Blue-green deployment using nginx and Docker

Blue-green deployment is a methodology for releasing new code into the production environment with the purpose of avoiding service interruptions to applications using two identical environments and a router to seamlessly switch service between environments. In this exercise, we simulate blue-green deployment of a simple web application written in Python. We use two docker containers to model the identical blue-green deployment environments and use nginx as the router.

1 Initial Setup

1) To begin, clone the following git repo: https://github.com/software-rebels/blue-green-deploy.git
2) Pull required docker images from docker hub - nginx:1-alpine and ecse437/hello_web_app.
3) In order to enable the nginx and web servers to easily communicate with each other using the other container's name or IP, create a new *docker network* named "lab5". All the containers you create for this exercise must connect to this network.

2 Write script to simulate Blue-Green deployment

Write a python/shell script called blue-green.(py|.sh) that will perform the following steps:

1) **Create the "Blue" environment**: Create a container named "hello_BLUE" from the ecse437/hello_web_app image. Pass an environment variable "APP_VERSION" to 1.0 to this container. The response of the web application changes according to APP_VERSION.
2) **Create the "Router"**: Create a container named "nginx" using the nginx:1-alpine image. The host forwards requests received on port 8080 to port 80 on the nginx container. The directory nginx-conf.d must be mapped to /etc/nginx/conf.d on the container.
3) **Create the "Green" Environment:** Create a container named "hello_GREEN" identical to the "Blue" environment, except for APP_VERSION, which must be set to 2.0.
4) **Update nginx configuration**: Modify the configuration file so that nginx forwards the requests to the "Green" environment instead. After modifying the configuration, *reload* the nginx configuration for changes to be reflected.

The objective of this exercise is to switch-over from one environment to another without disruption of service. The "test.sh" script continuously tries to access the web application and displays the response. You can use this script to verify if the service is disrupted at any time.
HINT: Make sure to run all containers in the background.

3 Deliverables

For grading, you must submit a zipped up copy of blue-green.(py|.sh) through mycourses. The complete exercise is due by the end of the lab period.

Grading Scheme

Part of Exercise	***Marks***
Puppet code:	/50
Functional correctness	40/50
Design of the solution	5/50
Style	5/50
Total Marks:	/50