SSH FS - Improving SSH with VSCode

Carlos Ayala Bellido
cayalabe@stanford.edu
October 14, 2022

1 Background

This is meant to be a guide to setting up and using SSH FS, an extension available in VSCode (Visual Studio Code). As moving around Terminal and Vim/Emacs can be a bit of a shock for those used to text editors like Eclipse or Jupyter, this not only makes it easier to SSH into terminal (with longer times between needing to reload into SSH) but also allow for editing files without needing to use Vim, Emacs, or any other Terminal language. This was an extension I was introduced to in my undergrad at Carnegie Mellon, so I hope this helps out others!

2 Setting Up SSH FS

2.1 Downloading SSH FS

Figure 1: The Extensions tab icon.

To start, we need to download SSH FS. On the left hand sidebar of VSCode, there should be an Extensions tab, with the icon seen in Figure 1. Once you have entered the Extension tab, search for "SSH FS", with the correct application by Kevin Schoofs. You have the correct extension if it matches the page shown in Figure 2.

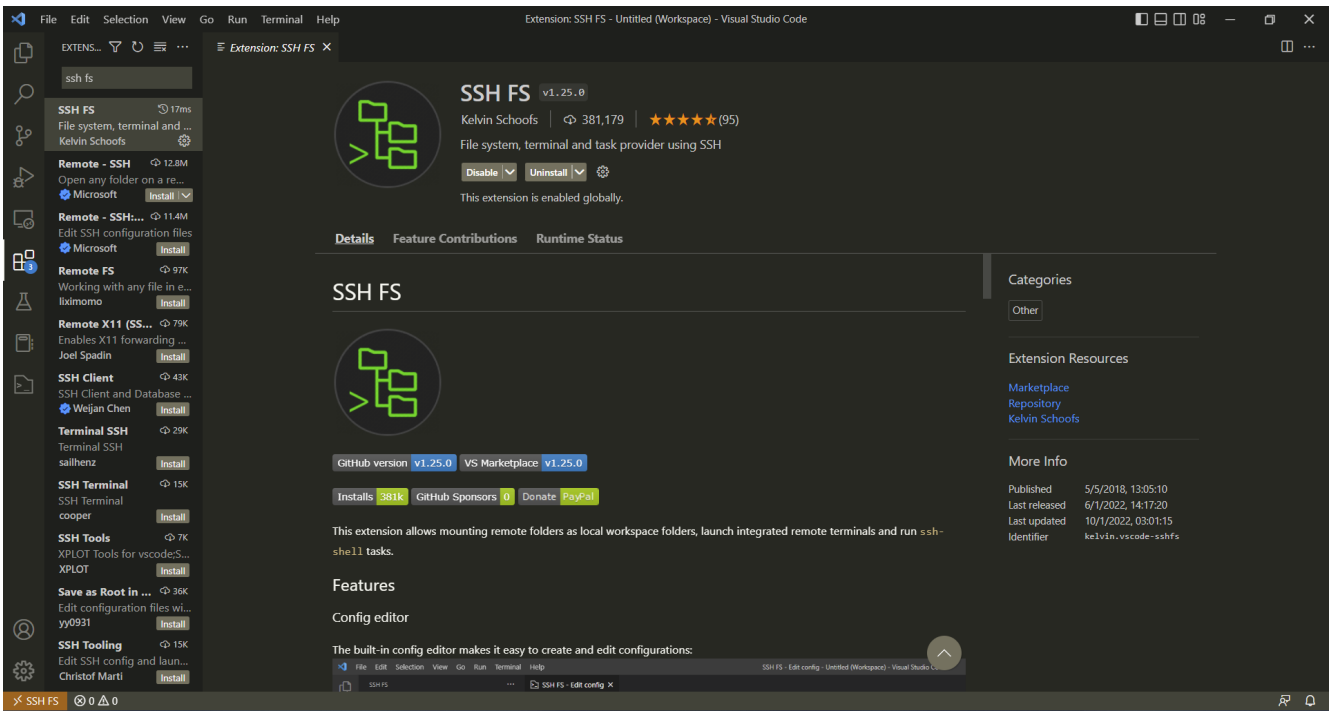

Figure 2: The SSH FS Extension page in the Extensions tab.

Install this extension, and you should see a new icon at the bottom of the left sidebar, specifically the icon seen in Figure 3.

Figure 3: The SSH FS tab icon.

2.2 Setting Up Your Stanford SSH Configuration

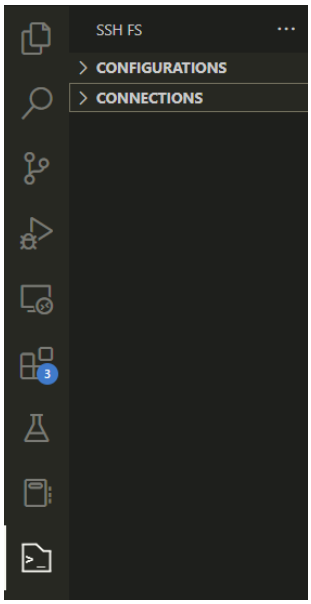

Figure 4: The initial SSH FS tab setup, which should be empty given a lack of configurations and connections to start.

Enter the SSH FS tab by clicking on the icon in the left sidebar. At this point, the left side of your screen should match the image seen in Figure 4. The drop-down that matters to us at this point is "Configurations". Hovering your mouse over this drop-down will reveal four new icons, seen in Figure 5.

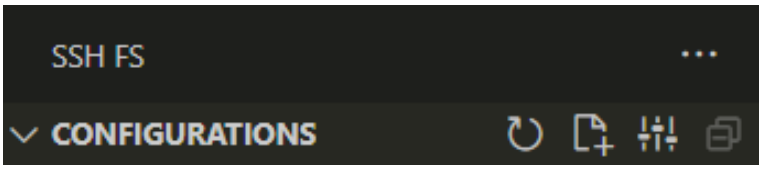

Figure 5: Hovering over the Configurations drop-down reveals these icons.

The refresh symbol is used to refresh for any updates to your configurations which were not automatically updated; the page with the plus symbol in the bottom-right serves to create new configurations; the sliders open up the full configurations page; and the double box collapses all open drop-down menus within the configurations tab.

For now, the only icon which matters is the page with the bottom-right plus symbol, so we'll click on that. Doing so greets you with the page seen in Figure 6, which is the start to creating a new configuration.

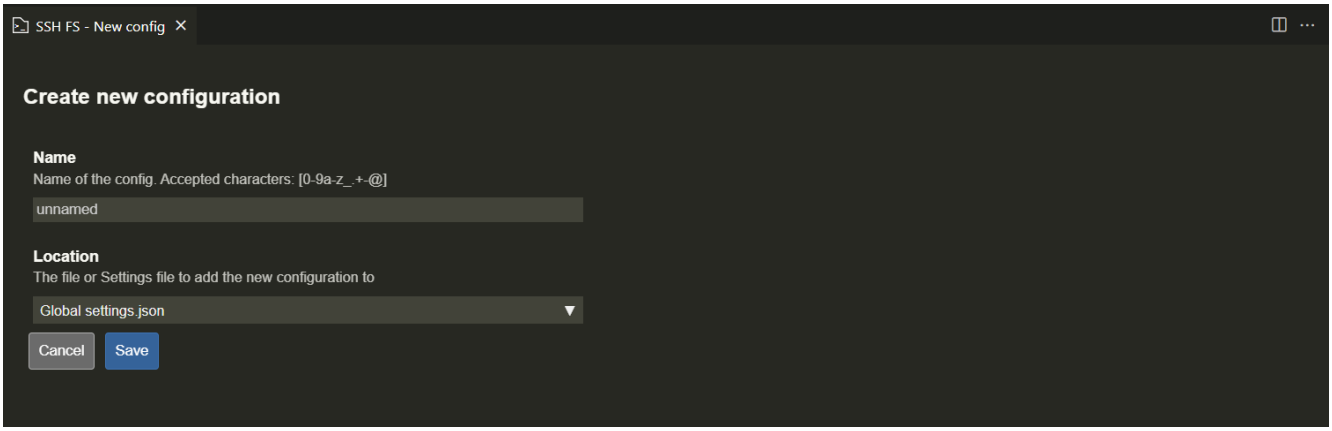

Figure 6: First page for new configuration setup with SSH FS.

Here, we will give whatever name (within their naming parameters) to our configuration, and keep location at "Global settings.json". Clicking save opens the next page, which may seem long and overwhelming. However, scrolling down reveals the only items we really care about: Host, Port, Root, Username, and Password. They are seen in Figure 7, and should be set up as follows:

Host: rice.stanford.edu (ie. host being used)
Port: 22 [leave unchanged if already 22]
Root: /farmshare/user_data/[SUNet ID]/ [ie. root path for directory and terminal]
Username: SUNet ID
Password: SUNet Password

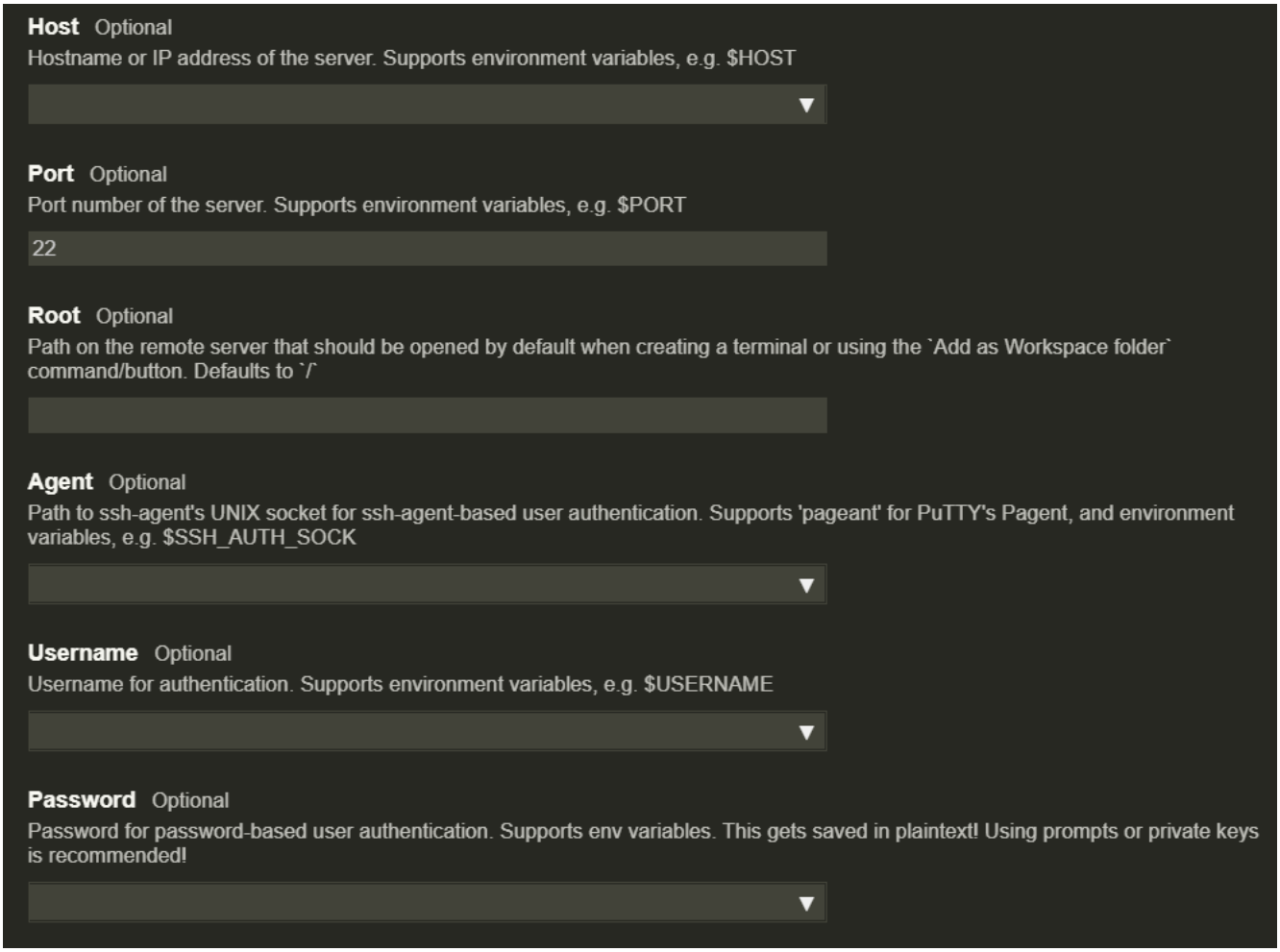

Figure 7: The items which matter from the second page for new configuration setup with SSH FS.

Once you have included all of this in your configuration, scroll to the bottom and click save. Now, your configuration tab should look somewhat like Figure 8.

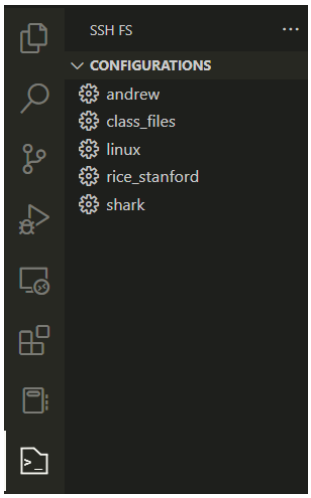

Figure 8: This is what your Configuration drop-down should now look like, specifically the addition of the new configurations.

3 Using SSH FS

Figure 9: Hovering over a specific configuration in the SSH FS tab reveals these icons.

Now that you have set up SSH FS and your Stanford configuration, we need to get this working. As with the Configurations drop-down menu, hovering over your specific configuration reveals icons, as shown in Figure 9. The folder with the bottom-right plus symbol adds your directory for access to your workspace; the box with the greater-than symbol opens a terminal to your directory specified in Root; the gear opens up that specific configuration's settings page; and the power plug disconnects your directory from your active workspace and exits your SSH. Clicking the folder or box will activate a Duo 2-Factor Push Notification, similar to when you SSH into your terminal, at the top of your screen and as seen in Figure 10.

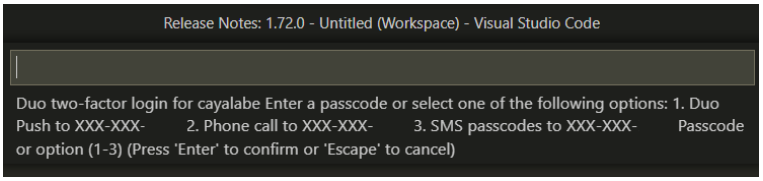

Figure 10: Duo 2-Factor Push Notification in VSCode.

Follow the instructions as you normally would, and you have now SSH'd into your directory. Note that, so long as your active (which can be seen when your Configurations is shown in the Connections drop-down like in Figure 11), you should not recieve another Duo 2F push. This means that if you only have either the directory or terminal open, you can open the other and not need to do a Duo 2F verification again.

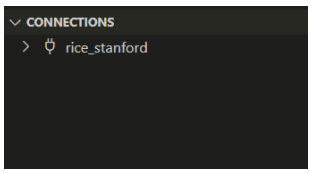

Figure 11: Connections drop-down in the SSH FS tab.

If you click on the folder (disregarding the Duo 2F Verification), you will be sent to the Explorer tab. If your SSH was successful, you should now see something similar to Figure 12 on the left side of your window. You should now be able to add, delete, edit and save files and folders without ever touching terminal again! Git operations still need to be handled through terminal though.

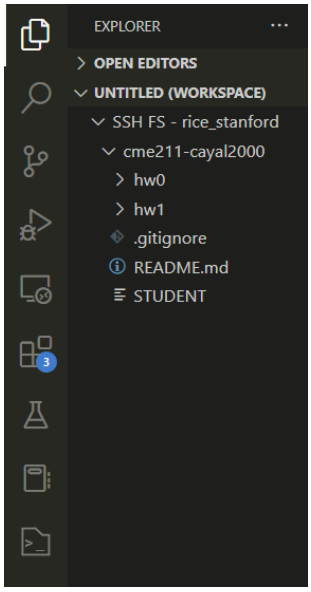

Figure 12: This is what your Explorer tab should now look like, specifically the addition of the "Untitled (Workspace)" drop-down.

If you click on the box (disregarding the Duo 2F Verification), you will be sent to the terminal at the bottom of your screen, as usual. If your SSH was successful, you should now see be SSH'd into the correct directory as listed by the Root variable set up in the Configuration Settings page!