


EXPERIMENT 5

AIM: Working with Dockerfile to Build and Push Docker Image

Steps to Complete:

1. Create following Dockerfile

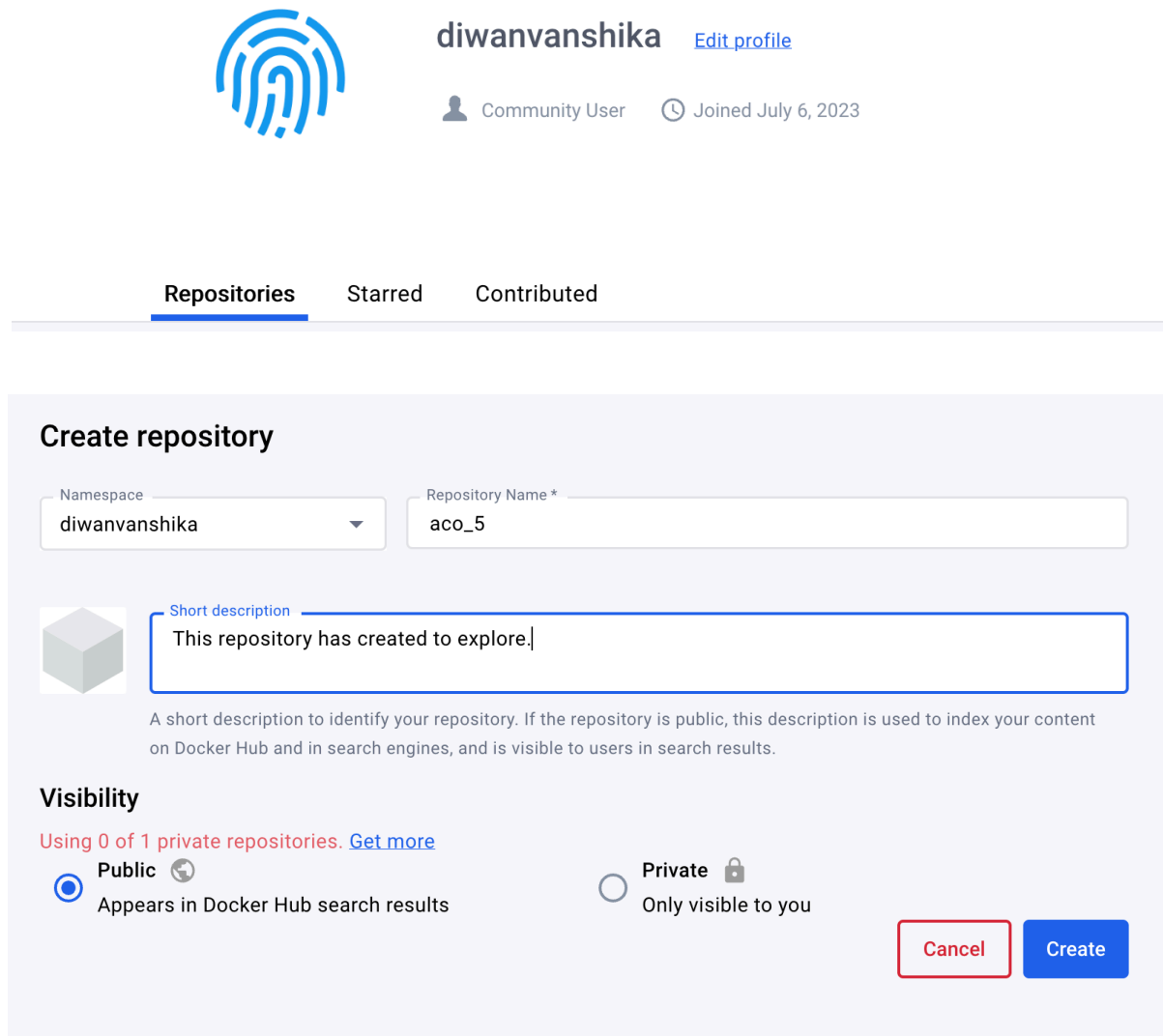
```
 Dockerfile
1  #Create following Dockerfile
2  FROM alpine
3  MAINTAINER SUDIPT
4  RUN apk update
5  RUN apk add nodejs
6  RUN mkdir /app
7  COPY index.js /app
8  WORKDIR /app
9  RUN node index.js
10 |
```

2. Now we have dockerized the app, we will Build image from Dockerfile.

```
docker build -t myimage:0.1 .
```

```
● (base) vanshika@VANSHIKAS-MacBook-Air AC0 % docker build -t myimage:0.1 .
[+] Building 9.7s (12/12) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 243B                                0.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load metadata for docker.io/library/alpine:latest  0.0s
=> CACHED [1/7] FROM docker.io/library/alpine                    0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 29B                                       0.0s
=> [2/7] RUN apk update                                           2.3s
=> [3/7] RUN apk add nodejs                                       6.6s
=> [4/7] RUN mkdir /app                                           0.3s
=> [5/7] COPY index.js /app                                       0.0s
=> [6/7] WORKDIR /app                                             0.0s
=> [7/7] RUN node index.js                                        0.3s
=> exporting to image                                             0.2s
=> => exporting layers                                              0.2s
=> => writing image sha256:04ced233c1615ff674299c14e38dac3049058757054de978cf47daed7b7b91 0.0s
=> => naming to docker.io/library/myimage:0.1                     0.0s
```

3. Create account on Dockerhub and create a repository in it.



The screenshot shows the Docker Hub profile for user 'diwanvanshika'. The profile includes a blue fingerprint icon, the username 'diwanvanshika' with an 'Edit profile' link, and the status 'Community User' with a 'Joined July 6, 2023' timestamp. Below the profile are tabs for 'Repositories', 'Starred', and 'Contributed'. The 'Repositories' tab is active, showing a 'Create repository' form. The form has a 'Namespace' dropdown set to 'diwanvanshika' and a 'Repository Name' field set to 'aco_5'. A 'Short description' field contains the text 'This repository has created to explore.' Below the description is a note: 'A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.' Under the 'Visibility' section, the 'Public' option is selected, with the text 'Using 0 of 1 private repositories. Get more' and 'Appears in Docker Hub search results'. The 'Private' option is unselected, with the text 'Only visible to you'. At the bottom right are 'Cancel' and 'Create' buttons.

diwanvanshika [Edit profile](#)

Community User Joined July 6, 2023

Repositories Starred Contributed

Create repository


Namespace: diwanvanshika Repository Name *: aco_5


Short description: This repository has created to explore.

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ Public  Appears in Docker Hub search results

☐ Private  Only visible to you

[Cancel](#) [Create](#)

4. Tag the recently created image using following command.

```
docker tag imageID Repositoryname
```

```
(base) vanshika@VANSHIKAs-MacBook-Air ACO % docker tag 04ced233c161 diwanvanshika/aco_5
```

5. Login to Dockerhub from console using following command.

`docker Login`

```
● (base) vanshika@VANSHIKAs-MacBook-Air AC0 % docker login

Authenticating with existing credentials...
Login Succeeded
```

6. Now push the image on Dockerhub using following command.

`docker push Repositoryname`

```
● (base) vanshika@VANSHIKAs-MacBook-Air AC0 % docker push diwanvanshika/aco_5
Using default tag: latest
The push refers to repository [docker.io/diwanvanshika/aco_5]
f3d692204a6b: Pushed
5f70bf18a086: Pushed
64ffdc148181: Pushed
3c2597658b61: Pushed
4f900eaede2c: Pushed
526404c401b2: Pushed
0faf9b67be60: Pushed
latest: digest: sha256:3f7c203ffeaef8237c9e89b46918c48e44937535f35da010f77676d58ce9737c size: 1777
○ (base) vanshika@VANSHIKAs-MacBook-Air AC0 %
```

7. Pull and Run the container of your deployed image on docker hub.

`Docker run -d image_name`

```
● (base) vanshika@VANSHIKAs-MacBook-Air AC0 % docker run -d myimage:0.1
edadabc19ff375f069b1d4c9a61157f9e0cacc8cad2bcb62dbc9f85c0ea34a4b
○ (base) vanshika@VANSHIKAs-MacBook-Air AC0 %
```