

1 Mục tiêu

- Github
- Sử dụng docker để triển khai ứng dụng node.js

2 Yêu cầu

- Tạo Github repo chứa và quản lý source code bài tập tuần đến tuần 08.
- Xây dựng docker script để triển khai ứng dụng tuần 08.
- Cần nộp: file links.txt chứa link Github repo của project tuần 08 (đa tạo docker, chọn public cho repo) và tên Docker image trên Hub.

3 Hướng dẫn

3.1 Tạo tài khoản Github và cài đặt GIT

- Truy cập <https://github.com> và tạo tài khoản.
- Truy cập <https://git-scm.com> rồi chọn Install và lựa chọn OS phù hợp để cài đặt GIT lên máy.
- Mở terminal trên máy lên và thực hiện lệnh sau:
`ssh-keygen -t ed25519 -C "<your_email@example.com>"`
- Khi được yêu cầu nhập tên file lưu (Enter file in which to save the key) thì nhập đầy đủ lưu trong folder .ssh để SSH agent tự động xử lý được. Ví dụ với windows 11:
`C:/Users/<username>/.ssh/id_ed25519`
- Phần nhập passphrase nên để trống (chỉ nhấn ENTER) là đã tạo thành công cặp key được lưu trong đường dẫn cụ thể sau:

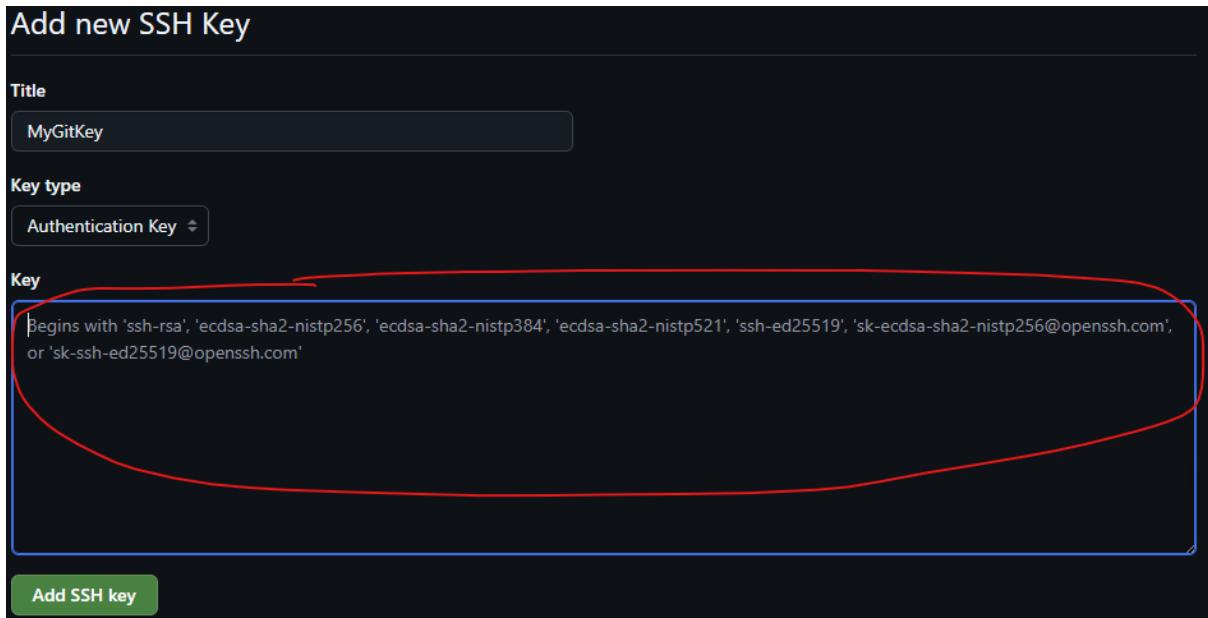
`C:/Users/<username>/.ssh/id_ed25519`

`C:/Users/<username>/.ssh/id_ed25519.pub`

- Đưa khóa private vào SSH agent bằng lệnh sau:

`ssh-add C:/Users/<username>/.ssh/id_ed25519`

- Trở lại github và vào Settings của tài khoản (nhấn vào biểu tượng tài khoản ở góc phải trên để mở menu).
- Chọn mục “SSH and GPG keys”.
- Sau đó dùng Notepad mở file id_ed25519.pub để copy public key và chép lên trên GitHub.

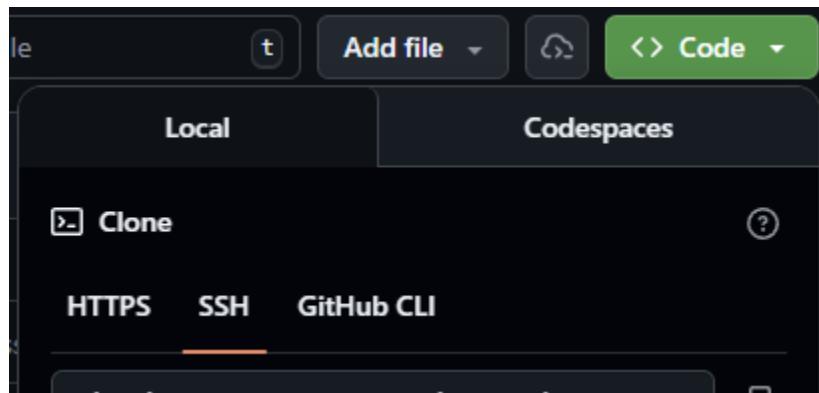


- Sau khi chép key xong thì kiểm tra đã thành công chưa bằng lệnh:

ssh -T git@github.com

3.2 Tạo repository và đẩy source lên Github

- Tại dashboard của Github, ở góc trái trên có button “New” để tạo repository và đặt tên là “**Web_W08**”, nên chọn visibility là **Private** đối với code cần bảo mật, phần Add .gitignore thì lựa chọn là **Node**.
- Trên máy mở terminal và di chuyển đến thư mục hiện hành là folder gốc của source code bài tuần 08 bằng lệnh: **cd /d <đường dẫn thư mục gốc>**
- Khởi tạo git cho source code tuần 08 bằng lệnh: **git init**
- Kết nối với repository đã tạo ở Github: **git remote add origin <URL-REPOSITORY>** (trong đó URL-REPOSITORY là address SSH lấy từ Github của repository)



- Đảm bảo nhánh chính tên ‘main’: **git branch -M main**
- Thực hiện lệnh pull trước tiên: **git pull --rebase origin main**
- Thực hiện commit đầu tiên: **git add .**

git commit -m “initial commit”

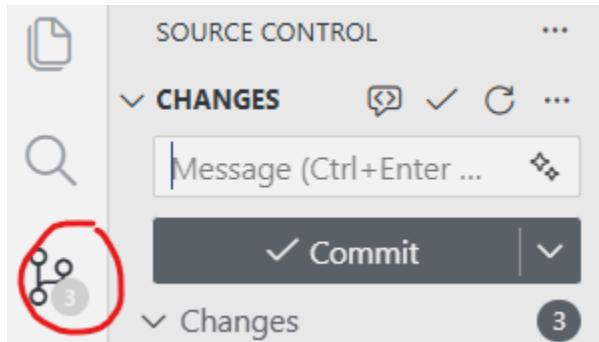
- Thực hiện đẩy lên Github repo: `git push -u origin main`
- Như vậy là đã hoàn thành đẩy source lên Github repo.
- Sau này khi cần đồng bộ với Github repo (lấy thay đổi mới) thì chỉ kéo về: `git pull`
- Hoặc đẩy thay đổi từ source lên Github repo bằng nhóm lệnh:

`git add .`

`git commit -m "message"`

`git push`

- Hoặc sử dụng giao diện của Visual Studio Code để thực hiện các quá trình lấy và đẩy source.



3.3 Tạo tài khoản Docker

- Truy cập <https://www.docker.com> và tạo tài khoản (có thể sử dụng chính tài khoản Github ở trên để làm tài khoản Docker).
- Đăng nhập vào tài khoản mới tạo.
- Tải ứng dụng Docker Desktop về máy và cài đặt.
- Chạy Docker Desktop và tiến hành đăng nhập vào tài khoản.

3.4 Tạo Docker Image cho project thực hành (đến tuần 08)

- Mở File explorer đến thư mục gốc của project thực hành tuần 08.
- Tạo file `.dockerignore` trong thư mục gốc của project (không có tên, chỉ có extension).
- Mở file `.dockerignore` bằng text editor và thêm nội dung sau:

```
node_modules
npm-debug.log
.git
.env
```

- Tạo file `Dockerfile` trong thư mục gốc của project (không có extension, chỉ có tên).
- Mở file `Dockerfile` bằng text editor và thêm nội dung sau:

```
# 1. Chọn Base Image (Nền tảng): Node.js bản nhẹ (Alpine)
FROM node:18-alpine

# 2. Tạo thư mục làm việc trong Container
WORKDIR /app
```

```

# 3. Copy file định nghĩa thư viện trước (để tận dụng Cache)
COPY package*.json ./

# 4. Cài đặt thư viện (Dependencies)
RUN npm install

# 5. Copy toàn bộ source code vào trong Container
# (Lệnh này sẽ bỏ qua những file đã liệt kê trong .dockerignore)
COPY ..

# 6. Khai báo port mà ứng dụng sẽ chạy (chỉ để document, không mở port thật sự)
EXPOSE 3000

# 7. Lệnh mặc định để chạy ứng dụng khi container khởi động
CMD ["npm", "start"]

```

- Thực hiện build Docker image bằng lệnh sau:

docker build -t <tên đăng ký trên docker>/<tên image>:v1 .

Ví dụ: *docker build -t matuan/umt-app:v1 .*

- Sau khi build thành công, kiểm tra trên app Docker Desktop phần Images sẽ thấy image vừa build



- Thực hiện chạy thử (run container):

docker run -d -p 3000:3000 --env-file .env --name <tên container> <tên image>

Ví dụ: *docker run -d -p 3000:3000 --env-file .env --name umt-1 matuan/umt-app:v1*

- Sau đó sử dụng trình duyệt để truy cập: <http://localhost:3000> để kiểm tra ứng dụng đã chạy.

3.5 Đẩy Docker image lên hub để chia sẻ cũng như triển khai trên máy khác

- Có sử dụng Docker Desktop và đã login thì không cần thực hiện login trong terminal nữa.
- Thực hiện lệnh sau để đẩy image lên Hub:

docker push <tên image>

Ví dụ: *docker push matuan/umt-app:v1*

- Trên máy khác thì trước khi tải và chạy container từ image trên Hub cần phải tạo file .env (vì không đóng gói theo image vì dữ liệu nhạy cảm, có thể thử nghiệm bằng thư mục khác nhưng lưu ý thay đổi port của host).
- Tải và chạy container bằng lệnh sau:

docker run -d -p 3000:3000 --env-file .env --restart always --name <tên container> <tên image trên hub>