MediaMarkt Cloud Engineering challenge

Cloud Build/Artifact to generate the container (150 points).

Para implementar esta parte he creado un archivo cloudbuild.yaml en el repositorio. Este archivo es el que tiene la pipeline, esta pipeline tiene dos steps, uno que construye la imagen y la tagea con el commit sha y latest. Si este paso se completa correctamente la pipeline se sube al repositorio de Artifactory. Para generar los recursos de GCP de Cloud Build y Cloud Artifact se ha utilizado terraform, los modulos de artifactory, cb_connection y cb_triggers.

Generation of the Docker Composer YAML (150 points).

La parte del docker compose, he entendido que tenia que ir directamente en el repositorio. He creado un docker-compose para poderlo construir la imagen docker directamente desde el repositorio, el docker-compose directamente construye la imagen a partir del Dockerfile.

Creation of the Terraform Files (400 points).

En la parte de terraform he querido que toda la infraestuctura necesaria para la creación de este proyecto este creada con terraform, incluido, vpcs, toda la parte de cloudbuid y artifact y eks.

La creación del proyecto de terraform se ha pensado en todo momento en crear un código escalable, para eso se han utilizado diferentes módulos:

apis:

Este modulo se utiliza para activar las APIS de GCP necesarias para los otros modulo, es un modulo dependencia de los demas. Para poder utilizar esto es necesario activar previamente “Cloud Resource Manager API” en la consola de GCP.

artifactory:

Este modulo se utiliza para crear el repositorio de docker donde subiremos nuestra imagen generada.

cb_connection:

Este modulo se utiliza para conectar nuestra cuenta de GCP con nuestros repositorios de github, es necesariamente tener un oauth token de github, este oauth token lo guardamos como secreto dentro de GCP, también es necesario tener la app de Cloud Build instalada en nuestro github y obtener el id de instalación.

cb_triggers:

Este modulo se utiliza para crear los triggers, unicamente tenemos uno pero se ha construido de forma que se puedan añadir varios repositorios en una lista y se puedan crear varios. Es necesario tener la conexión que creamos antes con nuestro github.

vpc:

En este modulo he creado la red principal y la subred para posteriormente utilizarla en la creación de nuestro cluster, así una regla del firewall para abrir el puerto 30000 puerto que utilizaremos en el servicio de NodePort que crearemos mas tarde.

gke:

En este modulo creamos un cluster de gke y exportamos el edpoint, el certificado y el token para posteriormente poder hacer la conexión con el provider de kubernetes.

k8s-app:

Este modulo es el que utilizamos para hacer del deploy vía terraform, por lo tanto lo explico en el siguiente punto.

Commands for the Deployment through TF files (kubectl) (200 points).

Para poder hacer el deploy se ha utilizado el deployment de kubernetes y se ha creado un modulo. La conexión al cluster la hacemos a partir del cluster que creamos en el modulo de gke. En el modulo se para el repositorio de artifact que lo recogemos de otro modulo, se le pasa la imagen y el tag de la imagen. Se crea un recurso de Deployment que contiene la imagen de artifact y se crea un servicio de NodePort con el fin de poder comprobar que podemos acceder al servicio.

Solution of the IAM Role assignation (300 points).

Para la solución de IAM asignaría los siguientes roles:

DevOps Team Roles:

roles/container.admin

Finance Team Roles:

roles/billing.admin

Para aplicar estas políticas en la consola de GCP lo haría si:

Crear dos grupos desde “Grupos”. Seria un grupo de DevOps y otro de Finance.

Ahora aplicaría estos roles a los grupos desde IAM. Y añadiría a los usuarios que pertenezcan a cada grupo.