TD DevOps 2023 4A-INT

Requirements

- Ubuntu 20.04 LTS Dekstop version virtual machine with Docker installed
- Docker hub account (free)
- A Repository in Docker hub
- A GitHub account (free)
- If you want to use windows to manipulate Git, so install Git Bash. If using Linux, install Git in command lines.
- Download the folder of source code here : https://drive.google.com/drive/folders/1eSg01rjL_HaJrQbemKIXGZYqz_P8dqig?usp=sharing

Part 1 – Build & Pipelines

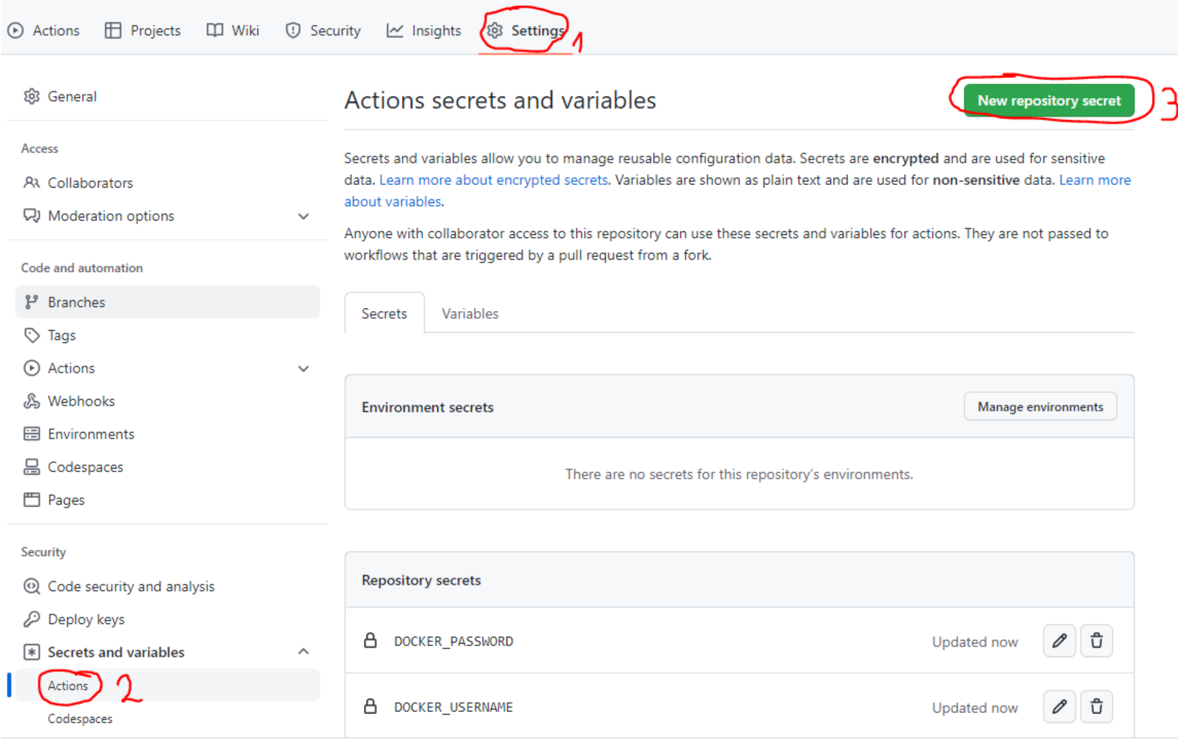
Mission: Build a very simple website and push it in your Docker hub repository by using GitHub actions and pipelines. Then turn on the container of the website in your Ubuntu, open the website with your navigator, give the answer of the 3 random questions, then finish **properly**.

Task 1

1) Create a public repository in github
2) In command lines, you may add your credential of your Git account, then clone the created repository, (Changing directory work is suggested, for example create a folder called TD_build and place yourself there). Go inside the downloaded repository in command line.
3) Add a simple readme file in your main branch, then add, commit and push it to github
4) In command lines, create and use the branch called feat/build_simple_website from the main branch.
5) Add the source code from the folder build_ci into the branch, don't push that to the branch.

Task 2

1) Read the push.yaml file in .github/workflows, understand what it does, you will explain me later.
2) Add your username and the password of your docker account as secrets in github actions website.
   Name them as is it in the yaml code.



Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Environment secrets

There are no secrets for this repository's environments.

Repository secrets

DOCKER_PASSWORD	Updated now
DOCKER_USERNAME	Updated now

3) Change the values of the tag with your username and the repository name of Docker hub.
4) Call the teacher to validate that, Be prepare to explain the code, you may look for some information through internet.
5) Push the code in the branch feat/build_simple_website
6) See your github actions, if everything is successfully done, so pull the new image and run it in your ubuntu machine with the options needed to open the website in the navigator. Open the website in the navigator.
7) Answer the 3 questions, then call the teacher to explain the answers.
8) Invite your partner to your repository, then create a pull_request to merge the code to the main branch, your partner must validate (approve) this, then you complete by merging it.
9) Call the teacher to validate it.

Part 2 – Test & Pipelines

Mission : TEST & BUILD a very simple website and push it in your Docker hub repository by using GitHub actions and pipelines. Then turn on the container of the website in your Ubuntu, open the website with your navigator, give the answer of the 3 random questions, then finish **properly**.

Task 1

1) In command lines, you may add your credential of your Git account, then clone the created repository, (Changing directory work is suggested, for example create a folder called TD_build and place yourself there). Go inside the downloaded repository in command line.
2) In command lines, create a branch called feat/testbuild_simple_website from the main branch.
3) Add the source code from the folder test_ci into the branch and push it to your repository.

Task 2

1) Read the test.yaml file in .github/workflows, understand what does it do, you will explain me later.
2) Add your username and the password of your docker account as secrets in github actions website.
   Name them as is it in the yaml code.
3) Change the values of the tag with your username and the repository name of Docker hub.
4) Call the teacher to validate that, Be prepare to explain the code, you may look for some information through internet.
5) After the first push, you will find some errors in the python code, repair them, push the code again, check the result of the pipelines, try until everything is good.
6) Pull and run the container in your machine, when having the webpage opened, you should have 3 numbers, call the teacher.
7) Answer the 3 questions, then call the teacher to explain the answers.
8) Invite your partner to your repository, then create a pull_request to merge the code to the main branch, your partner must validate (approve) this, then you complete by merging it.

Tips: You may search some stuffs through internet BUT it's suggested to follow the official documentation, not tutorials made in blogs.

Part 3 – Test & Pipelines

Mission : TEST, BUILD & DEPLOY IN PRODUCTION a very simple website in your machine (it's suggested to use a virtual machine, for example the one that is delivered by the teacher), then finish **properly**.

We will use the same repository and yaml pipeline from the Part 2, we will CONTINUE to fill code on it.

Task 1

1) Add your machine as a self-hosted runner in your repository
   Go to your repository->Settings->General->Actions->Runners-> New self-hosted runner.
   Follow the instructions to install the agent in your VM, it is suggested to use the vm provided by the teacher

Task 2

1) Add a second job to your pipeline from the end of the file, after the first job.
2) The new job depends on the previous job, add this property (needs: previous job name)
3) Name the new job
4) Specify the runner, tells that the new job will be run in the self-hosted agent

Task 3

In order to give the rights of executing Docker to github, we need to add the user of github (your username) in the self-host agent. In ubuntu, you can do this with the following commands

```
$ sudo adduser $your_github_username
```

Follow the instructions, then:

```
$ sudo groupadd docker
```

```
$ sudo usermod -aG docker $USER
```

Task 4

In the last "steps", you need to deploy your website in the self-hosted agent, add the command lines that run the website in docker. (read the usefull commands).

Call the teacher when the website is deployed.

Usefull commands

Execute shell files in github
In your terminal, give the permission like this :

```
git update-index --chmod=+x ./.github/scripts/backend_decrypt.sh
```

Pull and run website container from your repository

```
docker pull {docker_username}/repository:tag
docker run -P -d {docker_username}/repository:tag
```

Set specific port for your container

```
docker run -d -p 5003:80 {github_username}/repository:tag
```

(-d means to deploy in background)

Check the port used to allocate the container and visit the web page

```
docker container ls
http:://localhost:port
```

Stop container

```
docker container ls
docker stop container_name
```

Remove container and image

```
docker container ls
docker container rm <container_id>
docker image ls
docker image rm <image_id>
```

Login to github in command line

```
git config --global user.name "username"
git config --global user.password "password"
```

Git commit & push

```
git add .
git commit -m “My message”
git push origin feat/my_branch
```

For each random question

For each random question you have to answer:

1) What is that? Explain with simple sentences, images, etc, and give an example of use.
2) Is it free? Which licenses does it have?
3) Usefulll for DevOps? How?