

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

The screenshot shows the Docker Hub page for the `uifd/ui-for-docker` repository. The page indicates that the repository is deprecated and development continues at `portainer/portainer`. A Docker Pull Command box shows the command `docker pull uifd/ui-for-docker`.

Below the Docker Hub page, a Docker Playground session is shown. The session ID is `cd9an2u3_cd9av060qau0008hbjs0`. The IP address is `192.168.0.13`. The SSH command is `ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-with-docker.com`.

The terminal output shows the following commands and results:

```
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibility.
#
# The PWD team.
=====
(root@) (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
6411946080a8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbad244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
(root@) (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
c590dd163101ae795bdcea0eb1dd98f6fe549cb3f24daab9ff7c1931923fe0d
(root@) (local) root@192.168.0.13 ~
c
```

UI For Docker

Dashboard Containers Containers Network Images Networks Volumes Info Refresh

# UI For Docker


The UI for Docker container engine

Learn more.

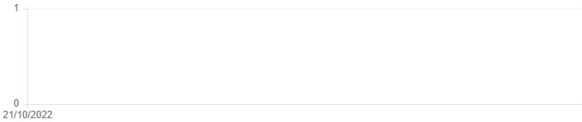
Running Containers

- beautiful\_goldwasser Up About a minute

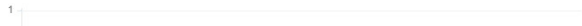
Status



Containers created



Images created

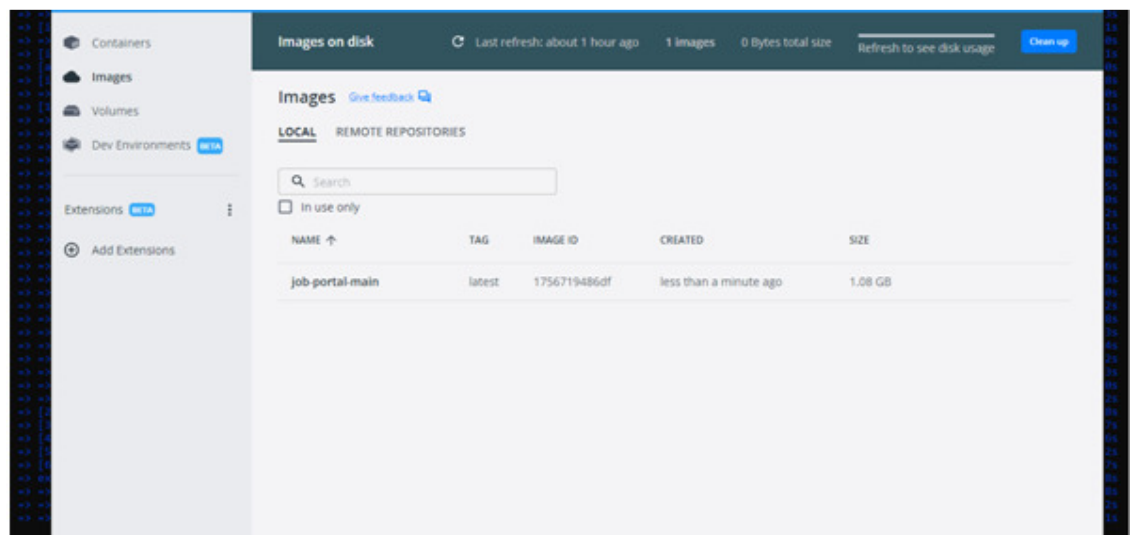


Running Stopped Ghost

21-10-2022

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: 687B
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afa88c25f6022354d547d892501807aad826a7f4a0d19d9f380a6f6c
Resolving docker.io/library/python:3.6@sha256:f8652afa88c25f6022354d547d892501807aad826a7f4a0d19d9f380a6f6c
-> sha256:f8652afa88c25f6022354d547d892501807aad826a7f4a0d19d9f380a6f6c 1.81kB / 1.86kB
-> sha256:080744807ade079d8fac31872358c2de118f8214c0448e936303b376d3668d 2.22kB / 2.22kB
-> sha256:54268638087c5e3ad24c6e21fc089abbc848a27634c089288eff71f3f44b184 9.27kB / 9.27kB
-> sha256:0e29546d541c0bd389281d21a73a8d1db78665c1095b74f323089e0b77ade13 54.92kB / 54.92kB
-> sha256:0bd29c735c1b92107d5c07a84f08736921995a296c714053a32a6b7d392315f0 5.15kB / 5.15kB
-> sha256:c1b77a034722f809ec0a7f2023a023ba85d61d4d9c0c5a99a057a0f8d06 18.87kB / 18.87kB
-> sha256:0494a4811822631c827ccac322ca463037f0805f56a010e6f15c91a0b718783 54.57kB / 54.57kB
-> sha256:0f9f74896df093fe8172f504faba5e0b4e4ab481a0fef09112efc7e4d3c7877 196.51kB / 196.51kB
-> sha256:5e3b213efc56598e7bd0602083945c164de2a37205e06ac2dad823124dc743 6.29kB / 6.29kB
-> extracting sha256:0e29546d541c0bd389281d21a73a8d1db78665c1095b74f323089e0b77ade13
-> sha256:0f9df0d5631af3ade6f7e2410f5e7859c48ed18c5478070f41c1344b096752 14.11kB / 14.11kB
-> extracting sha256:0494a4811822631c827ccac322ca463037f0805f56a010e6f15c91a0b718783
-> extracting sha256:c5b7a081722f8780c253f5823ed21ba85d610505c05205ab553748cd050
-> sha256:404f02044bac0432ca522c308f254b1c91fcea0806bfee760b0243b2f11ba07 235B / 235B
-> sha256:c4f42e2be530908bffc040c1d713de518434ccc5f5d954a5084a8169a3a3f 2.21kB / 2.21kB
-> extracting sha256:0494a4811822631c827ccac322ca463037f0805f56a010e6f15c91a0b718783
-> extracting sha256:0f9f74896df093fe8172f504faba5e0b4e4ab481a0fef09112efc7e4d3c7877
-> extracting sha256:5e3b213efc56598e7bd0602083945c164de2a37205e06ac2dad823124dc743
-> extracting sha256:0f9df0d5631af3ade6f7e2410f5e7859c48ed18c5478070f41c1344b096752
-> extracting sha256:404f02044bac0432ca522c308f254b1c91fcea0806bfee760b0243b2f11ba07
-> extracting sha256:c4f42e2be530908bffc040c1d713de518434ccc5f5d954a5084a8169a3a3f
-> [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 lib_mh
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df082fd56e385c522513f2ff2a2b49abd242b22a28a4f8379f19
-> 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>
```



## 1. Create a IBM container registry and deploy helloworld app

