Assignment - 4

Docker and Kubernetes

Team ID : PNT2022TMID50292

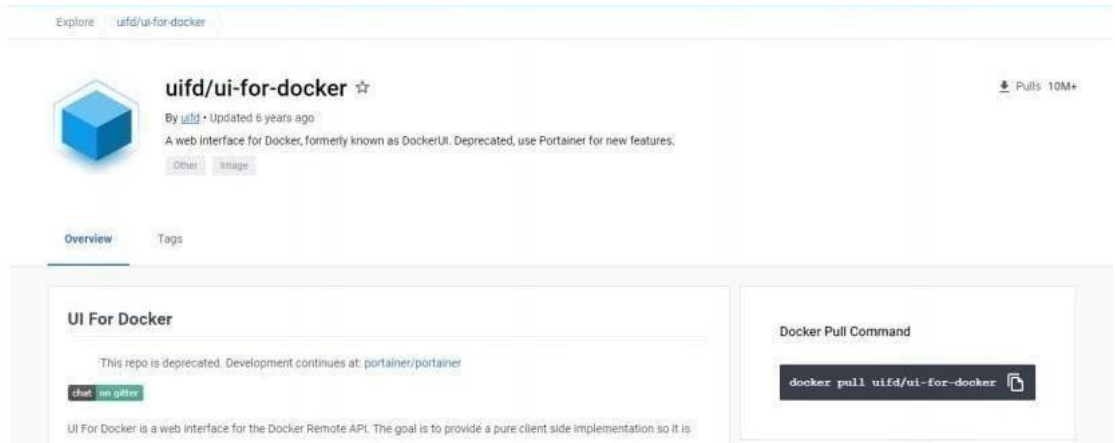
Project Name : Smart Fashion Recommender Application

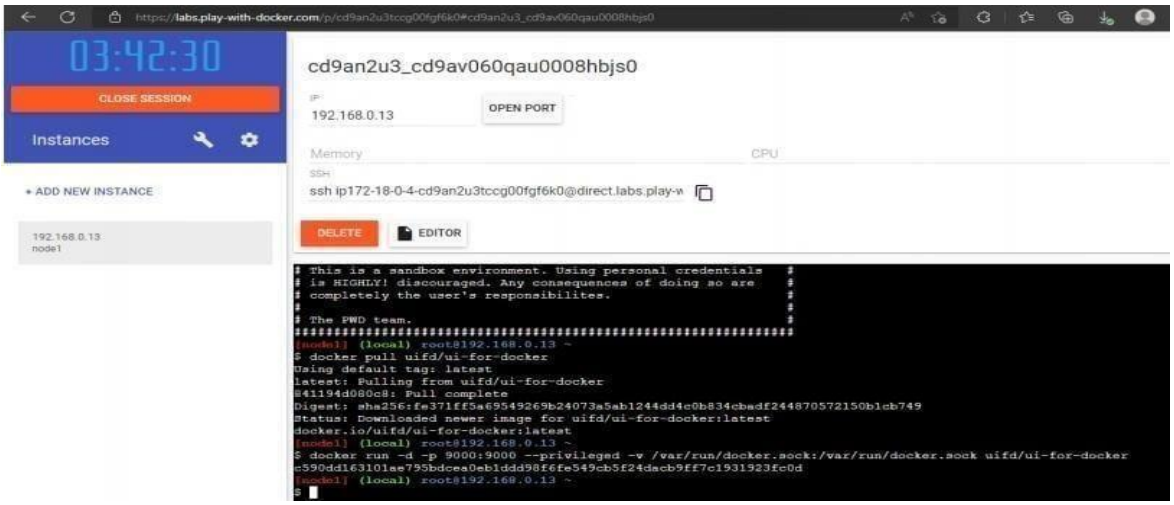
1. PULL AN IMAGE FROM DOCKER HUB AND RUN IT IN DOCKER PLAYGROUND:

docker run --rm -p 8787:8787 rocker/verse docker
pull rocker/verse
docker login --username=amishyaj --email=amirenjai@gmail.com
WARNING: login credentials saved in
/home/amishyaj/.docker/config.json Login Succeeded

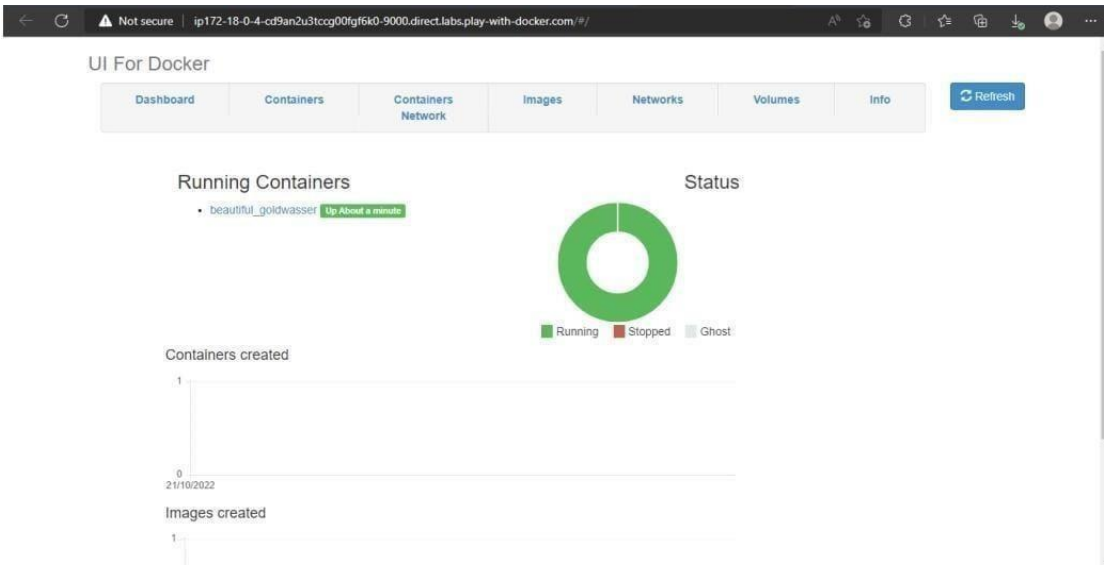
REPOSITORY TAG IMAGE ID CREATED SIZE
verse_gapminder_gsl latest 023ab91c6291 3 minutes ago 1.975 GB
verse_gapminder latest bb38976d03cf 13 minutes ago 1.955 GB
rocker/verse latest 0168d115f220 3 days ago 1.954 GB
docker tag bb38976d03cf amishyaj
/verse_gapminder:firsttry docker
push amishyaj /verse_gapminder

Saving and loading images docker save
verse_gapminder docker save verse_gapminder >
verse_gapminder.tar docker load --input
verse_gapminder.tar docker load --input
verse_gapminder.tar

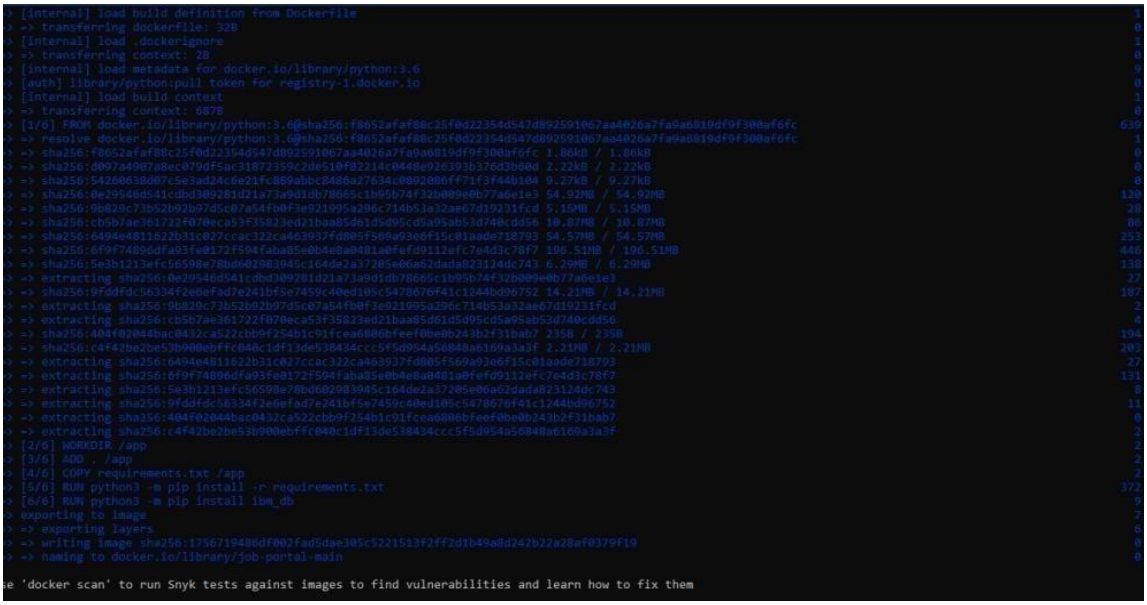


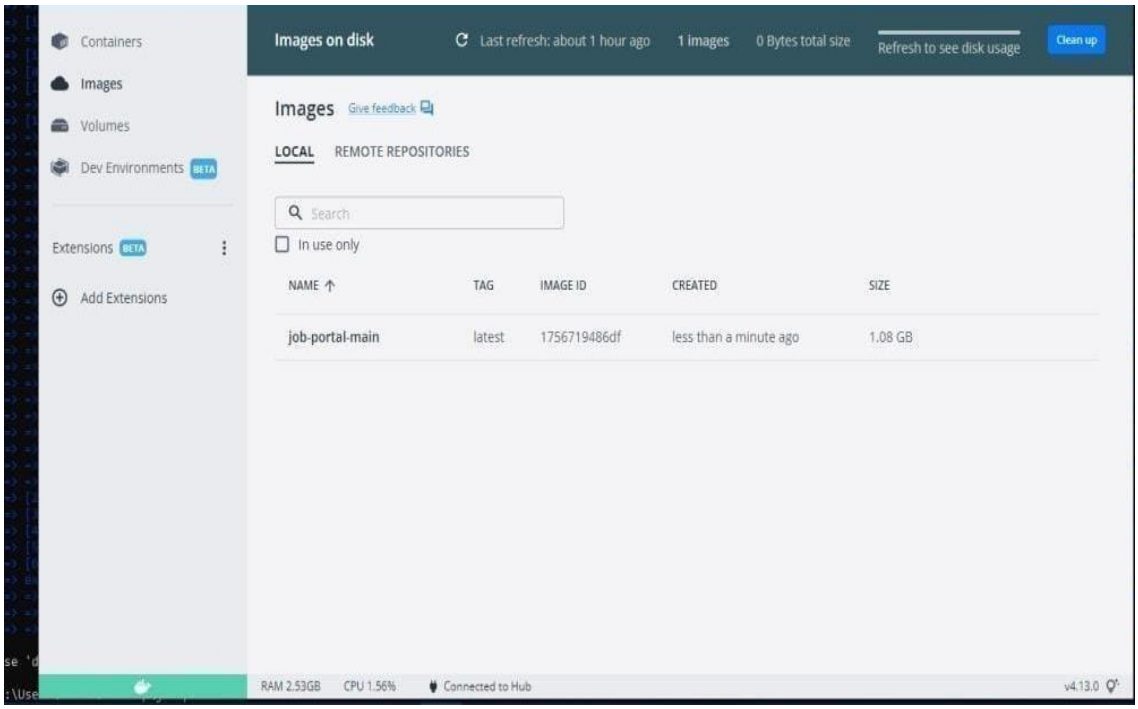

https://labs.play-with-docker.com/p/cd9an2u3tccg00fgf6k0#cd9an2u3_cd9av060qau0008hbjs0
03:42:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
cd9an2u3_cd9av060qau0008hbjs0
IP
192.168.0.13
OPEN PORT
Memory
CPU
SSH
ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-w
DELETE
EDITOR
This is a sandbox environment. Using personal credentials
is HIGHLY! discouraged. Any consequences of doing so are
completely the user's responsibilites.
The PWD team.
[node1] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ef5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.13 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dacb9ff7c1931923fc0d
[node1] (local) root@192.168.0.13 ~
$


Not secure | ip172-18-0-4-cd9an2u3tccg00fgf6k0-9000.direct.labs.play-with-docker.com/#/
UI For Docker
Dashboard
Containers
Containers Network
Images
Networks
Volumes
Info
Refresh
UI For Docker
The UI for Docker container engine
Learn more.
Running Containers
beautiful_goldwasser
Up About a minute
Status



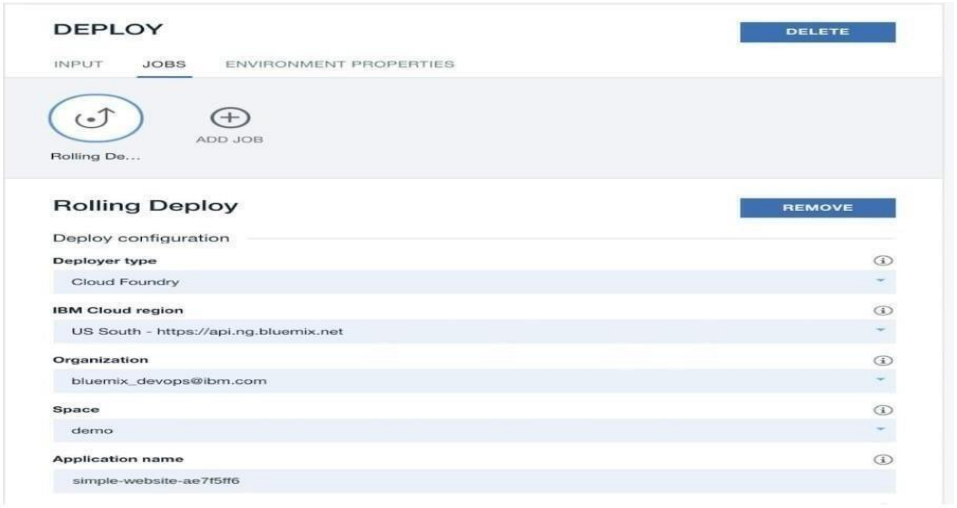
2. CREATE A DOCKER FILE FOR THE JOBPORTAL APPLICATION AND DEPLOY IT IN DOCKER DESKTOP APPLICATION:



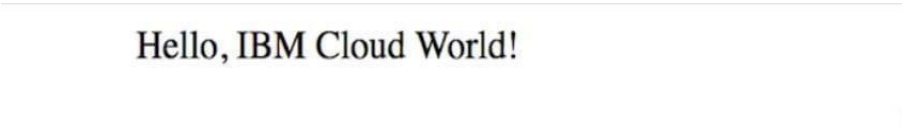


3. CREATE A IBM CONTAINER REGISTRY AND DEPLOY HELLOWORLD APP OR JOBPORTALAPP:

```
        <html>
   <body>
      Hello, IBM Cloud World!
   </body> </html>---
applications:
- buildpack: https://github.com/cloudfoundry/staticfile-buildpack.git
  host: simple-website-${random} name: simple-website-${random}
  memory: 64M
  stack: cflinuxfs2
```



```
{
  "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
  "Params": {
    "Target": {
      "Url": "https://api.ng.bluemix.net",
      "Org": "bluemix_devops@ibm.com",
      "Space": "demo"
    },
    "Name": "simple-website-ae7f5ff6",
    "Instrumentation": {}
  },
  "Path": "manifest.yml",
  "Type": "Cloud Foundry"
}
```



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:

ibmcloud target -g <resource_group_name>ibmcloud cr nishanthc-add <your_ amishyaj >ibmcloud resource service-instance-create example-postgresql databases-forpostgresql standard us- southibmcloud ks cluster-service-bind mycluster default examplepostgresqlgit clone -b node git@github.com:IBM-Cloud/clouddatabases-helloworld-kubernetesexamples.gitspec:

replicas: 3name: cloudpostgres-nodejs-app image:

"registry.<region>.bluemix.net/<namespace>/icdpg" # Edit me

imagePullPolicy: Alwaysibmcloud cr regionYou are targeting region 'us-south', the registry is 'registry.ng.bluemix.net'.ibmcloud cr build -t registry.ng.bluemix.net/<namespace>/icdpg .ibmcloud cr images

env:

- name: BINDING valueFrom:

secretKeyRef: name: <postgres-secret-

name> # Edit me key: binding

```
apiVersion: v1 kind:
Service
metadata: name:
cloudpostgres-service labels:
run: clouddb-demo spec:
  type: NodePort
  selector: run:
    clouddb-demo
  ports:
  - protocol:  TCP
    port:          8080
    nodePort: 30081
    kubectl apply -f
    clouddb-
    deployment.yml
    deployment.app
    s/icdpostgres-
    app        created
    service/cloudpo
    stgres-service
    created
kubectl get pods -o wideibmcloud ks workers <your_cluster_name>
```

Hello World!

Thanks for creating an IBM Cloud Databases for PostgreSQL database.

Add a word to the database

The word hello is defined as a greeting Add

Database output

```
The word bye is defined as a goodbye
The word bye is defined as a farewell
The word hello is defined as a greeting
The word hello is defined as a greeting
The word hello bob is defined as a greeting
The word hello bob is defined as a greeting
```