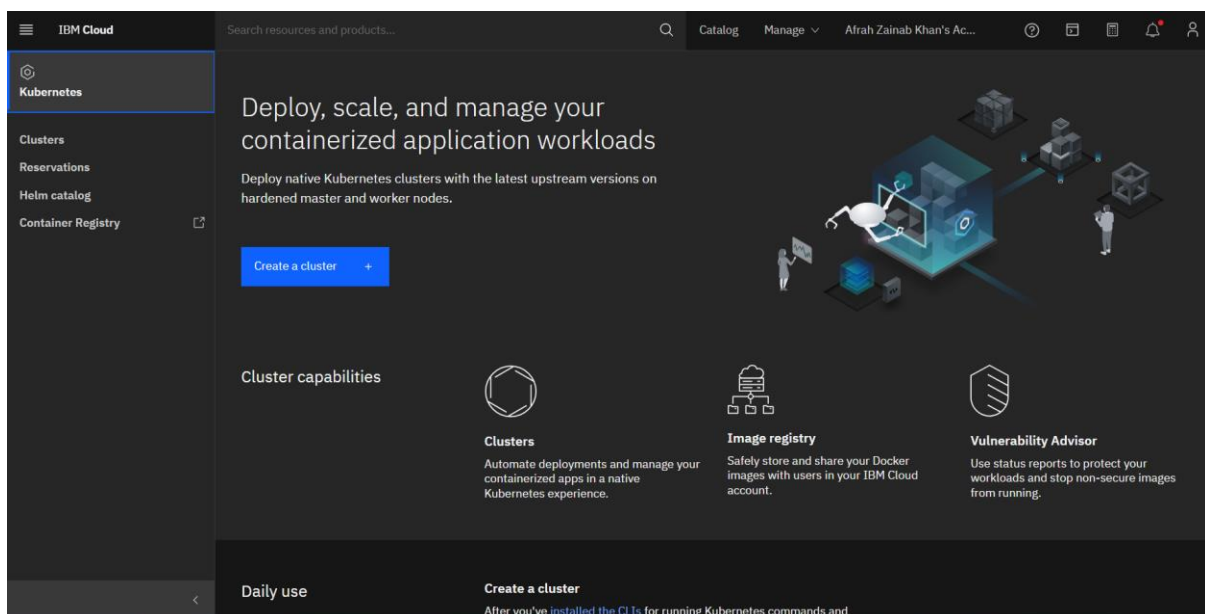
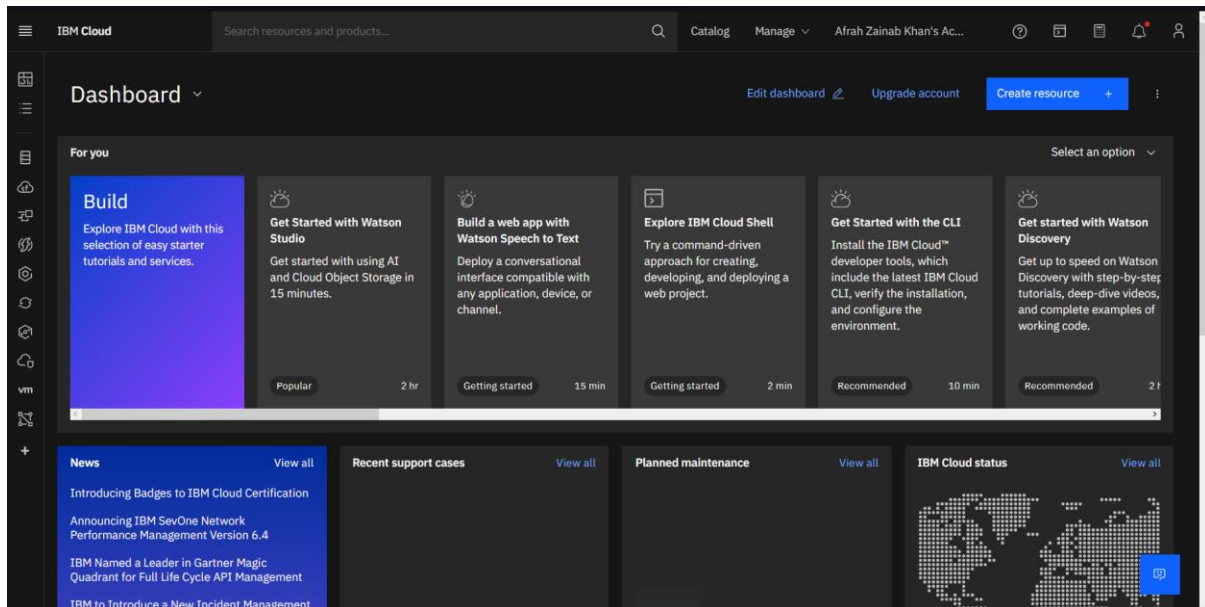


# CONTAINERIZE THE APP

Date	19 November 2022
Team ID	PNT2022TMID32708
Project Name	Job/Skill Recommender

## STEP 1:

Sign in to IBM Cloud Dashboard and Open Kubernetes Service



## STEP 2:

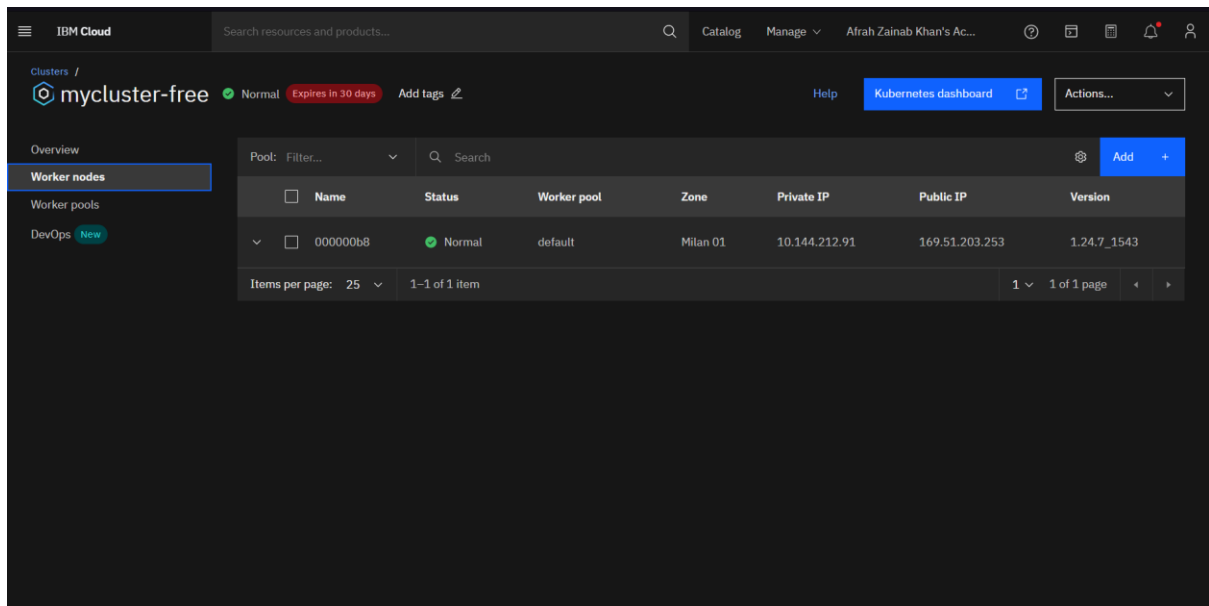
Create the cluster. The Region and cluster type are needed to create the cluster.

The screenshot shows the IBM Cloud 'Kubernetes cluster' creation page. The header includes the IBM Cloud logo, a search bar, and navigation links like 'Catalog' and 'Manage'. The main content area is titled 'Kubernetes cluster' and includes a 'Create' button. A sidebar on the right shows the 'Summary' for the 'Worker node' plan, which is 'Free' and includes '2 vCPUs, 4GB RAM, Virtual - shared, Ubuntu 18'. The 'Total estimated cost' is 'Free/mo'. A 'Create' button is at the bottom of the sidebar. The main content area also has a 'Create' button and a note: 'You have already created your one free cluster. Ready to create more clusters? Choose the Standard cluster type.'

### STEP 3:

After creating the cluster click on the work node to note the public IP.

The screenshot shows the IBM Cloud 'Kubernetes cluster' overview page for a cluster named 'mycluster-free'. The cluster is in a 'Normal' state and 'Expires in 30 days'. The 'Overview' tab is selected, showing a summary of the cluster's status. The 'Node status' section shows '1 of 1' nodes in a 'Normal' state. The 'Add-on status' section shows '0 of 0' add-ons in a 'Normal' state. The 'Master status' section shows 'Normal' status. The 'Ingress status' section shows 'Healthy' status. The 'Details' section provides information about the cluster, including the 'Cluster ID' (cdscq15f041f3tac08g), 'Version' (1.24.8\_1544), 'Infrastructure' (Classic), 'Zones' (Milan 01), 'Created' (11/19/2022, 6:04 PM), 'Resource group' (Default), and 'Image security enforcement' (Enable). The 'Node health' section shows '1 total nodes'.



#### STEP 4:

Next, create a Docker file in Flask app and type the 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 write this command to build an image from our Docker file, “ docker build -t <image\_name>:<tag>.”

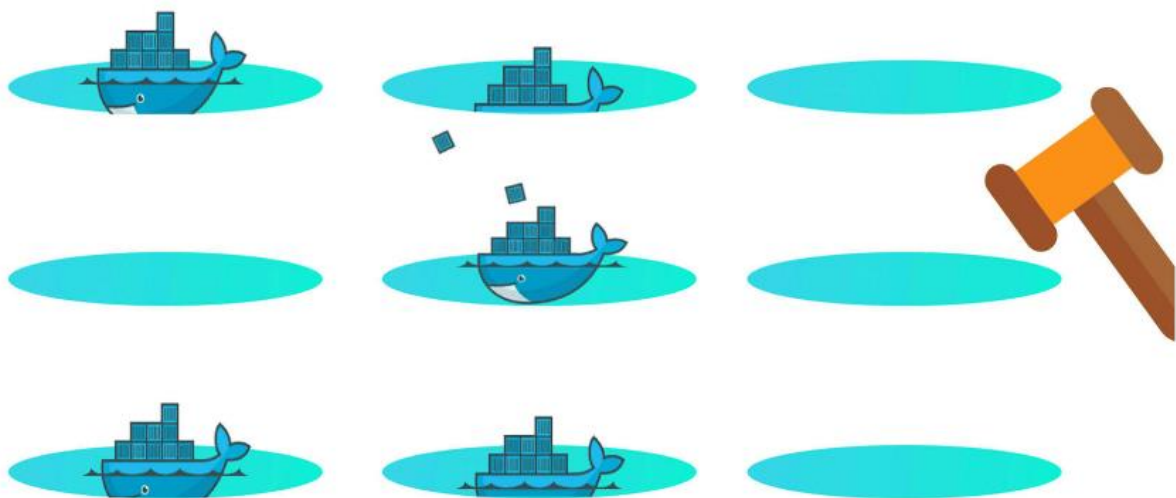
```

kunal@kali:~/web_kunalmalhotra$ 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
--> 6262a134e0e
Step 4/8 : COPY ./app
--> f07f7377899f
Step 5/8 : WORKDIR /app
Removing intermediate container f9010699d2fe
--> 0dccc6f20e3d
Step 6/8 : RUN pip install -r requirements.txt
--> Running in 815304060007
Collecting click==6.7 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/34/c1/8806f99713ddb993c5366c362b2f908f18269f8d792aff1dbfd700775a77/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/e7/08578774ed4536d324b14dacb4696386634607af824ed997202cd8ed94b/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/d5/24/a680dcda945c00f6d608d8975131ab3f29a22f2hcf1dab21165194b2d4/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2==2.10 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27d016a7bed8e8686763ba4d277a78ca76f32659228a731/1/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/32d741db316d8fdb7688822dd37001ef7a448255de9699db4bfcdbf4172b/MarkupSafe-1.0.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e2375aa29c4764e70d1b8f3efa682bef8d0aa04fe335243/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/61/5399631c1554768c6290b08c82c72d7317910374ca682ff1e5
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/ebef9a5c612fffe1c5a63214b16596f9e64676768661e1e46
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 815304060007
--> 66d263eb97bc
Step 7/8 : ENTRYPOINT [ "python" ]
--> Running in bdc1c83815e1
Removing intermediate container bdc1c83815e1
--> 73kefc38a61c
Step 8/8 : CMD [ "app.py" ]
--> Running in a784d430dd6f
Removing intermediate container a784d430dd6f
--> d8b6b83763a5
Successfully built d8b6b83763a5

```

## STEP 6:

After building image successfully, Type “docker run -d -p 5000:5000 app.



## STEP 7:

Docker Image Created Successfully.

