

Assignment -4

Docker and Kubernetes

Student Name	Tamilarasan K
Student Roll Number	712219104027
Maximum Marks	2 Marks

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

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a timer at 03:58:45, a 'CLOSE SESSION' button, and an 'Instances' section showing a single instance named 'node1' with IP 192.168.0.18. The main area displays the instance details for 'cdhtqrm3_cdhtqte3tccg00fmtje0', including its IP (192.168.0.18), memory usage (1.63%), CPU usage (0.16%), and an SSH command. Below this is a terminal window showing the following commands and output:

```
wait Block until one or more containers stop, then print their exit codes
Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.18 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab124dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
1c759ed73863a957fad6fba18edbe6388eed7ad0b867b0eacb15e722e10e4bb9
[node1] (local) root@192.168.0.18 ~
$
```

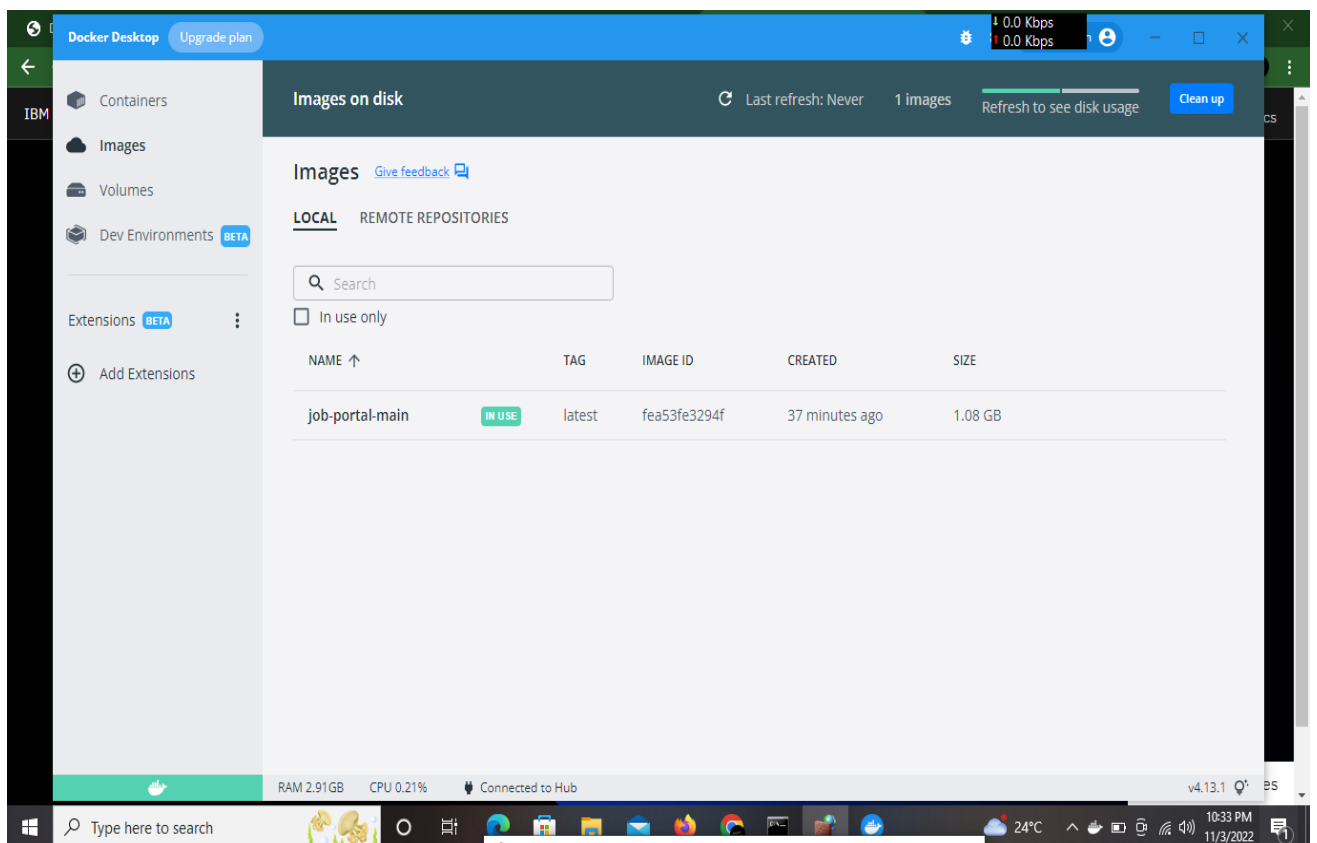
The screenshot shows the 'UI For Docker' dashboard. At the top, there's a navigation bar with tabs for Dashboard, Containers, Containers Network, Images, Networks, Volumes, and Info. Below this, the 'Running Containers' section shows a single container named 'hungry_fermat' which is 'Up 14 seconds'. To the right, there's a 'Status' section with a donut chart showing the status of containers: Running (green), Stopped (red), and Ghost (grey). Below the donut chart, there are two line graphs: 'Containers created' and 'Images created', both showing a count of 1 on 11/3/2022.

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

```
Command Prompt
=> => transferring context: 28
=> [internal] load metadata for docker.io/library/python:3.6
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
=> => resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
=> => sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
=> => sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
=> => sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
=> => sha256:54260638d07c5e3ad24c6e21fc889abb8486a27634c0892086ff7f1f3f44b104 9.27kB / 9.27kB
=> => sha256:0e29546d541cbbd389281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB
=> => sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
=> => sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
=> => sha256:6f9f74896df93fe0172f594faba85e0b4e8a0481a0fef9112efc7e4d3c78f7 196.51MB / 196.51MB
=> => sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 6.29MB / 6.29MB
=> => sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
=> => sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 2358 / 2358
=> => sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB
=> => extracting sha256:0e29546d541cbbd389281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3
=> => extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd
=> => extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56
=> => extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793
=> => extracting sha256:6f9f74896df93fe0172f594faba85e0b4e8a0481a0fef9112efc7e4d3c78f7
=> => extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743
=> => extracting sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752
=> => extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7
=> => extracting sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6169a3a3f
=> [internal] load build context
=> => transferring context: 6258
=> [auth] library/python:pull token for registry-1.docker.io
=> [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:fea53fe3294f2f702eba74475f589b629df42a18ed0e87bbddacffefc58bf00c
=> => naming to docker.io/library/job-portal-main

C:\Users\user\Desktop\job-portal-main>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
job-portal-main latest fea53fe3294f 29 minutes ago 1.08GB

C:\Users\user\Desktop\job-portal-main>
```



3.Create a IBM container registry and deploy helloworld app

The screenshot shows the IBM Cloud Container Registry interface. The left sidebar has a 'Quick start' section with 'Namespaces' highlighted, showing a count of 2. The main content area is titled 'Namespaces' and shows a table of existing namespaces. The table has columns for Name, Resource group, Repository count, Image count, and Retention policy. Two namespaces are listed: 'jobportalap' and 'jobportalapplication', both in the 'Default' resource group. The 'jobportalap' namespace has 1 repository and 1 image, while 'jobportalapplication' has 0 repositories and 0 images. Both have a retention policy of 'Retain all images'. A 'Create' button is visible in the top right of the table area.

Name	Resource group	Repository count	Image count	Retention policy
jobportalap	Default	1	1	Retain all images
jobportalapplication	Default	0	0	Retain all images

The screenshot shows the IBM Cloud Container Registry interface, now displaying the 'Repositories' page. The left sidebar has 'Repositories' highlighted, showing a count of 1. The main content area is titled 'Repositories' and shows a table of existing repositories. The table has columns for Name, Image count, Namespace, and Last updated. One repository is listed: 'testrepo' in the 'jobportalap' namespace, with 1 image and updated 3 days ago. The repository path is 'icr.io/jobportalap/testrepo'. A 'Create' button is visible in the top right of the table area.

Name	Image count	Namespace	Last updated
testrepo icr.io/jobportalap/testrepo	1	jobportalap	3 days ago

```
Command Prompt - docker run -p 4000:4000 icr.io/jobportalap/testrepo@sha256:ef3608d0cf4c21ce5e9c5e42caa10349538d25697a6e8660476a5b5558c89898
11936051f93b: Layer already exists
jobportaltest: digest: sha256:ef3608d0cf4c21ce5e9c5e42caa10349538d25697a6e8660476a5b5558c89898 size: 3259

C:\Users\user> ibmcloud cr image-list
Listing images...

Repository      Tag              Digest           Namespace        Created          Size    Security status
icr.io/jobportalap/testrepo  jobportaltest    ef3608d0cf4c     jobportalap      3 days ago      435 MB  -

OK

C:\Users\user> docker run -p 5070:5070 icr.io/jobportalap/testrepo
Unable to find image 'icr.io/jobportalap/testrepo:latest' locally
docker: Error response from daemon: manifest for icr.io/jobportalap/testrepo:latest not found: manifest unknown: manifest unknown.
See 'docker run --help'.

C:\Users\user> docker pull icr.io/jobportalap/testrepo
Using default tag: latest
Error response from daemon: manifest for icr.io/jobportalap/testrepo:latest not found: manifest unknown: manifest unknown

C:\Users\user> docker run -p 4000:4000 icr.io/jobportalap/testrepo@sha256:ef3608d0cf4c21ce5e9c5e42caa10349538d25697a6e8660476a5b5558c89898
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```

4. Create a Kubernetes cluster in IBM cloud and deploy hello world image and also expose the same app to run in node port.

The screenshot displays the IBM Cloud Kubernetes Dashboard for a cluster named 'mycluster-free'. The cluster is currently in a 'Preparing master, workers...' state. A prominent banner at the top indicates that the cluster 'Expires in 30 days' and advises backing up data. The main content area shows the following status: 1 of 1 Node (Pending), 0 of 0 Add-ons (Normal), Master status is Unknown, and Ingress status is Pending. Below this, a 'Details' section provides specific information: Cluster ID is cdk2eopf0u0ekek, Version is 1.24.7_1542, Infrastructure is Classic, Zones are Milan 01, Created on 11/7/2022 at 3:00 AM, and Image security enforcement is currently disabled with an 'Enable' button. The left sidebar offers navigation options including Overview, Worker nodes, Worker pools, and DevOps. The right sidebar contains a 'Help' section with links to log in, deploy, expose, and manage the cluster.

mycluster-free - IBM Cloud

Inbox (10) - devamaha632@...

IBM-Project-40862-1660636

Getting Started with Kubernetes

Install and Set Up kubect...

cloud.ibm.com/kubernetes/clusters/cdk2eopf0u0ekek974e0/nodes

IBM Cloud

Search resources and products...

Catalog

Manage

Mahadevan M's Account

Clusters / mycluster-free

Help

Kubernetes dashboard

Actions...

Expires in 30 days

Add tags

Overview

Worker nodes

Worker pools

DevOps New

Search

Name	Status	Worker pool	Zone	Private IP	Public IP
000000d9	Normal	default	Milan 01	10.144.194.66	169.51.205.176

Items per page: 251-1 of 1 item11 of 1 page

Help

Log in to your cluster

Deploy your app

Expose your app

Add storage to your app

Connect integrations

Install add-ons

Troubleshoot

Type here to search

23°C Mostly clear

3:16 AM 11/7/2022