ML Pipelines on Google Cloud

Course ·
29% complete
Professional Machine Learning Engineer Certification Learning Path navigate_next ML Pipelines on Google Cloud navigate_next Pipeline orchestration with TFX

TFX on Cloud AI Platform Pipelines

2 hours 40 minutes Free

Overview

In this lab, you use utilize the following tools and services to deploy and run a TFX pipeline on Google Cloud that automates the development and deployment of a TensorFlow 2.3 WideDeep Classifier to predict forest cover from cartographic data:

- The TFX CLI utility to build and deploy a TFX pipeline.
- A hosted AI Platform Pipeline instance (Kubeflow Pipelines) for TFX pipeline orchestration.
- A AI Platform Training job for model training and flock management for parallel tuning trials.
- AI Platform Prediction as a model server destination for blessed pipeline model versions.
- CloudTuner and AI Platform Vizier for advanced model hyperparameter tuning using the Vizier algorithm.
- You will then create and monitor pipeline runs using the TFX CLI as well as the KFP UI.

Objectives

- Use the TFX CLI to build a TFX pipeline.
- Deploy a TFX pipeline version without tuning to a hosted AI Platform Pipelines instance.
- Create and monitor a TFX pipeline run using the TFX CLI.
- Deploy a new TFX pipeline version with tuning enabled to a hosted AI Platform Pipelines instance.
- Create and monitor another TFX pipeline run directly in the KFP UI.

Setup

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time.
   There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
   If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

Note: Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Activate Cloud Shell

Cloud Shell is a virtual machine that contains development tools. It offers a persistent 5-GB home directory and runs on Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources. `gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab completion.

1. Click the **Activate Cloud Shell** button (>_) at the top right of the console.
2. Click **Continue**.
   It takes a few moments to provision and connect to the environment. When you are connected, you are also authenticated, and the project is set to your *PROJECT_ID*.

Sample commands

- List the active account name:

```
gcloud auth list
```

(Output)

```
Credentialed accounts: - <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts: - google1623327_student@qwiklabs.net
```

- List the project ID:

```
gcloud config list project
```

(Output)

```
[core] project = <project_ID>
```

(Example output)

```
[core] project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide.

Initialize lab resources

1. In Cloud Shell, run the following commands to initialize the lab resources:

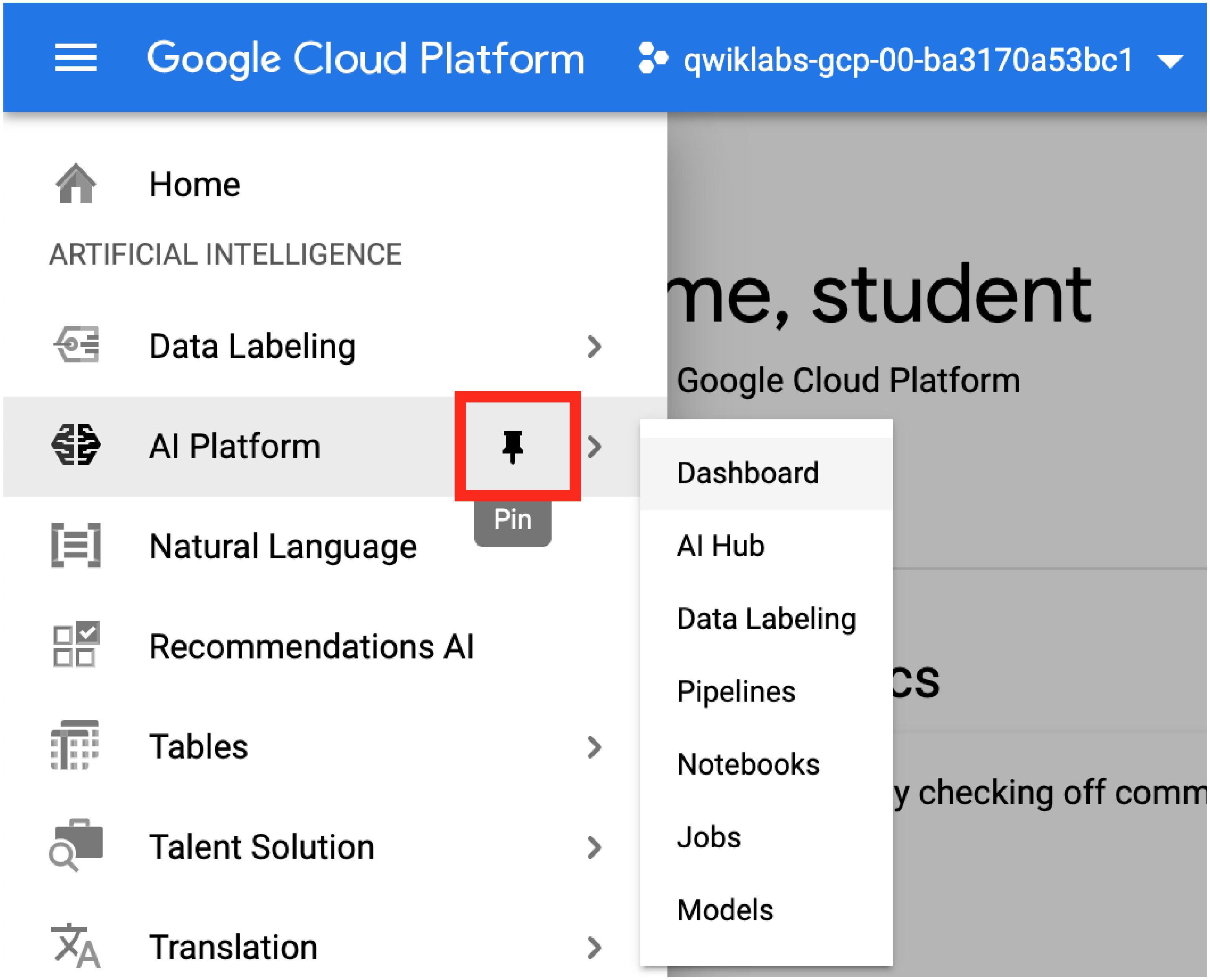
```
gsutil cp gs://cloud-training/TFX-Script/tfx-init.sh . sh tfx-init.sh
```

Note: If there are warnings or errors, feel free to ignore them.

Task 1. Create an instance of AI Platform Pipelines

In this task, you deploy Kubeflow Pipelines as a Kuberenetes App, which are solutions with simple click to deploy to Google Kubernetes Engine and that have the flexibility to deploy to Kubernetes clusters on-premises or in third-party clouds. You will see Kubeflow Pipelines integrated into your Google Cloud environment as **AI Platform Pipelines**. If interested, learn more about Kubeflow Pipelines in the Kubeflow Introduction during installation steps.

1. In the Google Cloud Console, on the Navigation menu, scroll down to **AI Platform** and pin the section for easier access later in the lab.



2. Navigate to **AI Platform** > **Pipelines**.
3. Then click **New Instance**.
4. Click **Configure**.

5. To create cluster select **Zone** as then check **Allow access to the following Cloud APIs**, leave the name as is, and then click **Create New Cluster**.

NOTE that for Endpoint, do not include the "https://" part, all other parts are OK. See below example

Note: The cluster creation will take 3 - 5 minutes. You need to wait until this step completes before you proceed to the next step.

6. Scroll to the bottom of the page, accept the marketplace terms, and click **Deploy**. You will see the individual services of KFP deployed to your GKE cluster. Wait for the deployment to finish before proceeding to the next task.
7. In **Cloud Shell**, run the following to configure kubectl command line access

```
gcloud container clusters get-credentials cluster-1 --zone {{{project_0.default_zone|place_holder_text}}} --project {{{project_0.project_id|place_holder_text}}}
```

8. In **Cloud Shell**, run the following to get the ENDPOINT of your KFP deployment

kubectl describe configmap inverse-proxy-config | grep googleusercontent.com Important: In a later task, you will need to set the endpoint for your KFP in one of the cells in your notebook. Remember to use the above output as your **ENDPOINT**.

Click **Check my progress** to verify the objective. Creating an instance of AI Platform Pipelines . **Endpoint**

NOTES:

As per GCP, https://9f1a88f39dc64af-dot-us-central1.pipelines.googleusercontent.com

client = kfp.Client(host= 9f1a88f39dc64af-dot-us-central1.pipelines.googleusercontent.com)

Store

gs://qwiklabs-gcp-04-f6502ae2aede-kubeflowpipelines-default/

Service acct

tfx-tuner-caip-service-account@qwiklabs-gcp-04-f6502ae2aede.iam.gserviceaccount.com

```
env: PIPELINE_NAME=tfx_covertype_continuous_training
env: MODEL_NAME=tfx_covertype_classifier
env: DATA_ROOT_URI=gs://cloud-training/OCBL203/workshop-datasets
```

```
env: KUBEFLOW_TFX_IMAGE=gcr.io/qwiklabs-gcp-04-f6502ae2aede/tfx_covertype_continuous_training
env: RUNTIME_VERSION=2.3
env: PYTHON_VERIONS=3.7
env: USE_KFP_SA=False
env: ENABLE_TUNING=False
```

Task 2. Access AI Platform Notebooks

An instance of AI Platform Notebooks is used as a primary experimentation/development workbench.

To launch AI Platform Notebooks:

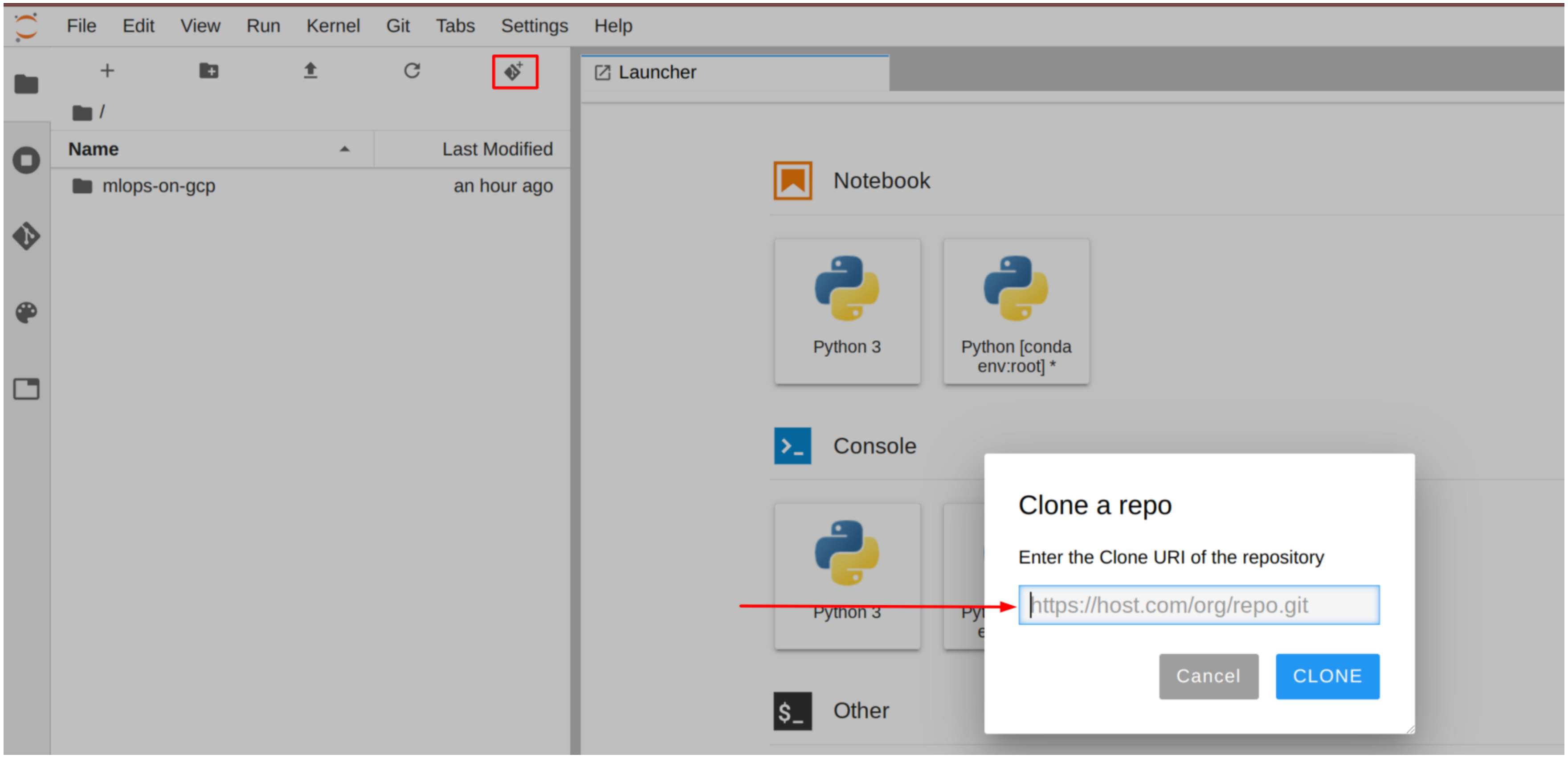
1. Click on the **Navigation Menu** and navigate to **AI Platform**, then to **Workbench**.
2. You should see the `tfx-on-googlecloud` notebook preprovisioned for you. If not, wait a few minutes and refresh the page.
3. Click **Open JupyterLab**. A JupyterLab window will open in a new tab.

Task 3. Clone the example repo within your AI Platform Notebooks instance

To clone the `mlops-on-gcp` notebook in your JupyterLab instance:

1. In JupyterLab, click the **Terminal** icon to open a new terminal.
2. At the command-line prompt, type in the following command and press Enter:

   git clone https://github.com/GoogleCloudPlatform/mlops-on-gcp **Note:** If the cloned repo does not appear in the JupyterLab UI, you can use the top line menu and under **Git** > **Clone a repository**, clone the repo (https://github.com/GoogleCloudPlatform/mlops-on-gcp) using the UI.



Clone a repo

Enter the Clone URI of the repository

https://host.com/org/repo.git

3. Confirm that you have cloned the repository by double clicking on the `mlops-on-gcp` directory and ensuring that you can see its contents. The files for all the Jupyter notebook-based labs throughout this course are available in this directory.

Task 4. Navigate to the lab notebook

1. From the `mlops-labs/workshops/tfx-caip-tf23` folder execute the `install.sh` script to install TFX and KFP SDKs:

cd mlops-on-gcp/workshops/tfx-caip-tf23 ./install.sh

2. In the AI Platform Notebook, navigate to `mlops-labs/workshops/tfx-caip-tf23/lab-02-tfx-pipeline/labs` and open `lab-02.ipynb`.
3. Clear all the cells in the notebook (look for the **Clear** button on the notebook toolbar) and then **Run** the cells one by one.
4. When prompted, come back to these instructions to check your progress in the *Task 5. Run your training job in the cloud* section.

If you need more help, you may take a look at the complete solution by navigating to `mlops-on-gcp/workshops/tfx-caip-tf23/lab-02-tfx-pipeline/solutions` open `lab-02.ipynb`.

Task 5. Run your training job in the cloud

Test completed tasks - Compile the kubeflow pipeline

Click **Check my progress** to verify the objective. Complile the kubeflow pipeline

Test completed tasks - Deploy the pipeline package to AI Platform Pipelines

Click **Check my progress** to verify the objective. Deploy the pipeline package to AI Platform Pipelines

Test completed tasks - Create kubeflow pipeline run using TFX CLI

Click **Check my progress** to verify the objective. Create kubeflow pipeline run using TFX CLI

Congratulations!

In this lab, you learned how to manually build and deploy a TFX pipeline to AI Platform Pipelines and trigger pipeline runs from a notebook.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2022 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.

NOTES:

The solution seems to be installing CUDA tool kit *10.1*.

```
sudo apt-get install cuda-cudart-10-1 – OK
===
sudo apt-get update – NOT OK, as this pulls latest ver
sudo apt install nvidia-cuda-toolkit
==
sudo apt install libcudart.so.10.1  -- OK
sudo apt-get install libcudart10.1 -- ???

-----
2023-10-21 09:28:22.221204: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /usr/lo
...

--
```

```
pip install python-memcache
apt-get install python-memcache – ubuntuDebian
```

```
2023-10-21 09:28:22.221204: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror:
libcudart.so.10.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /usr/local/lib 2023-10-21 09:28:22.221252: I
tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
INFO:absl:tensorflow_ranking is not available: No module named 'tensorflow_ranking' INFO:absl:tensorflow_text is not available: No module named
'tensorflow_text' INFO:absl:Running driver for Pusher INFO:absl:MetadataStore with gRPC connection initialized INFO:absl:Adding KFP pod name tfx-covertype-
continuous-training-bkwb2-4135672384 to execution INFO:absl:Running executor for Pusher INFO:absl:Attempting to infer TFX Python dependency for beam
INFO:absl:Copying all content from install dir /tfx-src/tfx to temp dir /tmp/tmp4jadn89x/build/tfx INFO:absl:Generating a temp setup file at
/tmp/tmp4jadn89x/build/tfx/setup.py INFO:absl:Creating temporary sdist package, logs available at /tmp/tmp4jadn89x/build/tfx/setup.log INFO:absl:Added --
extra_package=/tmp/tmp4jadn89x/build/tfx/dist/tfx_ephemeral-0.25.0.tar.gz to beam args WARNING:googleapiclient.discovery_cache:file_cache is unavailable
when using oauth2client >= 4.0.0 or google-auth Traceback (most recent call last): File "/usr/local/lib/python3.7/dist-
packages/googleapiclient/discovery_cache/__init__.py", line 36, in autodetect from google.appengine.api import memcache ModuleNotFoundError: No module
named 'google.appengine' During handling of the above exception, another exception occurred: Traceback (most recent call last): File
"/usr/local/lib/python3.7/dist-packages/googleapiclient/discovery_cache/file_cache.py", line 33, in from oauth2client.contrib.locked_file import LockedFile
ModuleNotFoundError: No module named 'oauth2client.contrib.locked_file' During handling of the above exception, another exception occurred: Traceback
(most recent call last): File "/usr/local/lib/python3.7/dist-packages/googleapiclient/discovery_cache/file_cache.py", line 37, in from oauth2client.locked_file
import LockedFile ModuleNotFoundError: No module named 'oauth2client.locked_file' During handling of the above exception, another exception occurred:
Traceback (most recent call last): File "/usr/local/lib/python3.7/dist-packages/googleapiclient/discovery_cache/__init__.py", line 42, in autodetect from . import
file_cache File "/usr/local/lib/python3.7/dist-packages/googleapiclient/discovery_cache/file_cache.py", line 41, in "file_cache is unavailable when using
oauth2client >= 4.0.0 or google-auth" ImportError: file_cache is unavailable when using oauth2client >= 4.0.0 or google-auth INFO:googleapiclient.discovery:URL
being requested: GET https://www.googleapis.com/discovery/v1/apis/ml/v1/rest INFO:absl:Deploying to model with version v1697880528 to AI Platform for
serving: {'model_name': 'tfx_covertype_classifier', 'project_id': 'qwiklabs-gcp-02-37d0e2e14014', 'pythonVersion': '3.7', 'regions': ['us-east1'], 'runtimeVersion':
'2.3'} INFO:googleapiclient.discovery:URL being requested: POST https://ml.googleapis.com/v1/projects/qwiklabs-gcp-02-37d0e2e14014/models?alt=json
INFO:googleapiclient.discovery:URL being requested: POST https://ml.googleapis.com/v1/projects/qwiklabs-gcp-02-
37d0e2e14014/models/tfx_covertype_classifier/versions?alt=json Traceback (most recent call last): File "/tfx-
src/tfx/extensions/google_cloud_ai_platform/runner.py", line 420, in deploy_model_for_aip_prediction body=body, parent=model_name).execute() File
"/usr/local/lib/python3.7/dist-packages/googleapiclient/_helpers.py", line 134, in positional_wrapper return wrapped(*args, **kwargs) File
"/usr/local/lib/python3.7/dist-packages/googleapiclient/http.py", line 898, in execute raise HttpError(resp, content, uri=self.uri)
googleapiclient.errors.HttpError: During handling of the above exception, another exception occurred: Traceback (most recent call last): File "/tfx-
src/tfx/orchestration/kubeflow/container_entrypoint.py", line 360, in main() File "/tfx-src/tfx/orchestration/kubeflow/container_entrypoint.py", line 353, in
main execution_info = launcher.launch() File "/tfx-src/tfx/orchestration/launcher/base_component_launcher.py", line 209, in launch
copy.deepcopy(execution_decision.exec_properties)) File "/tfx-src/tfx/orchestration/launcher/in_process_component_launcher.py", line 72, in _run_executor
copy.deepcopy(input_dict), output_dict, copy.deepcopy(exec_properties)) File "/tfx-src/tfx/extensions/google_cloud_ai_platform/pusher/executor.py", line
112, in Do job_labels, File "/tfx-src/tfx/extensions/google_cloud_ai_platform/runner.py", line 429, in deploy_model_for_aip_prediction .format(e))
RuntimeError: Creating model verseion to AI Platform failed:
```