Completion time: one day

Prerequisites:

- Github account with a new repository created. Ensure that your GitHub repository is publicly accessible.
- Docker Desktop installed.

Submission Guidelines:

- The test consists of three independent tasks that can be completed in any order.
- **The code should be uploaded in the public Github repository created as a prerequisite.**
- Useful resources mentioned in the exercises can be found at the following link: https://github.com/anddragn/Useful-Resources
- Every exercise should be uploaded in a different folder of the same repository, in order to have a clear separation between them.
- Every folder should contain a separate README. Each one of these documents should include any relevant information regarding the process of solving the task, such as output snippets, screenshots, commands used that are not included in the scripts provided as a solution, any relevant observations or challenges faced during the validation process etc.
- **We do not expect you to write all the code without consulting the documentation or any other resources that you consider relevant.**
- **Your submissions should prove that you can operate with basic DevOps concepts, which are going to play a significant part during the evaluation process**

I. /etc/passwd file manipulation

Create a Linux container and use its passwd file. In order to do this, the following steps are provided:

1. After you have installed Docker Desktop, please download an Ubuntu image of your choice (hint: find the image in Docker Hub and use ‘docker pull’ command). You can check that the image is present locally using Docker Desktop or ‘docker images’ command in terminal.
2. In order to start a container based on the downloaded image, use the following command:

   ```
   docker run -it --name my_linux_container <linux-image:tag>
   ```

   Where <linux-image:tag> is the downloaded image.

After completing the steps above, you can proceed with the next steps:

Inside the container, create a new user **john** first, then copy the **create_large_file.sh** file to the Linux container and run the script there. The file mentioned can be found at the link given for Useful Resources, in the folder **1-linux**.

Write a bash script that reads the passwd file, checks if the file name is “passwd” and does the following actions:

1. Print the home directory
2. List all usernames from the passwd file
3. Count the number of users
4. Find the home directory of a specific user (prompt to enter the username value)
5. List users with specific UID range (e.g. 1000-1010)
6. Find users with standard shells like /bin/bash or /bin/sh
7. Replace the “/” character with “\” character in the entire /etc/passwd file and redirect the content to a new file
8. Print the private IP
9. Print the public IP
10. Switch to john user
11. Print the home directory

Bonus / Nice to have:

Error handling - Implement error handling to check if the script is executed with appropriate privileges and exit gracefully with an informative message if not.

Use a function for each request.

II. Choose one of the applications in the folder **2-simpleapp**, that can be found at the link provided for Useful Resources**.** Containerize it with Docker, and automate the build and push processes.
In order to complete this task, the following recommendations are provided:

- Choose an Application:
    - There are two applications in the folder mentioned, one in Java (**Main.java**) and one in Python (**app.py**). Please choose one of these and use it for the following steps of this task.

- Create a Dockerfile:
    - Write a Dockerfile to containerize the application chosen at the previous step and include necessary instructions to set up the environment and dependencies.

- Local Testing:
    - Build the Docker image locally, then test the Dockerized application to ensure it functions correctly within the container. Include all the steps in the README file.

- Set Up a Docker Registry (Ex: https://hub.docker.com/):
    - Choose a Docker registry service and create an account and a repository there.

- Automation:
    - Automate the following steps (using GitHub Actions):
      - Trigger the build whenever changes are pushed to the repository on branch main/master
      - Build the Docker image using the Dockerfile
      - Tag the Docker image with a version or commit hash and the latest tag
      - Push the Docker image to the Docker registry

Bonus / Nice to have:

Configure environment variables for sensitive information, such as credentials.

III. Query a MySQL database.

This exercise is going to use the **company.sql** dump, that you can find at the link provided for Useful Resources, in the folder **3-database**.

In order to solve this task, the following steps are required:

- Create a MySql container. Create a database named “company”.
- Import the company.sql file in your company database.
- Create a user and assign all the permissions required for the database “company”
- Find the average salary for each department.
- Write in the README file all the steps/commands

Bonus / Nice to have: Mount a volume in the database container in order to preserve the data from the database.