

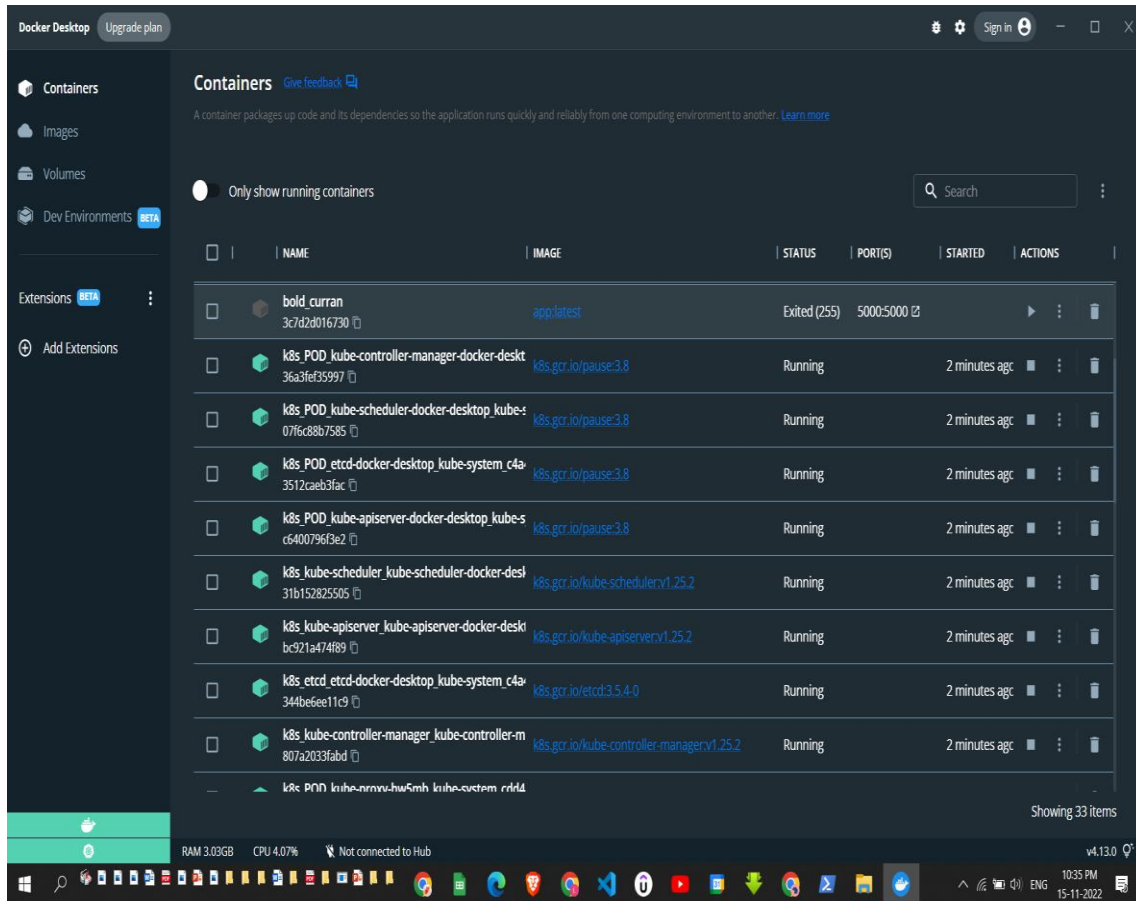
Containerizing the application

1. Docker image creation

```
278 app.run(host='0.0.0.0', debug=True)
```

```
PS C:\Users\posit\Pictures\z\IBM\Project Development Phase\Sprint 2> docker build -t apps:latest .
[+] Building 72.4s (9/10)
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 32B                                              0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.9                  3.5s
=> [1/6] FROM docker.io/library/python:3.9@sha256:475fe86bf1da48ea27009a8f7d7e96231af4142de918a68010d48d0abb9c9c5 0.0s
=> [internal] load build context                                              0.1s
=> => transferring context: 31.27kB                                           0.1s
=> CACHED [2/6] RUN apt-get update                                             0.0s
=> CACHED [3/6] RUN mkdir /app                                                0.0s
=> CACHED [4/6] WORKDIR /app                                                  0.0s
=> [5/6] COPY . /app                                                         0.0s
=> [6/6] RUN pip install -r requirements.txt                                  68.3s
=> => Collecting ibm-db
=> => # Downloading ibm_db-3.1.3.tar.gz (1.4 MB)
=> => # 1.4/1.4 MB 1.4 MB/s eta 0:00:00
=> => # Installing build dependencies: started
=> => # Installing build dependencies: finished with status 'done'
=> => # Getting requirements to build wheel: started
```

2. Docker image for flask application



The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (with a BETA badge), Extensions (with a BETA badge), and Add Extensions. The main area is titled 'Containers' and includes a 'Give feedback' link. Below the title, there is a toggle for 'Only show running containers' and a search bar. A table lists the containers, with columns for NAME, IMAGE, STATUS, PORT(S), STARTED, and ACTIONS. The table shows one exited container and several running containers, including Kubernetes components. The bottom status bar displays system information: RAM 3.03GB, CPU 4.07%, and a 'Not connected to Hub' status. The taskbar at the bottom shows various application icons and the system clock indicating 10:35 PM on 15-11-2022.

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
bold_curran 3c7d2d016730	app:latest	Exited (255)	5000:5000		
k8s_POD_kube-controller-manager-docker-deskt 36a3fef35997	k8s.gcr.io/pause:3.8	Running		2 minutes ago	
k8s_POD_kube-scheduler-docker-desktop_kube-s 07fc88b7585	k8s.gcr.io/pause:3.8	Running		2 minutes ago	
k8s_POD_etcd-docker-desktop_kube-system_c4a 3512caeb3fac	k8s.gcr.io/pause:3.8	Running		2 minutes ago	
k8s_POD_kube-apiserver-docker-desktop_kube-s c6400796f3e2	k8s.gcr.io/pause:3.8	Running		2 minutes ago	
k8s_kube-scheduler_kube-scheduler-docker-desk 31b152825505	k8s.gcr.io/kube-schedulerv1.25.2	Running		2 minutes ago	
k8s_kube-apiserver_kube-apiserver-docker-desk bc921a474f89	k8s.gcr.io/kube-apiserverv1.25.2	Running		2 minutes ago	
k8s_etcd_etcd-docker-desktop_kube-system_c4a 344befe11c9	k8s.gcr.io/etcd:3.5.4-0	Running		2 minutes ago	
k8s_kube-controller-manager_kube-controller-m 807a2033fabd	k8s.gcr.io/kube-controller-managerv1.25.2	Running		2 minutes ago	
k8s_POD_kube-nmnu-hw5mh_kube-system_cdd4					

Showing 33 items

RAM 3.03GB CPU 4.07% Not connected to Hub v4.13.0

10:35 PM 15-11-2022