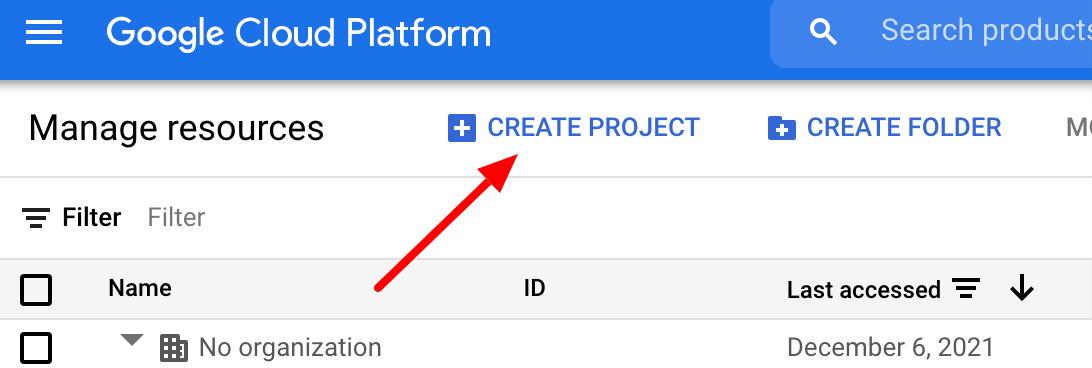
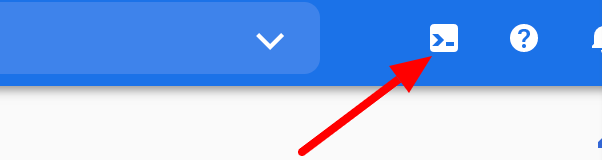
This file documents creating a ML Pipeline using Vertex Pipelines.

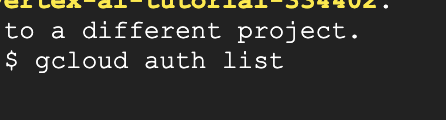
1. The first step is to create a Google Project and enable billing in it.



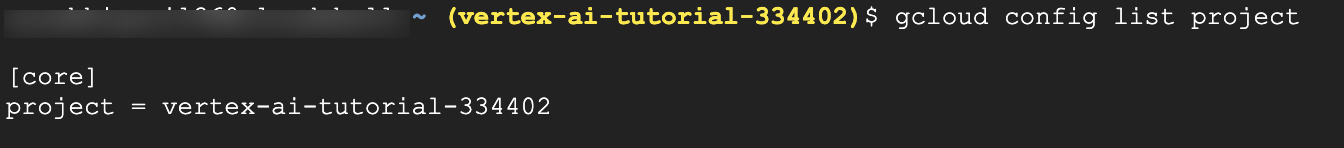
Then from the console activate cloud shell



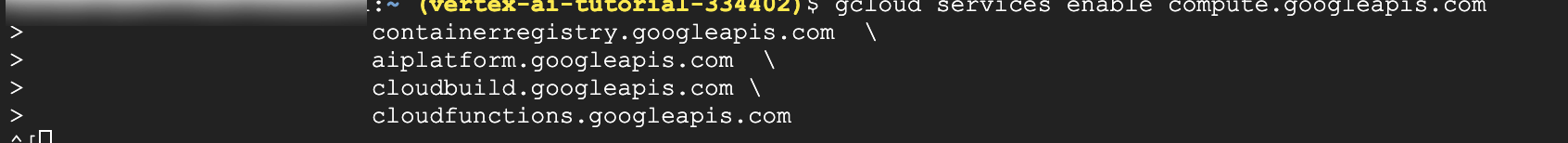
Check cloud shell is authorized by running the following command in the terminal:



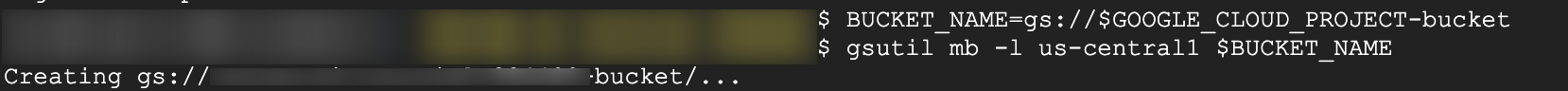
Check and ensure google cloud recognizes project using:



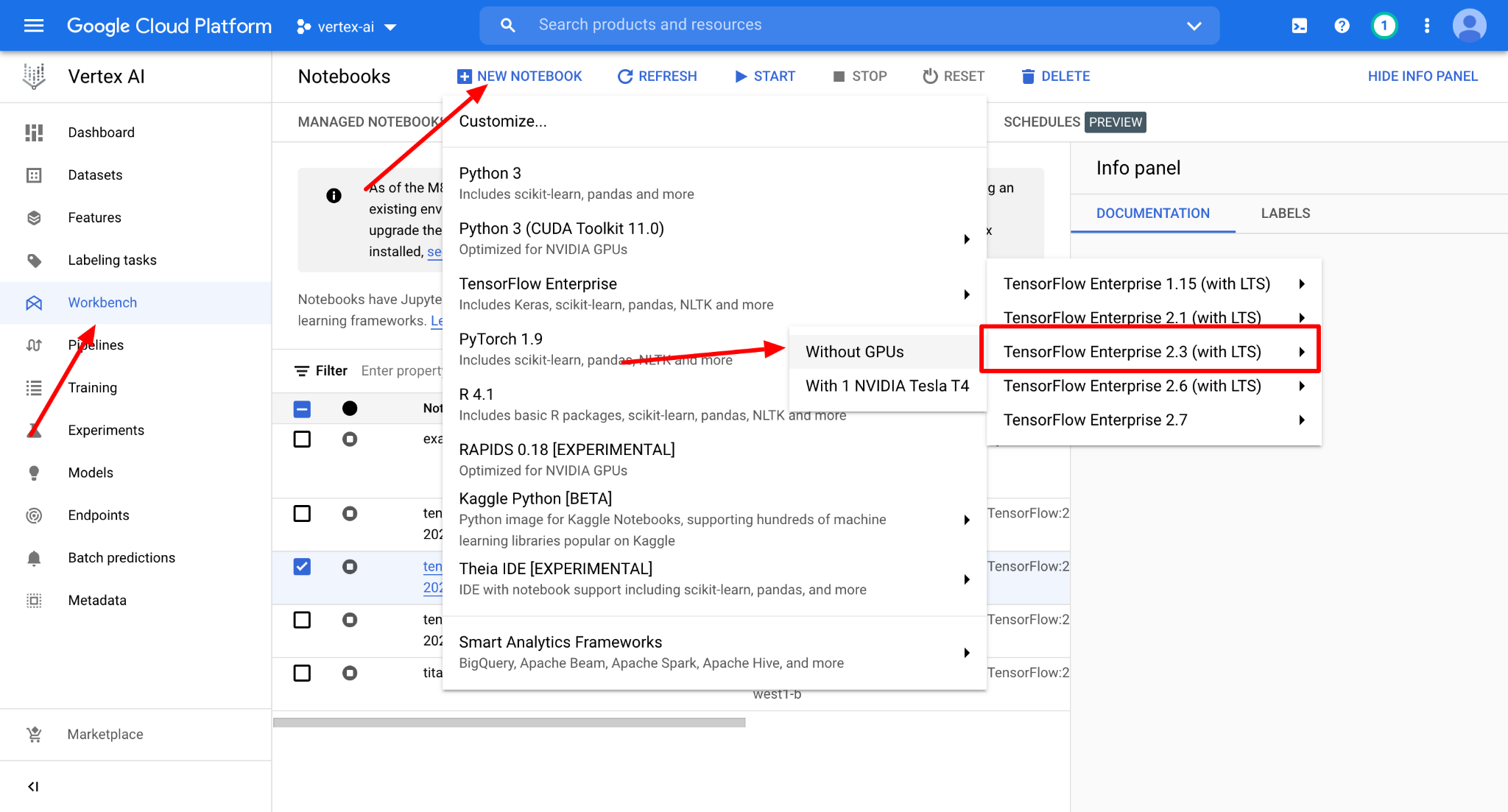
The following commands give project the necessary rights:



2. Then create a cloud storage bucket using the following commands in a terminal in the project. This is required to save model assets and datasets for custom training.

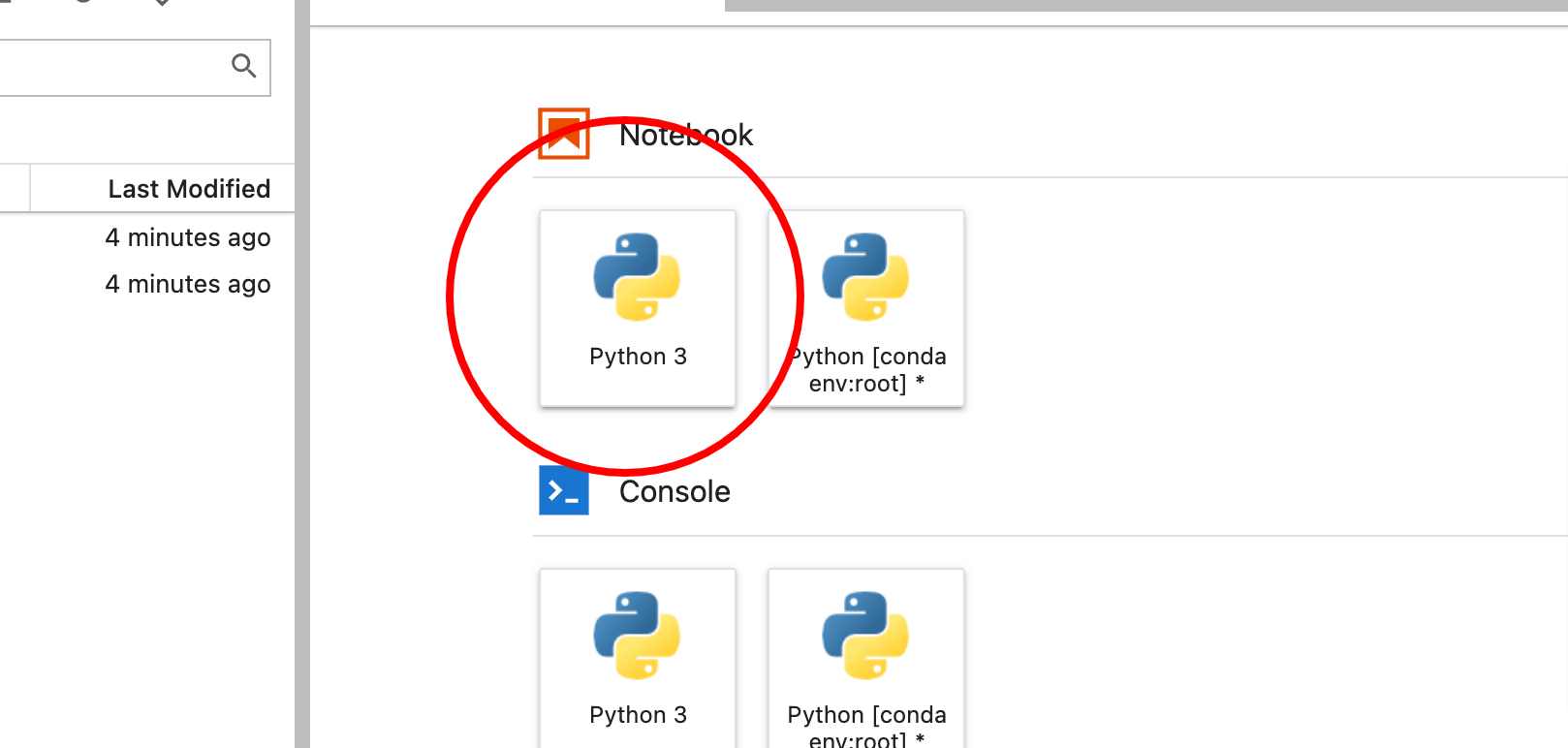


4. Creating a Vertex AI Workbench: From google console Vertex AI select workbench and create a new notebook within the user-managed notebooks section.



Letting all the defaults selected as it is click Create to create a new notebook.

5. Vertex Pipeline Setup: From the new instance click OPEN JUPYTERLAB. From the launcher create new python3 notebook.

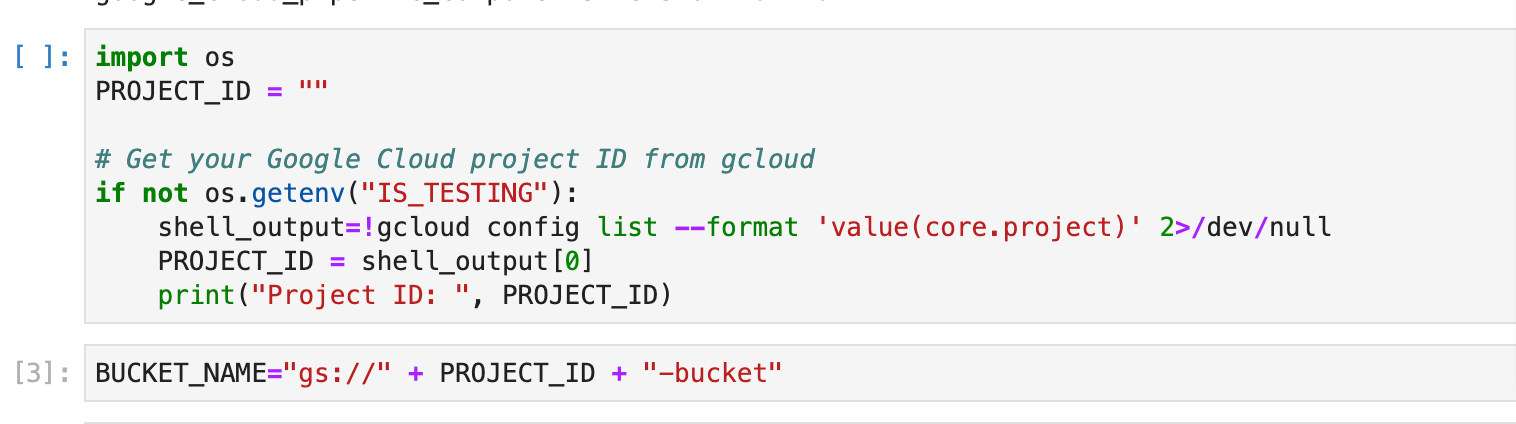


Here Kubeflow Pipelines is the SDK that will be used to build the pipeline. Google Cloud Pipeline Components which provides pre-built components that make it easier to interact with Vertex AI services from your pipeline steps.

Following command install both the libraries:



Next Project details should be set using this:



6. End to end ML Pipeline will be created using this command:

Here the UCI Machine Learning [Dry beans dataset](https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset) is used. This pipeline will:

* Create a [Dataset](https://cloud.google.com/vertex-ai/docs/datasets/datasets) in Vertex AI
* Train a tabular classification model with [AutoML](https://cloud.google.com/vertex-ai/docs/training/automl-api)
* Get evaluation metrics on this model
* Based on the evaluation metrics, decide whether to deploy the model using conditional logic in Vertex Pipelines
* Deploy the model to an endpoint using [Vertex Prediction](https://cloud.google.com/vertex-ai/docs/predictions/deploy-model-api)

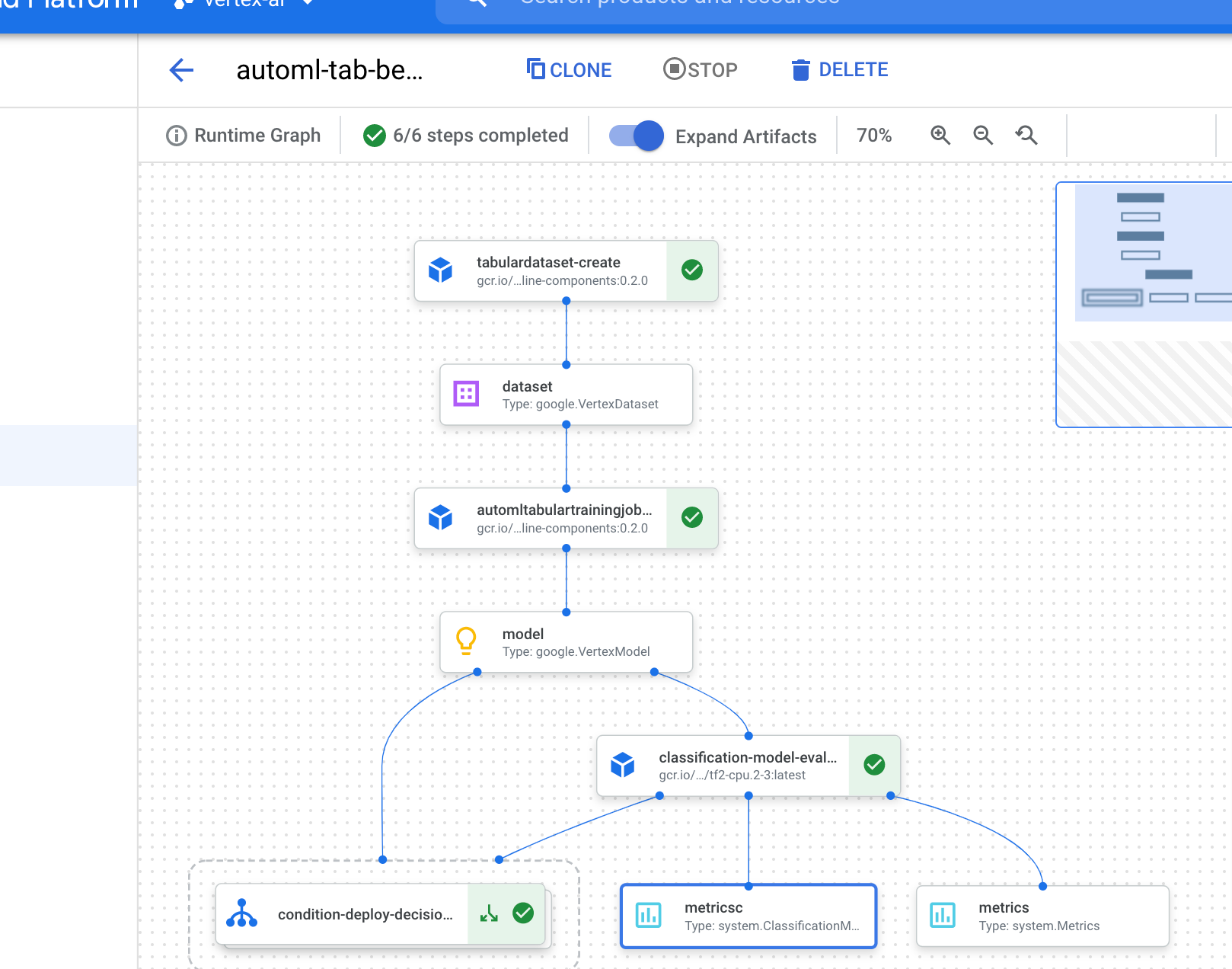


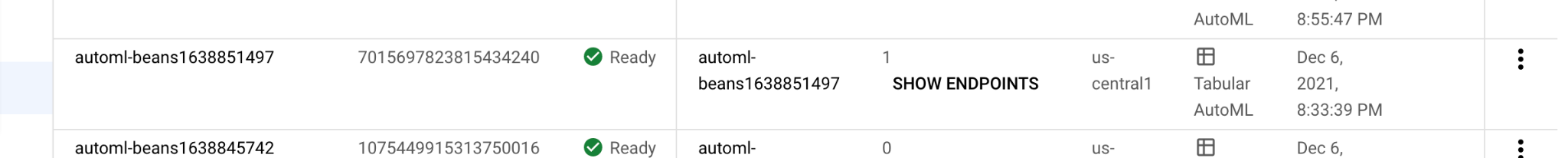
Custom component defined to be used at the end to evaluate and parse them in Vertex UI pipeline can be created using:

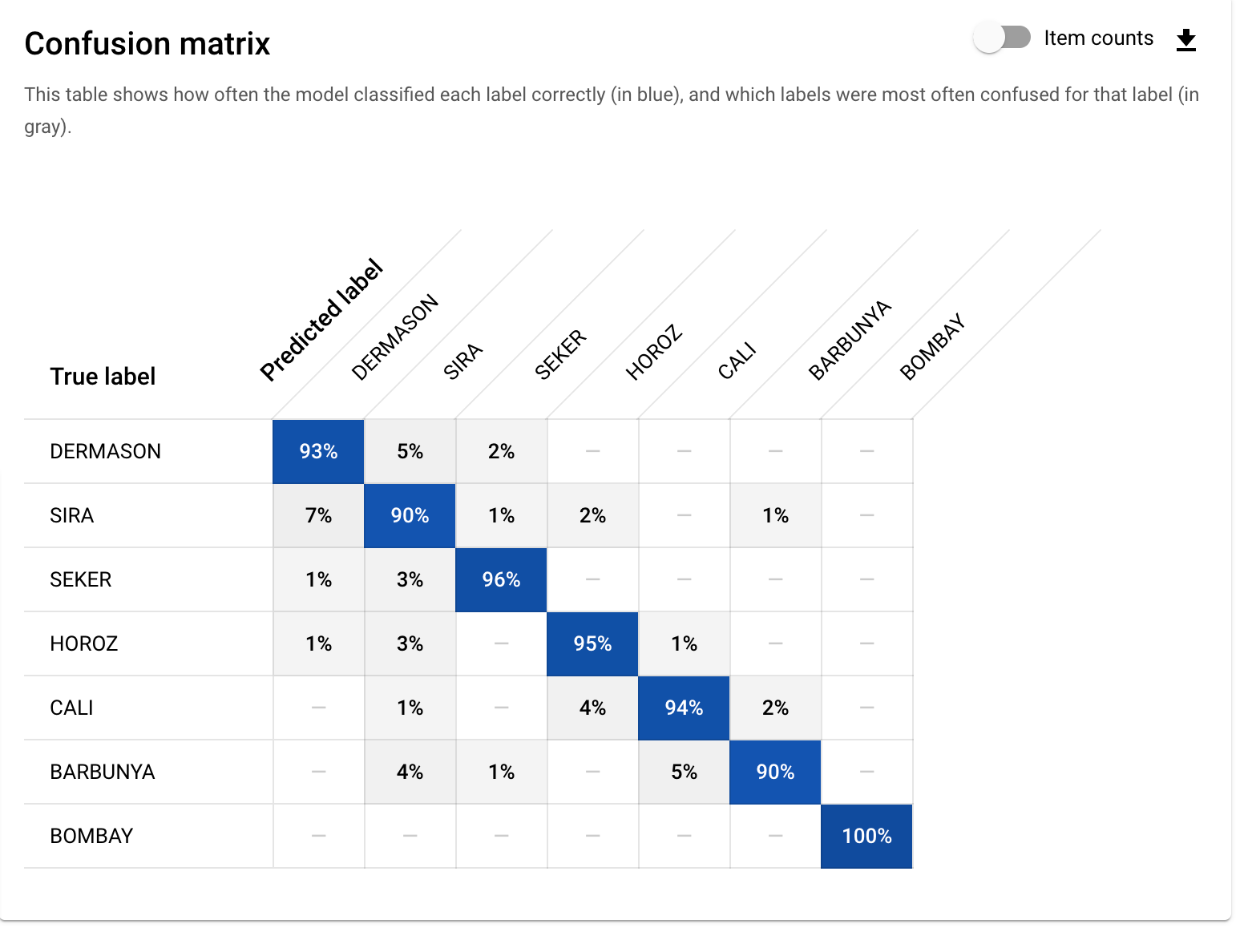


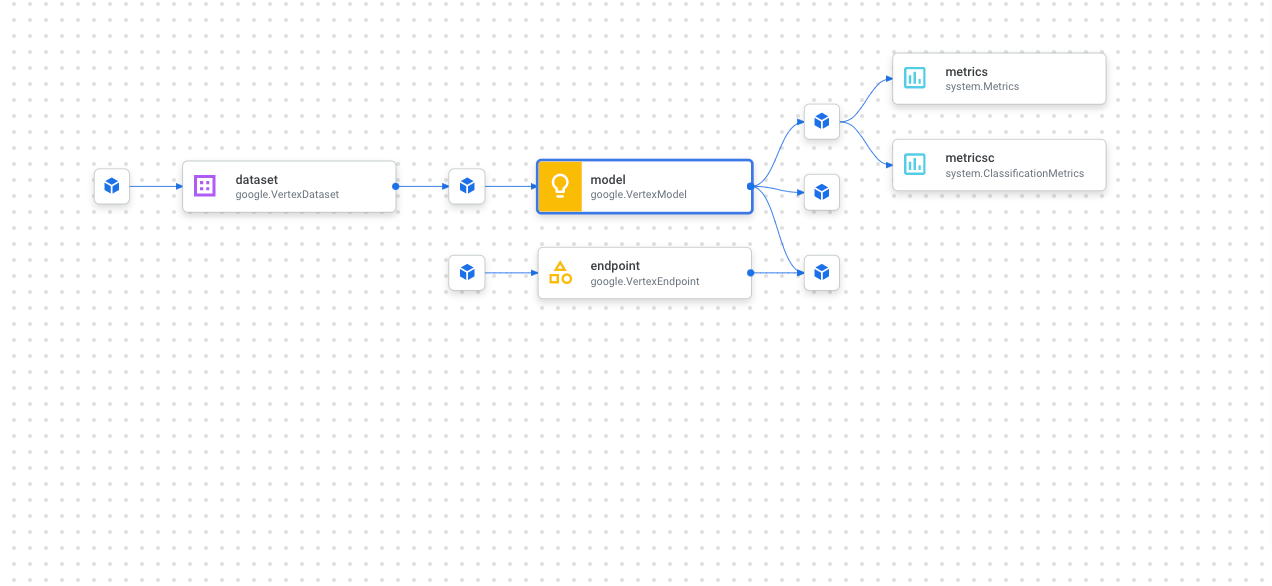
Link to the complete notebook.

Completed pipeline run will look like:









7. Pipeline can be run twice and compared to see how accuracy of training changes:

8. Cleanup:

Deleting notebook created in vertex AI workbench, deleting endpoints created, deleting endpoint and undeploying model from endpoint. This avoids unnecessary charges on the account when the instances are not being used.

