

Kubernetes project using GKE and Google Cloud Functions

Business Overview

Google Cloud is a collection of physical assets, such as computers and hard disk drives, and virtual resources, such as virtual machines (VMs), housed in Google data centers worldwide. This resource distribution has various advantages, including redundancy in a failure and decreased latency by putting resources closer to customers. This release also presents some guidelines for combining resources.

GCP offers a web-based graphical user interface for managing Google Cloud projects and resources. If a user prefers to work at the command line, the G-Cloud command-line tool can handle most Google Cloud activities.

This is the fourth project in the GCP Roadmap project series. The [previous projects](#) utilize services such as PubSub, Compute Engine, Cloud Storage, BigQuery, and GCP Dataflow with Apache Beam. **This project will automate and deploy the dataflow service code from the previous project as a microservice template using Docker, Google Kubernetes Engine (GKE), and Google Cloud Functions.**

Tech Stack

→ Language: Python3

→ Libraries: FastAPI

→ Services: Cloud Storage, Dataflow, Apache Beam, BigQuery, G-Cloud SDK, Google Kubernetes Engine, Google Container Registry, Google Cloud Functions, kubectl, Docker

Cloud Storage

Cloud Storage is a service that allows users to store their data on the Google Cloud. An object is an immutable piece of data that consists of a file in any format. Objects can be stored in containers known as buckets. All buckets are related to a project, and the user may organize their projects into organizations. After starting a project, users may create Cloud Storage buckets, upload things to the buckets, and get objects. Users can also give rights to make data accessible to certain domains or for specific use cases such as establishing a website.

BigQuery

Google Bigquery is a Cloud Datawarehouse powered by Google, which is Serverless, highly scalable, and cost-effectively designed for making data-driven business decisions quickly. It offers both the batch and streaming insertion capabilities and is integrated with Tensorflow as well to perform machine learning using SQL-like dialects.

Apache Beam

Apache Beam is a batch and streaming data processing unified programming model. It offers many APIs for interacting with various data sources and processing data using various backends, such as Spark or Dataflow. As a result, the data may be stored elsewhere, and computation can be performed on it in a serverless manner or on a specified backend.

Dataflow

Google Cloud Dataflow is a cloud-based data processing service that can handle batch and real-time data streaming. It allows users to build processing pipelines for integrating, preparing, and analyzing massive data sets, which is typical of big data processing.

Google Kubernetes Engine

Google Kubernetes Engine (GKE) provides a controlled environment for deploying, maintaining, and scaling containerized applications on Google infrastructure. The GKE environment is made up of many Compute Engine instances joined together to form a cluster. The Kubernetes open-source cluster management technology powers GKE clusters and provide tools for interacting with your cluster. Kubernetes commands and resources are used to deploy and manage applications, execute administrative activities, define policies, and evaluate the health of your deployed workloads.

Cloud Functions

Cloud Functions is a serverless Function as a Service (FaaS) provided by Google to run your code in the cloud, supporting many programming languages. Functions naturally scale and are highly available and fault-tolerant. Cloud Functions are ideal for designing serverless backends, real-time data processing, and adaptive apps.

Key Takeaways

- Running Dataflow template created in dataflow series
- Creating microservice to run the dataflow job using a template
- Understanding Docker
- Installing Docker
- Creating a docker image for microservice
- Running microservice in docker container
- Overview of Kubernetes and GKE implementation of Kubernetes
- Installing and configuring kubectl
- Pushing image to google container registry
- Creating GKE cluster and namespace
- Deployment and service configs for GKE deployment
- Deploying microservice in GKE and running it.
- Cloud function to call microservice automatically when a new file arrives.
- Scaling the application in GKE