Kubeflow Project: Deep Dive

Jeremy Lewi(jlewi@google.com)
David Aronchick (aronchick@google.com)

This talk is a deep dive aimed at current/potential contributors

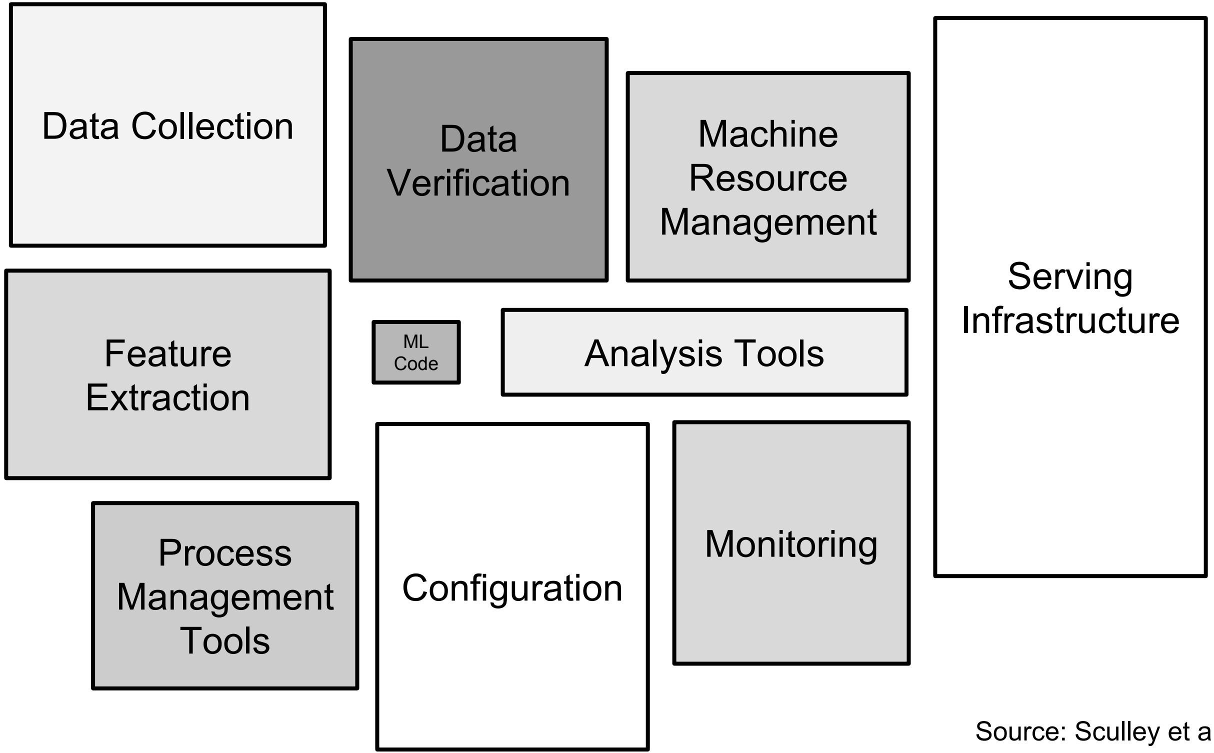
Several Talks Related To Kubeflow

- Tuesday, May 1:
  - Red Hat OpenShift Commons Machine Learning Reception Panel
- Wednesday, May 2:
  - Kubeflow Intro - Michał Jastrzębski & Ala Raddaoui, Intel
- Thursday, May 3:
  - Kubeflow Deep Dive - Jeremy Lewi, Google
  - Build ML Products With Kubeflow - Jeremy Lewi, Google & Stephan Fabel, Canonical
  - Compliant Data Management and Machine Learning on Kubernetes - Daniel Whitenack, Pachyderm
- Friday, May 4:
  - Keynote: Kubeflow ML on Kubernetes - David Aronchick & Vishnu Kannan, Google
  - Conquering a Kubeflow Kubernetes Cluster with ksonnet, Ark, and Sonobuoy - Kris Nova, Heptio & David Aronchick, Google
  - Serving ML Models at Scale with Seldon and Kubeflow - Clive Cox, Seldon.io

Agenda

- What is Kubeflow
- Roadmap
- Core Principles
- Why ksonnet

ML Requires DevOps; lots of it



Data Collection

Data Verification

Machine Resource Management

Serving Infrastructure

Feature Extraction

ML Code

Analysis Tools

Process Management Tools

Configuration

Monitoring

Source: Sculley et al.: Hidden Technical Debt in Machine Learning Systems

Kubeflow: Build Portable ML Solutions Using Kubernetes

What is Kubeflow?

- Community
  - Who: Datascientists, ml researchers, software engineers, product managers
  - What: K8s native platform for ML
  - Why: Because building a platform is too big a problem to tackle alone
- A K8s native platform for ML
  - K8s custom resources for managing ML tasks (distributed training, orchestration, model deployment etc...)
  - microservices for ML (data registries, model databases, hyperparameter tuning, etc...)
  - ksonnet packages to manage infrastructure declaratively
- **Result: E2E ML solutions built on Kubeflow that are portable**
  - Onprem <--> cloud
  - Across problems/domains

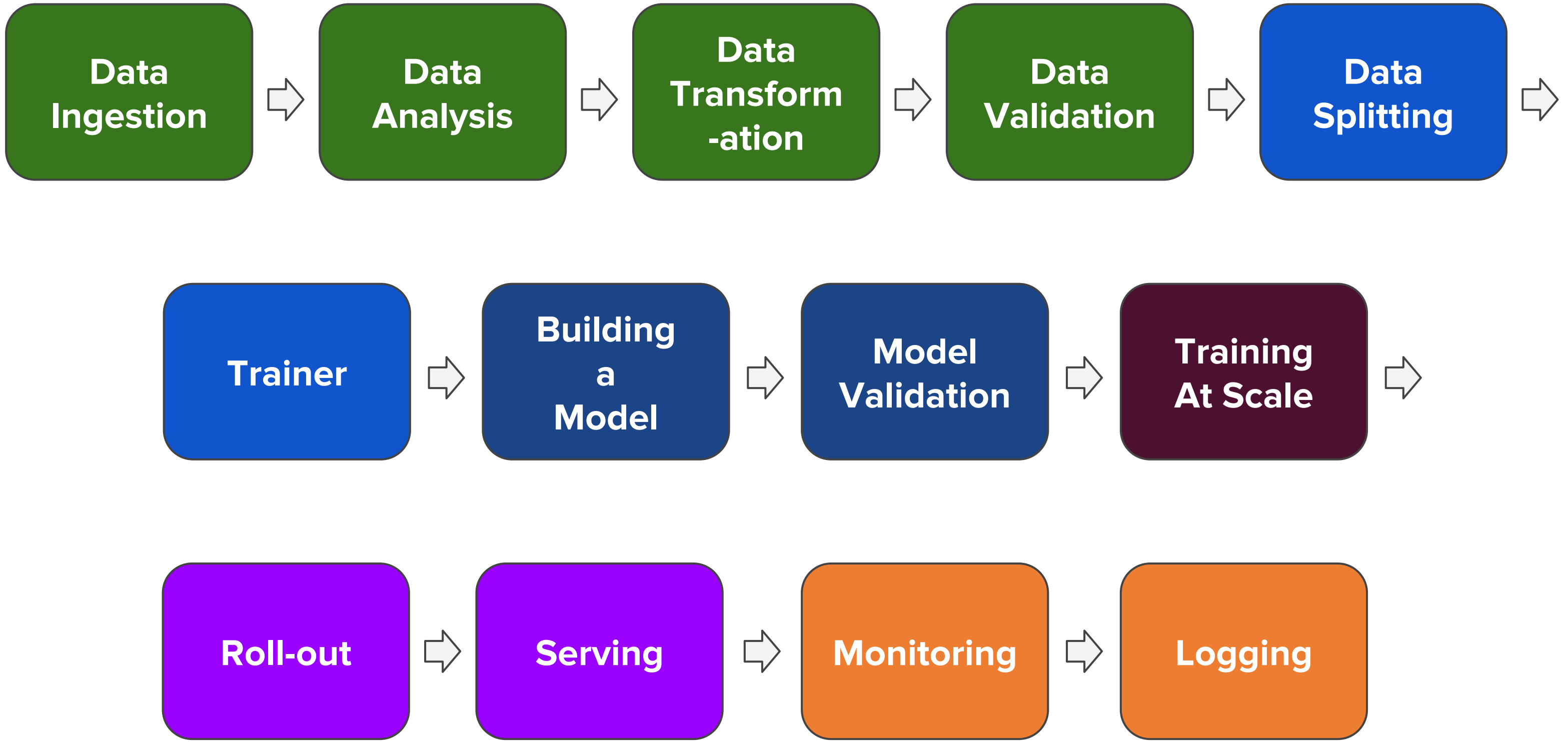
The Community (kubeflow/community)

- 66 individual members
- 12 Organizations
  - Alibaba Cloud, Caicloud, Canonical, Cisco, Datawire, Dell, Github, Google, Heptio, Huawei, Intel, Microsoft, Momenta, Pachyderm, Project Jupyter, Red Hat, Seldon, Weaveworks
- ~ 1000 GitHub events per week (8249 total)
- ~ 44 contributors per week (109 total)
- ~ 40 commits per week (408 total)
- ~ 65 commenters per week (163 total)
- ~ 35 PR creators per week (77 total)
- ~ 85 PRs created per week (601 total)

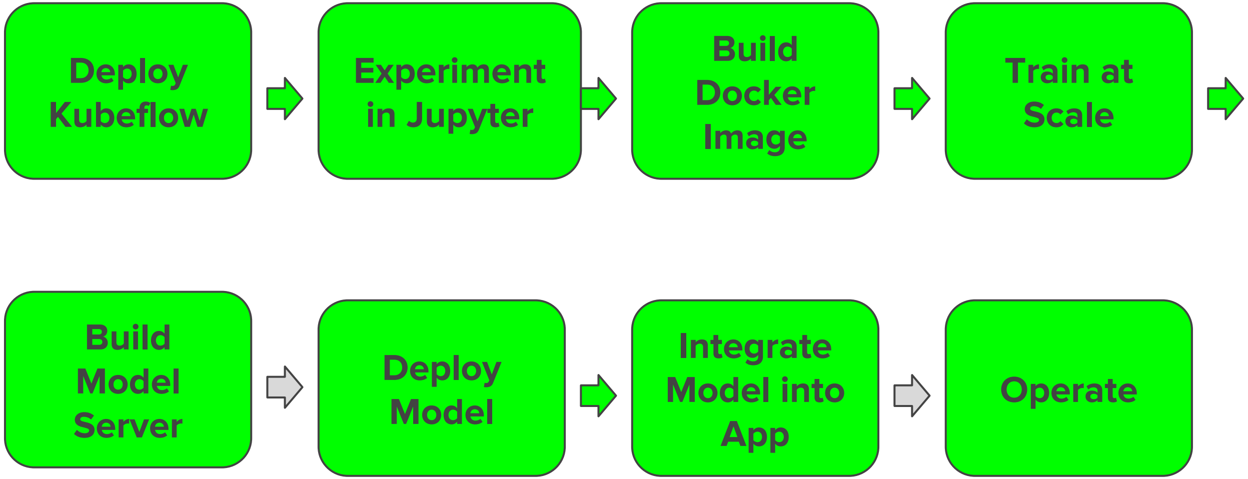
Using cncf/devstats; make dashboards public kubeflow/community#106

Kubeflow ML Platform

ML Platform


Data Ingestion
Data Analysis
Data Transform-ation
Data Validation
Data Splitting
Trainer
Building a Model
Model Validation
Training At Scale
Roll-out
Serving
Monitoring
Logging

User Experience



2 Types of Components within Kubeflow

- Components being developed within Kubeflow
  - Source is in https://github.com/kubeflow/
- Components developed elsewhere but integrated with Kubeflow
  - Source is owned/maintained outside Kubeflow
  - Packages are integrated with Kubeflow
  - **Not subject to Kubeflow governance**
  - No well defined criteria; yet

Projects being developed within Kubeflow

- K8s CRDs for several ML frameworks
  - tf-operator, PyTorch Operator, caffe-2,
  - Horvod for TF
- KVC
  - Kubernetes volume controller
  - Efficiently manage data for ML workloads
- Katib
  - Hyperparameter tuning system Clone of Vizier (Google's HP Tuning System)
- Docker images for ML
  - TFServing images
  - Curated Jupyter Notebook Images

Projects integrated with Kubeflow

- Argo
  - CRD for workflows
- JupyterHub
  - Multi-user server for Jupyter notebooks
- Pachyderm
  - deploy and manage multi-stage data pipelines while maintaining complete reproducibility and provenance
- SeldonIO
  - CRD and tooling for serving and deploying models
- Tensor2Tensor
  - Library of TensorFlow models and datasets for a variety of applications
- TFX Libraries
  - OSS libraries from Google's TensorFlow based platform ML platform (TFX)
  - Currently available: TF Serving, TF Transform and TF Model Analysis (TFMA)

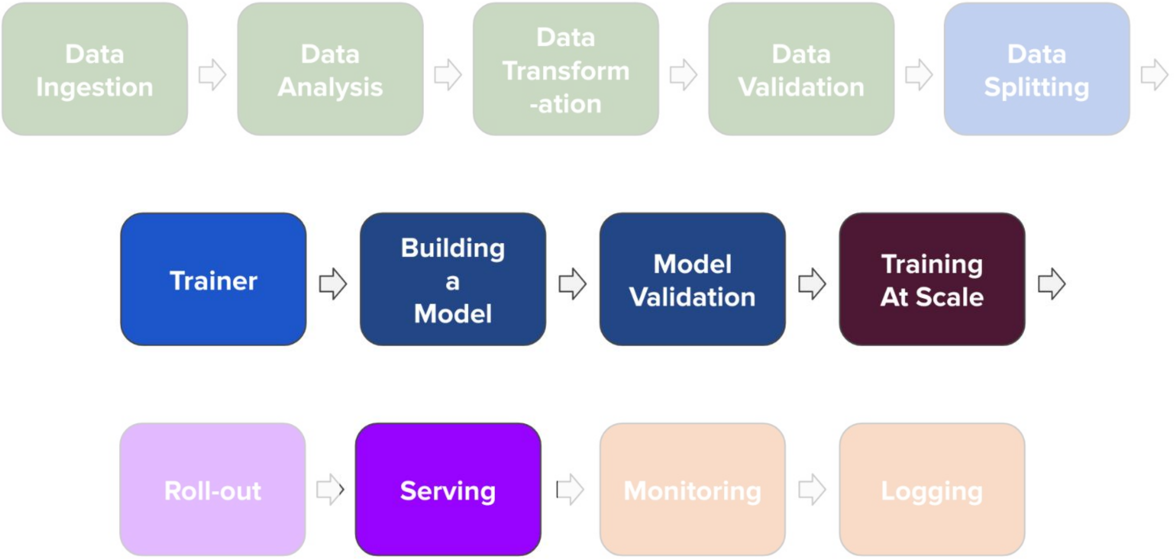
Roadmap

Adding components is easy

- Take yaml manifests -> turn them into ksonnet -> PR
  - Instructions
- Would like to include more components to give users a complete ML platform
  - Model management
  - Experiment management
  - Model analysis
  - Data management
  - Connectors to common data sources

0.1 Release 04/04

- Core components
  - Argo
  - JupyterHub
  - TfJob - v1alpha1
  - Seldon
  - TFServing



Getting to 1.0

- Aiming for 1.0 by EOY (Kubecon USA)
- Core components ready for production
- Core CUJ: Continuous integration & deployment of models
    - Every night my model is automatically retrained on my latest logs
    - If the new model is better it is automatically pushed into production
- **Eventually… Foundation (CNCF? Other?)**

0.2 Release ETA EOQ2

- New Components
  - Katib for HP Tuning
  - PyTorch operator
  - Batch inference
  - Horovod integration
  - Central UI
  - Easier deployment "click to deploy"
- Improvements to existing components
  - TfJob v1alpha2
  - Better error reporting for JupyterHub
  - Improved monitoring for serving
    - ISTIO integration

3 Core Principles

Open

- Why
  - Building an ML platform is too big a challenge to do alone
  - Kubernetes' success illustrates the value of building a broad, energetic community
- What this means
  - All members of the community equal opportunity
    - Except: Google is currently sole owner of kubeflow.org domain
  - All test/release infrastructure is community owned
    - Release/test teams include members from multiple organizations
- **Success will depend on everyone carrying water and chopping wood**
  - # PRs per week is 2x # commits -> Need more reviewers

Low bar; high ceiling

- Low bar - make it super easy to get started
    - Minimize number of K8s concepts/APIs users need to learn just to get started
    - Optimize Kubeflow deployments
        - Work with sig-apps to define appropriate scaffolding for apps
    - **Very active area in the community**
- High ceiling - allow system administrators to do complex customizations
    - Extensibility has been critical to K8s success
    - Users should be able to easily customize individual components

Kubernetes Native

- Run anywhere Kubernetes runs
- Reuse K8s concepts/APIs; don't reinvent the wheel
- Hard dependency on K8s
  - Kubeflow will not invest in running on other platforms

Applying these Principles

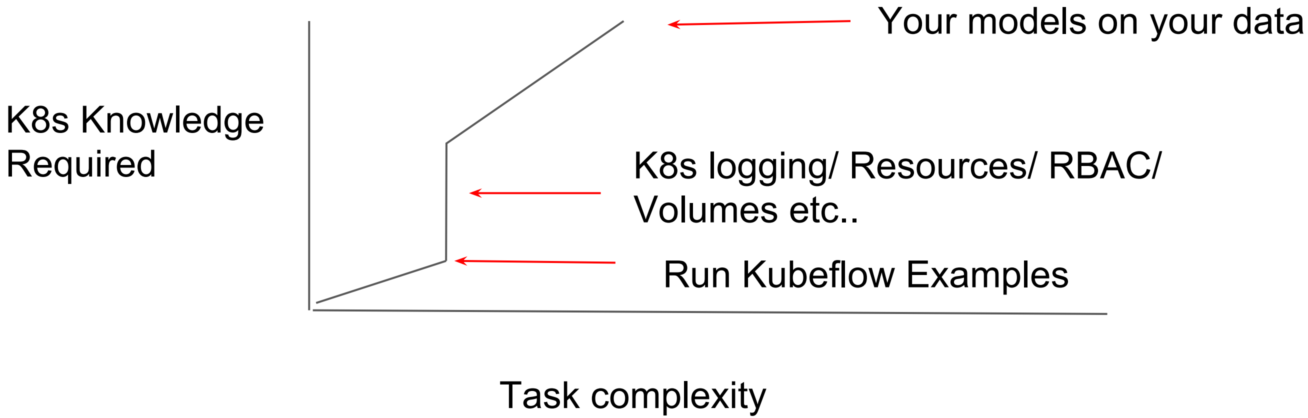
How is Kubeflow K8s Native?

- Kubeflow uses K8s APIs and concepts
    - TfJob & other controllers don't hide K8s APIs
        - Use requests/limits for resource scheduling
        - Let users customize image, arguments, environment variables etc...
    - Volumes for storage
- Kubeflow is managed declaratively matching K8s best practices
    - config intended to be checked into source control
    - embracing GitOps
- Leveraging the K8s ecosystem
    - Use CRDs
    - Want to align with sig-apps app CRD for app management

Can we reconcile K8s Native & Low bar?

- **Hot topic in the community**
- K8s is a steep learning curve for datascientists
- Can we make K8s approachable and avoid users falling off a cliff
  - Learn as you go

What we want to avoid



Your models on your data

K8s Knowledge
Required

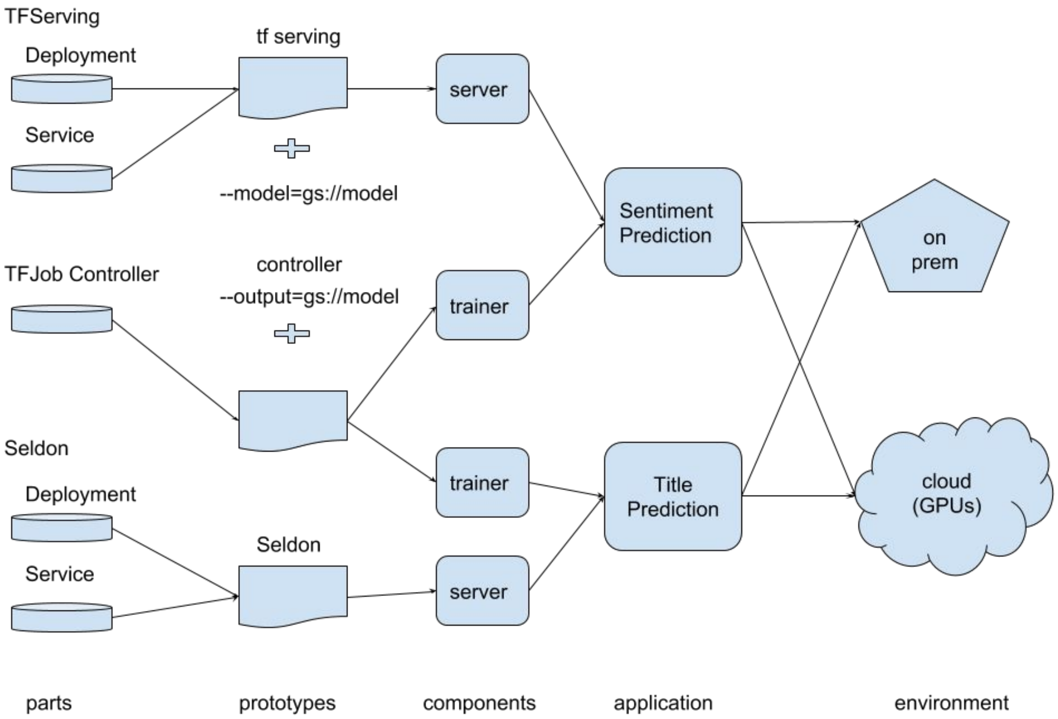
K8s logging/ Resources/ RBAC/
Volumes etc..

Run Kubeflow Examples

Task complexity

Why ksonnet?

Portability is our mission



- Use ksonnet to build ML applications
- Move those applications between environments
  - local -> cloud
  - dev -> test -> prod

Where to go from here

- Main repo: https://github.com/kubeflow/kubeflow
- Community: https://github.com/kubeflow/community
- slack: kubeflow (http://kubeflow.slack.com)
- twitter: @kubeflow
- Mailing list: kubeflow-discuss@googlegroups.com