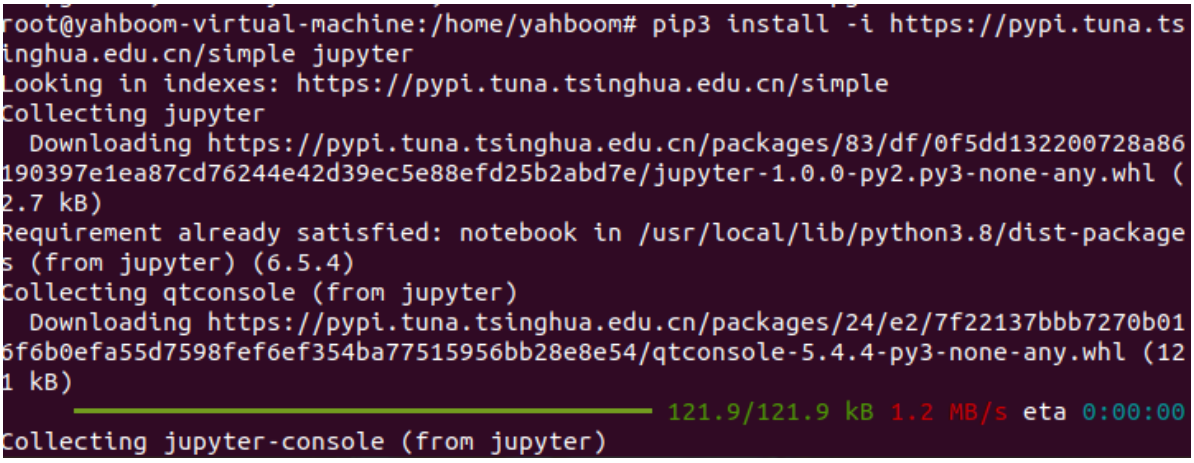
1.Install ffi library

```
apt-get install libffi-dev
```



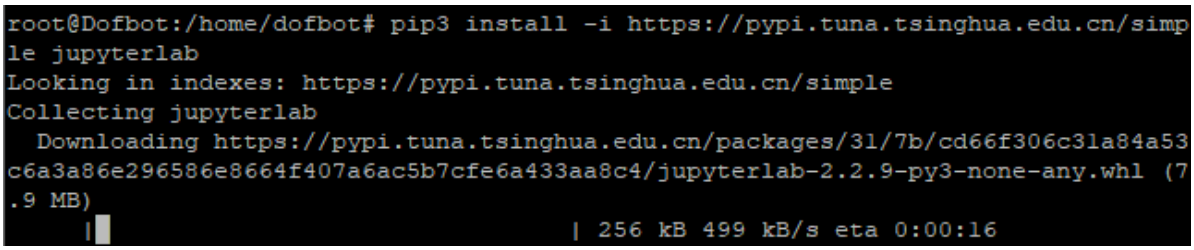
2.Install jupyter

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



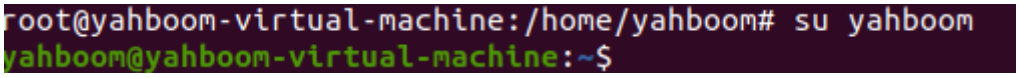
3.Install jupyter lab

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



4.Switch back to regular users after installation is completed

```
su jetson
```



4.Configure JupyterLab

1.Generate configuration file

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

```
yahboom@yahboom-virtual-machine:~/.jupyter$ jupyter notebook --generate-config
Overwrite /home/yahboom/.jupyter/jupyter_notebook_config.py with default config?
 [y/N]y
Writing default config to: /home/yahboom/.jupyter/jupyter_notebook_config.py
yahboom@yahboom-virtual-machine:~/.jupyter$
```

2. Use iPhone to generate the login password for jupyterlab.

   ipython

   In [1], enter: from notebook.auth import passwd

   In [2], enter: passwd()

Then, enter the same password twice and press enter to confirm. Note that the password here is used to log in to the jupyterab interface. For ease of memory, it can be kept consistent with the user's password.

Then the system will output the ciphertext of the password and copy it as a whole. The ciphertext generated by each password may be different.

Please copy the actual output ciphertext.

In [3], enter: exit()

```
yahboom@yahboom-virtual-machine:~/.jupyter$ ipython
Python 3.8.10 (default, May 26 2023, 14:05:08)
Type 'copyright', 'credits' or 'license' for more information
IPython 8.12.2 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from notebook.auth import passwd

In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$13Ta0o+Xl7FAkUAS5FCqkg$yuuIX/MJ9
WIRw2g8wWXv/q4KsDMqFOWcURRi9mfz6M0'

In [3]: exit()
yahboom@yahboom-virtual-machine:~/.jupyter$
```

3.Compile the configuration file for jupyter

nano ~/.jupyter/jupyter_notebook_config.py

Go directly to the back and add the following content:

c.NotebookApp.ip = '0.0.0.0'

c.NotebookApp.open_browser = False

c.NotebookApp.password = The entire password ciphertext

c.NotebookApp.port = 8888

```
this value
#          will correspond to the value of the Gateway url with 'ws' in place of
'http'.  (JUPYTER_GATEWAY_WS_URL env var)
#  Default: None
# c.GatewayClient.ws_url = None

#------------------------------------------------------------------------------
# TerminalManager(LoggingConfigurable) configuration
#------------------------------------------------------------------------------
##

## Timeout (in seconds) in which a terminal has been inactive and ready to be
culled.
#          Values of 0 or lower disable culling.
#  Default: 0
# c.TerminalManager.cull_inactive_timeout = 0

## The interval (in seconds) on which to check for terminals exceeding the
#  inactive timeout value.
#  Default: 300
# c.TerminalManager.cull_interval = 300

c.NotebookApp.ip = '0.0.0.0'

c.NotebookApp.open_browser = False

c.NotebookApp.password =
'argon2:$argon2id$v=19$m=10240,t=10,p=8$13Ta0o+Xl7FAkUAS5FCqkg$yuuIX/MJ9WIRw2g8wWXv/-
q4KsDMqFOWcURRi9mfz6M0'

c.NotebookApp.port = 8888
```

Finally, save with Ctrl+S and exit with Ctrl+X.

5. Install the jupyteralab plugin

1. Install nodejs and npm

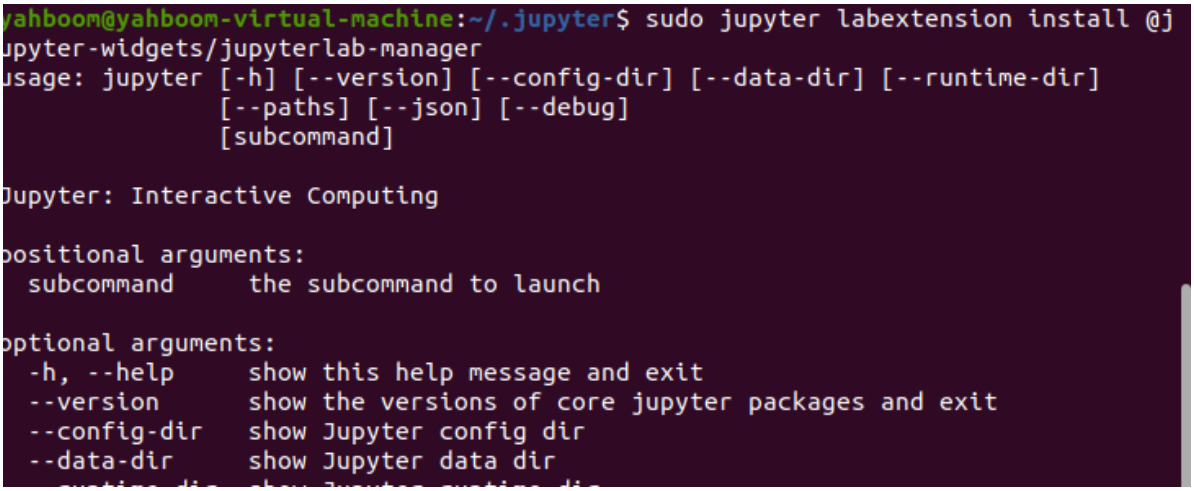
```
sudo apt install nodejs npm
```

```
yahboom@yahboom-virtual-machine:~/.jupyter$ sudo apt install nodejs npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
nodejs is already the newest version (10.19.0~dfsg-3ubuntu1).
npm is already the newest version (6.14.4+ds-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 277 not upgraded.
yahboom@yahboom-virtual-machine:~/.jupyter$
```

2.Install the jupyter widget extension.

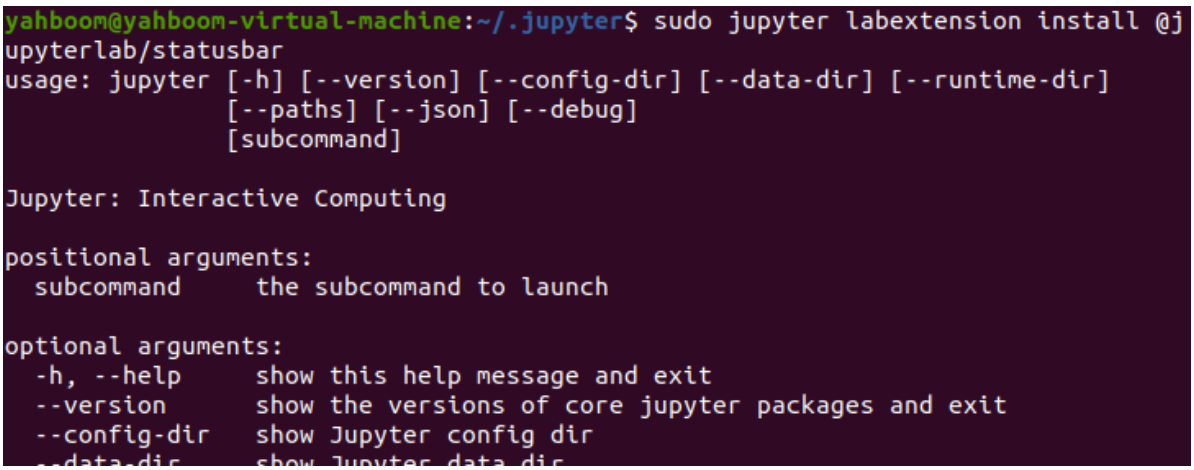
Due to the need for download and compilation, it takes a long time to run and may cause errors. If an error occurs, run the installation again.

```
sudo jupyter labextension install @jupyter-widgets/jupyterlab-manager
```



3. Install the Statusbar plugin

```
sudo jupyter labextension install @jupyterlab/statusbar
```

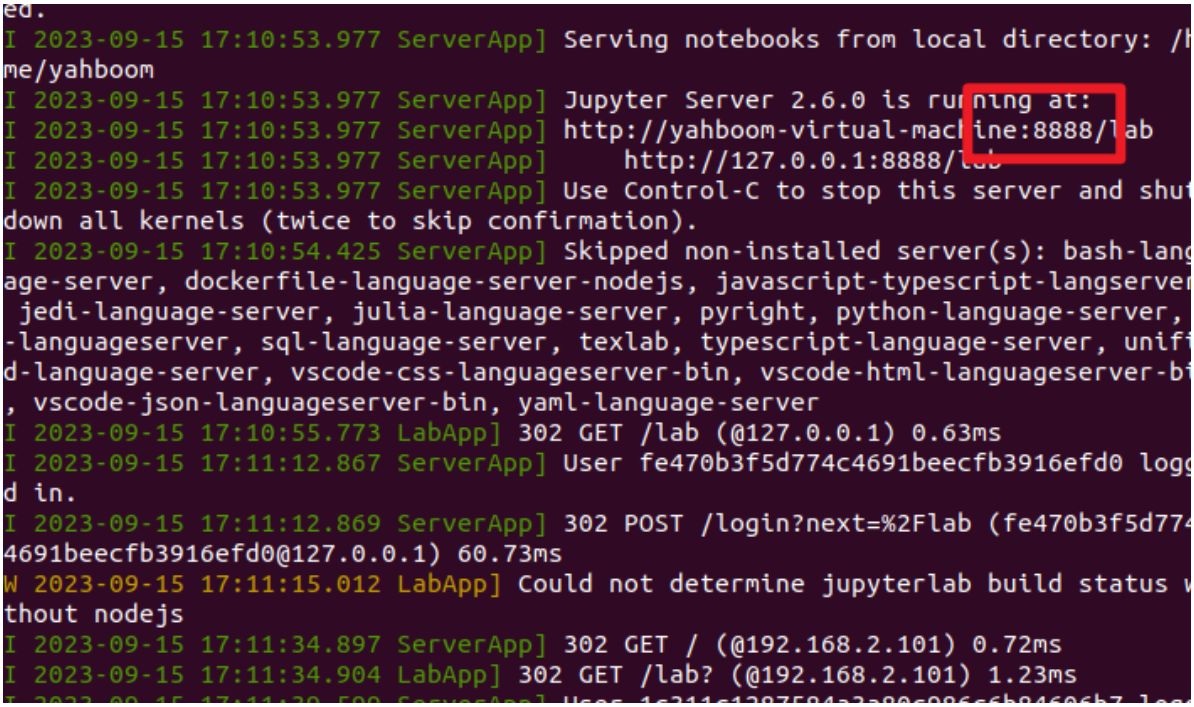


4. The installation of jupyterlab has been completed at this point.

6. Start jupyterlab

1. Enter the directory where you want to run the code, taking entering the root directory as an example.
2. Open jupyterlab, open the terminal, and enter the following command

```
jupyter lab
```



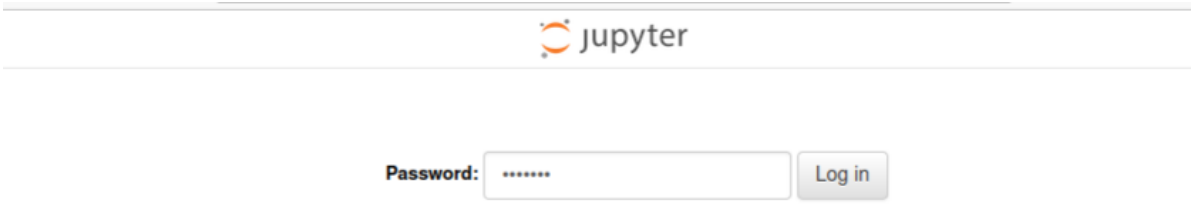
This port number is the one we need to access, which defaults to c. in the previous jupyter configuration file The port number referred to by NotebookApp. port.

If an additional jupyteralab service is opened, the port number will automatically+1, which can distinguish different jupyteralab services.

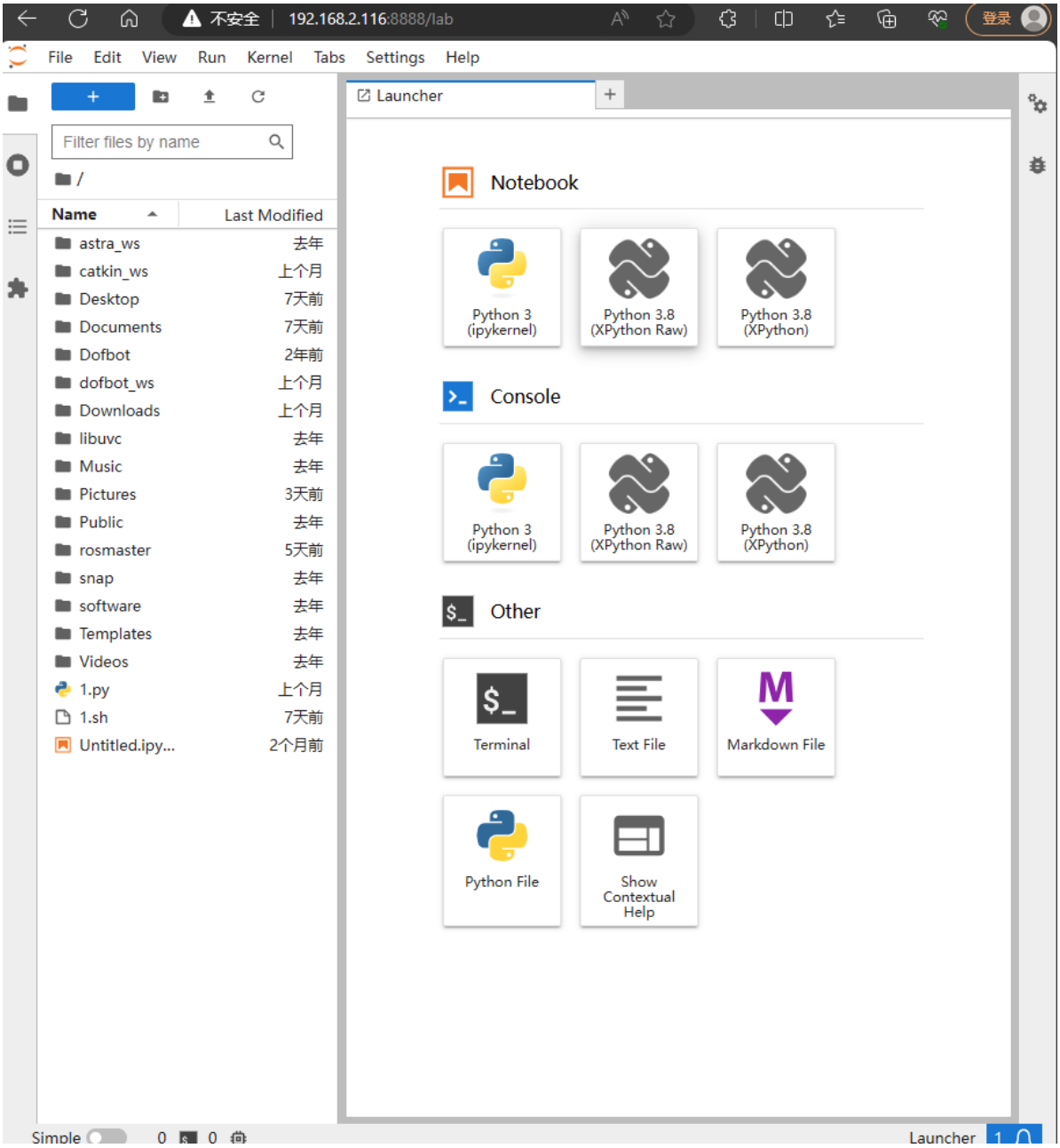
7. Remote access to jupyterlab

1. Open a browser on the computer (Chrome or Firefox browser is recommended), and then enter the IP: port number of the virtual machine. Taking IP 192.168.2.116 and port 8888 as an example.

   The interface for jupyterlab will pop up and require a password to be entered. Simply fill in the login password for jupyterlab that has been set up above.



2. Upon seeing the following interface, it indicates that the remote login to jupyterlab has been successful, and a new Python 3 program can be created to run.



8. Exit jupyterlab

Press Ctrl+C twice on the terminal where jupyterab was just opened to exit jupyterab.