

CI/CD for a KFP pipeline

Learning Objectives:

1. Learn how to create a custom Cloud Build builder to pilote CAIP Pipelines
2. Learn how to write a Cloud Build config file to build and push all the artifacts for a KFP
3. Learn how to setup a Cloud Build Github trigger to rebuild the KFP

In this lab you will walk through authoring of a **Cloud Build** CI/CD workflow that automatically builds and deploys a KFP pipeline. You will also integrate your workflow with **GitHub** by setting up a trigger that starts the workflow when a new tag is applied to the **GitHub** repo hosting the pipeline's code.

Configuring environment settings

Update the `ENDPOINT` constat with the settings reflecting your lab environment.

Then endpoint to the AI Platform Pipelines instance can be found on the [AI Platform Pipelines](#) page in the Google Cloud Console.

1. Open the *SETTINGS* for your instance
2. Use the value of the `host` variable in the *Connect to this Kubeflow Pipelines instance from a Python client via Kubeflow Pipelines SKD* section of the *SETTINGS* window.

```
In [1]: ENDPOINT = '<YOUR_ENDPOINT>'
PROJECT_ID = !(gcloud config get-value core/project)
PROJECT_ID = PROJECT_ID[0]
```

Creating the KFP CLI builder

Review the Dockerfile describing the KFP CLI builder

```
In [1]: !cat kfp-cli/Dockerfile

FROM gcr.io/deeplearning-platform-release/base-cpu
RUN pip install kfp==0.2.5
ENTRYPOINT ["/bin/bash"]
```

Build the image and push it to your project's Container Registry.

```
In [ ]: IMAGE_NAME='kfp-cli'
TAG='latest'
IMAGE_URI='gcr.io/{}/{}:{}'.format(PROJECT_ID, IMAGE_NAME, TAG)
```

```
In [ ]: !gcloud builds submit --timeout 15m --tag {IMAGE_URI} kfp-cli
```

Understanding the Cloud Build workflow.

Review the `cloudbuild.yaml` file to understand how the CI/CD workflow is implemented and how environment specific settings are abstracted using **Cloud Build** variables.

The CI/CD workflow automates the steps you walked through manually during `lab-02-kfp-pipeline` :

1. Builds the trainer image
2. Builds the base image for custom components
3. Compiles the pipeline
4. Uploads the pipeline to the KFP environment
5. Pushes the trainer and base images to your project's **Container Registry**

Although the KFP backend supports pipeline versioning, this feature has not been yet enable through the **KFP** CLI. As a temporary workaround, in the **Cloud Build** configuration a value of the `TAG_NAME` variable is appended to the name of the pipeline.

The **Cloud Build** workflow configuration uses both standard and custom [Cloud Build builders](#). The custom builder encapsulates **KFP CLI**.

Manually triggering CI/CD runs

You can manually trigger **Cloud Build** runs using the `gcloud builds submit` command.

```
In [ ]: SUBSTITUTIONS="\"
        _ENDPOINT={},\
        _TRAINER_IMAGE_NAME=trainer_image,\
        _BASE_IMAGE_NAME=base_image,\
        TAG_NAME=test,\
        _PIPELINE_FOLDER=.,\
        _PIPELINE_DSL=covertypes_training_pipeline.py,\
        _PIPELINE_PACKAGE=covertypes_training_pipeline.yaml,\
        _PIPELINE_NAME=covertypes_continuous_training,\
        _RUNTIME_VERSION=1.15,\
        _PYTHON_VERSION=3.7,\
        _USE_KFP_SA=True,\
        _COMPONENT_URL_SEARCH_PREFIX=https://raw.githubusercontent.com/kubeflow/pipeline
        \"\"\".format(ENDPOINT).strip()
```

```
In [ ]: !gcloud builds submit . --config cloudbuild.yaml --substitutions {SUBSTITUTIONS}
```

Setting up GitHub integration

In this exercise you integrate your CI/CD workflow with **GitHub**, using [Cloud Build GitHub App](#). You will set up a trigger that starts the CI/CD workflow when a new tag is applied to the **GitHub** repo managing the pipeline source code. You will use a fork of this repo as your source GitHub repository.

Create a fork of this repo

[Follow the GitHub documentation](#) to fork this repo

Create a Cloud Build trigger

Connect the fork you created in the previous step to your Google Cloud project and create a trigger following the steps in the [Creating GitHub app trigger](#) article. Use the following values on the **Edit trigger** form:

Field	Value
Name	[YOUR TRIGGER NAME]
Description	[YOUR TRIGGER DESCRIPTION]
Event	Tag
Source	[YOUR FORK]
Tag (regex)	.*
Build Configuration	Cloud Build configuration file (yaml or json)
Cloud Build configuration file location	/workshops/kfp-caip-sklearn/lab-03-kfp-cicd/cloudbuild.yaml

Use the following values for the substitution variables:

Variable	Value
_BASE_IMAGE_NAME	base_image
_COMPONENT_URL_SEARCH_PREFIX	https://raw.githubusercontent.com/kubeflow/pipelines/0.2.5/component
_ENDPOINT	[Your inverting proxy host]
_PIPELINE_DSL	covertime_training_pipeline.py
_PIPELINE_FOLDER	workshops/kfp-caip-sklearn/lab-03-kfp-cicd
_PIPELINE_NAME	covertime_training_deployment
_PIPELINE_PACKAGE	covertime_training_pipeline.yaml
_PYTHON_VERSION	3.7
_RUNTIME_VERSION	1.15
_TRAINER_IMAGE_NAME	trainer_image
_USE_KFP_SA	False

Trigger the build

To start an automated build [create a new release of the repo in GitHub](#). Alternatively, you can start the build by applying a tag using `git`.

```
git tag [TAG NAME]
git push origin --tags
```

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.