

Assignment - 4

Docker and Kubernetes

Team ID : PNT2022TMID52244

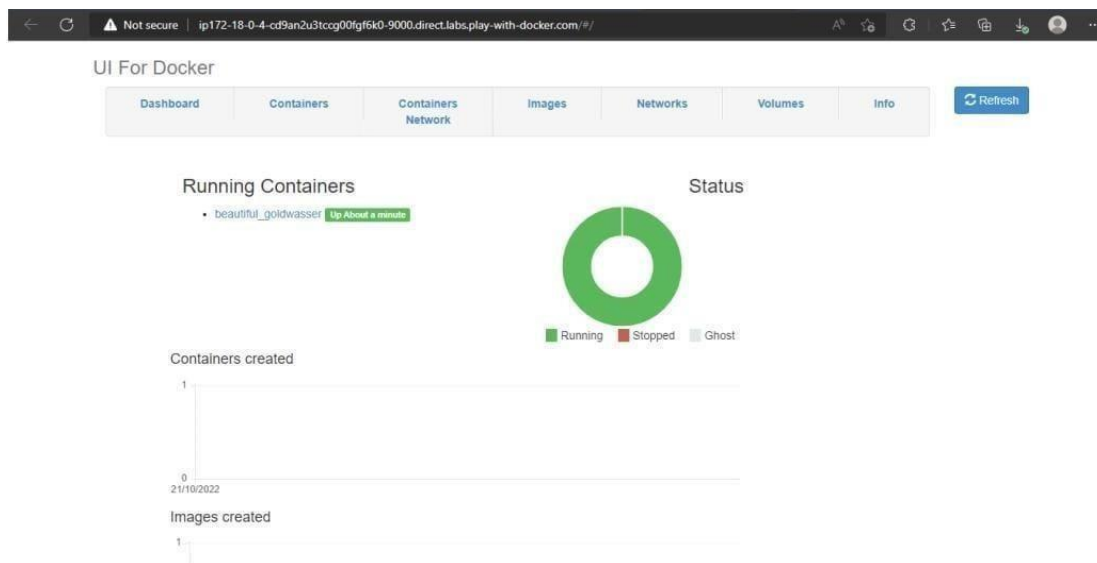
Project Name : Smart Fashion Recommender Application

Submitted by : JEEVITHAN.S(963519104027)

1. Pull an Image from docker hub and run it in docker playground:

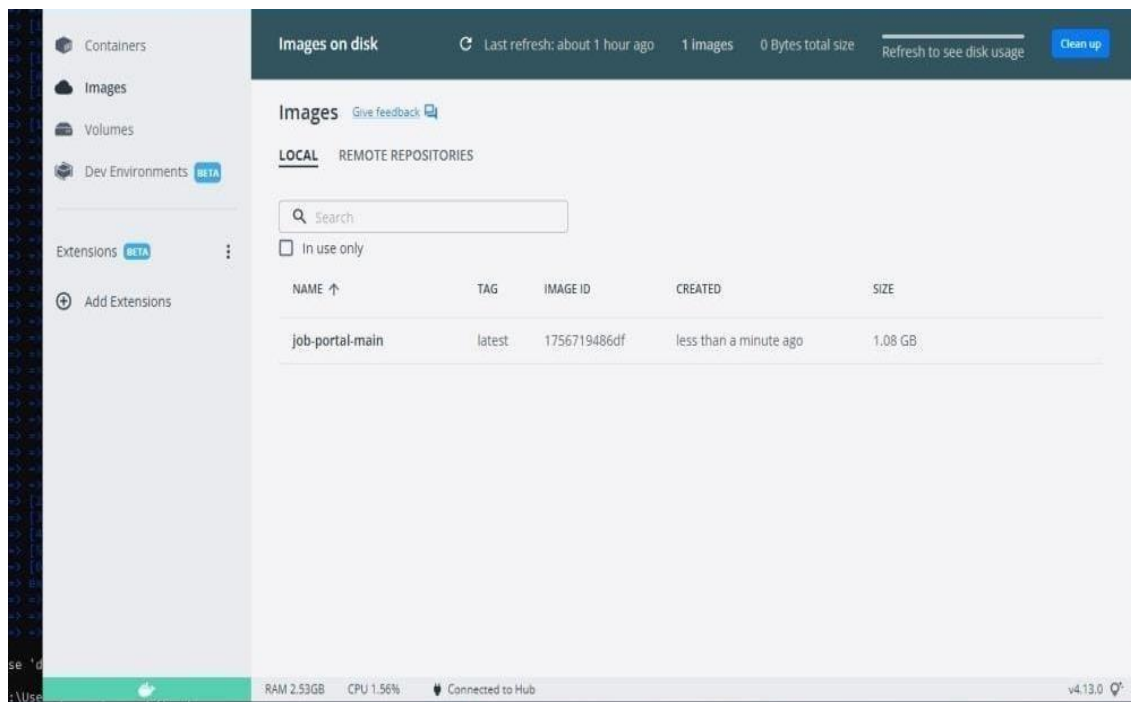
The image shows a composite screenshot of a web browser and a Docker Playground interface. The top part displays the Docker Hub page for the repository `uifd/ui-for-docker`. It includes the repository name, a star icon, and a note stating "This repo is deprecated. Development continues at: [portainer/portainer](#)". Below this, there's a section for "UI For Docker" with a description: "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". To the right, the "Docker Pull Command" is shown as `docker pull uifd/ui-for-docker`.

The bottom part of the image shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:42:30, a "CLOSE SESSION" button, and a list of instances. The main area displays the instance details for `cd9an2u3_cd9av060qau0008hbj0`, including its IP address `192.168.0.13` and an "OPEN PORT" button. Below this, there's a terminal window showing the command `docker pull uifd/ui-for-docker` being executed, followed by the output: `latest: Pulling from uifd/ui-for-docker`, `441194d080c8: Pull complete`, `Digest: sha256:fe371ff5a9549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749`, `Status: Downloaded newer image for uifd/ui-for-docker:latest`, and `docker.io/uifd/ui-for-docker:latest`. The terminal also shows the command `docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker` being executed, followed by the output: `c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dadb9ff7c1931923fc0d`.

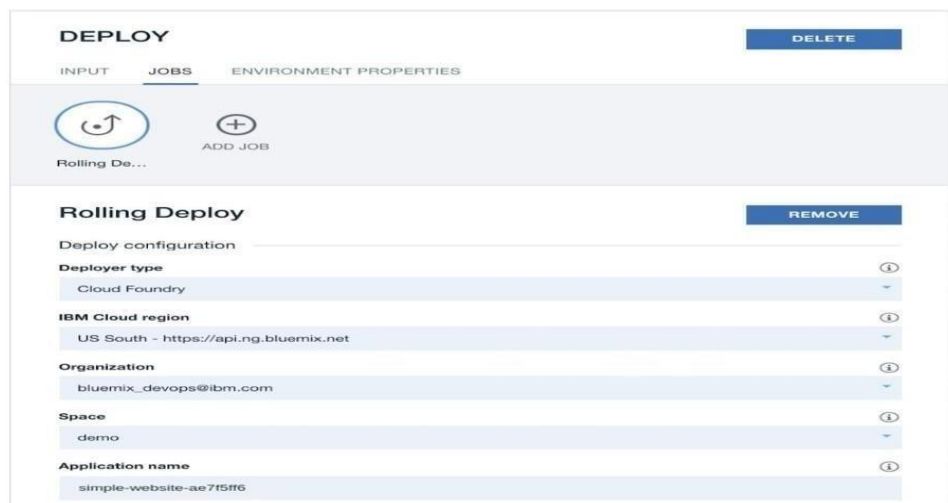


2. Create a docker file for the jobportal application and deploy it in Docker desktop application:

```
[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:f8652afaf88c25f8d22354d547d802591967aa4026a7fa9a0819df9f300af0fc
=> resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f8d22354d547d802591967aa4026a7fa9a0819df9f300af0fc
=> sha256:f8652afaf88c25f8d22354d547d802591967aa4026a7fa9a0819df9f300af0fc 1.86kB / 1.86kB
=> sha256:d897a4907d8ec870df5ac31872359c2de510f82214c0446e926303b376d3b6bd 2.22kB / 2.22kB
=> sha256:54260638d07c5e3ad24c6e21fc889abb8c486a27634c8803086ff71f3f44b184 9.27kB / 9.27kB
=> sha256:6e29546d541c0bd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB
=> sha256:9b83c73b5209207d5c07a54f60f3a92199c226c714b53a32a6c7d19231fcd 5.15MB / 5.15MB
=> sha256:cb507aee01722f070ecae3f353823ed1baa85d61d509cda95a5b3d744e0d56 10.87MB / 10.87MB
=> sha256:6494e4811622b31c027ccac322ca63037f4085f569a93a6f15c81a9d718793 54.57MB / 54.57MB
=> sha256:6f9f74890df93fa0172f594fabu85ebba08a481a0fef09112efc7e4d3c78f7 196.51MB / 196.51MB
=> sha256:5e3b1213efc56598e78bd002983045c164de2a37285e06a62dada023124dc743 6.29MB / 6.29MB
=> extracting sha256:6e29546d541c0bd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3
=> sha256:9fd0fde56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
=> extracting sha256:9b820c73b5209207d5c07a54f60f3a92199c226c714b53a32a6c7d19231fcd
=> extracting sha256:cb507aee01722f070ecae3f353823ed1baa85d61d509cda95a5b3d744e0d56
=> sha256:4f42be2be53b900ebffcc04c1df13de538434cc3f5d954a56848e106a3a3f
=> extracting sha256:4f42be2be53b900ebffcc04c1df13de538434cc3f5d954a56848e106a3a3f
=> extracting sha256:6494e4811622b31c027ccac322ca63037f4085f569a93a6f15c81a9d718793
=> extracting sha256:6f9f74890df93fa0172f594fabu85ebba08a481a0fef09112efc7e4d3c78f7
=> extracting sha256:5e3b1213efc56598e78bd002983045c164de2a37285e06a62dada023124dc743
=> extracting sha256:9fd0fde56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752
=> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fca0880bfeef0be0b243b2f31bab7
=> extracting sha256:c4f42be2be53b900ebffcc04c1df13de538434cc3f5d954a56848e106a3a3f
[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 job_db
=> exporting image
=> exporting layers
=> writing image sha256:1756719488df002fad5dae305c5221513f2ff2d3b49a8d242b22a28ef0379f19
=> naming to docker.io/library/job-portal-main
[+] 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```



3. Create a IBM container registry and deploy helloworld app or jobportalapp:



```

1  {
2    "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
3    "Params": {
4      "Target": {
5        "Url": "https://api.ng.bluemix.net",
6        "Org": "bluemix_devops@ibm.com",
7        "Space": "demo"
8      },
9      "Name": "simple-website-ae7f5ff6",
10     "Instrumentation": {}
11   },
12   "Path": "manifest.yml",
13   "Type": "Cloud Foundry"
14 }

```

Hello, IBM Cloud World!

4. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport:

