# GCP Get Started Guide

THE UNIVERSITY OF TENNESSEE KNOXVILLE

BIG ORANGE. BIG IDEAS.®

Chengcheng Li

Date: 08/29/2018

# Announcement

- Before we get started, there is one thing we need to clarify. Thanks to GCP. They provide some Always Free Products, which can help us to do a lot of projects. Among these project, there is a preinstalled **virtual machine**. When I say "**local**" or "**locally**", I mean run the code with this free preinstalled virtual machine not on your own computer.
- I'm also new to GCP and there are a lot I need to learn. I hope this guide can help you get started and develop more skills.

# Reference

- https://cloud.google.com/ml-engine/docs/tensorflow/getting-started-training-prediction
- https://cloud.google.com/solutions/running-distributed-tensorflow-on-compute-engine
- Python
- Basic Linux commands
- **Comments**: Python and basic Linux commands are required. I learned most of the tricks from the first two references. **If you can go through these links, I think you don't need to use this guide.**

# Four Ways to Run Codes

- FREE Run codes locally without ml-engine
- FREE Run codes locally with ml-engine
- FREE Run codes on cloud with ml-engine and preinstalled instance
- Run codes on cloud with ml-engine and self-created instance
- **Comments**: I would suggest you to try the first three free methods first. If they don’t work for your projects, try the fourth one. I guess the free methods should work for a few projects. At least, I tested they work for DL project 1.

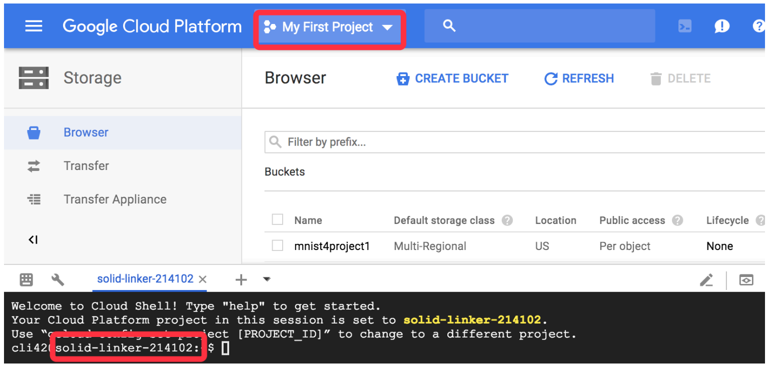
# Prerequisites

- Set up and test your Cloud environment
  - Set up your GCP project
  - Set up your environment
- Prepare your data
  - The codes for Project 1 can automatically download MNIST.
- Prepare your codes
  - Write or download codes
- **Comments**: No matter what kind of method you use, you need to do the above steps. You can learn how to set up environment and bucket with Reference 1 and my last slides.

https://cloud.google.com/ml-engine/docs/tensorflow/getting-started-training-prediction

# Run codes locally without ml-engine

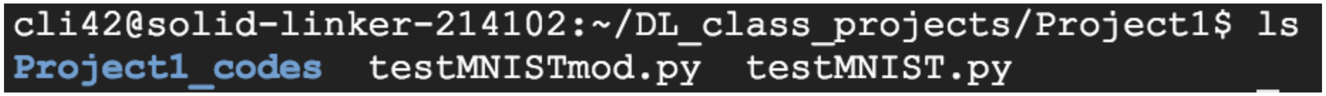
- Activate gcloud
  - If you activate your gcloud under a project, your working project is automatically set to that project.



| Name | Default storage class | Location | Public access | Lifecycle |
|---|---|---|---|---|
| mnist4project1 | Multi-Regional | US | Per object | None |

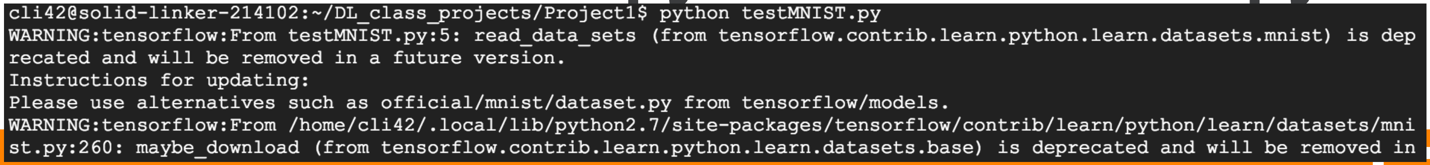
```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to solid-linker-214102.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
cli42@solid-linker-214102:~$
```

- Prepare your codes



```
cli42@solid-linker-214102:~/DL_class_projects/Project1$ ls
Project1_codes  testMNISTmod.py  testMNIST.py
```

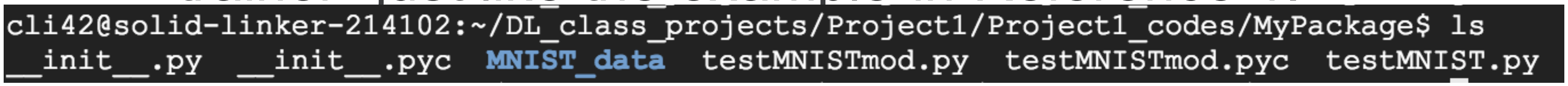
- Run the code with “**python testMNIST.py**”



```
cli42@solid-linker-214102:~/DL_class_projects/Project1$ python testMNIST.py
WARNING:tensorflow:From testMNIST.py:5: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is dep
recated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /home/cli42/.local/lib/python2.7/site-packages/tensorflow/contrib/learn/python/learn/datasets/mni
st.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in
```

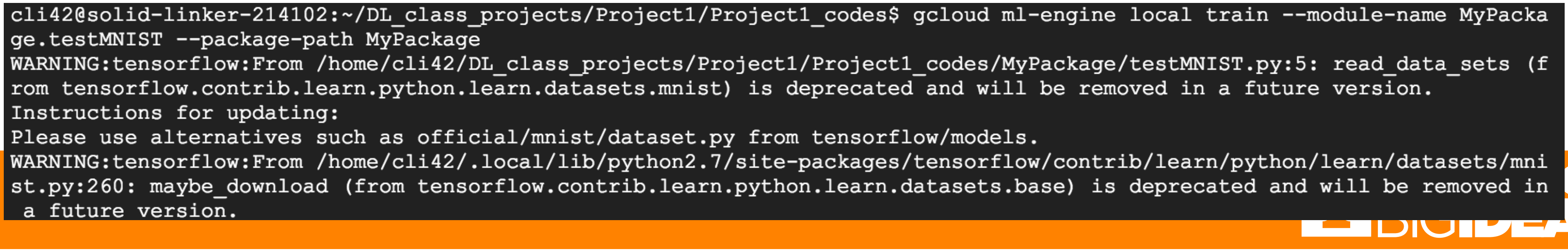
# Run codes locally with ml-engine

- Activate gcloud, verify Google Cloud SDK
- Prepare your codes
  - You need to modify the code and change it into a package. Here I used “MyPackage”. You can use whatever you like. It’s suggested to use the name “trainer” just like the example in Reference 1.



```
cli42@solid-linker-214102:~/DL_class_projects/Project1/Project1_codes/MyPackage$ ls
__init__.py  __init__.pyc  MNIST_data  testMNISTmod.py  testMNISTmod.pyc  testMNIST.py
```

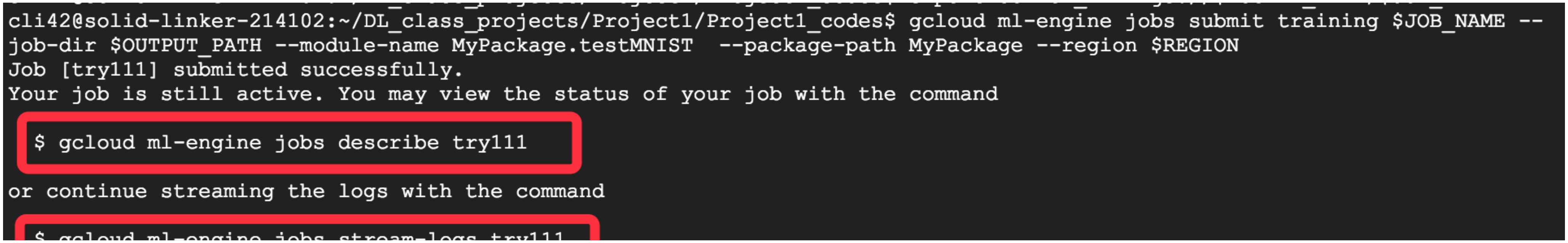
- Run the code with “gcloud ml-engine local train --module-name MyPackage.testMNIST --package-path MyPackage”



```
cli42@solid-linker-214102:~/DL_class_projects/Project1/Project1_codes$ gcloud ml-engine local train --module-name MyPacka
ge.testMNIST --package-path MyPackage
WARNING:tensorflow:From /home/cli42/DL_class_projects/Project1/Project1_codes/MyPackage/testMNIST.py:5: read_data_sets (f
rom tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /home/cli42/.local/lib/python2.7/site-packages/tensorflow/contrib/learn/python/learn/datasets/mni
st.py:260: maybe_download (from tensorflow.contrib.learn.python.learn.datasets.base) is deprecated and will be removed in
 a future version.
```

# Run codes on cloud with ml-engine and preinstalled instance

- Activate gcloud; verify Google Cloud SDK
- Set up a bucket
  - If you run on cloud, at least you need a bucket to store your logs. For Project 1, you don’t need a bucket to store your training data and output, but usually you do.
- Prepare your codes
- Run the code with
  - gcloud ml-engine jobs submit training $JOB_NAME --job-dir $OUTPUT_PATH --module-name MyPackage.testMNIST  --package-path MyPackage --region $REGION
  - I don’t want to bore you with setting up environmental variables. You can find the above environmental variables from Reference 1



```
cli42@solid-linker-214102:~/DL_class_projects/Project1/Project1_codes$ gcloud ml-engine jobs submit training $JOB_NAME --
job-dir $OUTPUT_PATH --module-name MyPackage.testMNIST  --package-path MyPackage --region $REGION
Job [try111] submitted successfully.
Your job is still active. You may view the status of your job with the command

  $ gcloud ml-engine jobs describe try111

or continue streaming the logs with the command

  $ gcloud ml-engine jobs stream-logs try111
```

# Run codes on cloud with ml-engine and self-created instance

- Create your own instance
- Activate gcloud; verify Google Cloud SDK
- Connect to your instance
  - gcloud compute ssh instance_name
- Install pip and other necessary packages
- Do similar things as method 3. The only difference is that now you're using self-created instance instead of preinstalled instance.

https://cloud.google.com/solutions/running-distributed-tensorflow-on-compute-engine

THANK
YOU