

ASSIGNMENT – 4

KUBERNETES AND DOCKER

Assignment Date	22 October 2022
Student Name	Ms. R SHARRANYA
Student Roll Number	960519106059
Maximum Marks	2 Marks

Question – 1:

Pull an image from Dockers hub and run it in Dockers playground.

Solution:

app.py:

```
from flask import Flask
import os
app = Flask(__name__)

@app.route("/")
def home():
    return "Welcome To Docker Playground"

if __name__ == "__main__":
    port = int(os.environ.get('PORT',5000))
    app.run(host='0.0.0.0',port=port)
```

Requirements.txt:

```
python
flask
tqdm
colorama
numpy
```

To-do.txt:

```
download python
install python
install flask pip install -r requirements.txt
run app -python app.py
```

```
[node1] (local) root@192.168.0.13 ~  
$ docker pull hello-world  
Using default tag: latest  
latest: Pulling from library/hello-world  
Digest: sha256:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7  
Status: Image is up to date for hello-world:latest  
docker.io/library/hello-world:latest  
[node1] (local) root@192.168.0.13 ~  
$ docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

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

Question – 2:

Create a Dockerfile for the Jobportal application and deploy it in docker desktop application.

Solution:

app.py

```
from flask import Flask,render_template
import os

doc_desk = Flask(__name__)

@doc_desk.route("/")
def index():
    return render_template("index.html")

if __name__=="__main__":
    port =os.environ.get("PORT",5000)
    doc_desk.run( host="0.0.0.0",port=port)
```

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index</title>
    <style>
        body{
            background-color: black;

        }
        #container{
            background-color: aqua;
            margin-left:400px;
            margin-top: 100px;
            height: 100px;
            width: 400px;
            color:black ;
            font-family: cursive;
            text-align: center;
```

```

padding-top: 30px;

    }
</style>
</head>
<body >
  <h1 id="container">Welcome to Docker</h1>
  {% comment %} <script>
    function getRandomnum(maxNum){
      return Math.floor(Math.random()*maxNum)

    }
    const getRandomColor =()=>{
      const h =getRandomnum(360)
      const s = getRandomnum(100)
      const l= getRandomnum(100)
      //"hsl("+h+", "+s+"%, "+l+"%)"
      return `hsl(${h}deg,${s}%,${l}%)`
    }
    const setBackgroundColor = ()=>{
      const randomColor =getRandomColor()
      document.getElementById("container").style.backgroundColor=randomColor
    }
    setInterval(()=>{
      setBackgroundColor()
    },1500);
  </script> {% endcomment %}
</body>
</html>

```

Dockerfile:

```

FROM python
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
EXPOSE 5000

```

requirements.txt:

```
flask
```

Output:

The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Extensions (marked BETA), and Add Extensions. The main panel is titled 'Images on disk' and shows a summary: 'Last refresh: about 16 hours ago', '3 images', and '0 Bytes total size'. Below this is a section for 'Images' with a 'Give feedback' link. There are tabs for 'LOCAL' and 'REMOTE REPOSITORIES'. A search bar is present. A checkbox for 'In use only' is visible. A table lists the local images:

NAME ↑	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	716a6b564ec2	25 days ago	258.42 MB

Terminal:

The screenshot shows a terminal window with the following content:

```
PS D:\IBM\Assignment 4\Docker_Desktop> docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
hello-world         latest      716a6b564ec2  3 weeks ago   258MB
sharrah01/new       2.0        716a6b564ec2  3 weeks ago   258MB
icr.io/sharanyatest/hello s1         716a6b564ec2  3 weeks ago   258MB
PS D:\IBM\Assignment 4\Docker_Desktop> docker run -p 5000:5000 hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (windows-amd64, nanoserver-1809)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run a Windows Server container with:
PS C:\> docker run -it mcr.microsoft.com/windows/servercore:1809 powershell

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

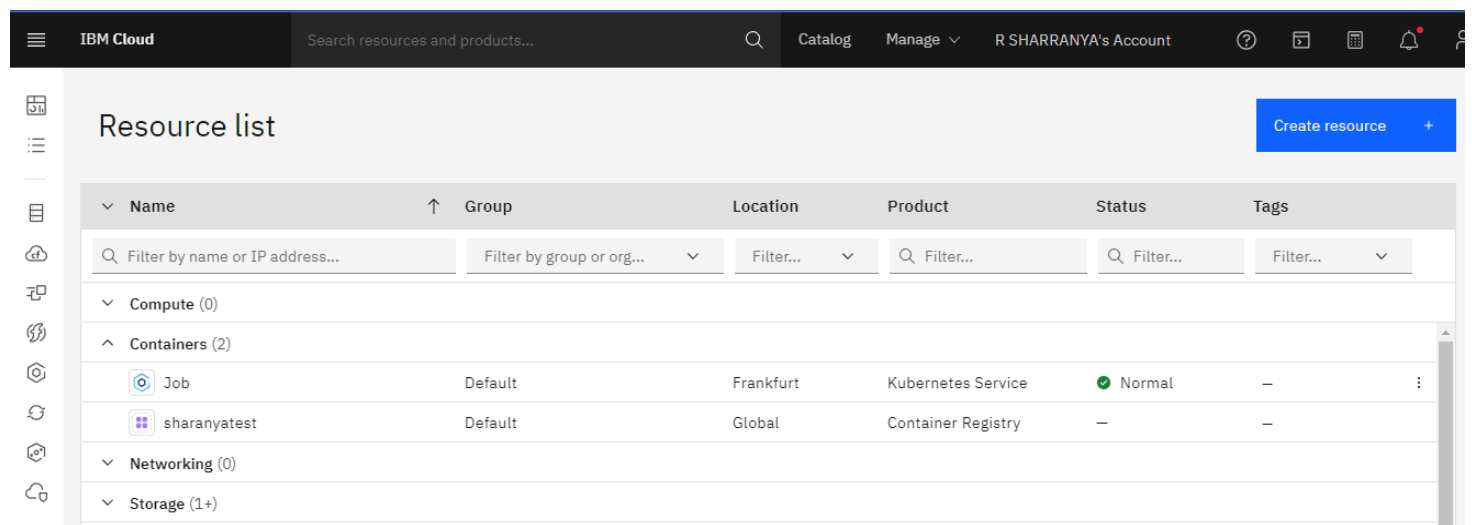
PS D:\IBM\Assignment 4\Docker_Desktop>
```

Question-3:

Create IBM container registry and deploy helloworld or jobportal app

Solution:

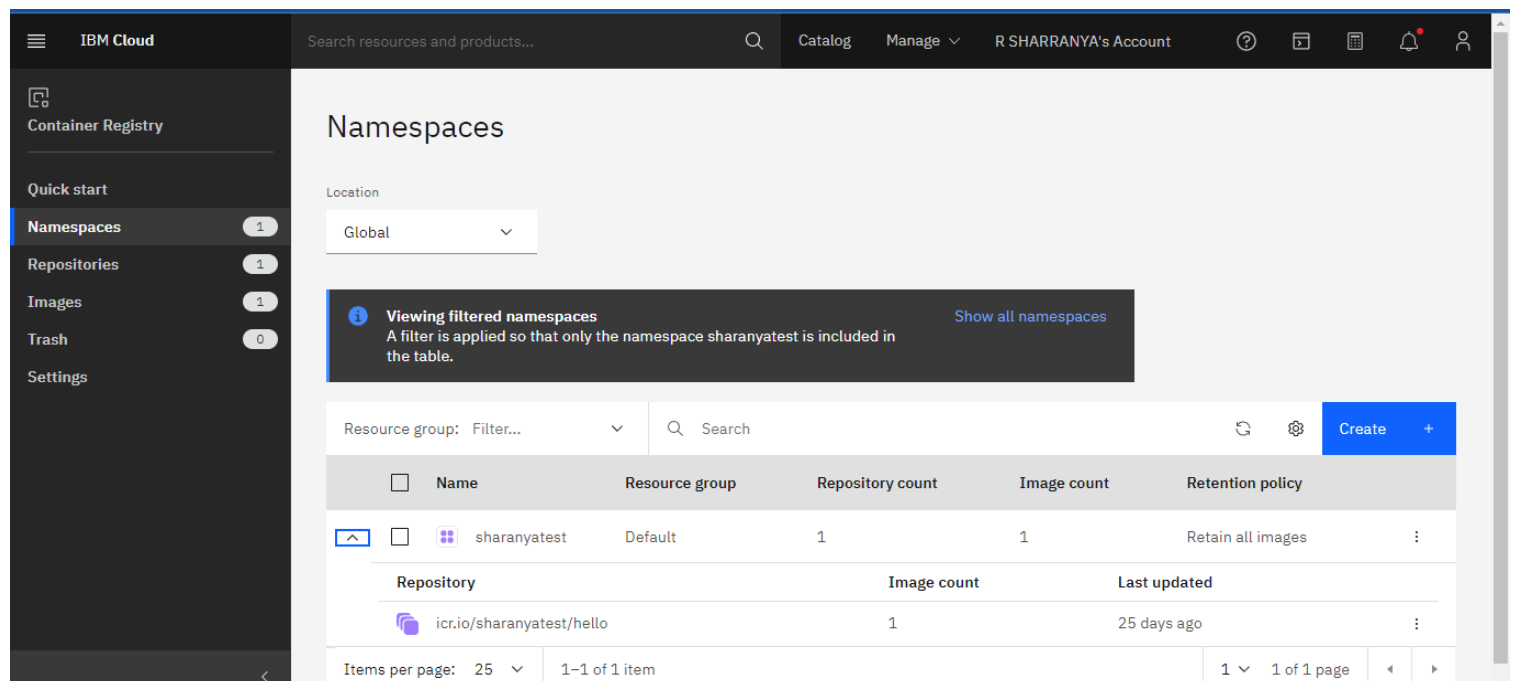
Resource List:



The screenshot shows the IBM Cloud Resource List page. The 'Containers' section is expanded, showing two resources: 'Job' and 'sharanyatest'. The 'Job' resource is a Kubernetes Service located in Frankfurt, and the 'sharanyatest' resource is a Container Registry located globally. Both resources have a status of 'Normal'.

Name	Group	Location	Product	Status	Tags
Job	Default	Frankfurt	Kubernetes Service	Normal	—
sharanyatest	Default	Global	Container Registry	—	—

Container Registry:



The screenshot shows the IBM Cloud Container Registry 'Namespaces' page. The 'Global' location is selected. A message indicates that a filter is applied, showing only the 'sharanyatest' namespace. The table below lists the namespaces and their details.

Name	Resource group	Repository count	Image count	Retention policy
sharanyatest	Default	1	1	Retain all images

Repository	Image count	Last updated
icr.io/sharanyatest/hello	1	25 days ago

Pulled Image:

IBM Cloud

Container Registry

Quick start

Namespaces1

Repositories1

Images1

Trash0

Settings

Search resources and products...

CatalogManage

R SHARRANYA's Account

?

Images

Location

Global

View by: Digest

Search

Create

Repository@digest	Tags	Manifest type	Created	Size	Security status
<div><div></div><div>sharanyatest/hello@sha256:314cc0309465...</div></div>	<div>s1</div> <div></div>	Docker	25 days ago	103 MB	<div>Unscanned</div> <div></div>

Items per page: 25

1-1 of 1 item

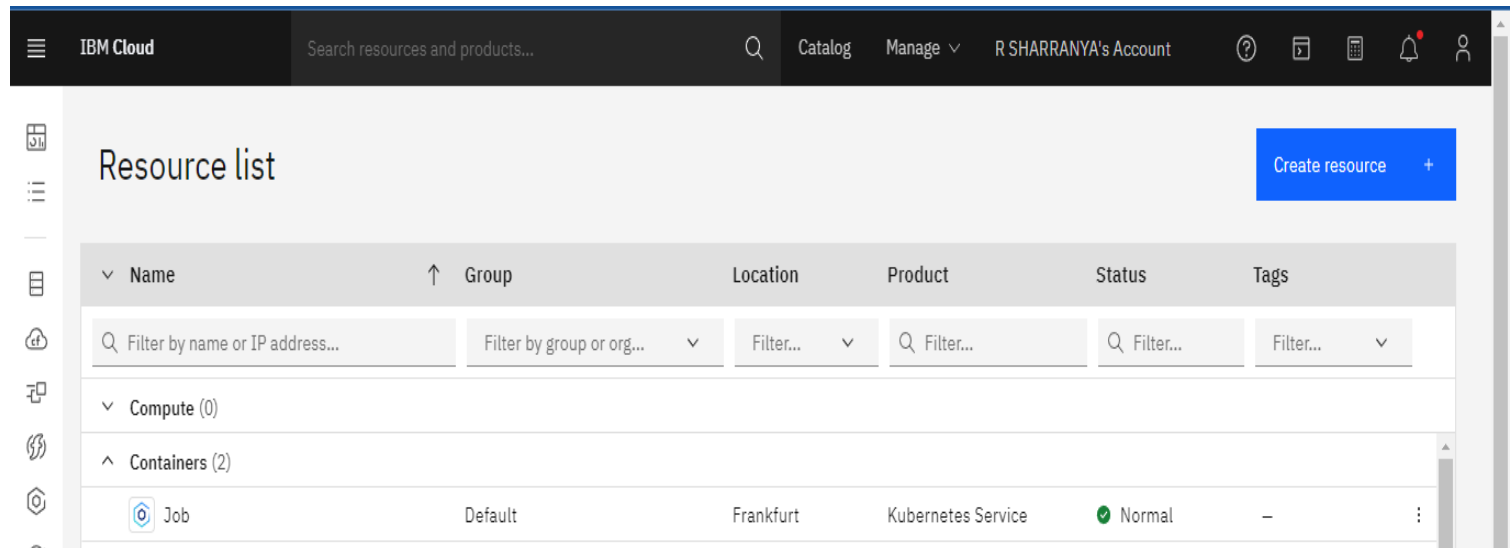
1

1 of 1 page

Question-4:

Create Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in node port.

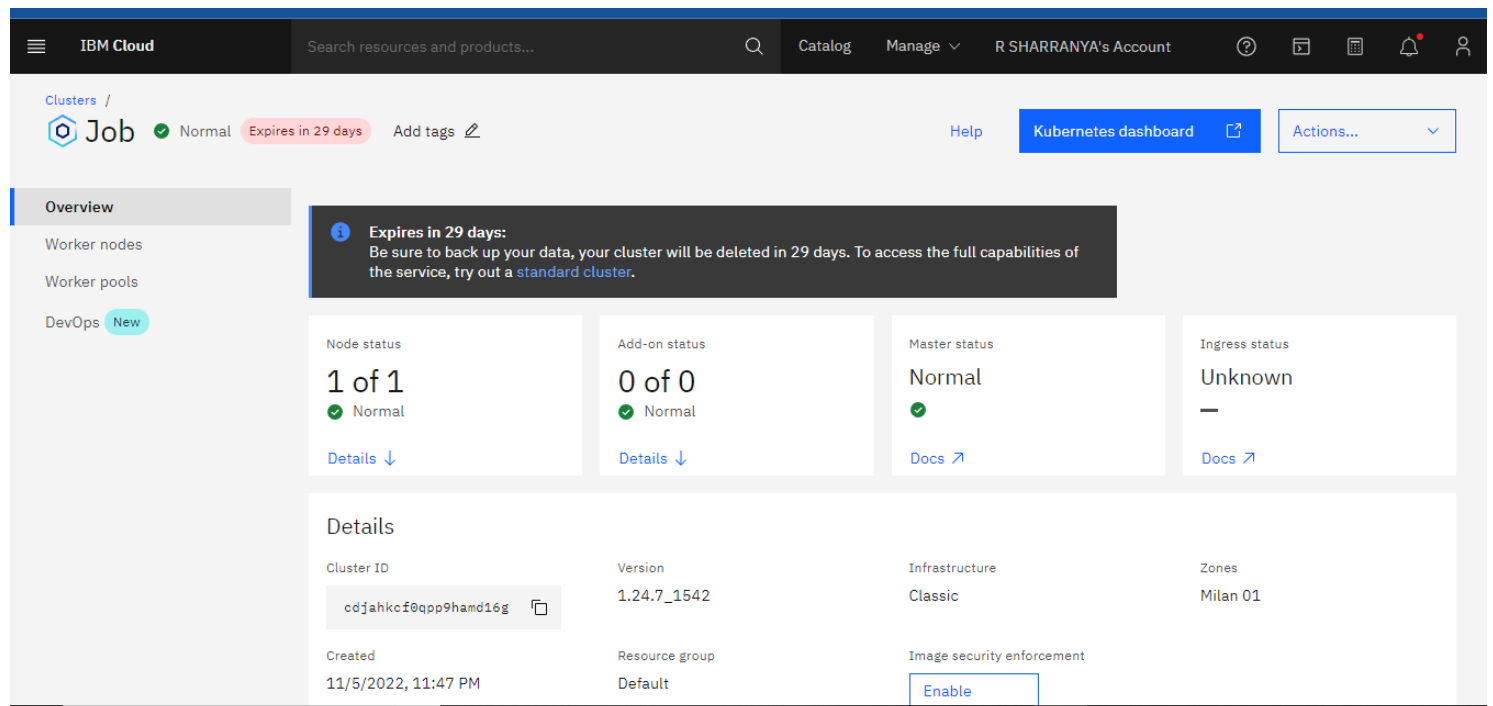
Resource List:



The screenshot shows the IBM Cloud 'Resource list' page. The top navigation bar includes the IBM Cloud logo, a search bar, and links for Catalog, Manage, and the user's account (R SHARRANYA's Account). The main content area is titled 'Resource list' and features a 'Create resource' button. Below the title is a table with columns: Name, Group, Location, Product, Status, and Tags. The table has a search bar and filter options for each column. The 'Containers' section is expanded, showing a list of resources. One resource is visible: 'Job' in the 'Default' group, located in 'Frankfurt', with the product 'Kubernetes Service' and status 'Normal'.

Name	Group	Location	Product	Status	Tags
Job	Default	Frankfurt	Kubernetes Service	Normal	-


Kubernetes:




The screenshot shows the IBM Cloud Kubernetes dashboard. The top navigation bar includes the IBM Cloud logo, a search bar, and links for Catalog, Manage, and the user's account (R SHARRANYA's Account). The main content area is titled 'Clusters / Job' and features a 'Kubernetes dashboard' button. Below the title is a section for 'Overview' with a warning banner: 'Expires in 29 days: Be sure to back up your data, your cluster will be deleted in 29 days. To access the full capabilities of the service, try out a standard cluster.' The 'Overview' section includes four status cards: 'Node status' (1 of 1, Normal), 'Add-on status' (0 of 0, Normal), 'Master status' (Normal), and 'Ingress status' (Unknown). Below these cards is a 'Details' section with a table of cluster information.



Cluster ID	Version	Infrastructure	Zones
cdjahkcf0qpp9hamd16g	1.24.7_1542	Classic	Milan 01

Created: 11/5/2022, 11:47 PM
Resource group: Default
Image security enforcement: [Enable](#)

kubernetes

default

 Search

+

Cluster > Events

Cluster Role Bindings

Cluster Roles

Events N

Namespaces

Network Policies N

Nodes

Persistent Volumes

Role Bindings N

Roles N

Service Accounts N


Custom Resource Definitions

Settings


About



Events

Name	Reason	Message	Source	Object	Count	First Seen	Last Seen
sharrah-656bb45bfc-w6p74.1724ec3424300	BackOff	Back-off pulling image "sharrah01/new:hello"	kubelet 10.144.186.145	Pod/sharrah-656bb45bfc-w6p74	984	3 hours ago	9 seconds ago
hello-7748bb5c6d-ncnmr.1724ec53aa0c1e	Pulled	Container image "sharrah01/new:2.0" already present on machine	kubelet 10.144.186.145	Pod/hello-7748bb5c6d-ncnmr	1019	3 hours ago	2 minutes ago
sharranya.1724eae493f	EnsuringLoadBalancer	Ensuring load balancer	service-controller	Service/sharranya	55	4 hours ago	3 minutes ago
sharranya.1724eae49e3	CreatingCloudProvider	Error on cloud load balancer a89268663632c4ba38666d4054108528 for service default/sharranya with UID 89268663-632c-4ba3-8666-d4054108528d: Failed to get available cloud provider IPs for load balancer services: Clusters with one node must use services of type NodePort. See https://ibm.biz/ib-debug(IKS) or https://ibm.biz/oc-lb-debug (Openshift) for more details.	ibm-cloud-provider	Service/sharranya	55	4 hours ago	3 minutes ago
sharrah-cdccbfc-ctmq.1724ec438c920f	BackOff	Back-off pulling image "sharrah01/new:hello"	kubelet 10.144.186.145	Pod/sharrah-cdccbfc-ctmq	959	3 hours ago	4 minutes ago
sharrah-1724ec322a5fa	EnsuringLoadBalancer	Ensuring load balancer	service-controller	Service/sharrah	50	3 hours ago	4 minutes ago

kubernetes

default

 Search

+

Workloads > Pods

Workloads N

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses N

Ingress Classes

Services N

Config and Storage

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
hello-7748bb5c6d-ncnmr	Show all	Show all	10.144.186.145	CreateContainerComplete	0	-	-	3 hours ago
sharrah-cdccbfc-ctmq	Show all	Show all	10.144.186.145	ImagePullBackOff	0	-	-	3 hours ago
sharrah-656bb45bfc-w6p74	Show all	Show all	10.144.186.145	ImagePullBackOff	0	-	-	3 hours ago

Workloads

Workloads N

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses N
- Ingress Classes
- Services N

Config and Storage

