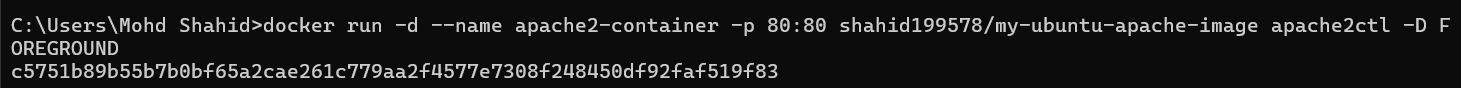
**Task 1: Launch the Apache2 Container**

1. **Run the Apache2 Container:**

**docker run -d --name apache2-container -p 80:80 shahid199578/my-ubuntu-apache-image apache2ctl -D FOREGROUND**

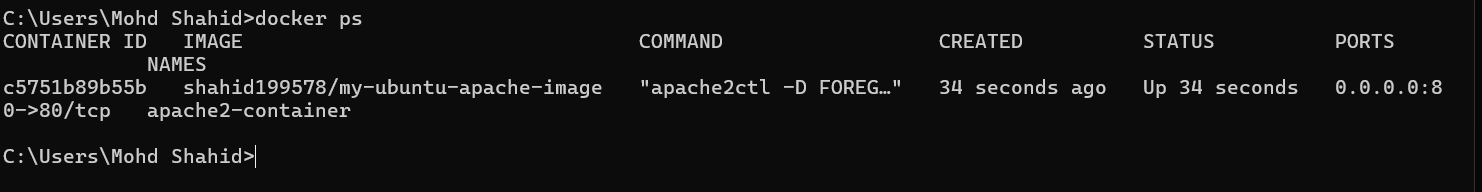
****

* + -d: Runs the container in detached mode.
  + --name apache2-container: Names the container apache2-container.
  + -p 80:80: Maps the container's port 80 to the host's port 80.
  + apache2-image: The name of the Apache2 Docker

1. **Verify the Container is Running:**

You can check the running containers with the command:

**docker ps**

****

**Task 2: Create a Docker Volume on /var/www/html**

Docker volumes are used to persist data between container restarts. To create a Docker volume and mount it to the /var/www/html directory inside the Apache2 container:

1. **Create the Volume:**

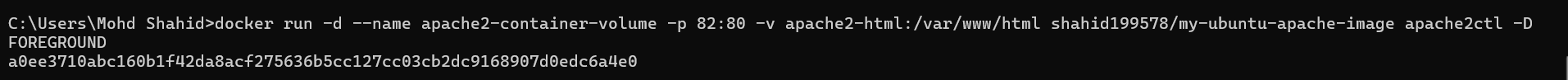
**docker volume create apache2-html**

****

1. **Launch the Container with the Volume:**

Now, launch or recreate the Apache2 container with the volume mounted to /var/www/html:

**docker run -d --name apache2-container-volume -p 82:80 -v apache2-html:/var/www/html shahid199578/my-ubuntu-apache-image apache2ctl -D FOREGROUND**

****

-v apache2-html:/var/www/html: Mounts the volume apache2-html to the /var/www/html directory inside the container.

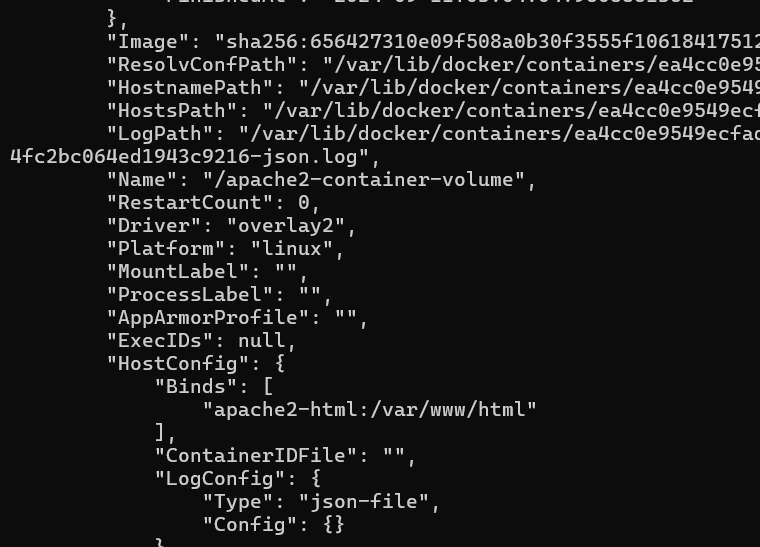
1. **Verify Volume is Mounted:**

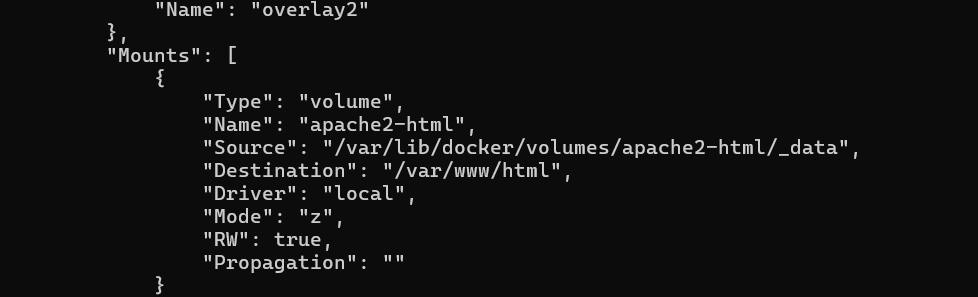
inspect the container to verify the volume is properly mounted:

**docker inspect apache2-container**

****

**docker inspect apache2-container-volume**

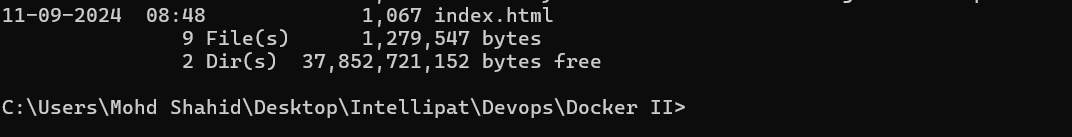
****

****

Check the "Mounts" section to ensure that apache2-html is mounted to /var/www/html.

This will ensure that any data in /var/www/html persists even if the container is restarted.

**Prepare Your HTML Files:**

* Place the HTML files you want to serve in a directory on your host system. For example, let's assume you place them in C:\Users\Mohd Shahid\Desktop\Intellipat\Devops\Docker II\.
* 

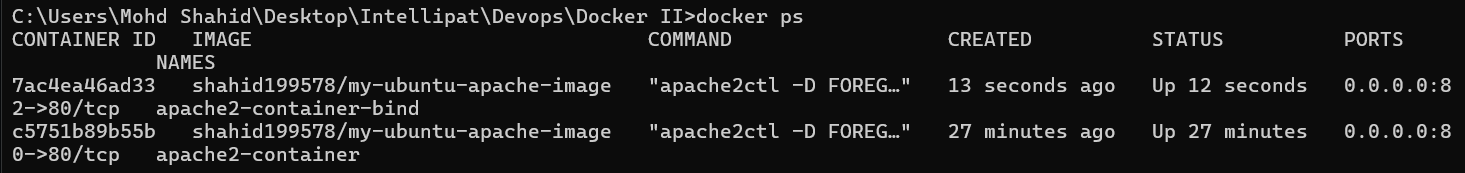
**Stop and Remove the Existing Container:** If you already have a container running with the volume, you might need to stop and remove it:

**docker stop apache2-container-volume**

**docker rm apache2-container-volume**

**Run the Apache2 Container with Bind Mount:** Use the following command to start the container with a bind mount:

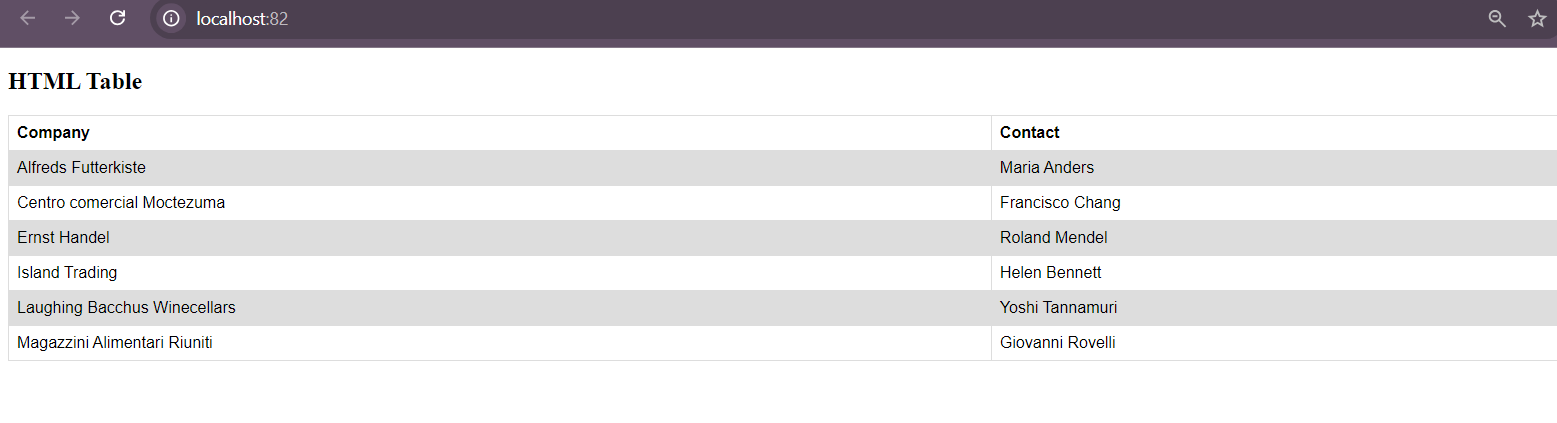
**docker run -d --name apache2-container-bind -p 82:80 -v "C:\Users\Mohd Shahid\Desktop\Intellipat\Devops\Docker II\:/var/www/html" shahid199578/my-ubuntu-apache-image apache2ctl -D FOREGROUND**

****

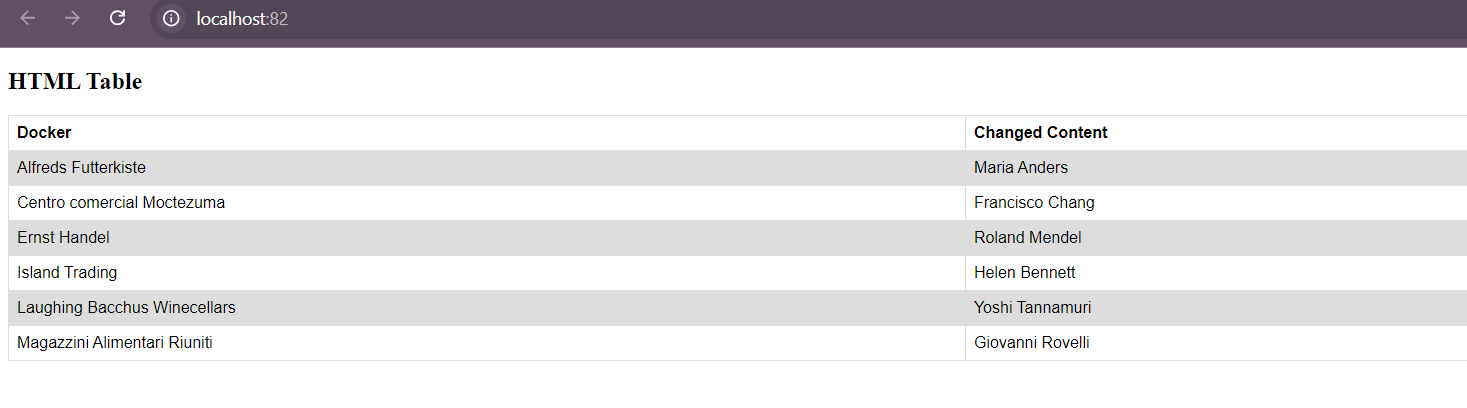
* This command mounts the host directory (C:\Users\Mohd Shahid\Desktop\Intellipat\Devops\Docker II\) to /var/www/html inside the container.

**Verify the Setup:**

* Access your Apache2 server at <http://localhost:82> .



* Any changes made to the HTML files in /path/to/your/html on your host will be reflected immediately in the container.



Deploy 5 custom containers using Docker Compose, each serving a unique default page. The containers should use an Apache web server and be accessible on different ports.

**Requirements**

1. **Custom Containers**: Create 5 containers from a custom Apache image.
2. **Unique Default Pages**: Each container should serve a different HTML page.
3. **Port Mapping**: Map the containers to ports 81 through 85 on the host machine.

**Docker Compose Configuration**

Create a docker-compose.yml file with the following content:

version: '3'

services:

  web1:

    image: shahid199578/my-ubuntu-apache-image

    container\_name: web1

    ports:

      - "81:80"

    volumes:

      - ./page1.html:/var/www/html/index.html

    command: apache2ctl -D FOREGROUND

  web2:

    image: shahid199578/my-ubuntu-apache-image

    container\_name: web2

    ports:

      - "82:80"

    volumes:

      - ./page2.html:/var/www/html/index.html

    command: apache2ctl -D FOREGROUND

  web3:

    image: shahid199578/my-ubuntu-apache-image

    container\_name: web3

    ports:

      - "83:80"

    volumes:

      - ./page3.html:/var/www/html/index.html

    command: apache2ctl -D FOREGROUND

  web4:

    image: shahid199578/my-ubuntu-apache-image

    container\_name: web4

    ports:

      - "84:80"

    volumes:

      - ./page4.html:/var/www/html/index.html

    command: apache2ctl -D FOREGROUND

  web5:

    image: shahid199578/my-ubuntu-apache-image

    container\_name: web5

    ports:

      - "85:80"

    volumes:

      - ./page5.html:/var/www/html/index.html

    command: apache2ctl -D FOREGROUND

**Custom Default Pages**

Create the following HTML files in the same directory as your docker-compose.yml file:

* **page1.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Page 1</title>

    <style>

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }

        h1 { color: #3498db; }

    </style>

</head>

<body>

    <h1>Welcome to Page 1</h1>

    <p>This is the default page for container 1.</p>

</body>

</html>

* **page2.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Page 2</title>

    <style>

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }

        h1 { color: #2ecc71; }

    </style>

</head>

<body>

    <h1>Welcome to Page 2</h1>

    <p>This is the default page for container 2.</p>

</body>

</html>

* **page3.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Page 3</title>

    <style>

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }

        h1 { color: #e74c3c; }

    </style>

</head>

<body>

    <h1>Welcome to Page 3</h1>

    <p>This is the default page for container 3.</p>

</body>

</html>

* **page4.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Page 4</title>

    <style>

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }

        h1 { color: #f39c12; }

    </style>

</head>

<body>

    <h1>Welcome to Page 4</h1>

    <p>This is the default page for container 4.</p>

</body>

</html>

* **page5.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Page 5</title>

    <style>

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; }

        h1 { color: #9b59b6; }

    </style>

</head>

<body>

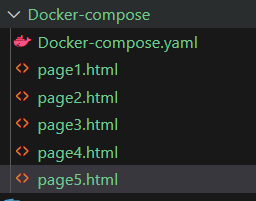
    <h1>Welcome to Page 5</h1>

    <p>This is the default page for container 5.</p>

</body>

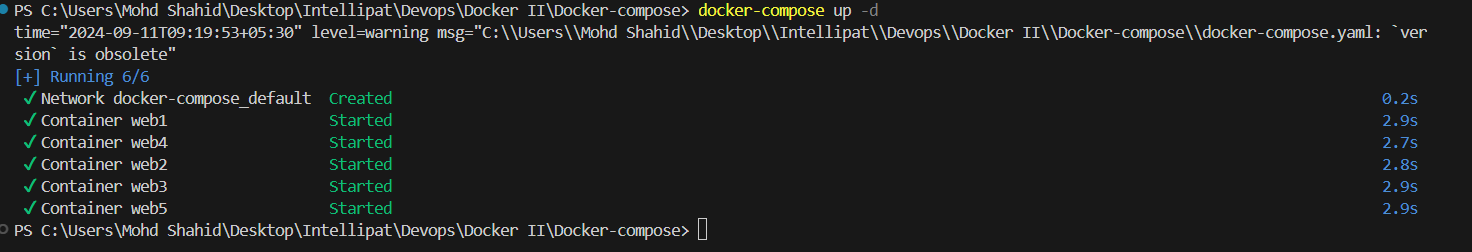
</html>

1. Save the docker-compose.yml file and the HTML files in the same directory.



1. Run docker-compose up from the directory where the files are located to start the containers.

**docker-compose up -d**



1. Verify the deployment by navigating to http://localhost:81, http://localhost:82, http://localhost:83, http://localhost:84, and http://localhost:85 in your web browser. Each URL should display a different default page.



