**\*Lab 3: Conditional Bash Scripts with Docker\***

**Tasks:**

**1. Write a script that checks if a Docker container is running.**

**2. If not running, the script should start it**

Certainly, here's a Bash script that checks if a Docker container is running and starts it if it's not:

| #!/bin/bash  # Name of the Docker container CONTAINER\_NAME="my-container"  # Check if the container is already running if docker ps -f name=$CONTAINER\_NAME | grep -q $CONTAINER\_NAME; then  echo "Container is already running." else  echo "Container is not running. Starting..."  docker start $CONTAINER\_NAME  if [ $? -eq 0 ]; then  echo "Container started successfully."  else  echo "Failed to start the container."  fi fi |
| --- |

**Here's how the script works:**

* It defines the name of the Docker container you want to check and potentially start.
* It uses docker ps -f name=$CONTAINER\_NAME to check if the container is running. The -f flag is used to filter containers by name, and grep is used to search for the container name in the output.
* If the container is running, it prints a message indicating that the container is already running.
* If the container is not running, it prints a message indicating that it's starting the container, uses docker start to start the container, and then checks the exit code to determine if the start operation was successful.

**Remember to make the script executable using chmod +x script\_name.sh and replace "my-container" with your actual container's name before running the script.**