



Kubeflow: Machine Learning em escala.

Marcio Junior Vieira
CEO & Data Scientist, Ambiente Livre
Pesquisador da UFG.

Mini-CV

- 25 anos de experiência em TI, vivência em desenvolvimento e análise de sistemas, gestão empresarial e ciência de dados.
- CEO da Ambiente Livre atuando como Cientista de Dados, Engenheiro de Dados e Arquiteto de Software.
- Já lecionou nos MBAs em Big Data & Data Science, Inteligência Artificial e Business Intelligence e Analytics da Universidade Positivo e MBA Artificial Intelligence e Machine Learning da FIAP.
- Trabalhando com Free Software e Open Source desde 2000 com serviços de consultoria e treinamento.
- Graduado em Tecnologia em Informática(2004) e pós-graduado em Software Livre(2005) ambos pela UFPR.
- Pesquisador pelo UFG/CIAP (Centro de Colaboração Interinstitucional de Inteligência Artificial Aplicada às Políticas Públicas).
- Atuou com Pesquisador do Laboratório de Tecnologias para Tomada de Decisão da Universidade de Brasília (Unb/Latitude).
- Palestrante FLOSS em: FISL, TDC, Latinoware, Campus Party, Pentaho Day, Ticonova, PgDay e FTSL.
- Organizador Geral: Pentaho Day 2017, 2015, 2019 e apoio nas ed. 2013 e 2014.
- Data Scientist, instrutor e consultor de Big Data e Data Science com tecnologias abertas.
- Ajudou a capacitar equipes de Big Data na IBM, Accenture, Tivit, Serpro, Natura, MP, Netshoes, Embraer entre outras.
- Especialista em implantação e customização de Big Data com Hadoop, Spark, Pentaho, Cassandra e MongoDB.
- Contribuidor de projetos internacionais, tais como Pentaho, LimeSurvey, SuiteCRM e Camunda.
- Especialista em implantação e customização de ECM com Alfresco e BPM com Activiti, Flowable e Camunda.
- Certificado (Certified Pentaho Solutions) pela Hitachi Vantara (Pentaho).

Nosso Ecossistema de Serviços

Data Driven	CRM, CMS e ITSM	ECM e BPM	Infra - IAC - DevOps
<p>Painéis de Indicadores Cubos e Relatórios Análise Preditiva Processamento Distribuído Banco de Dados Colunares</p> <p>Dashboards e OLAP Data Integration e Data Mining Big Data & Data Lake Machine Learning Business Intelligence & Analytics</p> <p>Consultoria Treinamento Projeto</p>	<p>Help Desk e Inventory Pesquisas Online Marketing e Vendas SAC e Pós-vendas Portais de Conteúdo</p> <p>IT Service Management Customer Relationship Management Content Management System Content Management Framework EAD e LMS</p> <p>Consultoria Treinamento Projeto</p>	<p>Gestão de Documentos Gerenciamento de Mídias Processo de Negócio BPMN e BPMS Microserviços</p> <p>Enterprise Content Management Records Management Business Process Management Microservices Orchestration</p> <p>Consultoria Treinamento Projeto</p>	<p>DepOps DevSecOps MLOps e DataOps Native Cloud Distributed Systems</p> <p>Web Server Kubernetes-as-a-Service Object Storage Containers Building Blocks</p> <p>Consultoria Treinamento Projeto</p>



Conceitos.

- Projeto Open Source.
- Específico para Machine Learning (ML).
- Torça fácil o desenvolvimento, implantação e gerenciamento de ML.
- Portátil e escalável.
- Foi construído sobre o Kubernetes.
- Microsserviços de ML.
- ML + K8s
- Os 3 Princípios básicos do Kubeflow:
 - * Capacidade de composição.
 - * Portabilidade.
 - * Escalabilidade.
- Lançado em 2017 pela Google (gerenciar tensorflow)

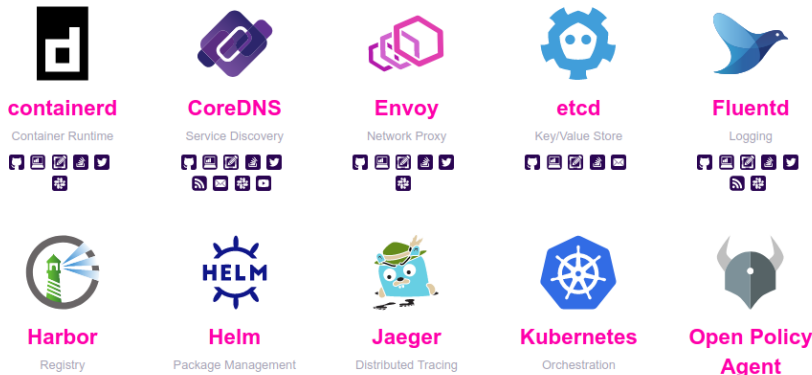


Kubeflow

CNCF

- Fundação que hospeda componentes críticos da infraestrutura de tecnologia global.
- Reúne os principais desenvolvedores, usuários finais e fornecedores do mundo e executa as maiores conferências de desenvolvedores de código aberto.
- Faz parte da organização sem fins lucrativos Linux Foundation.
- Criada da parceria da Google com a Linux Foundation.
- Google ofereceu o Kubernetes como uma tecnologia base.
- 286 mil contribuidores.
- 720 instituições membras.
- 140 Distribuições e plataformas certificadas.
- 163 mil membros (Meetups).
- 31 Projetos graduados 36 incubados e 143 sandbox.

Alguns projetos graduados da CNCF



Kubeflow Componentes



Pipelines



Notebooks



Dashboard



AutoML



Model Training

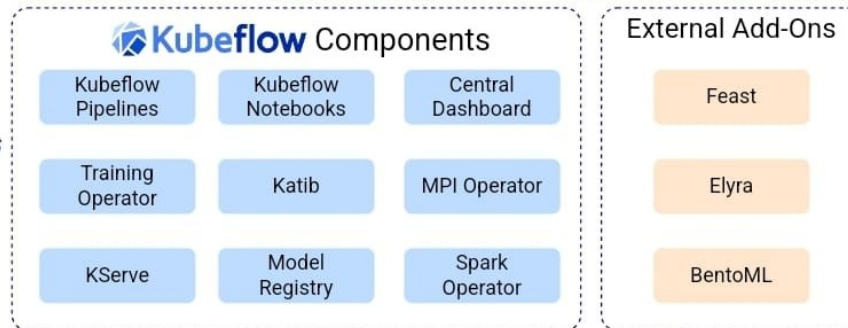
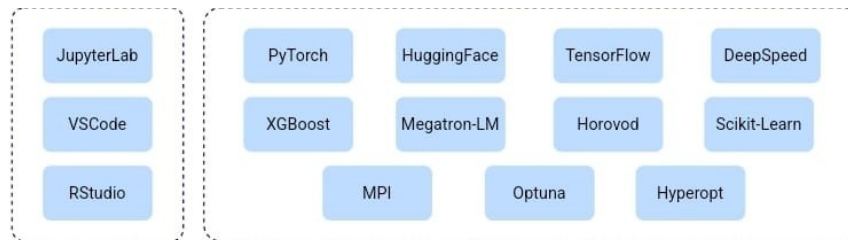


Model Serving

Kubeflow Ecosystem

Integrations

*Kubeflow Components
and
External Add-Ons*



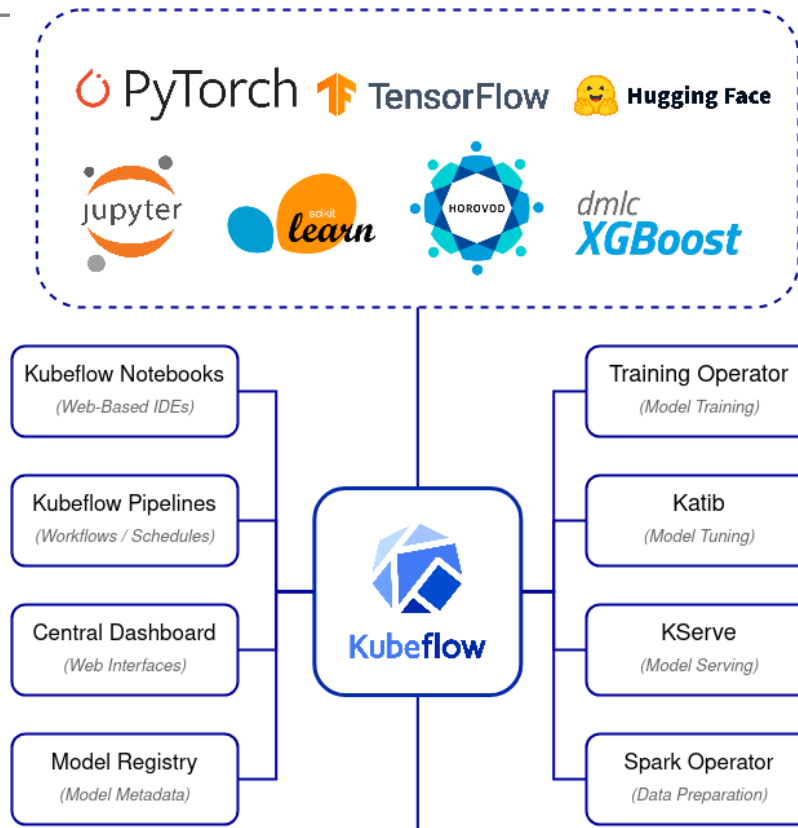
Infrastructure



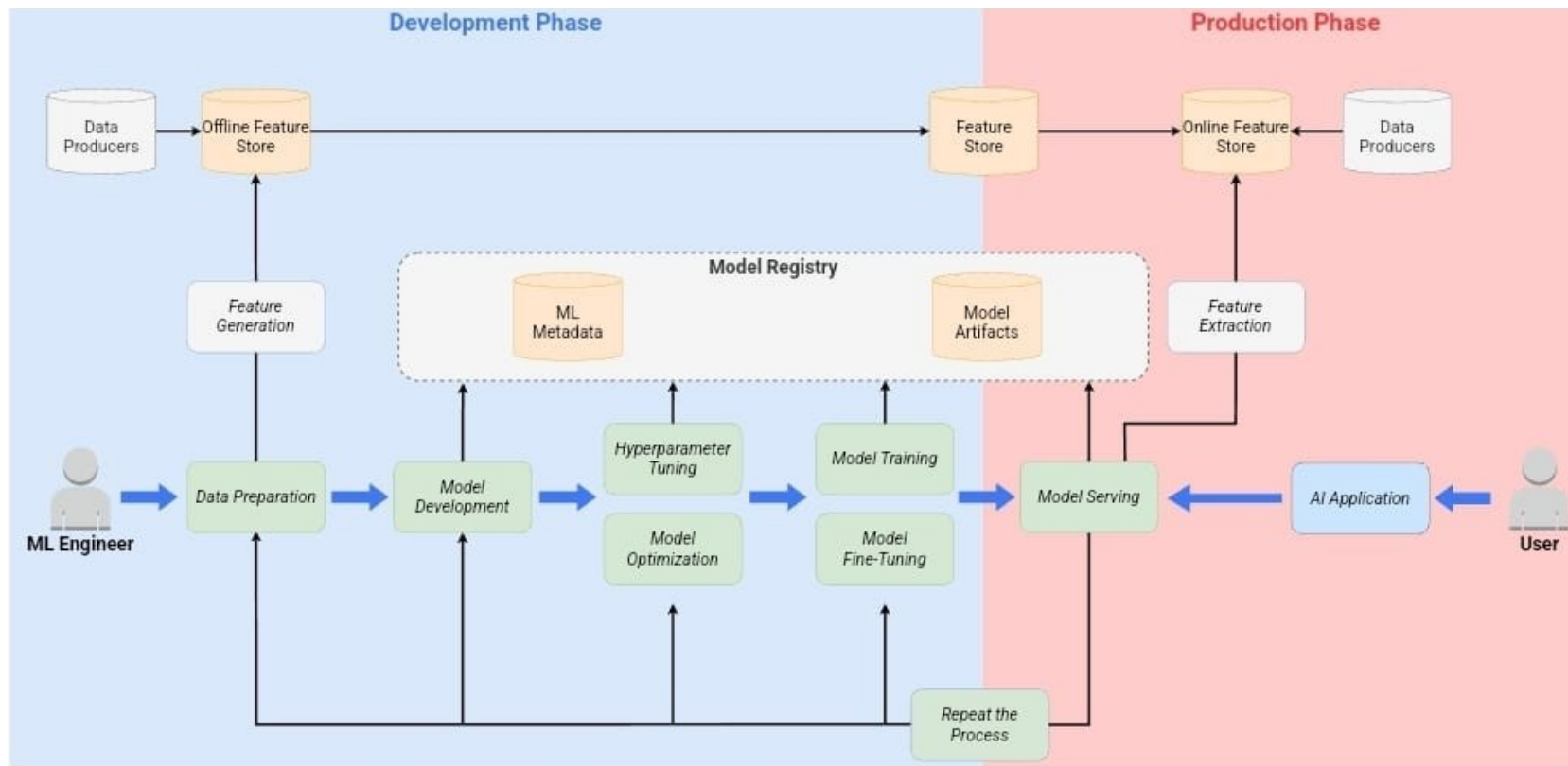
Hardware

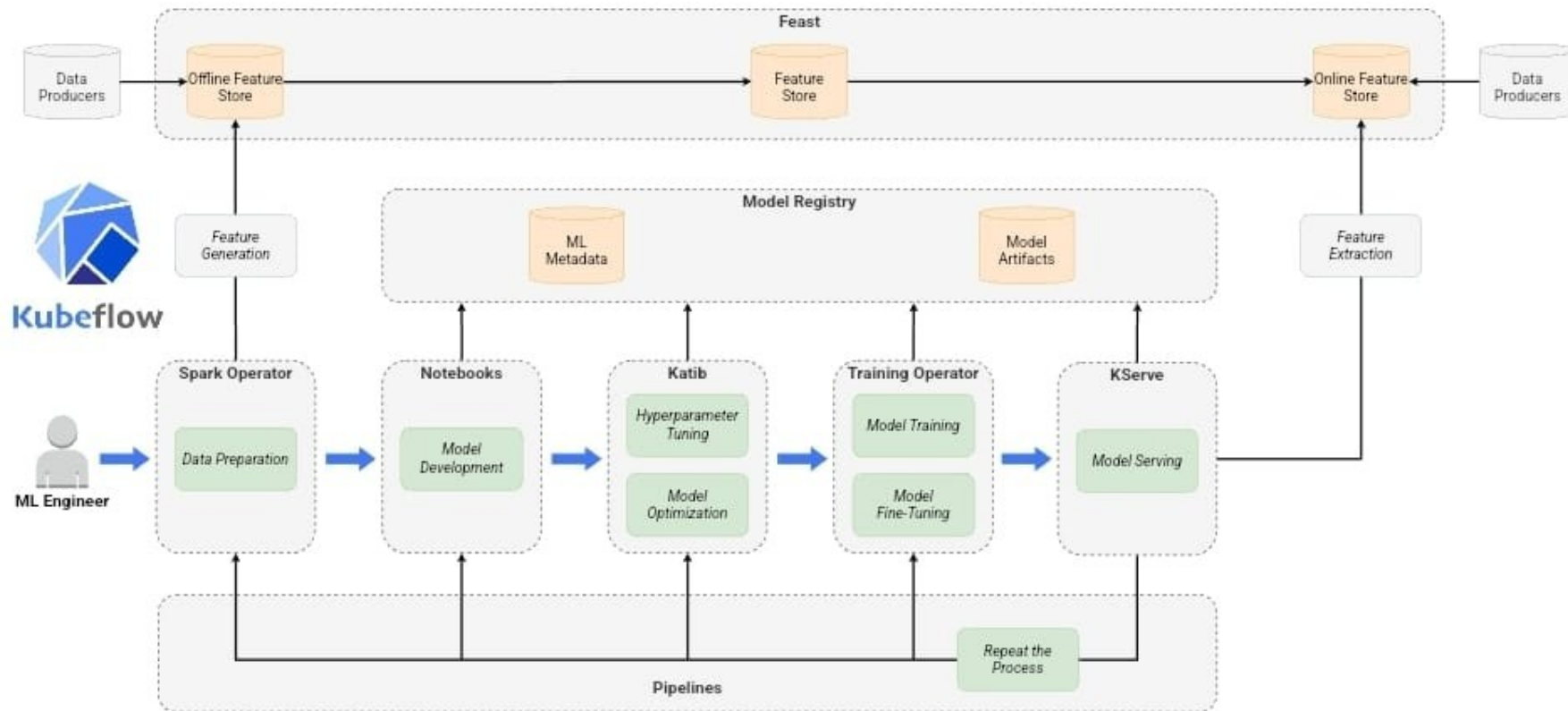


Kubeflow Arquitetura



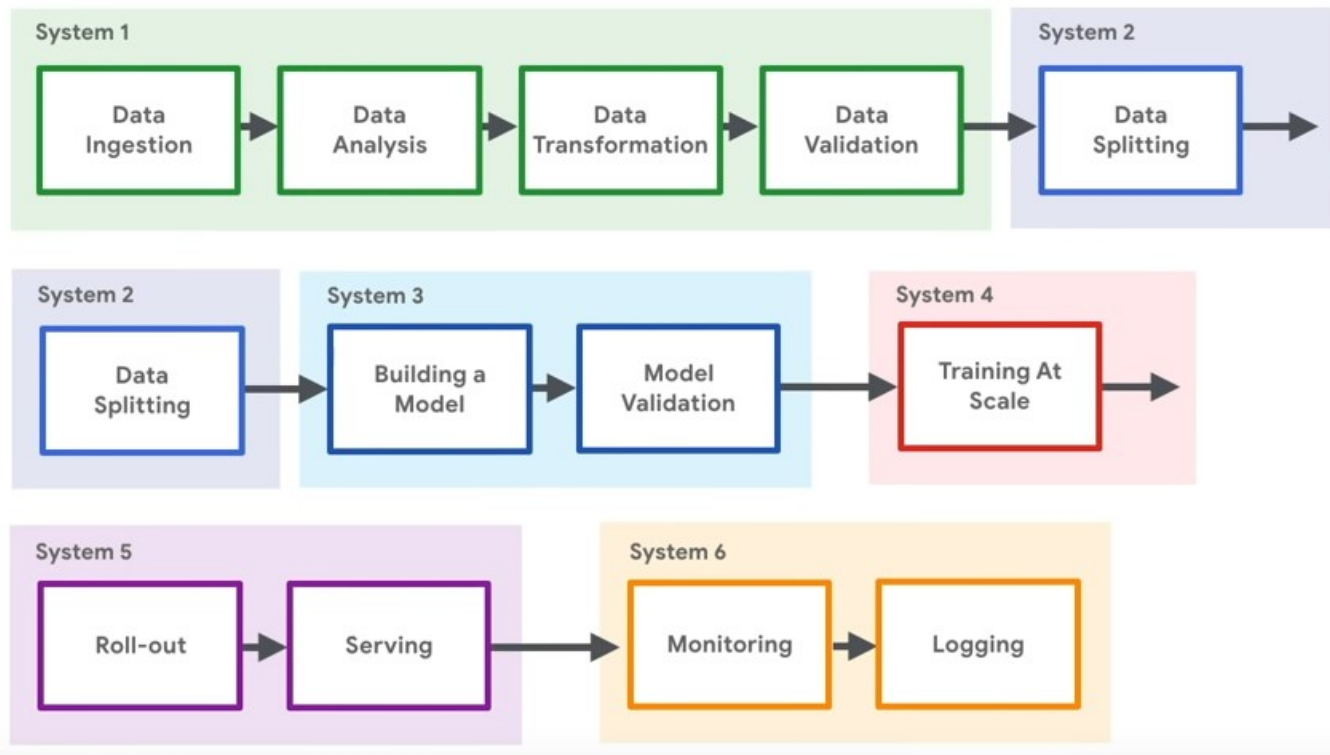
Arquitetura AutoML





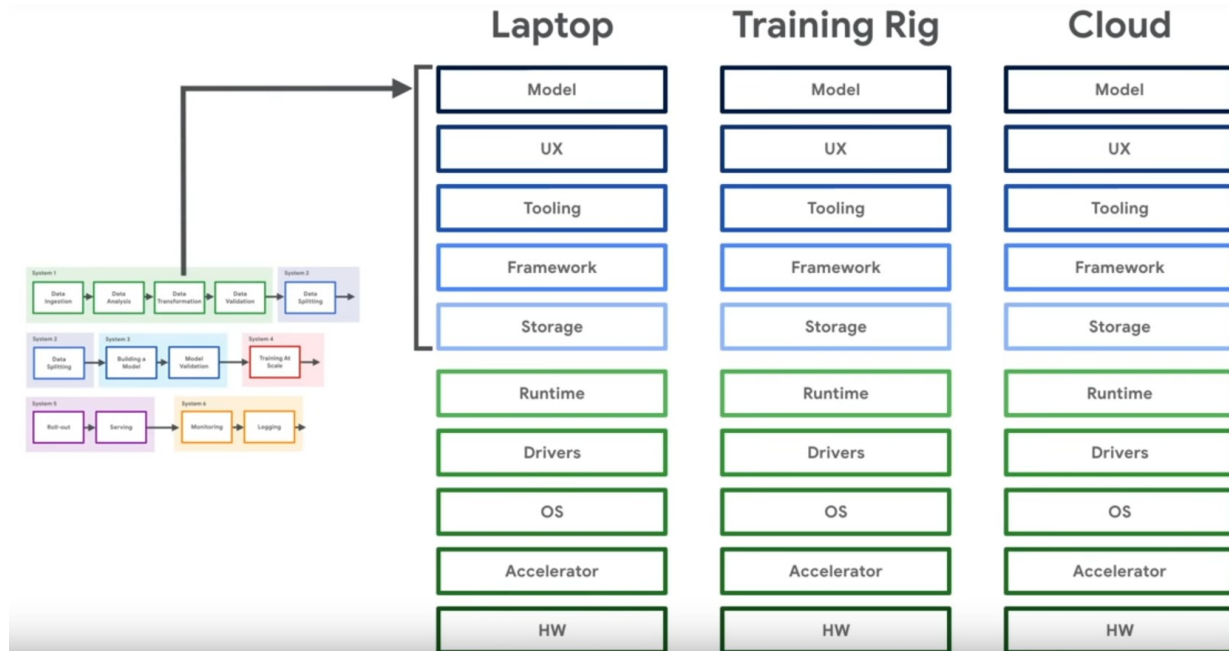
Projeto de ML

- Tem diversos estágios.
- Build Blocks.
- Pode usar diferentes bibliotecas de ML.
- Pode usar diferentes versão das bibliotecas. (Ex: tensorflow.)



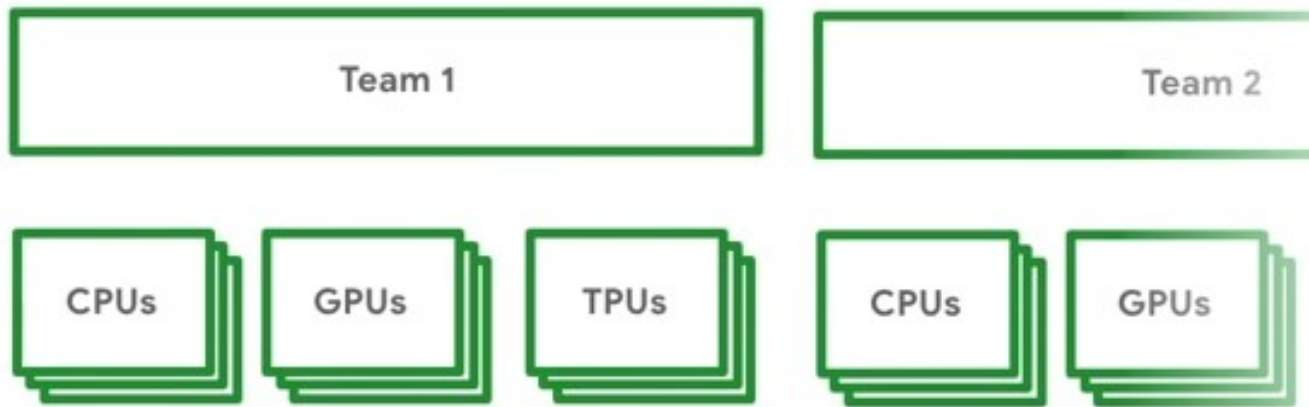
Foco no ML

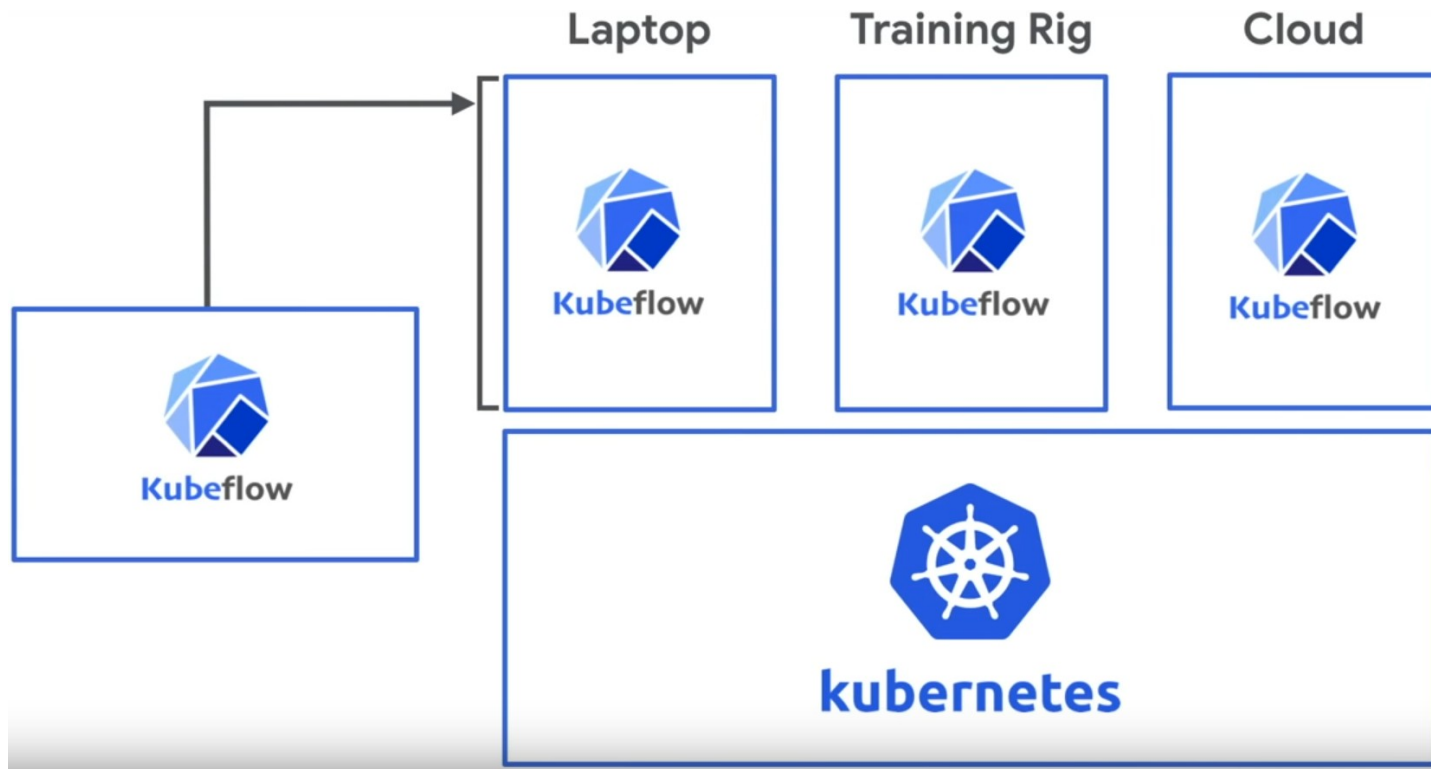
- Foca seu projeto no ML.
- Seleciona onde quer executar. Cloud Publica, Cloud Privada.

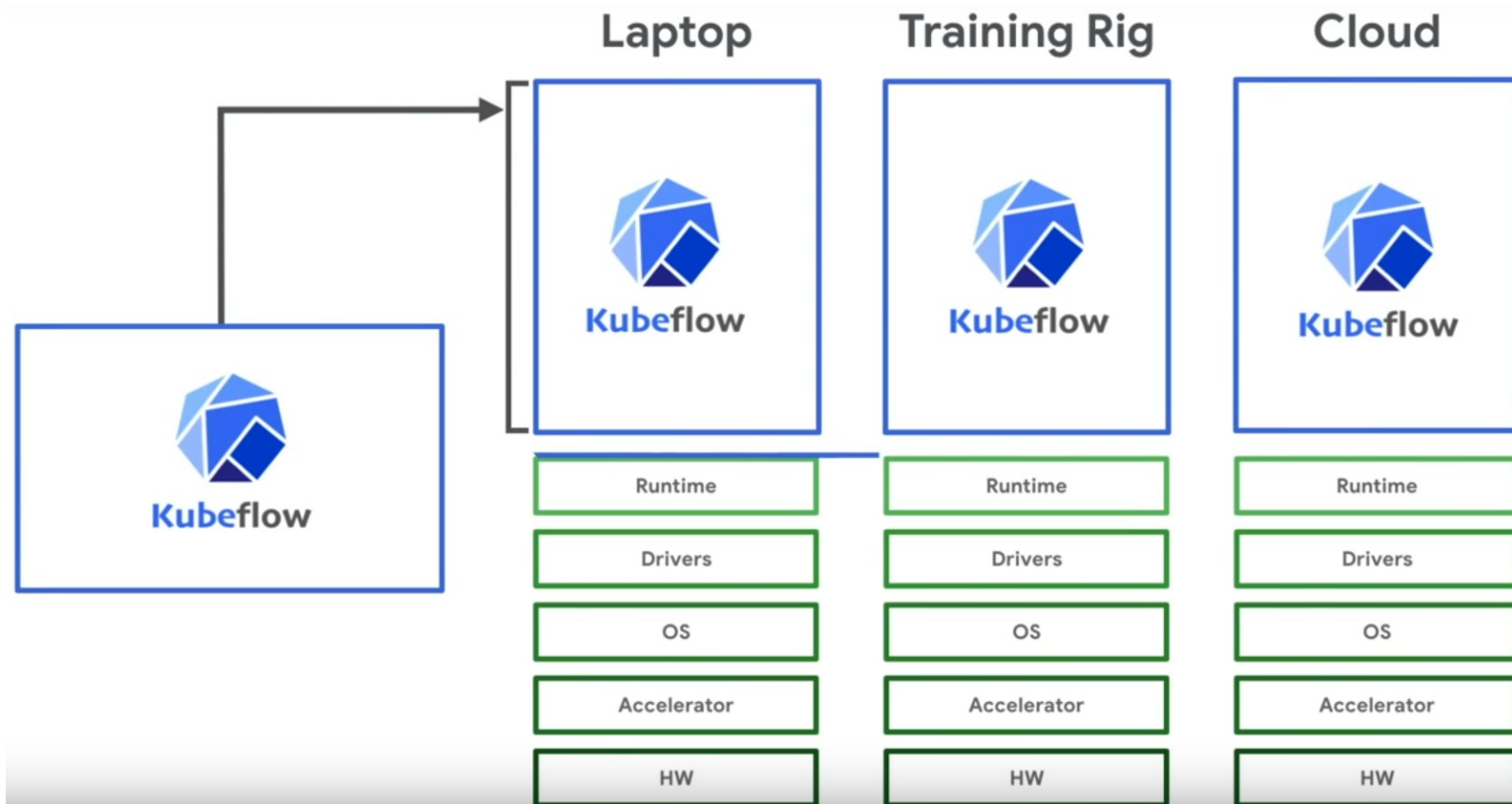


Foco no ML

- Trabalha com ambientes heterogêneos de computação (CPUs, GPUs, TPUs, etc)
- Pode organizar equipes com ambientes diferentes.







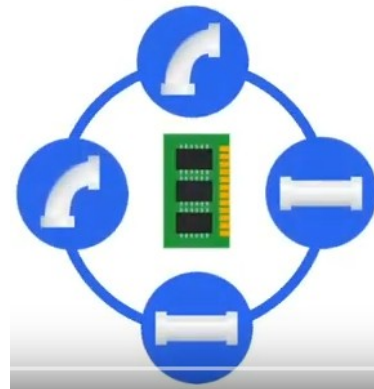
KFP

- É um dos principais componentes do Kubeflow.
- Todo ambiente Kubernetes native.
- Orquestração de Machine Learning pipeline.
- Permite experimentações, reproduções e compartilhar pipelines.
- Reutilização de componentes (building blocks).
- Monitoramento de execução.
- Agendamento de fluxos de trabalhos.
- Registro de metadados e controle de versão.



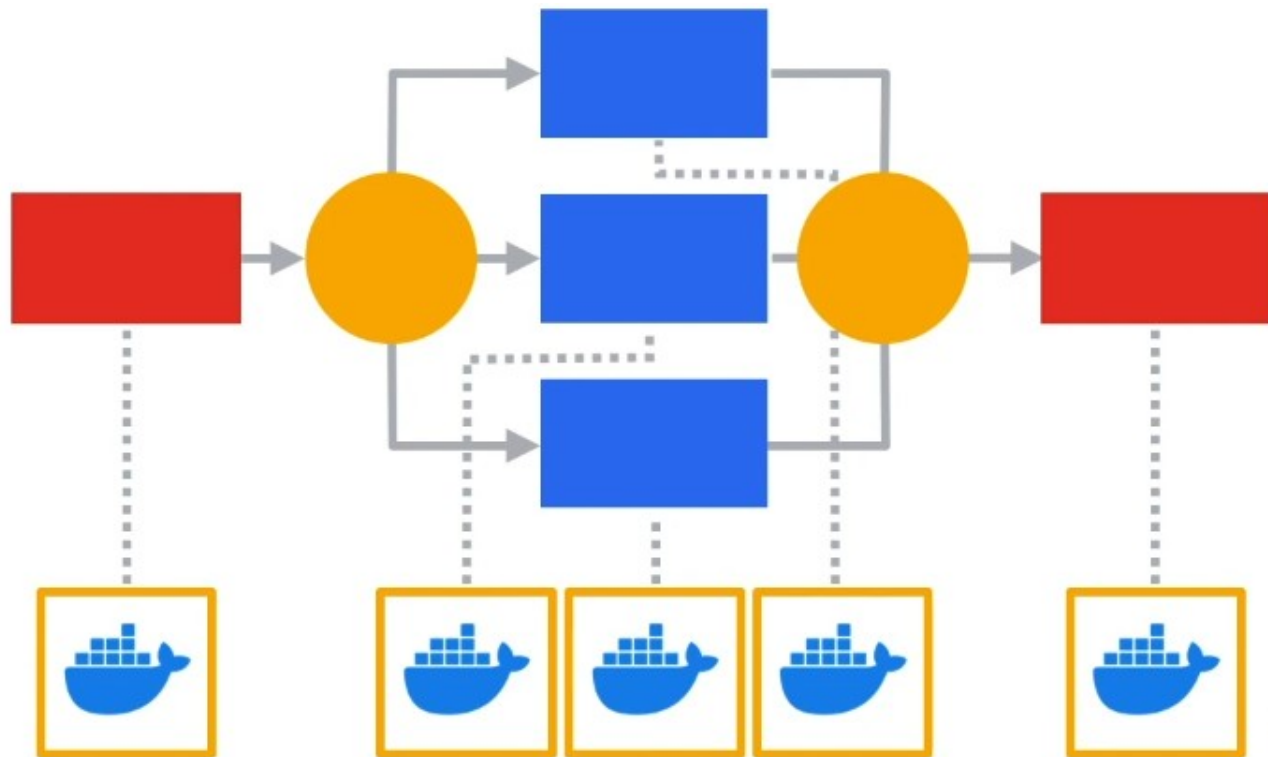
Kubeflow

Pipelines



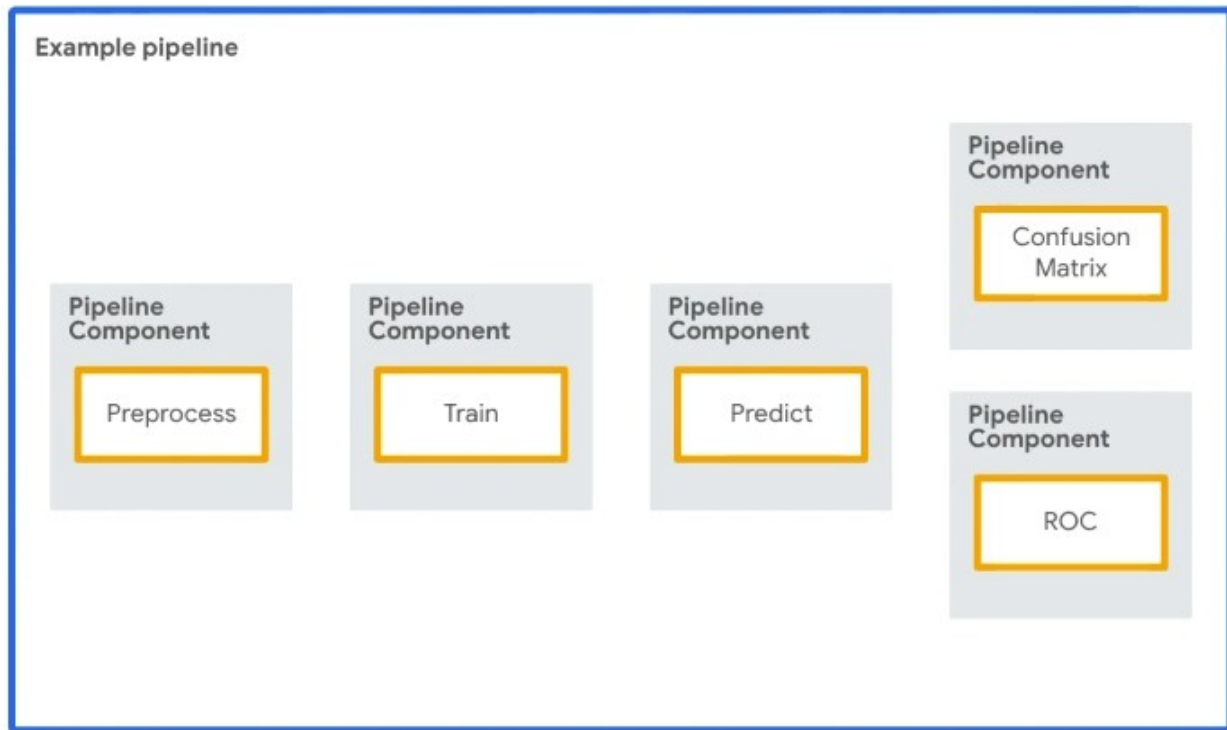
Pipelines

- Executado em contêineres
- Portabilidade
- Repetibilidade.
- Encapsulamento.



Pipelines

- Visão gráfica.
- Entradas e parâmetros.
- Tem diversos componentes que são etapa do workflow.



Componente - Step

- Visão gráfica.
- Entradas e parâmetros.
- Cada componente é uma etapa do workflow.
- Executa uma tarefa específica.
- Simila a uma função nas linguagens de programação.
- Fazem pré-processamento.
- Transformações.
- Treinamento de modelos.

```
In [14]: import kfp.dsl as dsl

def my_pipeline_step(step_name, param1, param2, ...):
    return dsl.ContainerOp(
        name = step_name,
        image = '<path to my container image>',
        arguments = [
            '--param1', param1,
            '--param2', param2,
            ...
        ],
        file_outputs = {
            'output1': '/output1.txt',
            'output2': '/output2.json',
            ...
        }
    )
```

Entradas

- Entradas e parâmetros.

```
In [14]: import kfp.dsl as dsl

def my_pipeline_step(step_name, param1, param2, ...):
    return dsl.ContainerOp(
        name = step_name,
        image = '<path to my container image>',
        arguments = [
            '--param1', param1,
            '--param2', param2,
            ...
        ],
        file_outputs = {
            'output1': '/output1.txt',
            'output2': '/output2.json',
            ...
        }
    )
```

Saídas

- Saída de Valores.

```
In [14]: import kfp.dsl as dsl

def my_pipeline_step(step_name, param1, param2, ...):
    return dsl.ContainerOp(
        name = step_name,
        image = '<path to my container image>',
        arguments = [
            '--param1', param1,
            '--param2', param2,
            ...
        ],
        file_outputs = {
            'output1': '/output1.txt',
            'output2': '/output2.json',
            ...
        }
    )
```

Lógica de ML

- Exemplo de componente usando Pandas e Sklearn

```
[2]: def treina(feato:str, label:str,file:str) -> (float,float) :  
    import pandas  
    from sklearn.linear_model import LinearRegression  
    df = pandas.read_csv(file)  
    reglin = LinearRegression()  
    reglin.fit(df[[feat]], df[label])  
    return (reglin.coef_[0], reglin.intercept_)
```

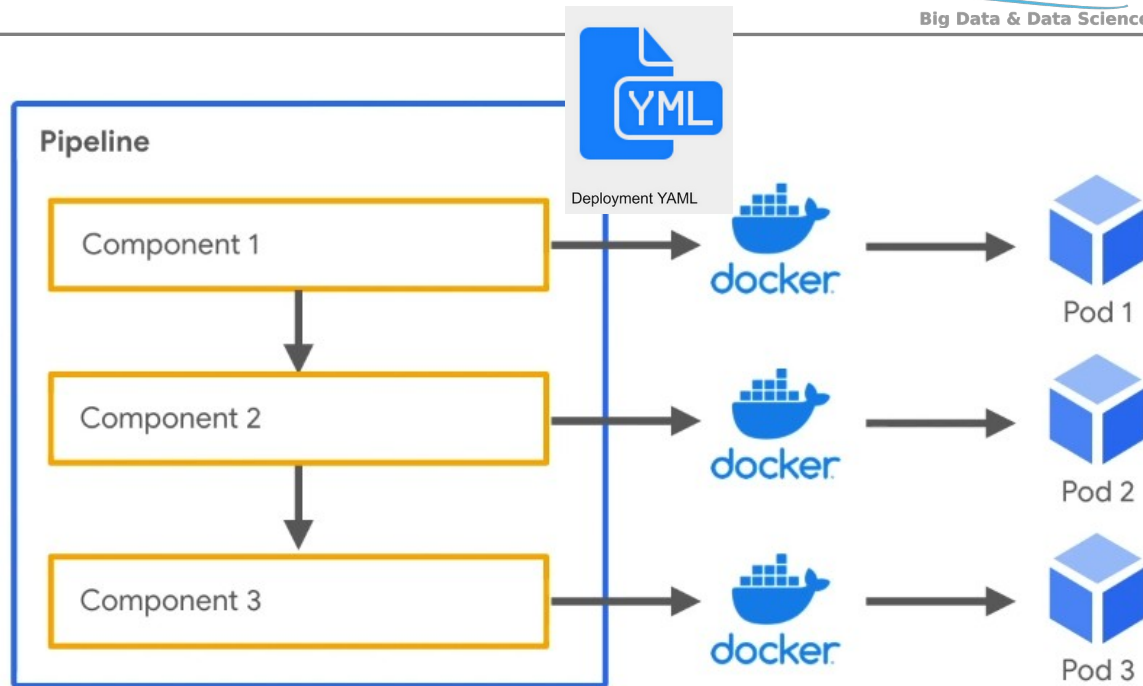
```
[3]: comp_treina = comp.create_component_from_func(treina,output_component_file='treina_component.yaml' base_image=
```


Entradas

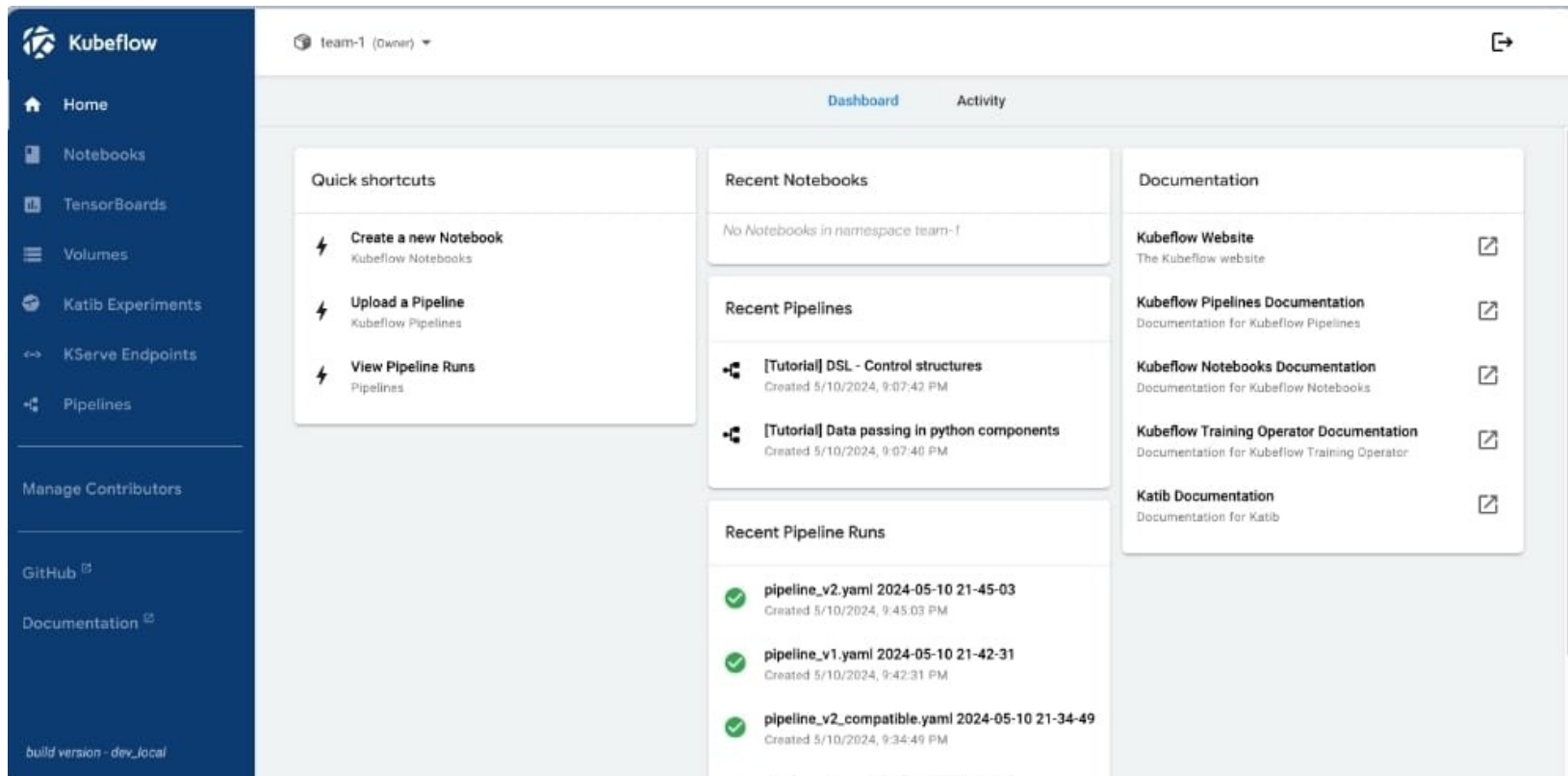
- Cada componente é composto de um código empacotado.
- Um componente inicia 1 ou mais pods do Kubernetes a cada etapa.
- Componentes previamente desenvolvidos podem ser encontrados no github do Kubeflow.

```
kubectll get pods -n kubeflow
```

```
pipeline-iris-8d82r-245036923    0/2    ContainerCreating    0    2m58s
pipeline-texto-curto-h5f4f-2925007992    0/2    ContainerCreating    0    5m54s
workflow-controller-789b47d74-dzdrh    1/1    Running              0    6h30m
```



Interface Gráfica - Kubeflow UI



The screenshot displays the Kubeflow UI interface. On the left is a dark blue sidebar with the Kubeflow logo and navigation links: Home, Notebooks, TensorBoards, Volumes, Katib Experiments, KServe Endpoints, Pipelines, Manage Contributors, GitHub, and Documentation. The main content area is titled 'team-1 (Owner)' and features a 'Dashboard' tab. It is organized into three columns. The first column, 'Quick shortcuts', contains links to 'Create a new Notebook', 'Upload a Pipeline', and 'View Pipeline Runs'. The second column has three sections: 'Recent Notebooks' (empty), 'Recent Pipelines' (listing two tutorial pipelines), and 'Recent Pipeline Runs' (listing three successful pipeline runs with their IDs and creation times). The third column, 'Documentation', provides links to the Kubeflow Website, Pipelines Documentation, Notebooks Documentation, Training Operator Documentation, and Katib Documentation.

Kubeflow

team-1 (Owner)

Dashboard Activity

Quick shortcuts

- Create a new Notebook
Kubeflow Notebooks
- Upload a Pipeline
Kubeflow Pipelines
- View Pipeline Runs
Pipelines

Recent Notebooks

No Notebooks in namespace team-1

Recent Pipelines

- [Tutorial] DSL - Control structures
Created 5/10/2024, 9:07:42 PM
- [Tutorial] Data passing in python components
Created 5/10/2024, 9:07:40 PM

Recent Pipeline Runs

- pipeline_v2.yaml 2024-05-10 21-45-03
Created 5/10/2024, 9:45:03 PM
- pipeline_v1.yaml 2024-05-10 21-42-31
Created 5/10/2024, 9:42:31 PM
- pipeline_v2_compatible.yaml 2024-05-10 21-34-49
Created 5/10/2024, 9:34:49 PM

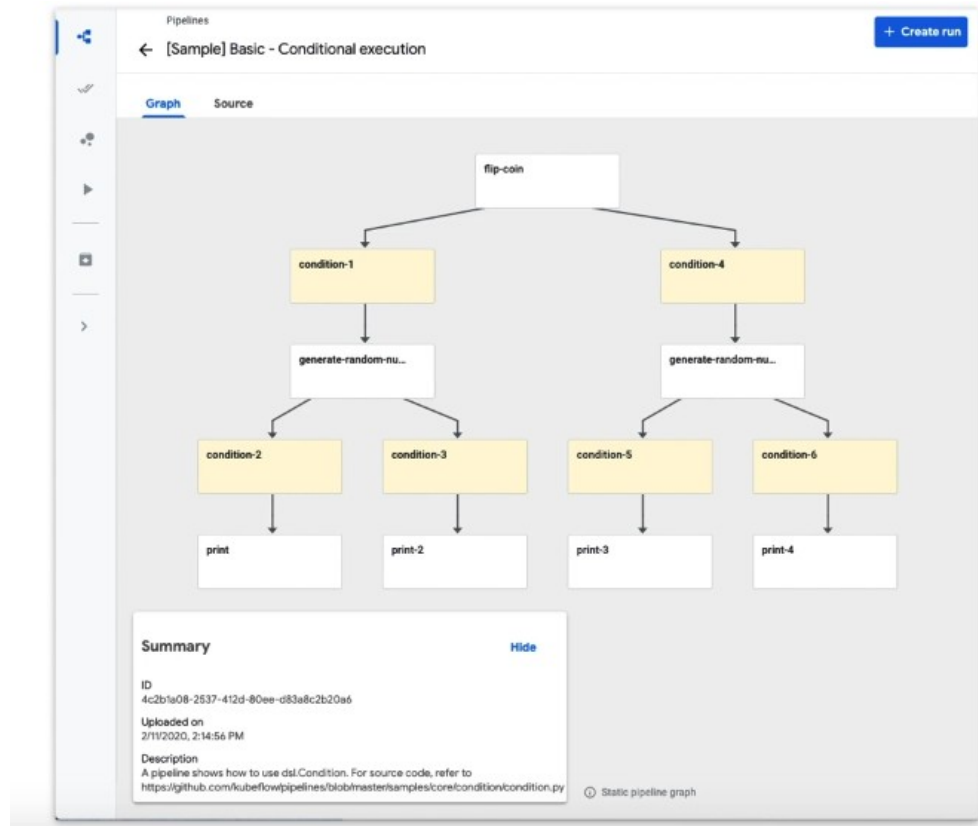
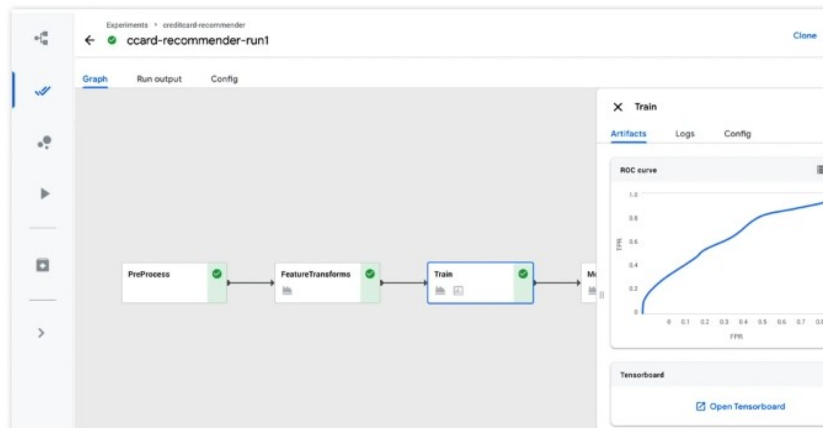
Documentation

- Kubeflow Website
The Kubeflow website
- Kubeflow Pipelines Documentation
Documentation for Kubeflow Pipelines
- Kubeflow Notebooks Documentation
Documentation for Kubeflow Notebooks
- Kubeflow Training Operator Documentation
Documentation for Kubeflow Training Operator
- Katib Documentation
Documentation for Katib

build version - dev_local

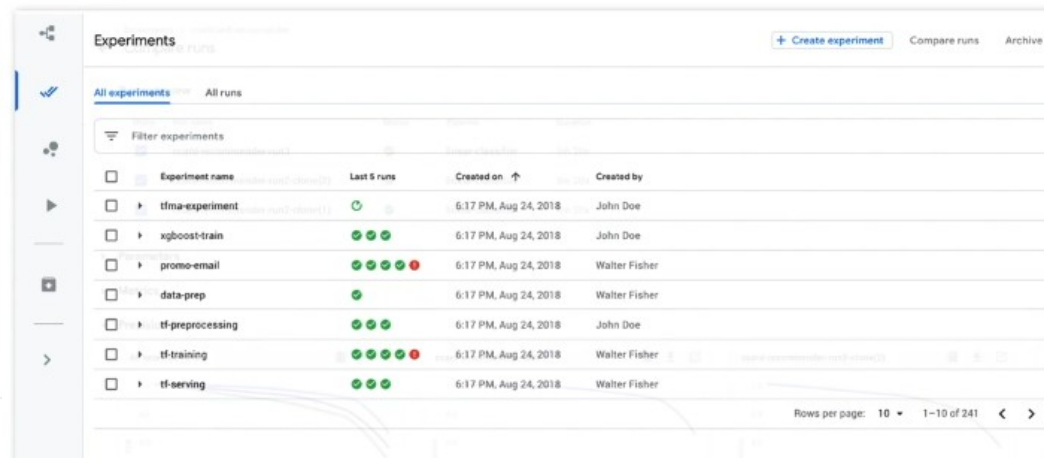
Kubeflow UI

- Status de execução.
- Conclusão de etapas.
- Visualização do artefato de saída.



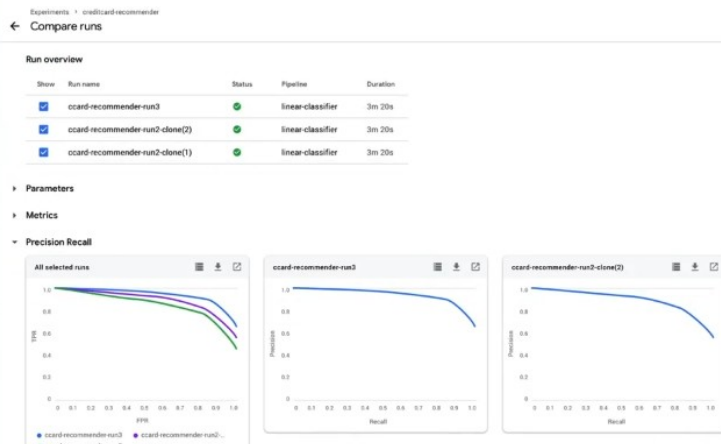
Experimentos.

- Administração de experimentos.
- Estatísticas de desempenho.
- Comparação entre execuções.
- Testar diferentes configurações.



The screenshot shows the 'Experiments' page in the Kubeflow UI. It features a table of experiments with columns for 'Experiment name', 'Last 5 runs', 'Created on', and 'Created by'. The table lists several experiments, including 'tfma-experiment', 'xgboost-train', 'promo-email', 'data-prep', 'tf-preprocessing', 'tf-training', and 'tf-serving'. Each experiment has a status indicator (green circles) and a 'Created on' timestamp. The page also includes a 'Filter experiments' search bar and a 'Rows per page' dropdown set to 10.

Experiment name	Last 5 runs	Created on	Created by
tfma-experiment	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	John Doe
xgboost-train	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	John Doe
promo-email	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	Walter Fisher
data-prep	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	Walter Fisher
tf-preprocessing	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	John Doe
tf-training	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	Walter Fisher
tf-serving	🟢 🟢 🟢 🟢 🟢	6:17 PM, Aug 24, 2018	Walter Fisher



Usuários Kubeflow no Mundo



Google Cloud

**Kubeflow SaaS
ML Services Scale**



Recomendação Musical



multi-cloud



Física de partículas.

Bloomberg

**Hiperparâmetro de
Previsão Financeira.**

Uber

Previsão de demanda e rotas

Fonte: <https://theirstack.com/es/technology/kubeflow/br>

Usuários Kubeflow no Brasil

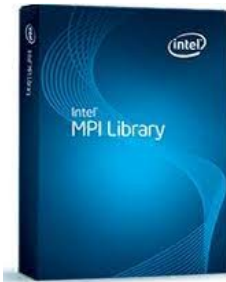


Fontes: <https://theirstack.com/es/technology/kubeflow/br>

<https://medium.com/grupoolxtech/dados-no-jupyter-notebooks-com-o-kubeflow-na-olx-brasil-d475c45d9995>

Conceitos




- Padrão de comunicação para computação paralela e distribuída de alta performance (HPC - High Performance Computing).
- Permite que processos independentes em diferentes nós de um cluster de computadores compartilhem informações e coordenem suas atividades de maneira eficiente, gerando melhorias significativas no desempenho.
- Amplamente utilizado em vários domínios, incluindo ciência, engenharia, finanças e pesquisa.
- API com rotinas para enviar e receber mensagens entre processos em um ambiente paralelo. Inclui rotinas para comunicação coletiva, como reduções e operações de dispersão.
- Diversas implementações: Open MPI, MPICH e Intel MPI.
- Permite que os desenvolvedores aproveitem as vantagens da computação em cluster, distribuindo cargas de trabalho em vários nós e reduzindo o tempo de execução do aplicativo.



OPEN MPI

Suporte

- Ao contrário de outros operadores no Kubeflow, como TF Operator e PyTorch Operator, que suportam apenas uma estrutura de aprendizado de máquina, o operador MPI é **desacoplado** da estrutura subjacente para que possa funcionar bem com muitas estruturas, como Horovod (Uber), TensorFlow, PyTorch, Apache MXNet e vários coletivos implementações de comunicação como OpenMPI.

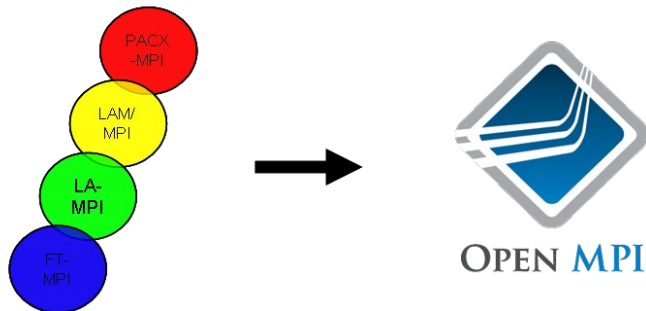
	TF Operator	PyTorch Operator	MPI Operator
Framework Support	 TensorFlow	 PyTorch	 TensorFlow/Keras Apache MXNet/PyTorch/OpenMPI
Distribution Strategy & Backend	tf.distribute MPI/NCCL/PS/TPU	torch.distributed Gloo/MPI/NCCL	horovod DistributedOptimizer Gloo/MPI/NCCL

Conceitos

- Implementação Open Source da biblioteca de passagem de mensagens do MPI.
- Oferece uma plataforma para programadores escreverem aplicativos paralelos que podem ser executados em clusters de computadores.
- Suporta múltiplas linguagens de programação, incluindo C, C++, Fortran, Python e Java, e pode ser executado em vários sistemas operacionais (Linux, Unix, macOS e Windows).
- Desenvolvido e mantido por uma equipe internacional de programadores.
- Amplamente utilizado em várias áreas, incluindo simulação numérica, processamento de imagens, bioinformática, modelagem climática e muito mais.
- Tem Licença BSD.

Histórico

- Criado em 2003 por um grupo que trabalhavam em projetos relacionados a HPC e MPI.
- O objetivo era criar uma implementação de código aberto do MPI que pudesse ser usada em uma ampla variedade de plataformas de hardware e sistemas operacionais.
- Os criadores uniram esforços com outros projetos de MPI de código aberto existentes, e decidiram unificar seus esforços para criar uma nova implementação de MPI.
- Estabeleceram um consórcio de empresas e organizações para fornecer financiamento e suporte para o desenvolvimento do projeto.
- Hoje é amplamente utilizado em todo o mundo em ambientes HPC, e é reconhecido como uma das implementações MPI mais robustas e escaláveis disponíveis.



Time - Open MPI - Empresas Parceiras

ORACLE®

M ADVANCED RESEARCH COMPUTING
UNIVERSITY OF MICHIGAN

absoft

amazon
web services

AMD

ARM®

AUBURN
UNIVERSITY

IBM®

BROADCOM

BULL

ZIH
Center for Information Services &
High Performance Computing

Chelsio
Communications
Accelerate

CISCO

CHEMNITZ UNIVERSITY
OF TECHNOLOGY

UNIVERSITY of WISCONSIN
LA CROSSE™

coverity

CS@UH



facebook

Linaro®

intel®



FUJITSU

Hochschule
für Technik
Stuttgart



Sandia
National
Laboratories

OAK
RIDGE
National Laboratory



NVIDIA®



Vantagens

- Ferramentas simples e eficientes para análise preditiva de dados.
- Acessível a todos e reutilizável em vários contextos.
- Construído em NumPy, SciPy e matplotlib.
- Código aberto, comercialmente utilizável - licença BSD.



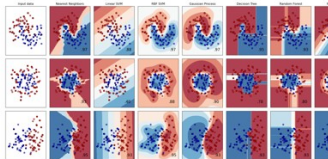
Algoritmos - Scikit-learn

Classificação

Identificar a qual categoria um objeto pertence.

Aplicações: Detecção de spam, reconhecimento de imagem.

Algoritmos: SVM , vizinhos mais próximos , floresta aleatória e muito mais...

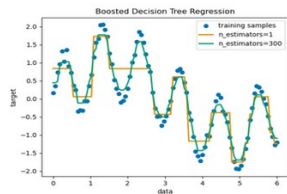


Regressão

Prevendo um atributo de valor contínuo associado a um objeto.

Aplicações: Resposta a medicamentos, Preços de ações.

Algoritmos: SVR , vizinhos mais próximos , floresta aleatória e muito mais...

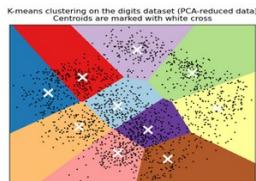


Agrupamento

Agrupamento automático de objetos semelhantes em conjuntos.

Aplicações: segmentação de clientes, agrupamento de resultados de experimentos

Algoritmos: k-Means , agrupamento espectral , média-deslocamento e muito mais...

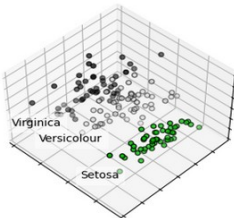


Redução de dimensionalidade

Reduzindo o número de variáveis aleatórias a serem consideradas.

Aplicações: Visualização, Maior eficiência

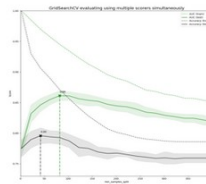
Algoritmos: PCA , seleção de características , fatoração de matriz não negativa , e muito mais...



Seleção de modelo

Comparar, validar e escolher parâmetros e modelos.

Aplicações: Precisão aprimorada por meio de algoritmos de ajuste de parâmetros: pesquisa em grade , validação cruzada , métricas e muito mais...

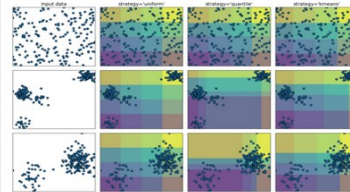


Pré-processando

Extração e normalização de atributos.

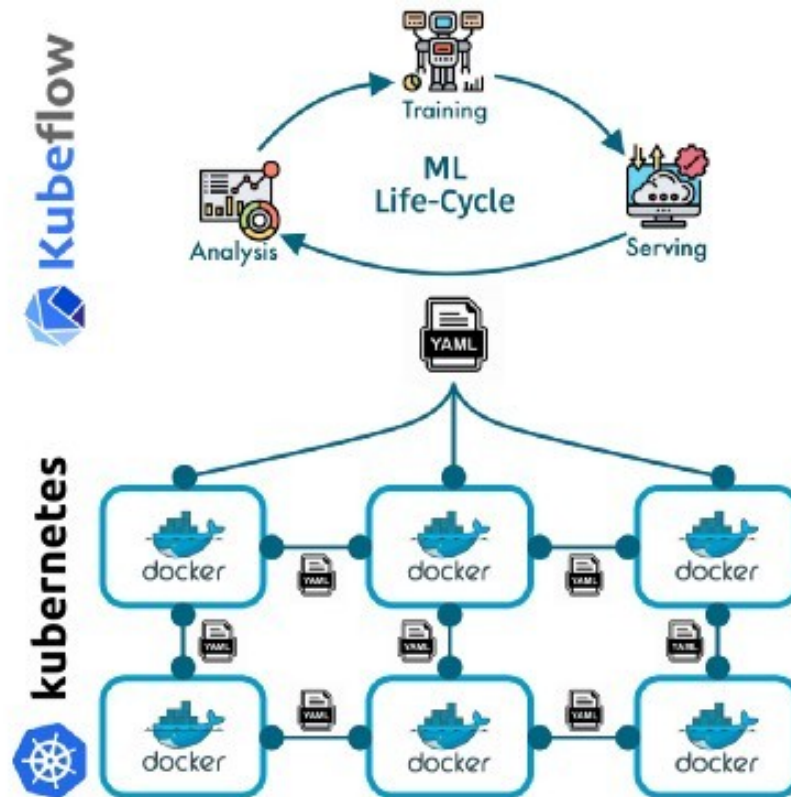
Aplicações: transformação de dados de entrada, como texto, para uso com algoritmos de aprendizado de máquina.

Algoritmos: pré-processamento , extração de recursos e muito mais...



K8s

- Todo Kubeflow e containerizado.



Recomendado

- 16 GB memoria
- 6 CPU
- 45 GB de espaço em disco.

Mínimo

- 10 GB memória.
- 6 CPU.
- 30 GB de espaço em disco.

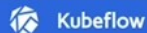
Requisitos

- Todo Kubeflow e containerizado (Native Kubernetes)
- Kubernetes 1.22.;
- Istio 1.14.1;
- Cert-manager 1.5.0;
- Dex 2.31.2;
- Kustomize 3.2.0;
- Knative Serving 1.2.5 (Opcional para usar Kserver Ccomponent);
- Knative Eventing 1.2.4 (Opcional para usar Kserver Ccomponent);

GKE

- GKE é o Google Kubernetes Engine.
- Kubeflow é parte do Google Cloud IA Platform.
- Kubeflow em apenas alguns cliques.





Deploy on GCP

To deploy Kubeflow on Google Cloud Platform:

- Enter the Project ID of the GCP project you want to use
- Pick a name for your deployment
- Choose how to connect to kubeflow service
- (Optional) Choose GKE zone where you want Kubeflow to be deployed
- (Optional) Choose Kubeflow version
- Click Create Deployment
- If your deployment include endpoint, will redirect once endpoint is ready

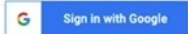
Notice:

- When you click deploy, a service account will be created in target project. The service account will issue a short lived access token which will be sent to Kubeflow deploy service, granting access to necessary GCP resources in target project.
- The Kubeflow deploy service uses this to create Kubeflow GCP resources on your behalf
- If you don't want to delegate a credential to the service please use our [CLI to deploy Kubeflow](#)
- [Terms](#)
- [Privacy](#)



Sign in to deploy Kubeflow

Your credentials are needed to perform GCP actions.



Create a Kubeflow deployment

Project ID *

Deployment name *
kubeflow

Choose how to connect to kubeflow service: *
Login with GCP IAP

- An endpoint protected by GCP IAP will be created for accessing kubeflow. Follow these [instructions](#) to create an OAuth client and then enter as IAP OAuth Client ID and Secret

IAP OAuth client ID *

IAP OAuth client secret *

GKE zone: *
us-central1-a

Kubeflow version: *
v1.0.0

☒ Share Anonymous Usage Report

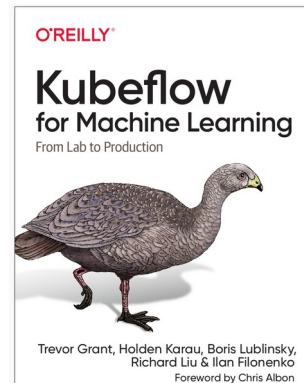
Create Deployment

Kubeflow Service
Endpoint

View YAML

Livros

- Kubeflow for Machine Learning From Lab to Production



Web

- **Kubeflow 101** - Google Cloud Tech -
https://www.youtube.com/playlist?list=PLlivdWyY5sqLS4IN75RPDEyBgTro_YX7x

Obrigado



Marcio Junior Vieira

marcio@ambientelivre.com.br

@marviojvieira @ambientelivre

@ambientelivreopensource

<https://www.linkedin.com/in/mvieira1/>

Blog: <http://blogs.ambientelivre.com.br/marcio/>

<https://github.com/ambientelivre/labs>

https://github.com/ambientelivre/samples_kubeflow