Question 1. Knowing docker tags

Run the command to get information on Docker

```
docker --help
```

Now run the command to get help on the "docker build" command:

```
docker build --help
```

Do the same for "docker run".

Which tag has the following text? - *Automatically remove the container when it exits*

- `--delete`
- `--rc`
- `--rmc`
- **`--rm`**

```
wongs@LAPTOP-6FC3HDVG MINGW64 ~/OneDrive/Desktop/DE Zoom Camp
$ docker run --help
```

```
      --rm                             Automatically remove the container
                                       when it exits
```

Question 2. Understanding docker first run

Run docker with the python:3.9 image in an interactive mode and the entrypoint of bash. Now check the python modules that are installed (use `pip list`).

What is version of the package *wheel* ?

- **0.42.0**
- 1.0.0
- 23.0.1
- 58.1.0

```
wongs@LAPTOP-6FC3HDVG MINGW64 ~/OneDrive/Desktop/DE Zoom Camp
$ winpty docker run -it python:3.9 pip list
Package    Version
---------- -------
pip        23.0.1
setuptools 58.1.0
wheel      0.42.0
```

Question 3. Count records

How many taxi trips were totally made on September 18th 2019?

Tip: started and finished on 2019-09-18.

Remember that `lpep_pickup_datetime` and `lpep_dropoff_datetime` columns are in the format timestamp (date and hour+min+sec) and not in date.

- 15767
- **15612**
- 15859
- 89009

Query Query History

```
SELECT COUNT(*) FROM public.green_taxi_data
WHERE DATE(lpep_pickup_datetime) = '2019-09-18' AND DATE(lpep_dropoff_datetime) = '2019-09-18'
```

Data Output Messages Notifications

	count bigint
1	15612

SELECT COUNT(*) FROM public.green_taxi_data
WHERE DATE(lpep_pickup_datetime) = '2019-09-18' AND DATE(lpep_dropoff_datetime) = '2019-09-18'

Question 4. Largest trip for each day

Which was the pick up day with the largest trip distance Use the pick up time for your calculations.

- 2019-09-18
- 2019-09-16
- **2019-09-26**
- 2019-09-21

Query Query History

```
SELECT DATE(lpep_pickup_datetime) as pickup_date, MAX(trip_distance) as max_distance FROM public.green_taxi_data
GROUP BY pickup_date
ORDER BY max_distance DESC
```

Data Output Messages Notifications

	pickup_date date	max_distance double precision
1	2019-09-26	341.64
2	2019-09-21	135.53
3	2019-09-16	114.3
4	2019-09-28	89.64
5	2019-09-24	82.12
6	2019-09-18	70.28
7	2019-09-10	69.67
8	2019-09-27	68.41

Question 5. Three biggest pick up Boroughs

Consider lpep_pickup_datetime in '2019-09-18' and ignoring Borough has Unknown

Which were the 3 pick up Boroughs that had a sum of total_amount superior to 50000?

- **"Brooklyn" "Manhattan" "Queens"**
- "Bronx" "Brooklyn" "Manhattan"
- "Bronx" "Manhattan" "Queens"
- "Brooklyn" "Queens" "Staten Island"

```
SELECT * FROM
(SELECT
    z."Borough" AS pickup_loc,
    SUM(gt."total_amount") AS total_sum
FROM
    public.green_taxi_data gt
    LEFT JOIN zones z ON gt."PULocationID" = z."LocationID"
WHERE DATE(gt."lpep_pickup_datetime") ='2019-09-18'
GROUP BY z."Borough"
ORDER BY SUM(gt."total_amount") DESC LIMIT 10) p WHERE total_sum > 50000
```

Data Output Messages Notifications

	pickup_loc text	total_sum double precision
1	Brooklyn	96333.24000000482
2	Manhattan	92271.30000000489
3	Queens	78671.71000000215

SELECT * FROM
(SELECT
z."Borough" AS pickup_loc,
SUM(gt."total_amount") AS total_sum
FROM
public.green_taxi_data gt
LEFT JOIN zones z ON gt."PULocationID" = z."LocationID"
WHERE DATE(gt."lpep_pickup_datetime") ='2019-09-18'
GROUP BY z."Borough"
ORDER BY SUM(gt."total_amount") DESC LIMIT 10) p WHERE total_sum > 50000

Question 6. Largest tip

For the passengers picked up in September 2019 in the zone name Astoria which was the drop off zone that had the largest tip? We want the name of the zone, not the id.

Note: it's not a typo, it's `tip` , not `trip`

- Central Park
- Jamaica
- **JFK Airport**

- Long Island City/Queens Plaza

```sql
SELECT
    zpu."Zone" AS "pickup_loc",
    zdo."Zone" AS "dropoff_loc",
    MAX(tip_amount)
FROM
    green_taxi_data t
    JOIN zones zpu ON t."PULocationID" = zpu."LocationID"
    JOIN zones zdo ON t."DOLocationID" = zdo."LocationID"
WHERE
    zpu."Zone" = 'Astoria'
    AND to_char(lpep_pickup_datetime, 'YYYY-MM') = '2019-09'
GROUP BY 1, 2
ORDER BY 3 DESC LIMIT 10;
```

Data Output Messages Notifications

	pickup_loc text	dropoff_loc text	max double precision
1	Astoria	JFK Airport	62.31
2	Astoria	Woodside	30
3	Astoria	Kips Bay	28
4	Astoria	NV	25
5	Astoria	Astoria	20
6	Astoria	Upper West Side South	20
7	Astoria	[null]	19.28
8	Astoria	Newark Airport	19.01

Total rows: 10 of 10 Query complete 00:00:00.894

```sql
SELECT
        zpu."Zone" AS "pickup_loc",
  zdo."Zone" AS "dropoff_loc",
        MAX(tip_amount)
FROM
        green_taxi_data t
        JOIN zones zpu ON t."PULocationID" = zpu."LocationID"
        JOIN zones zdo ON t."DOLocationID" = zdo."LocationID"
WHERE
  zpu."Zone" = 'Astoria'
        AND to_char(lpep_pickup_datetime, 'YYYY-MM') = '2019-09'
GROUP BY 1, 2
ORDER BY 3 DESC LIMIT 10;
```

Question 7. Creating Resources

After updating the main.tf and variable.tf files run:

```
terraform apply
```

Paste the output of this command into the homework submission form.

```
C:\Users\wongs\OneDrive\Desktop\zoomcamp2024\Week 1 Homework>terraform apply -var="project=mindful-future-412612"
google_bigquery_dataset.demo_dataset: Refreshing state... [id=projects/mindful-future-412612/datasets/demo_dataset]

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_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                    = "US"
      + name                        = "mindful-future-412612-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: 1 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_storage_bucket.demo-bucket: Creating...
google_storage_bucket.demo-bucket: Creation complete after 2s [id=mindful-future-412612-terra-bucket]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Users\wongs\OneDrive\Desktop\zoomcamp2024\Week 1 Homework>
```

C:\Users\wongs\OneDrive\Desktop\zoomcamp2024\Week 1 Homework>terraform apply

```
-var="project=mindful-future-412612"
google_bigquery_dataset.demo_dataset: Refreshing state...
[id=projects/mindful-future-412612/datasets/demo_dataset]

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_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                    = "US"
      + name                        = "mindful-future-412612-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: 1 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_storage_bucket.demo-bucket: Creating...
google_storage_bucket.demo-bucket: Creation complete after 2s
[id=mindful-future-412612-terra-bucket]
```

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```