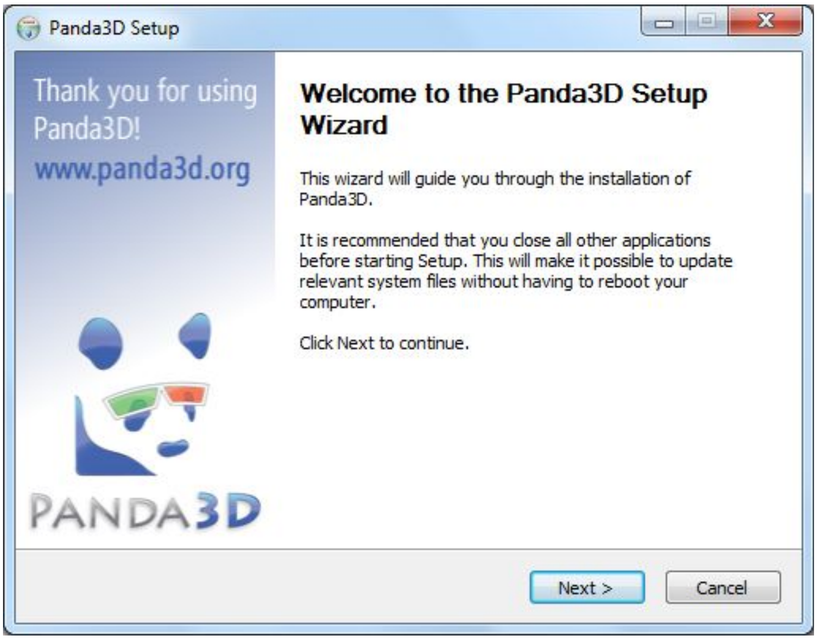
Docker Basics

Agenda

- What is containerization?
- Why we need containerization?
- What is docker?
- Installing Docker (Linux, Mac only)
- Dockerfile Jupyter example - creating and running

What is containerization ?

- Containers are programs with all dependencies installed. (like those of windows, where you click Next)
- Like Virtual-Machines, but lighter.



Welcome to the Panda3D Setup Wizard

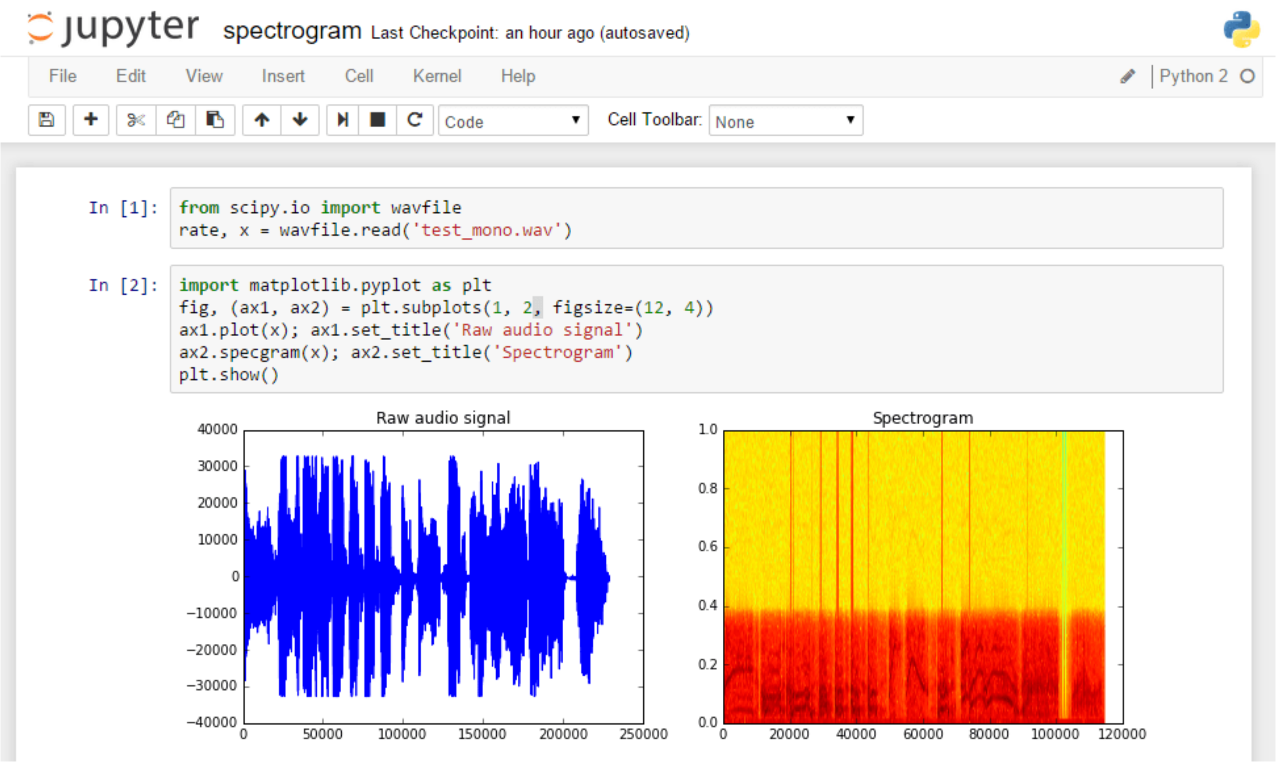
This wizard will guide you through the installation of Panda3D.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Why would you need containers?

- Analysis with Jupyter notebook which you want to share with your friends, so they can tinker around with it and run themselves.



In [1]:

```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

In [2]:

```
import matplotlib.pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram')
plt.show()
```

Issues

- First they will have to setup/install correct libraries (specific versions)
- Everyone might be on different OS - different instructions for all (Linux, Windows, Mac)
- Each person had to do that by himself

Containers Solve these Issues

- Software Dependency Resolution (OS, Libraries)
- No Setup Time

What is Docker?

- Company which provides this containerization technology - a synonym for containerization

- Other compaies do exist :
    - rkt (aka rocket)
    - Mesos Containers
    - Windows Server Containers

Installing Docker

Ubuntu

In [ ]:

```
sudo apt-get update
# install
sudo apt install docker.io
# enable docker when system boots up
sudo systemctl start docker
sudo systemctl enable docker
# check version
docker --version
```

MAC

Go to this link and download the stable version

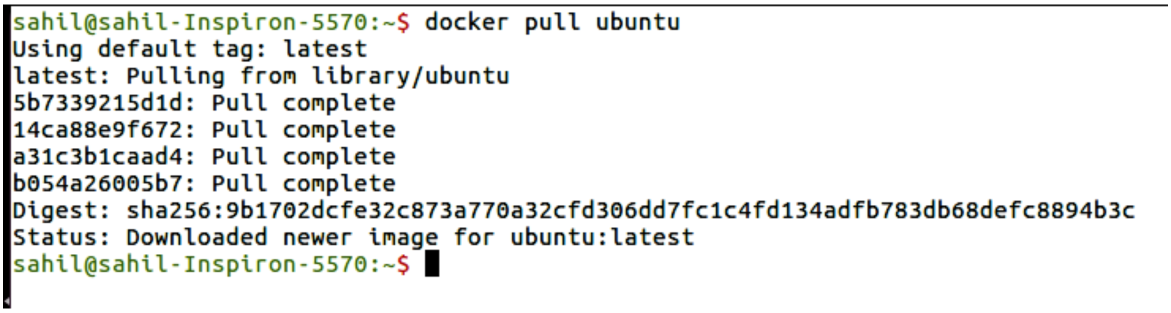
In [ ]:

```
https://docs.docker.com/v17.12/docker-for-mac/install/#download-docker-for-mac
```

Demo Create containers

In [ ]:

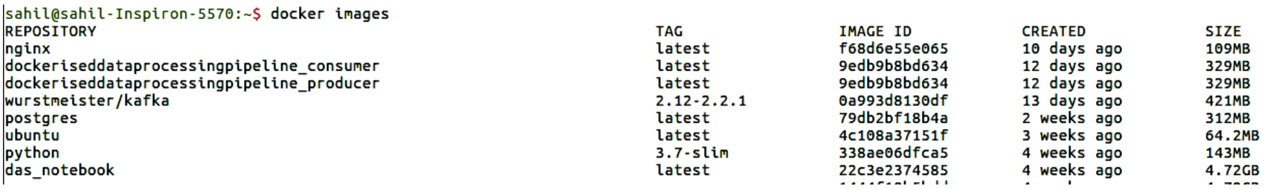
```
docker pull ubuntu
```



```
sahil@sahil-Inspiron-5570:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5b7339215d1d: Pull complete
14ca88e9f672: Pull complete
a31c3b1caad4: Pull complete
b054a26005b7: Pull complete
Digest: sha256:9b1702dcfe32c873a770a32cfd306dd7fc1c4fd134adfb783db68defc8894b3c
Status: Downloaded newer image for ubuntu:latest
sahil@sahil-Inspiron-5570:~$
```

In [ ]:

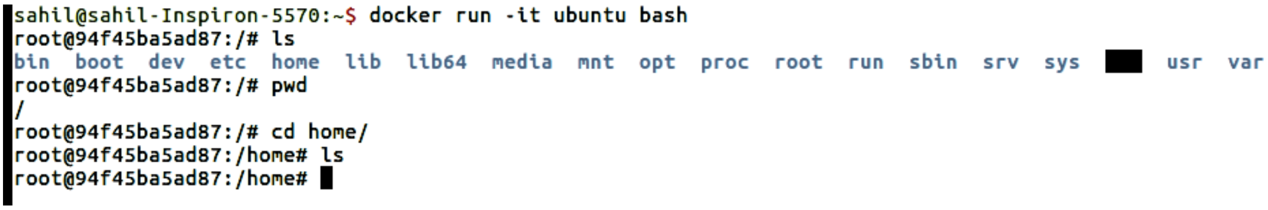
```
docker images
```



```
sahil@sahil-Inspiron-5570:~$ docker images
REPOSITORY                                        TAG                 IMAGE ID            CREATED             SIZE
nginx                                             latest              f68d6e55e065        10 days ago         109MB
dockeriseddataprocessingpipeline_consumer         latest              9edb9b8bd634        12 days ago         329MB
dockeriseddataprocessingpipeline_producer         latest              9edb9b8bd634        12 days ago         329MB
wurstmeister/kafka                                2.12-2.2.1          0a993d8130df        13 days ago         421MB
postgres                                          latest              79db2bf18b4a        2 weeks ago         312MB
ubuntu                                            latest              4c108a37151f        3 weeks ago         64.2MB
python                                            3.7-slim            338ae06dfca5        4 weeks ago         143MB
das_notebook                                      latest              22c3e2374585        4 weeks ago         4.72GB
```

In [ ]:

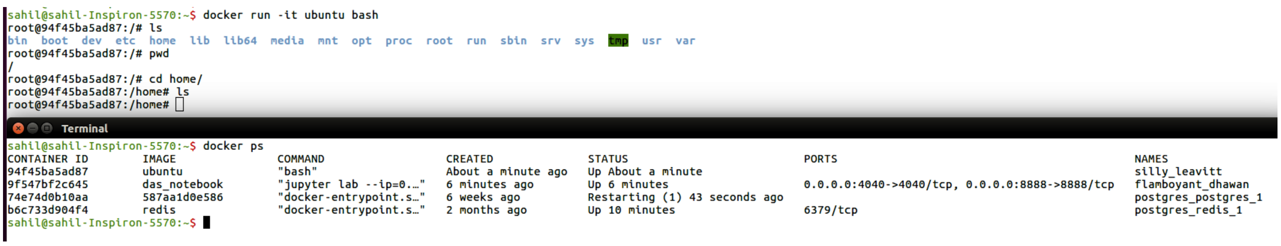
```
docker run -it ubuntu bash
```



```
sahil@sahil-Inspiron-5570:~$ docker run -it ubuntu bash
root@94f45ba5ad87:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  [redacted]  usr  var
root@94f45ba5ad87:/# pwd
/
root@94f45ba5ad87:/# cd home/
root@94f45ba5ad87:/home# ls
root@94f45ba5ad87:/home# 
```

In [ ]:

```
docker ps
```



```
sahil@sahil-Inspiron-5570:~$ docker run -it ubuntu bash
root@94f45ba5ad87:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@94f45ba5ad87:/# pwd
/
root@94f45ba5ad87:/# cd home/
root@94f45ba5ad87:/home# ls
root@94f45ba5ad87:/home# 
```

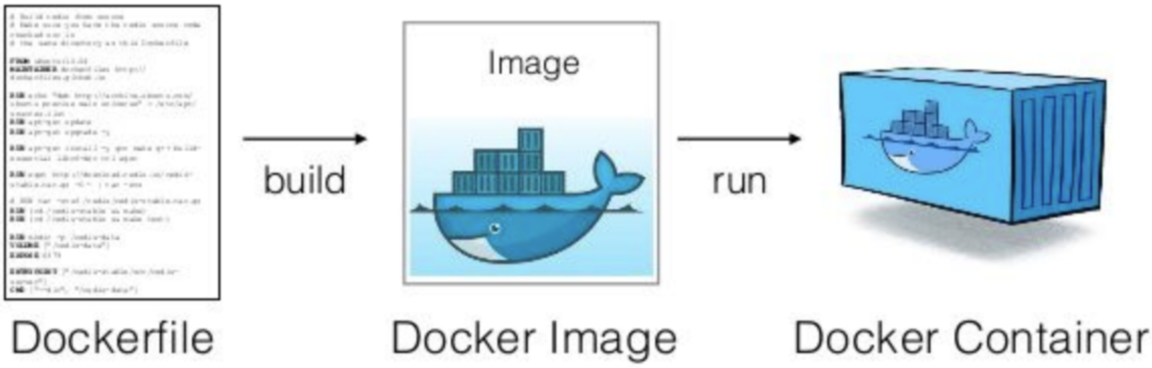
```
sahil@sahil-Inspiron-5570:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS                          PORTS                                            NAMES
94f45ba5ad87        ubuntu              "bash"                   About a minute ago   Up About a minute                                                                silly_leavitt
9f547bf2c645        das_notebook        "jupyter lab --ip=0.…"   6 minutes ago        Up 6 minutes                    0.0.0.0:4040->4040/tcp, 0.0.0.0:8888->8888/tcp   flamboyant_dhawan
74e74d0b10aa        587aa1d0e586        "docker-entrypoint.s…"   6 weeks ago          Restarting (1) 43 seconds ago                                                    postgres_postgres_1
b6c733d904f4        redis               "docker-entrypoint.s…"   2 months ago         Up 10 minutes                   6379/tcp                                         postgres_redis_1
sahil@sahil-Inspiron-5570:~$ 
```

Try out different docker images

- docker pull python
- docker pull postgres
- docker pull

Dockerfiles

Dockerfile (text file) create docker images which we run as containers



Lets Create our Own Jupyter Dockerfile

Create a file called **Dockerfile** and paste the following lines,

In [ ]:

```
FROM ubuntu

RUN apt-get update
RUN apt-get upgrade -y

RUN apt-get install -y python3 python3-dev python3-pip
```

Create the image using,

In [ ]:

```
docker build . -t my_python_image
```

Check if the image is built using,

In [ ]:

```
docker images
```

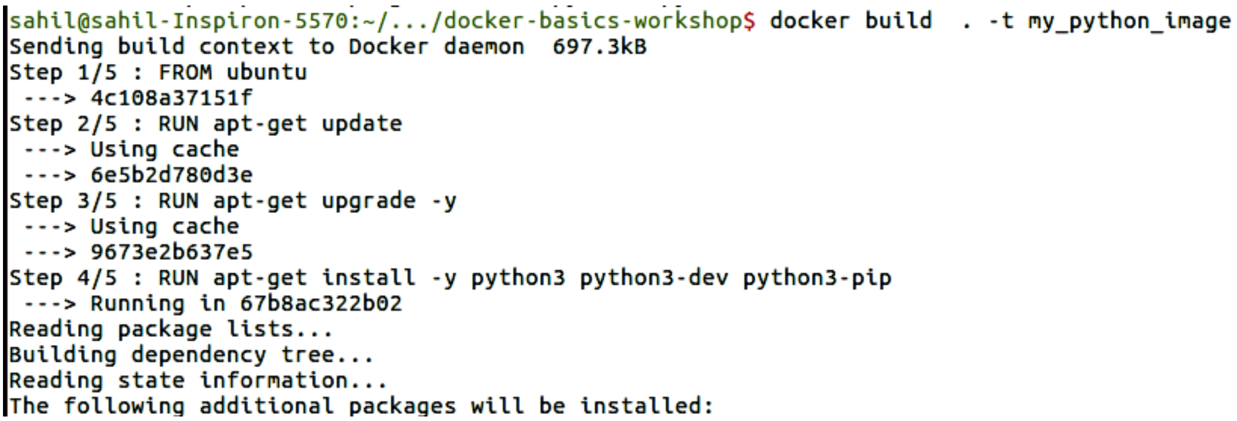
Now we can install all our popular python packages

Rebuild the image using the same command as previous time

In [ ]:

```
FROM ubuntu

RUN apt-get update
RUN apt-get upgrade -y

RUN apt-get install -y python3 python3-dev python3-pip

RUN pip3 install jupyter pandas numpy scipy
```

You will notice that only the new line is run. All previous instructions are loaded from cache. So only new changes are run henceforth.



```
sahil@sahil-Inspiron-5570:~/.../docker-basics-workshop$ docker build  . -t my_python_image
Sending build context to Docker daemon  697.3kB
Step 1/5 : FROM ubuntu
 ---> 4c108a37151f
Step 2/5 : RUN apt-get update
 ---> Using cache
 ---> 6e5b2d780d3e
Step 3/5 : RUN apt-get upgrade -y
 ---> Using cache
 ---> 9673e2b637e5
Step 4/5 : RUN apt-get install -y python3 python3-dev python3-pip
 ---> Running in 67b8ac322b02
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
```

Lets run the Docker container with jupyter notebook

Add the entrypoint command and rebuild the image

In [ ]:

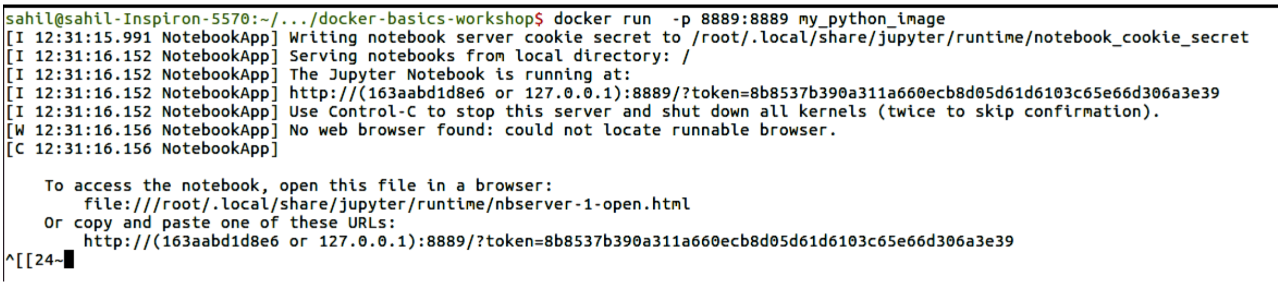
```
FROM ubuntu

RUN apt-get update
RUN apt-get upgrade -y

RUN apt-get install -y python3 python3-dev python3-pip

RUN pip3 install jupyter numpy

ENTRYPOINT ["jupyter", "notebook", "--ip=0.0.0.0" , "--allow-root", "--port=8889"]
```

Run the container

We have mapped the port 8889 which is on the contaier to port 8889 on the host

In [ ]:
```
docker run -p HOST_PORT:CONTAINER_PORT image-name
```

In [ ]:
```
docker run -p 8889:8889 my_python_image
```



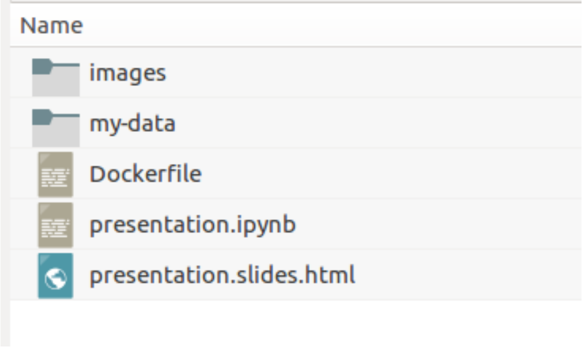
```
sahil@sahil-Inspiron-5570:~/.../docker-basics-workshop$ docker run  -p 8889:8889 my_python_image
[I 12:31:15.991 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[I 12:31:16.152 NotebookApp] Serving notebooks from local directory: /
[I 12:31:16.152 NotebookApp] The Jupyter Notebook is running at:
[I 12:31:16.152 NotebookApp] http://(163aabd1d8e6 or 127.0.0.1):8889/?token=8b8537b390a311a660ecb8d05d61d6103c65e66d306a3e39
[I 12:31:16.152 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 12:31:16.156 NotebookApp] No web browser found: could not locate runnable browser.
[C 12:31:16.156 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///root/.local/share/jupyter/runtime/nbserver-1-open.html
    Or copy and paste one of these URLs:
        http://(163aabd1d8e6 or 127.0.0.1):8889/?token=8b8537b390a311a660ecb8d05d61d6103c65e66d306a3e39
^[[24~
```

Persisting Data

If we want to save our files, we will attach a folder from our computer to another folder inside the docker container. Create a directory called **my-data** in your computer, we will store all our jupyter notebook analysis in here.

In [ ]:
```
docker run -v /HOST/DIRECTORY/FULL/PATH:/container/directory image-name
```

In [ ]:
```
docker run -p 8889:8889 -v "$(pwd)/my-data/":/home/ my_python_image
```



Name

images

my-data

Dockerfile

presentation.ipynb

presentation.slides.html