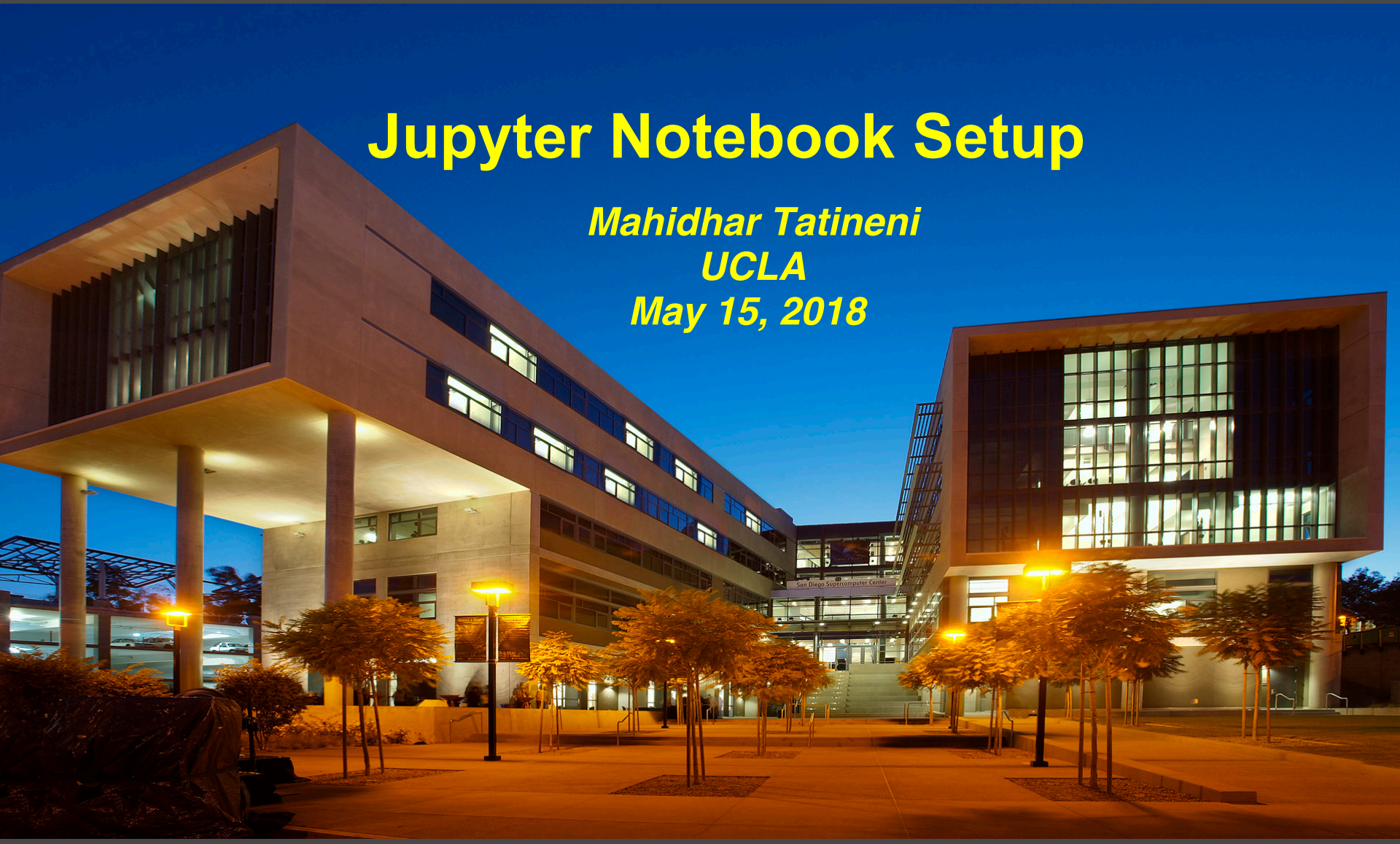
Jupyter Notebook Setup

Mahidhar Tatineni
UCLA
May 15, 2018



Installation

- **Several options to install Jupyter notebooks**
- **Download and install Anaconda - will have Jupyter Notebook included. This is the recommended approach**
- **Use Python's package manager "pip" to install :**
  - pip3 install --upgrade pip
  - pip3 install jupyter
- **Can you "pip" for python2 (only for jupyter versions < 6.0)**

Installing kernels

- **You can have python 2 and 3 kernels at the same time.**
- **For example if jupyter was installed with python3**
  - python2 -m pip install ipykernel
  - python2 -m ipykernel install --user
- **Can also use Conda**
  - conda create -n ipykernel_py2 python=2 ipykernel
  - activate ipykernel_py2
  - python -m ipykernel install --user
- **Last command creates kernel spec file**

Example image: sdsc_ubuntu_tf1.1_keras_R.img
Includes: ipython2, 3 and R kernels

- **ipython2 kernel file:**
  - /usr/local/share/jupyter/kernels/python2/kernel.json

```
{
 "display_name": "Python 2",
 "language": "python",
 "argv": [
  "python",
  "-m",
  "ipykernel_launcher",
  "-f",
  "{connection_file}"
 ]
}
```

Other kernels

- https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

- PPA for Ubuntu 15.10 (wily)/16.04 (xenial) has following kernels:
  - ihaskell
  - ijulia
  - ijavascript
  - irkernel
  - iruby
  - gophernotes

Installing irkernel

- https://irkernel.github.io/

- Step 1: Install from R console
  - install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))
    devtools::install_github('IRkernel/IRkernel')

- Step 2: Make it available to Jupyter
  - IRkernel::installspec(user = FALSE)

Example image: sdsc_ubuntu_tf1.1_keras_R.img
Includes: ipython2, 3 and R kernels

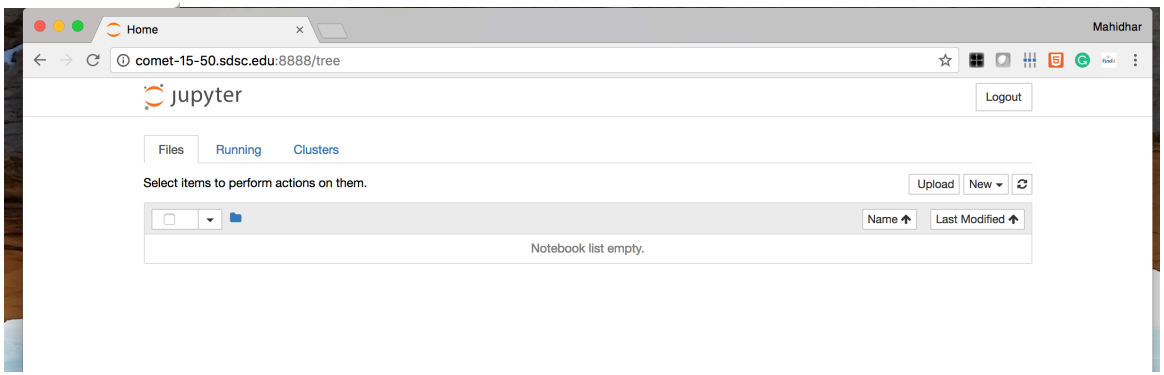
- **irkernel kernel file:**
  - /usr/local/share/jupyter/kernels/ir/kernel.json

```
{
  "argv": ["/usr/lib/R/bin/R", "--slave", "-e", "IRkernel::main()"
, "--args", "{connection_file}"],
  "display_name": "R",
  "language": "R"
}
```

Notebook security

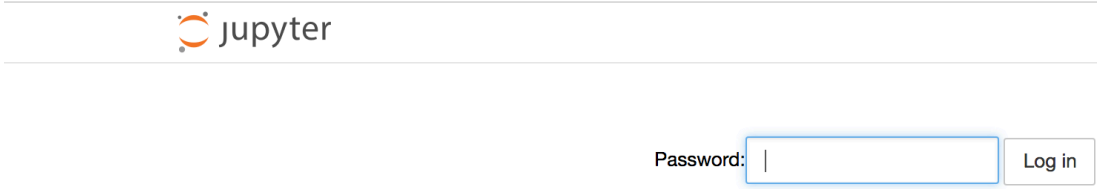
- By default you get a token (that's what we will do today).

```
mahidhar — etrain72@comet-ln2:~ — ssh etrain72@comet.sdsc.edu — 80×24
etrain72@comet-15-50:~$ jupyter notebook --no-browser --ip="*" &
[1] 12160
etrain72@comet-15-50:~$ [I 14:19:55.371 NotebookApp] Writing notebook server coo
kie secret to /home/etrain72/.local/share/jupyter/runtime/notebook_cookie_secret
[W 14:19:55.389 NotebookApp] WARNING: The notebook server is listening on all IP
 addresses and not using encryption. This is not recommended.
[I 14:19:55.398 NotebookApp] Serving notebooks from local directory: /home/etrai
n72
[I 14:19:55.398 NotebookApp] 0 active kernels
[I 14:19:55.398 NotebookApp] The Jupyter Notebook is running at: http://[all ip
addresses on your system]:8888/?token=c97f4e9c66eec3e00bdd4eb8a58157b261d289cea5
1621d7
[I 14:19:55.398 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 14:19:55.401 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=c97f4e9c66eec3e00bdd4eb8a58157b261d289cea51
621d7

etrain72@comet-15-50:~$ [I 14:20:26.798 NotebookApp] 302 GET /?token=c97f4e9c66e
ec3e00bdd4eb8a58157b261d289cea51621d7 (12.27.68.35) 0.53ms
```



Adding a password

- **Configure**
  - jupyter notebook --generate-config
- **Setup the password**
  - from notebook.auth import passwd
  - Passwd()
- **Add to json config file**
  - c.NotebookApp.password = ….



SSL for encrpytion

- **Self signed option:**
  - openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mykey.key -out mycert.pem
  - jupyter notebook --certfile=mycert.pem --keyfile mykey.key --no-browser --ip="*" &
- **Using Let's Encrypt**
  - Create a Let's Encrypt certificate
  - Update configuration

```
# Set options for certfile, ip, password, and toggle off
# browser auto-opening
c.NotebookApp.certfile = u'/absolute/path/to/your/certificate/fullchain.pem'
c.NotebookApp.keyfile = u'/absolute/path/to/your/certificate/privkey.pem'
# Set ip to '*' to bind on all interfaces (ips) for the public server
c.NotebookApp.ip = '*'
c.NotebookApp.password = u'sha1:bcd259ccf...<your hashed password here>'
c.NotebookApp.open_browser = False
```

Jupyter Notebook: Today's Tutorial

[1] Get an interactive node:

```
cd $HOME
srun --pty --nodes=1 --ntasks-per-node=24 -p compute --reservation=UCLA2018Res -t 02:00:00 --wait 0 /bin/bash
```

[2] Load the singularity module and get an interactive shell

```
module load singularity
singularity shell /share/apps/gpu/singularity/sdsc_ubuntu_tf1.1_keras_R.img
```

[3] Launch the notebook

```
jupyter notebook --no-browser --ip="*" &
```

This will give you an address which has localhost in it and a token. Something like:

http://localhost:8888/?token=389587c9d1b69f8f595e7d8bfdd83c9961ed26b8b3f1bb3e

You can replace localhost with comet-XX-YY.sdsc.edu and then paste it into your browser. That should get you into the running notebook. From there everything should be working as a regular notebook. Note: This token is your auth so don't email/send it around (I already stopped the above link).