

PROFESSIONAL DATA ENGINEER - Google Cloud Platform

TITLE: "Introduction to Google Cloud - Google Kubernetes Engine"

Author:

- *Name: "Vignesh Sekar S"*
- *Designation: "Multi Cloud Architect"*
- *Tags: [Google Cloud, DataEngineer, Python, PySpark, SQL, BigData]*

Lab on Google Cloud - Kubernetes Engine (CLI)

1. Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure.
2. The Kubernetes Engine environment consists of multiple machines (specifically Compute Engine instances) grouped to form a container cluster.

Cluster orchestration with Google Kubernetes Engine

- Google Kubernetes Engine (GKE) clusters are powered by the Kubernetes -- an open source cluster management system.
- Kubernetes provides the mechanisms through which you interact with your container cluster. You use Kubernetes commands and resources to deploy and manage your applications, perform administrative tasks, set policies, and monitor the health of your deployed workloads.
- Kubernetes draws on the same design principles that run popular Google services and provides the same benefits:
 - a. automatic management
 - b. monitoring and liveness probes for application containers
 - c. automatic scaling
 - d. rolling updates and more.

- A GKE cluster consists of at least one cluster master machine and multiple worker machines called nodes. a. Nodes are Compute Engine virtual machine (VM) instances that run the Kubernetes processes necessary to make them part of the cluster.
- GKE uses Kubernetes objects to create and manage your cluster's resources. Kubernetes provides the Deployment object for deploying stateless applications like web servers.
- Service objects define rules and load balancing for accessing your application from the internet. Kubernetes on Google Cloud
- When you run a GKE cluster, you also gain the benefit of advanced cluster management features that Google Cloud provides. These include:
 - Load balancing for Compute Engine instances
 - Node pools to designate subsets of nodes within a cluster for additional flexibility
 - Automatic scaling of your cluster's node instance count
 - Automatic upgrades for your cluster's node software
 - Node auto-repair to maintain node health and availability
 - Logging and Monitoring with Cloud Monitoring for visibility into your cluster

Task 1. Set a default compute zone

- Set the default compute region * `gcloud config set compute/region us-east4`
- Set the default compute zone * `export ZONE=$(gcloud config set compute/zone us-east4-a) * export ZONE=$(gcloud config get-value compute/zone)`

Task 2. Create a GKE cluster

- Cluster names must start with a letter and end with an alphanumeric, and cannot be longer than 40 characters. * `gcloud container clusters create --machine-type=e2-medium --zone=$ZONE lab-cluster`

Task 3. Get authentication credentials for the cluster

- Authenticate with the cluster
 - `gcloud container clusters get-credentials lab-cluster`

Task 4. Deploy an application to the cluster

- You can now deploy a containerized application to the cluster. For this lab, you'll run hello-app in your cluster.
- To create a new Deployment hello-server from the hello-app container image, run the following kubectl create command:
 - `kubectl create deployment hello-server-gcp --image=gcr.io/google-samples/hello-app:1.0`
- NOTE This Kubernetes command creates a Deployment object that represents hello-server.

- In this case,
 - `--image` specifies a container image to deploy. The command pulls the example image from a Container Registry bucket. `gcr.io/google-samples/hello-app:1.0` indicates the specific image version to pull. If a version is not specified, the latest version is used.
- To create a Kubernetes Service, which is a Kubernetes resource that lets you expose your application to external traffic, run the following `kubectl expose` command:
 - `kubectl expose deployment hello-server-gcp --type=LoadBalancer --port 8080`
- NOTE: In this command:
 1. `--port` specifies the port that the container exposes.
 2. `type="LoadBalancer"` creates a Compute Engine load balancer for your container.
- To inspect the hello-server Service, run `kubectl get`: a. `kubectl get service`
- To view the application from your web browser, open a new tab and enter the following address, replacing [EXTERNAL IP] with the EXTERNAL-IP for hello-server.
 - `http://[EXTERNAL-IP]:8080` `http://35.245.179.251:8080 (http://35.245.179.251:8080)`

Task 5. Deleting the cluster

- To delete the cluster, run the following command:
 - `gcloud container clusters delete lab-cluster`
- When prompted, type Y to confirm.
 - Deleting the cluster can take a few minutes.

copyright © 2022—2023 Cloud & AI Analytics. All rights reserved