

This tutorial shows you how to train the Bidirectional Encoder Representations from Transformers (BERT) model on Cloud TPU.

BERT is a method of pre-training language representations. Pre-training refers to how BERT is first trained on a large source of text, such as Wikipedia. You can then apply the training results to other Natural Language Processing (NLP) tasks, such as [question answering](https://en.wikipedia.org/wiki/Question_answering) (https://en.wikipedia.org/wiki/Question_answering) and [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis) (https://en.wikipedia.org/wiki/Sentiment_analysis). With BERT and Cloud TPU, you can train a variety of NLP models in about 30 minutes.

For more information about BERT, see the following resources:

- [Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing](https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html) (<https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>)
 - [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](https://arxiv.org/abs/1810.04805) (<https://arxiv.org/abs/1810.04805>)
-
- Create a Cloud Storage bucket to hold your dataset and model output.
 - Clone the BERT repository and other required files.
 - Run the training job.
 - Verify the output results.

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Cloud TPU

- Cloud Storage

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

1. [Sign in \(https://accounts.google.com/Login\)](https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account \(https://accounts.google.com/SignUp\)](https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[Go to the project selector page \(https://console.cloud.google.com/projectselector2/home/dashboard\)](https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project \(/billing/docs/how-to/modify-project\)](/billing/docs/how-to/modify-project).

This walkthrough uses billable components of Google Cloud. Check the [Cloud TPU pricing page \(/tpu/docs/pricing\)](/tpu/docs/pricing) to estimate your costs. Be sure to [clean up \(#clean_up\)](#) resources you create when you've finished with them to avoid unnecessary charges.

This section provides information on setting up Cloud Storage storage, VM, and Cloud TPU resources for tutorials.

Important: Set up all resources in the same region/zone to reduce network latency and network costs.

You need a Cloud Storage bucket to store the data you use to train your model and the training results. The `ctpu up` tool used in this tutorial sets up default permissions for the Cloud TPU service account. If you want finer-grain permissions, review the [access level permissions](/tpu/docs/storage-buckets) (/tpu/docs/storage-buckets).

The bucket location must be in the same region as your virtual machine (VM) and your TPU node. VMs and TPU nodes are located in [specific zones](/tpu/docs/types-zones#types) (/tpu/docs/types-zones#types), which are subdivisions within a region.

1. Go to the Cloud Storage page on the Cloud Console.

[Go to the Cloud Storage page](https://console.cloud.google.com/storage/browser) (https://console.cloud.google.com/storage/browser)

2. Create a new bucket, specifying the following options:

- A unique name of your choosing.
- Select `Region` for Location type and `us-centra1` for the Location (zone)
- Default storage class: `Standard`
- Location: Specify a bucket location in the same region where you plan to create your TPU node. See [TPU types and zones](/tpu/docs/types-zones#types) (/tpu/docs/types-zones#types) to learn where various TPU types are available.

This section demonstrates using the [Cloud TPU provisioning tool](https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (`ctpu`) for creating and managing Cloud TPU project resources. The resources are comprised of a virtual machine (VM) and a Cloud TPU resource that have the same name. **These resources must reside in the same region/zone as the bucket you just created.**

You can also set up your VM and TPU resources using `gcloud` commands or through the [Cloud Console](https://console.cloud.google.com/) (https://console.cloud.google.com/). See the [creating and deleting TPUs](/tpu/docs/creating-deleting-tpus) (/tpu/docs/creating-deleting-tpus) page to learn all the ways you can set up and manage your Compute Engine VM and Cloud TPU resources.

1. Open a Cloud Shell window.

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

2. Run `gcloud config set project <var>your-project</var>` to set the project where you want to create Cloud TPU.
3. Run `ctpu up` specifying the flags shown for either a Cloud TPU device or Pod slice. If you do not specify `tpu-size`, the default is a v2-8 Cloud TPU. Refer to [CTPU Reference](/tpu/docs/ctpu-reference) (/tpu/docs/ctpu-reference) for flag options and descriptions.
4. Set up a Cloud TPU device:

★ **Note:** If you have more than one project, you must specify the project name. If `--name` is not specified, it defaults to your username. If `--zone` is not specified, it defaults to `us-central1-b`. Make sure the zone matches the zone where you set up the storage bucket.

5. The configuration you specified appears. Enter `y` to approve or `n` to cancel.
6. 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 the name of your VM:

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

From your Compute Engine virtual machine (VM), clone the BERT repository.

This tutorial uses the General Language Understanding Evaluation (GLUE) benchmark to evaluate and analyze the performance of the model. To use this benchmark, download the `download_glue_data.py` script using the following `git clone` command:

Next, run the `download_glue_data.py` on your Compute Engine VM.

Next, define several parameter values that are required when you train and evaluate your model:

- `STORAGE_BUCKET`. This is the name of the Cloud Storage bucket that you created in the **Before you begin** section.
- `BERT_BASE_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.

- `GLUE_DIR`. This is the directory that contains the GLUE data. For this tutorial, the directory is `~/glue_data`.
- `TASK_NAME`. This is the type of training you want to perform. For this tutorial, use the value `MRPC`, which specifies the Microsoft Research Paraphrase Corpus.
- `TPU_NAME`. The name of the TPU. The `ctpu up` command automatically creates this variable for you.

From your Compute Engine VM, run the following command.

When the training script completes, you should see results similar to the following:

To avoid incurring charges to your GCP account for the resources used in this topic:

1. Disconnect from the Compute Engine VM:

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 Cloud TPU to delete your Compute Engine 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).