

Assignment -4
Docker and Kubernetes

Assignment Date	03 November 2022
Student Name	Likesh Kumar B M
Student Roll Number	113119UG07050
Maximum Marks	2 Marks

1.Pull an image from docker hub and run it in docker Playground

Type your text

uifd/ui-for-docker

Explore Repositories Organizations Help

Upgrade

parameshwari

Explore

uifd/ui-for-docker

uifd/ui-for-docker

By uiid • Updated 6 years ago

A web interface for Docker, formerly known as DockerUI. Deprecated, use Portainer for new features.

Other Image

Pulls 10M+

Overview

Tags

UI For Docker

This repo is deprecated. Development continues at: portainer/portainer

chat on gitter

UI For Docker is a web interface for the Docker Remote API. The goal is to provide a pure client side implementation so it is effortless to connect and manage docker.

Goals

Docker Pull Command

docker pull uifd/ui-for-docker

03:42:30

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13

node1

cd9an2u3_cd9av060qau0008hbjs0

IP

192.168.0.13

OPEN PORT

Memory

CPU

SSH

ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-w

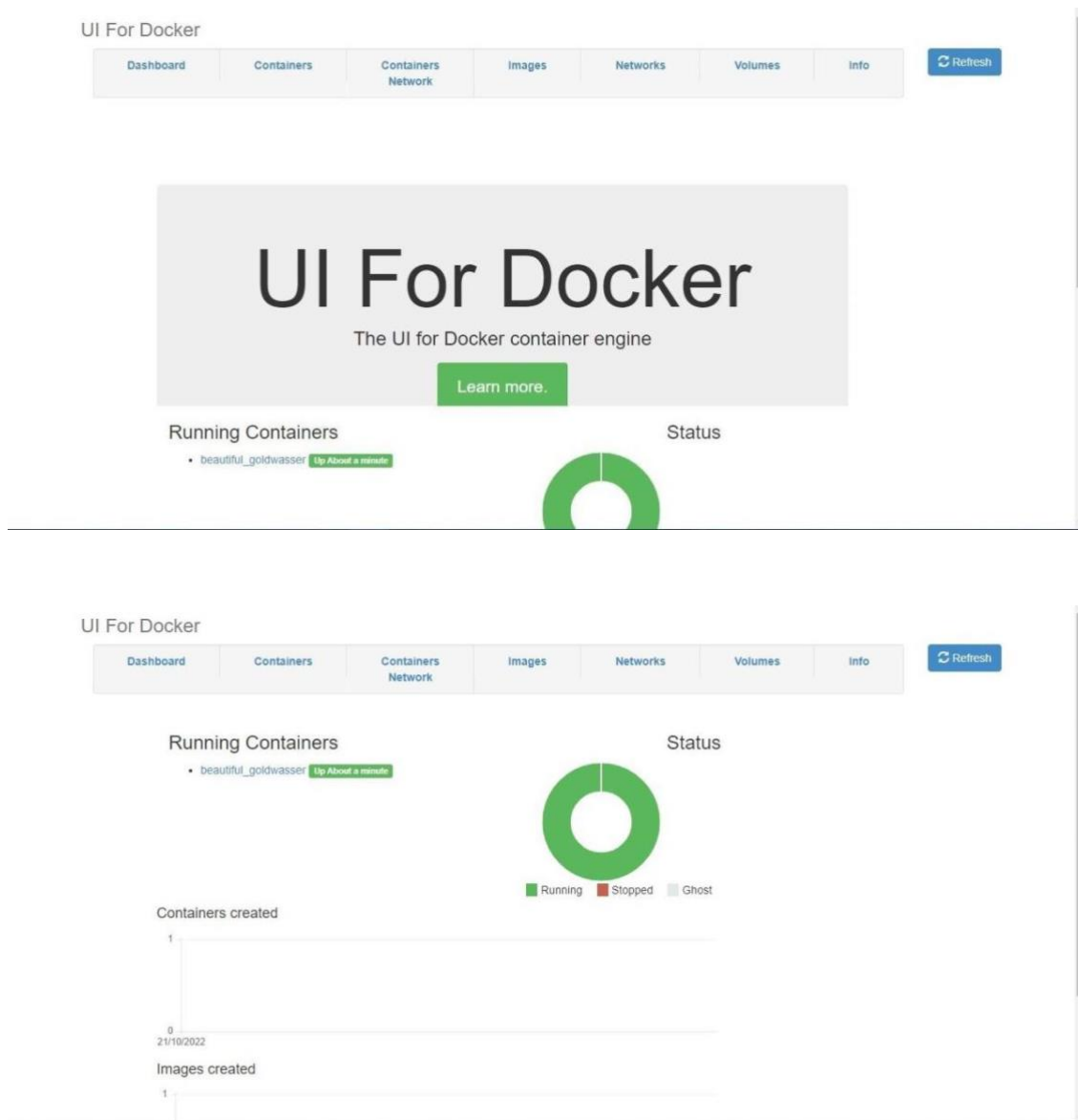
DELETE

EDITOR

```

# This is a sandbox environment. Using personal credentials #
# is HIGHLY discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
# #
# The FWD team. #
#####
(node1) (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
(node1) (local) root@192.168.0.13 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
c590dd163101ae793bdcea0eb1ddd98f6fe549cb5f24dacb9ff7c1931923fc0d
(node1) (local) root@192.168.0.13 ~
$

```

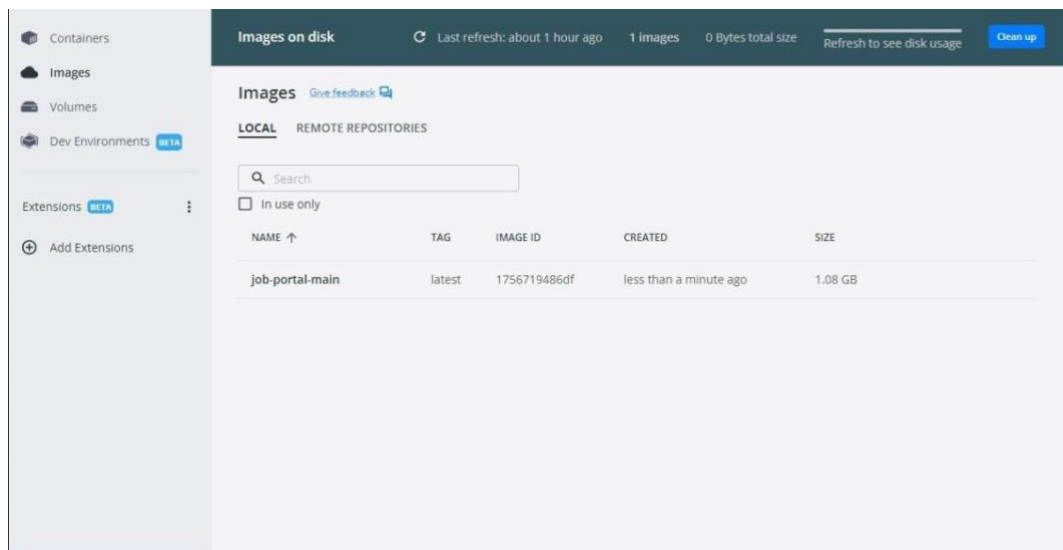


2.Create a docker file for the job portal application and deploy it in Docker desktop application

Type your text

```
C:\Windows\System32\cmd.exe
-> [internal] load build definition from Dockerfile
-> -- transferring dockerfile: 32B
-> [internal] load .dockerignore
-> -- transferring context: 2B
-> [internal] load metadata for docker.io/library/python:3.6
-> [auth] library/python:pull token for registry-1.docker.io
-> [internal] load build context
-> -- transferring context: 687B
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652aef88c25f0d22354d547d802591067aa4026a7f69a81bd9f9308aef8fc
-> resolve docker.io/library/python:3.6@sha256:f8652aef88c25f0d22354d547d802591067aa4026a7f69a81bd9f9308aef8fc
-> sha256:f8652aef88c25f0d22354d547d802591067aa4026a7f69a81bd9f9308aef8fc 1.86GB / 1.86GB
-> sha256:0072a087d8e079d85a31672325d56c18f62214e0448929300976d0d0d 2.22GB / 2.22GB
-> sha256:54290638067c5a3d2a7e21fc80a00c8480a2763d:0021006f771f9744b104 9.27GB / 9.27GB
-> sha256:0e20544d541cd0100201d21a73a0d1d078065c1b95074f32b009a0077a0e1a3 54.92MB / 54.92MB
-> sha256:0b829c73b5202b07d5c07a54f0b7e021995a296c714b53a32a07d19231fcd 5.15MB / 5.15MB
-> sha256:c8507ae3d1722f070eac3f35823ed31ba05d61d5d95c05e5a053d740cd056 10.87MB / 10.87MB
-> sha256:0094e411821d11c077cc322c446307f70806f569a3a0f15c01a0e718793 54.57MB / 54.57MB
-> sha256:0f972400c0a9f0e0172f504f6a00a0e0a81d09fcd01120fc7a4327077 190.51MB / 190.51MB
-> sha256:5a3b2123efc56506e7b0a062081945c164de2a37205e0a030a0a823124c743 8.29MB / 8.29MB
-> extracting sha256:0e20544d541cd0100201d21a73a0d1d078065c1b95074f32b009a0077a0e1a3
-> sha256:0fdd0d5633af20e0ad7a2a1b75a7459c48ed185c5478670741c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:0b829c73b5202b07d5c07a54f0b7e021995a296c714b53a32a07d19231fcd
-> extracting sha256:0107ae301722f070eac3f35823ed31ba05d61d5d95c05e5a053d740cd056
-> sha256:48407040a0e0432c05220b072401c3efce0800f0a700a0b14102f71b0b7 215B / 215B
-> sha256:c4f42be2be530090ebff0a0c10f13de538a3dccc5f54054a5684a0160a3a3f 2.21MB / 2.21MB
-> extracting sha256:6494a4811622b31c027cc322c446307f70806f569a3a0f15c01a0e718793
-> extracting sha256:0f972400c0a9f0e0172f504f6a00a0e0a81d09fcd01120fc7a4327077
-> extracting sha256:5e1b1211efc56506e7b0a062081945c164de2a37205e0a030a0a823124c743
-> extracting sha256:0f407d03031af20e0ad7a2a1b75a7459c48ed185c5478670741c1244bd96752
-> extracting sha256:4004f02040a0b432c05220b072401c3efce0800f0a700a0b14102f71b0b7
-> extracting sha256:c4f42be2be530090ebff0a0c10f13de538a3dccc5f54054a5684a0160a3a3f
-> [2/6] WORKDIR /app
-> [3/6] ADD . /app
-> [4/6] COPY requirements.txt /app
-> [5/6] RUN python3 -m pip install -r requirements.txt
-> [6/6] RUN python3 -m pip install llm_db
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df062f0ddae305c5221513f2f221b4f0a0324b32a28af03779f10
-> naming to docker.io/library/job-portal-main

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\VK-PC\Desktop\job-portal-main>
```



3.Create a IBM container registry and deploy helloworld app