Build an Event Driven Machine Learning Pipeline on Kubernetes

Yasushi Osonoi
Developer Advocate IBM

Animesh Singh
STSM, IBM
kubeflow kfserving maintainer
Center for Open Source Data and AI Technologies (CODAIT)

**Code** – Build and improve practical frameworks to enable more developers to realize immediate value.

**Content** – Showcase solutions for complex and real-world AI problems.

**Community** – Bring developers and data scientists to engage with IBM

- Team contributes to over 10 open source projects
- **17 committers** and many contributors in Apache projects
- Over 1100 JIRAs and **66,000 lines of code** committed to Apache Spark itself; over 65,000 LoC into SystemML
- Over **25 product lines** within IBM leveraging Apache Spark
- Speakers at over 100 conferences, meetups, unconferences and more

CODAIT
codait.org
Who We Are

We are a group of data scientists and open source developers based out of IBM’s Watson West building in San Francisco. CODAIT was formerly known as the Spark Technology Center. In addition to the Apache Spark data science stack, the Center’s expanded mission will include core frameworks for deep learning. We aim to make AI models dramatically easier to create, deploy, and manage in the enterprise.
DEVELOPER ADVOCATE in TOKYO
Tokyo Team is a part of Worldwide Developer Advocate Teams!

Developer Advocate City Leader
AKIRA ONISHI

WW Developer Advocate
NORIKO KATO

WW Developer Advocate
KYOKO NISHITO

Client Developer Advocate
YASUSHI OSONOI

Program Manager
TOSHIO YAMASHITA

WW Developer Advocate
TAIJI HAGINO

WW Developer Advocate
AYA TOKURA

Digital Developer Advocate
JUNKI SAGAWA

@taiponrock
Code Patterns

- **Object tracking in video with OpenCV and Deep Learning**
  - Get the Code
  - Technologies: Artificial Intelligence, Cloud

- **Locate and count items with object detection**
  - Get the Code
  - Technologies: Artificial Intelligence, IBM PowerAI

- **Build a secure e-voting app**
  - Get the Code
  - Technologies: Blockchain, Hyperledger Fabric

- **Serverless image processing with Cloud Object Storage**
  - Get the Code
  - Technologies: Cloud, Data stores

- **Enhance customer helpdesks with Smart Document Understanding**
  - Get the Code
  - Technologies: Machine Learning

- **Create a cognitive news search app**
  - Get the Code
  - Technologies: Watson, Natural Language Processing

- **Get customer sentiment insights from product reviews**
  - Get the Code
  - Technologies: Watson, Natural Language Processing

- **Build fault-tolerant microservices**
  - Get the Code
  - Technologies: Cloud, Containerization

https://developer.ibm.com/patterns/
Code の力で日本の未来を変えよう

生産性を高めアプリ開発を加速する 140 以上の日本語版 Code Patterns、スキルアップに役立つ 6,000 を超える技術記事

お勧めコンテンツ

【5月16日】IBM Developer Dojo 始動のお知らせ

無料で使える IBM Cloud ライトアカウントを作成しよう

従量課金アカウントへのアップグレード方法をご紹介

IBM Cloud Internet Services(CIS)の機能紹介

このページを共有

→ Code Patterns を見る
→ デベロッパーアドボケイトとは？
→ ニュースレター購読

https://developer.ibm.com/jp/
Please follow me @osonoi
IBM's history of actively fostering balanced communities

“For more than 20 years, IBM and Red Hat have paved the way for open communities to power innovative IT solutions.” – Red Hat
IBM recently open sourced some key technologies for AI, including:

• The **AI Fairness 360 toolkit** (AIF360), an open source software toolkit that can help detect and remove bias in machine learning models
• The **Adversarial Robustness Toolbox** for rapid crafting and analysis of attack and defense methods for machine learning models
• **Fabric for Deep Learning** (FfDL, pronounced fiddle), a deep learning platform offering TensorFlow, Caffe, PyTorch etc. as a Service on Kubernetes

This space is as hot as they come, so look for more open source innovation from IBM in the months to come.
1997
IBM Watson
Jeopardy
2011

Apple's
releases Siri
1997

Facebook's
face recognition
2015

Siri gets
deep learning
2016

IBM Deep Blue
chess
2012

Progress in Deep Learning
2012

Introduced
deep learning
with GPUs
2017

AlphaGo
2017
Project Debater

Project Debater is the first AI system that can debate humans on complex topics. The goal is to help people build persuasive arguments and make well-informed decisions.

Watch a live debate
Progress in Deep Learning

1997: IBM Deep Blue chess

2011: IBM Watson Jeopardy

2011: Apple’s releases Siri

2012: AlexNet

2015: Facebook’s face recognition

2016: Siri gets deep learning

2017: AlphaGo

...
Deep Learning = Training Artificial Neural Networks

- 25 million “neurons”
- 100 million connections (parameters)

A human brain has:

- 200 billion neurons
- 32 trillion connections between them
ML
Code

*Source: Hidden Technical Debt in Machine Learning Systems*
In reality …
Neural Network Design Workflow

1. Design neural network
2. Data
3. HPO
   - neural network structure
   - hyperparameters
4. Performance meets needs?
   - Optimal hyperparameters
   - Start another experiment
   - No
Neural Network Design Workflow

1. Design neural network
2. Collect domain data
3. Perform hyperparameter optimization (HPO)
   - neural network structure
   - hyperparameters
4. Evaluate performance
   - Performance meets needs?
   - Yes: Deploy trained model
      - Cloud
   - No: Start another experiment
     - BAD
     - Still good!
     - Optimal hyperparameters
5. Decide on next steps
   - Design neural network

Let’s understand it from the context of an AI Lifecycle
We need a Cloud native AI Platform to build, train, deploy and monitor Models.
Many tools available to build initial models
Neural Network Modeller within Watson Studio
An intuitive drag-and-drop, no-code interface for designing neural network structure

Real-time validation of network flow

Drag-and-drop network layers

- Customize layer by setting hyperparameters
- Generate CPU or GPU compatible code
- Save as popular framework code
- Export as a python notebook
- Execute as batch experiment

CODE PATTERN

Validate computer vision deep learning models

Compare inference results with ground truth test data to continuously evaluate model accuracy

by Mark Sturdivant | Updated June 17, 2019 - Published June 14, 2019

Many tools to train machine learning and deep learning models

- Jupyter
- R Studio
- SPSS Modeler
- IBM Watson Studio
- Anaconda
- TensorFlow
- PaddlePaddle
- PyTorch
- Caffe
- Keras
- XGBoost
- MxNet
Training is accomplished. Model is ready – Can we trust it?
What does it take to trust a decision made by a machine?
(Other than that it is 99% accurate)?

Is it fair?  
Is it easy to understand?  
Did anyone tamper with it?  
Is it accountable?
Our vision for Trusted AI
Pillars of trust, woven into the lifecycle of an AI application

FAIRNESS
EXPLAINABILITY
ROBUSTNESS
ASSURANCE

supported by an instrumented platform
AIOpenScale
So let`s start with vulnerability detection of Models?

Is the model vulnerable to adversarial attacks?
Enter: Adversarial Robustness Toolbox
IBM Adversarial Robustness Toolbox

https://github.com/IBM/adversarial-robustness-toolbox

ART is a library dedicated to adversarial machine learning. Its purpose is to allow rapid crafting and analysis of attack and defense methods for machine learning models. The Adversarial Robustness Toolbox provides an implementation for many state-of-the-art methods for attacking and defending classifiers.

The Adversarial Robustness Toolbox contains implementations of the following attacks:

- Deep Fool (Moosavi-Dezfooli et al., 2015)
- Fast Gradient Method (Goodfellow et al., 2014)
- Jacobian Saliency Map (Papernot et al., 2016)
- Universal Perturbation (Moosavi-Dezfooli et al., 2016)
- Virtual Adversarial Method (Moosavi-Dezfooli et al., 2015)
- C&W Attack (Carlini and Wagner, 2016)
- NewtonFool (Jang et al., 2017)

The following defense methods are also supported:

- Feature squeezing (Xu et al., 2017)
- Spatial smoothing (Xu et al., 2017)
- Label smoothing (Warde-Farley and Goodfellow, 2016)
- Adversarial training (Szegedy et al., 2013)
- Virtual adversarial training (Miyato et al., 2017)
Implementation for state-of-the-art methods for attacking and defending classifiers.

Evasion attacks
- FGSM
- JSMA
- BIM
- PGD
- Carlini & Wagner
- DeepFool
- NewtonFool
- Universal perturbation

Evasion defenses
- Feature squeezing
- Spatial smoothing
- Label smoothing
- Adversarial training
- Virtual adversarial training
- Thermometer encoding
- Gaussian data augmentation

Poisoning detection
- Detection based on clustering activations
- Proof of attack strategy

Evasion detection
- Detector based on inputs
- Detector based on activations

Robustness metrics
- CLEVER
- Empirical robustness
- Loss sensitivity

Unified model API
- Training
- Prediction
- Access to loss and prediction gradients
ART Demo: https://art-demo.mybluemix.net/
Integrate adversarial attacks in a model training pipeline

Use a Jupyter notebook to integrate the Adversarial Robustness Toolbox into a neural network model training pipeline to find model vulnerabilities

by Animesh Singh, Anupama Murthy, Christian Kadner | Published June 25, 2018

モデルのトレーニング・パイプラインに敵対者からの攻撃を統合する

モデルの脆弱性を見つけるために、Jupyter Notebook を使用してニューラル・ネットワーク・モデルのトレーニング・パイプラインに Adversarial Robustness Toolbox を統合する

Robustness check accomplished. How do we check for bias throughout lifecycle?

- Is the dataset biased?
- Are predictions biased?
- Are model weights biased?

Steps:
1. Create Model
2. Prepare and Analyze Data
3. Assign Hyperparameters and Train
4. Monitor
5. Validate and Deploy
6. Trained Model
7. Deployed Model
8. Initial Model

Questions:
- Are model weights biased?
- Are predictions biased?
- Is the dataset biased?
Unwanted bias and algorithmic fairness
Machine learning, by its very nature, is always a form of statistical discrimination

Discrimination becomes objectionable when it places certain privileged groups at systematic advantage and certain unprivileged groups at systematic disadvantage

Illegal in certain contexts
Unwanted bias and algorithmic fairness

Machine learning, by its very nature, is always a form of statistical discrimination.

Unwanted bias in training data yields models with unwanted bias that scale out.

Prejudice in labels

Undersampling or oversampling
Google apologizes for mis-tagging photos of African Americans

Google was quick to respond over the weekend to a user after Google Photos app had mis-categorized a photo of him and 1 and offensive way.

Amazon scraps secret AI recruiting tool that showed bias against women

SAN FRANCISCO (Reuters) - Amazon.com Inc's (AMZN.O) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.
Google Photosが黒人をゴリラと認識した事件で開発者が謝罪

アマゾンの採用AIツール、女性差別でシャットダウン

Isobel Asher Hamilton
Oct. 15, 2018, 05:30 AM | TECH INSIDER | 6,121
AIF360 toolkit is an open-source library to help detect and remove bias in machine learning models.

The AI Fairness 360 Python package includes a comprehensive set of metrics for datasets and models to test for biases, explanations for these metrics, and algorithms to mitigate bias in datasets and models.

**Toolbox**

- Fairness metrics (30+)
- Fairness metric explanations
- Bias mitigation algorithms (10+)

**Supported bias mitigation algorithms**

- Optimized Preprocessing (Calmon et al., 2017)
- Disparate Impact Remover (Feldman et al., 2015)
- Equalized Odds Postprocessing (Hardt et al., 2016)
- Reweighing (Kamiran and Calders, 2012)
- Reject Option Classification (Kamiran et al., 2012)
- Prejudice Remover Regularizer (Kamishima et al., 2012)
- Calibrated Equalized Odds Postprocessing (Pleiss et al., 2017)
- Learning Fair Representations (Zemel et al., 2013)
- Adversarial Debiasing (Zhang et al., 2018)
- Meta-Algorithm for Fair Classification (Celis et al., 2018)

**Supported fairness metrics**

- Comprehensive set of group fairness metrics derived from selection rates and error rates
- Comprehensive set of sample distortion metrics
- Generalized Entropy Index (Speicher et al., 2018)
AI Fairness 360 Open Source Toolkit

This extensible open source toolkit can help you examine, report, and mitigate discrimination and bias in machine learning models throughout the AI application lifecycle. Containing over 70 fairness metrics and 10 state-of-the-art bias mitigation algorithms developed by the research community, it is designed to translate algorithmic research from the lab into the actual practice of domains as wide-ranging as finance, human capital management, healthcare, and education. We invite you to use it and improve it.

Not sure what to do first? Start here!

Read More
Learn more about fairness and bias mitigation concepts, terminology, and tools before you begin.

Try a Web Demo
Step through the process of checking and remediating bias in an interactive web demo that shows a sample of capabilities available in this toolkit.

Watch Videos
Watch videos to learn more about AI Fairness 360.

Read a paper
Read a paper describing how we designed AI Fairness 360.

Use Tutorials
Step through a set of in-depth examples that introduces developers to code that checks and mitigates bias in different industry and application domains.

Ask a Question
Join our AIF360 Slack Channel to ask questions, make comments and tell stories about how you use the toolkit.

View Notebooks
Open a directory of Jupyter Notebooks in GitHub that provide working examples of bias detection and mitigation in sample datasets. Then share your own notebooks!

Contribute
You can add new metrics and algorithms in GitHub. Share Jupyter notebooks showcasing how you have examined and mitigated bias in your machine learning applications.
Ensure loan fairness

Get a demonstration of the AI Fairness 360 toolkit for bias metrics, explanations, and remediation

by Michael Hind, Karthikeyan Natesan Ramamurthy | Published September 19, 2018

Summary

Model is trained, tested and validated. Then we can deploy it. Do we need anything else?

How can I monitor and trace the model?

How can I manage multiple version of my model to enable dark launch, A/B test and traffic shift easily?

How can I add and enforce policies to the model easily?
Istio
An open service mesh platform to connect, observe, secure, and control microservices.

Connect: Traffic Control, Discovery, Load Balancing, Resiliency

Observe: Metrics, Logging, Tracing

Secure: Encryption (TLS), Authentication, and Authorization of service-to-service communication

Control: Policy Enforcement
Manage microservices traffic using Istio

Enable your microservices with advanced traffic management and request tracing capabilities using Istio

So AI in general and Deep Learning in particular are very iterative and repetitive. And they need Cloud. Why?
AI requires the strength of HPC & GPUs
Ability to scale AI workloads on demand
Ability to utilize various technologies and achieve high performance computing.

1. Model/Data Parallelism

2. MPI/NCCL

NCCL (pronounced "Nickel") is a stand-alone library of standard collective communication routines for GPUs, implementing all-reduce, all-gather, reduce, broadcast, and reduce-scatter. It has been optimized to achieve high bandwidth on platforms using PCIe, NVLink, NVswitch, as well as networking using InfiniBand Verbs or TCP/IP sockets. NCCL supports an arbitrary number of GPUs installed in a single node or across multiple nodes, and can be used in either single- or multi-process (e.g., MPI) applications.
To scale we need to go Cloud native for AI
Access to elastic compute leveraging Kubernetes

Auto-allocation means infrastructure is used only when needed

Source code

Training definition

Training artifacts

Kubernetes container

TensorFlow, PyTorch, Caffe, Keras

Compute cluster

NVIDIA Tesla K80, P100, V100

Training assets are managed and tracked.

Cloud Object Storage
Model training distributed across containers

- NVIDIA GPUs
- Kubernetes
- container orchestration
- training runs
- containers
- server cluster
- dataset
- Cloud Object Storage
Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.
Oh, you want to use ML on K8s?

First, can you become an expert in ...

- Containers
- Packaging
- Kubernetes service endpoints
- Persistent volumes
- Scaling
- Immutable deployments
- GPUs, Drivers & the GPL
- Cloud APIs
- DevOps
- ...

Source: kubeCon Barcelona 2019
Numpy
Jupyter
TF.Transform
TF.Estimator
Docker
Seldon

Experiment Tracking
Declarative Deployment
HP Tuning
Profiling
Validation

Resource
Scheduling
Access
Drivers
Orchestration
Lifecycle
Networking

Source: kubeCon Barcelona 2019
Kubeflow architecture

- Make it super easy to deploy and administer a platform
  - Leverage KF & non KF components
- Tie it together using
  - Orchestration
    - Combine components into complex workflows
  - Metadata
    - Collect data from multiple components
Getting Started

- **Getting started with Kubeflow**
  Quickly get running with your ML workflow on an existing Kubernetes installation

- **Microk8s for Kubeflow**
  Quickly get Kubeflow running locally on native hypervisors

- **Minikube for Kubeflow**
  Quickly get Kubeflow running locally

- **Kubernetes Engine for Kubeflow**
  Get Kubeflow running on Google Cloud Platform. This guide is a quickstart to deploying Kubeflow on Google Kubernetes Engine

- **Requirements for Kubeflow**
  Get more detailed information about using Kubeflow and its components
@dsl.pipeline(
    name='Object detection',
    description='Object detection'
)
def object_detection(worker=3):
    getData = get_data()
    pre_process = pre_process(getData.output)
    hpo = hyperparameter_tune(pre_process.output)
    train = start_train(hpo.output, worker)
    r_check = robustness_check(train.output)
    f_check = fairness_check(train.output)
    deploy = deploy_model(r_check.output, f_check.output)

# dsl-compile --py object_detection.py --output object_detection.tgz
Kubeflow pipeline

- **Pipelines**
  - Experiments
  - Archive
  - Pipeline name
    - object detection
    - [Sample] Basic - Condition
    - [Sample] Basic - Exit Handler
    - [Sample] Basic - Immediate Value
    - [Sample] Basic - Parallel Join
    - [Sample] Basic - Sequential
    - [Sample] ML - TFX - Taxi Tip Prediction Model Trainer
    - [Sample] ML - XGBoost - Training with Confusion Matrix

- **Graph**
  - get-data
  - pre-process
  - hyperparameter-tune
  - train
  - fairness-check
  - robustness-check
  - deploy-model
Overview of the Kubeflow pipelines service

Kubeflow is a machine learning (ML) toolkit that is dedicated to making deployments of ML workflows on Kubernetes simple, portable, and scalable.

Kubeflow pipelines are reusable end-to-end ML workflows built using the Kubeflow Pipelines SDK.

The Kubeflow pipelines service has the following goals:

• End to end orchestration: enabling and simplifying the orchestration of end to end machine learning pipelines
• Easy experimentation: making it easy for you to try numerous ideas and techniques, and manage your various trials/experiments.
• Easy re-use: enabling you to re-use components and pipelines to quickly cobble together end to end solutions, without having to re-build each time.

Documentation

Get started with your first pipeline and read further information in the Kubeflow Pipelines overview.

See the various ways you can use the Kubeflow Pipelines SDK.

See the Kubeflow Pipelines API doc for API specification.

Consult the Python SDK reference docs when writing pipelines using the Python SDK.

Blog posts

• Getting started with Kubeflow Pipelines (By Amy Unruh)
• How to create and deploy a Kubeflow Machine Learning Pipeline (By Lak Lakshmanan)
  • Part 1: How to create and deploy a Kubeflow Machine Learning Pipeline
  • Part 2: How to deploy data scientists and developers on Kubernetes using Kubeflow and ML pipelines

Acknowledgments

This Demo will go over how to leverage Kubeflow Pipeline into the AI LifeCycle
Get Kubeflow up and running on a private cloud

Create a portable and scalable on-premises solution for enterprises that need to protect data.

Winnie Tsang, Raymond Wong | Updated September 20, 2018 - Published September 19, 2018

Today more and more companies use artificial intelligence (AI) to improve the user experiences for their products. These enterprises have the following goals:

Infact, we need a transparent, trusted and **automated AI Pipeline**

- Has the training data changed?
- Is the dataset biased?
- Are predictions biased?
- Are model predictions less accurate?
- Are model weights biased?
- Is model training showing increasing loss?
- Are Hyperparameters suboptimal?
- Is the model vulnerable to adversarial attacks?
Transparent, trusted, automated, event driven and auditable AI Pipeline

- **Prepare Data**
  - Train
  - Deploy
- **Optimize Hyperparameters**
  - Train
- **Implement Defense**
  - Train
  - Deploy
- **Harden**
  - Train
  - Deploy

- **Trigger**: Trained Model is showing increasing loss
- **Trigger**: Data changed
- **Trigger**: Bias Detected
- **Trigger**: Model is vulnerable to attack
- **Trigger**: Model performance is suboptimal

- **Actions**

- **Initial Model**
- **Assign Hyperparameters and Train**
- **Create Model**
- **Monitor**
- **Validate and Deploy**
- **Trained Model**
- **Deployed Model**
Trigger: Trained Model is showing increasing loss

Prepare Data
Train
Deploy

Create Model
Assign Hyperparameters
and Train

Monitor

Validate and Deploy

Initial Model

Trained Model

Deployed Model

Trigger: Data changed

Trigger: Bias Detected

Actions

Trigger: Model is vulnerable to attack

Trigger: Model performance is biased

Optimize Hyperparameters
Train

Implement Defense
Train
Deploy

Debias
Train
Deploy

Prepare and Analyzed Data

Transparent, trusted, automated, event driven, auditable AI Pipeline as a Service
Build
Provides easy-to-use, simple source-to-container builds, so you can focus on writing code and know how to build it. Knative solves for the common challenges of building containers and runs it on cluster.

Serving
Run serverless containers on Kubernetes with ease, Knative takes care of the details of networking, autoscaling (even to zero), and revision tracking. You just have to focus on your core logic.

Eventing
Universal subscription, delivery, and management of events. Build modern apps by attaching compute to a data stream with declarative event connectivity and developer-friendly object model.
**Build** — Source-to-container build orchestration

**Knative Build Components**

- **Build**
- **Builder**
- **BuildTemplate**

For example, you can write a build that uses Kubernetes-native resources to obtain your source code from a repository, build a container image, then run that image.

- A Build can include multiple steps where each step specifies a Builder.
- A Builder is a type of container image that you create to accomplish any task, whether that's a single step in a process, or the whole process itself.
- The steps in a Build can push to a repository.
- A BuildTemplate can be used to defined reusable parameterized templates.
Knative serving

**Serving** — Request-driven compute model, scale to zero, autoscaling, routing and managing traffic

**Knative Serving components**

- **Configuration**
  - Desired current state of deployment (#HEAD)
  - Records both code and configuration (separated, ala 12 factor)
  - Stamps out builds / revisions as it is updated

- **Revision**
  - Code and configuration snapshot
  - k8s infra: Deployment, ReplicaSet, Pods, etc

- **Route**
  - Traffic assignment to Revisions (fractional scaling or by name)
  - Built using Istio

- **Service**
  - Provides a simple entry point for UI and CLI tooling to achieve common behavior
  - Acts as a top-level controller to orchestrate Route and Configuration.
**Knative eventing**

**Broker** and **Trigger** are CRDs providing an event delivery mechanism that hides the details of event routing from the event producer and event consumer.

The **Event Registry** maintains a catalog of the event types that can be consumed from the different Brokers.

**Event Sources** are Kubernetes Custom Resources which provide a mechanism for registering interest in a class of events from a particular software system.

**Channels** are Kubernetes Custom Resources which define a single event forwarding and persistence layer.
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Support Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS SQS</td>
<td>Proof of Concept</td>
<td>Brings AWS Simple Queue Service messages into Knative.</td>
</tr>
<tr>
<td>Apache Camel</td>
<td>Proof of Concept</td>
<td>Allows to use Apache Camel components for pushing events into Knative.</td>
</tr>
<tr>
<td>Apache Kafka</td>
<td>Proof of Concept</td>
<td>Brings Apache Kafka messages into Knative.</td>
</tr>
<tr>
<td>BitBucket</td>
<td>Proof of Concept</td>
<td>Registers for events of the specified types on the specified BitBucket organization/repository. Brings those events into Knative.</td>
</tr>
<tr>
<td>Cron Job</td>
<td>Proof of Concept</td>
<td>Uses an in-memory timer to produce events on the specified Cron schedule.</td>
</tr>
<tr>
<td>GCP PubSub</td>
<td>Proof of Concept</td>
<td>Brings GCP PubSub messages into Knative.</td>
</tr>
<tr>
<td>GitHub</td>
<td>Proof of Concept</td>
<td>Registers for events of the specified types on the specified GitHub organization/repository. Brings those events into Knative.</td>
</tr>
<tr>
<td>GitLab</td>
<td>Proof of Concept</td>
<td>Registers for events of the specified types on the specified GitLab repository. Brings those events into Knative.</td>
</tr>
<tr>
<td>Google Cloud Scheduler</td>
<td>Active Development</td>
<td>Create, update, and delete Google Cloud Scheduler Jobs. When those jobs are triggered, receive the event inside Knative.</td>
</tr>
<tr>
<td>Google Cloud Storage</td>
<td>Active Development</td>
<td>Registers for events of the specified types on the specified Google Cloud Storage bucket and optional object prefix. Brings those events into Knative.</td>
</tr>
<tr>
<td>Kubernetes Api Server</td>
<td>Active Development</td>
<td>Brings Kubernetes resource changes into Knative as references or as full resources.</td>
</tr>
</tbody>
</table>
Event Driven ML pipeline

def object_detection(
    worker=3,
    new_image_name="hougangliu/object_detection:latest"):
    getData = get_data()
    pre_process = pre_process(getData.output)
    new_image = build_image(new_image_name)
    hpo = hyperparameter_tune(pre_process.output, new_image.output)
    train = start_train(hpo.output, new_image.output, worker)
    r_check = robustness_check(train.output)
    f_check = fairness_check(train.output)
    deploy = deploy_model(r_check.output, f_check.output)

---

```yaml
apiVersion: build.knative.dev/v1alpha1
kind: Build
metadata:
  name: build-objective-detection
spec:
  serviceAccountName: build-auth
  source:
    git:
      url: https://github.com/hougangliu/object_detection.git
      revision: master
  steps:
  - image: hougangliu/image-build:latest
    args: ["make", "build"]
    name: build-image
  - image: hougangliu/image-push:latest
    args: ["make", "push"]
    name: push-image
  volumeMounts:
  - name: docker-socket-example
    mountPath: /var/run/docker.sock
```

---

Graph:

```
get-data
  |> pre-process
  |> build-new-image
  |> hyperparameter-tune
  |> train
  |- robustness-check
  |- fairness-check
  |   |- deploy-model
```
Deploy a Knative application using Tekton Pipelines

Learn how to use the Tekton Pipelines open source project to build and deploy a Knative app

Gregory Dritschler | Published June 5, 2019

Tekton Pipelines is an open source project to configure and run continuous integration and continuous delivery (CI/CD) pipelines within a Kubernetes cluster. In this tutorial you learn the following concepts and skills:

- The basic concepts used in the Tekton Pipelines project
- Examples of creating a pipeline to build and deploy a Knative application

Introducing Data Asset eXchange (DAX)

The challenge: Data is the fuel for AI, but data quality, licensing, and format vary significantly

In support of open data, IBM announced the Data Asset eXchange (DAX), a place to find curated free and open datasets under open data licenses

- Standardized dataset formats and metadata
- Ready for use in enterprise AI applications
- Complement to the Model Asset eXchange (MAX)

ibm.biz/data-asset-exchange
The Community Data License Agreement
http://cdla.io

- Linux Foundation initiative to create a new legal framework that meets the needs of data licensing
- Enables collaboration in data much like open source licenses enable community collaboration
- IBM is a major supporter
- When possible, CDLA will be used for DAX datasets
Together: A Transparent, and trusted event driven Open Source AI Pipeline
Build an Event Driven Machine Learning Pipeline on Kubernetes

THANKS