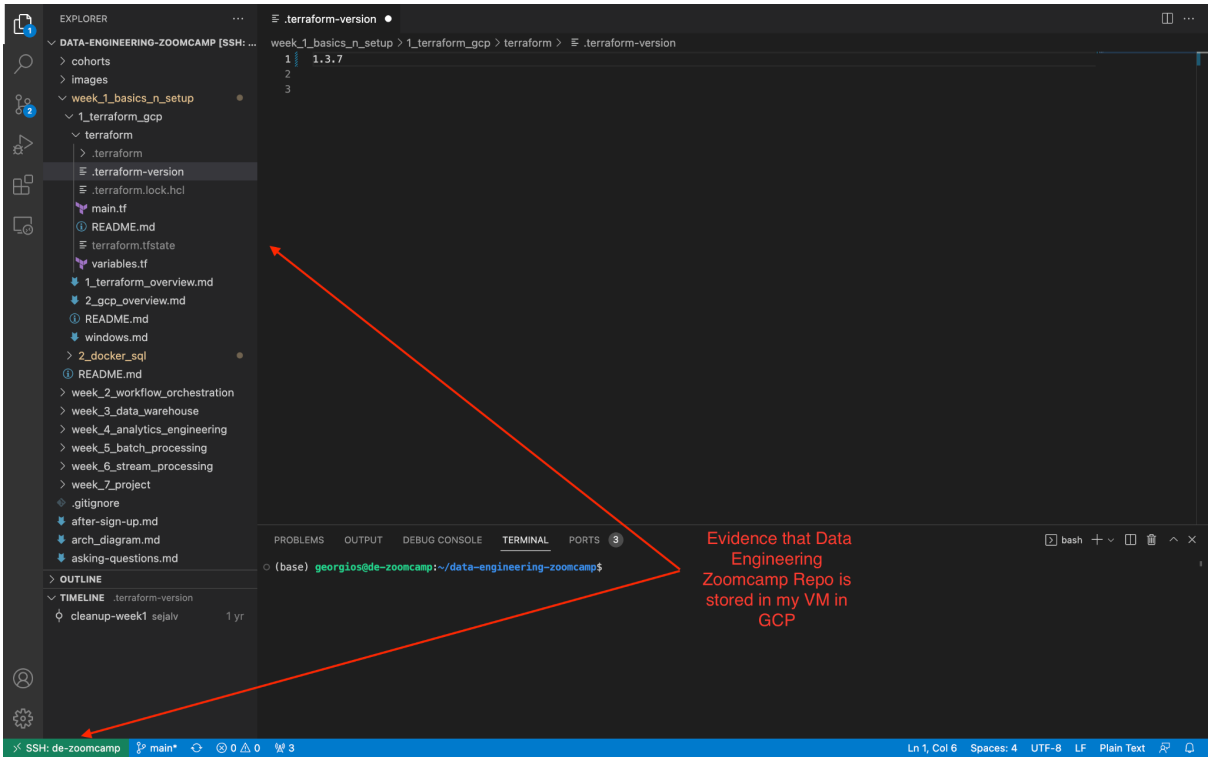
Week 1 Homework: GCP & Terraform

Georgios Grigoriou
georgiosvgrigoriou@gmail.com
Deadline: 30 January (Thursday), 22:00 CET

In this homework I prepared the environment by creating resources in GCP with Terraform as part of the Data Engineering Zoomcamp course.

In my VM on GCP, I installed Terraform, as evidenced in the image below

I copied the files from the course repo to my VM, as evidenced in image below



Question 1. Creating Resources

Modify the files as necessary to create a GCP Bucket and Big Query Dataset.After updating the main.tf and variable.tf files run:

```
terraform apply
```

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

Solutions' headings are in italic and highlighted in yellow under each Question's instruction.

Solution

Below is the commands that I ran on the terminal to direct to terraform in VM in GCP and authenticate

```
(base) georgios@de-zoomcamp:~$ ls
Anaconda3-2022.10-Linux-x86_64.sh  anaconda3  bin  data-engineering-zoomcamp  snap
(base) georgios@de-zoomcamp:~$ cd data-engineering-zoomcamp
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp$ ls
README.md         arch_diagram.md      cohorts     images                week_2_workflow_orchestration  week_4_analytics_engineering  week_6_stream_processing
after-sign-up.md  asking-questions.md  dataset.md  week_1_basics_n_setup  week_3_data_warehouse          week_5_batch_processing       week_7_project
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp$ cd week_1_basics_n_setup
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup$ ls
1_terraform_gcp  2_docker_sql  README.md
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup$ cd 1_terraform_gcp
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp$ ls
1_terraform_overview.md  2_gcp_overview.md  README.md  terraform  windows.md
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp$ cd terraform
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform$ ls
README.md  main.tf  terraform.tfstate  terraform.tfstate.backup  variables.tf
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform$ export GOOGLE_APPLICATION_CREDENTIALS=~/.gc/ny-rides.json
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform$ gcloud auth activate-service-account --key-file $GOOGLE_APPLICATION_CREDENTIALS
Activated service account credentials for: [dtc-de-[redacted].iam.gserviceaccount.com]
(base) georgios@de-zoomcamp:~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform$
```

Commands in terminal of VM in GCP

(base) **georgios@de-zoomcamp**:~$ ls
Anaconda3-2022.10-Linux-x86_64.sh **anaconda3** **bin** **data-engineering-zoomcamp** **snap**
(base) **georgios@de-zoomcamp**:~$ cd data-engineering-zoomcamp
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp**$ ls
README.md arch_diagram.md **cohorts** **images** **week_2_workflow_orchestration** **week_4_analytics_engineering** **week_6_stream_processing**
after-sign-up.md asking-questions.md dataset.md **week_1_basics_n_setup** **week_3_data_warehouse** **week_5_batch_processing** **week_7_project**
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp**$ cd week_1_basics_n_setup
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp/week_1_basics_n_setup**$ ls
1_terraform_gcp **2_docker_sql** README.md
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp/week_1_basics_n_setup**$ cd 1_terraform_gcp
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp**$ ls
1_terraform_overview.md 2_gcp_overview.md README.md **terraform** windows.md
(base) **georgios@de-zoomcamp**:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp**$ cd terraform
(base)
georgios@de-zoomcamp:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform**$ ls
README.md main.tf terraform.tfstate terraform.tfstate.backup variables.tf
(base)
georgios@de-zoomcamp:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform**$ export GOOGLE_APPLICATION_CREDENTIALS=~/.gc/ny-rides.json
(base)
georgios@de-zoomcamp:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform**$ gcloud auth activate-service-account --key-file $GOOGLE_APPLICATION_CREDENTIALS
Activated service account credentials for: [dtc-de-user@<project-id>.iam.gserviceaccount.com]

terraform init & terraform apply

Please find below the commands on the terminal. Project ID was replaced by <project id> for confidentiality purposes.

(base)
georgios@de-zoomcamp:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform**$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/google from the dependency lock file
- Using previously-installed hashicorp/google v4.50.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
(base)
georgios@de-zoomcamp:**~/data-engineering-zoomcamp/week_1_basics_n_setup/1_terraform_gcp/terraform**$ terraform apply
var.project
Your GCP Project ID

Enter a value: <project-id>

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.dataset will be created
+ resource "google_bigquery_dataset" "dataset" {
    + creation_time              = (known after apply)
    + dataset_id                 = "trips_data_all"
    + delete_contents_on_destroy = false
    + etag                       = (known after apply)
    + id                         = (known after apply)
    + labels                     = (known after apply)
    + last_modified_time         = (known after apply)
    + location                   = "europe-west6"
    + project                    = "<project-id>"
    + self_link                  = (known after apply)

    + access {
        + domain        = (known after apply)
        + group_by_email = (known after apply)
        + role          = (known after apply)
        + special_group  = (known after apply)
        + user_by_email  = (known after apply)

        + dataset {
            + target_types = (known after apply)

            + dataset {
                + dataset_id = (known after apply)
                + project_id = (known after apply)
              }
          }

        + routine {
            + dataset_id = (known after apply)
            + project_id = (known after apply)
            + routine_id = (known after apply)
          }

        + view {
            + dataset_id = (known after apply)
            + project_id = (known after apply)
            + table_id   = (known after apply)
          }
      }
  }

# google_storage_bucket.data-lake-bucket will be created
+ resource "google_storage_bucket" "data-lake-bucket" {
    + force_destroy               = true
    + id                          = (known after apply)
    + location                    = "EUROPE-WEST6"
    + name                        = "dtc_data_lake_<project-id>"
    + project                     = (known after apply)
    + public_access_prevention    = (known after apply)
    + self_link                   = (known after apply)
```

```
    + storage_class               = "STANDARD"
    + uniform_bucket_level_access = true
    + url                         = (known after apply)

    + lifecycle_rule {
        + action {
            + type = "Delete"
          }

        + condition {
            + age                   = 30
            + matches_prefix        = []
            + matches_storage_class = []
            + matches_suffix        = []
            + with_state            = (known after apply)
          }
      }

    + versioning {
        + enabled = true
      }

    + website {
        + main_page_suffix = (known after apply)
        + not_found_page   = (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.

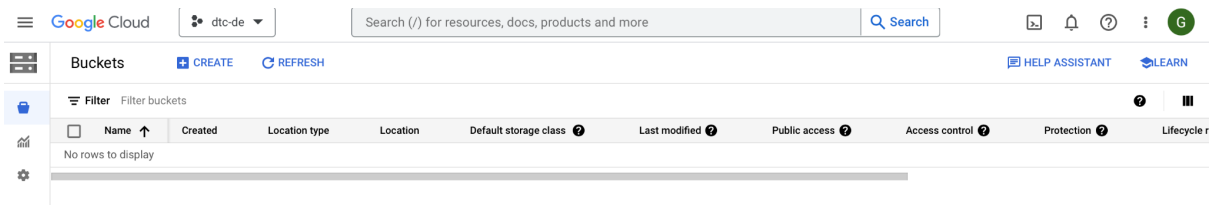
Enter a value: yes

google_bigquery_dataset.dataset: Creating...
google_storage_bucket.data-lake-bucket: Creating...
google_bigquery_dataset.dataset: Creation complete after 1s [id=projects/<project-id>/datasets/trips_data_all]
google_storage_bucket.data-lake-bucket: Creation complete after 1s [id=dtc_data_lake_<project-id>]

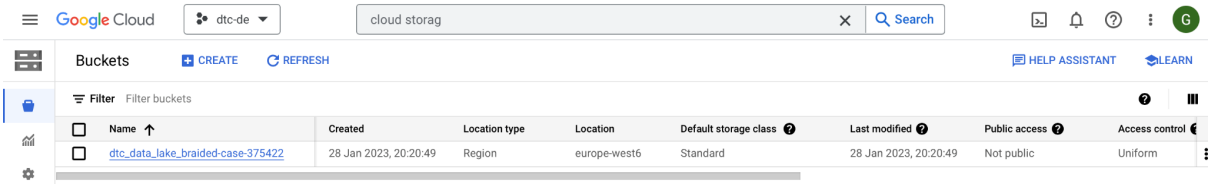
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Bucket and BigQuery before and after I run terraform apply

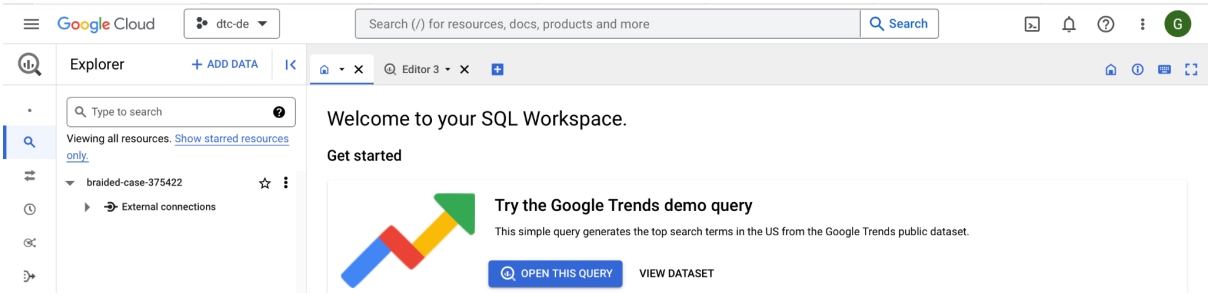
Before:



After:



Before:



After: