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.getElementaryById("container").style.backgroundColor=randomColor
        }
        setInterval(()=>{
            setBackgroundColor()
        },1500);
    </script> {% endcomment %}
</body>
</html>
```

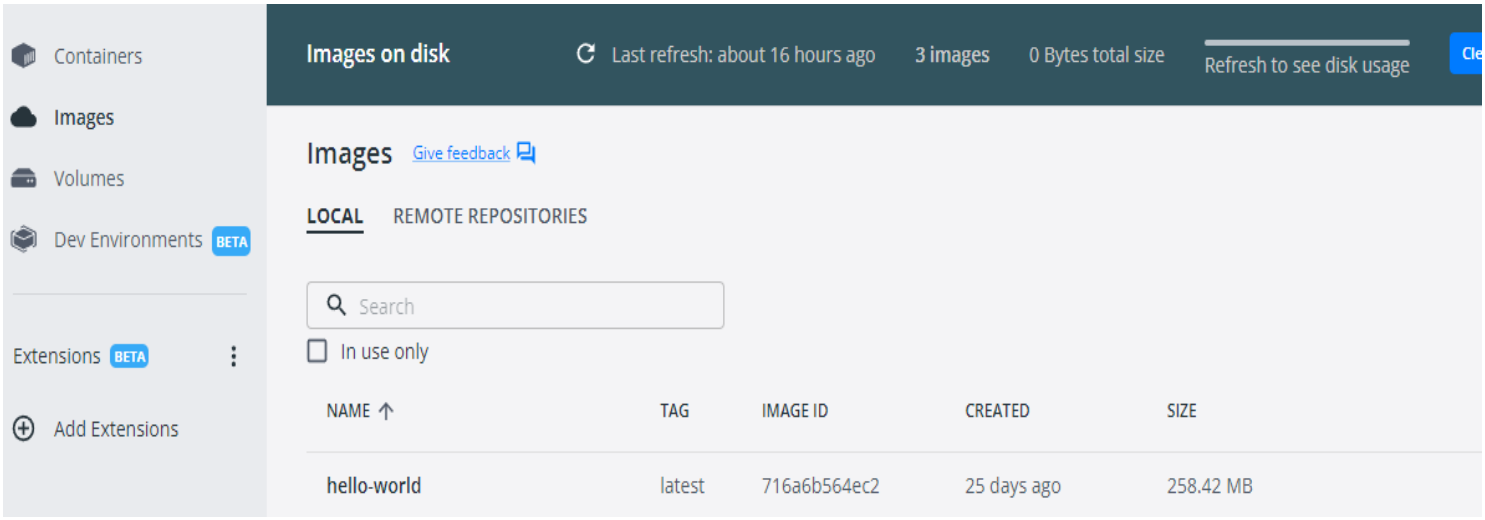
Dockerfile:

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

requirments.txt:

```
flask
```

Output:



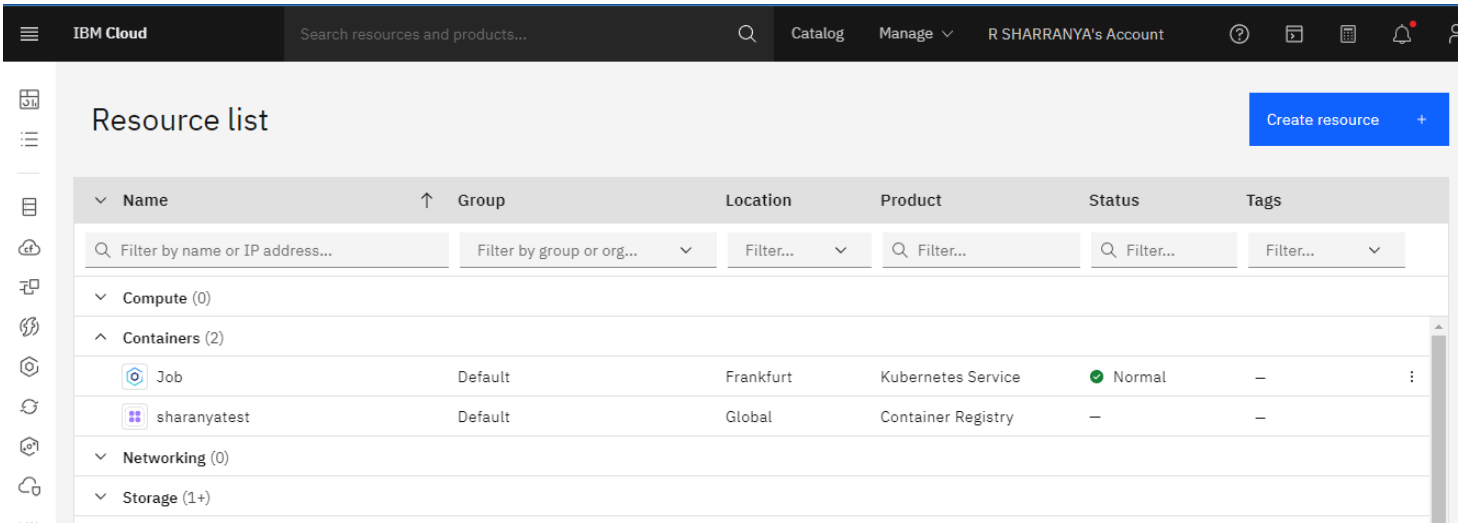
Terminal:

Question-3:

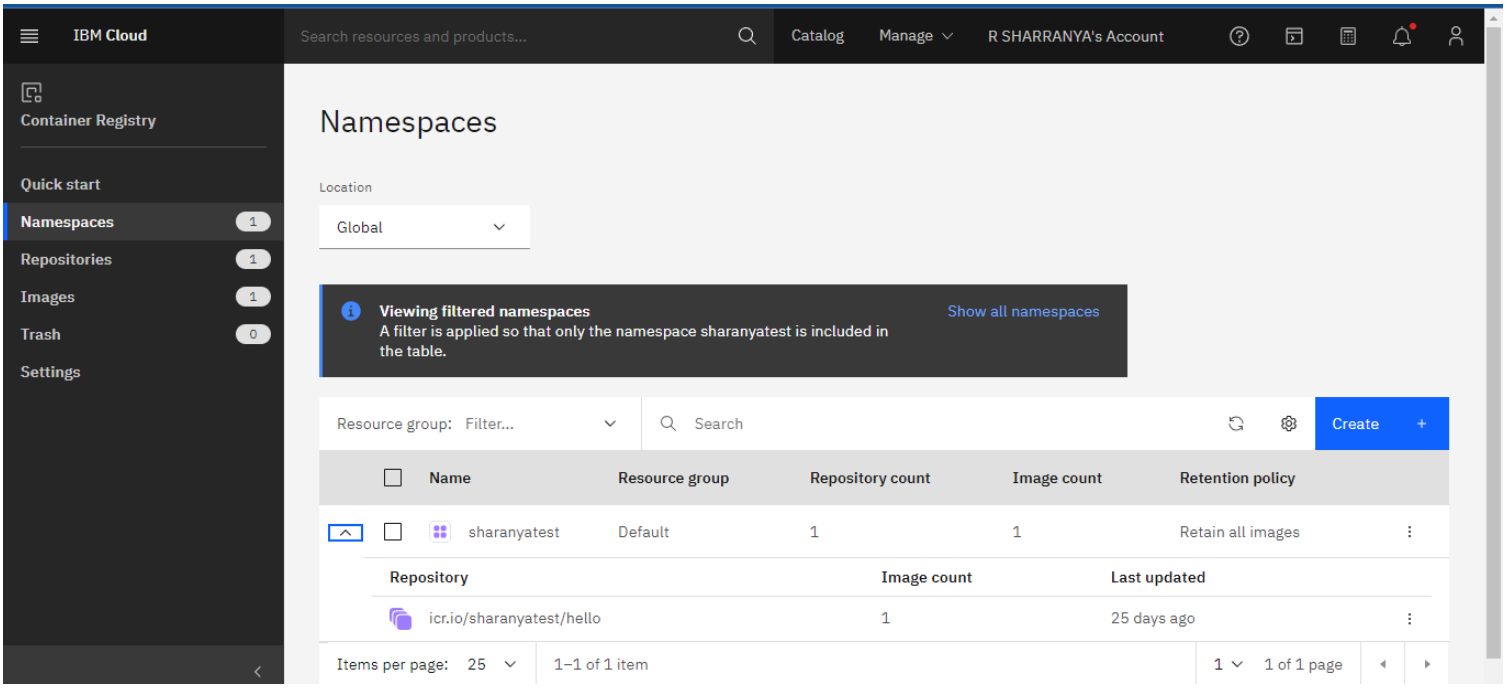
Create IBM container registry and deploy helloworld or jobportal app

Solution:

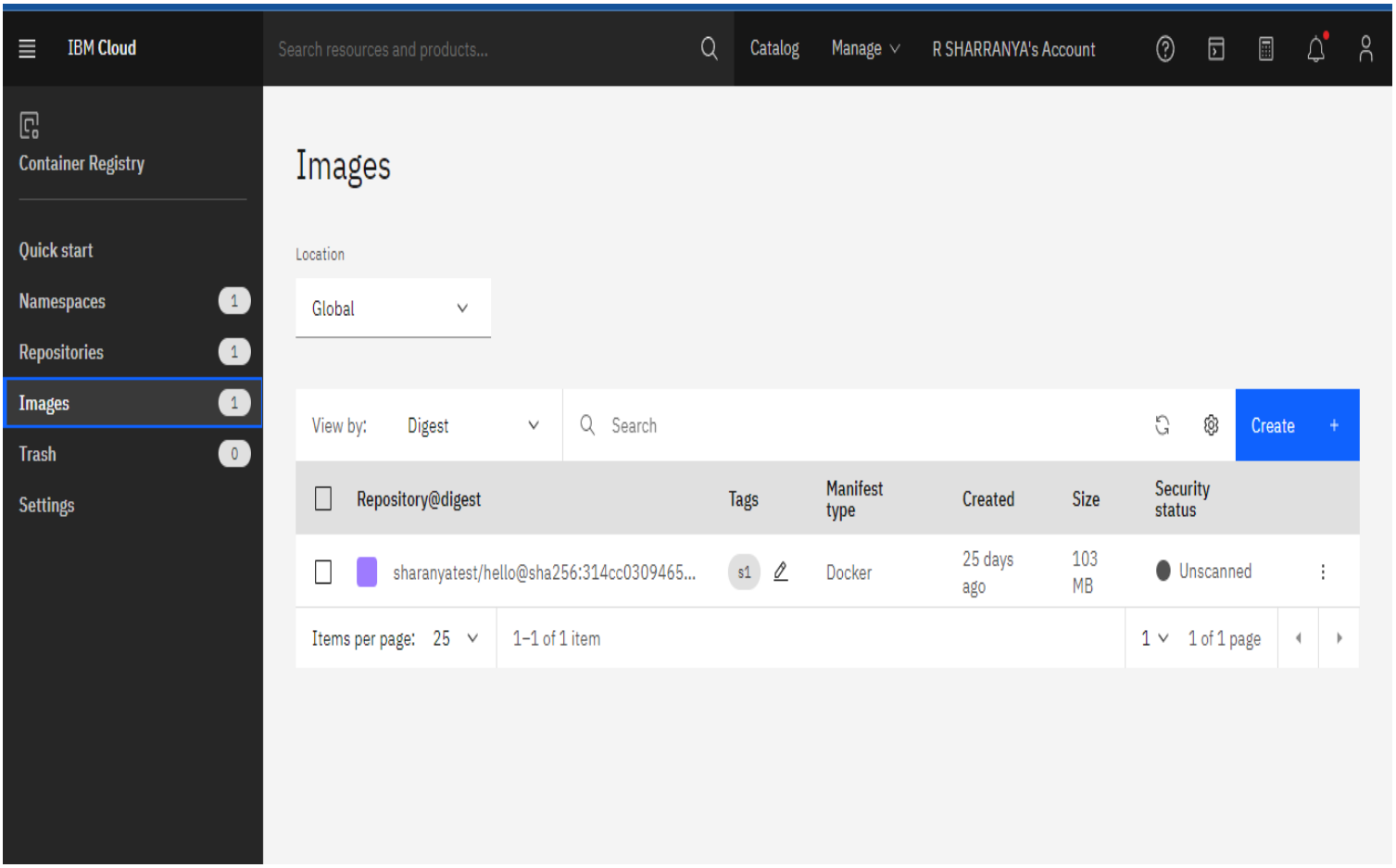
Resource List:



Container Registry:



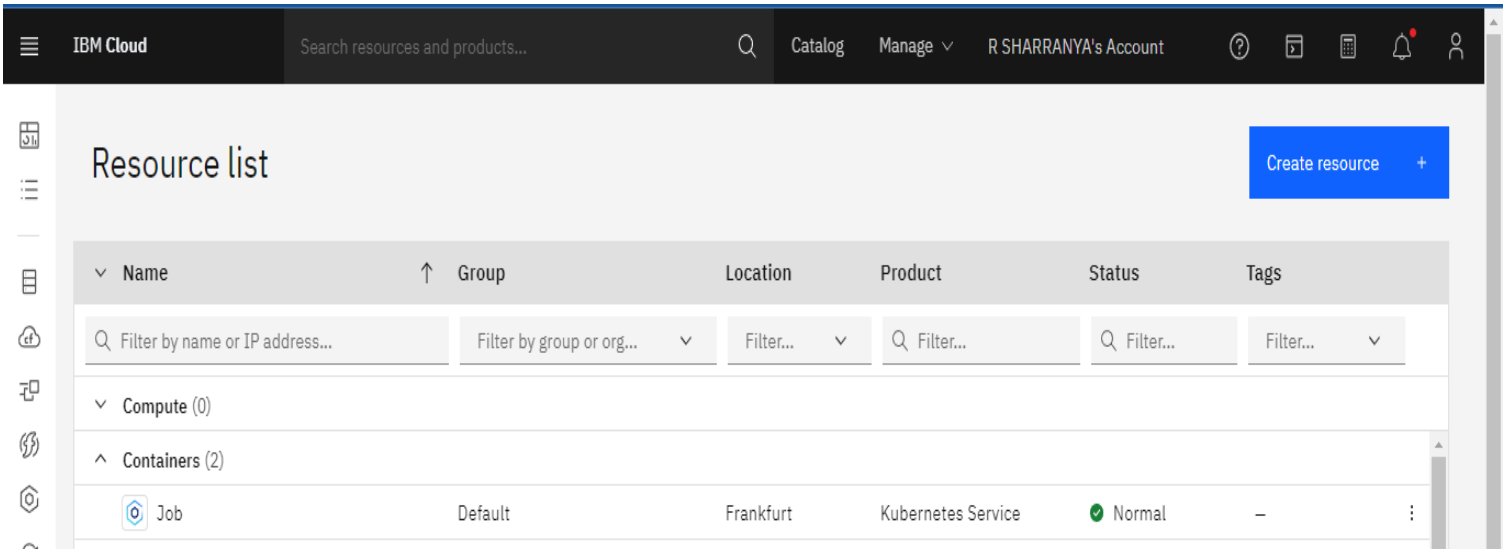
Pulled Image:



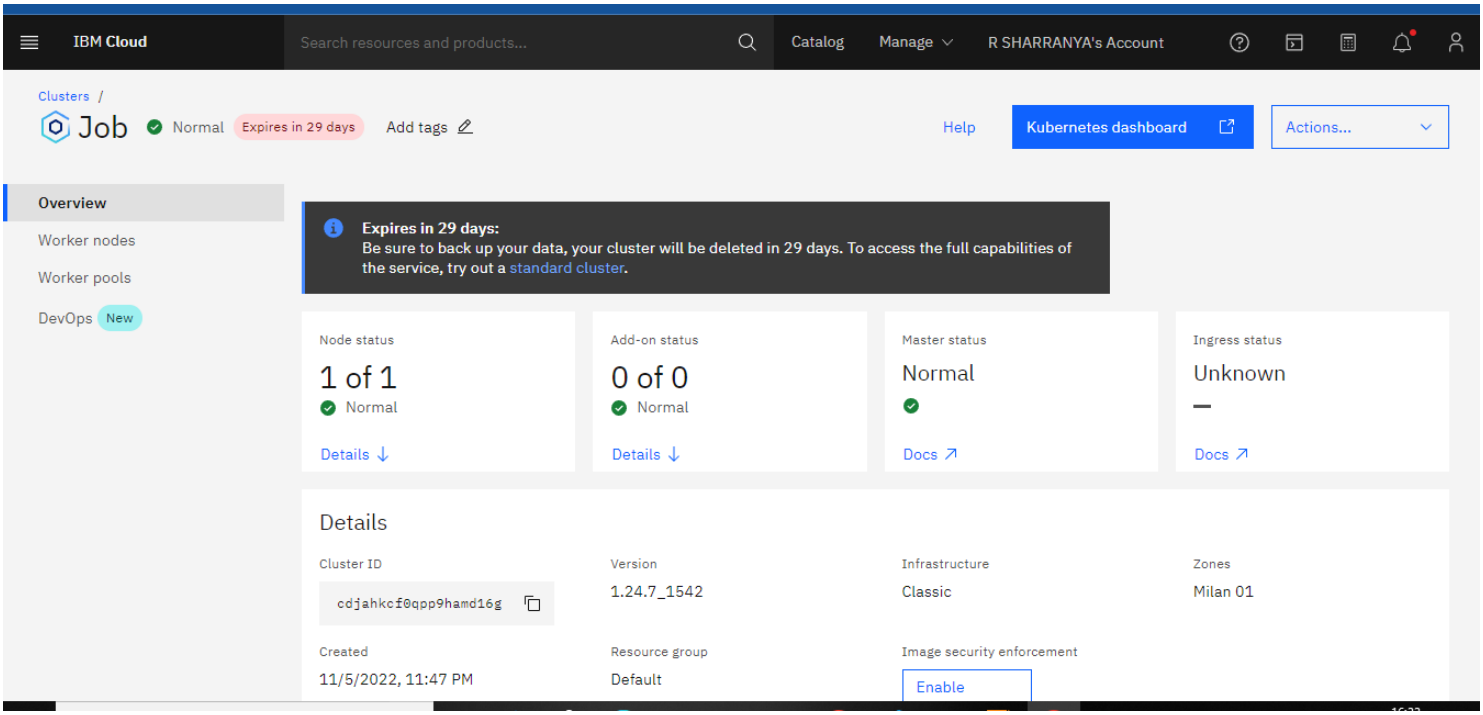
Question-4:

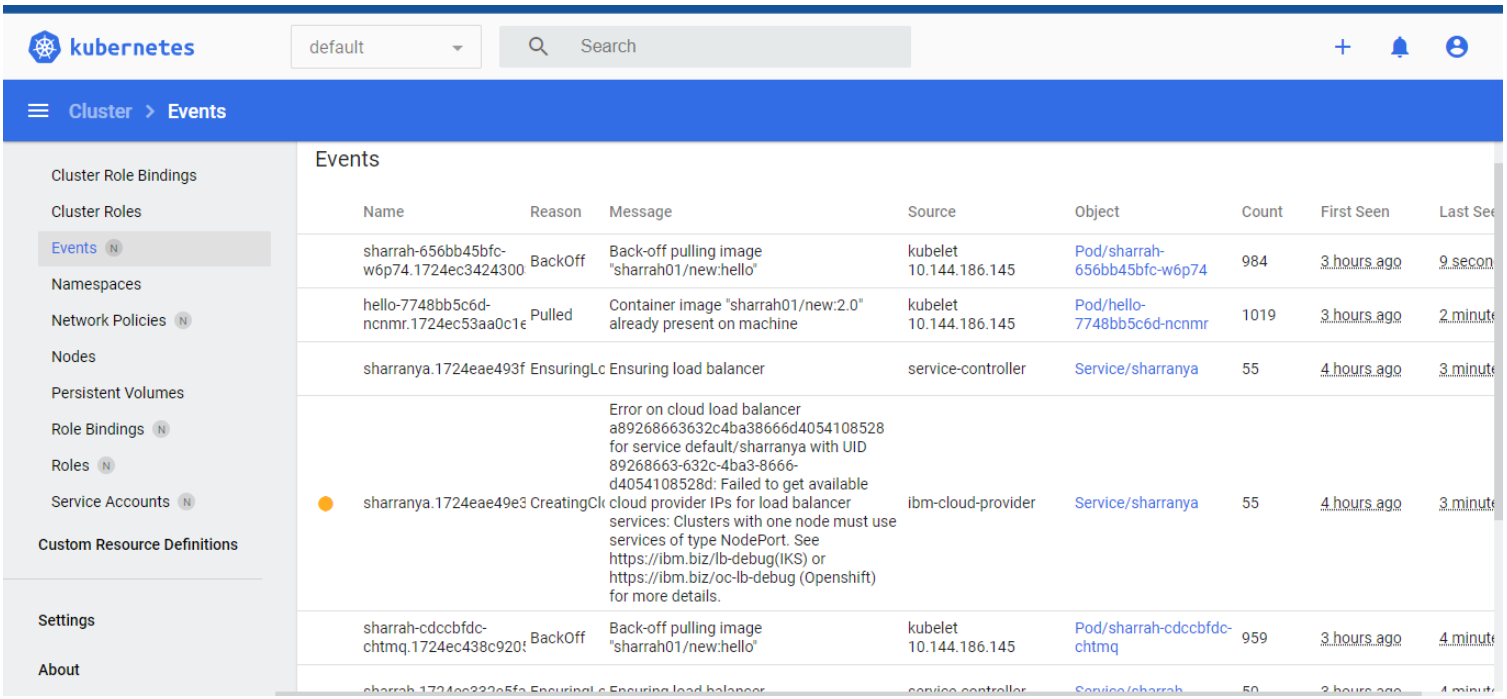
Create Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in node port.

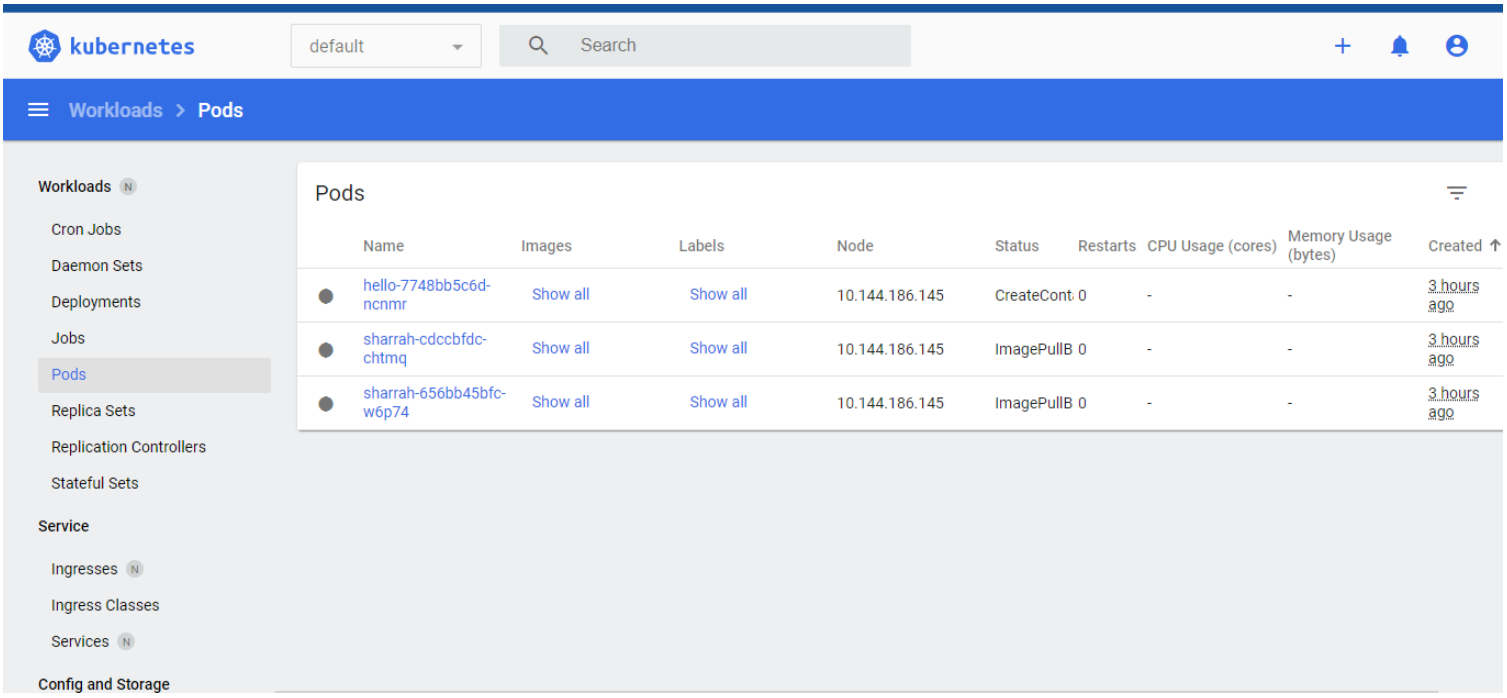
Resource List:

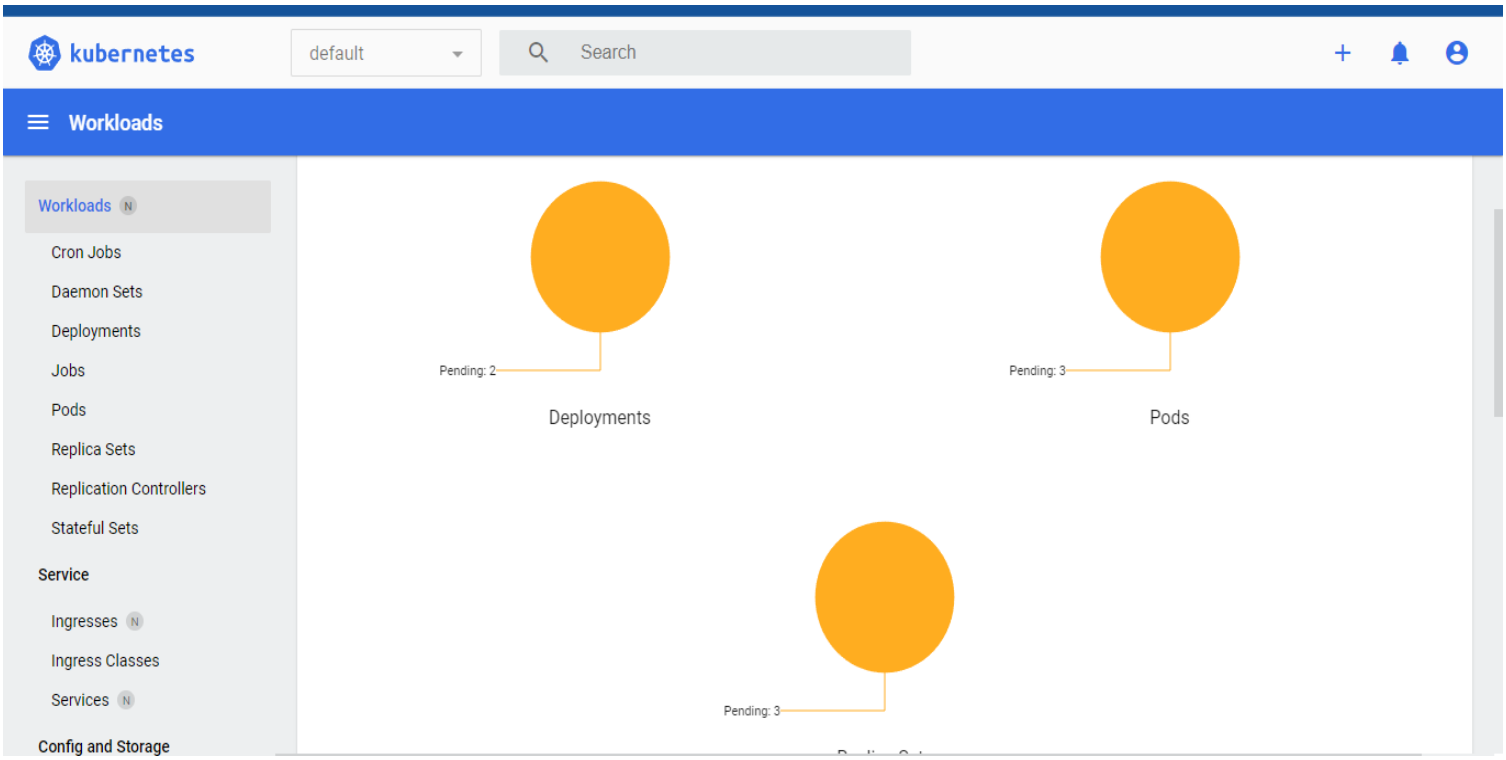


Kubernetes:




kubernetes
default
Search
Cluster > Events
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Network Policies
Nodes
Persistent Volumes
Role Bindings
Roles
Service Accounts
Custom Resource Definitions
Settings
About
Events
Name
Reason
Message
Source
Object
Count
First Seen
Last See
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 secon
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 minute
sharranya.1724eae493f
EnsuringLc
Ensuring load balancer
service-controller
Service/sharranya
55
4 hours ago
3 minute
sharranya.1724eae49e3
CreatingCl
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/lb-debug(IKS) or https://ibm.biz/oc-lb-debug (Openshift) for more details.
ibm-cloud-provider
Service/sharranya
55
4 hours ago
3 minute
sharrah-cdccbfdc-chtmq.1724ec438c920
BackOff
Back-off pulling image "sharrah01/new:hello"
kubelet 10.144.186.145
Pod/sharrah-cdccbfdc-chtmq
959
3 hours ago
4 minute


kubernetes
default
Search
Workloads > Pods
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
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
CreateCont
0
-
-
3 hours ago
sharrah-cdccbfdc-chtmq
Show all
Show all
10.144.186.145
ImagePullB
0
-
-
3 hours ago
sharrah-656bb45bfc-w6p74
Show all
Show all
10.144.186.145
ImagePullB
0
-
-
3 hours ago


kubernetes
default
Search
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
Pending: 2
Deployments
Pending: 3
Pods
Pending: 3