A.I. For Software Testing and Reverse Engineering CS4110

This document provides instructions on how to build the Docker container using the Dockerfile that we provide you. 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

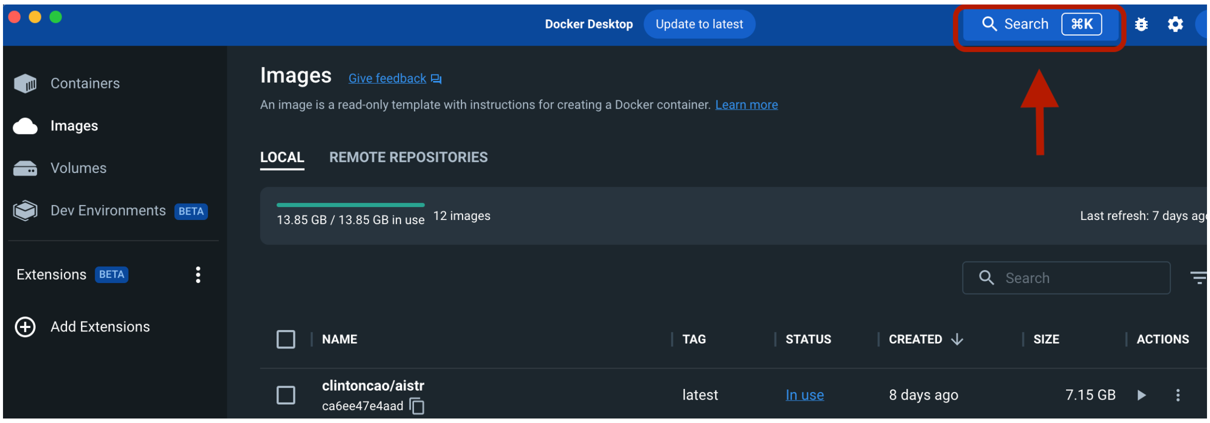
Head over to https://www.docker.com/products/docker-desktop and follow the install instructions that listed on the website to install Docker on your machine.

Pulling the pre-built Docker image

The simplest way to collect the Docker image that we will be using for the lab is to pull the pre-built image from Dockerhub. To pull the pre-built image, you can run the following command in your terminal or PowerShell:

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

Alternatively, you can also pull the docker image directly using the Docker Desktop app. For this option, open the search bar that is located on top of the application (see Figure 1).



An image is a read-only template with instructions for creating a Docker container. Learn more

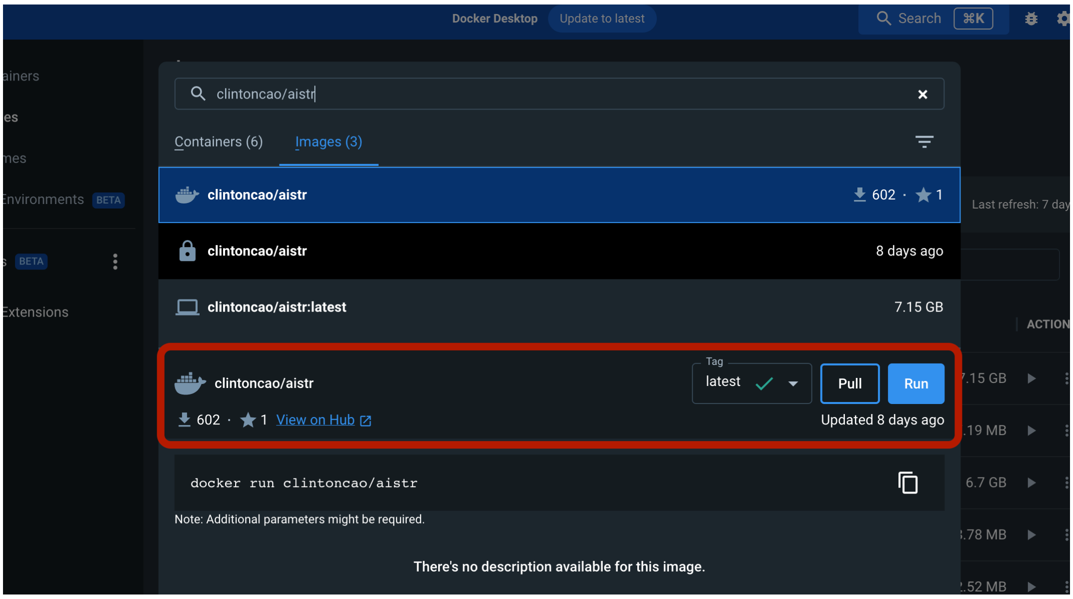
13.85 GB / 13.85 GB in use 12 images

Last refresh: 7 days ag

□	NAME	TAG	STATUS	CREATED	SIZE	ACTIONS
□	clintoncao/aistr ca6ee47e4aad	latest	In use	8 days ago	7.15 GB	▶ ⋮

Figure 1. Search bar in the Docker desktop application

By typing “*clintoncao/aistr*” you should see image pop up in the search results (see Figure 2). You can click on the pull button and Docker automatically pulls the image to your local machine.



```
docker run clintoncao/aistr
```

Note: Additional parameters might be required.

There's no description available for this image.

Figure 2. Docker image showing up in the search results.

Running the Docker container

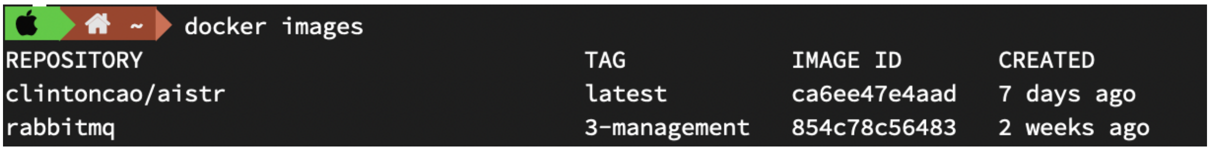
If you have Docker desktop installed on your machine, you can go to images and run the via from Docker desktop. You can then also give a name to the Docker container.

Alternatively, 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 similar to what is shown below in your terminal/PowerShell (Figure 3):



```
 ~ docker images
REPOSITORY                                  TAG            IMAGE ID       CREATED
clintoncao/aistr                            latest         ca6ee47e4aad   7 days ago
rabbitmq                                    3-management   854c78c56483   2 weeks ago
```

Figure 3. List of images after running “*docker images”* in the terminal.

To run the docker container we need to run the following command:

docker run -it --name CONTAINER_NAME IMAGE_NAME

The it flags is used to tell the docker that you want to start an interactive session with tty attached. 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. IMAGE_NAME is the name of the image that you want to use to create the container To exit the container, simply run exit and this will stop the container from running.

To run a docker container using our pre-built docker image, we use the following command:

```
docker run -it --name aistr_container clintoncao/aistr:latest
```

Alternatively, you can also create a container from the Docker desktop application. In Figure 2, you can see a “run” button. Clicking on the “run” should start a container with a random name.

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. **Make sure** that the folder that you are using for the volume is **completely empty** (**no hidden files** are allowed either)! 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 clintoncao/aistr:latest /bin/bash
```

Note: This was tested on MacOS and Linux. We have noticed that sometimes Windows might save files in a different Unicode and the files can then not be parsed by the docker container. If you encounter this issue, please drop by during the lab and ask a TA for assistance.