

Assignment -4
Docker and Kubernetes

Assignment Date	26 October 2022
Student Name	R. DEEPAK
Student Roll Number	210519106015
Maximum Marks	2 Marks

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

uifd/ui-for-docker

Explore

Repositories

Organizations

Help

Upgrade

Mridhul

Explore

uifd/ui-for-docker

uifd/ui-for-docker

By uifd • Updated 6 years ago

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

Other

Image

Overview

Tags

UI For Docker

This repo is deprecated. Development continues at: [portainer/portainer](#)

chat

on github

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

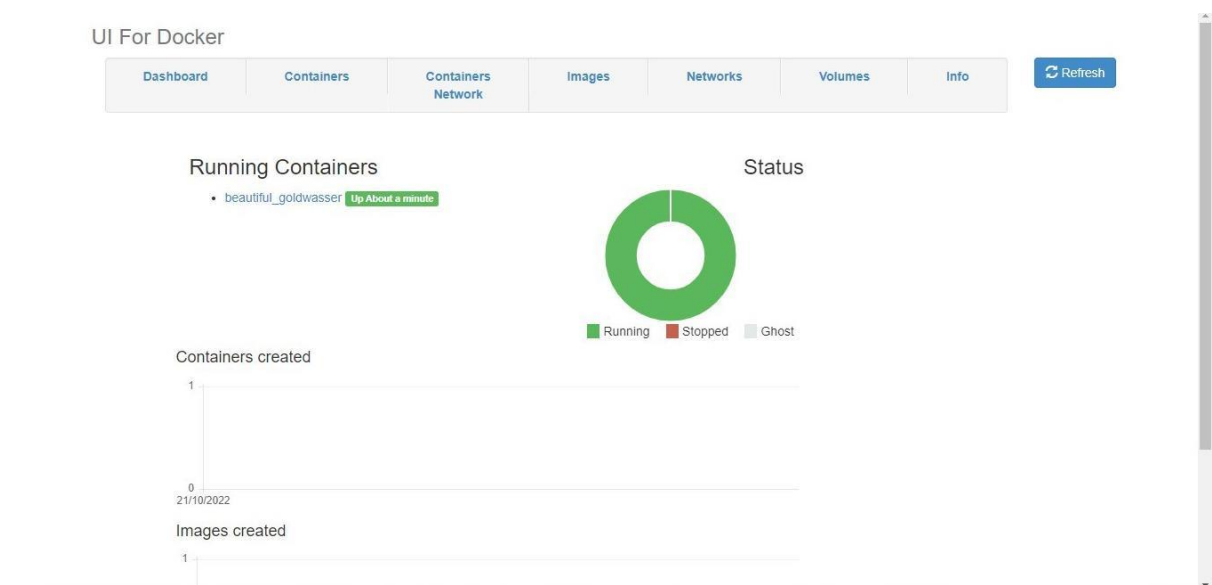
DELETE

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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dadb9ff7c1931923fc0d
[node1] (local) root@192.168.0.13 ~
$

```

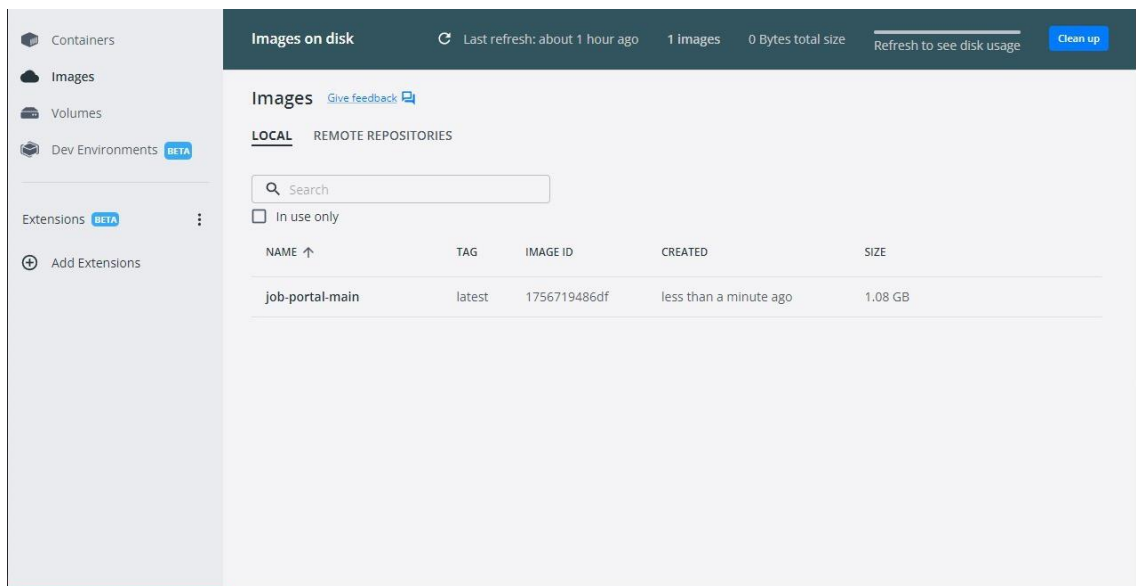


2.Create a docker file for the job portal application and deploy it in Docker desktop application

```
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: 887B
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> => resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> => sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
-> => sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
-> => sha256:54200638d07c5e3ad24c6e21fc889abbc8486a27634c0892006ff71f3f44db104 0.27kB / 0.27kB
-> => sha256:0e29546c541cddb309261021a73a9d1db78665c1b95b74f32b009e0b7796c1e3 54.92MB / 54.92MB
-> => sha256:0829c73b52b92b705c07a54f00f3a921995a296c714b33a32ae07d19231fcd 5.15MB / 5.15MB
-> => sha256:cb5b7ae361722f0ecae53f35823ad21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
-> => sha256:6404a4811622b31c027ccac322ca463937fd805f560a930ef15c01aade718793 54.57MB / 54.57MB
-> => sha256:6f9f74896dffa93fe0172f594faba85e0b4e8a0481a0fef09112efc7e4d3c78f7 196.51MB / 196.51MB
-> => sha256:5e3b1213efc56598e78bd002983945c164de2a37205e06ae2dada823124dc743 6.29MB / 6.29MB
-> => extracting sha256:0e29546c541cddb309261021a73a9d1db78665c1b95b74f32b009e0b7796c1e3
-> => sha256:9fddfdcd56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
-> => extracting sha256:0829c73b52b92b705c07a54f00f3a921995a296c714b53a32ae07d19231fcd 2.35
-> => extracting sha256:cb5b7ae361722f0ecae53f35823ad21baa85d61d5d95cd5a95ab53d740cdd56 4.00
-> => sha256:404f02044bac0432ca522cbb9f254b1c91fcae680bfeef0be0b243b2f31bab7 235B / 235B
-> => sha256:c4f42be2be53b000ebffcc040c1d13de538434ccc5f5d954a56048a6169a3a3f 2.21MB / 2.21MB
-> => extracting sha256:6404a4811622b31c027ccac322ca463937fd805f560a930ef15c01aade718793 27.35
-> => extracting sha256:6f9f74896dffa93fe0172f594faba85e0b4e8a0481a0fef09112efc7e4d3c78f7 131.45
-> => extracting sha256:5e3b1213efc56598e78bd002983945c164de2a37205e06ae2dada823124dc743 8.25
-> => extracting sha256:9fddfdcd56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 11.35
-> => extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcae680bfeef0be0b243b2f31bab7 0.05
-> => extracting sha256:c4f42be2be53b000ebffcc040c1d13de538434ccc5f5d954a56048a6169a3a3f 2.85
-> [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 ibm_db
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df002fad5dae385c5221513f2ff2d1b49a8d242b22a28af0379f19
-> naming to docker.io/library/job-portal-main
-> 0.15

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

