# Using Jupyter Notebooks on Atmos

Contributors: Barron H. Henderson and Ben Murphy and content from Ed Anderson
Date: 2020-06-23
Last Updated: 2020-07-04

## General Overview

Jupyter can be run on atmos as a server, which can then be accessed graphically in several ways. The conventional approach is to forward a linux application window to your desktop computer (Figure 1, option b), but you can also connect directly from your desktop computer and access the Jupyter application via a web browser (Figure 1, option a). This latter approach requires setting a Jupyter password and starting a "server process" (i.e. a Jupyter notebook) on atmos and then setting up a port tunnel. Port tunneling is good because it runs in the background, does not require X11, is therefore much faster, and is thus the preferred method. Both methods actually use ports, so there is a brief explanation of ports and how to set one up. Then, an explanation follows on how to start a Jupyter Server. Finally, a description is given of how to configure a tunnel for that Jupyter Server instance.

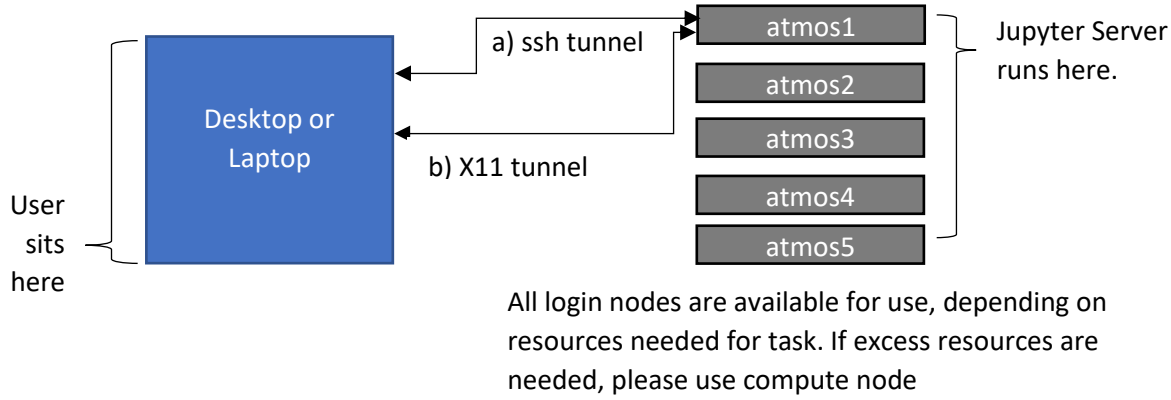


*Figure 1 Schematic of relationship between atmos and desktop or laptop computer*

To summarize, there are three basic steps shown in the list below.

- Set a Jupyter password (*1 time only, see Jupyter Password*)
- Start a "Jupyter Server process" (*once in a while, see Jupyter for Tunnel*)
- Start a Tunnel (*can be automatic, see Setup a Tunnel*)

The first step is setting a password, which is done once and need not be done again (see *Jupyter Password*). The second is starting your own "Jupyter Server" and needs to be done only if you do not already have a "Jupyter Server" running (see *Jupyter Server for Tunnel*) -- Mine runs for weeks without needing to restart. The third is starting a tunnel on your machine and can be made to start automatically with MobaXterm or be scripted to start each day (see *Setup a Tunnel*).

# Table of Contents

General Overview ..... 1
Choose a PORT ..... 2
Jupyter Password ..... 2
Start Your Own Jupyter Server ..... 2
a) Jupyter Server for Tunnel (faster, better, stronger…work it, make it, do it, makes us) ..... 3
b) Jupyter Server for X11 Web Browser (very slow) ..... 3
Setup a Tunnel ..... 3
a) Tunneling Support: Mobaxterm ..... 3
b) Tunneling Support: PuTTy ..... 4
c) Tunneling Support: Shell ..... 5

## Choose a PORT

If using tunneling, you will need to connect a port on your computer to a port on atmos. The port should be in the custom range: between 8888 and higher. It must be identical to the port your Jupyter server is using.

If EPA really likes notebooks, we could setup a more formal server environment and then people wouldn't need their own port.

## Jupyter Password

To use Jupyter via tunnel, you need a server password. You set this up once, and it never needs to change. I suggest setting one up even if you plan to use X11. It requires only one command shown below.

```
$ /work/ROMO/anaconda_envs/basic38/bin/jupyter notebook password
```

You will be prompted for a password, which you can enter when connecting to a Jupyter Server.

## Start Your Own Jupyter Server

Open a terminal on atmos and start a "background" Jupyter process. There are several Jupyter installations. If you want libraries for Air Quality Management, I suggest you start with mine. I am using an anaconda installation available at:

```
/work/ROMO/anaconda_envs/basic38
```

To use that installation, you can run with a linux browser or with a tunnel to your local desktop.

Both methods start by calling commands on atmos, so open a terminal.

### a) Jupyter Server for Tunnel (faster, better, stronger…work it, make it, do it, makes us)

This ssh tunnel approach is *very* fast, and requires a small additional step. For use with tunneling, open a screen session and invoke the 'no-browser' option. You only have to run this command once per session (I run once a week or so) unless the process dies or the server is rebooted.

```
$ screen
```

```
$ /work/ROMO/anaconda_envs/basic38/bin/jupyter lab --no-browser --port=9308 --ip=localhost
```

Check to see that the port it started on is the one you asked for – I chose port number 9308 here, and you may choose any port larger than 8888. If the one you ask for is in use, Jupyter will try the next available one. Once you are confident of the port, use the key combinations Ctrl+a then Ctrl+d to "detach" from screen. To check that the process is still running in the background, `screen -ls` will show your detached process. This Jupyter process will run until it either dies or the computer resets. You can reattach `screen -r <ID>` where the ID is from `screen -ls` and disconnect again (Ctrl+a Ctrl+d).

### b) Jupyter Server for X11 Web Browser (very slow)

The x11 browser is very slow over VPN. I do not recommend this method, but it is the easiest to setup. If you are using the Tunnel option, skip to "Setup a Tunnel." If there is a good reason to use the X11 browser option, you will be running the server interactively so you do not need screen. You will need to select a port and a web browser. The atmos system will not choose a browser, so you have to specify it. The command below uses port 9308 and the firefox browser.

```
$ /work/ROMO/anaconda_envs/basic38/bin/jupyter lab --browser=firefox --port=9308
```

The browser takes a while to open, and has not been very responsive for any user to date, but your setup may be different.

## Setup a Tunnel

If using a Jupyter-with-tunnel-option configuration, you will need to setup a tunnel. This can be done easily in PuTTy, MobaXterm, or from any terminal emulator (Cygwin, etc). The three sections below explain how to setup that tunnel.

### a) Tunneling Support: Mobaxterm

In Mobaxterm, there is a Tunnel configuration interface. Figure 2 below has the Tunnel button behind a configuration screen. Click the Tunnel button, and then the tunnel manager will open. Make a "New SSH tunnel" and configure it similar to the picture below. I have selected port 9308 on the server and client side. I am using the atmos1.nesc.epa.gov login node. I am using my username (bhenders) and the ssh call must use port 22 for the ssh call.

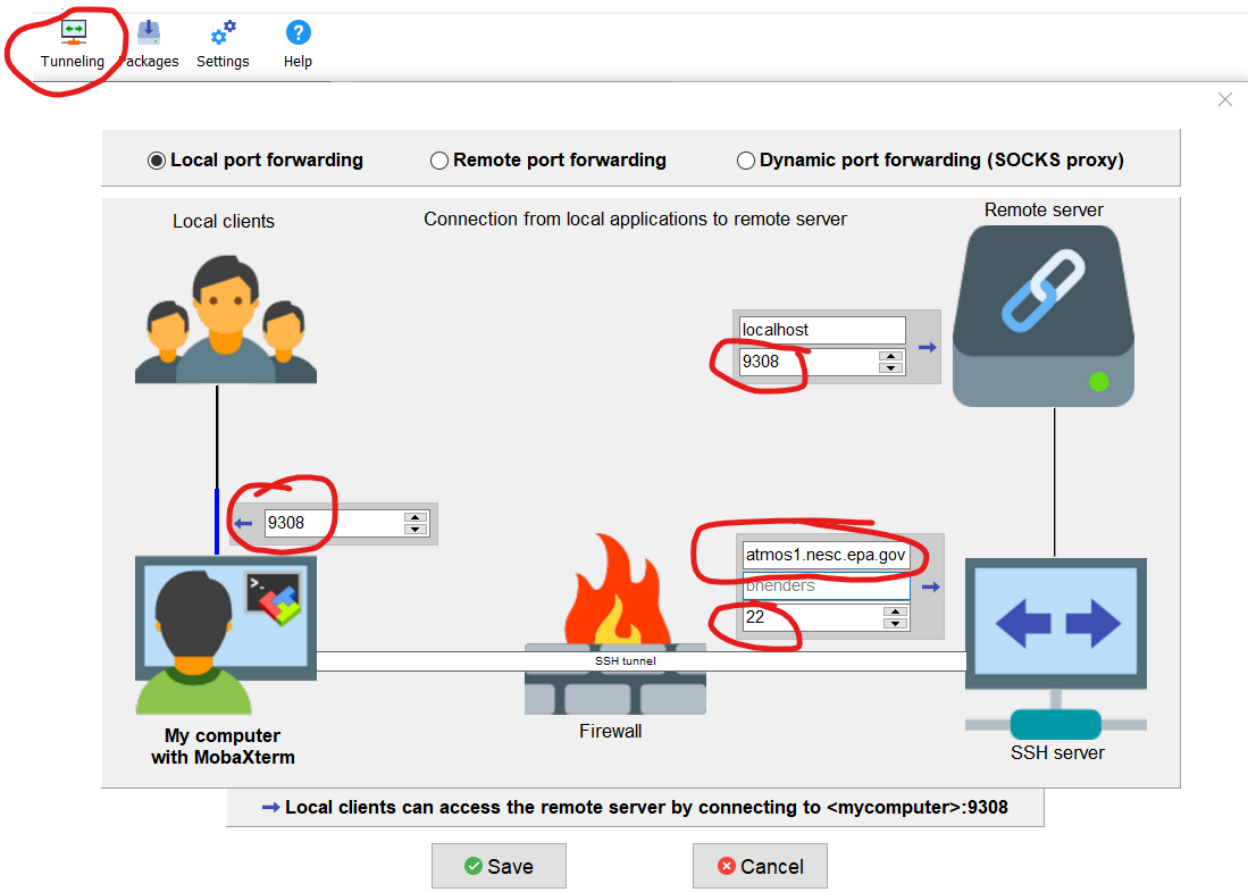


*Figure 2 MobaXterm Tunnel Configuration example.*

## b) Tunneling Support: PuTTy

<Copied from an email from Ed>

In your SSH client, you will need to configure an SSH tunnel. If you use PuTTY, you probably already have entries for the atmos login nodes. Select one of these and go to Connection->SSH->Tunnels. Choose a port number and enter it in the box labelled "Source port", and put "localhost:<port>" in the box labelled "Destination", then click "Add". For example, I created a tunnel using the port 9308 in example Figure 3:

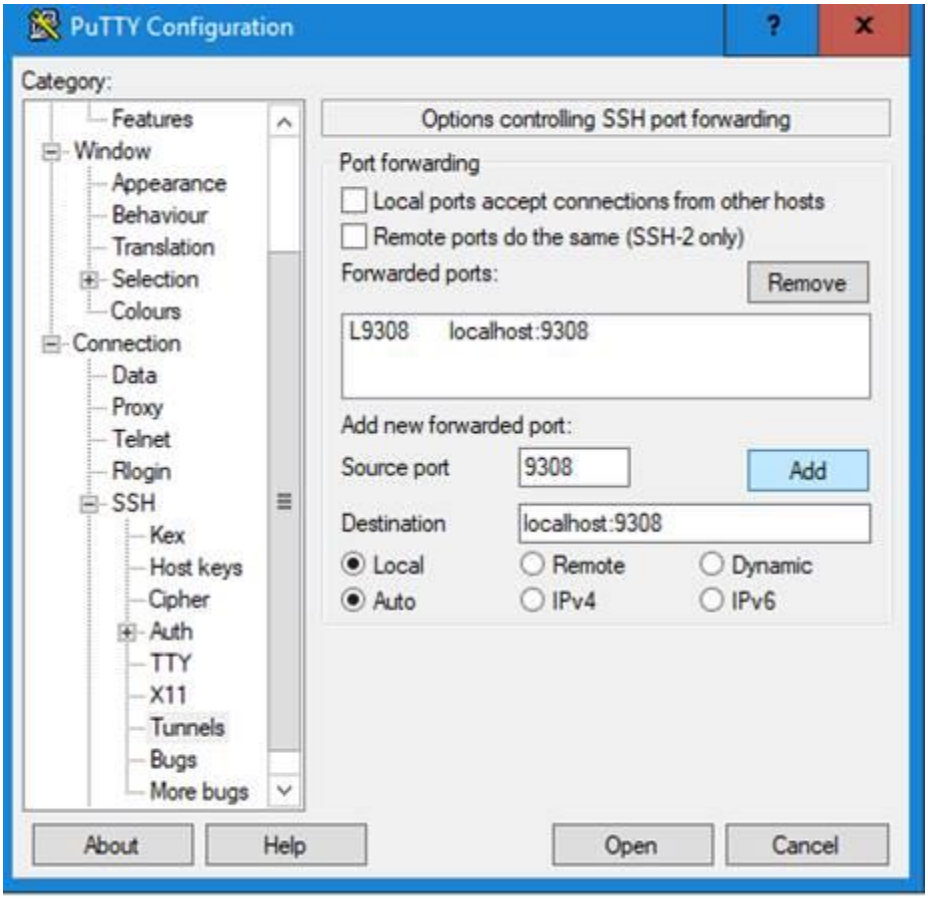


*Figure 3 PuTTy Tunnel Configuration example.*

Go back to Session, give it a name like "atmos2tunnel", and click Save. Now use this session to open a window to atmos.

### c) Tunneling Support: Shell

The line below can be run from a terminal shell to start a tunnel.

```
ssh -N -f -L localhost:<PORT>:localhost:<PORT> <USERNAME>@atmos1.nesc.epa.gov
```

This is using the same port on server and local sides. The username should be your atmos username.

*Example Shell Workflow (Start your Jupyter session in two keystrokes, from Ben):*

On atmos, put a shell script called, for example, “run_jupyter.csh” in your home directory. It’s contents could be

```
#!/bin/csh -f
      if ($#argv == 1 ) then
            Set port = $1
      else
```

```
            set port = 8585
      endif
       /work/ROMO/anaconda_envs/basic38/bin/jupyter-lab --no-browser --
port=${port} --ip=localhost
```

Now you can start Jupyter immediately when you log into atmos by typing `./run_jupyter.csh 9010` The Jupyter server will usually take a second or two to start up.

Next, get your MobaXterm environment on your desktop finished off. I use this script to allow me to enter the atmos login-node name and the port number and it starts a Google Chrome window for me to get going on my session. I named it "jupyter_browser.csh"

```
#!/bin/bash

# Detect Server Name and use default if it's absent
if [ $# -eq 2  ]; then
   server=$1
   port=$2
else
   echo "Improper Command Line Arguments were given."
   echo "syntax: jupyter_browser.sh <server name> <port number>"
   exit 22
fi

# Open Chrome Browser
echo "Giving Jupyter 10 seconds or so to start up on Atmos..."
sleep 5
echo "Giving Jupyter 5 seconds or so to start up on Atmos..."
sleep 3
echo "Giving Jupyter 2 seconds or so to start up on Atmos..."
sleep 1
echo "Giving Jupyter 1 seconds or so to start up on Atmos..."
sleep 1
echo "Starting Jupyter Browser..."

ssh -f -L localhost:${port}:localhost:${port} bmurphy@${server}.nesc.epa.gov
sleep 10; [path-to-google-chrome]/google-chrome /new-window
http://localhost:$port
```

There are a couple of features about this script I really like:

1) There's some junk at the front that helps me remember what info I need to give to make it go. If your server from above is running on atmos3, then execute it like so:
   ```
   ./jupyter_browser.csh atmos3 9010
   ```
2) The timing block gives the Jupyter server just a few moments to get going before trying to start the tunnel. If Moba tries to start the tunnel and the port is open yet, then you might have to try again, or even kill the Jupyter server and restart it on a different port.
3) The specific SSH options and `sleep 10` command are cool. They make sure that the tunnel stays open as long as the Google Chrome browser is open. Once I close the browser, the tunnel shuts down and the port is freed up. No waiting for synchronization, etc.

4) The “new-window” option means Google Chrome will start a brand new window for you and your Jupyter Adventure.
5) The server name and port number are automatically passed to the ssh command and to Google Chrome, so you don’t have to enter it 1,000,000 times.

   To Do: Be sure to update to your username in the ssh call. Also replace [path-to-google-chrome] with your local path to your Google Chrome executable. Alternatively, I just created a symbolic link to Google Chrome in my MobaXterm home directory and replaced [path-to-google-chrome] with “./”