Assignment #4

TA Email: kentma@nyu.edu

Introduction

The goal of this assignment is to gain hands on experience with securely deploying your application. To achieve this you will be working with Docker, Docker compose, Docker secrets, Notary (Docker Content Trust), and Kubernetes. By the end of this assignment you should have an understanding of how to secure the deployment and updates of an application after the code is built and tested. The assignment is to containerize and securely deploy your spell checking Web service. This document outlines the steps required to achieve this goal.

Setup

Completion time – 2 to 3 hours

To complete this assignment you will need to install Docker, Notary, and Kubernetes or Docker Swarm. Docker should be available in most modern package managers, and can also be found at https://docs.docker.com/install. Information on using Notary with docker (Docker Content Trust) can be found at https://docs.docker.com/engine/security/trust/content_trust/. Information on installing and using Kubernetes for the purpose of this assignment can be found at https://kubernetes.io/docs/tasks/tools/install-minikube/.

Deliverables & Grading

The configuration files and reports are to submitted through NYU Classes. Your submission should be a compressed archive file which contains all configuration files and the report in PDF, ODT, or text format.

1. Correctness: 50 pts.
2. Report: 50 pts.

Total 100 pts.

Docker and Secrets

For this part of the assignment, you are tasked with dockerizing your application. To complete this portion of the assignment, you must create a Dockerfile for your Web service. All sensitive information such as secret keys (for instance for generating CSRF tokens) and default admin passwords must be supplied using Docker secrets or Kubernetes secrets (see next section). The dockerfile should be self-contained, a person who wants to deploy your service should have to perform no other actions other than building your docker container. This means that your dockerfile must install the dependencies for your application or use a parent image that already contains these dependencies. In order to allow us to easily grade, we require that your Web service be exposed on port 8080.

After the Dockerfile is created for your sevice, these image should be placed in a .docker-compose.yml file. In order for someone to deploy your application, they should only need to run the command:

```
$ docker-compose up -d
```

This would be more useful if you Web service and database were separated, and both services were to be launched using docker-compose. However, due to time constrains and the design of the previous assignments you will only be using one docker image.

To complete this portion of the assignment, you will need to turn in a Dockerfile and a .docker-compose.yml file.

Docker Content Trust

After your Web service has been containerized, one must ensure that updates can be delivered securely. We will do this using Docker Content Trust, which is built on notary. Information on Docker Content Trust can be found at https://docs.docker.com/engine/security/trust/content_trust/.

To avoid pushing too many images to the Docker Hub, in this portion of the write-up you can just explain how you would use Docker Content Trust and what it would add for your program. You are not required to actually perform these steps.

Kubernetes or Docker Swarm

We want to deploy the service with many replicas so that our service can maintain availability. We will do this with kubernetes or with docker swarm. We also want to limit our pods to have resource limitations so we do not overburden or hog the machine our pods are running on. Our Web service should have four replicas with limited CPU and memory.

For this assignment we assume load balancing is being performed by a load balancing solution or by a service mesh like Istio, but you are not required to implement this due to time constraints on the assignment.

For this portion of the assignment you need to turn all files and/or the commands you executed to get your replicated service up and running.

Writeup

After you have completed setting up all of the software listed above, write a paragraph for each tool explaining how you set it up (providing commands where necessary, or reffering to the submitted configuration files otherwise) and what problem(s) the software solves.

Extra Credit

Extra credit may be granted by the graders for assignments that go above and beyond the requirements specified in this document.

Hints

You may want to view tutorials for how to use each of the applications mentioned here. Configurations can be complex, and following a walkthrough can help translate the theory that you learned about in class into practice. You should also check if there are tools available to easily convert between docker-compose and kubernetes to save some effort in setting up your kubernetes configuration.

Late Policy

Late assignments will not be accepted.