

Lab 11.1: Cloud Build

ID720 Virtualisation

Introduction

Last time we talked about how tools like Kubernetes facilitate CI/CD pipelines, and about how efficient pipelines correlate strongly with overall project success. Today we'll see how to use Google's *Cloud Builder*, which can fill a key role in a CI/CD pipeline.

To start today's lab, create a GCP project named "lab11".

1 Enable needed services

For this lab we will need two GCP tools, Cloud Build and Container Registry. You'll find both of these in the main GCP menu (left side of the page) under the "CI/CD" section. Enable both of these tools for your account, if necessary.

You'll probably see some notice about Artifact Registry when activating the Container Registry. Artifact Registry will replace Container Registry, but for the time being I found some problems using it, so we'll use Container Registry for this lab.

2 Create application

We need something to build and deploy, so you'll find a simple Flask app in the `lab11` directory. Copy the `lab11` folder and its three files into your Cloud Shell working directory. You won't need to modify these files, but you may want to check their contents, particularly the `Dockerfile`.

Configure the cloud build In the `lab11` directory in your Cloud Shell, add a new file named `cloudbuild.yaml`. This file will specify the steps to be carried out in executing our cloud build. The first step is

```
# Docker Build
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t',
        'gcr.io/${PROJECT_ID}/lab11',
        '.']
```

This step will build a Docker container image using the files in our `lab11` directory. Coincidentally, it does this in a Docker container whose image is specified with the `name` attribute above. If our cloud build was meant to produce something different, say perhaps a `.jar` file, we would use a different image.

Now add our second step to the file as follows:

```
# Docker Push
- name: 'gcr.io/cloud-builders/docker'
  args: ['push',
        'gcr.io/${PROJECT_ID}/lab11']
```

This step will push our newly built image into Google's Container Registry

3 Execute our build

With all of these files saved in our Cloud Shell, we can execute our build with the command

```
gcloud builds submit --region=us-west2 --config=cloudbuild.yaml .
```

Note that you need to enter this command in your Cloud Shell while inside your `lab11` directory. The build will take a few minutes and print output to your shell while it is in progress. When it finishes, check your Container Registry's contents in your browser. You should see your new image.

4 Deploy and expose your app

We already know how to deploy and expose an app from previous labs. Briefly,

1. Launch a cluster;
2. Create a YAML file with a **Deployment** manifest (You should have examples from earlier labs);
3. Apply the manifest;
4. Expose the deployment (The app listens on port 5000) with a **LoadBalancer** service.
5. Check to see that your application is visible online.

Once everything is done, don't forget to remove your load balancer and cluster.