

Editing Locally and Using SFTP: the FileZilla-Sublime-Terminal Flow

Matthew Salim, 20 May 2016

This guide focuses on effective and efficient offline editing on Sublime Text. The key is to use SFTP for Sublime Text, which allows you to edit files locally and upload them efficiently. Instead of relying on the traditional download-edit-upload-confirm_upload-build-loop, all we need to do is to save the file, and it'll automatically upload the file to the Shark machines. Users have found it to be effective as the method is efficient and they can use the full capabilities of Sublime Text.

We will guide you in setting up your system as part of working towards Datalab. This should take no more than 10 minutes to complete.

This guide is applicable for Windows, Mac, and Ubuntu. We haven't tested it on Atom (another editor), but it should work as well with its own SFTP plugin.

Please note that both Sublime Text and the SFTP plugin are not free. They will nag you once in a while, and you can simply click cancel to continue using both software, but please do support them if you enjoy using them!

First Run: Setting Up Your System through Datalab

Things you'll need to install:

- FileZilla (<https://filezilla-project.org/>)
- Sublime Text 3 (<http://www.sublimetext.com/3>)
- Terminal
 - On Windows: PuTTY (<http://www.putty.org/>), Remote Terminal (on Windows Store), Linux Subsystem (Windows 10 Insider Preview)

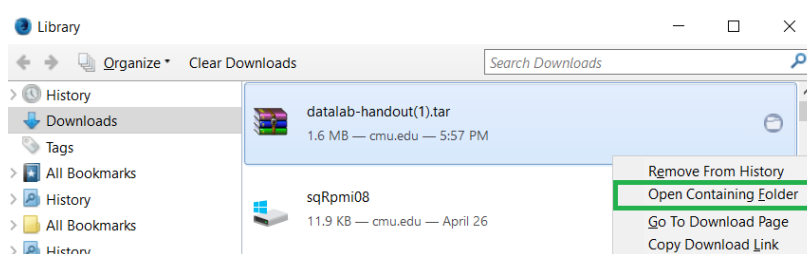
The steps are:

1. Downloading assignment handout from Autolab
2. Using FileZilla
3. Copying to Shark
4. Extracting assignment handout
5. Setting up Sublime Text for SFTP
6. Editing locally
7. Save-build-run loop
8. Downloading assignment handin from Shark
9. Uploading assignment handin to Autolab

1. Downloading Assignment Handout

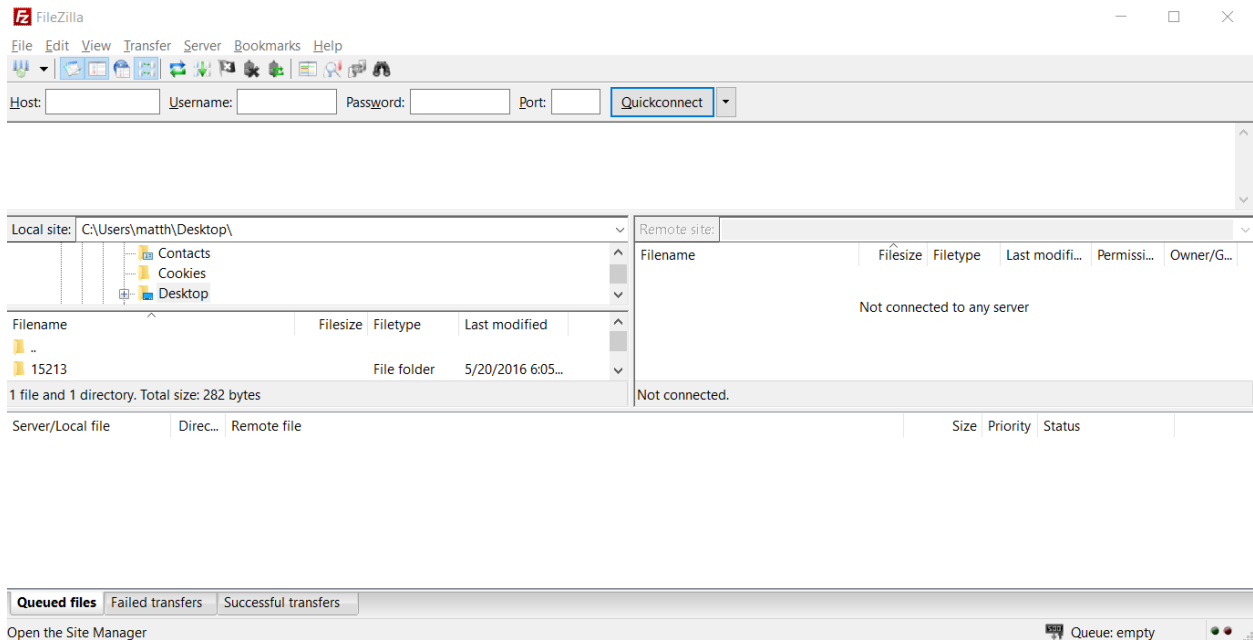
Download the Datalab assignment handout (.tar) from [Autolab](#) locally to your computer.

Make sure to take note where you save your file locally, since we'll need the location to copy it to Shark in step 3. You can right click on the downloaded handout and click open containing folder to check its location.



2. Using FileZilla

FileZilla provides a GUI for you to transfer files onto Shark. If you find scp to be more efficient, feel free to skip steps 2 and 3. FileZilla will initially look like this:



The left side reflects folders in your computer, and the right side reflects folders in the remote machine, which will be Shark. What we need to do is to fill in the host, username, password, and port fields on top of the window with the following details:

- Host: shark.ics.cs.cmu.edu
- Username: (your Andrew ID)
- Password: (your Andrew ID password)
- Port: 22

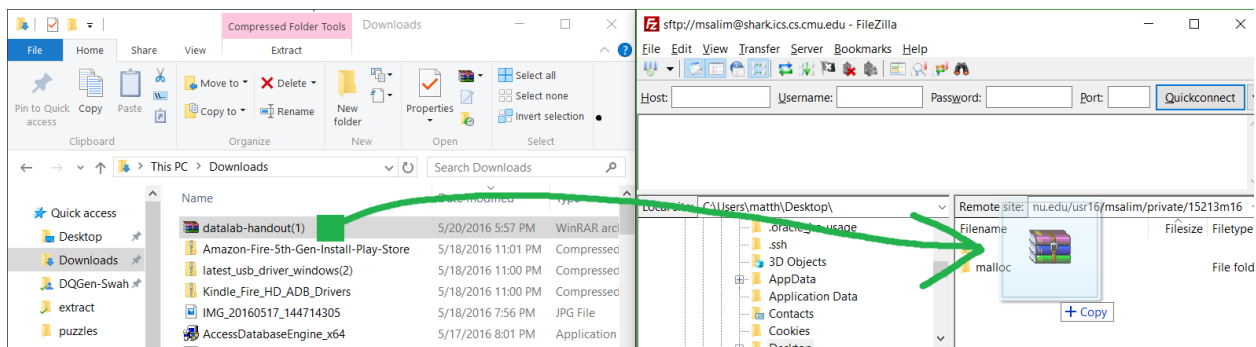
Then, press Enter or click the Quickconnect button. If you allow FileZilla to remember your password, you will be able to connect without entering your information again by clicking on the drop down arrow on the right of Quickconnect.



Now, you have a GUI to browse through your folders on Shark. Make sure that everything related to 15-213 / 18-213 / 15-513 is in your private folder. Navigate into your private folder, and, if you prefer, create a 15213 / 18213 / 15513 folder and enter it.

3. Copying to Shark

On FileZilla's left column (or Windows Explorer, GNOME Files, or Mac OS X Finder), open the folder where you've downloaded the assignment handout (based on step 1). Drag the file into your 15213 / 18213 / 15513 folder and drop it there.



That's all!

4. Extracting assignment handout

We now need to use the Terminal. Open your terminal and ssh to Shark:

```
ssh hbovik@shark.ics.cs.cmu.edu
```

Enter your password when prompted. Note that they will not show stars (*) for each character that you enter, but it is entered.

Navigate to the folder that your handout is in—for example:

```
cd private/15213
```

You can use `ls` to list the files and folders in the current folder. Then, run the following command—for instance, for Datalab, run:

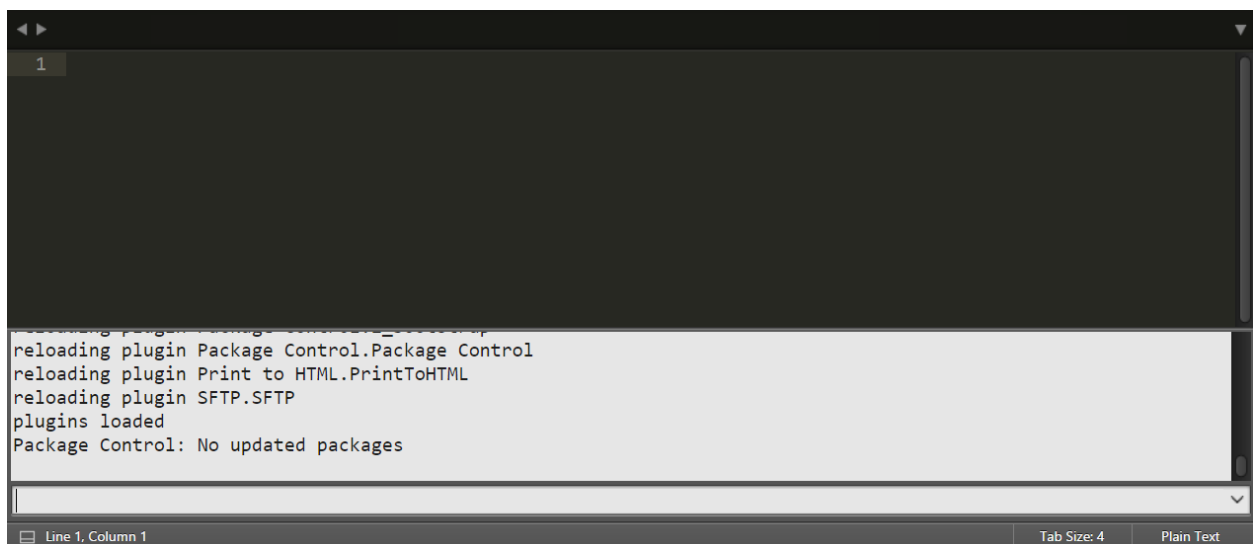
```
tar -xvf datalab-handout.tar
```

It will create a folder named `data1ab-handout` that contains all the assignment materials. You can navigate into the directory (`cd data1ab-handout`) to see what's in it.

Don't close the terminal! We'll need it in the next step.

5. Setting Up Sublime Text for SFTP

Here comes the fun! This will allow you to edit your files locally and upload them efficiently. First thing: open Sublime, and press `Ctrl + `` (Mac: `Command + ``) to bring up the console.

A screenshot of the Sublime Text console window. The console is dark-themed and shows the following text: "reloading plugin Package Control.Package Control", "reloading plugin Print to HTML.PrintToHTML", "reloading plugin SFTP.SFTP", "plugins loaded", and "Package Control: No updated packages". The status bar at the bottom indicates "Line 1, Column 1", "Tab Size: 4", and "Plain Text".

```
reloading plugin Package Control.Package Control
reloading plugin Print to HTML.PrintToHTML
reloading plugin SFTP.SFTP
plugins loaded
Package Control: No updated packages
```

On a web browser, go to the Sublime Text Package Control webpage at <https://packagecontrol.io/installation>. It should look like this:

Package Control

INSTALLATION

Simple

The simplest method of installation is through the Sublime Text console. The console is accessed via the `ctrl+`` shortcut or the `View > Show Console` menu. Once open, paste the appropriate Python code for your version of Sublime Text into the console.

SUBLIME TEXT 3

SUBLIME TEXT 2

```
import urllib.request,os,hashlib; h =
'2915d1851351e5ee549c20394736b442' +
'8bc59f460fa1548d1514676163dafc88'; pf = 'Package
Control.sublime-package'; ipp =
sublime.installed_packages_path();
urllib.request.install_opener(
urllib.request.build_opener(
urllib.request.ProxyHandler()) ); by =
urllib.request.urlopen( 'http://packagecontrol.io/' +
pf.replace(' ', '%20')).read(); dh =
hashlib.sha256(by).hexdigest(); print('Error validating
download (got %s instead of %s), please try manual
install' % (dh, h) if dh != h else open(os.path.join(
ipp, pf), 'wb' ).write(by)
```

This code creates the Installed Packages folder for you (if necessary), and then downloads the `Package Control.sublime-package` into it. The download will be done over HTTP instead of HTTPS due to Python standard library limitations, however the file will be validated using SHA-256.

Copy the entire string under the “Sublime Text 3” tab, paste it into the console, and press Enter. It should take a while to install. Afterwards, close and restart Sublime Text.

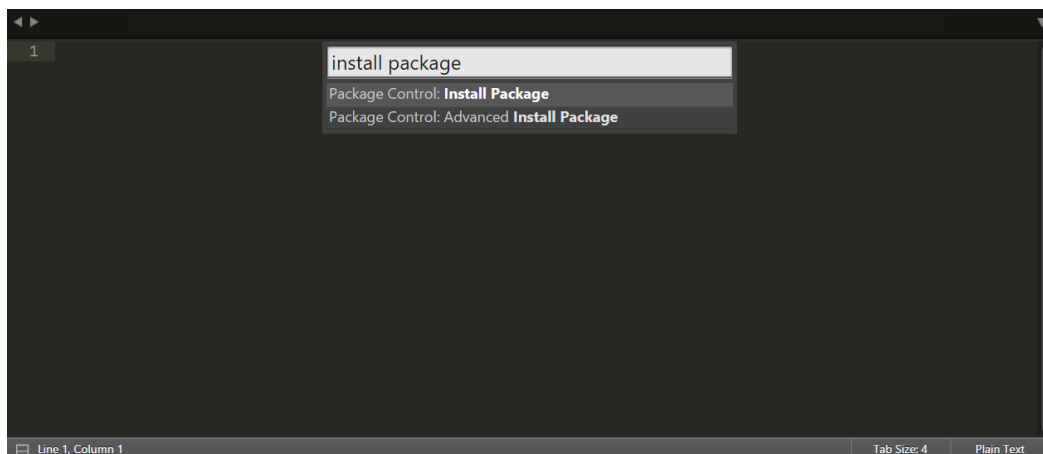
In Sublime Text, then, go to Tools → Command Palette (Ctrl + Shift + P / Command + Shift + P). Type in “Install Package”, and click on “Package Control: Install Package”. Also—if you prefer the keyboard—instead of clicking, you can navigate using the arrow keys and enter.

Usage Instructions ▶

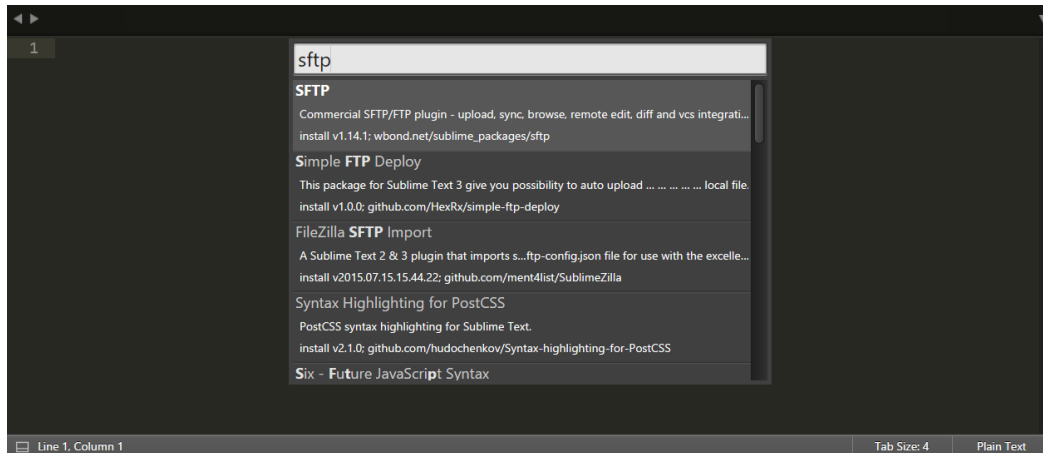
Manual

If for some reason the console installation instructions do not work for you (such as having a proxy on your network), perform the following steps to manually install Package Control:

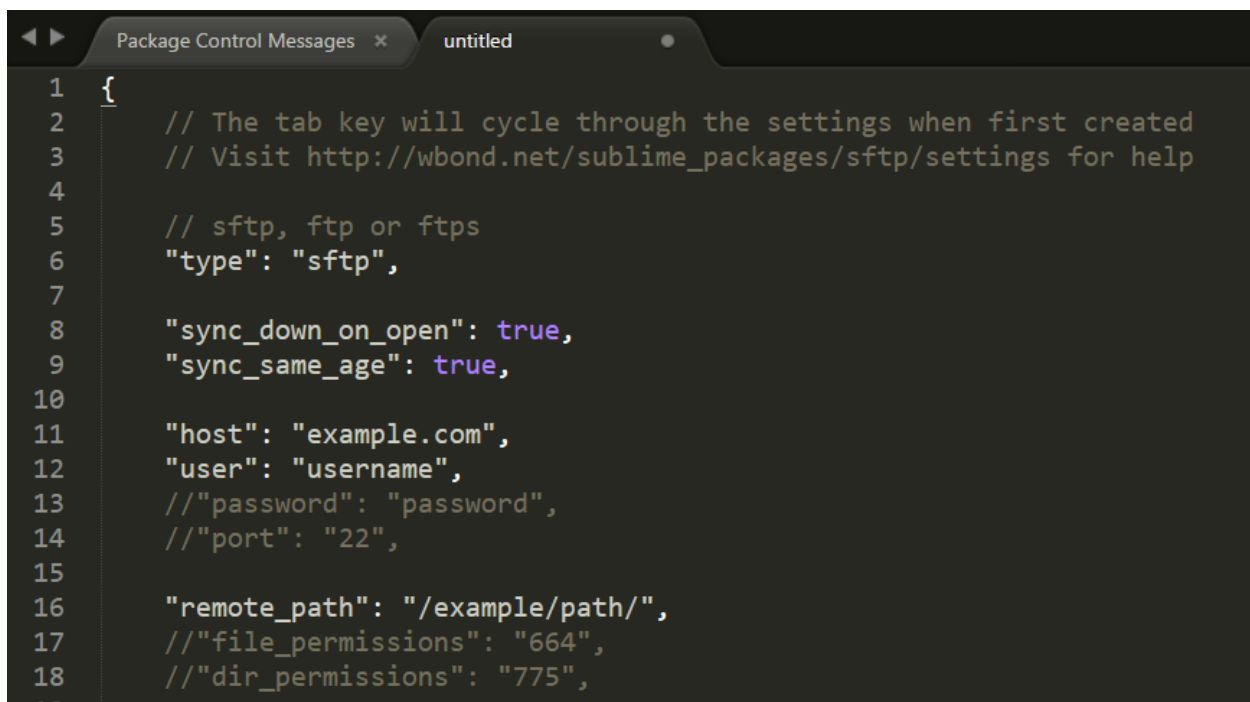
1. Click the `Preferences > Browse Packages...` menu
2. Browse up a folder and then into the `Installed Packages/` folder
3. Download `Package Control.sublime-package` and copy it into the `Installed Packages/` directory
4. Restart Sublime Text



Search for SFTP, then click on the one by wbond.net. This will install the SFTP plugin.



After installation, you will see the text titled Package Control Messages which you can ignore. Instead, browse to File → SFTP/FTP → Setup Server... and you'll get a page like this:



We need to edit the host, user, port, and remote_path fields. You can add the password so that you won't be prompted for your password every time you are accessing Shark, but the password is saved in plaintext. If you wish to do so, do so at your own risk.

- "host": "shark.ics.cs.cmu.edu"
- "username": "(your Andrew ID)"

- "port": "22" (uncomment this)
- "remote_path": see below

For remote_path, we need to figure out the absolute path (starting with /afs/.../private) to your private folder. To do so, go back to your terminal, and type in

pwd

to show the absolute path in your current folder. For instance:

```
root@FIRECROW: ~  
-bash-4.1$ pwd  
/afs/andrew.cmu.edu/usr16/msalim/private  
-bash-4.1$
```

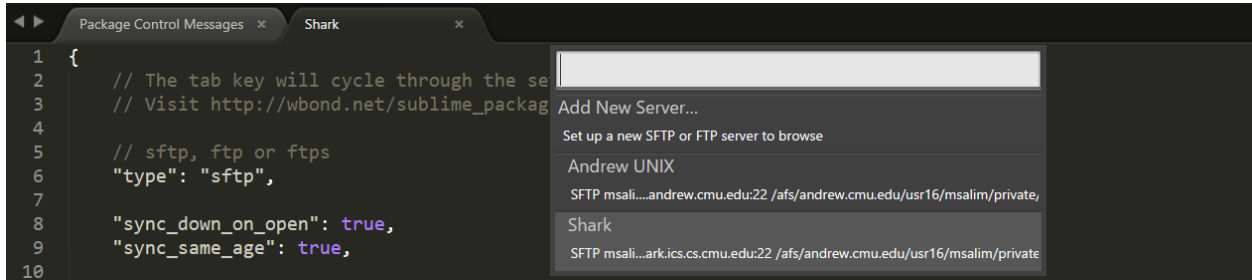
I would put in "/afs/andrew.cmu.edu/usr16/msalim/private" for my remote_path.

Here is an example of a filled-in one.

```
Package Control Messages x untitle  
1 {  
2 // The tab key will cycle through the settings when first created  
3 // Visit http://wbond.net/sublime_packages/sftp/settings for help  
4  
5 // sftp, ftp or ftps  
6 "type": "sftp",  
7  
8 "sync_down_on_open": true,  
9 "sync_same_age": true,  
10  
11 "host": "shark.ics.cs.cmu.edu",  
12 "user": "msalim",  
13 // "password": "password",  
14 "port": "22",  
15  
16 "remote_path": "/afs/andrew.cmu.edu/usr16/msalim/private",  
17 // "file_permissions": "664",  
18 // "dir_permissions": "775",  
19
```

Afterwards, save the file, and when prompted, insert the title of the connection (e.g. "Shark"). Don't change the folder—you should be given the correct folder when saving (sftp_servers).

Assuming that you do not save your password, you will then be prompted to enter the password. You can do so immediately. The general way to log in to SFTP is to go to File → SFTP / FTP → Browse Server... and click the connection you have just created.

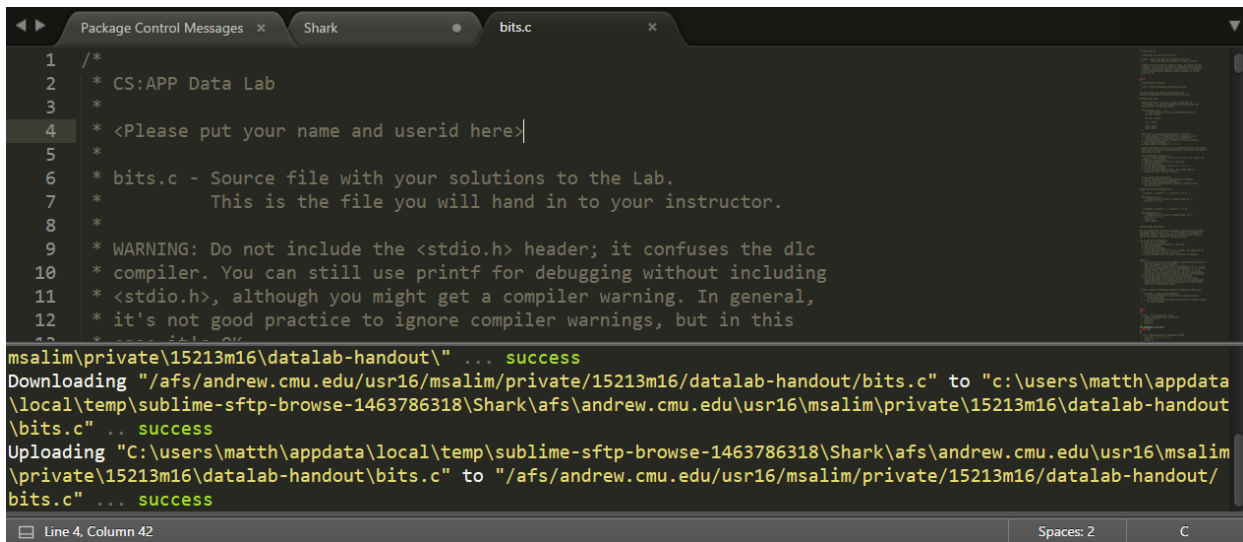


It will give you the password prompt. Given a correct password, you will be given a list of files and folders that exists in the given remote_path folder—you're all set!

6. Editing locally

You can now edit your assignment file (e.g. bits.c) locally and efficiently. To do so, browse to the folder that contains the file (e.g. datalab-handout), and click on the file and click Edit. You can enter folders by clicking on the folder name. (Of course, pointing and clicking can be substituted with arrow keys + Enter).

Now, whenever you save the file, it will save it on Shark, given that it says success (instead of failure) on the console.



If you want to save the file with another name (e.g. bits-old1.c for backup), you can do so by using the Save As... command, and the plugin will automatically upload the file onto Shark.

Careful: the SFTP plugin saves without confirmation, which makes your workflow using offline editor way more efficient. However, it's important to note that it will never ask you about overwriting a file with the same name! Be careful when using the Save As... command.

Warning: the SFTP plugin does not do file autosaving, so you must save your file and make sure that the success notification appears before closing the tab or Sublime Text entirely.

One note: the SFTP plugin may disconnect when the servers reboot, and it may disconnect after you put your computer to sleep. If it happens, it will try to reconnect and prompt you to enter your password again. However, it is urged to not take any risk and to save often!

7. Save-build-run loop

We said that this workflow is effective—and it is! Here's a typical use case which we call the save-build-run loop:

- When we're done editing the assignment, save the file.
- Try building the file on the Shark machine (make). If it fails to build, re-edit the file.
- If it builds successfully, run it. If it fails to run well, re-edit the file.

Since we frequently do re-editing and re-building, it is nice to be able to save fast and build fast. We can do so with two terminals with VIM, but we can also do that just as fast with Sublime Text and one terminal. Here's one way:

1. Open both Sublime Text and terminal. Go to the directory the assignment files are in.
2. When you're done editing, press Ctrl + S (or Command + S).
3. Type "make" in terminal (or press the up arrow key) and press enter.
 - If it doesn't build, go back to step 2.
4. If it works, run the test (e.g. ./driver.pl).
 - If it doesn't run as expected, go back to step 2.

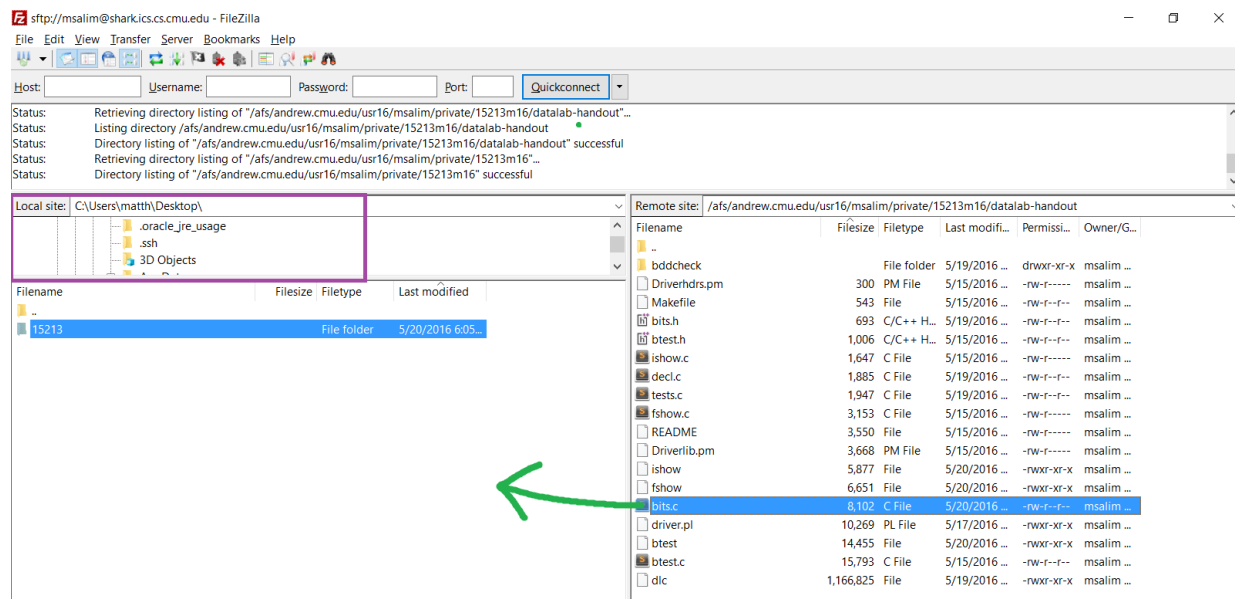
Typing Ctrl + S and pressing the up arrow key (at most twice) is efficient—and for non-avid command line users, using Sublime for its GUI, desktop keyboard shortcut, and mouse-based navigation may be more effective for productivity than using command line tools.

An additional monitor will be helpful!

8. Downloading assignment handin from Shark

Follow the writeup for details on which file(s) to hand in! If you find scp to be more efficient, feel free to skip this step and go to step 9.

On FileZilla's right column, navigate to the folder that contains the file(s) you need to hand in. Drag that file onto the left column's bottom row and drop it—into a location in your computer. You can navigate to your desired copy location in your computer on the left column's top row, as shown in purple.



Remember the local directory that contains the file for step 9.

9. Uploading assignment handin to Autolab

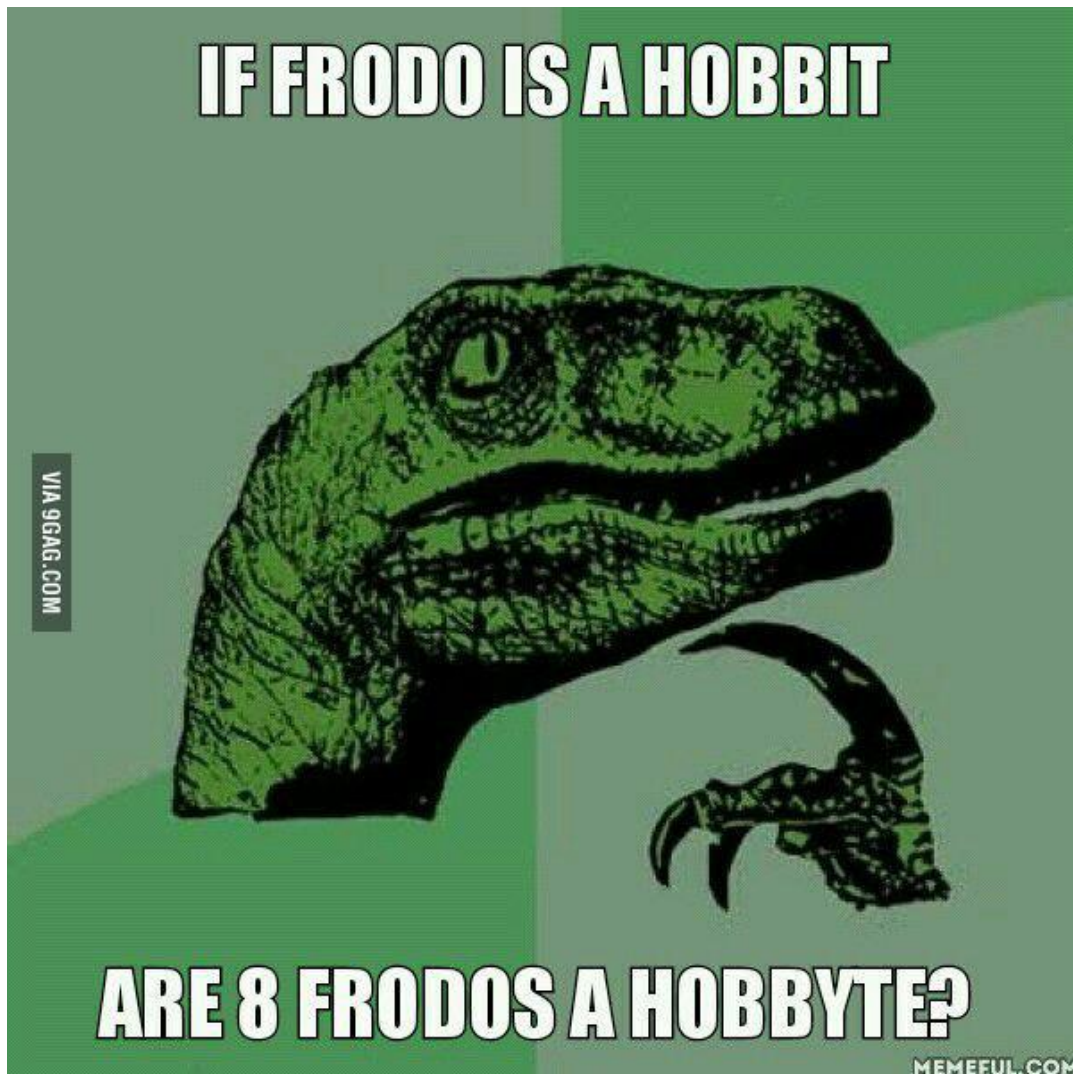
Now, all you need to do is to open [Autolab](#), click the assignment, and submit the file.

That's it for now!

If you have any questions and/or suggestions regarding this writeup, feel free to post a question on Piazza, or email me directly at msalim@cmu.edu. Good luck with Datalab! Oh, and also:

- Read the writeup. Please read the writeup!
- Ask questions—on Piazza, in Office Hours, or through one-on-one video calls.
- Start early!

-Matthew



From <http://9gag.com/gag/agNDXXr/if-frodo-is-a-hobbit>