

Q3 – Docker and Container Internals

1. Cloning the repository

```
root@master:~# git clone https://github.com/devops-experience/k8s-helloworld.git
Cloning into 'k8s-helloworld'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 2), reused 1 (delta 1), pack-reused 4 (from 1)
Unpacking objects: 100% (9/9), 2.29 KiB | 780.00 KiB/s, done.
root@master:~# cd k8s-helloworld/
root@master:~/k8s-helloworld# ls
app.py  README.md  requirements.txt
```

2. Writing Docker file

```
GNU nano 4.8 Dockerfile
FROM python:3.8-slim
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["sh", "-c", "python app.py"]
```

3. Docker Image Building

```
root@master:~/k8s-helloworld# docker build -t gowri5877/k8shelloworldimage .
[+] Building 28.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.0s
=> => transferring dockerfile: 162B                                              0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim                2.7s
=> [auth] library/python:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/4] FROM docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613 0.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 29.87kB                                               0.0s
=> CACHED [2/4] WORKDIR /app                                                       0.0s
=> [3/4] COPY . .                                                                0.1s
=> [4/4] RUN pip install -r requirements.txt                                     25.1s
=> exporting to image                                                            0.3s
=> => exporting layers                                                            0.3s
=> => writing image sha256:64dd504c0b375612385e27bf847154df8096a31afe548b4f968978fe810d9bb5 0.0s
=> => naming to docker.io/gowri5877/k8shelloworldimage                         0.0s
```

4. Running the docker container

```
root@master:~/k8s-helloworld# docker run -dit gowri5877/k8shelloworldimage
dfa3c95b486768b57f0cd349cb556a6aebcd4327d7e6edf0401cbb6ce00b71f05
```

5. Read write layer on host is present in /var/lib/docker/containers

And in that dfa3c... folder

```
root@master:/var/lib/docker/containers# ls
226dceec820ec3f8f9c07f90077c9af6b65e2ec1eaf3227672e3ca9784da5565
7acbda153f85441ead40bca075dbcb7f8b9bba314866df1d71c5f53ae596c941
7d2dab7d64a307c7774e7e861fdbcb8fde867700c2b4c09d0b5dea564d7bbeaa
99c9ae9a398ebadb3c534e200f2b984fc24ffdbeae1c7ddadb2ab8f1d5bef875
9f8eb0a6ae80807b3d6baf995ed387995648121eb8cf8f7929f57a011f80d7c2
dfa3c95b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05
```

6. Working directory of the container

```
root@master:~/k8shellworld# docker exec -it dfa pwd
/app
```

7. Writable layer directory

```
{
  "StartedAt": "2025-04-09T09:05:38.622775172Z",
  "FinishedAt": "0001-01-01T00:00:00Z"
},
{
  "Image": "sha256:64dd504c0b375612385e27bf847154df8096a31afe548b4f968978fe810d9bb5",
  "ResolvConfPath": "/var/lib/docker/containers/dfa3c95b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05/
  solv.conf",
  "HostnamePath": "/var/lib/docker/containers/dfa3c95b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05/ho
  name",
  "HostsPath": "/var/lib/docker/containers/dfa3c95b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05/hosts
  ",
  "LogPath": "/var/lib/docker/containers/dfa3c95b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05/dfa3c95
  b486768b57f0cd349cb556a6aebd4327d7e6edf0401cbb6ce00b71f05-json.log",
}
```

Commands used

Git clone <https://github.com/Gowrisankar5877/k8s-helloworld.git>

cd k8s-helloworld/

ls

nano Dockerfile

docker build -t gowri5877/k8shellworldimage .

docker run -dit gowri5877/k8shellworldimage

docker ps

cd /var/lib/docker/containers/

docker exec -it dfa pwd

docker container inspect dfa