Project Setup

Harvard CS109b Spring 2017

This document is editable by anyone with the link. Like a wiki, please enhance it as you find areas to improve.

Introduction

Keras and TensorFlow have numerous dependencies. Unlike CS109a, there may be issues using these libraries with Anaconda. In order to alleviate issues, we have created two environments for you to use in your project:

For your local machine, there is a Linux-based VirtualBox image.

For your AWS instance, there is a Linux-based AMI.

Since both environments run Linux, there will be some consistency between them, at least more than you get otherwise for all possible student configuration permutations.

You are not required to use these images, but it should save significant time over installing all the dependent Python and LaTeX packages yourself. It will also ensure a certain degree of consistency between teammates on the project. See Appendix A for a list of packages that may be necessary should you elect to go independently.

Note that we will support both Python 2.7 and 3.5 users.

Part 1 - VirtualBox

VirtualBox is a free virtual machine environment for Windows, MacOS and Linux.mm

1.1 Install VirtualBox

Go to https://www.virtualbox.org/

Download and install the appropriate VirtualBox host for your local machine. (**Note:** Installation may fail if you have a previous version of VirtualBox on your machine. Make sure you remove it completely before installing the latest version. You may need to reboot.)

1.2 Download and Import the CS109b Virtual Box Image

Download the image file here for all platforms: https://drive.google.com/open?id=0BzQIe6y82PgQTEtBQW5ERzRDcjg This is an OVA file. In VirtualBox, choose File, Import and then select the downloaded `cs109b.ova` file. This file is 8 GB.

[**If you downloaded prior to Apr 1, 2pm, read this section. Otherwise you may safely ignore it**

For Mac, you can save 1 Gb of download by trying to download https://drive.google.com/open?id=0B8_XcYtJw05aVE1FOFVfWFR6bFU (Python V2 default, Date: 3/31, 12pm) and unzipping it. The download is approximately 7 GB. Once unzipped, it is approximately 16 GB. Unzip the file into a fresh local directory. To save local disk space, you can then remove the downloaded zip file.

If you downloaded the VirtualBox image prior to 3/31, 12pm, there are two small issues:

1. The tensorflow was installed on Python version 3.5. (not 2.7).
2. The IMDbPY library is not present.

Rather than downloading another 7 Gb image, both of these issues can be addressed by running the following commands:

```
sudo apt-get install python-pip python-dev
pip install tensorflow
pip install IMDbPY
pip install keras
pip install jupyter

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

python3 -m pip install ipykernel
python3 -m ipykernel install --user
pip install requests wget
pip install matplotlib seaborn scikit-learn
```

If you downloaded the zip file, it will likely be sufficient to navigate to the directory containing the unzipped contents and double-click on the `cs109b.vbox` file. If VirtualBox does not recognize the .vbox file, some hacky machinations are required. (Hopefully, we will find a more elegant procedure)

1. Select “New” and create a new empty Virtual Machine. Call it something simple like “x”. It won’t be needed for long. Select Linux as the OS, do not add any storage and ignore the warning.
2. Go back to the main VirtualBox console. Select VM “x” and then select Settings. Choose Storage. Click on the small +icon to add storage. Select an existing volume. Navigate to the .VDI file in the folder where you expanded the zip file. Click OK to dismiss the settings dialog.

3. Now that VirtualBox “knows” about the VDI, you can double-click on the `cs109b.vbox` file. It should recognize the VM. Now you can hit the green arrow to start the VM.

] End of section for early downloads.

1.3 Start up the image and login

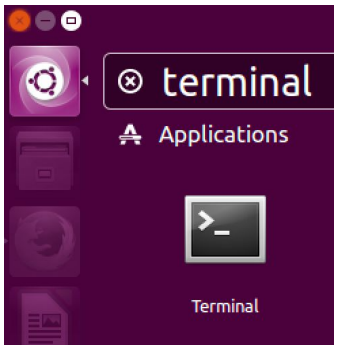
Launch virtualbox on your local machine. Once you have successfully imported the OVA file, just hit the green Start arrow.

The username is cs109b. The password is also cs109b. This user has root privileges. When the VM starts you will be automatically logged in to Linux.

NOTE for PC (Windows and Linux) users: You may get an error that VT-X is not enabled in your computer’s BIOS setting. This means you have to shutdown your machine, enter the BIOS configuration upon boot and turn on Virtualization. For more information: https://www.howtogeek.com/213795/how-to-enable-intel-vt-x-in-your-computers-bios-or-uefi-firmware/

1.4 Test that it is working

Click on the top left icon and type in `terminal` in the search box. It should look like this:



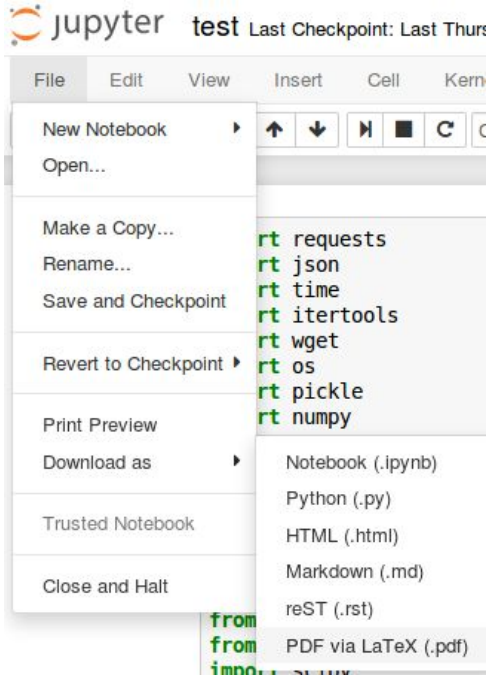
Click on the terminal icon to launch a shell / command / terminal window.

Once the terminal is launched, enter

```
cd Documents
ls
jupyter notebook test.ipynb
```

`ls` lists the contents of the directory. The notebook should come up correctly. The list of libraries give you an idea of some of possible libraries you may use over the duration of the project. Ensure that everything is working correctly by executing each of the cells (via Shift-Enter or clicking on the Play > toolbar icon. The Keras and TensorFlow version numbers should appear.

Next, generate a PDF. From the Jupyter menu bar, select File, Download As, PDF. It should look like this:



The resultant `test.pdf` should look correct.

1.5 Create a share

Create a share to move files easily between the Virtual Machine and your local host or to use your favorite local text editor while executing code on the virtual machine.

Follow the instructions at
http://helpdeskgeek.com/virtualization/virtualbox-share-folder-host-guest/

Make the share permanent so it appears on each reboot of the virtual machine.

(Select the Linux command window and then you should see a Devices tab in the top menu)

If you run into some problems mounting the Guest additions CD, it is always possible to install the Additions manually. Open a terminal window in the Linux Virtual Machine and enter:

```
sudo apt install virtualbox-guest-utils
```

We do not recommend sharing a local clone of your git repository. Due to the way git keeps track of the repository state and potential differences in git versions, keep two separate clones of your git repository: on your local machine and on the virtual machine.

1.6 Tune your environment

Depending on the availability of RAM and processors on your machine, you may decide to allocate more (or less) resources to your virtual machine.

Shutdown down the Linux machine using the Gear icon on the top right.

Once it has shut successfully, go the VirtualBox Manager, select the cs109b VM, choose settings, then System. From there, you can assign more memory or processors to the virtual machine.

If the host machine has limited resources you may consider switching to a lightweight desktop environment. In the virtual machine terminal enter

```
sudo apt-get updates
sudo apt-get install lubuntu-desktop
```

and restart the virtual machine. If it automatically logs you into the same desktop as before, log out and switch to the Lubuntu desktop, as shown below.

jupy

1.7 SSH into the virtual machine

If you'd prefer not to work inside the virtual environment, you can SSH into the virtual machine using a terminal and work from your default environment (source):

1. Power off the virtual machine if it is already running
2. Open the network settings of your virtual machine and ensure that you are using a NAT adapter.
3. Click on advanced, then open the port forwarding window.
4. Create a rule with name = SSH, protocol = TCP, host port = 3022, and guest port = 22
5. Install the SSH server on the VM with `sudo apt install openssh-server`
6. SSH into the VM with `ssh -p 3022 cs109b@127.0.0.1`

If you're doing this, odds are you won't want to keep a VirtualBox window minimized all the time. You can launch the VM in a headless mode or go to Machine -> Detach GUI to leave the VM running in the background without VirtualBox open. Assuming that the VirtualBox binaries have been added to the path, you should be able to launch the VM in headless mode from your terminal with `VBoxManage startvm <vm name> --type headless`; use `VBoxManage controlvm <vm name> savestate` or `VBoxManage controlvm <vm name> acpishutdown` to stop it.

1.8 Hosting Jupyter notebooks on the virtual machine

If you're hosting a jupyter notebook on the VM, you can use port forwarding to access it from your local environment (source):

1. Launch your notebooks with `jupyter notebook --no-browser --port=8889 filename.ipynb`
    a. To avoid typing this long command every time, you can create an alias for it in your `~/.bashrc` file. For instance, if you'd like to launch notebooks with just `jnb filename.ipynb`, you can add the following line to the top of your `~/.bashrc` file: `alias jnb='jupyter notebook --no-browser --port=8889'`
2. Run `ssh -p 3022 -N -f -L localhost:8888:localhost:8889 cs109b@127.0.0.1` on your local machine
3. Open localhost:8888 on your local machine

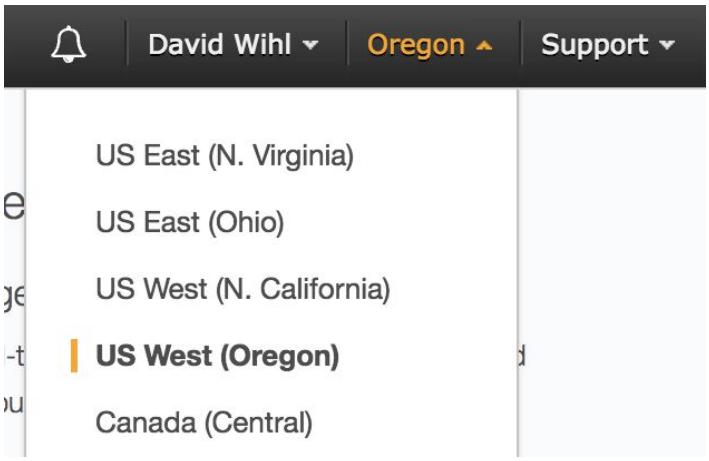
Part 2 - AWS

2.1 Login to AWS

https://aws.amazon.com

2.2 Switch to US West

From the region button on top right, select US West(Oregon)



AWS may open instances in US East. For now, they recommend US West. In any case, there should not be large data transfers needed on a regular basis.

2.3 Create instance

- Search for “EC2” in the search bar. Click on “EC2” to start the EC2 wizard.
- Click on the big blue “Launch instance” button
- Select “Community AMIs”.
- Search for “cs109b” (Select AMI with ID - ami-ee8e198e)
- For now, select “t2.micro” (Free tier) (when doing Deep Learning, use “p2.xlarge”)
- Click “Next: Configure Instance Details”
- Choose 1 instance
- When using p2.xlarge, click on “Request Spot Instances”. Spot instances are significantly cheaper than normal instances. It is a way for Amazon to sell excess capacity at reduced prices.

- (Optional) Click on “Enable CloudWatch Detailed Monitoring”. This will enable additional services like automatically shutting down an idle instance. You will be warned that additional charges may be incurred, which will go against your allotment. Consider it like buying insurance.
- You will also need to set “Configure Security Group” option for Jupyter notebooks and Tensorboard. It should look like following:

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8888	0.0.0.0/0
Custom TCP Rule	TCP	6006	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

- When you attempt to launch the instance it will ask if you have a keypair.
    - If you have not used SSH before and do not have a keypair, select “Create a new keypair”
    - Download the .pem file and store it in a safe place
    - You will need to import the .pem file to connect your instance over ssh. See below.

- Launch the instance

2.4 Login to the instance

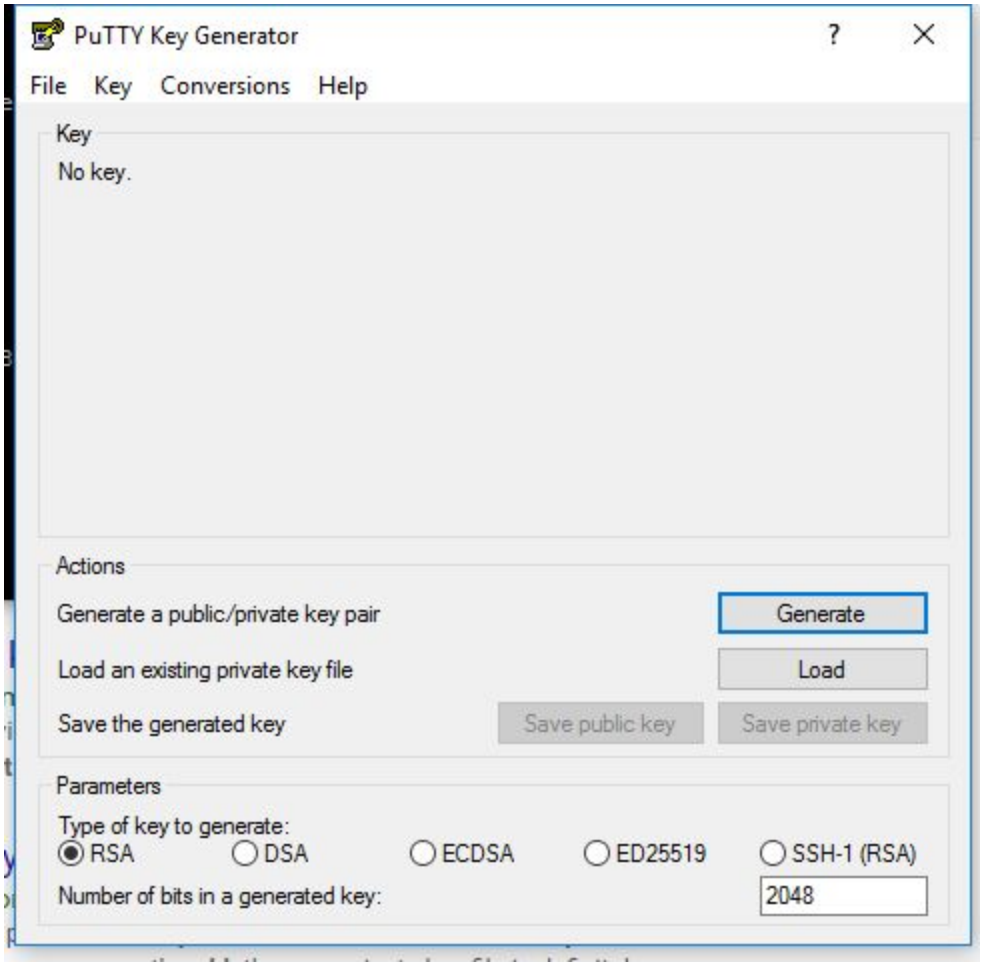
Connect to the instance over ssh which establishes a terminal session to your newly created instance.

For Windows Users

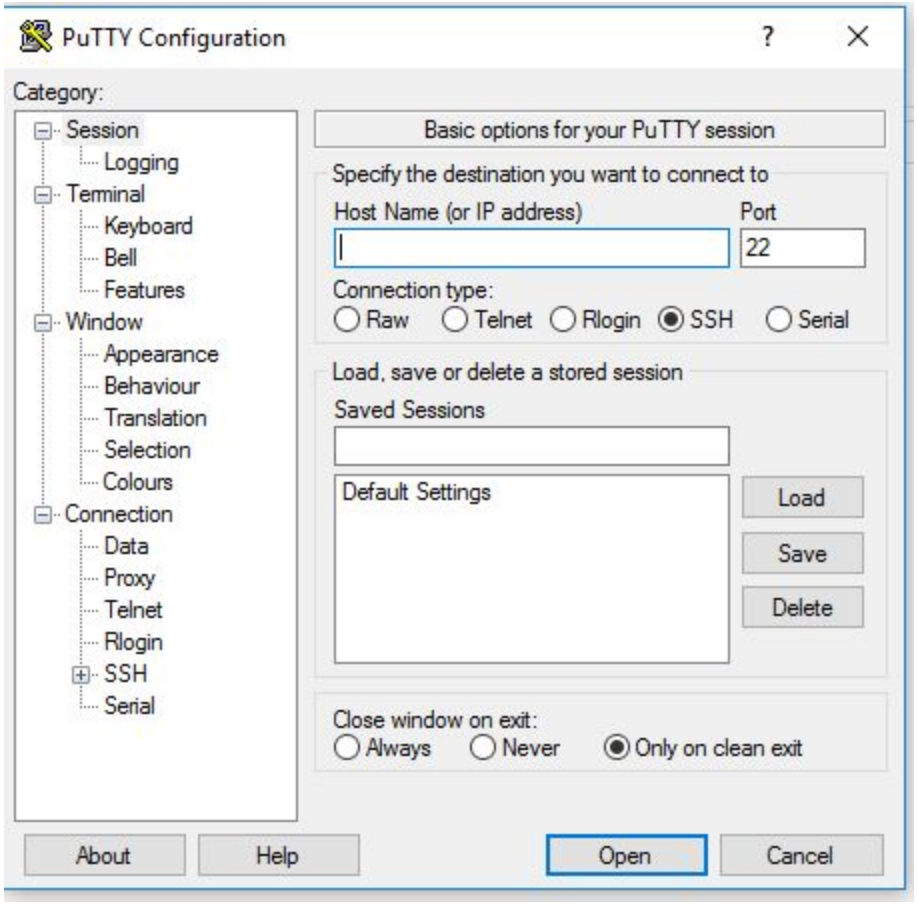
Windows users may not have an ssh client installed. If you need ssh for Windows, download PuTTY.

You will need to convert the `.pem` key from AWS into a `.ppk` key in PuTTY Key Generator

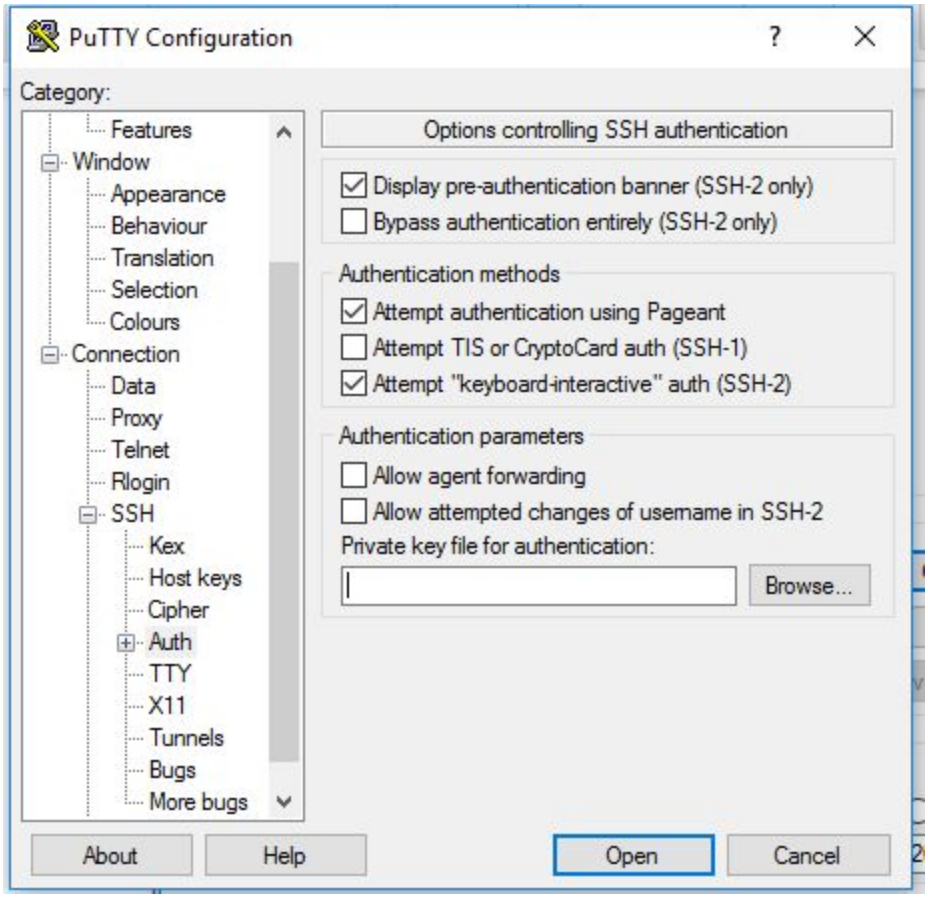
- Select Import Key under Conversions
- Save as Private Key with RSA selected


PuTTY Key Generator
File Key Conversions Help
Key
No key.
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
ED25519
SSH-1 (RSA)
Number of bits in a generated key:
2048

Launch PuTTY and you will see the below screen.



Navigate to SSH > Auth, browse to add the `.ppk` file and Open



Login as user `ubuntu`

For MacOS and Linux users

Open a terminal window.

Note: On a Mac or on Linux you may need to change the permissions of your ssh keyfile if you get the following error when attempting to ssh to your instance:

“Permission denied (publickey)”

Change the permission of your ssh keyfile as follows if you get the above error message.

```
chmod 600 <ssh_key>
```
e.g.
```
chmod 600 myfile.pem.txt
```

Use the public IP of the running instance and username “ubuntu”. For example:

```
ssh -i <your_ssh_keyfile> ubuntu@<your_AWS_public_DNS_name_or_IP_address>
```

Example:

```
ssh -i myfile.pem.txt ubuntu@54.12.34.56
```

2.5 Launch Python and check libraries

Try some Linux commands, e.g.

- `ls; cd; mkdir;` etc…

Launch python

- `python`

To start Jupyter, it is a two step process:

1. Launch the Jupyter server on AWS:

   `jupyter notebook`

   Note that since AWS is running “headless,” i.e. no windowing support, do not expect a browser window to pop open. You are simply starting a remote server.

2. Start a local browser on your machine, and then navigate to the Jupyter server running on AWS:

   `https://<your_AWS_public_DNS_name_or_IP_address>:8888`

   e.g.

   `https://53.12.34.56:8888`

Logout (you also press Control-D at the shell prompt)

NOTE: As of this writing, AWS has not yet enabled p2.xlarge instances for our class. When they do procedures may change. We will update this document appropriately.

2.6 Copy a file from AWS to your local machine

Using SCP is one way to do this:

Note: “\” means line continuation in Unix parlance

```
scp -i <your_ssh_keyfile> \
ubuntu@<your_AWS_public_DNS_name_or_IP_address:<path_to your_AWS_file>/<your_AWS_file>  ./
```

2.7 Copy a file from your local machine to AWS

Using SCP is one way to do this:

Note: "\" means line continuation in Unix parlance

```
scp -i <your_ssh_keyfile>  \
<path_to_your_local_file>/<your_local_file>ubuntu@<your_AWS_public_DNS_name_or_IP_address:~/
```

2.8 Shutdown the instance

Shutdown the instance from the AWS Console using either Actions/Instance State: "Stop" or "Terminate"

Make sure you always shut down your instance after you have completed your work!

2.9 Change Instance Settings

TBD, enabling CloudWatch for idle instances, warnings for overages...

Part 3 - Spark

3.1 Running Spark on the Virtual Machine

If you would like to play with Spark, it is already installed on the VirtualBox VM. From a terminal window, enter `pyspark` and then:

```
>>> a = 5
>>> b = 3
>>> a+b
8
>>> print("Welcome to Spark")
Welcome to Spark
## type Ctrl-d to exit
```

Thanks to https://www.santoshsrinivas.com/installing-apache-spark-on-ubuntu-16-04/ for Spark install instructions.

3.2 Running Spark in the Cloud

To learn more about Spark, see https://sparkhub.databricks.com/resources/

DataBricks provides a mini 6 GB cluster for free which allows public sharing. See https://databricks.com/try-databricks

Appendix A - Possible Python Libraries

If you want to run your own configuration of Python, etc. we will not be able to offer support. As mentioned in lecture, Windows users in particular will have to run TensorFlow on Python 3 which is mutually exclusive with some of the other libraries. Here is the list of some libraries we used when preparing a proof of concept solution (we will not be sharing this PoC solution with you even after the course).

```
requests
json
time
itertools
wget
os
pickle
numpy
random
matplotlib
seaborn
sklearn
scipy
tensorflow
pandas
keras
```

All of the above libraries are already installed on the VM and AMI.

Following libraries are not on AMI/VM:

```
tmdbsimple
```

Use following commands to install tmdbsimple:

```
   (1)   sudo apt-get install libssl-dev libffi-dev
   (2)   pip install tmdbsimple # Python2.7
            and/or
         pip3 install tmdbsimple #Python3.5
```

```
h5py
```

Use following command to install h5py

```
      pip install h5py   # Python2.7
      pip 3 install h5py # Python3.5
```