Working on a Server 101

Set up and Installation:

Before attempting to get on a server, make sure that the following is installed on your computer:

1) XQuartz: enables graphics to be displayed on a computer from a server
2) XCode: can be downloaded through the app store (make sure you have the most recent macOS update)
3) Cisco AnyConnect: application used to connect to a vpn which is necessary when working remotely from home or any place that is not UCSD.
4) Homebrew: Homebrew installs packages that Apple (or your Linux) system does not have (such as wget, etc.).

For any questions about using the Unix Shell, go to software carpenter website which will have documentation, lessons and tutorials on how to use Unix. Use links below:
http://swcarpentry.github.io/shell-novice/03-create/index.html
or
http://swcarpentry.github.io/shell-novice/

In addition to these two applications, in order to get many functions necessary for every Unix coding, one needs to download homebrew. Homebrew includes functions such as wget and others. Here is a link to the homebrew installation process. This is only necessary for mac computers.

Getting on a server and transferring files:

If off campus, use the cisco UCSD vpn. To download cisco on to a computer, use the link https://blink.ucsd.edu/technology/network/connections/off-campus/VPN/index.html . Using a vpn when off campus is necessary because in order to login to a ucsd server, one needs to be on UCSD-Protected internet. Name of the vpn for ucsd is: vpn.ucsd.edu

If you are on campus, make sure are on UCSD-Protected (if it still does not work, go on to the vpn)

To log on to a server use code:

```
ssh –Y username@server
```

where ssh =>used to communicate between two machines
    -Y => enables graphics on server to be displayed
    username = username on server
    server = server website address

example:

```
ssh –Y lcolosi@fig.ucsd.edu
```

where lcolosi => username
        @fig.ucsd.edu => @server

One can also place an IP of the server after the username@ where an IP is the identity of a particular machine you are trying to communicate with (internet protocol)

Next, you will be asked to put in a password. This password is your UCSD campus password (the password you use to get into tritonlink or to access your UCSD email account)

When on the server, if the graphics are still not showing up, try to create a text editor (when you are off the server and are just in your computers home directory in bash) in order to explicitly tell the computer to bring up graphics when on the server along with a few other things. Use the following command:
Command: vi name
where the flag “name” will either bring you to a pre-existing text editor if the name is already used or will name a new text editor if the name has not been used.

Purpose: vi is a text editor document where one can establish a command state which will run a series of commands. In this case, tell the computer that all graphics on the host server should be displayed. Along with this command, we also tell the computer to log into fig.

Once the text editor is up, press i (for insert) in order to get into insert mode and be able to write on the text editor

Type the following to enable graphics:
```
Host server
          Hostname server address
          User username
       ForwardX11 yes
```

Example:
```
     Host fig
          Hostname fig.ucsd.edu
          User lcolosi
          ForwardX11 yes
```

Press esc to bring the text editor to command mode
and then write the following
```
          :wq
```
where : gets into command mode
            w execute command
            q quit

Side note: nano is also a text editor, however, it may have other syntax not covered here.

If that does not work, another way is to start up the XQuartz application. In the toolbar on the top left of your screen, click on applications and then in the drop down menu, click on terminal. This will bring up a screen on your screen a terminal window. From here, type in the following code:

```
ssh -o ForwardX11=yes user@server
```

Ex:

```
ssh -o ForwardX11=yes lcolosi@fig.ucsd.edu
```

Next, type in your password. Once on the server, bring up the matlab program using one of the two commends below. The graphics should come up as matlab loads. Note however that the graphics seem to really lag or be completely unresponsive when using this method.

If that does not work, another way is to type in the same line of code as before but into the regular terminal on your computer. XCode and XQuartz must be downloaded on to your computer!!!

Now that the graphics are working, start-up Matlab on the server by simply typing:

i. For matlab R2016b version: matlab
ii. For matlab R2018b version: (starting at to base directory of the server) /usr/local/MATLAB/R2018b/bin/matlab

To look for a file in any directory use the following:
which filename

Now that Matlab is working, the data and pre-existing mfiles in order to run matlab must be placed on the server:

Begin by making folders for the data and mfiles:

1) ssh –Y username@server (get onto server)
2) type in password
3) pwd (check to see what directory you are in (this is not really needed because you should always be in the home directory i.e. /home/username when you get on the server)
4) mkdir filename (create folder in server)
   a. i.e. mkdir data
   b. i.e. mkdir mfiles
5) ls (check to make sure the new folder is there)

Once the folders are created on the server, place the data and the mfiles on the computer onto the server. This is done by using rsync:

rsync copies files, data, figures, etc. from the any machine to another machine. rsync has a couple very nice features:

1) If the syncing/downloading process has been interrupted by say closing the computer, when the computer is opened again, the files will continue to download from the place it was stopped
2) After the data is downloaded, you can check that all of the data is at the desired destination (the server or your computer) by running the same code again where rsync will go through each piece of data and see if it is downloaded yet (if the file is already there rsync moves on without copying the same file again)

Note that rsync will only transfer data between two machines. It will not transfer between the cloud and a machine. In order to do this, use rclone (procedure to get rclone on will be discussed later).

In order to use rsync to bring files from your computer over to the server, use the follow example as a template:
1) Begin by being in your home directory in your computer (i.e. /User/username)
2) Start a screen: screen –S rsync_CCMP2_data
3) Write the following code in the screen terminal:

```
rsync -av --partial --progress ~/Documents/Research\ Lab/surfacewaves/IFREMER-data -e ssh
lcolosi@fig.ucsd.edu:/home/lcolosi/test
```

where ~/Documents/Research\ Lab/surfacewaves/IFREMER-data = source (where and what data you are transferring)

~ = tells the computer to go to the home directory

-e ssh lcolosi@fig.ucsd.edu:/home/lcolosi/test = destination

-e = tell the computer that the destination is remote

Now that the mfiles and the data are on the server, programs can be ran and figures can be saved on the server without taking up space on your computer.

In order to transfer files from the cloud (i.e. google drive) to the server, use rclone:

Installing rclone: Go on to the following website and follow the installation steps:
https://github.com/pageauc/pi-timolo/wiki/How-to-Setup-rclone

After installing rclone, go onto your home directory in your computer and start a new screen.

Use the following code to transmit data from the cloud onto a server:

```
rclone copy source destination
```

Example for data saved on google drive:

```
rclone copy source: path on google drive to data from home directory on computer destination:
directory on server
```

If you want to bring data from one server to another, use the wget command. Before executing this command, make sure of the following:

1) The current directory is the directory in which the data will be copied to. Therefore, make sure you are in the right directory.
2) If many files must be copied with multiple wget commands:
   a. Use a for loop to perform the task
   b. If a for loop is not used, make sure the current directory matches with the structure in which the data being copied is in.

Example of copying data from the server ftp://ftp.remss.com:

```
wget -c –r --user=username –password=password ftp://ftp.remss.com/ccmp/v02.0/Y2015/
```

where -c = “continue” getting a partially downloaded file
  -r = turn on “recursive” retrieving
  --user=username --password=password = log in username and password onto the server
  ftp://ftp.remss.com/ccmp/v02.0/Y2015/ = path to data from home directory in server

<u>Other Processes in the terminal:</u>

Unzipping multiple files:

```
I.   Method 1: cd /home/lcolosi/data/   %place yourself in the correct directory
         bzip2 -dk *.bz2 %unzip all documents ending in .zip
            ls -l   %check to make sure all the files have been unzipped by listing them
II.  Method 2: for z in 01 02 03 04 05 06 07 08 09 10 11 12
            > do
            > cd $z
             > bzip2 -dk  *.bz2
            > cd ..
            > done
```

Wget data onto server:

Method 1: for loop for copying files using wget from the Ifremer server to the fig server

```
#set variables
ftp=ftp://ftp.ifremer.fr/ifremer/ww3/HINDCAST/GLOBAL

#create a loop that will go through each file in Ifremer to obtain hs for 2004-2011 for the
#European ww3 model run.
for y in `printf "%02d " {12..16}` # adds zeros until 2-digit number
do
    d=20${y}_ECMWF
   `wget -c -r -np -R "index.html*" -nH --cut-dirs=7  ${ftp}/${d}/hs`
done
```

```
#create a loop that will go through each file in Ifremer to obtain wnd for 2004-2011 for the
#European ww3 model run.
for y in `printf "%02d " {1..11}`

do
    d=20${y}_ECMWF
    ` wget -c -r -np -R "index.html*" -nH --cut-dirs=7  ${ftp}/${d}/wnd`
done
```

Renaming directories:
Use the mv function in order to rename a directory:

```
mv /home/user/oldname /home/user/newname
```

Make sure that you are in the current parent directory of the directory you want to change the name of.

Viewing files on bash with ncdump and ncview:
On the server, they most likely have the ncdump and ncview commands installed and ready for use. These functions allow one to look at the data within a netCDF file without having to call the data within jupyter notebooks or on matlab. Here is the basic syntax and description of these functions:

1) ncdump:

a. Syntax: >> ncdump –h file_name.nc # displays only the name, dimensions and attributes of each variable
>> ncdump –c file_name.nc #displays the name, dimensions and attributes of each variable along with printing the time, lon, and lat variables
>> ncdump file_name.nc #displays the name, dimensions and attributes of each variable along with printing all variables
Note that the file name and path can be in the second input variable

2) ncview:

a. Syntax: >> ncview file_name.nc # displays a graphical interface of controls and a movie of the data gridded on a

In order to have these functions on your own computer, here is a link to install these functions onto a computer.

Shell scripts:
In order to make a shell script (shell scripts are analogous to an mfile in matlab):

1) Create a text editor file using the command below and make sure to add the .sh extension:
Vim name_of_editor.sh
2) Write the your code in the vim file
3) Quit the vim file and make the script executable by using the command:
chmod +x name_of_editor.sh
4) Now, we can run the shell script using the command:
./ name_of_editor.sh

Important note:
By detaching from a screen running on a server by using the command ctrl + A + D, the code running in the screen will continue running even if I close my computer, move away from internet, etc.
The power of wind cards:
Here is an example of using wind cards to move several data files from many subdirectories:
mv -i /zdata/downloads/Ifremer/altimeter_data/wind/data/*/*/*.nc /zdata/downloads/Ifremer/altimeter_data/

Reference code for Bash:
Press tab to auto complete
man command = documentation on specified command
rm folder = remove folder
rm –r folder = removes directory
mv past_location new_location = move file to another location in the server
mv –i = ask for confirmation before moving a file because the mv "move" command overwrites any file with the same name
mkdir name = make a new directory with specified name
cd = bring one back to the home directory
cd .. = move one step up in directory
cd ~ = bring one back to the home directory
ctrl + a + d = detach from a screen
screen –list = show active screens
screen –r name = resume a detached screen
screen –x name = resume an attached screen
"Documents/Research Lab" = quotes enable one to write a directory path without \ for spaces
screen -ls | grep Detached | cut -d. -f1 | awk '{print $1}' | xargs kill = kills all screens that are running
screen –X quit = kills all screens attached or detached
.. = go to parent directory (i.e. one above the current directory)
~ = go to the home directory
. = referring to the current directory
fc -l command = "finds command" and list the past times command was used
ls -l = lists files and directories with permissions granted within current directory
ls –la = lists hidden and visible files and directories with permissions granted within current directory
Reference code for vim commands:
For all commands for vim, one must be in the command mode by pressing *esc:*

:set paste = paste mode in order for any pasted text retains its form when pasted into vim
:0 = move to top of vim file
:d1000 = delete 1000 line of code in the vim text editor

To change to insert mode to adjust the text in the vim file, press *i.*
To save and quit the vim file, go into command mode and press the following:

:wq = quit and save
:x = quit (without saving)