

# Deployment Of App In IBM Cloud

## Containerize The App

Team ID	PNT2022TMID04791
Project Name	Plasma Donor Application

```
Administrator: Command Prompt
Cloud integration: v1.0.29
Version: 20.10.20
API version: 1.41
Go version: go1.18.7
Git commit: 9fdeb9c
Built: Tue Oct 18 18:28:44 2022
OS/Arch: windows/amd64
Context: default
Experimental: true

Server: Docker Desktop 4.13.0 (89412)
Engine:
Version: 20.10.20
API version: 1.41 (minimum version 1.12)
Go version: go1.18.7
Git commit: 03df974
Built: Tue Oct 18 18:18:35 2022
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.6.8
GitCommit: 9cd3357b7f7d7218e4aec3cae239db1f68a5a6ec6
name:
Version: 1.1.4
GitCommit: v1.1.4-0-g5f4dc4d
docker-init:
Version: 0.19.0
GitCommit: d64bad9

C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
C:\WINDOWS\system32>
```

```
Administrator: Command Prompt
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
C:\WINDOWS\system32>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
e9de957aee07: Pull complete
Digest: sha256:4b1d0c4a2d2aaf63b3711f34eb9fa89fa1bf3dd6e4ca954d47caebca4805c2
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

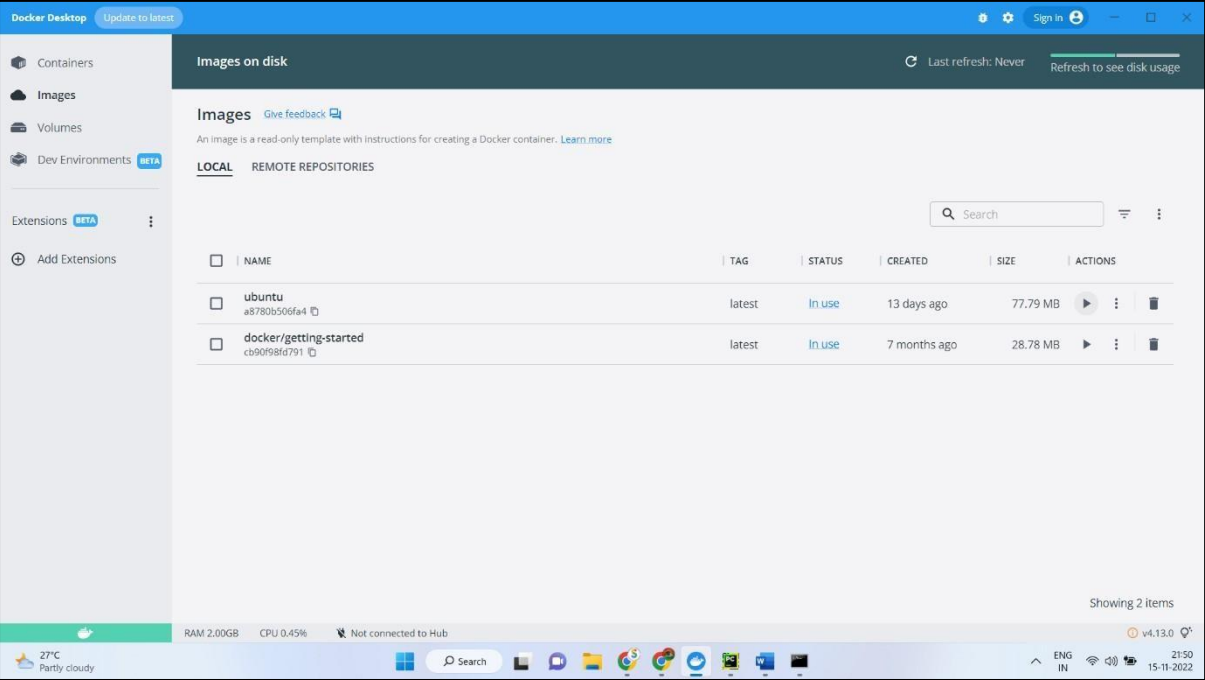
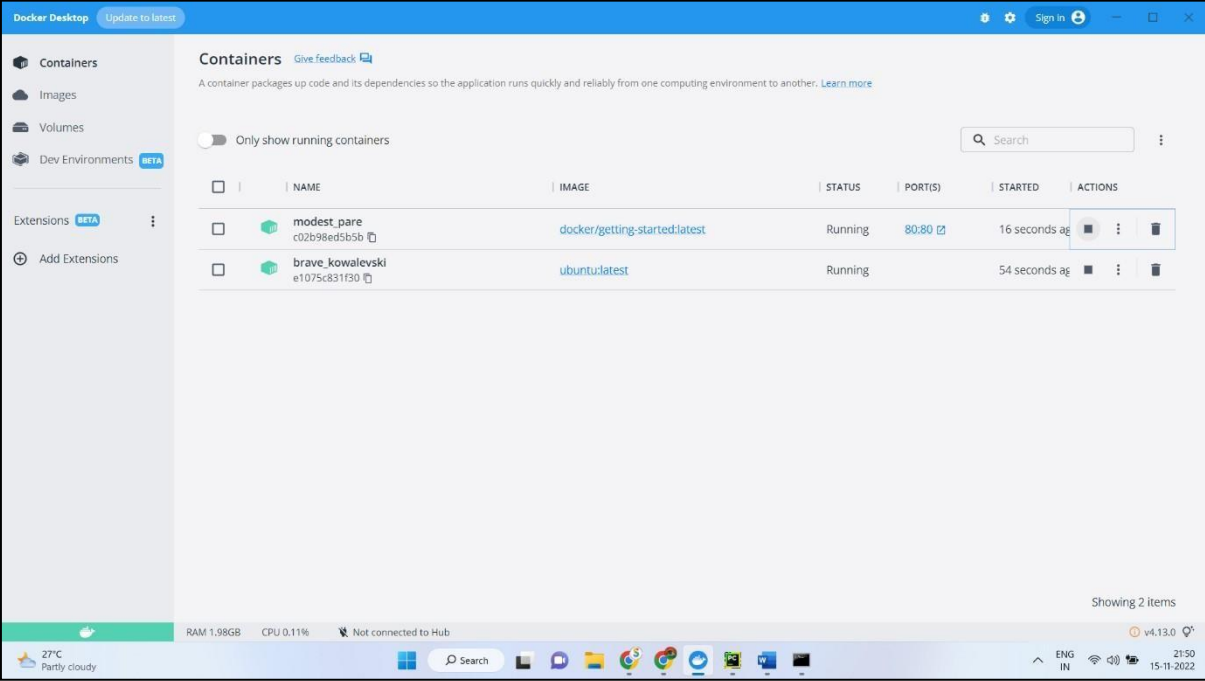
C:\WINDOWS\system32>docker pull plasma donar app
"docker pull" requires exactly 1 argument.
See 'docker pull --help'.

Usage:  docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Pull an image or a repository from a registry

C:\WINDOWS\system32>docker pull plasmadonar
Using default tag: latest
Error response from daemon: pull access denied for plasmadonar, repository does not exist or may require 'docker login': denied: requested access to the resource is denied

C:\WINDOWS\system32>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest   a8780b506faa   12 days ago    77.8MB
docker/getting-started latest   cb90f98fd791   7 months ago   28.8MB
C:\WINDOWS\system32>
```



The screenshot shows the PyCharm IDE interface. The main editor displays the `main.py` file with the following code:

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route("/")
6 def home():
7     return "<h1>Hello, World!</h1> \
8         "<p> Welcome to Plasma Donor Application</p>"
9
10 if __name__ == '__main__':
11     app.run(debug=True)
```

The Run console at the bottom shows the following output:

```
Run: main
"C:\Users\ransa\PycharmProjects\python flask\venv\Scripts\python.exe" "C:/Users/ransa/PycharmProjects/python flask/main.py"
+ Serving Flask app 'main'
+ Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5986
Press CTRL+C to quit
+ Restarting with stat
+ Debugger is active!
+ Debugger PIN: 169-685-378
127.0.0.1 - - [15/Nov/2022 21:42:08] "GET / HTTP/1.1" 200 -
```

