EECS Tutorial: cslab Linux Environment

FAQ for remote access into cslab Linux environment

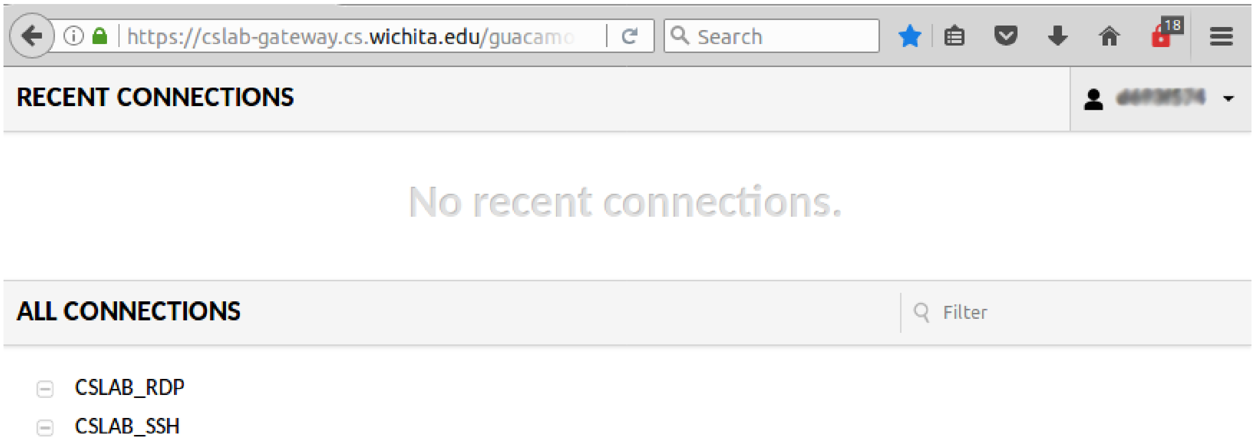
How do I access the cslab Linux environment via a web-browser?

1. Open your favorite HTML5 compatible web-browser. To test the compatibility of your browser go to html5test.com

2. In your browser go to cslab-gateway.cs.wichita.edu

3. At the login screen enter your myWSU ID and password and you will be presented with the *Apache Guacamole* home screen.

   Enter your myWSU ID with only lowercase letters for the login username.



4. To connect into an LXDE graphical RDP desktop session for working on programming assignments click on `CSLAB_RDP`.

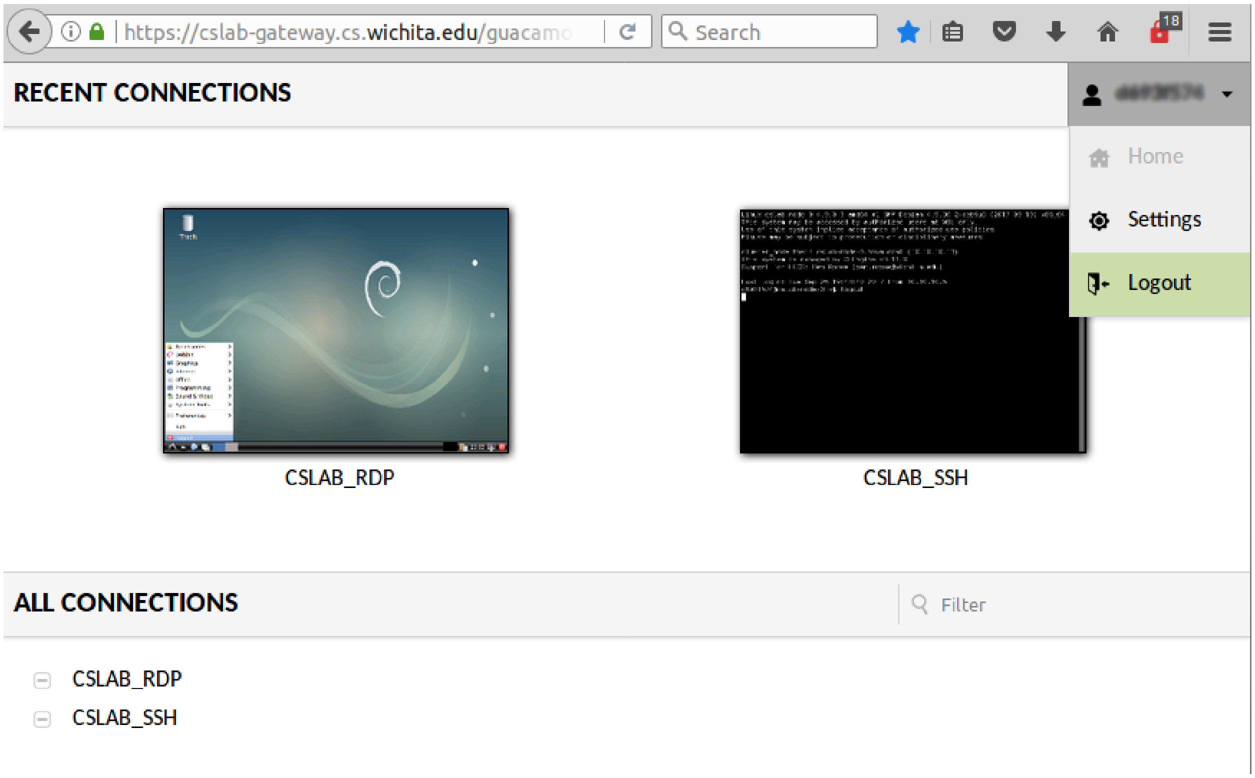
5. To connect into a command-line SSH terminal session for working on programming assignments click on `CSLAB_SSH`.

6. When you next log into cslab-gateway.cs.wichita.edu the *Apache Guacamole* home screen will show clickable thumbnails of your recent connections.

7. To log out of *Guacamole* from the home screen, click on your `myWSU_ID` at the top right. This drop-down list also enables you to open the `Settings` menu which includes displaying an on-screen keyboard for mobile devices.

8. During your initial connection into an RDP desktop session, occasionally you may see a policykit error message pop-up window. This is a known software bug which is being worked on. Clicking the OK button will close the error and it should not affect the rest of your login session.

9. To open/close the *Guacamole* menu sidebar while in the cslab environment press the key combination **Ctrl+Alt+Shift**. The *Guacamole* menu sidebar enables you to log out, disconnect, change settings, upload/download files, and use a remote clipboard.



10. **When you finish your work session, please make sure to logout from your connection within the cslab environment:**

    - **by using the [logout] taskbar button on bottom right or [logout] menu item on bottom left within the RDP desktop session,**
    - **by typing `exit` or pressing Ctl+D within the SSH terminal session, or**
    - **by using the *Guacamole* menu sidebar.**

11. **NOTE: All cslab-nodes automatically reboot every night between 2am and 3am. Any programs/processes left running will be killed during the reboot cycle.**

12. For further help on using the *Guacamole* interface go to Using Guacamole Guide

What am I allowed to do within the cslab Linux environment?

- You are allowed to use all software installed on the cslab Linux environment for their intended purpose of educatio and training in your programming and computer science classes at Wichita State University.

- When you log into the cslab environment you are accepting responsibility and accountabily for the commands, programs, and proccesses run by your Linux user account and the data stored within your user account. This is a position of trust: the university trusts that you will use cslab responsibly.

- If it is deemed by your instructor or the EECS systems administrator that commands, programs, or processes have been used by your user account on the cslab-nodes and associated servers for malicious purposes or outside of their intended purposes, then your access into cslab may be temporarily or permanently removed and there may be additional academic consequences to your actions.

Never share your myWSU password with other users.
Never attempt to use root or sudo privileged commands within cslab environment!

What software is available within the cslab Linux environment?

- cslab environment gives you both graphical and command-line Linux tools for writing, compiling, and debugging your CS programming class assignments. The Linux operating system running in cslab is Debian 9 (stretch) with a default LXDE desktop and Bash shell. Software tools/packages installed on the cslab-nodes include:
  - Text and code editors: leafpad, nano, vim, and emacs.
  - Integrated development environments (IDE): atom, geany, and eclipse.
  - Compiling tools: GNU C compiler (gcc), g++, make, haskell-compiler, prolog, perl, python, and java.
  - Debugging tools: GNU debugger (gdb) and data display debugger (ddd).
  - Latex tools: pdflatex, texlive, and texmaker.
  - Version control tools: git and subversion.
  - GUI terminal emulators: terminator, lxterminal, and xterm.
  - CLI terminal multiplexers: screen and tmux.
- To check if a specific software package or version is installed within the cslab-nodes, connect into the SSH terminal session or open a terminal emulator in the RDP desktop session and type:

```
apt list --installed specify_package_name_here
```

- If a Linux software package is not installed within the cslab Linux environment which you require to complete your class assignment, then please ask your instructor whether this package can be installed for you by the EECS systems administrator.

Do not attempt to install any software packages yourself!

How do I copy text within the cslab Linux environment?

Copying text from your local computer to the remote environment:

1. Copy the required text to the clipboard within your local computer application using your preferred method of copying, i.e. **Ctrl+C**.

2. Within the cslab environment in your browser, open the *Guacamole* menu sidebar by pressing the key combination **Ctrl+Alt+Shift**.

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

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

5. Within the RDP desktop session, any text shown in the *Guacamole* [Clipboard] can be pasted into a remote cslab application by normal methods, i.e. **Ctrl+V**.

6. Within the SSH terminal session, text in the *Guacamole* [Clipboard] can be pasted into the terminal by right-clicking on the browser window with your mouse or by pressing the key combination **Ctrl+Shift+V**.

Copying text from the remote environment to your local computer:

1. Within the RDP desktop session, text can be cut or copied from any cslab application by normal cut/copy methods, i.e. **Ctrl+C**.

2. Within the SSH terminal session, text to be copied is ”highlighted” using the mouse.

3. Open the *Guacamole* menu sidebar by pressing the key combination **Ctrl+Alt+Shift**.

4. The copied or ”highlighted” text will appear in the remote *Guacamole* [Clipboard] and can then be selected and copied to your local computer clipboard, i.e. **Ctrl+C**.

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

How do I download/upload files within the cslab Linux environment?

Using guacctl/guacget to download a file:

- Within the SSH terminal session, you can use the *Guacamole terminal session control utility (guacctl)* for downloading files. To download a file from the SSH terminal session to your local computer via the web-browser type:

  ```
  guacget file_to_be_downloaded
  ```

- `guacget` is an alias for the command `guacctl --download`. To see all the option flags available for this command type `guacctl`.
- **NOTE: `guacget` only works in SSH terminal sessions. `guacget` does not work in RDP desktop sessions.**

Drag-and-drop to upload a file:

- You can drag-and-drop a file from your local computer onto the cslab web-browser window. This can be used in both RDP desktop and SSH terminal sessions.
- By default the file is uploaded into your user home directory on the remote cslab-node.
- You can set a custom destination directory for future uploaded files when using drag-and-drop by typing:

  ```
  guacctl -s custom_upload_directory
  ```

Guacamole file browser to upload or download a file:

1. Open the *Guacamole* menu sidebar by pressing the key combination **Ctrl+Alt+Shift**.
2. Click on the disk drive icon under [Devices] to open a file browser of the remote cslab-node.
3. Browse to your user home directory on the remote server. You can then browse to subdirectories within your user home. Your home directory full path on the remote cslab-node will look like the following:

   ```
   /opt/homes/stu##/your_myWSU_id/
   ```

4. If you are unsure where your home directory is located on the remote cslab-node, in a terminal or in the SSH terminal session type `pwd` to show you the full path of your present working directory.

5. Downloads are initiated by double-clicking on any file shown, while uploads are initiated by clicking the [Upload Files] button. Clicking [Upload Files] will open a file browsing dialog where you can choose one or more files from your local computer, ultimately uploading the selected files to the directory currently displayed within the remote cslab-node file browser.

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

Can I open multiple browser tabs/windows into the cslab Linux environment?

- cslab restricts RDP desktop sessions to a single connection per user. You can view the remote RDP desktop session from only one web-browser tab/window at a time. If you attempt to open more than one web-browser tab/window and connect to a second RDP desktop session, then the gateway will present an error message in the second tab.

- cslab restricts SSH terminal sessions to a maximum of four concurrent connections per user. You can open up to four web-browser tabs/windows and connect into remote SSH terminal sessions at the same time. Each open SSH terminal session tab/window will run a separate command-line shell instance on the same remote cslab-node.

- You can use the `tmux` or `screen` terminal multiplexer commands to run multiple command-line shells concurrently in the same SSH session tab/window.

- You can open one web-browser tab/window to connect to a graphical RDP desktop session and open additional tabs/windows to connect to SSH terminal sessions concurrently. However, the *Apache Guacamole* system cannot guarantee that your RDP desktop session and your SSH terminal session(s) will connect to the same remote cslab-node.

How do I access the cslab Linux environment on port 22 via an SSH client?

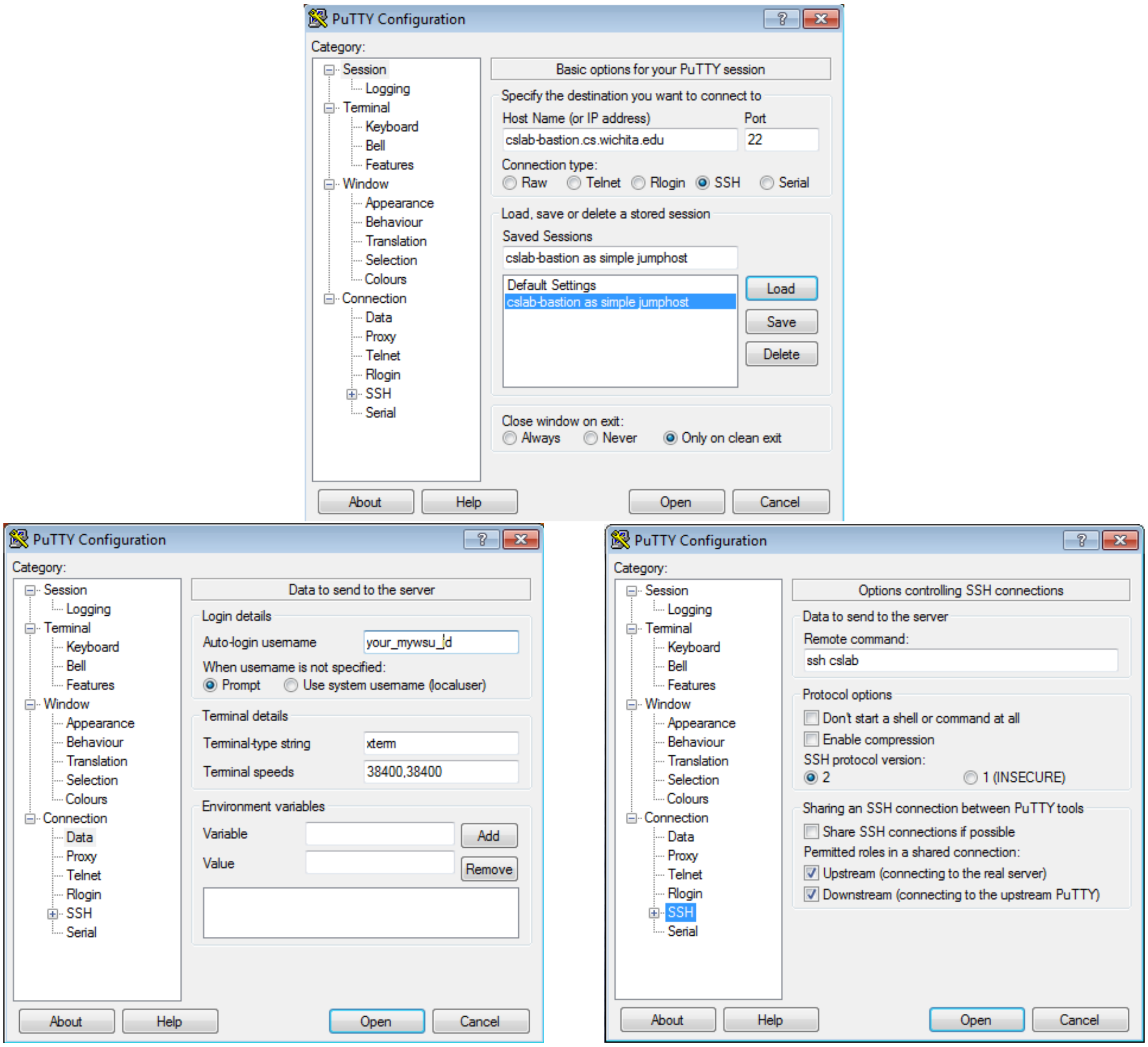
- The cslab-nodes are located inside a virtual private network. These internal nodes are only SSH accessible via the cslab-bastion.cs.wichita.edu jumphost. Any external connection into the cslab environment must proxy connect through the cslab-bastion and the bastion should be used only as an SSH jumphost/proxy.

- 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.

- **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 *Guacamole* web-browser interface.**

Using *PuTTY* SSH client on Microsoft Windows:

1. Download *PuTTY* from Simon Tatham's official download webpage and install.
2. Open *PuTTY* and in [Session] category add a new SSH connection with [Host Name] as `cslab-bastion.cs.wichita.edu` and [Port] as `22`.
3. In [Connection–Data] category add your myWSU_ID into [Auto-login username].
4. In [Connection–SSH] category add the text `ssh cslab` into [Remote command].
5. In [Session] category add a name for the configuration, such as "cslab environment" and click *Save*. Saving as "Default Settings" will always open with this configuration.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): cslab-bastion.cs.wichita.edu

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: cslab-bastion as simple jumphost

Default Settings
cslab-bastion as simple jumphost

Close window on exit: ○ Always ○ Never ● Only on clean exit

Data to send to the server

Login details

Auto-login username: your_mywsu_id

When username is not specified: ● Prompt ○ Use system username (localuser)

Terminal details

Terminal-type string: xterm

Terminal speeds: 38400,38400

Environment variables

Variable

Value

Options controlling SSH connections

Data to send to the server

Remote command: ssh cslab

Protocol options

□ Don't start a shell or command at all

□ Enable compression

SSH protocol version: ● 2 ○ 1 (INSECURE)

Sharing an SSH connection between PuTTY tools

□ Share SSH connections if possible

Permitted roles in a shared connection:

☑ Upstream (connecting to the real server)

☑ Downstream (connecting to the upstream PuTTY)

Figure 1: *PuTTY* configuration for cslab

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

7. Ensure the cslab-bastion and cslab host key fingerprints match one of the following SHA256 or MD5 hashes before clicking *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
```

8. You will be prompted twice to enter your myWSU password, once for the cslab-bastion (SSH jumphost/proxy) and once for the internal cslab-node.

9. You may see a `Could not chdir to home directory....` warning. Do not be concerned, this is not a critical error.

10. If the SSH connection completed successfully, then you will be presented with a standard shell prompt:

```
your_mywsu_id@cslab-node-#:~$
```

Using *OpenSSH* client on Linux or Mac OSX:

1. Add 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
  HostKeyAlias cslab.cs.wichita.edu
  User your_mywsu_id
```

2. In a local CLI terminal 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 and 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. You will be prompted twice to enter your myWSU password, once for the cslab-bastion (SSH jumphost/proxy) and once for the internal cslab-node.

6. You may see a `Could not chdir to home directory....` warning. Do not be concerned, this is not a critical error.

7. If the SSH connection completed successfully, then you will be presented with a standard shell prompt:

```
your_mywsu_id@cslab-node-#:~$
```

How do I use graphical (GUI) applications in cslab via an SSH client?

Graphical applications cannot currently be accessed in Microsoft Windows using the *PuTTY* SSH client. Please use the *Guacamole* web-browser interface instead.

Using *OpenSSH* client with X11 forwarding on Linux or Mac OSX:

- Ensure you have followed the directions in using *OpenSSH* client first.
- SSH allows for graphical applications to run on a local computer from the remote cslab Linux environment using X11 forwarding.
- To use X11 forwarding on a per session basis append the `-X` option flag to your SSH command, i.e.

  ```
  ssh -X cslab
  ```

- To always use X11 forwarding for connections to cslab, instead of using the `-X` option flag, add the following line to the `Host cslab cslab-last....` entry in your `~/.ssh/config` file:

  ```
    ForwardX11 yes
  ```

- If you are using Mac OSX, then you may need to install XQuartz before using X11 forwarding. Download XQuartz for Mac and install the software package.

How do I copy files from/to the cslab Linux environment via an SSH client?

Using *OpenSSH* client on Linux or Mac OSX:

- Ensure you have followed the directions in using *OpenSSH* client first.
- 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:~
  ```

- To copy a file from your user home directory on cslab to a local directory on your local computer, type

  ```
  scp cslab:~/remote_filename_or_path local_directory
  ```

How do I access the last accessed cslab-node via an SSH client?

- When connecting to the cslab Linux environment using an SSH client, the ballast load-balancer on cslab-bastion 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 using SSH.

- **Use the following instructions to connect to the last used cslab-node only if/when you need access into a previously running SSH connection. For normal use, your best option is always to connect via `ssh cslab` and let the ballast load-balancer automatically connect to an available node.**

Using *OpenSSH* client on Linux or Mac OSX:

1. If you need to connect to the last cslab-node you previously accessed using SSH, just append `-last` to the SSH command, i.e.

   ```
   ssh cslab-last
   ```

Using *PuTTY* SSH client on Microsoft Windows:

1. Open *PuTTY* and load the previously created “cslab environment” session.

2. Change [Connections–Data] configuration category [Remote command] entry to `ssh cslab-last`.

3. *Save* changed configuration as a new [Session] with a different name, such as “cslab environment last used node”.

4. If you need to connect to the last cslab-node you previously accessed using SSH, connect using this newly created *PuTTY* session.

What are the currently known bugs/glitches within the cslab Linux environment?

Due to the cslab environment being so new, there are a few software bugs and glitches which you may see at times. Most of these bugs occur during operations that involve the new cslab environment interacting with the older CS servers and functions and are not critical issues.
Known bugs include:

- A policykit error message pops up when user first logs into a *Guacamole* RDP desktop session if other users are also logged in. This is a minor bug within policykit and can be easily mitigated by clicking *Okay* in the pop-up window.

- When the *handin* command is used in the cslab environment for programming assignment submission it may produce a `segmentation fault` error. This is caused by the older 32-bit *handin* program running on the cslab 64-bit processor architecture. The error is minor and does not affect submission of student assignments for grading.

- When accessing cslab via an SSH client through the cslab-bastion, a `Could not chdir to home directory....` warning may be displayed. This is caused by cslab-bastion not having access to user home directories as an SSH connection is established. cslab-bastion is only used as an SSH jumphost to allow for external access into the cslab-nodes, and does not require access to user home directories for this function. The warning can be disregarded.

- Graphical (GUI) applications cannot currently be accessed in Microsoft Windows using the *PuTTY* SSH client. This is due to a limitation with the *PuTTY* client which requires SSH public key authentication for using the cslab-bastion as a secure SSH proxy. Adding public key authentication for SSH access into cslab is planned as a future option. For now, please use the *Guacamole* web-browser interface when graphical applications are required from a Microsoft Windows computer.

If you experience a bug or error when using the cslab Linux environment which stops you from completing your programming assignments, please inform your instructor at your earliest convenience.