Arrikto

Kubernetes & Kubeflow

How Kubernetes and Kubeflow come together

Kubernetes

What is Kubernetes (K8s)? ***

- Portable, extensible, open-source platform
- Designed for managing containerized workloads and services
- Facilitates both declarative configuration and automation.
- Framework to run distributed systems resiliently.

Why Relevant to Kubeflow Ecosystem?

Kubernetes can be described as a platform itself but in our case its a bit of a platform for a platform, in this case Kubeflow!

***Taken from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Kubernetes Clusters

What is a Kubernetes Cluster? ***

- Set of worker machines, called nodes, that run containerized applications.
- The worker nodes host the Pods which have Containers (subsequent slides)
- Pods host the the components of distributed stateful application and workloads

Why Relevant to Kubeflow Ecosystem?

Kubeflow, running on Kubernetes Cluster, provides Data Scientists and MLOps professionals a way to automate execution, resource management and administration of environment.

***Partially taken from https://kubernetes.io/docs/concepts/overview/components/

Containers in K8s

Arrikto

What is a Container? ***

- Similar to VMs w/ relaxed isolation properties to share the Operating System (OS) among the applications.
- Filesystem, share of CPU, memory, process space, and more.
- Decoupled from the underlying infrastructure therefore are portable across clouds and OS distributions.

Why Relevant to Kubeflow Ecosystem?

Kubeflow takes advantage of container isolation to ensure machine learning stages are easily portable and reproducible across machines and clouds.

***Taken from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

Pods in K8s

What is a Pod? ***

- Smallest deployable units of computing that you can create and manage in Kubernetes.
- Group of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

Why Relevant to Kubeflow Ecosystem?

Kubeflow components run in Pods which simplifies container management as well as resource management during Notebook Creation, Pipeline Execution and Model Serving.

***Partially taken from https://kubernetes.io/docs/concepts/overview/components/

Operators & Controllers in K8s

Arrikto

What is an Operator?

Enables automation of common processes in Kubernetes and extends the API. This includes the application or infrastructure we want to manage, a way to declare it (yaml!) and a loop to constantly babysit and adjust our clusters state.

What is a Controller?

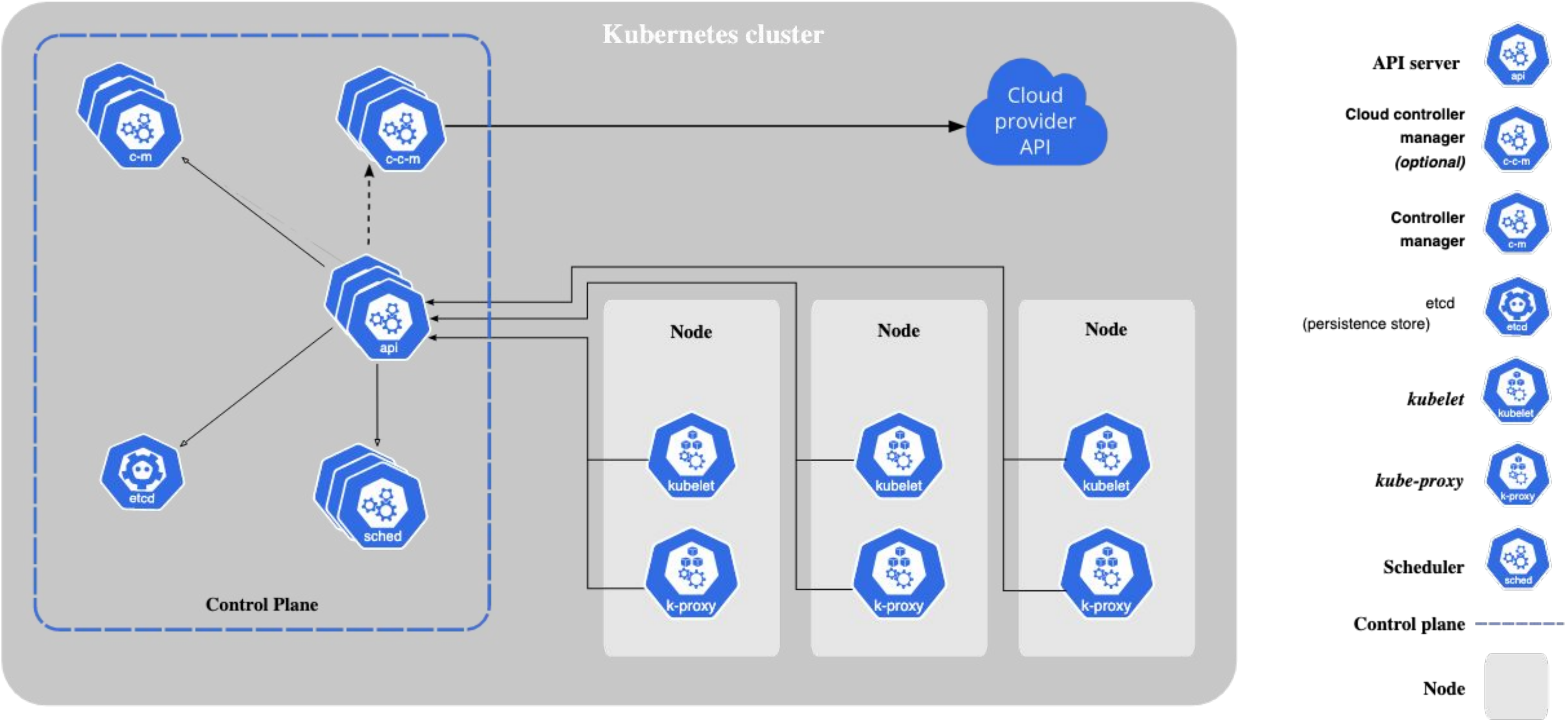
A loop that continuously regulates a system. In this case the controller is continuously regulating the operator which is enabling automation of common processes.

Why Relevant to Kubeflow Ecosystem?

Kubeflow operators and controllers support MLOps and Data Science Model Development Life Cycle, some examples are:

- **Training Operator:** Kubernetes custom resource that makes it easy to run distributed or non-distributed TensorFlow/PyTorch/Apache MXNet/XGBoost/MPI jobs on Kubernetes.
- **Notebook Controller:** allows users to create the custom resource "Notebook"

Visualizing Kubernetes



***Taken from https://kubernetes.io/docs/concepts/overview/components/

Kubeflow & Kubernetes

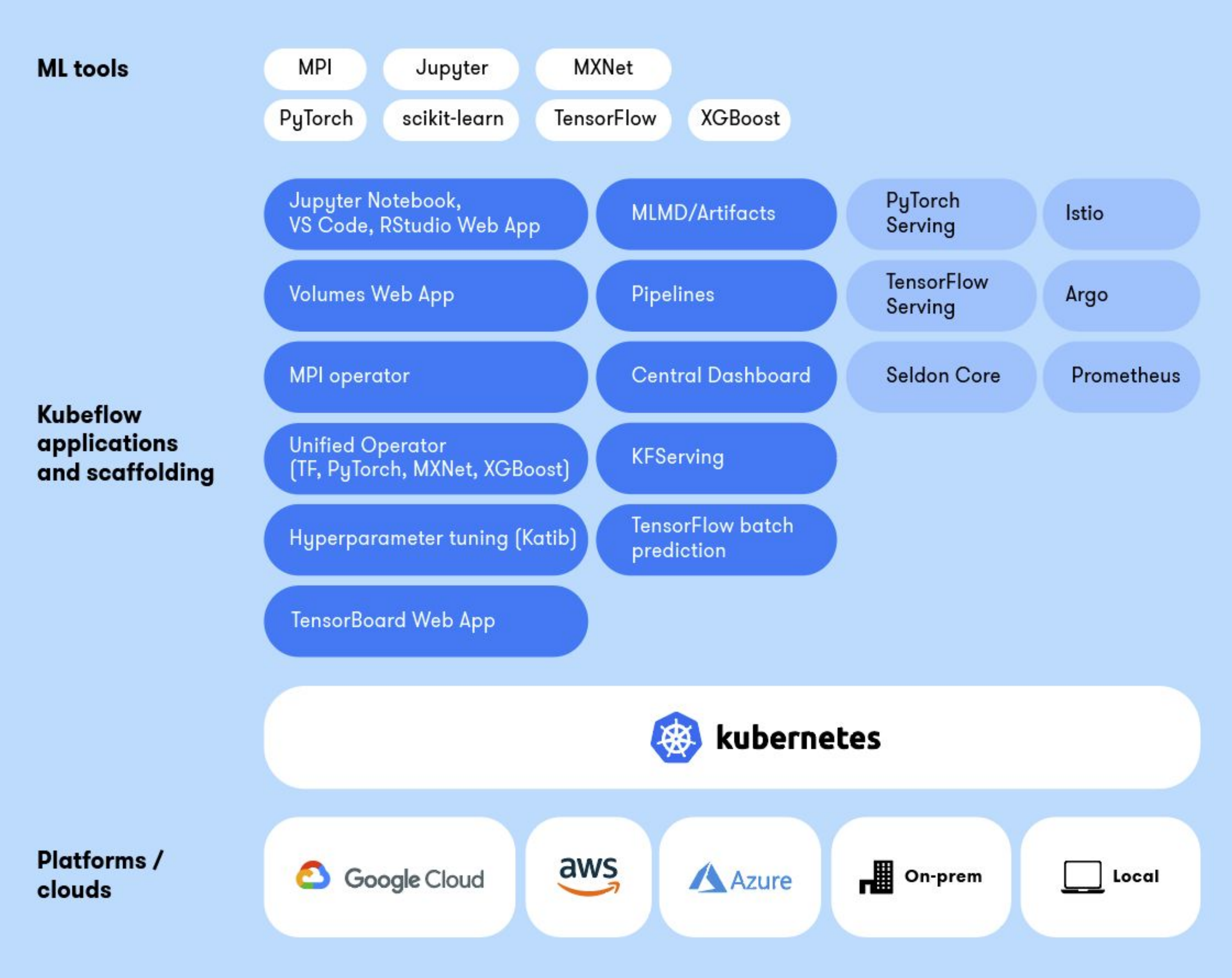
What is Kubeflow Relationship w/ K8s

- Specialized ML platform that is built for Kubernetes.
- Collection of Pods and provides container images to run ML workloads and IDES, such as JupyterLab Notebooks, in Kubernetes Pods.

Why Relevant to Kubeflow Ecosystem?

Kubeflow is an ML obsessed platform that leverages the power of Kubernetes to really improve the Model Development Lifecycle by abstracting away the K8s complexity so data scientists can focus on data science.

Visualizing Kubernetes & Kubeflow



ML tools

MPI · Jupyter · MXNet

PyTorch · scikit-learn · TensorFlow · XGBoost

Kubeflow applications and scaffolding

- Jupyter Notebook, VS Code, RStudio Web App
- Volumes Web App
- MPI operator
- Unified Operator (TF, PyTorch, MXNet, XGBoost)
- Hyperparameter tuning (Katib)
- TensorBoard Web App
- MLMD/Artifacts
- Pipelines
- Central Dashboard
- KFServing
- TensorFlow batch prediction
- PyTorch Serving
- TensorFlow Serving
- Seldon Core
- Istio
- Argo
- Prometheus

kubernetes

Platforms / clouds

Google Cloud · aws · Azure · On-prem · Local

Kubeflow Notebook Servers are Stateful Applications

Arrikto

What is the relationship?

- During notebook server creation and during execution of Kubeflow pipelines the Kubernetes Pods will request storage with a PersistentVolumeClaim and will host PersistentVolumes that are provisioned in response.
- Pods facilitate greater system resilience since data is coupled with deployed code.

Why Relevant to Kubeflow Ecosystem?

Kubeflow running in Kubernetes Pods manages the requests for volumes behind the scenes so the Data Scientists and MLOps professionals can focus on their work.

Kubeflow Notebook Servers & K8’s Resources

Arrikto

What is the relationship?

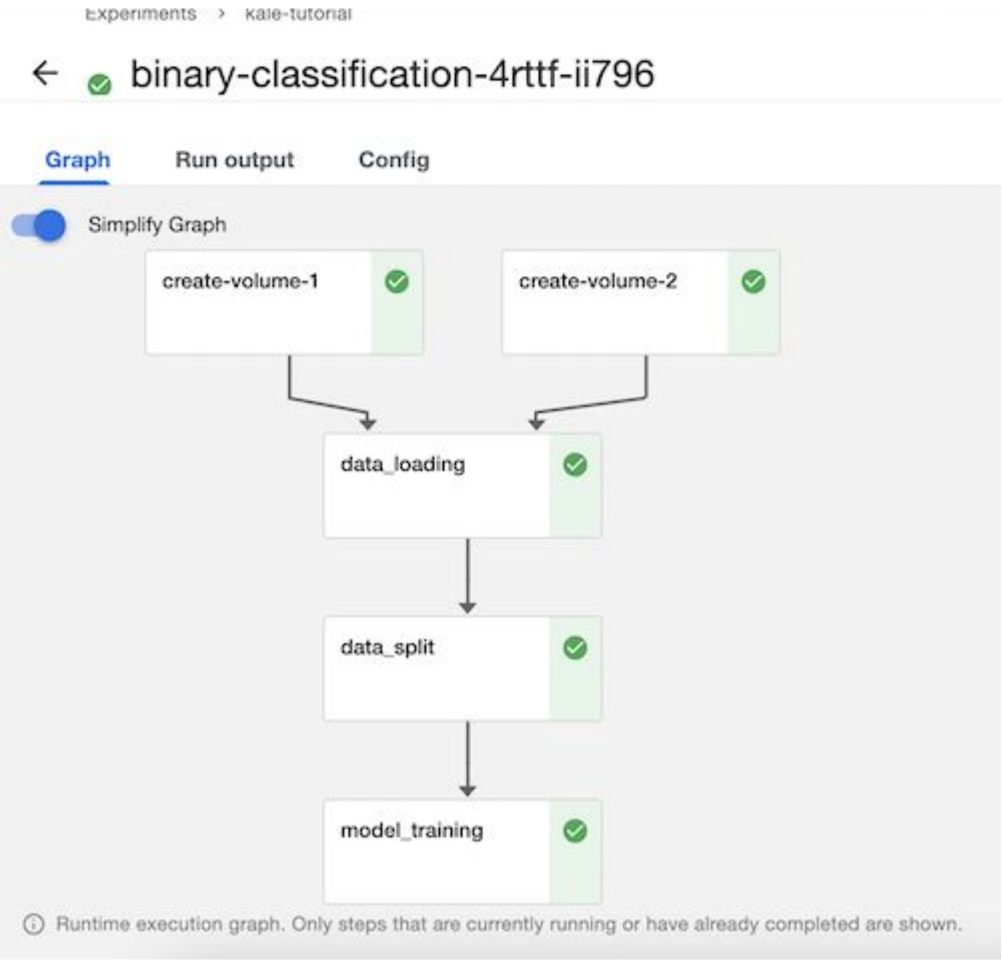
- Notebook Server runs in a container in a Pod, which request CPUs in support of the Notebook Server.
- Pipeline steps execute within Pod, which can request GPUs if necessary.

Why Relevant to Kubeflow Ecosystem?

Developers to focus only on their code while Kubeflow is responsible for the resource management.

A Pipeline! (Visualizing Kubernetes & Kubeflow)

Pipeline created with Jupyter Notebook and Kale



Experiments › kale-tutorial

binary-classification-4rttf-ii796

Runtime execution graph. Only steps that are currently running or have already completed are shown.

Pipeline executed in Kubeflow on Kubernetes

All Together Now! (Addendum)

Arrikto

- Building Kubeflow Pipelines from Jupyter Notebooks or Python Code with Kale makes it easy to take advantage of the relationship between Kubeflow and Kubernetes since Kale ensures that many of the complex considerations that one has to manually account for are automatically addressed during pipeline creation and deployment.
- Notebooks leveraging Arrikto's Rok are StatefulSets which is a special kind of pod that makes sure it gets its very own volume and volume claim. As a result we can restart the pod and know that the exact volume with our data on it is present. We can also take snapshots and rapidly restore environments.
- Guarding against unnecessary GPU usage can help keep costs low, using Kubeflow Notebook Servers allows you to set taints and tolerations to prevent this. Working with Arrikto's EKF you can further manage your node groups (GPU / non-GPU) to manage resource consumption during Pipeline Execution.

Arrikto

THANK YOU!

arrikto.com