

# Dockert Task 2

## *Task Description:*

Create a dockerfile, docker-compose file which when executed must display your basic details in the website.

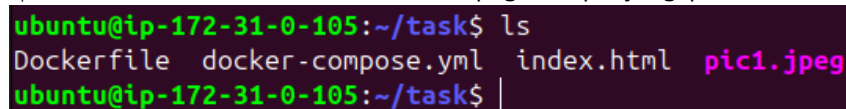
## *Solution:*

## Personal Profile Website with Docker

This project creates a simple personal profile website using Docker and docker-compose.

### Project Structure

```
task/
├── Dockerfile          # Instructions to build the Docker image
├── docker-compose.yml  # Service configuration for docker-compose
└── index.html          # The webpage displaying personal details
```



```
ubuntu@ip-172-31-0-105:~/task$ ls
Dockerfile  docker-compose.yml  index.html  pic1.jpeg
ubuntu@ip-172-31-0-105:~/task$ |
```

### Dockerfile

```
FROM nginx:alpine
```

```
# Copy custom HTML file to replace default nginx page
COPY index.html /usr/share/nginx/html/index.html
```

```
# Expose port 80
EXPOSE 80
```

```
# Default command to start nginx
CMD ["nginx", "-g", "daemon off;"]
```

```
ubuntu@ip-172-31-0-105: ~/task
GNU nano 6.2 Dockerfile
FROM nginx:alpine

# Copy custom HTML file to replace default nginx page
COPY index.html /usr/share/nginx/html/index.html

# Expose port 80
EXPOSE 80

# Default command to start nginx
CMD ["nginx", "-g", "daemon off;"]
```

## Docker-compose.yml

version: '3'

services:

webapp:

build:

context: .

dockerfile: Dockerfile

container\_name: my-profile-webapp

ports:

- "8080:80"

restart: unless-stopped

networks:

- profile-network

networks:

profile-network:

driver: bridge

```
ubuntu@ip-172-31-0-105: ~/task
GNU nano 6.2 docker-compose.yml
version: '3'

services:
  webapp:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: my-profile-webapp
    ports:
      - "8080:80"
    restart: unless-stopped
    networks:
      - profile-network

networks:
  profile-network:
    driver: bridge
```

## Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>My Personal Profile</title>

<style>

body {

font-family: Arial, sans-serif;

line-height: 1.6;

margin: 0;

padding: 0;

background-color: hsl(170, 95%, 46%);

color: #333;

}

.container {

width: 80%;

```
    max-width: 800px;
    margin: 0 auto;
    padding: 2rem;
    background-color: #4ebba0;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    border-radius: 5px;
    margin-top: 2rem;
}
```

```
h1 {
    color: #2c3e50;
    border-bottom: 2px solid #3498db;
    padding-bottom: 0.5rem;
}
```

```
.info-group {
    margin-bottom: 1.5rem;
}
```

```
.info-label {
    font-weight: bold;
    color: #3498db;
}
```

```
.profile-pic {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    margin: 1rem auto;
    display: block;
    object-fit: cover;
    box-shadow: 0 0 5px rgba(0,0,0,0.2);
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
  
```

```
  <h1>My Personal Profile</h1>
```

```
  <div class="info-group">
```

```
    <p><span class="info-label">Name:</span> Surya Pkh</p>
```

```
    <p><span class="info-label">Role:</span> DevOps Engineer</p>
```

```
    <p><span class="info-label">Email:</span> Suryaprakash27032001@gmail.com</p>
```

```
  </div>
```

```
  <div class="info-group">
```

```
    <h2>About Me</h2>
```

```
    <p>I am a passionate DevOps Engineer with experience in Docker, Kubernetes, and AWS. I  
    enjoy automating deployment processes and building efficient CI/CD pipelines.</p>
```

```
  </div>
```

```
  <div class="info-group">
```

```
    <h2>Skills</h2>
```

```
    <p><span class="info-label">DevOps:</span> Docker, Kubernetes, Terraform, Ansible</p>
```

```
    <p><span class="info-label">Cloud:</span> AWS, Azure, GCP</p>
```

```
    <p><span class="info-label">Programming:</span> Python, JavaScript, Bash</p>
```

```
    <p><span class="info-label">CI/CD:</span> Jenkins, GitHub Actions, GitLab CI</p>
```

```
  </div>
```

```
  <div class="info-group">
```

```
    <h2>Hobbies</h2>
```

```
    <p>Hiking, Photography, Reading Technical Books, Contributing to Open Source</p>
```

```
  </div>
```

```
  <div class="info-group">
```

```
    <p><em>Container ID: </em><span id="container-id">Loading...</span></p>
```

```
  </div>
```

<script>

```
// Simulate loading container ID
```

```
document.getElementById('container-id').textContent = 'Docker Container';
```

&lt;/script&gt;

&lt;/body&gt;

```

GNU nano 6.2                                index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Personal Profile</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 0;
      padding: 0;
      background-color: hsl(170, 95%, 46%);
      color: #333;
    }
    .container {
      width: 80%;
      max-width: 800px;
      margin: 0 auto;
      padding: 2rem;
    }
  </style>
</html>

```

[ Read 88 lines ]

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location  
 ^X Exit    ^R Read File    ^\_ Replace    ^U Paste    ^J Justify    ^\_ Go To Line

## How it Works

1. The `Dockerfile` creates an image based on Nginx Alpine (lightweight) and copies our custom HTML file
2. The `docker-compose.yml` file defines the service configuration and networking
3. The `index.html` file contains a simple personal profile page

## Instructions

## Prerequisites

- ## 10 Docker installed on your machine

## ⑩ Docker Compose installed on your machine

### Running the Application

1. Create all the files in a new directory:

```
mkdir task
cd task
# Create Dockerfile, docker-compose.yml, and index.html using your editor
```

```
ubuntu@ip-172-31-0-105:~$ mkdir task
ubuntu@ip-172-31-0-105:~$ ls
composetest  shell.sh  task
ubuntu@ip-172-31-0-105:~$ cd task
ubuntu@ip-172-31-0-105:~/task$ ls
ubuntu@ip-172-31-0-105:~/task$ nano Dockerfile
ubuntu@ip-172-31-0-105:~/task$ nano docker-compose.yml
ubuntu@ip-172-31-0-105:~/task$ nano index.html
```

2. Build and start the container:

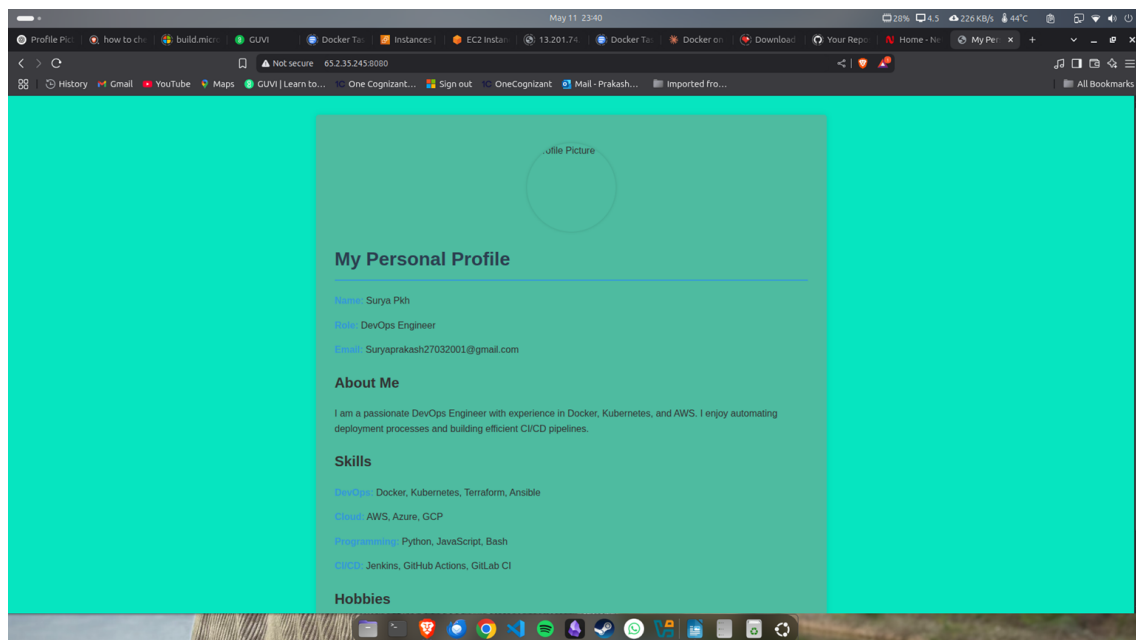
```
docker-compose up -d
```

```
ubuntu@ip-172-31-0-105:~/task$ docker compose up -d
WARN[0000] /home/ubuntu/task/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 4.6s (8/8) FINISHED                                docker:default
=> [webapp internal] load build definition from Dockerfile      0.1s
=> == transferring dockerfile: 258B                             0.0s
=> [webapp internal] load metadata for docker.io/library/nginx:alpine 2.3s
=> [webapp internal] load .dockerignore                        0.0s
=> == transferring context: 2B                                   0.0s
=> [webapp internal] load build context                        0.0s
=> == transferring context: 2.95kB                               0.0s
=> [webapp 1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a06 1.9s
=> == resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8 0.0s
=> == sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f 10.33kB / 10.33kB 0.0s
=> == sha256:62223d644fa234c3a1cc785ee14242ec47a77364226 2.50kB / 2.50kB 0.0s
=> == sha256:6769dc3a703c719c1d2756bda113659be28a016cf 10.79kB / 10.79kB 0.0s
=> == sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b532 1.79MB / 1.79MB 0.3s
=> == sha256:b464cfdcf2a6319875aeb27359ec549790ce14d8214fcb16 629B / 629B 0.5s
=> == sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa294 955B / 955B 0.7s
=> == extracting sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc1 0.1s
=> == sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73 402B / 402B 0.7s
=> == extracting sha256:b464cfdcf2a6319875aeb27359ec549790ce14d8214fcb16e 0.0s
=> == sha256:197eb75867ef4fceed4724f17b0972ab0489436860a 1.21kB / 1.21kB 0.7s
=> == extracting sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947 0.0s
```

```
=> => sha256:6769dc3a703c719c1d2756bda113659be28ae16cf 10.79kB / 10.79kB 0.0s
=> => sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b532 1.79MB / 1.79MB 0.3s
=> => sha256:b464cfd72a6319875aeb27359ec549790ce14d8214fcb16 629B / 629B 0.5s
=> => sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa294 955B / 955B 0.7s
=> => extracting sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc1 0.1s
=> => sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73 402B / 402B 0.7s
=> => extracting sha256:b464cfd72a6319875aeb27359ec549790ce14d8214fcb16e 0.0s
=> => sha256:197eb75867ef4fceed4724f17b0972ab0489436860a 1.21kB / 1.21kB 0.7s
=> => extracting sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947 0.0s
=> => sha256:34a64644b756511a2e217f0508e11d1a572085d66cd 1.40kB / 1.40kB 0.9s
=> => sha256:39c2ddf6010082a4a646e7ca44e95aca9bf3eae 15.52MB / 15.52MB 1.1s
=> => extracting sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73c 0.0s
=> => extracting sha256:197eb75867ef4fceed4724f17b0972ab0489436860a594a9 0.0s
=> => extracting sha256:34a64644b756511a2e217f0508e11d1a572085d66cd6dc9a 0.0s
=> => extracting sha256:39c2ddf6010082a4a646e7ca44e95aca9bf3eae 0.6s
=> [webapp 2/2] COPY index.html /usr/share/nginx/html/index.html 0.1s
=> [webapp] exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:49420a5bc44cbf80b5117dfe8f2ac14e918f5c206fb25 0.0s
=> => naming to docker.io/library/task-webapp 0.0s
=> [webapp] resolving provenance for metadata file 0.0s
[+] Running 3/3
✓ webapp Built 0.0s
✓ Network task_profile-network Created 0.1s
✓ Container my-profile-webapp Started 0.5s
ubuntu@ip-172-31-0-105:~/task$ ls
```

3. Access the website:

⑩ Open your browser and go to: <http://<serverip>:8080>



## Stopping the Application

`docker-compose down`

## Customization

You can customize the personal information by editing the `index.html` file before building the container. For example:



- ⑩ Change the name, role, and email
- ⑩ Update the About Me section
- ⑩ Modify the skills list
- ⑩ Add or remove sections
- ⑩ Change the styling

After making changes, rebuild and restart the containers:

```
docker-compose down
docker-compose up -d --build
```

## Advanced Options

### Persistent Data

If you want to store data persistently, you can add volumes to the docker-compose.yml file:

```
services:
  webapp:
    # ... existing configuration ...
    volumes:
      - ./data:/usr/share/nginx/html/data
```

### Environment Variables

You can use environment variables for dynamic configuration:

```
services:
  webapp:
    # ... existing configuration ...
    environment:
      - NAME=Your Name
      - ROLE=Your Role
```

Then modify the index.html to use these variables with a server-side language or inject them at build time.