

Deployment of App in IBM Cloud

Containerize the App (Docker Image Creation)

Date	11 NOV 2022
Team ID	PNT2022TMID42228
Project Name	Skill and Job Recommender
Team Members:	Roja sri varthini A K , Kamali G , Arthi C , Anitha K

Step 1 :

Sign in to your [IBM Cloud Dashboard](#) And Open **IBM Kubernetes Service**.

IBM Cloud

Search resources and products...

Catalog Manage ▾ Rajeswari Ramesh's Ac...

Kubernetes

Clusters

Reservations

Helm catalog

Container Registry

Save on your cluster costs by creating a reservation

Your reservations are not physical instances, but rather a billing discount set up with certain attributes like location, flavor, and infrastructure.

[Request access](#)

Pricing

Calculate the worker node costs without a reservation and your potential savings with a 1 or 3 year contract.

Customize cluster details

Infrastructure ⓘ Classic ▾

Reservation ⓘ None ▾

Worker nodes 3 - +

Zones 3 - +

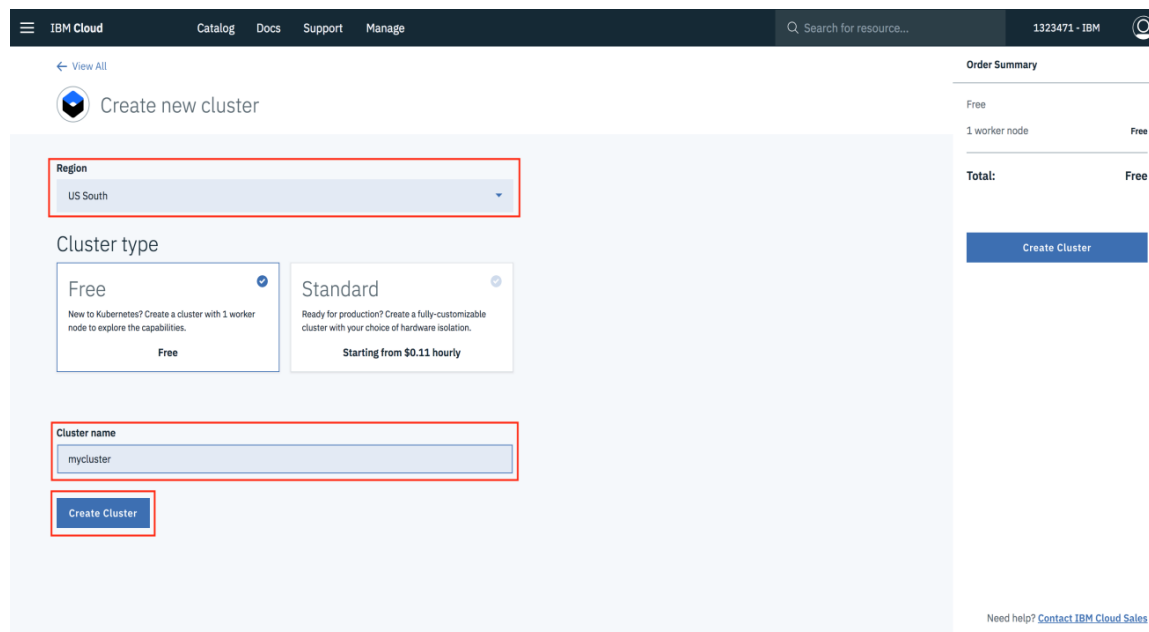
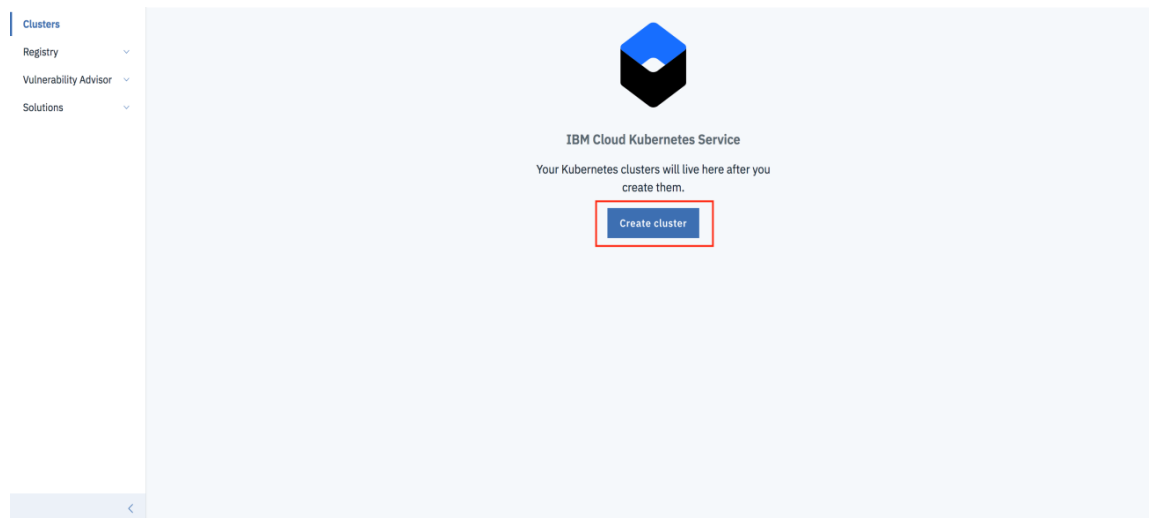
Calculated cost

\$1,905.12

per month estimated

Step 2 :

Create the **Cluster**. Give the Region and cluster type are need to create cluster.



Step 3 :

After creating the cluster Click the Work Node to Note the pubic Id.

The screenshot shows the IBM Cloud Clusters console. At the top, there's a navigation bar with 'IBM Cloud', 'Catalog', 'Docs', 'Support', and 'Manage'. A search bar and the account '1386681 - IBM' are on the right. Below the navigation bar, the breadcrumb 'Clusters / cluster_kunal' is shown. The cluster name 'cluster_kunal' is displayed with a status 'Expires in a month' and 'Normal'. A 'Kubernetes Dashboard' button is on the right. The 'Worker Nodes' tab is selected and highlighted with a red box. Below the tab, the 'Worker Nodes' section contains a search bar and an 'Add Nodes' button. A table lists the worker nodes:

<input type="checkbox"/>	Name	Status	Worker Pool	Zone	Private IP	Public IP	Kubernetes Version
<input type="checkbox"/>	w1	Normal	default	hou02	10.47.79.201	184.172.233.151	1.9.8_1517

At the bottom of the table, it says 'Items per page: 10 | 1-1 of 1 items' and '1 of 1 pages'.

Step 4 :

Next Create a Docker file in Flask App and Type a Following Code within it

```
FROM python:2.7
LABEL maintainer="Kunal Malhotra, kunal.malhotra1@ibm.com"
RUN apt-get update
RUN mkdir /app
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT [ "python" ]
CMD [ "app.py" ]
```

Step 5 :

Open the terminal and type this command to build an image from your Dockerfile: `docker build -t <image_name>:<tag> .`

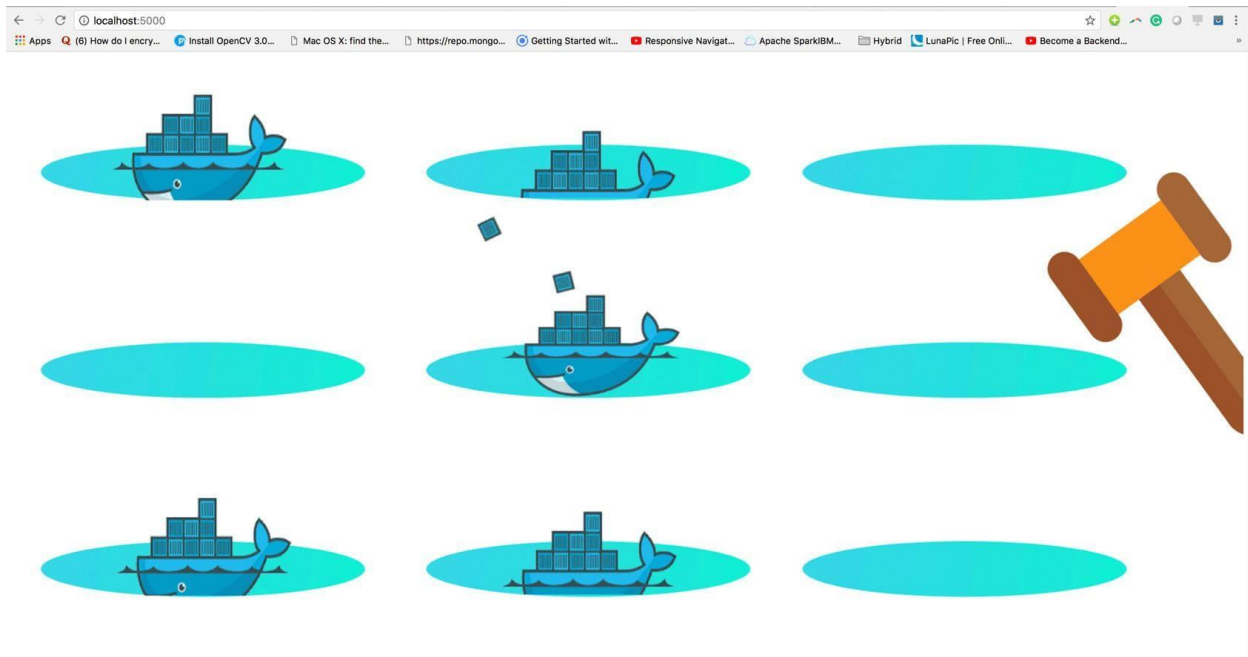
```

kunal@mbp:~$ docker build -t app:latest .
Sending build context to Docker daemon 348.2kB
Step 1/8 : FROM python:2.7
----> 6c76e39e7cfe
Step 2/8 : LABEL maintainer="Kunal Malhotra, kunal.malhotra@ibm.com"
----> Using cache
----> d8b57d41591c
Step 3/8 : RUN apt-get update
----> Using cache
----> 6262a134e40e
Step 4/8 : COPY ./app
----> f07f7378b99f
Step 5/8 : WORKDIR /app
Removing intermediate container f9010b99d2fe
----> 0bcc6af20e3d
Step 6/8 : RUN pip install -r requirements.txt
----> Running in 8153040b00b7
Collecting click==6.7 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/34/c1/8806f99713ddb993c5366c362b2f908f18269f8d792aff1abfd700775a77/click-6.7-py2.py3-none-any.whl (71kB)
Collecting Flask==1.0.2 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/7f/af/08578774ed4536d3242b14dcb4696386634607af824ed997202cd0edb4b/Flask-1.0.2-py2.py3-none-any.whl (91kB)
Collecting itsdangerous==0.24 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/dc/b4/a60bcbda945c00f6d608d8975131ab3f25b22f2bcef1dab221165194b2d4/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2==2.10 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8686763ba4d277a78ca76f32659220a731/Jinja2-2.10-py2.py3-none-any.whl (126kB)
Collecting MarkupSafe==1.0 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/4d/de/32d741db316d8fdb7680822dd37001ef7a448255de9699ab4bfcdbf4172b/MarkupSafe-1.0.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e78d1b8f3efa6682bef8d0aae04fe335243/Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
Building wheels for collected packages: itsdangerous, MarkupSafe
  Running setup.py bdist_wheel for itsdangerous: started
  Running setup.py bdist_wheel for itsdangerous: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/2c/4a/61/5599631c1554768c6290b008c02c72d7317910374ca082ff1e5
  Running setup.py bdist_wheel for MarkupSafe: started
  Running setup.py bdist_wheel for MarkupSafe: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/33/56/20/ebef49a5c612fffe1c5a632146b16596f9e6467676861e4e46
Successfully built itsdangerous MarkupSafe
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.0.2 Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 itsdangerous-0.24
Removing intermediate container 8153040b00b7
----> 66d2636a97bc
Step 7/8 : ENTRYPOINT [ "python" ]
----> Running in bdc1c83815e1
Removing intermediate container bdc1c83815e1
----> 73cefc38a1c
Step 8/8 : CMD [ "app.py" ]
----> Running in a784d430dd6f
Removing intermediate container a784d430dd6f
----> d86bb83763a5
Successfully built d86bb83763a5
Successfully tagged app:latest
kunal@mbp:~$

```

Step 6 :

After you build your image successfully, type: `docker run -d -p 5000:5000 app`.



Step 7 :

Docker Image Created Successfully.

