

Triển khai CI/CD cho DSM Web Server

CI/CD deploy web lên web hosting chạy trên DSM

- [CI/CD sử dụng GitHub Actions](#)
- [CI/CD sử dụng Jenkins](#)

CI/CD sử dụng GitHub Actions

Cấu hình trên DSM

Tạo shared folder

Trên DSM

- Control Panel > Shared Folder > Create Shared Folder > Name: docker
- Trong folder tạo thêm folder cho các web khác nhau
 - docker/Web/web_x
 - docker/Web/web_y
 - docker/Web/web_z

Shared Folder Creation Wizard

Set up basic information

Name *:

Description:

Location:

Volume 1: Btrfs

☐ Hide this shared folder in "My Network Places"
☐ Hide sub-folders and files from users without permissions ⓘ
☐ Enable Recycle Bin
☒ Restrict access to administrators only

Note: [How to set up a Recycle Bin emptying schedule](#)

* This field is required.

Next

Tạo container web hosting

Trên DSM truy cập Container Manager. Tạo một project và sử dụng docker compose để cấu hình container mới.

Với web tĩnh chỉ có html thì sử dụng image `nginx:alpine`

```
services:
  web-cv:
    image: nginx:alpine
    container_name: web-cv # Tên container
    volumes:
      - /volume1/docker/Web/cv:/usr/share/nginx/html:ro # th? m?c code # ro = readonly
    ports:
      - "9901:80" # port ?? truy c?p. Trong tr??ng h?p này là 192.168.1.88:9901
    restart: unless-stopped
```

Với web động sử dụng php thì sử dụng image `php:8.3-apache`

```
services:
  web-upkgs:
    image: php:8.3-apache
    container_name: web-upkgs
    volumes:
      - /volume1/docker/Web/upkgs:/var/www/html:rw # rw = read write
```

```
ports:
  - "9902:80"
restart: unless-stopped
```

Cần chú ý map đúng thư mục code.

Copy source code vào đúng thư mục đã mapping.

Chạy container mới tạo. Nếu truy cập được qua local (ex: 192.168.1.88:9902) là thành công là thành công.

Tạo user deploy và phân quyền

Trên DSM tạo một user mới là deploy để dùng cho việc deploy web

User này cần phải có quyền **administrators** để có thể sử dụng được ssh

Bật SSH

DSM > Control Panel > Terminal & SNMP > Enable SSH service > **Apply**

Tạo SSH key cho GitHub Actions

- Trên máy tính Windows/macOS/Linux khác có ssh-keygen
 - `ssh-keygen -t ed25519 -C "github-actions-to-dsm"`
 - Đặt tên cho key (ex: id_ed25519_web_deploy)
 - Passphrase có thể để trống
 - Khi thành công sẽ có thông báo dạng như sau
Your identification has been saved in id_ed25519_web_deploy
Your public key has been saved in id_ed25519_web_deploy.pub
- Xem nội dung public key
 - `cat id_ed25519_web_deploy.pub`

Thêm public key vào user deploy

- Truy cập vào DSM thông qua ssh bằng user deploy
`ssh deploy@IP_C?A_DSM`
- Tạo thư mục .ssh và file authorized_keys
`mkdir -p ~/.ssh`
`chmod 700 ~/.ssh`
vi ~/.ssh/authorized_keys
- Dán nội dung id_ed25519_web_deploy.pub đã tạo ở trên vào file này
- Set quyền `chmod 600 ~/.ssh/authorized_keys`
- Thử đăng nhập bằng key
`ssh -i id_ed25519_dsm deploy@IP_C?A_DSM`
- Nếu login thành công mà không hỏi password là thành công.

Lưu ý

- `chmod 700` cấp quyền đọc ghi và thực thi cho chủ sở hữu
- `chmod 600` cấp quyền đọc ghi cho chủ sở hữu

Lưu ý

Nếu đăng nhập vào vẫn yêu cầu password

- Cần đảm bảo chắc chắn user deploy có **quyền admin**
- Đảm bảo cấp đúng quyền `chmod 700` và `chmod 600`
- Đảm bảo gõ đúng tên key đã tạo (ex `id_ed25519_web_deploy` và `id_ed25519_web_deploy.pub`)
- Nếu vẫn không được thì dùng `sudo grep -E "PubkeyAuthentication|AuthorizedKeysFile" /etc/ssh/sshd_config` để kiểm tra config
 - Kết quả mong đợi
`PubkeyAuthentication yes`
`AuthorizedKeysFile .ssh/authorized_keys`
 - Nếu không đúng 100% (`PubkeyAuthentication no`, `AuthorizedKeysFile` không chính xác, có dấu `#` comment phía trước) thì cần sửa lại cho đúng `sudo vi /etc/ssh/sshd_config`
 - Restart ssh `sudo systemctl restart sshd`
 - Thử truy cập lại `ssh -i id_ed25519_web_deploy -o IdentitiesOnly=yes deploy@192.168.1.88`
- Nếu vẫn chưa thành công
 - Thử kiểm tra quyền bằng lệnh `ls -ld ~`
 - Nếu quyền quá rộng
`777 (drwxrwxrwx)`
`775 (drwxrwxr-x)`
thì cần đưa về mặc định `755 (drwxr-xr-x)`
`sudo chmod 755 ~`
 - Restart ssh `sudo systemctl restart sshd`
 - Thử truy cập lại `ssh -i id_ed25519_web_deploy -o IdentitiesOnly=yes deploy@192.168.1.88`

Lưu ý:

Để đảm bảo bảo mật hãy tắt PasswordAuthentication

- `sudo vi /etc/ssh/sshd_config`
- Sửa `PasswordAuthentication` thành `no`
- Restart ssh `sudo systemctl restart sshd`
- Thử truy cập lại
 - `ssh -i id_ed25519_web_deploy -o IdentitiesOnly=yes deploy@192.168.1.88` => Thành công
 - `ssh deploy@192.168.1.88` => Thất bại

Setup GitHub Secrets

M? port

Trước tiên để GitHub có thể truy cập tới DSM thông qua SSH thì cần mở port 22 của máy chủ DSM. Có thể map bất kỳ cổng nào của router vào cổng 22 của DSM

Virtual Server for	:	Single IPs Account/ WAN1	
Index	:	3 ▼	
Application	:	DSM_SSH	▼
Protocol	:	TCP ▼	
Start Port Number	:	14322	
End Port Number	:	14322	
Local IP Address	:	192.168.1.88	
Start Port Number(Local)	:	22	
End Port Number(Local)	:	22	

Thêm secrets

Trên GitHub repo > Settings > Secrets and variables > Actions > New repository secret

- `DSM_HOST`: IP của router (hoặc domain name)
- `DSM_PORT`: Port vừa mở phía trên.
- `DSM_USER`: Username của account deploy vừa tạo
- `DSM_PATH`: Folder chứa source code trên DSM (ex: `/volume1/docker/Web/cv`)
- `DSM_SSH_KEY`: Nội dung private key đã tạo
 - `cat id_ed25519_web_deploy` để xem nội dung private key

Tạo GitHub Actions *(Từ phần này chỉ bao gồm lý thuyết, chưa test)*

Trong repo trên GitHub tạo file `.github/workflows/deploy.yml`

```

name: Deploy to DSM via SSH + rsync

on:
  push:
    branches: [ main ]      # ??i l?i branch b?n dùng, ví d? 'master' ho?c 'prod'
    workflow_dispatch:      # cho phép ch?y tay t? tab Actions

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Start SSH agent
        uses: webfactory/ssh-agent@v0.9.0
        with:
          ssh-private-key: ${ secrets.DSM_SSH_KEY }

      - name: Add DSM host to known_hosts
        run: |
          mkdir -p ~/.ssh
          ssh-keyscan -p ${ secrets.DSM_PORT } ${ secrets.DSM_HOST } >> ~/.ssh/known_hosts

      - name: Deploy via rsync
        run: |
          rsync -avz --delete \
            --exclude ".git" \
            --exclude ".github" \
            ./ \
            ${ secrets.DSM_USER }@${ secrets.DSM_HOST }:${ secrets.DSM_PATH }

      # (Tùy ch?n) Set l?i owner cho ?úng user ch?y web trên DSM
      # N?u c?n, b? comment ?o?n này và ch?nh 'http:http' theo user c?a Synology/Container
      # - name: Fix permissions on DSM
      #   run: |
      #     ssh ${ secrets.DSM_USER }@${ secrets.DSM_HOST } -p ${ secrets.DSM_PORT } \
      #       "chown -R http:http ${DSM_PATH}"

      # Gi?i thích nhanh:
      # actions/checkout@v4: clone code t? repo vào runner.
      # webfactory/ssh-agent: load private key vào ssh-agent ?? rsync/ssh dùng.
      # ssh-keyscan: thêm host DSM vào known_hosts ?? tránh prompt t??ng tác.
      # rsync -avz --delete:
      #   -a: archive (gi? permission, symlink, v.v...)
      #   -v: verbose
      #   -z: nén khi truy?n
      #   --delete: xóa file trên DSM n?u ?ã xóa trong repo
      #   --exclude: không ??y .git, .github lên DSM.

```

Ok như vậy là đã xong. Hãy đảm bảo đã mount chính xác folder cho các container php, nginx như bước đầu tiên

- php: `/volume1/docker/webapp -> /var/www/html`
- Nginx: `/volume1/docker/webapp -> /usr/share/nginx/html`

Test

1. Thực hiện commit trên repo
2. Push lên main (hoặc branch đã chọn)
3. Vào tab Actions của repo → chọn workflow Deploy to DSM via SSH + rsync:
 - Xem log:
 - Nếu lỗi SSH: check lại DSM_HOST, DSM_PORT, firewall/router.
 - Nếu lỗi quyền: check quyền user `deploy` với folder `/volume1/docker/webapp`.
4. Nếu job xanh (success), SSH vào DSM:
 - Kiểm tra:
 - `ls -l /volume1/docker/Web/cv`
 - Dùng DSM web truy cập tới thư mục được mount và kiểm tra.
 - Truy cập và kiểm tra thực tế trên web

CI/CD sử dụng Jenkins