**\*\*Lab 2: Basic Bash Script with Docker\*\***

**\*\*Objective\***\*:

Grasp fundamental interactions between Bash and Docker.

**\*\*Tasks\*\*:**

1. Write a Bash script to pull an Nginx Docker image.

2. Run the Docker container via the script.

| **#!/bin/bash  # Task 1: Pull Nginx Docker image echo "Task 1: Pulling Nginx Docker image" docker pull nginx  # Check if image pull was successful if [ $? -eq 0 ]; then  echo "Image pull successful: Nginx Docker image" else  echo "Image pull failed: Nginx Docker image"  exit 1 fi  # Task 2: Run Docker container echo "Task 2: Running Docker container" docker run -d -p 8080:80 --name nginx-container nginx  # Check if container is running if docker ps | grep -q nginx-container; then  echo "Nginx container is running and accessible at http://localhost:8080" else  echo "Failed to start Nginx container."  exit 1 fi  # End of script echo "Lab 2 tasks completed."** |
| --- |

**Instructions:**

* Save the script to a file named lab2\_script.sh.
* Make the script executable: chmod +x lab2\_script.sh.
* Run the script: ./lab2\_script.sh.

**Documentation:**

* The script begins with a shebang (#!/bin/bash) to indicate that it's a Bash script.
* Comments within the script explain each step of the tasks and their objectives.
* In Task 1, the script pulls the Nginx Docker image using docker pull nginx.
* The exit code ($?) of the image pull operation is checked to determine success.
* In Task 2, the script runs an Nginx container in detached mode, mapping port 8080 on the host to port 80 in the container.
* The docker ps command checks if the container is running, and grep is used to search for the container name.
* Appropriate messages are displayed based on the success or failure of the tasks.
* The script concludes by indicating the completion of Lab 2 tasks.