Project Summary: Kubernetes Microservice

In this project I will be using AWS Codebuild, Elastic Kubernetes Service, Cloudwatch, Elastic Container Registry, and Docker to build a docker image, store the docker image, deploy the docker image and monitor the activity of the deployment. In the Kubernetes cluster I will setup 2 pods, one that hosts the PostgreSQL database application (the database server) and one that will run a python script (app2.py, the analytics application) that will connect to the pod with the PostgreSQL database and run hard coded SQL queries to retrieve the latest data in the database. The pod that hosts the analytics application is a microservice that provides a service to analysts by retrieving data that is routinely queried from the PostgreSQL database through an API. The pod that hosts the database application is also a microservice that provides a service that stores data.

The following items are used to deploy the service.

1. App2.py is the python code for the coworking analytics application that will be run in a pod in the kubernetes cluster
2. buildspec.yaml is used by AWS Codebuild for the build steps to build/create the docker image that will be stored in the Elastic Container Registry. The source of the buildspec.yaml file is my github repo, which is linked to my AWS Codebuild project.
3. config.py file is referenced by the app.py file to get the environment variables.
4. requirements_new.txt is used by dockerfile to install the appropriate dependencies in the docker image.
5. dockerfile is the file used by AWS Codebuild to build the docker image, which the source is my github repo
6. ConfigMap.yaml is the file to be loaded into kubernetes with the environment variables that the service needs to access the database
7. Secret.yaml is the file to be loaded into kubernetes with the password to the database
8. deployment.yaml is the file to deploy the docker image stored in the elastic container service to a pod
9. service.yaml is the file to create the service which will facilitate communication to the deployed pod