Data Engineering Bootcamp

First Delivery Report

Autor: Alejandra Elizabeth Moreno Morales

Task description

Based on the self-study material, recorded and live session, and mentorship covered until this deliverable, we suggest you perform the following:

Take as reference Terraform reference, identify and select the corresponding terraform blocks to build your own Airflow Cluster.

Airflow Cluster must be built with GKS in Google or EKS in AWS.

In case of some difficulties, take advantage of templates provided by Wizeline to build and start your Airflow Cluster.

Take your notes about any blocker and your lessons learned to be discussed during Q&A and Mentoring sessions.

Target

Implement configuration for run an Airflow cluster in a Google Kubernetes Services (GKS) of Google Cloud Platform (GCP) using Terraform as tool to declare the configuration files.

Prerequisites

- Terraform configuration

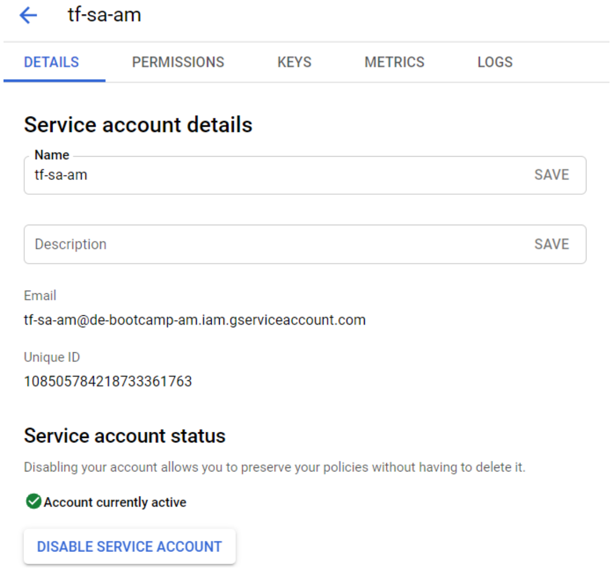
To run configuration files with the declaration of the infrastructure as code.

- GCP account

To deploy resources requested

Be aware of:

-Configure a Service account to be used on behalf of terraform



tf-sa-am

Service account details

Name
tf-sa-am

Description

Email
tf-sa-am@de-bootcamp-am.iam.gserviceaccount.com

Unique ID
108505784218733361763

Service account status

Disabling your account allows you to preserve your policies without having to delete it.

✓ Account currently active

-Enable Compute Engine API and Kubernetes Engine API in order to Terraform could work on this configuration.

- GCloud SDK tool configuration

To be used as an access for GCP using your user account credentials and therefore Terraform be allowed to provision resources on GCloud.

- Kubectl tool configuration

To be able to control Kubernetes clusters

- Helm3

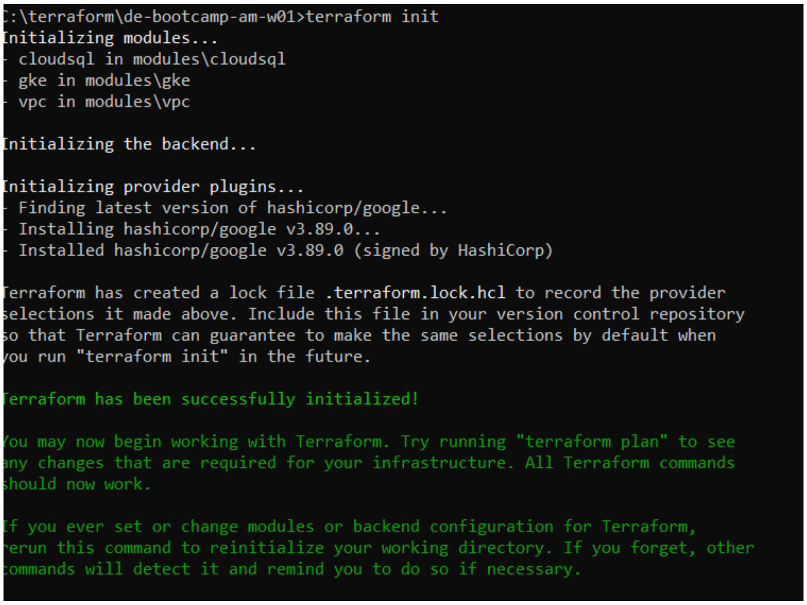
To manage the Kubernetes application.

Terraform configuration files

- main.tf. Declare provisions needed. A VPC and subnet will be created. Also deploy a 2-node separately managed node pool GKE cluster. And SQL resource running PostgreSQL 12.
- output.tf. Define outputs values after creation.
- provider.tf. Configures the specified provider to Terraform uses to create and manage your resources.
- terraform.tfvars. Template used to set the values for variables
- variables.tf. Declare the variables name, description and default value to be used in the project.

Implementation

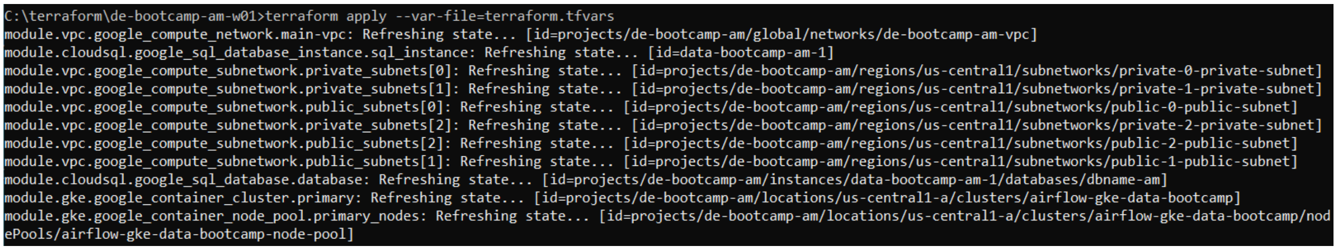
1. Create and customize the Terraform files to deploy the infrastructure needed. See Terraform configuration files
2. Initialize your Terraform workspace to install the plugins needed to manage the infrastructure



```
:\terraform\de-bootcamp-am-w01>terraform init
nitializing modules...
- cloudsql in modules\cloudsql
- gke in modules\gke
- vpc in modules\vpc

nitializing the backend...

nitializing provider plugins...
- Finding latest version of hashicorp/google...
- Installing hashicorp/google v3.89.0...
- Installed hashicorp/google v3.89.0 (signed by HashiCorp)

erraform has created a lock file .terraform.lock.hcl to record the provider
election it made above. Include this file in your version control repository
o that Terraform can guarantee to make the same selections by default when
ou run "terraform init" in the future.

erraform has been successfully initialized!

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

f you ever set or change modules or backend configuration for Terraform,
erun this command to reinitialize your working directory. If you forget, other
ommands will detect it and remind you to do so if necessary.
```

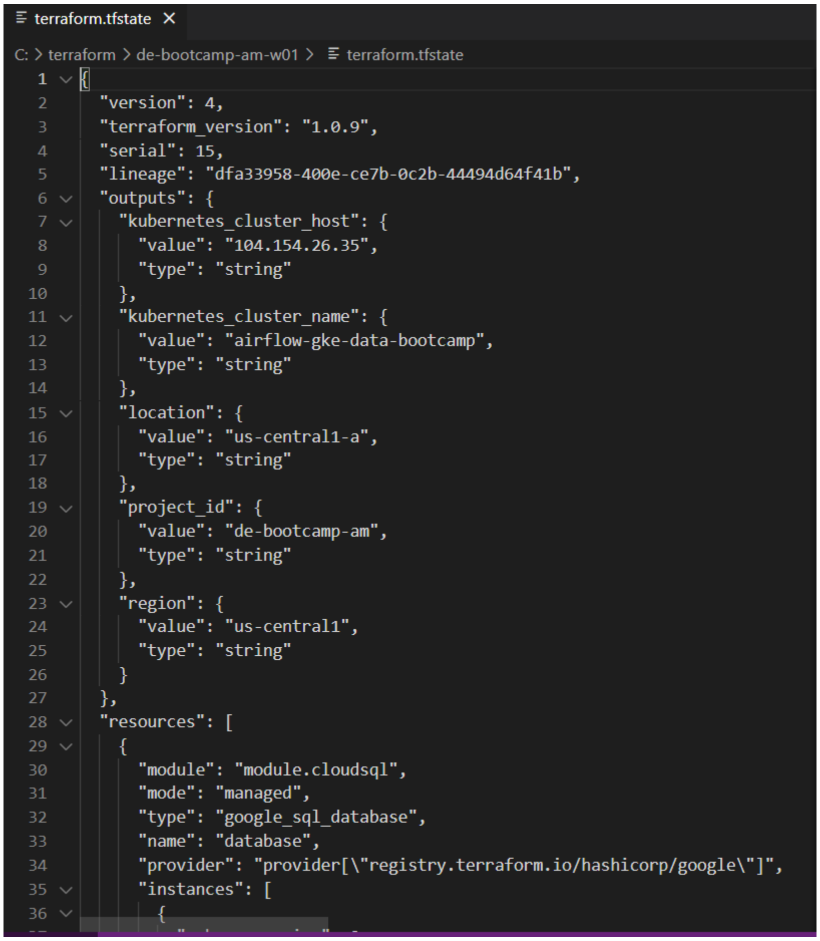
Create EKS cluster

3. Run Terraform apply and review the planned actions. Your terminal output should indicate the plan is running and what resources will be created. Confirm the apply.



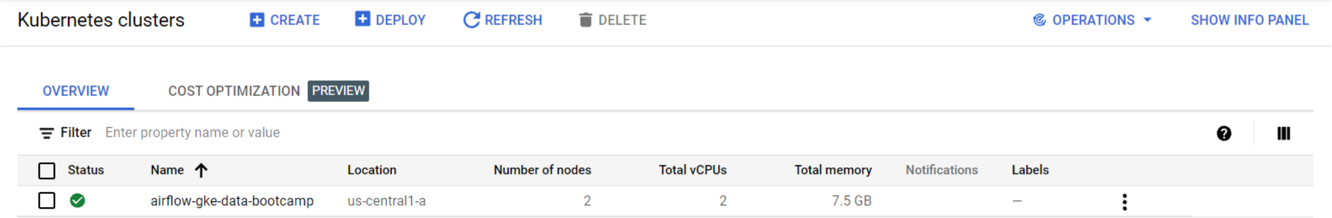
```
C:\terraform\de-bootcamp-am-w01>terraform apply --var-file=terraform.tfvars
module.vpc.google_compute_network.main-vpc: Refreshing state... [id=projects/de-bootcamp-am/global/networks/de-bootcamp-am-vpc]
module.cloudsql.google_sql_database_instance.sql_instance: Refreshing state... [id=data-bootcamp-am-1]
module.vpc.google_compute_subnetwork.private_subnets[0]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/private-0-private-subnet]
module.vpc.google_compute_subnetwork.private_subnets[1]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/private-1-private-subnet]
module.vpc.google_compute_subnetwork.public_subnets[0]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/public-0-public-subnet]
module.vpc.google_compute_subnetwork.private_subnets[2]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/private-2-private-subnet]
module.vpc.google_compute_subnetwork.public_subnets[2]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/public-2-public-subnet]
module.vpc.google_compute_subnetwork.public_subnets[1]: Refreshing state... [id=projects/de-bootcamp-am/regions/us-central1/subnetworks/public-1-public-subnet]
module.cloudsql.google_sql_database.database: Refreshing state... [id=projects/de-bootcamp-am/instances/data-bootcamp-am-1/databases/dbname-am]
module.gke.google_container_cluster.primary: Refreshing state... [id=projects/de-bootcamp-am/locations/us-central1-a/clusters/airflow-gke-data-bootcamp]
module.gke.google_container_node_pool.primary_nodes: Refreshing state... [id=projects/de-bootcamp-am/locations/us-central1-a/clusters/airflow-gke-data-bootcamp/nod
ePools/airflow-gke-data-bootcamp-node-pool]
```

4. Upon successful application, your terminal prints the outputs and save in terraform.tfstate

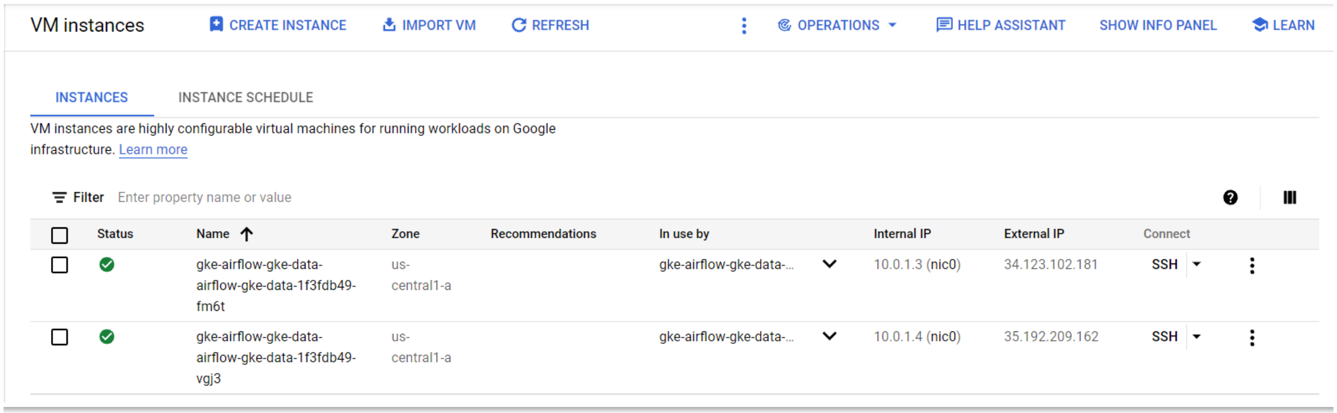


```
{
  "version": 4,
  "terraform_version": "1.0.9",
  "serial": 15,
  "lineage": "dfa33958-400e-ce7b-0c2b-44494d64f41b",
  "outputs": {
    "kubernetes_cluster_host": {
      "value": "104.154.26.35",
      "type": "string"
    },
    "kubernetes_cluster_name": {
      "value": "airflow-gke-data-bootcamp",
      "type": "string"
    },
    "location": {
      "value": "us-central1-a",
      "type": "string"
    },
    "project_id": {
      "value": "de-bootcamp-am",
      "type": "string"
    },
    "region": {
      "value": "us-central1",
      "type": "string"
    }
  },
  "resources": [
    {
      "module": "module.cloudsql",
      "mode": "managed",
      "type": "google_sql_database",
      "name": "database",
      "provider": "provider[\"registry.terraform.io/hashicorp/google\"]",
      "instances": [
        {
```

5. Review resources created in GCP console

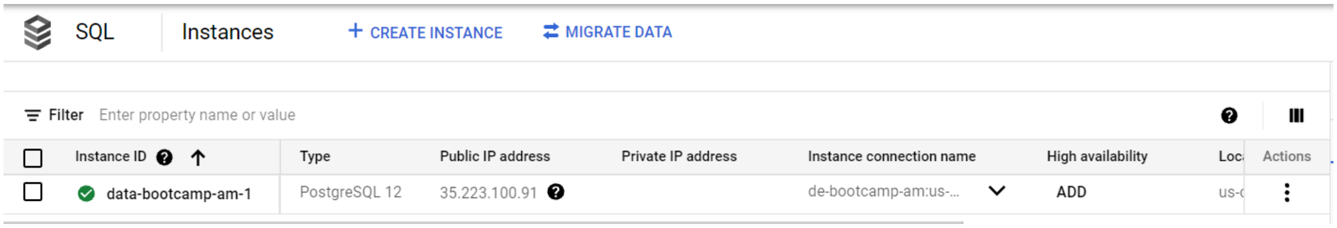


□	Status	Name ↑	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
□	✓	airflow-gke-data-bootcamp	us-central1-a	2	2	7.5 GB		—

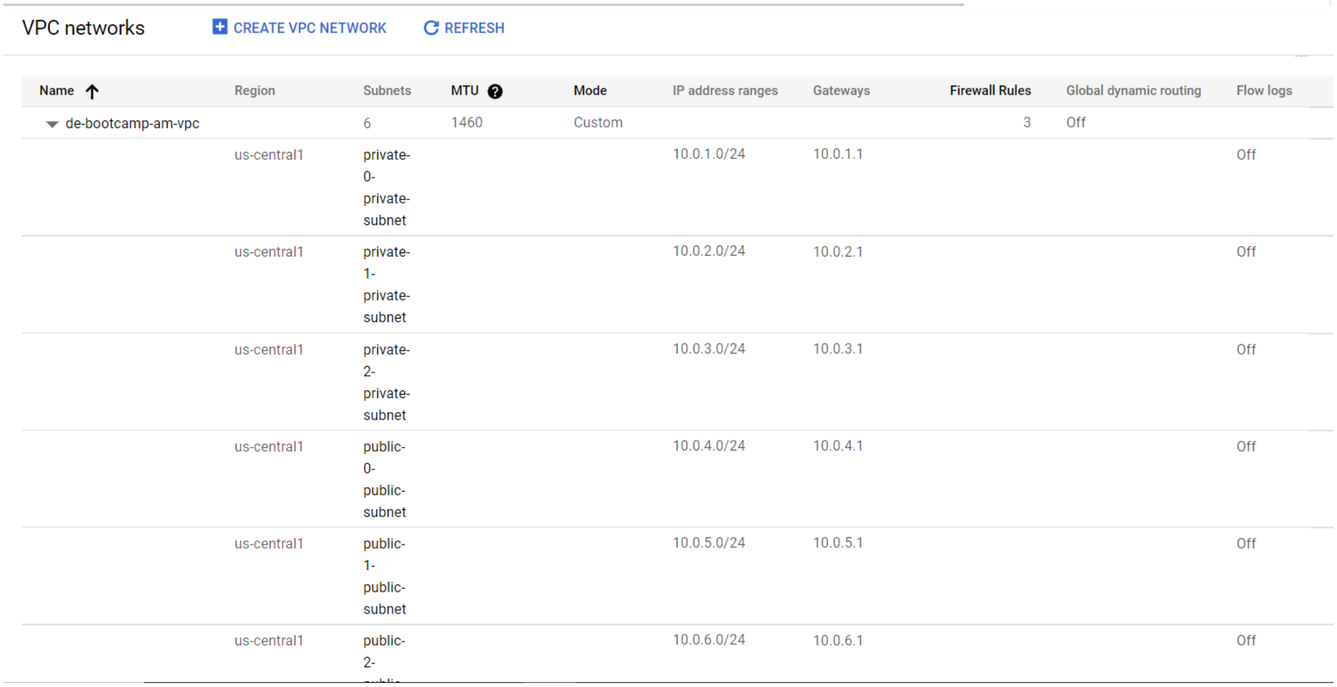


VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	gke-airflow-gke-data-airflow-gke-data-1f3fdb49-fm6t	us-central1-a		gke-airflow-gke-data-...	10.0.1.3 (nic0)	34.123.102.181	SSH
□	✓	gke-airflow-gke-data-airflow-gke-data-1f3fdb49-vgj3	us-central1-a		gke-airflow-gke-data-...	10.0.1.4 (nic0)	35.192.209.162	SSH

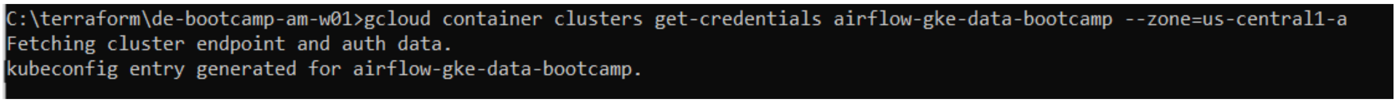


	Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Loc	Actions
□	✓ data-bootcamp-am-1	PostgreSQL 12	35.223.100.91		de-bootcamp-am:us-...	ADD	us-c	



Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global dynamic routing	Flow logs
de-bootcamp-am-vpc		6	1460	Custom			3	Off	
	us-central1	private-0-private-subnet			10.0.1.0/24	10.0.1.1			Off
	us-central1	private-1-private-subnet			10.0.2.0/24	10.0.2.1			Off
	us-central1	private-2-private-subnet			10.0.3.0/24	10.0.3.1			Off
	us-central1	public-0-public-subnet			10.0.4.0/24	10.0.4.1			Off
	us-central1	public-1-public-subnet			10.0.5.0/24	10.0.5.1			Off
	us-central1	public-2-			10.0.6.0/24	10.0.6.1			Off

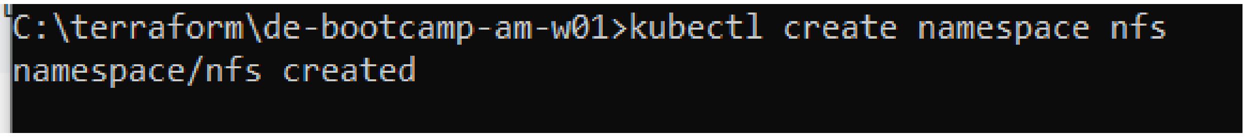
6. Once that the cluster is created, set the kubectl context



```
C:\terraform\de-bootcamp-am-w01>gcloud container clusters get-credentials airflow-gke-data-bootcamp --zone=us-central1-a
Fetching cluster endpoint and auth data.
kubeconfig entry generated for airflow-gke-data-bootcamp.
```

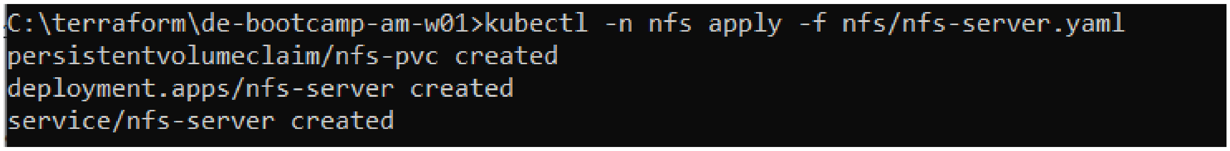
Create NFS Service

7. Create a namespace for the nsf service



```
C:\terraform\de-bootcamp-am-w01>kubectl create namespace nfs
namespace/nfs created
```

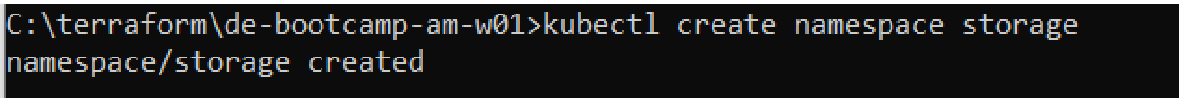
8. Create the nfs server



```
C:\terraform\de-bootcamp-am-w01>kubectl -n nfs apply -f nfs/nfs-server.yaml
persistentvolumeclaim/nfs-pvc created
deployment.apps/nfs-server created
service/nfs-server created
```

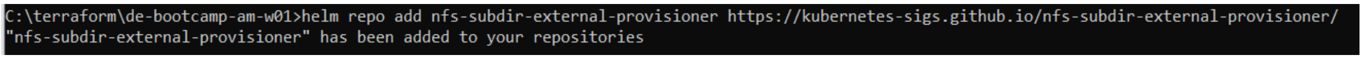
Create Storage

9. Create a namespace for storage deployment



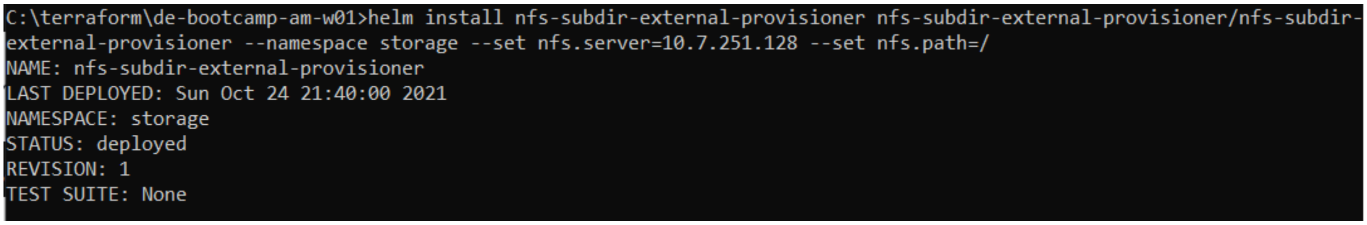
```
C:\terraform\de-bootcamp-am-w01>kubectl create namespace storage
namespace/storage created
```

10. Add the chart for the nfs-provisioner



```
C:\terraform\de-bootcamp-am-w01>helm repo add nfs-subdir-external-provisioner https://kubernetes-sigs.github.io/nfs-subdir-external-provisioner/
"nfs-subdir-external-provisioner" has been added to your repositories
```

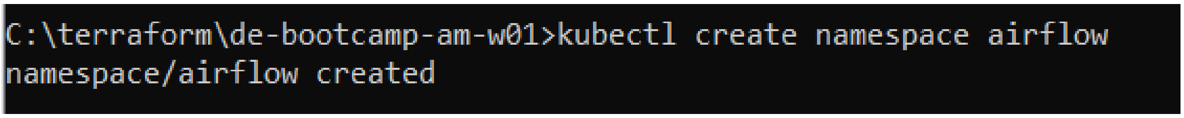
11. Install nfs-external-provisioner



```
C:\terraform\de-bootcamp-am-w01>helm install nfs-subdir-external-provisioner nfs-subdir-external-provisioner/nfs-subdir-
external-provisioner --namespace storage --set nfs.server=10.7.251.128 --set nfs.path=/
NAME: nfs-subdir-external-provisioner
LAST DEPLOYED: Sun Oct 24 21:40:00 2021
NAMESPACE: storage
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

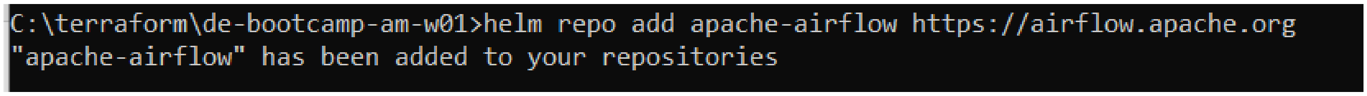
Create Airflow

12. Create a namespace for airflow deployment



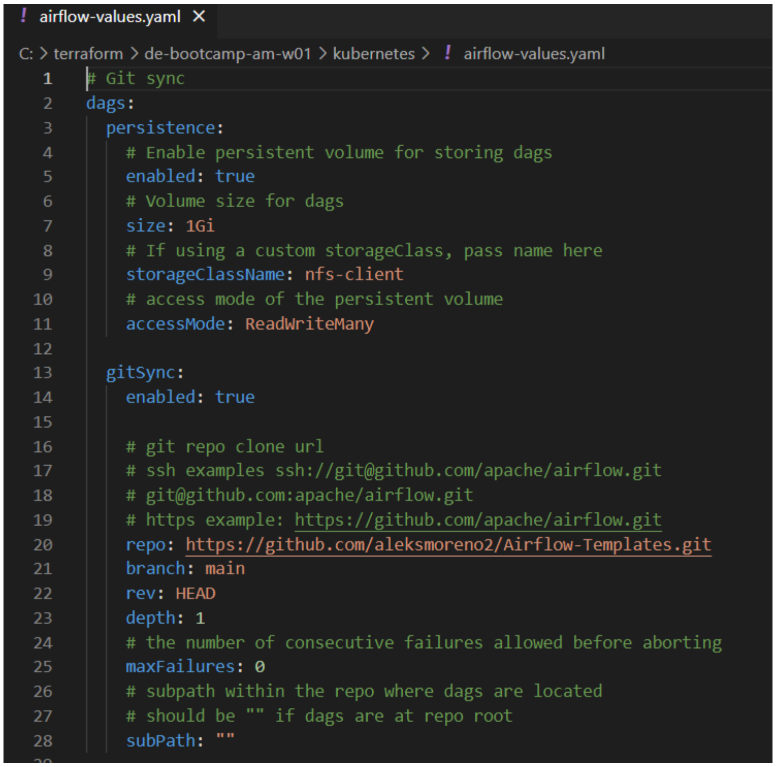
```
C:\terraform\de-bootcamp-am-w01>kubectl create namespace airflow
namespace/airflow created
```

13. Add the chart repository

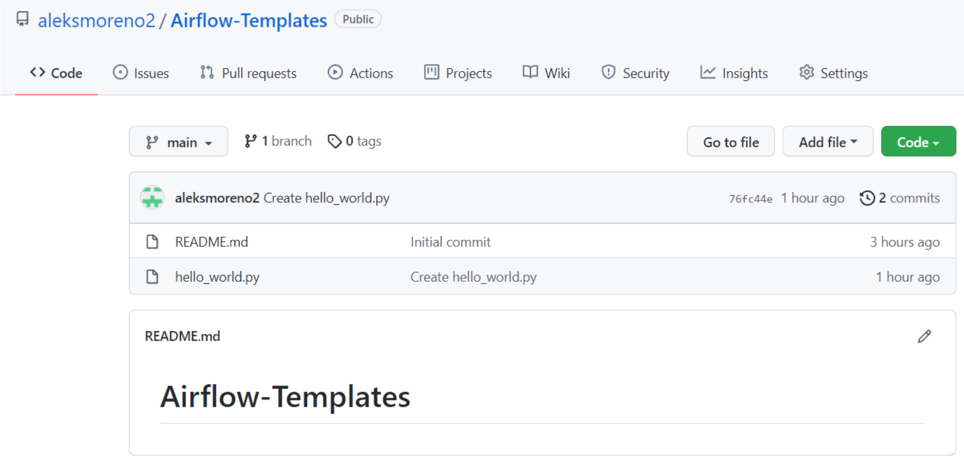


```
C:\terraform\de-bootcamp-am-w01>helm repo add apache-airflow https://airflow.apache.org
"apache-airflow" has been added to your repositories
```

14. Update airflow-values.yaml file with the project values


airflow-values.yaml ×
C: > terraform > de-bootcamp-am-w01 > kubernetes > airflow-values.yaml

```yaml
# Git sync
dags:
  persistence:
    # Enable persistent volume for storing dags
    enabled: true
    # Volume size for dags
    size: 1Gi
    # If using a custom storageClass, pass name here
    storageClassName: nfs-client
    # access mode of the persistent volume
    accessMode: ReadWriteMany

  gitSync:
    enabled: true

    # git repo clone url
    # ssh examples ssh://git@github.com/apache/airflow.git
    # git@github.com:apache/airflow.git
    # https example: https://github.com/apache/airflow.git
    repo: https://github.com/aleksmoreno2/Airflow-Templates.git
    branch: main
    rev: HEAD
    depth: 1
    # the number of consecutive failures allowed before aborting
    maxFailures: 0
    # subpath within the repo where dags are located
    # should be "" if dags are at repo root
    subPath: ""
```



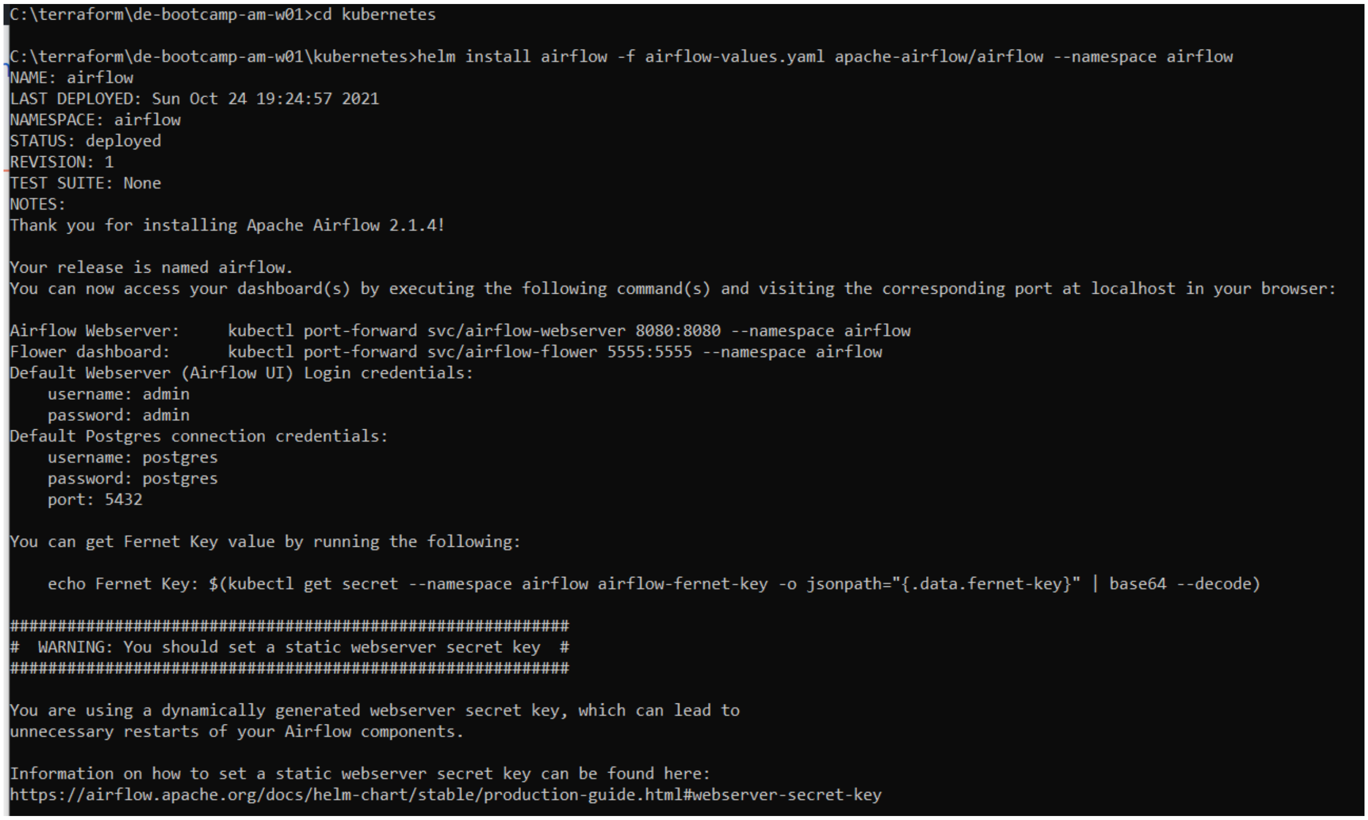
aleksmoreno2 Create hello_world.py 76fc44e 1 hour ago 2 commits

README.md	Initial commit	3 hours ago
hello_world.py	Create hello_world.py	1 hour ago

README.md

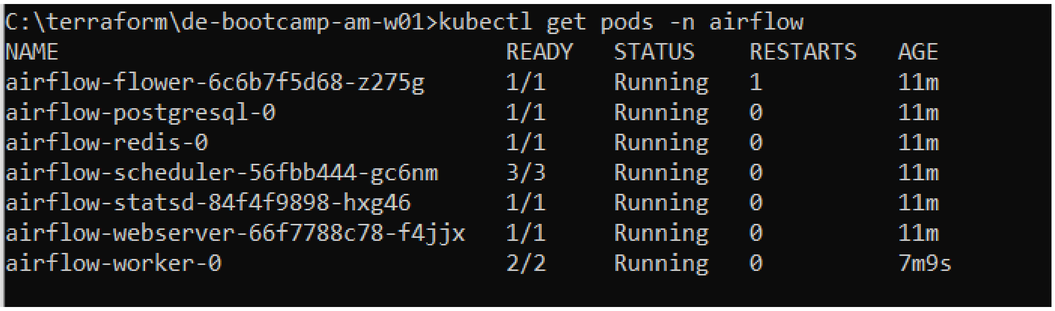
Airflow-Templates

15. Install the airflow chart from the repository



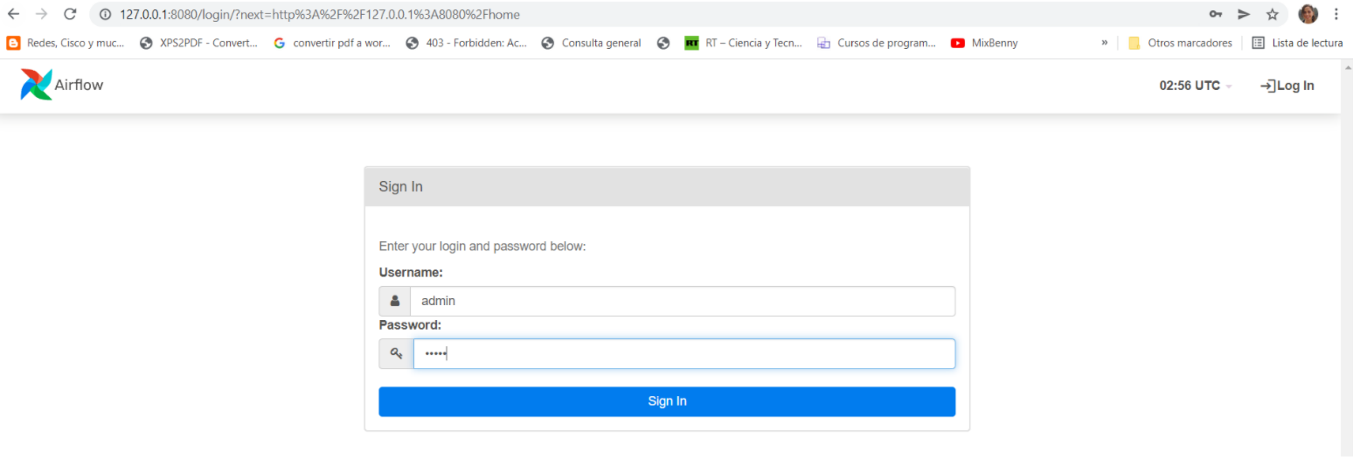
```
C:\terraform\de-bootcamp-am-w01>cd kubernetes

C:\terraform\de-bootcamp-am-w01\kubernetes>helm install airflow -f airflow-values.yaml apache-airflow/airflow --namespace airflow
NAME: airflow
LAST DEPLOYED: Sun Oct 24 19:24:57 2021
NAMESPACE: airflow
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing Apache Airflow 2.1.4!

Your release is named airflow.
You can now access your dashboard(s) by executing the following command(s) and visiting the corresponding port at localhost in your browser:

Airflow Webserver:     kubectl port-forward svc/airflow-webserver 8080:8080 --namespace airflow
Flower dashboard:      kubectl port-forward svc/airflow-flower 5555:5555 --namespace airflow
Default Webserver (Airflow UI) Login credentials:
    username: admin
    password: admin
Default Postgres connection credentials:
    username: postgres
    password: postgres
    port: 5432

You can get Fernet Key value by running the following:

    echo Fernet Key: $(kubectl get secret --namespace airflow airflow-fernet-key -o jsonpath="{.data.fernet-key}" | base64 --decode)

###########################################################
#  WARNING: You should set a static webserver secret key  #
###########################################################

You are using a dynamically generated webserver secret key, which can lead to
unnecessary restarts of your Airflow components.

Information on how to set a static webserver secret key can be found here:
https://airflow.apache.org/docs/helm-chart/stable/production-guide.html#webserver-secret-key
```

16. Verify that our pods are up and running



```
C:\terraform\de-bootcamp-am-w01>kubectl get pods -n airflow
NAME                                 READY   STATUS    RESTARTS   AGE
airflow-flower-6c6b7f5d68-z275g      1/1     Running   1          11m
airflow-postgresql-0                 1/1     Running   0          11m
airflow-redis-0                      1/1     Running   0          11m
airflow-scheduler-56fbb444-gc6nm     3/3     Running   0          11m
airflow-statsd-84f4f9898-hxg46       1/1     Running   0          11m
airflow-webserver-66f7788c78-f4jjx   1/1     Running   0          11m
airflow-worker-0                     2/2     Running   0          7m9s
```

17. Accessing to Airflow dashboard



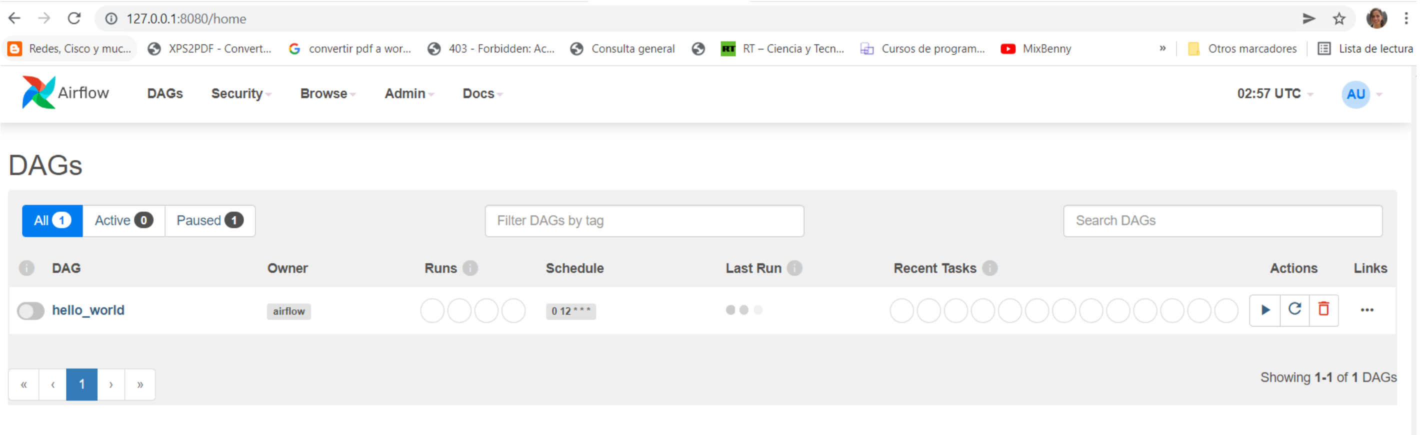
127.0.0.1:8080/login/?next=http%3A%2F%2F127.0.0.1%3A8080%2Fhome

Sign In

Enter your login and password below:

Username: admin

Password: •••••



DAGs

DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
hello_world	airflow		0 12 * * *				

Showing **1-1** of **1** DAGs