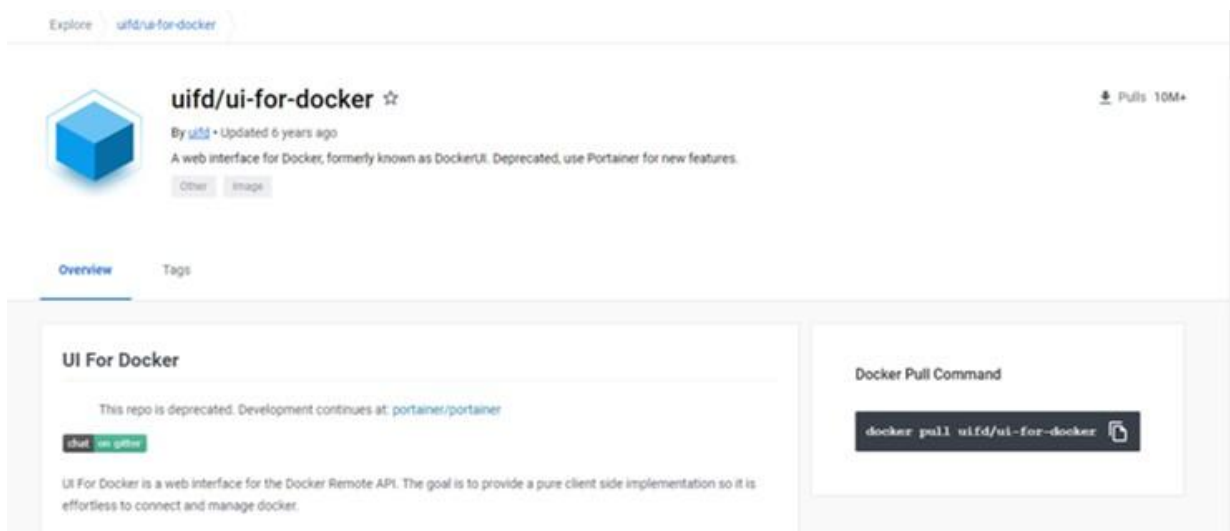


Assignment - 4 Docker and Kubernetes

Batch no	11
Student Name	TEJASHWINI T S
Student Reg Number	921319104209

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



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

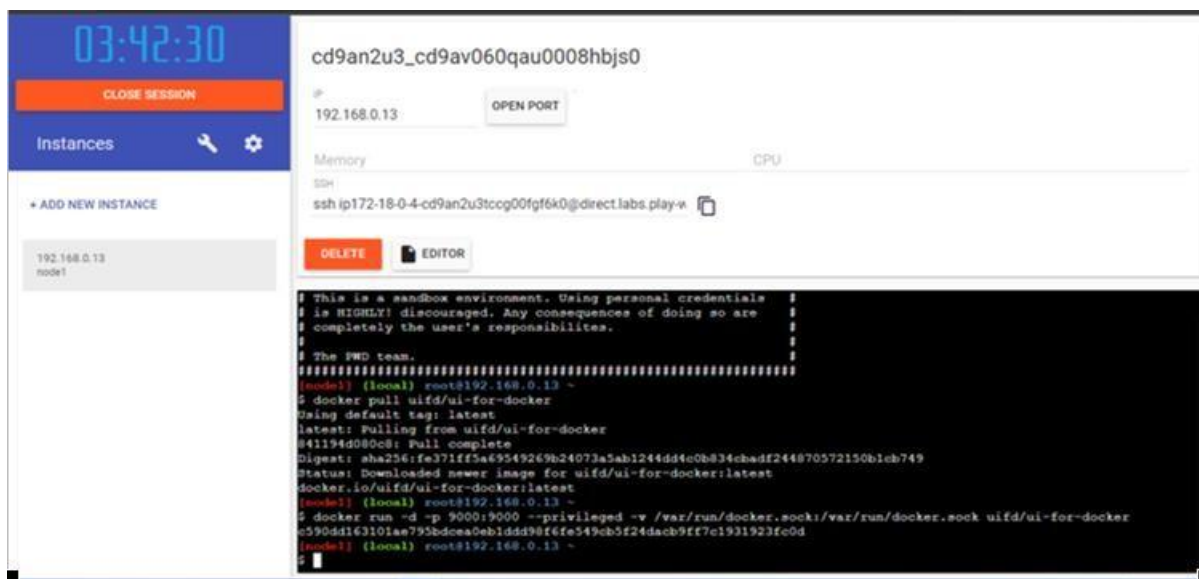
This repo is deprecated. Development continues at: [portainer/portainer](#)

[chat](#) [join 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.

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

192.168.0.13 OPEN PORT

Memory CPU

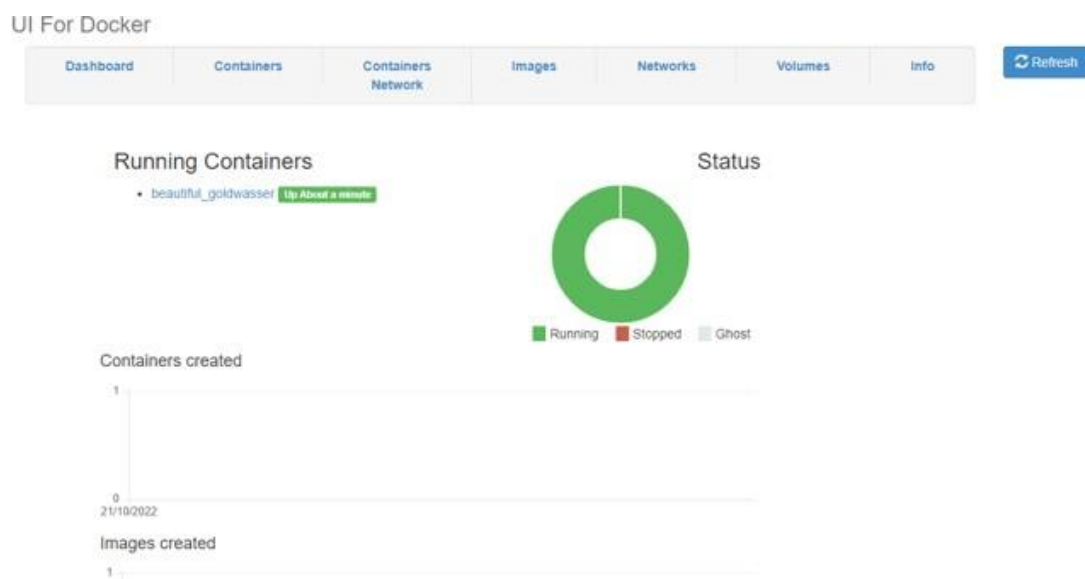
SSH

ssh ip172-18-0-4-cd9an2u3tccg00fg6k0@direct.labs.play-v

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.
#####
[roob1] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
41194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1eb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[roob1] (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
c590dd163101ae795bdcea0eb1dd498f6fe549cb5f24dab9ff7c1931923fcd0d
[roob1] (local) root@192.168.0.13 ~
$
```

UI For Docker



```

-> [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: 607B
-> [1/8] FROM docker.io/library/python:3.6@sha256:f8052a7a986c25f6d22354d547d892591867aa4026a7faba81bd9f180aaf0fc
-> resolve docker.io/library/python:3.6@sha256:f8052a7a986c25f6d22354d547d892591867aa4026a7faba81bd9f180aaf0fc
-> sha256:f8052a7a986c25f6d22354d547d892591867aa4026a7faba81bd9f180aaf0fc 1.05kB / 1.05kB
-> sha256:d097a4907a8ec070d5ac11022359c2de510f82114c0440e926303b376d3d60d 2.22kB / 2.22kB
-> sha256:542600380073e3ac74c8e21fc688abbc8486a27634c0002000ff71f3f44b104 9.27kB / 9.27kB
-> sha256:0e29546d541c0bd309281d21a73abdddb78665c1b95b74f32b00000077ade1e3 54.92MB / 54.92MB
-> sha256:9b828c73052b02b07d5c07a54f10f3e921095a290c714053a32ae67d182211fd 5.15MB / 5.15MB
-> sha256:c0503ae181722f070eac53f50821ed71ba88581d5095cd5a95ab532748cd56 10.87MB / 10.87MB
-> sha256:6454a4211622031c027cc322c443137f0805f54ba3a6f15c01ade718791 54.37MB / 54.53MB
-> sha256:0f9f74800d903f01727594fand5e0044a0401a9fef001120fc764d3c70f7 100.51MB / 100.51MB
-> sha256:5e301215efc50598c78bd007081045c164de2a7705e06a62dadab21124dc743 6.20MB / 6.20MB
-> extracting sha256:0e29546d541c0bd309281d21a73abdddb78665c1b95b74f32b00000077ade1e3
-> sha256:0f0f0158134f5a6d7e241b5a745c40a0105c5878070f41c1344b00752 14.11MB / 14.21MB
-> extracting sha256:9b828c73052b02b07d5c07a54f10f3e921095a290c714053a32ae67d182211fd
-> extracting sha256:c0503ae181722f070eac53f50821ed71ba88581d5095cd5a95ab532748cd56
-> sha256:484f0204d4ac0412c52c0b0754b1c0f1e6d000f4e98005436f31b0b7 2.95B / 2.95B
-> sha256:c442be2be51b000bffc040c10f13de538434ccc5f5d054a5608a0100aba3f
-> extracting sha256:8404e811822031c027cc322c443137f0805f54ba3a6f15c01ade718791
-> extracting sha256:0f9f74800d903f01727594fand5e0044a0401a9fef001120fc764d3c70f7
-> extracting sha256:5e301215efc50598c78bd007081045c164de2a7705e06a62dadab21124dc743
-> extracting sha256:0f0f0158134f5a6d7e241b5a745c40a0105c5878070f41c1344b00752
-> extracting sha256:484f0204d4ac0412c52c0b0754b1c0f1e6d000f4e98005436f31b0b7
-> extracting sha256:c442be2be51b000bffc040c10f13de538434ccc5f5d054a5608a0100aba3f
-> [2/8] WORKDIR /app
-> [3/8] ADD . /app
-> [4/8] COPY requirements.txt /app
-> [5/8] RUN python3 -m pip install -r requirements.txt
-> [6/8] RUN python3 -m pip install lib_d
-> exporting to image
-> exporting layers
-> writing image sha256:175b7194800f002fa25dae385c322333f2f2a1b40a0124202a20f0370f10
-> 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\WK-PC\Desktop\job-portal-main>

create a docker file for the job portal application and deploy it in Docker desktop application

