

1) To pull an image from docker hub and run it in docker playground

```
sethuran@sethu:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sethuran@sethu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mongo         latest   d98599dfdf65   3 months ago   696MB
mongo         <none>   aad77ae58e0c   15 months ago   682MB
redis         latest   aa4d65e670d6   15 months ago   105MB
mongo-express  latest   f8fe1b7702e7   16 months ago   136MB
hello-world    latest   d1165f221234   28 months ago   13.3kB
redis         4.0      191c4017dcdd   2 years ago    89.3MB
sethuran@sethu:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

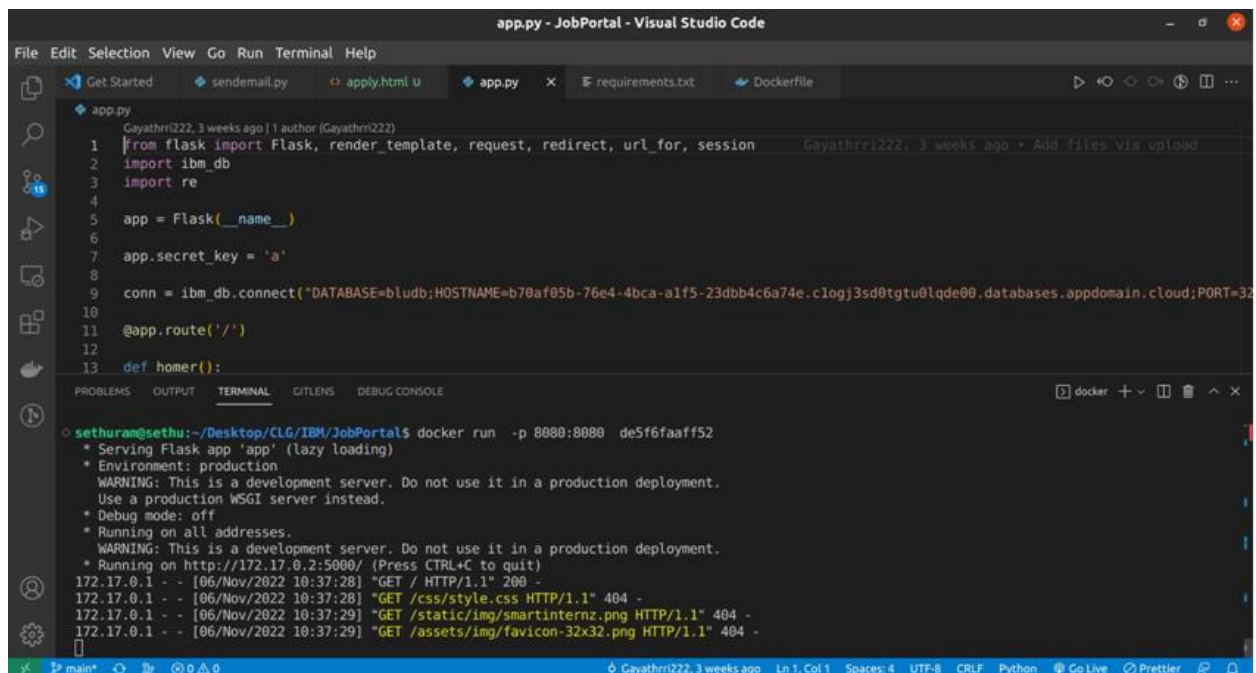
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

sethuran@sethu:~$ docker run mongo
{"t":{"date":"2022-11-06T08:26:09.463+00:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-", "msg":"Initialized wire specification","att
r":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13}}}}
```

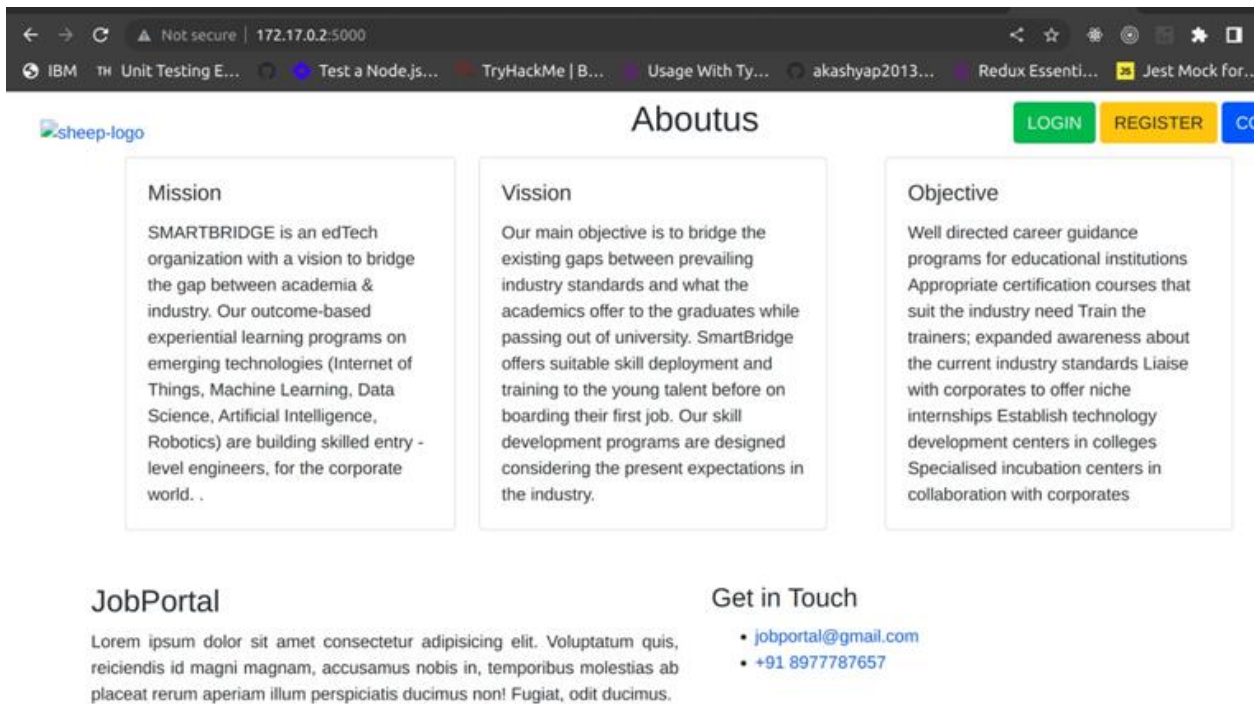
2) Create Docker file for the job portal application and deploy it in Docker Desktop Application



The screenshot shows the Visual Studio Code interface with a file named `app.py` open. The code is a Flask application that connects to a MongoDB database and has a `homer()` endpoint. Below the code editor, the terminal shows the command `docker run -p 8080:8080 de5f6faaff52` and its output, which includes Flask's startup logs and HTTP requests to the application.

```
File Edit Selection View Go Run Terminal Help
Get Started sendemail.py apply.html u app.py x requirements.txt Dockerfile
app.py
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 import re
4
5 app = Flask(__name__)
6
7 app.secret_key = 'a'
8
9 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b78af05b-76e4-4bca-a1f5-23dbb4c6a74e.clogj3sd0tgu0lqde00.databases.appdomain.cloud;PORT=32
10
11 @app.route('/')
12
13 def homer():

PROBLEMS OUTPUT TERMINAL GITLENS DEBUG CONSOLE
sethuran@sethu:~/Desktop/CLG/IBM/JobPortal$ docker run -p 8080:8080 de5f6faaff52
* Serving Flask app '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: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [06/Nov/2022 10:37:28] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [06/Nov/2022 10:37:28] "GET /css/style.css HTTP/1.1" 404 -
172.17.0.1 - - [06/Nov/2022 10:37:29] "GET /static/img/smartinternz.png HTTP/1.1" 404 -
172.17.0.1 - - [06/Nov/2022 10:37:29] "GET /assets/img/favicon-32x32.png HTTP/1.1" 404 -
```



- 3) Create a IBM Container Registry and deploy job portal app or hello world app
- 4)

```
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ ibmcloud cr namespace-add sethuran52001
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

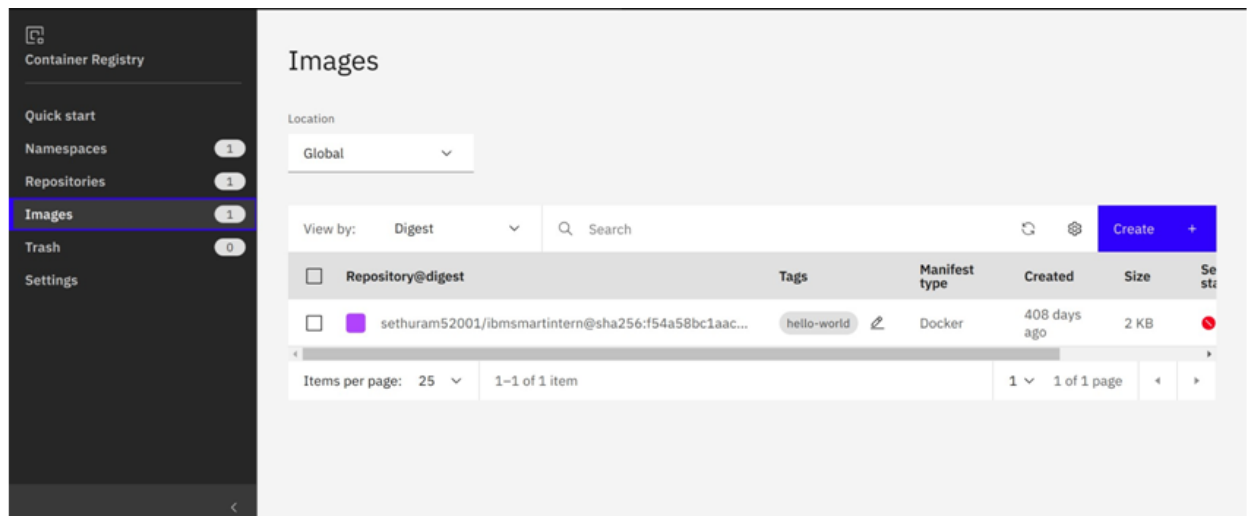
Adding namespace 'sethuran52001' in resource group 'Default' for account Sethuran Venkatesan's Account in registry icr.io...
Successfully added namespace 'sethuran52001'

OK
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ ibmcloud cr login
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.

OK
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ docker tag hello-world icr.io/sethuran52001/ibmsmartintern:hello-world
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ docker push icr.io/sethuran52001/ibmsmartintern:hello-world
The push refers to repository [icr.io/sethuran52001/ibmsmartintern]
e07ee1baac5f: Pushed
hello-world: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$ ibmcloud cr image-list
Listing images...

Repository          Tag          Digest          Namespace       Created      Size      Security status
icr.io/sethuran52001/ibmsmartintern hello-world    f54a58bc1aac    sethuran52001  1 year ago  2.5 kB    -

OK
sethuran@sethu:~/Downloads/IBM_Cloud_CLI_2.12.1_and64/Bluenix_CLI$
```



5) Create Kubernetes Cluster in IBM cloud and deploy hello world image or job portal image and expose the app to run in nodeport

