EECS Tutorial: cslab Linux Environment SSH Access

FAQ for SSH access into cslab Linux environment

General Information regarding accessing cslab Linux environment using SSH

- **Please only connect into the cslab environment with an SSH client if you have previous experience in using SSH and the Linux command-line. If you are new to Linux, please use the cslab *Guacamole* web-browser interface.**

- SSH uses network port 22. If you are accessing cslab via SSH on WSU campus, ensure you are connected wirelessly to "WSU Secure" or using an Ethernet connected computer. "WSU Guest" wireless prohibits port 22 connections and your SSH session will fail to connect.

- You can only access cslab using SSH public key authentication. You cannot access cslab over SSH using your myWSU password.

- The cslab-nodes are located inside a virtual private network. These internal nodes are only SSH accessible via the `cslab-bastion.cs.wichita.edu` jumphost. Any external connection into the cslab environment must proxy connect through the `cslab-bastion`.

- `cslab-bastion.cs.wichita.edu` and `cslab-sftp.cs.wichita.edu` can be used as SCP or SFTP file-server hosts with SSH clients, such as *OpenSSH*, or graphical SFTP clients, such as *Filezilla*.

- This tutorial will help you configure your SSH client to make a proxy connection into cslab via `cslab-bastion` and to set up SSH public key authentication.

- **ONLY USE THE CSLAB-BASTION AS A PROXY/JUMPHOST INTO THE CSLAB-NODES OR AS AN SCP/SFTP SERVER. DO NOT DIRECTLY SSH INTO THE CSLAB-BASTION FOR OTHER TASKS!**

How do I access cslab Linux environment via *OpenSSH* on Linux or Mac?

Configuring the *OpenSSH* client for cslab access:

1. On your Linux or Mac desktop/laptop computer, copy the following host entry into your local user `~/.ssh/config` file:

```
Host cslab cslab-last cslab.cs.wichita.edu cslab-last.cs.wichita.edu
  ProxyCommand ssh your_mywsu_id@cslab-bastion.cs.wichita.edu ballast %h
  User your_mywsu_id
  IdentityFile ~/.ssh/cslab_rsa
  HostKeyAlias cslab.cs.wichita.edu

Host cslab-sftp cslab-sftp.cs.wichita.edu
  HostName cslab-sftp.cs.wichita.edu
  User your_mywsu_id
  IdentityFile ~/.ssh/cslab_rsa
  HostKeyAlias cslab.cs.wichita.edu
```

(Note: replace “your_mywsu_id” with your own 8 character ID number in both `ProxyCommand` and `User` lines.)

2. Open a command-line terminal emulator window and generate a new SSH public/private key pair for your local user by typing
   `ssh-keygen -t rsa -b 4096 -f ~/.ssh/cslab_rsa`

3. Follow the prompts in the command-line as your new SSH key is generated and enter a passphrase you will remember!
   It is highly recommended to use a passphrase for your SSH key to keep your Linux user account secure.

4. Open the newly generated SSH public key by typing
   `less ~/.ssh/cslab_rsa.pub`

5. Select all the text displayed within *less*, starting with `ssh-rsa` and ending with the hostname of your local computer.

6. Copy the text either by right-clicking on the terminal emulator window and selecting copy or by pressing the key combination **Ctrl+Shift+C**.

7. Open a web-browser application and follow the *eecs_tutorial_cslab_web_access* to access cslab-gateway.cs.wichita.edu

8. Once logged into *guacamole*, open a [cslab_SSH_CLI_terminal] connection.

9. Within the cslab SSH terminal session in your browser, open the *Guacamole* menu sidebar by pressing the key combination **Ctrl+Alt+Shift**.

10. Paste the copied text to the remote *Guacamole* [Clipboard] field using your preferred method, i.e. **Ctrl+V**.

11. Close the *Guacamole* menu sidebar by pressing the key combination **Ctrl+Alt+Shift**.

12. Within the cslab SSH terminal session, open your `authorized_keys` file by typing `nano ~/.ssh/authorized_keys`

13. Paste your locally copied SSH public key into the terminal session by right-clicking on the browser window with your mouse or by pressing the key combination **Ctrl+Shift+V**.

14. Ensure you have a blank line at end of the text file by pressing **Enter** after the text.

15. Quit *nano* by pressing **CTRL+X** and follow the prompts at the bottom of the screen to ensure you save the `authorized_keys` file.

16. Ensure correct permissions are set on the `authorized_keys` file by typing `chmod 600 ~/.ssh/authorized_keys`

17. You have now configured your local Linux or Mac computer to directly access cslab via *OpenSSH*. Make sure to properly disconnect and log out of the cslab *guacamole* web-interface once you are done.

18. **If you wish to set up more than one local computer with different SSH public/private key pairs for accessing cslab, then you can append additional SSH public keys in your cslab user `authorized_keys` file. Make sure to remove no longer used SSH public keys from this file.**

Logging into cslab using the *OpenSSH* client:

1. Ensure you have first followed the directions in the previous section to configure your *OpenSSH* client.

2. In your local computer open a command-line terminal emulator window and connect to the cslab Linux environment by typing
   ```
   ssh cslab
   ```

3. The first time you connect to the cslab environment using SSH, you will be asked to confirm the authenticity of each SSH remote host, i.e.

   ```
   The authenticity of host 'cslab.cs.wichita.edu' can't be established.
   ECDSA key fingerprint is [SHA256 or MD5 hash value].
   Are you sure you want to continue connecting (yes/no)?
   ```

4. Ensure the cslab-bastion and cslab host key fingerprints match one of the following SHA256 or MD5 hashes before typing `yes`:

   ```
   ECDSA key fingerprint is
   SHA256:X6dBKj4sqYYPWol6MXSQvGhpIQ6qBxh7mBQhnSw8n64
   MD5:d8:ba:c6:1c:86:fa:7f:f6:92:4f:c1:02:30:ce:ab:99

   ed25519 key fingerprint is
   SHA256:zzozIV7cP1T9C77PLRaevzdzCu21k44lbjd8jaJKS8Q
   MD5:6d:3d:8e:3a:db:f6:de:33:af:77:01:40:f3:71:1d:14

   RSA key fingerprint is
   SHA256:0CUyGZAYMdOd8vTOK3AtM2XTX3lMaGA2NP73rR7s6Ns
   MD5:75:5a:16:53:1a:7c:c2:4b:99:66:2d:e3:1e:76:f9:c9

   DSA key fingerprint is
   SHA256:7zW122xr+aoBb5yiRI96nvdx8Ml07qLKHYwG2Wu6jIM
   MD5:27:59:53:18:5a:67:71:f6:32:f1:e1:15:e9:e5:fe:b1
   ```

5. When prompted, enter your previously created passphrase to unlock your SSH private key and authenticate into cslab.

6. If the SSH connection completed successfully, then you will be presented with a standard shell prompt within a cslab node:
   ```
   your_mywsu_id@cslab-node-#:~$
   ```

Logging into your last used cslab-node-# using the *OpenSSH* client:

- When connecting to the cslab Linux environment using an SSH client, the ballast load-balancer will redirect you to one of the available and least used cslab-nodes at time of connection. Since load-balancing is calculated by ballast on a one minute cycle, you may not be redirected to the same cslab-node the next time you connect into cslab.
- **Use the following instructions to connect to the last used cslab-node only if you need access into a previously running SSH session. For normal use, your best option is always to connect via `ssh cslab` and let the ballast load-balancer automatically connect you to an available node.**
- To connect to the last node you previously accessed using SSH type
  `ssh cslab-last`

Running graphical (GUI) applications using the *OpenSSH* client:

- SSH allows for graphical applications to run on a local computer from the remote cslab Linux environment using X11 forwarding.
- If you are using Mac OSX, then you may need to install the XQuartz software before using X11 forwarding. Download XQuartz for Mac and install the software package.
- To use X11 forwarding on a per session basis append the `-X` option flag to your SSH command:
  `ssh -X cslab`
- To always use X11 forwarding for connections into cslab, instead of using the `-X` option flag, add the following line to the `Host cslab cslab-last ...` entry in your local user `~/.ssh/config` file:

  ```
  ForwardX11 yes
  ```

Copying files to/from cslab via SCP:

- To copy a file from your local computer to your user home directory on cslab using Secure Copy (SCP), type
  `scp local_filename_or_path cslab-sftp:~`
- To copy a file from your user home directory on cslab to a local directory on your local computer, type
  `scp cslab-sftp:~/remote_filename_or_path local_directory`