Getting Started with Docker for Linux_final_project

Welcome to the Linux_final_project! This guide includes comprehensive instructions on installing Docker, pulling and using the Docker image kangjiecan/repository-name:NSCC_Linux_Final_assignment, and utilizing Docker commands to streamline your development process.

Prerequisites: Before starting, you'll need Docker installed on your system.

Using the Docker Image kangjiecan/repository-name:NSCC_Linux_Final_assignment

This Docker image is pre-configured with an Apache HTTP Server to serve the content of the Linux_final_project.

Pulling the Docker Image

- At first, ensure docker service is running on your system.
  ```
  sudo service docker status
  ```
- if not, start the service
  ```
  sudo service docker start
  ```
- Second, pull the Docker image from Docker Hub:
  ```
  docker pull kangjiecan/repository-name:NSCC_Linux_Final_assignment
  ```

Running and stop the Docker Image

- To run the image and start the Apache server:
  ```
  docker run -d -p 8080:80 kangjiecan/repository-name:NSCC_Linux_Final_assignment
  ```
- -d runs the container in detached mode (in the background).
- -p 8080:80 maps port 80 from inside the container (Apache) to port 8080 on your host machine.
- Access the web server by navigating to http://localhost:8080 in your web browser.
- For stopping the Docker container using its container ID:
  ```
  docker stop [CONTAINER_ID]
  ```
- Find the container ID with
  ```
  docker ps
  ```

Installing Docker Compose

- For most systems, you can install docker-compose with this command:
  ```
  sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  ```
- Then, apply executable permissions to the binary:
  ```
  sudo chmod +x /usr/local/bin/docker-compose
  ```
- Check if it's installed correctly by running:
  ```
  docker-compose --version
  ```

Creating a docker-compose.yml File

- Create a directory for your project if you haven't already:
  ```
  mkdir linux_final_project
  cd linux_final_project
  ```
- Inside this directory, create a file named docker-compose.yml.
- Open the docker-compose.yml file with a text editor and define your service:
  ```
  version: '3.8'
  services:
    web:
      image: kangjiecan/repository-name:NSCC_Linux_Final_assignment
      ports
        - "8080:80"
      container_name: linux_final_web
  ```

In this docker-compose.yml file:

- version: Specifies the version of the Docker Compose file format.
- services: Under this key, you define your application's services (containers).
- web: The name of the first service. You can choose any name here.
- image: Specifies the Docker image to use for this service.
- ports: Maps port 80 from inside the Docker container to port 8080 on the host machine.
- container_name: An optional field to specify a custom container name.

Docker Compose base

To start your services, run:

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

This will start the Apache server in a Docker container as defined in your docker-compose.yml file.

To stop and remove all the services defined in the docker-compose.yml file, run:

```
docker-compose down
```

To view the running services, use:

```
docker-compose ps
```

To see the logs for a service, run:

```
docker-compose logs web
```