Setting up a Jupyter Lab remote server

2 minute read

If you haven't yet used Jupyter Lab I highly recommend it. In any case, this post is for both Jupyter Lab and Notebook users who want to set up a remote server. I've done this a few times and promptly went and forgot how. So these instructions are primarily for me, but if it ends up helping someone else, then great!

So why do you need to setup a remote Jupyter server, you ask? Well you might find yourself in a situation where you have a powerful GPU cluster, that you train your deep learning models on, but also need the flexibility of an interactive notebook. Since servers usually run headless, ie. without a graphical user interface, you'll need to access Jupyter remotely via the internet or an SSH connection. I imagine this is a rather common scenario, given the rising popularity of MOOCs (http://course.fast.ai/) that require setting up cloud instances. I personally use my university's cluster, but it's almost identical to the scenario explained above except for some issues with port forwarding.

The instructions shown below is the minimum subset of the more comprehensive documentation available here (http://jupyter-notebook.readthedocs.io/en/stable/public_server.html#notebook-server-security), that is still secure.

Step 1: Password Setup

```
$ jupyter notebook --generate-config
$ jupyter notebook password
Enter password:  ****
Verify password: ****
[NotebookPasswordApp] Wrote hashed password to /Users/you/.jupyter/jupyter_notebook_config.json
```

Use this hashed password when editing jupyter_notebook_config.json in step 3

Step 2: Using SSL for Encrypted Communication

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mycert.pem -out mycert.pem
```

The above command is slightly different from the one in official Jupyter docs, which didn't work for me for some reason.

Step 3: Running a Public notebook server (via the web)

Open /Users/you/.jupyter/jupyter_notebook_config.py with your favourite text editor and edit the following

```
# Set options for certfile, ip, password, and toggle off
# browser auto-opening
c.NotebookApp.certfile = u'/absolute/path/to/your/certificate/mycert.pem'
c.NotebookApp.keyfile = u'/absolute/path/to/your/certificate/mycert.pem'
# Set ip to '*' to bind on all interfaces (ips) for the public server
c.NotebookApp.ip = '*'
c.NotebookApp.password = u'sha1:bcd259ccf...<your hashed password here>'
c.NotebookApp.open_browser = False

# It is a good idea to set a known, fixed port for server access
c.NotebookApp.port = 9999
```

Step 4: Run Jupyter Lab/Notebook

Jupyter Lab and Notebook share the same configuration files, so there is no need to follow different processes for each. To start the server, simply run

```
$ jupyter lab
or
$ jupyter notebook
```

Step 5: Open Jupyter Lab/Notebook on your local machine

Run the following on your local machine to start an SSH connection to the server, in the background. The specifics of the command will differ for you based on your configuration.

```
$ ssh -N -f -L 8888:localhost:9999 user@domain.com #Change the specifics as required
```

Open a web-browser on your local computer and navigate to *https://localhost:8888/* (*https://localhost:8888/*) or whatever you configured it to be, with an emphasis on *https*.

That's it! Hope this helps.

Keep in mind that this tutorial is for single user setups. For multi-user servers, look at JupyterHub (https://github.com/jupyterhub/jupyterhub)

Tags: Jupyter (https://agent-jay.github.io/tags/#jupyter)

Updated: March 07, 2018

LEAVE A COMMENT

Gilly • 10 days ago

Thank you for this! work fine for me. One question, do I have to run the command `ssh -N -f -L 8888:localhost:9999 user@domain.com #Change the specifics as required` every time I want to connect to the server? If so how do I create a permanent connection?

• Reply • Share ›

Mo • 2 months ago

I receive the error "channel 2: open failed: connect failed: Connection refused" when trying to open the https://localhost:... Any idea as to why that might be?" I am allowed to use the remote server and can access via remote desktop, but would prefer SSH.

• Reply • Share ›

ibmua • 3 months ago • edited

Actually,
jupyter lab
on the host and then
ssh -N -f -L 8888:localhost:8888 user@server
- ssh tunnel on the client
is all you need.
You don't need any additional security stuff, as by default jupyter only listens to localhost connections. You can verify this via
lsof -i -P -n | grep LISTEN
after turning Jupyter on.

• Reply • Share ›

Shih Hwa Lai • a year ago

Hi, I cannot start the server with the "c.NotebookApp.ip" set to '*', I successfully start the server with that setting set to '0.0.0.0'

• Reply • Share ›

Alber Blanco Garcés ➚ Shih Hwa Lai • 9 months ago

Same but with "c.NotebookApp.ip" set to '127.0.0.1'

Great post anyway, very useful and informative. Thanks!

• Reply • Share ›