Developing applications for the cloud increasingly requires building and deployment of this functionality as containerized microservices. Increasingly, artificial intelligence (AI) is at the core of cloud applications. Addressing the need to build and deploy containerized AI models within cloud applications, more vendors of application development tools are building support for containerization into their data-science workbenches [...]
Developing applications for the cloud increasingly requires building and deployment of this functionality as containerized microservices.

Increasingly, artificial intelligence (AI) is at the core of cloud applications. Addressing the need to build and deploy containerized AI models within cloud applications, more vendors of application development tools are building support for containerization into their data-science workbenches and for programming of these applications using languages such as Python, Java, and R.

Data science workbenches are the focus of much AI application development. The latest generation of these tools is leveraging cloud-native interfaces to a steady stream of containerized machine learning models all the way to the edge.

Productionization of the AI DevOps pipeline is no easy feat. It requires a wide range of tooling and infrastructure capabilities, ranging from the workbenches that provide access to such popular AI modeling frameworks as TensorFlow and PyTorch to big data analytics, data governance, and workflow management platforms. In a cloud-native context, it also requires the ability to deploy containerized machine learning and other AI microservices over Kubernetes orchestration backbones in public, private, hybrid, multi-cloud, and even edge environments.

As AI applications begin to infuse every nook and cranny of the cloud-computing universe, it’s absolutely essential that there be open, flexible standards for this DevOps pipeline. This could enable an AI application built in one workbench or framework to be trained, served, executed, benchmarked, and managed downstream in diverse cloud-native application environments that all ride a common end-to-end Kubernetes backplane.

Recognizing this imperative, the AI community has in the past year rapidly coalesced around an open-source project that has built a platform to drive the machine learning DevOps pipeline over Kubernetes. Developed by Google and launched in late 2017, Kubeflow provides a framework-agnostic pipeline for productionizing AI microservices across a multi-framework, multi-cloud cloud-native ecosystem.

Kubeflow supports the entire DevOps lifecycle for containerized machine learning. It simplifies the creation of production-ready AI microservices, ensures the mobility of containerized AI apps between Kubernetes clusters, and supports scaling of AI DevOps workloads to any cluster size. And it’s designed to support any workload in the end-to-end AI DevOps pipeline, ranging from upfront data preparation to iterative modeling and training, and thence too downstream serving, evaluation, and management of containerized AI microservices.

Though it began as an internal Google project for simplified deployment of TensorFlow models to the cloud, Kubeflow is designed to be independent of the specific frameworks in which machine learning models are created, to be agnostic the underlying hardware accelerators used for training and inferencing, and to
Kubeflow Shows Promise in Standardizing the AI DevOps Pipeline

productionize containerized AI apps anywhere in the multicloud that implements Kubernetes, Docker, and other core cloud-native platforms.

Though it’s been going as a community project for less than a year, Kubeflow, currently in version 0.3, has evolved rapidly to include the following rich functionality:

- **Modeling**: Kubeflow supports Jupyter-based AI modeling in the TensorFlow framework, with the community planning to support other popular frameworks—including PyTorch, Caffe2, MXNet, Chainer, and more—in the near future, via Seldon Core, an open source platform for running non-TensorFlow serving and inferencing workloads.

- **Collaboration**: Kubeflow facilitates framework-agnostic creation of AI models in interactive Jupyter notebooks, execution of those models in Jupyter notebook servers, and team-based sharing and versioning in multi-user JupyterHub.

- **Orchestration**: Kubeflow supports deployment of containerized AI applications to cloud-native computing platforms over the open-source Kubernetes orchestration environment, leveraging the cloud-native Ambassador API, Envoy proxy service, Ingress load balancing and virtual hosting service, and Pachyderm data pipelines.

- **Productionization**: Kubeflow incorporates features for managing AI DevOps workflows, including the deployment of TensorFlow in a distributed cloud. It also has extensions for enhancing distributed training performance, performing model benchmarking, hyperparameter tuning, measurement, and testing. It provides a command-line interface for administration of Kubernetes application manifests in support of complex DevOps pipeline deployments that comprise multiple microservice components. And it enables scheduling and execution of distributed training and inferencing on containerized AI models through a controller that can be configured to use either CPUs or GPUs and can be dynamically adjusted to the size of the Kubernetes cluster.

Befitting the critical importance of such a project to scalable AI apps in the cloud, the Kubeflow project has broad industry participation and contributions. The project now has around 100 contributors in 20 different organizations. Organizations that have gone on record as contributing to or otherwise participating in the Kubeflow community include Alibaba Cloud, Altoros, Amazon Web Services, Ant Financial, Argo Project, Arrikto, Caicloud, Canonical, Cisco, Datawire, Dell, Github, Google, H20.ai, Heptio, Huawei, IBM, Intel, Katacoda, MapR, Mesosphere, Microsoft, Momenta, NVIDIA, Pachyderm, Primer, Project Jupyter, Red Hat, Seldon, Uber, and Weaveworks.

However, Kubeflow is far from mature and has only been adopted in a handful of commercial AI workbench and DevOps solutions. Here are a few early adopters among vendors of AI tools that support cloud-native productionization of containerized models:

- **Alibaba Cloud**: The cloud provider’s open-source Arena workbench incorporates Kubeflow within a command-line tool shields AI DevOps professionals from the complexities of low-level resources, environment administration, task scheduling, and GPU scheduling and assignment. It accelerates the tasks of submitting TensorFlow AI training jobs and checking their progress.

- **Amazon Web Services**: The cloud provider supports Kubeflow on their public cloud’s Amazon Elastic Container Service For Kubernetes. They leverage the open-source platform to support data science pipelines that serve machine learning models created in Jupyter notebooks to GPU worker nodes for scalable training and inferencing as microservices on Kubernetes.

- **Cisco**: The vendor supports Kubeflow both on premises and in Google Cloud. Users that run Kubeflow on
Cisco’s unified computing system server platform can provision containerized AI apps and other analytic workloads to it that were built in frameworks such as TensorFlow, PyTorch, and Spark; third-party AI modeling workbenches such as Anaconda Enterprise and **Cloudera Data Science Workbench**; and big data platforms from Cloudera and Hortonworks.

- **H2O.ai**: The vendor supports deployment of their **H2O 3** AI DevOps toolchain on Kubeflow over Kubernetes to reduce the time that data scientists spend on tasks such as tuning model hyperparameters.

- **IBM**: The vendor supports Kubeflow in its Cloud Private platform to support easy configuration and administration of scalable Kubernetes-based AI pipelines in enterprise data centers, Leveraging Kubeflow with IBM Cloud Private-Community Edition, data scientists can collaborate in DevOps pipelines within private cloud environment in their enterprise data centers.

- **Weaveworks**: The vendor provides DevOps tools for automated productionization, observability, and monitoring of cloud-native application workloads over Kubernetes. Its tooling enables users to leverage its Weave Cloud to simplify the observability, deployments, and monitoring of Kubeflow running on Kubernetes clusters.

Within the coming year, Wikibon expects most provider of AI development workbenches, team-oriented machine-pipeline automation tools, and DevOps platforms to fully integrate Kubeflow into their offerings. At the same time, we urge the Kubeflow community to submit this project to the **Cloud Native Computing Foundation** for development within a dedicated working group.
Jim is Wikibon’s Lead Analyst for Data Science, Deep Learning, and Application Development. Previously, Jim was IBM’s data science evangelist. He managed IBM’s thought leadership, social and influencer marketing programs targeted at developers of big data analytics, machine learning, and cognitive computing applications. Prior to his 5-year stint at IBM, Jim was an analyst at Forrester Research, Current Analysis, and the Burton Group. He is also a prolific blogger, a popular speaker, and a familiar face from his many appearances as an expert on theCUBE and at industry events.”

James Kobielus
@jameskobielus
james.kobielus@wikibon.org