Kedro Vertex AI Plugin

Release 0.1.0

GetInData

Mar 22, 2022

CONTENTS:

CHAPTER

ONE

INTRODUCTION

1.1 What is GCP VertexAI Pipelines?

Vertex AI Pipelines is a Google Cloud Platform service that aims to deliver Kubeflow Pipelines functionality in a fully managed fashion. Vertex AI Pipelines helps you to automate, monitor, and govern your ML systems by orchestrating your ML workflow in a serverless manner.

1.2 Why to integrate Kedro project with Vertex AI Pipelines?

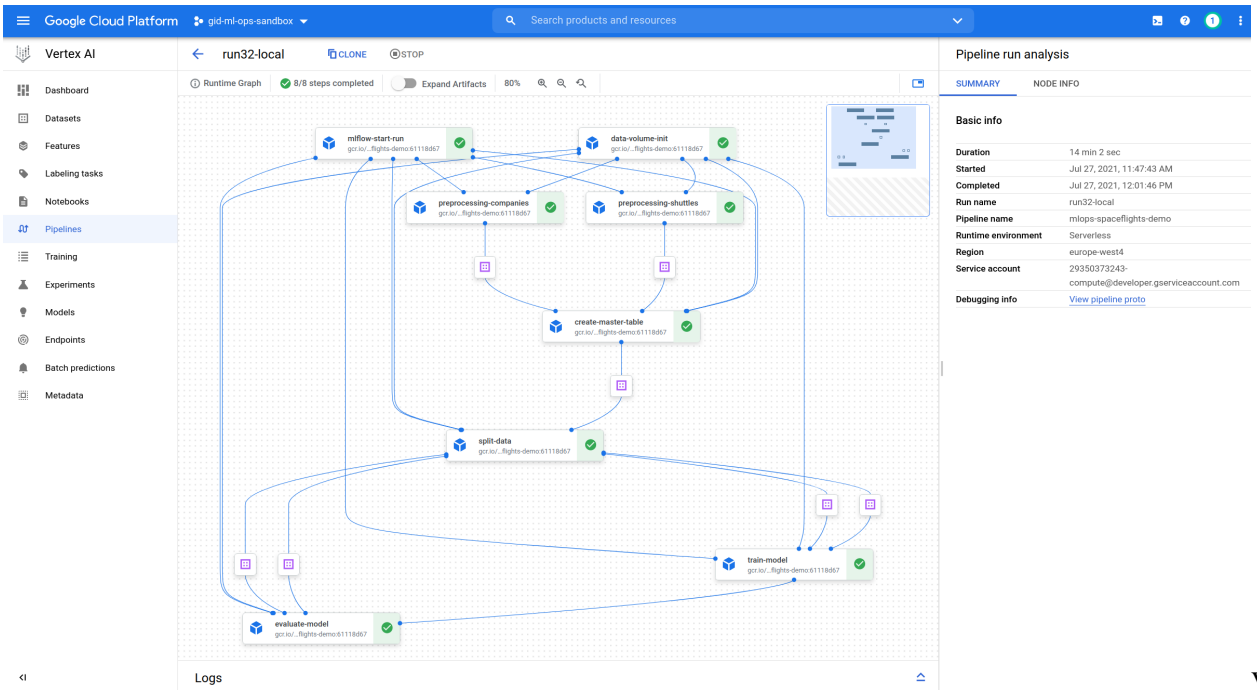
Throughout couple years of exploring ML Ops ecosystem as software developers we've been looking for a framework that enforces the best standards and practices regarding ML model development and Kedro Framework seems like a good fit for this position, but what happens next, once you've got the code ready?

It seems like the ecosystem grown up enough so you no longer need to release models you've trained with Jupyter notebook on your local machine on Sunday evening. In fact there are many tools now you can use to have an elegant model delivery pipeline that is automated, reliable and in some cases can give you a resource boost that's often crucial when handling complex models or a load of training data. With the help of some plugins **You can develop your ML training code with Kedro and execute it using multiple robust services** without changing the business logic.

We currently support:

- Kubeflow kedro-kubeflow
- Airflow on Kubernetes kedro-airflow-k8s

And with this **kedro-vertexai** plugin, you can run your code on GCP Vertex AI Pipelines in a fully managed fashion


Google Cloud Platform
gid-ml-ops-sandbox
Search products and resources
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Notebooks
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
run32-local
CLONE
STOP
Runtime Graph
8/8 steps completed
Expand Artifacts
80%
mlflow-start-run
data-volume-init
preprocessing-companies
preprocessing-shuttles
create-master-table
split-data
train-model
evaluate-model
Logs
Pipeline run analysis
SUMMARY
NODE INFO
Basic info
Duration 14 min 2 sec
Started Jul 27, 2021, 11:47:43 AM
Completed Jul 27, 2021, 12:01:46 PM
Run name run32-local
Pipeline name mlops-spaceflights-demo
Runtime environment Serverless
Region europe-west4
Service account 29350373243-compute@developer.gserviceaccount.com
Debugging info View pipeline proto

VertexAi

CHAPTER

TWO

INSTALLATION

2.1 Installation guide

2.1.1 Kedro setup

First, you need to install base Kedro package

```
$ pip install 'kedro'
```

2.1.2 Plugin installation

Install from PyPI

You can install `kedro-vertexai` plugin from `PyPi` with `pip`:

```
pip install --upgrade kedro-vertexai
```

Install from sources

You may want to install the develop branch which has unreleased features:

```
pip install git+https://github.com/getindata/kedro-vertexai.git@develop
```

2.1.3 Available commands

You can check available commands by going into project directory and runnning:

```
$ kedro vertexai
Usage: kedro vertexai [OPTIONS] COMMAND [ARGS]...

  Interact with GCP Vertex AI Pipelines

Options:
  -e, --env TEXT  Environment to use.
  -h, --help      Show this message and exit.

Commands:
```

(continues on next page)

(continued from previous page)

```
  compile          Translates Kedro pipeline into YAML file with Kubeflow...
  init             Initializes configuration for the plugin
  list-pipelines   List deployed pipeline definitions
  run-once         Deploy pipeline as a single run within given experiment.
  schedule         Schedules recurring execution of latest version of the...
  ui               Open Kubeflow Pipelines UI in new browser tab
  upload-pipeline  Uploads pipeline to Kubeflow server
```

init

`init` command takes one argument (that is the kubeflow pipelines root url) and generates sample configuration file in `conf/base/vertexai.yaml`. The YAML file content is described in the Configuration section.

ui

`ui` command opens a web browser pointing to the currently configured VertexAI Pipelines UI on GCP web console.

list-pipelines

`list-pipelines` uses Kubeflow Pipelines to retrieve all registered pipelines

compile

`compile` transforms Kedro pipeline into Vertex AI workflow. The resulting `yaml` file can be uploaded to Vertex AI Pipelines via web UI.

run-once

`run-once` is all-in-one command to compile the pipeline and run it in the GCP Vertex AI Pipelines environment.

2.2 Configuration

Plugin maintains the configuration in the `conf/base/vertexai.yaml` file. Sample configuration can be generated using `kedro vertexai init`:

```yaml
# Configuration used to run the pipeline
project_id: my-gcp-mlops-project
region: us-central1
run_config:
  # Name of the image to run as the pipeline steps
  image: us.gcr.io/my-gcp-mlops-project/example_model:${commit_id}

  # Pull policy to be used for the steps. Use Always if you push the images
  # on the same tag, or Never if you use only local images
  image_pull_policy: IfNotPresent
```

(continues on next page)

(continued from previous page)

```yaml
# Location of Vertex AI GCS root
root: my-gcp-mlops-project-staging-bucket

# Name of the kubeflow experiment to be created
experiment_name: example-model

# Name of the run for run-once, templated with the run-once parameters
run_name: example-model-${run_id}

# Name of the scheduled run, templated with the schedule parameters
scheduled_run_name: example-model

# Optional pipeline description
#description: "Very Important Pipeline"

# Flag indicating if the run-once should wait for the pipeline to finish
wait_for_completion: False

# How long to keep underlying Argo workflow (together with pods and data
# volume after pipeline finishes) [in seconds]. Default: 1 week
ttl: 604800

# What Kedro pipeline should be run as the last step regardless of the
# pipeline status. Used to send notifications or raise the alerts
# on_exit_pipeline: notify_via_slack

# Optional section allowing adjustment of the resources
# reservations and limits for the nodes
resources:

  # For nodes that require more RAM you can increase the "memory"
  data_import_step:
    memory: 2Gi

  # Training nodes can utilize more than one CPU if the algoritm
  # supports it
  model_training:
    cpu: 8
    memory: 1Gi

  # GPU-capable nodes can request 1 GPU slot
  tensorflow_step:
    nvidia.com/gpu: 1

  # Default settings for the nodes
  __default__:
    cpu: 200m
    memory: 64Mi
```

2.2.1 Dynamic configuration support

`kedro-vertexai` contains hook that enables TemplatedConfigLoader. It allows passing environment variables to configuration files. It reads all environment variables following `KEDRO_CONFIG_<NAME>` pattern, which you can later inject in configuration file using `${name}` syntax.

There are two special variables `KEDRO_CONFIG_COMMIT_ID`, `KEDRO_CONFIG_BRANCH_NAME` with support specifying default when variable is not set, e.g. `${commit_id|dirty}`

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search