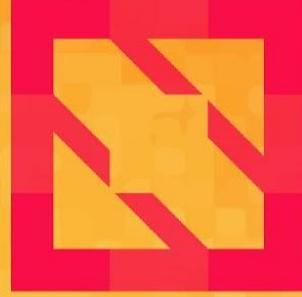




KubeCon



CloudNativeCon

North America 2019





KubeCon



CloudNativeCon

North America 2019

From Notebook to Kubeflow Pipelines

An End-to-End Data Science Workflow

Jeremy Lewi, Google @jeremylewi

Michelle Casbon, Google @texasmichelle

Stefano Fioravanzo, FBK @sfioravanzo

Ilias Katsakoris, Arrikto @elikatsis



What is Kubeflow



Kubeflow

The Kubeflow project is dedicated to making deployments of machine learning (ML) workflows on Kubernetes: simple, portable and scalable.

Platforms Critical to Success With ML



KubeCon

CloudNativeCon

North America 2019

Platform

Lyft Learn

Bloomberg

Stripe
Railyard

AirBnB
BigHead

Google TFX

Many
Others ..

Applications



Infrastructure

Kubernetes

Spark

Borg

An Open Platform For Everyone



KubeCon



CloudNativeCon

North America 2019

Platform

Lyft Learn

Bloomberg

Stripe
Railyard

AirBnB
BigHead

Google TFX

Many
Others ..

Kubeflow

Applications



argo



TensorFlow



jupyter



scikit
learn

XGBoost



Infrastructure

Kubernetes

Spark

Borg

ML Applications Are Distributed Systems



KubeCon



CloudNativeCon

North America 2019

ML
Code

"Hidden Technical Debt in Machine Learning Systems"

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

CI/CD Critical For Managing Complexity

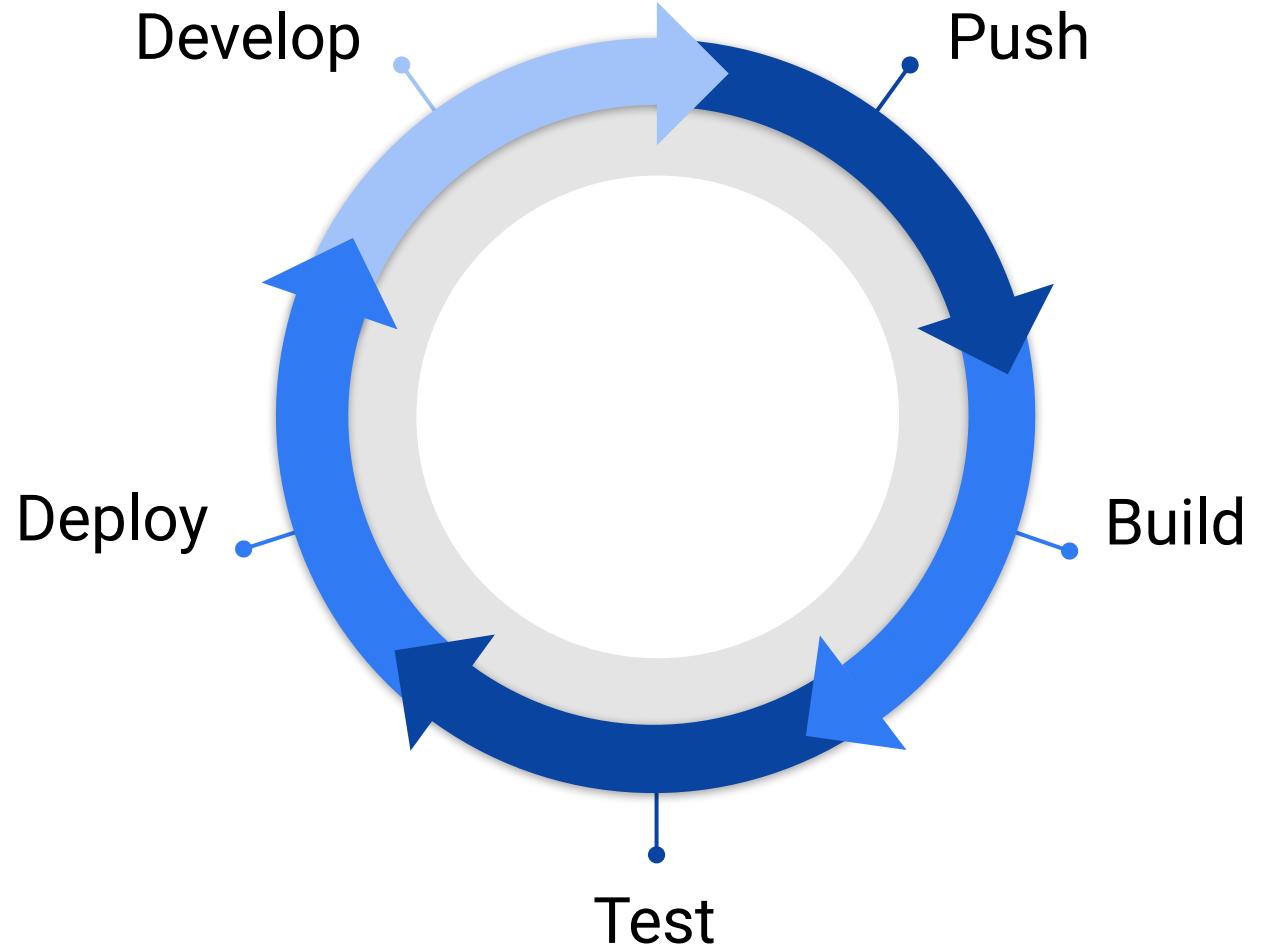


KubeCon

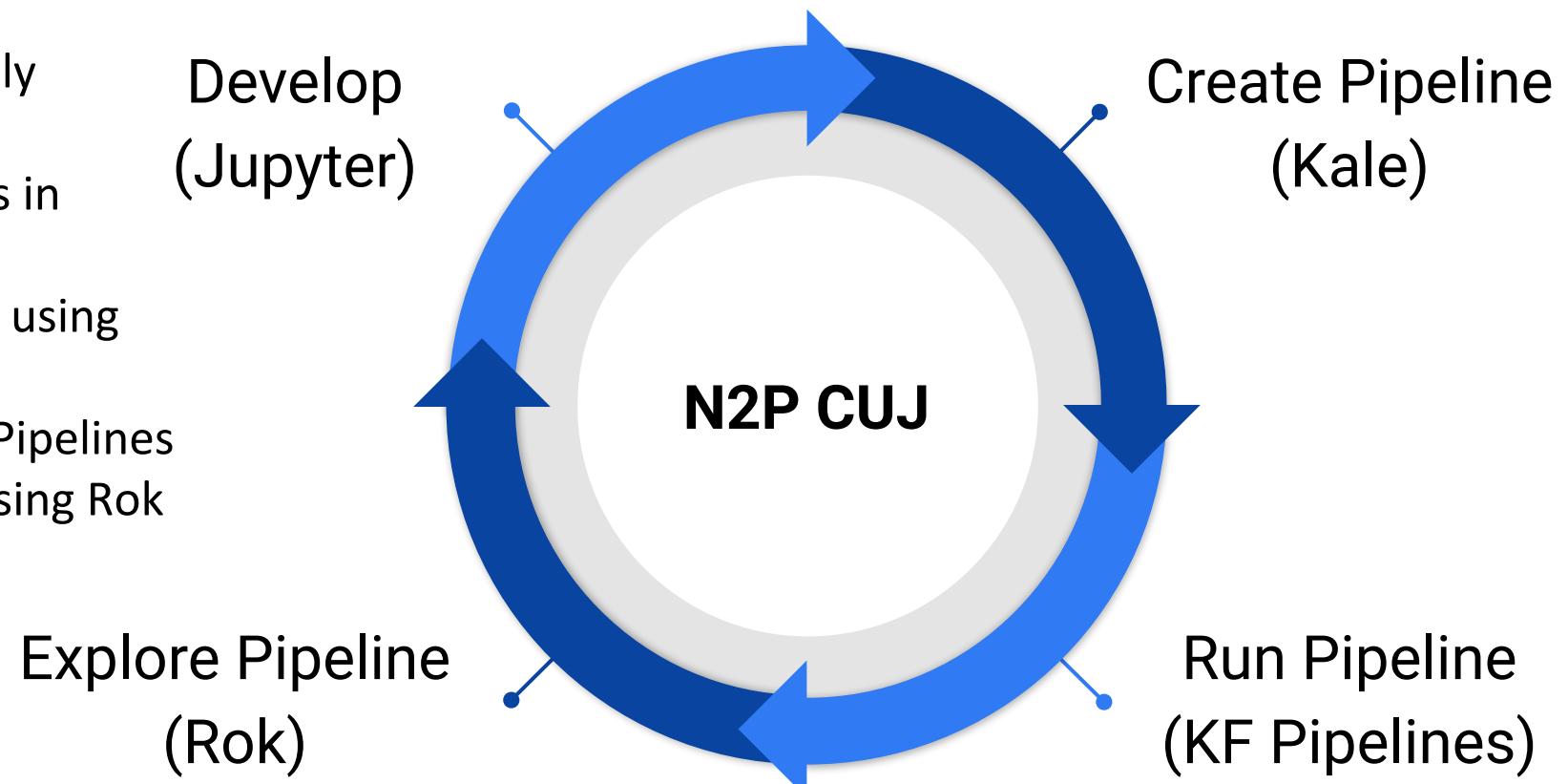
CloudNativeCon

North America 2019

- How do we manage complexity?
 - Push On Green ->
 - Fewer problems
 - Problems less severe
- References
 - <https://cloud.google.com/devops/>
 - <https://ai.google/research/pubs/pub42576>



- How can data scientists continually improve and validate models?
 - Develop models and pipelines in Jupyter
 - Convert notebook to pipeline using Kale
 - Run pipeline using Kubeflow Pipelines
 - Explore and debug pipeline using Rok



Data Science with Kubeflow



KubeCon

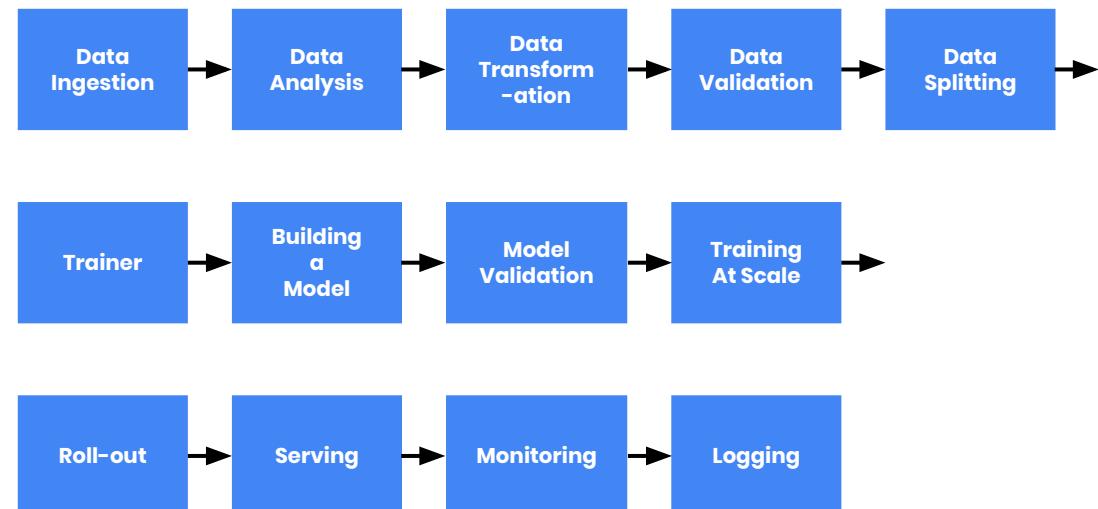
CloudNativeCon

North America 2019

Kubeflow Pipelines exists because Data Science and ML are inherently **pipeline processes**

This workshop will focus on two essential aspects:

- **Low barrier to entry:** deploy a Jupyter Notebook to Kubeflow Pipelines in the Cloud using a fully GUI-based approach
- **Reproducibility:** automatic data versioning to enable reproducibility and better collaboration between data scientists



Data Science with Kubeflow



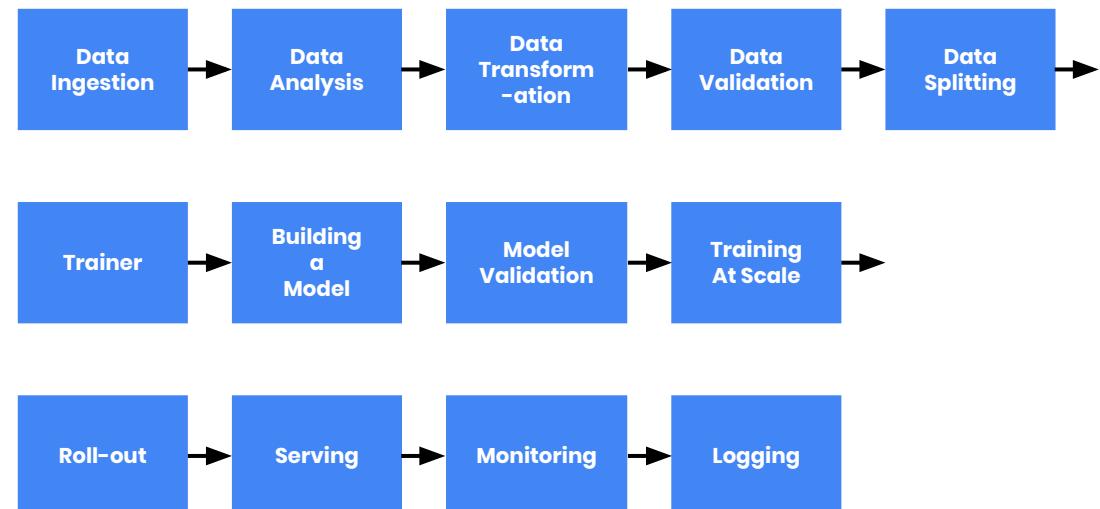
KubeCon

CloudNativeCon

North America 2019

Kubeflow Pipelines exists because Data Science and ML are inherently **pipeline processes**

- This workshop will focus on the following aspects:
- **Low barrier to entry:**  **Kale** Notebook to Kubeflow Pipelines on the cloud using a fully GUI-based approach
 - **Reproducibility:**  **Arritko** versioning to enable reproducibility and better collaboration between data scientists

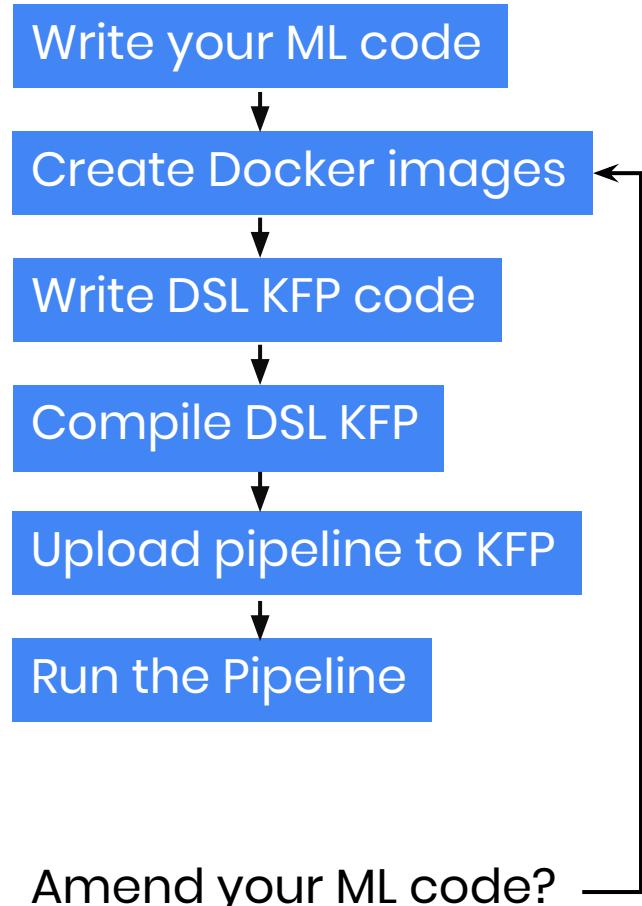


Benefits of running a Notebook as a Pipeline

- The steps of the workflow are clearly defined
- Parallelization & isolation
 - Hyperparameter tuning
- Data versioning
- Different infrastructure requirements
 - Different hardware (GPU/CPU)

Workflow

Before



Workflow

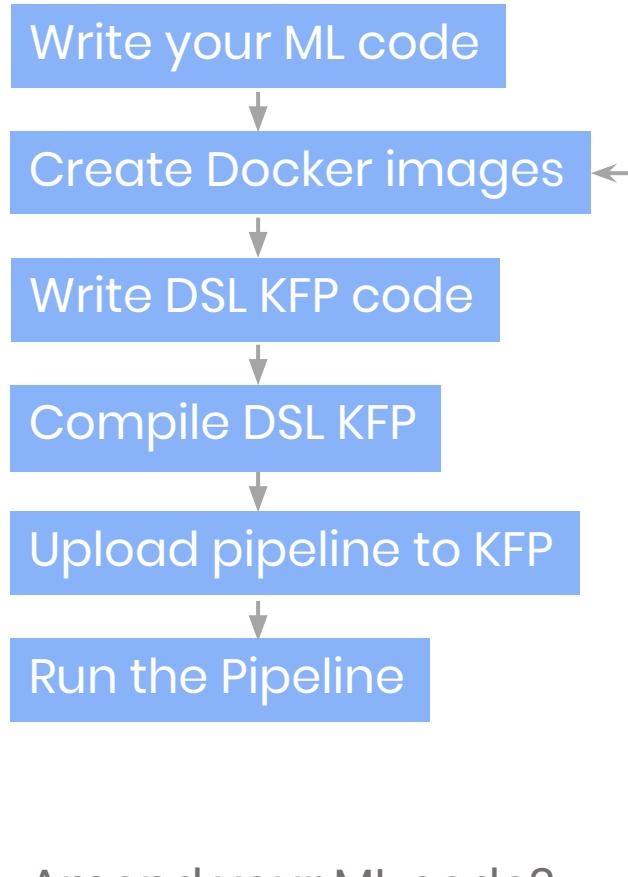


KubeCon

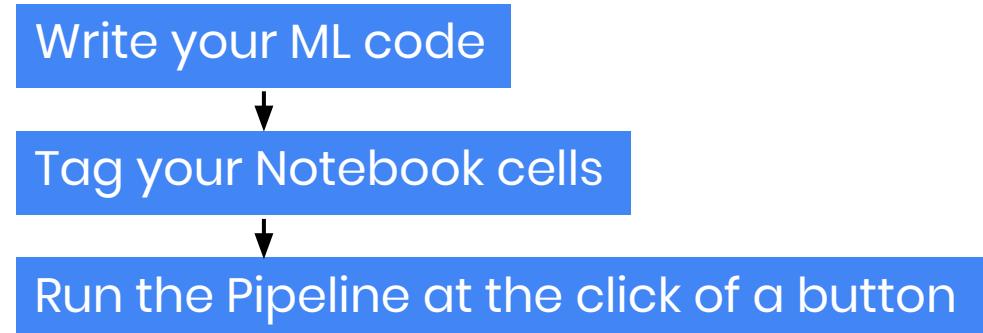
CloudNativeCon

North America 2019

Before



After



Amend your ML code?

Amend your ML code? → Just edit your Notebook!

Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

1

Set up GCP and
install MiniKF

2

Explore the ML code of
the Titanic challenge

4

Reproducibility with
Volume Snapshots

5

Debugging the pipeline

3

Convert notebook to
a Kubeflow pipeline

6

Clean up

Zones:

us-central1-*

us-west1-*

us-west2-*

Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

1

Set up GCP and
install MiniKF

2

Explore the ML code of
the Titanic challenge

4

Reproducibility with
Volume Snapshots

5

Debugging the pipeline

3

Convert notebook to
a Kubeflow pipeline

6

Clean up

Zones:

us-central1-*

us-west1-*

us-west2-*

What is MiniKF?

- Kubeflow on GCP, your laptop, or on-prem infrastructure in just a few minutes
- All-in-one, single-node, Kubeflow distribution
- Very easy to spin up on your own environment on-prem or in the cloud
- MiniKF = MiniKube + Kubeflow + Arrikto's Rok Data Management Platform

KDD 2017 Applied Data Science Paper

KDD'17, August 13–17, 2017, Halifax, NS, Canada

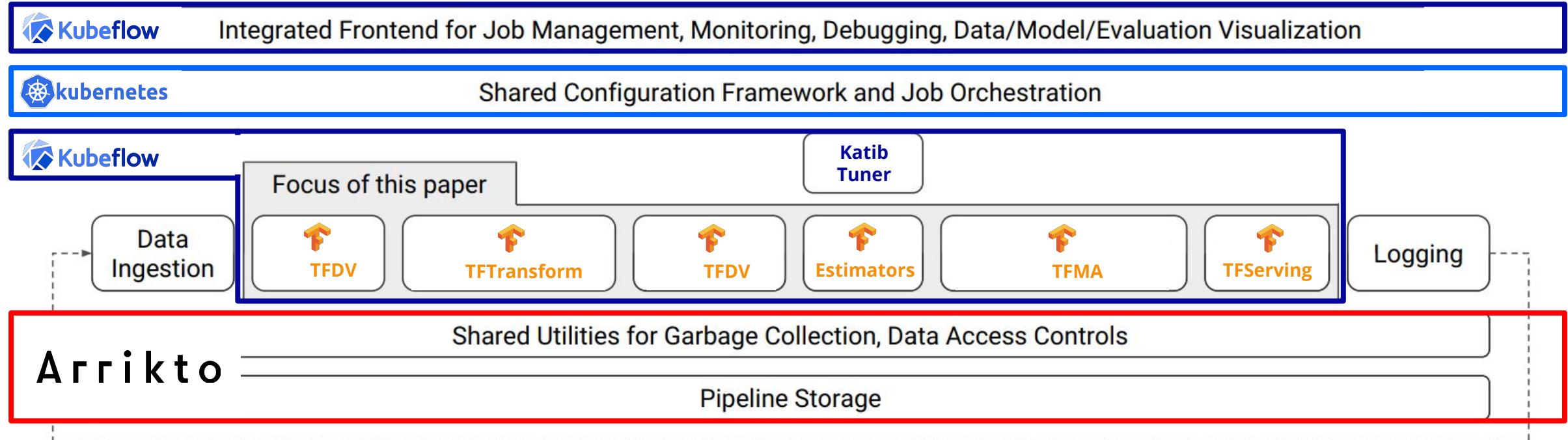
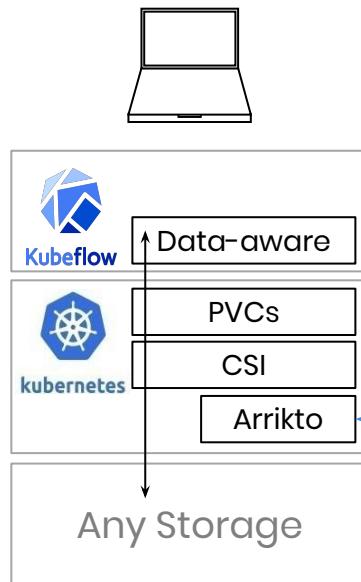


Figure 1: High-level component overview of a machine learning platform.

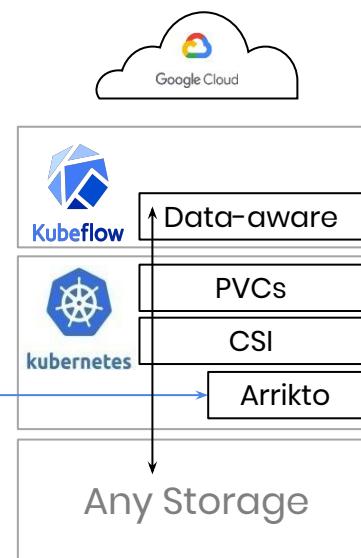
Data Versioning, Packaging, and Sharing

Across teams and cloud boundaries for complete Reproducibility, Provenance, and Portability

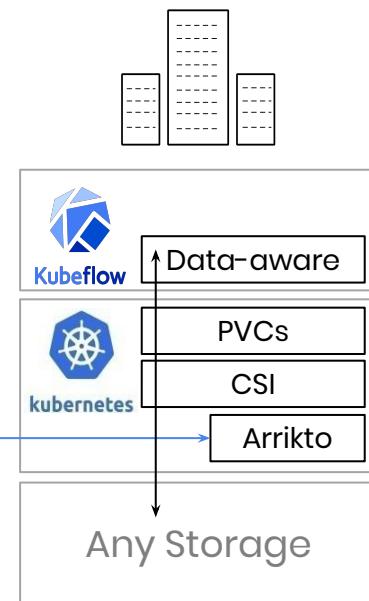
Experimentation

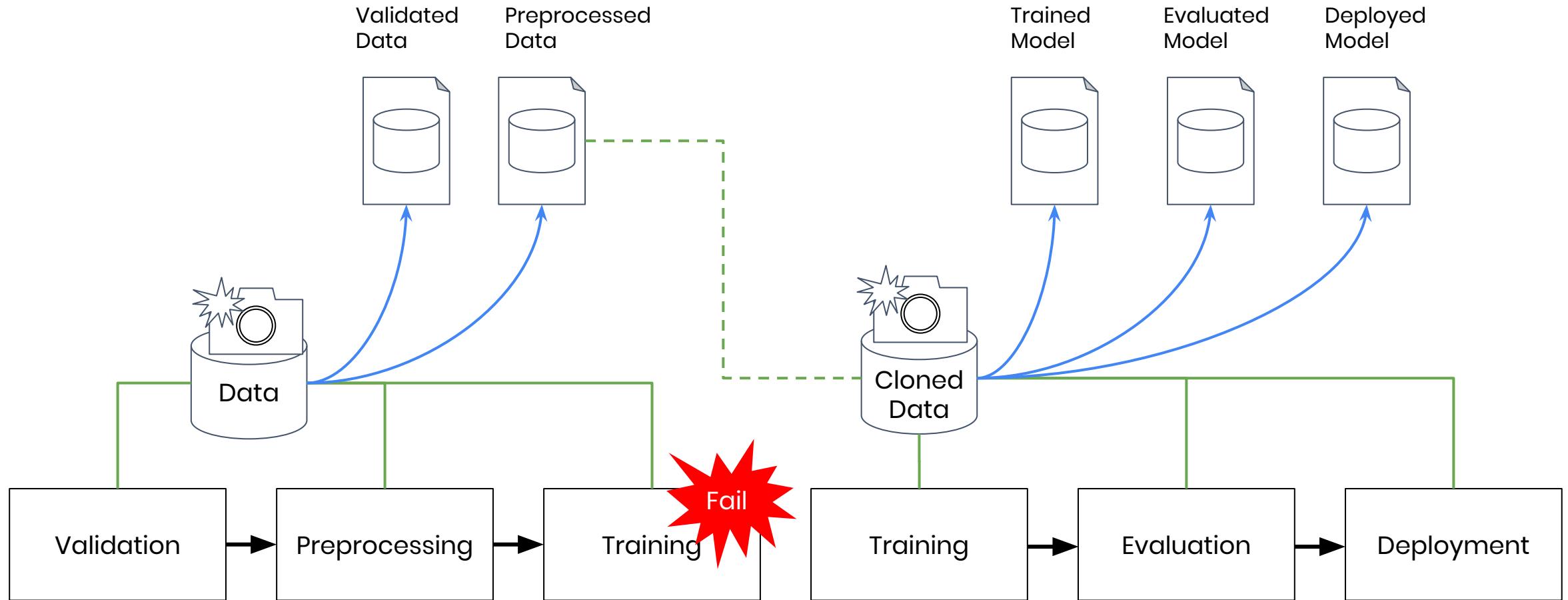


Training



Production





Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

1

Set up GCP and
install MiniKF

2

Explore the ML code of
the Titanic challenge

4

Reproducibility with
Volume Snapshots

5

Debugging the pipeline

Zones:

us-central1-*

us-west1-*

us-west2-*

3

Convert notebook to
a Kubeflow pipeline

6

Clean up

Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

1

Set up GCP and
install MiniKF

2

Explore the ML code of
the Titanic challenge

4

Reproducibility with
Volume Snapshots

5

Debugging the pipeline

Zones:

us-central1-*

us-west1-*

us-west2-*

3

Convert notebook to
a Kubeflow pipeline

6

Clean up

KALE – Kubeflow Automated PipeLines Engine



KubeCon

CloudNativeCon

North America 2019

- Python package + JupyterLab extension
- Convert a Jupyter Notebook to a KFP workflow
- No need for Kubeflow SDK



KALE - Modules



KubeCon

CloudNativeCon

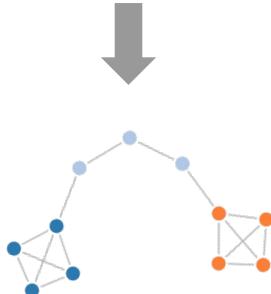
North America 2019

nbparser

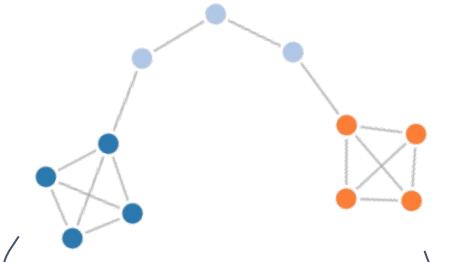


```
[3]: C = np.transpose(A)
      print(C)
```

```
[4]: D = np.matmul(A, B)
      print(D)
```



static_analyzer



Pipeline Step create-matrices

```
[2]: A = np.random.random((10, 10))
      B = np.random.random((10, 10))
```

Pipeline Step matmul

```
[4]: D = np.matmul(A, B)
      print(D)
```

marshal

-- Pipeline Step create-matrices --

```
[2]: A = np.random.random((10, 10))
      B = np.random.random((10, 10))
      kale.marshal.save(A)
      kale.marshal.save(B)
```

-- Pipeline Step matmul --

```
[4]: A = kale.marshal.load("A.npy")
      B = kale.marshal.load("B.npy")
      D = np.matmul(A, B)
      print(D)
```

Derive pipeline
structure

Identify
dependencies

Inject data
objects

codegen

```
def {{ function_name }}{{ function_args|join(' ', '') }}:
    from kale.converter.odo import resource_save, resource_load
    _odo_data_directory = "/data/{{ pipeline_name }}/_odo_data/"
    _input_data_folder = '/data/{{ pipeline_name }}/'

    # ----- DATA LOADING -----
    {% for in_var in in_variables %}
    [...]
    {{ in_var }} = resource_save(
        _odo_data_directory + _odo_load_file_name)
    {% endfor %}
    # ----- DATA LOADING -----
```



```
{% for block in function_blocks %}
{{block|indent(4, True)}}
{% endfor %}
# ----- DATA SAVING -----
{% for out_var in out_variables %}
[...]
resource_load(
    {{ out_var }}, _odo_data_directory + "{{ out_var }}")
# ----- DATA SAVING -----
```



Generate &
deploy pipeline

Contribute!



KubeCon



CloudNativeCon

North America 2019

github.com/kubeflow-kale

 **Kubeflow Kale**
Automation tool to deploy Jupyter Notebooks to Kubeflow Pipelines
<https://kubeflow-kale.github.io>

[Repositories 4](#) [Packages](#) [People 3](#) [Teams](#) [Projects](#) [Settings](#)

Pinned repositories

 [kale](#) 
Convert a JupyterNotebook to a Kubeflow Pipeline deployment.
Python 22 5

 [jupyterlab-kubeflow-kale](#) 
JupyterLab extension to provide a Kubeflow specific left area for Notebooks deployment
TypeScript 2 3

Kale Intro on Medium: <https://bit.ly/2qjXXhF>

Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

1

Set up GCP and
install MiniKF

2

Explore the ML code of
the Titanic challenge

4

Reproducibility with
Volume Snapshots

5

Debugging the pipeline

Zones:

us-central1-*

us-west1-*

us-west2-*

3

Convert notebook to
a Kubeflow pipeline

6

Clean up

Agenda

g.co/codelabs/kubeflow-minikf-kale

- 1 Set up GCP and install MiniKF
- 2 Explore the ML code of the Titanic challenge
- 3 Convert notebook to a Kubeflow pipeline
- 4 Reproducibility with Volume Snapshots
- 5 Debugging the pipeline
- 6 Clean up

Zones:
us-central1-*
us-west1-*
us-west2-*

Agenda



KubeCon



CloudNativeCon

North America 2019

g.co/codelabs/kubeflow-minikf-kale

- 1 Set up GCP and install MiniKF
- 2 Explore the ML code of the Titanic challenge
- 3 Convert notebook to a Kubeflow pipeline
- 4 Reproducibility with Volume Snapshots
- 5 Debugging the pipeline
- 6 Clean up

Zones:

us-central1-*

us-west1-*

us-west2-*

What's new in v0.7



KubeCon

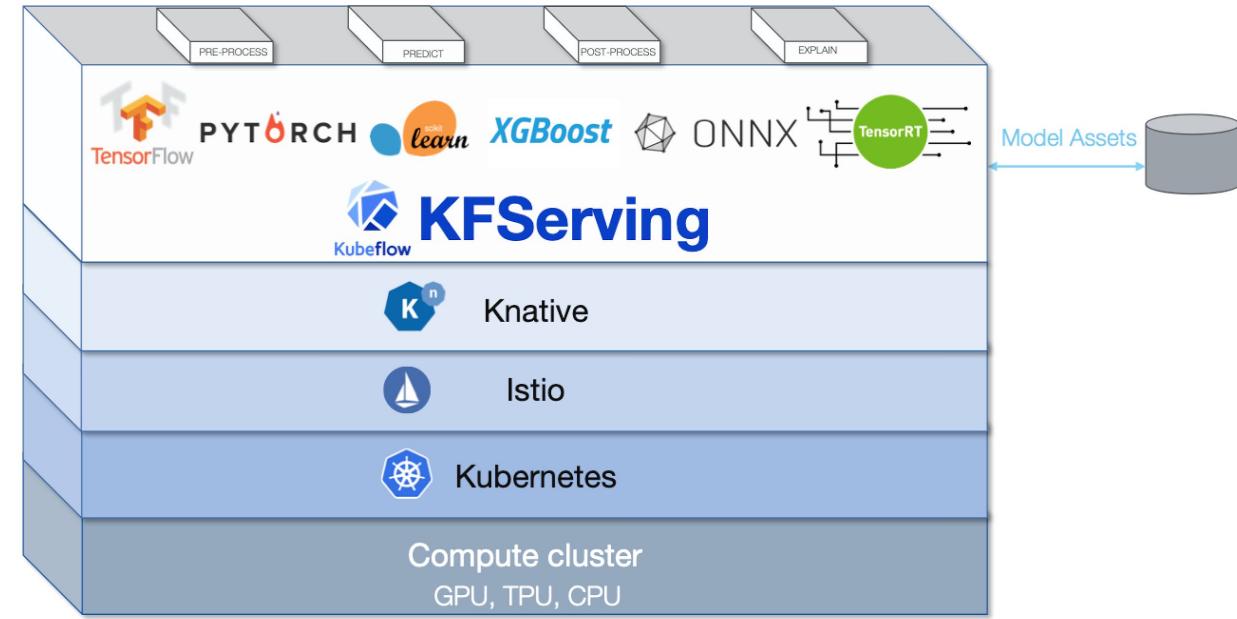
CloudNativeCon

North America 2019

- KFServing for model deployment and management
- kfctl simpler syntax – deploy with 1 command

```
kfctl apply -f kfdef.yaml
```

- Improved multi-user support
 - Aggregated roles
- Hyperparameter tuning
 - A “Suggestions CR” that provides suggestions to improve experiments
 - A more robust metric collector and prometheus runtime metrics and counters
 - More back-end database options



What's new in v0.7



KubeCon

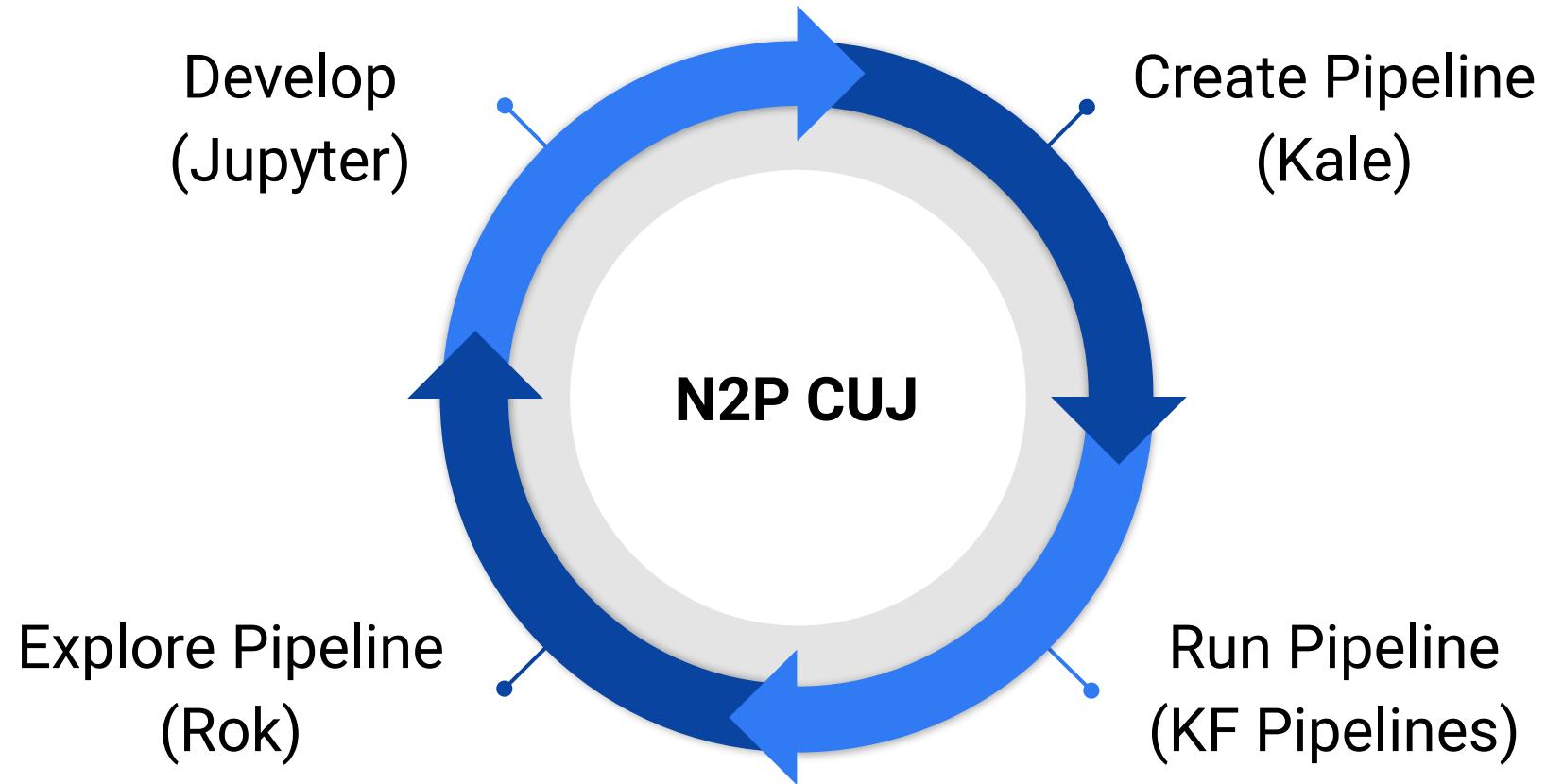


CloudNativeCon

North America 2019

- Pipelines
 - Performance improvements
 - Automatic metadata logging for TFX pipelines
 - New looping constructs `withItems` and `withParams`

Notebook-to-Pipeline CUJ



Ecosystem-supported CUJ for Kubeflow 1.0 coming in Jan 2020

Community



KubeCon

CloudNativeCon

North America 2019

Kubeflow is open

- Open community
- Open design
- Open source
- Open to ideas

Get involved

- github.com/kubeflow
- kubeflow.slack.com
- @kubeflow
- kubeflow-discuss@googlegroups.com
- Community call on Tuesdays

