

Challenges of Operationalizing Data Science in Production

Machine Learning Operations Meet-Up #1
July 4

Agenda

Real-world Data Science Challenges

- Section 1: Business Aspects
- Section 2: Technology and Operational Aspects
- Demo

Speakers

Santanu Dey



[@Santanu_Dey](https://twitter.com/Santanu_Dey)



santanud@iguazio.com

Santanu Dey is a Solutions Architect helping customers with their Digital Transformations journey, solutions involving Cloud, Analytics, Microservices etc.

Over 18 years of proven track record of designing and operationalizing high-volume, mission critical, distributed systems.

Rasmi Mohapatra



<https://www.linkedin.com/in/rasmi-m-428b3a46/>

Rasmi's primary background is in product and technology management. His secondary background covers business transformation and operations functions across enterprise and startup environment. He currently is a Product Owner at Experian's APAC innovation Hub - XLabs.

Section 1: Business Aspects

What is Data Science?

DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC
DATA SCIENCE IS NOT MAGIC



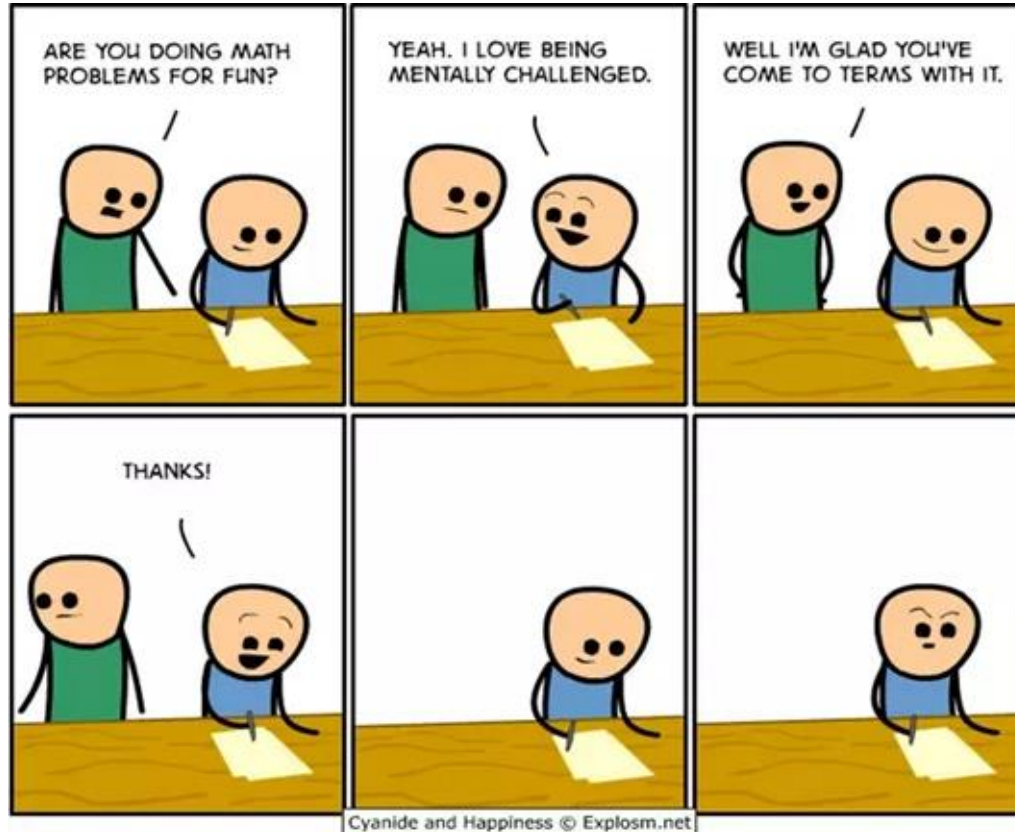
1. Domain Knowledge!

© 2013 Ted Goff

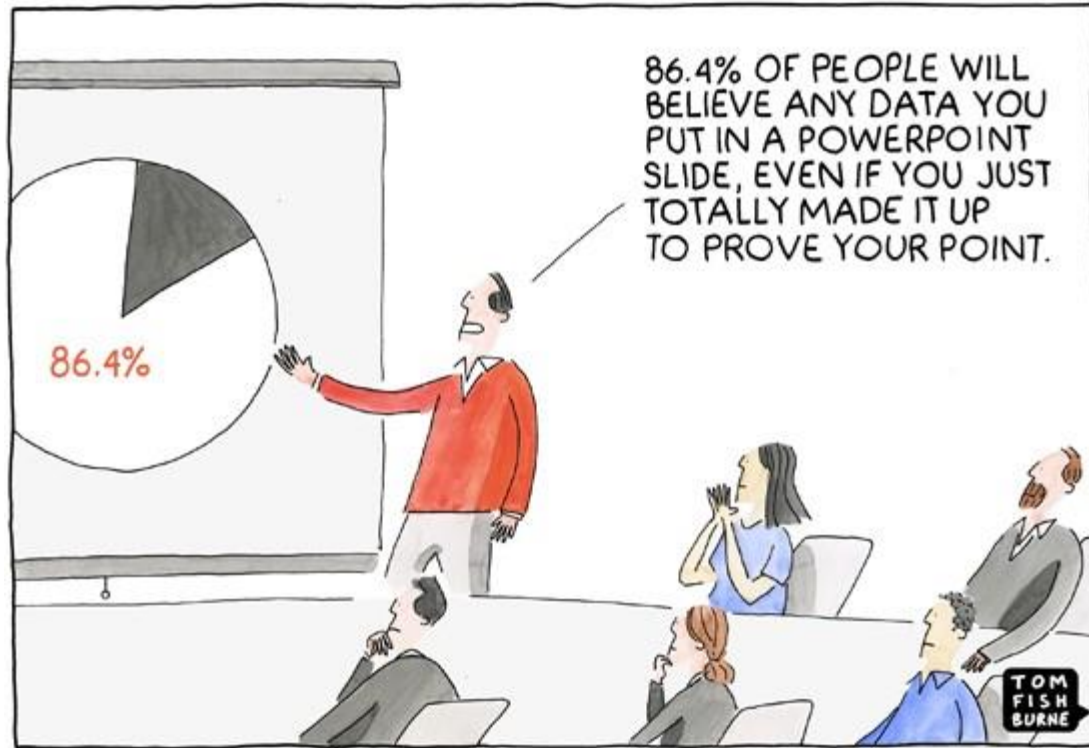


**“You can’t keep adjusting the data
to prove that you would be the best
Valentine’s date for Scarlett Johansson.”**

2. Actually understanding math!

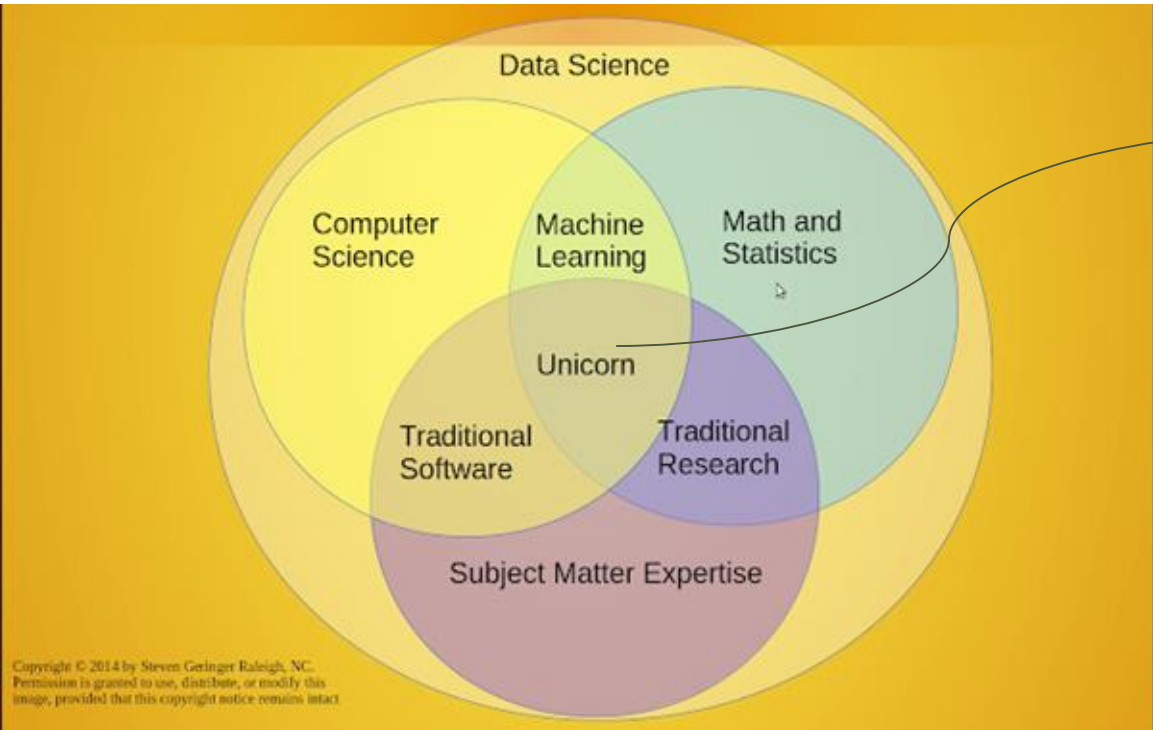


3. Visualisation!



Business Issue #0

Don't stop until you find your data scientist!



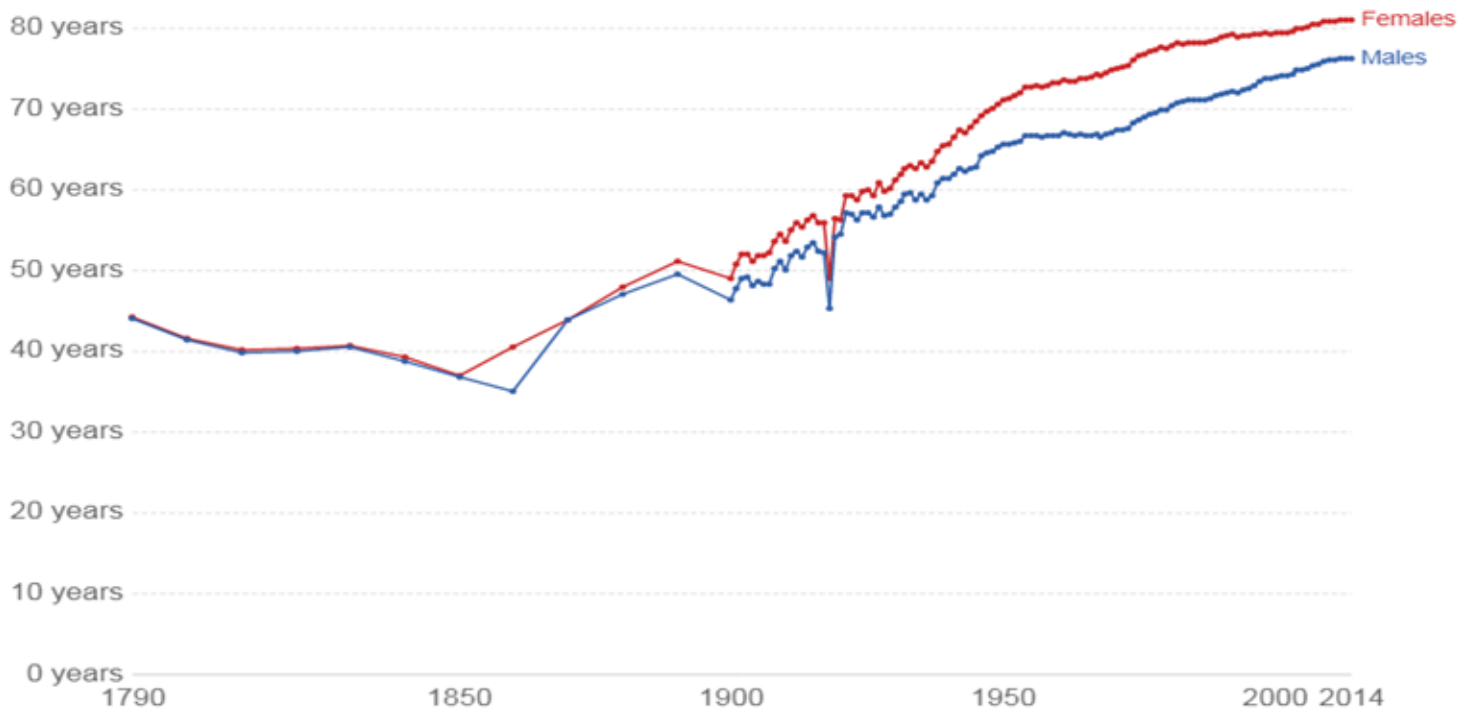
Business Issue #1

Giving into hype - Underestimating “small” data

Life expectancy at birth, United States

Estimates are based on period life tables.

Our World
in Data



Source: Human Mortality Database (2018) and others

CC BY-SA

Business Issue #2

Being unaware of regulation, compliance



Business Issue #3

Can't explain it right to right people? You probably losing your hard work!!!

Why did the tree fall down?



"I agree."

"It was the wind. It is the simpler explanation."



Two Explanations

1. The wind knocked down the tree.
2. Two meteorites. One hit the tree and knocked it down. Then it hit the other meteorite, thus obliterating evidence of its existence.

When there are two explanations, choose the simpler one

Business Issue #4

Accessing clean data



Business Challenges Summary

- # 0- Focus on your domain and business requirements
- # 1- Business cares about outcomes - not Big Data or Small
- # 2- Be aware of regulatory implications of Data
- # 3- Focus on Explain-ability & “Good enough” accuracy
- # 4- Making the dataset usable for Data Science

Section 2: Technology Aspects

Data Science Journey



Data Scientist



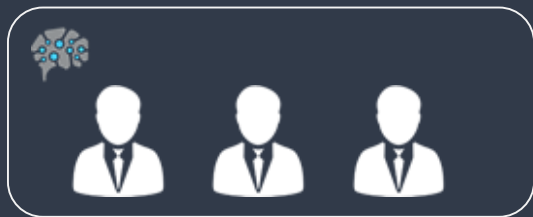
Workstation

Sample data set
at rest



Models

What does it take to Productionize AI Apps



Data Science Team



Large Data Sets



Productionize Smart Applications

Where are the key challenges?

Landmines!



Data Engineer



Data Scientist



ML-Ops



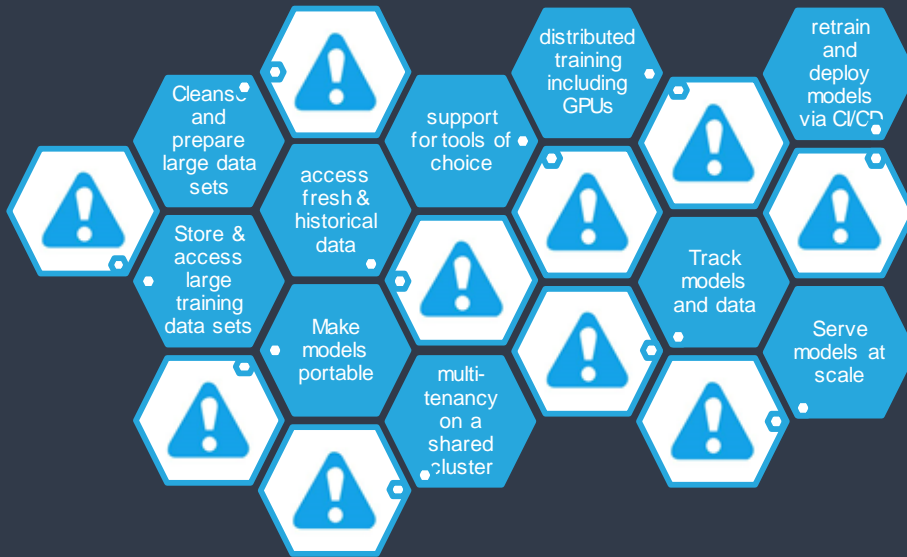
Large Data Sets



Platform & Tooling

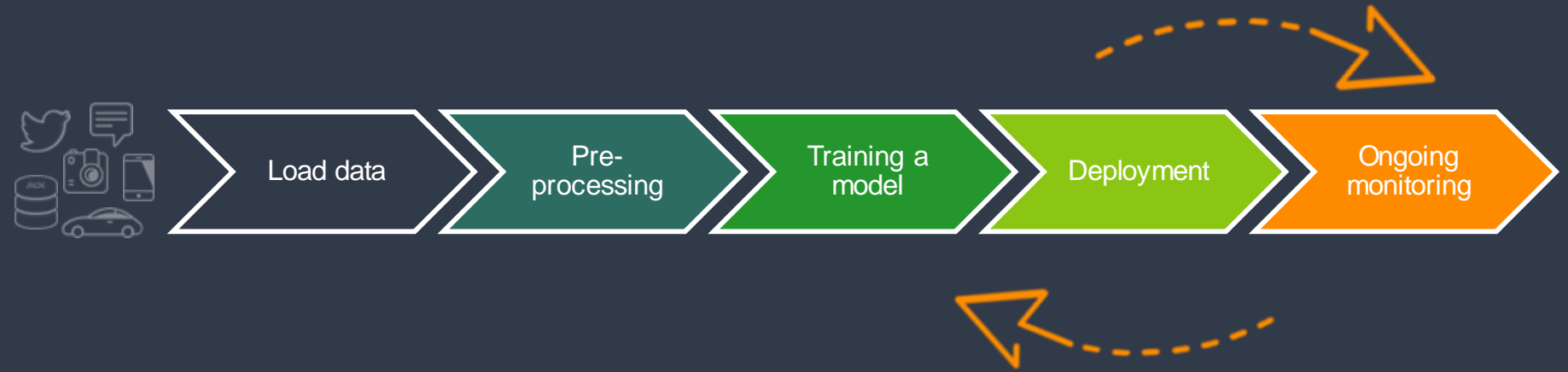


Infra & SLA aspects



Productionize Smart Applications

Data Science Lifecycle



DS Lifecycle & IT Implications

I/O Intensive

- High Volume
- Legacy Touchpoints
- Often long wait
- Multiple silos

Compute Driven

- CPU
- GPU
- Multi-node
- Scheduling / Sharing
- Elasticity

SLA Driven

- Latency
- Automation
- Monitoring
- Accuracy



Load data

Pre-processing

Training a model

Deployment

Ongoing monitoring



DS Lifecycle & IT Implications

Customer
Driven

I/O Intensive


- High Volume
- Legacy Touchpoints
- Often long wait
- Multiple silos

Compute Driven

- CPU
- GPU
- Multi-node
- Scheduling / Sharing
- Elasticity

SLA Driven

- Latency
- Automation
- Monitoring
- Accuracy



Load data

Pre-
processing

Training a
model

Deployment

Ongoing
monitoring



Data Science Lifecycle & Multiple Roles



Data Engineer



Data Scientist



ML-Ops



Load data

Pre-processing

Training a model

Deployment

Ongoing monitoring



Friction Across Data Science Lifecycle & Roles



Data Engineer



Data Scientist



ML-Ops



Load data

Pre-processing

Training a model

Deployment

Ongoing monitoring

- ✓ Get fresh and relevant data from actual system
- ✗ Dependency on IT for data access and data prep
- ✗ Old data, unclean, wrong granularity

- ✓ Self-service data prep by Data Scientist
- ✗ Am I forced to use specific services / tools?
- ✗ data prep code not scaling for large dataset

- ✓ Support Jupyter and ALL my DS tools
- ✗ Rewrite the training code to scale the training
- ✗ Experiments are not repeatable
- ✗ IT blocks my tools

- ✓ Portable model artifact
- ✗ Too much CI/CD work
- ✗ Data input is different for inference phase
- ✗ Difficult to version/track/rollback models

- ✓ scale and recover elastically
- ✗ Security / Scalability/ Monitoring of deployed models
- ✗ No feedback loop

DIY Approach

DIY Approach



AWS Kinesis



AWS
DynamoDB



AWS
Batch



AWS
Lambda



AWS S3



AWS EMR



Load data

Pre-
processing

Training a
model

Deployment

Ongoing
monitoring

DIY Approach



AWS Kinesis



AWS
DynamoDB



AWS
Batch



AWS SageMaker
Notebooks



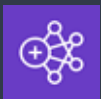
ML Training
Cluster



AWS
Lambda



AWS S3



AWS EMR



AWS S3



AWS SageMaker
Models



Load data

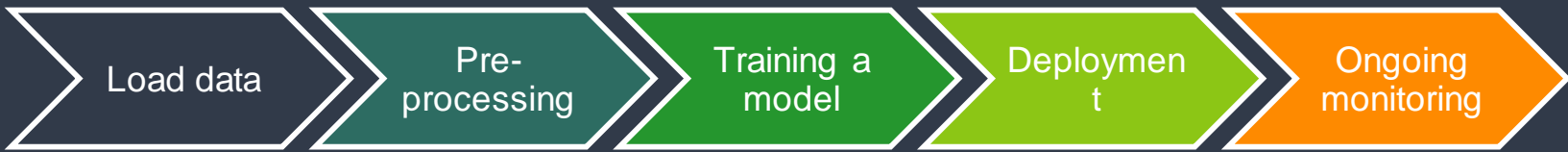
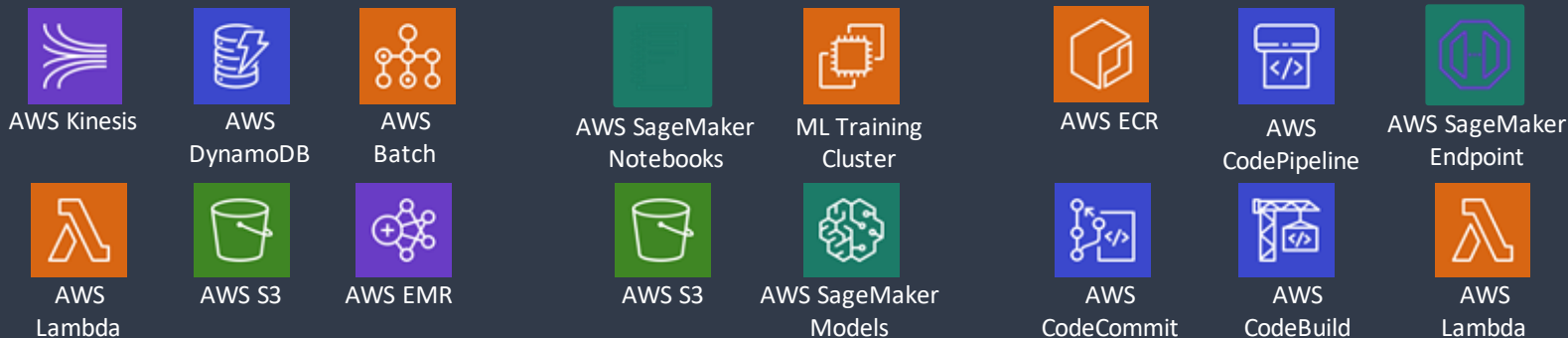
Pre-
processing

Training a
model

Deployment

Ongoing
monitoring

DIY Approach



DIY Approach



Data Engineer



Data Scientist



ML-Ops



AWS Kinesis



AWS
DynamoDB



AWS
Batch



AWS
Lambda



AWS S3



AWS EMR



AWS SageMaker
Notebooks



ML Training
Cluster



AWS S3



AWS SageMaker
Models



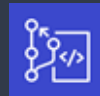
AWS ECR



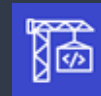
AWS
CodePipeline



AWS SageMaker
Endpoint



AWS
CodeCommit



AWS
CodeBuild



AWS
Lambda



Load data

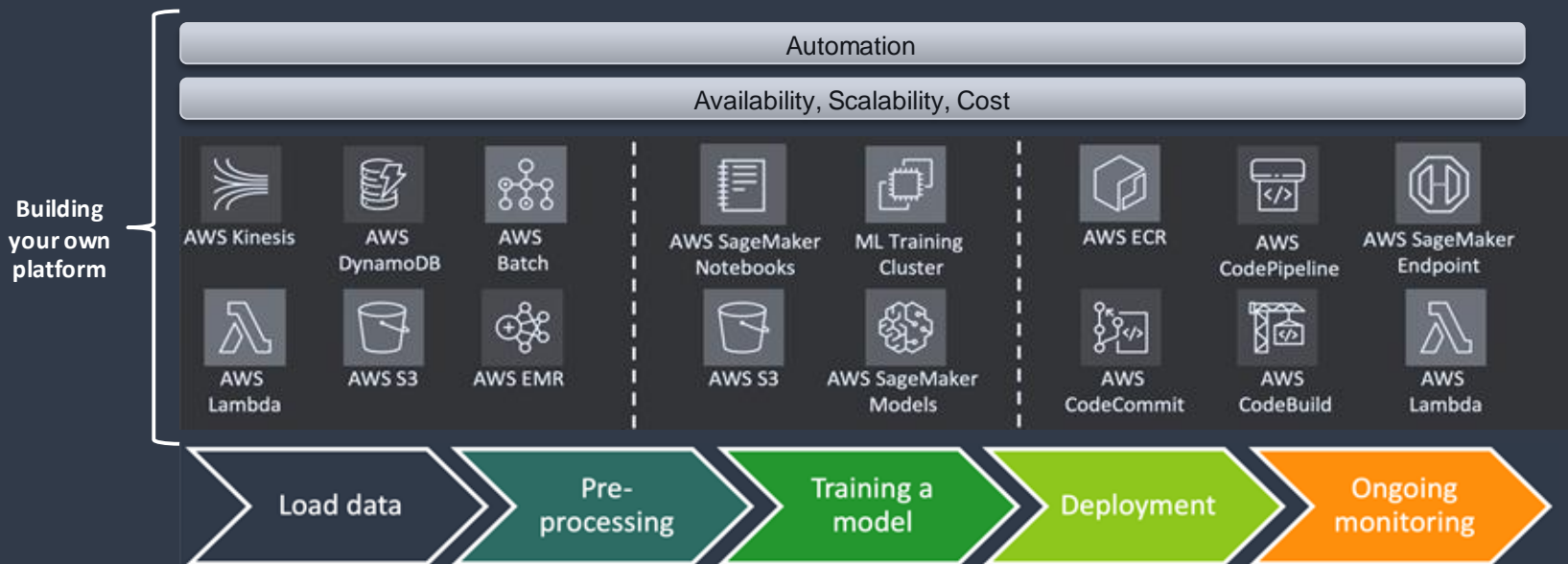
Pre-
processing

Training a
model

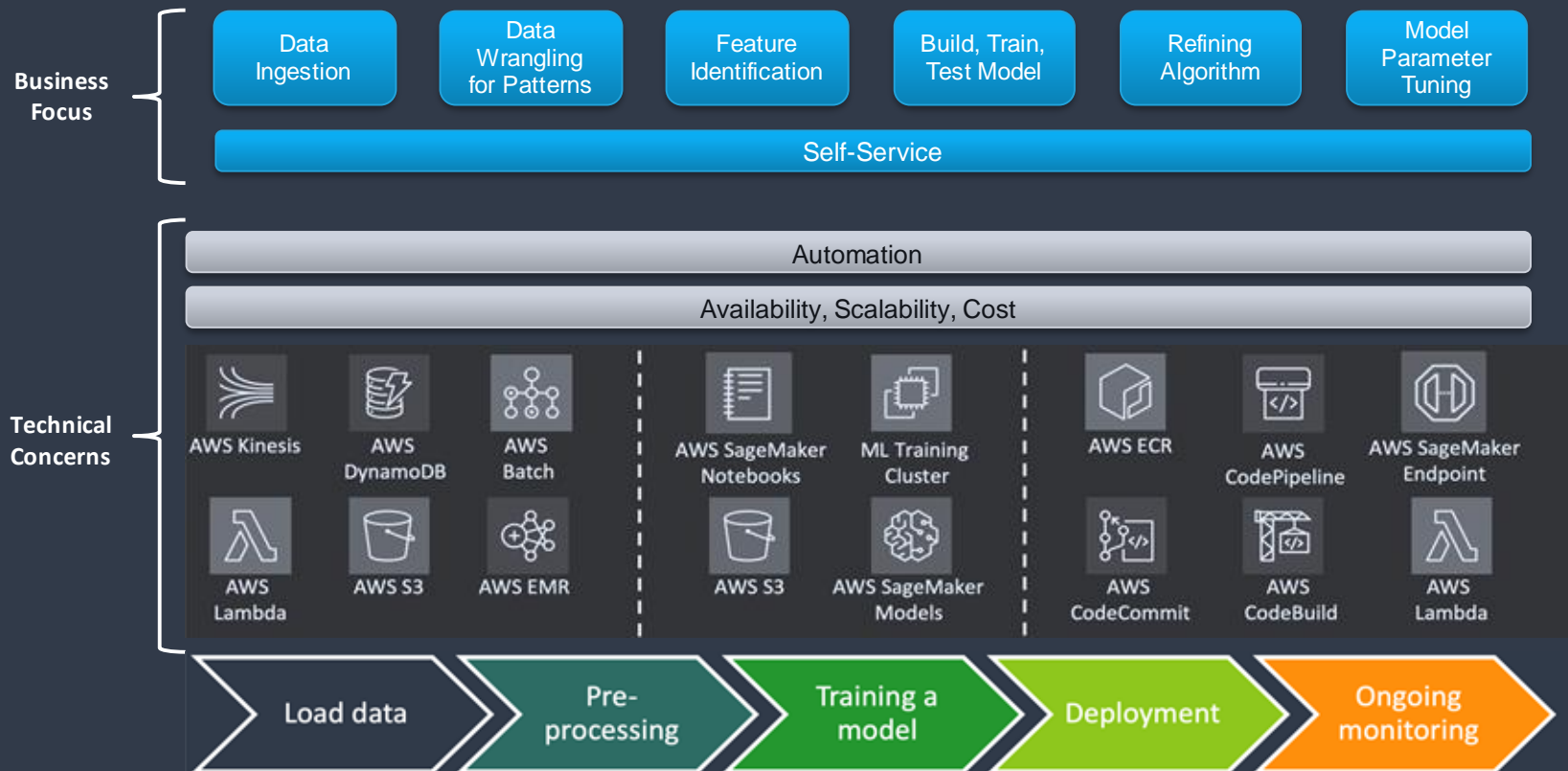
Deployment

Ongoing
monitoring

DIY Approach



DIY Approach



DIY Approach

Business Focus

Data Ingestion

Data Wrangling for Patterns

Feature Identification

Build, Train, Test Model

Refining Algorithm

Model Parameter Tuning

Self-Service

Focus on the business challenges as outlined in section 1

Automation

Availability, Scalability, Cost

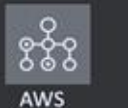
Technical Concerns



AWS Kinesis



AWS DynamoDB



AWS Batch



AWS SageMaker Notebooks



ML Training Cluster



AWS ECR



AWS CodePipeline



AWS SageMaker Endpoint



AWS Lambda



AWS S3



AWS EMR



AWS S3



AWS SageMaker Models



AWS CodeCommit



AWS CodeBuild



AWS Lambda

Load data

Pre-processing

Training a model

Deployment

Ongoing monitoring

Platform Approach

Platform Approach



Data Science Platform Services

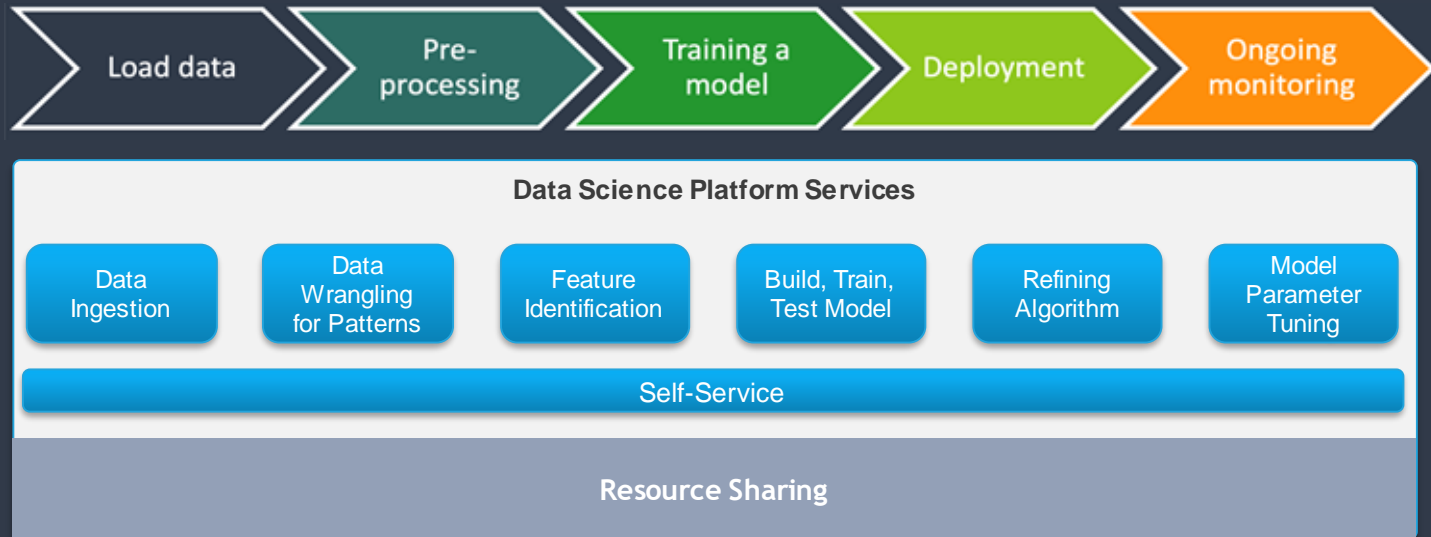
Platform Approach



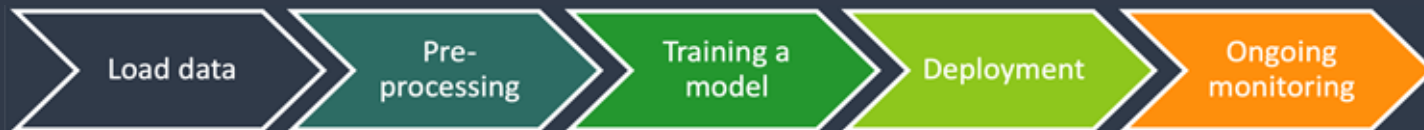
Data Science Platform Services



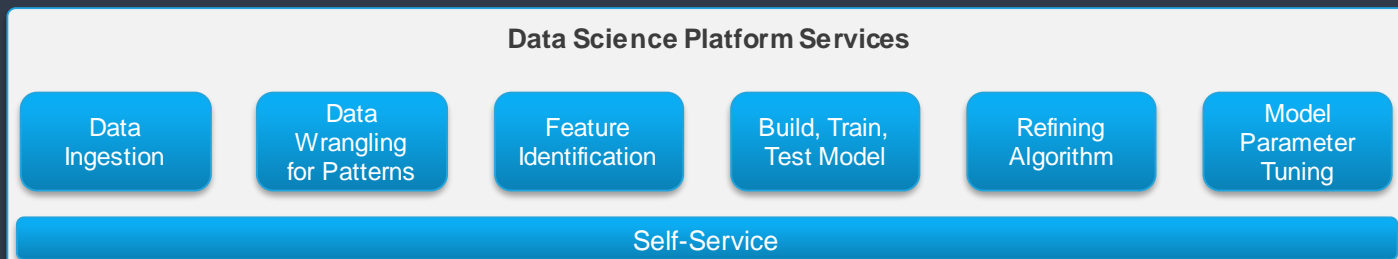
Platform Approach



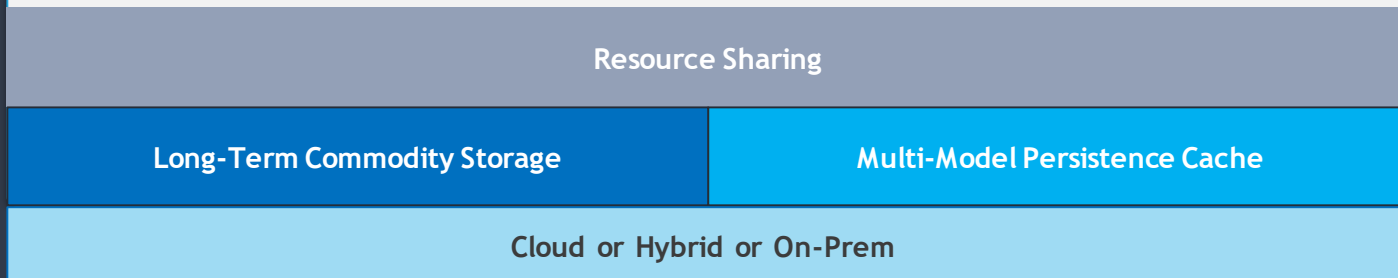
Platform Approach



Business Focus



A platform should hide the technical concerns



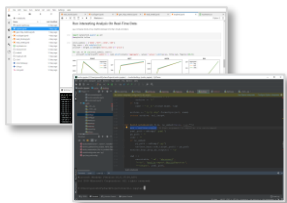
Demo

Summary

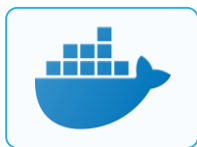
Code/Model Development Is Just The FIRST Step

Every piece of code, data science algorithm, or data processing task must be built for production

Develop/Experiment



Package



- Dependencies
- Parameters
- Run scripts
- Build

Scale-out



- Load-balance
- Data partitions
- Model distribution
- Hyper params

Tune



- Parallelism
- GPU support
- Query tuning
- Caching

Instrument



- Monitoring
- Logging
- Versioning
- Security

Automate



- CI/CD
- Workflows
- Rolling upgrades
- A/B testing

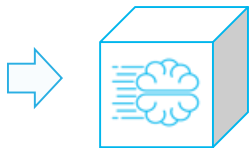
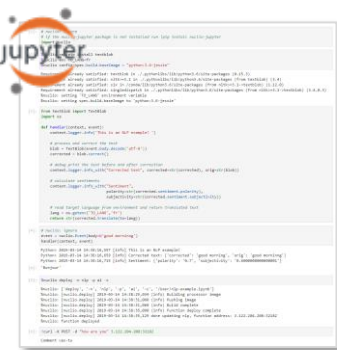


Weeks with one
data scientist

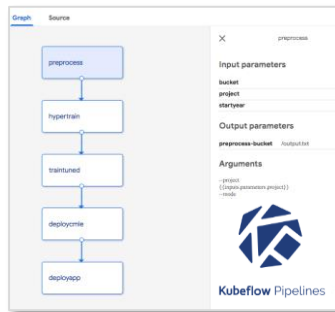


Months with a large team of developers,
scientists, data engineers and DevOps

Nuclio: Fast Serverless for Data Science & RT Analytics



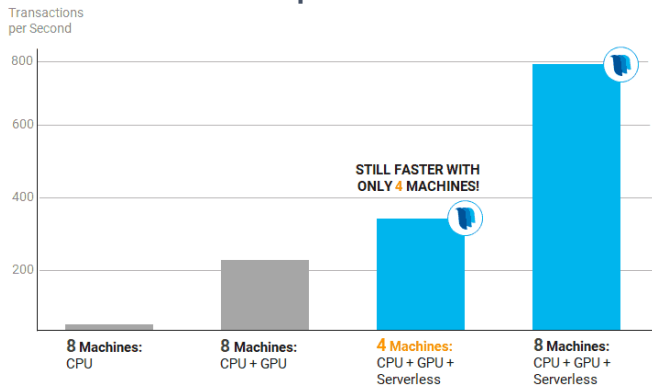
magic commands from notebook to function



Extending ML Pipelines from batch:

1. Parallel processing
2. Code build/deployment
3. Stream processing
4. API/Model Serving

High-performance IO and Computation + GPU Optimizations

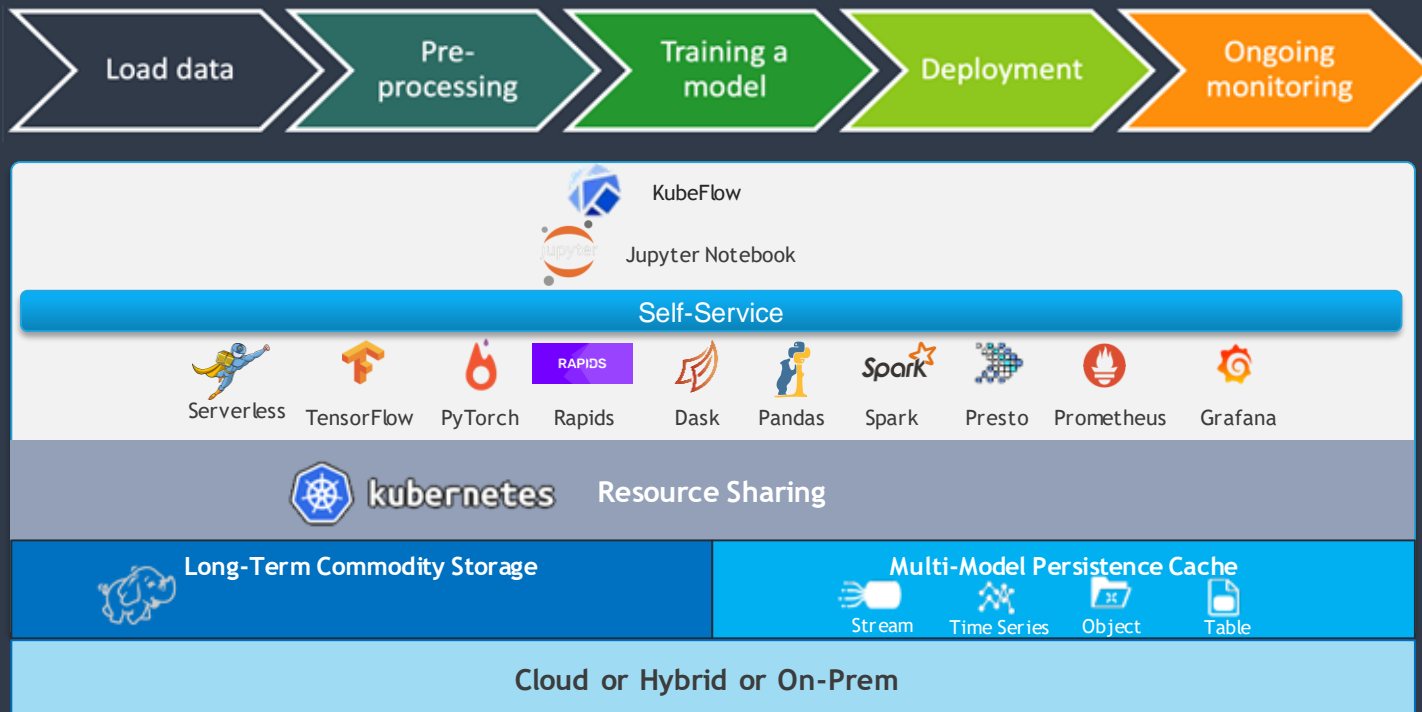


Code + DevOps Automation:

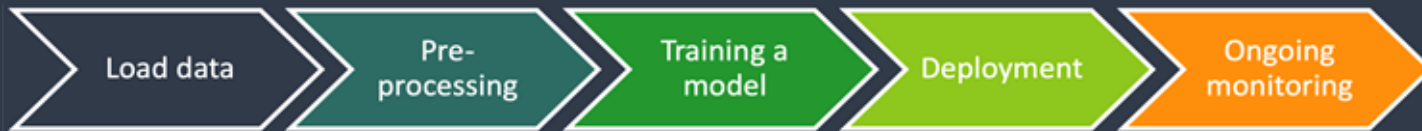
1. Auto-scaling (to zero)
2. Automated logging & monitoring
3. Security hardening
4. Auto-build and CI/CD
5. Workload mobility (cloud/edge/..)



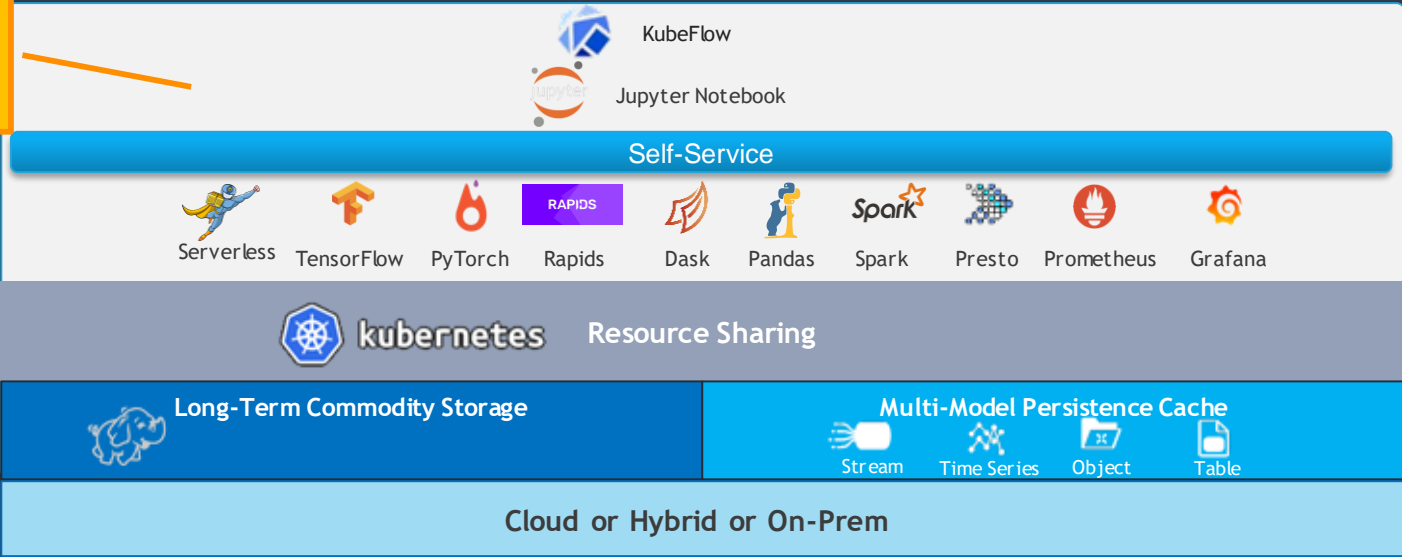
Platform Approach - Summary



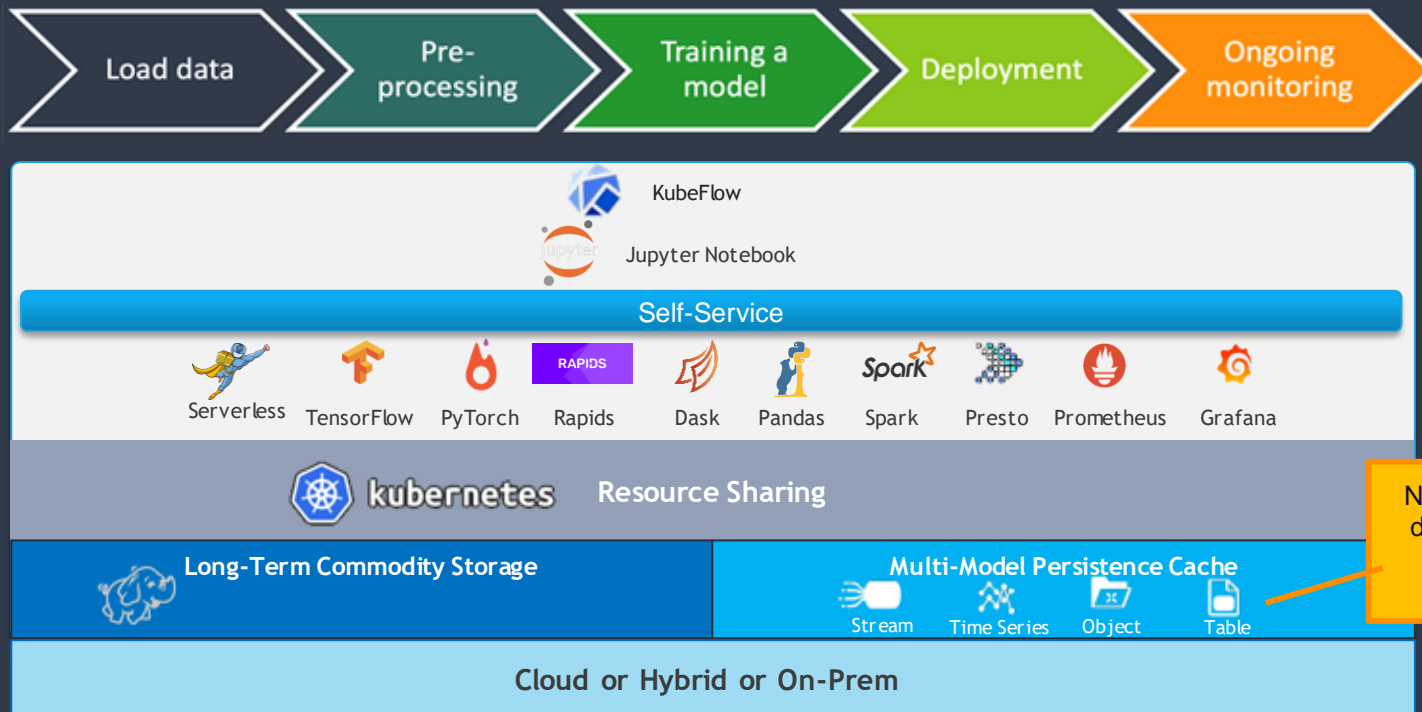
Platform Approach - Summary



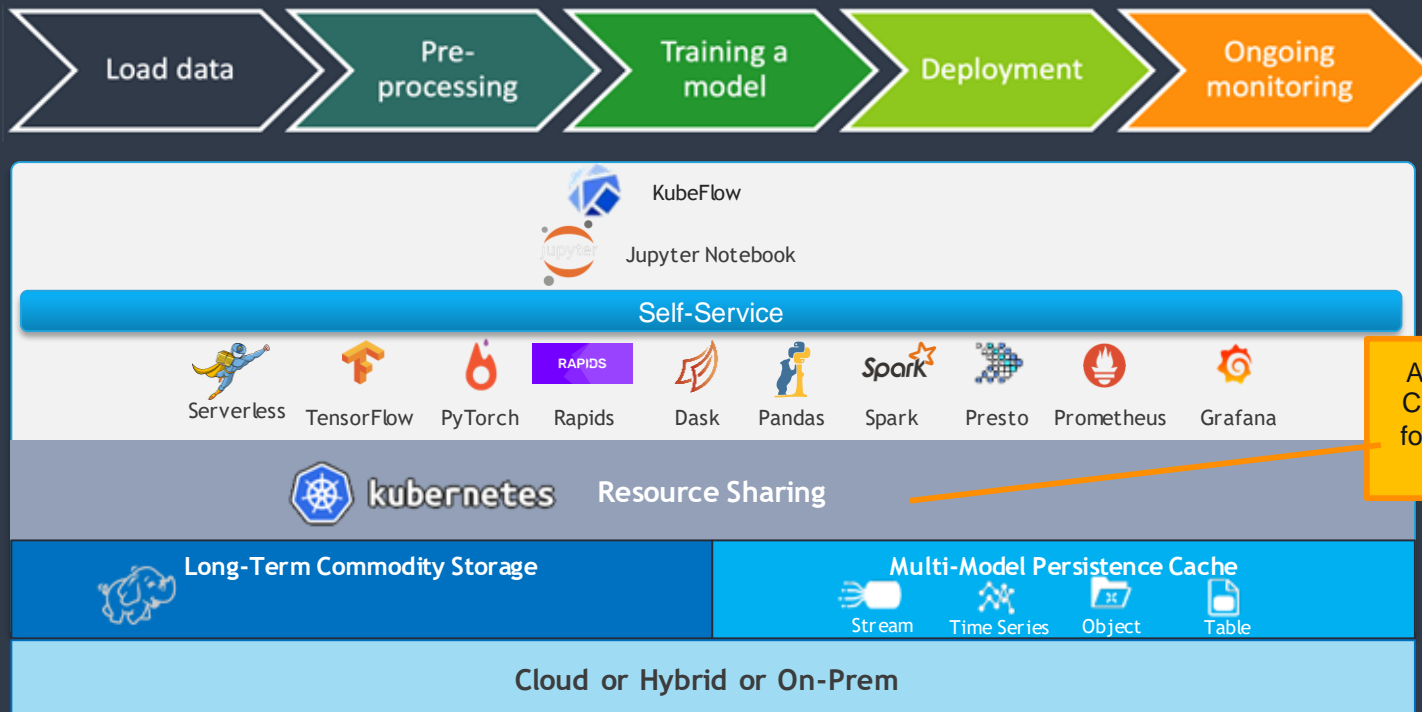
No new skills to learn – use most all common ML tools e.g. Jupy, Kubeflow



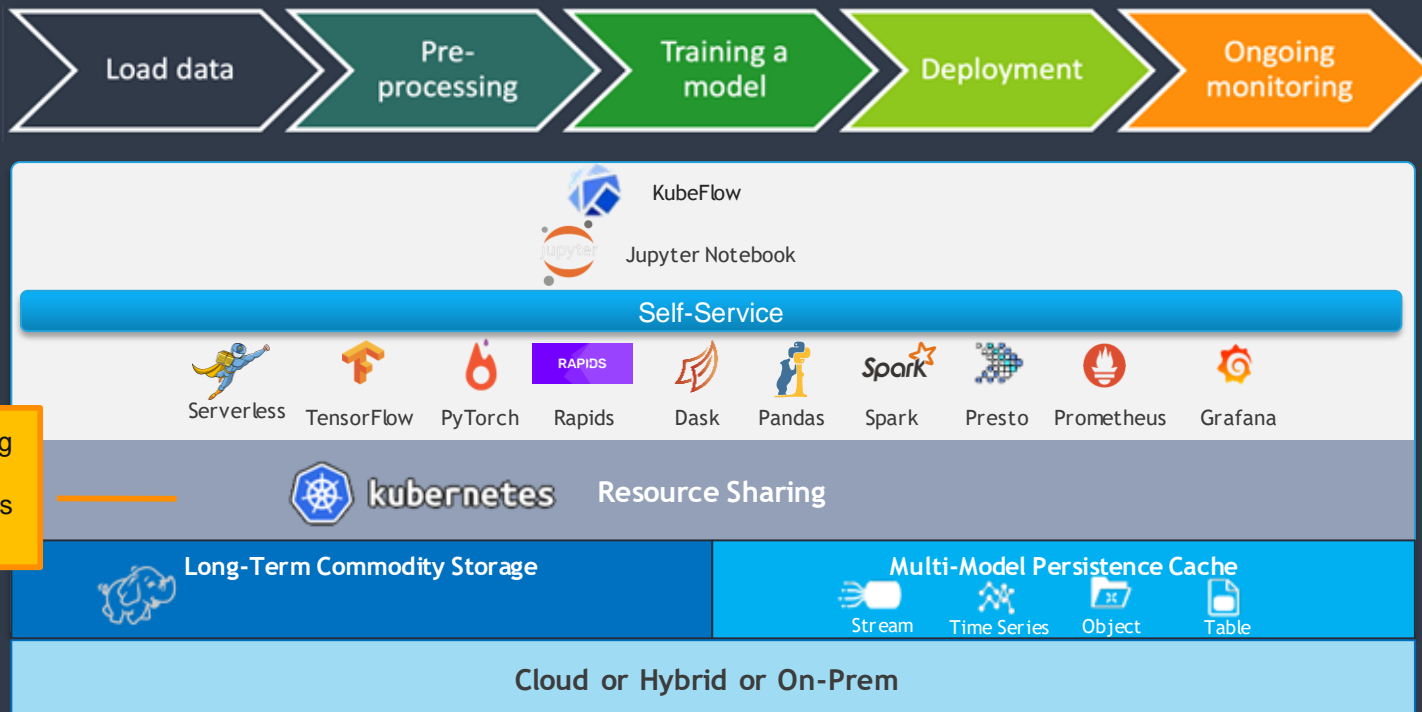
Platform Approach - Summary



Platform Approach - Summary



Platform Approach - Summary



scaling, monitoring and sharing of platform resources using K8S

Q&A

santanud@iguazio.com

Upcoming Meet-up Sessions

- ML Pipelines for Production: KubeFlow
- Why use GPUs to Accelerate ML Projects
- How Serverless Simplifies ML Model Development & Deployment

Thanks