

Assignment - 4

Kubernetes Assignment

Assignment Date	19-10-2022
Student Name	GOKUL. S
Student Roll Number	7179KCTKCTKCTKCTKCTKCT19BCS033
Maximum Marks	2 Marks

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

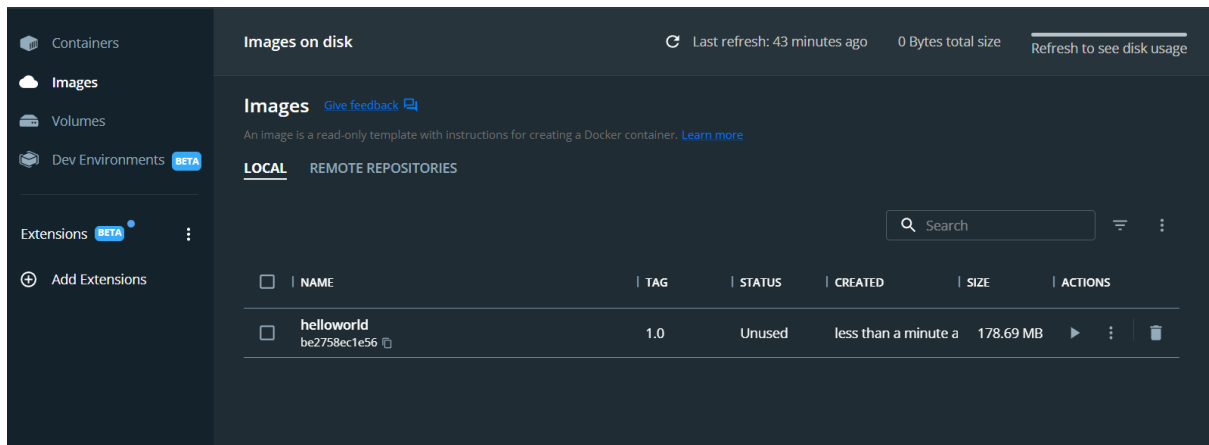
```
[node1] (local) root@192.168.0.8 ~
$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest
[node1] (local) root@192.168.0.8 ~
$ docker run alpine
[node1] (local) root@192.168.0.8 ~
$
```

2. Create a docker file for the hello world application and deploy it in Docker desktop application.

Building the image from the file: docker build -t helloworld:0.1 .

```
Command Prompt
C:\Users\Lenovo\Desktop\HelloWorld>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
C:\Users\Lenovo\Desktop\HelloWorld>docker build -t helloworld:1.0 .
[+] Building 9.5s (17/17) FINISHED
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 32B 0.0s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 2B 0.0s
=> resolve image config for docker.io/docker/dockerfile:1 2.6s
=> [auth] docker/dockerfile:pull token for registry-1.docker.io 0.0s
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:9ba7531bd80fb0a858632727cf7a112fbfd19b17e94c4e84ced81e24ef1a0dbc 0.0s
=> [internal] load .dockerignore 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> [internal] load metadata for docker.io/library/python:3.7-alpine 1.7s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [1/6] FROM docker.io/library/python:3.7-alpine@sha256:6e24a01708ff88a4a99c1f4c357a96190b7efe47230c3bed184b8f4ad368fb9f 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 292B 0.0s
=> CACHED [2/6] WORKDIR /code 0.0s
=> CACHED [3/6] RUN apk add --no-cache gcc musl-dev linux-headers 0.0s
=> [4/6] COPY requirements.txt requirements.txt 0.1s
=> [5/6] RUN pip install -r requirements.txt 4.1s
=> [6/6] COPY . . 0.1s
=> exporting image 0.3s
=> => exporting layers 0.3s
=> => writing image sha256:be2758ec1e5667cc608e59d6ebf0a2a7ebd48513b676d7c600e2ce94b8398597 0.0s
=> => naming to docker.io/library/helloworld:1.0 0.0s
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\Lenovo\Desktop\HelloWorld>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
helloworld 1.0 be2758ec1e56 14 seconds ago 179MB
C:\Users\Lenovo\Desktop\HelloWorld>
```

Docker desktop app:



Run the container:

```
Command Prompt - docker run -p 5000:5000 be2758ec1e56

C:\Users\Lenovo\Desktop\HelloWorld>docker run -p 5000:5000 be2758ec1e56
* Serving Flask app 'app.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [10/Nov/2022 16:38:10] "GET / HTTP/1.1" 200 -
```

Output:



3. Create a IBM container registry and deploy helloworld app.

```
C:\Users\Lenovo>ibmcloud plugin install container-registry
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 1.0.2' found in repository 'IBM Cloud'
Attempting to download the binary file...
 11.90 MiB / 11.90 MiB [=====] 100.00% 0s
12476416 bytes downloaded
Installing binary...
OK
Plug-in 'container-registry 1.0.2' was successfully installed into C:\Users\Lenovo\.bluemix\plugins\container-registry. Use 'ibmcloud plugin show container-registry' to show its details.

C:\Users\Lenovo>ibmcloud cr region-set global
The region is set to 'global', the registry is 'icr.io'.

OK
```

```
C:\Users\Lenovo>ibmcloud cr namespace-add apphelloworld
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'apphelloworld' in resource group 'Default' for account Aswin Venkat's Account in registry icr.io...

Successfully added namespace 'apphelloworld'

OK

C:\Users\Lenovo>
```

```
C:\Users\Lenovo>ibmcloud cr login
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.

OK
```

```
C:\Users\Lenovo>docker tag helloworld:1.0 icr.io/apphelloworld/helloworld:1.0

C:\Users\Lenovo>docker push icr.io/apphelloworld/helloworld:1.0
The push refers to repository [icr.io/apphelloworld/helloworld]
6710d00f007c: Pushed
0be4a798de9f: Pushed
7504a231c5f8: Pushed
a406de449b24: Pushed
680c5f8d2408: Pushed
ed37f306c964: Pushed
31e0f81c5bec: Pushed
b3aede2981e3: Pushed
6666686122fd: Pushed
994393dc58e7: Pushed
1.0: digest: sha256:d521cb0ae002ab12a99e28e7c3873cbc2032a1a4e15a7d9cab72cfbd60b3250e size: 2412

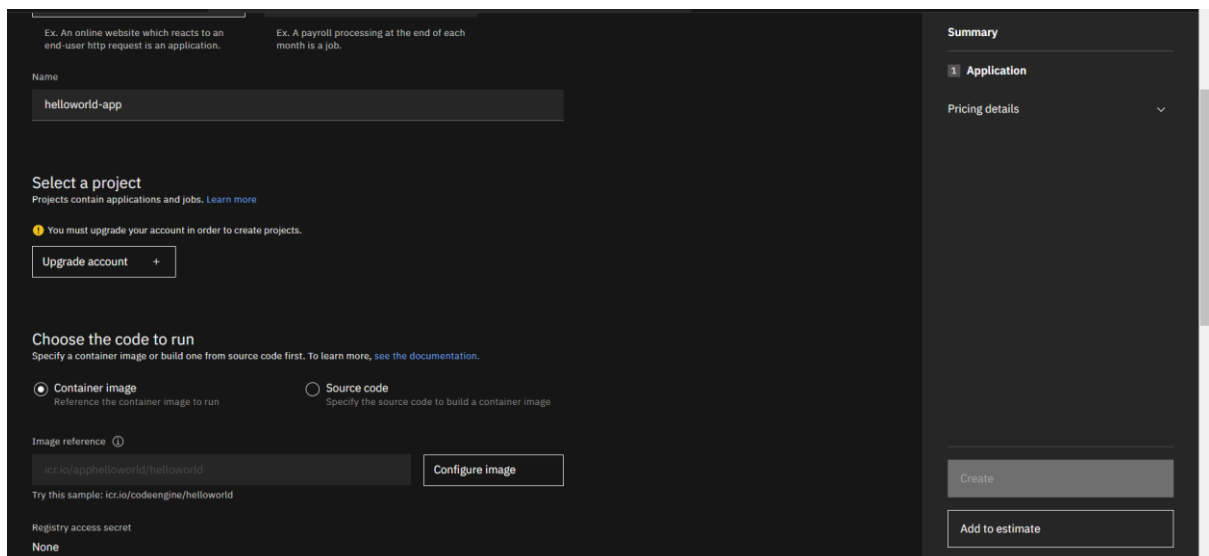
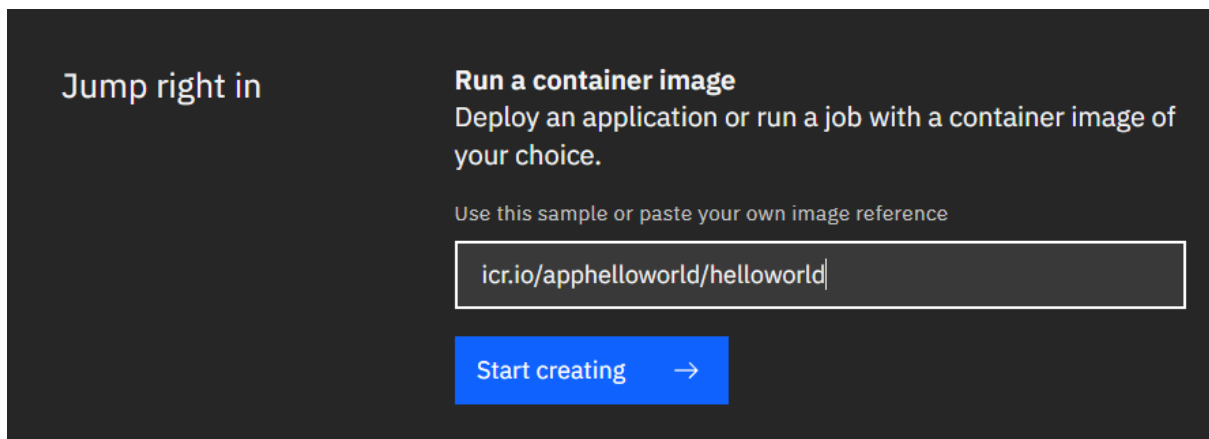
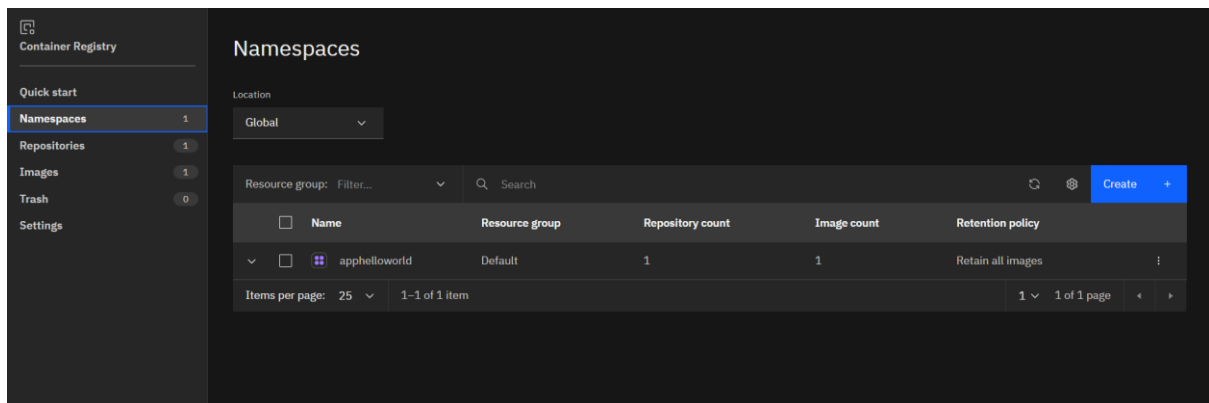
C:\Users\Lenovo>
```

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

Repository          Tag    Digest          Namespace        Created    Size    Security status
icr.io/apphelloworld/helloworld  1.0    d521cb0ae002    apphelloworld    1 hour ago  68 MB    -

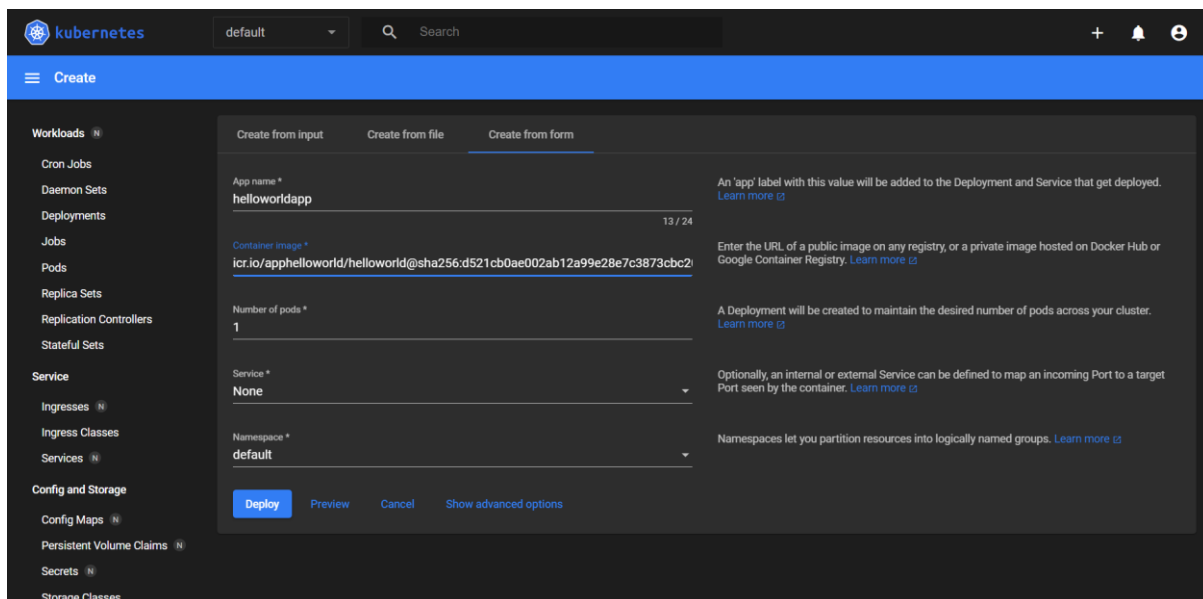
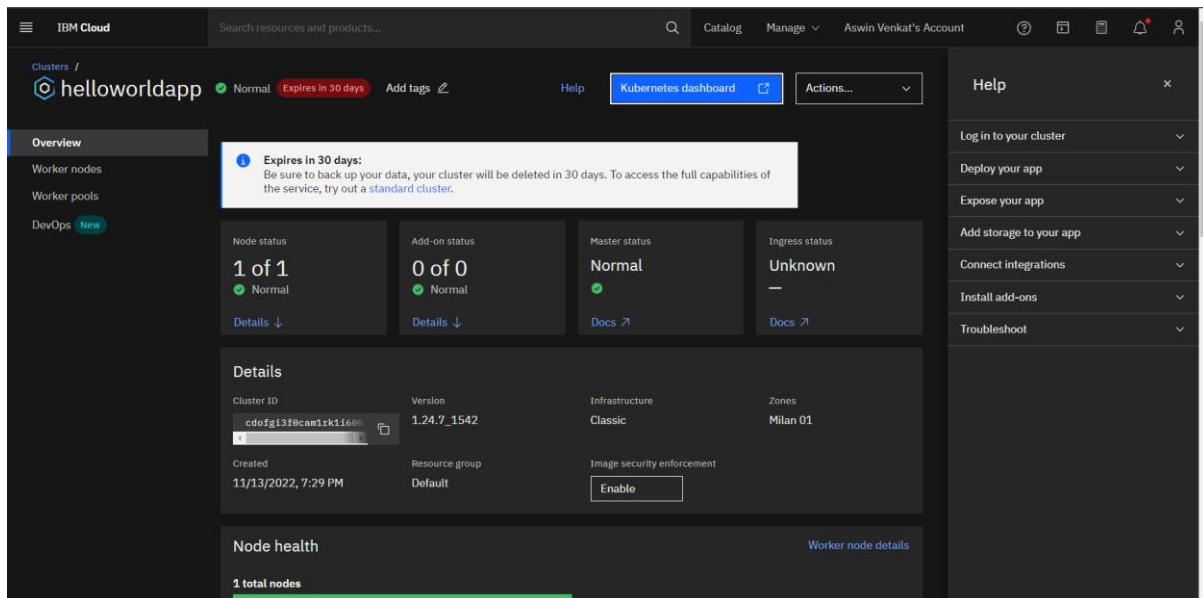
OK
```

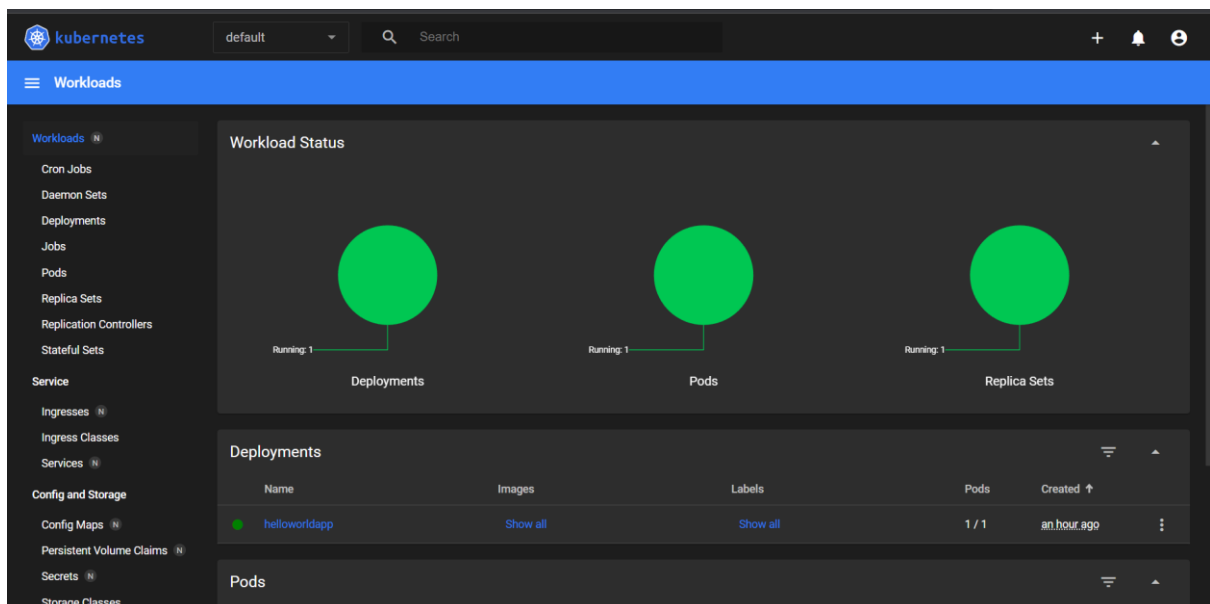
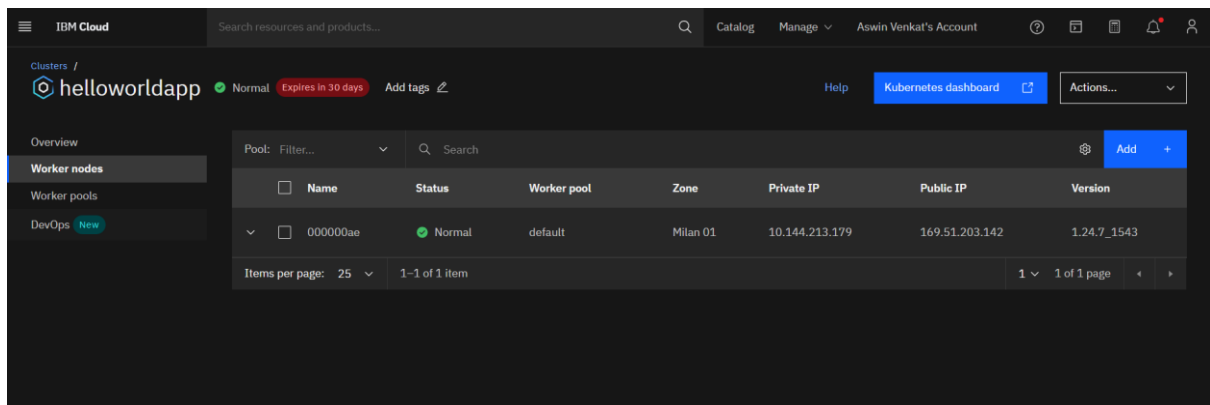
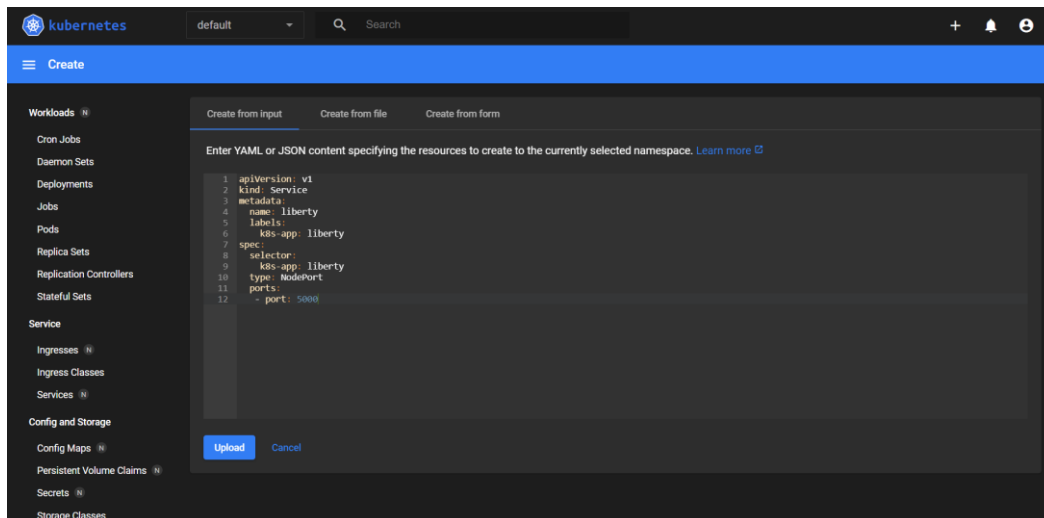
The image is pushed to the namespace.



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

Cluster:





Deployments

Name	Images	Labels	Pods	Created ↑
<div><div></div><div>helloworldapp</div></div>	Show all	Show all	1 / 1	<div>an hour ago</div> <div></div>

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
<div><div></div><div>helloworldapp-5df9ff585b-tc6rl</div></div>	Show all	Show all	10.144.213.179	Running	0	<div>1.00m</div>	<div>19.79Mi</div>	<div>an hour ago</div> <div></div>

Replica Sets

Name	Images	Labels	Pods	Created ↑
<div><div></div><div>helloworldapp-5df9ff585b</div></div>	Show all	Show all	1 / 1	<div>an hour ago</div> <div></div>

← → ↻ 🌐 169.51.203.142:31857