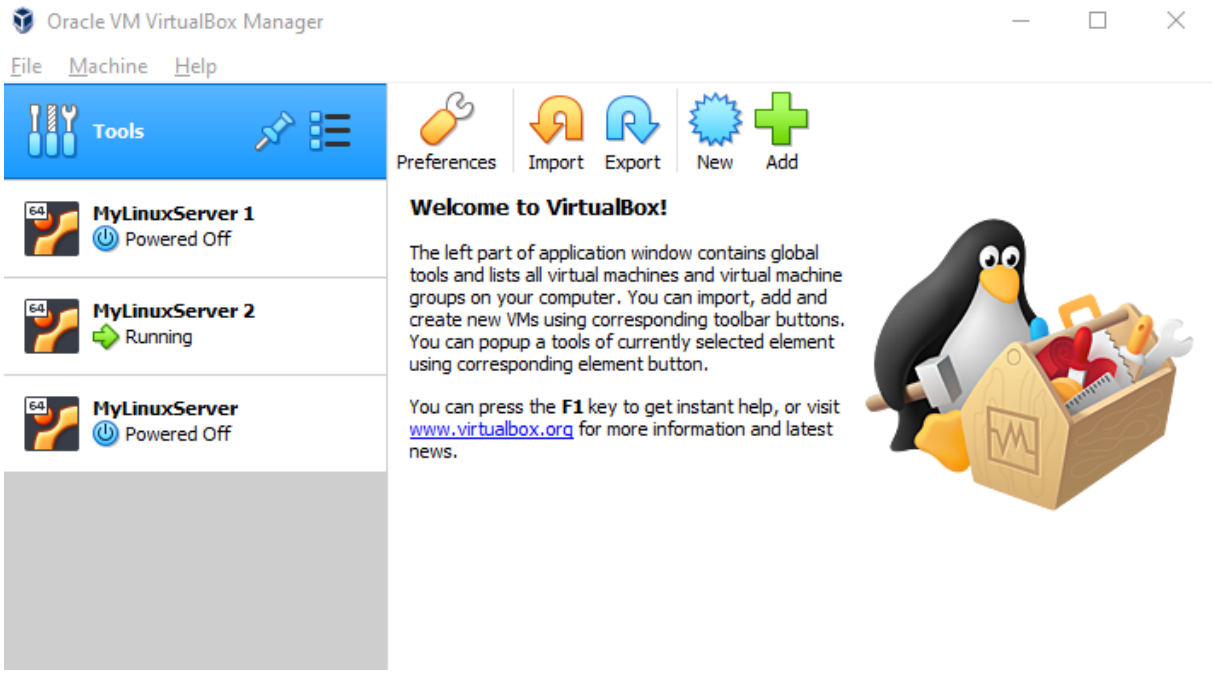
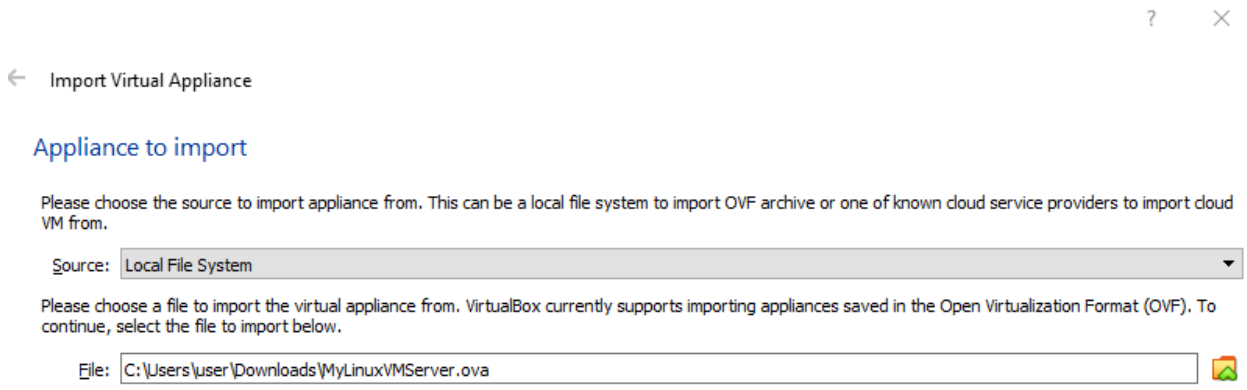
1. Download VirtualBox for Windows (or OS X). VirtualBox is a virtualization program that allows you to run Virtual Machines.
2. Also download the .ova file that is uploaded. The .ova is the virtual machine image file and Please access the image file here:

   https://drive.google.com/file/d/1QxeBSAAUduCeSreEDpNismqMwxN0SaXP/view?usp=sharing

3. Install VirtualBox (https://www.virtualbox.org)
4. Open VirtualBox.
   3.1 Click on Tools-> Import. It should bring you to the Import Virtual Appliance page



3.2 Import the Virtual Image file you downloaded from Luminus. This has already been configured with a Linux installation



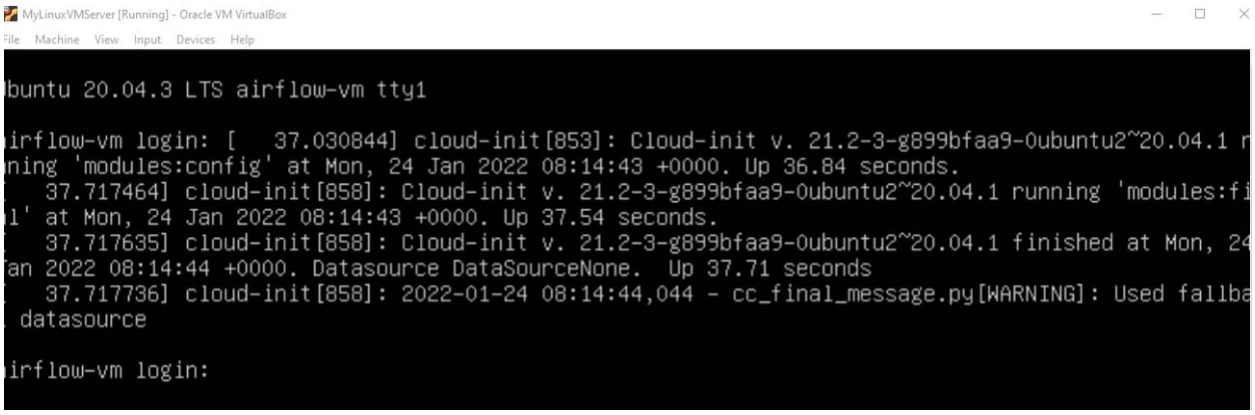
4.3 Click Next. It will take some time to import a 1.8 GB image file. This Linux image has been configured to use up to 4 GB of your RAM. You may adjust it downwards if you have problems but it is recommended you leave 4 GB for the Linux machine especially if you're going to use it for your project [I suspect you will need more if you plan to run more intensive data processing applications later]

4.4 Once it's done, you should see MyLinuxVMServer on the VirtualBox main screen. Double click it or select it and press "Start"

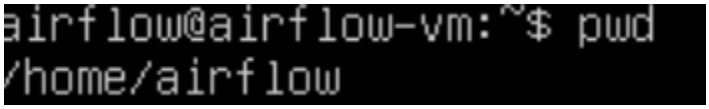
4.5 You may resize VirtualBox if the scaling is not suitable for your screen: Machine -> Display -> Scale Factor

4.6 Hit enter. You should see a prompt for the login of the VM machine. The VM is named airflow-vm.

The username/password combination has been set to airflow/airflow. Use the credentials to login.



4.7 Type pwd to check that you're in the home root directory folder.

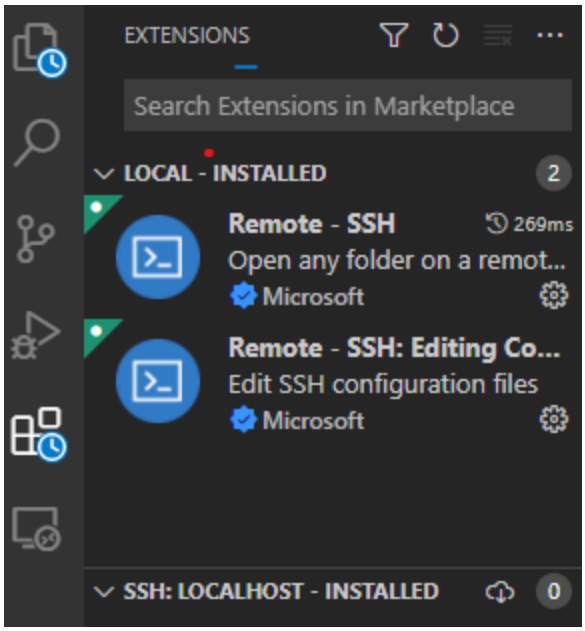


4.8 Congratulations! You have just accessed your first Linux virtual machine on VirtualBox!

5. Optional, access VM through Visual Studio code via SSH

The above steps are optional but recommended for developers looking to develop on Linux VMs from their local machines. The following sequence of steps describes the steps if you wish to remote access a running VM (that means you still have to run the VM first) from Visual Studio Code

4.1 Run Visual Studio Code. On the main screen, look for extensions. We want to install an extension "Remote – SSH". Download it and install it.

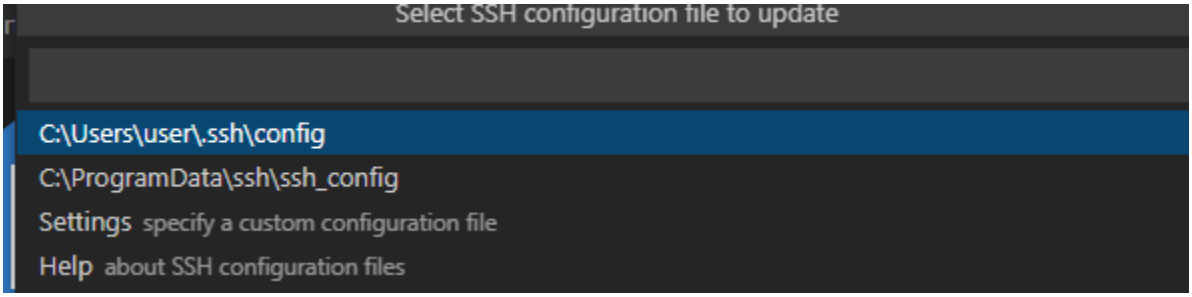


4.2 You should see “Remote Explorer” on the left pane below Extensions if you installed SSH correctly. Click on it.

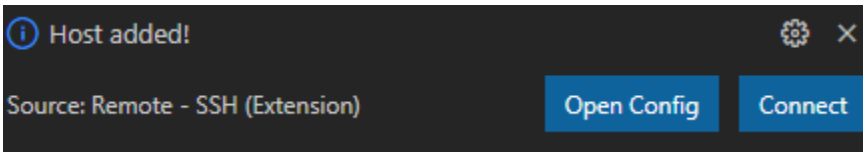
4.3 On the main window, Click + to add a new SSH connection. This allows you to set up a remote access connection. Enter SSH Connection Command Prompt, type ssh –p 2222 airflow@localhost to establish a connection to your VM.

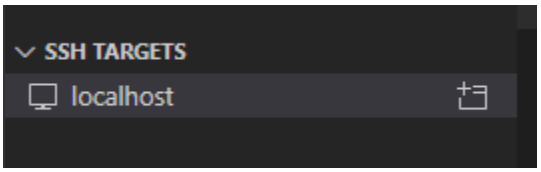
(The reason this works is because I had preconfigured basic NAT in the VM to redirect port 2222 to port 22 of host – Port 22 is the standard port of SSH, so attempts to reach port 2222 in the local computer will be redirected to port 22 of the remote computer (Port 22 is the standard port for SSH connections. Well this is slightly tedious to explain completely in this post and it is not needed for you to understand it now, but if you’re really curious you can ask me in a regular consult how it works)

4.4 You will be prompted where to save your SSH config file. Click the first one it comes up.



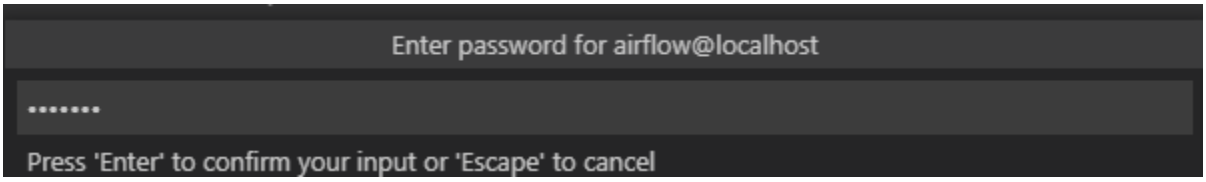
4.5 On the bottom right, it should show Host added. Just connect to it.





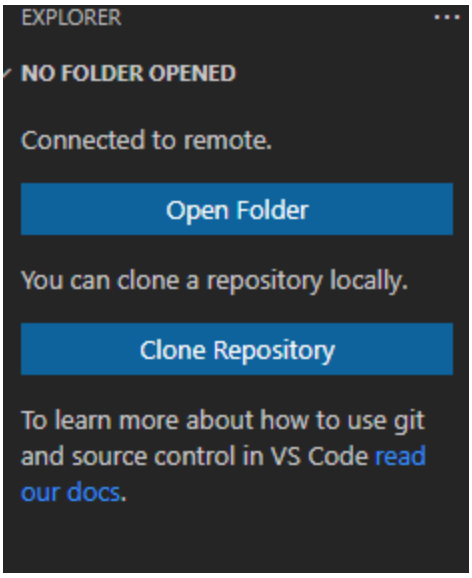
Alternatively, on the left pane, right click localhost and click “Connect Host in New/Current Window” (doesn’t matter what you choose, you can choose new window if you prefer to keep it in another window)

4.6 Type in the password for the SSH connection



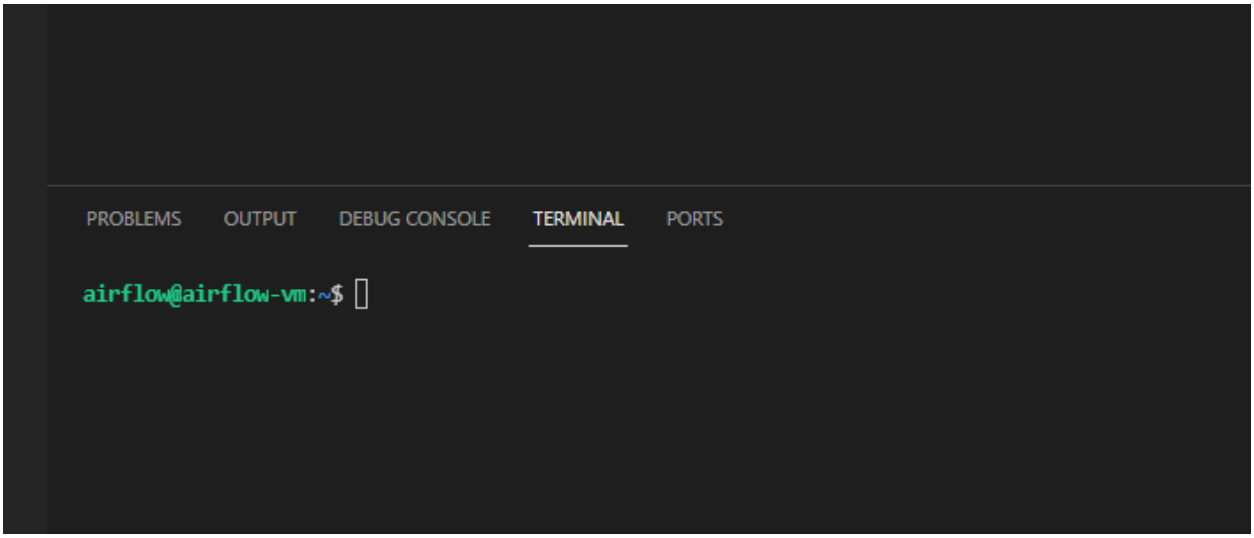
Since we have attempted to establish a SSH connection with the credentials airflow@localhost, we need to key in the password to log in. The password is airflow.

4.7 Go back to Explorer View of Visual Studio on left pane. Click on Open Folder. We want to access the root remote directory /home/airflow



4.8 Press OK

4.9 Now, access terminal. Click on the terminal tab on the bottom of VS:



Alternatively, hit the Terminal Tab and Create New Terminal.

Type pwd to verify you're in root directory. If you're inside, congratulations! You have just made your first remote access to a running linux VM!