

[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](#)

# Running Deeplab-v3 on Cloud TPU

This tutorial shows you how to train the Deeplab-v3 model on Cloud TPU.

This model is an image semantic segmentation model. Image semantic segmentation models focus on identifying and localizing multiple objects in a single image. This type of model is frequently used in machine learning applications such as autonomous driving, geospatial image processing, and medical imaging.

In this tutorial, you'll run a training model against the PASCAL VOC 2012 dataset. For more information on this data set, see [The PASCAL Visual Object Classes Homepage](http://host.robots.ox.ac.uk/pascal/VOC/) (<http://host.robots.ox.ac.uk/pascal/VOC/>).

## 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/projectselector) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECTOR](https://console.cloud.google.com/projectselector))

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. Be sure to

clean up (#clean\_up) resources you create when you've finished with them to avoid unnecessary charges.

## Set up your resources

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.

### Create a Cloud Storage bucket

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](https://cloud.google.com/tpu/docs/storage-buckets) (<https://cloud.google.com/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](https://cloud.google.com/tpu/docs/types-zones#types) (<https://cloud.google.com/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.
  - 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](https://cloud.google.com/tpu/docs/types-zones#types) (<https://cloud.google.com/tpu/docs/types-zones#types>) to learn where various TPU types are available.

### Use the `ctpu` tool

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](https://cloud.google.com/tpu/docs/creating-deleting-tpus) (<https://cloud.google.com/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.

## Run `ctpu up` to create 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 <Your-Project>` to use 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. Refer to [CTPU Reference](https://cloud.google.com/tpu/docs/ctpu-reference) (<https://cloud.google.com/tpu/docs/ctpu-reference>) for flag options and descriptions.
4. Set up a Cloud TPU device:

```
$ ctpu up
```

★ **Note:** If you have more than one project, you must specify the project name.

The following configuration message appears:

```
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]:

Press `y` to create your Cloud TPU resources.

**Note:** The first time you run `ctpu up` on a project it takes about 5 minutes to perform startup tasks such as SSH key propagation and API turnup.

The `ctpu up` command creates a virtual machine (VM) and Cloud TPU services.

From this point on, a prefix of `(vm)$` means you should run the command on the Compute Engine VM instance.

## Verify your Compute Engine VM

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.

## Install additional packages

For this model, you need to install the following additional packages on your Compute Engine instance:

- `python-pil`
- `python-numpy`
- `jupyter`
- `matplotlib`
- `PrettyTable`

```
(vm)$ sudo apt-get install python-pil python-numpy && \
pip install --user jupyter && \
```



```
pip install --user matplotlib && \  
pip install --user PrettyTable
```

## Clone the TensorFlow models and tpu repositories

Run the following command to clone the TensorFlow models repository to your Compute Engine instance:

```
(vm)$ git clone https://github.com/tensorflow/models.git && \  
git clone https://github.com/tensorflow/tpu.git
```

## Add tensorflow/models/research/ to PYTHONPATH

Next, add the models in `tensorflow/models/research` to the `PYTHONPATH` variable:

```
(vm)$ cd ~/models/research && \  
export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
```

## Download and convert the PASCAL VOC 2012 dataset

This model uses the PASCAL VOC 2012 dataset for training and evaluation. Run the following script to download the dataset and convert it to TensorFlow's TFRecord format:

```
(vm)$ cd ~/models/research/deeplab/datasets/ && \  
bash download_and_convert_voc2012.sh && \  
cd ~/
```

## Download the pretrained checkpoint

In this step, you download the a modified resnet 101 pretrained checkpoint. To start, download the checkpoint:

```
(vm)$ wget http://download.tensorflow.org/models/resnet_v1_101_2018_05_04.tar.gz
```

Then, extract the contents of the tar file:

```
(vm)$ tar -vxf resnet_v1_101_2018_05_04.tar.gz
```

## Upload data to your Cloud Storage bucket

At this point, you can now upload your data to the Cloud Storage bucket you created earlier:

```
(vm)$ gsutil cp -r ~/models/research/deeplab/datasets/pascal_voc_seg/tfrecord gs://  
gsutil -m cp -r ~/resnet_v1_101 gs://YOUR-BUCKET-NAME
```

## Train the model

You're now ready to train the model. Be sure to change *YOUR-BUCKET-NAME* to the name of your Cloud Storage bucket.

**Note:** This step can take up to 10 hours to complete.

```
(vm)$ python ~/tpu/models/experimental/deeplab/main.py \  
--mode='train' \  
--num_shards=8 \  
--alsologtostderr=true \  
--model_dir=gs://YOUR-BUCKET-NAME \  
--dataset_dir=gs://YOUR-BUCKET-NAME/tfrecord \  
--init_checkpoint=gs://YOUR-BUCKET-NAME/resnet_v1_101/model.ckpt \  
--model_variant=resnet_v1_101_beta \  
--image_pyramid=1. \  
--aspp_with_separable_conv=false \  
--multi_grid=1 \  
--multi_grid=2 \  
--multi_grid=4 \  
--decoder_use_separable_conv=false \  
--train_split='train'
```

## Evaluate the model

When the training completes, you can evaluate the model. To do so, change the `--mode` flag from `train` to `eval`:

```
(vm)$ python ~/tpu/models/experimental/deeplab/main.py \  
--mode='eval' \  
--num_shards=8 \  
--alsologtostderr=true \  
--model_dir=gs://YOUR-BUCKET-NAME \  
--dataset_dir=gs://YOUR-BUCKET-NAME/tfrecord \  
--init_checkpoint=gs://YOUR-BUCKET-NAME/resnet_v1_101/model.ckpt \  
--model_variant=resnet_v1_101_beta \  
--image_pyramid=1. \  
--aspp_with_separable_conv=false \  
--multi_grid=1 \  
--multi_grid=2 \  
--multi_grid=4 \  
--decoder_use_separable_conv=false \  
--train_split='train'
```

## 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>).

## What's next

- Learn more about `ctpu` (<https://github.com/tensorflow/tpu/tree/master/tools/ctpu>), including how to install it on a local machine.
- 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.*