Homework Week 1

1. Answer: --rm
2. Run: docker run -it --entrypoint=bash python:3.9  
   Answer: 0.42.0
3. Query:  
   SELECT COUNT(\*)

FROM green\_taxi

WHERE

DATE(lpep\_pickup\_datetime) = '2019-09-18'

AND DATE(lpep\_dropoff\_datetime) = '2019-09-18';  
  
Answer: 15612

1. Query:  
     
   SELECT

DATE(lpep\_pickup\_datetime) AS pick\_up\_day,

MAX(trip\_distance) AS largest\_trip\_distance

FROM green\_taxi

GROUP BY pick\_up\_day

ORDER BY largest\_trip\_distance DESC

LIMIT 1;  
  
Answer: 2019-09-26

1. Brookly, manhatten, queens  
     
   Select

"Borough" as borough,

SUM(total\_amount) as amount

FROM taxi\_zones as zones

INNER JOIN

green\_taxi as taxi

ON

"PULocationID" = "LocationID"

WHERE

DATE(taxi.lpep\_pickup\_datetime) = '2019-09-18'

AND "Borough" != 'Unknown'

GROUP BY

borough

HAVING

SUM(total\_amount) > 50000;

1. JFKAirport  
   select taxi\_zones."Zone" from taxi\_zones

join

(Select

"Borough", "Zone", "tip\_amount", "DOLocationID"

FROM

taxi\_zones

INNER JOIN

green\_taxi

ON

"LocationID" = "PULocationID"

WHERE

"Zone" = 'Astoria'

AND DATE(green\_taxi.lpep\_pickup\_datetime) BETWEEN '2019-09-01' AND'2019-09-30'

ORDER BY

tip\_amount DESC limit 1) as q1

on "LocationID" = "DOLocationID"

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# google\_bigquery\_dataset.demo\_dataset will be created

+ resource "google\_bigquery\_dataset" "demo\_dataset" {

+ creation\_time = (known after apply)

+ dataset\_id = "demo\_dataset"

+ default\_collation = (known after apply)

+ delete\_contents\_on\_destroy = false

+ effective\_labels = (known after apply)

+ etag = (known after apply)

+ id = (known after apply)

+ is\_case\_insensitive = (known after apply)

+ last\_modified\_time = (known after apply)

+ location = "EU"

+ max\_time\_travel\_hours = (known after apply)

+ project = "dataengineering-411512"

+ self\_link = (known after apply)

+ storage\_billing\_model = (known after apply)

+ terraform\_labels = (known after apply)

}

# google\_storage\_bucket.demo-bucket will be created

+ resource "google\_storage\_bucket" "demo-bucket" {

+ effective\_labels = (known after apply)

+ force\_destroy = true

+ id = (known after apply)

+ location = "EU"

+ name = "dataengineering-411512-terra-bucket"

+ project = (known after apply)

+ public\_access\_prevention = (known after apply)

+ self\_link = (known after apply)

+ storage\_class = "STANDARD"

+ terraform\_labels = (known after apply)

+ uniform\_bucket\_level\_access = (known after apply)

+ url = (known after apply)

+ lifecycle\_rule {

+ action {

+ type = "AbortIncompleteMultipartUpload"

}

+ condition {

+ age = 1

+ matches\_prefix = []

+ matches\_storage\_class = []

+ matches\_suffix = []

+ with\_state = (known after apply)

}

}

}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

google\_bigquery\_dataset.demo\_dataset: Creating...

google\_storage\_bucket.demo-bucket: Creating...

google\_bigquery\_dataset.demo\_dataset: Creation complete after 3s [id=projects/dataengineering-411512/datasets/demo\_dataset]

google\_storage\_bucket.demo-bucket: Creation complete after 3s [id=dataengineering-411512-terra-bucket]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.