Setting up Stampede2: A Brief Guide

Contributors: John Walker, Cian Roche, Tri Nguyen, Nora Shipp

Notation: Steps in a process are denoted: step 1 > step 2 > etc, and commands will be in quotes, eg. “ssh **username**@stampede2.tacc.utexas.edu” where **bold** indicates that the text changes depending on personal settings. The usual English quotes will be replaced by single quotes ‘xxx’. The home directory of your operating system will be represented by the tilde ~/ which on windows will be C:\Users**windows_username**\

1. Creating Account:

Once your advisor has set up an account for you (?) go to https://portal.tacc.utexas.edu/ and take the following steps:

> Login with TACC account

> forgot password (maybe not necessary?)

> follow instructions to receive email with password reset instructions

> login with new password

> go to home > Account Profile (url is https://portal.tacc.utexas.edu/account-profile)

> take note of TACC username. This username will be referred to as **username** for the remainder of the document. Should look like tgxxxxxx where x is a number.

Note: When asked to add a device for 2-factor authentication, don’t download the mentioned app and scan the QR code within that app. Instead, if you have the ‘duo’ app for other 2-factor logins, simply scan the QR code on your phone and it will automatically connect with duo. Open the app after adding and you should see a passcode titled **username**.

2. Accessing Cluster:

Standard login procedure is:

> Open terminal

> Run “ssh **username**@stampede2.tacc.utexas.edu” (ssh command should work on windows 10, mac and linux, but if it doesn’t you may need to install ssh. On windows you could get the ‘windows terminal’ app from the ‘Microsoft store’ windows app and make sure you run these commands from a powershell instance, not a cmd instance.)

```
PS C:\Users\CianM> ssh tg876846@stampede2.tacc.utexas.edu
To access the system:

1) If not using ssh-keys, please enter your TACC password at the password prompt
2) At the TACC Token prompt, enter your 6-digit code followed by <return>.
```

> You should be prompted for your password (the one set in section 1) and then a code which youll get in the duo app (or the tacc app if you went that route). Should then be logged in. We will improve the bash interface in section 3.

Nicer way to do that in future: SSH keys and config files

SSH keys will allow you to skip the stage where you type your password, and the config file allows you to skip typing the above command “ssh tg...” and replace it by something like “ssh stampede2”.

> Open a new terminal (powershell on windows). Generate an SSH key for your device by typing “ssh-keygen” (should generate a randomart image). This key is stored in ~/.ssh

> Navigate to the ~/.ssh folder (again, on windows this will be C:\Users**windows_username**\.ssh) and run “ls” to make sure ‘id_rsa’ is present

Option 1: Pull the config file from github

> Run the command

“curl -O https://raw.githubusercontent.com/CianMRoche/stampede2Setup/main/config”

then within this file ‘config’ change ‘USERNAME’ to your **username** (probably looks like tgxxxxxx) via some text editor.

Option 2: Manually make config file

> Create a file called “config” (if one doesn’t exist already) and enter the following contents:

Host stampede2

    HostName stampede2.tacc.utexas.edu

User **username**

IdentityFile ~/.ssh/id_rsa

then save. The final line may be unnecessary as it is the default location. Remember to change ~ to appropriate home directory on windows.

Now that the config file is present:

In the config file we have created a shortcut, such that we can type “ssh stampede2” in terminal and the username, host name and ssh key (which replaces password) are all sent automatically.

> If not on windows, in your terminal (personal system, not one logged in to stampede) run “ssh-copy-id -i ~/.ssh/id_rsa **username**@stampede2.tacc.utexas.edu”

And sign in as usual when prompted. Now stampede should know what your ssh key is (the one referred to by your config file).

> That command wont work on windows, so instead use (replace ~ and **username**)

“ cat ~/.ssh/id_rsa.pub | ssh **username**@stampede2.tacc.utexas.edu ”cat >> ~/.ssh/authorized_keys" ”

Error fixing: If ssh-copy-id or windows version doesn’t work

A. If you get an error like this:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions for '/Users/username/.ssh/id_rsa' are too open.
It is required that your private key files are NOT accessible by o
thers.
This private key will be ignored.
```

Then youll need to reset the permissions on your ssh keys. Best to google it, but try running “chmod 600 ~/.ssh/config” in your systems terminal, which sets the file to be readable and writable only by the user and not accessible by others. Then try the above step again

B. If you want to copy the key onto the server manually:

> Run “cat id_rsa.pub” in ~/.ssh and copy the entire output (on windows, ‘cat’ won’t work in command prompt, but will in powershell.) This is **your key** which you will now add to stampede’s ‘list of people who can access this account’.

> Go back to your terminal which is logged in to stampede (if you closed it, simply go through the steps at the start of section 2 again). Navigate to ~/.ssh via “cd ~/.ssh”

> Run “echo “**paste key here**” >> authorized_keys” to add your key to the list of allowed keys. Could also do manually via “vim authorized_keys” > press i for insert mode > press the Esc key and then press Shift + G to move cursor to end of file > paste as usual > press Esc > type :wq > press enter

Result of section 2:

In future you should sign in as follows:

> Open terminal

> run “ssh stampede2”

> Shouln’t need to enter password, only the TACC token code.

3. Making the Stampede Terminal Nicer

A terminal’s appearance is controlled by the ~/.bashrc file, so we will put a .bashrc on our stampede home directory to make interfacing with stampede easier.

> Log into stampede via “ssh stampede2”

> navigate to home directory via “cd $HOME”

> Pull a premade .bashrc (courtesy of Tri) from github via

“curl -O https://raw.githubusercontent.com/CianMRoche/stampede2Setup/main/.bashrc”

> check it’s there by running “ls -la” (list everything) and you should see .bashrc

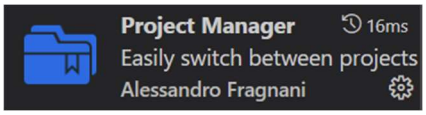
> Next time you log in the terminal should display the current file path and has some other general quality of life features. Feel free to edit this file with any customizations you may want for your terminal.

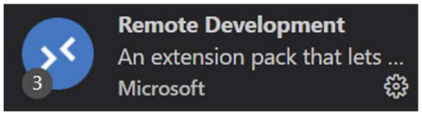
4. Using VS Code as Your Interface with Stampede

If you've set everything up using the defaults so far, this should be very simple, and a great quality of life improvement.

> Install visual studio code

> Go to the extensions manager (click squares on left hand side of vs code) and search for & install both 'Project manager' and 'Remote Development' which look like this:

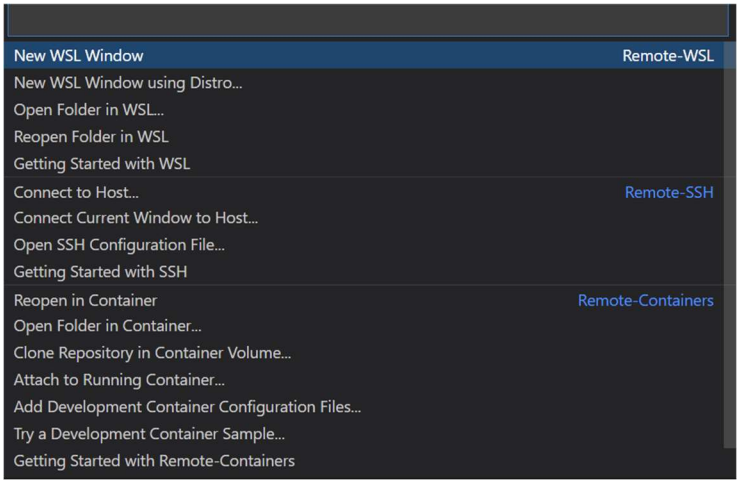




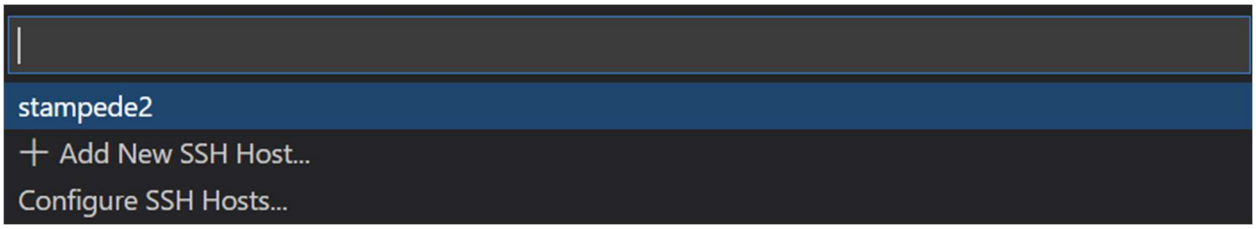
> File > preferences > settings > search for 'remote.ssh show login terminal' and make sure that setting box is checked (we want to show the terminal so we can type in our 2-factor code when vs code tries to connect to stampede)

> Click the green pair of arrows on the bottom left

> Click 'connect to host' in the remote-ssh menu



> If you used the default location for your 'config' file, then vs code should see it and you should see 'stampede2' as an option.



> If you don't see that as an option, click 'configure ssh hosts' and direct it to your config file from earlier (perhaps via 'specify a custom configuration file'). For more help, click the green button on the bottom left, then 'getting started with SSH'.

> Now vs code should open a new window and start connecting to stampede2. When it prompts for a tacc token in the terminal at the bottom of the vs code window, provide it and press enter as usual

> You should now be in the home directory. Changing to your work directory requires another 2-factor code (bug, should be fixed at some point in the future) so I do the following:

> Open a terminal inside vs code by clicking 'terminal' at the top, then new terminal. It should look like the one from section 3, since its using that .bashrc

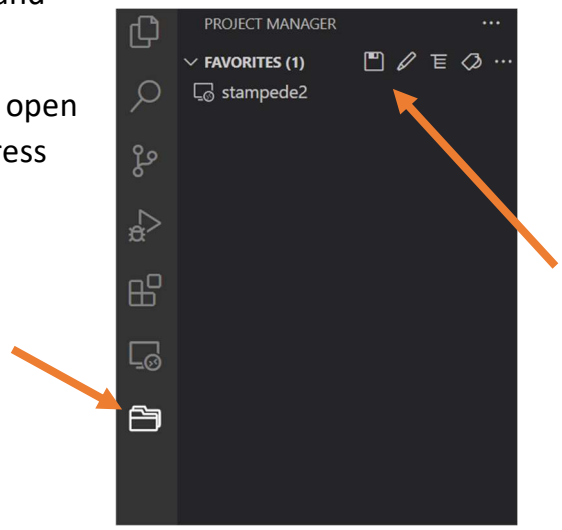
> type cd $WORK then copy the directory path youre put in. Should look like /work/08385/**username**/stampede2 where those number are likely different for you. Note that if you prefer you can do this with a different folder like scratch or even some subfolder. I just chose work because ist where I will write and edit the most code.

> In vs code window which is running stampede click file > open folder > paste the work directory path you just copied > press enter

> do the 2-factor thing again

> Now youre in the work folder, click the project manager extension on the left (looks like a folder)

> Click the save button (floppy disk icon beside the word 'favorites') and I choose to keep the default name



> Now you can close that window, and whenever you launch vs code you can simply click project manager on the left, then ‘stampede2’ (if you used the default name). This takes you directly to the work directory (after 2-factor) and you can interface with the home directory by opening a terminal in vs code and typing “cd ~ ”

Result: All Sections

To work on stampede2 you:

> Open vs code

> Click project manager on the left

> click ‘stampede2’ project (or whatever you named it)

> Do the 2-factor authentication

> Open a terminal inside vs code to work with folders other than your work directory and to run general commands. This terminal acts exactly like the terminal you would use without all this vs code stuff

> If you’ll work a lot in $HOME or scratch, then you can open that folder in vs code but it will require a 2-factor authentication code (for now).

5. Set up Anaconda Environment

> With a terminal connected to stampede2, run “cd $WORK” if youre not there already, then run

“curl -O https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh”

> then run the command “Anaconda3-2021.05-Linux-x86_64.sh” (the filename of the installer we just downloaded)

> Go through install with all defaults. Update anaconda after installing if update is available (you will be notified)

> Download an anaconda environment setup file courtesy of Nora via

“curl -O https://raw.githubusercontent.com/CianMRoche/stampede2Setup/main/py3.yml”

Or just define your own environment in a similar way

> Run “conda env create -f py3.yml” and let it run through.

> to activate the environment, run “conda activate py3” and you should see (base) turn to (py3) in your terminal

> Congrats, now you have a very robust anaconda environment loaded in stampede (and maybe also inside vs code!)

Special Thanks:

Lots of credit to John Walker for VS Code knowledge and general wisdom