



CI/CD for Kubeflow Pipelines on AI Platform



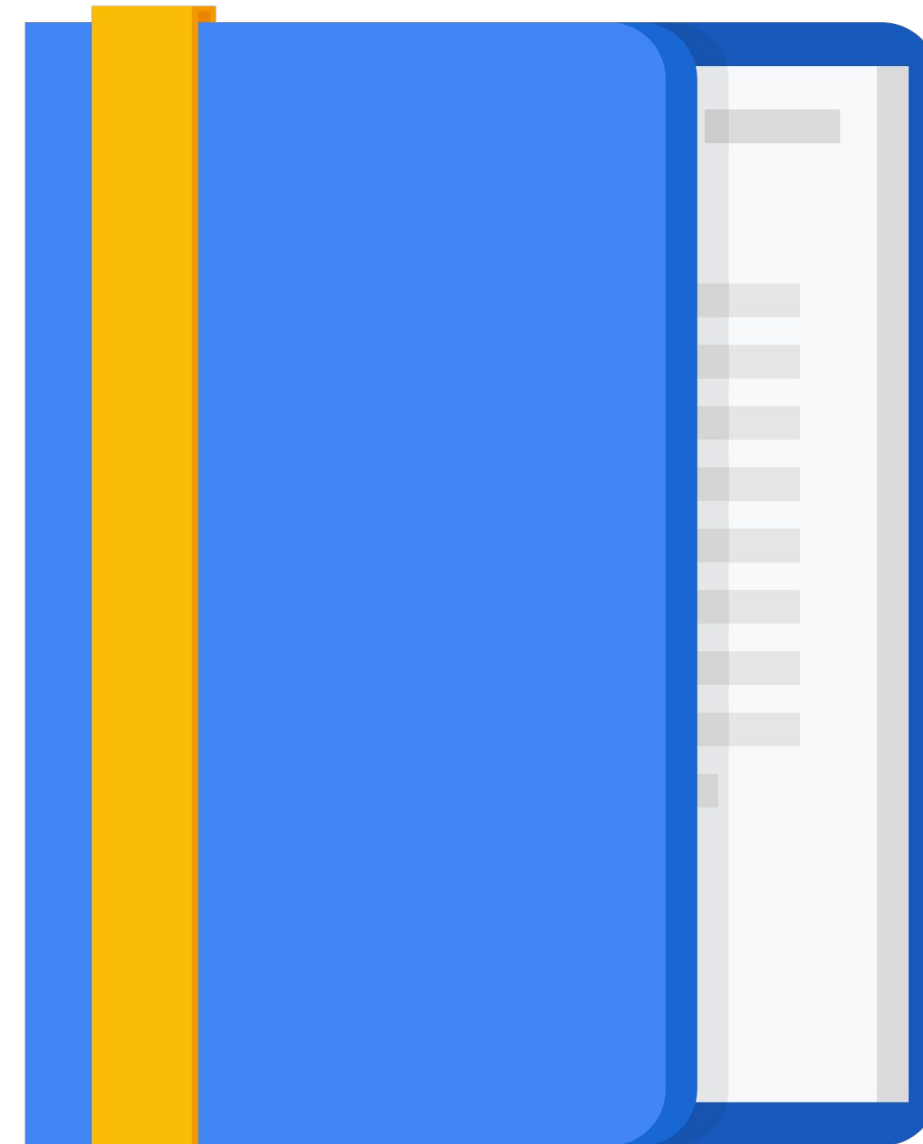
Agenda

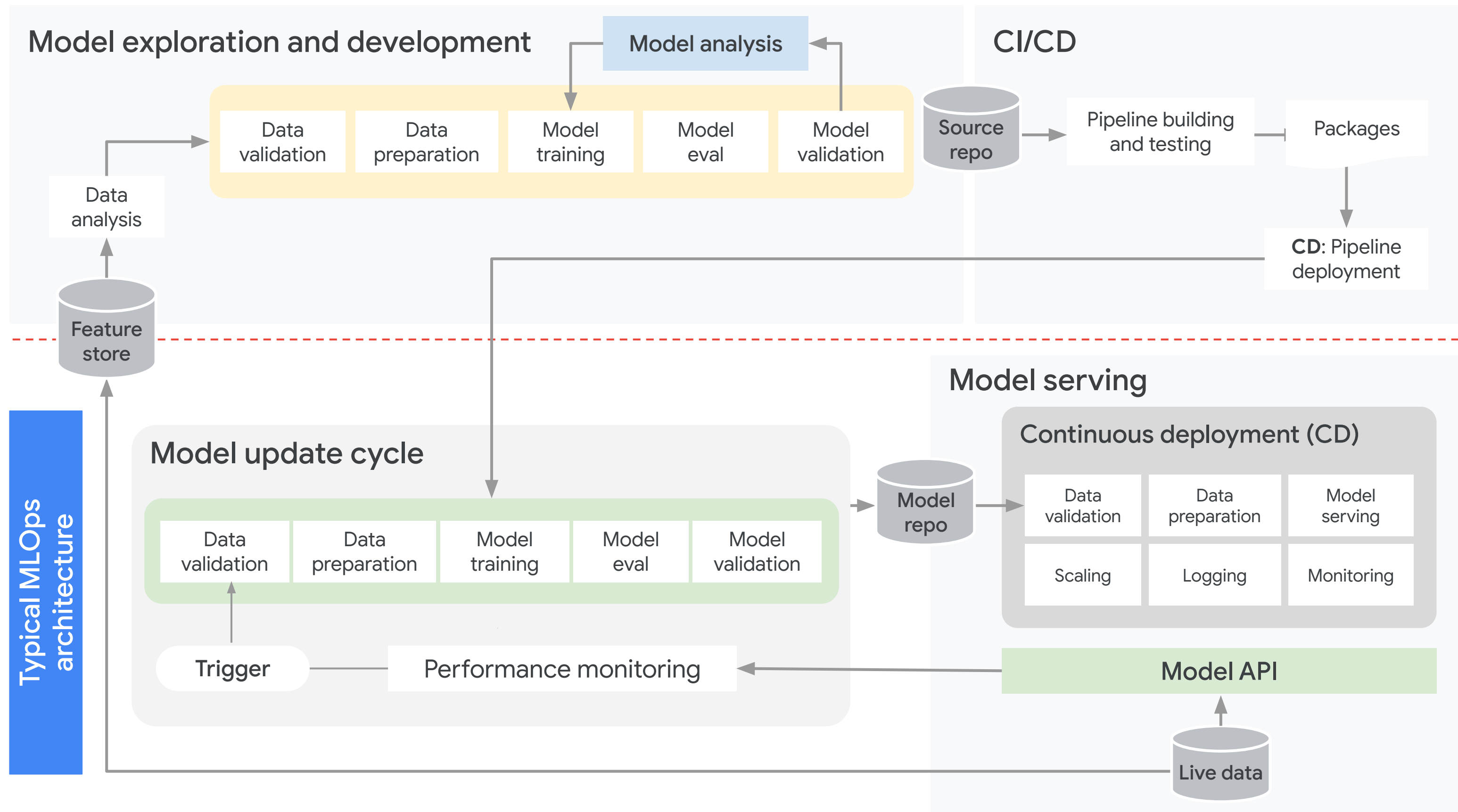
Concept Overview

Cloud Build Builders

Cloud Build Configuration

Cloud Build Triggers





Every container is a self-contained directory in repo

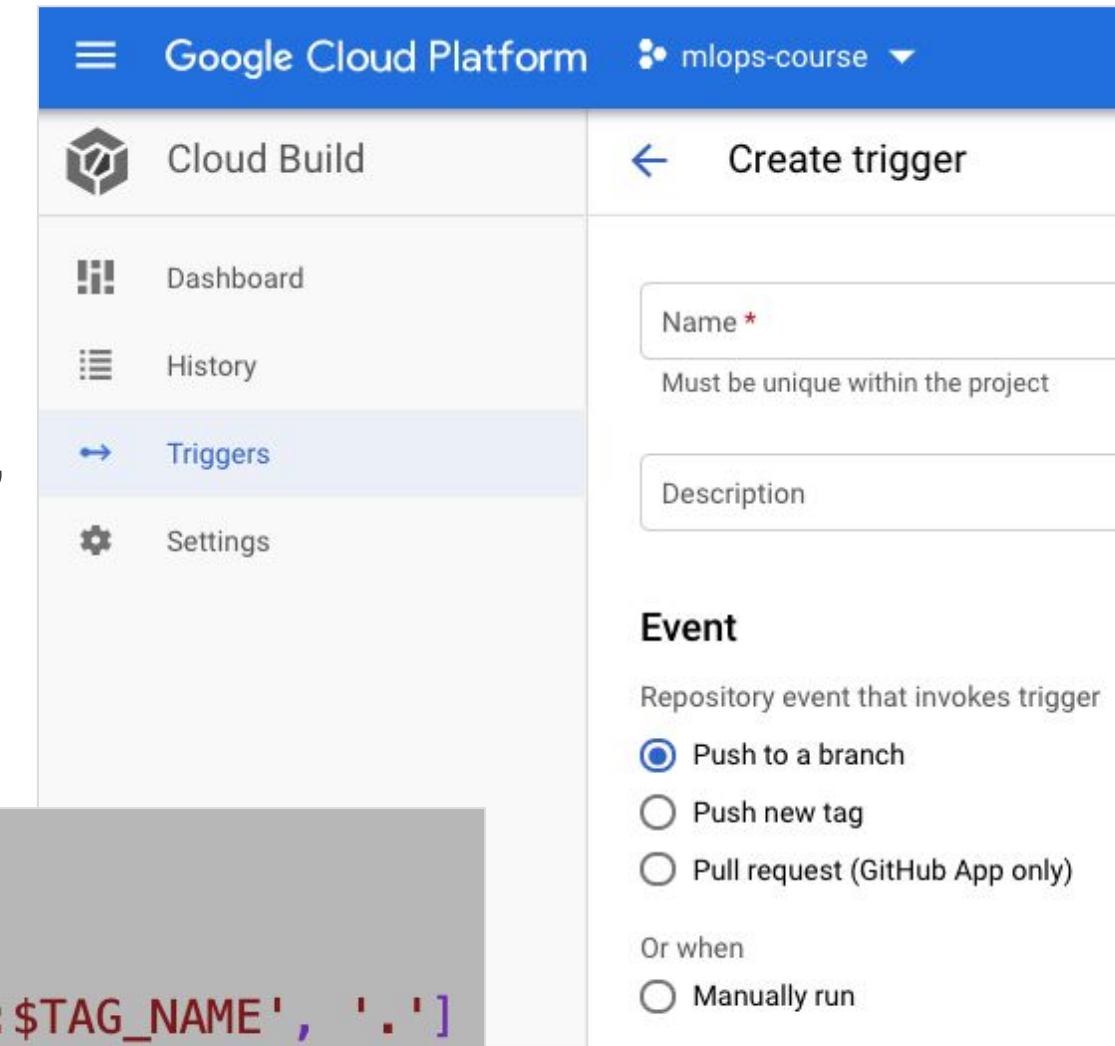
If any of these files is changed, you need to rebuild and push the Docker image.

Branch: master ▾ mlops-on-gcp / workshops / kfp-caip-sklearn / lab-03-kfp-cicd /		
BenoitDherin committed 195f772 3 days ago ... ✓		
..		
base_image	Initial draft	
exercises	created cloudbuild.yaml exercises	
kfp-cli	Initial draft	
pipeline	Initial draft	
trainer_image	Initial draft	
README.md	Initial draft	
cloudbuild.yaml	Initial draft	
lab-03.ipynb	reordered paragraph to help craft exercises	

For CI/CD, use a GitHub trigger to rebuild ML artifacts

Connect the GitHub repository to your Google Cloud account, and then trigger a Cloud Build from a GitHub trigger. Do this for every container in your ML pipeline.

```
20 steps:
21 # Build the trainer image
22 - name: 'gcr.io/cloud-builders/docker'
23   args: ['build', '-t', 'gcr.io/$PROJECT_ID/_TRAINER_IMAGE_NAME:$TAG_NAME', '.']
24   dir: $_PIPELINE_FOLDER/trainer_image
25
26 # Build the base image for lightweight components
27 - name: 'gcr.io/cloud-builders/docker'
28   args: ['build', '-t', 'gcr.io/$PROJECT_ID/_BASE_IMAGE_NAME:$TAG_NAME', '.']
29   dir: $_PIPELINE_FOLDER/base_image
30
31 # Compile the pipeline
32 - name: 'gcr.io/$PROJECT_ID/kfp-cli'
33   args:
34     - '-c'
```



Google Cloud Platform mlops-course

Cloud Build

← Create trigger

Name *

Must be unique within the project

Description

Event

Repository event that invokes trigger

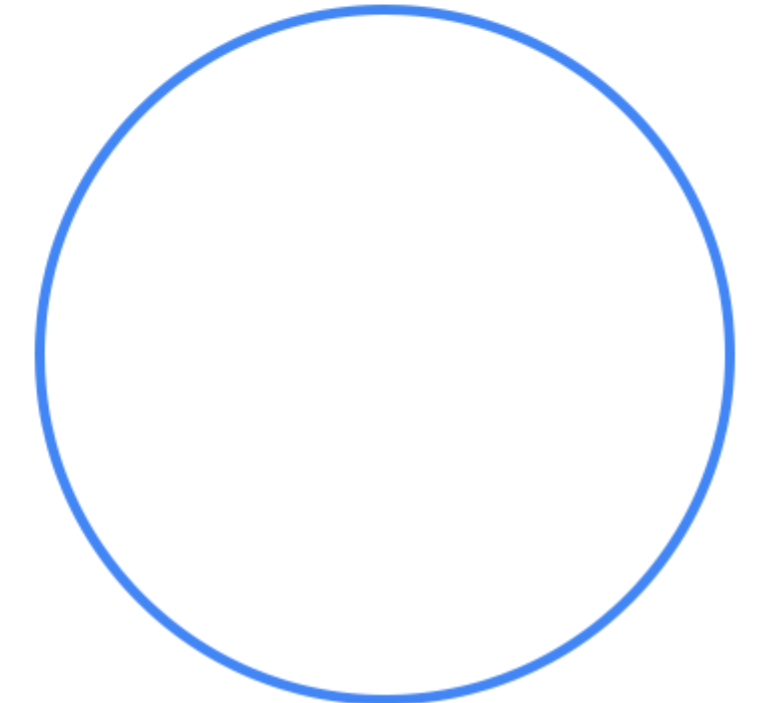
☒ Push to a branch

☐ Push new tag

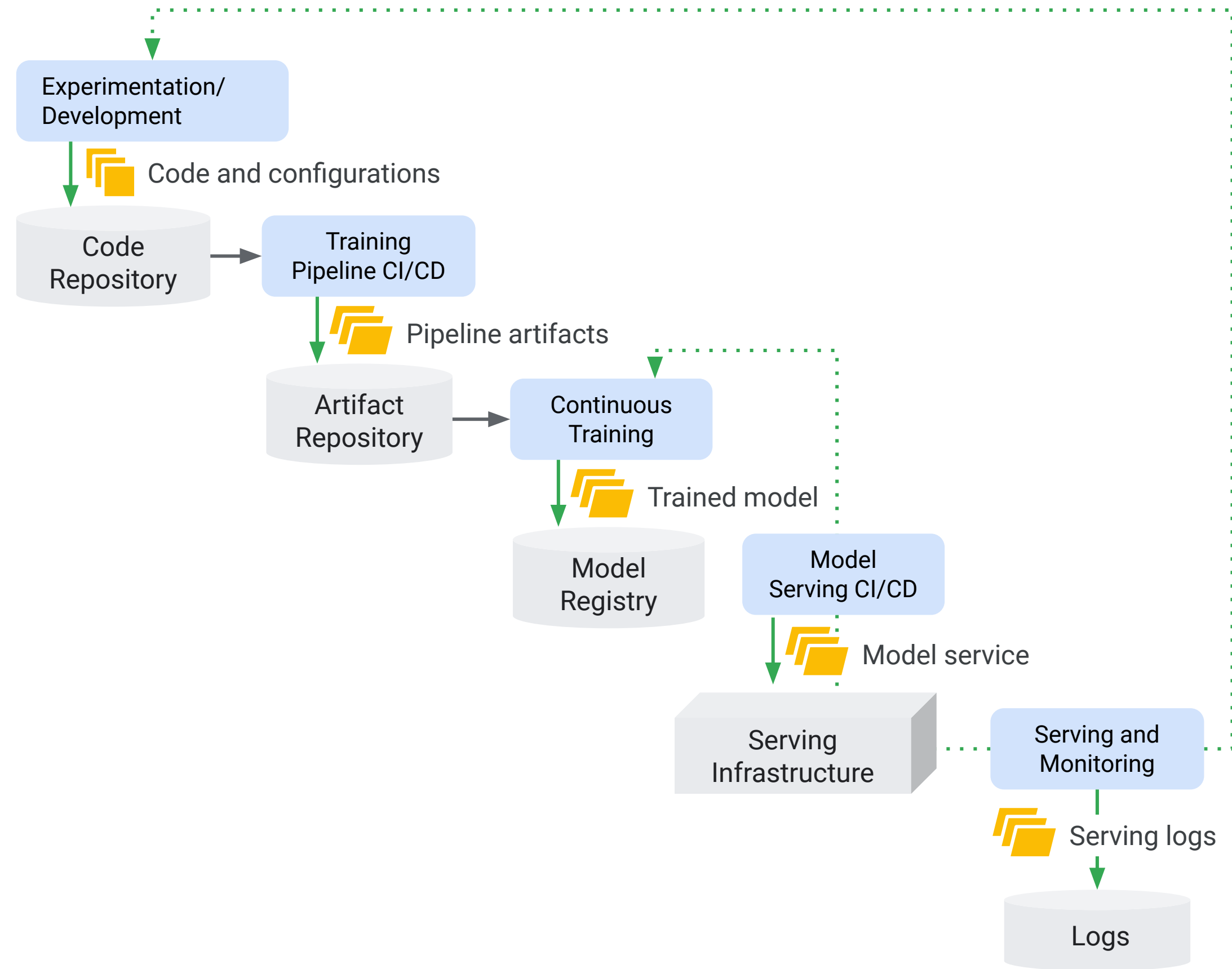
☐ Pull request (GitHub App only)

Or when

☐ Manually run



Where we are
headed next



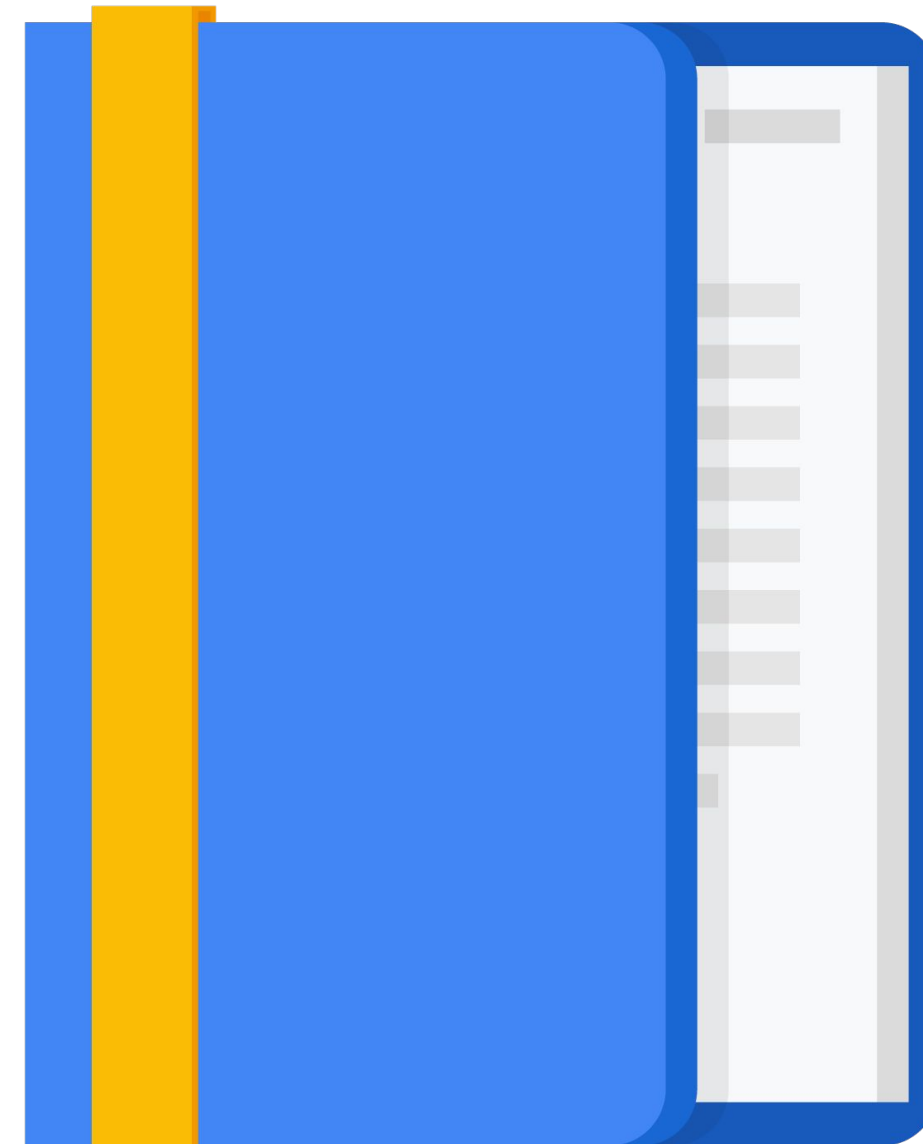
Agenda

Concept Overview

Cloud Build Builders

Cloud Build Configuration

Cloud Build Triggers



What are cloud builders?

Cloud configuration/provisioning actions that are packaged as Docker containers

Typical cloud builder actions:

- Building a Docker image from a Dockerfile
- Pushing a Docker image into a Google Cloud project registry
- Deploying a VM instance on Compute Engine
- Uploading a Kubeflow pipeline on CAIP Pipelines

What are cloud builders?

Standard

Already packaged config actions

Docker Registry:

`gcr.io/cloud-builders/`

Container Code:

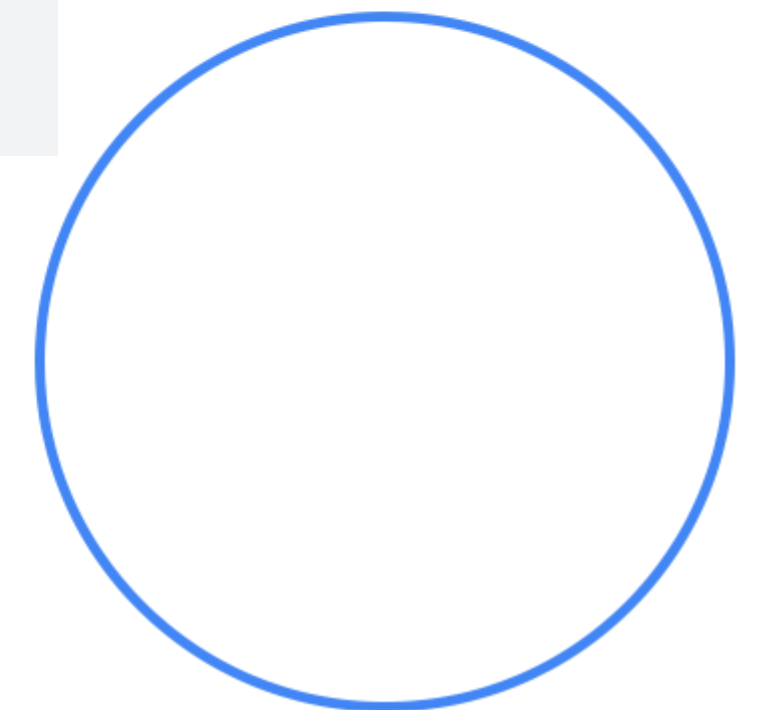
[GoogleCloudPlatform/cloud-builders](https://cloud.google.com/cloud-build/docs/container-code)

Custom

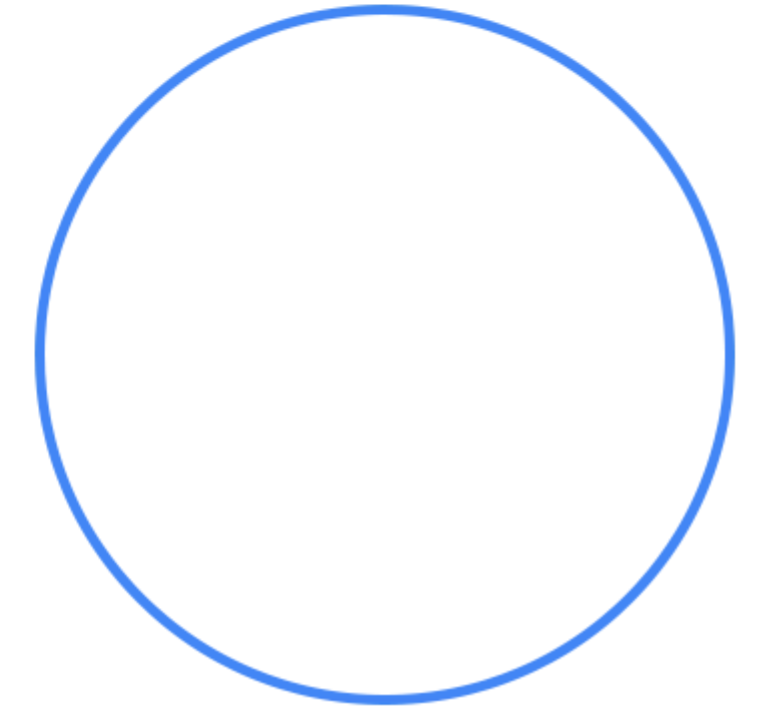
config actions packaged by you

Docker Registry:

`gcr.io/<YOUR_PROJECT>/`



Standard Cloud Builders




Standard cloud builders

Builder	Name	Example
bazel	<code>gcr.io/cloud-builders/bazel</code>	bazel example
docker	<code>gcr.io/cloud-builders/docker</code>	docker example
git	<code>gcr.io/cloud-builders/git</code>	git example
gcloud	<code>gcr.io/cloud-builders/gcloud</code>	gcloud example
gke-deploy	<code>gcr.io/cloud-builders/gke-deploy</code>	gke-deploy example


...wrap standard config tools as Docker containers















- `bazel` : runs the [bazel](#) tool
- `curl` : runs the [curl](#) tool
- `docker` : runs the [docker](#) tool
- `dotnet` : run the [dotnet](#) tool
- `gcloud` : runs the [gcloud](#) tool
- `gcs-fetcher` : efficiently fetches objects from Google Cloud
- `git` : runs the [git](#) tool
- `gke-deploy` : deploys an application to a Kubernetes cluster,
- `go` : runs the [go](#) tool
- `gradle` : runs the [gradle](#) tool
- `gsutil` : runs the [gsutil](#) tool
- `javac` : runs the [javac](#) tool
- `kubectl` : runs the [kubectl](#) tool
- `mvn` : runs the [maven](#) tool
- `npm` : runs the [npm](#) tool
- `wget` : runs the [wget](#) tool
- `yarn` : runs the [yarn](#) tool


[GoogleCloudPlatform](#) / [cloud-builders](#)

[Code](#)
[Issues 14](#)
[Pull requests](#)
[Actions](#)
[Projects](#)
[Security](#)
[Insights](#)


Branch: master
Go to file
Add file
Clone


bendory committed 237d8e7 7 days ago
663 commits
1 branch
0 tags


	.github	Create issue template	17 months ago
	bazel	Active voice.	29 days ago
	curl	Fix argument forwarding in notice entrypoints. (#686)	28 days ago
	docker	Fix argument forwarding in notice entrypoints. (#686)	28 days ago
	dotnet	Fix argument forwarding in notice entrypoints. (#686)	28 days ago
	gcloud	Configure git in cloud-sdk to use ADC. (#694)	22 days ago
	gcs-fetcher	Use the Google-hosted cloud-sdk image.	last month
	git	Whitespace.	22 days ago
	gke-deploy	Add kubectl apply --server-dry-run flag to gke-deployer (#696)	15 days ago
	go	Fixed broken modules example. (#701)	7 days ago
	gradle	Fix argument forwarding in notice entrypoints. (#686)	28 days ago
	gsutil	Configure git in cloud-sdk to use ADC. (#694)	22 days ago
	javac	Active voice.	29 days ago
	kubectl	Configure git in cloud-sdk to use ADC. (#694)	22 days ago

<https://github.com/GoogleCloudPlatform/cloud-builders/tree/master/docker>


Example: Docker cloud builder


 [GoogleCloudPlatform](#) / [cloud-builders](#)


[Code](#) [Issues 14](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)


 **rafikk** committed f5f0767 28 days ago [...](#)


..

 [examples/helloworld](#)

 [Dockerfile-19.03.8](#)

 [README.md](#)

 [cloudbuild.yaml](#)

 [notice.sh](#)

GCB rebrand. ([#309](#))

Document use of alternative official supported images. ([#680](#))

Active voice.

Deprecated docker builder. ([#671](#))

Fix argument forwarding in notice entrypoints. ([#686](#))


```
FROM launcher.gcr.io/google/ubuntu16_04
```

```
ARG DOCKER_VERSION=5:19.03.8~3-0~ubuntu-xenial
```

```
RUN apt-get -y update && \  
    apt-get -y install \  
        apt-transport-https \  
        ca-certificates \  
        curl \  
        make \  
        software-properties-common && \  
    curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add - && \  
    apt-key fingerprint 0EBFCD88 && \  
    add-apt-repository \  
        "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
        xenial \  
        edge" && \  
    apt-get -y update && \  
    apt-get -y install docker-ce=${DOCKER_VERSION} docker-ce-cli=${DOCKER_VERSION}
```

```
COPY notice.sh /usr/bin
```

```
ENTRYPOINT ["/usr/bin/notice.sh"]
```

When ran, /usr/bin/notice.sh is executed

Example: Docker cloud builder

notice.sh

```
#!/bin/sh
```

```
echo '
```

```
***** NOTICE *****
```

```
Alternative official `docker` images, including multiple versions across  
multiple platforms, are maintained by the Docker Team. For details, please  
visit https://hub.docker.com/\_/docker.
```

```
***** END OF NOTICE *****
```

```
'
```

```
/usr/bin/docker "$@"
```

When run, the container
passes its args to the
docker command.

Custom Cloud Builders



Custom Cloud Build builder

Dockerfile

```
FROM gcr.io/deeplearning-platform-release/base-cpu

RUN pip install kfp==0.2.5

ENTRYPOINT ["/bin/bash"]
```

Your custom Docker file

`IMAGE_URI='gcr.io/PROJECT_ID/IMAGE_NAME:TAG'`

Your project registry

`gcloud builds submit --timeout 15m --tag IMAGE_URI PATH`

Path to the directory where Docker package code is located

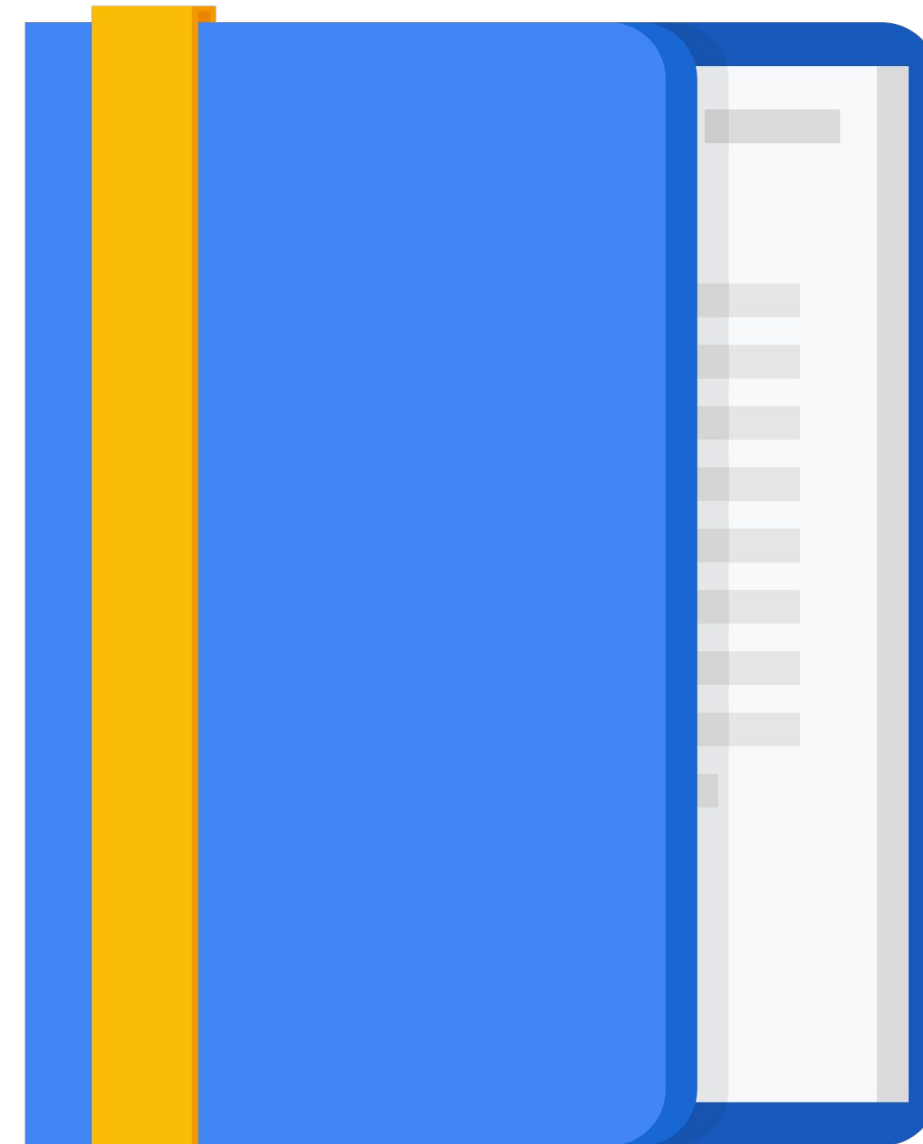
Agenda

Concept Overview

Cloud Build Builders

Cloud Build Configuration

Cloud Build Triggers



Cloud Build configuration file

cloudbuild.yaml

Describe the cloud builders to be run step by step

steps:

- name: 'gcr.io/cloud-builders/docker' ←
args: ['build', '-t', 'gcr.io/\$PROJECT_ID/\$_TRAINER_IMAGE_NAME:\$TAG_NAME', '.']
dir: \$_PIPELINE_FOLDER/trainer_image
- name: 'gcr.io/cloud-builders/docker' ←
args: ['build', '-t', 'gcr.io/\$PROJECT_ID/\$_BASE_IMAGE_NAME:\$TAG_NAME', '.']
dir: \$_PIPELINE_FOLDER/base_image

Running Cloud Build

```
gcloud builds submit . --config cloudbuild.yaml --substitutions $SUBSTITUTIONS
```

Path to the build
code directory

Cloud Build config file
describing the cloud
builders to be run

Variable substitutions to be
made in the cloudbuild.yaml
config file

A simple Cloud Build step

steps:

- name: 'gcr.io/cloud-builders/docker'
args: ['build', '-t', 'gcr.io/\$PROJECT_ID/\$_TRAINER_IMAGE_NAME:\$TAG_NAME', '.']
dir: \$_PIPELINE_FOLDER/trainer_image

The cloud builder container URI to be run

Substitution

The CWD in the Docker container from which the entrypoint is executed

The arguments to be passed to the container entrypoint



Persistence dir:

steps:

- **name:**
args:
dir: <rel path>

A cloud builder container has its current working directory:

- Set by default to /workspace in the container.
- Shared between steps.

This can be modified by specifying a **dir path** that:

- Will resolve to /workspace/<path>.
- Will still be shared between steps **if the path is relative**.

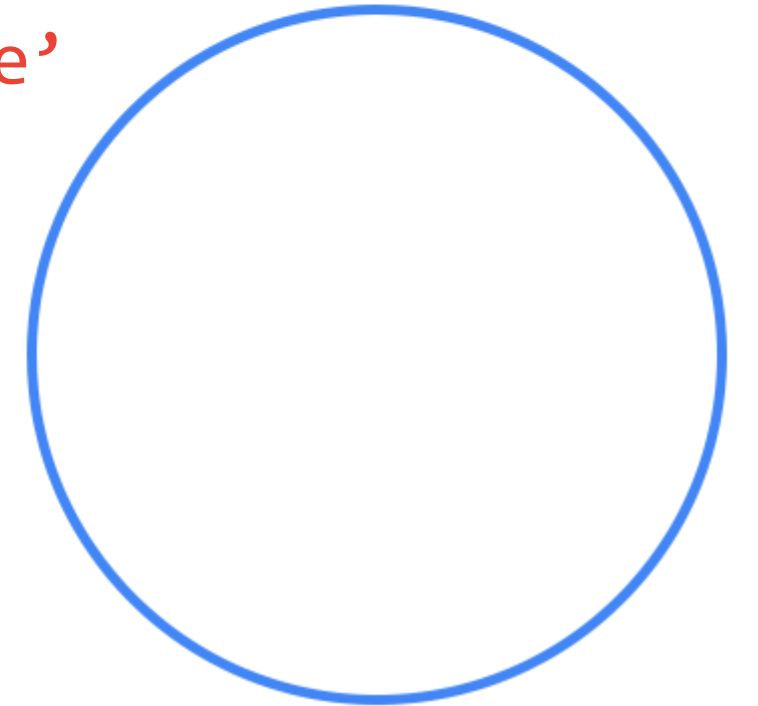
Caution: If the path is absolute, the CWD will not persist between steps.

Substitutions: \$_VARIABLE_NAME

steps:

```
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t', 'gcr.io/$PROJECT_ID/$_TRAINER_IMAGE_NAME:$TAG_NAME', '.']
  dir: $_PIPELINE_FOLDER/$_IMAGE_NAME
```

```
gcloud builds submit . --config cloudbuild.yaml \
  --substitutions '_PIPELINE_FOLDER=.,_IMAGE_NAME=trainer_base'
```



Custom cloud builder

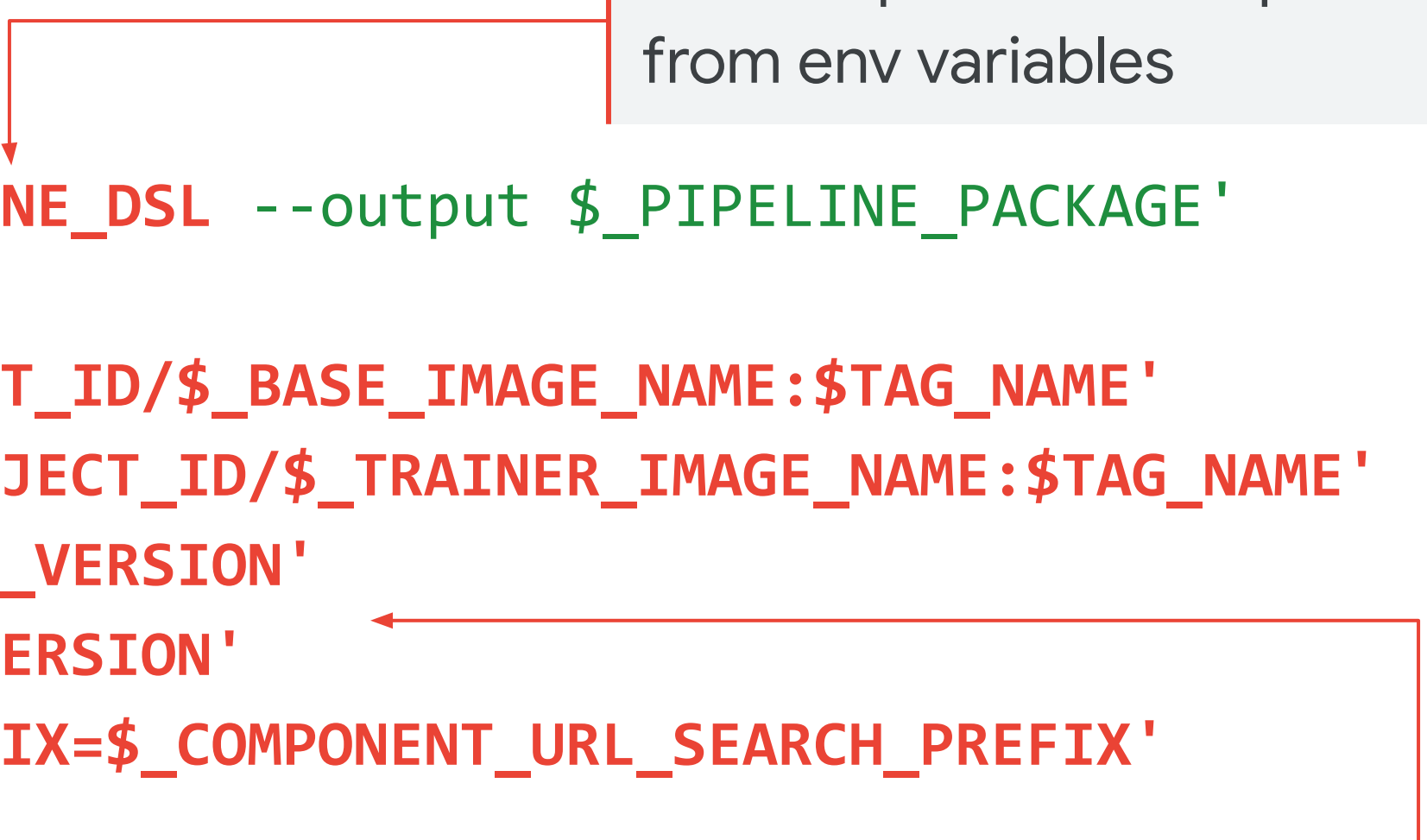
```
# Upload the pipeline
- name: 'gcr.io/$PROJECT_ID/kfp-cli'
  args:
    - '-c'
    - |
      kfp --endpoint $_ENDPOINT pipeline upload \
        -p ${_PIPELINE_NAME}_${TAG_NAME} $_PIPELINE_PACKAGE
  dir: $_PIPELINE_FOLDER/pipeline
```

No different from the standard cloud builders (but the registry is your own project registry)



Passing environment variables

```
- name: 'gcr.io/$PROJECT_ID/kfp-cli'
  args:
    - '-c'
    - 'dsl-compile --py $_PIPELINE_DSL --output $_PIPELINE_PACKAGE'
  env:
    - 'BASE_IMAGE=gcr.io/$PROJECT_ID/$_BASE_IMAGE_NAME:$TAG_NAME'
    - 'TRAINER_IMAGE=gcr.io/$PROJECT_ID/$_TRAINER_IMAGE_NAME:$TAG_NAME'
    - 'RUNTIME_VERSION=$_RUNTIME_VERSION'
    - 'PYTHON_VERSION=$_PYTHON_VERSION'
    - 'COMPONENT_URL_SEARCH_PREFIX=$_COMPONENT_URL_SEARCH_PREFIX'
    - 'USE_KFP_SA=$_USE_KFP_SA'
  dir: $_PIPELINE_FOLDER/pipeline
```



This script takes its input from env variables

That's how the env variables are passed to the script

Pushing images to Container Registry

Build the image locally on the build node

steps:

- name: 'gcr.io/cloud-builders/docker'
args: ['build', '-t', 'gcr.io/\$PROJECT_ID/\$_TRAINER_IMAGE_NAME:\$TAG_NAME', '.']
dir: \$_PIPELINE_FOLDER/trainer_image

Images:

- 'gcr.io/\$PROJECT_ID/\$_TRAINER_IMAGE_NAME:\$TAG_NAME'
- 'gcr.io/\$PROJECT_ID/\$_BASE_IMAGE_NAME:\$TAG_NAME'

Push the image to the registry

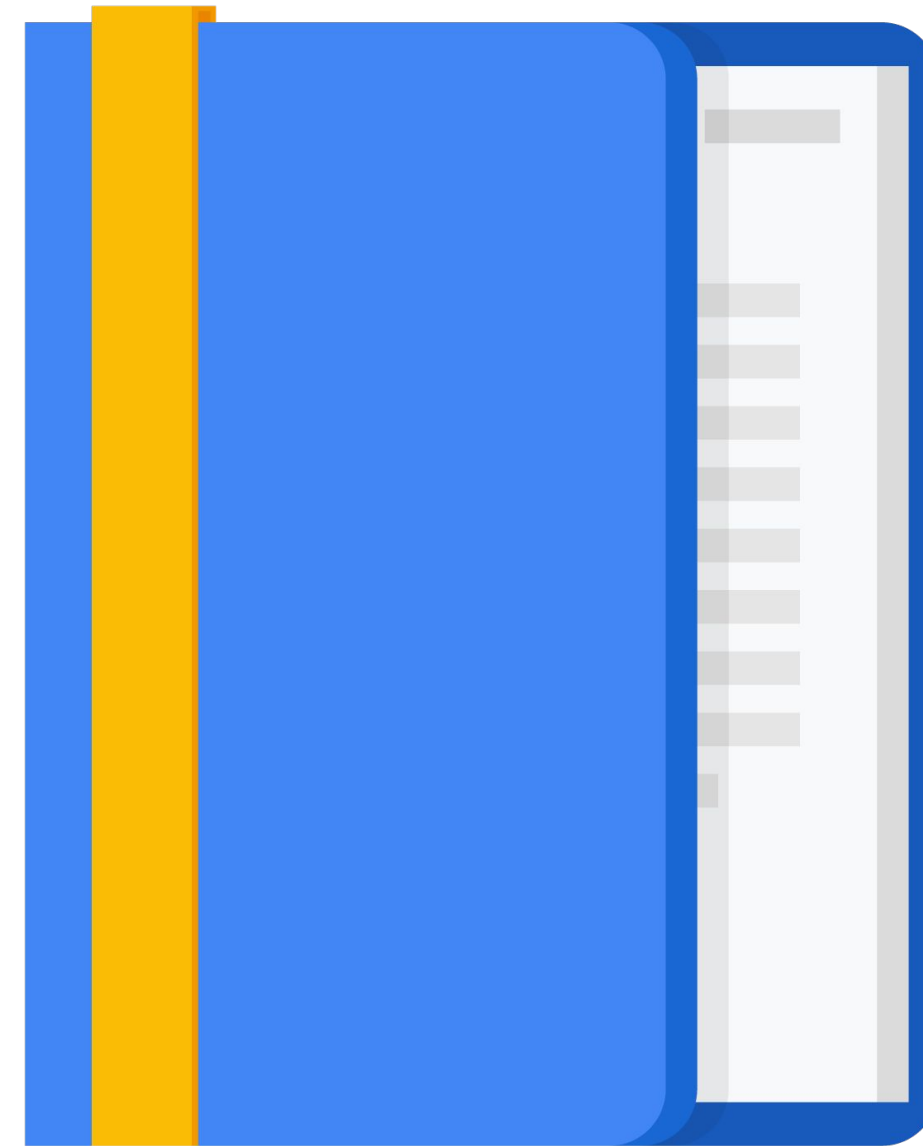
Agenda

Concept Overview

Cloud Build Builders

Cloud Build Configuration

Cloud Build Triggers



Manually executing a Cloud Build

1. Clone the Github repo with your cloudbuild.yaml and ML code on a build node.
 - The build node can be any node with the gcloud sdk properly authenticated:
 - JupyterLab VM
 - Cloud Shell
(if the containers to build are small)
 - Your laptop
 - A dedicated build VM on Cloud Compute
2. Run `gcloud builds submit` on the cloudbuild.yaml with the proper substitutions.

Manually executing a Cloud Build

```
In [ ]: 1 SUBSTITUTIONS=""
2 _ENDPOINT={},\
3 _TRAINER_IMAGE_NAME=trainer_image,\
4 _BASE_IMAGE_NAME=base_image,\
5 TAG_NAME=test,\
6 _PIPELINE_FOLDER=.,\
7 _PIPELINE_DSL=covertypes_training_pipeline.py,\
8 _PIPELINE_PACKAGE=covertypes_training_pipeline.yaml,\
9 _PIPELINE_NAME=covertypes_continuous_training,\
10 _RUNTIME_VERSION=1.15,\
11 _PYTHON_VERSION=3.7,\
12 _USE_KFP_SA=True,\
13 _COMPONENT_URL_SEARCH_PREFIX=https://raw.githubusercontent.com/kubeflow/pipelines/0.2.5/components/gcp/
14 """.format(ENDPOINT).strip()
```

```
In [ ]: 1 !gcloud builds submit . --config cloudbuild.yaml --substitutions {SUBSTITUTIONS}
```

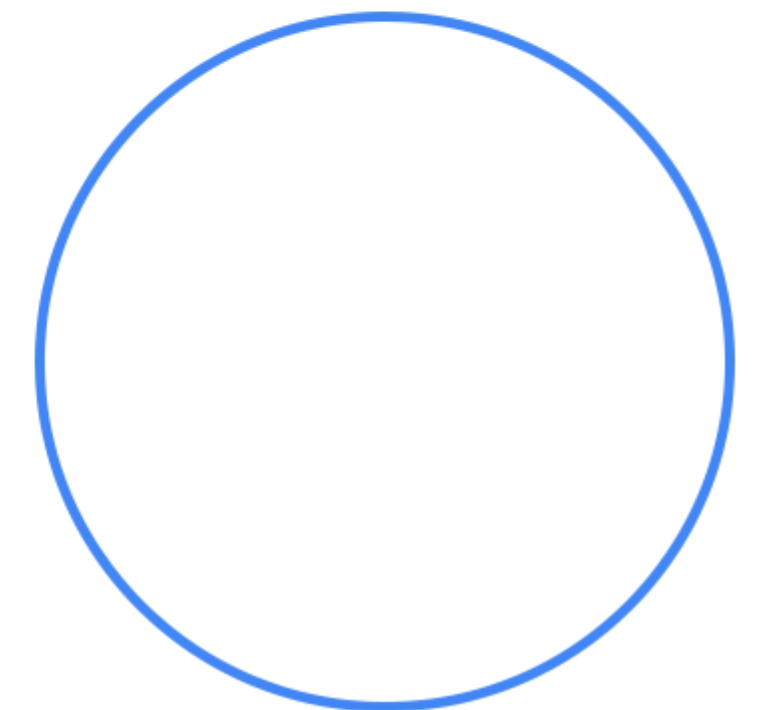
CI/CD: Automated Cloud Build triggers

When the ML code changes in Github via:

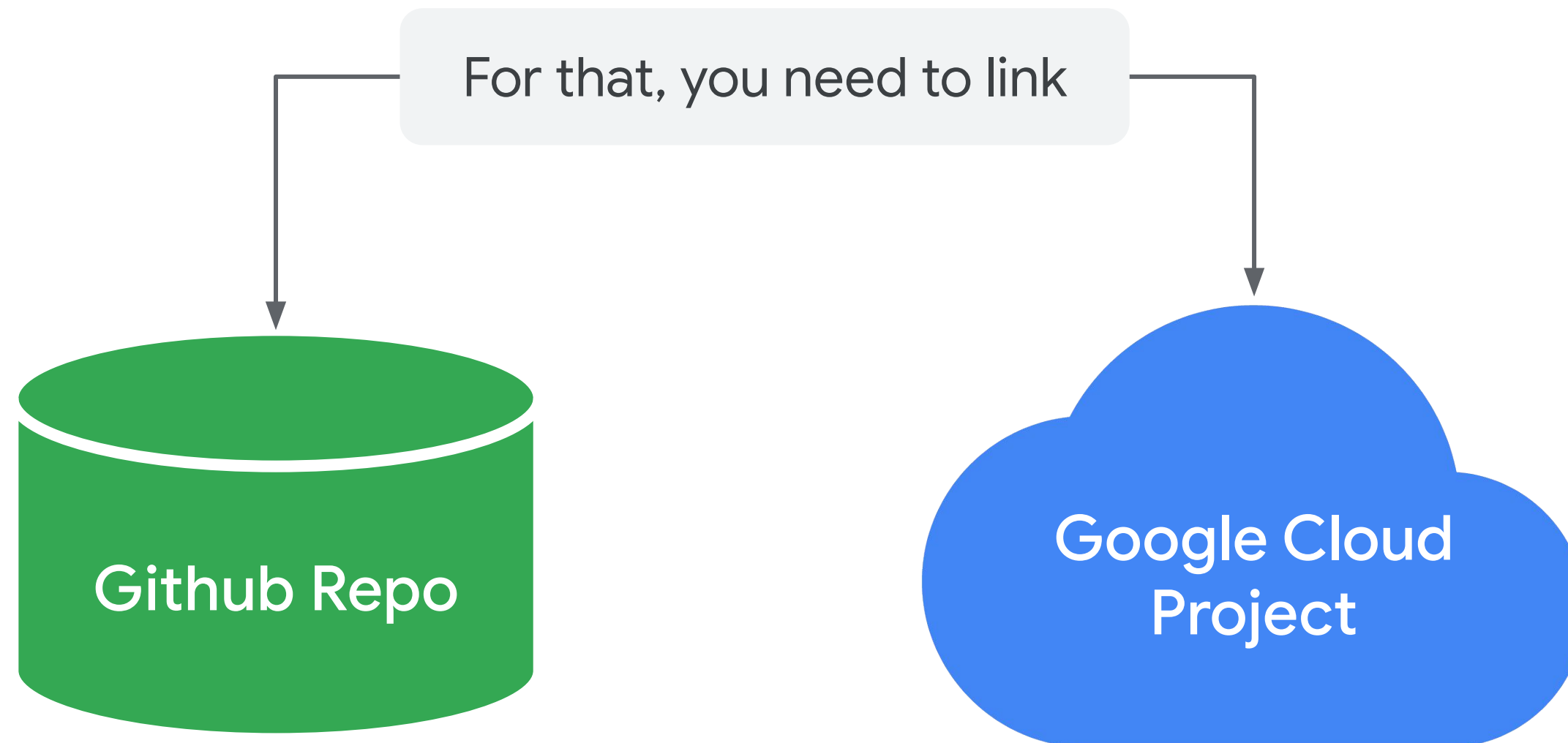
- Push on a branch
- New tag
- Pull request
- etc.

...you want a Cloud Build to be triggered automatically, so that the training artifacts are updated from the new code:



- Training containers
- Kubeflow pipelines
- etc.




CI/CD: Automated Cloud Build triggers



Set up your github repo to work with Cloud Build




[Marketplace](#) / [Apps](#) / Google Cloud Build




Application

Google Cloud Build

 You've already granted this app access to GitHub outside of GitHub Marketplace.

[Set up a plan](#) [Configure access](#)

 **Verified by GitHub**
GitHub confirms that this app meets the [requirements for verification](#).

Categories

[Continuous integration](#)
[GitHub Enterprise](#)

Google Cloud Build

Google Cloud Build lets you create fast, consistent, reliable builds across all languages. Automatically build containers or non-container artifacts on commits to your GitHub repository. Get complete control over defining custom workflows for building, testing, and deploying across multiple environments such as VMs, serverless, Kubernetes, or Firebase.

Allow your repo to be accessed by ~~Google~~ Cloud Build

Repository access

☐ **All repositories**

This applies to all current *and* future repositories.

☒ **Only select repositories**


 **Select repositories** ▼


Select at least one repository.

Save

Cancel

Add your repo to Cloud Build

 Google Cloud Platform

 **Google Cloud Build**

Connect your selected GitHub repositories to your Google Cloud Platform projects

With this GitHub application, you can connect your repositories to your Google Cloud projects and start using Cloud Build.

1 Project settings

2 Repository selection

3 Trigger settings

Project


mlops-course ▼ or Create new project


☒ I understand that GitHub content for the selected repositories will be transferred to this GCP project to provide the connected service. Members of this GCP project with sufficient permissions will be able to create and run triggers on these repositories, based on transferred GitHub content. I also understand that content from all GitHub app triggers in this GCP project may be transferred to GitHub in order to provide functionality like showing trigger names in GitHub build results. This will apply to all existing and future GitHub App triggers in this project. [Learn more](#)

Next

Cancel

Add your repo to Cloud Build

 Google Cloud Platform


 **Google Cloud Build**

Connect your selected GitHub repositories to your Google Cloud Platform projects

With this GitHub application, you can connect your repositories to your Google Cloud projects and start using Cloud Build.

☒ **Project settings** 2 **Repository selection** 3 Trigger settings

Select repositories to connect with your project



 Filter repositories


☒ **Select all repositories**


☒ BenoitDherin/mlops-on-gcp


Connect repository Cancel


Set up triggers


 Google Cloud Platform 

 Cloud Build

 Dashboard

 History


 Triggers

 Settings

Dashboard

Your dashboard shows the results of the latest builds for each build trigger. To populate your dashboard, you will need to set up build triggers.

[SET UP BUILD TRIGGERS](#)

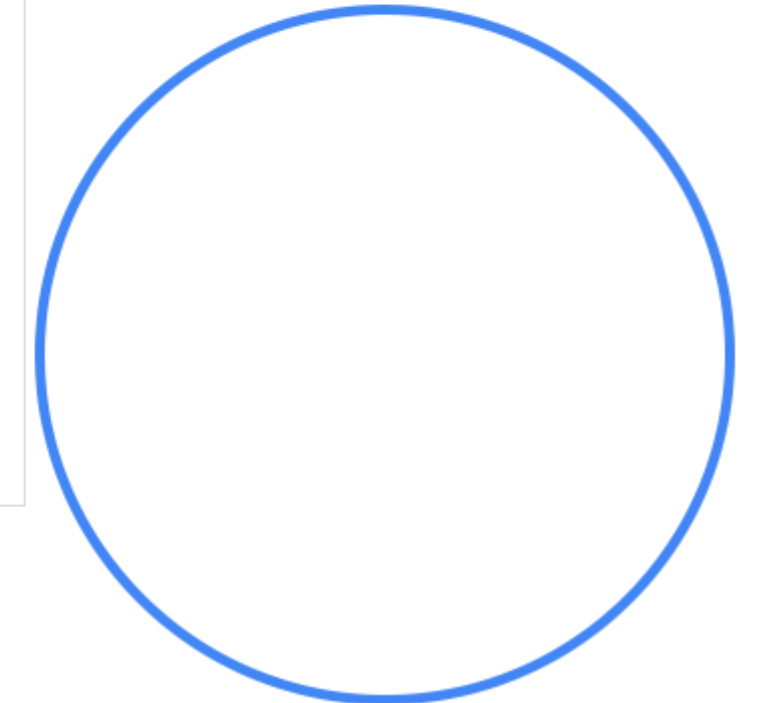
 Filter triggers

Specify the location of the cloudbuild.yaml file

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

Set up the substitution variable values

Variable	Value
<input type="text" value="_BASE_IMAGE_NAME"/>	<input type="text" value="base_image"/>
<input type="text" value="_COMPONENT_URL_SEARCH_I"/>	<input type="text" value="https://raw.githubusercontent.com/kubeflow/pipelines/0.2.5/components/gcp/"/>
<input type="text" value="_ENDPOINT"/>	<input type="text" value="Value"/>



Lab example

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

A Cloud Build trigger is now listed

☰

Google Cloud Platform

mlops-course ▾

🔍 Search products and resources ▾

🗨

?

🔔

⋮

🏠 Cloud Build

📊 Dashboard

📜 History

➡ Triggers

⚙ Settings

Triggers

➡ CONNECT REPOSITORY

+ CREATE TRIGGER

Integrate your working repositories in order to start creating build triggers. Build triggers automatically build containers based on source code or tag changes in a repository.

Repositories

ACTIVE INACTIVE ?

☰ Filter repositories

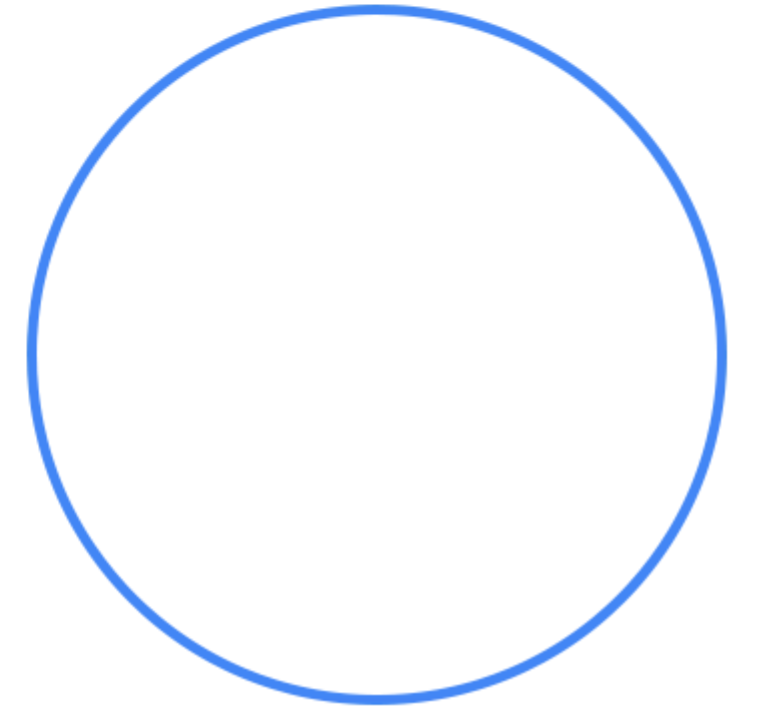
🔗 BenoitDherin/mlops-on-gcp ↗

Cloud Build GitHub App ⋮

Name	Description	Event	Filter	Build configuration	Status
mlops-tag-trigger	—	Push to tag	.*	workshops/kfp-caip-sklearn/lab-03-kfp-cicd/cloudbuild.yaml	Enabled ▾ Run trigger

Pushing a new tag will now trigger a Cloud Build

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



The build can be monitored

Google Cloud Platform

mlops-course

Search products and resources

Cloud Build

Dashboard

History

Triggers

Settings

Dashboard

Filter triggers

Running: BenoitDherin/mlops-on-gcp - mlops-tag-trigger

Latest Build	Duration	Trigger description	Source	Commit
6/25/20, 12:50 PM	00:00:14	-	BenoitDherin/mlops-on-gcp	3cbdf6d

Build History

[View all](#)

Average Duration

-

Pass - Fail %

-

The build can be monitored

Google Cloud Platform

mlops-course

Search products and resources

Cloud Build

Dashboard

History

Triggers

Settings

Build details

Running: 4c7c5e40

Started on Jun 25, 2020, 12:50:36 PM

Trigger

mlops-tag-trigger

Source

BenoitDherin/mlops-on-gcp

Tag

v1

Commit

3cbdf6d

Steps	Duration	BUILD LOG	EXECUTION DETAILS	BUILD ARTIFACTS
<div>Build Summary</div> <div>4 Steps</div>	00:01:00	<div><div><input type="checkbox"/> Wrap lines</div><div><input type="checkbox"/> Show newest entries first</div><div><div>↑</div><div>↓</div></div><div><div>EXPAND</div><div>VIEW R</div></div></div>		
<div>0: gcr.io/cloud-builders/docker</div> <div>build -t gcr.io/mlops-course/trainer_ima...</div>	-	<div>1 starting build "4c7c5e40-e6b0-40c0-8a14-d1d03ab99446"</div> <div>2</div> <div>3 FETCHSOURCE</div> <div>4 Fetching storage object: gs://354711844521.cloudbuild-source.googleusercontent.com/3cbdf6d764b6eaa0ee35349ec5ea43e45e081</div> <div>5 Copying gs://354711844521.cloudbuild-source.googleusercontent.com/3cbdf6d764b6eaa0ee35349ec5ea43e45e0817f0-c370416c-835a</div> <div>6 / [0 files][0.0 B/ 3.9 MiB]</div> <div>7 / [1 files][3.9 MiB/ 3.9 MiB]</div> <div>8 Operation completed over 1 objects/3.9 MiB.</div> <div>9 BUILD</div> <div>10 Starting Step #0</div> <div>11 Step #0: Already have image (with digest): gcr.io/cloud-builders/docker</div> <div>12 Step #0:</div> <div>13 Step #0: ***** NOTICE *****</div> <div>14 Step #0:</div> <div>15 Step #0: Alternative official `docker` images, including multiple versions across</div> <div>16 Step #0: multiple platforms, are maintained by the Docker Team. For details, please</div> <div>17 Step #0: visit https://hub.docker.com/_/docker.</div> <div>18 Step #0:</div>		
<div>1: gcr.io/cloud-builders/docker</div> <div>build -t gcr.io/mlops-course/base_imag...</div>	-			
<div>2: gcr.io/mlops-course/kfp-cli</div> <div>-c dsl-compile --py covertime_training_pi...</div>	-			
<div>3: gcr.io/mlops-course/kfp-cli</div> <div>-c kfp --endpoint 4979ddc3166669ff-dot...</div>	-			

After the build, new artifacts are available

☰

Google Cloud Platform

mlops-course ▾

🔍 Search products and resources

📦 Container Registry

📄 Images

⚙️ Settings

Repositories

🔄 REFRESH

mlops-course

🔍 Filter

All hostnames ▾ ?

Name ^	Hostname
📦 base_image	gcr.io
📦 kfp-cli	gcr.io
📦 kfp-dev	gcr.io
📦 trainer_image	gcr.io

After the build, new artifacts are available

Getting Started

Pipelines

Experiments

Artifacts

Executions

Archive

Pipelines

Filter pipelines

	Pipeline name	Description
<input type="checkbox"/>	▶ covertype_continuous_training	
<input type="checkbox"/>	▶ [Tutorial] DSL - Control structures	source code Shows how to use conditional execution and exit handlers. This pipeline will randomly fail to demonstr...
<input type="checkbox"/>	▶ [Tutorial] Data passing in python com...	source code Shows how to pass data between python components.
<input type="checkbox"/>	▶ [Demo] TFX - Taxi Tip Prediction Mod...	source code GCP Permission requirements . Example pipeline that does classification with model analysis based on...
<input type="checkbox"/>	▶ [Demo] XGBoost - Training with Conf...	source code GCP Permission requirements . A trainer that does end-to-end distributed training for XGBoost models.

Now the new
pipeline is ready
to be run

Getting Started

Pipelines

Experiments

Artifacts

Executions

Archive

Documentation

Github Repo

AI Hub Samples

<

Experiments

← Start a run

Run details

Pipeline *

covertime_continuous_training

Choose

Pipeline Version *

covertime_continuous_training

Choose

Run name *

Run of covertime_continuous_training (27ea4)

Description (optional)

This run will be associated with the following experiment

Experiment *

Choose

Run Type

☒ One-off

☐ Recurring

Run parameters

Specify parameters required by the pipeline

project_id

region

source_table_name

gcs_root

Lab

CI/CD for a KFP Pipeline

In this lab you will walk through authoring a Cloud Build CI/CD workflow that automatically builds and deploys a KFP pipeline.



[https://github.com/GoogleCloudPlatform/mlops-on-gcp/blob/master/
workshops/kfp-caip-sklearn/lab-03-kfp-cicd/exercises/lab-03.ipynb](https://github.com/GoogleCloudPlatform/mlops-on-gcp/blob/master/workshops/kfp-caip-sklearn/lab-03-kfp-cicd/exercises/lab-03.ipynb)

cloud.google.com