Containers
Introduction

- Containers
  - Separate applications and its dependencies.
  - Remove physical hardware requirements.
- Docker is based on Linux containers
Docker Stack

Host Hardware

Host OS

Docker Engine

Libraries

Application 1

Application 2

Application 3

CONTAINER
Docker

- Build an app once and run anywhere
- Containers take fewer resources
- Docker hub contains images available for use.
Docker Engine

- To do anything with docker you need to install docker engine.
- Get an OS specific image from [https://docs.docker.com/get-docker/](https://docs.docker.com/get-docker/)

To check that it is installed: `docker --version`
Docker Engine

- To check that it is installed: `docker --version`

Docker version 19.03.8, build afacb8b
Docker Client

- It is the user interface

- You communicate with it.

- It communicates with the Docker Daemon
Docker Daemon

- It executes the commands you send to the client.
- For example, building, running, and distributing your containers.
Dockerfile

- It contains the instructions to build a Docker image.

- For example, to install a software, set environmental variables etc.

- Use `docker build` command to build an *image* from it.

- Docker uses a Union file system.

- Once you run the image it is a *container*
Docker Project - dockerApp

- Python file: `code.py`

```python
import numpy as np
print('Numpy version:', np.__version__)
```

- `Requirements.txt` file

```
-i https://pypi.org/simple
numpy = 1.17.4
pandas = 0.24.2
matplotlib = 3.1.2
```
FROM python:3.7-slim

COPY requirements.txt /dockerApp/
WORKDIR /dockerApp

RUN pip install --upgrade pip \\
    && pip install --trusted-host pypi.python.org --requirement requirements.txt

COPY code.py /dockerApp

CMD ["python","code.py"]
Dockerfile

FROM python:3.7-slim

COPY requirements.txt /dockerApp/
WORKDIR /dockerApp

RUN pip install --upgrade pip \\ & & pip install --trusted-host pypi.python.org --requirement requirements.txt

COPY code.py /dockerApp

CMD ["python","code.py"]

- FROM: get a parent image from the docker hub
  Each command in the docker file uses this parent image.
- Copy requirements file into the dockerApp folder
Dockerfile

```dockerfile
FROM python:3.7-slim
COPY requirements.txt /dockerApp/
WORKDIR /dockerApp

RUN pip install --upgrade pip \ 
    && pip install --trusted-host pypi.python.org --requirement requirements.txt

COPY code.py /dockerApp

CMD ["python","code.py"]
```

- Set `dockerApp` as the working directory
- Run pip commands:
  - First to upgrade pip in the parent image
  - Second install all the dependencies in `requirements.txt` in the parent image.
Dockerfile

```
FROM python:3.7-slim

COPY requirements.txt /dockerApp/
WORKDIR /dockerApp

RUN pip install --upgrade pip \
    && pip install --trusted-host pypi.python.org --requirement requirements.txt

COPY code.py /dockerApp

CMD ["python","code.py"]
```

- Copy the app, `code.py` to the dockerApp
- Define what command gets executed.
Docker Image

docker build --tag=msmldockerimage.

Sending build context to Docker daemon 5.12kB
Step 1/6 : FROM python:3.7-slim
 | --> 74ac77e99873a
Step 2/6 : COPY requirements.txt /dockerApp/
 | --> e6e640d1c0db
Step 3/6 : WORKDIR /dockerApp
 | --> Running in ee49306b6437
Removing intermediate container ee49306b6437
 | --> 79ca3e4240516
Step 4/6 : RUN pip install --upgrade pip &
pip install --trusted-host pypi.python.org
 | --> Running in 95f472b978fe
--requirement requirements.txt
Requirement already up-to-date: pip in /usr/local/lib/python3.7/site-packages (20.0.2)
Collecting numpy==1.17.4
 | --> 0.24.2
- Downloading numpy-1.17.4-cp37-cp37m-manylinux1_x86_64.whl (20.0 MB)
Collecting pandas==0.24.2
 | --> 0.24.2-cp37-cp37m-manylinux1_x86_64.whl (10.1 MB)
Collecting matplotlib==3.1.2
 | --> 3.1.2-cp37-cp37m-manylinux1_x86_64.whl (13.1 MB)
Collecting pytz==2019.3
 | --> 2019.3-py2.py3-none-any.whl (509 kB)
Collecting python-dateutil==2.5.0
 | --> 2.5.0
- Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting kiwisolver==1.2.0
 | --> 1.2.0-cp37-cp37m-manylinux1_x86_64.whl (88 kB)
Collecting cycler==0.10
 | --> 0.10.0-cp37-cp37m-manylinux1_x86_64.whl (6.5 kB)
Collecting pyparsing==2.0.4,>2.1.6,<2.2.0
 | --> 2.4.7-py2.py3-none-any.whl (67 kB)
Collecting six==1.5
 | --> 1.14.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: numpy, pytz, six, python-dateutil, pandas, kiwisolver, cycler, pyparsing, matplotlib
Successfully installed cycler-0.10.0 kiwisolver-1.2.0 matplotlib-3.1.2 numpy-1.17.4 pandas-0.24.2 pyparsing-2.4.7 python-dateutil-2.8.1 pytz-2019.3 six-1.14.0
Removing intermediate container 95f472b978fe
 | --> b34a6b08f847
Step 5/6 : COPY code.py /dockerApp
 | --> 0c1636293309
Step 6/6 : CMD ['python","code.py"]
 | --> Running in 676d95b34c9f
Removing intermediate container 676d95b34c9f
 | --> ce20b48ceeb4
Successfully built ce20b48ceeb4
Successfully tagged msmldockerimage:latest
# List all local images

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>msmldockerimage</td>
<td>latest</td>
<td>ce20b48cecb4</td>
<td>About a minute ago</td>
<td>386MB</td>
</tr>
<tr>
<td>firstdockerimage</td>
<td>latest</td>
<td>d7e061293f39</td>
<td>18 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>c43425b4027e</td>
<td>18 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>4f6c8ffa8e58</td>
<td>22 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>337518bfbc18</td>
<td>26 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>dockimage</td>
<td>latest</td>
<td>037e9db66dca</td>
<td>16 hours ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>8b7f78c036ce</td>
<td>16 hours ago</td>
<td>179MB</td>
</tr>
<tr>
<td>helloapp</td>
<td>v1</td>
<td>b9485b71c61c</td>
<td>46 hours ago</td>
<td>1.22MB</td>
</tr>
<tr>
<td>python</td>
<td>3.7-slim</td>
<td>74ac77e9873a</td>
<td>2 weeks ago</td>
<td>179MB</td>
</tr>
<tr>
<td>busybox</td>
<td>latest</td>
<td>83aa35a1c79</td>
<td>5 weeks ago</td>
<td>1.22MB</td>
</tr>
<tr>
<td>jupyter/datascience-notebook</td>
<td>latest</td>
<td>029fd3e52059</td>
<td>11 months ago</td>
<td>5.49GB</td>
</tr>
<tr>
<td>hello-world</td>
<td>latest</td>
<td>fce289e99eb9</td>
<td>15 months ago</td>
<td>1.84kB</td>
</tr>
</tbody>
</table>

```bash
docker run --rm msmldockerimage
```

Numpy version: 1.17.4
Docker hub

- There is a container registry at Docker hub

- 1 free private repository

- Go to https://hub.docker.com/

- Create a username followed by a new repository, msml605 (private)
Docker hub

- `<username>/msml605 (private)`

Docker commands

To push a new tag to this repository,

```
docker push nayeemnz/msml605:tagname
```

- Associate your image to this repository on docker hub

```
docker tag msmldockerimage:latest nayeemnz/msmldockerimage:latest
```
List docker images

- Associate your image to this repository on docker hub

```
docker tag msmldockerimage:latest nayeemnz/msmldockerimage:latest
```

- `docker image ls`

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>msmldockerimage</td>
<td>latest</td>
<td>ce20b48cecb4</td>
<td>31 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>nayeemnz/msmldockerimage</td>
<td>latest</td>
<td>ce20b48cecb4</td>
<td>31 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>firstdockerimage</td>
<td>latest</td>
<td>d7e061293f39</td>
<td>48 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>c43425b4027e</td>
<td>49 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>4fdd8fa8e58</td>
<td>52 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>337518f8bc18</td>
<td>56 minutes ago</td>
<td>386MB</td>
</tr>
<tr>
<td>dockimage</td>
<td>latest</td>
<td>037e9db16dca</td>
<td>17 hours ago</td>
<td>386MB</td>
</tr>
<tr>
<td>&lt;none&gt;</td>
<td>&lt;none&gt;</td>
<td>8b7f78c036ce</td>
<td>17 hours ago</td>
<td>179MB</td>
</tr>
<tr>
<td>helloapp</td>
<td>v1</td>
<td>b9485b71c61c</td>
<td>47 hours ago</td>
<td>1.22MB</td>
</tr>
<tr>
<td>python</td>
<td>3.7-slim</td>
<td>74ac77e9873a</td>
<td>2 weeks ago</td>
<td>179MB</td>
</tr>
<tr>
<td>busybox</td>
<td>latest</td>
<td>83aa35a1c79</td>
<td>5 weeks ago</td>
<td>1.22MB</td>
</tr>
<tr>
<td>jupyter/datascience-notebook</td>
<td>latest</td>
<td>029fd3e52059</td>
<td>11 months ago</td>
<td>5.49GB</td>
</tr>
<tr>
<td>hello-world</td>
<td>latest</td>
<td>fce289e99eb9</td>
<td>15 months ago</td>
<td>1.84kB</td>
</tr>
</tbody>
</table>
Push local image to docker

- You may need to login

```
docker login

docker push nayeemmmz/msmldockerimage
```

The push refers to repository [docker.io/nayeemmmz/msmldockerimage]
9f1eefabe514: Pushed
a158808d8ac7: Pushed
50f7a330b9f2: Pushed
76d8d23d2ffce: Mounted from library/python
3d93c3dfdc9a: Mounted from library/python
17e50df399c: Mounted from library/python
07081806a448: Mounted from library/python
c3a984abe8a8: Mounted from library/python
latest: digest: sha256:1d9db42b7b8a7394cb863ae6fb3ff24dc1cb2f50f676b4587e67d4b15bdaabee size: 1996
Docker hub

- There is a pull request
Docker hub

- Run the remote image

  ```
  docker run --rm nayeemmnz/msmldockerimage
  ```

- Or pull from docker hub and run locally

  ```
  docker pull nayeemmnz/msmldockerimage
  ```
Requirements file

- Dependency manager — pipenv

- Uses pip and virtualenv

- For user installation

```bash
pip install --upgrade setuptools wheel
pip install --user pipenv
```
Set up the PATH

- vi ~/.bash_profile

```bash
export PATH="/Users/nayeem/.local/bin:$PATH"
```

- In the terminal

```bash
source ~/.bash_profile

echo $PATH | tr ':' '
'
```
pipenv

Usage: pipenv [OPTIONS] COMMAND [ARGS]...

Options:
    --where         Output project home information.
    --venv          Output virtualenv information.
    --py            Output Python interpreter information.
    --envs          Output Environment Variable options.
    --rm            Remove the virtualenv.
    --bare          Minimal output.
    --completion    Output completion (to be eval'd).
    --man           Display manpage.
    --support       Output diagnostic information for use in GitHub issues.
    --site-packages Enable site-packages for the virtualenv. [env var: PIPENV_SITE_PACKAGES]
    --python TEXT   Specify which version of Python virtualenv should use.
    --three / --two Use Python 3/2 when creating virtualenv.
    --clear         Clears caches (pipenv, pip, and pip-tools). [env var: PIPENV_CLEAR]
    -v, --verbose   Verbose mode.
    --pypi-mirror TEXT Specify a PyPI mirror.
    --version       Show the version and exit.
    -h, --help      Show this message and exit.
pipenv - shell to install packages

```
(base) Mohammds-MBP:dockerApp nayeem$ pipenv shell
Creating a virtualenv for this project...
Pipfile: /Users/nayeem/Documents/UMD/MSML605/Code/dockerApp/Pipfile
Using /anaconda3/bin/python (3.7.3) to create virtualenv...
  Creating virtual environment...created virtual environment CPython3.7.3.final.0-64 in 686ms
  creator CPython3Posix(dest=/Users/nayeem/.local/share/virtualenvs/dockerApp-t1Fthxfm, clear=False, global=False)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app_data_dir=/Users/nayeem/Library/Application Support/virtualenv/seed-app-data/v1.0.1)
  activators BashActivator, CShellActivator, FishActivator, PowerShellActivator, PythonActivator, XonshActivator

✔ Successfully created virtual environment!
Virtualenv location: /Users/nayeem/.local/share/virtualenvs/dockerApp-t1Fthxfm
requirements.txt found, instead of Pipfile! Converting...
✔ Success!
```
pipenv - to install packages

```bash
(dockerApp) bash-3.2$ pipenv install pandas
Installing pandas...
Adding pandas to Pipfile's [packages]...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
✓ Success!
Updated Pipfile.lock (0b6fa6)!
Installing dependencies from Pipfile.lock (0b6fa6)...

(dockerApp) bash-3.2$ pipenv install numpy
Installing numpy...
Adding numpy to Pipfile's [packages]...
✓ Installation Succeeded
Installing dependencies from Pipfile.lock (0b6fa6)...

(dockerApp) bash-3.2$ pipenv run python code.py

Numpy version: 1.17.4
```
pipenv - requirements file

```sh
pipenv lock -r >requirements.txt
```

```
(dockerApp) bash-3.2$ cat requirements.txt
-i https://pypi.org/simple
cycler==0.10.0
kiwisolver==1.2.0
matplotlib==3.1.2
numpy==1.17.4
pandas==0.24.2
pyparsing==2.4.7
python-dateutil==2.8.1
pytz==2019.3
six==1.14.0
```