

cse15l-lab-reports

Lab Report 1

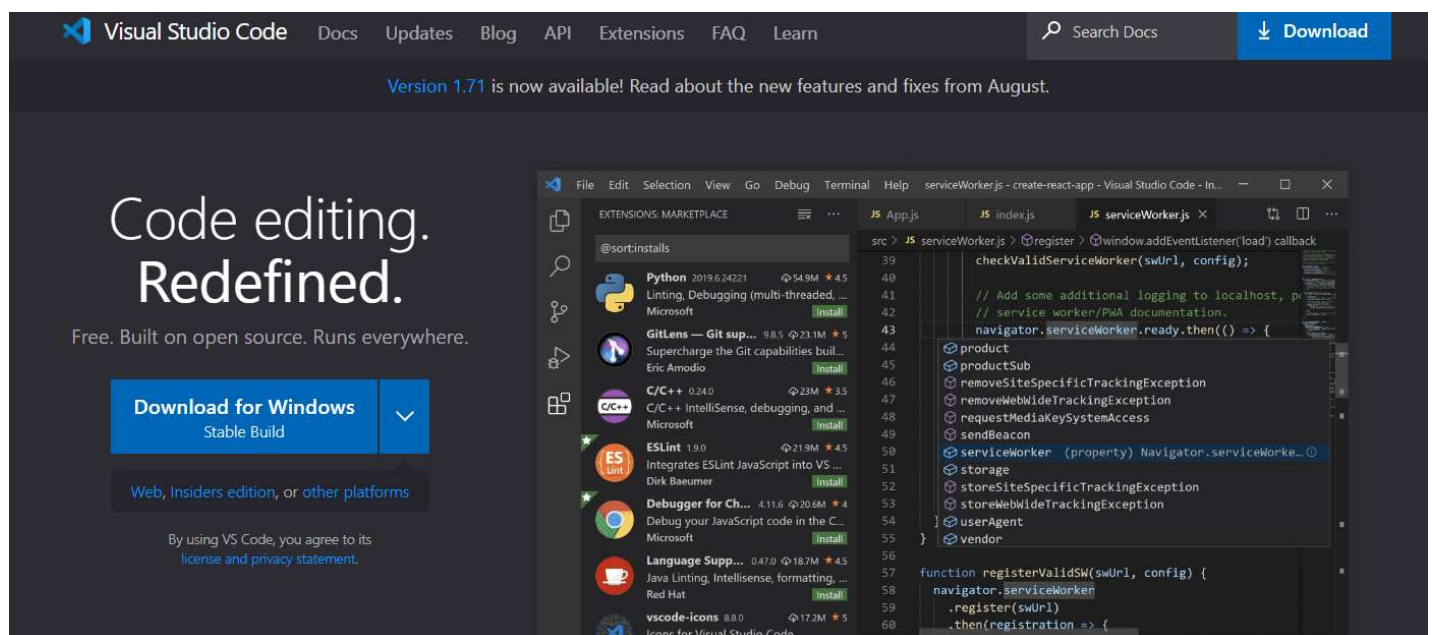
In order to set up remote connection for CSE 15L, it is important that we need to know how to connect to the three computers that we will be using, labeled as ieng6.

This is the tutorial for how to remote in 6 Steps:

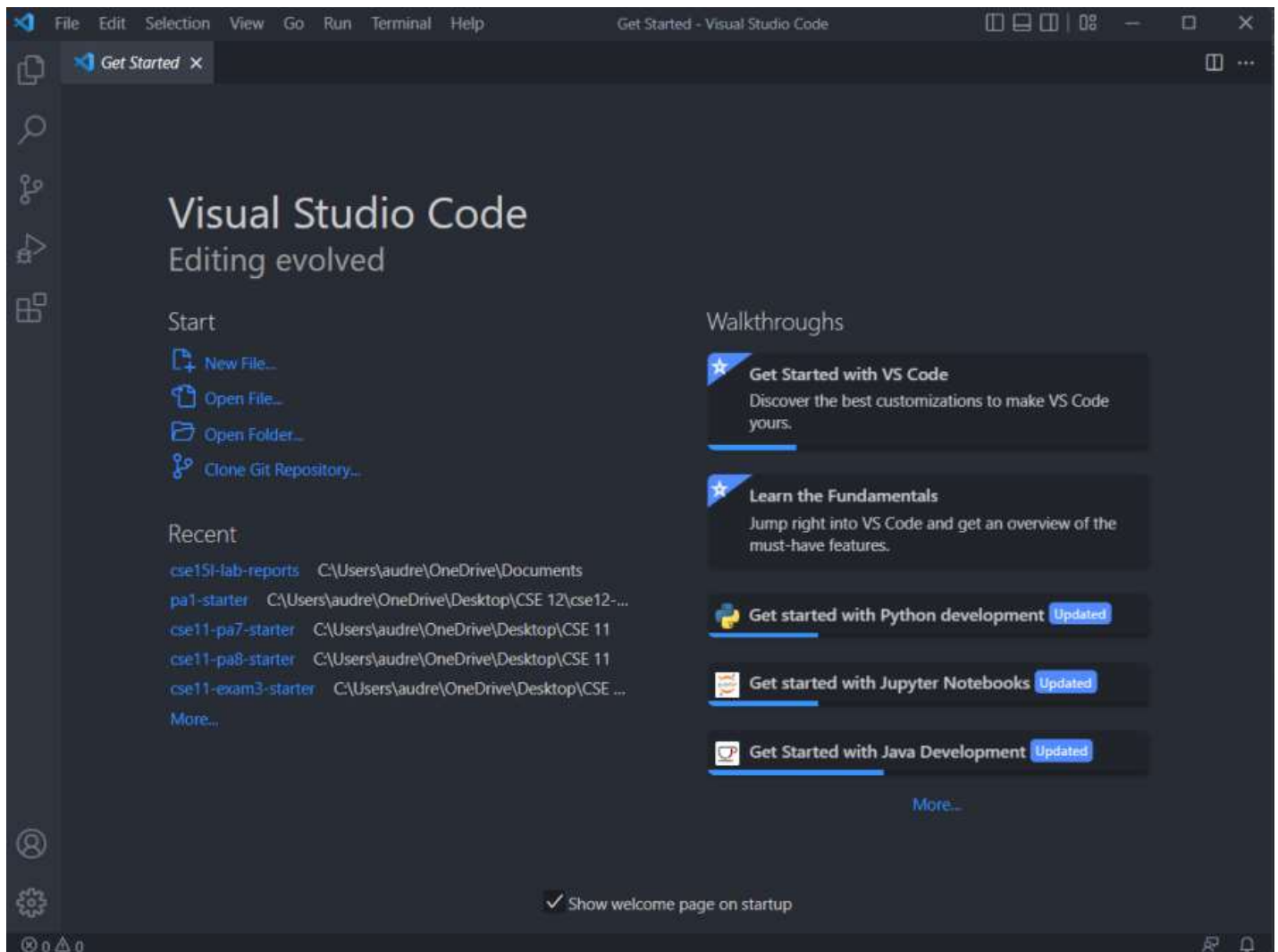
- Installing VSCode
- learning how to Remotely Connect
- Simple Commands
- Moving Files with `scp`
- Setting up SSH (Windows OS)
- Optimizing Remote Running

Step 1: VS Code:

[Visual Studio Code](#) is an IDE that will become very helpful in editing code. You can download it in the link above for Windows, Mac, and Linux and read the descriptions to download it to your computer.



After downloading, you will be able to see this startup page:



Open the terminal using the command `Ctrl + Shift + ~`, or starting a new terminal in the command bar above using `Terminal > New Terminal`.

Step 2: Remotely Connecting:

Before going further on the terminal, you need to first find your CSE 15L account. You can do this by going to the [CSE 15L Account Lookup](#) and inputting your UCSD SSO credentials. Then, click on the button that says `cs15lfa22xx` (the `xx` is replaced by your custom username, mine is `ni`) and click reset your password. You can use the same password as your SSO.

Account Lookup ⓘ

Username *

jsmith

Student ID *

a12345678

Submit

Clear Fields

Forgot Username or New Student?

Last Name *

Smith

Student ID *

a12345678

Submit

Clear Fields











Wait around 15-20 minutes and then go back to the VS Code Terminal.

I operate on Windows, so I had to install Open SSH on my computer. Here are the steps to do that:

(Note: you must be running on an admin account to do these steps)

- Go to **Settings > Apps > Optional Features**
 - Scroll through the already added features, and check to see if Open SSH Client is already installed. Personally, I use this feature to remote into other computers for work, so I already had it installed.

Apps > Optional features

	Chinese (Simplified) Supplemental Fonts	152 MB	▼
	Internet Explorer mode	3.28 MB	▼
	Math Recognizer	30.5 MB	▼
	Microsoft Quick Assist	3.12 MB	▼
	Notepad (system)	705 KB	▼
	OpenSSH Client	10.3 MB	▼
	Steps Recorder	1.13 MB	▼
	Windows Fax and Scan	18.6 MB	▼
	Windows Hello Face	146 MB	▼
	Windows Media Player	79.0 MB	▼

- If it isn't, go to `Add a feature` and search for Open SSH Client. Click **install**. Return to the VS Code terminal.

In terminal, type `ssh cs15lfa22ni@ieng6.ucsd.edu`. For Windows, it will look slightly different than Mac OS, but it's possible to use Powershell to operate now.

When prompted for the fingerprint, type `yes` and input your password that you set earlier. Your screen should look something like this if successful:

```
PS C:\Users\audre\OneDrive\Documents\cse15l-lab-reports> ssh cs15lfa22ni@ieng6.ucsd.edu
Last login: Sat May 28 01:18:42 2022 from 128.54.173.166
Hello cs15lfa22ni, you are currently logged into ieng6-202.ucsd.edu
```

```
You are using 0% CPU on this system
```

Cluster Status

Hostname	Time	#Users	Load	Averages
ieng6-201	20:35:01	10	0.24,	0.06, 0.06
ieng6-202	20:35:01	13	0.05,	0.03, 0.05
ieng6-203	20:35:01	18	0.03,	0.04, 0.05

```
Fri Sep 30, 2022 8:36pm - Prepping cs15lfa22
[cs15lfa22ni@ieng6-202]:~:18$
```

Step 3: Simple Commands

Good job getting here!

For the purposes of this tutorial, only `ls`, `dir`, `cat`, `who` and `cd` will be used as simple commands.

- `ls` and `dir` lists out the files in the current directory
- `cat` prints out the contents of a file
- `who` gives information about who else is connected to the remote server
- `cd` changes the directory
 - Using `cd` will give you the ability to open files, while `cd ..` brings you back to the previous directory.

Run these commands to see how they work.

```
[cs15lfa22ni@ieng6-202]:~:19$ who
cs15lfa22nd pts/0      Sep 30 20:17 (23-119-226-140.lightspeed.sndgca.sbcglobal.net)
zol008 pts/1        Sep 30 18:58 (rrcs-24-43-123-85.west.biz.rr.com)
cs15lfa22gs pts/2      Sep 30 20:28 (128.54.209.29)
cs15lfa22gg pts/4      Sep 30 18:49 (128.54.202.137)
cs15lfa22cc pts/5      Sep 30 19:09 (128.54.153.76)
cs15lfa22qo pts/6      Sep 30 20:23 (69.196.34.48)
jih077 pts/7        Sep 30 18:19 (cpe-70-95-200-113.san.res.rr.com)
cs15lfa22re pts/8      Sep 30 20:17 (107-132-35-38.lightspeed.sndgca.sbcglobal.net)
cs15lfa22ml pts/9      Sep 30 19:40 (cpe-76-88-4-33.san.res.rr.com)
cs15lfa22hc pts/10     Sep 30 20:33 (128.54.174.1)
cs15lfa22ni pts/11     Sep 30 20:36 (76.82.31.145)
cs15lfa22em pts/14     Sep 30 20:33 (128.54.151.246)
cs12fa22pe pts/17     Sep 30 18:50 (cpe-75-80-109-135.san.res.rr.com)
cs15lfa22bq pts/12     Sep 30 20:21 (cpe-76-176-146-203.san.res.rr.com)
[cs15lfa22ni@ieng6-202]:~:19$
```

Log out of the remote computer using the command `exit` .

Step 4: Moving files with `scp`

With the `scp` command, you can copy files from your local computer to the server.

1. Create a java file. For this, I create a file called `WhereAmI.java` with code that was provided from the course. You could create one as simple as a file that only prints out `"Hello World!"` .
2. Save it
3. In terminal, type `scp "fileName" directory` . For instance, since I am transferring over `WhereAmI.java`, I would be typing `scp WhereAmI.java cs15lfa22ni@ieng6.ucsd.edu:~/` to transfer to your remote server.
4. Input your password and press **enter**

```
PS C:\Users\audre\OneDrive\Documents\cse15l-lab-reports> scp WhereAmI.java cs15lfa22ni@ieng6.ucsd.edu:~/
WhereAmI.java: No such file or directory
PS C:\Users\audre\OneDrive\Documents\cse15l-lab-reports> cd ..
PS C:\Users\audre\OneDrive\Documents> cd ..
PS C:\Users\audre\OneDrive> cd ..
PS C:\Users\audre> scp WhereAmI.java cs15lfa22ni@ieng6.ucsd.edu:~/
WhereAmI.java
```

Note: You have to be in the directory that the file is in, or else you won't be able to find the file, like I did above

Step 5: Generating an SSH Key

To save time, generating an SSH Key is easier so you don't have to keep inputting passwords and waiting for longer than you have to transfer and copy files.

In local terminal, type `ssh-keygen`, which will generate a public and private RSA key pair. There are different types, but for this, I used the default, which is RSA.

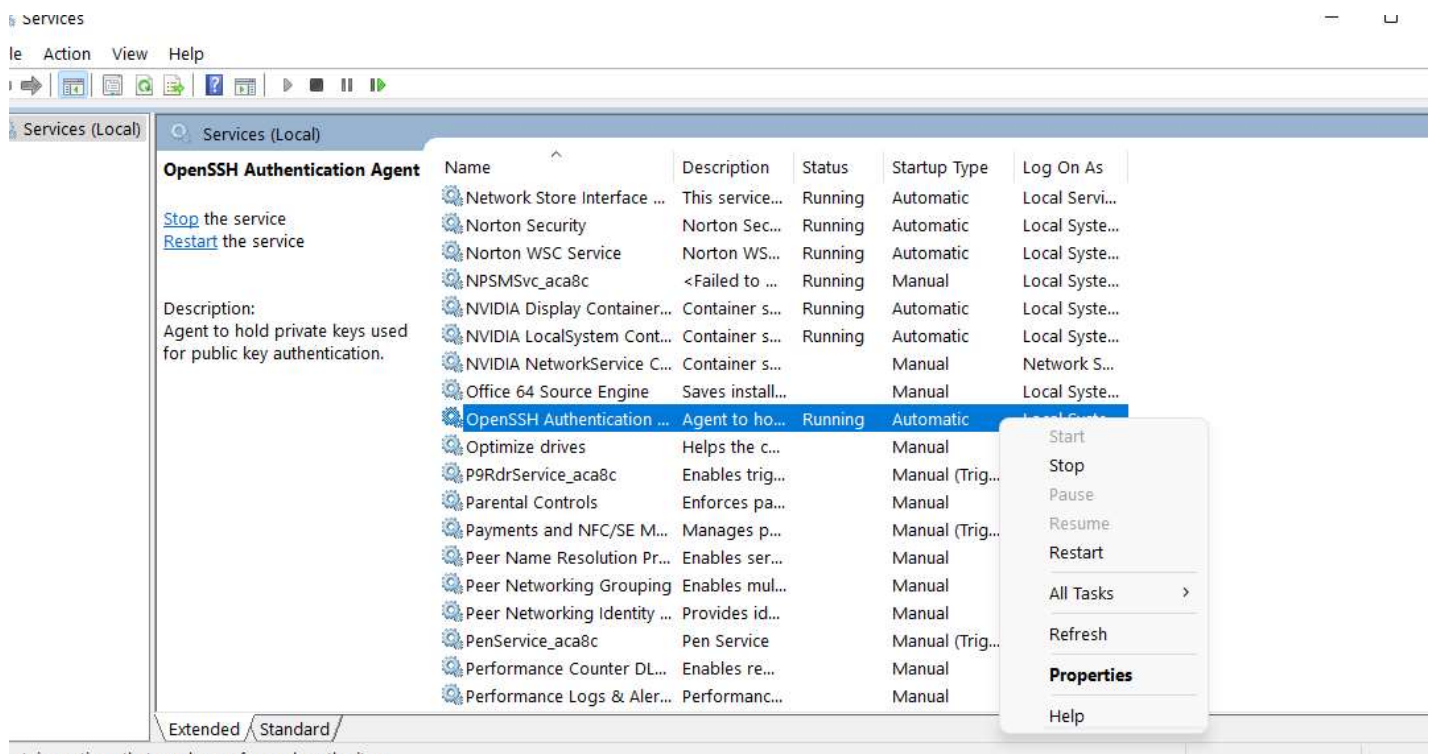
You will be prompted to input a file to save the keypair to, and then input a passcode. For both, I just pressed **Enter**, which will save it to the default, and then set an empty passcode, so I won't have to input a passcode.

```
PS C:\Users\audre> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\audre\.ssh\id_rsa): cdcd
Enter passphrase (empty for no passphrase):
```

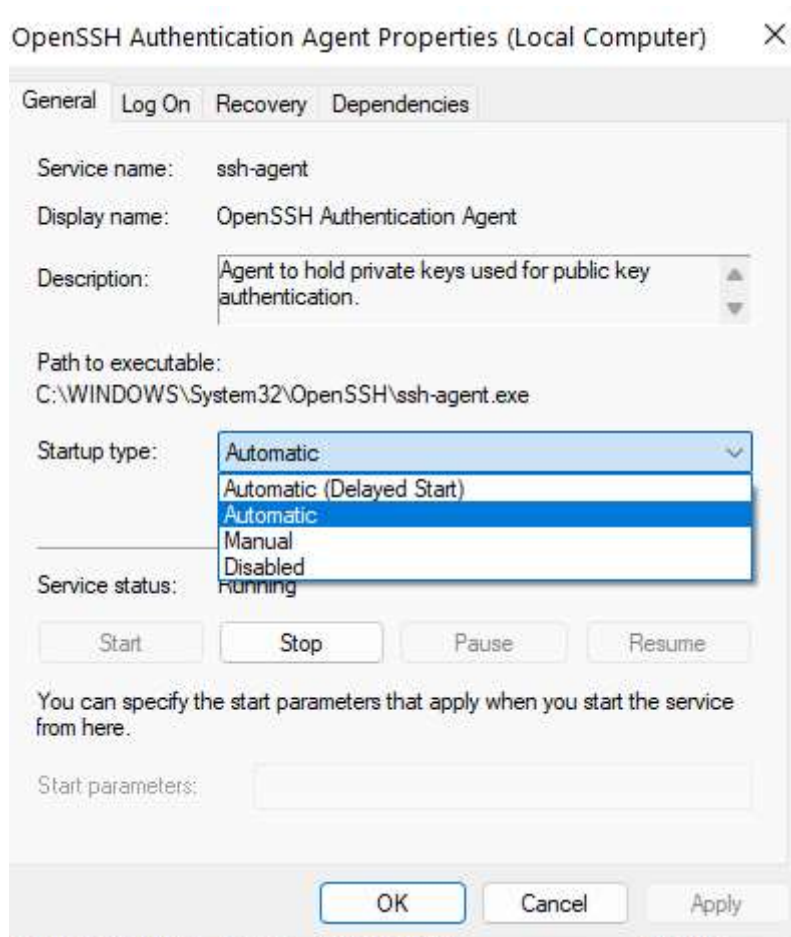
For Windows, there is an extra step. I was told to follow the `ssh-add` function on the Microsoft website, but it didn't work for me, so I did it without going through command prompt and finding it.

[Here are the directions to the command prompt directions.](#)

If that also doesn't work for you, I went into Services through finding the **Windows Start menu > Services**. Find **Open SSH Authentication Agent** and click on Properties. By default, Windows has this feature disabled.



Afterwards, toggle the Startup type to **Automatic**. Click **Start** and then **apply**.



Finally, go back to vscode and type `ssh-add`. You should get an output that looks like this.

```
PS C:\Users\audre> ssh-add
Identity added: C:\Users\audre\.ssh/id_rsa (audre@AUDREYPC)
PS C:\Users\audre>
```

Go back to VS Code terminal, then type `ssh cs15lfa22zz@ieng6.ucsd.edu` and enter your password. Then, on the server, type `mkdir .ssh`, which will make a directory called `.ssh`.

Logout of the server. Then, locally, type `scp /Users/audre/.ssh/id_rsa.pub cs15lfa22ni@ieng6.ucsd.edu:~/ .ssh/authorized_keys`. Change around the type to fit your credentials. Enter your password.

Congrats, you are now able to get into the remote server quicker!

Step 6: Optimizing Remote Running

To make things easier for myself, I tried out running things now that I don't have to put in a password. Every command I did above was able to be recalled using the up arrow on the keyboard. So, I used the up arrow to do several things:

- copy a file of my WhereAmI.java file to the server
- log in using ssh
- running commands on the server before even logging into the server
- compiling and running the WhereAmI.java file

In total, this saved about 80% of the time it would have usually taken me to do all the steps without the ssh key.

```
PS C:\Users\audre> scp WhereAmI.java cs15lfa22ni@ieng6.ucsd.edu
1 file(s) copied.
PS C:\Users\audre> ssh cs15lfa22ni@ieng6.ucsd.edu "ls"
WhereAmI.class
WhereAmI.java
perl5
PS C:\Users\audre> javac WhereAmI.java
PS C:\Users\audre> java WhereAmI
Windows 11
audre
audre
C:\Users\audre
PS C:\Users\audre> █
```

Step 7: Celebrate

You are now able to efficiently log in remotely!