Case Study: Implementing Port Forwarding with Docker.

Introduction:

Port forwarding in Docker plays a crucial role in enabling communication between containers and the host system or between containers themselves. In this case study, we'll explore how to set up port forwarding for Apache HTTP Server (httpd) and Ngnix containers.

Example 1: Running Apache HTTP Server Inside a Docker Container

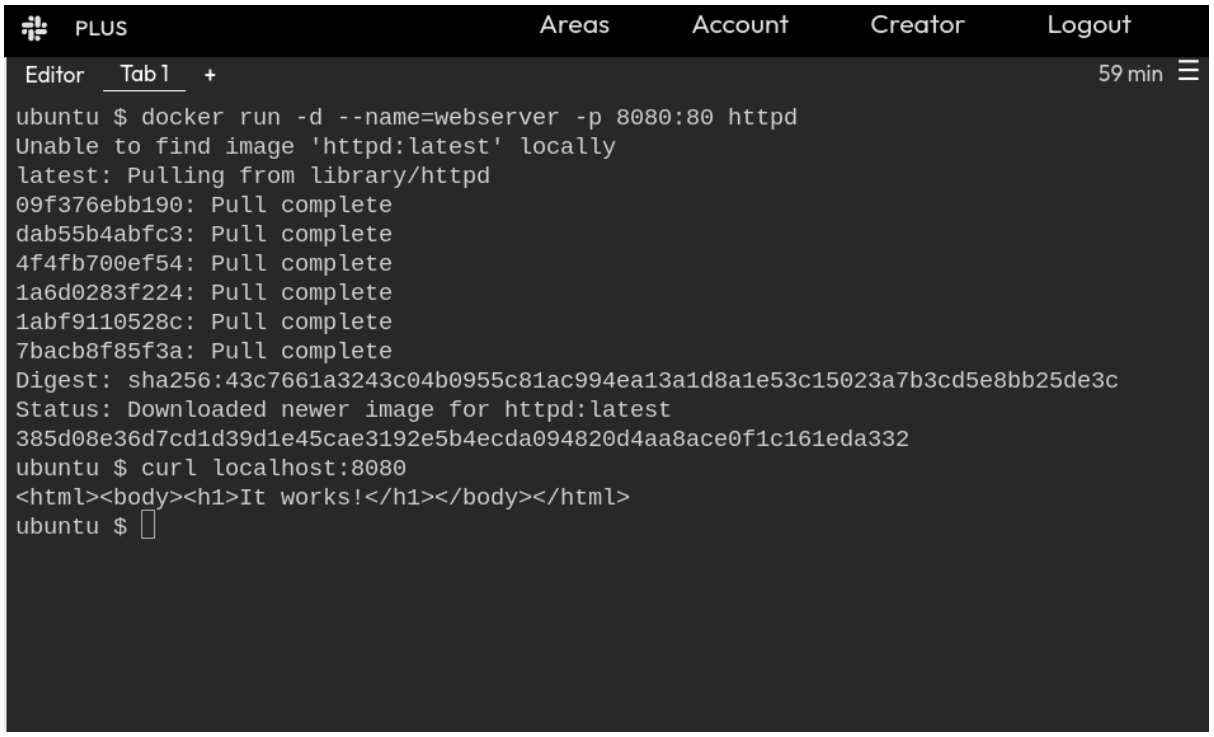
Scenario:

A web developer wants to run a website using Apache HTTP Server inside a Docker container and access it from the host system's web browser.

Implementation:

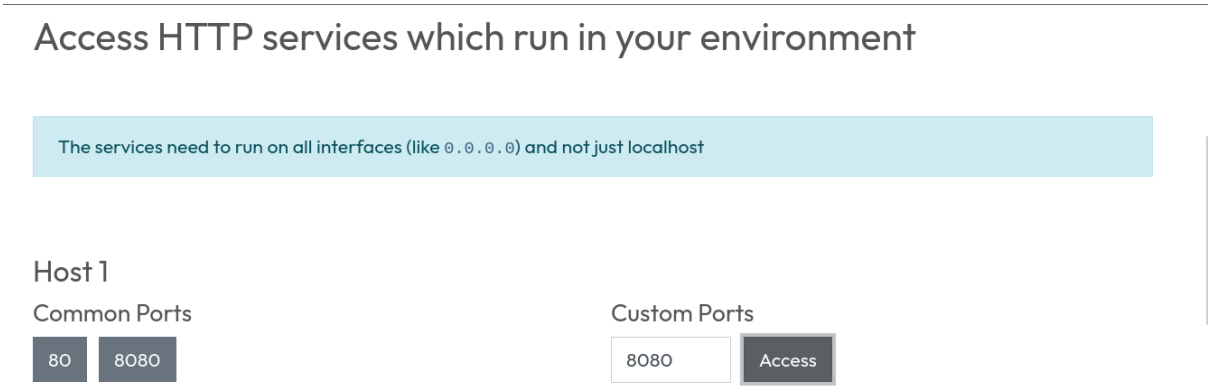
1. Run the Docker Container with Port Forwarding:

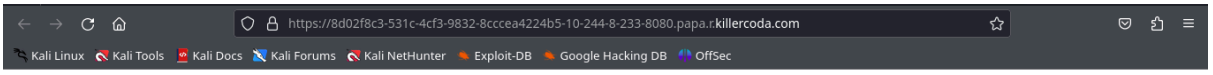
```
docker run -dit --name=webserver -p 8080:80 httpd
```



2. Access the Apache Server:

Open a web browser on the host system and navigate to http://localhost:8080 to access the website served by the Apache HTTP Server running inside the Docker container.





It works!

Example 2: Running Ngnix Inside a Docker Container

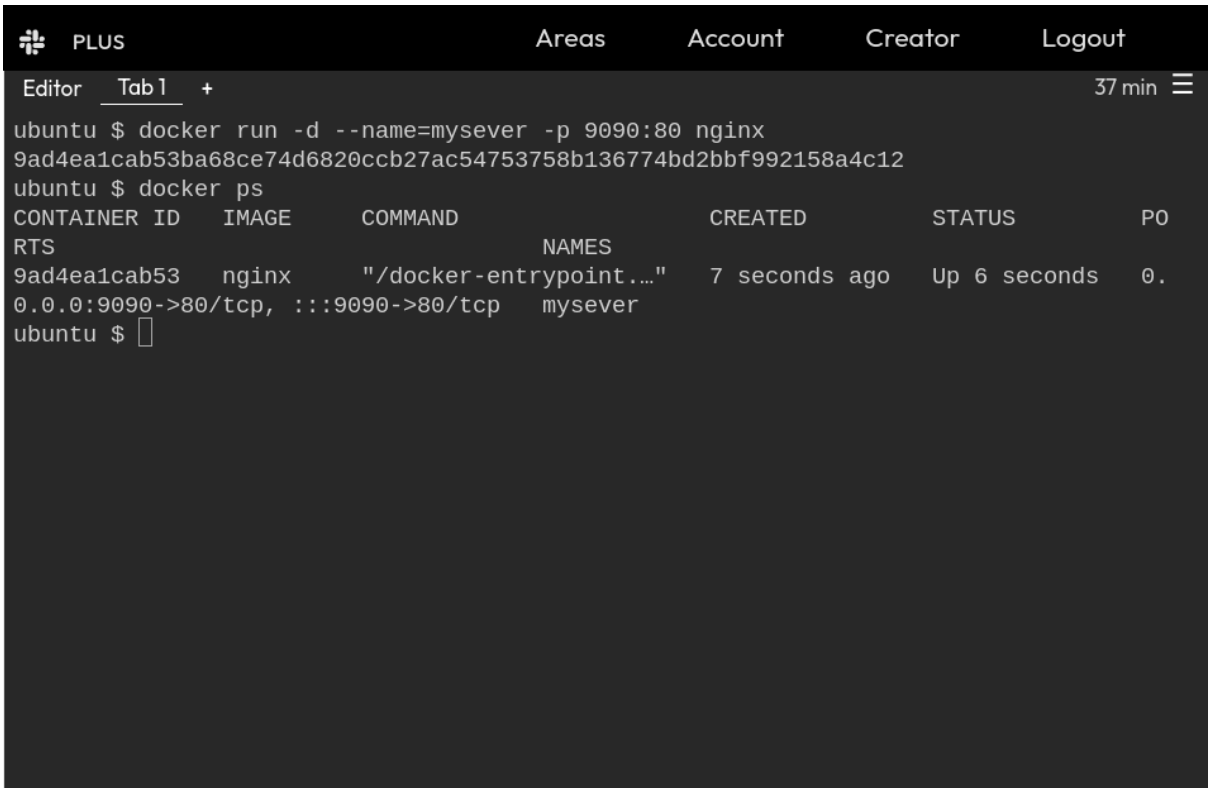
Scenario:

An organization wants to deploy a website using NGINX inside a Docker container and ensure it is accessible from the host system's web browser.

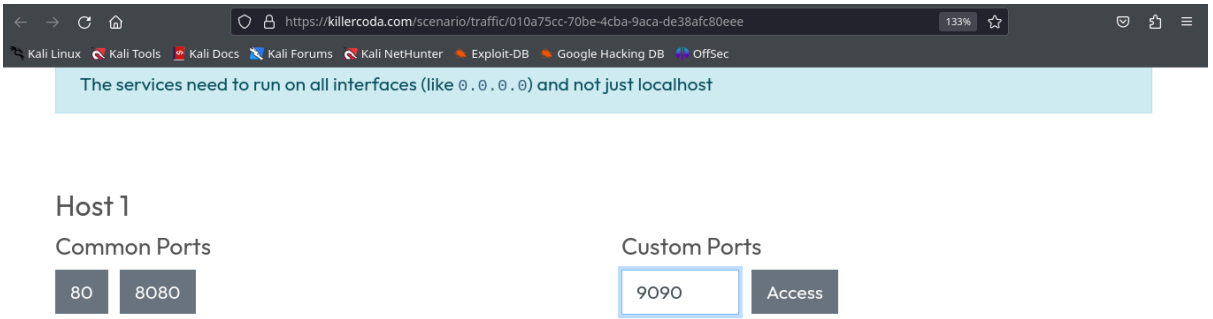
Implementation:

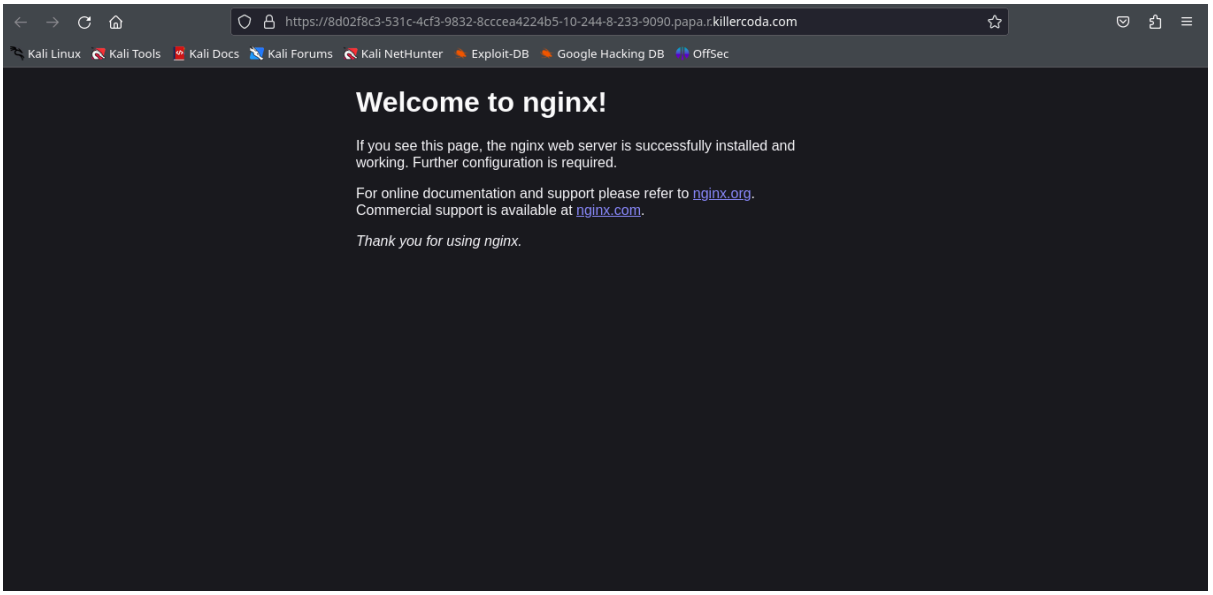
1.Run the Docker Container with Port Forwarding::

```
docker run -d --name=ng -p 9090:80 nginx
```



2. **Access the NGINX Server:**
Open a web browser on the host system and navigate to http://localhost:9090 to access the website served by NGINX running inside the Docker container.





Conclusion:

Port forwarding in Docker simplifies the process of accessing services running inside containers from the host system or other containers. By following the examples provided in this case study, users can effectively set up port forwarding for Apache HTTP Server and Ngnix containers, enabling seamless communication between containers and the host system.