

# Machine Learning Made Easy on Kubernetes: MLOps

Brian Redmond, Azure



# Brian Redmond – Who am I?

- Cloud Architect @ Microsoft (18 years)
- Azure Global Black Belt Team
- Live in Pittsburgh, PA, USA
- Avid marathon runner and outdoors enthusiast
- World traveler



# Let's start with a demo!

“Real world” application with Game of Thrones

What Game of Thrones character is this?

- Let's use Tensorflow and build a image classifier



<http://aka.ms/thrones>



# What is Machine Learning?



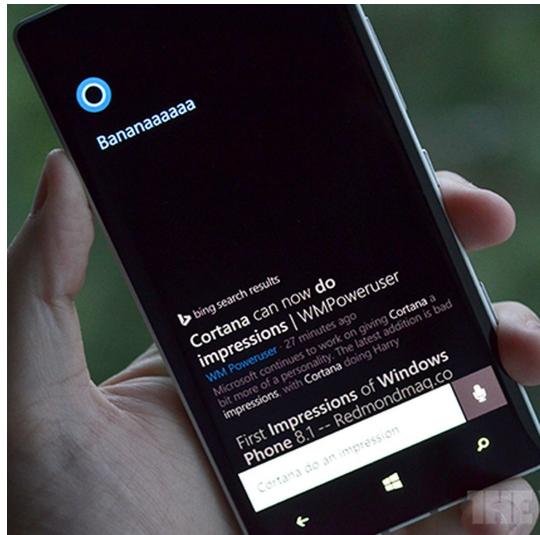
*“The ability to learn without being explicitly programmed,” Arthur Samuel*



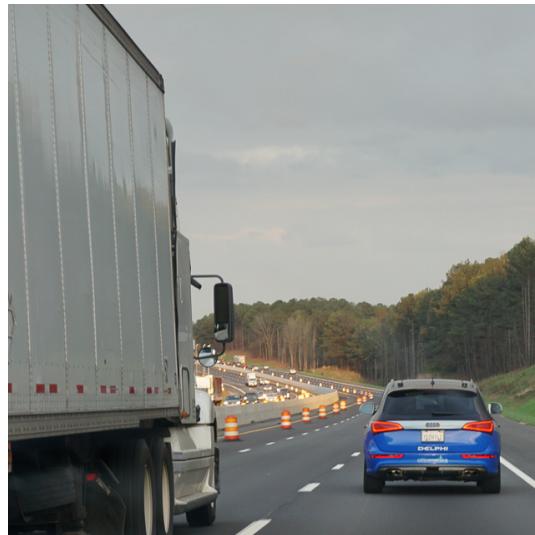
# Machine Learning is super powerful



OpenAI created “MuseNet” which can generate 4-minute musical compositions with 10 different instruments (2019)



Microsoft reaches 'human parity' with new speech recognition system (2016)



Delphi self driving car drove cross country with a computer handling 99% of the drive (2015)



Google DeepMind's AlphaGo program beat a world class Go player in multiple games (2016)



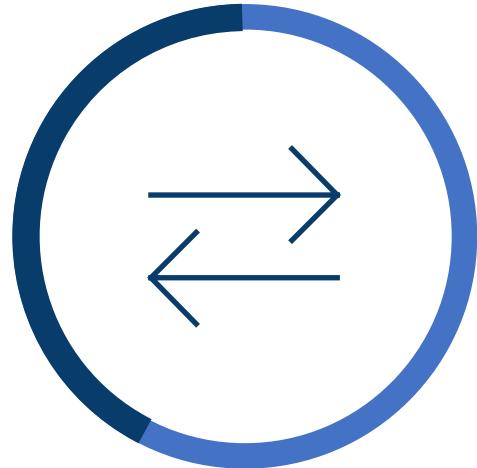
But Machine Learning is hard!



# Kubernetes

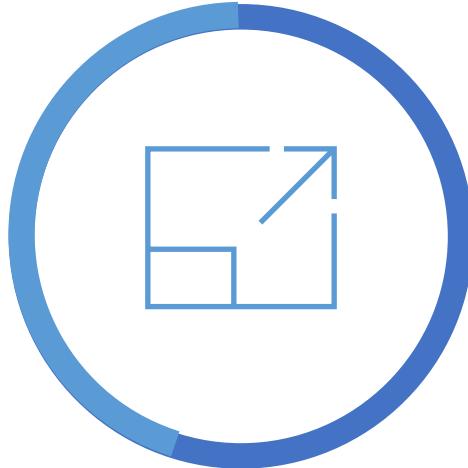
# What's behind the growth?

Kubernetes: the leading orchestrator shaping the future app development and management



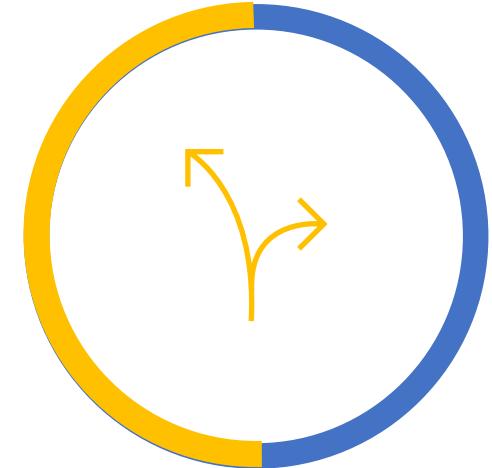
42%

portability



45%

scalability



50%

agility

---

The perceived benefits of Kubernetes

---

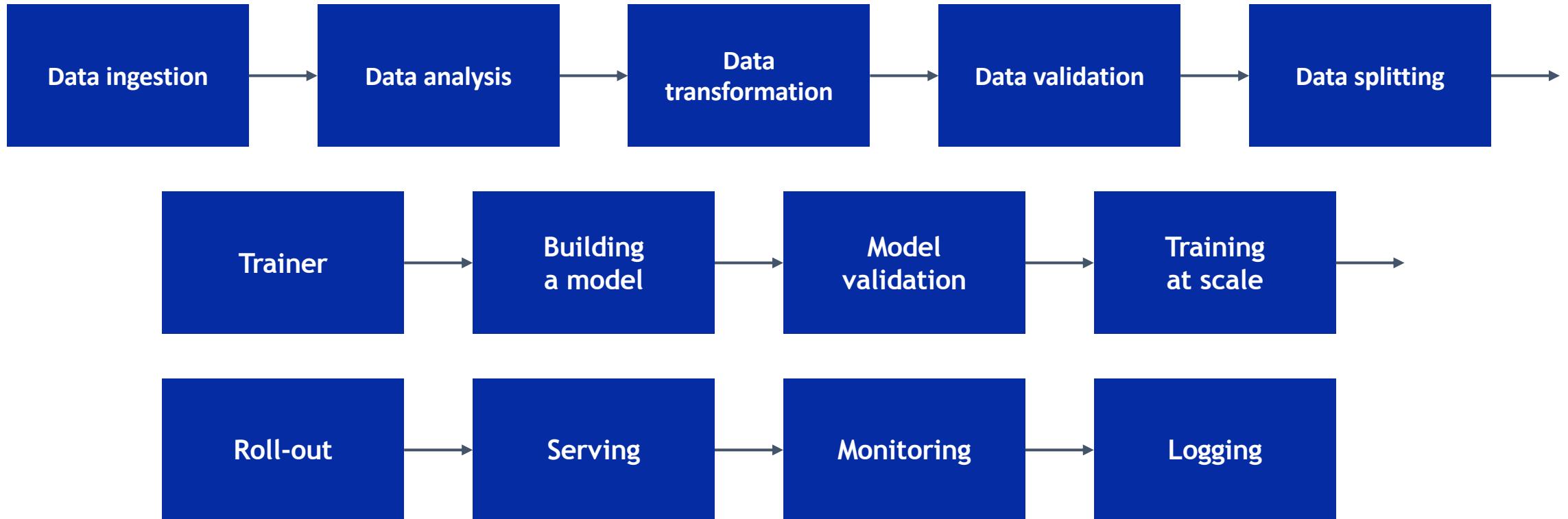
# Cloud Native Apps

# Cloud Native ML?

# Platform

Building a model

# Platform



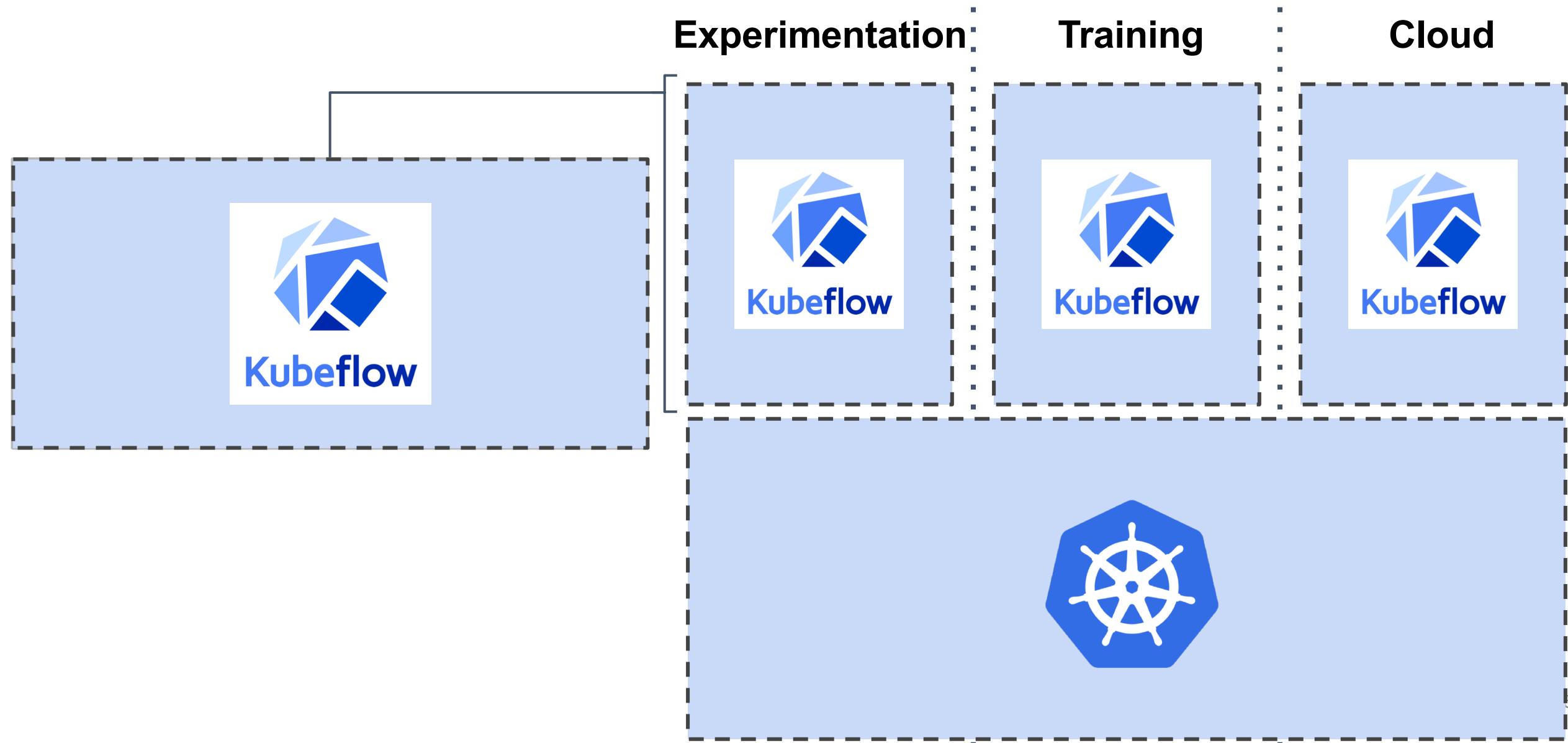
# Kubecon 2017

The image is a composite of three parts. On the left is a large dark slide with the text "Introducing Kubeflow" in white. In the center is a photograph of a presentation stage. Two people are standing behind a blue podium; the man on the left is gesturing with his hands while speaking, and the woman on the right is also gesturing. The podium has the KubeCon and CloudNativeCon logos and the text "North America 2017". In the foreground, the backs of audience members' heads are visible. On the right is a dark footer bar containing the KubeCon logo (a stylized ship wheel icon), the text "KubeCon | CloudNativeCon", and "North America 2017".

Introducing Kubeflow

KubeCon CloudNativeCon  
North America 2017

**Make it Easy for Everyone  
to Develop, Deploy and Manage  
Portable, Distributed ML  
on Kubernetes**

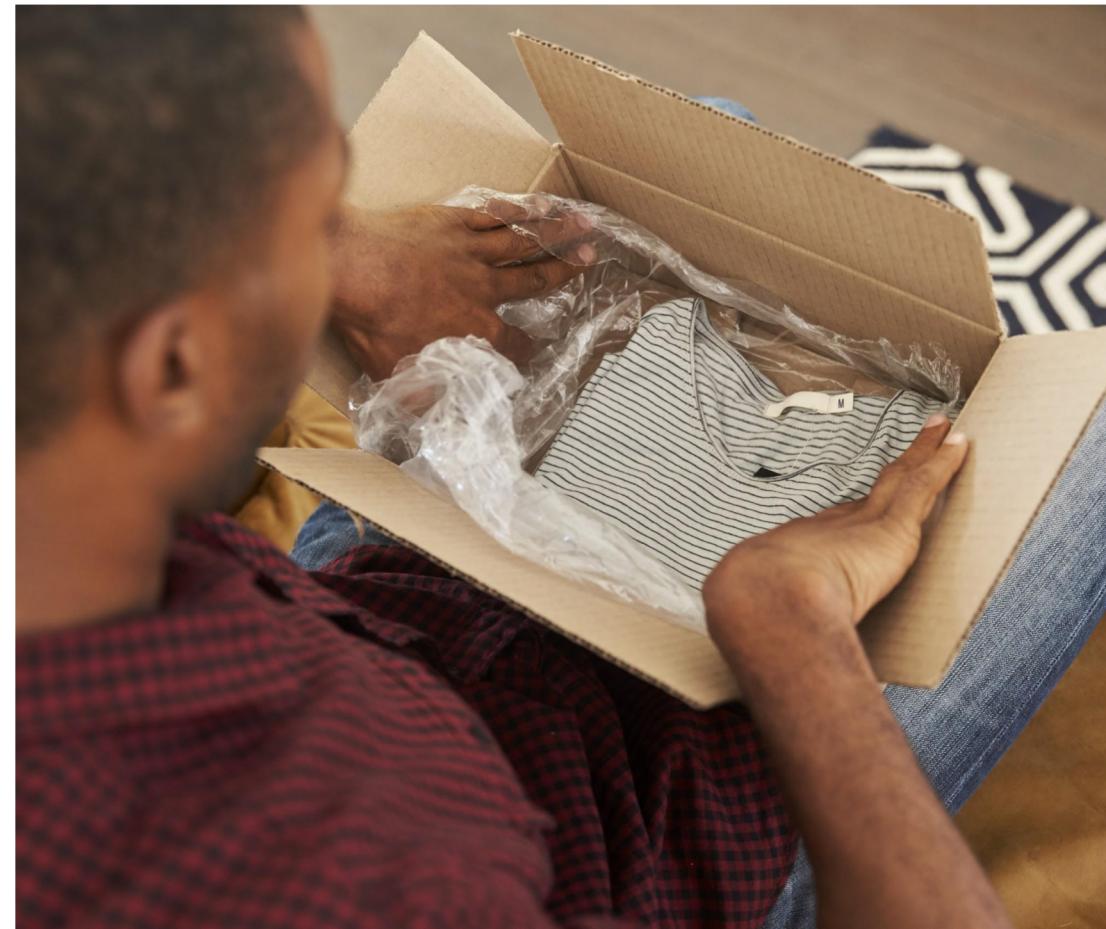


# Cloud Native ML!

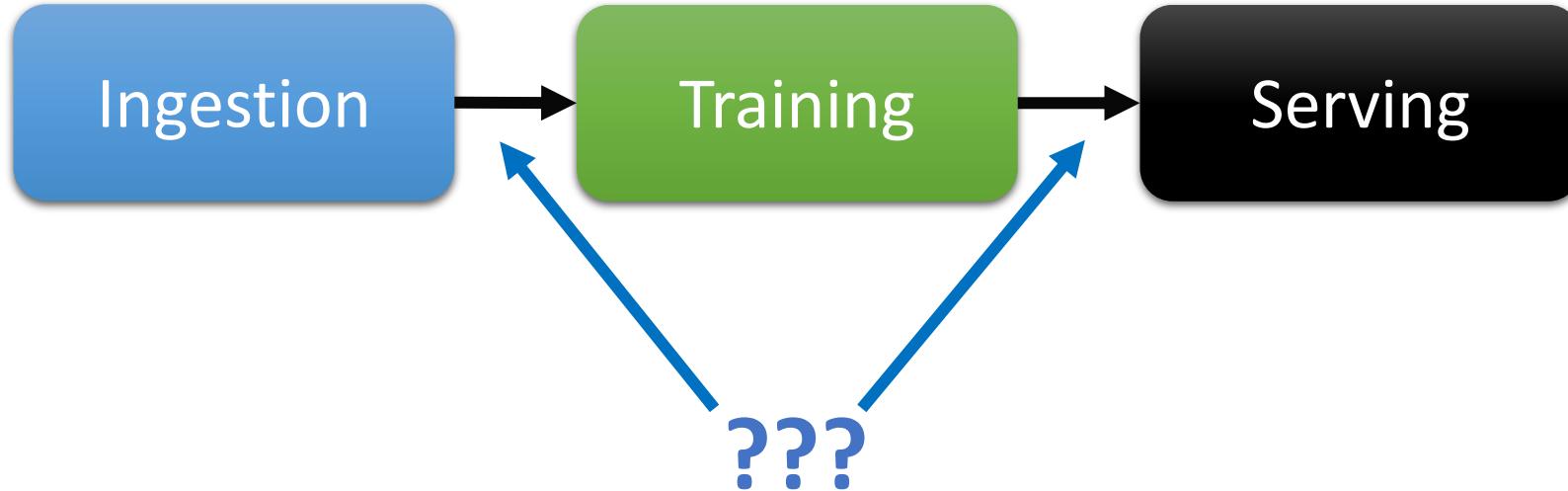
# Kubeflow - What's in the box?

## Key Features

- Ingest your data with Pachyderm
- Services for spawning and managing Jupyter notebooks
- Operators for Tensorflow, PyTorch, MXNet, Chainer jobs
- Serving
  - TF Serving
  - Seldon
  - TensorRT (NVidia)
- Apache Beam (batch and stream processing)
- Katib (hyperparameter tuning)
- Kubebench (benchmarking)
- Pipelines
- Argo CD (GitOps)



# Rich Container Based Pipelines



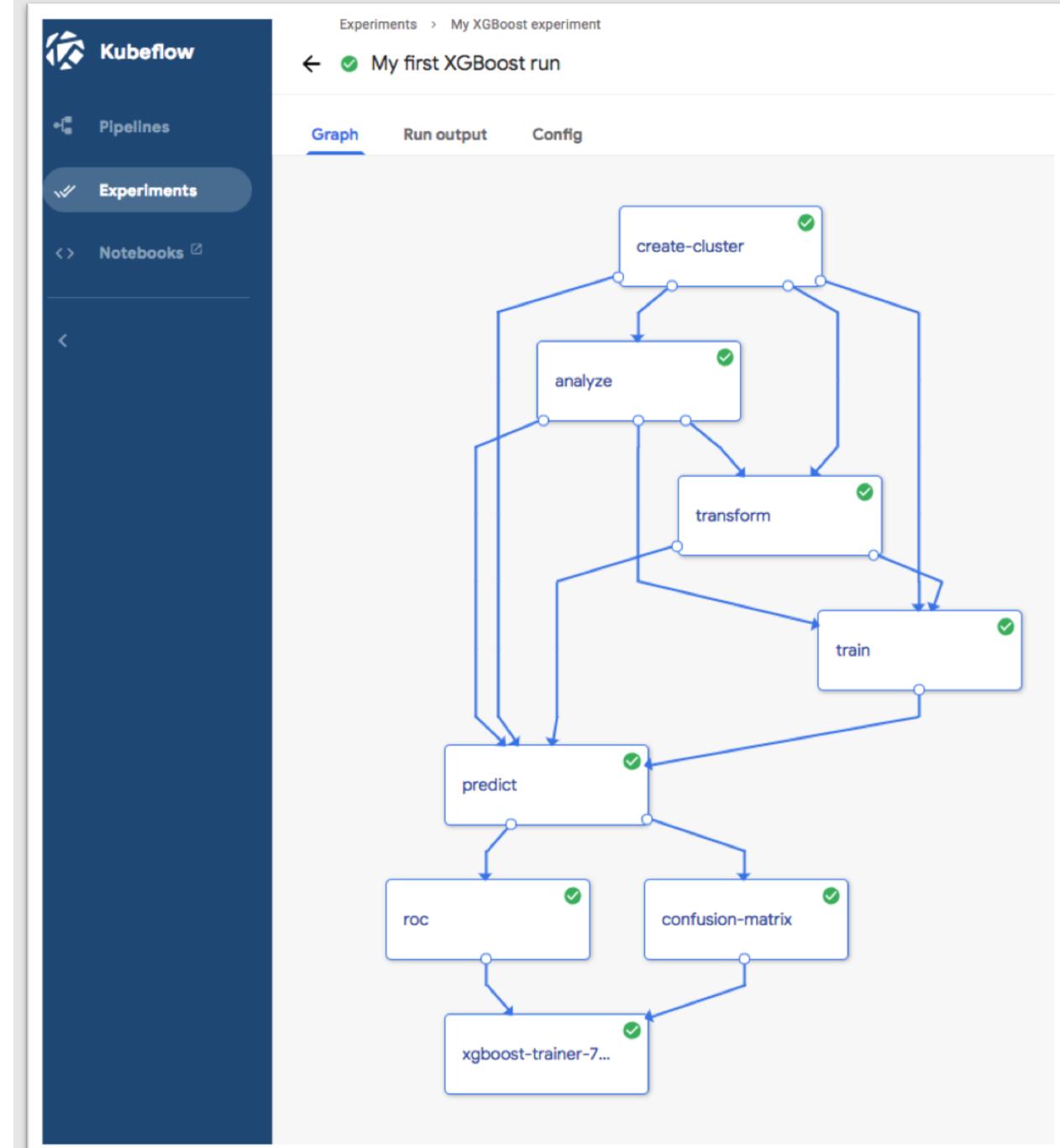
**User Goal** = Repeatable, multi-stage ML training

## Problem

- Tools not built to be containerized/orchestrated
- Coordinating between steps often requires writing custom code
- Different tools have different infra requirements

# Kubeflow Pipeline Details

- Containerized implementations of ML Tasks
  - Escapsulates all the dependencies of a step with no conflicts
  - Step can be singular or distributed
  - Can also involve external services
- Specified via Python SDK
- Inputs/outputs/parameters can be chained together



# Rich Container Based Pipelines



```
ingestStep = dsl.ContainerOp(image=tft_image, <params>,
                             file_outputs={'bucket': '/output.txt'})
```

```
trainStep = dsl.ContainerOp(image=tfjob_image, <params>,
                           arguments=[ingestStep.outputs['bucket']])
```

```
servingStep = dsl.ContainerOp(image=dfs_image, <params>,
                             arguments=[convertStep.outputs['bucket']])
```

# What Game of Thrones character is this?

- Let's use Tensorflow!
- Build a model based on Inception image classification
- Train using TFJob in Kubernetes
- Tensorflow Serving
- Hyperparameter Optimization
- Pipeline workflow
- Jupyter Notebook



<http://aka.ms/thrones>

# Questions?

Find me at @chzbrgr71



Source for demos: <https://github.com/chzbrgr71/got-image-classification>

