

EEC-289 Student Software Manual

Welcome to EEC-289

Follow these instructions to prepare your machine for EEC-289 for this Fall.

Note that installing the eec289 software will require you to download around 10 GB data

If you have a slow network connection, please be patient with downloading.

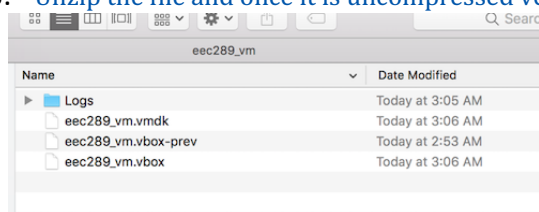
Please contact the course staff ASAP if these requirements present challenges so we can discuss alternative options.

1. VirtualBox

- 1.1. Go to <https://www.virtualbox.org/wiki/Downloads> ,download it and install it.
 - 1.1.1. **Windows** : <https://download.virtualbox.org/virtualbox/6.1.26/VirtualBox-6.1.26-145957-Win.exe>
 - 1.1.2. **OS X**: <https://download.virtualbox.org/virtualbox/6.1.26/VirtualBox-6.1.26-145957-OSX.dmg>
 - 1.1.3. **Ubuntu 20.04**: <https://download.virtualbox.org/virtualbox/6.1.26/virtualbox-6.1.26-145957~Ubuntu~eoan amd64.deb>

1.2. Download the VM

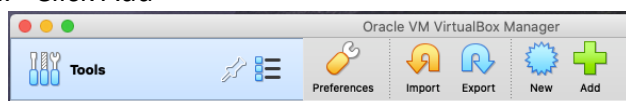
- 1.2.1. Download the VirtualBox Image for ECE 289 from [here](#)
[<https://docs.google.com/open?id=1MnPNXmMxpO3pVZoGeYHsnPDhxDfDghv>]
- 1.2.2. Please confirm the filename of the file you downloaded is `eec289_vm.zip`
- 1.2.3. Unzip the file and once it is uncompressed verify the files are as shown below:



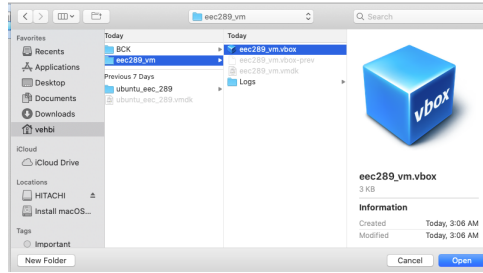
1.3. Create a new VM and attach the downloaded VirtualBox Image.

Image based instructions:

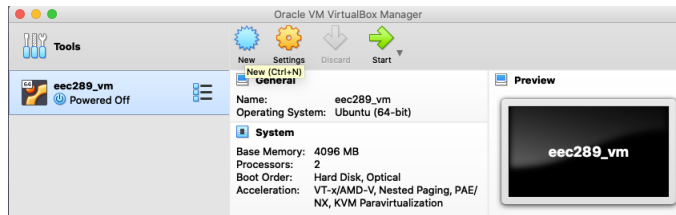
1.3.1. Click Add



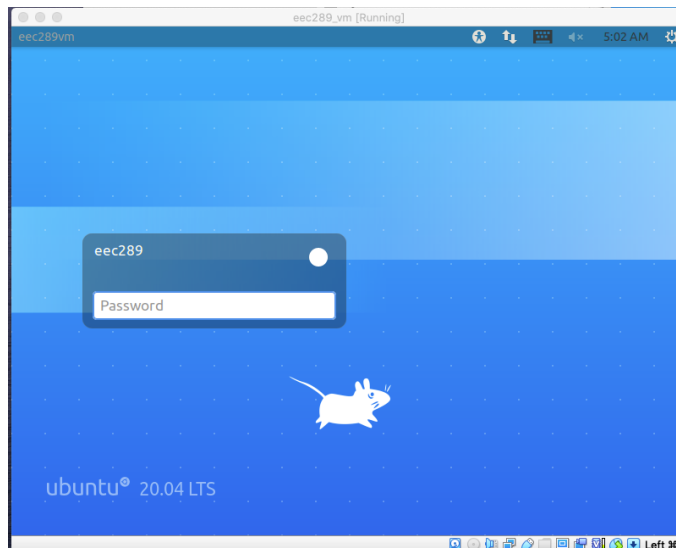
1.3.2. Select eec289_vm.vbox in the unzipped folder and click open



1.3.3. Click start to run the VM.

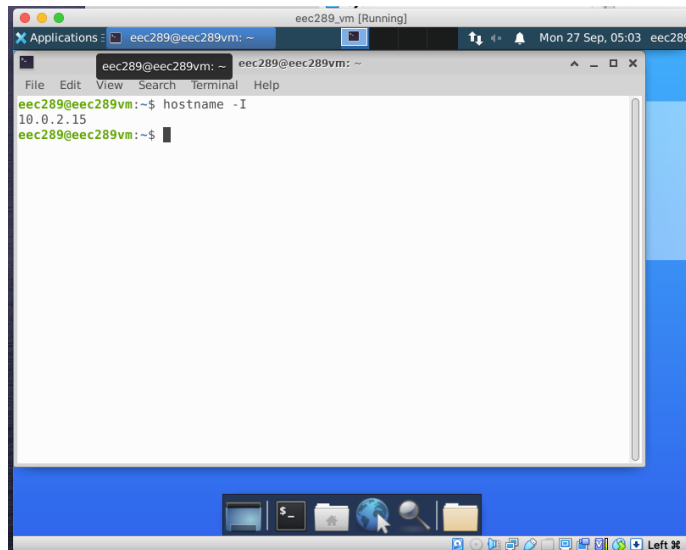


1.3.4. Once you are on the login screen, use the password “eec289”

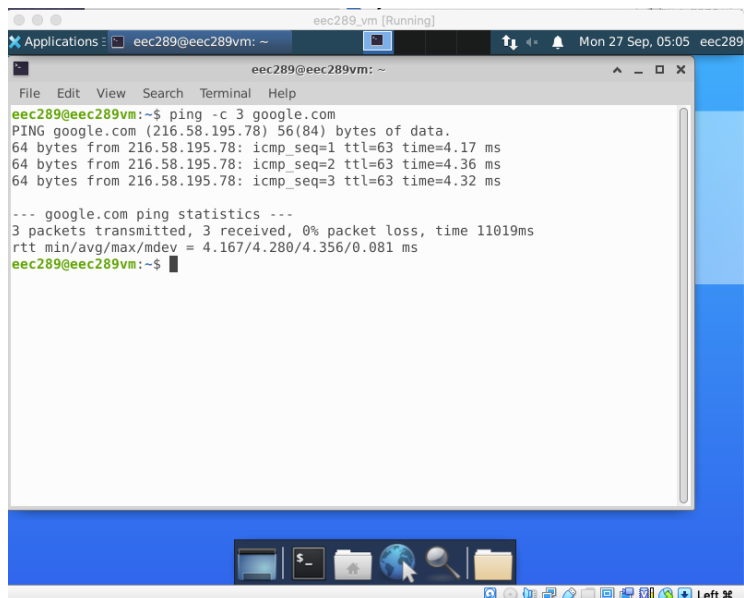


1.4. Configure your VM

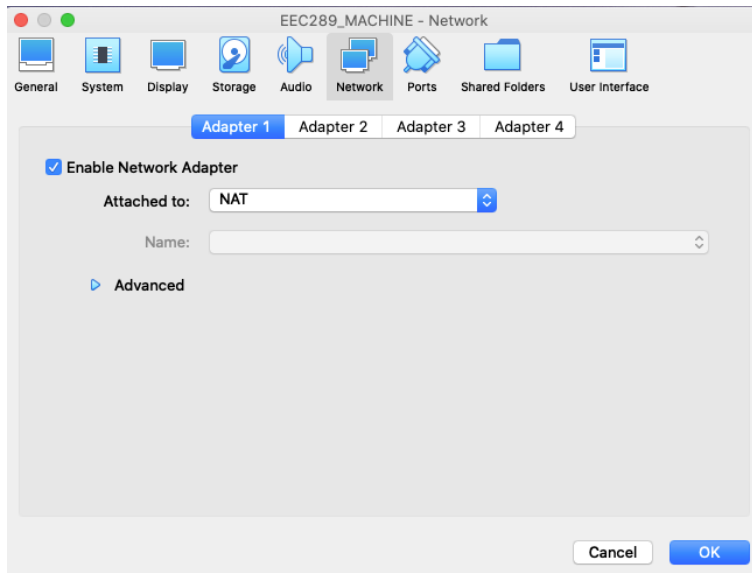
1.4.1. You will need to run `'hostname -I'` inside your eec289 VM to discover its IP address. You should be able to see an IP address just as shown below.



1.4.2. Ensure your VM is connected to the network by pinging google "ping -c 3 google.com" and make sure that it doesn't fail.



1.4.3. If either of these commands does not look right, you can turn of the VM and go to settings > networks > Adapter 1, make sure you have selected NAT



1.4.4. If you have resolve issues in your VM like below:

```
eec289@eec289vm:~$ ping -c 3 google.com
ping: google.com: Temporary failure in name resolution
```

You can turn off your VM and you open a terminal or a powershell or a command line on your host machine and run the following command: (change EEC289_MACHINE to whatever you named your VM)

```
VBoxManage modifyvm "EEC289_MACHINE" --natdnshostresolver1 on
```

2. Install required software

2.1. Run an update

```
eec289@eec289vm:~$ sudo apt-get update
```

2.2. Install git

```
eec289@eec289vm:~$ sudo apt-get install git
```

2.3. Clone the following repo by “git clone <https://github.com/eec289q-f21/stdsw.git>”

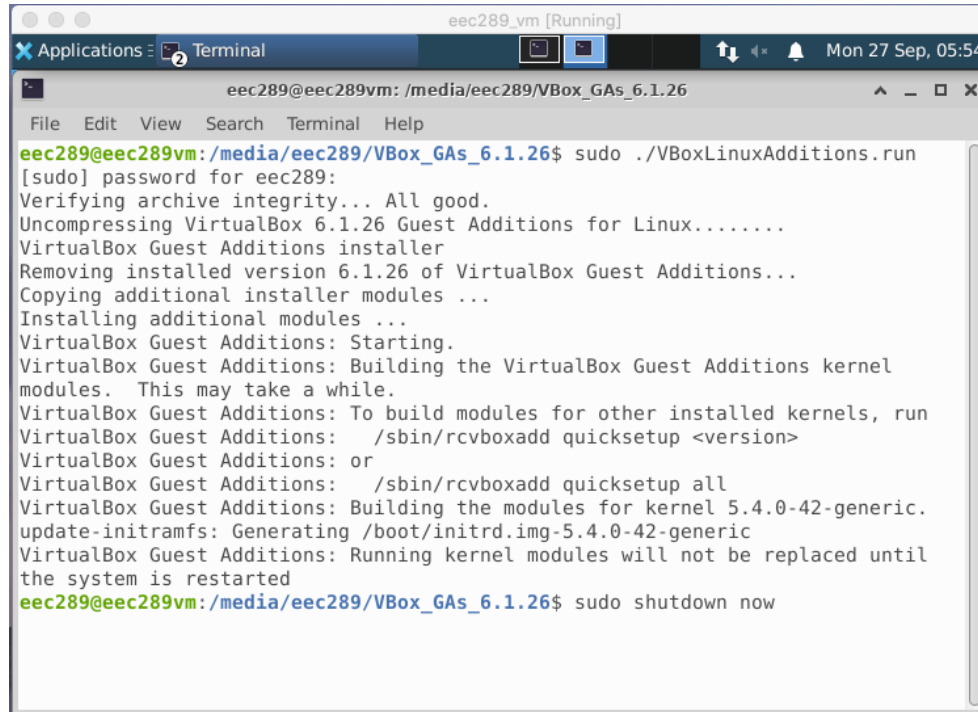
```
eec289@eec289vm:~$ git clone https://github.com/eec289q-f21/stdsw.git
Cloning into 'stdsw'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 1), reused 7 (delta 1), pack-reused 0
Unpacking objects: 100% (11/11), 3.65 KiB | 1.83 MiB/s, done.
```

2.4. Go to stdsw/vm and run “vmsetup.sh”, this may take a while! And can ask for password 2 times.

```
eec289@eec289vm:~$ cd stdsw/vm/
eec289@eec289vm:~/stdsw/vm$ ./vmsetup.sh
```

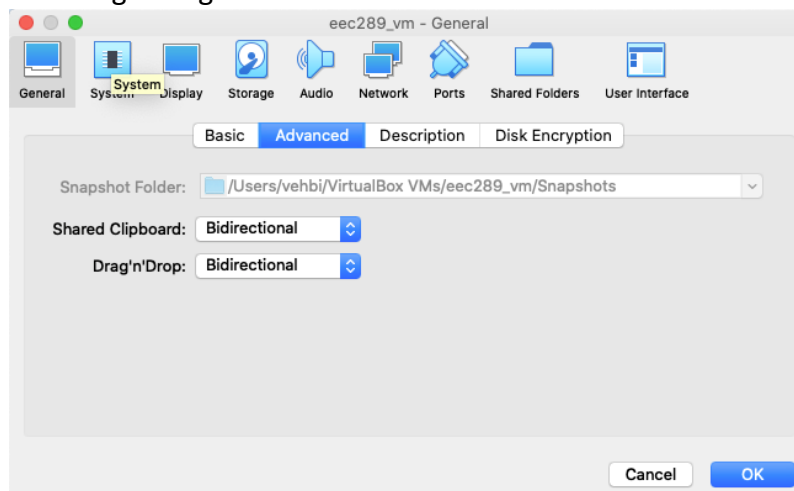
3. Some more configuration.

- 3.1.1. Guest Additions installation is needed to fix the screen resolution issues. please click on “Insert Guest Additions CD Image” from the devices menu and install it by opening a terminal within VM and switching to the directory “/media/eec289/VBOX_GAs_6.1.26” and running “VBOXLinuxAdditions.run” file. This may also take a while!. Once it is done, please shut down your VM.

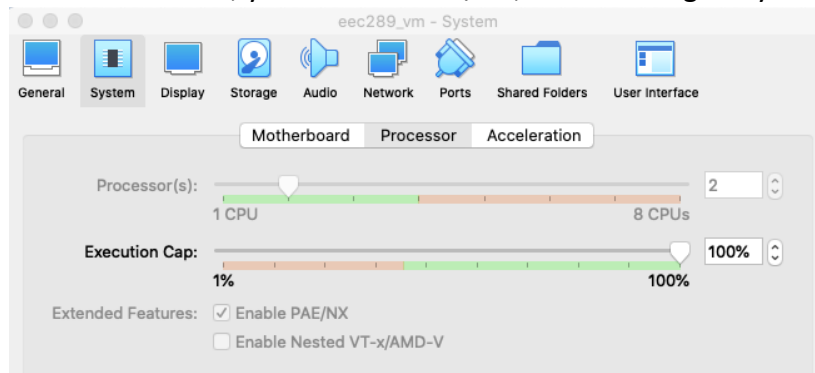


```
eec289_vm [Running]
Applications Terminal
eec289@eec289vm: /media/eec289/VBox_GAs_6.1.26
File Edit View Search Terminal Help
eec289@eec289vm:/media/eec289/VBox_GAs_6.1.26$ sudo ./VBoxLinuxAdditions.run
[sudo] password for eec289:
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.26 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 6.1.26 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.4.0-42-generic.
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
eec289@eec289vm:/media/eec289/VBox_GAs_6.1.26$ sudo shutdown now
```

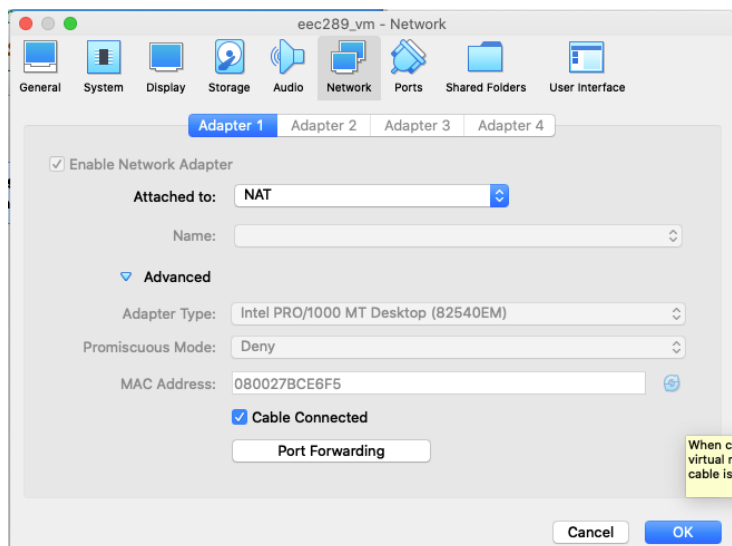
- 3.1.2. Once your VM is turned off, Go to settings > General > Advanced. Make the following configurations



3.1.3. Make sure also, you enabled PAE/NX, Go to settings > System > Processor.



3.1.4. Port-forwarding for SSH must be configured. Go to Settings > Network > Port Forwarding



Add the following rule:

	Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
ssh		TCP		3022		22

4. Configure Your Development Environment (Optional)

4.1.1. Install VS Code on your computer (NOT inside the VM),

4.1.1.1. download and install Visual Studio Code from

<https://code.visualstudio.com/download>

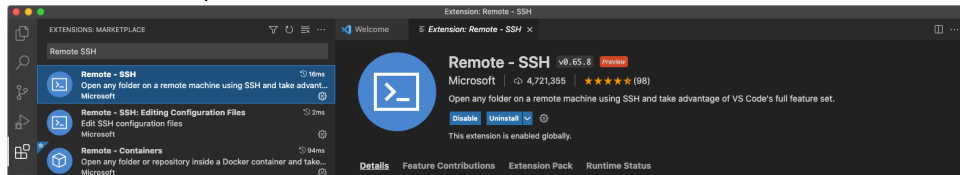
4.1.2. Install the Remote-SSH extension by

4.1.2.1. Open VS Code

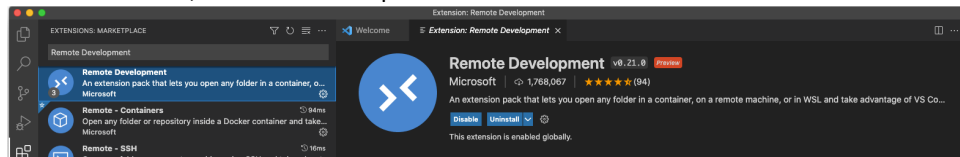
4.1.2.2. Press Ctrl-Shift-P (on Windows/Linux) or Cmd-Shift-P (on Mac)

4.1.2.3. Type `Extensions: Install Extensions` and select that option

4.1.2.4. In the search box, paste: `ms-vscode-remote-ssh` or `Remote – SSH`



4.1.2.5. In the search box, “Remote Development”



4.1.2.6. Install these extensions

4.1.3. Restart VS Code

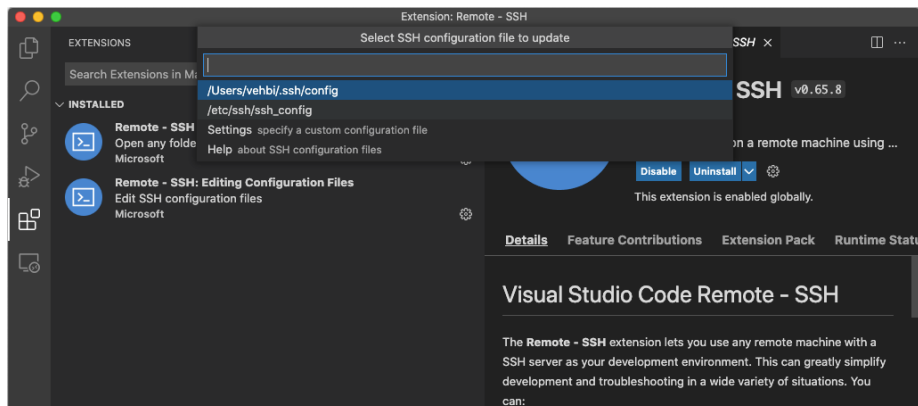
4.1.4. Configure SSH Config

4.1.4.1. Open VS Code

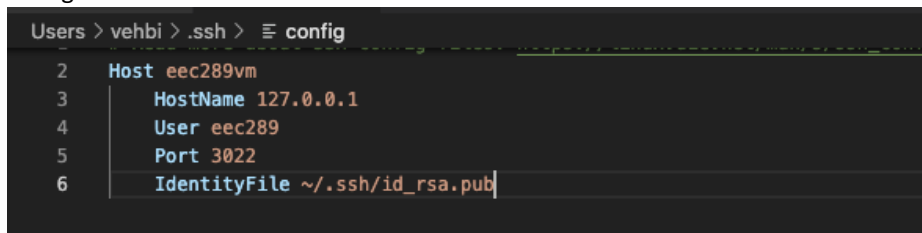
4.1.4.2. Press Ctrl-Shift-P (on Windows/Linux) or Cmd-Shift-P (on Mac)

4.1.4.3. Type `Remote-SSH: Open Configuration File` and select that option

4.1.4.4. Choose the first file

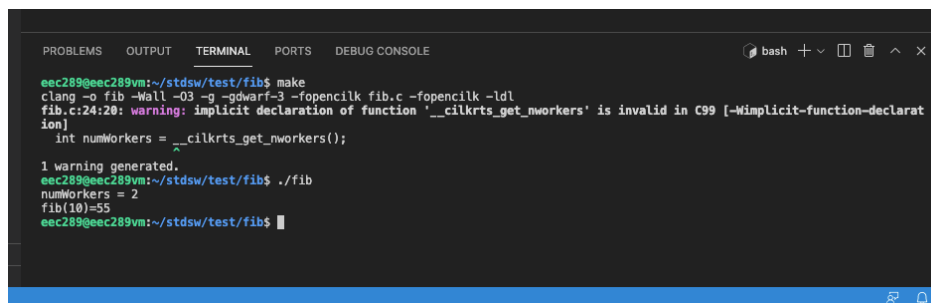
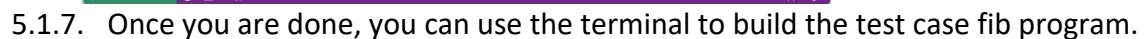
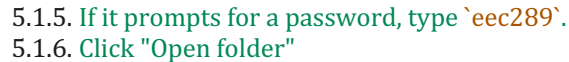


4.1.4.5. Configure as follows



For windows users can generate them opening a command line and running 'ssh-keygen -t rsa -b 4096 -f %USERPROFILE%/id_rsa.pub' and give the C:/Users/<your user>/id_rsa.pub

- 5.1.1. Open VS Code
- 5.1.2. Press Ctrl-Shift-P (on Windows/Linux) or Cmd-Shift-P on Mac
- 5.1.3. Type `Remote-SSH: Connect to Host` and select that option
- 5.1.4. Select eec289vm



6. RUNNING ON AWS!

So far, what we have done is to establish a development environment for the assignments to come.

You will do your own development, compiling and debugging on the VM that you have setup. Once you are done with your development and feel like it is time to get some performance numbers, that means you are ready to test your mettle.

Here are the steps to run your code on AWS:

1. Introduce your AWS credentials from the email you received. Open a terminal and enter “aws configure” and fill out the information as follows:



```
eec289@eec289vm: ~/eec-289
File Edit View Search Terminal Help
eec289@eec289vm:~/eec-289$ aws configure
AWS Access Key ID [None]: AKIAUIJQK4QBJJTPZUNL
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-1
Default output format [None]: json
```

Don't get any ideas, this access codes are invalidated 😊

If you get the following error:

when I run aws configure in the vm, I get this:

Traceback (most recent call last):

File "/usr/bin/aws", line 19, in <module>

import awscli.clidriver

File "/usr/lib/python3/dist-packages/awscli/clidriver.py", line 36, in <module>

from awscli.help import ProviderHelpCommand

File "/usr/lib/python3/dist-packages/awscli/help.py", line 23, in <module>

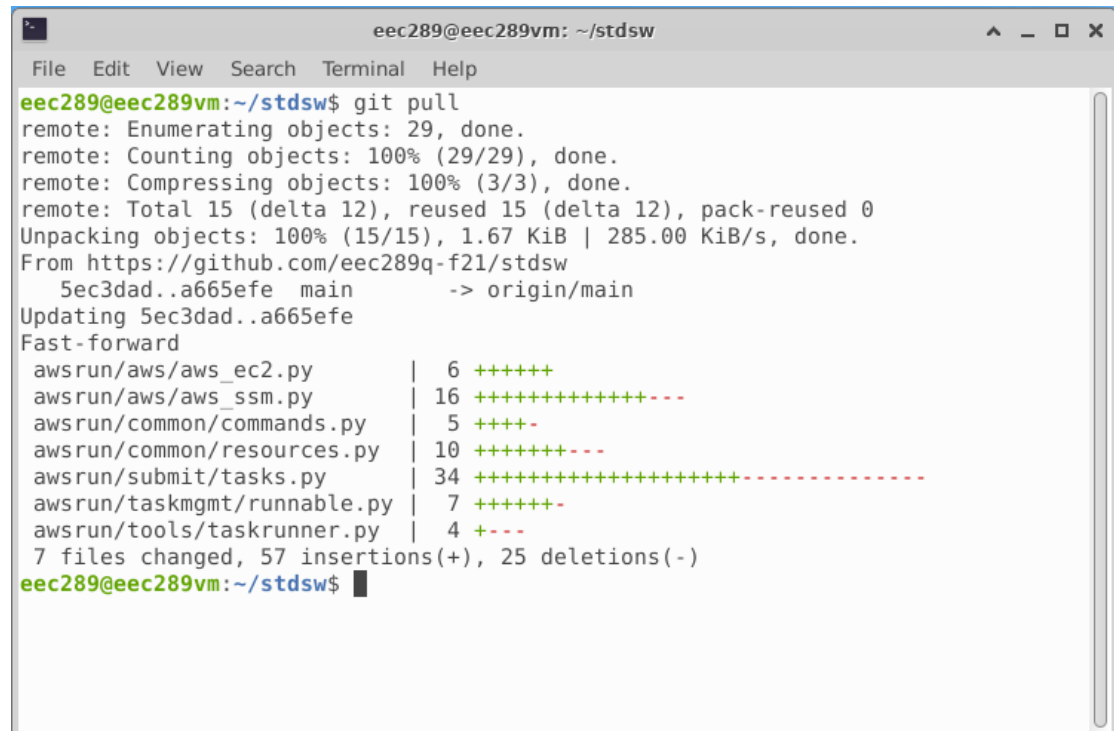
from botocore.docs.bcdoc import docevents

ImportError: cannot import name 'docevents' from 'botocore.docs.bcdoc' (/usr/local/lib/python3.8/dist-packages/botocore/docs/bcdoc/__init__.py)

Please update your aws by

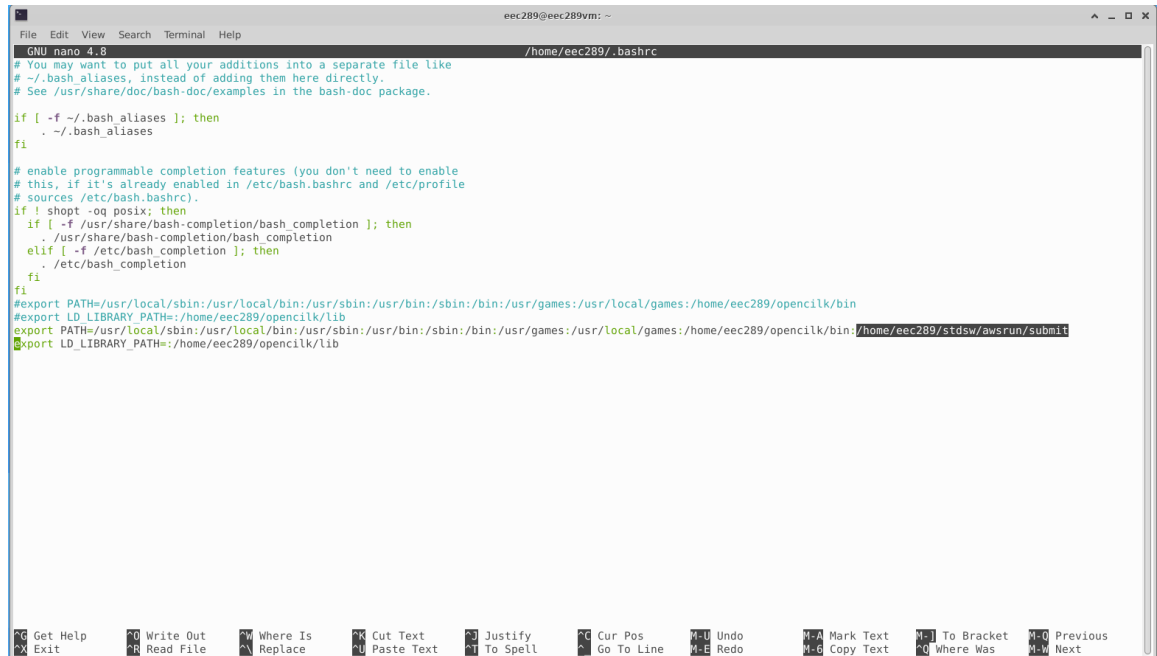
```
pip3 install --upgrade pip
pip3 install --upgrade awscli
```

2. Make sure that your stdsw repo is up to date. Go under the folder stdsw and run “git pull”. You better do this step with every new released assignment or when we say you do it in the slack channel.



```
eec289@eec289vm: ~/stdsw
File Edit View Search Terminal Help
eec289@eec289vm:~/stdsw$ git pull
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 15 (delta 12), reused 15 (delta 12), pack-reused 0
Unpacking objects: 100% (15/15), 1.67 KiB | 285.00 KiB/s, done.
From https://github.com/eec289q-f21/stdsw
   5ec3dad..a665efe  main       -> origin/main
Updating 5ec3dad..a665efe
Fast-forward
 awsrun/aws/aws_ec2.py      | 6 ++++++
 awsrun/aws/aws_ssm.py     | 16 ++++++++-----
 awsrun/common/commands.py | 5 ++++-
 awsrun/common/resources.py | 10 ++++++---
 awsrun/submit/tasks.py    | 34 ++++++++-----
 awsrun/taskmgmt/runnable.py | 7 ++++++-
 awsrun/tools/taskrunner.py | 4 +---
 7 files changed, 57 insertions(+), 25 deletions(-)
eec289@eec289vm:~/stdsw$
```

3. Make sure that you have included the stdsw tools in your PATH. This is a one-time step unlike the second step. Feel free to use your favorite editor for this step. Open a terminal and type “nano ~/.bashrc” and go to line where you export PATH variable and add “/home/eec289/stdsw/awsrun/submit” and you are done. Save and exit!!!



```
GNU nano 4.8 /home/eec289/.bashrc
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

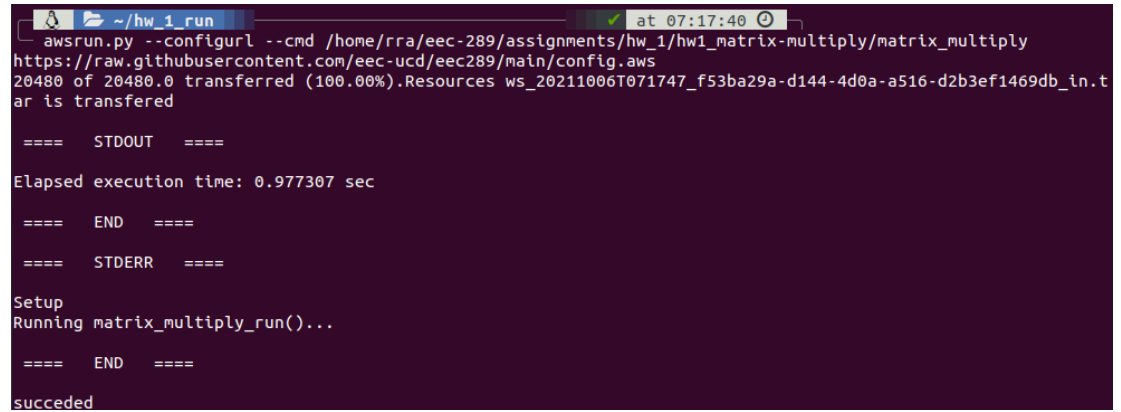
#export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/home/eec289/opencilk/bin
#export LD_LIBRARY_PATH=/home/eec289/opencilk/lib
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/home/eec289/opencilk/bin:/home/eec289/stdsw/awsrun/submit
export LD_LIBRARY_PATH=/home/eec289/opencilk/lib
```

Then you can source the bashrc by “`./~/.bashrc`”



```
eec289@eec289vm: ~/eec-289
File Edit View Search Terminal Help
eec289@eec289vm:~/eec-289$ ./~/.bashrc
eec289@eec289vm:~/eec-289$
```

4. Run the following command to run your executable on AWS. Here, I will run the `matrix_multiply` example:



```
awsrun.py --configurl --cmd /home/rra/eec-289/assignments/hw_1/hw1_matrix-multiply/matrix_multiply
https://raw.githubusercontent.com/eec-ucd/eec289/main/config.aws
20480 of 20480.0 transferred (100.00%).Resources ws_20211006T071747_f53ba29a-d144-4d0a-a516-d2b3ef1469db_in.t
ar is transferred

==== STDOUT ====

Elapsed execution time: 0.977307 sec

==== END ====

==== STDERR ====

Setup
Running matrix_multiply_run()...

==== END ====

succeeded
```

You also have the stdout and stderr from the run under the same folder where you ran `awsrun`:



```
ls
stderr stdout
```