Nilay Gupta <nilayguptaforwork@gmail.com>

assingment

Deepesh Mishra <mishradeepesh72@gmail.com> 17 November 2023 at 14:46
To: nilayguptaforwork@gmail.com

last step, which involves using Docker Compose to define a multi-container application. Docker Compose is a tool for defining and running multi-container Docker applications. In this example, we'll create a simple web application with Nginx and a backend service.

1. **Create a Docker Compose File:**
Create a file named `docker-compose.yml` in your project directory. This file will define the services, networks, and volumes for your application. Here's a basic example:

```yaml
version: '3'

services:
  web:
    image: my-web-app
    ports:
      - "8081:80"

  backend:
    image: some-backend-image
    ports:
      - "8082:3000"
```

Replace `my-web-app` with the name of the image you built in a previous step and `some-backend-image` with the image for your backend service.

2. **Run Docker Compose:**
In your terminal, navigate to the directory containing the `docker-compose.yml` file and run:

```bash
docker-compose up
```

This command will start the services defined in the `docker-compose.yml` file. You should see the output for each service.

3. **Access Your Application:**
Open your web browser and go to http://localhost:8081. This should display the web app served by Nginx. If you have a backend service, you can access it at http://localhost:8082.

4. **Stop and Remove Containers:**
When you're done testing, you can stop and remove the containers using:

```bash
docker-compose down
```

This command will stop and remove the containers, networks, and volumes defined in your `docker-compose.yml` file.

Docker Compose simplifies the management of multi-container applications, allowing you to define complex setups with ease. Customize the `docker-compose.yml` file according to your application's needs, adding more services, networks, or volumes as required.

[Quoted text hidden]