Git cheat sheet

Efim Abrikosov

April 13, 2019

1 Main commands

- `git help functionname` — display help information about Git
- `git init [--template=templatedirectory]` — create an empty Git repository or reinitialize an existing one. Files and directories in the template directory whose name do not start with a dot will be copied to the directory after it is created.
- `git clone repository [directory]` — clone a repository into a new directory. Optionally supply the name of a new directory to clone into.
- `git add [filename]` — add file contents to the index
- `git commit [-a][-m text]` — record changes to the repository
- `git diff [cached]`
- `git status`

1.1 Base workflow cases

2 Setting up a Google Cloud project

- Log in your google account
- Browse to cloud.google.com

- Click on “Go to Console”
- Go to Navigation menu (three horizontal lines in the top left corner)
- Select “Compute Engine”
- Click “Create”
- Select “Allow full access to all Cloud APIs”
- Now click “Create”
- Click on “SSH” field in the VM list to open the console
- To update the system configuration type “sudo apt-get update”
- To install Git type in “sudo apt-get -y -qq install git”
- Go to Navigation menu
- Select “Storage”
- Click “Create bucket”
- Select appropriate settings
- Now click “Create”
- In console type gsutil cp [filename] gs:://[bucketname]/[pathname]
- To publish cloud storage files to the web run gsutil acl ch -u AllUsers:R gs://[bucketname]/[pathname]
- To launch Cloud Datalab, open a Cloud Shell in the Platform page (the icon is in the top right corner)
- In Cloud Shell type “gcloud compute zones list”
- In Cloud Shell type “datalab create mydatalabvm –zone [zonename]”
- Creating Datalab VM may take several minutes
- Click on “Web Preview” button in the top of the Cloud Shell and change port to 8081

- Go to Navigation menu
- Select “BigQuery”
- In More Options click “Query settings”
- Under Additional Settings ensure that Legacy is not enabled
- In the query textbox type necessary SQL commands to extract data from big datasets
- Create a notebook in Datalab
- Define a valid query string in the notebook
- Use the following logic:
    1. import google.datalab.bigquery as bq
    2. df = bq.Query(query).execute().result().to_dataframe()
    3. df.head()
- Launch Cloud Datalab
- To download git repository contents use the logic
    1. %bash
    2. git clone [repositoryaddress] m -rf [pathname]
- Select APIs&Services from Cloud Platform Navigation Menu
- Click “Library” and search for required API (e.g. Cloud Vision API, Translate API, Speech API, or Natural Language API)
- Click “Enable” if necessary
- In APIs&Services click “Credentials” and create “API key” if necessary. This key will be used in Datalab code to invoke various APIs
- In Datalab use APIKEY generated in credentials as “developerKey” parameter in Datalab code
- In Datalab, run “!pip install –upgrade google-api-python-client”

  - Translate API
    1. from googleapiclient.discovery import build
    2. service = build('translate', 'v2', developerKey=APIKEY)
  - Vision API
    1. import base64
    2. vservice = build('vision', 'v1', developerKey=APIKEY)
  - Natural Language API
    1. lservice = build('language', 'v1beta1', developerKey=APIKEY)
  - Speech API
    1. build('speech', 'v1beta1', developerKey=APIKEY)

- Launch Cloud Datalab
- Use the following logic
  - google.datalab.bigquery as bq
  - qry = ''' SELECT * FROM ...'''
  - bq.Query(qry).execute().result().to_dataframe()

3 Useful links

- Git cheet sheet