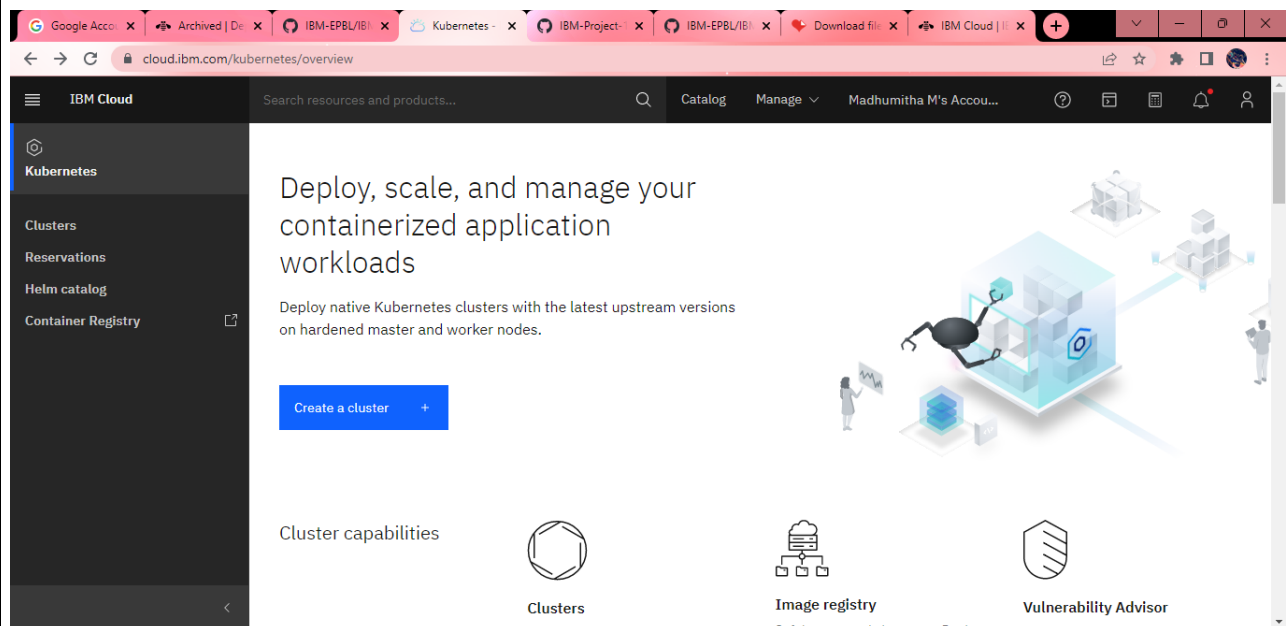


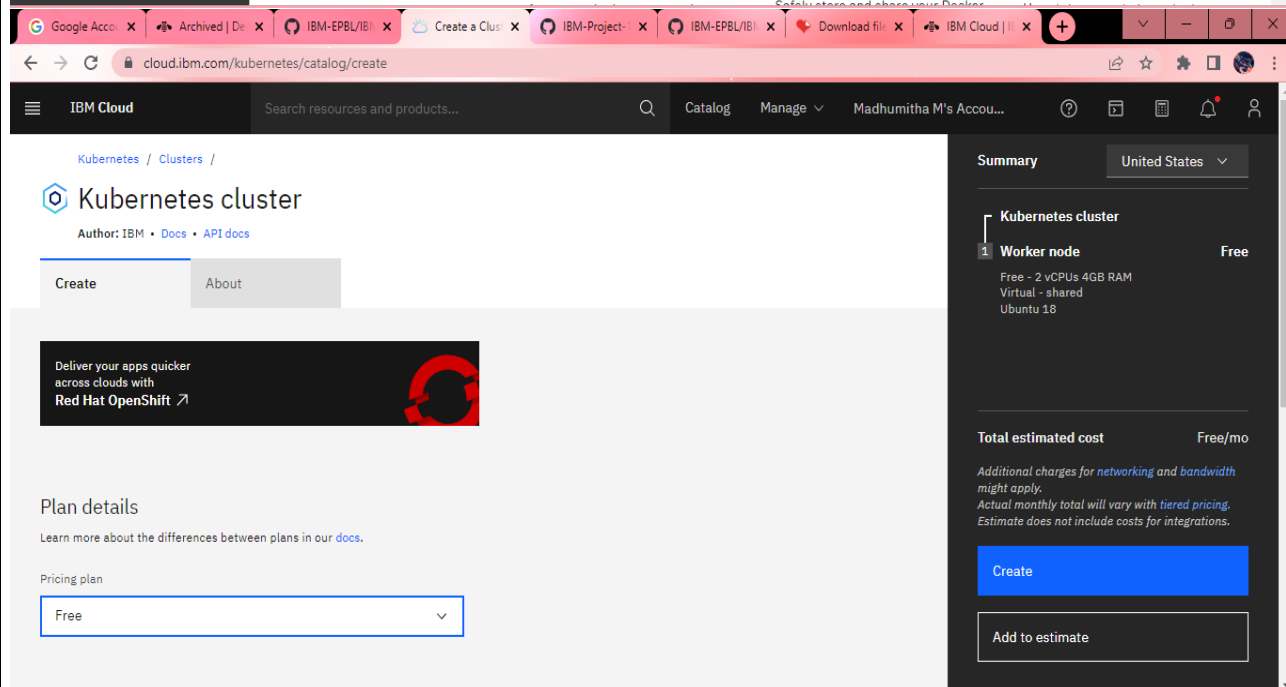
Upload Image To IBM Container Registry

Team ID	PNT2022TMID37226
Project Name	Project - NEWS TRACKER APPLICATION

Creation of kubernetes cluster:



The screenshot shows the IBM Cloud Kubernetes overview page. The left sidebar contains navigation links for Kubernetes, Clusters, Reservations, Helm catalog, and Container Registry. The main content area features the heading "Deploy, scale, and manage your containerized application workloads" and a "Create a cluster" button. Below this, there are icons for "Cluster capabilities", "Clusters", "Image registry", and "Vulnerability Advisor".



The screenshot shows the IBM Cloud Kubernetes cluster creation page. The left sidebar contains navigation links for Kubernetes / Clusters / Create and About. The main content area features the heading "Kubernetes cluster" and a "Create" button. Below this, there is a "Plan details" section with a "Pricing plan" dropdown menu set to "Free". On the right, there is a "Summary" section showing the "Kubernetes cluster" details, including "Worker node" specifications (Free - 2 vCPUs 4GB RAM, Virtual - shared, Ubuntu 18) and a "Total estimated cost" of "Free/mo". A "Create" button is visible at the bottom right.

The top screenshot shows the IBM Cloud Kubernetes dashboard for a cluster named 'mycluster-free'. The 'Overview' page displays four status cards: Node status (1 of 1 Pending), Add-on status (Error), Master status (Unknown), and Ingress status (Pending). Below these is a 'Details' section showing cluster information like ID, version, infrastructure, and zones. A 'Node health' section shows a bar chart indicating 100% of nodes are in a 'Pending' state.

The bottom screenshot shows the 'Worker nodes' page for the same cluster. It displays a table with one node in 'Provision pending' state. The table columns are Name, Status, Worker pool, Zone, Private IP, Public IP, and Version. The node's status is 'Provision pending - Preparing to provision worker'.

Name	Status	Worker pool	Zone	Private IP	Public IP	Version
00000018	Provision pending - Preparing to provision worker	default	Milan 01			1.24.7_1543

Dockerfile:

```
FROM python:2.7
LABEL maintainer="Nivetha, 310119205020@smartinternz.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" ]
```

Build an image from the Docker file:

```
writing build context to docker daemon... Done
Step 1/8 : FROM python:2.7
--> b069d96fbc
Step 2/8 : USER maintainer="Karel Melchior, karel.melchior@flan.com"
--> bc0c330dcf
--> Using cache
--> d8b57d4109c6
Step 3/8 : RUN apt-get update
--> Using cache
--> c82e6c39ade0
Step 4/8 : COPY . /app
--> f07ff7700af0
Step 5/8 : WORKDIR /app
Warning: Internal error container f901bf664dfc
--> bc0c330dcf
Step 6/8 : RUN pip install -r requirements.txt
--> Running in 8153848600cf
Collecting click==6.7 (from -r requirements.txt [line 1])
  Downloading https://files.pythonhosted.org/packages/d4/c2/88ee8f111ebcd181bb070fa43716bcb007307cc11ce-6.7-py2.py3-none-any.whl (79kB)
Collecting Flask==0.12 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/f7/f7/b87577ca9de3ba323754ac9a5284480180ae097b07dbdaef91ea-1.0.7-py2.py3-none-any.whl (101kB)
Collecting Itsdangerous==0.24 (from -r requirements.txt [line 3])
  Downloading https://files.pythonhosted.org/packages/e9/90/a909000000000000000000000000000000000000000000000000000000000000-0.24.tar.gz (4kB)
Collecting Jinja2==2.10 (from -r requirements.txt [line 4])
  Downloading https://files.pythonhosted.org/packages/71/71/7ab0000000000000000000000000000000000000000000000000000000000000-2.10-py2.py3-none-any.whl (124kB)
Collecting MarkupSafe==2.0 (from -r requirements.txt [line 5])
  Downloading https://files.pythonhosted.org/packages/d0/86/865d51defbe1d0fe8867f0c5886770ad78447f550ec60e04f0dd4770/workupafe-1.0.10n-gp
Collecting Werkzeug==0.14.1 (from -r requirements.txt [line 6])
  Downloading https://files.pythonhosted.org/packages/d0/dc/d0dc6475d575a2c94794a01bd181e0addfbf608a9435585/Werkzeug-0.14.1-py2.py3-none-any.whl (329kB)
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'
Storing directory /root/.cache/pip/wheels/53/56/28/beb48d3c2fffefc5e62148c029676e64770802e9e6
Successfully built itsdangerous MarkupSafe
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-0.12 Jinja2-2.10 MarkupSafe-2.0 Werkzeug-0.14.1 click-6.7 itsdangerous-0.24
Warning: Internal error container 8153848600cf
--> b069d96fbc
Step 7/8 : ENVIRONMENT { "python": }
--> Running in bc0c330dcf
Warning: Internal error container bc0c330dcf
--> 73efc330dc
Step 8/8 : CMD ["app.py"]
--> Running in a784438a0f
Warning: Internal error container a784438a0f
--> d8b57d4109c6
Successfully built d8b57d4109c6
Successfully tagged app:latest
```

Push the image to the IBM Cloud Registry:

- 1) go to IBM Cloud Kubernetes Service.
- 2) select Private Repositories.
- 3) `ibmcloud plugin install container-registry -r "IBM Cloud"`
- 4) `ibmcloud login -a <cloud_foundry_end_point_for_the_region>`
- 5) `ibmcloud cr namespace-add <namespace>`
- 6) `ibmcloud cr login`
- 7) `docker tag <image_name> <region_url>/<namespace>/<image_name>:<tag>`
- 8) `docker push <region_url>/<namespace>/<image_name>:<tag>`
- 9) `ibmcloud cr image-list`

```
kunals-mbp:web kunalmalhotra$ ibmcloud cr image-list
Listing Images...

REPOSITORY                                TAG      DIGEST      NAMESPACE  CREATED      SIZE      SECURITY STATUS
registry.ng.bluemix.net/Flask-node/app    latest   b721dd768fe0 flask-node   1 day ago   366 MB    3 Issues

OK
kunals-mbp:web kunalmalhotra$
```

Create configuration files for Kubernetes:



```
└─ web
  ├── static
  ├── templates
  ├── venv
  ├── .dockerignore
  ├── .gitignore
  ├── app.py
  ├── deployment.yaml
  ├── Dockerfile
  ├── requirements.txt
  ├── service.yaml
  └── README.md
```

deployment.yaml :

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
        - name: flasknode
          image: registry.ng.bluemix.net/flask-node/app
          imagePullPolicy: Always
          ports:
```

- containerPort: 5000

service.yaml:

apiVersion: v1

kind: Service

metadata:

name: flask-node-deployment

spec:

ports:

- port: 5000

targetPort: 5000

selector:

app: flasknode

Deploy your application to Kubernetes:

- 1) ibmcloud cs region-set us-south
- 2) a) ibmcloud cs cluster-config cluster_kunal
- 2.b) > export KUBECONFIG=/Users/\$USER/.bluemix/plugins/container-service/clusters/<cluster_name>/<cluster_configuration_file.yaml>
- 3) kubectl get nodes
- 4) kubectl create -f deployment.yaml
- 5) kubectl create -f service.yaml
- 6) Look at the Kubernetes dashboard from the IBM Kubernetes Service overview page.

The screenshot shows the IBM Kubernetes Service dashboard. The left sidebar contains a navigation menu with options like Cluster, Namespaces, Nodes, Persistent Volumes, Roles, Storage Classes, Namespace (default), Overview, Workloads (Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets), Discovery and Load Balancing (Ingresses, Services), and Config and Storage (Config Maps, Persistent Volume Claims). The main content area shows the 'Overview' page with several sections:

- Deployments:** A table with columns Name, Labels, Pods, Age, and Images. It shows one deployment: 'flask-node-deployment' with label 'app: flasknode', 1/1 pods, 5 minutes age, and image 'registry.ng.bluemix.net/flask-node/app'.
- Pods:** A table with columns Name, Node, Status, Restarts, Age, CPU (cores), and Memory (bytes). It shows one pod: 'flask-node-deployment-5cd9fcf5bc-d6n6x' on node '10.47.79.201', status 'Running', 0 restarts, 5 minutes age, 0 CPU cores, and 19.352 Mi memory.
- Replica Sets:** A table with columns Name, Labels, Pods, Age, and Images. It shows one replica set: 'flask-node-deployment-5cd9fcf5bc' with label 'app: flasknode', 1/1 pods, 5 minutes age, and image 'registry.ng.bluemix.net/flask-node/app'.
- Discovery and Load Balancing:** A section for Services.
- Services:** A table with columns Name, Labels, Cluster IP, Internal endpoints, External endpoints, and Age. It shows two services: 'kubernetes' (Cluster IP: 172.21.0.1, Internal endpoints: kubernetes:443 TCP, kubernetes:0 TCP, External endpoints: -, Age: a minute) and 'flask-node-deployment' (Cluster IP: 172.21.104.14, Internal endpoints: flask-node-deployment:5000 TCP, flask-node-deployment:0 TCP, External endpoints: -, Age: a minute).
- Config and Storage:** A section for Config Maps and Persistent Volume Claims.

- 7) Finally, go to your browser and ping the Public IP of your worker node

