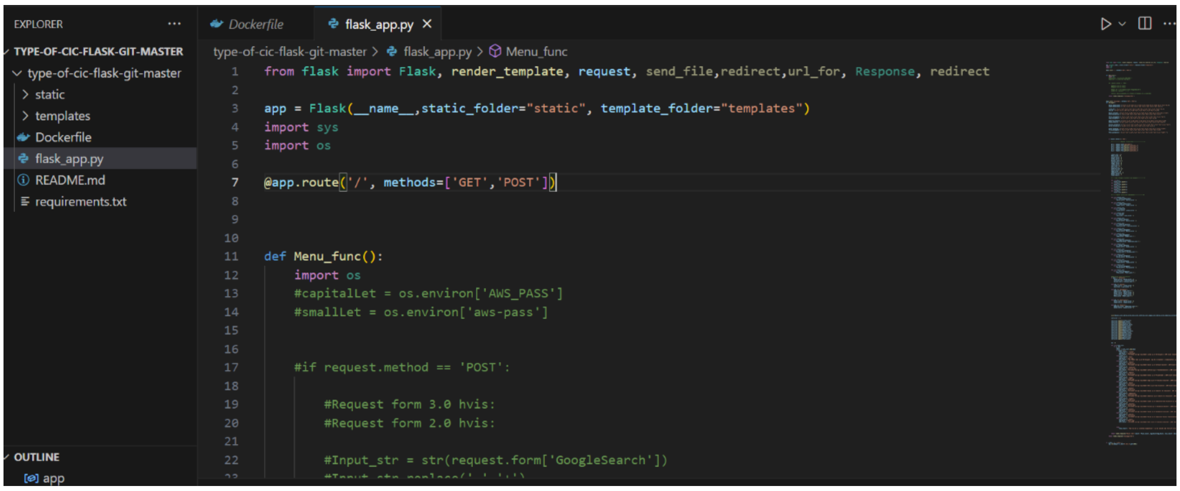
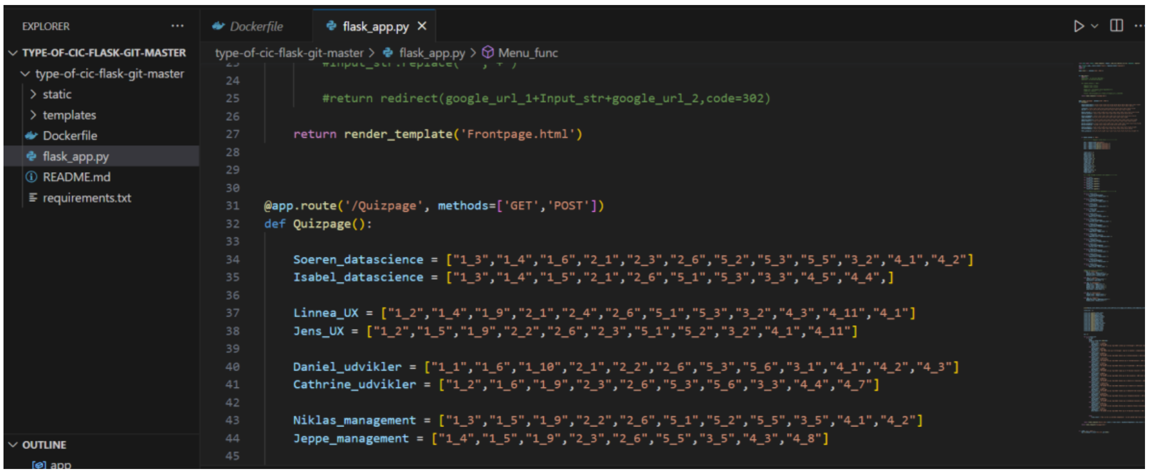
OPENSHIFT

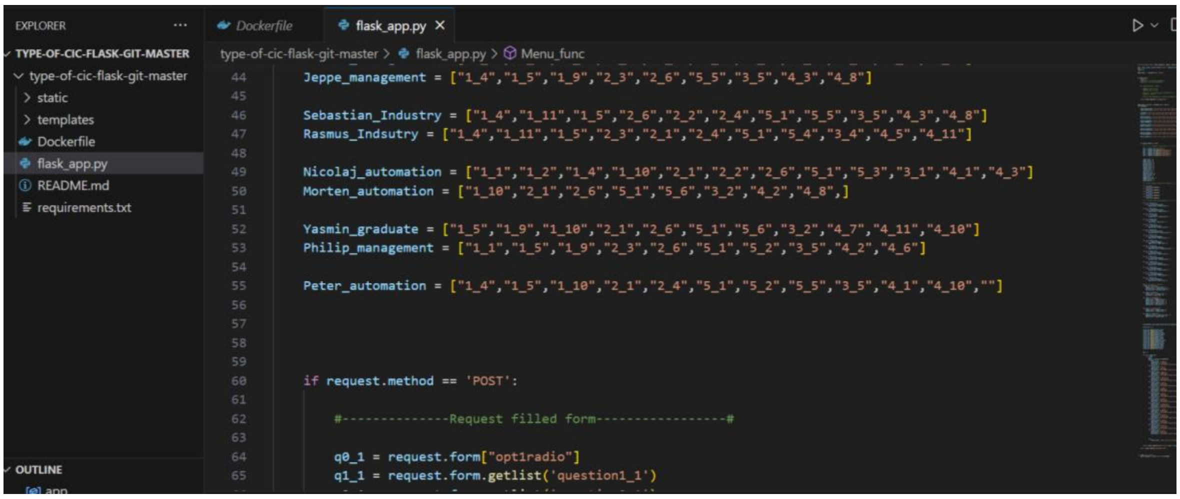
Python-Flask File: -


EXPLORER
TYPE-OF-CIC-FLASK-GIT-MASTER
type-of-cic-flask-git-master
static
templates
Dockerfile
flask_app.py
README.md
requirements.txt
OUTLINE
app
Dockerfile
flask_app.py
type-of-cic-flask-git-master > flask_app.py > Menu_func

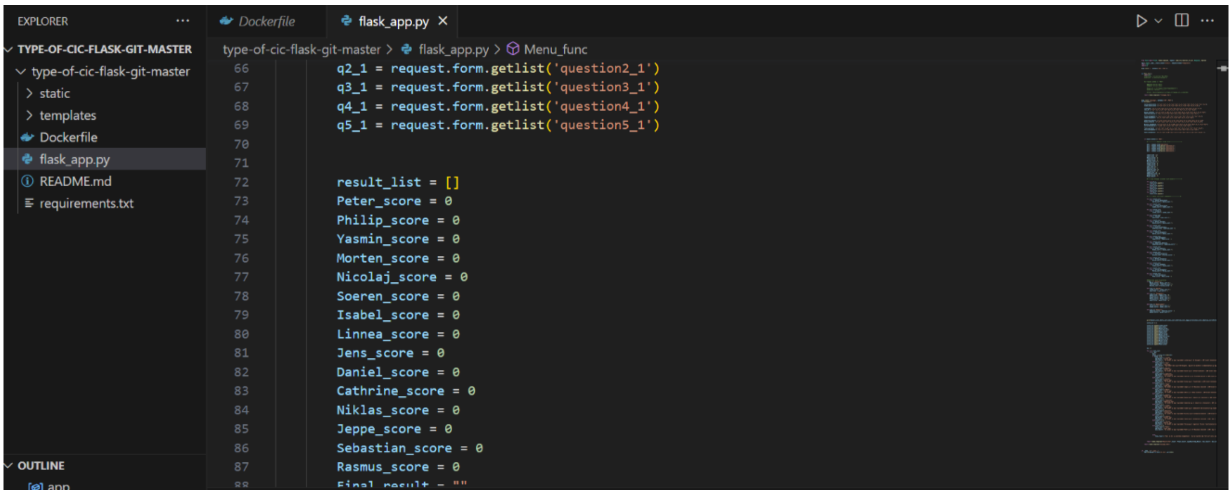
```python
from flask import Flask, render_template, request, send_file,redirect,url_for, Response, redirect

app = Flask(__name__,static_folder="static", template_folder="templates")
import sys
import os

@app.route('/', methods=['GET','POST'])



def Menu_func():
    import os
    #capitalLet = os.environ['AWS_PASS']
    #smallLet = os.environ['aws-pass']


    #if request.method == 'POST':

        #Request form 3.0 hvis:
        #Request form 2.0 hvis:

        #Input_str = str(request.form['GoogleSearch'])
```


EXPLORER
TYPE-OF-CIC-FLASK-GIT-MASTER
type-of-cic-flask-git-master
static
templates
Dockerfile
flask_app.py
README.md
requirements.txt
OUTLINE
app
Dockerfile
flask_app.py
type-of-cic-flask-git-master > flask_app.py > Menu_func

```python

        #return redirect(google_url_1+Input_str+google_url_2,code=302)

    return render_template('Frontpage.html')



@app.route('/Quizpage', methods=['GET','POST'])
def Quizpage():

    Soeren_datascience = ["1_3","1_4","1_6","2_1","2_3","2_6","5_2","5_3","5_5","3_2","4_1","4_2"]
    Isabel_datascience = ["1_3","1_4","1_5","2_1","2_6","5_1","5_3","3_3","4_5","4_4",]

    Linnea_UX = ["1_2","1_4","1_9","2_1","2_4","2_6","5_1","5_3","3_2","4_3","4_11","4_1"]
    Jens_UX = ["1_2","1_5","1_9","2_2","2_6","2_3","5_1","5_2","3_2","4_1","4_11"]

    Daniel_udvikler = ["1_1","1_6","1_10","2_1","2_2","2_6","5_3","5_6","3_1","4_1","4_2","4_3"]
    Cathrine_udvikler = ["1_2","1_6","1_9","2_3","2_6","5_3","5_6","3_3","4_4","4_7"]

    Niklas_management = ["1_3","1_5","1_9","2_2","2_6","5_1","5_2","5_5","3_5","4_1","4_2"]
    Jeppe_management = ["1_4","1_5","1_9","2_3","2_6","5_5","3_5","4_3","4_8"]

```

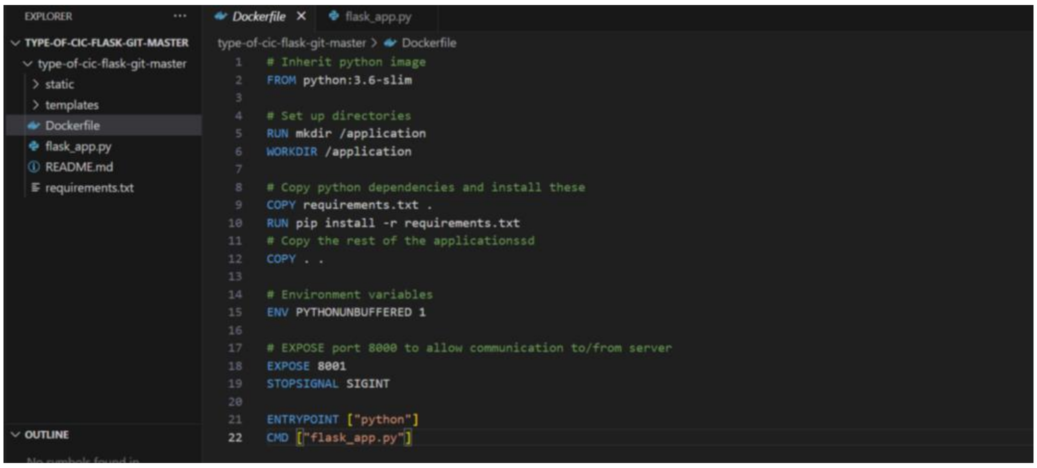

EXPLORER
TYPE-OF-CIC-FLASK-GIT-MASTER
type-of-cic-flask-git-master
static
templates
Dockerfile
flask_app.py
README.md
requirements.txt
OUTLINE
app
Dockerfile
flask_app.py
type-of-cic-flask-git-master > flask_app.py > Menu_func

```python
    Jeppe_management = ["1_4","1_5","1_9","2_3","2_6","5_5","3_5","4_3","4_8"]

    Sebastian_Industry = ["1_4","1_11","1_5","2_6","2_2","2_4","5_1","5_5","3_5","4_3","4_8"]
    Rasmus_Indsutry = ["1_4","1_11","1_5","2_3","2_1","2_4","5_1","5_4","3_4","4_5","4_11"]

    Nicolaj_automation = ["1_1","1_2","1_4","1_10","2_1","2_2","2_6","5_1","5_3","3_1","4_1","4_3"]
    Morten_automation = ["1_10","2_1","2_6","5_1","5_6","3_2","4_2","4_8",]

    Yasmin_graduate = ["1_5","1_9","1_10","2_1","2_6","5_1","5_6","3_2","4_7","4_11","4_10"]
    Philip_management = ["1_1","1_5","1_9","2_3","2_6","5_1","5_2","3_5","4_2","4_6"]

    Peter_automation = ["1_4","1_5","1_10","2_1","2_4","5_1","5_2","5_5","3_5","4_1","4_10",""]




    if request.method == 'POST':

        #-------------Request filled form----------------#

        q0_1 = request.form["opt1radio"]
        q1_1 = request.form.getlist('question1_1')
```


EXPLORER
TYPE-OF-CIC-FLASK-GIT-MASTER
type-of-cic-flask-git-master
static
templates
Dockerfile
flask_app.py
README.md
requirements.txt
OUTLINE
app
Dockerfile
flask_app.py
type-of-cic-flask-git-master > flask_app.py > Menu_func

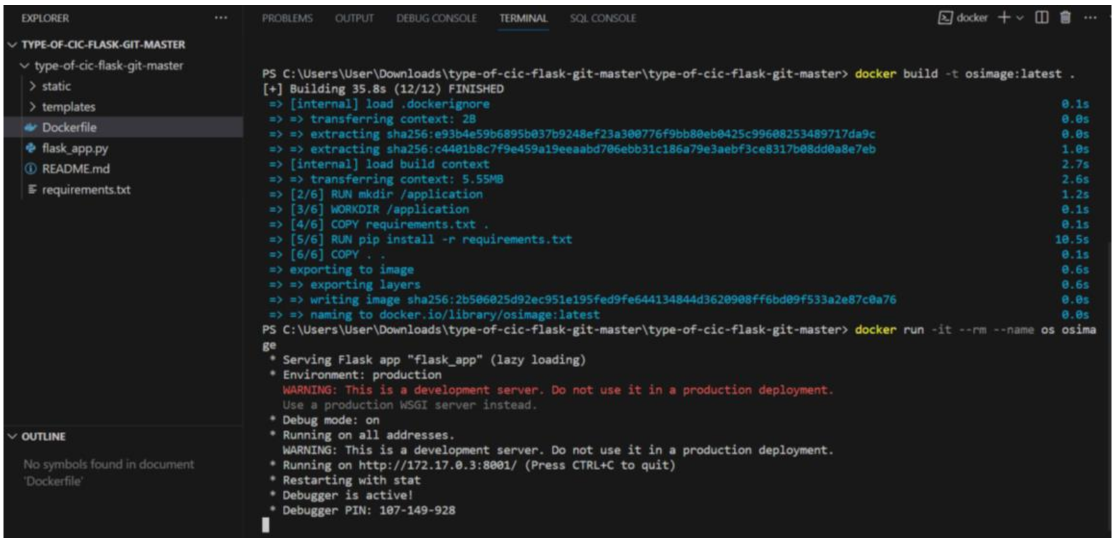
```python
            q2_1 = request.form.getlist('question2_1')
            q3_1 = request.form.getlist('question3_1')
            q4_1 = request.form.getlist('question4_1')
            q5_1 = request.form.getlist('question5_1')


            result_list = []
            Peter_score = 0
            Philip_score = 0
            Yasmin_score = 0
            Morten_score = 0
            Nicolaj_score = 0
            Soeren_score = 0
            Isabel_score = 0
            Linnea_score = 0
            Jens_score = 0
            Daniel_score = 0
            Cathrine_score = 0
            Niklas_score = 0
            Jeppe_score = 0
            Sebastian_score = 0
            Rasmus_score = 0
            Final_result = ""
```

DockerFile:



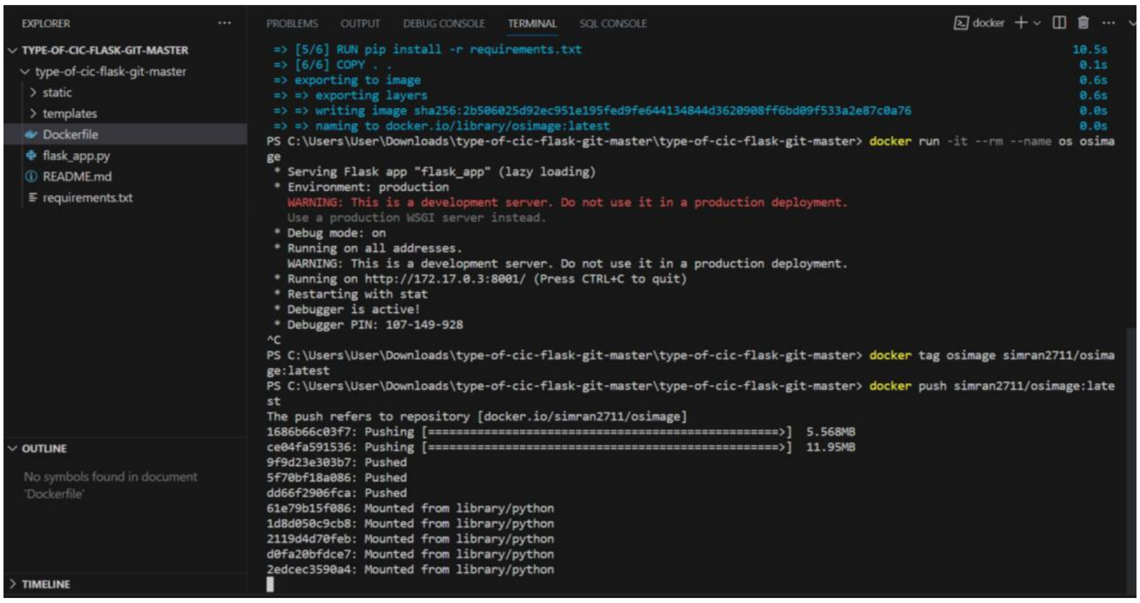
```dockerfile
# Inherit python image
FROM python:3.6-slim

# Set up directories
RUN mkdir /application
WORKDIR /application

# Copy python dependencies and install these
COPY requirements.txt .
RUN pip install -r requirements.txt
# Copy the rest of the applicationssd
COPY . .

# Environment variables
ENV PYTHONUNBUFFERED 1

# EXPOSE port 8000 to allow communication to/from server
EXPOSE 8001
STOPSIGNAL SIGINT

ENTRYPOINT ["python"]
CMD ["flask_app.py"]
```

Docker image building and running the container:



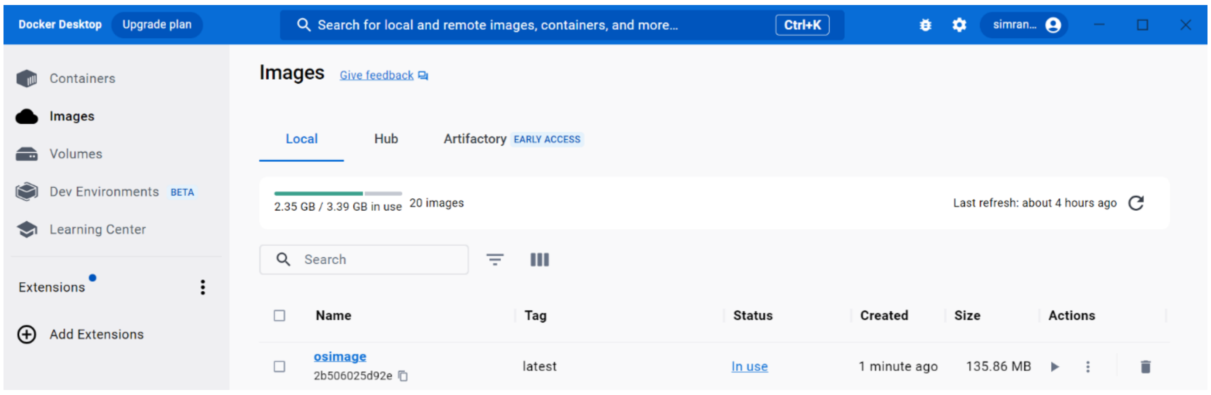
```
PS C:\Users\User\Downloads\type-of-cic-flask-git-master\type-of-cic-flask-git-master> docker build -t osimage:latest .
[+] Building 35.8s (12/12) FINISHED
 => [internal] load .dockerignore                                                                          0.1s
 => => transferring context: 2B                                                                            0.0s
 => => extracting sha256:e93b4e59b6895b037b9248ef23a300776f9bb80eb0425c99608253489717da9c                  0.0s
 => => extracting sha256:c4401b8c7f9e459a19eeaabd706ebb31c186a79e3aebf3ce8317b08dd0a8e7eb                  1.0s
 => [internal] load build context                                                                          2.7s
 => => transferring context: 5.55MB                                                                        2.6s
 => [2/6] RUN mkdir /application                                                                           1.2s
 => [3/6] WORKDIR /application                                                                             0.1s
 => [4/6] COPY requirements.txt .                                                                          0.1s
 => [5/6] RUN pip install -r requirements.txt                                                             10.5s
 => [6/6] COPY . .                                                                                         0.1s
 => exporting to image                                                                                     0.6s
 => => exporting layers                                                                                    0.6s
 => => writing image sha256:2b506025d92ec951e195fed9fe644134844d3620908ff6bd09f533a2e87c0a76               0.0s
 => => naming to docker.io/library/osimage:latest                                                          0.0s
PS C:\Users\User\Downloads\type-of-cic-flask-git-master\type-of-cic-flask-git-master> docker run -it --rm --name os osima
ge
 * Serving Flask app "flask_app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.3:8001/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 107-149-928
```

Image tag and push to docker hub:



```
 => [5/6] RUN pip install -r requirements.txt                                                      10.5s
 => [6/6] COPY . .                                                                                  0.1s
 => exporting to image                                                                              0.6s
 => => exporting layers                                                                             0.6s
 => => writing image sha256:2b506025d92ec951e195fed9fe644134844d3620908ff6bd09f533a2e87c0a76        0.0s
 => => naming to docker.io/library/osimage:latest                                                   0.0s
PS C:\Users\User\Downloads\type-of-cic-flask-git-master\type-of-cic-flask-git-master> docker run -it --rm --name os osima
ge
 * Serving Flask app "flask_app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.3:8001/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 107-149-928
^C
PS C:\Users\User\Downloads\type-of-cic-flask-git-master\type-of-cic-flask-git-master> docker tag osimage simran2711/osima
ge:latest
PS C:\Users\User\Downloads\type-of-cic-flask-git-master\type-of-cic-flask-git-master> docker push simran2711/osimage:late
st
The push refers to repository [docker.io/simran2711/osimage]
1686b66c03f7: Pushing [==================================================>]  5.568MB
ce04fa591536: Pushing [==================================================>]  11.95MB
9f9d23e303b7: Pushed
5f70bf18a086: Pushed
dd66f2906fca: Pushed
61e79b15f086: Mounted from library/python
1d8d050c9cb8: Mounted from library/python
2119d4d70feb: Mounted from library/python
d0fa20bfdce7: Mounted from library/python
2edcec3590a4: Mounted from library/python
```

Image creation:

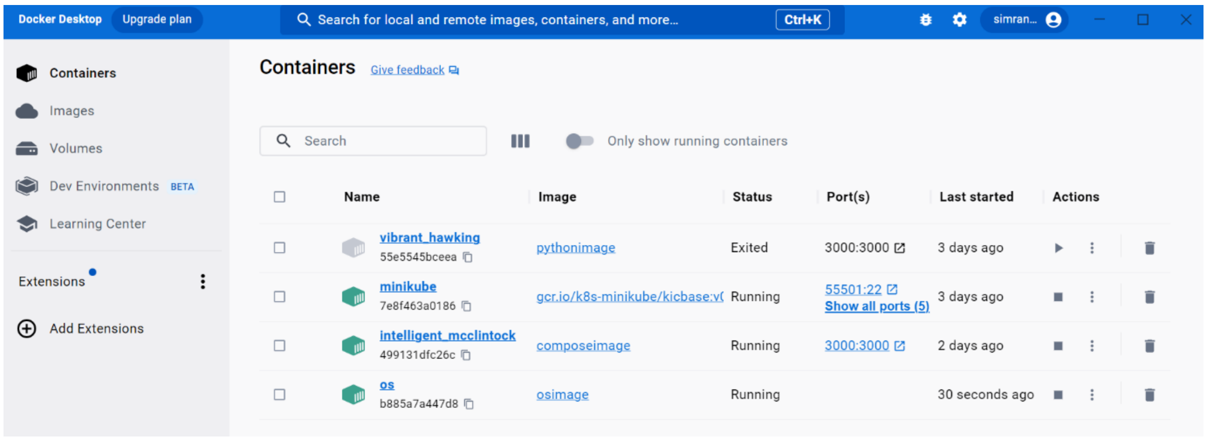


Images

2.35 GB / 3.39 GB in use   20 images   Last refresh: about 4 hours ago

Name	Tag	Status	Created	Size	Actions
osimage 2b506025d92e	latest	In use	1 minute ago	135.86 MB	

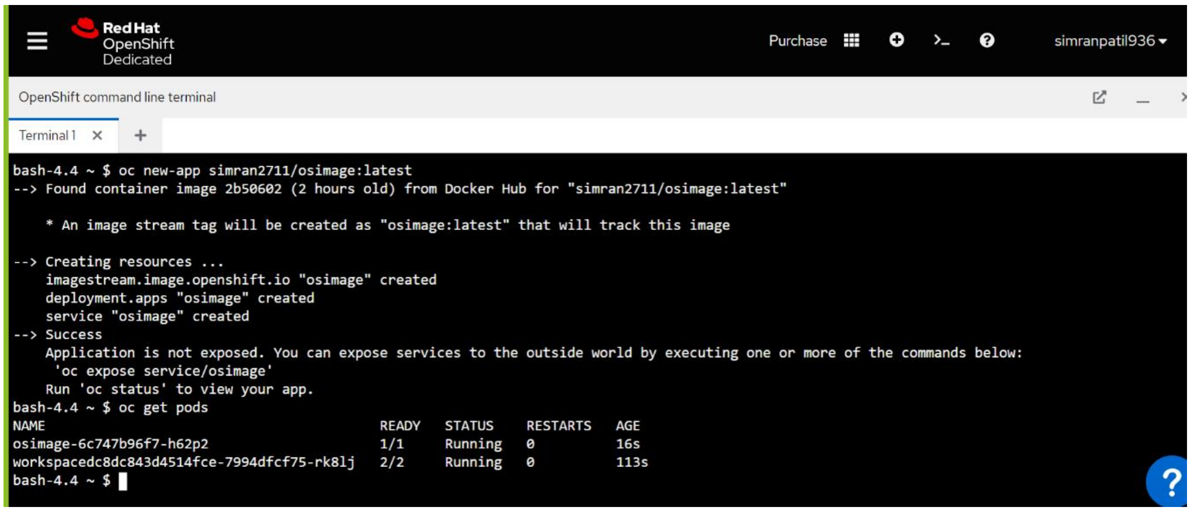
Containers:



Containers

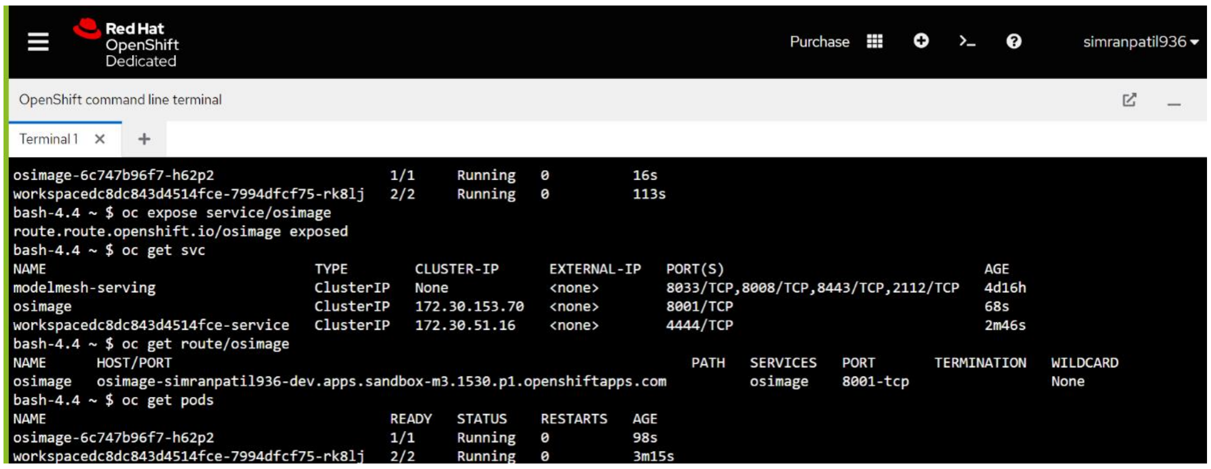
Name	Image	Status	Port(s)	Last started	Actions
vibrant_hawking 55e5545bceea	pythonimage	Exited	3000:3000	3 days ago	
minikube 7e8f463a0186	gcr.io/k8s-minikube/kicbase:v(	Running	55501:22 Show all ports (5)	3 days ago	
intelligent_mcclintock 499131dfc26c	composeimage	Running	3000:3000	2 days ago	
os b885a7a447d8	osimage	Running		30 seconds ago	

Running to pods using CLI:



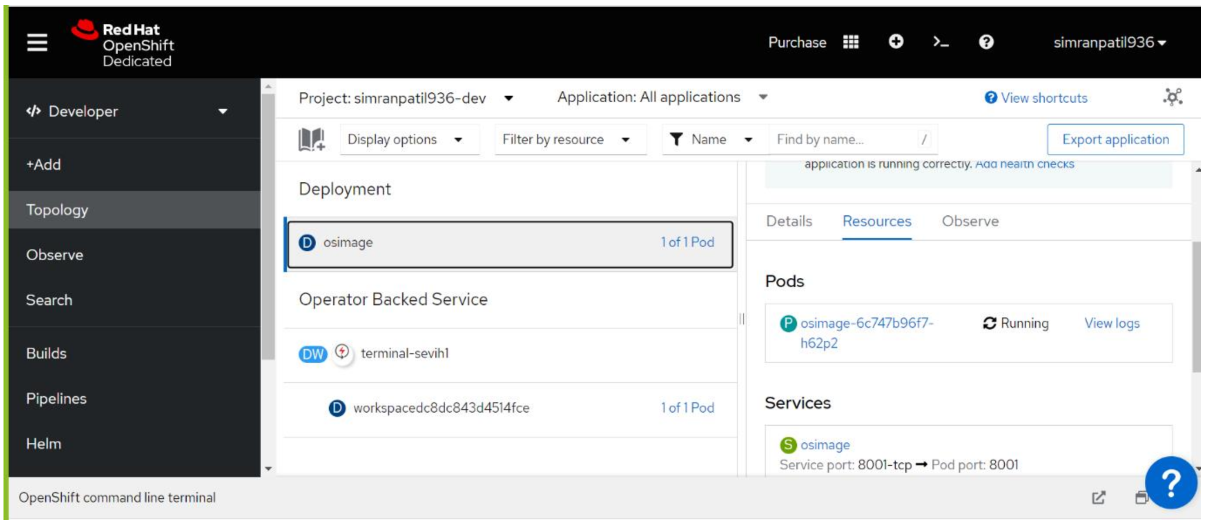
```
bash-4.4 ~ $ oc new-app simran2711/osimage:latest
--> Found container image 2b50602 (2 hours old) from Docker Hub for "simran2711/osimage:latest"

    * An image stream tag will be created as "osimage:latest" that will track this image

--> Creating resources ...
    imagestream.image.openshift.io "osimage" created
    deployment.apps "osimage" created
    service "osimage" created
--> Success
    Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
     'oc expose service/osimage'
    Run 'oc status' to view your app.
bash-4.4 ~ $ oc get pods
NAME                                          READY   STATUS    RESTARTS   AGE
osimage-6c747b96f7-h62p2                      1/1     Running   0          16s
workspacedc8dc843d4514fce-7994dfcf75-rk8lj    2/2     Running   0          113s
bash-4.4 ~ $ 
```

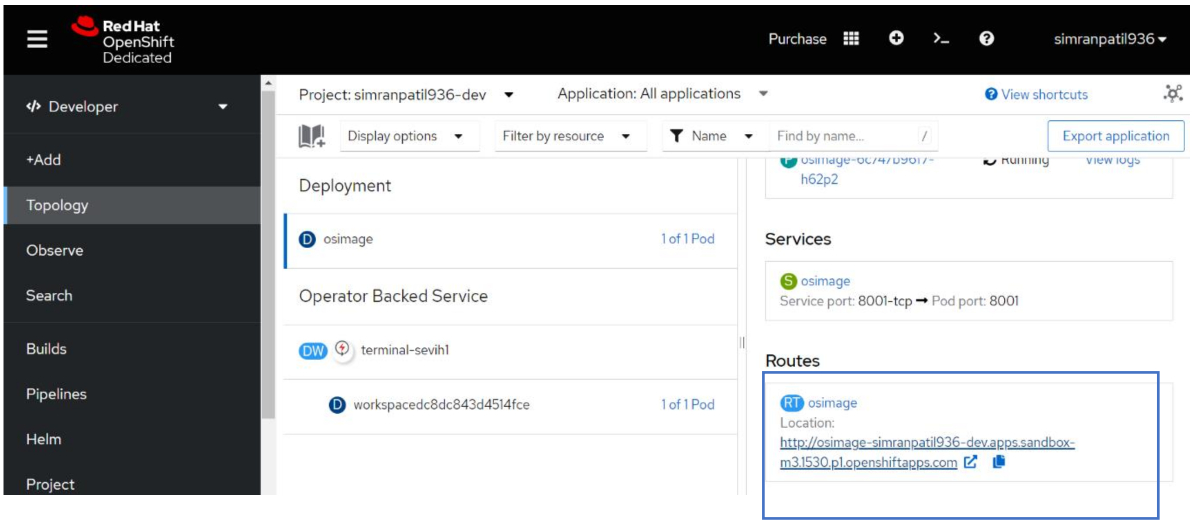
Exposing the route of pods and services



```
osimage-6c747b96f7-h62p2                      1/1     Running   0          16s
workspacedc8dc843d4514fce-7994dfcf75-rk8lj    2/2     Running   0          113s
bash-4.4 ~ $ oc expose service/osimage
route.route.openshift.io/osimage exposed
bash-4.4 ~ $ oc get svc
NAME                                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                                AGE
modelmesh-serving                    ClusterIP   None            <none>        8033/TCP,8008/TCP,8443/TCP,2112/TCP    4d16h
osimage                              ClusterIP   172.30.153.70   <none>        8001/TCP                               68s
workspacedc8dc843d4514fce-service    ClusterIP   172.30.51.16    <none>        4444/TCP                               2m46s
bash-4.4 ~ $ oc get route/osimage
NAME      HOST/PORT                                                       PATH   SERVICES   PORT       TERMINATION   WILDCARD
osimage   osimage-simranpatil936-dev.apps.sandbox-m3.1530.p1.openshiftapps.com          osimage    8001-tcp                 None
bash-4.4 ~ $ oc get pods
NAME                                          READY   STATUS    RESTARTS   AGE
osimage-6c747b96f7-h62p2                      1/1     Running   0          98s
workspacedc8dc843d4514fce-7994dfcf75-rk8lj    2/2     Running   0          3m15s
```

Pod Status

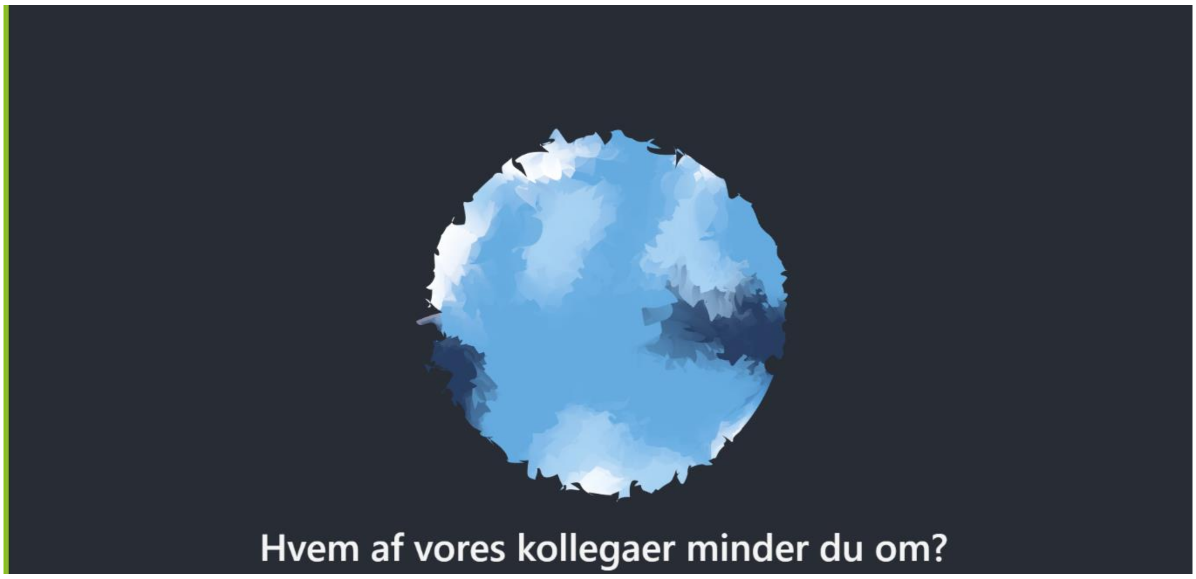


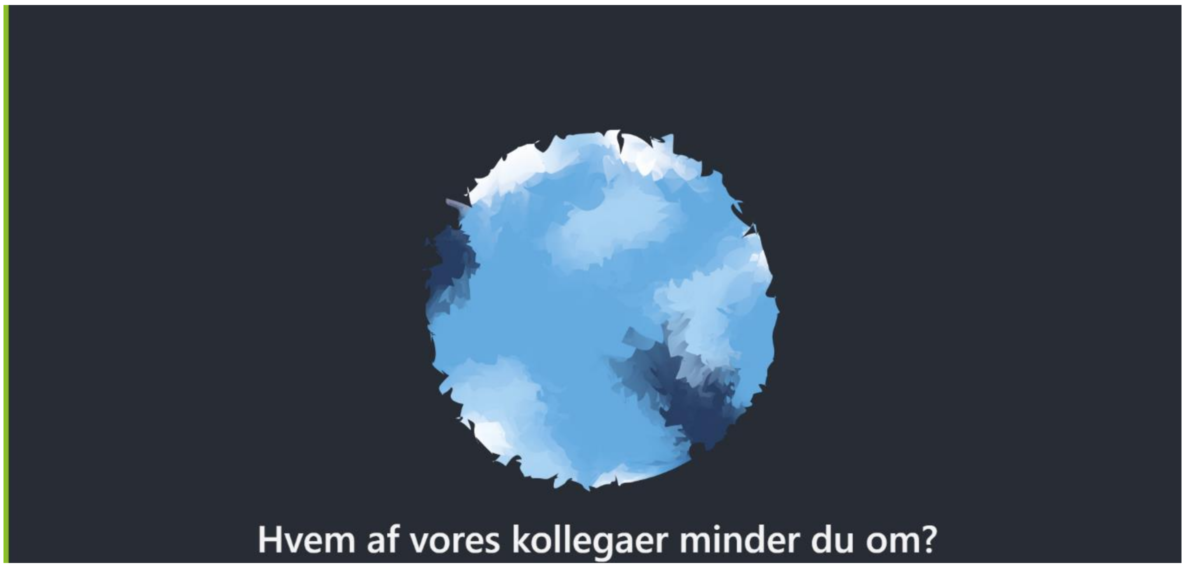


Service port: 8001-tcp → Pod port: 8001

Location:
http://osimage-simranpatil936-dev.apps.sandbox-m3.1530.p1.openshiftapps.com

Final Output:


Hvem af vores kollegaer minder du om?


Hvem af vores kollegaer minder du om?