Final Deliverables Report

Team ID	PNT2022TMID21037
Project Name	Inventory Management System for Retailers

Team members and their Contribution:

Name	**Roll no**	**Contribution**
Arun Kumar	714019104015	Frontend – 5 Pages, Deployment of using docker and Kubernetes.
Haneef	714019104030	Frontend – 5 Pages, Documentation
Hareesh Kumaran	714019104031	Frontend – 4 Pages, Documentation
Kaushik	714019104045	Backend Fully , Integration of IBM Cloud, Deployment of using docker and Kubernetes.

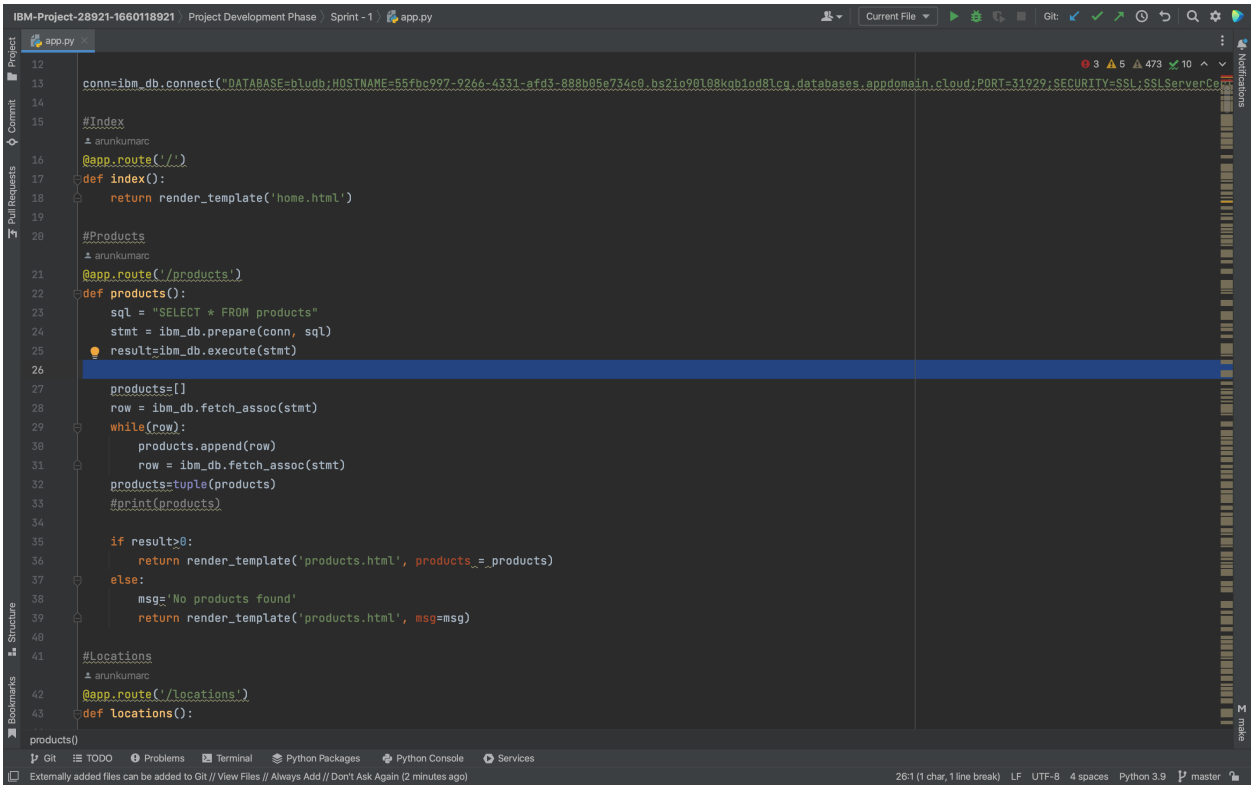
Introduction:

1. Sprint 1 -Backend
2. Sprint 2 -Frontend
3. Sprint 3 -IBM cloud integration
4. Sprint 4 -Docker and Kubernetes

Sprint 1 – Backend:

All the routes to each page and APIs are created.

Example, (For Products page)

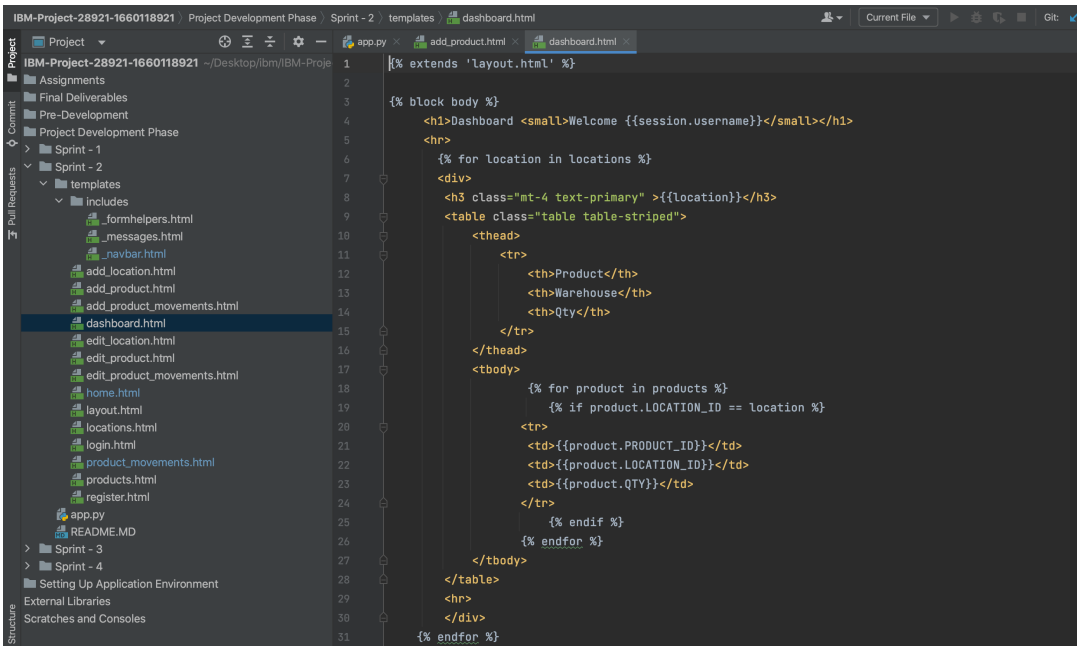


```
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLServerCe

#Index
arunkumarc
@app.route('/')
def index():
    return render_template('home.html')

#Products
arunkumarc
@app.route('/products')
def products():
    sql = "SELECT * FROM products"
    stmt = ibm_db.prepare(conn, sql)
    result=ibm_db.execute(stmt)

    products=[]
    row = ibm_db.fetch_assoc(stmt)
    while(row):
        products.append(row)
        row = ibm_db.fetch_assoc(stmt)
    products=tuple(products)
    #print(products)

    if result>0:
        return render_template('products.html', products_=_products)
    else:
        msg='No products found'
        return render_template('products.html', msg=msg)

#Locations
arunkumarc
@app.route('/locations')
def locations():
```

Sprint 2 – Frontend:

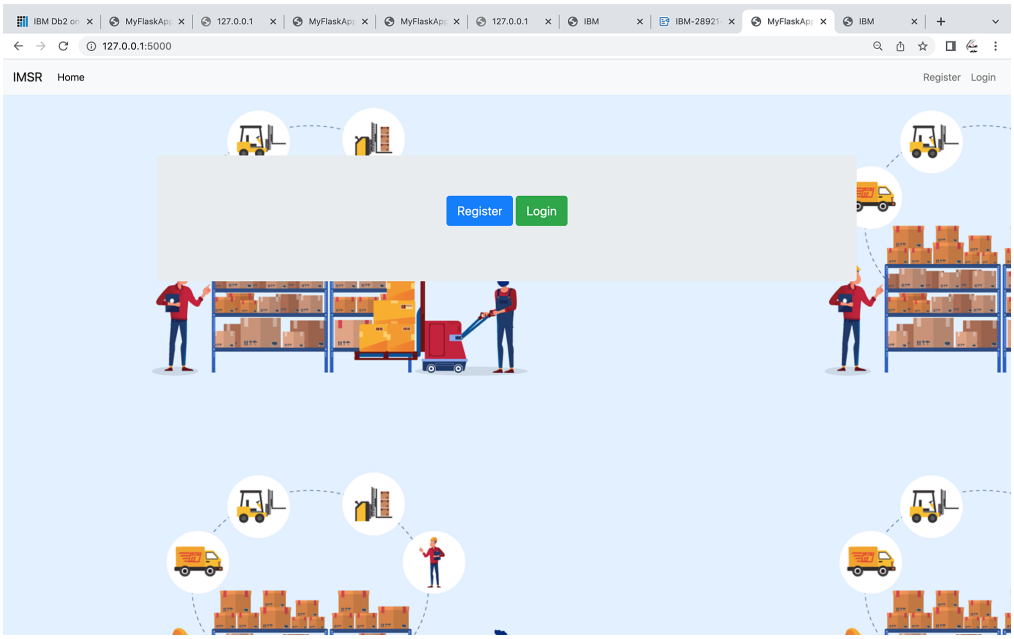
The frontend is written using HTML, CSS (using Bootstrap) and JavaScript for all the pages to which the routes created in Sprint 1.

For Example, (The Hierarchy of different pages and the code for login page)

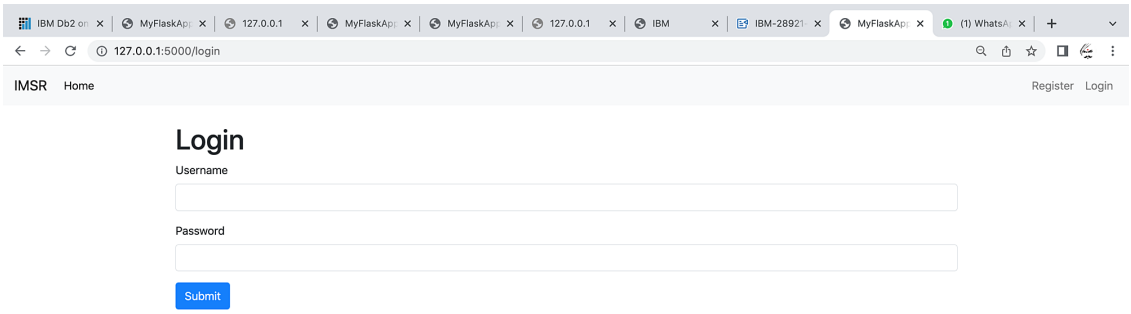


Sample FrontEnd Pages,

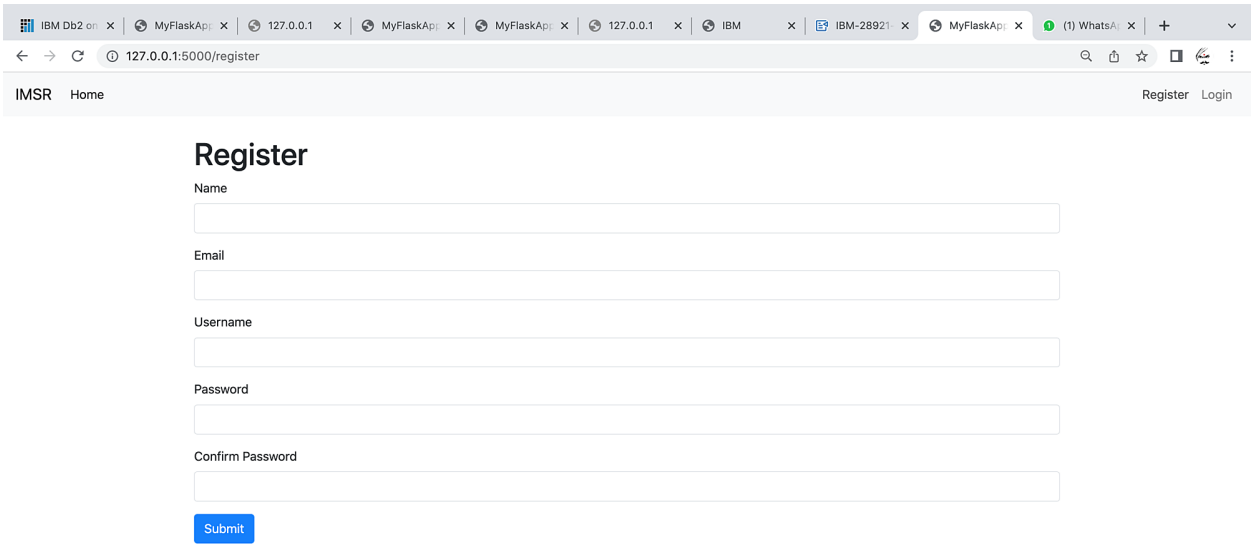
Home Page,



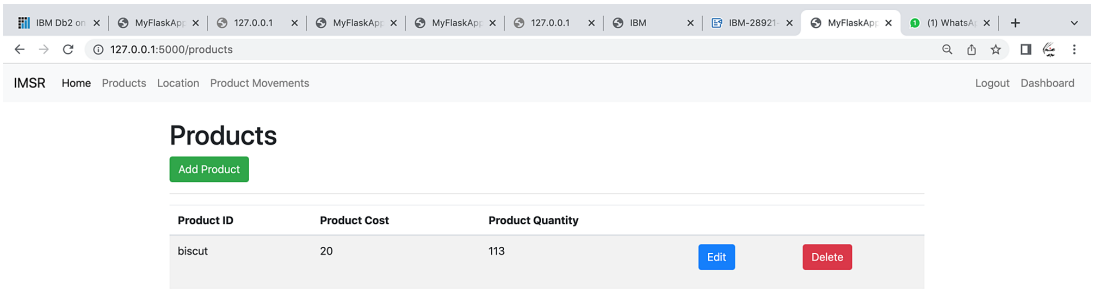
Login page,



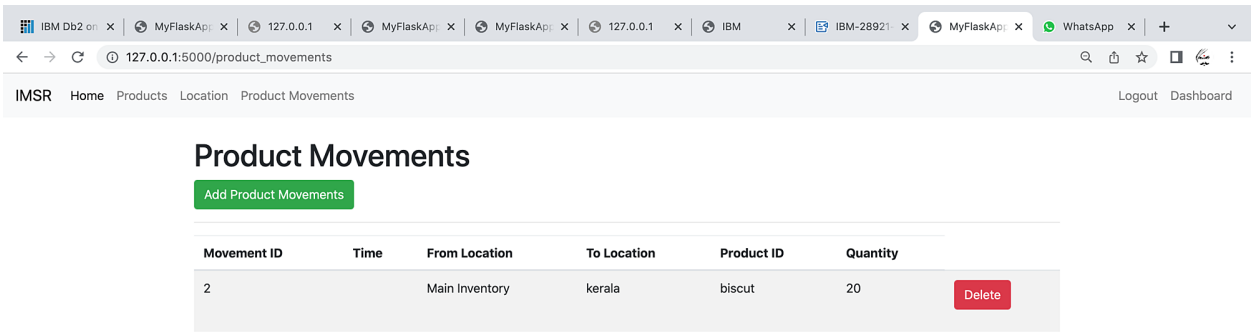
Register,



products page,



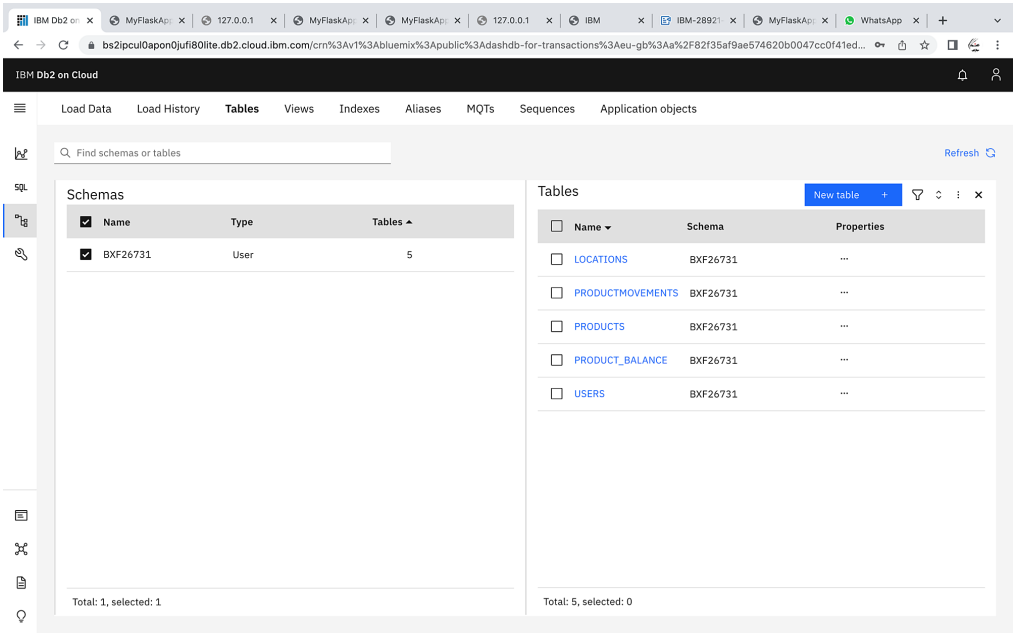
product movents page,



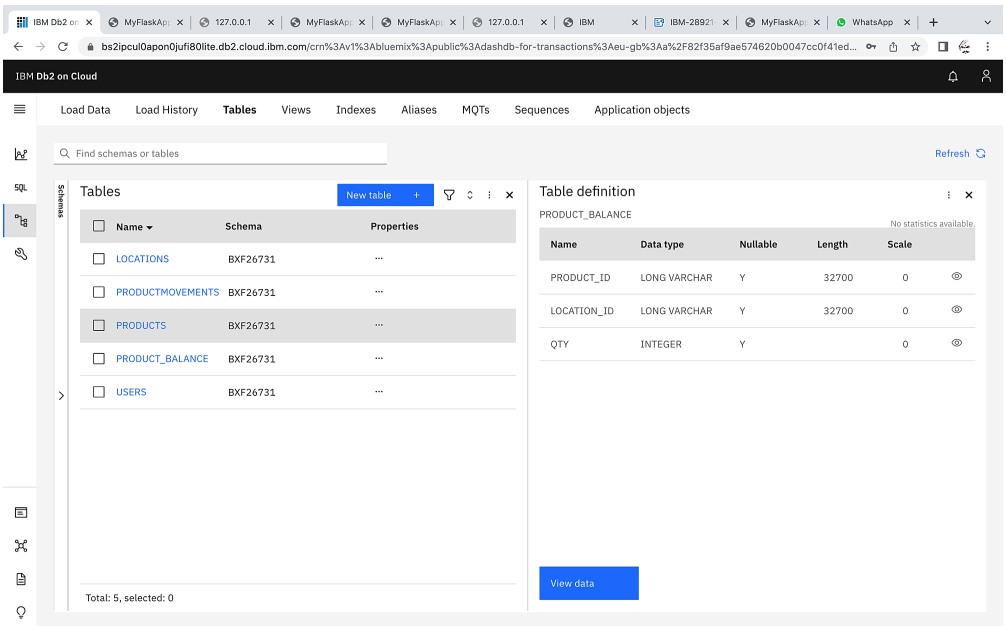
Sprint 3 - IBM Cloud Integration

IBM Cloud Integration:

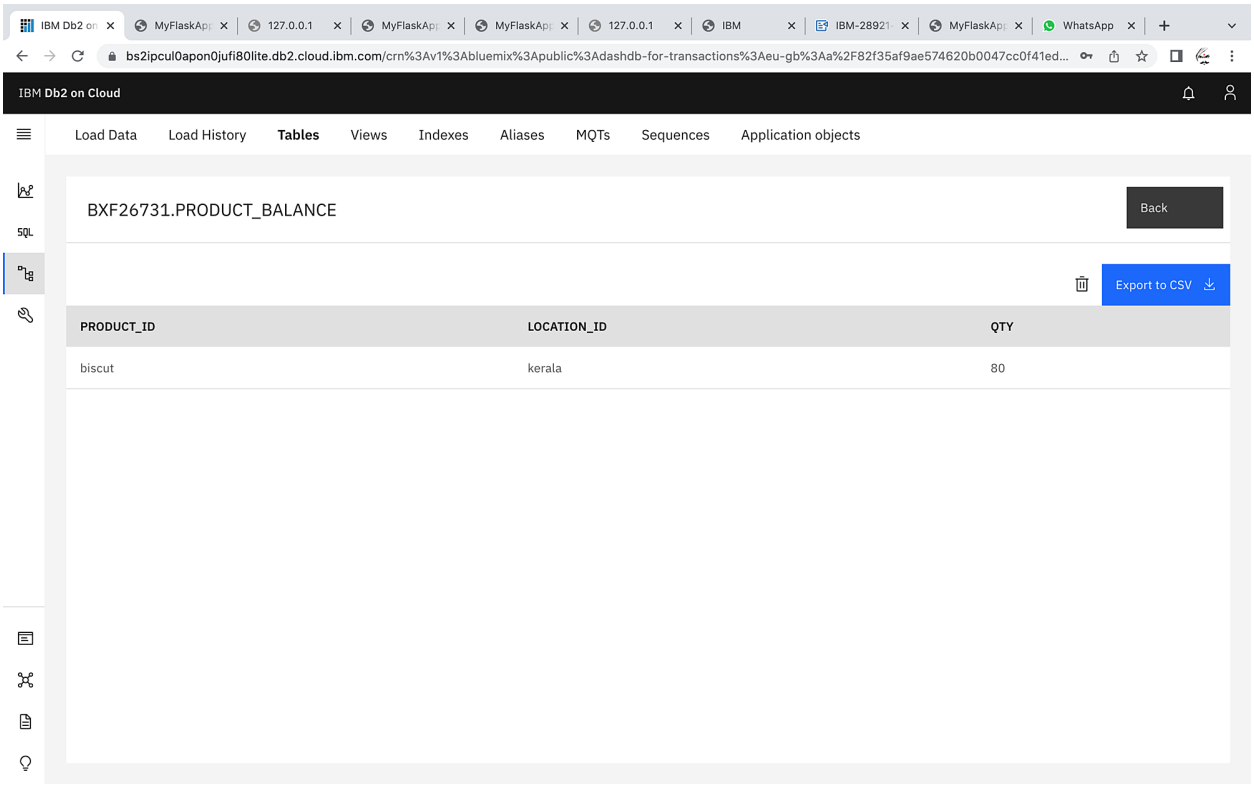
5 tables created for our project,



Schema of the particular table (For Example, Product_Balance)



Data of a particular table (For Example, Product_Balance)



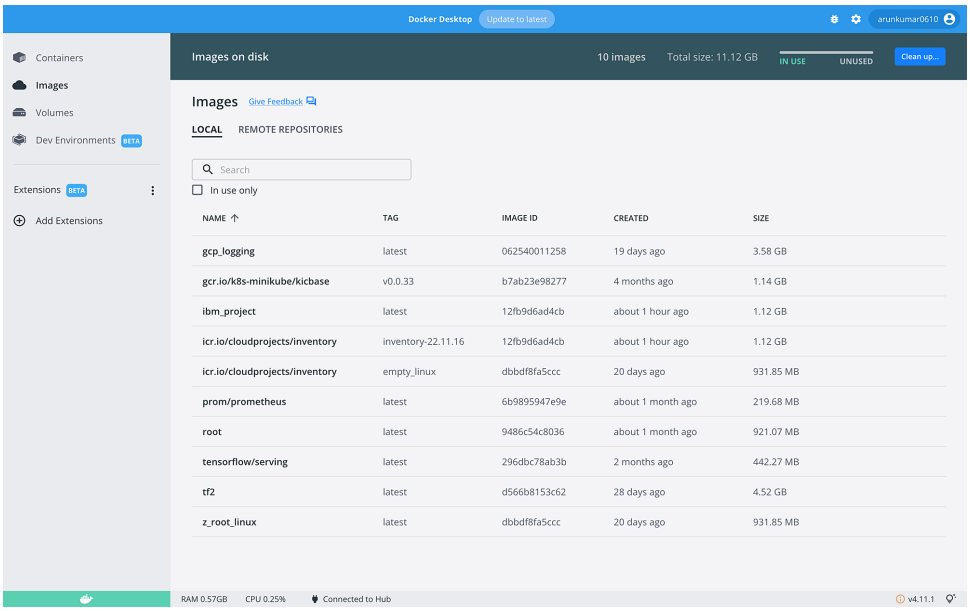
Code for Connection of IBM Database,

Note: DigiCertGlobalRootCA.crt should be downloaded and configured within the project folder.

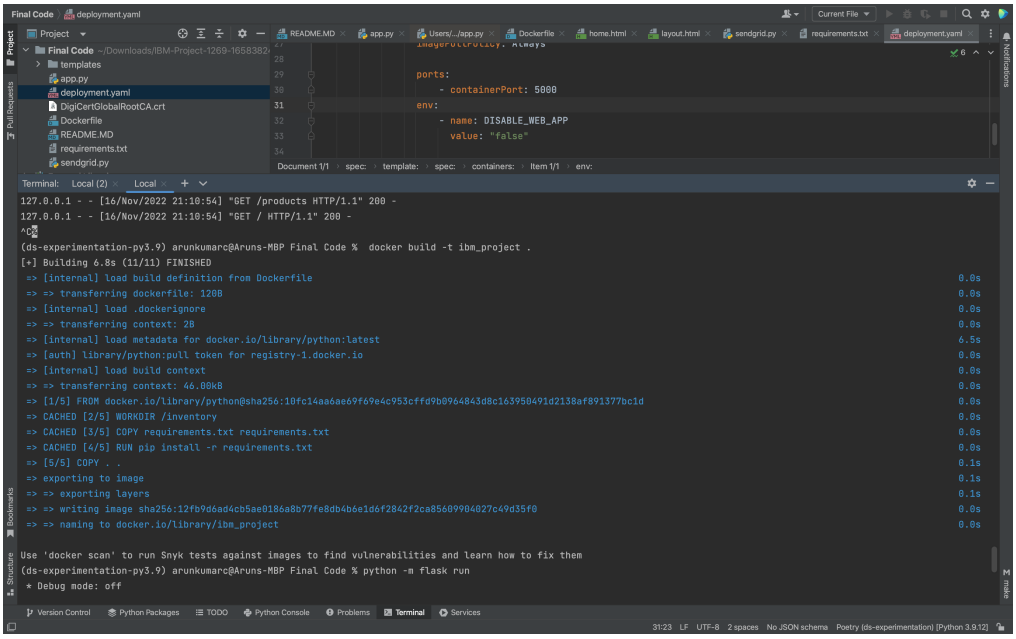
Sprint 4 (Deploying the application using Docker and Kubernetes):

Note: Make sure to create a Dockerfile in the project folder.

Login into DockerHub in Project Folder using command prompt. This connects local docker desktop to cloud docker hub.



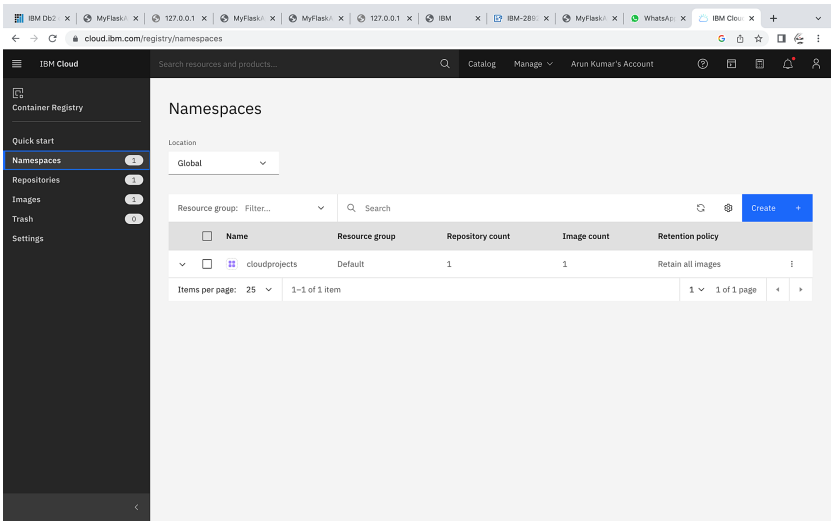
Building an image for our project,



Create a valid Deployment.yaml file,

```
OUTPUT  PROBLEMS  DEBUG CONSOLE  TERMINAL  JUPYTER

arunkumarc@Aruns-MBP Untitled Folder % kubectl apply -f dep.yaml
pod/mypod unchanged
arunkumarc@Aruns-MBP Untitled Folder %
```

Create a namespace in IBM Container registry,

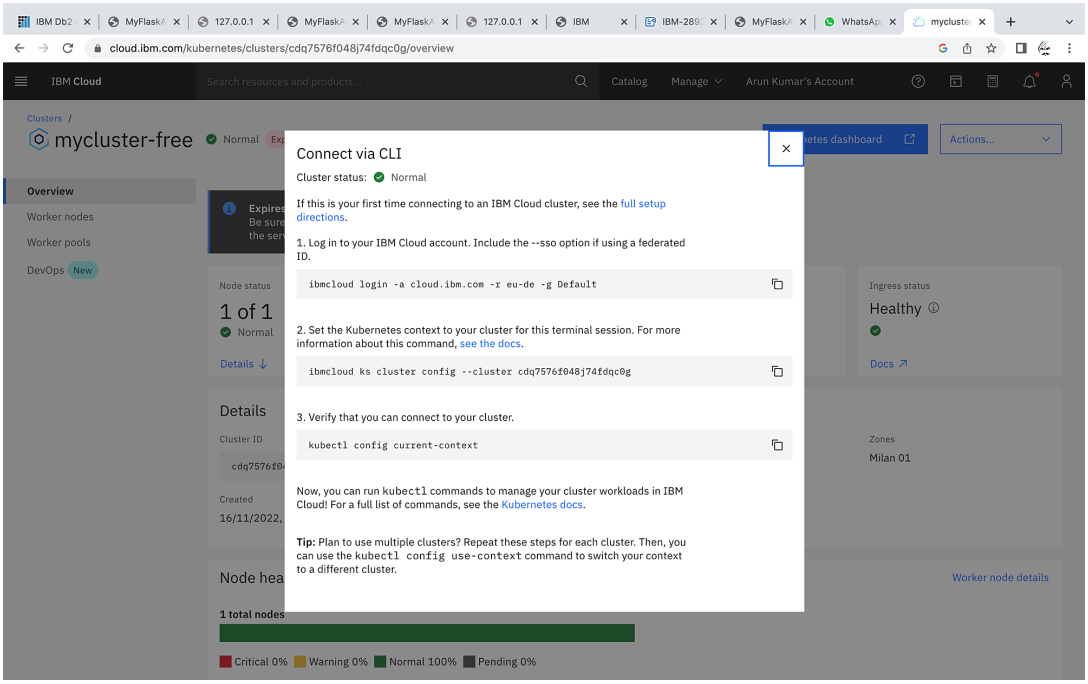


Pushing the project into IBM container Registry,

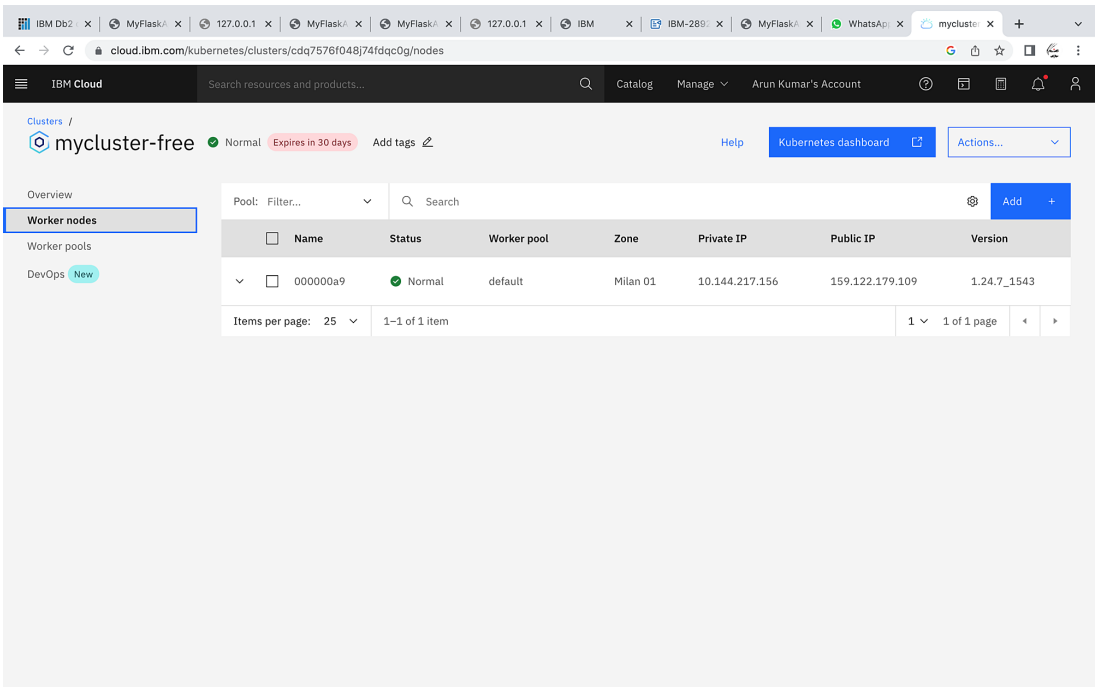
```
Last login: Wed Nov 16 21:12:25 on ttys002
arunkumarcBAruns-MacBook-Pro ~ % ibmcloud login
API endpoint: https://cloud.ibm.com

Email> carunkumar19cs@srishakthi.ac.in

Password>
Authenticating...
Credentials were rejected.
Code: BXNIM0602E, message: The credentials you entered for the user 'carunkumar19cs@srishakthi.ac.in' are incorrect.

Password>
Authenticating...
OK

Targeted account Arun Kumar's Account (82f35af9ae574620b0047cc0f41edfcd)


Select a region (or press enter to skip):
1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. kr-seo
6. eu-de
7. eu-gb
8. ca-tor
9. us-south
10. us-east
11. br-sao
Enter a number>


API endpoint:      https://cloud.ibm.com
Region:
User:              carunkumar19cs@srishakthi.ac.in
Account:           Arun Kumar's Account (82f35af9ae574620b0047cc0f41edfcd)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:
arunkumarcBAruns-MacBook-Pro ~ % ibmcloud cr login
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.

OK
arunkumarcBAruns-MacBook-Pro ~ % docker push icr.io/cloudprojects/inventory:inventory-22.11.16
The push refers to repository [icr.io/cloudprojects/inventory]
5f87197dcc0c: Layer already exists
8b826a7cddaf: Pushing [>                                                  ]  2.181MB/191.6MB
0d36aab5d077: Layer already exists
681842495a3: Layer already exists
345c9e42b8e4: Pushed
24bf8dd8c4a6: Pushed
18bbb218c890: Pushing [===>                                               ]  4.939MB/67.85MB
e6e9854ca999: Pushed
397a239e853b: Pushing [==========================>                        ]  286.1MB/529MB
89c3244a87b2: Pushed
80231db1194c: Pushed
f1c1f2985b4c: Pushed
ccba29d69370: Waiting
```

Note: Create a Kubernetes Cluster in IBM Cloud and wait for the work node to get fully deployed.

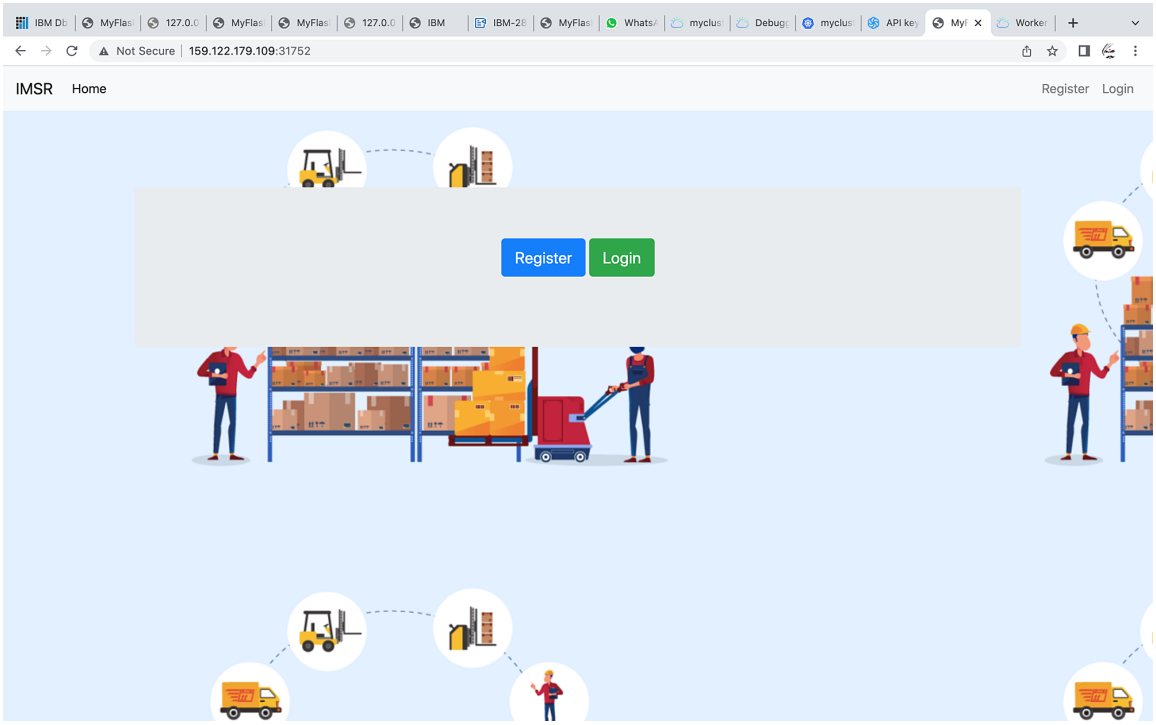
Then, Login into Kubernetes Cluster using the following commands,



Check for the public IP address in your IBM Kubernetes Cluster under Worker Node,



Thus we have the Public IP address and the Nodeport. so can access application



Our application ip :

http://159.122.179.109:31752/

Result:

Thus In this way We developed a “Inventory management System for Retailers” using Python, Sendgrid and IBM Cloud Services (IBM DB2, IBM Container registry, IBM Kubernetes).

Thank You!