ECE Linux SSH + VS Code Setup Guide

Author: Jeff Huang / jzxhuang

How to set up SSH for ECE linux "the right way". No VPN is required to SSH into ECE linux. Notably, with this setup:

- Skip `eceterm` completely and go directly to `eceubuntu/ecetesla` - you only need to type the SSH command ONCE (and enter your password ONCE).
- Configure VS Code to work over SSH. Edit remote files while maintaining your VS Code features like Intellisense, syntax highlighting, themes, etc.

Skip to the bottom of the document for a summary (TL;DR).

*Disclaimer: I have only tested on macOS. It should work fine for *nix systems, but not sure about Windows. The instructions are up-to-date as of the last update of the Gist. Some Windows-specific notes are included in some sections thanks to feedback from classmates using Windows.*

1. SSH Setup

Covers SSH key generation, SSH authorized key setup, and SSH config file to use `ProxyJump`.

1.1. Generate SSH Key

The first step is to generate SSH private/public pair. If you've already done this before, skip this step.

Run

```
ssh-keygen -t rsa
```

Create a password to protect this key if you'd like (optional).

I HIGHLY recommend naming the files something meaningful (ie. `ecelinux`) rather than `id_rsa`. You should still save it under the path `~/.ssh`.

For Windows, if `ssh-keygen` isn't available, look for guides online on how to generate SSH keys for Windows :(

1.2. Configure SSH Remotely (Authorized Keys)

Again, skip this step if you've already done this before. Essentially, you need to copy the public key you generated into the file `~.ssh/authorized_keys` on the remote ecelinux machine.

For example, you could copy it into your clipboard, SSH into the remote machine, and paste it into that file. Here's a step-by-step breakdown, if you need that:

1. Copy your SSH public key to eceterm (replace `id_rsa` with your file name):

```
scp ~/.ssh/id_rsa.pub username@eceterm.uwaterloo.ca
```

1. Connect to the remote machine, ie. over SSH (the password will be your Nexus PW):

```
ssh yourusername@eceterm.uwaterloo.ca
```

1. If the folder and file `~/.ssh/authorized_keys` does not exist yet, create the folder and file. Then, append your public key to the end of the file.

```
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

1. Delete the public key from your remote machine:

```
rm ~/id_rsa.pub
```

Test it out by disconnecting from the remote machine, then connect again using `ssh username@eceterm.uwaterloo.ca`

This time, you shouldn't have to enter your Nexus ID, only your SSH passphrase if you set one.

1.3. Configure SSH Locally (Skip eceterm!)

This step allows you connect directly to eceubuntu/ecetesla, skipping eceterm completely.

On your local machine, open the file `~/.ssh/config` (if it doesn't exist yet, create it).

Add the following to the file. Replace `User` with your username, and make sure `IdentifyFile` points to your SSH private key.

```
Host eceterm eceterm1 eceterm2
    HostName %h.uwaterloo.ca
    User jzxhuang
    ForwardAgent yes
    UseKeychain yes
    AddKeysToAgent yes
    IdentityFile ~/.ssh/ecelinux

Host eceubuntu1 eceubuntu2 eceubuntu4 ecetesla0 ecetesla1 ecetesla2 ecetesla3
    HostName %h.uwaterloo.ca
    User jzxhuang
    ProxyJump eceterm
    ForwardAgent yes
    UseKeychain yes
    AddKeysToAgent yes
    IdentityFile ~/.ssh/ecelinux
```

Some notes:

- The most important fields are `ForwardAgent` and `ProxyJump`. This allows you to SSH directly to `eceubuntu*` and `ecetesla*` without going through `eceterm` first.
- `UseKeychain` and `AddKeysToAgent` are used to save the SSH passphrase to your OS keychain, ie. in macOS. This is optional. If you're not sure if your OS supports this, delete these fields.

If you followed these steps correctly, you should now be able to SSH directly to eceubuntu. For example, `ssh eceubuntu1` should work directly!

2. VS Code

If you aren't comfortable with terminal-based editors like Vim, a great way to work on ECE linux is through VS Code's Remote Development extension. It allows you to use VS Code on any remote machine, while maintaining all your preferences and themes!

In order to do this, you **must** have set up SSH correctly, especially the automatic `ProxyJump` (part 1.3 of this guide). If you haven't, please follow those steps first!

2.1. Install VS Code and Remote Development Extension

If don't have VS Code, install it (Google is your friend). Then, install the `Remote Development` extension pack by Microsoft.

2.2. Connect with Remote SSH

Use the extension to connect to eceubuntu through VS Code! Yep, it's that easy.

`Command Palette > Remote-SSH: Connect to Host... > eceubuntu*`

Assuming your SSH config is set up correctly, this should work. If it doesn't work, check you SSH config. If it still doesn't work, sorry :(

Note for Windows: Apparently, the default Windows SSH client doesn't support ProxyJump properly. A solution is to use something like Git Bash. In your local VS Code settings, find the settings for the Remote SSH extension, find the field that sets the path of your SSH client, and point it to your SSH client that supports ProxyJump correctly (ie. `C:\Program Files\Git\usr\bin\ssh.exe`).

2.3. Configure IntelliSense/Language/Autocompletion

You'll need to install extensions remotely on the ECE linux servers. Simply do so by installing extensions after you've Remote-SSH'ed in. For example, you might want to install the Java development pack, if you're working with Java.

2.4. Additional Tips

Here are some tips:

- Install Intellisense extensions as needed on the remote instance of VS Code. For example, VS Code is great for Java with the `Java` extension pack!
- After opening a Java project, in the sidebar, go to `Java Dependencies > Referenced Libraries` and add appropriate paths. For example, this might be `/lib`, `/gen-java`, etc.

2.5. ECE 454 Specific Tips

A collection of tips for ECE 454, written for the S2020 offering of the class. In general, you will need to specifically add pre-compiled `.jar` files to the Java Referenced Libraries to get Intellisense to work perfectly.

A1

VS Code doesn't recognize the source files in `gen-java`. A hacky workaround is to copy `BCryptService.java` and `IllegalArgument.java` into your root directory for A1. You'll need to update `build.sh` to avoid compiling those files, unless you want a big mess of files in your root directory.

REMEMBER TO REVERT YOUR BUILD.SH BEFORE SUBMITTING.

A2

You can SSH direclty to `ecehadoop`. Add it as a host in your SSH config file.

You can easily get VS Code IntelliSense working with Hadoop by adding the following to your Referenced Libraries (assuming you've Remote-SSH to `ecehadoop`):

- `/opt/hadoop-latest/hadoop/share/hadoop/common`
- `/opt/hadoop-latest/hadoop/share/hadoop/mapreduce`

Can't figure out how to get Scala Intellisense working. I don't think it's as necessary though since typing is much less strict and all that's really required are the basic FP operators (map, reduce, filter, etc.).

Summary (TL;DR)

Generate SSH key pair. Add public to `AuthorizedKeys` on the ECE server. Add the following to your local `~/.ssh/config` file:

```
Host eceterm eceterm1 eceterm2
    HostName %h.uwaterloo.ca
    User jzxhuang
    ForwardAgent yes
    UseKeychain yes
    AddKeysToAgent yes
    IdentityFile ~/.ssh/ecelinux

Host eceubuntu1 eceubuntu2 eceubuntu4 ecetesla0 ecetesla1 ecetesla2 ecetesla3
    HostName %h.uwaterloo.ca
    User jzxhuang
    ProxyJump eceterm
    ForwardAgent yes
    UseKeychain yes
    AddKeysToAgent yes
    IdentityFile ~/.ssh/ecelinux
```

To use VS Code over SSH, install the `Remote Development` extension pack. You can use it to directly edit remote files in VS Code
(`Command Palette > Remote-SSH: Connect to Host... > eceubuntu*`).

An alternative to this workflow is to mount the remote drive SSHFS. I found this option to be slow and doesn't

meet all my needs, but it's another valid option!

I hope you found this useful. Share it with your classmates and star/bookmark this for your own convenience!