Kubernettes Setup

Quickstart

This quickstart shows you how to deploy a containerized application with Kubernetes Engine.

Before you begin

Take the following steps to enable the Kubernetes Engine API:

1. Visit the Kubernetes Engine page in the Google Cloud Platform Console.
2. Create or select a project.
3. Wait for the API and related services to be enabled. This can take several minutes.
4. Make sure that billing is enabled for your project.
5. Learn how to enable billing

Choosing a shell

To complete this quickstart, you can use either Google Cloud Shell or your local shell.

Google Cloud Shell is a shell environment for managing resources hosted on Google Cloud Platform (GCP). Cloud Shell comes preinstalled with the gcloud and kubectl command-line tools. gcloud provides the primary command-line interface for GCP, and kubectl provides the command-line interface for running commands against Kubernetes clusters.

If you prefer using your local shell, you must install the gcloud and kubectl command-line tools in your environment.

Cloud ShellLocal Shell

To launch Cloud Shell, perform the following steps:

1. Go to Google Cloud Platform Console.
2. Google Cloud Platform Console
3. From the top-right corner of the console, click the Activate Google Cloud Shell button:

A Cloud Shell session opens inside a frame at the bottom of the console. You use this shell to run gcloud and kubectl commands.

Configuring default settings for gcloud

Before getting started, you should use gcloud to configure two default settings: your default project and compute zone.

Your project has a project ID, which is its unique identifier. When you first create a project, you can use the automatically-generated project ID or you can create your own.

Your compute zone is an approximate regional location in which your clusters and their resources live. For example, us-west1-a is a zone in the us-west region.

Configuring these default settings makes it easier to run gcloud commands, since gcloud requires that you specify the project and compute zone in which you wish to work. You can also specify these settings or override default settings by passing operational flags, such as --project, --zone, and --cluster, to gcloud commands.

When you create Kubernetes Engine resources after configuring your default project and compute zone, the resources are automatically created in that project and zone.

Note: gcloud might return an error if these settings are either not configured or not specified.

Setting a default project

To set a default project, run the following command from Cloud Shell:

gcloud config set project PROJECT\_ID

Replace PROJECT\_ID with your project ID.

Setting a default compute zone

To set a default compute zone, run the following command:

gcloud config set compute/zone COMPUTE\_ZONE

where COMPUTE\_ZONE is the desired geographical compute zone, such as us-west1-a.

Creating a Kubernetes Engine cluster

A cluster consists of at least one cluster master machine and multiple worker machines called nodes. Nodes are Compute Engine virtual machine (VM) instances that run the Kubernetes processes necessary to make them part of the cluster. You deploy applications to clusters, and the applications run on the nodes.

To create a cluster, run the following command:

gcloud container clusters create CLUSTER\_NAME

where CLUSTER\_NAME is the name you choose for the cluster.

Note: It might take several minutes to finish creating the cluster.

Get authentication credentials for the cluster

After creating your cluster, you need to get authentication credentials to interact with the cluster.

To authenticate for the cluster, run the following command:

gcloud container clusters get-credentials CLUSTER\_NAME

This command configures kubectl to use the cluster you created.

Deploying an application to the cluster

Now that you have created a cluster, you can deploy a containerized application to it. For this quickstart, you can deploy our example web application, hello-app.

Note: To read the source code for hello-app, refer to the hello-app code review section.

Kubernetes Engine 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.