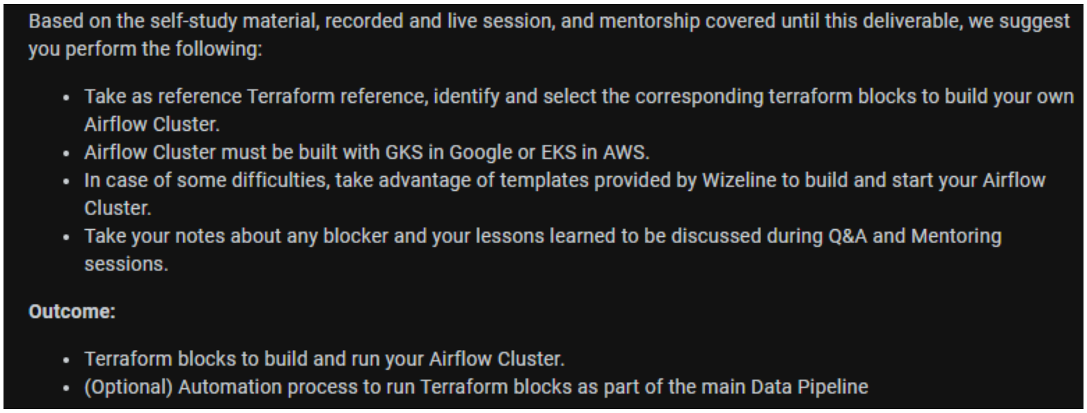
Geovanni V. 10/24/2021

Data Engineering Bootcamp Deliverable 1



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.

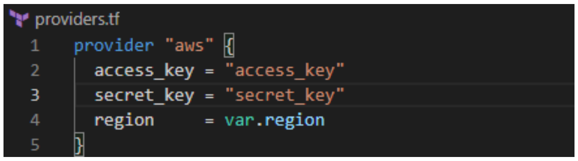
Outcome:

- Terraform blocks to build and run your Airflow Cluster.
- (Optional) Automation process to run Terraform blocks as part of the main Data Pipeline

For this deliverable I decided to use the provided templates for AWS (https://github.com/wizelineacademy/data-bootcamp-terraforms/tree/master/aws).

Preparation

I proceeded to install every tool needed which included: Terraform, Kubernetes, Helm and awscli. The templates needed me to add my aws credentials which I included in the providers.tf file.



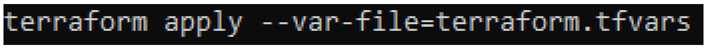
providers.tf

```
provider "aws" {
  access_key = "access_key"
  secret_key = "secret_key"
  region     = var.region
}
```

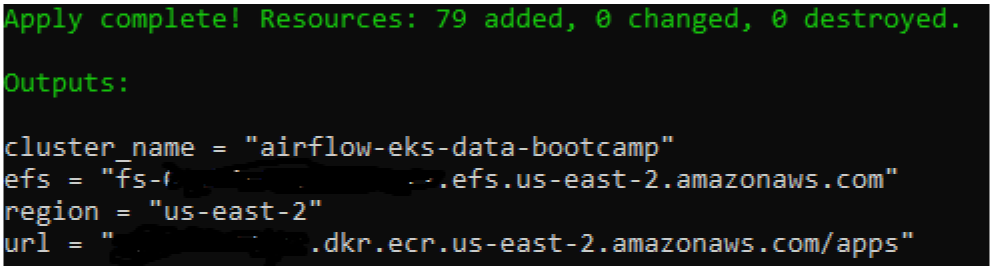
Process

I ran the following commands.

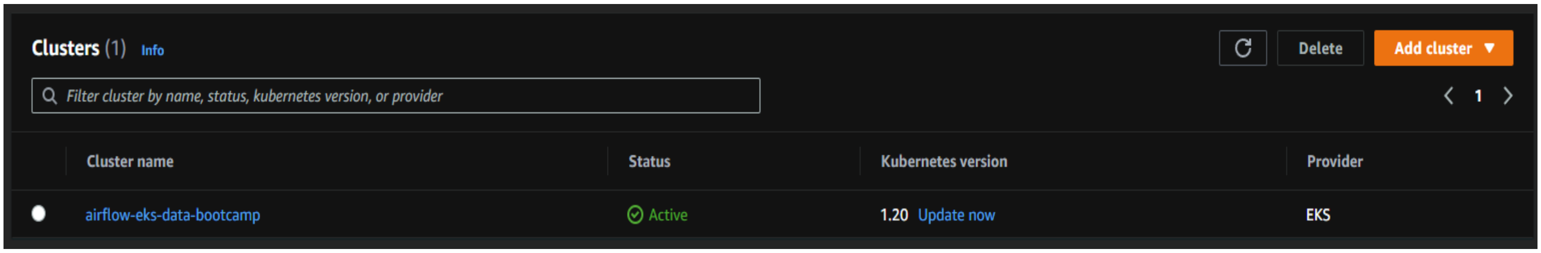
```
terraform init
```



```
terraform apply --var-file=terraform.tfvars
```

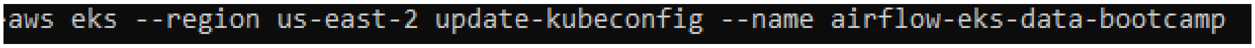


```
Apply complete! Resources: 79 added, 0 changed, 0 destroyed.

Outputs:

cluster_name = "airflow-eks-data-bootcamp"
efs = "fs-[redacted].efs.us-east-2.amazonaws.com"
region = "us-east-2"
url = "[redacted].dkr.ecr.us-east-2.amazonaws.com/apps"
```



Clusters (1) Info
Delete
Add cluster ▼
Filter cluster by name, status, kubernetes version, or provider
‹ 1 ›

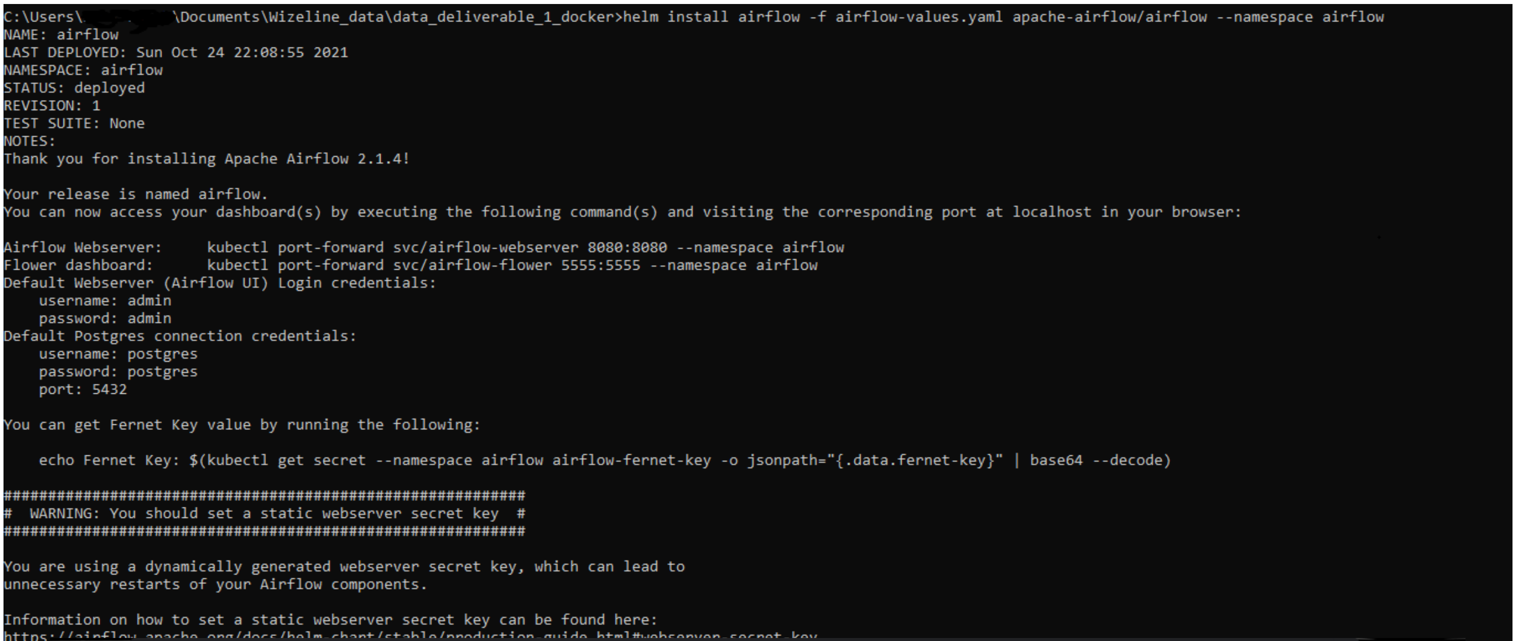
	Cluster name	Status	Kubernetes version	Provider
○	airflow-eks-data-bootcamp	✓ Active	1.20 Update now	EKS



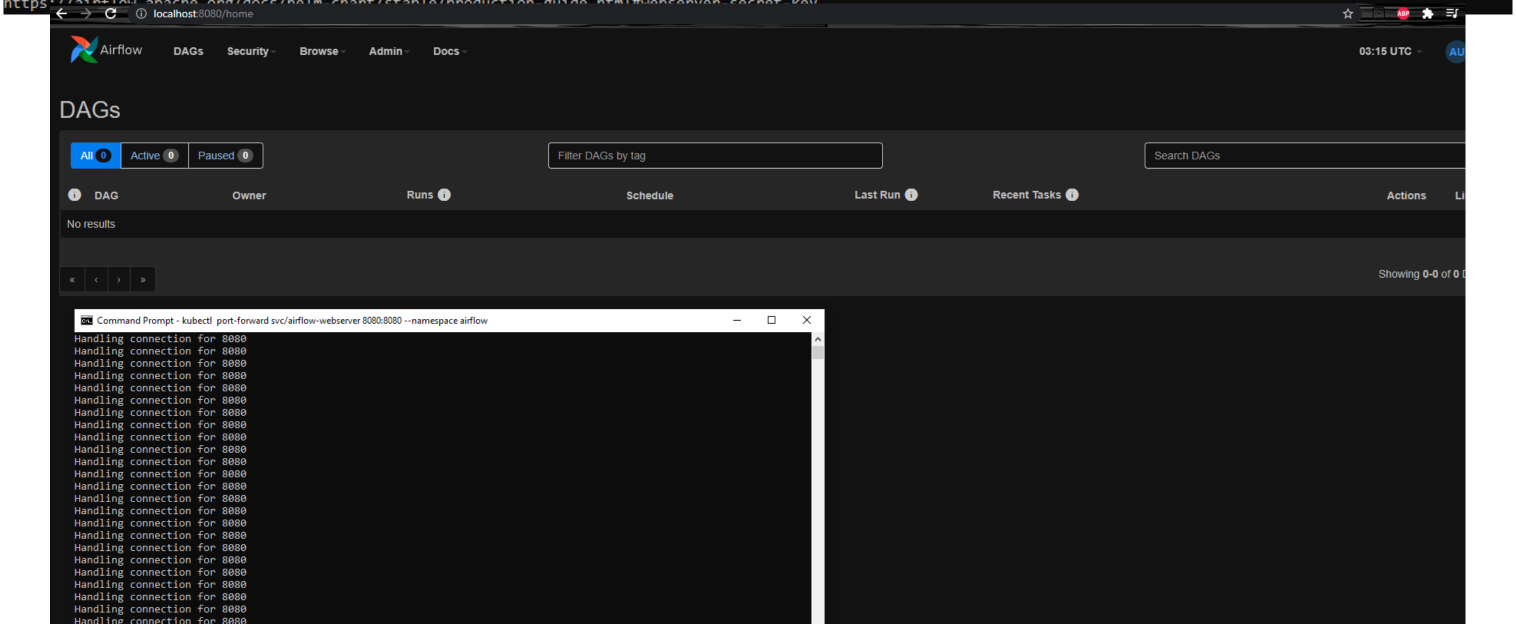
```
aws eks --region us-east-2 update-kubeconfig --name airflow-eks-data-bootcamp
```

Where on this last line the selected region is us-east-2 and the cluster name is airflow-eks-data-bootcamp.

After this I followed the steps in (https://github.com/wizelineacademy/data-bootcamp-terraforms/blob/master/kubernetes/README.md) to install Airflow on the EKS cluster. I could not find the yaml file described in the process so I decided to create my own from scratch which is included in the files.



```
C:\Users\[redacted]\Documents\Wizeline_data\data_deliverable_1_docker>helm install airflow -f airflow-values.yaml apache-airflow/airflow --namespace airflow
NAME: airflow
LAST DEPLOYED: Sun Oct 24 22:08:55 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
```



No results

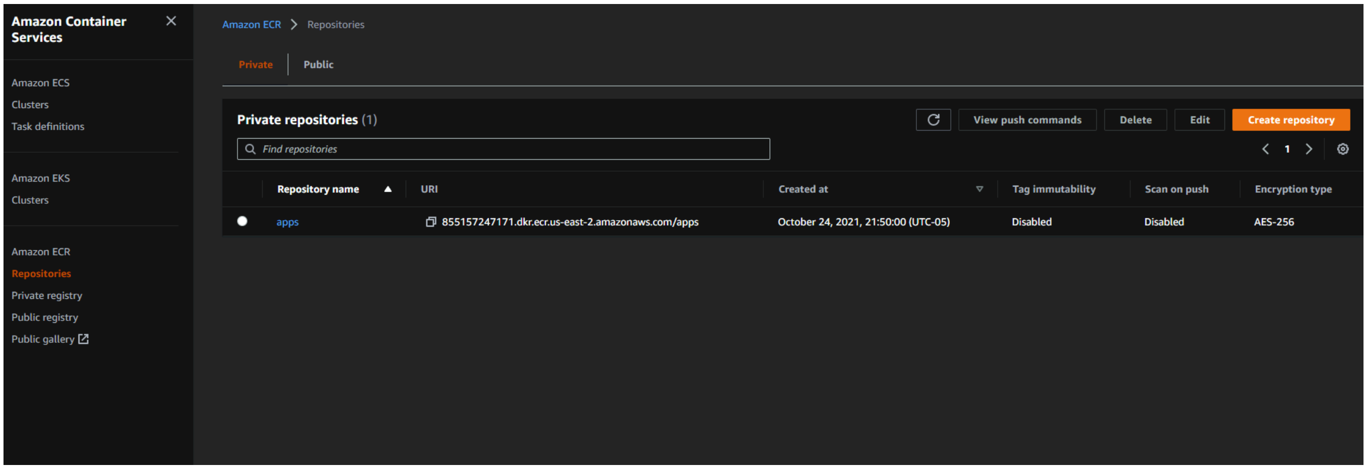
Command Prompt - kubectl port-forward svc/airflow-webserver 8080:8080 --namespace airflow

```
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```

As we can see the airflow cluster is up and running on AWS EKS.

Moving on to the task of creating a docker container on the cloud I searched on what could be done to do this. I found an AWS service called ECR which is a fully managed Docker container registry. Unfortunately, I ran out of time. The module needed to open a repository on that container is included in the code so that when the terraform code is run it will open a repo on ECR and build the Airflow cluster on EKS.

ECR Repository created.



Private repositories (1)

	Repository name	URI	Created at	Tag immutability	Scan on push	Encryption type
○	apps	855157247171.dkr.ecr.us-east-2.amazonaws.com/apps	October 24, 2021, 21:50:00 (UTC-05)	Disabled	Disabled	AES-256