

[Docs](#) | [Patch notes](#) | [Log in](#)[Platform](#) ▾ [Solutions](#) ▾ [Learn](#) ▾ [Blog](#)[Get a demo](#)[← VIEW ALL BLOG POSTS](#)A large hero image with a teal background. In the center, the text 'Kubeflow VS MLflow' is displayed. 'Kubeflow' and 'MLflow' are in a large, bold, dark teal font, while 'VS' is in a smaller, white font between two horizontal white lines. A faint silhouette of a shark is visible in the background.

Kubeflow VS MLflow

A Comprehensive Comparison Between Kubeflow and MLflow

Henrik Skogström / August 11, 2021

As a data scientist or a machine learning engineer, you have probably heard about [Kubeflow](#) and [MLflow](#). These are the two most popular open-source tools under the machine learning platforms umbrella. Because these platforms are the open-source category leaders, they are often compared against each other despite being quite different.

Both products today offer a rather extensive set of capabilities for developing and deploying machine learning models. However, the products started from very different perspectives, with Kubeflow being more orchestration and pipeline-focused and MLflow being more

experiment tracking-focused.

In this article, you will learn about the similarities and significant differences between Kubeflow and MLflow.

Kubeflow and its Components

Kubeflow offers a scalable way to train and deploy models on Kubernetes. It is an orchestration medium that enables a cloud application framework to operate smoothly. Some of the components of Kubeflow include the following:

- **Notebooks:** It offers services for creating and managing interactive Jupyter notebooks in corporate settings. Also included is the ability for users to build notebook containers or pods directly in clusters.
- **TensorFlow model training:** Kubeflow comes with a custom TensorFlow job operator that makes it easy to configure and run model training on Kubernetes. Kubeflow also supports other frameworks through bespoke job operators, but their maturity may vary.
- **Pipelines:** Kubeflow pipelines allow you to build and manage multistep machine learning workflows run in Docker containers.
- **Deployment:** Kubeflow offers several ways to deploy models on Kubernetes through external addons.

MLflow and its Components

MLflow is an open-source framework for tracking the whole machine learning cycle from start to finish, from training to deployment. Among the functions it offers are model tracking, management, packaging, and centralized lifecycle stage transitions. Some of the components of MLflow include the following:

- **Tracking:** While executing your machine learning code, there's an API and UI for logging parameters, code versions, metrics, and output files so you can visualize them later.
- **Project:** They provide a standard style for packaging reusable data science code;

nonetheless, each project is a code directory or a Git repository that uses a descriptor file to indicate dependencies and how to run the code.

- **Models:** MLflow models are a standard for distributing machine learning models in a variety of flavors. There are several tools available to assist with the deployment of various models. Each model is then saved as a directory with arbitrary files and an ML model description file that identifies the flavors in which it can be used.

Similarities between Kubeflow and MLflow

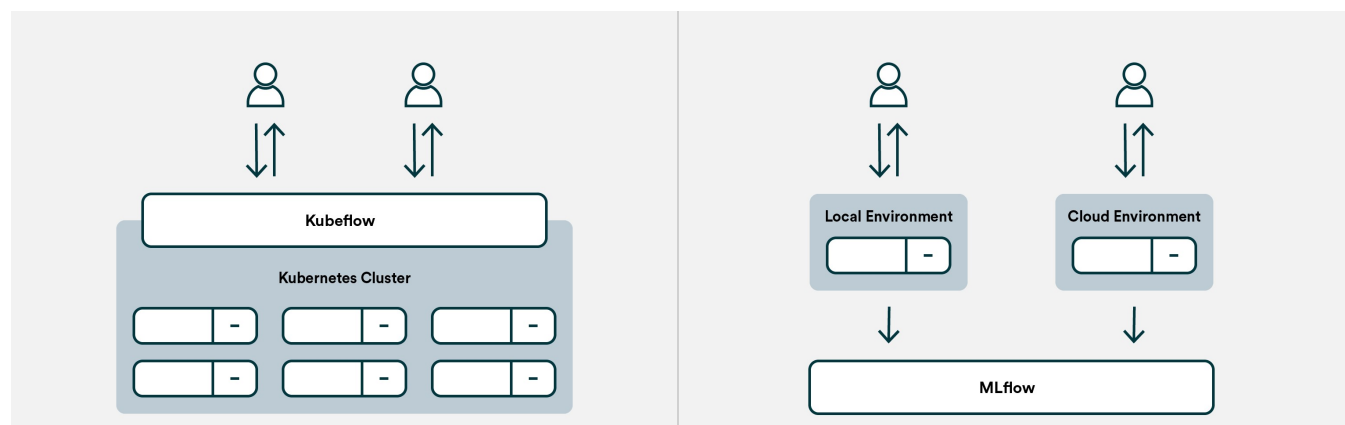
It's crucial to note that both projects are open-source platforms that have received broad support from prominent organizations in the data analytics industry. Here are some similarities between the two platforms.

1. Both technologies aid in creating a collaborative environment for model development
2. Both are scalable, portable, and customizable.
3. Both of them can be characterized as machine learning platforms.

Differences between Kubeflow and MLflow

Both open-source projects are supported by their respective major player in the tech scene. Kubeflow is originated from within Google, while MLflow is supported by Databricks (the authors of Spark).

Using this key difference as the bedrock of each platform, we will now explore four significant differences between Kubeflow and MLflow.



1. **Different approaches:** If there's one thing you remember from this article, it should be this. Kubeflow is, at its core, a container orchestration system, and MLflow is a Python program for tracking experiments and versioning models. Think of it this way: When you train a model in Kubeflow, everything happens within the system (or within the Kubernetes infrastructure it orchestrates), while with MLflow, the actual training happens wherever you choose to run it, and the MLflow service merely listens in on parameters and metrics.

This fundamental difference is also the reason why MLflow is often more favored among data scientists. MLflow is easier to set up since it is just a single service, and it's also easier to adapt your ML experiments to MLflow as the tracking is done via a simple import in your code.

On the other hand, Kubeflow is often characterized as an overly complex tool, but most of the added complexity is due to the infrastructure orchestration capabilities (which require an understanding of Kubernetes). Kubeflow ensures reproducibility to a greater extent than MLflow because it manages the orchestration.

2. **Collaborative environment:** Experiment tracking is at the core of MLflow. It favors the ability to develop locally and track runs in a remote archive via a logging process. This is suitable for exploratory data analysis (EDA). The same capability is made possible through Kubeflow metadata. However, it requires higher technical know-how.
3. **Pipelines and scale:** Orchestrating both parallel and sequential jobs is what Kubeflow was originally built for. For use cases where you may be running end-to-end ML pipelines or large-scale hyperparameter optimization, and you need to utilize cloud computing, Kubeflow is the choice of the two.
4. **Model deployment:** Both have methods for model deployment, but they do so in very different ways. In Kubeflow, this is achieved through Kubeflow pipelines, a distinct component that focuses on model deployment and continuous integration and delivery (CI/CD). Kubeflow pipelines may be used, independent of the rest of Kubeflow's capabilities. MLflow achieves this by utilizing the model registry. Through this, MLflow provides organizations with a central location to share machine learning models as well as a space for collaboration on how to move them forward for implementation and approval in the real world. The MLflow model registry has a set of APIs and UIs to manage the

complete lifecycle of the MLflow model more collaboratively. The registry also provides model versioning, model lineage, annotations, and stage transitions.

MLflow makes it easy to promote models to API endpoints on different cloud environments like Amazon Sagemaker. Also, if you do not want to use a cloud vendor's API endpoint, MLflow has a REST API endpoint that you can use. KubeFlow, on the other hand, allows for a collection of serving components on top of a Kubernetes cluster. This may be at the expense of a larger amount of development effort and time.

Summary

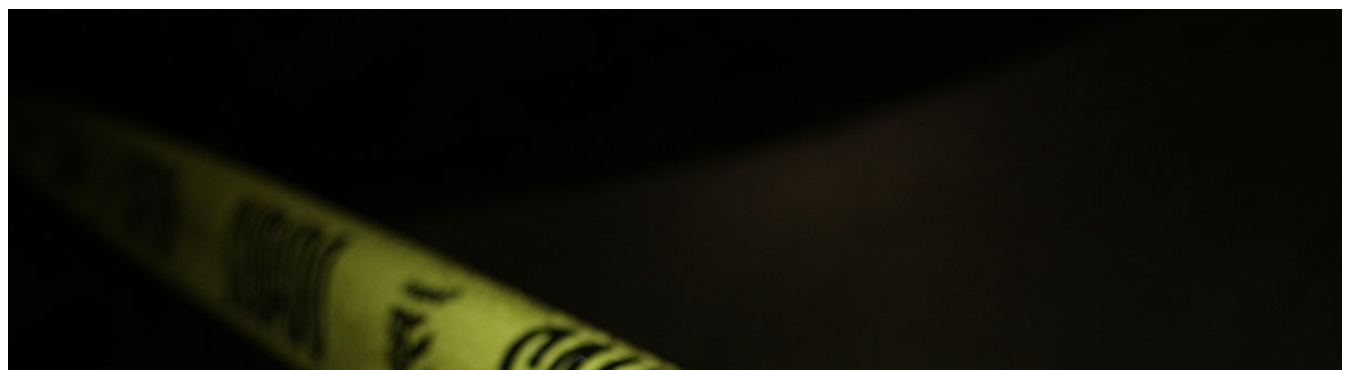
MLflow and KubeFlow are category leaders in the open-source machine learning platforms, but they are very different. To put it simply, KubeFlow solves infrastructure orchestration and experiment tracking with the added cost of being rather demanding to set up and maintain, while MLflow just solves experiment tracking (and model versioning).

KubeFlow meets the requirements of larger teams responsible for delivering custom production ML solutions. These teams often have more specialized roles and have the required resources to manage the Kubernetes infrastructure.

MLflow, on the other hand, more meets the needs of data scientists looking to organize themselves better around experiments and machine learning models. For such teams, ease of use and setup is often the main driver.

Valohai as an Alternative for KubeFlow and MLflow

[CAUTION: Opinions ahead] We didn't just write this article to help choose between KubeFlow and MLflow; we think we've built a better alternative.





Valohai is an MLOps platform that offers Kubeflow-like machine orchestration and MLflow-like experiment tracking without any setup. The gap we see in the market is for teams that have a demanding use case for machine learning (i.e. building production ML systems), but they don't have the resources to manage their own Kubernetes infrastructure and maintain a complex system like Kubeflow.

Unlike Kubeflow and MLflow, Valohai is not an open-source platform but rather a managed one. We take care of setting up, maintaining the environment, and ensuring your team is successfully onboarded. While you can't modify Valohai's source code, we've built the system API-first so other systems can be connected with it (e.g. CI/CD pipelines, monitoring tools, visualization tools). Valohai can also be set up on any cloud or on-premise environment (i.e. data never has to leave your infrastructure).

	Valohai	MLflow	Kubeflow
<i>Type</i>	Managed	Open-source	Open-source
<i>Setup time</i>	Hours	Days to weeks	Weeks to months
<i>Maintenance</i>	None	Required	Required
<i>Features</i>	Experiment tracking Model management Machine orchestration Pipeline automation	Experiment tracking Model management	Experiment tracking Model management Machine orchestration Pipeline automation
<i>Support</i>	Chat, calls & online resources	Online resources	Online resources

embracing MLOps with all of its perks.

If you are interested in learning more, check out:

- [Valohai Product](#)
- [Valohai and Kubeflow Comparison](#)
- [MLOps Platform: Build vs Buy](#)
- [MLOps Platforms Compared](#)

More Kubeflow comparisons

This article continues our series on common tools teams are comparing for various machine learning tasks. You can check out some of our previous Kubeflow comparison articles:

Other Kubeflow comparison articles

[Kubeflow and MLflow](#)[Kubeflow and Airflow](#)[Kubeflow and Metaflow](#)[Kubeflow and Databricks](#)[Kubeflow and SageMaker](#)[Kubeflow and Argo](#)



EBOOK PREVIEW **NEW** 🎉

MLOps in the Wild

MLOps in the Wild is a collection of short & visual MLOps case studies. Download the preview edition to read the first three case studies before everyone else.

First name*

Last name*

Work email*

Company name*

Download the eBook Preview!

Keep reading 🧐

[Top 7 AI Trends in 2022](#)

Sunny Kellman / January 19, 2022

[Managing AI Products: Feasibility, Desirability and Viability](#)

Henrik Skogström / January 17, 2022

[Running Weights & Biases Experiments on Valohai Pipelines](#)

Eikku Koponen / January 14, 2022

[Git for Data Science: What every data scientist should know about Git](#)

Juha Kiili / January 04, 2022

[Data-Centric AI and How to Adopt This Approach](#)

Eikku Koponen and Jean-Emmanuel Wattier / December 20, 2021

[Observability in Production: Monitoring Data Drift with WhyLabs and Valohai](#)

Eikku Koponen / December 02, 2021

[A Comprehensive Comparison Between Kubeflow and Argo](#)

Henrik Skogström / December 01, 2021

[A Comprehensive Comparison Between Kubeflow and Metaflow](#)

Henrik Skogström / November 25, 2021

[Product Update: Human Validation and Confusion Matrices](#)

Juha Kiili / November 24, 2021

[A Comprehensive Comparison Between Kubeflow and SageMaker](#)

Henrik Skogström / November 16, 2021

[A Comprehensive Comparison Between Kubeflow and Databricks](#)

Henrik Skogström / November 09, 2021

[From Notebook to Production: How to Bridge the Gap between Data Science and Engineering?](#)

Eikku Koponen / November 08, 2021



Platform

[Valohai overview](#)

[Why Valohai?](#)

[Success stories](#)

[Pricing](#)

[Book a demo](#)

Technical Resources

[Get started](#)

[Documentation](#)

[Patch notes](#)

[Open standards](#)

Other Resources

[All about MLOps](#)

[Blog posts](#)

[eBooks and whitepapers](#)

Nobody cares...

[About us](#)

[Careers](#)

[Terms of service](#)

[Privacy](#)

;