Principles of Urban Informatics
Class I
Instructor: Mohit Sharma
TA: Vamsi, Aishwarya
Sep 6. 2017
Contributors: Mohit Sharma, Christian Moscardi, Alex Shannon

Setting up your machine

CDF Jupyter notebook

Windows users, you guys are using bitvise client, keep using that. It is the most reliable solution for Windows.

1. Creating alias (Mac and Linux Users only)

a. Mac Users:

i. Press CMD + SpaceBar; type terminal; and hit enter

ii. type: `nano ~/.bash_profile`

iii. Add this line: `alias notebook="ssh cuspid@gw.cusp.nyu.edu -L 8000:compute.cusp.nyu.edu:8000"`. (you can replace "`notebook`" with a name of your choice)

iv. Save the file by typing `Ctrl + o` and exit using `Ctrl + x`

v. Either close the terminal and open it again or type `source ~/.bash_profile`

b. Linux Users:

i. Press `Ctrl + Alt + t` OR press windows/ cmd key and search terminal

ii. type: `nano ~/.bashrc`

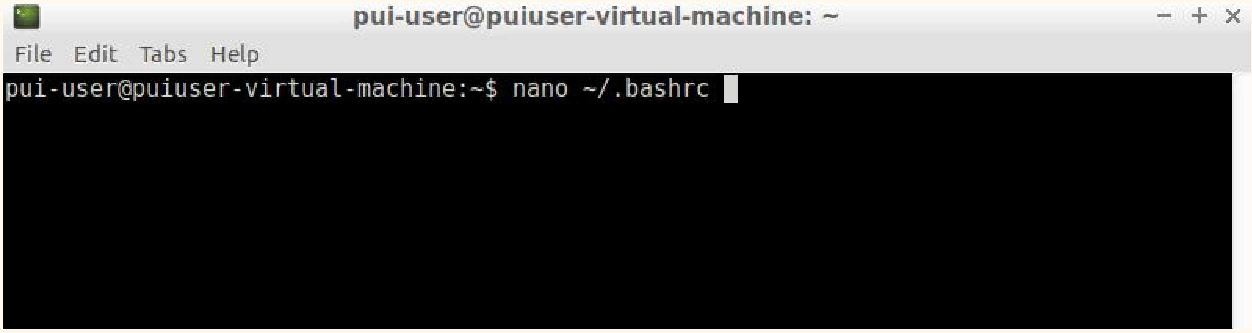
iii. Add this line: `alias notebook="ssh cuspid@gw.cusp.nyu.edu -L 8000:compute.cusp.nyu.edu:8000"`. (you can replace "`notebook`" with a name of your choice)

iv. Save the file by typing `Ctrl + o` and exit using `Ctrl + x`

v. Either close the terminal and open it again or type `source ~/.bashrc`

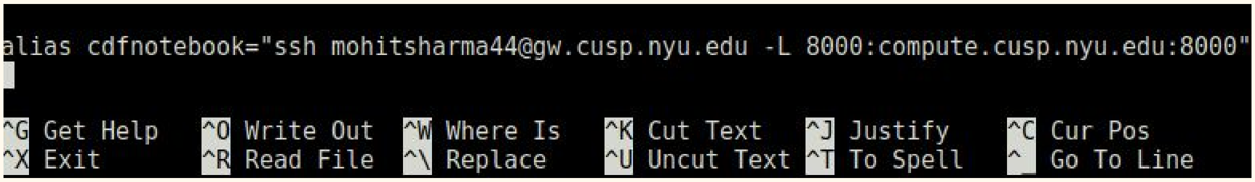
Some Screenshots that may help:

Opening bashrc (for linux) [Mac users replace bashrc with bash_profile]



```
pui-user@puiuser-virtual-machine:~$ nano ~/.bashrc
```

Adding alias

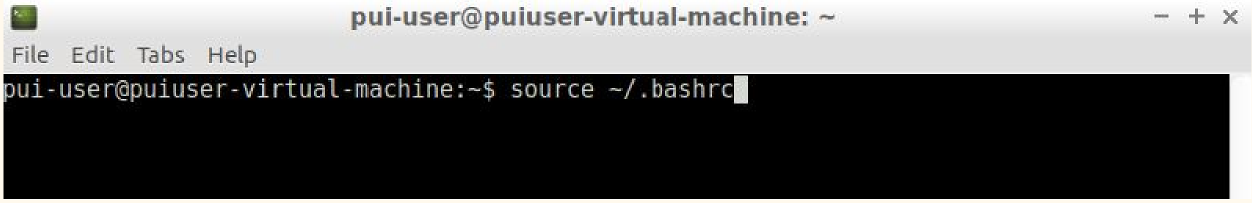


```
alias cdfnotebook="ssh mohitsharma44@gw.cusp.nyu.edu -L 8000:compute.cusp.nyu.edu:8000"

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Close the editor by typing Ctrl + o and Ctrl + x

Source bashrc (for linux) [Mac users, replace bashrc with bash_profile]



```
pui-user@puiuser-virtual-machine:~$ source ~/.bashrc
```

Setting up passwordless access to CDF (for Mac and Linux users only)

1. Opening Terminal

a. Mac Users: Press CMD + SpaceBar; type terminal; and hit enter

b. Linux Users: Press `Ctrl + Alt + t` OR press windows/ cmd key and search terminal

2. Generate SSH keys

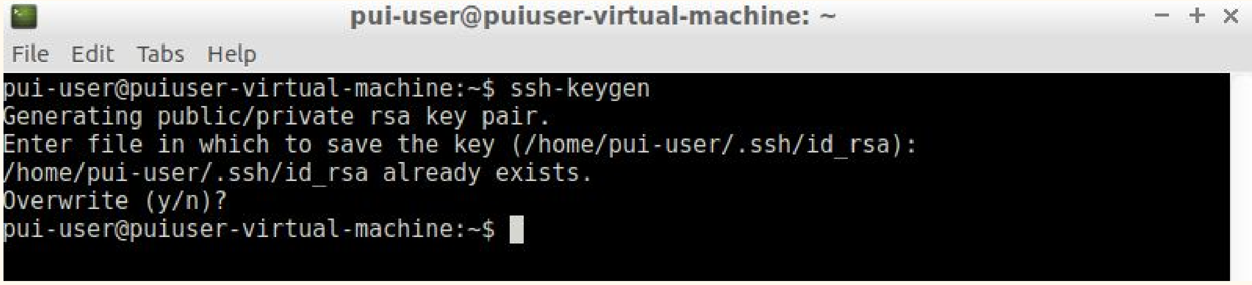
To login to cdf, you can either use your passwords (which we don't want to) or use a secure private key. Let's generate a secure key

a. Mac and Linux Users [open terminal]:

i. `ssh-keygen` (and then hit enter a bunch of times.. Don't worry about the output for now).

ii. If it asks you to "`overwrite (y/n)`" press `Ctrl + c` or simply press Enter to not overwrite. This means that you already have an ssh key that you can use. You don't need to generate a new one.

Screenshot: I already have keys

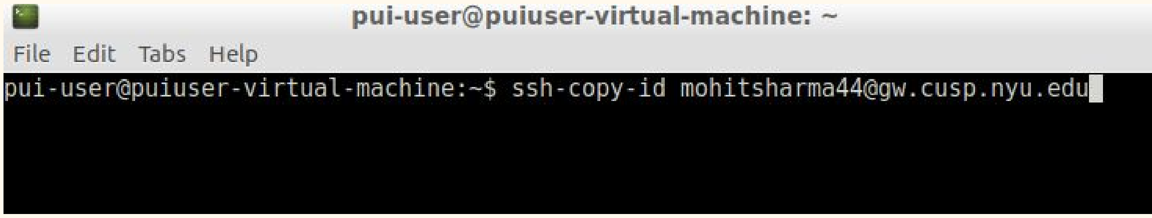


```
pui-user@puiuser-virtual-machine:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pui-user/.ssh/id_rsa):
/home/pui-user/.ssh/id_rsa already exists.
Overwrite (y/n)?
pui-user@puiuser-virtual-machine:~$
```

3. Copy SSH keys

a. Mac and Linux Users:

i. `ssh-copy-id` cuspid@gw.cusp.nyu.edu and hit enter. This is the last time you need to enter your password.



```
pui-user@puiuser-virtual-machine:~$ ssh-copy-id mohitsharma44@gw.cusp.nyu.edu
```

4. Test the alias and passwordless authentication

a. Mac and Linux Users:

i. Open terminal and type your alias name. For example, in my case it is notebook. (This time it shouldn't prompt you for password). You need to keep this terminal open as long as you are using the notebook

ii. Open your web browser and in the address bar, type: https://localhost:8000 and you should be welcomed with jupyterhub!

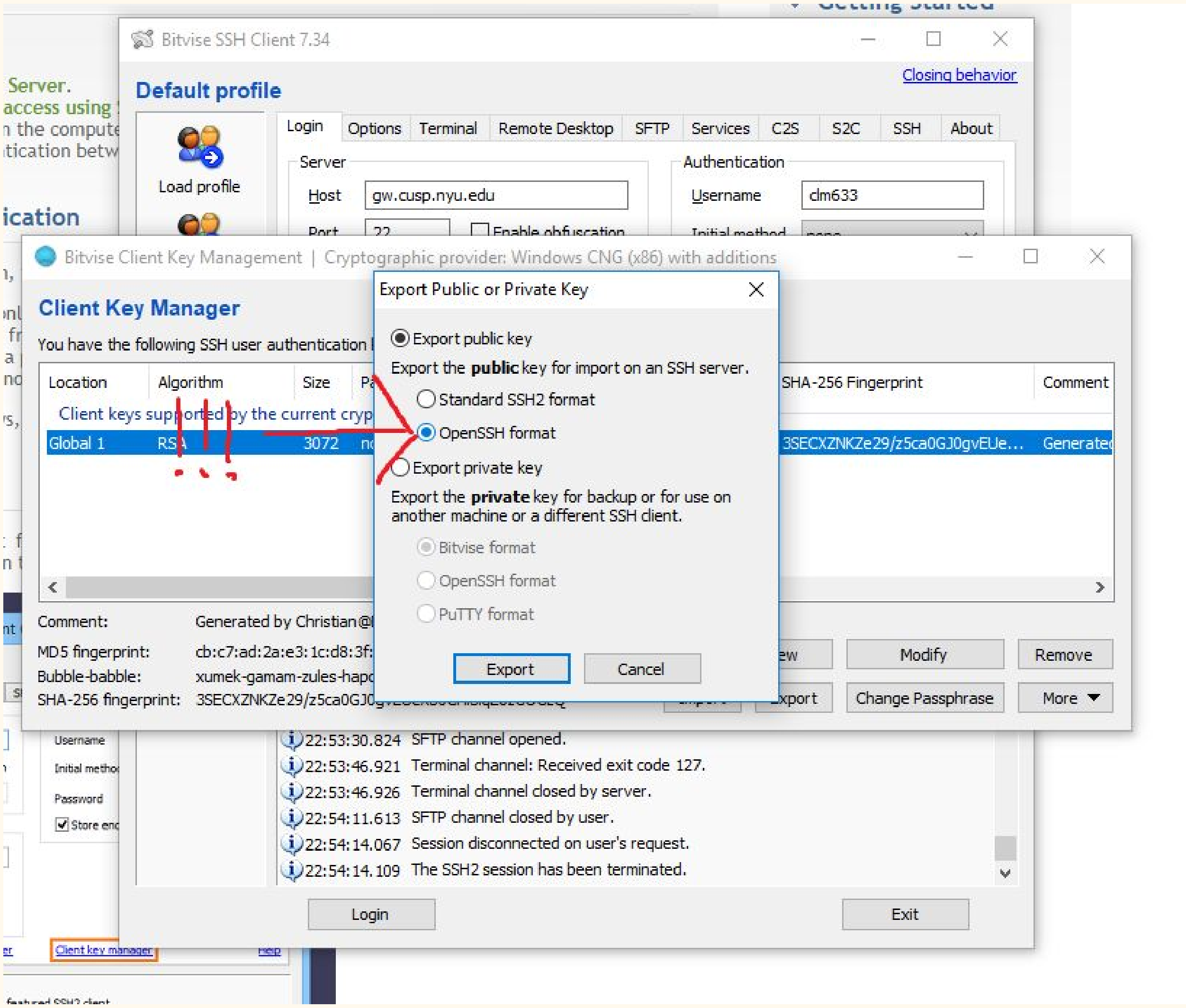
iii. Remember, whenever you are done, go back to the same terminal where you typed your alias and type `Ctrl + c` a bunch of times and then you can safely close the terminal.

Setting up passwordless access to CDF (for Windows users only)

- Christian Moscardi

1. Follow the instructions under the “Generate Public/Private Keypair” heading here. Make sure to STOP at the end of this section - do NOT go on to “Upload public key to Bitvise SSH Server”

2. On the same page, follow the instructions under “Alternative: Export public key”. One small change: when doing this, **make sure to check the “OpenSSH format” box** instead of the “standard SSH2 format” box. You’ll save this public key as a file - remember where you saved it, as you’ll need it later. See the photo below:



Host gw.cusp.nyu.edu

Username dm633

Port 22

□ Enable obfuscation

You have the following SSH user authentication

Client keys supported by the current cryp

Global 1 RSA 3072 3SECXZNKZe29/z5ca0GJ0gvEUe... Generated

● Export public key

Export the **public** key for import on an SSH server.

○ Standard SSH2 format

● OpenSSH format

○ Export private key

Export the **private** key for backup or for use on another machine or a different SSH client.

○ Bitvise format

○ OpenSSH format

○ PuTTY format

Comment: Generated by Christian@

MD5 fingerprint: cb:c7:ad:2a:e3:1c:d8:3f:

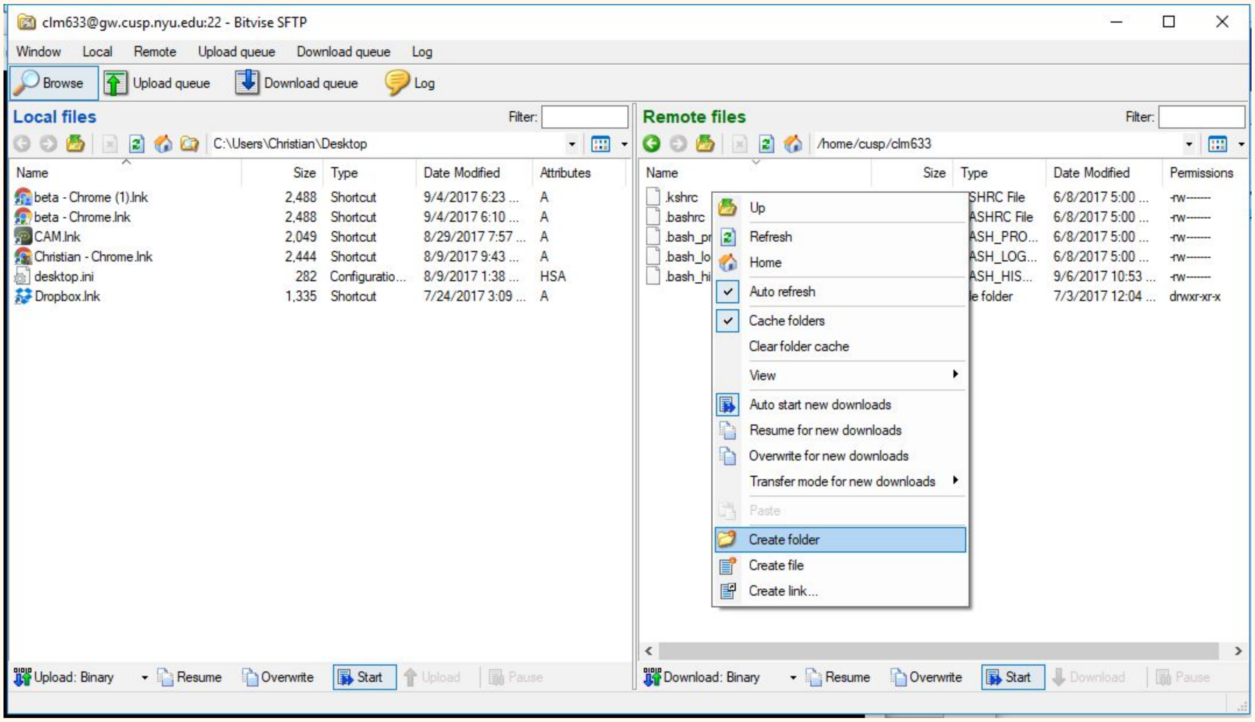
Bubble-babble: xumek-gamam-zules-hapo

SHA-256 fingerprint: 3SECXZNKZe29/z5ca0GJ0

```
22:53:30.824 SFTP channel opened.
22:53:46.921 Terminal channel: Received exit code 127.
22:53:46.926 Terminal channel closed by server.
22:54:11.613 SFTP channel closed by user.
22:54:14.067 Session disconnected on user's request.
22:54:14.109 The SSH2 session has been terminated.
```

3. Now, SSH into gw.cusp.nyu.edu as usual (with your password).

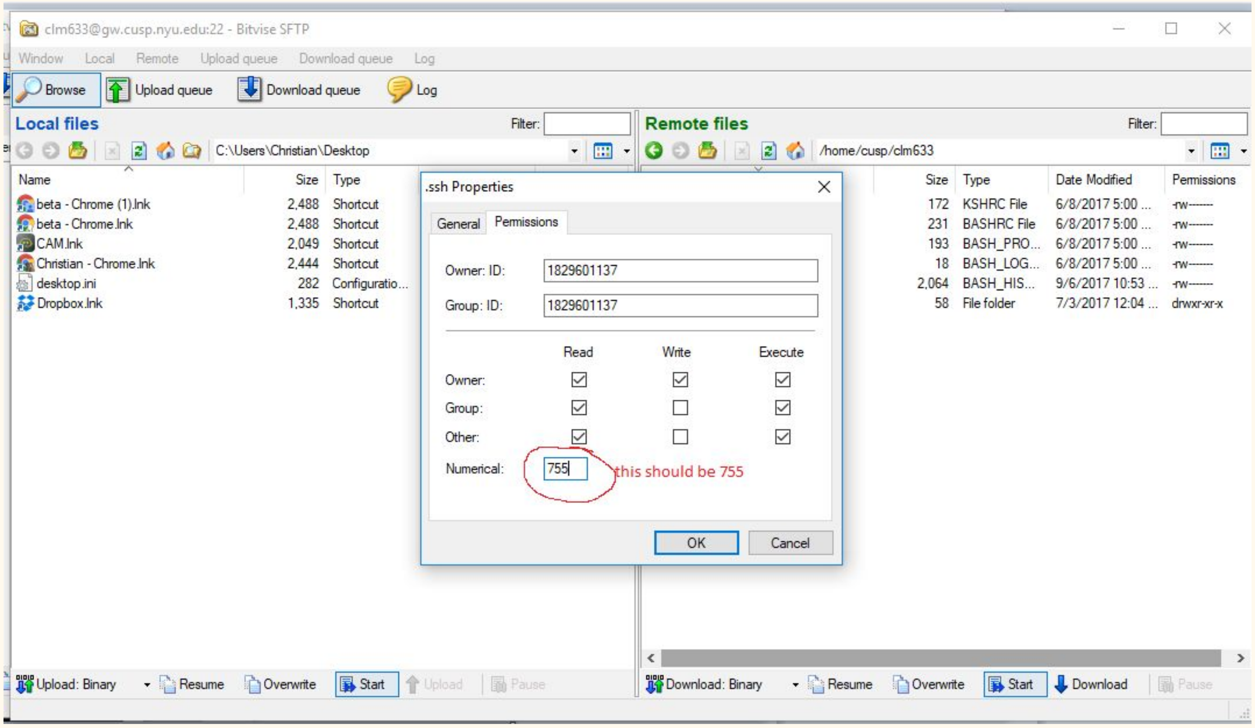
4. In the bitvise file browser, right-click and create the folder **.ssh** in your home directory. If it exists, great, don't worry. (Screenshot below)



Name	Size	Type	Date Modified	Attributes
beta - Chrome (1).lnk	2,488	Shortcut	9/4/2017 6:23 ...	A
beta - Chrome.lnk	2,488	Shortcut	9/4/2017 6:10 ...	A
CAM.lnk	2,049	Shortcut	8/29/2017 7:57 ...	A
Christian - Chrome.lnk	2,444	Shortcut	8/9/2017 9:43 ...	A
desktop.ini	282	Configuratio...	8/9/2017 1:38 ...	HSA
Dropbox.lnk	1,335	Shortcut	7/24/2017 3:09 ...	A

Name	Size	Type	Date Modified	Permissions
.kshrc		SHRC File	6/8/2017 5:00 ...	-rw-------
.bashrc		ASHRC File	6/8/2017 5:00 ...	-rw-------
.bash_pr		ASH_PRO...	6/8/2017 5:00 ...	-rw-------
.bash_lo		ASH_LOG...	6/8/2017 5:00 ...	-rw-------
.bash_hi		ASH_HIS...	9/6/2017 10:53 ...	-rw-------
		e folder	7/3/2017 12:04 ...	drwxr-xr-x

5. Now, right click on the folder (whether or not it existed) and select **“Properties”**. Then, click on the **“Permissions”** tab and set the numeric value to **755**. Hit OK. See screenshot below.



Name	Size	Type
beta - Chrome (1).lnk	2,488	Shortcut
beta - Chrome.lnk	2,488	Shortcut
CAM.lnk	2,049	Shortcut
Christian - Chrome.lnk	2,444	Shortcut
desktop.ini	282	Configuratio...
Dropbox.lnk	1,335	Shortcut

Size	Type	Date Modified	Permissions
172	KSHRC File	6/8/2017 5:00 ...	-rw-------
231	BASHRC File	6/8/2017 5:00 ...	-rw-------
193	BASH_PRO...	6/8/2017 5:00 ...	-rw-------
18	BASH_LOG...	6/8/2017 5:00 ...	-rw-------
2,064	BASH_HIS...	9/6/2017 10:53 ...	-rw-------
58	File folder	7/3/2017 12:04 ...	drwxr-xr-x

Owner: ID: 1829601137

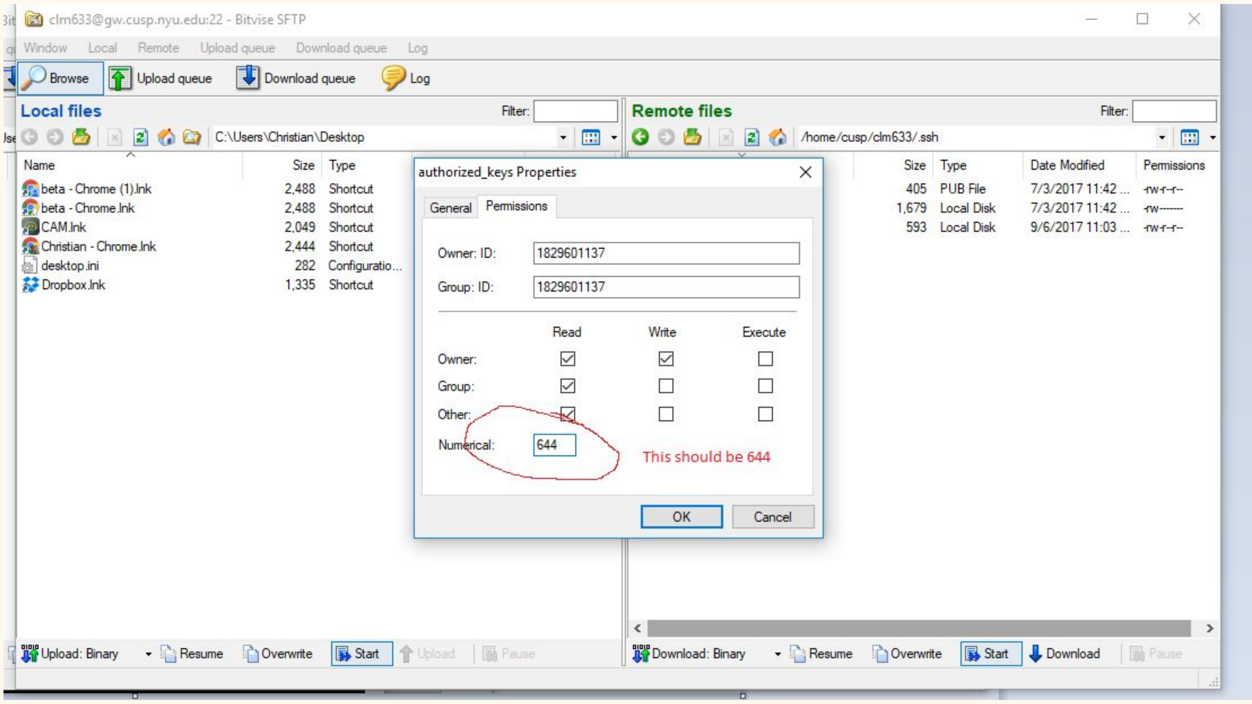
Group: ID: 1829601137

	Read	Write	Execute
Owner:	☑	☑	☑
Group:	☑	□	☑
Other:	☑	□	☑

Numerical: 755

this should be 755

6. Click into **.ssh/** and right-click to create a file. Name it **authorized_keys**. Then, go into Properties -> Permissions (as with the .ssh folder) and set the permissions to **644**. See screenshot below.



C:\Users\Christian\Desktop

Name	Size	Type
beta - Chrome (1).lnk	2,488	Shortcut
beta - Chrome.lnk	2,488	Shortcut
CAM.lnk	2,049	Shortcut
Christian - Chrome.lnk	2,444	Shortcut
desktop.ini	282	Configuratio...
Dropbox.lnk	1,335	Shortcut

/home/cusp/clm633/.ssh

Size	Type	Date Modified	Permissions
405	PUB File	7/3/2017 11:42 ...	-rw-r--r--
1,679	Local Disk	7/3/2017 11:42 ...	-rw-------
593	Local Disk	9/6/2017 11:03 ...	-rw-r--r--

authorized_keys Properties

General | Permissions

Owner: ID: 1829601137

Group: ID: 1829601137

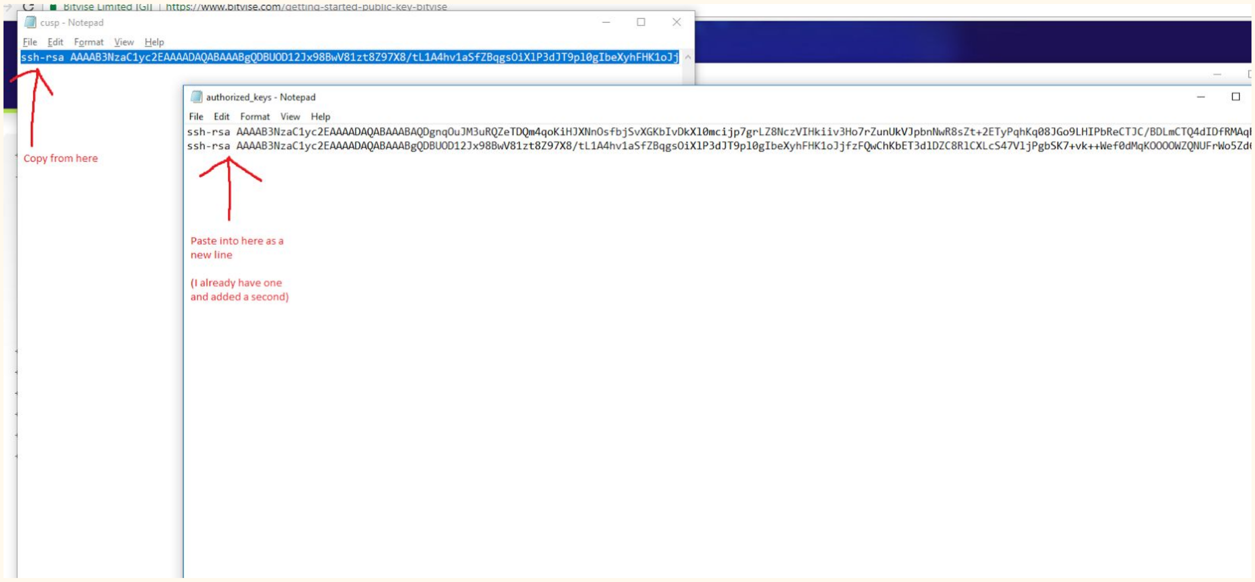
	Read	Write	Execute
Owner:	☑	☑	□
Group:	☑	□	□
Other:	☑	□	□

Numerical: 644

This should be 644

OK | Cancel

7. Copy the contents of the SSH key you exported in step 2. into the authorized_keys file. I used Notepad to do this… note, I have 2 keys, so you will probably just have 1 line here.


Bitvise Limited [GI] | https://www.bitvise.com/getting-started-public-key-bitvise
File Edit Format View Help
File Edit Format View Help

cusp - Notepad

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDBUOD12Jx98BwV81zt8Z97X8/tL1A4hv1aSfZBqgsOiX1P3dJT9pl0gIbeXyhFHK1oJj
```

Copy from here

authorized_keys - Notepad

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDgnqOuJM3uRQZeTDQm4qoKiHJXNnOsfbjSvXGKbIvDkX10mcijp7grLZ8NczVIHkiiv3Ho7rZunUkVJpbnNwR8sZt+2ETyPqhKq08JGo9LHIPbReCTJC/BDLmCTQ4dIDfRMAq
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDBUOD12Jx98BwV81zt8Z97X8/tL1A4hv1aSfZBqgsOiX1P3dJT9pl0gIbeXyhFHK1oJjfzFQwChKbET3dlDZC8R1CXLcS47V1jPgbSK7+vk++Wef0dMqKOOOOWZQNUFrWo5Zd
```

Paste into here as a new line

(I already have one and added a second)

8. Back in the bitvise tutorial, follow the instructions under the "Configure Public Key Authentication" header and you should be good to go.

Local Jupyter notebook

There are two ways of running things locally on your machine.
1. Installing a virtual machine and then installing things on the "guest" machine(safest)
2. Installing packages on your "host" machine (a.k.a your laptop)

Virtual Machine

1. Download Vmware or Virtualbox(for Windows, Mac and Linux):

a. Windows and Linux users, download VMware Workstation Free from here: bit.ly/puilinuxusers

b. Mac users, download virtualbox from here: bit.ly/puimacusers

2. Download prebuilt Linux Virtual Machine (windows, linux and mac):

a. Download image from here: bit.ly/puilinux (~3.5GB) Password: Cusp_19

b. Extract the above image (simply double clicking should work)

c. Save it in a folder and make sure you never delete it.

3. We'll take it from here in the next lecture..

Local Installation

Make sure you understand the risks of installing things locally. Most likely you shouldn't be able to break your system but I won't underestimate you guys ;)

1. Install brew (for Mac users only):

a. Open Terminal

b. Install brew (it is basically a package manager for Mac, just like App Store on Steroids):

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

c. It will prompt you for a password. This is your local machine's password (not your cusp password)

d. Let it do its thing.

2. Setup the system:

a. In your terminal:
Mac users: `brew install python2 python3` ← Mac users won't need anything else
Linux users: `sudo apt-get install python python-dev python-pip python3 python3-dev python3-pip`

b. Now, using pip3 (which is automagically installed by brew), install virtualenv, ipython and jupyter

```
sudo pip3 install virtualenv
pip3 install ipython jupyter --user
```

c. If you want to use python2 and python3 both kernels in jupyter notebook, run these commands:

```
python2 -m pip install ipykernel
python2 -m ipykernel install --user
```

i. Now, create a virtual environment for your PUI class (in a `devel` directory, for example) and activate it:

```
mkdir -p ~/devel/pui17
cd ~/devel/pui17
```

`virtualenv venv --python=python3` ← you can select python2 if you want

```
source venv/bin/activate
```

d. Finally, you can install the following scientific packages in your pui17 environment:

```
pip install numpy scipy matplotlib pandas scikit-learn
```

This command will install all the above packages in virtual environment! We'll see in the class why this is important.

e. If you want, you can also setup custom python kernel for your projects:

```
python -m ipykernel install --user --name venv
--display-name "Python venv"
```

← You can rename venv to whatever your virtualenv name is and replace the display-name from "Python venv" to whatever you would like to see when you run jupyter notebook.

f. To Exit from the virtualenv, simply type `deactivate`

FullScreen Mode:

For VirtualBox

- Alex Shannon

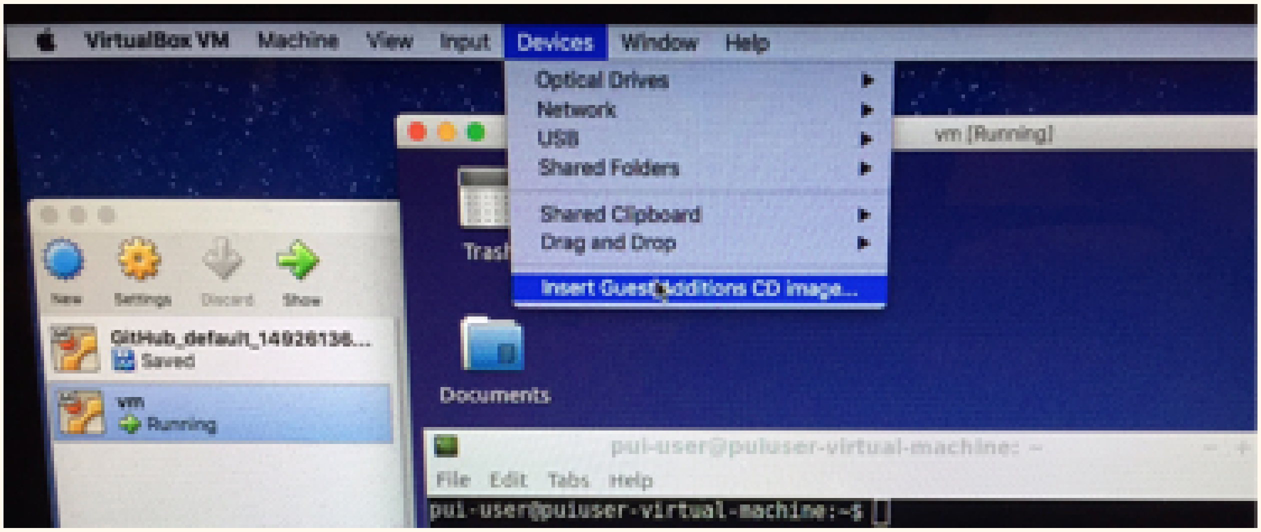
You’ve got your virtual machine up and running, and you’re ready to start coding, so you go to enter fullscreen, and… *wait*… Something’s not right here – the VM window doesn’t scale with the view. You’ve just spent all this time to work in an annoyingly small box? I don’t think so… Let’s take some steps to get things working more naturally.

(corresponding photos below)

[1] Select the VM window – on the bar at the top of the screen; select “*Devices*” and scroll to the bottom. Select “*Insert Guest Additions CD Image…*” If this works – great! You can skip to [4]. If you get an error however, you need to add an optical drive. To do so, go back to the VitrualBox tab [2], and select “*Settings*”; once in settings, [3] go to the “*Storage*” tab, and select “*Controller: SATA*” – below this you will find that you can add an optical drive; do this, and select “*Leave Empty*”. Now you should be able to complete step [1] successfully, and a “*VBOXADDITIONS…*” [4] should be on your VM Desktop (if it opens in File Manager, you can close the window).

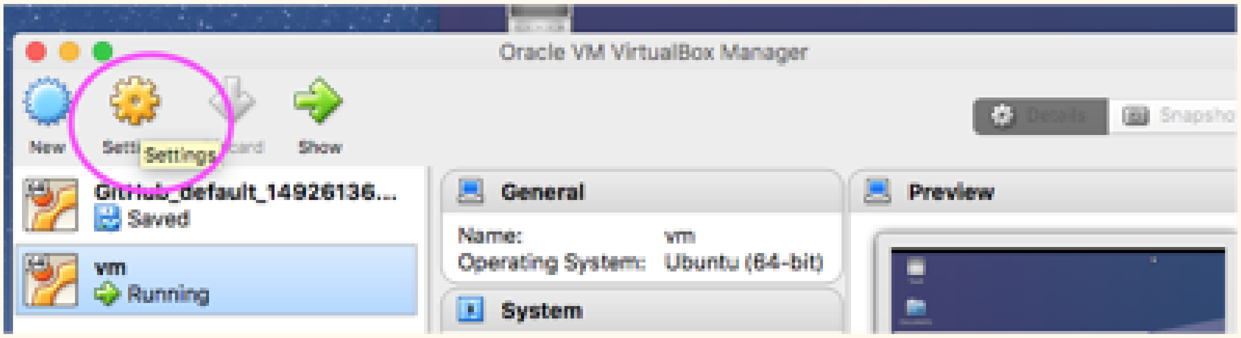
[5] Use ‘*Command + Click*’ to show options; select open in Terminal. Once in the terminal [6], run the code ‘sudo ./VBoxLinuxAdditions.run’ – let it run (remember your password is ‘*Cusp_19’*!), and you should be good to go! Use ‘*Command +f*’ to toggle between full screen and window view.

[1]



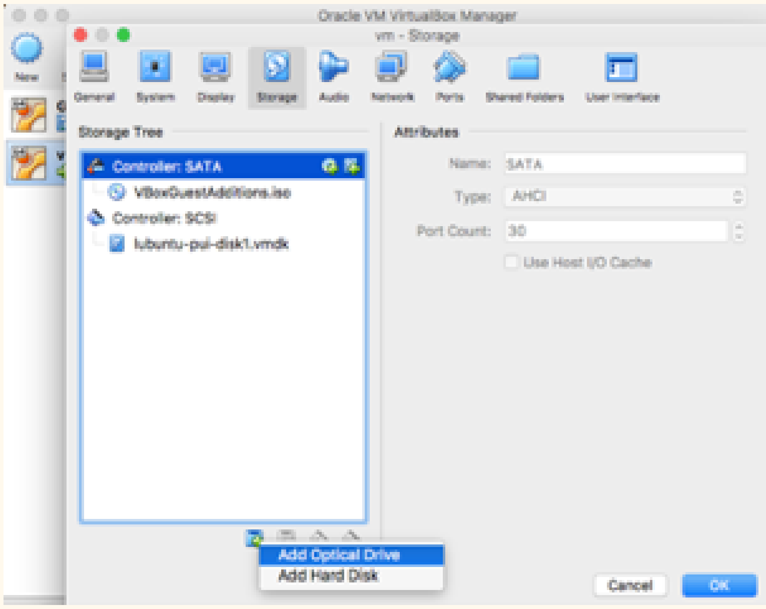
```
pui-user@puiuser-virtual-machine:~$
```

[2]



Name: vm
Operating System: Ubuntu (64-bit)

[3]



Attributes

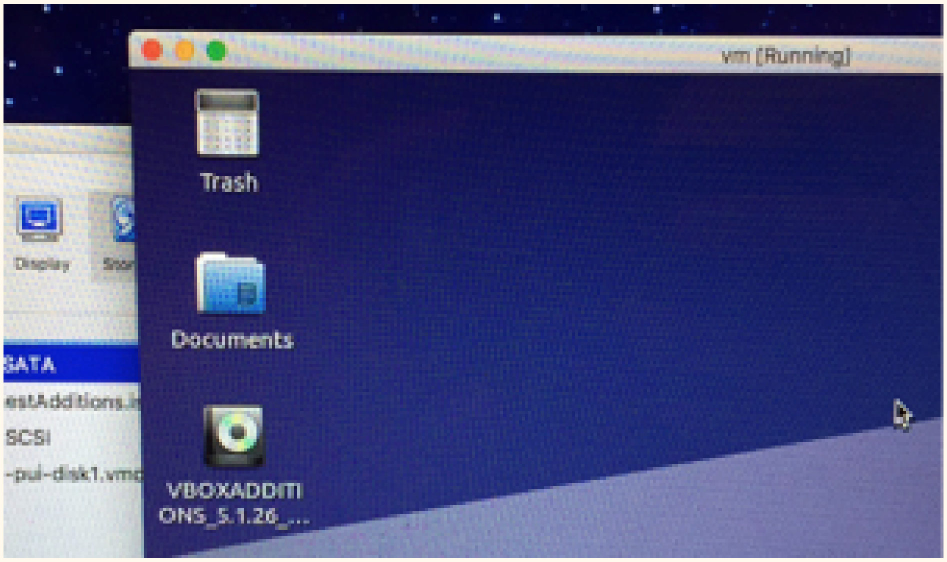
Name: SATA

Type: AHCI

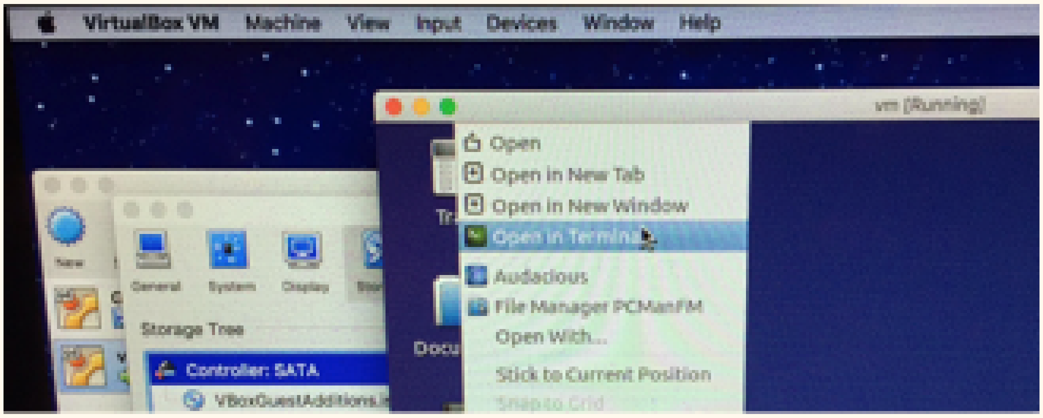
Port Count: 30

□ Use Host I/O Cache

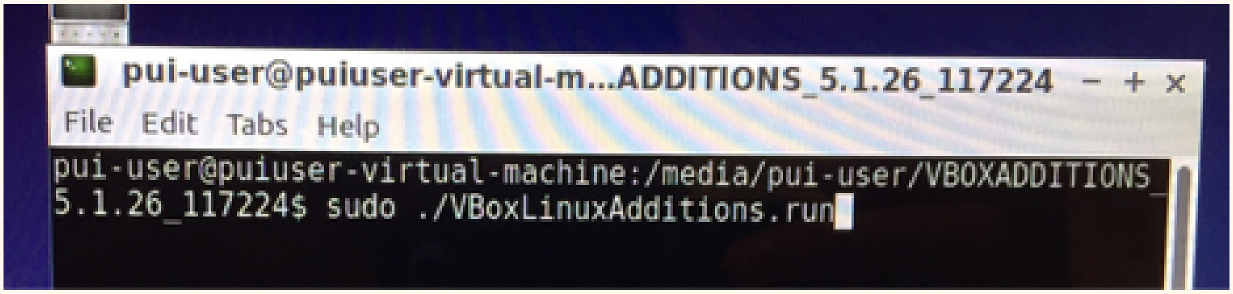
[4]



[5]



[6]



```
pui-user@puiuser-virtual-machine:/media/pui-user/VBOXADDITIONS_
5.1.26_117224$ sudo ./VBoxLinuxAdditions.run
```

Make sure to participate in the Poll

Participate

http://etc.ch/rrCp

View Results [real-time]

https://directpoll.com/r?XDbzPBd3ixYqg8ZEB9sOmlYCnM2hDOwSFhLYxdxe