**Lab 6: Functions in Bash for Modular Docker Scripts**

Objective:*Use functions in Bash to modularize Docker operations.*

Tasks:

1. Create separate functions for Docker pull, run, and stop operations.

2. Use the functions in a main Bash script workflow.

Documentation:

- The utility of functions in Bash.

- Structuring Docker operations in modular scripts

Prerequisites:

1- An AWS account with administrative access.

2- Docker Deep Dive Course

3- Bash Script Deep Dive Course

4- Complete Previous labs

Implementation Documentation:

**1. Introduction**

In this lab, we will explore the use of functions in Bash to modularize Docker operations. By breaking down Docker-related tasks into separate functions, we can create more maintainable and reusable scripts. Specifically, we will create functions for Docker pull, run, and stop operations.

**2. Script Implementation**

Here's the Bash script that accomplishes the tasks:

| #!/bin/bash  # Function to pull a Docker image pull\_image() {  local image\_name="$1"  echo "Pulling the Docker image: $image\_name"  docker pull "$image\_name" }  # Function to run a Docker container run\_container() {  local image\_name="$1"  echo "Running the Docker container based on: $image\_name"  docker run -it --rm -name "$image\_name" "$image\_name" }  # Function to stop a Docker container stop\_container() {  local container\_name="$1"  echo "Stopping the Docker container: $container\_name"  docker stop "$container\_name" }  # Main script workflow main() {  read -p "Enter the name of the Docker image to pull and run: " image\_name   # Pull the Docker image  pull\_image "$image\_name"   # Run the Docker container  run\_container "$image\_name"   # Ask if the user wants to stop the container  read -p "Do you want to stop the container? (yes/no): " stop\_choice  if [ "$stop\_choice" == "yes" ]; then  # Stop the Docker container  stop\_container "$image\_name"  else  echo "Container is still running. You can manually stop it later."  fi }  # Call the main function to start the script main |
| --- |

**Explanation of the Script**

We have defined three separate functions for Docker operations: pull\_image, run\_container, and stop\_container. These functions take the relevant parameter (image name or container name) and perform the respective operation.

In the main function, we first accept the name of the Docker image from the user.

We then call the pull\_image function to pull the specified Docker image.

Next, we call the run\_container function to run a container based on the pulled image.

After running the container, we ask the user if they want to stop it. If the user chooses to stop it, we call the stop\_container function.

**3. Running the Script**

To run the script, follow these steps:

Save the script in a .sh file, e.g., **docker\_modular\_script.sh.**

Make the script executable by running **chmod +x docker\_modular\_script.sh**.

Execute the script by running **./docker\_modular\_script.sh.**

Follow the prompts to specify the Docker image, run the container, and optionally stop it.

**4. Conclusion**

In this lab, we have demonstrated how to use functions in Bash to modularize Docker operations. By organizing the code into functions, we improve code reusability and maintainability. This approach makes it easier to manage Docker-related tasks and create more structured and readable scripts.