## Lab 4-02: Deploying To Google Kubernetes Engine Cluster

### Introduction

Suppose a system administrator calls the developer to inform her that her application has crashed, but it is working just fine for the developer. They checked the logs and other possible errors to find the root problem and saw that the version of inconsistencies and right dependencies were missing. They installed the missing dependencies to solve the problem but still faced the same issue. They decided to fix the problem finally by using containers. **Containers** disengage application dependencies and code from the OS. This idea will allow the system administrator to log in to each machine and instruct it to run the container. It will pull down only the files that have changed since the last container and run the code. However, the system administrator needs to deal with more than one developer, so she decides to go with **Kubernetes** as the solution. Kubernetes is a portable and open-source platform for managing containers, but it is not as simple to manage Kubernetes. So the viable solution is **Google Kubernetes Engine,** which is a managed service to run Kubernetes that, along with cluster creation, offers advanced cluster management services.

### Problem

An organization must deploy a sample application to a cluster of 4 nodes. How will this be done?

### Solution

They can use Google Kubernetes Engine to deploy an application to the cluster.

|  |
| --- |
| **Step 1: Create Kubernetes Cluster**  1. Sign in to the GCP console <https://accounts.google.com/>.  2. Navigate to **Clusters** in **Kubernetes Engine** under the **Compute** section**.**    3. Click **Create** on the following pop-up menu.    4. Click **Configure** in front of the **GKE Standard** option found in the following pop-up window.     1. Enter the name and location of your cluster.      1. In the section of **Node Pool Details**, change the number of nodes from 3 to 4.     7. Select your **Zone** and choose **Release channel** under the **Control plane version.**    8. Click on **Create**.    **Step 2: Get Files**   1. Clone the git hub repository by using the git clone command.   **https://github.com/linuxacademy/content-gcpro-developer**   1. Change the directory with the command **cd/content-gcpro-developer/kubernetes-engine-lab** to move to the kubernetes-engine-lab folder under the content-gcpro-developer directory.   Description: Text  Description automatically generated  **Step 3: Configuring the ‘config.yaml’ file**   1. Click on **Open Editor.**      1. Navigate to **config.yaml** under **kubernetes-engine-lab** under **content-gcpro-developer.**      1. On line 32 of the code, enter your project name before the image string.      1. Click on **Save** or press CTRL+Sto save changes.   **Step 4: Push the Containerized App into the Container Registry**   1. To build the containerized docker image, open the cloud shell and write the following command.   **docker build -t la-container-image**   1. In the cloud shell, write the following command to configure Docker.   **gcloud auth configure-docker**   1. Tag the image with the registry and project names using the following command.   **docker tag la-container-image gcr.io/[project name]/la-container-image:v1**   1. Push this image using the following command.   **docker push gcr.io/[project name]/la-container-image:v1**  Description: Text  Description automatically generated   1. To confirm the operation, navigate to **Images** in the **Container Registry** pageunder the **CI/CD** section.      1. You will see the **la-container-image** there.     **Step 4: Deploying Workload**   1. Navigate to **Workloads** on the **Kubernetes Engine** page under the **Compute** section.      1. Click **Deploy** on the following pop-up menu.      1. Choose the **Existing container image** and click **Continue**.      1. Click **Deploy.**      1. Now to increase the number of pods, navigate to the **YAML** tab.      1. Click **Edit** change the number of replicas to **4,** and click **Save**. 2. Click **Expose** on the deployment page.      1. Set the **Service type** to the **Load balancer** and click **Expose**.      1. Confirm by clicking the **external link.**      1. The following page will appear.   Description: Text  Description automatically generated |