A.I. For Software Testing and Reverse Engineering CS4110

This document provides instructions on how to build the Docker container using the Dockerfile that is provide on BrightSpace. Please follow the instructions that are specific to the operating system of your machine. If there are any questions regarding the setup, please drop by during lab hours or send us a message on Mattermost.

Step 1: Download and Install Docker

Head over to https://www.docker.com/products/docker-desktop and follow the install instructions that listed on the website to install Docker on your machine.

Building the Docker Container

There two ways to build the Docker container. Below we list both methods to build the container.

Using the Dockerhub way:

Step 1: Pull the Docker image from Dockerhub

Head over to https://hub.docker.com/repository/docker/clintoncao/aistr and pull the Docker image using the following command:

```
docker pull clintoncao/aistr:latest
```

That's it!

Building the image yourself (You can neglect this part if you used the Dockerhub way):

Step 1: Download the Dockerfile and scripts

Head over to https://github.com/ClintonCao/AISTR_Docker and download the Dockerfile and scripts that are listed in the repository.

Step 2: Build Docker container using the Dockerfile

Once you have downloaded the Dockerfile and scripts, start your terminal (or PowerShell on Windows) and navigate to the folder where the Dockerfile is located. Then run the following command to build the Docker container:

```
docker build -t IMAGE_NAME .
```

The -t flag stands for tag and that is the name that you want to give the Docker image. IMAGE_NAME is where you will fill in the name that you want to give the image.

Note 1: You might need to have administrator/sudo rights to run docker commands.
Note 2: As we are installing and building all the tools for you, it might take some time for the build to finish!

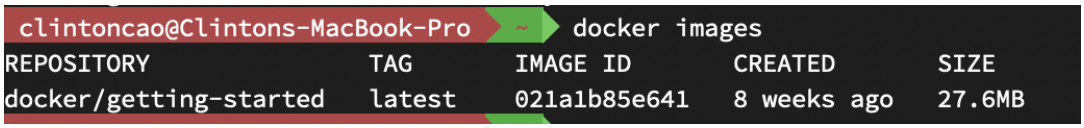
Running the Docker container

If you have Docker desktop installed on your machine, you can go to images and run the via from Docker desktop. You can then also give a name to the Docker container. You can also start the container using the terminal/PowerShell:

First, list all the images that are on your machine:

```
docker images
```

You should then see something like the picture below in your terminal/PowerShell:



Find the image that that you have created using the Dockerfile that we provided you. Now run the Docker container using the following command:

```
docker run -it --name CONTAINER_NAME IMAGE_NAME /bin/bash
```

The it flags is used to tell the docker that you want to start an interactive session with tty attached. The argument/bin/bash gives a bash shell. The --name flag is used to tell the name that you want to give the container and CONTAINER_NAME is where you should fill in the name for the container. To exit the container, simply run exit and this will stop the container from running.

Syncing changes of your host machine with the docker container

You can create a volume in docker to sync the changes that you have made on a file on your host machine to the docker volume. The files on the container will also be updated with the changes. This method copies the filesystem of the container to the volume and then link it with your host machine. You can sync such volume using the following commands:

```
docker volume create --opt type=none --opt device=YOUR/LOCAL/DIRECTORY --opt o=bind --name strdata
```

```
docker run -it -v strdata:/home/str IMAGE_NAME /bin/bash
```

Note1: this was tested on MacOS and Linux, we do not guarantee that this will also work on Windows.
Note2: Make sure your local directory is **empty** (i.e it **cannot** contain any hidden files) for volume linking.
Note3: If you have pulled the Docker image from Dockerhub you can replace IMAGE_NAME with clintoncao/aistr:latest