DevOps Technical Test 2.0a

Introduction

This test is designed to test your problem solving ability, your DevOps knowledge and your understanding of linux concepts. The aim is to assess your problem solving approach, your ability to turn your solution into a working system and the way in which you structure your implementation.

The test must be returned to your test coordinator within 3 days of being issued, though the expectation isn't to spend more than half a day on it, obviously this is just to give you enough time to fit it in. Once your test is completed please give us access to your private github repo, or if this isn't possible let us know and we can negotiate upload via Google Drive. Our github usernames are Litostorg and pgilmar.

If you need clarification on any of the requirements then please email your test coordinators for advice.

Environment Setup

To attempt this test you will need a working development environment setup that includes:

- Internet access.
- An operating system with the latest version of docker installed.

This test does not require an advanced knowledge of docker in order to be attempted although existing experience in docker will be advantageous. A candidate with no prior docker knowledge should familiarise themselves with the basic concepts of the technology before attempting the test.

Design Requirements

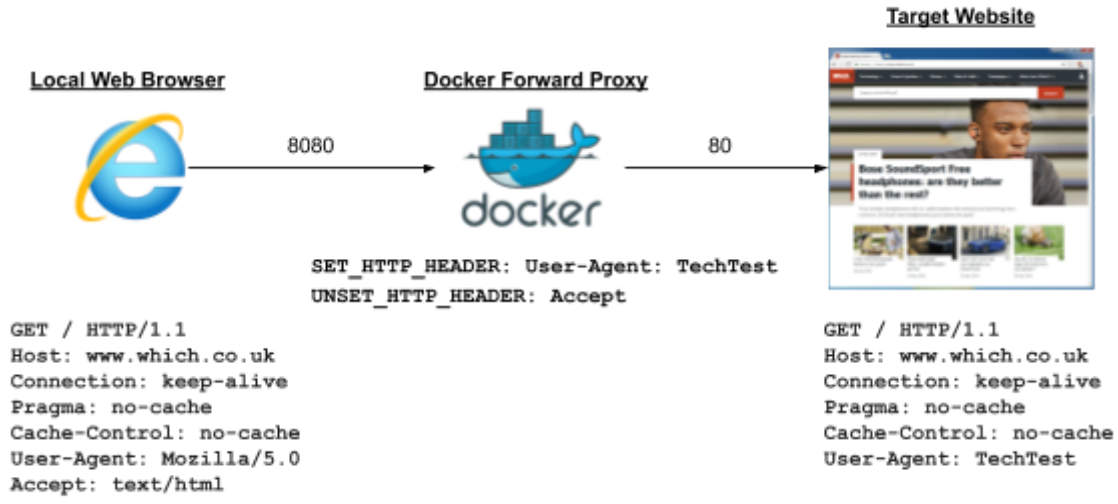
The goal of this test is to create a forward proxy using apache that can be used to manipulate HTTP requests as well as providing a monitoring page that displays statistics on the container. This task has two parts whose requirements are described in more detail below.

Forward Proxy Requirements

The implementation of the forward proxy has the following requirements:

1. The forward proxy must be run inside a docker container that uses the official apache docker image as its base image, see: https://hub.docker.com/_/httpd/.
2. The forward proxy should be accessible on port 8080 from a web browser on the host operating system. The port the forward proxy runs on inside the container is up to you.
3. The docker container should accept the following variables whose effect is described below:
    a. UNSET_HTTP_HEADER - This variable should accept a comma separated list of HTTP headers that will be removed in the outbound HTTP request, e.g. UNSET_HTTP_HEADER : Accept-Encoding, User-Agent, Cookie.
    b. SET_HTTP_HEADER - This variable should accept a comma separated list of HTTP headers that should be added or amended in the outbound HTTP request, e.g. SET_HTTP_HEADER: User-Agent: TechTest, Host: www.which.co.uk
4. The method that you use to apply the docker variables to any configuration files within the container is completely at your discretion.

A diagram showing the request flow from the local browser to the external target website can be seen below:



Monitoring Page Requirements

The implementation of the monitoring page has the following requirements:

1. The monitoring page must be served by the same docker container that the forward proxy runs within, e.g. The entire solution (both forward proxy and monitoring page) must consist of a single docker container only.
2. The monitoring page should be available to the host web browser on a port of your choice, e.g. http://127.0.0.1:8081/status. The same port that the forward proxy exposes (8080) can be used to serve the status page if requirement.
3. The status page should not be generated in real time and should instead be created on a schedule every X seconds.
4. A docker variable called “STATUS_REFRESH_PERIOD” should be implemented to control the refresh period frequency.
5. The status page should show the following information:
    a. The date the status page was generated.
    b. The number of running processes within the container and their CPU utilisation.
    c. The memory usage within the container.
    d. The disk usage within the container.
6. The formatting of the status page content is completely at your discretion but functionality should be considered more important than appearance.
7. The method that you use to generate the status page is completely at your discretion.

Tips

1. The entrypoint into a docker container can be overridden to allow for custom commands to be run before other processes are started.
2. There are many standard linux programs available from the command line that can assist in the creation of the monitoring page.
3. Note that the requirements of this test are not indicative of docker best practice or the kind of work you would be doing day to day basis, they just make for an interesting problem to solve.