

## Desafío de Ingeniería en la Nube de MediaMarkt

### Tabla de contenido

---

<b>INICIO DE DESAFIO .....</b>	<b>1</b>
<b>CREAR IMAGEN CON BUILD Y PUBLICAR EN ARTIFACTS .....</b>	<b>2</b>
<b>DOCKER COMPOSE .....</b>	<b>3</b>
<b>TERRAFORM.....</b>	<b>4</b>
<b>KUBERNETES.....</b>	<b>5</b>
<b>IAM QUESTION TO WRITE .....</b>	<b>7</b>

### Inicio de Desafío

---

Comenzamos con el correo de las credenciales y ID del proyecto

```
Good Morning Luis,  
  
You are receiving this email because you've joined a Nuwe Cloud Challenge that it's  
about to start.  
  
In order to complete the challenge you only need the following credentials:  
  
email: SvBn4TzV@train.nuwe.io  
password: q3S8gjdJyR  
projectId: aermymllrfng9baekdhnlglzunissm  
Then simply login to: https://console.cloud.google.com/  
  
And select nuwe.io as your organization in order to find your project.  
  
NUWE TEAM 🍷
```

Desde la consola iniciamos una variable de entorno para facilitar su uso. Seleccionamos la cuenta y por último hacemos login.

```
PROJECT_ID=aermym11rfng9baekdhnlglzunissm
gcloud config set account SvBn4TzV@train.nuwe.io
gcloud auth application-default login
```

Seleccionamos el proyecto desde la variable de entorno

```
gcloud config set project $PROJECT_ID
```

```
gcloud services enable cloudresourcemanager.googleapis.com
gcloud services enable artifactregistry.googleapis.com
gcloud services enable cloudbuild.googleapis.com
```

## Crear Imagen con Build y publicar en Artifacts

---

Creamos el Repositor en Artifacts

```
gcloud artifacts repositories create docker-repo \
--repository-format=docker \
--location=us \
--description="Docker repository"
```

Comprobamos su creación

```
gcloud artifacts repositories list
```

Listing items under project aermym11rfng9baekdhnlglzunissm, across all locations.

			ARTIFACT_REGISTRY			
REPOSITORY	FORMAT	MODE	DESCRIPTION	LOCATION	LABELS	ENCRYPTION
docker-repo	DOCKER	STANDARD_REPOSITORY	Docker repository	us		Google-managed key

Comenzamos con clonar el GIT

```
git clone https://github.com/nuwe-io/mms-cloud-skeleton.git
```

Accedemos al directorio

```
cd mms-cloud-skeleton
```

Una vez dentro del directorio mms-cloud-skeleton hacemos un clon de nuestro cloudbuild.yaml

```
git clone https://github.com/wexmaster/media-artifact.git
```

Construimos la imagen y publicamos en el Repositor de Google

```
PROJECT_ID=$PROJECT_ID gcloud builds submit --config=media-artifact/cloudbuild.yaml
```

Una vez creada la imagen comprobamos con el comando

```
gcloud container images list-tags us-docker.pkg.dev/$PROJECT_ID/docker-repo/mms-cloud-skeleton
```

Obtendremos la imagen creada

DIGEST	TAGS	TIMESTAMP
0dd4e08433ac	latest	2023-03-27T15:52:08

## Docker Compose

---

Clonamos nuestro archivo en github

```
git clone https://github.com/wexmaster/media-docker-compose.git
```

Accedemos al directorio

```
cd media-docker-compose
```

Una vez dentro del directorio lanzamos el comando

```
PROJECT_ID=$PROJECT_ID docker-compose up -d
```

La salida de este comando mostrara lo siguiente

```
[+] Running 2/2
  :: Network media-docker-compose_default          Created   0.0s
  :: Container media-docker-compose-webmediamarkt-1 Started
```

Si accedemos a <http://localhost:3000/> comprobamos que se muestra la web

## Welcome to the Nuwe's vite vue skeleton

Here you would find a Vue bootstrapped webapp where you could start coding without taking care about configuring so many things, and just focus on the code!

You could start modifying the `src/App.vue` file to see any change in the webapp.

Paramos Docker compose

```
PROJECT_ID=$PROJECT_ID docker-compose down
```

Salida del comando anterior

```
[+] Running 2/2
  :: Container media-docker-compose-webmediamarkt-1 Removed 0.1s
  :: Network media-docker-compose_default              Removed
```

## Terraform

Utilizaremos Terraform para crear la infraestructura necesaria ,lo que nos permitirá crear un cluster para kubernetes. Si se usa para otro proyecto se deben modificar los archivos `var.tf` y `provider.tf` ya que contiene información que debemos completar.

El archivo `provider.tf` en la línea 10 colocamos el name de BUCKET el cual lo obtenemos del comando: `gcloud storage buckets list`

El archivo `var.tf` contiene el ID del proyecto en la línea 3

```
default      = "aermym11rfng9baekdhnlglzunissm"
```

En la línea 20 tenemos que colocar delante de .svc.id.goog el ID del proyecto

```
default = "aermym11rfng9baekdhnlglzunissm.svc.id.goog"
```

Para el proyecto actual descargamos de GIT

```
git clone https://github.com/wexmaster/media-terraform.git
```

Accedemos al directorio

```
cd media-terraform
```

Ahora iniciamos terraform

```
terraform init
```

Planificamos con

```
terraform plan
```

Por último, aplicamos la configuración, este paso crea la infraestructura en la nube para el entorno de kubernetes.

```
terraform apply
```

Este proceso suele demorar por lo que recomiendo esperar para continuar con Kubernetes.

## Kubernetes

---

En la consola ejecutamos

```
gcloud iam service-accounts list
```

Buscamos el subfijo que tenga @developer.gserviceaccount.com

DISPLAY NAME	EMAIL	DISABLED
registry-gcp	private-regi.... kubernetes@.....	False
Compute Engine default service account	549146317650-compute@developer.gserviceaccount.com	False

Se utilizará para vincular la imagen que se aloja en Artifact-Registry

Registramos **us-docker.pkg.dev**

```
gcloud auth configure-docker us-docker.pkg.dev
```

Nos conectamos en gcloud para el uso de Kubernetes en nuestra consola:

```
gcloud container clusters get-credentials primary \
--zone us-central1-a \
--project aermymllrfng9baekdhnlglzunissm
```

Prueba Básica para comprobar que estamos conectados.  
**kubectl get nodes**

NAME	STATUS	ROLES	AGE	VERSION
gke-primary-general-113c8839-f641	Ready	<none>	48s	v1.24.9-gke.3200
gke-primary-general-34673a6c-mwk5	Ready	<none>	47s	v1.24.9-gke.3200

Es necesario crear el archivo key.json

```
gcloud iam service-accounts keys create key.json \
--iam-account 549146317650-compute@developer.gserviceaccount.com
```

Una vez realizado, creamos el secret para Kubernetes

```
kubectl create secret docker-registry artifact-registry \
--docker-server=https://us-docker.pkg.dev \
--docker-username=_json_key \
--docker-password="$(cat key.json)" \
--docker-email=549146317650-compute@developer.gserviceaccount.com
```

Descargamos de GIT

```
git clone https://github.com/wexmaster/media-deployment.git
```

Accedemos al directorio

```
cd media-deployment
```

Desplegamos

```
kubectl apply -f deployment.yaml
```

## Comprobamos los POD y Servicios

```
kubectl get all
```

## Ya podemos ver desplegado los servicios y POD

NAME	READY	STATUS	RESTARTS	AGE
pod/frontend-5468b6684f-87xkz	1/1	Running	0	43s
pod/frontend-5468b6684f-d9thx	1/1	Running	0	43s

  

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
AGE				
service/frontend	ClusterIP	10.52.0.213	<none>	3000/TCP
45s				
service/frontend-loadbalancer	LoadBalancer	10.52.15.88	35.226.81.24	3000:30221/TCP
44s				
service/kubernetes	ClusterIP	10.52.0.1	<none>	443/TCP
19m				

  

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/frontend	2/2	2	2	44s

Una vez finalizado, podemos acceder a la URL: <http://35.226.81.24:3000/>

## IAM QUESTION TO WRITE

---

Para asignar los roles apropiados en GCP para el equipo de DevOps y el equipo de Finanzas de MediaMarkt, primero debemos comprender sus responsabilidades y necesidades de acceso en GCP.

El equipo de DevOps es responsable de crear y administrar clústeres en Kubernetes en GCP. Por lo tanto, necesitan permisos para crear y administrar clústeres en GCP, así como para acceder a los recursos necesarios para ejecutar sus aplicaciones. El rol apropiado para el equipo de DevOps sería el de Kubernetes Engine Developer o Kubernetes Engine Admin, que les permitiría crear y administrar clústeres de Kubernetes en GCP.

Por otro lado, el equipo de Finanzas es responsable de la gestión de la facturación en GCP. Por lo tanto, necesitan permisos para ver y administrar la facturación y los costos de GCP. El rol

apropiado para el equipo de Finanzas sería el de Billing Account Administrator, que les permitiría administrar la facturación y los costos en GCP.

Para asignar estos roles en la consola IAM de GCP, siga estos pasos:

1. Inicie sesión en la consola de GCP y navegue hasta la sección IAM.
2. Haga clic en "Agregar" para agregar un nuevo usuario o grupo.
3. Seleccione el usuario o grupo para el que desea asignar el rol.
4. Seleccione el rol apropiado del menú desplegable de roles. Para el equipo de DevOps, seleccione el rol de Kubernetes Engine Developer o Kubernetes Engine Admin. Para el equipo de Finanzas, seleccione el rol de Billing Account Administrator.
5. Haga clic en "Agregar".

Ahora, el equipo de DevOps y el equipo de Finanzas tienen los roles apropiados para realizar sus tareas en GCP mientras se sigue el principio de menor privilegio en la asignación de roles.