**Lesson 06 Demo 01**

**Building a Secure Docker Container**

**Objective:** To secure Docker containers by granting access to a non-root user within the container to mitigate the risks associated with running processes as the root user

**Tools required:** Docker

**Prerequisites:** None

Steps to be followed:

1. Create a Dockerfile with a non-root user
2. Build the Docker image
3. Create and run a Docker container

**Step 1: Create a Dockerfile with a non-root user**

1. Create a directory for the Dockerfile using the following command: **mkdir test1  
     
   A close-up of a number

   Description automatically generated**
2. Navigate inside the created directory using the following command:  
   **cd test1**  
     
   A close-up of a number

   Description automatically generated
3. Create a Dockerfile using the following command: **vi Dockerfile  
     
   **
4. Edit the Dockerfile to create a non-root user named **myuser** using the following script:

**# Use an official Ubuntu base image**

**FROM ubuntu:latest**

**# Create a new user called 'myuser'**

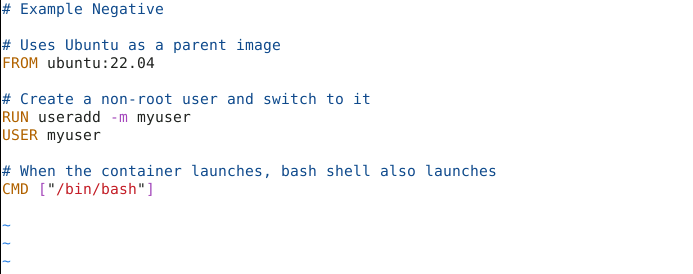
**RUN useradd -m myuser**

**# Set the user to 'myuser' for subsequent instructions**

**USER myuser**

**# The default command to run on the container start**

**CMD ["bash"]**

****

This Dockerfile sets up a basic Ubuntu container, creates a new user named **myuser**, and switches to that user before performing any other operations.

**Step 2: Build the Docker image**

1. Build the Docker image using the following command:

**docker build -t test1 .**

A close-up of a computer screen

Description automatically generated

**Step 3: Create and run a Docker container**

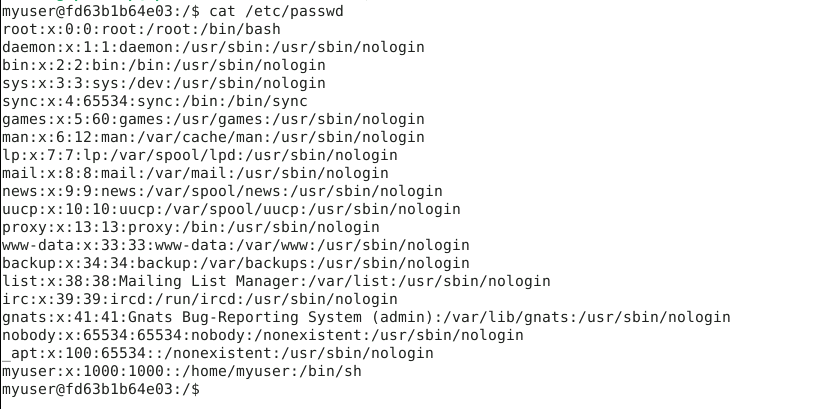
1. From the newly built Docker image, create and run a Docker container using the following command:

**docker run -it --name testcon1 test1**



1. Verify that the container is running as **myuser** using the following command:

**cat /etc/passwd**

****

This command displays the entry for **myuser** and confirms that the user exists within the container.

By following these steps, you have successfully secured Docker containers by granting access to a non-root user within the container, mitigating the risks associated with running processes as the root user.