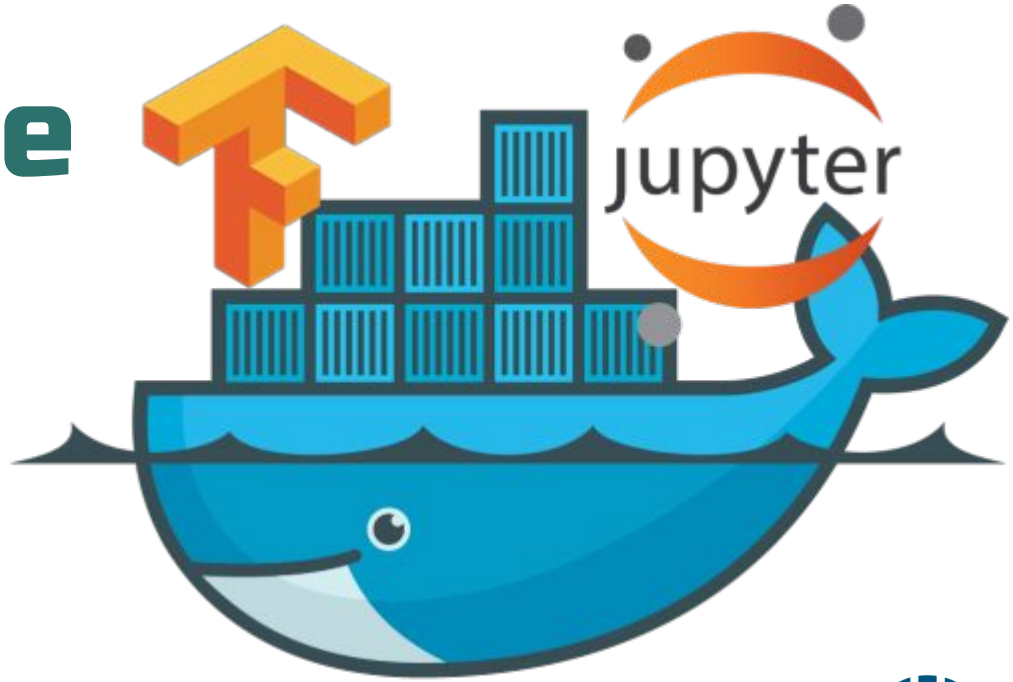
Docker for Data Science

Nottingham-Birmingham
Extragalactic Workshop,
September 10th-12th 2018



Nan Li
University of Nottingham

nan.li@nottingham.ac.uk

The University of Nottingham

Hi All,

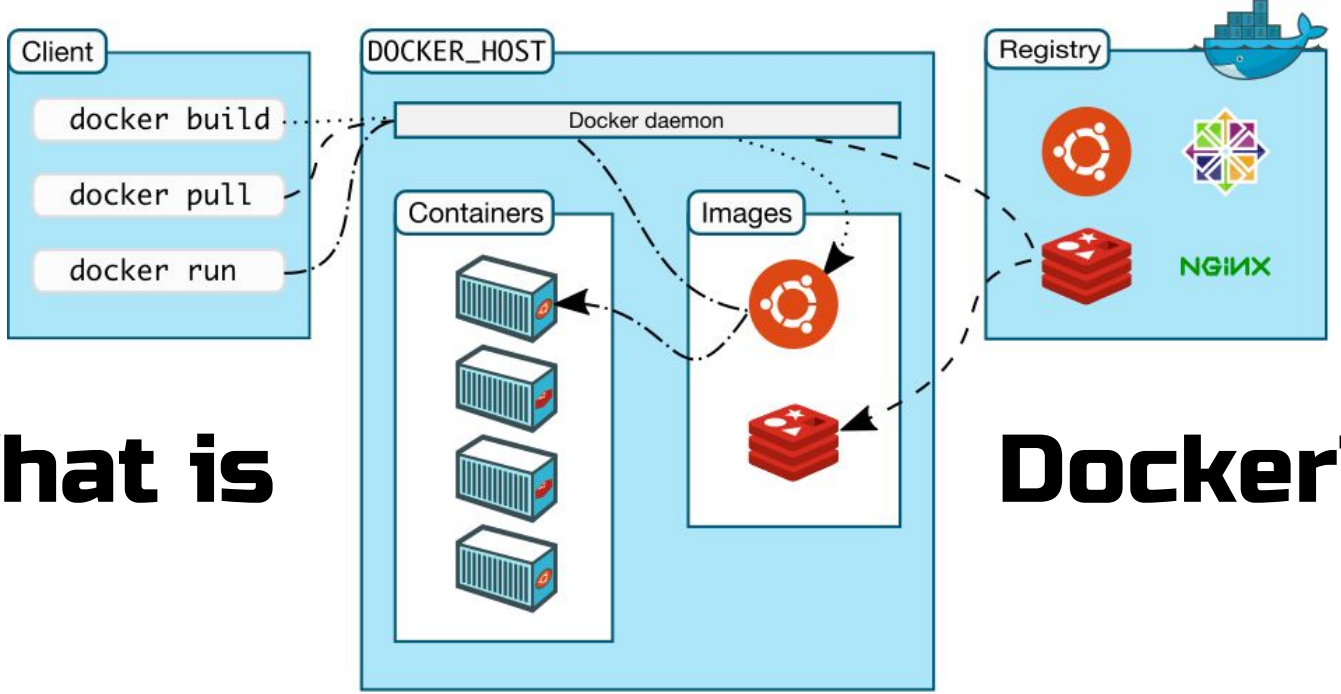
This presentation is for the tutorial on ***Applying Docker to Make Your Codes Portable*** during the Machine Learning Workshop at the University of Nottingham in September 2018.

I will keep maintaining the slides to make the tutorial clearer and will add more content about using ***nvidia-docker*** for deep learning with GPUs. Hopefully, it would be helpful for your project.

If you have any questions or suggestions, please do not hesitate to drop comments on the slides directly or send me an email. Your kind feedbacks are highly appreciated.

All the best,

Nan Li (nan.li@nottingham.ac.uk)

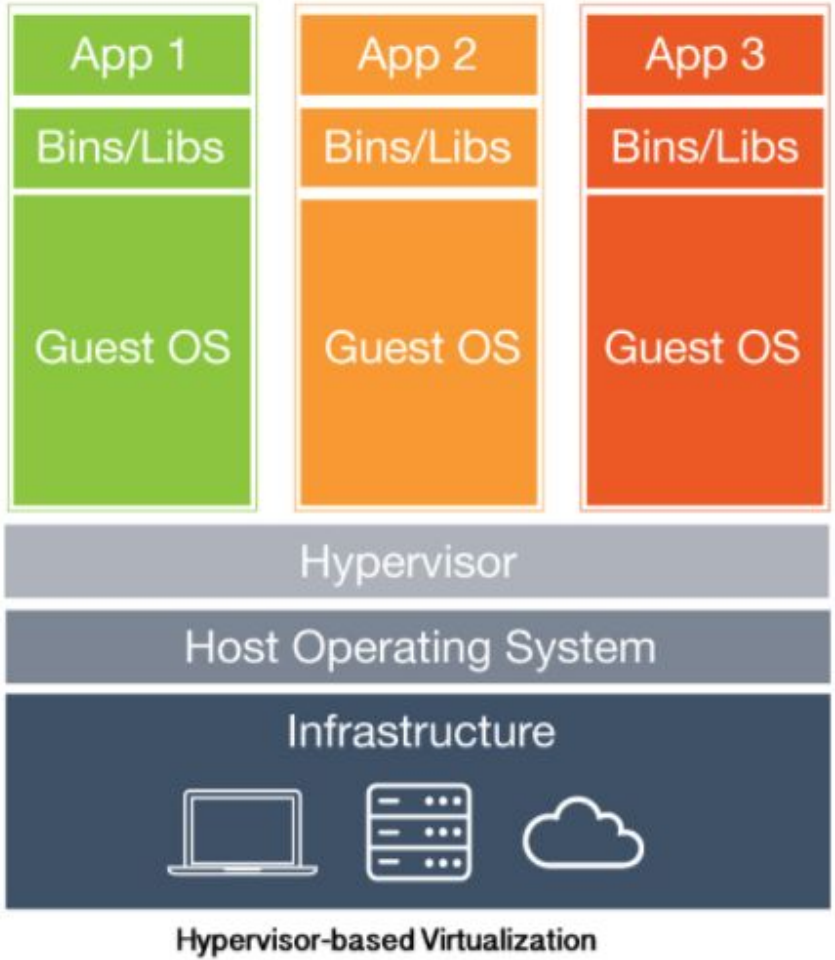


What is Docker?

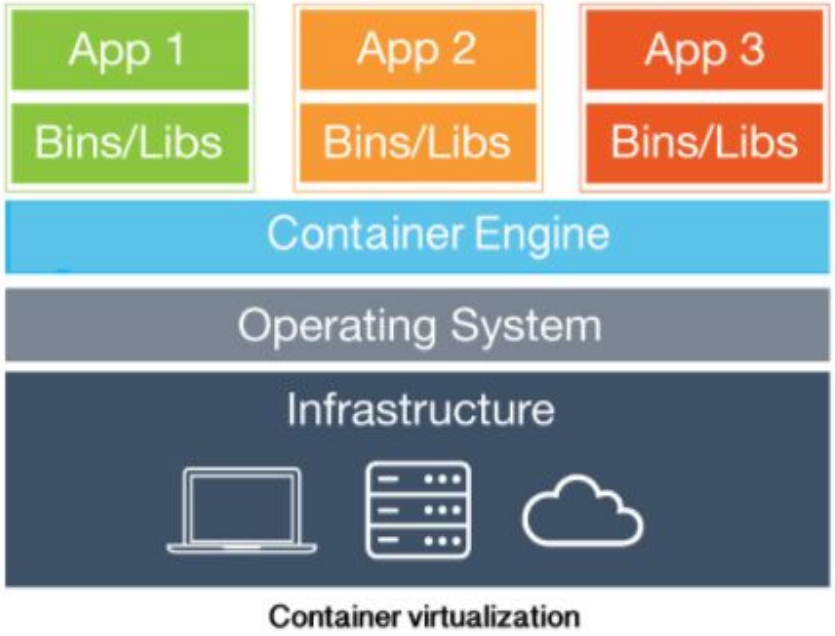
https://docs.docker.com/engine/docker-overview/

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

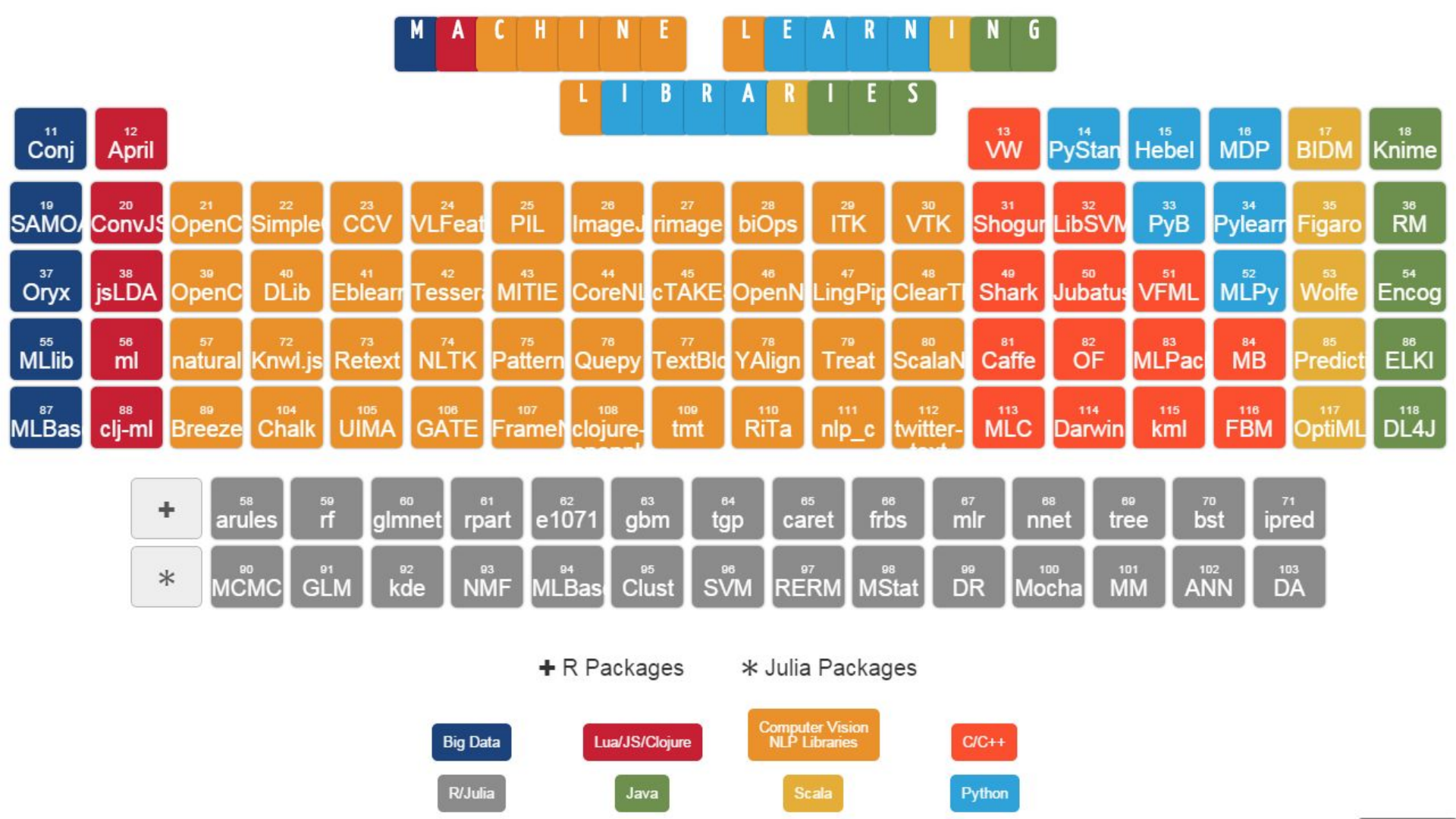
VM vs Docker



Hypervisor-based Virtualization

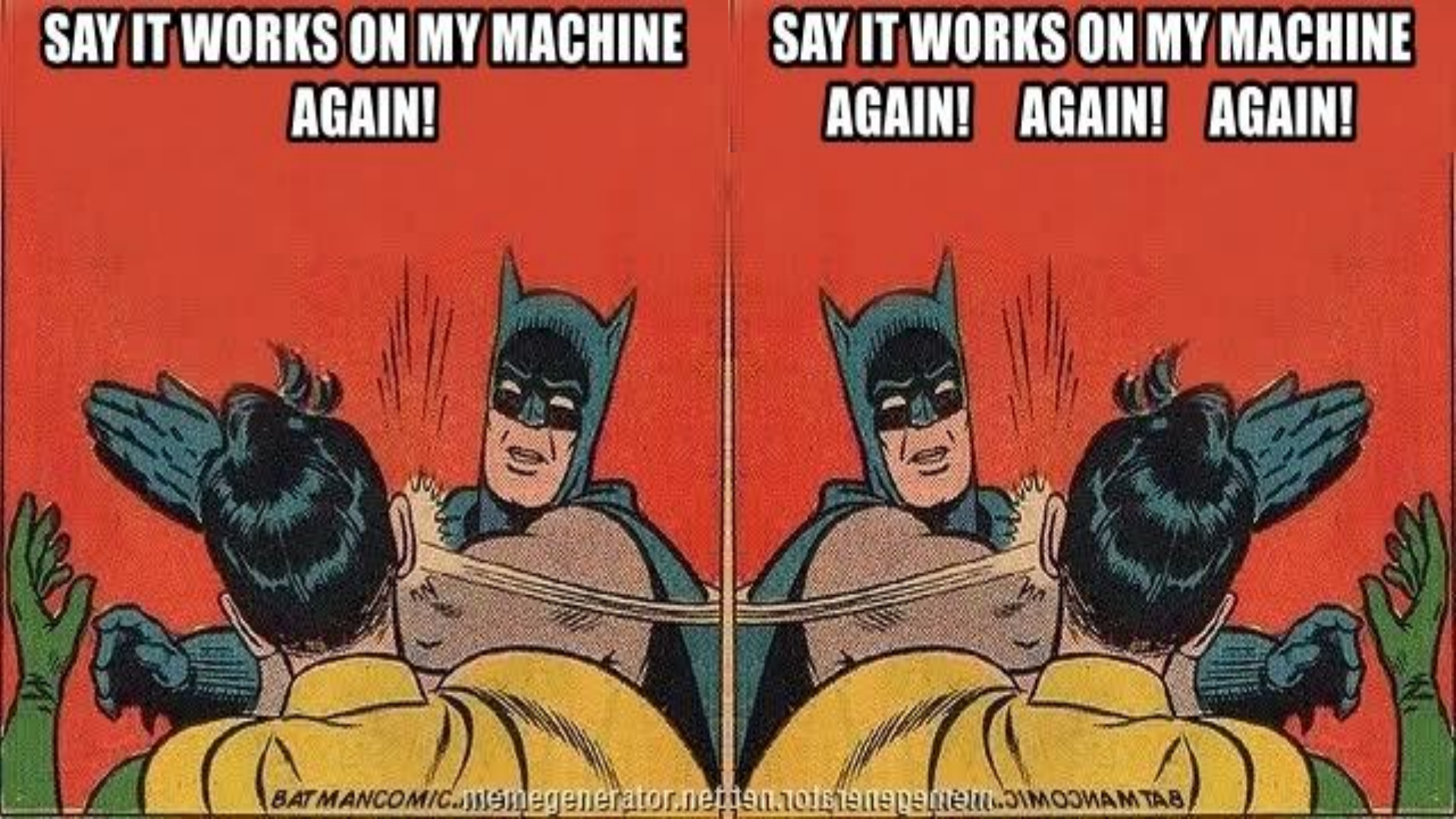


Container virtualization


MACHINE LEARNING
LIBRARIES
11 Conj
12 April
13 VW
14 PyStan
15 Hebel
16 MDP
17 BIDM
18 Knime
19 SAMOA
20 ConvJS
21 OpenC
22 Simple
23 CCV
24 VLFeat
25 PIL
26 ImageJ
27 rimage
28 biOps
29 ITK
30 VTK
31 Shogun
32 LibSVM
33 PyB
34 Pylearn
35 Figaro
36 RM
37 Oryx
38 jsLDA
39 OpenC
40 DLib
41 Eblearn
42 Tesser
43 MITIE
44 CoreNL
45 cTAKE
46 OpenN
47 LingPip
48 ClearTl
49 Shark
50 Jubatus
51 VFML
52 MLPy
53 Wolfe
54 Encog
55 MLlib
56 ml
57 natural
72 Knwl.js
73 Retext
74 NLTK
75 Pattern
76 Quepy
77 TextBlo
78 YAlign
79 Treat
80 ScalaN
81 Caffe
82 OF
83 MLPac
84 MB
85 Predict
86 ELKI
87 MLBas
88 clj-ml
89 Breeze
104 Chalk
105 UIMA
106 GATE
107 FrameN
108 clojure-
109 tmt
110 RiTa
111 nlp_c
112 twitter-
113 MLC
114 Darwin
115 kml
116 FBM
117 OptiML
118 DL4J
+
58 arules
59 rf
60 glmnet
61 rpart
62 e1071
63 gbm
64 tgp
65 caret
66 frbs
67 mlr
68 nnet
69 tree
70 bst
71 ipred
*
90 MCMC
91 GLM
92 kde
93 NMF
94 MLBas
95 Clust
96 SVM
97 RERM
98 MStat
99 DR
100 Mocha
101 MM
102 ANN
103 DA
+ R Packages
* Julia Packages
Big Data
Lua/JS/Clojure
Computer Vision
NLP Libraries
C/C++
R/Julia
Java
Scala
Python

IT WORKS

on my machine


SAY IT WORKS ON MY MACHINE AGAIN!
SAY IT WORKS ON MY MACHINE AGAIN! AGAIN! AGAIN!
BATMANCOMIC.

Install

https://docs.docker.com/docker-for-mac/install/

https://docs.docker.com/docker-for-windows/install/

https://docs.docker.com/install/linux/docker-ce/ubuntu/

https://goo.gl/rcidq9

Sign up

https://hub.docker.com/

Download the Files for this Tutorial

https://goo.gl/YqKT7Y

Setup the Workdir on the Host Machine

1. Turned off the firewall and close all the Jupyter notebooks on the host machine.
2. Unzip the file you downloaded from the link above and go to the directory.
3. Remember the absolute path to this directory. It is the value of the **</host/shared>** in the slides below.

Test the Installation of Docker

```
$ cd </host/shared>
$ docker info
$ docker run hello-world
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

Pull a Docker Image of Ubuntu

```
$ docker pull ubuntu:18.04
$ docker run -it -p 8888:8888 -p 6006:6006 \
   -v </host/shared>:/root/shared ubuntu:18.04
```

```
 |uranus| ~/Desktop/DL-workshop-NM pwd -P
/Users/uranus/Desktop/DL-workshop-NM
 |uranus| ~/Desktop/DL-workshop-NM docker run -it -p 8888:8888 -p 6006:6006 -v /Users/uranus/Desktop
/DL-workshop-NM:/root/shared ubuntu:18.04
root@96639a0f6e39:/# uname -a
Linux 96639a0f6e39 4.9.93-linuxkit-aufs #1 SMP Wed Jun 6 16:55:56 UTC 2018 x86_64 x86_64 x86_64 GNU/
Linux
root@96639a0f6e39:/#
```

Install dependencies for Your Machine Learning Code

```
docker# apt-get update
docker# apt-get install python3-pip
docker# pip3 --no-cache-dir install \
            pandas h5py tensorflow \
            keras jupyter matplotlib
```

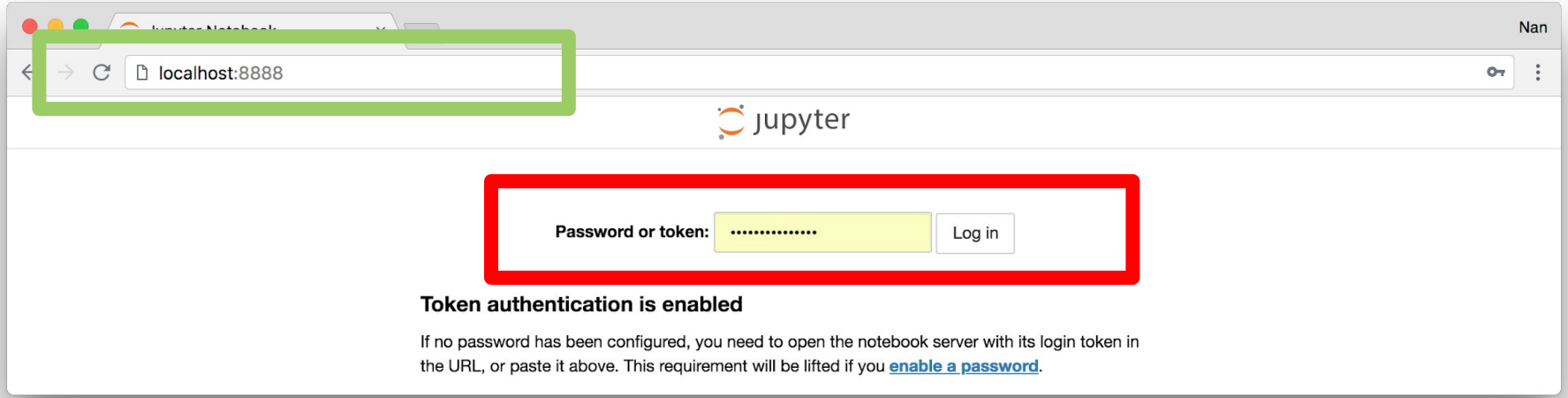
```
docker# cp -r /root/shared/jupyter /.jupyter
docker# cp -r /root/shared/notebooks /
docker# cp -r /root/shared/run_jupyter.sh /
```

Run the code

```
docker# jupyter-notebook --ip=0.0.0.0 \
          --allow-root
or
docker# /run_jupyter.sh --ip=0.0.0.0 \
          --allow-root
```

```
[I 12:11:18.252 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to s
kip confirmation).
[C 12:11:18.253 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://(f74dcc0af61a or 127.0.0.1):8888/?token=921ac5c09ff89061ccf4df4baaf3283b0f929a975213d
ad6
[I 12:11:30.572 NotebookApp] 302 GET / (172.17.0.1) 0.81ms
```



- Open a browser on the host machine and visit the address of “localhost:8888”.
- Copy the “token” highlighted in the previous slide and paste it into the red box, then log in.

- If this code runs well, it means that you have successfully constructed a container and run a deep learning code in it.
- Next, let's **save the change** and **upload it to your dockerhub**.

Push and Pull the Image

```
# Before save the change, you have to find the <container-id> of the running container.
$ docker ps -a
```

```
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS
                      PORTS                                           NAMES
5ebbacaeef32        linan7788626/dl-docker   "/run_jupyter.sh --i…"   19 minutes ago      Up 19 min
utes                  0.0.0.0:6006->6006/tcp, 0.0.0.0:8888->8888/tcp  goofy_darwin
dc4ed755c390        09c2764e2391           "/bin/sh -c '[\"jupyt…"  About an hour ago   Exited (2
) About an hour ago                                                 vigilant_albattani
e5a8ea2fa52e        09c2764e2391           "/bin/sh -c '[\"jupyt…"  About an hour ago   Exited (2
) About an hour ago                                                 happy_chaplygin
```

Push and Pull the Image

```
$ docker commit -m "<What did you do>" \
   -a "<Your name>" <container-id> \
   <repo_name>/<image_name>
$ docker push <repo_name>/<image-name>
```

```
$ docker pull <repo_name>/<image_name>
$ docker run -it -p 8888:8888 -p 6006:6006 \
   -v </host/shared>:/root/shared \
   <repo_name>/<image_name>
```

Dockerfile

You can construct the container with **Dockerfile** automatically. An example of Dockerfile is saved in the link provided at the beginning of this talk.

```
$ docker build -f Dockerfile \
   -t <repo_name>/<image_name>:latest .
$ docker run -it -p 8888:8888 -p 6006:6006 \
   -v </host/shared>:/root/shared \
   <repo_name>/<image_name>
```

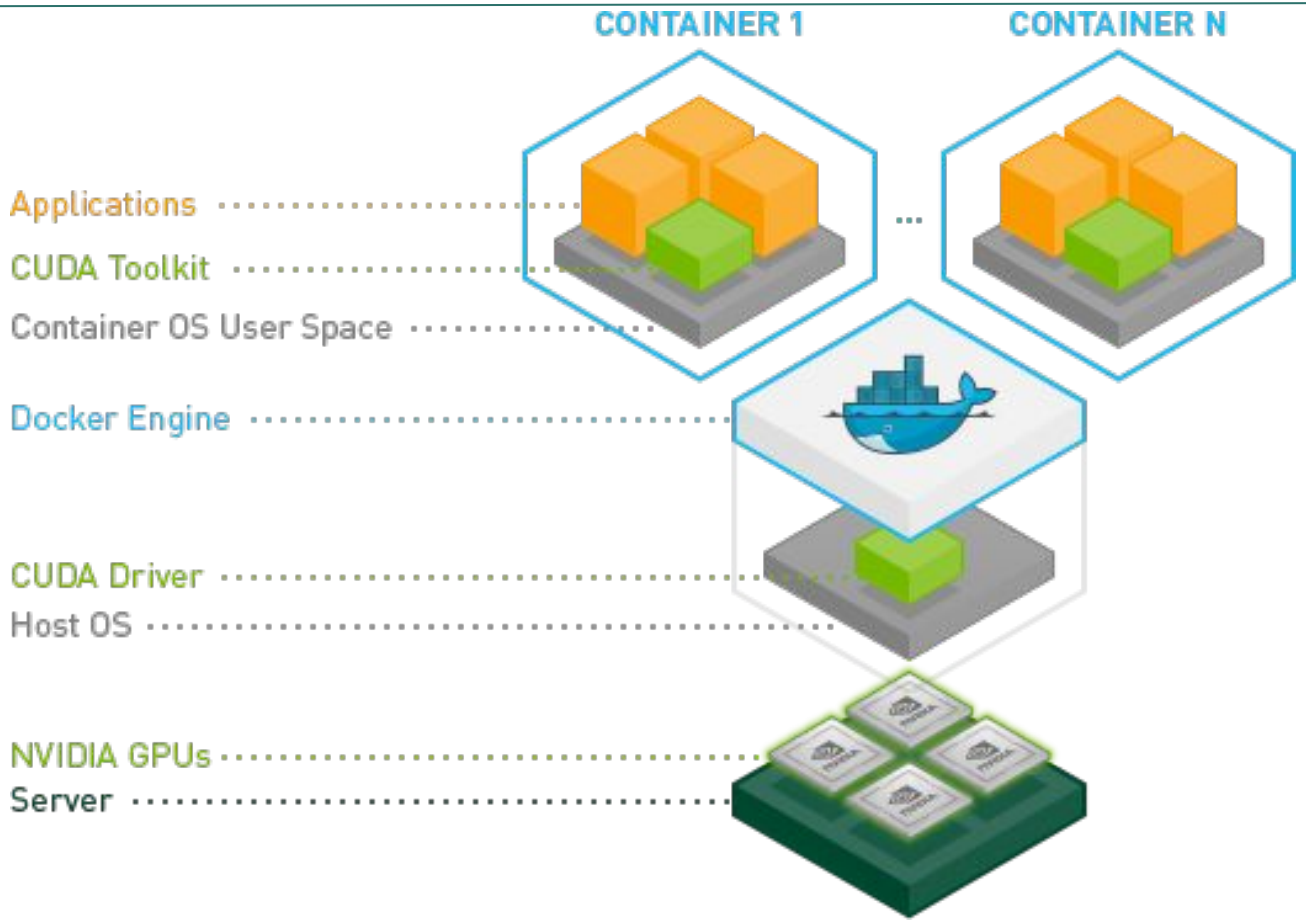
~/Desktop/DL-workshop-NM — vim Dockerfile

```
1  FROM ubuntu:18.04
2  LABEL maintainer="Nan Li <linan7788626@gmail.com>"
3
4  # Pick up some dependencies
5  RUN apt-get update && \
6      apt-get install -y python3-pip && \
7      apt-get clean && \
8      rm -rf /var/lib/apt/lists/*
9
10 RUN pip3 --no-cache-dir install \
11         h5py \
12         jupyter \
13         matplotlib \
14         pandas \
15         keras \
16         tensorflow
17
18 # Set up the notebook config.
19 COPY jupyter /root/.jupyter
20 COPY notebooks /notebooks
21 COPY run_jupyter.sh /
22
23 # TensorBoard
24 EXPOSE 6006
25 # Jupyter-notebook
26 EXPOSE 8888
27 # Setup Workdir
28 WORKDIR "/notebooks"
29
30 CMD ["/run_jupyter.sh", "--ip=0.0.0.0", "--allow-root"]
```

NORMAL 1:Dockerfile+ dockerfile utf-8[unix] 3% LN 1/30☰ : 1 ✹ mix-ind…

Nvidia-docker

https://goo.gl/NJer1E



Nvidia-docker

https://goo.gl/NJer1E

Coming Soon...

Useful Links

- https://docs.docker.com/get-started/
- https://awesome-docker.netlify.com/
- https://docker-curriculum.com/
- https://goo.gl/NjDiFB , Youtube Videos
- https://goo.gl/J9WKLw , Dockerfile Tutorial
- https://goo.gl/qCMQAj , Nvidia Docker