Assignment 3: Kubernetes and DevOps
ID720 Virtualisation

Introduction

For this assignment you will deploy a web application to a GKE Kubernetes cluster, along with its needed resources. You will deploy the application using a fully automated CI/CD pipeline to a staging cluster first, and then to a production cluster upon final approval.

This assignment is worth 25% of your mark in this paper.

1 Specifications

Create two k8s clusters: One for staging and one for production.

Your application will use two deployments with an associated service for each:

1. A Flask application - clone the repo at `https://github.com/tclark/assignment3-app`. Note that you won't interact with this repository apart from cloning it to get the code initially. When you set up your pipeline you will commit your versions of the code into a different repository. Your deployment should deploy two replicas of this pod. Expose this application with a load balancer.
2. A Redis server - use the standard image from Docker Hub. Expose this server internally with a Cluster IP.

In addition you will create a ConfigMap with two key-value pairs, one fpr the Redis hostname and one for the Redis port number.

For the Flask application, create a CI/CD pipeline on GCP that builds and deploys to your staging cluster whenever a new commit is pushed, and that deploys to production on approval. Note that the CI/CD pipeline is only used for the Flask application since it's the only part of the project for which you would update the code. The Redis deployment and other elements of the project are not managed by the pipeline.

After you deploy the full application stack, make a change to the Flask application code and push your changes to initiate a CI/CD rollout. For the assignment your pipeline logs should show **at least two** rollouts.

2 Submission

Your assignment will be marked according to the following criteria:

- The application should be accessible over the Internet and function properly.
- All of the elements specified above should be present and configured correctly.
- Demonstrate your CI/CD pipeline by rolling out at least one update to the Flask app after the initial deployment.

To submit your assignment, add `tclark@op.ac.nz` as a project editor on your GCP project. You may want to do this early so that I can check things should you have any questions while working on it. Once your project is completed, email the lecturer to let him know that it's ready.

This assignment is due at 11:59 PM on Friday, 11 November.

3 Appendix: Suggested Workflow

There are a lot of elements to this assignment, so a good plan is to start with something basic and build it up over time. Here's a possible approach:

- Start up one cluster and get the the various application elements running on it.
  - create the ConfigMap
  - deploy Redis
  - expose the Redis deployment
  - deploy the Flask app
  - expose the Flask app
- Set up a cloud build process for the Flask application
- Create your production and staging clusters
- Configure the CI/CD pipeline for the Flask app
- Verify that the pipeline works by rolling out an update

This plan will also avoid using too much GCP credit since you don't have to run the full stack until you get close to finishing.