Mediamarkt Hackathon

Register the Container generated by the DockerFile with Cloud Build / Artifacts.

In order to achieve this task it has been created a *cloudbuild.yaml* in the repository that uses docker to build the *Dockerfile* that is in the root of the repository. It tags the image properly and uploads it to the GCP Artifact Registry.

1. I have created the repository inside Artifact Registry, of type docker.
2. Then build and push the image by executing the following:

```
Unset
gcloud builds submit --config cloudbuild.yaml --region=global
```

There is also a *.gcloudignore* file that ignores the *.ssh*, *flux* and *terraform* directories.

Generation of the Docker Composer YAML

The docker-compose used here is pretty straightforward. It allows you to build, pull and serve the generated image. By using both keys *build* and *image* it builds the image and tags it properly.

The port section ensures that is only publishes for you local machine by binding it to *127.0.0.1*.

Creation of the Terraform Files

Inside the folder *terraform* there is a module that brings up a kubernetes cluster by making use of the *terraform-google-modules/kubernetes-engine/google//modules/safer-cluster* terraform module. This module also bootstraps the cluster with fluxcd that will help later on with the deployment of the resources.

In order for flux to synchronize with the repository it is necessary to generate a ssh key and grant access to it.

```
Unset
mkdir -p .ssh
ssh-keygen -t rsa -b 4096 -f .ssh/id_rsa
```

Apply the terraform module to generate all the resources.

Commands for the Deployment through TF files

Under the path *flux/cluster/default/mms-cloud-skeleton* there is a kustomization that will be synchronized with the flux controllers. And it should create a deployment with the project's image on the default namespace.

If a reconciliation is needed, issue the following command

```
Unset
flux reconcile ks flux-system
```

Note that the deployment is parametrized with the config in *flux/cluster/flux-system/cluster-settings.yaml*.

Solution of the IAM Role assignation

The finance team will have the rol *roles/billing.costsManager* as it allows to create, edit, and delete budgets, view billing account cost information and transactions, and manage the export of billing cost data to BigQuery. Does not confer the right to export *pricing* data or view *custom pricing* in the Pricing page. Also, does not allow the linking or unlinking of projects or otherwise managing the properties of the billing account.

And the DevOps team will be granted with the rol of *Kubernetes Engine Admin* as provides access to full management of clusters and their Kubernetes API objects.

To set a service account on nodes, you must also have the Service Account User role (roles/iam.serviceAccountUser) on the user-managed service account that your nodes will use.