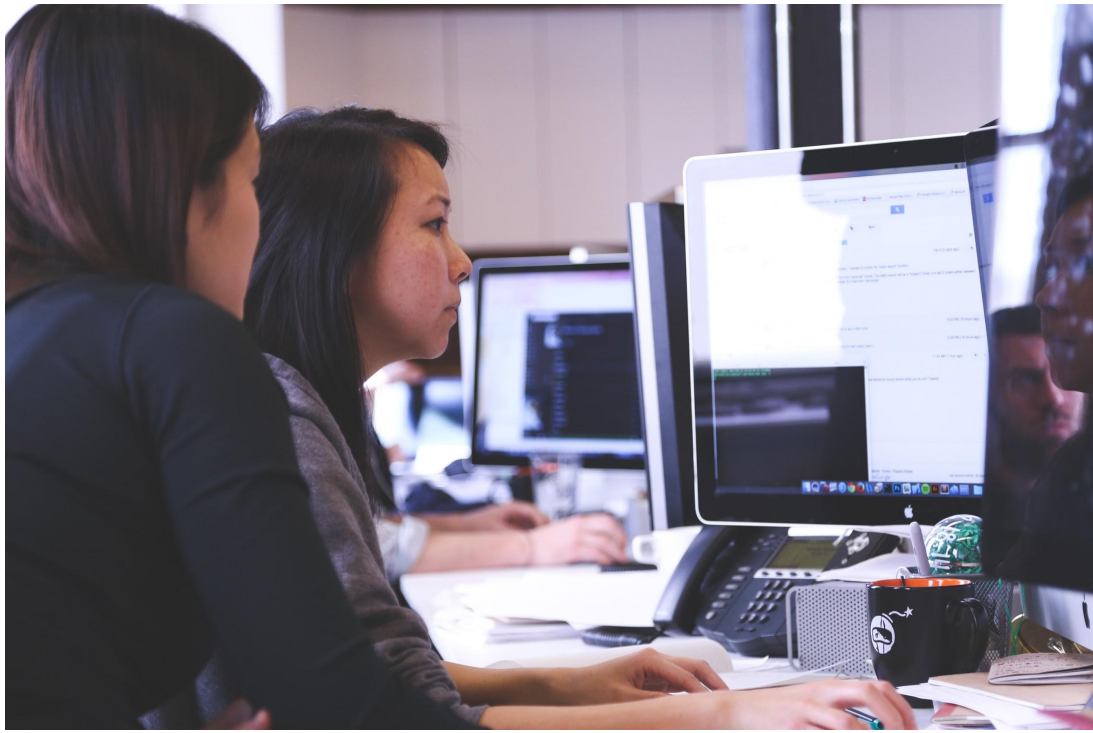
Part I

ML Models in Production

Getting your model ready for the real world

@justinjdn
Justin Norman



October 2020

ML Deployment Exercise

Let’s start with a simple example!

Run it yourself, or just follow along

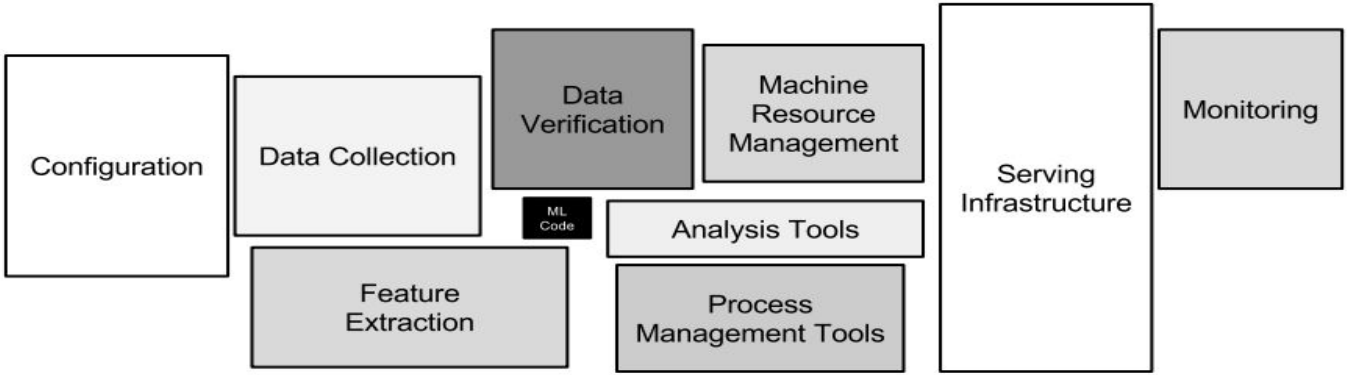
Code lives here: https://github.com/stbiadmin/USCProdML

Open: “ML Models in Production Workshop - USCMSBA.ipynb” in 

You’ll want to be running python 3+, everything else is in requirements.txt

Hidden technical debt in ML systems

Google Paper



Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

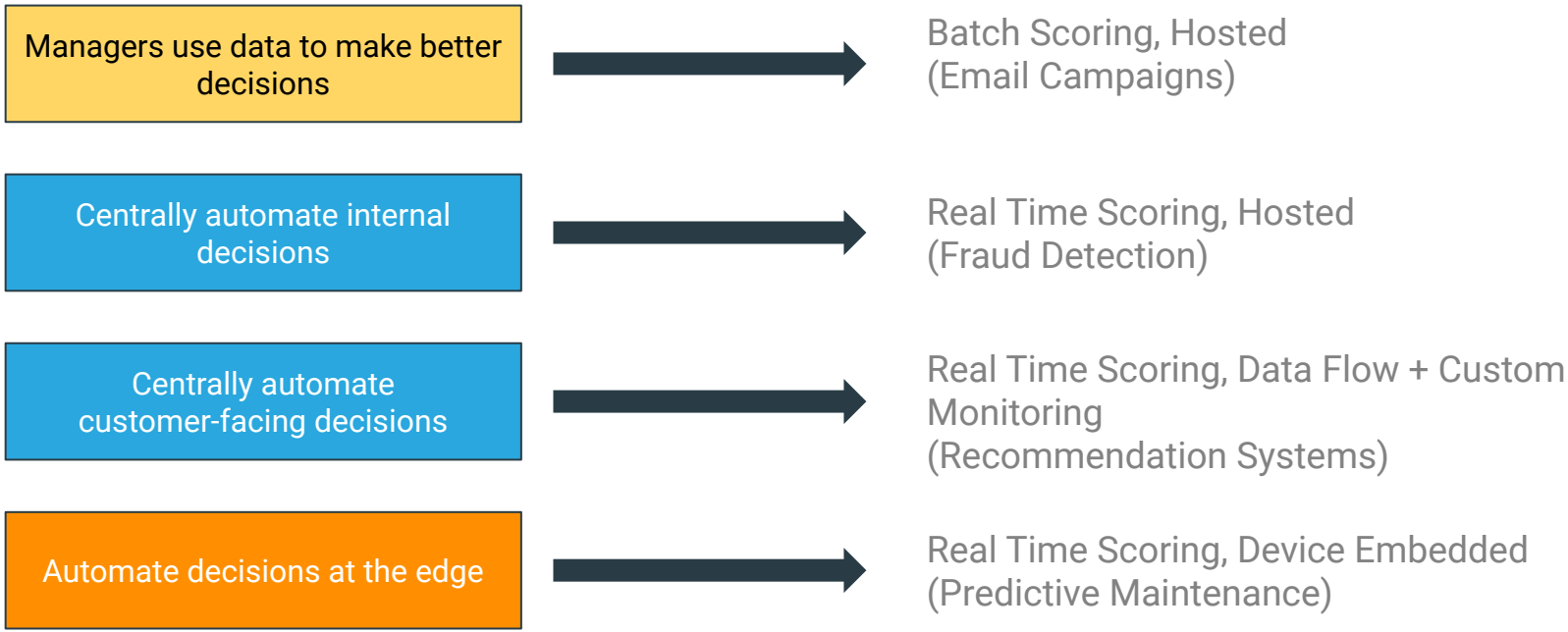
Source: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

ML Deployment Paradigms

What are the things we MUST remember to do?

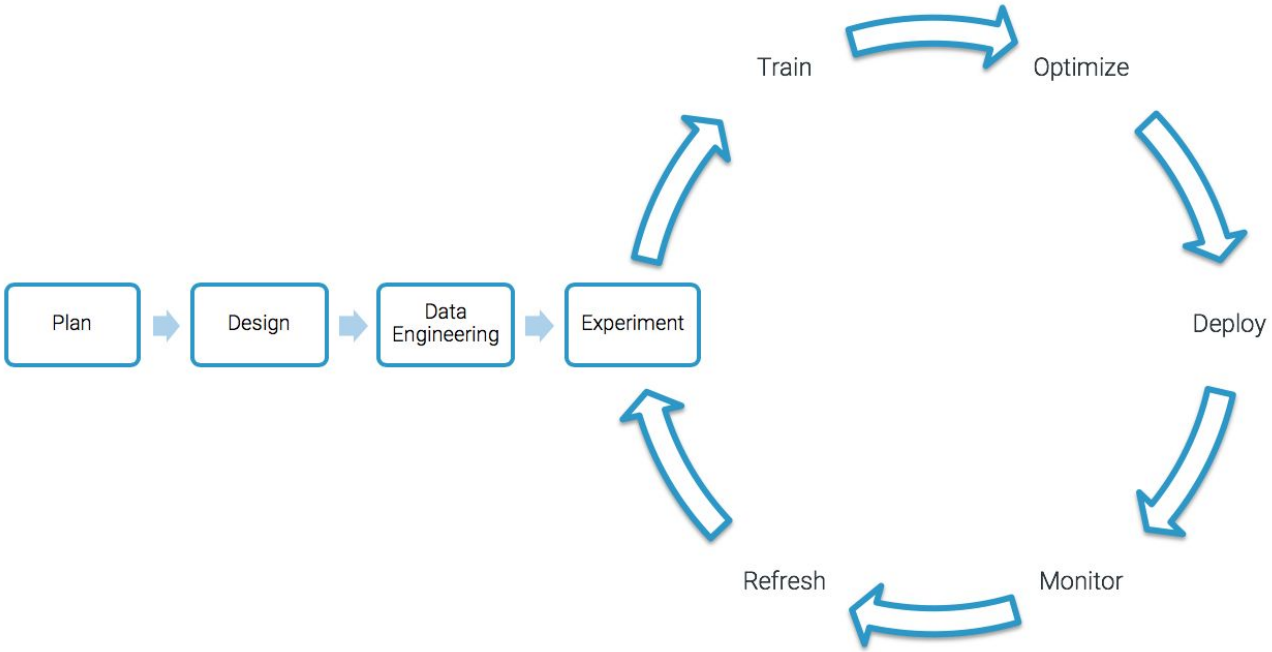
MODEL DEPLOYMENT PATTERNS

Knowing how business metrics will be improved help guide deployment options



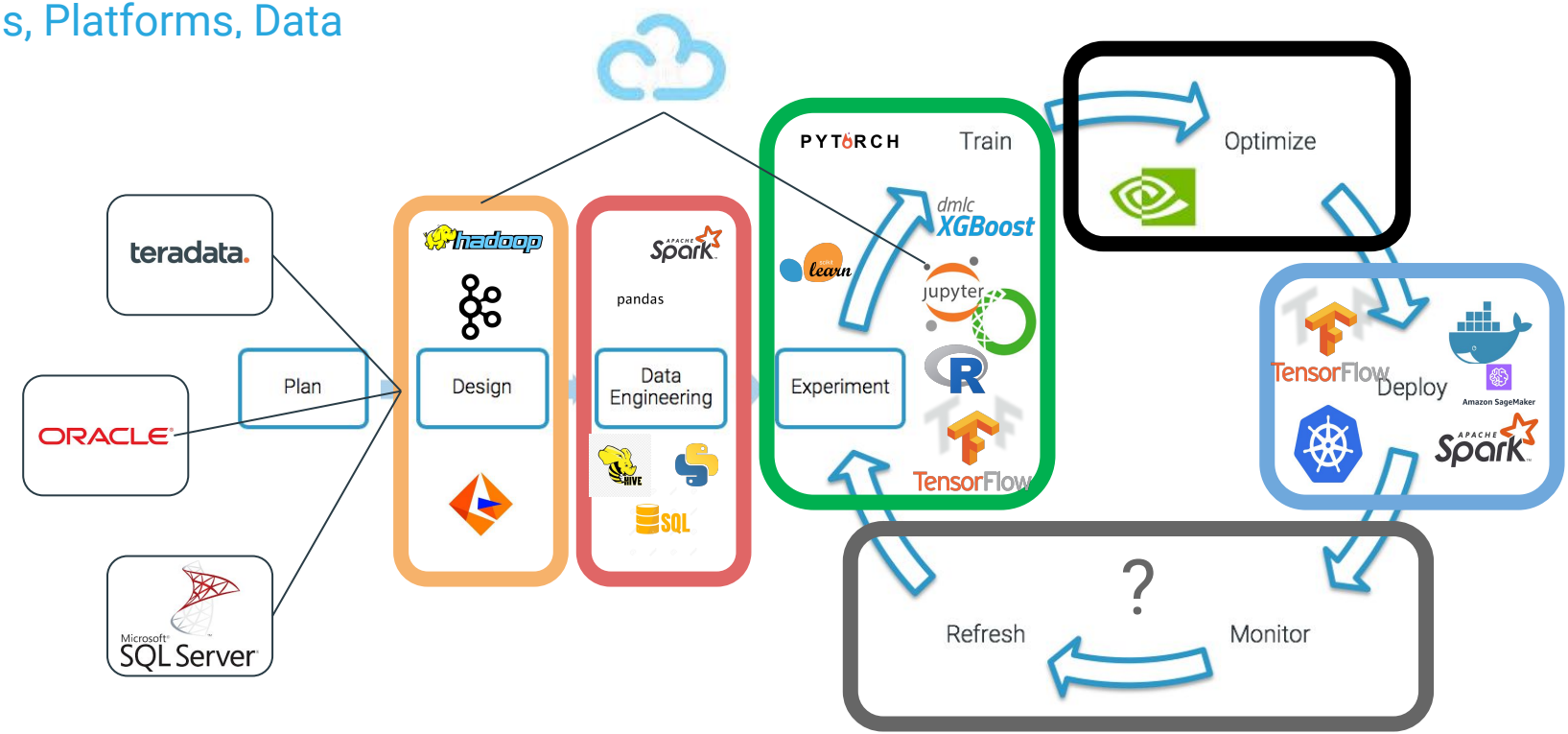
SAMPLE DATA SCIENCE / ML WORKFLOW

From Data Exploration to Action



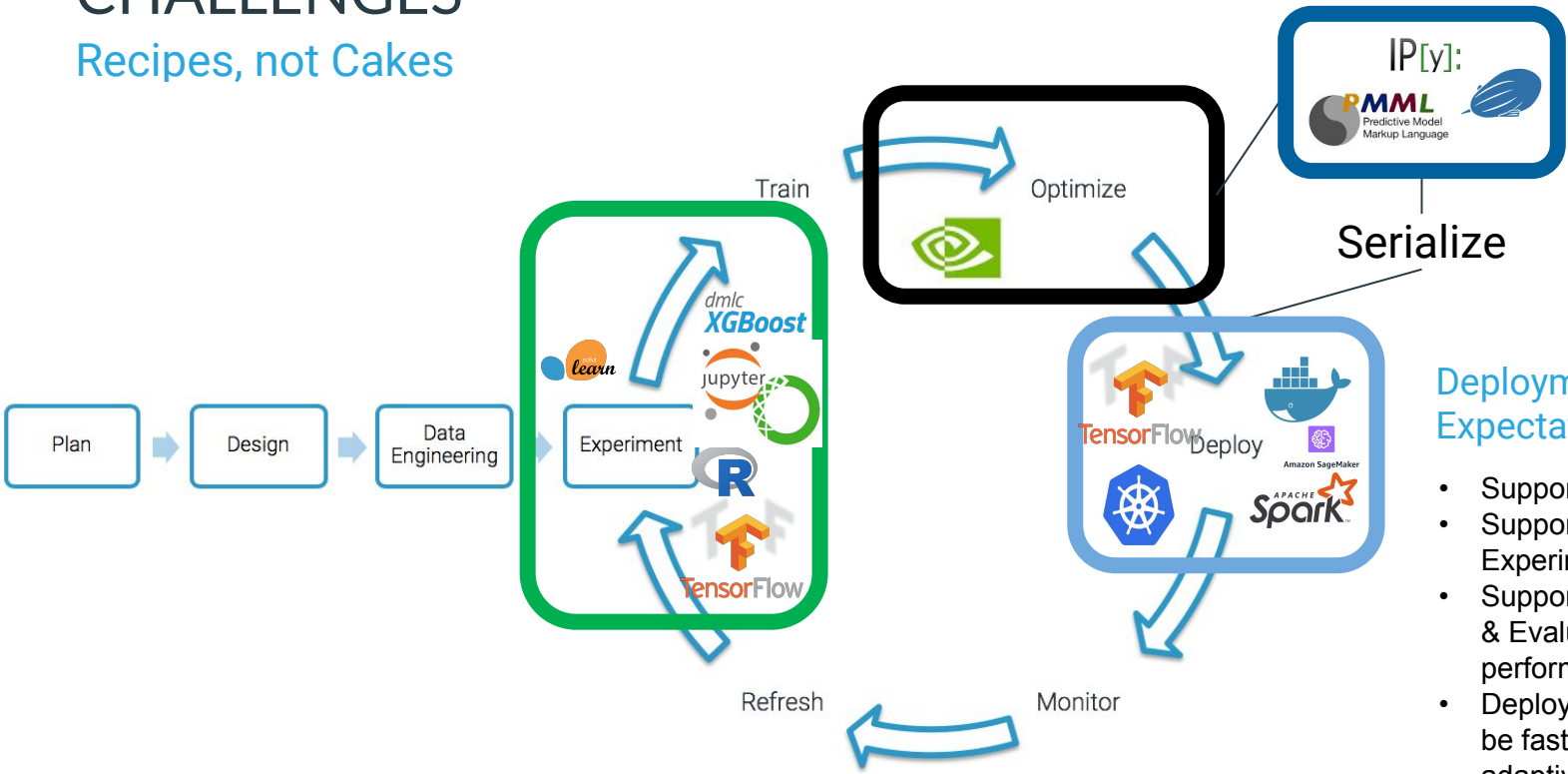
CHALLENGES

Tools, Platforms, Data



CHALLENGES

Recipes, not Cakes



Deployment Expectations

- Support A/B testing
- Support Experiments
- Support measuring & Evaluating model performance
- Deployment should be fast and adaptive to business needs

SUMMARY OF CHALLENGES

- Access

For sensitive data, secure clusters are difficult to access. No shared security

- Flexibility

IT typically doesn't want random packages installed on a secure cluster.

- Tools

Popular open source tools don't easily connect to these environments, or always support Hadoop data formats. Nothing supports full workflow

- Scale

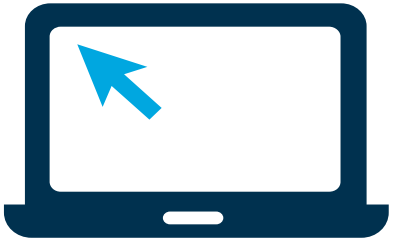
Laptops rarely have capacity for medium, let alone big data. This leads to a lot of sampling.

- Parallelism

Popular frameworks don't easily parallelize on a cluster. Typically code has to get rewritten for production.

- Security

Data being pulled into laptops



- Developer Experience

Notebooks, while awesome, don't easily support virtual environment and dependency management, especially for teams.

- Collaboration

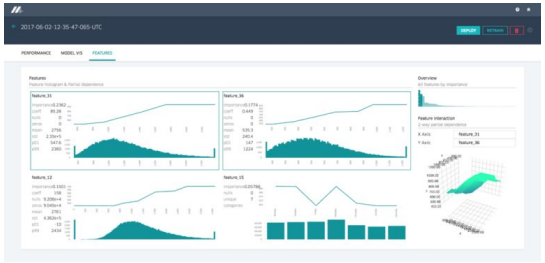
No easy way to share code between teams

- Deployment

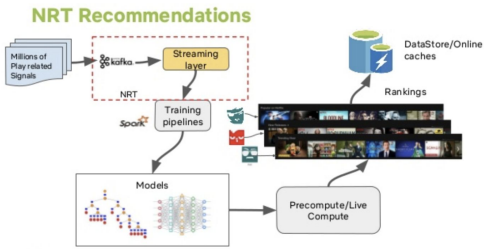
Notebooks are also challenging to "put into production."

MACHINE LEARNING AT UBER, NETFLIX, AND FACEBOOK

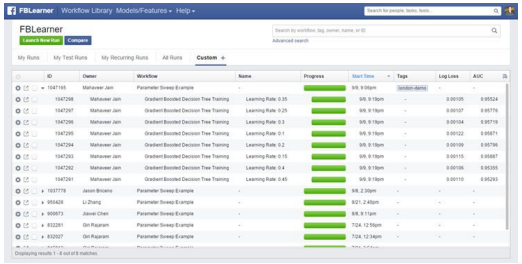
Industrialized AI requires new supporting tools and platforms



Uber
Michelangelo

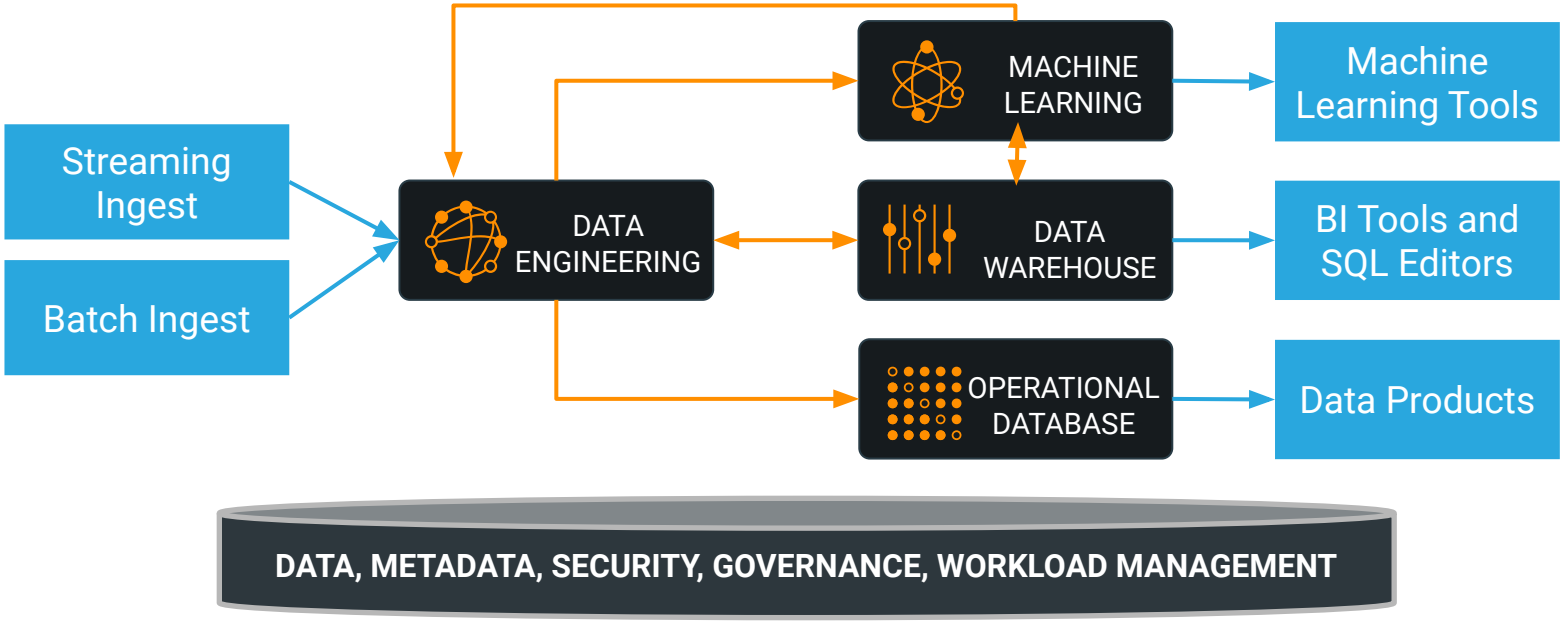


Netflix
Recommendation
Platform



Facebook
FBLearner

ML AT SCALE REQUIRES A UNIFIED DATA STRATEGY



EVERYONE HAS AN OPINION

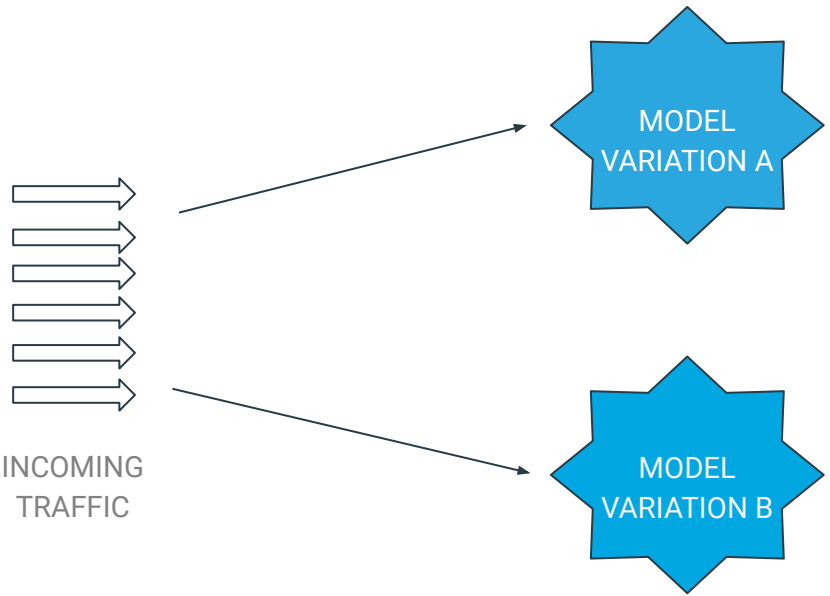
- Should enable collaboration and code reuse (git integration)
- Should support open-source frameworks and libraries
- Must handle dependencies and isolate dev environment for an individual session
- Can scale compute resources/up down when needed
- **Doesn't require you to move data to use it!**

A/B TESTING & MULTIVARIATE TESTING FOR THE MODEL

Is the best trained model indeed the best model, or does a different model perform better on new, unseen data?

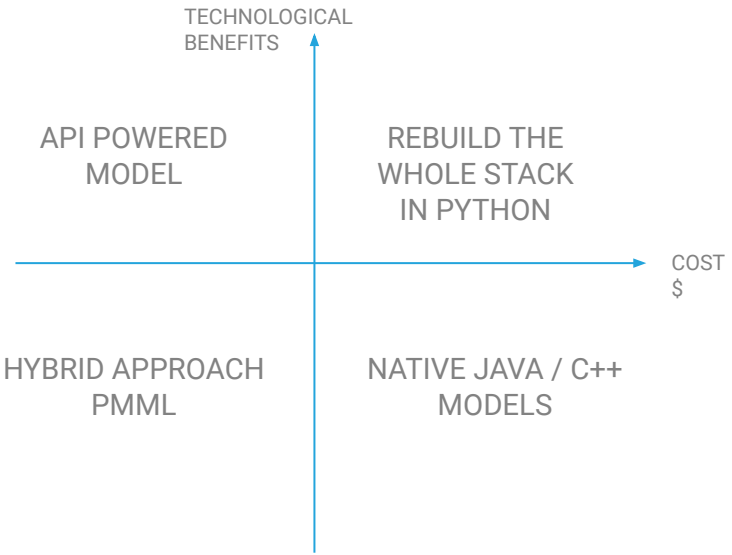
Data scientists need ...

- A framework to identify the best performers among a competing set of models
- To evaluate models which can maximize business KPIs
- Track specified model metrics, performance, and model artifacts
- Inspect & compare deployed models



MODEL DEPLOYMENT APPROACH : TECHNOLOGICAL VS COST BENEFITS

DIFFERENT MODEL DEPLOYMENT FORMATS



HYBRID APPROACH:

- Compatibility across multiple tools
- Non Agile
- Not flexible in terms of deployment

NATIVE JAVA/C++ MODEL

- Faster
- Limitation of Available Algo/DS Libraries

PYTHON STACK

- Python ML files are big due to dependencies
- Unit testing is tricky

API POWERED MODEL:

- Agile
- Scalable
- Can be used by both backend & fronted
- Faster

MONITORING STATS

SCHEDULE & MONITOR

Production ML needs...

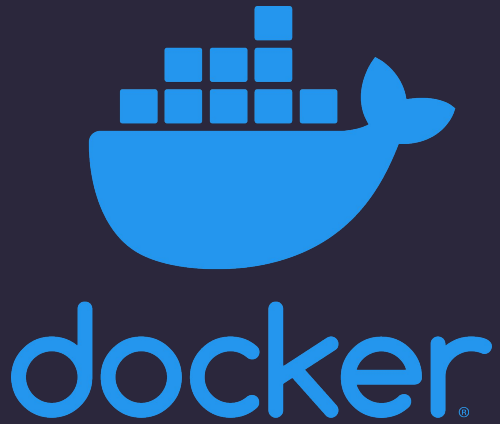
- A Monitoring mechanism that is model-agnostic
- Instrumentation of both the data flow in and the model performance metrics out
- To Collect Performance Metrics (e.g. accuracy, RMSE, Mean Absolute Error (MAE))

Run	Script	Arguments	Kernel	Comment	Submitter	Created At	numTrees	maxDepth	auroc	Status	Duration	Actions
479	dsfortelco_sklearn_exp.py	25 20	python3		training01	2/19/19 6:30 AM	25	20	0.862676056...	Success	1 mins	
478	dsfortelco_sklearn_exp.py	15 25	python3		training01	2/19/19 6:27 AM	15	25	0.853176855...	Success	1 mins	
477	dsfortelco_sklearn_exp.py	40 20	python3	First Random Forrest Experiment	training01	2/19/19 6:22 AM	40	20	0.836584394...	Success	1 mins	

Enabling Technologies

Let’s learn about the stuff that surrounds the ML code.

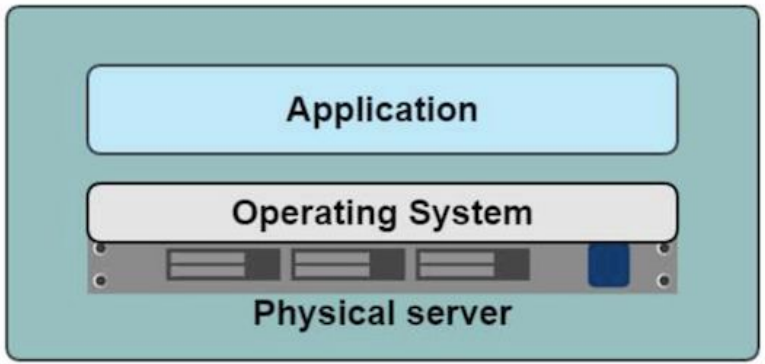
Introducing Containers



A History Lesson

In the Dark Ages

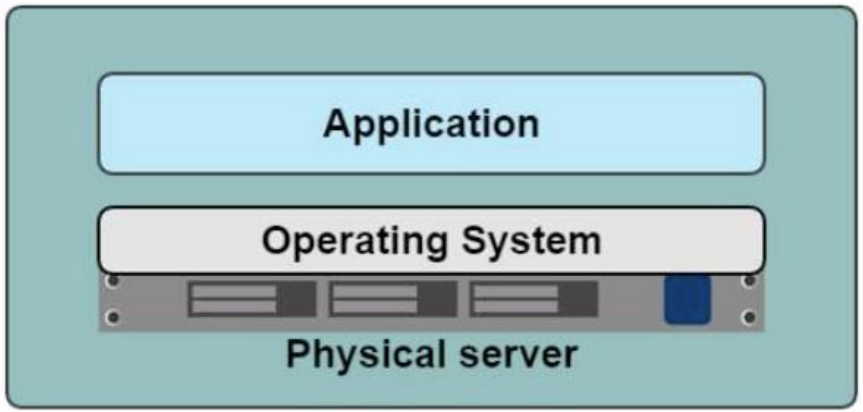
One application on one physical server



Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Historical limitations of application deployment

- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
- Vendor lock in

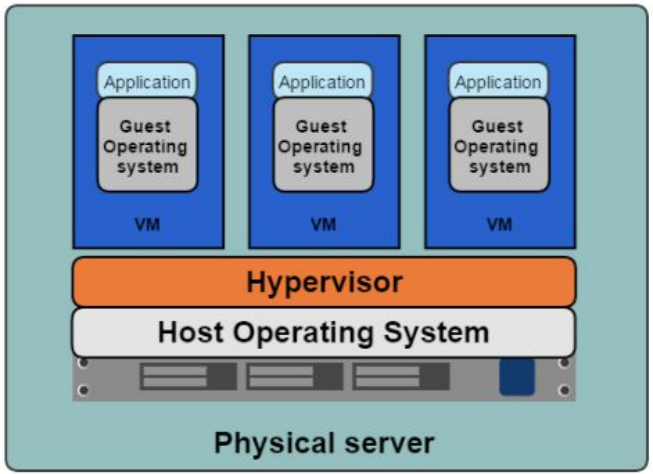


Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

A History Lesson

Hypervisor-based Virtualization

- One physical server can contain multiple applications
- Each application runs in a virtual machine (VM)



Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Benefits of VMs

- Better resource pooling
  - One physical machine divided into multiple virtual machines
- Easier to scale
- VMs in the cloud
  - Rapid elasticity
  - Pay as you go model

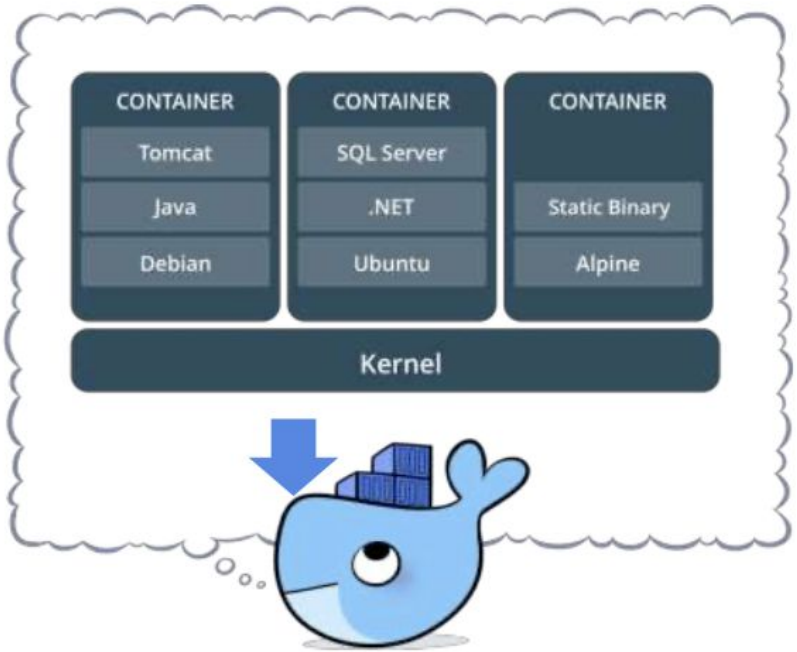
Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Limitations of VMs

- Each VM stills requires
  - CPU allocation
  - Storage
  - RAM
  - An entire guest operating system
- The more VMs you run, the more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed

Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

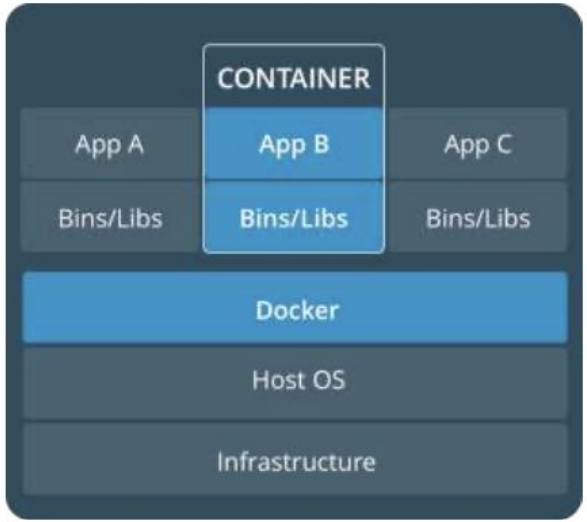
What is a container?



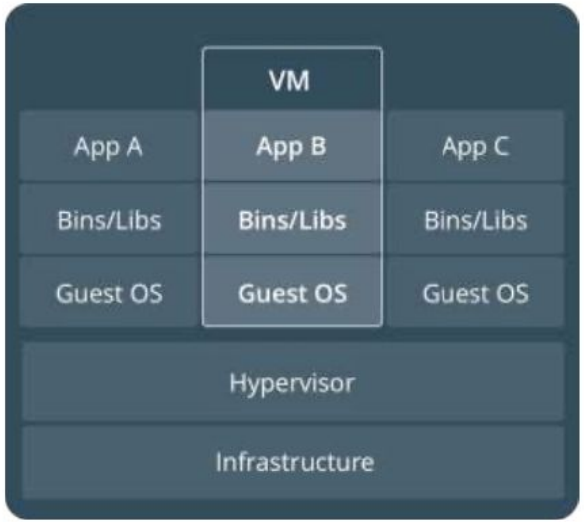
- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux and Windows Server

Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Comparing Containers and VMs



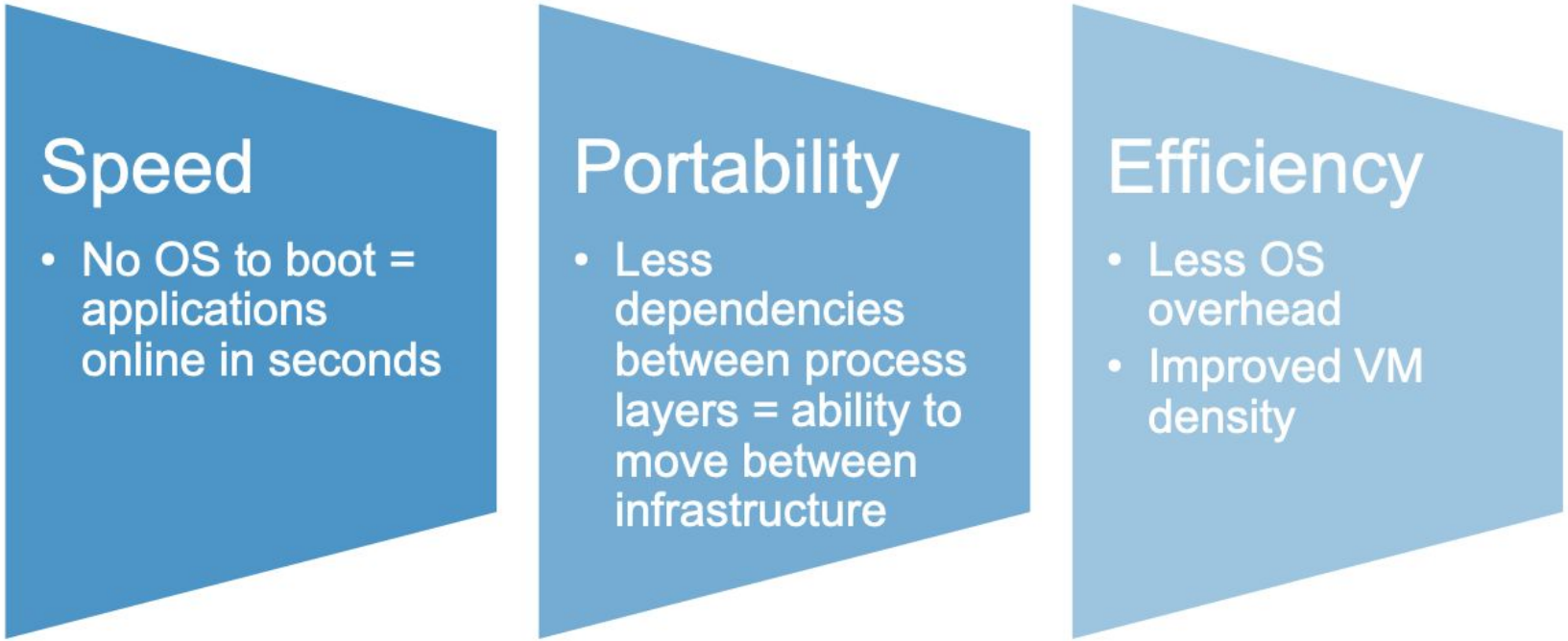
Containers are an app level construct



VMs are an infrastructure level construct to turn one machine into many servers

Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Key Benefits of Docker Containers



Source: https://www.slideshare.net/Docker/introduction-to-docker-2017/18

Introducing Containers

What Does “Kubernetes” Mean?

Greek for “pilot” or
“Helmsman of a ship”

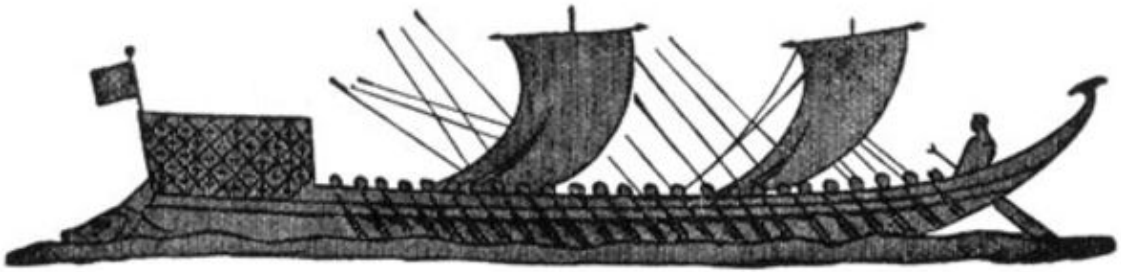


Image Source

Source Deck

What is Kubernetes?

- Project that was spun out of Google as an open source container orchestration platform.
- Built from the lessons learned in the experiences of developing and running Google's Borg and Omega.
- Designed from the ground-up as a **loosely coupled** collection of components centered around deploying, maintaining and scaling workloads.

Source Deck

What Does Kubernetes do?

- Known as the **linux kernel of distributed systems**.
- **Abstracts away the underlying hardware** of the nodes and provides a uniform interface for workloads to be both deployed and consume the shared pool of resources.
- Works as an engine for resolving state by converging actual and the **desired state** of the system.

Source Deck

Decouples Infrastructure and Scaling

- **All services** within Kubernetes are natively Load Balanced.
- Can scale up and down dynamically.
- Used both to enable self-healing and seamless upgrading or rollback of applications.

Source Deck

Self Healing

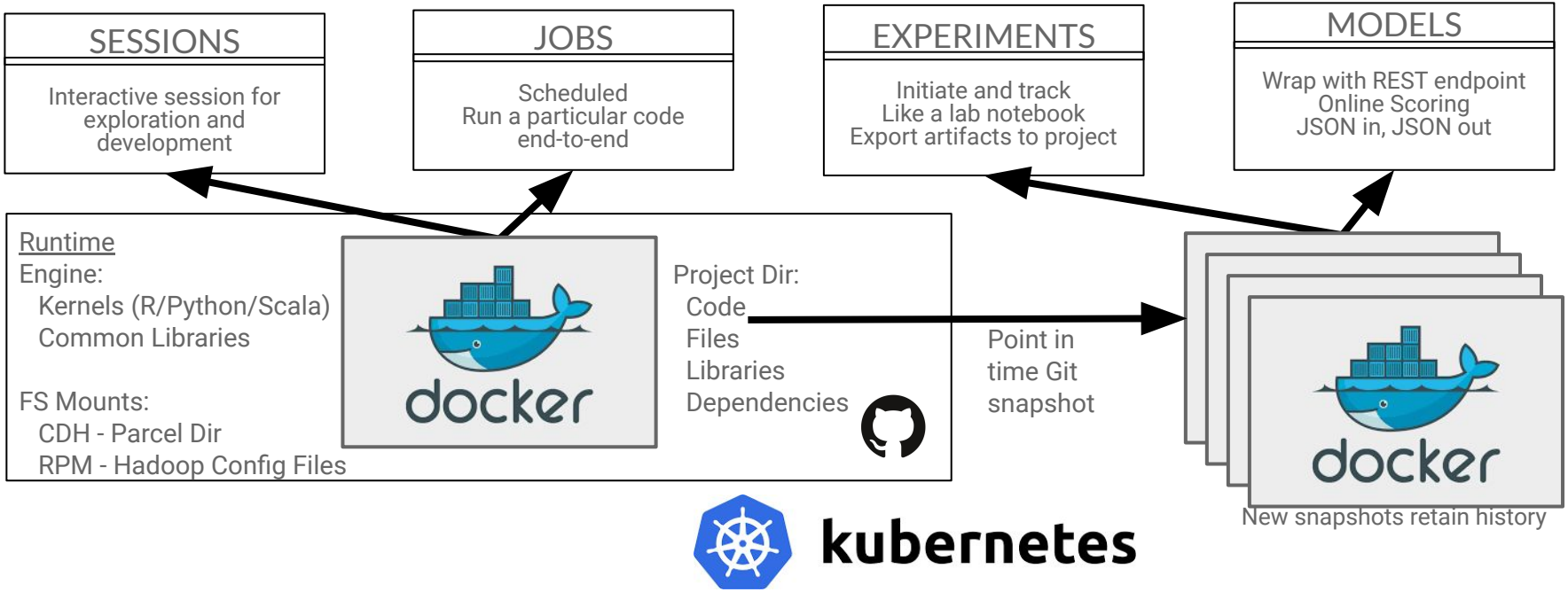
Kubernetes will **ALWAYS** try and steer the cluster to its desired state.

- **Me:** “I want 3 healthy instances of redis to always be running.”
- **Kubernetes:** “Okay, I’ll ensure there are always 3 instances up and running.”
- **Kubernetes:** “Oh look, one has died. I’m going to attempt to spin up a new one.”

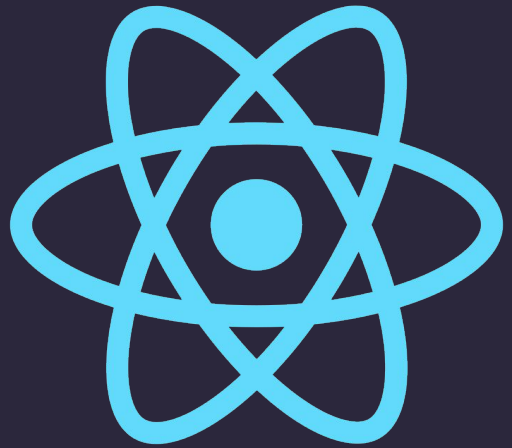
Source Deck

ACCELERATING MACHINE LEARNING

Lego Block for ML: Like a containerized edge node



Web Frameworks



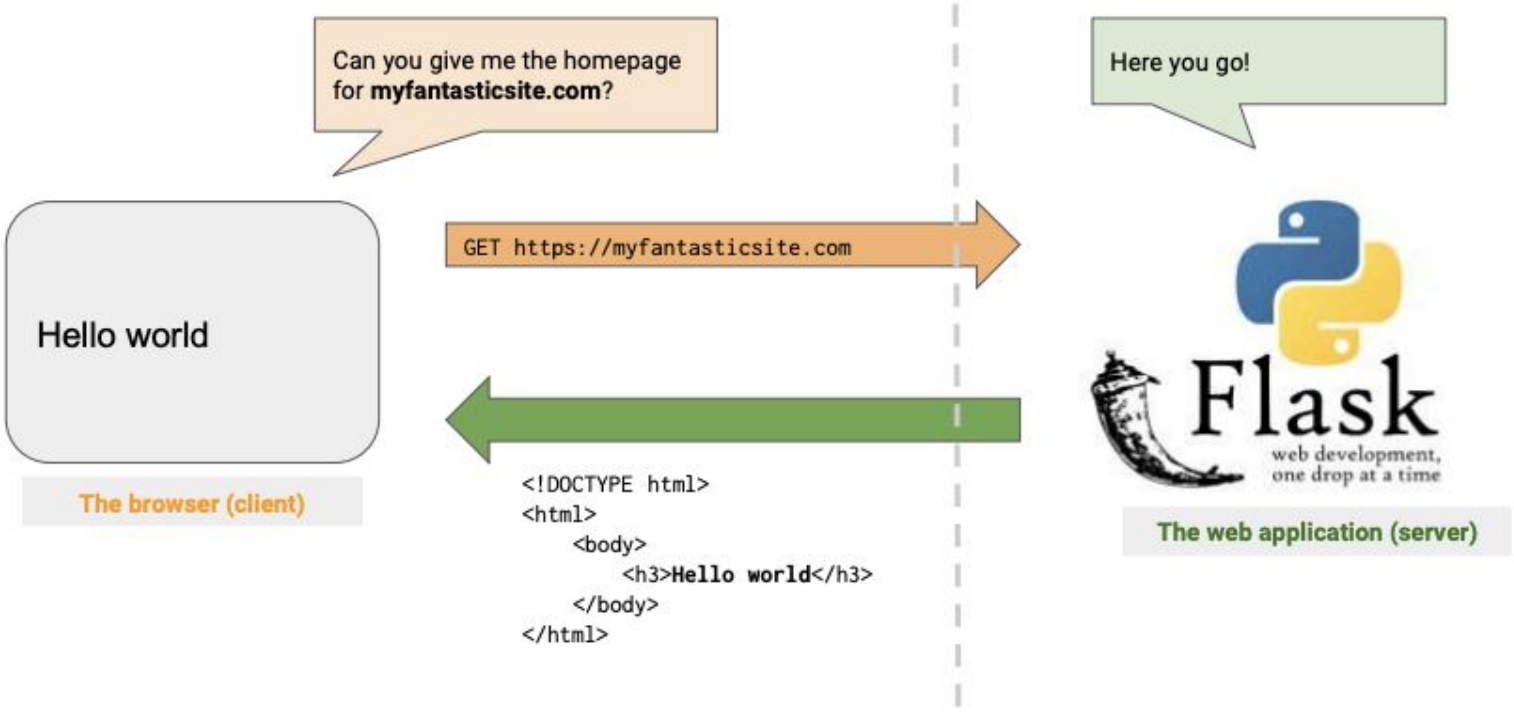
What is Flask?

1. A micro framework written in Python for the web--used to create websites and apis

2. Has a simple core but is also highly extensible

Flask

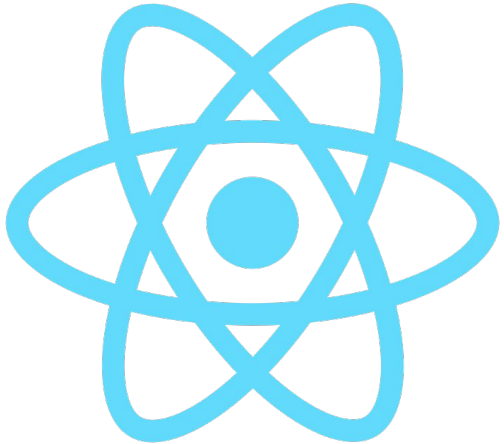
Framework for building web applications in Python



Source: https://www.slideshare.net/paragmujumdar/pythonflask-presentation

What is React?

1. A front-end javascript library developed at Facebook.
2. Usually used as a declarative and flexible way to build User Interfaces
3. Allows devs to build complex UIs from small and independent code blocks called “components”



Part II - Coming Soon

Let’s put it all together - scalable cloud-based ML applications