

BÁO CÁO BÀI TẬP

Môn học: Lập trình an toàn và khai thác lỗ hổng phần mềm

Lab 1: Automating Everything as Code (DevOps/CI-CD/Git)

GVHD: Nguyễn Hữu Quyền

THÔNG TIN CHUNG:

Lớp: NT521.P12.ANTT.2

STT	Họ và tên	MSSV	Email
1	Lại Quan Thiên	22521385	22521385@gm.uit.edu.vn
2	Mai Nguyễn Nam Phương	22521164	22521164@gm.uit.edu.vn
3	Đặng Đức Tài	22521270	22521270@gm.uit.edu.vn
4	Hồ Diệp Huy	22520541	22520541@gm.uit.edu.vn

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

1. Cho biết thông tin về lần commit vừa thực hiện:

- Commit ID
- Ngày giờ commit
- Thông điệp commit

```
(hohuy@192) - [~/labs/git-intro]
$ git log
commit 93fabfdf85c7b4af1fc033ed912e099db856580b (HEAD -> master)
Author: Ho Huy <hohuy0107@gmail.com>
Date: Sat Sep 28 23:32:44 2024 -0700

    Added additional line to file

commit a53e241adc1e155319b343da173a3b067ae9d1cc
Author: Ho Huy <hohuy0107@gmail.com>
Date: Sat Sep 28 23:29:02 2024 -0700

    Committing README>MD to begin tracking changes
```

- Commit id: 93fabfdf85c7b4af1fc033ed912e099db856580b
- Ngày giờ: Sat Sep 28 23:32:44 2024
- Thông điệp: Added additional line to file

2. Sinh viên hoàn thiện các bước 3 đến 11 trong tích hợp Git với Github, trình bày step-by-step có minh chứng

```
(hohuy@192)~/labs/git-intro
$ mkdir devops-study-team

(hohuy@192)~/labs/git-intro
$ cp README.MD devops-study-team

(hohuy@192)~/labs/git-intro
$ cd devops-study-team

(hohuy@192)~/labs/git-intro/devops-study-team
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/hohuy/labs/git-intro/devops-study-team/.git/

(hohuy@192)~/labs/git-intro/devops-study-team
$ git remote add origin https://github.com/hohuyy/NT521.git

(hohuy@192)~/labs/git-intro/devops-study-team
$ git add README.MD

(hohuy@192)~/labs/git-intro/devops-study-team
$ git commit -m "First commit"
[master (root-commit) b3cbe28] First commit
1 file changed, 4 insertions(+)
create mode 100644 README.MD

(hohuy@192)~/labs/git-intro/devops-study-team
$ git status
On branch master
nothing to commit, working tree clean

(hohuy@192)~/labs/git-intro/devops-study-team
$ git log
commit b3cbe28aff515ea23c43a3d3607d0084807f7e75 (HEAD -> master)
Author: Ho Huy <hohuy0107@gmail.com>
Date:   Sun Sep 29 12:27:49 2024 -0700

    First commit

(hohuy@192)~/labs/git-intro/devops-study-team
$ git branch -M main

(hohuy@192)~/labs/git-intro/devops-study-team
$ git push -u origin main
Username for 'https://github.com': hohuyy
Password for 'https://hohuyy@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/hohuyy/NT521.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

- **Bước 1:** Dùng lệnh mkdir để tạo thư mục devops-study-team:

```
(hohuy@192) - [~/labs/git-intro]  
$ mkdir devops-study-team
```

- **Bước 2:** Dùng lệnh cp để copy file README.MD vào thư mục vừa tạo:

```
(hohuy@192) - [~/labs/git-intro]  
$ cp README.MD devops-study-team
```

- **Bước 3:** Sử dụng lệnh cd để chuyển đến thư mục devops-study-team, sau đó dùng git init để khởi tạo Git repository:

```
(hohuy@192) - [~/labs/git-intro]  
$ cd devops-study-team  
  
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git init  
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:  
hint:  
hint:   git config --global init.defaultBranch <name>  
hint:  
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
hint: 'development'. The just-created branch can be renamed via this command:  
hint:  
hint:   git branch -m <name>  
Initialized empty Git repository in /home/hohuy/labs/git-intro/devops-study-team/.git/
```

- **Bước 4:** Để trở Git repository đến GitHub repository ta dùng lệnh git remote như sau:

```
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git remote add origin https://github.com/hohuy/NT521.git
```

- **Bước 5:** Stage và commit tập tin README.MD lần lượt bằng git add và git commit:

```
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git add README.MD  
  
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git commit -m "First commit"  
[master (root-commit) b3cbe28] First commit  
1 file changed, 4 insertions(+)  
create mode 100644 README.MD
```

- **Bước 6:** Kiểm tra commit bằng git log, git status:

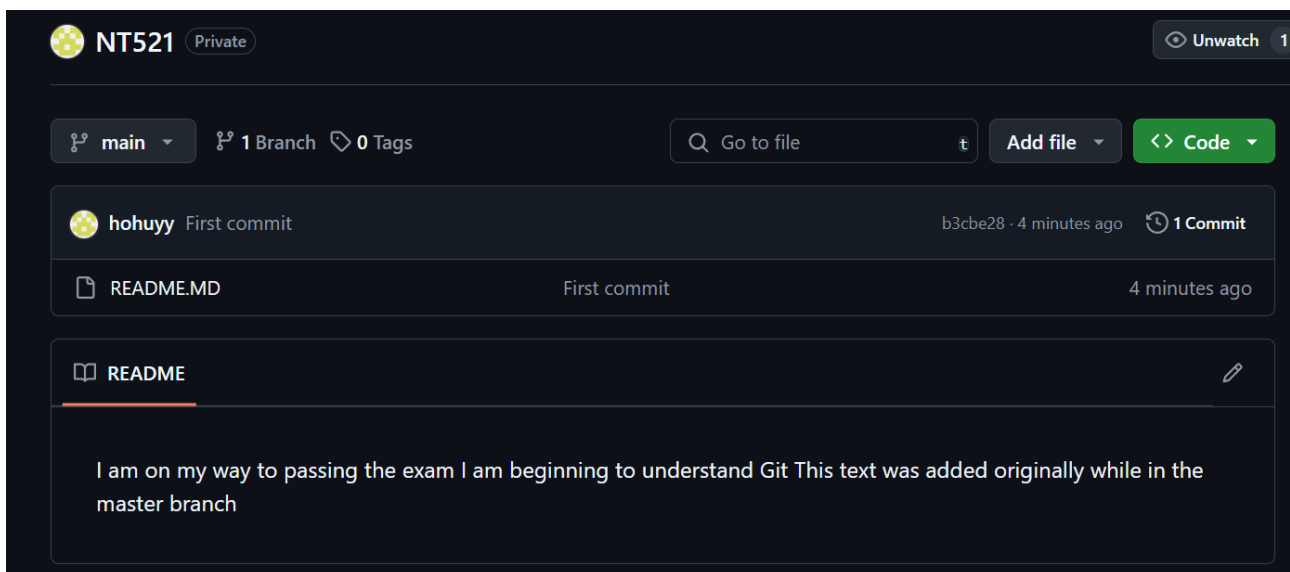
```
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git status  
On branch master  
nothing to commit, working tree clean  
  
(hohuy@192) - [~/labs/git-intro/devops-study-team]  
$ git log  
commit b3cbe28aff515ea23c43a3d3607d0084807f7e75 (HEAD -> master)  
Author: Ho Huy <hohuy0107@gmail.com>  
Date: Sun Sep 29 12:27:49 2024 -0700  
  
    First commit
```

- **Bước 6:** Đổi tên nhánh hiện tại thành main bằng git branch -M main, để gửi tập tin từ Git lên Github (nhánh main) ta dùng lệnh git push như sau:

```
(hohuy@192) - [~/labs/git-intro/devops-study-team]
$ git branch -M main

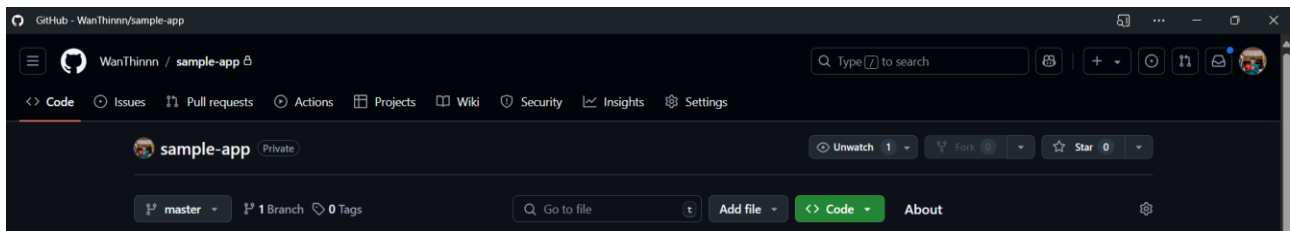
(hohuy@192) - [~/labs/git-intro/devops-study-team]
$ git push -u origin main
Username for 'https://github.com': hohuyy
Password for 'https://hohuyy@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/hohuyy/NT521.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

- **Bước 7:** Truy cập vào Github để kiểm tra tập tin:

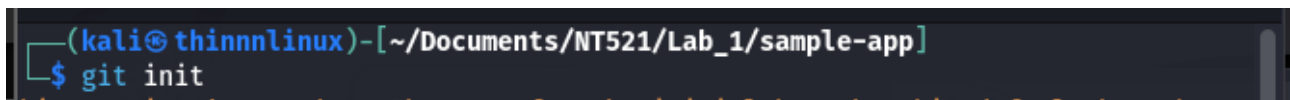


3. Sinh viên thực hiện commit và push code với thông điệp “Changed port from 8080 to 5050”, trình bày step-by-step có minh chứng

Bước 1: Tạo repository trên Github



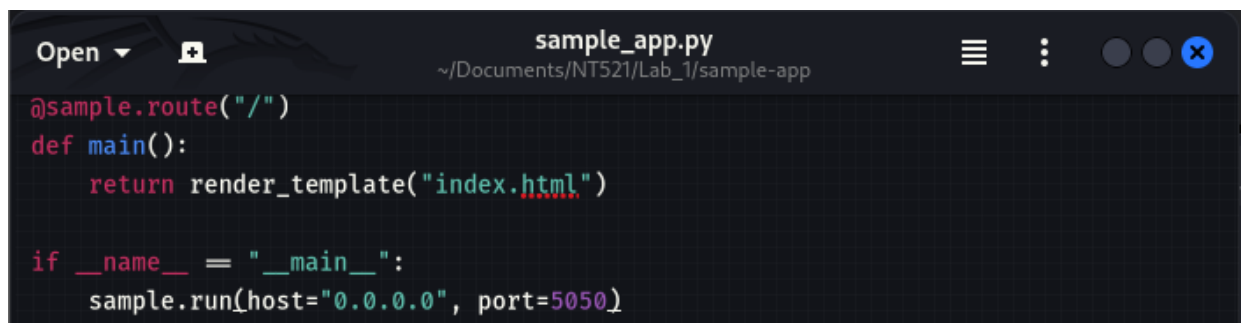
Bước 2: Khởi tạo Git repository trong thư mục sample-app: Mở Git Bash trong thư mục sample-app và chạy lệnh `git init`



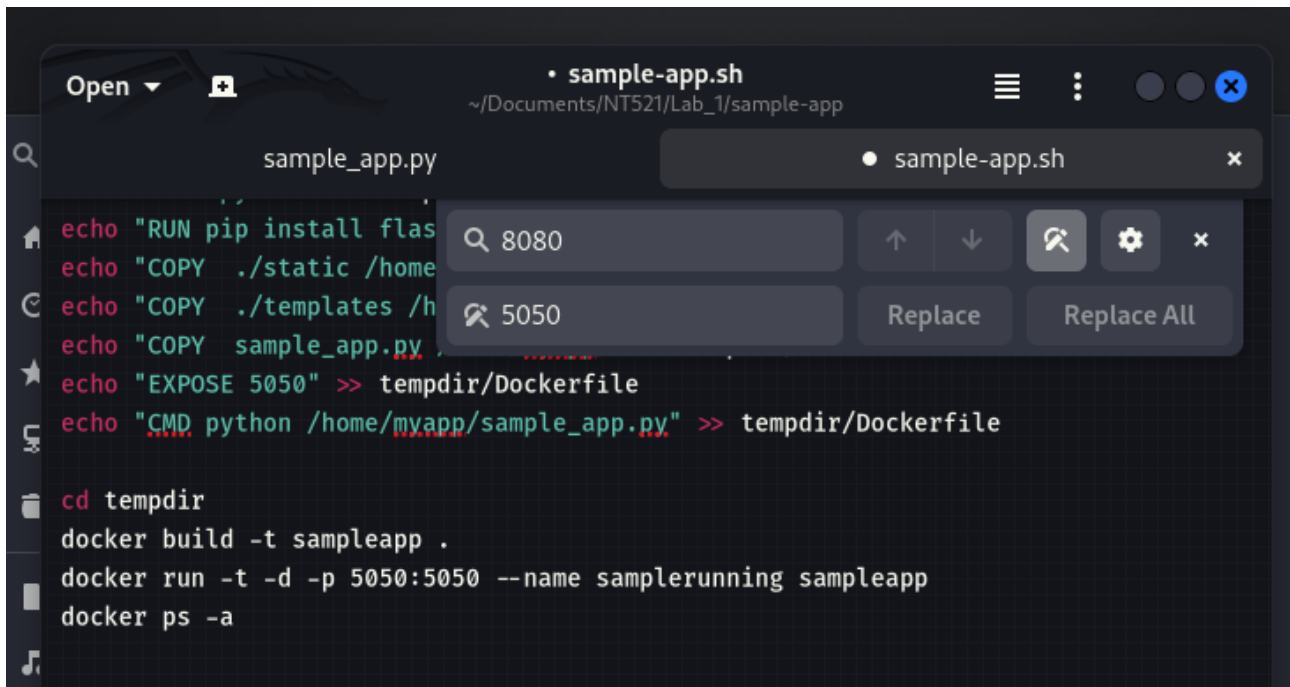
Bước 3: Kết nối Git repository với GitHub: `git remote add origin` <https://github.com/WanThinnn/sample-app.git>



Bước 4: Chỉnh sửa file `sample_app.py`: Mở file `sample_app.py` và thay đổi port từ 8080 thành 5050:



Bước 5: Chỉnh sửa file **sample-app.sh**: Tìm **port 8080** và thay thành **5050**:



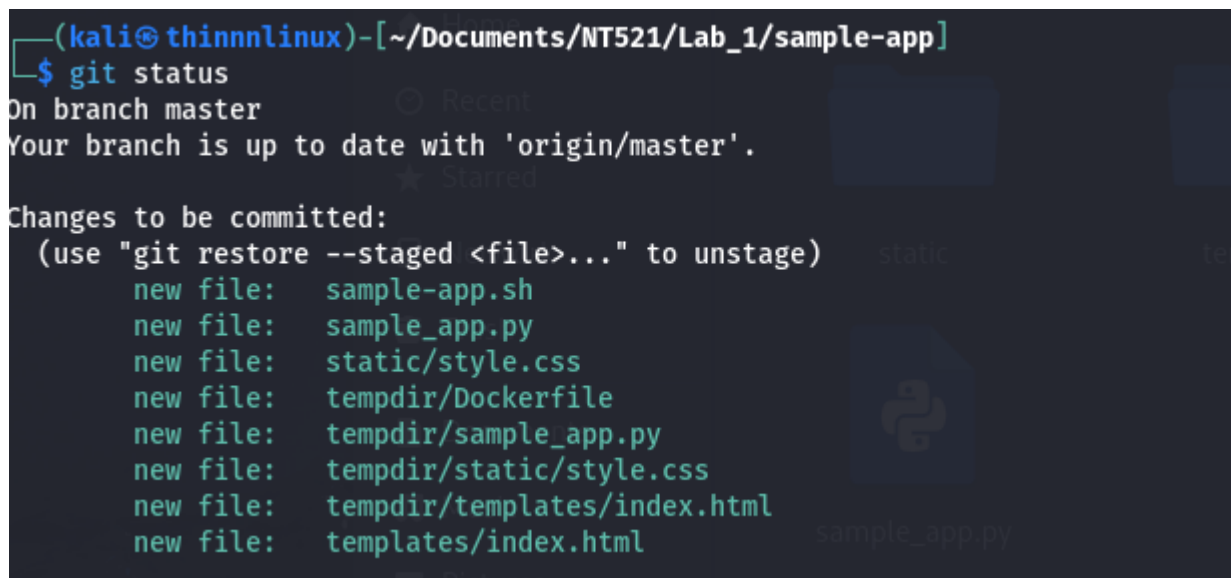
```
sample-app.sh
~/Documents/NT521/Lab_1/sample-app

sample_app.py
sample-app.sh

echo "RUN pip install flas
echo "COPY ./static /home
echo "COPY ./templates /h
echo "COPY sample_app.py .
echo "EXPOSE 5050" >> tempdir/Dockerfile
echo "CMD python /home/myapp/sample_app.py" >> tempdir/Dockerfile

cd tempdir
docker build -t sampleapp .
docker run -t -d -p 5050:5050 --name samplerunning sampleapp
docker ps -a
```

Bước 6: Stage các file trên: **git add .**



```
(kali@thinnnlinux)-[~/Documents/NT521/Lab_1/sample-app]
$ git status
On branch master
Your branch is up to date with 'origin/master'.

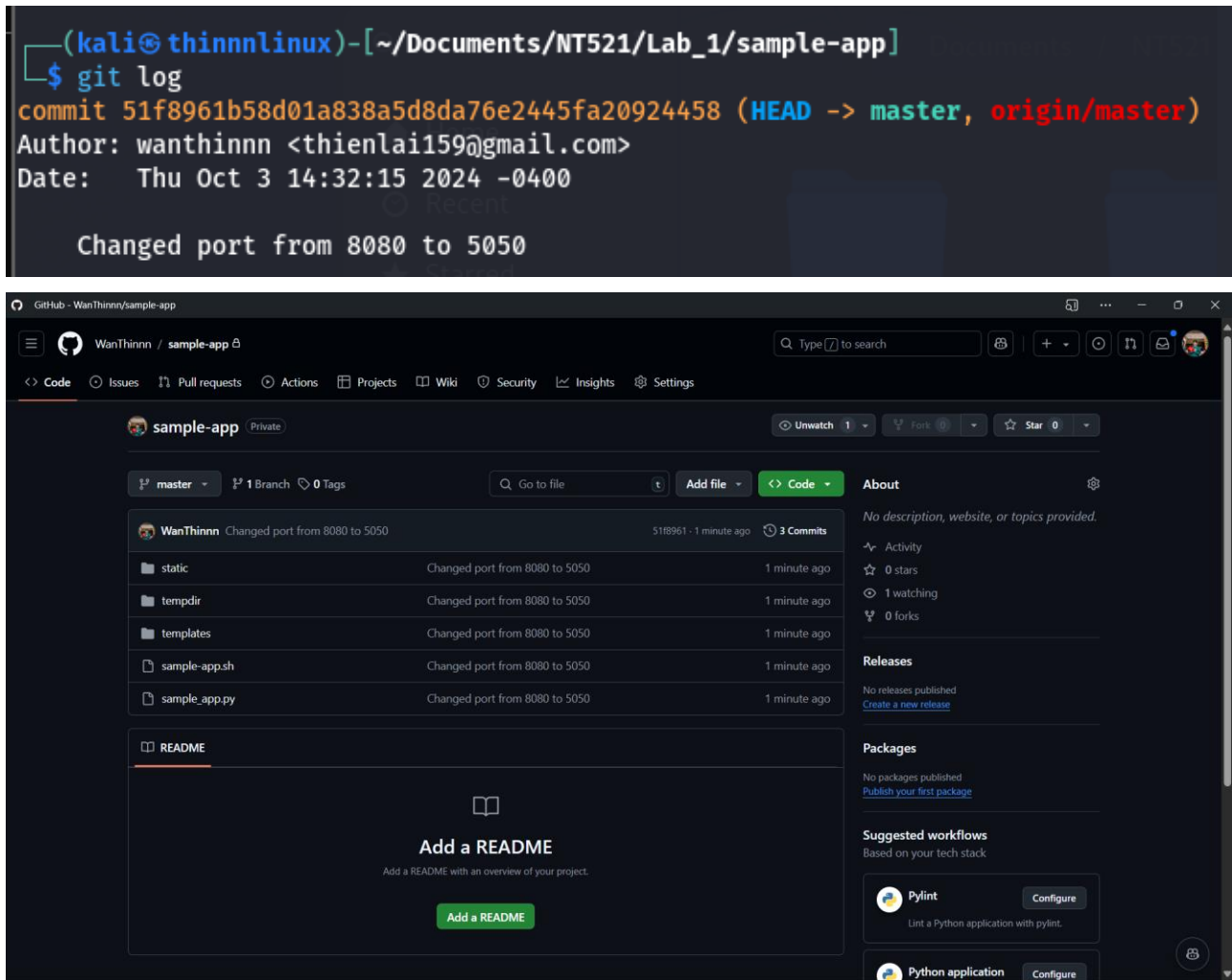
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   sample-app.sh
    new file:   sample_app.py
    new file:   static/style.css
    new file:   tempdir/Dockerfile
    new file:   tempdir/sample_app.py
    new file:   tempdir/static/style.css
    new file:   tempdir/templates/index.html
    new file:   templates/index.html
```


Bước 7: commit với thông điệp: `git commit -m "Changed port from 8080 to 5050"`

```
(kali@thinnlinux)-[~/Documents/NT521/Lab_1/sample-app]
$ git commit -m "Changed port from 8080 to 5050"
[master 51f8961] Changed port from 8080 to 5050
8 files changed, 75 insertions(+)
create mode 100755 sample-app.sh
create mode 100755 sample_app.py
create mode 100755 static/style.css
create mode 100644 tmpdir/Dockerfile
create mode 100755 tmpdir/sample_app.py
create mode 100755 tmpdir/static/style.css
create mode 100755 tmpdir/templates/index.html
create mode 100755 templates/index.html
```

Bước 8: Push thay đổi lên Github: `git push -u origin master`

```
(kali@thinnlinux)-[~/Documents/NT521/Lab_1/sample-app]
$ git push
Username for 'https://github.com': wanthinnn
Password for 'https://wanthinnn@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 6 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1.36 KiB | 1.36 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/WanThinnn/sample-app.git
553dc4e..51f8961 master -> master
```


Bước 9: Kiểm tra trên Github:

The image shows a terminal window and a GitHub repository page. The terminal window displays the output of a `git log` command, showing a commit with the message "Changed port from 8080 to 5050". The GitHub page shows the repository "sample-app" by "WanThinnn", with a commit history table and a README section.

```
(kali@thinnnlinux)-[~/Documents/NT521/Lab_1/sample-app]
$ git log
commit 51f8961b58d01a838a5d8da76e2445fa20924458 (HEAD -> master, origin/master)
Author: wanthinna <thienlai159@gmail.com>
Date: Thu Oct 3 14:32:15 2024 -0400

    Changed port from 8080 to 5050
```

GitHub - WanThinnn/sample-app

WanThinnn / sample-app

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

sample-app Private

master 1 Branch 0 Tags

Go to file Add file Code

WanThinnn Changed port from 8080 to 5050 51f8961 · 1 minute ago 3 Commits

static	Changed port from 8080 to 5050	1 minute ago
tempdir	Changed port from 8080 to 5050	1 minute ago
templates	Changed port from 8080 to 5050	1 minute ago
sample-app.sh	Changed port from 8080 to 5050	1 minute ago
sample_app.py	Changed port from 8080 to 5050	1 minute ago

README

Add a README

Add a README with an overview of your project.

Add a README

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Suggested workflows

Based on your tech stack

Pylint Configure

Lint a Python application with pylint.

Python application Configure

4. Sinh viên tự tìm hiểu và giải thích tùy chọn -v ở lệnh chạy Docker trên.

Lệnh sau sẽ khởi chạy Jenkins Docker container và sau đó cho phép các lệnh Docker được thực thi bên trong máy chủ Jenkins.

```
(kali@phaphajian) - [~/sample-app]
$ docker run --rm -u root -p 8080:8080 -v jenkins-
data:/var/jenkins_home -v $(which docker):/usr/bin/docker -v
/var/run/docker.sock:/var/run/docker.sock -v "$HOME":/home --name
jenkins_server jenkins/jenkins:lts
```

- Ta có: **-v <host_path>:<container_path>**

+ <host_path>: Đường dẫn đến thư mục hoặc file trên máy chủ (host).

+ <container_path>: Đường dẫn đến thư mục hoặc file trong container.

- Các -v trong lệnh trên:

-v jenkins-data:/data/var/jenkins_home

+ Ánh xạ dữ liệu từ volume có tên jenkins-data (hoặc tạo mới nếu chưa có) trên máy chủ đến thư mục /var/jenkins_home trong container.

+ Ý nghĩa: Đây là nơi Jenkins lưu trữ dữ liệu của mình, nên cần được gắn kết để dữ liệu Jenkins không bị mất khi container ngừng hoạt động.

-v \$(which docker) :/usr/bin/docker

+ Gắn kết tập tin thực thi Docker (which docker trả về đường dẫn của file Docker trên máy chủ) vào vị trí /usr/bin/docker trong container.

+ Ý nghĩa: Điều này cho phép Jenkins trong container có thể gọi Docker trên hệ thống máy chủ để quản lý các container khác.

-v /var/run/docker.sock:/var/run/docker.sock

+ Gắn kết socket Docker từ máy chủ (đường dẫn /var/run/docker.sock) vào container.

+ Ý nghĩa: Socket này giúp container Jenkins có thể tương tác với **Docker daemon** trên máy chủ để quản lý các container Docker khác từ bên trong Jenkins.

-v "\$HOME" :/home

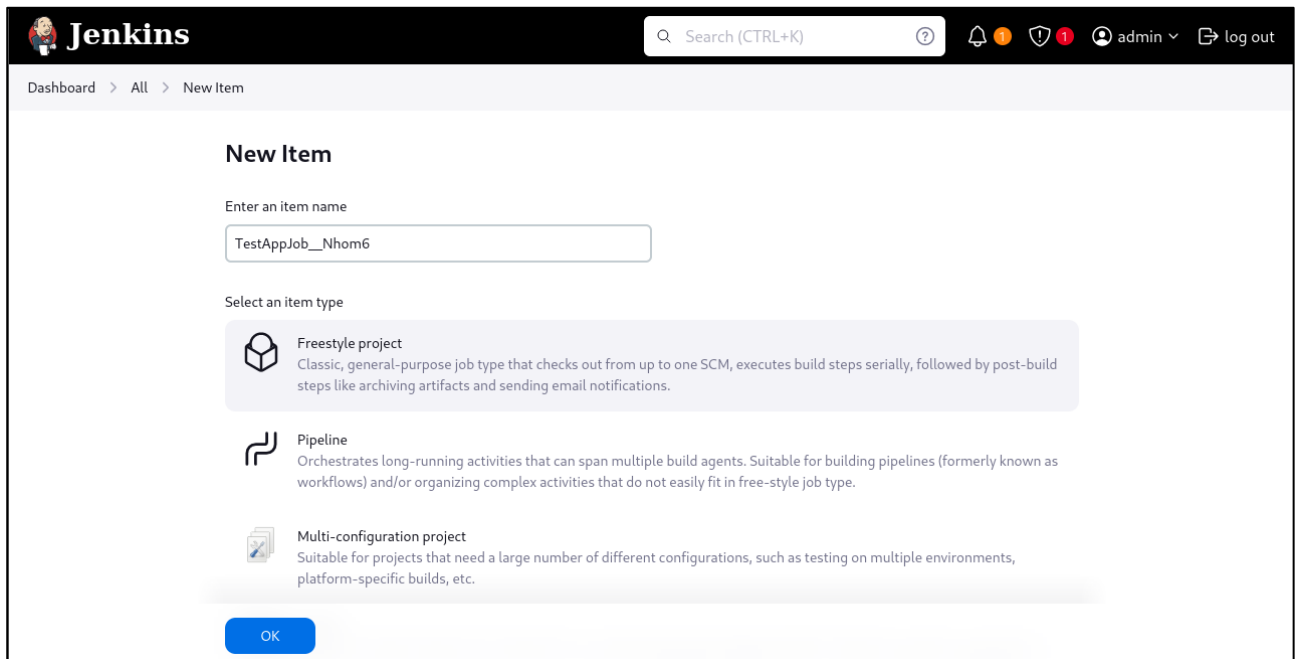
+ Gắn kết thư mục home của người dùng trên máy chủ với thư mục **/home** trong container.

+ Ý nghĩa: Điều này giúp container có thể truy cập vào thư mục home của người dùng trên máy chủ, cho phép chia sẻ dữ liệu giữa container và máy chủ.

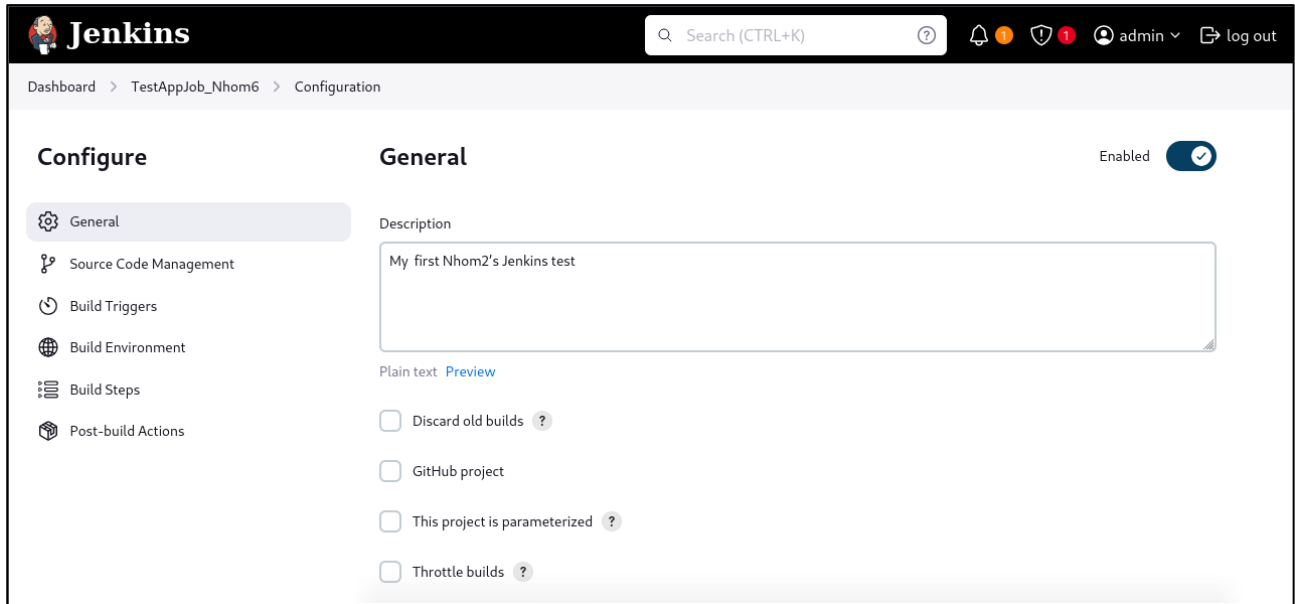
- **Tóm lại:** Tùy chọn -v giúp container có thể truy cập và chia sẻ tài nguyên từ hệ thống máy chủ một cách trực tiếp, cho phép container và máy chủ giao tiếp qua lại dễ dàng.

5. Sinh viên hoàn thành job kiểm tra bản dựng ứng dụng tự động theo gợi ý bên dưới, trình bày step-by-step có minh chứng.

- **Bước 1:** Tạo một job mới để kiểm tra app-sample bằng cách chọn **New item**, đặt tên là **TestAppJob_Nhom6** và chọn **Freestyle project**



- **Bước 2:** Viết **Description** như hướng dẫn và thực hiện chọn **Source Code Management** là **None** và **Build Triggers** như hình:



Dashboard > TestAppJob_Nhom6 > Configuration

Configure

Advanced ▾

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Source Code Management

☒ None
☐ Git ?

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?
☒ Build after other projects are built ?

Projects to watch

BuildAppJob

☒ Trigger only if build is stable

Save Apply

- **Bước 3:** Trong phần **Build Step** chọn **execute shell** với đoạn code như bên dưới:

Dashboard > TestAppJob_Nhom6 > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Build Steps

≡ Execute shell ?

Command

See the list of available environment variables

```
1:8080 | grep "Có làm thì mới có ăn :3" | awk '{if (length($0) == 0) exit 1; else exit 0;}'
```

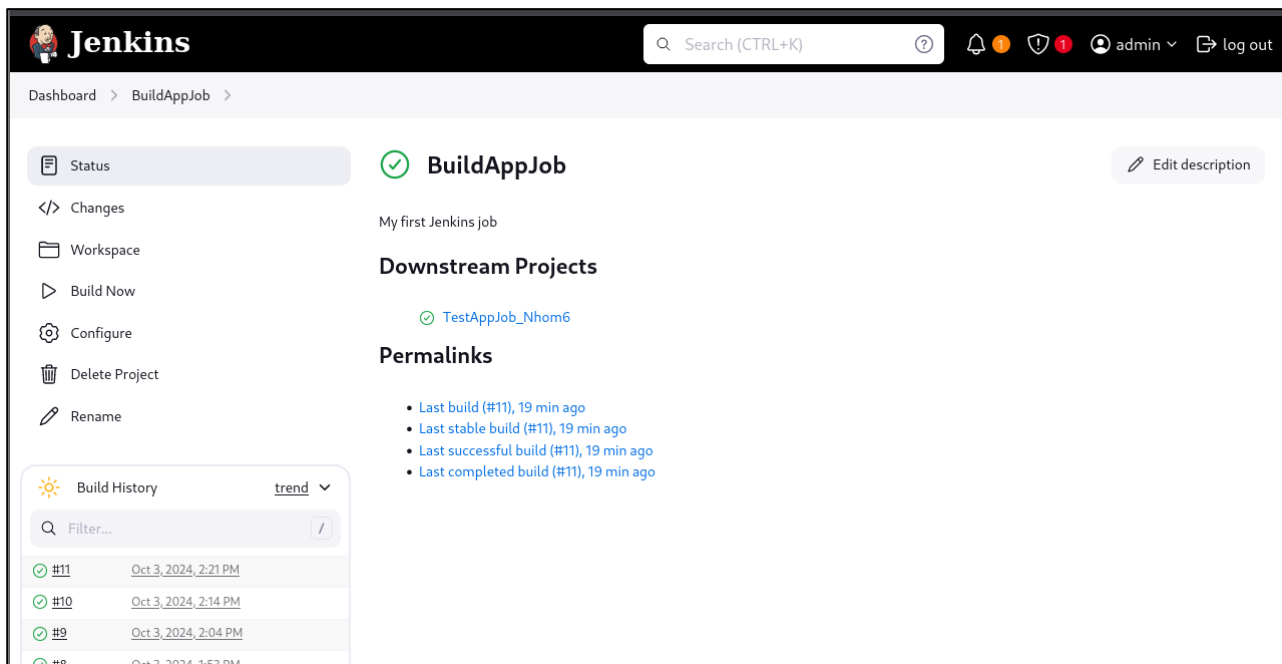
Advanced ▾

Add build step ▾

Save Apply

Chức năng của đoạn code: Sử dụng dữ liệu trả về thông qua **1:8080** để tìm kiếm nội dung có chưa câu **“Có làm thì mới có ăn :3”**. Nếu kết quả trả về là tồn tại dòng chữ đó trong nội dung sẽ thực hiện **exit 1**, ngược lại sẽ **exit 0**.

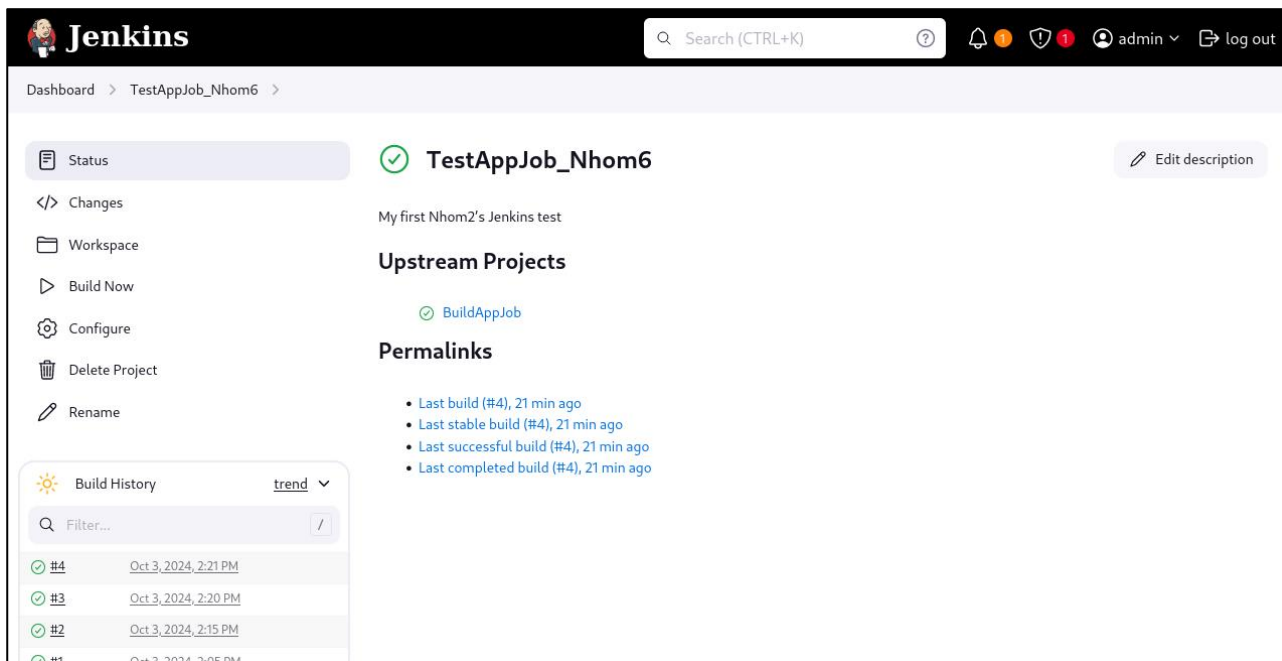
- Kết quả chạy BuildAppJob:



The screenshot shows the Jenkins dashboard for the 'BuildAppJob' project. The interface includes a top navigation bar with the Jenkins logo, a search bar, and user information. The left sidebar contains navigation links for Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area displays the job status as 'BuildAppJob' with a green checkmark icon. Below the job name, it says 'My first Jenkins job'. The 'Downstream Projects' section lists 'TestAppJob_Nhom6'. The 'Permalinks' section provides links for the last build, last stable build, last successful build, and last completed build, all of which are 19 minutes ago. A 'Build History' table is visible on the left, showing builds #8 through #11 with their respective timestamps.

Build Number	Timestamp
#11	Oct 3, 2024, 2:21 PM
#10	Oct 3, 2024, 2:14 PM
#9	Oct 3, 2024, 2:04 PM
#8	Oct 3, 2024, 1:53 PM

- Kết quả chạy TestAppJob_Nhom6:

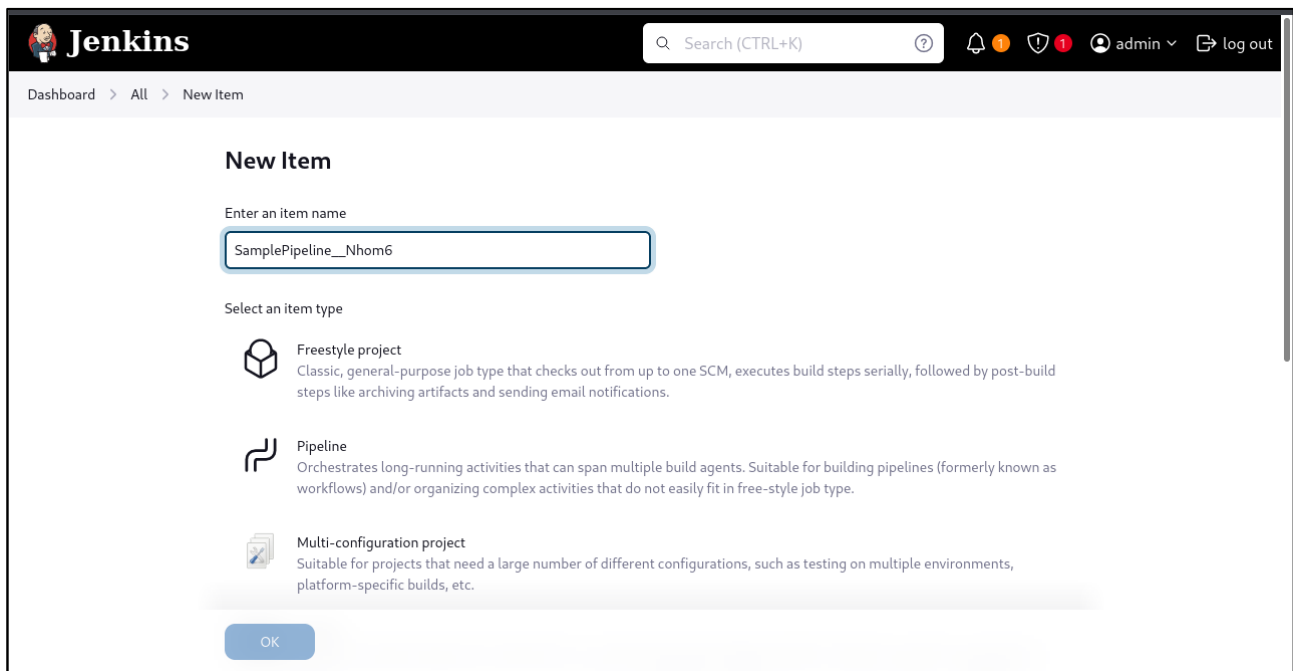


The screenshot shows the Jenkins dashboard for the 'TestAppJob_Nhom6' project. The interface is similar to the previous one, with a top navigation bar and a left sidebar. The main content area displays the job status as 'TestAppJob_Nhom6' with a green checkmark icon. Below the job name, it says 'My first Nhom2's Jenkins test'. The 'Upstream Projects' section lists 'BuildAppJob'. The 'Permalinks' section provides links for the last build, last stable build, last successful build, and last completed build, all of which are 21 minutes ago. A 'Build History' table is visible on the left, showing builds #1 through #4 with their respective timestamps.

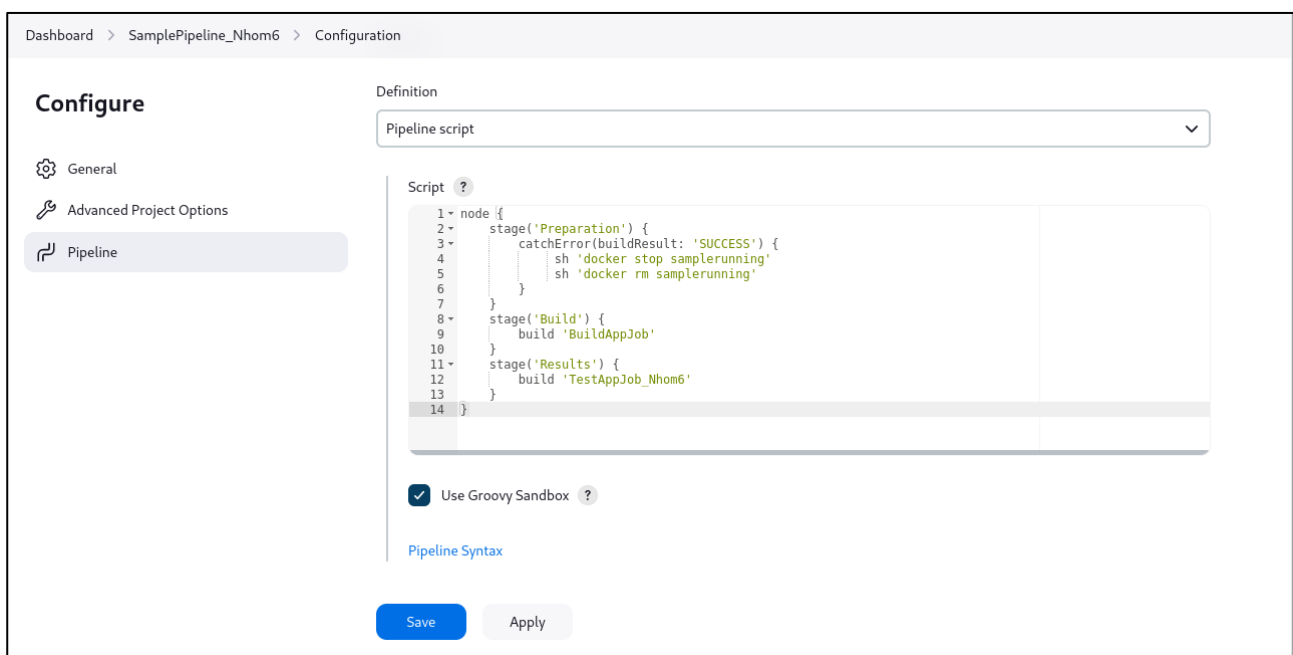
Build Number	Timestamp
#4	Oct 3, 2024, 2:21 PM
#3	Oct 3, 2024, 2:20 PM
#2	Oct 3, 2024, 2:15 PM
#1	Oct 3, 2024, 2:05 PM

6. Sinh viên hoàn thành pipeline của 2 ứng dụng Buid và Test theo gợi ý bên dưới, trình bày step-by-step có minh chứng.

- **Bước 1:** Tạo job Pipeline với tên và type như hình bên dưới:



- **Bước 2:** Thực hiện cấu hình job SamplePineline tại tab **Pipeline** và thực hiện điền script vào **Pipeline script**.



Giải thích đoạn code:

+ **Node** là một khối biểu thị môi trường, bên trong nó các câu lệnh sẽ được thực thi trên node được chỉ định.

+ **stage('Preparation')** là bước đầu tiên trong khối được đặt tên là *Preparation*, nó thực hiện bắt lỗi thông qua câu lệnh tiếp theo:

+ **catchError(buildResult: 'SUCCESS')** thực hiện bắt các lỗi, nếu có xảy ra mỗi thì kết quả vẫn là SUCCESS.

+ **sh 'docker stop samplerunning'** là lệnh dùng để dừng container có tên là samplerunning.

+ **sh 'docker rm samplerunning'** dùng để xóa docker container có tên là samplerunning.

+ **stage('Build')** là giai đoạn thứ hai dùng để thực hiện việc build ứng dụng.

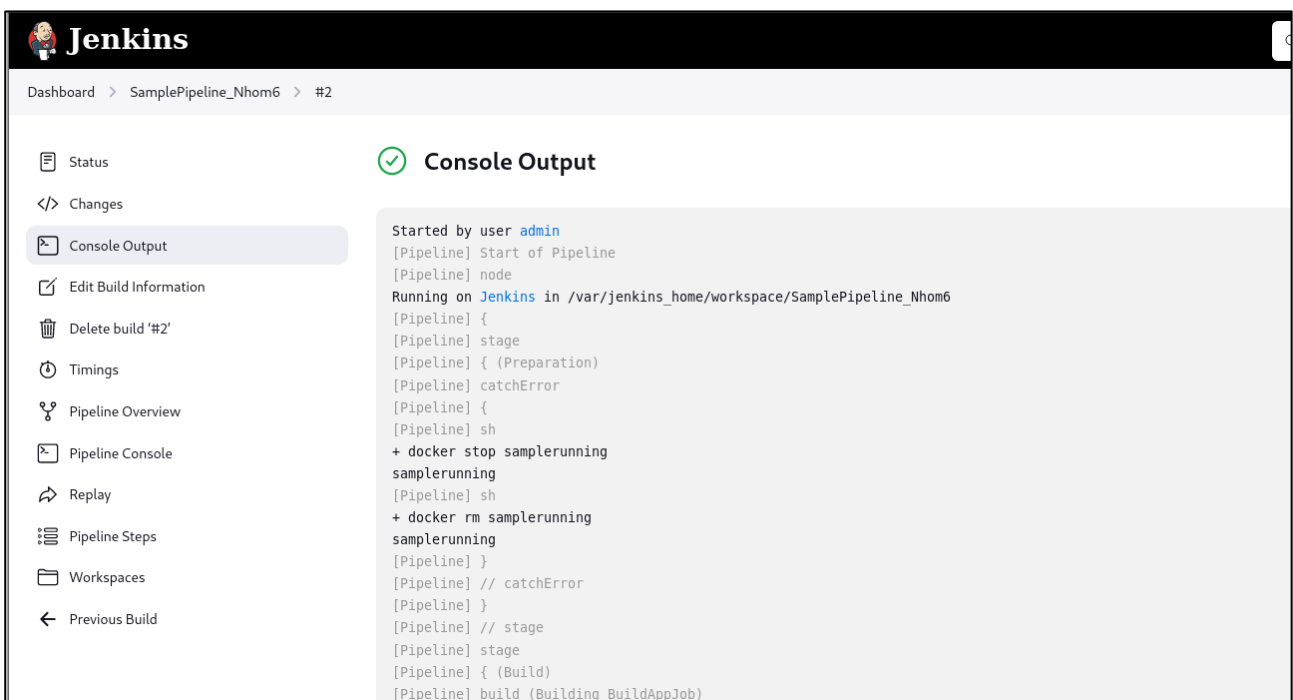
+ **build 'BuildAppJob'**: Câu lệnh này gọi job **BuildAppJob** để thực thi. Pipeline sẽ đợi job này trước khi chuyển sang giai đoạn khác

+ **stage('Results')** là giai đoạn thứ ba để thực hiện việc kiểm tra kết quả.

+ **build 'TestAppJob'** gọi job **TestAppJob**.

Tóm lại: Pipeline này thể hiện một quy trình CI/CD đơn giản với ba bước: chuẩn bị, build ứng dụng và kiểm tra kết quả build.

- **Bước 3:** Build SamplePipeline_Nhom6 ta được kết quả



```

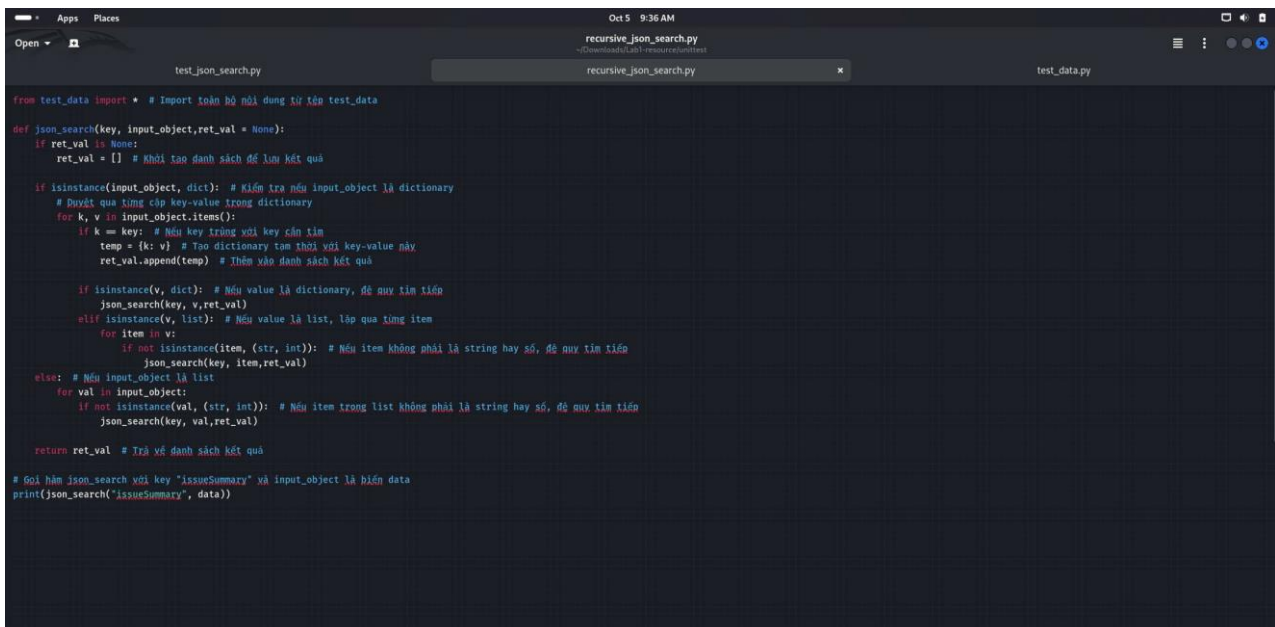
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/SamplePipeline_Nhom6
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Preparation)
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh
+ docker stop samplerunning
samplerunning
[Pipeline] sh
+ docker rm samplerunning
samplerunning
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] build (Building BuildAppJob)
  
```


7. Sinh viên sửa lại lỗi trên bằng cách di chuyển `ret_val=[]` và thực thi lại `recursive_json_search.py`, trình bày step-by-step có minh chứng

8. Sinh viên hãy tìm ra nguyên nhân, tìm cách sửa lỗi trên và thực thi lại `unittest`, trình bày step-by-step có minh chứng.

- Ở bài 7 và bài 8, ta có thể sửa lỗi của cả 2 yêu cầu bằng cách thiết lập lại cách hoạt động của hàm đệ quy và vị trí của `ret_val[]` như sau:

- Thiết lập lại tham số của hàm `json_search` thành `(key,input_object,ret_val=None)`
 ⇒ Thêm tham số `ret_val` sẽ giúp ta có `ret_val` chung trong suốt quá trình đệ quy, Ngoài ra khởi tạo giá trị của nó là `None` để khi lần đầu gọi hàm mà `ret_val` chưa được truyền vào thì `ret_val` sẽ trở thành 1 danh sách trống
- Mỗi khi có lời gọi đệ quy trong chương trình, ta sẽ truyền `ret_val` vào vốn sẽ là `ret_val` chung và duy nhất của chương trình để lưu lại các kết quả vào thay vì mỗi lần đệ quy là sẽ khởi tạo 1 biến `ret_val` mới rồi lưu kết quả vào `ret_val` đó như trong chương trình cũ
- Đoạn code sau khi sửa:



```

from test_data import * # Import toàn bộ nội dung từ tệp test_data

def json_search(key, input_object, ret_val=None):
    if ret_val is None:
        ret_val = [] # Khởi tạo danh sách để lưu kết quả

    if isinstance(input_object, dict): # Kiểm tra nếu input_object là dictionary
        # Duyệt qua từng cặp key-value trong dictionary
        for k, v in input_object.items():
            if k == key: # Nếu key trùng với key cần tìm
                temp = {k: v} # Tạo dictionary tạm thời với key-value này
                ret_val.append(temp) # Thêm vào danh sách kết quả

            if isinstance(v, dict): # Nếu value là dictionary, đệ quy tìm tiếp
                json_search(key, v, ret_val)
            elif isinstance(v, list): # Nếu value là list, lặp qua từng item
                for item in v:
                    if not isinstance(item, (str, int)): # Nếu item không phải là string hay số, đệ quy tìm tiếp
                        json_search(key, item, ret_val)
    else: # Nếu input_object là list
        for val in input_object:
            if not isinstance(val, (str, int)): # Nếu item trong list không phải là string hay số, đệ quy tìm tiếp
                json_search(key, val, ret_val)

    return ret_val # Trả về danh sách kết quả

# Gọi hàm json_search với key "issueSummary" và input_object là biến data
print(json_search("issueSummary", data))
  
```

- Kết quả chương trình:

```

kali@kali: ~/Downloads/Lab1-resource/unittest
$ python3 -m unittest test_json_search
[{"IssueSummary": "Network Device 10.10.20.82 Is Unreachable From Controller"}]
test_is_a_list (test_json_search.json_search_test.test_is_a_list)
Should return a list ... ok
test_search_found (test_json_search.json_search_test.test_search_found)
Key should be found, return list should not be empty ... ok
test_search_not_found (test_json_search.json_search_test.test_search_not_found)
Key should not be found, should return an empty list ... ok
Ran 3 tests in 0.000s
OK
kali@kali: ~/Downloads/Lab1-resource/unittest
$ python3 -m unittest test_json_search
[{"IssueSummary": "Network Device 10.10.20.82 Is Unreachable From Controller"}]
...
Ran 3 tests in 0.000s
OK
kali@kali: ~/Downloads/Lab1-resource/unittest
$ 

```

HẾT
