Completion time: two days

Prerequisites:

- GitHub account with a new repository created. Ensure that your GitHub repository is **publicly accessible**.
- Docker Desktop installed.
- Create an account on Docker Hub and a repository

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/Internship-Resources-2025
- **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.**
- **Make sure to double-check commands, irrelevant ones will lead to points being deducted.**
- **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. Linux essentials

The following tasks should be performed from within a Docker container.

Download and install Docker Desktop, then from terminal run an Ubuntu

container: docker run -it ubuntu

• Lookup the Public IP of cloudflare.com

```
root@f189dff529dd:~# nslookup cloudflare.com
Server:         192.168.65.7
Address:        192.168.65.7#53

Non-authoritative answer:
Name:   cloudflare.com
Address: 104.16.133.229
Name:   cloudflare.com
Address: 104.16.132.229
Name:   cloudflare.com
Address: 2606:4700::6810:84e5
Name:   cloudflare.com
Address: 2606:4700::6810:85e5
```

• Map IP address 8.8.8.8 to hostname google-dns

• Check if the DNS Port is Open for google-dns

• Modify the System to Use Google's

```
root@f189dff529dd:~# telnet 8.8.8.8 53
Trying 8.8.8.8...
Connected to 8.8.8.8.
Escape character is '^]'.
Connection closed by foreign host.
root@f189dff529dd:~# nano /etc/resolv.conf
root@f189dff529dd:~# nslookup cloudflare.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   cloudflare.com
Address: 104.16.133.229
Name:   cloudflare.com
Address: 104.16.132.229
Name:   cloudflare.com
Address: 2606:4700::6810:84e5
Name:   cloudflare.com
Address: 2606:4700::6810:85e5
```

Public DNS

- Change the nameserver to 8.8.8.8 instead of the default local configuration.
- Perform another public IP lookup for cloudflare.com and compare the results.

• Install and verify that Nginx service is running

• Find the Listening Port for Nginx

```
root@f189dff529dd:~# ss -tuln | grep nginx
root@f189dff529dd:~# ss -tuln
Netid    State     Recv-Q     Send-Q         Local Address:Port         Peer Address:Port    Process
tcp      LISTEN    0          511                  0.0.0.0:80              0.0.0.0:*
tcp      LISTEN    0          511                     [::]:80                 [::]:*
root@f189dff529dd:~# ^C
root@f189dff529dd:~# ^C
root@f189dff529dd:~# exit
exit
```

Bonus / Nice to Have:

• Change the Nginx Listening port to 8080

• Modify the default HTML page title from: "Welcome to nginx!" → "I have completed the Linux part of the DevOps internship project"

II. Choose one of the applications in the folder. Containerize it with Docker and automate the build and push processes.

To complete this task, the following recommendations are provided:

• Choose an application:

o There are two applications in the folder **2-app/**, one in Python(**calculator.py**), that has the dependencies located in **requirements.txt** file, and one in Node.js(**note.js)**, that has the dependencies located in **package.json** file. Please choose one of these and use it for the following steps of this task.

• Create a Dockerfile:

o Write a Dockerfile to containerize the application chosen at the previous step.

o Include necessary instructions to set up the environment and dependencies. o Ensure the application runs on **port 8080**.

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

```
PS E:\download\tremend> docker build -t python-calculator .
[+] Building 5.0s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 302B
 => [internal] load metadata for docker.io/library/python:3.9
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/python:3.9@sha256:bc2e05bca883473050fc3b7c134c28ab822be73126ba1ce29517d9e8b7f3703b
 => => resolve docker.io/library/python:3.9@sha256:bc2e05bca883473050fc3b7c134c28ab822be73126ba1ce29517d9e8b7f3703b
 => [internal] load build context
 => => transferring context: 70B
 => CACHED [2/4] ADD calculator.py .
 => [3/4] ADD requirements.txt .
 => [4/4] RUN pip install --no-cache-dir -r /requirements.txt
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:5d900c6090ff21a1558c62435e8d7bb12f42eb9cc750e5003f8dbdf3e7ad202f
 => => exporting config sha256:83e0a98fcafb458a17f5987fc6910f07657267d687ed32c65e0e7f9303c0774c
 => => exporting manifest list sha256:8921eab443c52177efa8a1046b37636adaab61f82e0c42778fa4682d59da5162
 => => naming to docker.io/library/python-calculator:latest
 => => unpacking to docker.io/library/python-calculator:latest
PS E:\download\tremend> ls
```

- Set Up a Docker Registry:
    - o Create an account on Docker Hub (https://hub.docker.com/).
    - o Create a repository for your image.
- Automation:
    - o 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 commit hash
        - ♣ Push the Docker image to the Docker registr

```
PS E:\download\tremend> docker push dafne78/tremend_dafne_stanciu:latest
The push refers to repository [docker.io/dafne78/tremend_dafne_stanciu]
tag does not exist: dafne78/tremend_dafne_stanciu:latest
PS E:\download\tremend> docker images
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
python-calculator   latest    65a8a2c70ca2   6 minutes ago   1.47GB
ubuntu              latest    72297848456d   7 weeks ago     117MB
PS E:\download\tremend> docker push dafne78/tremend_dafne_stanciu:python-calculator
The push refers to repository [docker.io/dafne78/tremend_dafne_stanciu]
tag does not exist: dafne78/tremend_dafne_stanciu:python-calculator
PS E:\download\tremend> docker push
docker: 'docker push' requires 1 argument

Usage:  docker push [OPTIONS] NAME[:TAG]
Run 'docker push --help' for more information
PS E:\download\tremend> docker push python-calculator
The push refers to repository [docker.io/library/python-calculator]
19781d6367b0: Waiting
7cd785773db4: Waiting
6e02a90e58ae: Waiting
f6d72b00ae7c: Waiting
a63ca3b77c10: Waiting
f299e0671245: Waiting
091eb8249475: Waiting
PS E:\download\tremend> docker push dafne78/tremend_dafne_stanciu/image:python-calculator
The push refers to repository [docker.io/dafne78/tremend_dafne_stanciu/image]
tag does not exist: dafne78/tremend_dafne_stanciu/image:python-calculator
```

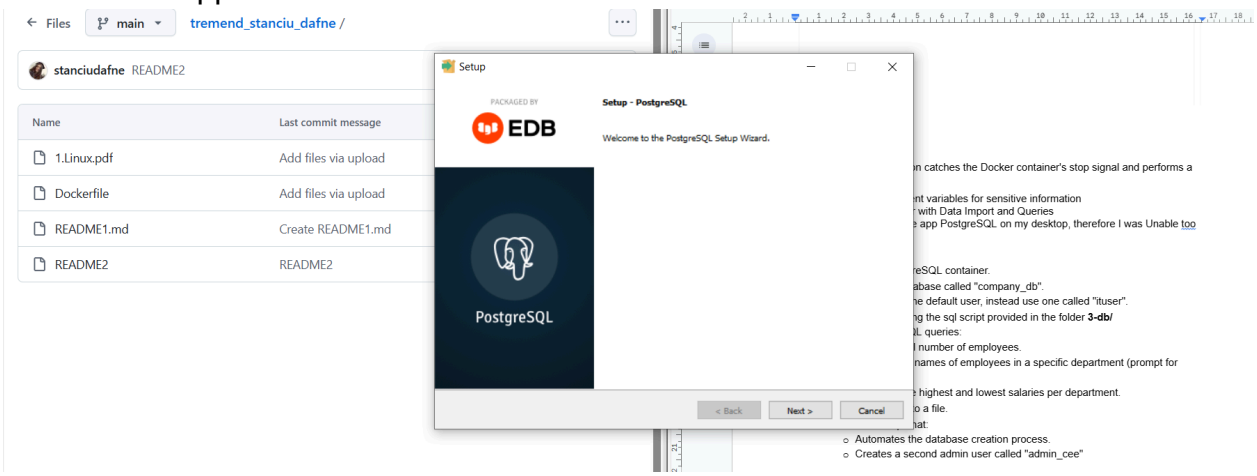
```
091eb8249475: Waiting
2a1f79c2396b: Waiting
255774e0027b: Waiting
6e02a90e58ae: Waiting
push access denied, repository does not exist or may require authorization: server message: insufficient_scope: authorization failed
PS E:\download\tremend> docker login
Authenticating with existing credentials... [Username: dafne78]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
PS E:\download\tremend> docker push python-calculator
Using default tag: latest
The push refers to repository [docker.io/library/python-calculator]
6e02a90e58ae: Waiting
a63ca3b77c10: Waiting
f6d72b00ae7c: Waiting
f299e0671245: Waiting
7cd785773db4: Waiting
255774e0027b: Waiting
19781d6367b0: Waiting
353e14e5cc47: Waiting
2a1f79c2396b: Waiting
775f8abf5c8c: Waiting
091eb8249475: Waiting
push access denied, repository does not exist or may require authorization: server message: insufficient_scope: authorization failed
PS E:\download\tremend> docker push python-calculator:sha256:65a8a2c70ca2b0408e76941fac3a9950158377f04937ce33c2277f00a5f31ad9
invalid reference format
```

Bonus / Nice to have:

- Ensure the application catches the Docker container's stop signal and performs a clean shutdown
- Configure environment variables for sensitive information

III. PostgreSQL Container with Data Import and Queries

I was unable to launch the app PostgreSQL on my desktop, therefore I was Unable too work on this app.



Every Time I was launching the app it was getting me to the initial steps of installing it, I was not able to succeed doing my tasks.

- Pull and run a PostgreSQL container.
  - Create a database called "company_db".

    - Do not use the default user, instead use one called "ituser".
- Create a dataset using the sql script provided in the folder **3-db/**
- Run the following SQL queries:
    - Find the total number of employees.
    - Retrieve the names of employees in a specific department (prompt for user input).
    - Calculate the highest and lowest salaries per department.
- Dump the dataset into a file.
- Write a Bash script that:
    - Automates the database creation process.
    - Creates a second admin user called "admin_cee"
    - Imports the dataset created at Step 4.
    - Executes the queries from Step 3 and outputs the results to a log file.

Bonus / Nice to Have:

- Mount a persistent volume to store PostgreSQL data.