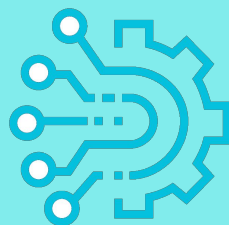
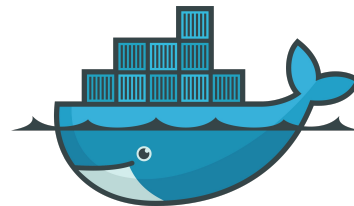


Why should we use..

Containers in Machine Learning



Machine
Learning



About me!

- Curious Learner
- Enterprise Architect by Profession
- Currently Learning “Maths for Programmers”



Contents

1

Introduction

Machine Learning | Deployments |
DevOps

2

Goals & Considerations

How do we utilize a Machine Learning
Model ?

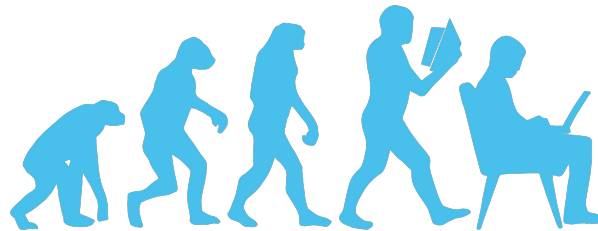
3

Containers | Docker | Demo

Deploying a Simple ML Application

Important information

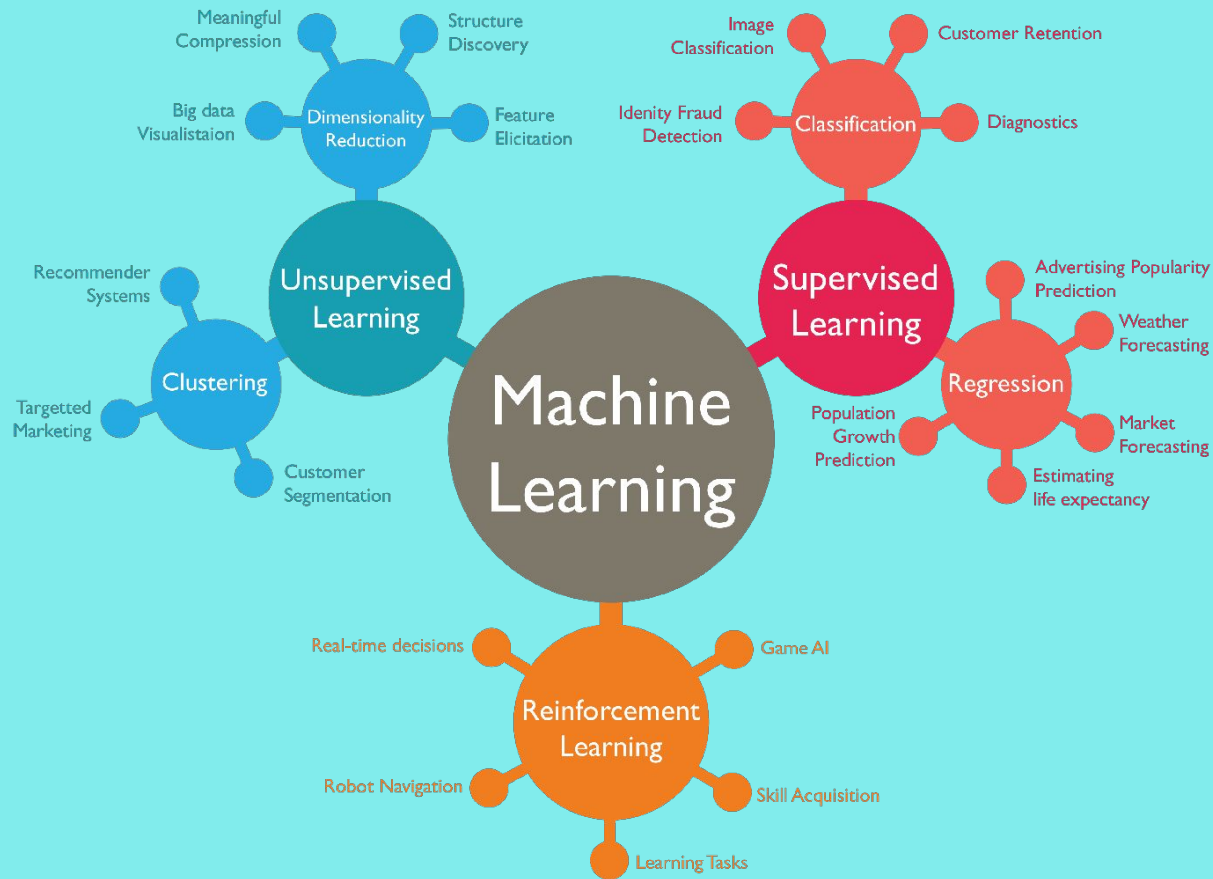
The Content presented
highlight my opinion and
understanding of the products.
Things could have evolved.



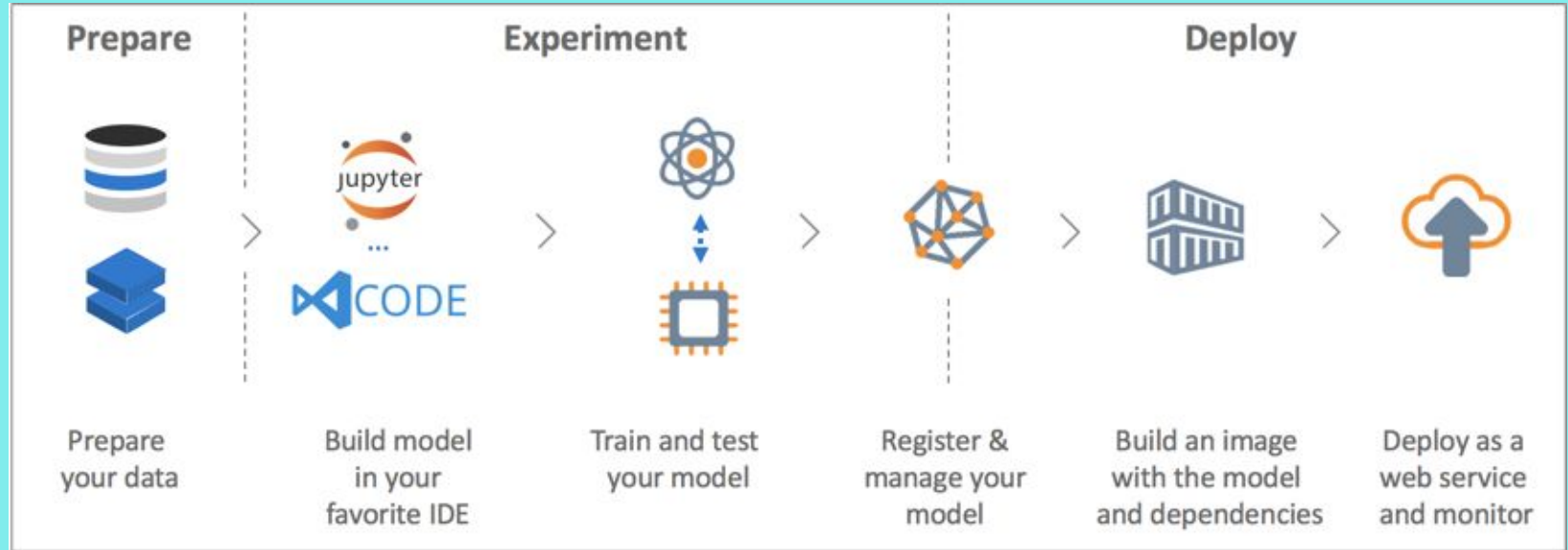


1. Introduction

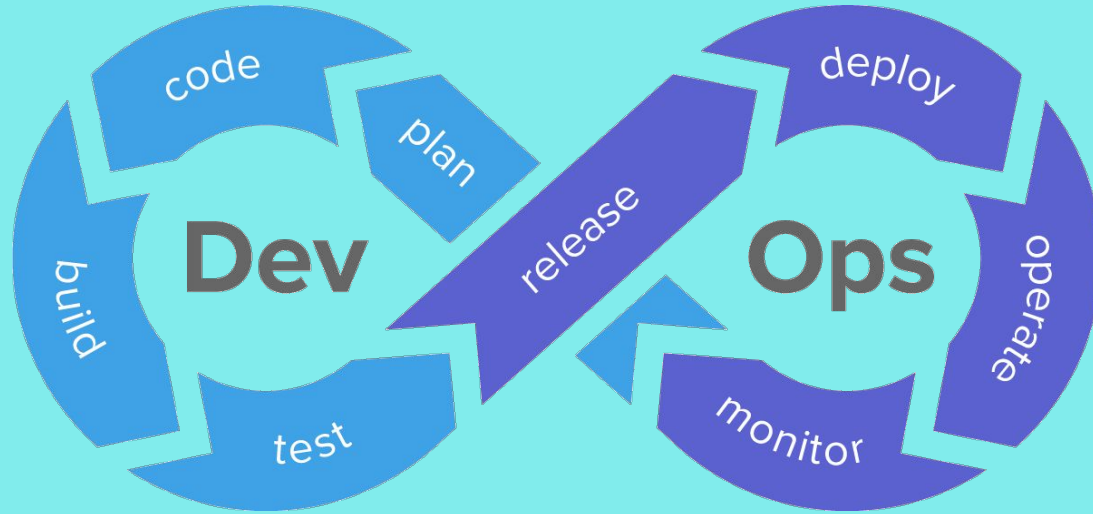
Machine Learning | Deployments | DevOps



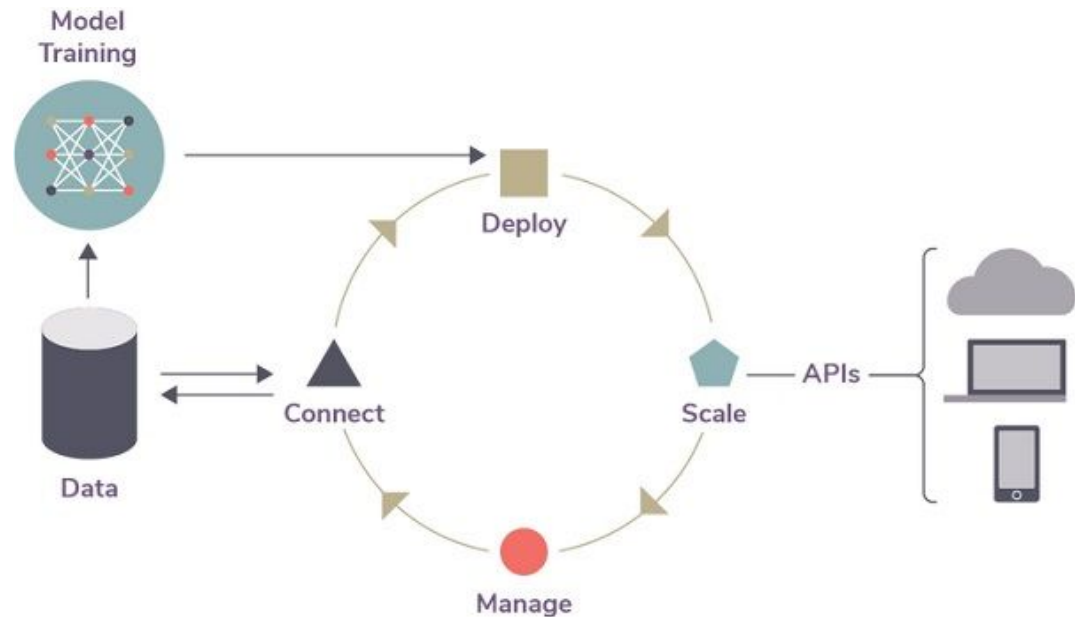
Machine Learning



Deployments



Dev Ops



Effectively it is a set of Practises to shorten the Software Lifecycle by Automating Routine tasks.

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK				
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS
	6 HOURS				2 MONTHS	2 WEEKS
	1 DAY					8 WEEKS
						5 DAYS

<https://xkcd.com/1205/>

Dev Ops



2. Goals & Considerations

How do we utilize a Machine Learning Model ?

“How many Mechanisms do we need to know to deploy our Code into Production ?”

How do we Standardize this ?

“How many Mechanisms do we need to know to deploy our Code into Production ?”

How do we Standardize this ?

Awesome Team





Flask



TENSORFLOW + KUBERNETES



Microsoft
Cognitive Services



Kubeflow



Amazon SageMaker



To Simplify..

1. Consume a Provider API (Azure ML, AWS etc)
2. Build and Deploy your own App which consumes a serialised model accessible over an API



Traditional Hosting (Option #2)

“How to keep my operating system updated without causing application downtime?”

“How to update/rollback applications without causing downtime?”

“How to provide HA/DR without incurring associated complexity?”

“How do I make sure my compute jobs can't hurt my database job?”

“How do I safely pack more work onto less machines?”

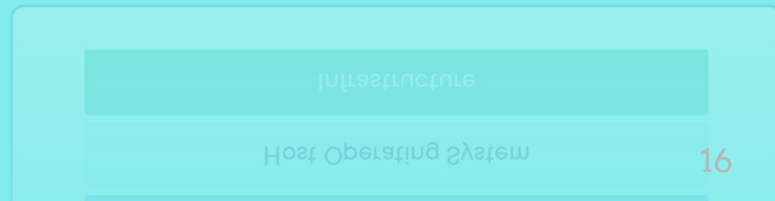
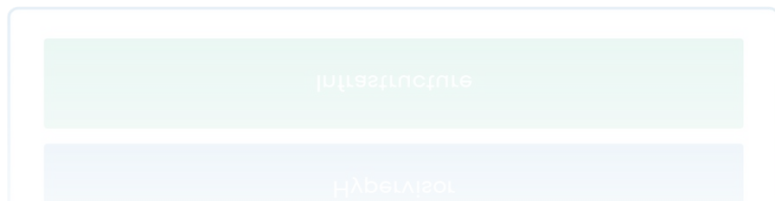
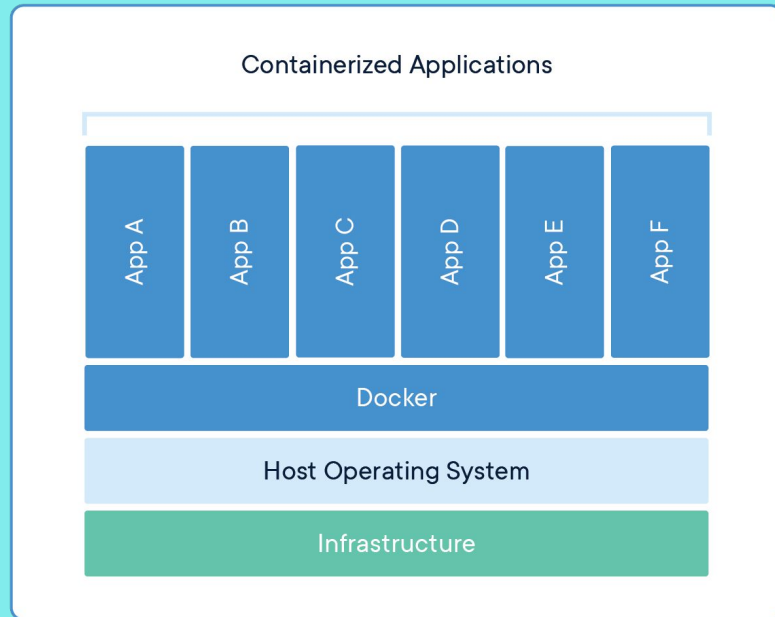
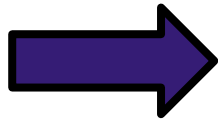
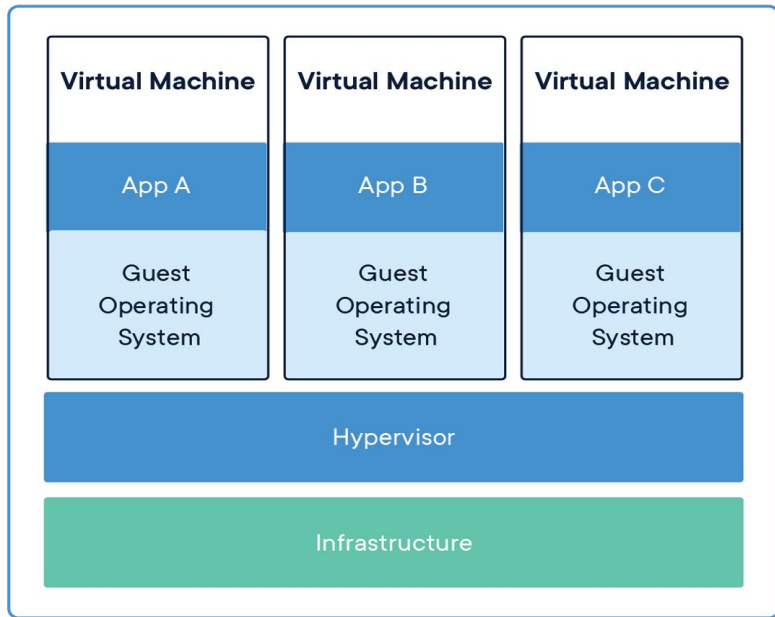
.war or .ear	Deployable Unit
Custom Configuration	JDBC driver, datasource, JMS queue, users
Application Server	Weblogic 10.x.y, Tomcat 9.x.y, WAS 8.x.y
Java Virtual Machine	Java 1.6.6_45 or Java 1.7.0_67
Operating System	Linux Kernel Version & Distribution
Server Hardware	4 Cores Intel Xeon, 32 GB Memory, 16 MB L3 Cache etc.

A large teal-colored geometric shape, resembling a stylized arrow or a corner, pointing towards the bottom right, located on the left side of the slide.

3. Containers | Docker | Demo

Deploying a Simple ML
Application

Build Once Run Anywhere



Container Terminology

Image: A packaged version of the dependencies (libraries and binaries) required to run a given process.

Container: A running instance of a given image.

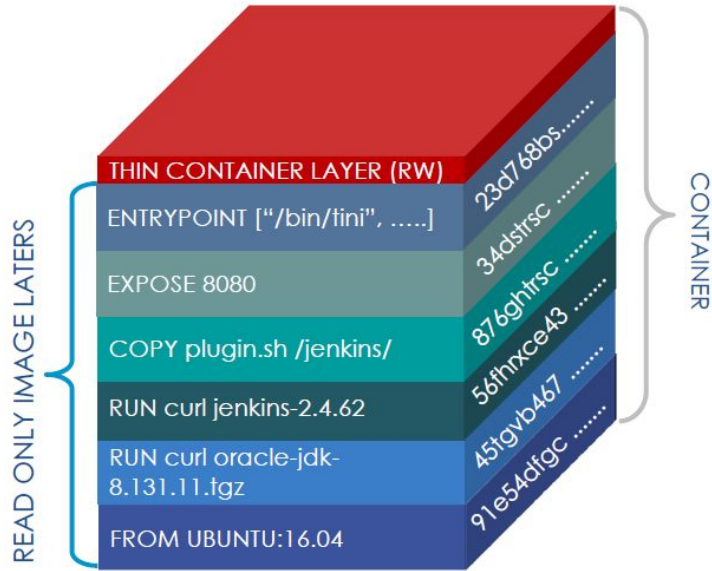
Engine: The Host OS process that is responsible for unpacking the image, abstracting the Host OS resources (disk, network, users, CPU, memory), and the lifecycle of the container.

Volume: Container storage for application data persistence.

Registry: A service that contains and can store and retrieve from repository of images remotely.

Cluster: A grouping of engines utilized by an orchestrator to host containers.

Docker Image



DOCKERFILE

```
FROM ubuntu:16.04
```

```
RUN curl http://download.oracle.com/jdk-8u131-linux-x64.tar.gz
```

```
RUN tar -zxvf jdk-8u131-linux-x64.tar.gz
```

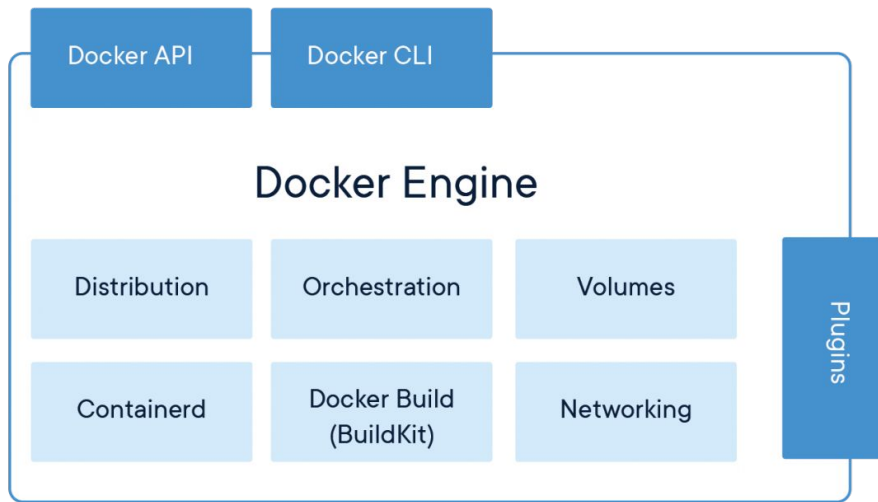
```
RUN curl http://mirrors.jenkins.io/warstable/latest/jenkins.war
```

```
COPY plugin.sh /jenkins/
```

```
EXPOSE 8080
```

```
ENTRYPOINT ["/bin/tini", "--", "/usr/local/bin/jenkins.sh"]
```

Docker Engine

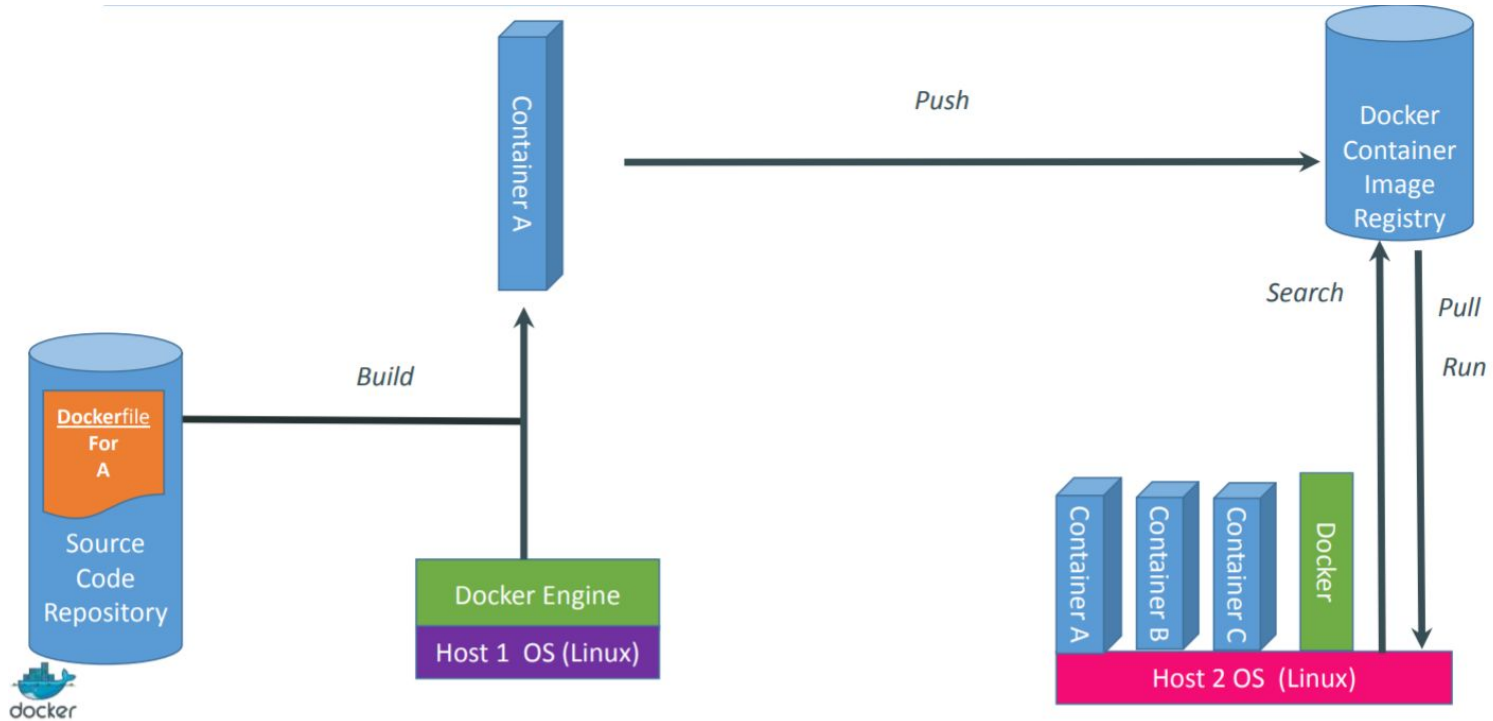


```
docker pull ubuntu
docker run -it --name ubuntu_example ubuntu /bin/bash
```

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers.

- Docker CLI
- Dockerfile
- Images
- Containers
- Docker Hub

A Simple Docker Workflow

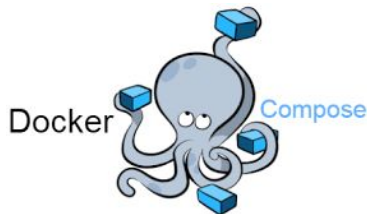


Steps to Dockerize any Application

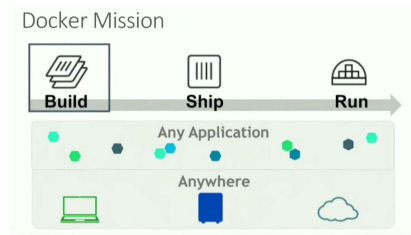
**Develop Your
App**



**Write a
Dockerfile**



**Build and Run
anywhere**



Docker #Key Commands

- `docker build` # Build an image from a Dockerfile
- `docker images` # List all images on a Docker host
- `docker run` # Run an image
- `docker ps` # List all running and stopped instances
- `docker stop` # Stop a running instances
- `docker rm` # Remove an instance
- `docker rmi` # Remove an image

Demo



Thank you
