RX-M

Docker Images

Lab – Dockerfiles II

This is a challenge lab and provides minimal instructions, thus requiring you to use the course book, your resourcefulness and the Docker references you have learned about in class to complete several Dockerfile and image oriented tasks.

The goal of the lab is to produce a single layer alpine image look alike that automatically runs a bash shell.

1. Export an Alpine container to create a tar file of the alpine file system

`docker create docker export`

2. Create a Dockerfile that builds your new alpine image with the bash executable as teh default command

`FROM scratch ADD CMD`

- What is the difference between the approach outlined here and simply using FROM alpine
- Can you guarantee that FROM will use a specific set of approved bits?
- Can you guarantee that FROM will always use an image with only one layer?

3. Content addressable FROM

The Dockerfile FROM command supports three possible forms:

- FROM
- FROM :
- FROM @

By using the alpine image repository name and the SHA digest you can guarantee that a specific set of bits is used as your base. Recreate the previous project using FROM with the Alpine digest.

4. Load/Save

- Run `docker ps` to display all of your images
- `docker save` (one of) the custom Alpine image repos you just created
- Remove the custom image and ensure that it is no longer present
- `docker load` the custom alpine image back from the saved file

Can you guarantee this image is the same as the prior image? If so how?

[OPTIONAL Extra Credit] Build the Python 2.7 Image

The official Python images are sourced here: https://github.com/docker-library/python

Build the image for Python 2.7 using the above URL, specifying the master branch and the Python 2.7 dockerfile.

Congratulations, you have completed the Dockerfile II lab!