Compliant Data Management & ML on k8s
Daniel Whitenack, @dwhitena
Outline

1. The what and why of compliance
2. Guidelines for and properties of compliant ML
3. A k8s-based solution
4. Demo!
5. Q&A
The what and why of compliance
The what and why of compliance

We need to explain (or make transparent):

• **Which data** was processed,
• **How** it was processed,
• **Who** processed the data, and
• **What corresponding data** was generated.
Guidelines for and properties of compliant ML
Guidelines for and properties of compliant ML
Guidelines for and properties of compliant ML

Data Ingestion
Data Cleaning
Feature Engineering
Model Selection, Parameter Search

Production ML/AI Model Training
ML/AI Model Inference or Prediction

@dwhitenan

#KubeCon + #CloudNativeCon
Guidelines for and properties of compliant ML

Data Ingestion
Data Cleaning
Feature Engineering
Model Selection, Parameter Search
Feature Transforms
Production ML/AI Model Training
Production Model Testing
Model Export & Optimization
ML/AI Model Inference or Prediction

@dwhitena

#KubeCon + #CloudNativeCon
Guidelines for and properties of compliant ML

- Data Ingestion
- Data Cleaning
- Feature Engineering
- Model Selection, Parameter Search
- Feature Transforms
- Production ML/AI Model Training
- Production Model Testing
- Model Export & Optimization
- Data Input Transforms
- ML/AI Model Inference or Prediction
- Post Processing

@dwhitena

#KubeCon + #CloudNativeCon
Guidelines for and properties of compliant ML

Data Ingestion

Data Cleaning

Feature Engineering

Model Selection, Parameter Search

Feature Transforms

Production ML/AI Model Training

Production Model Testing

Model Export & Optimization

Data Input Transforms

ML/AI Model Inference or Prediction

Post Processing

@dwhitena

#KubeCon + #CloudNativeCon
Guidelines for and properties of compliant ML

Data Ingestion -> Data Cleaning -> Feature Engineering -> Model Selection, Parameter Search

Feature Transforms -> Production ML/AI Model Training -> Production Model Testing -> Model Export & Optimization

Data Input Transforms -> ML/AI Model Inference or Prediction -> Post Processing

@dwhitena

#KubeCon + #CloudNativeCon
Guidelines for and properties of compliant ML

We need:
Guidelines for and properties of compliant ML

We need:

- Framework/infrastructure agnostic way to deploy/manage/define our workflow.
We need:

- Framework/infrastructure agnostic way to deploy/manage/define our workflow.
- Versioning system for all data, models, and processing in our workflow.
We need:

- Framework/infrastructure agnostic way to deploy/manage/define our workflow.
- Versioning system for all data, models, and processing in our workflow.
- Tools for combining the two features above.
A k8s-based solution
We need:

- Framework/infrastructure agnostic way to deploy/manage/define our workflow.
- Versioning system for all data, models, and processing in our workflow.
- Tools for combining the two features above.

A k8s-based solution
A k8s-based solution

We need:

• Framework/infrastructure agnostic way to deploy/manage/define our workflow.
• Versioning system for all data, models, and processing in our workflow.
• Tools for combining the two features above.

@dwhitena
A k8s-based solution

We need:

• Framework/infrastructure agnostic way to deploy/manage/define our workflow.
• Versioning system for all data, models, and processing in our workflow.
• Tools for combining the two features above.

@dwhitena

#KubeCon + #CloudNativeCon
A k8s-based solution

KubeFlow (https://github.com/kubeflow/kubeflow):

• Standards and custom resources for deploying distributed ML tooling on k8s
• Compatibility with supplementary tools including notebooks, serving, and monitoring
A k8s-based solution

**Pachyderm** ([https://github.com/pachyderm/pachyderm](https://github.com/pachyderm/pachyderm)):

- Data agnostic data versioning (model versioning) built on top of object storage
- Language/framework agnostic pipelining on k8s
- Data provenance
- Access controls for data
A k8s-based solution

Announcing!

Pachyderm (https://github.com/pachyderm/pachyderm):
• Data agnostic data versioning (model versioning)
• Language/framework agnostic pipelining
• Data provenance
• Access controls for data


@dwhitena
#KubeCon + #CloudNativeCon
KVC (https://github.com/kubeflow/experimental-kvc):

- Standards and tooling for exposing data to custom resources (e.g., TFJob)
A k8s-based solution

@dwhitena

#KubeCon + #CloudNativeCon
A k8s-based solution
A k8s-based solution

Pachyderm

Versioned Training Data

Pre-processing

Versioned Pre-processed Data

KVC

KubeFlow

TFJob

Master

PS(s)

worker(s)

Serving

Seldon

Training

Versioned model

Model export
A k8s-based solution
Acknowledgements, this work

@jlewi
@aronchick

@Ajay191191
@balajismaniam

@dwhitena
@jdoliner

@hamelsmu

@pdmack

@cliveseldon

@dwhitena

#KubeCon + #CloudNativeCon
Acknowledgements, Pachyderm ecosystem/integrations

Google, Intel, GitHub, Red Hat, SUSE, MINIO, Microsoft, Honeywell

@dwhitena  
#KubeCon + #CloudNativeCon
Resources

- Run this example: https://goo.gl/vFp9JE
- Pachyderm docs: https://goo.gl/ezRzTd
- Pachyderm Slack channel: https://goo.gl/MBLUKf
- KVC repo: https://goo.gl/b4VnFL
- KubeFlow docs/repo: https://goo.gl/JMigjf
- Slack/tweet @dwhitena

@dwhitena  #KubeCon + #CloudNativeCon