

Week 2 Homework: Workflow Orchestration and Observation

Georgios Grigoriou

georgiosvgrigoriou@gmail.com

Deadline: 06 February (Monday), 22:00 CET

In this homework, I prepared the environment and practice with Prefect, a workflow orchestration tool as part of the [Data Engineering Zoomcamp course](#). Solutions' headings are in italic and highlighted in yellow under each Question's instruction.

The code that was used can be found [here](#)

Week 2 Homework

The goal of this homework is to familiarise myself with workflow orchestration and observation.

Question 1. Load January 2020 data

Using the `etl_web_to_gcs.py` flow that loads taxi data into GCS as a guide, create a flow that loads the green taxi CSV dataset for January 2020 into GCS and run it. Look at the logs to find out how many rows the dataset has.

How many rows does that dataset have?

- **447,770**
- 766,792
- 299,234
- 822,132

Question 1. Load January 2020 data Prep and Solution

Using the `etl_web_to_gcs.py` file we configure the flow as follows:

color="green"

year="2020"

month="01"

```

EXPLORER          etl_web_to_gcs.py 3
flows > 02_gcp > etl_web_to_gcs.py > etl_web_to_gcs
  33     df.to_parquet(path, compression='gzip')
  34     return path
  35
  36
  37     @task()
  38     def write_gcs(path: Path) -> None:
  39         """Upload local parquet file to GCS"""
  40         gcs_block = GcsBucket.load("zoom-gcs")
  41         gcs_block.upload_from_path(from_path=path, to_path=path)
  42
  43
  44
  45     @flow()
  46     def etl_web_to_gcs() -> None:
  47         """The main ETL function"""
  48         color = "green"
  49         year = 2020
  50         month = "01"
  51
  52         dataset_file = f"{color}_tripdata_{year}-{month}"
  53         dataset_url = f"https://github.com/DataTalksClub/nyc-tlc-data/releases/download/{color}/{dataset_file}.csv.gz"
  54
  55         df = fetch(dataset_url)
  56         df_clean = clean(df)
  57         path = write_local(df_clean, color, dataset_file)
  58         write_gcs(path)
  59
  60     if __name__ == "__main__":
  61         etl_web_to_gcs()
  62

```

I found out that the correct answer is **447,770**

The figures below show the required information about the number of rows from the Logs Section of Prefect UI and the terminal of Visual Studio Code respectively.

Log Entry	Timestamp	Task / Action	Details
INFO Created task run 'fetch-b4598a4a-0' for task 'fetch'	Feb 6th, 2023 04:31:48	Task Run	
INFO Executing 'fetch-b4598a4a-0' immediately...	04:31:48	Task Run	
INFO Finished in state Completed()	04:31:50	Task Run	fetch-b4598a4a-0
INFO Created task run 'clean-b9fd7e03-0' for task 'clean'	04:31:50	Task Run	
INFO Executing 'clean-b9fd7e03-0' immediately...	04:31:50	Task Run	
INFO VendorID lpep_pickup_datetime lpep_dropoff_datetime ... payment_type trip_type congestion_surcharge 0 2.0 2019-12-18 15:52:30 2019-12-18 15:54:39 ... 1.0 1.0 0.0 1 2.0 2020-01-01 00:45:58 2020-01-01 00:56:39 ... 1.0 2.0 0.0	04:31:50	Data Preview	clean-b9fd7e03-0
INFO [2 rows x 20 columns]		Data Preview	
INFO columns: VendorID float64 lpep_pickup_datetime datetime64[ns] lpep_dropoff_datetime datetime64[ns] store_and_fwd_flag object RatecodeID float64 PULocationID int64 DOLocationID int64 passenger_count float64 trip_distance float64	04:31:50	Data Preview	clean-b9fd7e03-0

```

1pep_pickup_datetime      datetime64[ns]
1pep_dropoff_datetime    datetime64[ns]
store_and_fwd_flag        object
RatecodeID                float64
PULocationID              int64
DOLocationID              int64
passenger_count            float64
trip_distance              float64
fare_amount                float64
extra                      float64
mta_tax                     float64
tip_amount                 float64
tolls_amount                float64
ehail_fee                  float64
improvement_surcharge     float64
total_amount                float64
payment_type                float64
trip_type                  float64
congestion_surcharge       float64
dtype: object

INFO rows: 447770

INFO Finished in state Completed()

INFO Created task run 'write_local-f322d1be-0' for task 'write_local'

INFO Executing 'write_local-f322d1be-0' immediately...

INFO Finished in state Completed()

INFO Created task run 'write_gcs-1145c921-0' for task 'write_gcs'

INFO Executing 'write_gcs-1145c921-0' immediately...

INFO Getting bucket 'prefect-de-zoomcampg'.

INFO Uploading from PosixPath('data/green/green_tripdata_2020-01.parquet') to the bucket
'prefect-de-zoomcampg' path 'data/green/green_tripdata_2020-01.parquet'.

INFO Finished in state Completed()

```

```

EXPLORER ... etl_web_to_gcs.py 3
flows > 02_gcp > etl_web_to_gcs.py > clean
23     print(df.head(2))
24     print(f"columns: {df.dtypes}")
25     print(f"rows: {len(df)}")
26     return df
27
28
29 @task()
30 def write_local(df: pd.DataFrame, color: str, dataset_file: str) -> Path:
31     """Write Dataframe out locally as parquet file"""
32     path = Path(f"data/{color}/{dataset_file}.parquet")
33     df.to_parquet(path, compression="gzip")
34     return path
35
36
37 @task()
38 def write_gcs(path: Path) -> None:
39     """Upload local parquet file to GCS"""
40     gcs_block = GcsBucket.load("zoom-gcs")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

lpep_pickup_datetime      datetime64[ns]  INFO Task run 'clean-09d7e03-0' - rows: 447770
lpep_dropoff_datetime    datetime64[ns]  INFO Task run 'clean-09d7e03-0' - Finished in state Completed()
store_and_fwd_flag        object          INFO Flow run 'yellow-leech' - Created task run 'write_local-f322d1be-0' for task 'write_local'
RatecodeID                float64         INFO Flow run 'yellow-leech' - Executing 'write_local-f322d1be-0' immediately...
PULocationID              int64          INFO Flow run 'yellow-leech' - Finished in state Completed()
DOLocationID              int64          INFO Task run 'write_local-f322d1be-0' - Finished in state Completed()
passenger_count            float64         INFO Flow run 'yellow-leech' - Created task run 'write_gcs-1145c921-0' for task 'write_gcs'
trip_distance              float64         INFO Flow run 'yellow-leech' - Executing 'write_gcs-1145c921-0' immediately...
fare_amount                float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
extra                      float64         INFO Task run 'write_gcs-1145c921-0' - Uploading from PosixPath('data/green/green_tripdata_2020-01.parquet') to
mta_tax                     float64         INFO the bucket 'prefect-de-zoomcampg' path 'data/green/green_tripdata_2020-01.parquet'
tip_amount                 float64         INFO Task run 'write_gcs-1145c921-0' - Finished in state Completed()
tolls_amount                float64         INFO Flow run 'yellow-leech' - Finished in state Completed('All states completed.')
ehail_fee                  float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
improvement_surcharge     float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
total_amount                float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
payment_type                float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
trip_type                  float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
congestion_surcharge       float64         INFO Flow run 'yellow-leech' - Finished in state Completed()
dtype: object

```

Question 2. Scheduling with Cron

Cron is a common scheduling specification for workflows.

Using the flow in `etl_web_to_gcs.py`, create a deployment to run on the first of every month at 5am UTC. What's the cron schedule for that?

- 0 5 1 * *
- 0 0 5 1 *
- 5 * 1 0 *
- * * 5 1 0

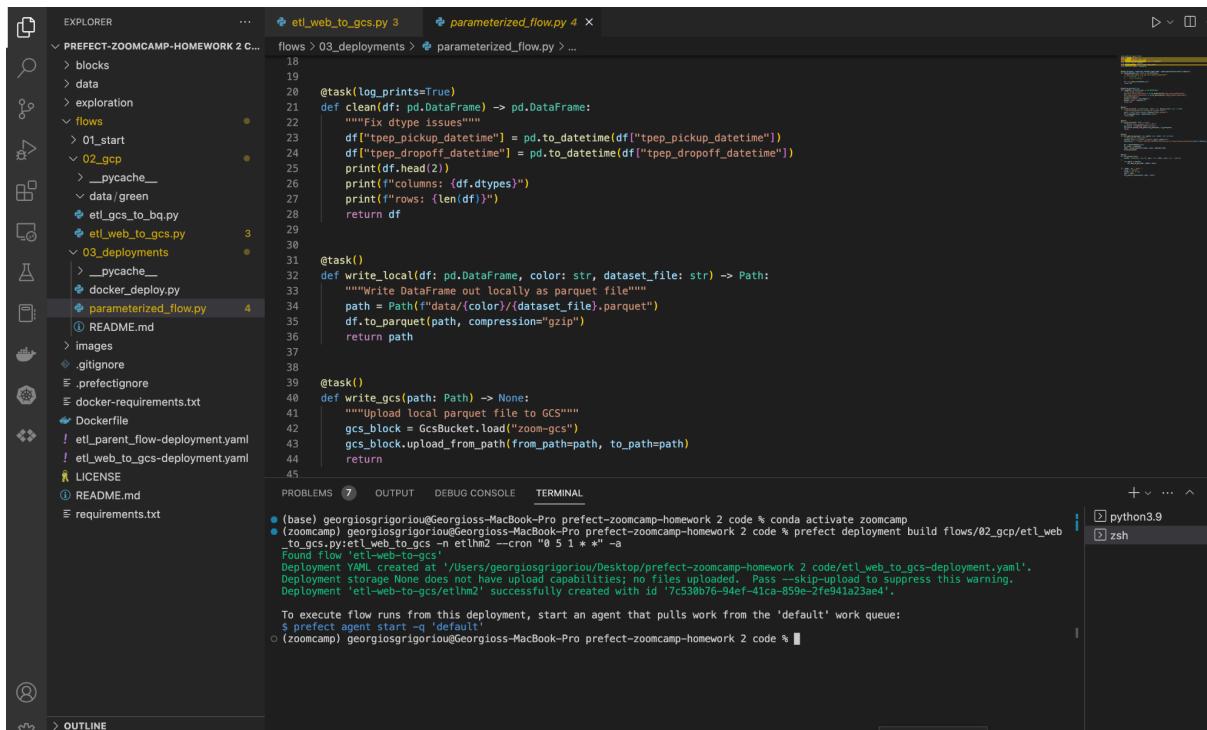
Question 2. Scheduling with Cron Solution

The correct answer is `0 5 1 * *`.

To get that, I ran the following command

```
prefect deployment build flows/02_gcp/etl_web_to_gcs.py:etl_web_to_gcs -n ethhm2 --cron  
"0 5 1 * *" -a
```

From the terminal in my virtual environment at Visual Studio Code, as illustrated in the figure below



The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree for a project named 'PREFECT-ZOOMCAMP-HOMEWORK 2 C...'. The main code editor window shows the contents of 'flows/02_gcp/etl_web_to_gcs.py'. The terminal window at the bottom shows the command being run:

```
(base) georgiosgrigoriou@Georgioss-MacBook-Pro prefect-zoomcamp-homework 2 code % conda activate zoomcamp
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro prefect-zoomcamp-homework 2 code % prefect deployment build flows/02_gcp/etl_web_to_gcs.py:etl_web_to_gcs -n ethhm2 --cron "0 5 1 * *"
Found flow 'etl_web_to_gcs'
Deployment Yaml created at '/Users/georgiosgrigoriou/Desktop/prefect-zoomcamp-homework 2/code/etl_web_to_gcs-deployment.yaml'.
Deployment storage engine does not have upload capabilities; no files uploaded. Pass --skip-upload to suppress this warning.
Deployment 'etl_web_to_gcs/ethhm2' successfully created with id '7c530b76-94ef-41ca-859e-2fe941a23ae4'.
```

The terminal also shows the command being run again:

```
To execute flow runs from this deployment, start an agent that pulls work from the 'default' work queue:
$ prefect agent start -q 'default'
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro prefect-zoomcamp-homework 2 code %
```

Now, if we go to my local Prefect UI, I can see the deployment named `ethhm2`, which was scheduled to run at 05:00 AM on day 1 of the month

Question 3. Loading data to BigQuery

Using `etl_gcs_to_bq.py` as a starting point, modify the script for extracting data from GCS and loading it into BigQuery. This new script should not fill or remove rows with missing values. (The script is really just doing the E and L parts of ETL).

The main flow should print the total number of rows processed by the script. Set the flow decorator to log the print statement.

Parametrize the entrypoint flow to accept a list of months, a year, and a taxi color.

Make any other necessary changes to the code for it to function as required.

Create a deployment for this flow to run in a local subprocess with local flow code storage (the defaults).

Make sure you have the parquet data files for Yellow taxi data for Feb. 2019 and March 2019 loaded in GCS. Run your deployment to append this data to your BigQuery table. How many rows did your flow code process?

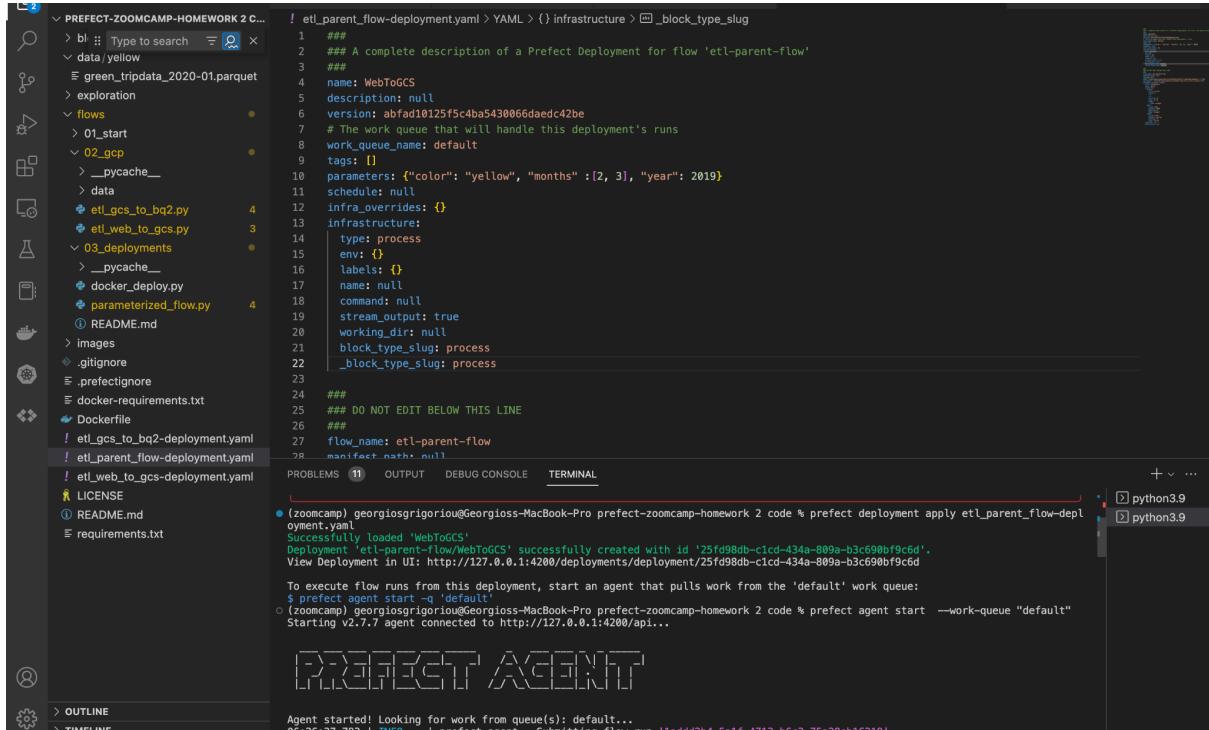
- **14,851,920**
- 12,282,990
- 27,235,753
- 11,338,483

Question 3. Loading data to BigQuery Solution

The correct answer is 14,851,920. Here are the Steps that were followed:

Step 1: Store parquet data files for Yellow taxi data for Feb. 2019 and March 2019 loaded in GCS.

To do that I ran a deployment called WebToGCS and apply the appropriate changes in the parameter key of the yaml file

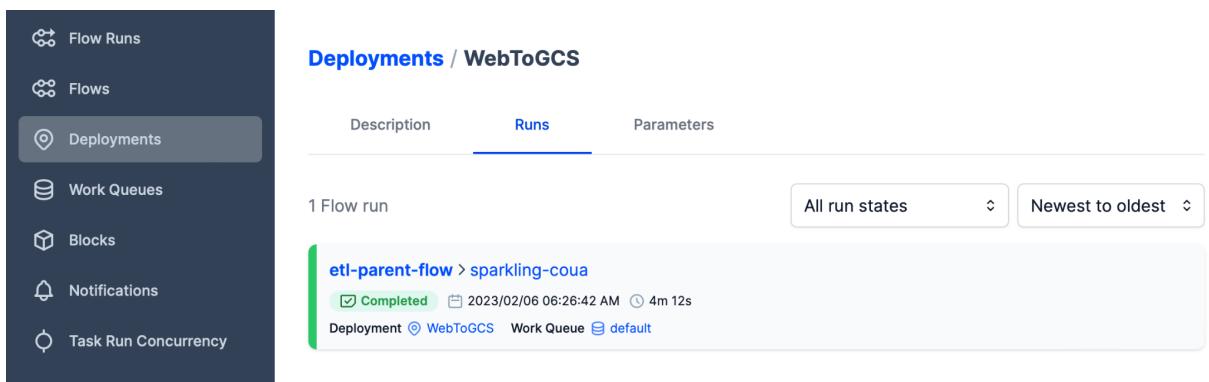


The screenshot shows a terminal window with the following command and output:

```
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro prefect-zoomcamp-homework 2 code % prefect deployment apply etl_parent_flow-deployment.yaml
Successfully loaded 'WebToGCS'
Deployment 'etl-parent-flow/WebToGCS' successfully created with id '25fd98db-c1cd-434a-809a-b3c690bf9c6d'.
View Deployment in UI: http://127.0.0.1:4200/deployments/deployment/25fd98db-c1cd-434a-809a-b3c690bf9c6d

To execute flow runs from this deployment, start an agent that pulls work from the 'default' work queue:
$ prefect agent start -q 'default'

(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro prefect-zoomcamp-homework 2 code % prefect agent start --work-queue "default"
Starting v2.7.7 agent connected to http://127.0.0.1:4200/api...
```



The screenshot shows the Prefect Agent interface. On the left, there is a sidebar with the following navigation items:

- Flow Runs
- Flows
- Deployments
- Work Queues
- Blocks
- Notifications
- Task Run Concurrency

The "Deployments" item is highlighted. On the right, the main area displays the "Deployments / WebToGCS" page. It shows a table with the following columns:

Description	Runs	Parameters
1 Flow run	All run states	Newest to oldest
etl-parent-flow > sparkling-coua	Completed	2023/02/06 06:26:42 AM 4m 12s
	Deployment	WebToGCS Work Queue default

Terminal commands (in *Italic*)

```
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro
prefect-zoomcamp-homework 2 code % prefect deployment build
flows/03_deployments/parameterized_flow.py:etl_parent_flow -n "WebToGCS"
```

Found flow 'etl-parent-flow'

Deployment YAML created at
 '/Users/georgiosgrigoriou/Desktop/prefect-zoomcamp-homework 2
 code/etl_parent_flow-deployment.yaml'.

Deployment storage None does not have upload capabilities; no files uploaded.
 Pass --skip-upload to suppress this warning.

```
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro
prefect-zoomcamp-homework 2 code % prefect deployment apply
etl_parent_flow-deployment.yaml
```

Successfully loaded 'WebToGCS'

Deployment 'etl-parent-flow/WebToGCS' successfully created with id
 '25fd98db-c1cd-434a-809a-b3c690bf9c6d'.

View Deployment in UI:

<http://127.0.0.1:4200/deployments/deployment/25fd98db-c1cd-434a-809a-b3c690bf9c6d>

To execute flow runs from this deployment, start an agent that pulls work from the
 'default' work queue:

\$ prefect agent start -q 'default'

```
(zoomcamp) georgiosgrigoriou@Georgioss-MacBook-Pro
prefect-zoomcamp-homework 2 code % prefect agent start --work-queue
"default"
```

```
Starting v2.7.7 agent connected to http://127.0.0.1:4200/api...
```

```
-----  
|_ \_ \_ | _| _/ _| _| / \_ | _| \| | _| | |
| _/ / _|| _|(_| _| / _\(_| _||`| _|  
|_| _|\_ | _| _\| _| _| / / \_ | _| \| | _|
```

```
Agent started! Looking for work from queue(s): default...
```

```
06:26:37.793 | INFO | prefect.agent - Submitting flow run
'1add2b4-5a1f-4713-b6c3-75a28eb16219'
```

```
06:26:37.872 | INFO | prefect.infrastructure.process - Opening process
'sparkling-coua'...
```

```
06:26:37.902 | INFO | prefect.agent - Completed submission of flow run
'1add2b4-5a1f-4713-b6c3-75a28eb16219'
```

```
/opt/anaconda3/envs/zoomcamp/lib/python3.9/runpy.py:127: RuntimeWarning:
'prefect.engine' found in sys.modules after import of package 'prefect', but prior to
execution of 'prefect.engine'; this may result in unpredictable behaviour
```

```
warn(RuntimeWarning(msg))
```

```
06:26:42.603 | INFO | Flow run 'sparkling-coua' - Downloading flow code from
storage at '/Users/georgiosgrigoriou/Desktop/prefect-zoomcamp-homework 2 code'
```

```
06:26:42.972 | INFO | Flow run 'sparkling-coua' - Created subflow run
'practical-armadillo' for flow 'etl-web-to-gcs'
```

```
06:26:43.050 | INFO | Flow run 'practical-armadillo' - Created task run
'fetch-ba00c645-0' for task 'fetch'
```

```
06:26:43.051 | INFO | Flow run 'practical-armadillo' - Executing 'fetch-ba00c645-0'
immediately...
```

```
06:28:22.554 | INFO | Task run 'fetch-ba00c645-0' - Finished in state Completed()
```

06:28:22.621 | INFO | Flow run 'practical-armadillo' - Created task run 'clean-2c6af9f6-0' for task 'clean'

06:28:22.621 | INFO | Flow run 'practical-armadillo' - Executing 'clean-2c6af9f6-0' immediately...

06:28:27.401 | INFO | Task run 'clean-2c6af9f6-0' - VendorID
tpep_pickup_datetime ... total_amount congestion_surcharge

0	1	2019-02-01 00:59:04	...	12.3	0.0
1	1	2019-02-01 00:33:09	...	33.3	0.0

[2 rows x 18 columns]

06:28:27.404 | INFO | Task run 'clean-2c6af9f6-0' - columns: VendorID
int64

tpep_pickup_datetime datetime64[ns]

tpep_dropoff_datetime datetime64[ns]

passenger_count int64

trip_distance float64

RatecodeID int64

store_and_fwd_flag object

PULocationID int64

DOLocationID int64

payment_type int64

fare_amount float64

extra float64

mta_tax float64

tip_amount float64

tolls_amount float64

improvement_surcharge float64

total_amount float64

congestion_surcharge float64

dtype: object

06:28:27.405 | INFO | Task run 'clean-2c6af9f6-0' - rows: 7019375

06:28:27.430 | INFO | Task run 'clean-2c6af9f6-0' - Finished in state Completed()

06:28:27.457 | INFO | Flow run 'practical-armadillo' - Created task run 'write_local-09e9d2b8-0' for task 'write_local'

06:28:27.457 | INFO | Flow run 'practical-armadillo' - Executing 'write_local-09e9d2b8-0' immediately...

06:28:48.256 | INFO | Task run 'write_local-09e9d2b8-0' - Finished in state Completed()

06:28:48.281 | INFO | Flow run 'practical-armadillo' - Created task run 'write_gcs-67f8f48e-0' for task 'write_gcs'

06:28:48.282 | INFO | Flow run 'practical-armadillo' - Executing 'write_gcs-67f8f48e-0' immediately...

06:28:48.411 | INFO | Task run 'write_gcs-67f8f48e-0' - Getting bucket 'prefect-de-zoomcampgg'.

06:28:48.676 | INFO | Task run 'write_gcs-67f8f48e-0' - Uploading from PosixPath('data/yellow/yellow_tripdata_2019-02.parquet') to the bucket 'prefect-de-zoomcampgg' path 'data/yellow/yellow_tripdata_2019-02.parquet'.

06:28:50.585 | INFO | Task run 'write_gcs-67f8f48e-0' - Finished in state Completed()

06:28:50.632 | INFO | Flow run 'practical-armadillo' - Finished in state Completed('All states completed.')

06:28:50.749 | INFO | Flow run 'sparkling-coua' - Created subflow run 'poetic-leech' for flow 'etl-web-to-gcs'

06:28:50.826 | INFO | Flow run 'poetic-leech' - Created task run 'fetch-ba00c645-0' for task 'fetch'

06:28:50.827 | INFO | Flow run 'poetic-leech' - Executing 'fetch-ba00c645-0' immediately...

06:30:23.751 | INFO | Task run 'fetch-ba00c645-0' - Finished in state Completed()

06:30:23.823 | INFO | Flow run 'poetic-leech' - Created task run 'clean-2c6af9f6-0' for task 'clean'

06:30:23.824 | INFO | Flow run 'poetic-leech' - Executing 'clean-2c6af9f6-0' immediately...

06:30:28.670 | INFO | Task run 'clean-2c6af9f6-0' - VendorID
tpep_pickup_datetime ... total_amount congestion_surcharge

0	1	2019-03-01 00:24:41	...	3.8	0.0
1	1	2019-03-01 00:25:27	...	15.0	0.0

[2 rows x 18 columns]

06:30:28.672 | INFO | Task run 'clean-2c6af9f6-0' - columns: VendorID
int64

tpep_pickup_datetime datetime64[ns]

tpep_dropoff_datetime datetime64[ns]

passenger_count int64

trip_distance float64

RatecodeID int64

store_and_fwd_flag object

PULocationID int64

DOLocationID int64

payment_type int64

fare_amount float64

extra float64

mta_tax float64

tip_amount float64

tolls_amount float64

improvement_surcharge float64

total_amount float64

congestion_surcharge float64

dtype: object

06:30:28.673 | INFO | Task run 'clean-2c6af9f6-0' - rows: 7832545

06:30:28.696 | INFO | Task run 'clean-2c6af9f6-0' - Finished in state Completed()

06:30:28.724 | INFO | Flow run 'poetic-leech' - Created task run 'write_local-09e9d2b8-0' for task 'write_local'

06:30:28.725 | INFO | Flow run 'poetic-leech' - Executing 'write_local-09e9d2b8-0' immediately...

06:30:49.796 | INFO | Task run 'write_local-09e9d2b8-0' - Finished in state Completed()

06:30:49.821 | INFO | Flow run 'poetic-leech' - Created task run 'write_gcs-67f8f48e-0' for task 'write_gcs'

06:30:49.822 | INFO | Flow run 'poetic-leech' - Executing 'write_gcs-67f8f48e-0' immediately...

06:30:49.954 | INFO | Task run 'write_gcs-67f8f48e-0' - Getting bucket 'prefect-de-zoomcampgg'.

06:30:50.185 | INFO | Task run 'write_gcs-67f8f48e-0' - Uploading from PosixPath('data/yellow/yellow_tripdata_2019-03.parquet') to the bucket 'prefect-de-zoomcampgg' path 'data/yellow/yellow_tripdata_2019-03.parquet'.

06:30:53.882 | INFO | Task run 'write_gcs-67f8f48e-0' - Finished in state Completed()

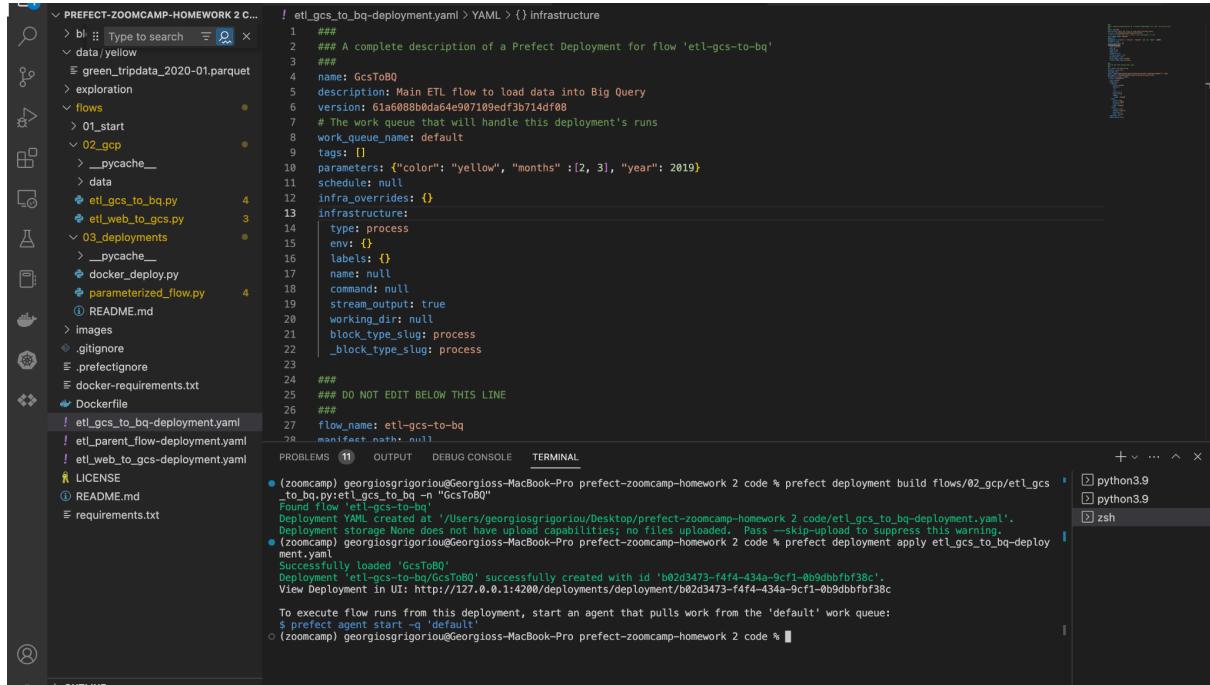
06:30:53.929 | INFO | Flow run 'poetic-leech' - Finished in state Completed('All states completed.')

06:30:53.958 | INFO | Flow run 'sparkling-coua' - Finished in state Completed('All states completed.')

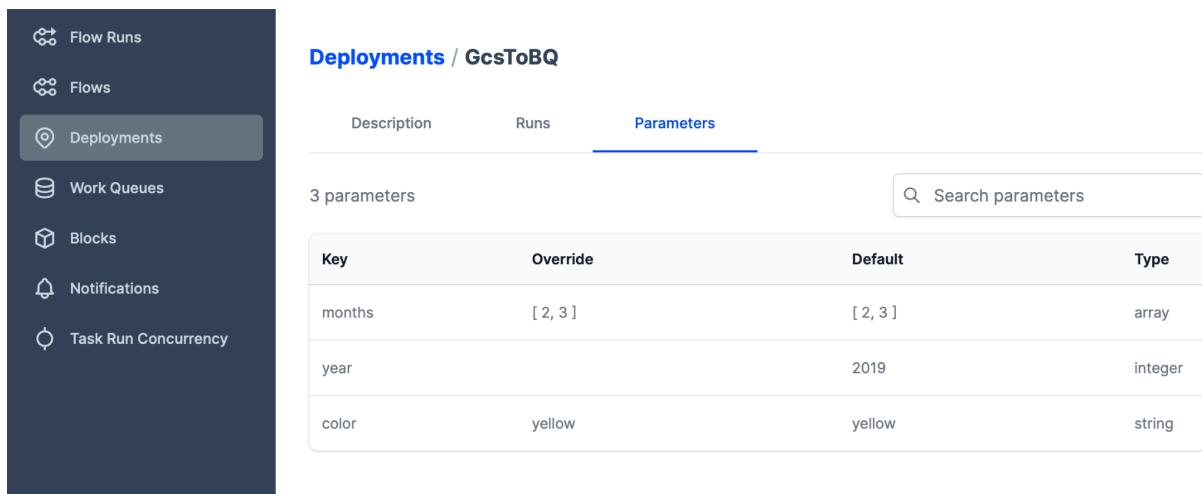
06:30:54.888 | INFO | prefect.infrastructure.process - Process 'sparkling-coua' exited cleanly.

Step 2: gcp to bq.py

With the same logic as Step 1, I built and applied deployment by modifying the `etl_gcs_to_bq.py` file in order to store data from Google Cloud Storage to Big Query
Modified script can be found below and in the github repo by clicking [here](#).



```
! etl_gcs_to_bq-deployment.yaml > YAML > {} infrastructure
1  ###
2  ### A complete description of a Prefect Deployment for flow 'etl-gcs-to-bq'
3  ###
4  name: GcsToBQ
5  description: Main ETL flow to load data into Big Query
6  version: 61a6088bbdd64e907109edf3b714df08
7  # The work queue that will handle this deployment's runs
8  work_queue_name: default
9  tags: []
10 parameters: {"color": "yellow", "months": [2, 3], "year": 2019}
11 schedule: null
12 infra_overrides: {}
13 infrastructure:
14   type: process
15   env: {}
16   labels: {}
17   name: null
18   command: null
19   stream_output: true
20   working_dir: null
21   block_type_slug: process
22   _block_type_slug: process
23
24 ###
25 ### DO NOT EDIT BELOW THIS LINE
26 ###
27 flow_name: etl-gcs-to-bq
28 manifest_path: null
29
```



Key	Override	Default	Type
months	[2, 3]	[2, 3]	array
year		2019	integer
color	yellow	yellow	string

Feb 6th, 2023

```
[INFO] Downloading flow code from storage at '/Users/georgiosgrigoriou/Desktop/prefect-zoomcamp-homework' 07:35:08 AM  
2 code'  
[INFO] Created task run 'extract_from_gcs-272ec809-0' for task 'extract_from_gcs' 07:35:09 AM  
[INFO] Executing 'extract_from_gcs-272ec809-0' immediately... 07:35:09 AM  
[INFO] Downloading blob named data/yellow/yellow_tripdata_2019-02.parquet from the prefect-de-zoomcampgg bucket to ../data/data/yellow/yellow_tripdata_2019-02.parquet 07:35:09 AM  
[INFO] extract_from_gcs-272ec809-0  
[INFO] Finished in state Completed() 07:35:11 AM  
[INFO] extract_from_gcs-272ec809-0  
[INFO] total_length: 7019375 07:35:14 AM  
[INFO] Created task run 'write_bq-f3b17cf5-0' for task 'write_bq' 07:35:14 AM  
[INFO] Executing 'write_bq-f3b17cf5-0' immediately... 07:35:14 AM  
[INFO] post: rows: 7019375 07:35:43 AM  
[INFO] write_bq-f3b17cf5-0  
[INFO] Finished in state Completed() 07:35:43 AM  
[INFO] write_bq-f3b17cf5-0  
[INFO] Created task run 'extract_from_gcs-272ec809-1' for task 'extract_from_gcs' 07:35:43 AM  
[INFO] Executing 'extract_from_gcs-272ec809-1' immediately... 07:35:43 AM  
[INFO] Downloading blob named data/yellow/yellow_tripdata_2019-03.parquet from the prefect-de-zoomcampgg bucket to ../data/data/yellow/yellow_tripdata_2019-03.parquet 07:35:43 AM  
[INFO] extract_from_gcs-272ec809-1  
[INFO] Finished in state Completed() 07:35:45 AM  
[INFO] extract_from_gcs-272ec809-1  
[INFO] total_length: 14851920 07:35:49 AM  
[INFO] Created task run 'write_bq-f3b17cf5-1' for task 'write_bq' 07:35:49 AM  
[INFO] Executing 'write_bq-f3b17cf5-1' immediately... 07:35:49 AM
```

```

26     if_exists="append"
27   )
28   print(f"post: rows: {len(df)}")
29
30
31 @flow(log_prints=True)
32 def etl_gcs_to_bq(
33   months: list[int] = [2, 3], year: int = 2019, color: str = "yellow"
34 ):
35   """Main ETL flow to load data into Big Query"""
36
37   total_length=0
38   for month in months:
39     path = extract_from_gcs(color, year, month)
40     df = pd.read_parquet(path)
41     total_length+=len(df)
42     print(f"total_length: {total_length}")
43     write_bq(df)
44
45 if __name__ == "__main__":
46   months = [2, 3]
47   year = 2019
48   color = "yellow"
49   etl_gcs_to_bq(months,year,color)

```

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL

```

t from the prefect-de-zoomcampg bucket to ./data/data/yellow/yellow_tripdata_2019-02.parquet
07:35:11.322 | INFO | Task run 'extract_from_gcs-272ec809-0' - Finished in state Completed()
07:35:14.899 | INFO | Flow run 'olive-pheasant' - total_length: 7019375
07:35:14.936 | INFO | Flow run 'olive-pheasant' - Created task run 'write_bq-f3b17cf5-0' for task 'write_bq'
07:35:14.936 | INFO | Flow run 'olive-pheasant' - Executing 'write_bq-f3b17cf5-0' immediately...
07:35:43.152 | INFO | Task run 'write_bq-f3b17cf5-0' - post: rows: 7019375
07:35:43.188 | INFO | Task run 'write_bq-f3b17cf5-0' - Finished in state Completed()
07:35:43.215 | INFO | Flow run 'olive-pheasant' - Created task run 'extract_from_gcs-272ec809-1' for task 'extract_from_gcs'
07:35:43.215 | INFO | Flow run 'olive-pheasant' - Executing 'extract_from_gcs-272ec809-1' immediately...
07:35:43.553 | INFO | Task run 'extract_from_gcs-272ec809-1' - Downloading blob named data/yellow/yellow_tripdata_2019-03.parquet from the prefect-de-zoomcampg bucket to ./data/data/yellow/yellow_tripdata_2019-03.parquet
07:35:45.532 | INFO | Task run 'extract_from_gcs-272ec809-1' - Finished in state Completed()
07:35:49.598 | INFO | Flow run 'olive-pheasant' - total_length: 14851920
07:35:49.638 | INFO | Flow run 'olive-pheasant' - Created task run 'write_bq-f3b17cf5-1' for task 'write_bq'
07:35:49.631 | INFO | Flow run 'olive-pheasant' - Executing 'write_bq-f3b17cf5-1' immediately...
07:36:31.542 | INFO | Task run 'write_bq-f3b17cf5-1' - post: rows: 7832545
07:36:31.575 | INFO | Task run 'write_bq-f3b17cf5-1' - Finished in state Completed()
07:36:31.610 | INFO | Flow run 'olive-pheasant' - Finished in state Completed('All states completed.')
07:36:32.412 | INFO | prefect.infrastructure.process - Process 'olive-pheasant' exited cleanly.

```

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH		PREVIEW
Row	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	
1	2.0	2019-03-14 20:56:08 UTC	2019-03-14 21:07:53 UTC	1.0	2.1	1.0	N	
2	2.0	2019-03-14 20:02:11 UTC	2019-03-14 20:16:42 UTC	1.0	2.24	1.0	N	
3	1.0	2019-03-14 20:43:11 UTC	2019-03-14 20:50:19 UTC	1.0	1.4	1.0	N	
4	2.0	2019-03-14 20:39:25 UTC	2019-03-14 21:07:21 UTC	1.0	8.83	1.0	N	
5	1.0	2019-03-14 20:26:31 UTC	2019-03-14 20:46:41 UTC	1.0	3.9	1.0	N	
6	1.0	2019-03-14 20:00:35 UTC	2019-03-14 20:08:00 UTC	2.0	1.4	1.0	N	
7	2.0	2019-03-14 20:03:46 UTC	2019-03-14 20:20:19 UTC	3.0	1.46	1.0	N	

Results per page: 50 ▾ 1 – 50 of 14851920 | < < > >|

Deployments / GcsToBQ

Description	Runs	Parameters
1 Flow run	All run states	Newest to oldest
etl-gcs-to-bq > olive-pheasant	Completed 2023/02/06 07:35:09 AM (1m 11s late) 1m 23s 4 task runs Deployment GcsToBQ Work Queue default	

Modified Script etl_gcp_to_bq.py:

```

from pathlib import Path
import pandas as pd
from prefect import flow, task

```

```

from prefect_gcp.cloud_storage import GcsBucket
from prefect_gcp import GcpCredentials

@task(retries=3)
def extract_from_gcs(color: str, year: int, month: int) -> Path:
    """Download trip data from GCS"""
    gcs_path = f"data/{color}/{color}_tripdata_{year}-{month:02}.parquet"
    gcs_block = GcsBucket.load("zoom-gcs")
    gcs_block.get_directory(from_path=gcs_path, local_path=f"../data/")
    return Path(f"../data/{gcs_path}")

@task(log_prints=True)
def write_bq(df: pd.DataFrame) -> None:
    """Write DataFrame to BigQuery"""
    gcp_credentials_block = GcpCredentials.load("zoom-gcs-creds")

    df.to_gbq(
        destination_table="dezoomcamp.rides",
        project_id="",
        credentials=gcp_credentials_block.get_credentials_from_service_account(),
        chunksize=500_000,
        if_exists="append"
    )
    print(f"post: rows: {len(df)}")

@flow(log_prints=True)
def etl_gcs_to_bq(
    months: list[int] = [2, 3], year: int = 2019, color: str = "yellow"
):
    """Main ETL flow to load data into Big Query"""

    total_length=0
    for month in months:
        path = extract_from_gcs(color, year, month)
        df = pd.read_parquet(path)
        total_length+=len(df)
        print(f"total_length: {total_length}")
        write_bq(df)

if __name__ == "__main__":
    months = [2, 3]
    year = 2019
    color = "yellow"
    etl_gcs_to_bq(months,year,color)

```

Question 4. Github Storage Block

Using the `web_to_gcs` script from the videos as a guide, you want to store your flow code in a GitHub repository for collaboration with your team. Prefect can look in the GitHub repo to find your flow code and read it. Create a GitHub storage block from the UI or in Python code and use that in your Deployment instead of storing your flow code locally or baking your flow code into a Docker image.

Note that you will have to push your code to GitHub, Prefect will not push it for you.

Run your deployment in a local subprocess (the default if you don't specify an infrastructure). Use the Green taxi data for the month of November 2020.

How many rows were processed by the script?

- 88,019
- 192,297
- **88,605**
- 190,225

Question 4. Github Storage Block Solution

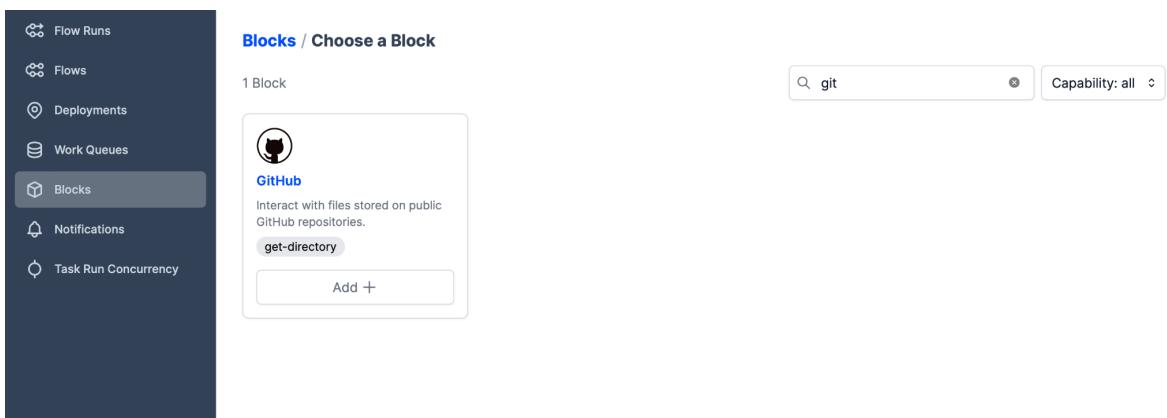
The correct answer is 88605. Change the flow from `etl_web_to_gcs.py` accordingly

Firstly, data from green taxi 2020 for the month of November (11)

```
@flow()
def etl_web_to_gcs() -> None:
    """The main ETL function"""
    color = "green"
    year = 2020
    month = "11"
    dataset_file = f"{color}_tripdata_{year}-{month}"
    dataset_url = f"https://github.com/DataTalksClub/nyc-tlc-data/releases/download/{color}/{dataset_file}.csv.gz"

    df = fetch(dataset_url)
    df_clean = clean(df)
    path = write_local(df_clean, color, dataset_file)
    write_gcs(path)
```

Block in Prefect



Follow the instruction to create the github block

The screenshot shows the 'Blocks / Choose a Block / GitHub / Create' screen. On the left is a sidebar with options: Flow Runs, Flows, Deployments, Work Queues, **Blocks**, Notifications, and Task Run Concurrency. The 'Blocks' option is selected. The main area has fields for 'Block Name' (set to 'dezoomgithub'), 'Repository' (set to 'https://github.com/Georgios-Grigoriou/data-engineering-zoomcamp-homework/tree/main/week_2_prefect'), 'Reference (Optional)', 'Access Token (Optional)', and a 'Create' button. A GitHub icon and a brief description are also present.

[Blocks / dezoomgithub](#)

The screenshot shows the 'Blocks / dezoomgithub' screen. It includes a code snippet for loading the block:

```
from prefect.filesystems import GitHub  
github_block = GitHub.load("dezoomgithub")
```

. Below it are sections for 'Repository' (set to 'https://github.com/Georgios-Grigoriou/data-engineering-zoomcamp-homework/tree/main/week_2_prefect'), 'Reference' (set to 'None'), and 'Access Token' (set to 'None'). A GitHub icon and a brief description are also present.

Import GitHuv and modify the task accordingly

```
from prefect.filesystems import GitHub  
  
@task()  
def write_gcs(path: Path) -> None:  
    """Upload local parquet file to GCS"""  
    github_block = GitHub.load("dezoomgithub")  
    github_block.upload_from_path(from_path=path, to_path=path)  
    return
```

Process ran locally and show the result in the terminal and Prefect UI respectively

```

flows > 02_gcp > etl_web_to_gcs.py > gcs_block
blocks
data
green
green_tripdata_2020-11.parquet
yellow
exploration
flows
01_start
02_gcp
__pycache__
data
etl_gcs_to_bq.py
etl_web_to_gcs.py
03_deployments
__pycache__
docker_deploy.py
parameterized_flow.py
README.md
images
.gitignore
.prefectignore
docker-requirements.txt
Dockerfile
! etl_gcs_to_bq-deployment.yaml
! etl_parent_flow-deployment.yaml
! etl_web_to_gcs-deployment.yaml
LICENSE
README.md
requirements.txt

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL

```

PULocationID           int64
DOLocationID          int64
passenger_count        float64
trip_distance          float64
fare_amount             float64
extra                  float64
mta_tax                float64
tip_amount              float64
tolls_amount            float64
ehail_fee               float64
improvement_surcharge float64
total_amount            float64
payment_type            float64
trip_type               float64
congestion_surcharge   float64
dtype: object
08:02:25.999 | INFO  | Task run 'clean-b9fd7e03-0' - rows: 88605
08:02:26.034 | INFO  | Task run 'clean-b9fd7e03-0' - Finished in state Completed()
08:02:26.070 | INFO  | Flow run 'rainbow-jerboa' - Created task run 'write_local-f322d1be-0' for task 'write_local'
08:02:26.070 | INFO  | Flow run 'rainbow-jerboa' - Executing 'write_local-f322d1be-0' immediately...
08:02:26.505 | INFO  | Task run 'write_local-f322d1be-0' - Finished in state Completed()
08:02:26.546 | INFO  | Flow run 'rainbow-jerboa' - Created task run 'write_gcs-1145c921-0' for task 'write_gcs'
08:02:26.546 | INFO  | Flow run 'rainbow-jerboa' - Executing 'write_gcs-1145c921-0' immediately...
08:02:26.669 | INFO  | Task run 'write_gcs-1145c921-0' - Getting bucket 'prefect-de-zoomcampg'
08:02:26.936 | INFO  | Task run 'write_gcs-1145c921-0' - Uploading from PosixPath('data/green/green_tripdata_2020-11.parquet') to the bucket 'prefect-de-zoomcampg' path 'data/green/green_tripdata_2020-11.parquet'.
08:02:27.208 | INFO  | Task run 'write_gcs-1145c921-0' - Finished in state Completed()
08:02:27.260 | INFO  | Flow run 'rainbow-jerboa' - Finished in state Completed('All states completed.')

```

INFO	columns: VendorID	float64	08:02:25 AM
	lpep_pickup_datetime	datetime64[ns]	clean-b9fd7e03-0
	lpep_dropoff_datetime	datetime64[ns]	
	store_and_fwd_flag	object	
	RatecodeID	float64	
	PULocationID	int64	
	DOLocationID	int64	
	passenger_count	float64	
	trip_distance	float64	
	fare_amount	float64	
	extra	float64	
	mta_tax	float64	
	tip_amount	float64	
	tolls_amount	float64	
	ehail_fee	float64	
	improvement_surcharge	float64	
	total_amount	float64	
	payment_type	float64	
	trip_type	float64	
	congestion_surcharge	float64	
	dtype: object		
INFO	rows: 88605		08:02:25 AM
INFO	Finished in state Completed()		clean-b9fd7e03-0
INFO	Created task run 'write_local-f322d1be-0' for task 'write_local'		08:02:26 AM
INFO	Executing 'write_local-f322d1be-0' immediately...		clean-b9fd7e03-0
INFO	Finished in state Completed()		08:02:26 AM
INFO	Created task run 'write_gcs-1145c921-0' for task 'write_gcs'		write_local-f322d1be-0
INFO	Executing 'write_gcs-1145c921-0' immediately...		08:02:26 AM
INFO	Getting bucket 'prefect-de-zoomcampg'.		08:02:26 AM
INFO	Uploading from PosixPath('data/green/green_tripdata_2020-11.parquet') to the bucket		write_gcs-1145c921-0

Push on github can be found [here](#)

Question 5. Email or Slack notifications

Q5. It's often helpful to be notified when something with your dataflow doesn't work as planned. Choose one of the options below for creating email or slack notifications.

The hosted Prefect Cloud lets you avoid running your own server and has Automations that allow you to get notifications when certain events occur or don't occur.

Create a free forever Prefect Cloud account at app.prefect.cloud and connect your workspace to it following the steps in the UI when you sign up.

Set up an Automation that will send yourself an email when a flow run completes. Run the deployment used in Q4 for the Green taxi data for April 2019. Check your email to see the notification.

Alternatively, use a Prefect Cloud Automation or a self-hosted Orion server Notification to get notifications in a Slack workspace via an incoming webhook.

Join my temporary Slack workspace with [this link](#). 400 people can use this link and it expires in 90 days.

In the Prefect Cloud UI create an [Automation](#) or in the Prefect Orion UI create a [Notification](#) to send a Slack message when a flow run enters a Completed state. Here is the Webhook URL to use:

<https://hooks.slack.com/services/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>

Test the functionality.

Alternatively, you can grab the webhook URL from your own Slack workspace and Slack App that you create.

How many rows were processed by the script?

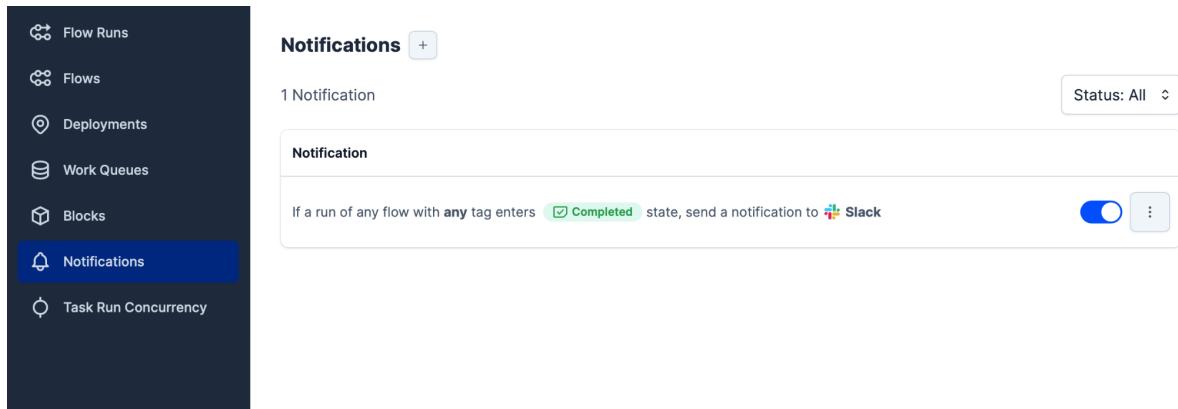
- 125,268
- 377,922
- 728,390
- **514,392**

Question 5. Email or Slack notifications Solution

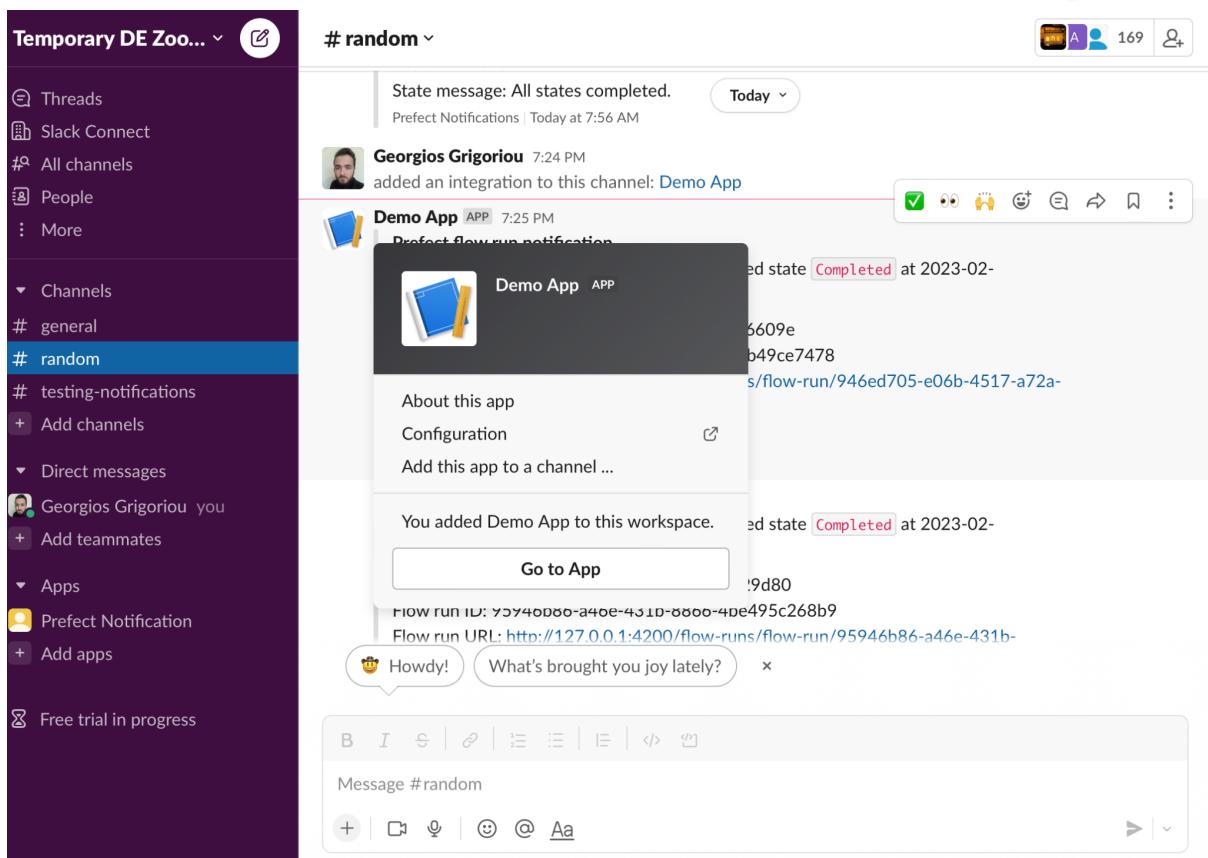
The correct answer is 514,392.

Steps:

Create Slack Notification



Using the Incoming Webhooks Slack API, I created a custom app and a webhook url following the instructions <https://api.slack.com/messaging/webhooks>.





Georgios Grigoriou 7:24 PM

added an integration to this channel: [Demo App](#)

Today ▾



Demo App APP 7:25 PM

Prefect flow run notification

Flow run etl-web-to-gcs/precious-jackal entered state [Completed](#) at 2023-02-06T19:25:49.116934+00:00.

Flow ID: 464fe2d2-97c6-409e-8a9e-f61bdc86609e

Flow run ID: 946ed705-e06b-4517-a72a-783b49ce7478

Flow run URL: <http://127.0.0.1:4200/flow-runs/flow-run/946ed705-e06b-4517-a72a-783b49ce7478>

State message: All states completed.

Prefect Notifications | Today at 7:25 PM

Prefect flow run notification

Flow run etl-parent-flow/amigurumi-lori entered state [Completed](#) at 2023-02-06T19:25:49.150829+00:00.

Flow ID: a13f3aa1-efcf-4074-9075-836308c29d80

Flow run ID: 95946b86-a46e-431b-8866-4be495c268b9

Flow run URL: <http://127.0.0.1:4200/flow-runs/flow-run/95946b86-a46e-431b-8866-4be495c268b9>

By running the flow etl_web_to_gcs for April 2019 green, I found that the processed rows are 514,392

```
> blocks
  data
    green
      green_tripdata_2019-04.parquet
      green_tripdata_2020-11.parquet
    yellow
  exploration
  flows
    01_start
      ingest_data_flow.py A
      ingest_data.py A
    README.md A
    02_gcp
      __pycache__
      data
        etl_gcs_to_bq.py A
        etl_web_to_gcs.py 3, M
        etl_web_to_github.py 4, U
    03_deployments
      __pycache__
      docker_deploy.py A
    parameterized_flow.py 4, M
    README.md M
  images
  .gitignore A
  .prefectignore A
  docker-requirements.txt A
  Dockerfile A
! etl_gcs_to_bq-deployment.yaml
! etl_parent_flow-deployment.yaml
! etl_web_to_gcs-deployment.yaml
LICENSE A
README.md A
requirements.txt A
OUTLINE
```

PROBLEMS	11	OUTPUT	DEBUG CONSOLE	TERMINAL	COMMENTS
tolls_amount		float64			
ehail_fee		float64			
improvement_surcharge		float64			
total_amount		float64			
payment_type		int64			
trip_type		int64			
congestion_surcharge		float64			
dtype: object					
19:31:58.878	INFO	Task run 'clean-0' - rows: 514392			
19:31:58.934	INFO	Task run 'clean-0' - Finished in state Completed()			
19:31:58.981	INFO	Flow run 'eggplant-bean' - Created task run 'write_local-0' for task 'write_local'			
19:31:58.981	INFO	Flow run 'eggplant-bean' - Executing 'write_local-0' immediately...			
19:32:00.481	INFO	Task run 'write_local-0' - Finished in state Completed()			
19:32:00.523	INFO	Flow run 'eggplant-bean' - Created task run 'write_gcs-0' for task 'write_gcs'			
19:32:00.524	INFO	Flow run 'eggplant-bean' - Executing 'write_gcs-0' immediately...			
19:32:00.659	INFO	Task run 'write_gcs-0' - Getting bucket 'prefect-de-zoomcampg'.			
19:32:00.889	INFO	Task run 'write_gcs-0' - Uploading from PosixPath('data/green/green_tripdata_2019-04.parquet') to the bucket 'prefect-de-zoomcampg'.			
19:32:01.389	INFO	Task run 'write_gcs-0' - Finished in state Completed()			
19:32:01.439	INFO	Flow run 'eggplant-bean' - Finished in state Completed('All states completed.')			
19:32:01.476	INFO	Flow run 'sepia-millipede' - Finished in state Completed('All states completed.')			

Question 6. Secrets

Prefect Secret blocks provide secure, encrypted storage in the database and obfuscation in the UI. Create a secret block in the UI that stores a fake 10-digit password to connect to a third-party service. Once you've created your block in the UI, how many characters are shown as asterisks (*) on the next page of the UI?

- 5
- 6
- **8**
- 10

Question 6. Secrets Solution

The correct answer is 8.

On Prefect Cloud, I searched on the secret block

The screenshot shows the Prefect Cloud interface. On the left, there's a sidebar with navigation links: Flow Runs, Flows, Deployments, Work Queues, **Blocks** (which is selected), Automations, and Task Run Concurrency. The main area is titled "Blocks / Choose a Block". It says "If you don't see a block for the service you're using, check out our Collections Catalog" and shows a search bar with "Sec". Below the search bar, it says "1 Block" and shows a card for the "Secret" block. The "Secret" block card has a key icon, the title "Secret", a description about representing a secret value, and a "Add +" button. In the bottom right corner of the main area, there's a smaller preview window titled "Blocks / desecret" showing code snippets and a preview of the secret value represented by asterisks.

We can see that the asterisks are 8.

Submitting the solutions

- Form for submitting: <https://forms.gle/PY8mBEGXJ1RvmTM97>
- You can submit your homework multiple times. In this case, only the last submission will be used.