

Machine Learning Operations (MLOps): Getting Started

Course · 1 day

61% complete

[Professional Machine Learning Engineer Certification Learning Path](#) navigate_next [Machine Learning Operations \(MLOps\): Getting Started](#)

navigate_next Vertex AI and MLOps on Vertex AI

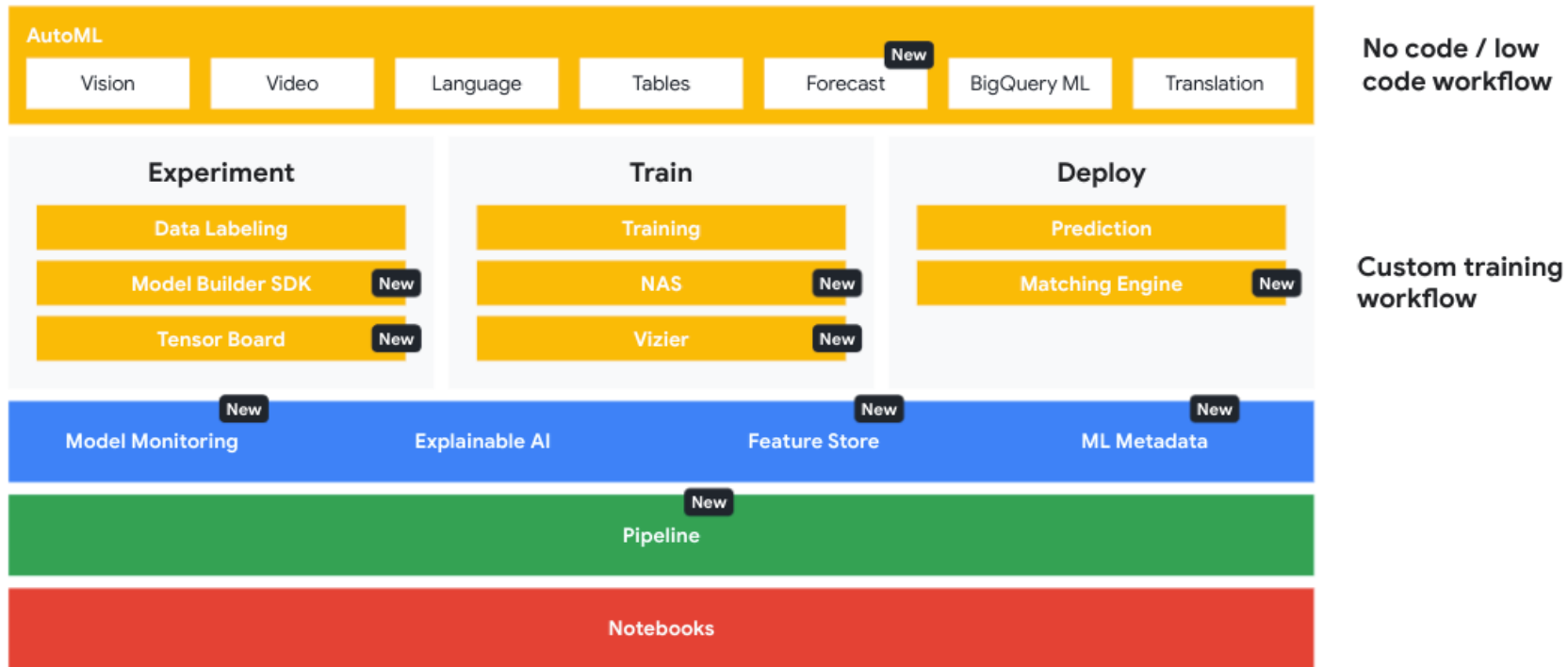
Training and Deploying a TensorFlow Model in Vertex AI

1 hour 30 minutes Free

Overview

In this lab, you will use [BigQuery](#) for data processing and exploratory data analysis and the [Vertex AI](#) platform to train and deploy a custom TensorFlow Regressor model to predict customer lifetime value. The goal of the lab is to introduce to Vertex AI through a high value real world use case - predictive CLV. You will start with a local BigQuery and TensorFlow workflow that you may already be familiar with and progress toward training and deploying your model in the cloud with Vertex AI.

Vertex AI ^{GA}



Vertex AI is Google Cloud's next generation, unified platform for machine learning development and the successor to AI Platform announced at Google I/O in May 2021. By developing machine learning solutions on Vertex AI, you can leverage the latest ML pre-built components and AutoML to significantly enhance development productivity, the ability to scale your workflow and decision making with your data, and accelerate time to value.

Learning objectives

- Train a TensorFlow model locally in a hosted [Vertex AI Workbench](#).
- Create a [managed Tabular dataset](#) artifact for experiment tracking.
- Containerize your training code with [Cloud Build](#) and push it to [Google Cloud Artifact Registry](#).
- Run a [Vertex AI custom training job](#) with your custom model container.
- Use [Vertex TensorBoard](#) to visualize model performance.
- Deploy your trained model to a [Vertex Online Prediction Endpoint](#) for serving predictions.
- Request an online prediction and explanation and see the response.

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](#).

Task 1. Enable Google Cloud services

- In Cloud Shell, use `gcloud` to enable the services used in the lab:

```
gcloud services enable \ compute.googleapis.com \ iam.googleapis.com \ iamcredentials.googleapis.com \ monitoring.googleapis.com \  
logging.googleapis.com \ notebooks.googleapis.com \ aiplatform.googleapis.com \ bigquery.googleapis.com \ artifactregistry.googleapis.com \  
cloudbuild.googleapis.com \ container.googleapis.com
```

Task 2. Create Vertex AI custom service account for Vertex Tensorboard integration

1. Create custom service account:

```
SERVICE_ACCOUNT_ID=vertex-custom-training-sa gcloud iam service-accounts create $SERVICE_ACCOUNT_ID \ --description="A custom  
service account for Vertex custom training with Tensorboard" \ --display-name="Vertex AI Custom Training"
```

2. Grant it access to Cloud Storage for writing and retrieving Tensorboard logs:

```
PROJECT_ID=$(gcloud config get-value core/project) gcloud projects add-iam-policy-binding $PROJECT_ID \ --  
member=serviceAccount:$SERVICE_ACCOUNT_ID@$PROJECT_ID.iam.gserviceaccount.com \ --role="roles/storage.admin"
```

3. Grant it access to your BigQuery data source to read data into your TensorFlow model:

```
gcloud projects add-iam-policy-binding $PROJECT_ID \ --  
member=serviceAccount:$SERVICE_ACCOUNT_ID@$PROJECT_ID.iam.gserviceaccount.com \ --role="roles/bigquery.admin"
```

4. Grant it access to Vertex AI for running model training, deployment, and explanation jobs:

```
gcloud projects add-iam-policy-binding $PROJECT_ID \ --  
member=serviceAccount:$SERVICE_ACCOUNT_ID@$PROJECT_ID.iam.gserviceaccount.com \ --role="roles/aiplatform.user"
```

Task 3. Launch Vertex AI Workbench notebook

1. In the Google Cloud Console, on the **Navigation Menu**, click **Vertex AI > Workbench**. Select **User-Managed Notebooks**.
2. On the Notebook instances page, click **Create Notebook**.
3. In the **Create instance** dialog, confirm the name of the deep learning VM, if you don't want to change the region and zone, leave it to default.
4. Click **Environment**, and select **TensorFlow Enterprise 2.6 (with LTS and Intel MKL-DNN/MKL)** from the dropdown.
5. Click **Machine type**, and select **E2 standard > e2-standard-2**.
6. Leave all settings as they are and then click **Create**. The new VM will take 2-3 minutes to start.
7. Click **Open JupyterLab**.
A JupyterLab window will open in a new tab.

Click *Check my progress* to verify the objective. Create a Vertex AI Notebook

Task 4. Clone the lab repository

Next you'll clone the `training-data-analyst` repo to your JupyterLab instance.

To clone the training-data-analyst notebook in your JupyterLab instance:

1. In JupyterLab, to open a new terminal, click the **Terminal** icon.
2. At the command-line prompt, run the following command:

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```

3. To confirm that you have cloned the repository, double-click on the training-data-analyst directory and ensure that you can see its contents.
The files for all the Jupyter notebook-based labs throughout this course are available in this directory.

It will take several minutes for the repo to clone.

Click *Check my progress* to verify the objective. Clone the lab repository

Task 5. Install lab dependencies and Run the Notebook

- Run the following to go to the `training-data-analyst/self-paced-labs/vertex-ai/vertex-ai-qwikstart` folder, then `pip3 install requirements.txt` to install lab dependencies:

`cd training-data-analyst/self-paced-labs/vertex-ai/vertex-ai-qwikstart` `pip3 install --user -r requirements.txt` **Note:** Ignore the incompatibility warnings and errors.

Navigate to lab notebook

1. In your notebook, navigate to **training-data-analyst > self-paced-labs > vertex-ai > vertex-ai-qwikstart**, and open **lab_exercise_long.ipynb**.
2. Continue the lab in the notebook, and run each cell by clicking the **Run** icon at the top of the screen.

Alternatively, you can execute the code in a cell with **SHIFT + ENTER**.

Read the narrative and make sure you understand what's happening in each cell.

Congratulations!

In this lab, you ran a machine learning experimentation workflow using Google Cloud BigQuery for data storage and analysis and Vertex AI machine learning services to train and deploy a TensorFlow model to predict customer lifetime value. You progressed from training a TensorFlow model locally to training on the cloud with Vertex AI and leveraged several new unified platform capabilities such as Vertex TensorBoard and prediction feature attributions.

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.

- [Overview](#)
- [Setup](#)
- [Task 1. Enable Google Cloud services](#)
- [Task 2. Create Vertex AI custom service account for Vertex Tensorboard integration](#)
- [Task 3. Launch Vertex AI Workbench notebook](#)
- [Task 4. Clone the lab repository](#)
- [Task 5. Install lab dependencies and Run the Notebook](#)
- [Congratulations!](#)
- [End your lab](#)