Overview: This quickstart provides a brief introduction to working with Cloud TPU. In this quickstart, you use Cloud TPU to run MNIST (https://developers.google.com/machine-learning/glossary/#MNIST), a canonical dataset of hand-written digits that is often used to test new machine learning approaches.

This topic is intended for users new to Cloud TPU. For a more detailed exploration of Cloud TPU, try running one of our colabs (/tpu/docs/colabs). You can also view one of the many examples in the Tutorials (/tpu/docs/tutorials/) section.

Before starting this tutorial, check that your Google Cloud project is correctly set up.

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Cloud TPU
- Cloud Storage

Use the pricing calculator (/products/calculator/) to generate a cost estimate based on your projected usage. New Google Cloud users might be eligible for a free trial (/free/).

This section provides information on setting up Cloud Storage storage and a Compute Engine VM.

Important: Set up your Compute Engine VM, your Cloud TPU node and your Cloud Storage bucket in the same region/zone to reduce network latency and network costs.

1. Open a Cloud Shell window.

   Open Cloud Shell (https://console.cloud.google.com/?cloudshell=true)

2. Create a variable for your project's name.

3. Configure `gcloud` command-line tool to use the project where you want to create Cloud TPU.

4. Create a Cloud Storage bucket using the following command:

★ **Note:** In the following command, replace ***bucket-name*** with the name you want to assign to your bucket.

This Cloud Storage bucket stores the data you use to train your model and the training results.

5. Launch a Compute Engine VM and Cloud TPU using the `ctpu up` command.

★ **Note:** If you have more than one project, you must specify the project name with the `--project` flag. If the Compute Engine VM name is not specified with the `--name` flag, it defaults to your username.

6. The configuration you specified appears. Enter y to approve or n to cancel.

7. When the `ctpu up` command has finished executing, verify that your shell prompt has changed from `username@project` to `username@tpuname`. This change shows that you are now logged into your Compute Engine VM.

★ **Note:** If you are not connected to the Compute Engine instance, you can connect by running the following commands, replacing ***vm-name*** with your VM name and ***tpu-name*** with your TPU name. The VM and TPU names default to your username:

As you continue these instructions, run each command that begins with `(vm)$` in your VM session window.

The MNIST TPU model is pre-installed on your Compute Engine VM. Change to the directory that stores the model:

The source code for the MNIST TPU model is available on GitHub (https://github.com/tensorflow/models/blob/master/official/vision/image_classification/mnist_main.py). You can run the model on a Cloud TPU. Alternatively, see how to run the model on a local machine (#run-local).

Export the following variables. Replace ***bucket-name*** with your bucket name:

1. Run the MNIST model:

Parameter	Description
`tpu`	The name of the Cloud TPU. If not specified when setting up the Compute Engine VM and Cloud TPU, defaults to your username.
`model_dir`	This is the directory that contains the model files. This tutorial uses a folder within the Cloud Storage bucket. You do not have to create this folder beforehand. The script creates the folder if it does not already exist.
`data_dir`	This is the directory that contains files used for training.
`download`	When set to true, the **`mnist_main.py`** script downloads and preprocesses the MNIST dataset, if it hasn't been downloaded already.

The model runs in under 5 minutes on a v3-8 Cloud TPU and displays output similar to:

To avoid incurring charges to your Google Cloud account for the resources used in this quickstart, follow these steps.

1. Disconnect from the Compute Engine instance, if you have not already done so:

   Your prompt should now be `user@projectname`, showing you are in the Cloud Shell.

2. In your Cloud Shell, run `ctpu delete` with the `--zone` flag you used when you set up the Compute Engine VM and Cloud TPU. This deletes both your VM and your Cloud TPU.

★ **Important:** If you set the TPU resources name when you ran `ctpu up`, you must specify that name with the `--name` flag when you run `ctpu delete` in order to shut down your TPU resources.

3. Run `ctpu status` to make sure you have no instances allocated to avoid unnecessary charges for TPU usage. The deletion might take several minutes. A response like the one below indicates there are no more allocated instances:

4. Run `gsutil` as shown, replacing `bucket-name` with the name of the Cloud Storage bucket you created for this tutorial:

Note: For free storage limits and other pricing information, see the Cloud Storage pricing guide (/storage/pricing).

This quickstart provided you with a brief introduction to working with Cloud TPU. At this point, you have the foundation for the following:

- Learning more about Cloud TPU
- Setting up Cloud TPU for your own applications

MNIST on Keras (https://colab.sandbox.google.com/github/tensorflow/tpu/blob/master/tools/colab/keras_mnist_tpu.ipynb

Product Overview (/tpu/index)

Cloud Tensor Processing Units (TPUs) (/tpu/docs/tpus)

Pricing (/tpu/docs/tpus)

Choosing a TPU service (/tpu/docs/deciding-tpu-service)	Understand different options for working with Cloud TPU, such as Compute Engine (/compute/docs), Google Kubernetes Engine (/kubernetes-engine/docs/), or AI Platform (/ml-engine/docs).
TPU types and zones (/tpu/docs/types-zones)	Learn what TPU types are available in each zone.
TPU versions (/tpu/docs/system-architecture#versions)	Understand the different TPU versions and learn how to select the right one for your application.