

5 easy steps to start editing python notebooks over SSH

Willem Meints

When you work with Linux machines on Amazon EC2 or Azure over SSH and want to edit interactive python code you've got a challenge. There is vim and other console based editors, but that might not be what you want. Here's how you can edit python code in your browser over SSH in 5 steps.

Step 1: Install jupyter on your remote machine

Jupyter is a webbased python editor that works with notebooks. Notebooks are interactive pieces of python mixed with markdown code.

It's cool because you can write several fragments of code and execute them one after another. When one of the fragments doesn't work you can simply edit and run it again. The global variables defined in the other fragments are still there.

Want some documentation with your code? Add a markdown fragment and go wild. The cool thing about notebooks is that you're not only editing python code and writing docs. It also visualizes graphs generated using pyplot.

The fact that you can do this kind of stuff in a notebook makes it ideal for creating experiments. It may not be suitable for production, but it is pretty much the only way I want to experiment when setting up prototypes.

You can install jupyter using the following command:

```
pip install jupyter
```

Execute this command on the remote machine you want to work on.
Once installed you can start editing notebooks.

Step 2: Start the jupyter notebook editor

Typically you'd start jupyter using the command `jupyter notebook`. But this opens up the browser, which is not available

through an SSH session. So instead run the following command to start the jupyter notebook server without opening a browser:

```
jupyter notebook --no-browser --port=8080
```

Notice the port setting. Keep this port in mind when executing the next step.

Step 3: Setup a SSH tunnel to your remote machine

To access the notebook on your remote machine over SSH, set up a SSH tunnel to the remote machine using the following command:

```
ssh -N -L 8080:localhost:8080 <remote_user>@<remote_host>
```

This command opens up a new SSH session in the terminal. I've added the option -N to tell SSH that I'm not going to execute any remote commands. This ensures that the connection cannot be used in that way, see this as an added security measure.

I've also added the -L option that tells SSH to open up a tunnel from port 8080 on the remote machine to port 8080 in my local machine.

Step 4: Open the notebook in your browser

Now that you have a tunnel to the remote machine, open up your browser and navigate to `http://localhost:8080/`. This will fire up the jupyter notebook web interface.

Step 5: Enjoy your python notebooks

Time to start hacking! Keep in mind that the notebooks you save will be stored on the remote machine. So if you want to backup the notebook files you will have to either push them to git or download them to your local machine.

Enjoy!