Infraestrutura de ML com Kubeflow

Agenda

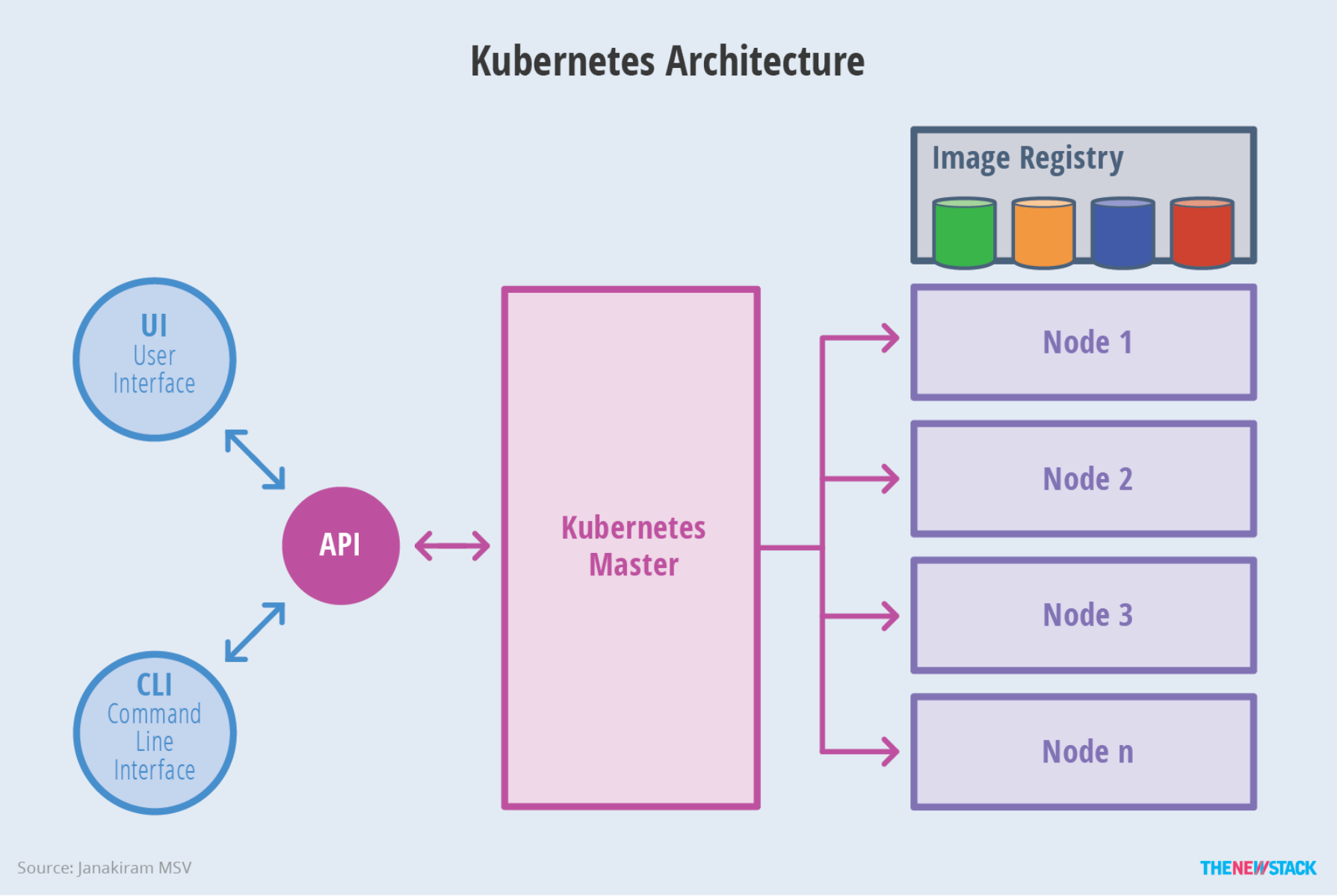
1. Kubernetes
2. Kubeflow
3. Kubeflow Pipelines
4. KFServing

Kubernetes

KUBERNETES

Introdução

Kubernetes (K8s) é um sistema open-source para automatizar a implantação, escalonamento e gerenciamento de aplicações em contêiners.


Kubernetes Architecture
Image Registry
UI
User
Interface
API
CLI
Command
Line
Interface
Kubernetes
Master
Node 1
Node 2
Node 3
Node n
Source: Janakiram MSV
THENEWSTACK

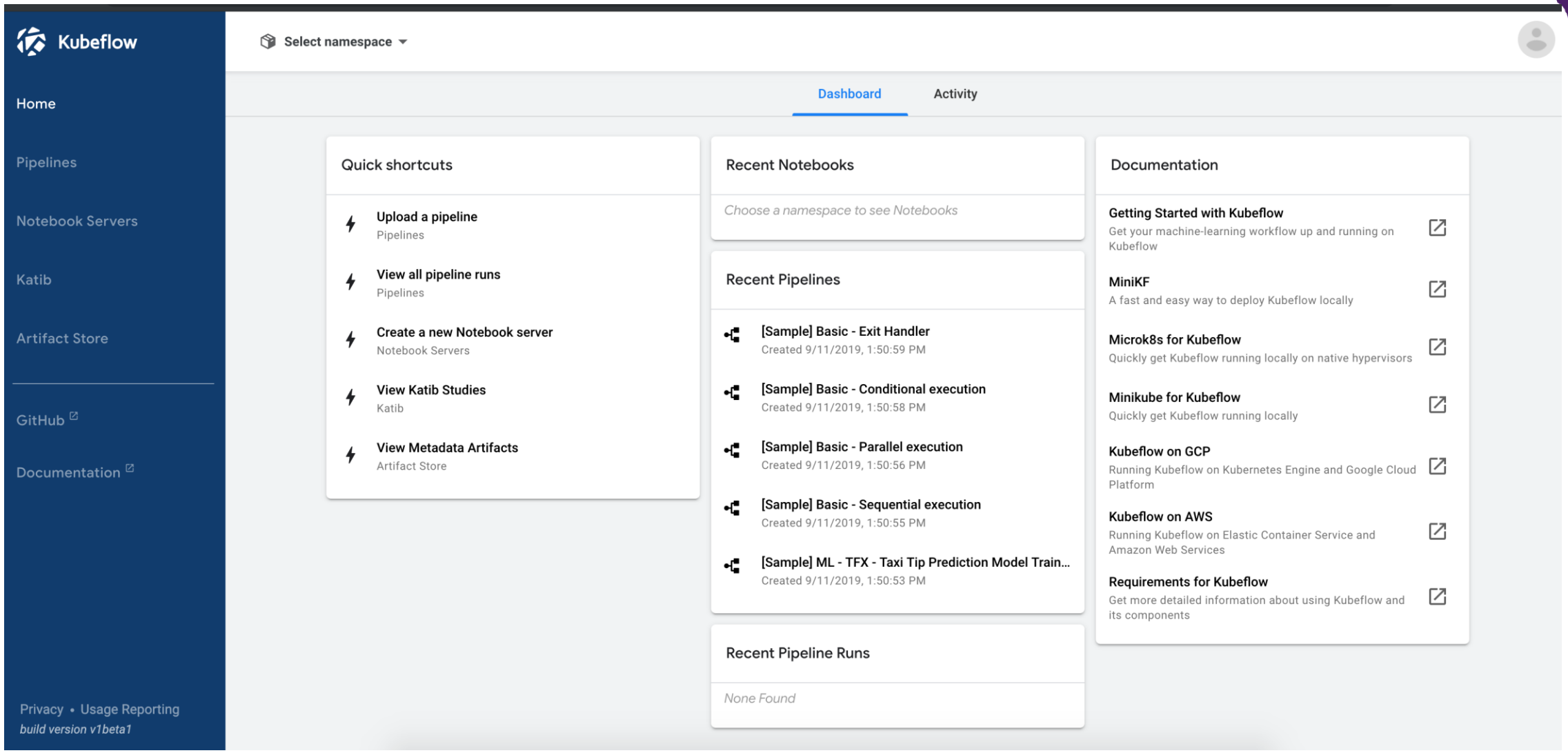
Kubeflow

KUBEFLOW

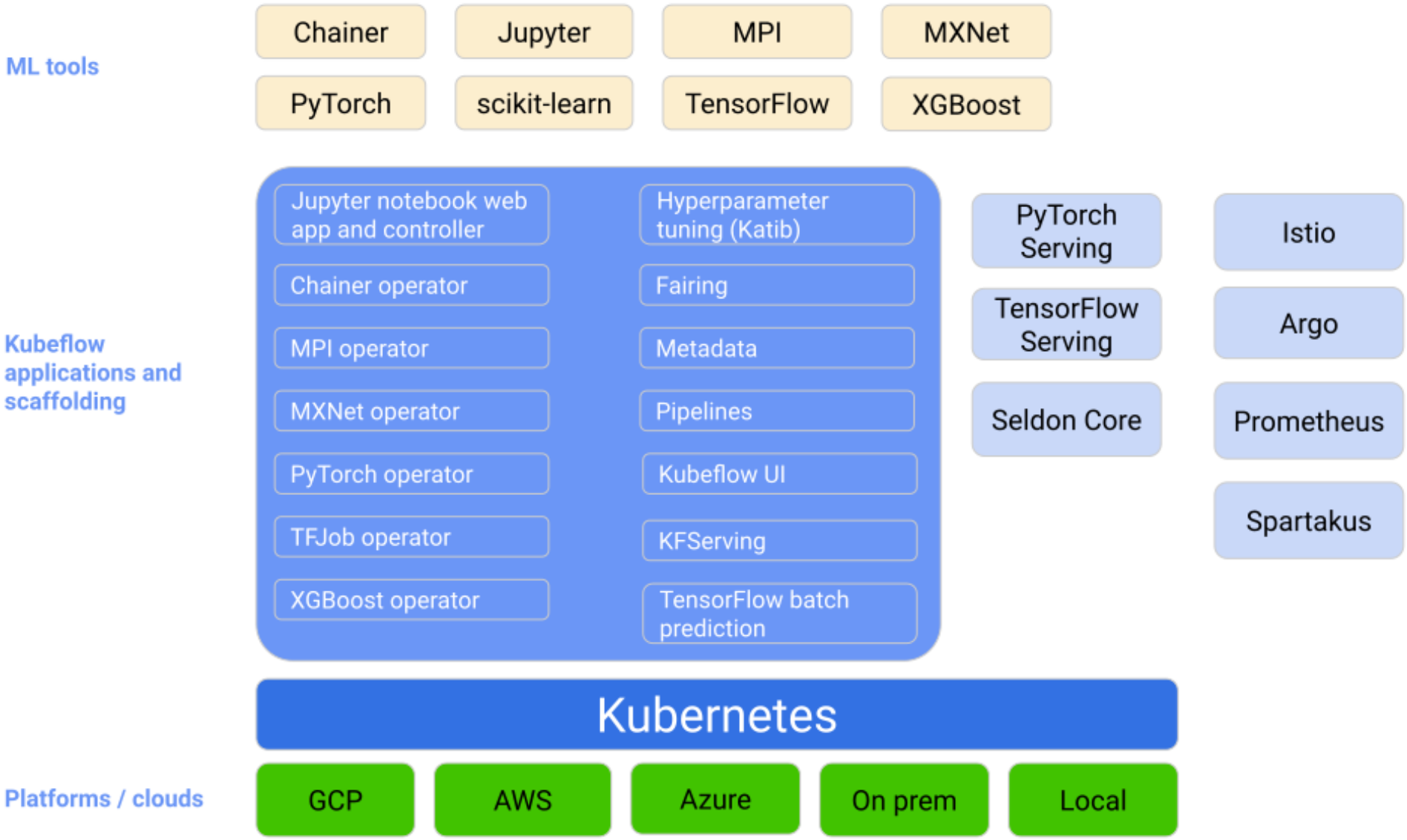
Introdução

Kubeflow é um toolkit de ML para Kubernetes. Ele foi feito para facilitar a implantação de workflows de ML em um cluster Kubernetes mais simples, portável e escalável.

KUBEFLOW


Kubeflow
Select namespace
Home
Pipelines
Notebook Servers
Katib
Artifact Store
GitHub
Documentation
Privacy • Usage Reporting
build version v1beta1
Dashboard
Activity
Quick shortcuts
Upload a pipeline
Pipelines
View all pipeline runs
Pipelines
Create a new Notebook server
Notebook Servers
View Katib Studies
Katib
View Metadata Artifacts
Artifact Store
Recent Notebooks
Choose a namespace to see Notebooks
Recent Pipelines
[Sample] Basic - Exit Handler
Created 9/11/2019, 1:50:59 PM
[Sample] Basic - Conditional execution
Created 9/11/2019, 1:50:58 PM
[Sample] Basic - Parallel execution
Created 9/11/2019, 1:50:56 PM
[Sample] Basic - Sequential execution
Created 9/11/2019, 1:50:55 PM
[Sample] ML - TFX - Taxi Tip Prediction Model Train...
Created 9/11/2019, 1:50:53 PM
Recent Pipeline Runs
None Found
Documentation
Getting Started with Kubeflow
Get your machine-learning workflow up and running on Kubeflow
MiniKF
A fast and easy way to deploy Kubeflow locally
Microk8s for Kubeflow
Quickly get Kubeflow running locally on native hypervisors
Minikube for Kubeflow
Quickly get Kubeflow running locally
Kubeflow on GCP
Running Kubeflow on Kubernetes Engine and Google Cloud Platform
Kubeflow on AWS
Running Kubeflow on Elastic Container Service and Amazon Web Services
Requirements for Kubeflow
Get more detailed information about using Kubeflow and its components

KUBEFLOW


ML tools
Chainer
Jupyter
MPI
MXNet
PyTorch
scikit-learn
TensorFlow
XGBoost
Kubeflow applications and scaffolding
Jupyter notebook web app and controller
Hyperparameter tuning (Katib)
Chainer operator
Fairing
MPI operator
Metadata
MXNet operator
Pipelines
PyTorch operator
Kubeflow UI
TFJob operator
KFServing
XGBoost operator
TensorFlow batch prediction
PyTorch Serving
TensorFlow Serving
Seldon Core
Istio
Argo
Prometheus
Spartakus
Kubernetes
Platforms / clouds
GCP
AWS
Azure
On prem
Local

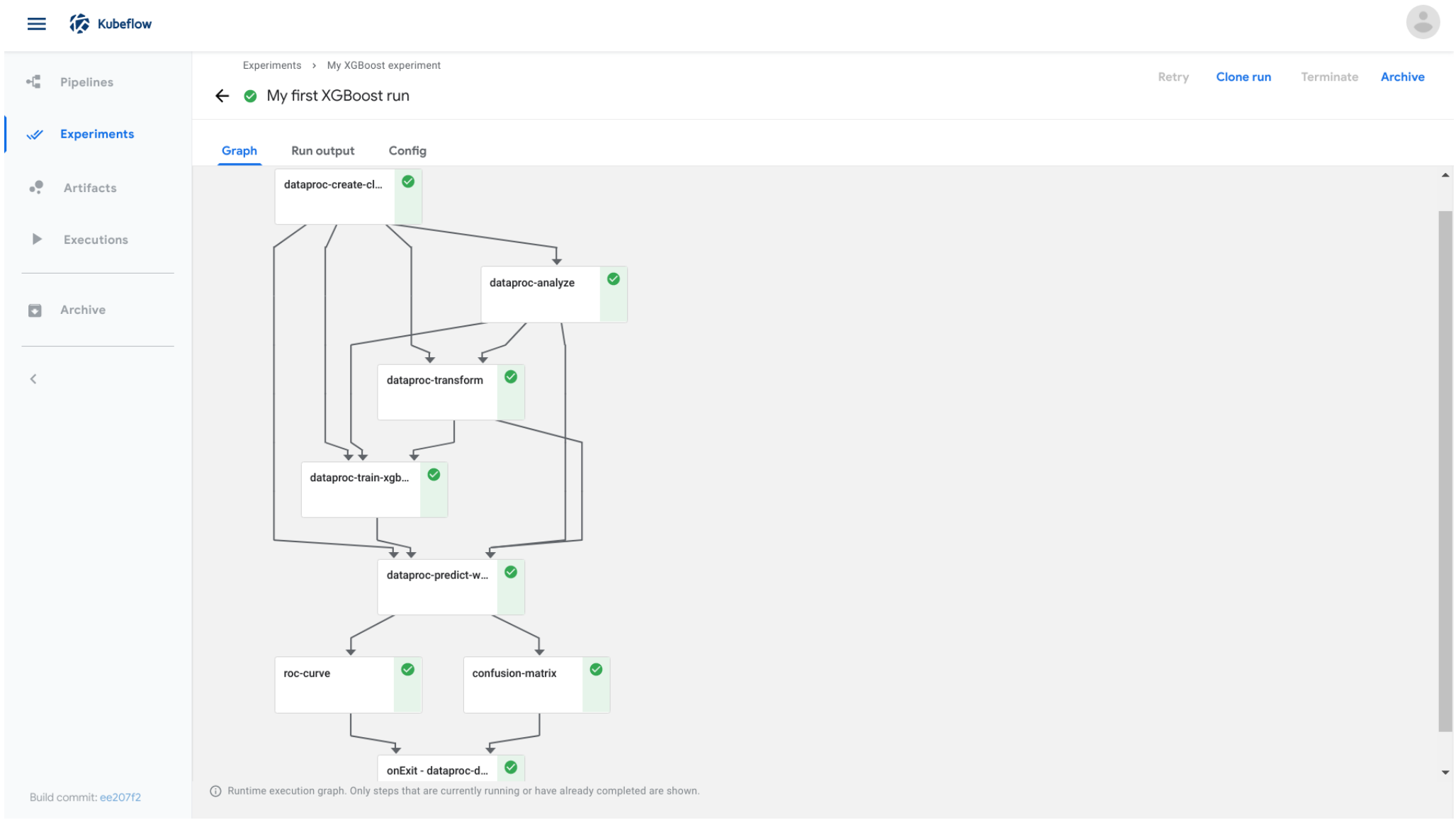
Kubeflow Pipelines

KUBEFLOW PIPELINES

Introdução

Kubeflow Pipelines é uma plataforma para construir e implantar workflows de ML end-to-end de forma portável e escalável, baseada em contêineres.

KUBEFLOW PIPELINES



KUBEFLOW PIPELINES

Desenvolvimento

- O desenvolvimento é feito em python
- Cada etapa do pipeline é feita através de um componente
- Os componentes tem uma imagem e rodam dentro do cluster
- O pipeline é compilado e é feito o upload para o Kubeflow

KUBEFLOW PIPELINES

```python
import kfp
from kfp import dsl

def echo_op():
    return dsl.ContainerOp(
        name='echo',
        image='library/bash:4.4.23',
        command=['sh', '-c'],
        arguments=['echo "hello world"']
    )

@dsl.pipeline(
    name='My first pipeline',
    description='A hello world pipeline.'
)
def hello_world_pipeline():
    echo_task = echo_op()

if __name__ == '__main__':
    kfp.compiler.Compiler().compile(hello_world_pipeline, __file__ + '.yaml')
```

KUBEFLOW PIPELINES

```
diagnose_me_op = components.load_component_from_url(

'https://raw.githubusercontent.com/kubeflow/pipelines/566dddfdfc0a6a725b6e50ea85e73d8d5578bbb9/
components/diagnostics/diagnose_me/component.yaml')
```

KUBEFLOW PIPELINES

```
_diagnose_me_op = diagnose_me_op(
    bucket=output,
    execution_mode=diagnostic_mode,
    project_id=project,
    target_apis=required_apis,
    quota_check=quota_check)
```

KUBEFLOW PIPELINES

```
import kfp.dsl as dsl
gpu_op = dsl.ContainerOp(name='gpu-op', ...).set_gpu_limit(2)
```

KUBEFLOW PIPELINES

```
name: xgboost4j - Train classifier
description: Trains a boosted tree ensemble classifier using xgboost4j

inputs:
- {name: Training data}
- {name: Rounds, type: Integer, default: '30', help: Number of training rounds}

outputs:
- {name: Trained model, type: XGBoost model, help: Trained XGBoost model}

implementation:
  container:
    image: gcr.io/ml-pipeline/xgboost-classifier-train@sha256:b3a64d57
    command: [
      /ml/train.py,
      --train-set, {inputPath: Training data},
      --rounds,    {inputValue: Rounds},
      --out-model, {outputPath: Trained model},
    ]
```

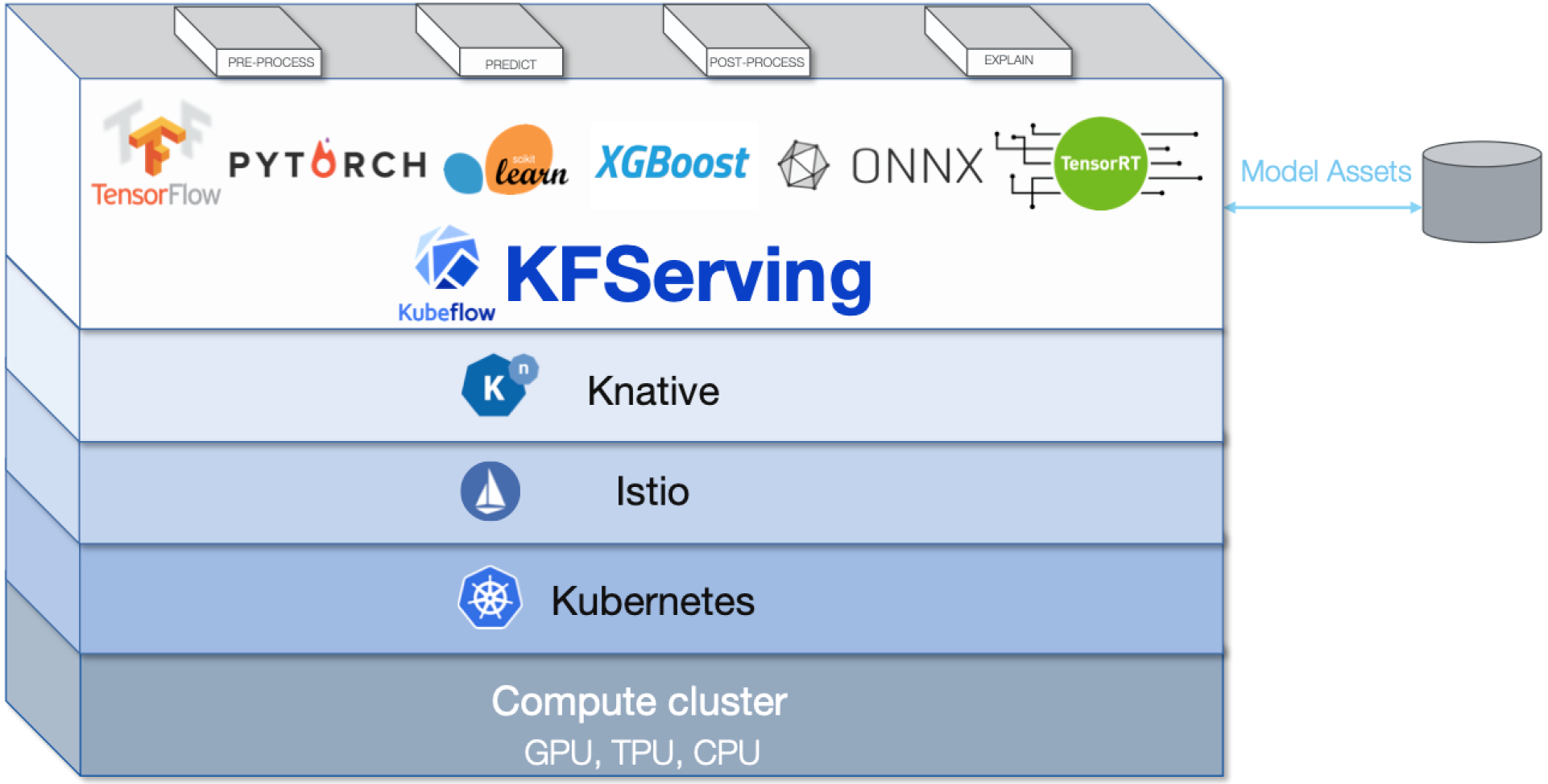
KFServing

KFSERVING

Introdução

O KFServing foi feito para facilitar o deploy de modelos de diversos frameworks (Tensorflow, XGBoost, scikit-learn, PyTorch e ONNX)

KFSERVING


PRE-PROCESS
PREDICT
POST-PROCESS
EXPLAIN
TensorFlow
PYTORCH
scikit learn
XGBoost
ONNX
TensorRT
Model Assets
Kubeflow
KFServing
Knative
Istio
Kubernetes
Compute cluster
GPU, TPU, CPU

KFSERVING

```yaml
apiVersion: "serving.kubeflow.org/v1alpha2"
kind: "InferenceService"
metadata:
  name: "sklearn-iris"
spec:
  default:
    predictor:
      sklearn:
        storageUri: "gs://kfserving-samples/models/sklearn/iris"
```

```
kubectl apply -f sklearn.yaml
```

KFSERVING

```
def kfservingPipeline(
    action = 'create',
    model_name='tensorflow-sample',
    default_model_uri='gs://kfserving-samples/models/tensorflow/flowers',
    canary_model_uri='gs://kfserving-samples/models/tensorflow/flowers-2',
    canary_model_traffic_percentage='10',
    namespace='kubeflow',
    framework='tensorflow',
    default_custom_model_spec='{}',
    canary_custom_model_spec='{}',
    autoscaling_target=0,
    kfserving_endpoint=''
):

    # define workflow
    kfserving = kfserving_op(action = action,
                             model_name=model_name,
                             default_model_uri=default_model_uri,
                             canary_model_uri=canary_model_uri,
                             canary_model_traffic_percentage=canary_model_traffic_percentage,
                             namespace=namespace,
                             framework=framework,
                             default_custom_model_spec=default_custom_model_spec,
                             canary_custom_model_spec=canary_custom_model_spec,
                             autoscaling_target=autoscaling_target,
                             kfserving_endpoint=kfserving_endpoint)
```

MUITO OBRIGADO!

Marcelo Pio

www.dtidigital.com.br