

[AI & Machine Learning Products](https://cloud.google.com/products/machine-learning/) (<https://cloud.google.com/products/machine-learning/>)

[Cloud TPU](https://cloud.google.com/tpu/) (<https://cloud.google.com/tpu/>)

[Documentation](https://cloud.google.com/tpu/docs/) (<https://cloud.google.com/tpu/docs/>) [Guides](#)

# Cloud Bigtable for Streaming Data

## Alpha

This feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages](https://cloud.google.com/products/#product-launch-stages) (<https://cloud.google.com/products/#product-launch-stages>).

[Cloud Bigtable](https://cloud.google.com/bigtable/docs) (<https://cloud.google.com/bigtable/docs>) is a low-latency, scalable, wide-row structured storage system that can store and serve training data for your machine learning model. Using Cloud Bigtable, you can stream your training data at ultra-high speed for an efficient use of Cloud TPU.

This tutorial shows you how to train the [TensorFlow ResNet-50 model](https://github.com/tensorflow/tpu/tree/master/models/official/resnet) (<https://github.com/tensorflow/tpu/tree/master/models/official/resnet>) on Cloud TPU using Cloud Bigtable to host your training data. The process uses [TensorFlow's integration with Cloud Bigtable](https://github.com/tensorflow/io/tree/master/tensorflow_io/bigtable) ([https://github.com/tensorflow/io/tree/master/tensorflow\\_io/bigtable](https://github.com/tensorflow/io/tree/master/tensorflow_io/bigtable)).

## Disclaimer

This tutorial uses a third-party dataset. Google provides no representation, warranty, or other guarantees about the validity, or any other aspects of, this dataset.

## Requirements and limitations

Note the following when defining your configuration:

- You must use TensorFlow 1.11 or above for Cloud Bigtable support.
- Cloud Bigtable is recommended for high performance (pod-scale) training jobs on massive amounts of data, processing hundreds of gigabytes (GB) to hundreds of

terabytes (TB) at tens-to-hundreds of gigabits per second (Gbps). We recommend that you use Cloud Storage for workloads that do not fit this description. Note: We also recommend Cloud Bigtable for use in reinforcement learning (RL) workloads where training data is generated on the fly.

## About the model and the data

The model in this tutorial is based on [Deep Residual Learning for Image Recognition](https://arxiv.org/pdf/1512.03385.pdf) (https://arxiv.org/pdf/1512.03385.pdf), which introduced the residual network (ResNet) architecture. This tutorial uses the 50-layer variant known as ResNet-50.

The training dataset is [ImageNet](http://www.image-net.org/challenges/LSVRC/2012/) (http://www.image-net.org/challenges/LSVRC/2012/), which is a popular choice for training image recognition systems.

The tutorial uses [TPUEstimator](https://www.tensorflow.org/api_docs/python/tf/contrib/tpu/TPUEstimator)

(https://www.tensorflow.org/api\_docs/python/tf/contrib/tpu/TPUEstimator) to train the model.

TPUEstimator is based on [tf.estimator](https://www.tensorflow.org/get_started/estimator) (https://www.tensorflow.org/get\_started/estimator), a high-level TensorFlow API, and is the recommended way to build and run a machine learning model on Cloud TPU. The API simplifies the model development process by hiding most of the low-level implementation, making it easier to switch between Cloud TPU and other platforms such as GPU or CPU.

## Before you begin

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

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 \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECT\)](https://console.cloud.google.com/projectselect)

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

This walkthrough uses billable components of Google Cloud. Check the [Cloud TPU pricing page](https://cloud.google.com/tpu/docs/pricing) (https://cloud.google.com/tpu/docs/pricing) to estimate your costs, and follow the instructions to clean up resources when you've finished with them.

4. Install the Cloud SDK including the `gcloud` command-line tool.

[INSTALL THE CLOUD SDK \(HTTPS://CLOUD.GOOGLE.COM/SDK/DOCS/\)](https://cloud.google.com/sdk/docs/)

In addition, this tutorial requires that you already have data stored in Cloud Bigtable. For more information on how to use Cloud Bigtable, see the [Cloud Bigtable documentation](https://cloud.google.com/bigtable/docs/) (https://cloud.google.com/bigtable/docs/).

## Create a Cloud Bigtable instance

Create a Cloud Bigtable instance for streaming your training data:

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

[GO TO THE CLOUD BIGTABLE PAGE \(HTTPS://CONSOLE.CLOUD.GOOGLE.COM/BIGTABLE\)](https://console.cloud.google.com/bigtable)

2. Create an instance, specifying the following options:
  - **Instance name:** A name to help you identify the instance.
  - **Instance ID:** A permanent identifier for the instance.
  - **Instance type:** Select **Production** for best performance.
  - **Storage type:** Select **HDD**.
  - **Cluster ID:** A permanent identifier for the cluster.
  - **Region:** Select a region. For example, `us-central1`. As a best practice, select the same region where you plan to create your TPU node. See the [TPU types and zones](https://cloud.google.com/tpu/docs/types-zones#types) (https://cloud.google.com/tpu/docs/types-zones#types) page to learn where various TPU types are available.

- **Zone:** Select a zone. For example, `us-central1-b`. As a best practice, select the same zone where you plan to create your TPU node. See the [TPU types and zones](https://cloud.google.com/tpu/docs/types-zones#types) (<https://cloud.google.com/tpu/docs/types-zones#types>) page to learn where various TPU types are available.

You can leave the other values at their default settings. For details, see the guide to [creating a Cloud Bigtable instance](https://cloud.google.com/bigtable/docs/creating-instance) (<https://cloud.google.com/bigtable/docs/creating-instance>).

## Download and set up the `ctpu` tool

This guide uses the [Cloud TPU Provisioning Utility](https://github.com/tensorflow/tpu/tree/master/tools/ctpu)

(<https://github.com/tensorflow/tpu/tree/master/tools/ctpu>) (`ctpu`) as a simple tool for setting up and managing your Cloud TPU. The guide also assumes that you want to run `ctpu` and the Cloud SDK locally rather than using Cloud Shell. The Cloud Shell environment is not suitable for lengthy procedures like the downloading of the ImageNet data, as Cloud Shell times out after a period of inactivity.

1. Follow the [ctpu guide](https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (<https://github.com/tensorflow/tpu/tree/master/tools/ctpu>) to download, install, and configure `ctpu`.
2. Set the project, zone, and region for the `gcloud` tool's environment, replacing `YOUR-PROJECT-ID` with your [Google Cloud project ID](https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) ([https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying\\_projects](https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects))

:

```
$ gcloud config set project YOUR-PROJECT-ID
$ gcloud config set compute/region us-central1
$ gcloud config set compute/zone us-central1-b
```

3. Type the following into your Cloud Shell, to check your `ctpu` configuration:

```
$ ctpu print-config
```

You should see a message like this:

```
2018/04/29 05:23:03 WARNING: Setting zone to "us-central1-b"
ctpu configuration:
  name: [your TPU's name]
```

```
project: [your-project-id]
zone: us-central1-b
```

If you would like to change the configuration for a single command invocation, If you would like to change the configuration generally, see `gcloud config con`

In the output message, the *name* is the name of your Cloud TPU (defaults to your username) and *zone* is the default geographic zone (defaults to `us-central1-b`) for your TPU node. See [TPU types and zones](https://cloud.google.com/tpu/docs/types-zones#types) (<https://cloud.google.com/tpu/docs/types-zones#types>) page to learn where various TPU types are available. You can change the name, zone, and other properties when you run `ctpu up` as described [below](#) (`#create_resources`).

## Set access permissions for Cloud TPU

Set the following access permissions for the Cloud TPU service account:

1. Give the Cloud TPU access to Cloud Bigtable for your Google Cloud project. (Note that `ctpu` applies some access permissions by default, but not this one.)

```
$ ctpu auth add-bigtable
```

You can check the Cloud TPU permissions by running this command:

```
$ ctpu auth list
```

2. (Optional) Update the Cloud Storage permissions. The `ctpu up` command sets default permissions for Cloud Storage. If you want finer-grained permissions, review and update the [access level permissions](#) ([https://cloud.google.com/tpu/docs/storage-buckets#storage\\_access](https://cloud.google.com/tpu/docs/storage-buckets#storage_access)).

## Create a Compute Engine VM and a Cloud TPU

Run the following command to set up a Compute Engine virtual machine (VM) and a Cloud TPU with associated services. This combination of resources and services is called a *Cloud TPU flock*:

```
$ ctpu up [optional: --name --zone]
```

You should see a message that follows this pattern:

```
ctpu will use the following configuration:
```

```
Name: [your TPU's name]
Zone: [your project's zone]
GCP Project: [your project's name]
TensorFlow Version: 1.14
VM:
  Machine Type: [your machine type]
  Disk Size: [your disk size]
  Preemptible: [true or false]
Cloud TPU:
  Size: [your TPU size]
  Preemptible: [true or false]
```

```
OK to create your Cloud TPU resources with the above configuration? [Yn]:
```

Enter `y` and press Enter to create your Cloud TPU resources.

The `ctpu up` command performs the following tasks:

- Enables the Compute Engine and Cloud TPU services.
- Creates a Compute Engine VM instance with the latest stable TensorFlow version pre-installed.
- Creates a Cloud TPU with the corresponding version of TensorFlow, and passes the name of the Cloud TPU to the Compute Engine VM instance as an environment variable (`TPU_NAME`).
- Ensures your Cloud TPU has access to resources it needs from your Google Cloud project, by granting specific IAM roles to your Cloud TPU service account.
- Performs a number of other checks.
- Logs you in to your new Compute Engine VM instance.

**Note:** The first time you run `ctpu up` on a project, it takes longer than usual because it needs to perform additional tasks including SSH key propagation and API turn-up.

You can run `ctpu up` as often as you like. For example, if you lose the SSH connection to the Compute Engine VM instance, run `ctpu up` to restore the connection. (If you changed the

default values for `--name` and `--zone`, you must specify them again each time you run `ctpu up`.) See the [ctpu documentation](https://github.com/tensorflow/tpu/tree/master/tools/ctpu) (https://github.com/tensorflow/tpu/tree/master/tools/ctpu) for details.

## Verify that you're logged in to your VM instance

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

## Set up some handy variables

In the following steps, a prefix of `(vm)$` means you should run the command on the Compute Engine VM instance in your Cloud TPU flock.

Set up some environment variables to simplify the commands in this tutorial:

1. Set up a `PROJECT` variable, replacing `YOUR-PROJECT-ID` with your [Google Cloud project ID](https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying_projects) (https://cloud.google.com/resource-manager/docs/creating-managing-projects#identifying\_projects)

:

```
(vm)$ export PROJECT=YOUR-PROJECT-ID
```

2. Set up a `STORAGE_BUCKET` variable, replacing `YOUR-BUCKET-NAME` with the name of your Cloud Storage bucket:

```
(vm)$ export STORAGE_BUCKET=gs://YOUR-BUCKET-NAME
```

3. Set up a `BIGTABLE_INSTANCE` variable, replacing `YOUR-INSTANCE-ID` with the ID of the Cloud Bigtable instance that you created earlier:

```
(vm)$ export BIGTABLE_INSTANCE=YOUR-INSTANCE-ID
```

## Prepare the data

Upload the data to bigtable. The training application expects your training data to be accessible in Cloud Bigtable. For instructions on how to upload your data to bigtable use [tensorflow\\_io/bigtable/README.md](https://github.com/tensorflow/io/blob/master/tensorflow_io/bigtable/README.md) ([https://github.com/tensorflow/io/blob/master/tensorflow\\_io/bigtable/README.md](https://github.com/tensorflow/io/blob/master/tensorflow_io/bigtable/README.md)).

## (Optional) Set up TensorBoard

TensorBoard offers a [suite of tools](https://cloud.google.com/tpu/docs/cloud-tpu-tools) (<https://cloud.google.com/tpu/docs/cloud-tpu-tools>) designed to present TensorFlow data visually. When used for monitoring, TensorBoard can help identify bottlenecks in processing and suggest ways to improve performance.

If you don't need to monitor the model's output at this time, you can skip the TensorBoard setup steps.

If you want to monitor the model's output and performance, follow the guide to [setting up TensorBoard](https://cloud.google.com/tpu/docs/tensorboard-setup) (<https://cloud.google.com/tpu/docs/tensorboard-setup>).

## Run the ResNet-50 model

If the data is in the bigtable database then you are now ready to train and evaluate the ResNet-50 model on your Cloud TPU, streaming the training data from Cloud Bigtable. The training application writes out the trained model and intermediate checkpoints to Cloud Storage.

Run the following commands on your Compute Engine VM instance:

1. Add the top-level `/models` folder to the Python path:

```
(vm)$ export PYTHONPATH="$PYTHONPATH:/usr/share/tpu/models"
```

2. Go to the directory on your Compute Engine VM instance where the ResNet-50 model is pre-installed:

```
(vm)$ cd /usr/share/tpu/models/official/resnet/
```

3. Run the training script:



```
(vm)$ python resnet_main.py \  
  --tpu=$TPU_NAME \  
  --model_dir=${STORAGE_BUCKET}/resnet \  
  --bigtable_project=$PROJECT \  
  --bigtable_instance=$BIGTABLE_INSTANCE \  
  --bigtable_table=$TABLE_NAME \  
  --bigtable_column_family=$FAMILY_NAME \  
  --bigtable_column_qualifier=$COLUMN_QUALIFIER \  
  --bigtable_train_prefix=$ROW_PREFIX_TRAIN \  
  --bigtable_eval_prefix=$ROW_PREFIX_EVAL
```

- `--tpu` specifies the name of the Cloud TPU. Note that `ctpu` passes this name to the Compute Engine VM instance as an environment variable (`TPU_NAME`).
- `--model_dir` specifies the directory where checkpoints and summaries are stored during model training. If the folder is missing, the program creates one. When using a Cloud TPU, the `model_dir` must be a Cloud Storage path (`gs://...`). You can reuse an existing directory to load current checkpoint data and to store additional checkpoints as long as the previous checkpoints were created using TPU of the same size and Tensorflow version.
- `--bigtable_project` specifies the identifier of the Google Cloud project for the Cloud Bigtable instance that holds your training data. If you do not supply this value, the program assumes that your Cloud TPU and your Cloud Bigtable instance are in the same Google Cloud project.
- `--bigtable_instance` specifies the ID of the Cloud Bigtable instance that holds your training data.
- `--bigtable_table` specifies the name of the Cloud Bigtable table that holds your training data.
- `--bigtable_column_family` specifies the name of the Cloud Bigtable family.
- `--bigtable_column_qualifier` specifies the name of the Cloud Bigtable column qualifier. See the [Cloud Bigtable overview](https://cloud.google.com/bigtable/docs/overview) (<https://cloud.google.com/bigtable/docs/overview>) for a description of the storage model.

## What to expect

The above procedure trains the ResNet-50 model for 90 epochs and evaluates the model every 1,251 steps. For information on the default values that the model uses, and the flags you can use to change the defaults, see the code and README for the [TensorFlow ResNet-50 model](https://github.com/tensorflow/tpu/tree/master/models/official/resnet) (<https://github.com/tensorflow/tpu/tree/master/models/official/resnet>). With the default flags, the model should train to above 76% accuracy.

In our tests, Cloud Bigtable provides very high throughput performance for workloads such as ImageNet training, supporting scan throughput at hundreds of megabytes per second per node.

## Clean up

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

1. Disconnect from the Compute Engine VM:

```
(vm)$ exit
```

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:

```
$ ctpu delete [optional: --zone]
```

★ **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:

```
2018/04/28 16:16:23 WARNING: Setting zone to "us-central1-b"  
No instances currently exist.  
    Compute Engine VM:    --  
    Cloud TPU:           --
```

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

```
$ gsutil rm -r gs://YOUR-BUCKET-NAME
```

**Note:** For free storage limits and other pricing information, see the [Cloud Storage pricing guide](https://cloud.google.com/storage/pricing) (<https://cloud.google.com/storage/pricing>).

1. When you no longer need your training data on Cloud Bigtable, run the following command in your Cloud Shell to delete your Cloud Bigtable instance. (The `BIGTABLE_INSTANCE` variable should represent the Cloud Bigtable instance ID that you used earlier.)

```
$ cbt deleteinstance $BIGTABLE_INSTANCE
```

## What's next

- Read the [RetinaNet-50 tutorial](https://cloud.google.com/tpu/docs/tutorials/retinanet) (<https://cloud.google.com/tpu/docs/tutorials/retinanet>) to understand the modifications made to the model to run on Cloud TPU.
- Explore the [TPU tools in TensorBoard](https://cloud.google.com/tpu/docs/cloud-tpu-tools) (<https://cloud.google.com/tpu/docs/cloud-tpu-tools>).

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated December 4, 2019.*