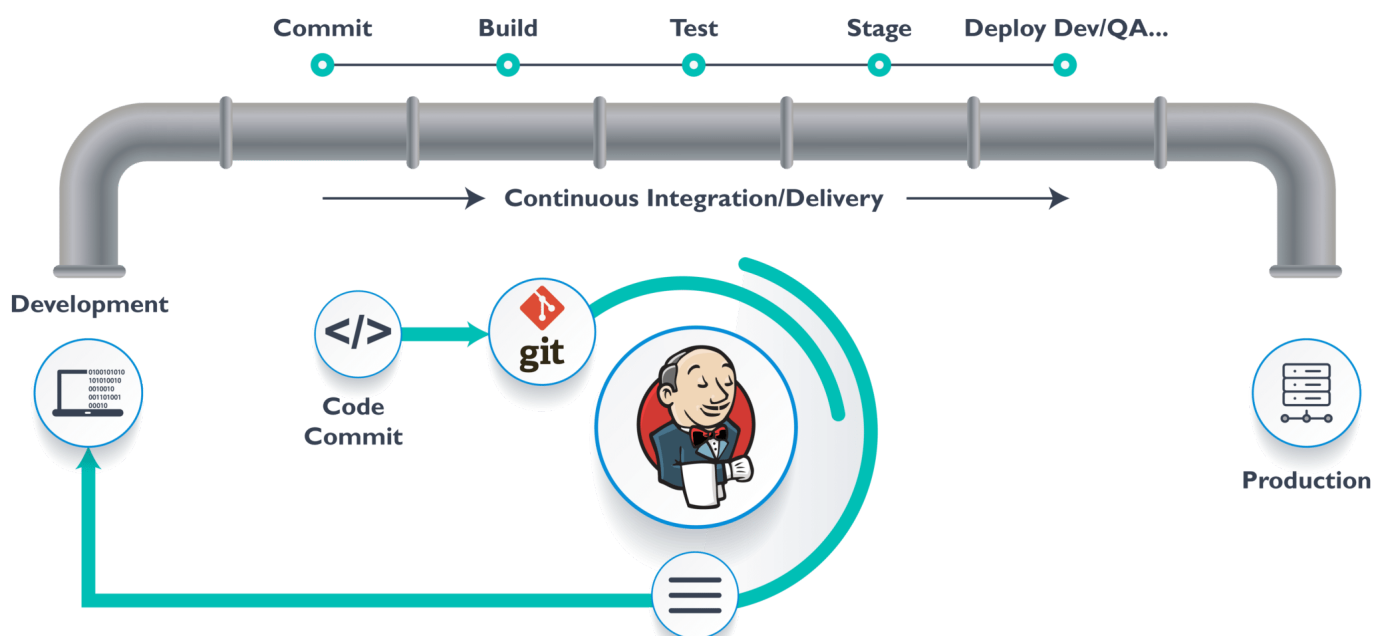


Tìm hiểu quy trình Continuous Integration với Jenkins

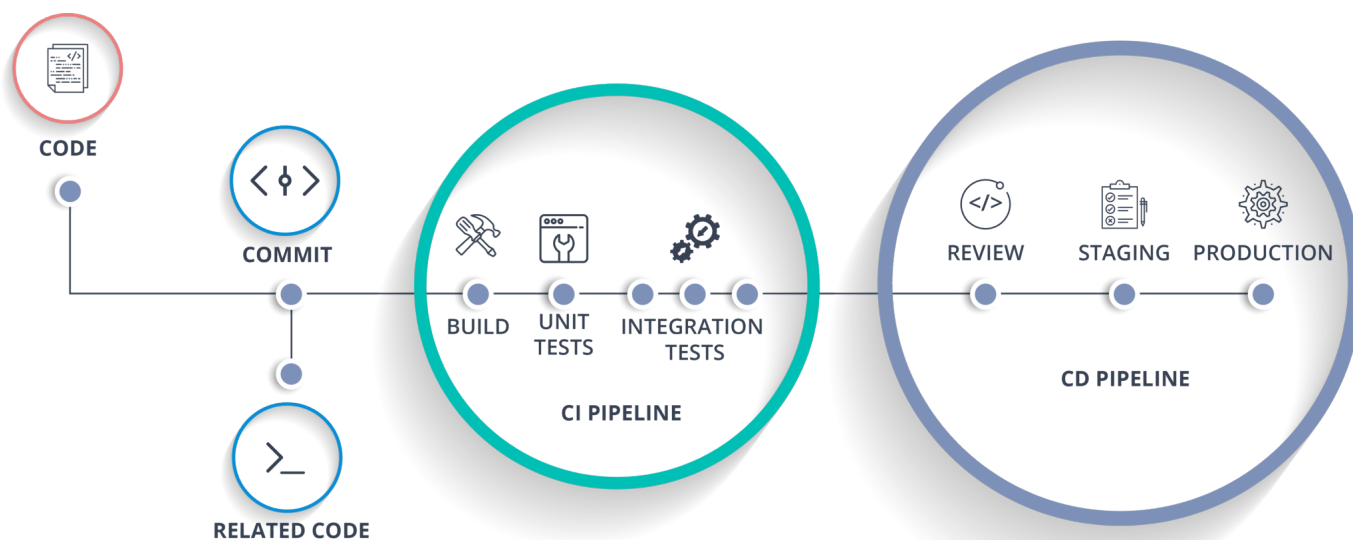
1. Jenkins là gì ?

1.1. Khái niệm

Jenkins là một opensource dùng để thực hiện chức năng tích hợp liên tục (gọi là **CI – Continuous Integration**) và xây dựng các tác vụ tự động hóa. Nó tích hợp các source code của các members trong team lại nhanh chóng một cách liên tục, theo dõi sự thực thi và trạng thái thông qua các bước kiểm thử (**Integration test, units test**). Tất nhiên là nhằm giúp sản phẩm chạy ổn định.



Quy trình CICD của Jenkins:



1.2. Chu trình làm việc

1. Bước đầu tiên, các thành viên trong team dev sẽ bắt đầu pull code mới nhất từ repo về branch để thực hiện các yêu cầu chức năng nhất định.

2. Tiếp đó là quá trình lập trình và test code để đảm bảo chất lượng của chức năng cũng như toàn bộ source code.
3. Thành viên code xong thì sẵn sàng cho việc commit vào branch develop của team.
4. Thành viên cập nhật code mới từ repo về local repo
5. Merge code và giải quyết conflict.
6. Build và đảm bảo code pass qua các tests dưới local.
7. Commit code lên repo
8. Máy chủ CI lắng nghe các thay đổi code từ repository và có thể tự động build/test, sau đó đưa ra các thông báo (pass/failure) cho các thành viên.

2. Cài đặt Jenkins

2.1. Chuẩn bị

- Chuẩn bị máy ảo chạy hệ điều hành Windows hoặc Linux cấu hình tối thiểu: 2 CPU, 2GB RAM, >= 50GB DISK.
- Cài đặt trước docker trên máy.

2.2. Cài đặt

Bước 1: Chỉnh sửa cấu hình network docker để có thể chạy docker trong docker (trong quá trình build Jenkins sẽ gọi đến docker của host thông qua kết nối socket để tạo docker phục vụ build code). Chỉnh sửa nội dung file `/lib/systemd/system/docker.service`. Chỉnh sửa nội dung như cấu hình sau đây:

```
ExecStart=/usr/bin/dockerd -H unix:///var/run/docker.sock -H
tcp://172.16.87.131 --containerd=/run/containerd/containerd.sock
```

với giao thức TCP sẽ chọn địa chỉ IP là IP của máy **Host**.

Bước 2: Tạo user Jenkins (tránh việc chạy Jenkins bằng Root).

```
# create new user
$ useradd -m -d /home/Jenkins -s /bin/bash Jenkins
# add Jenkins user to docker group
$ usermod -aG docker Jenkins

$ su - Jenkins
```

Bước 3: Chạy Jenkins bằng docker.

- Tạo thư mục backup dữ liệu cho Jenkins.

```
$ mkdir data
```

- Lấy UserId của user jenkins:

```
$ id
uid=1001(jenkins) gid=1001(jenkins) groups=1001(jenkins),999(docker)
```

- Khởi tạo docker container.

```
$ docker run -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):$(which docker) -v `PWD`/data:/var/jenkins_home -p 8080:8080 --user 1001:999 --name jenkins-server -d docker pull jenkins/jenkins
```

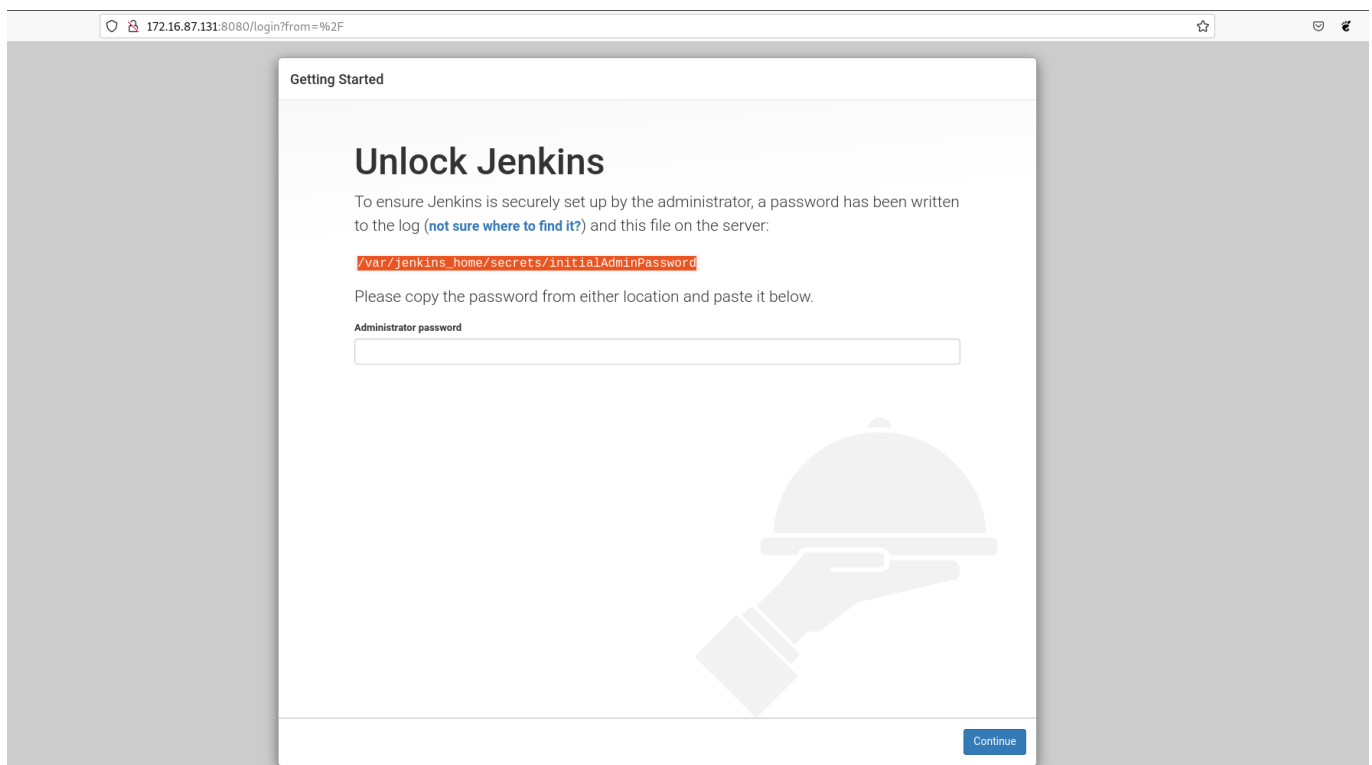
```
jenkins@ip-172-31-17-212:~$ docker container ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
4a530855b0de   jenkins/jenkins:lts   "/sbin/tini -- /usr/_..."   14 hours ago   Up 29 seconds   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   jenkins-server
```

- Sau khi chạy ổn định truy cập vào <http://localhost:8080> và lấy password đăng nhập tại file /var/jenkins_home/secrets/initialAdminPassword.

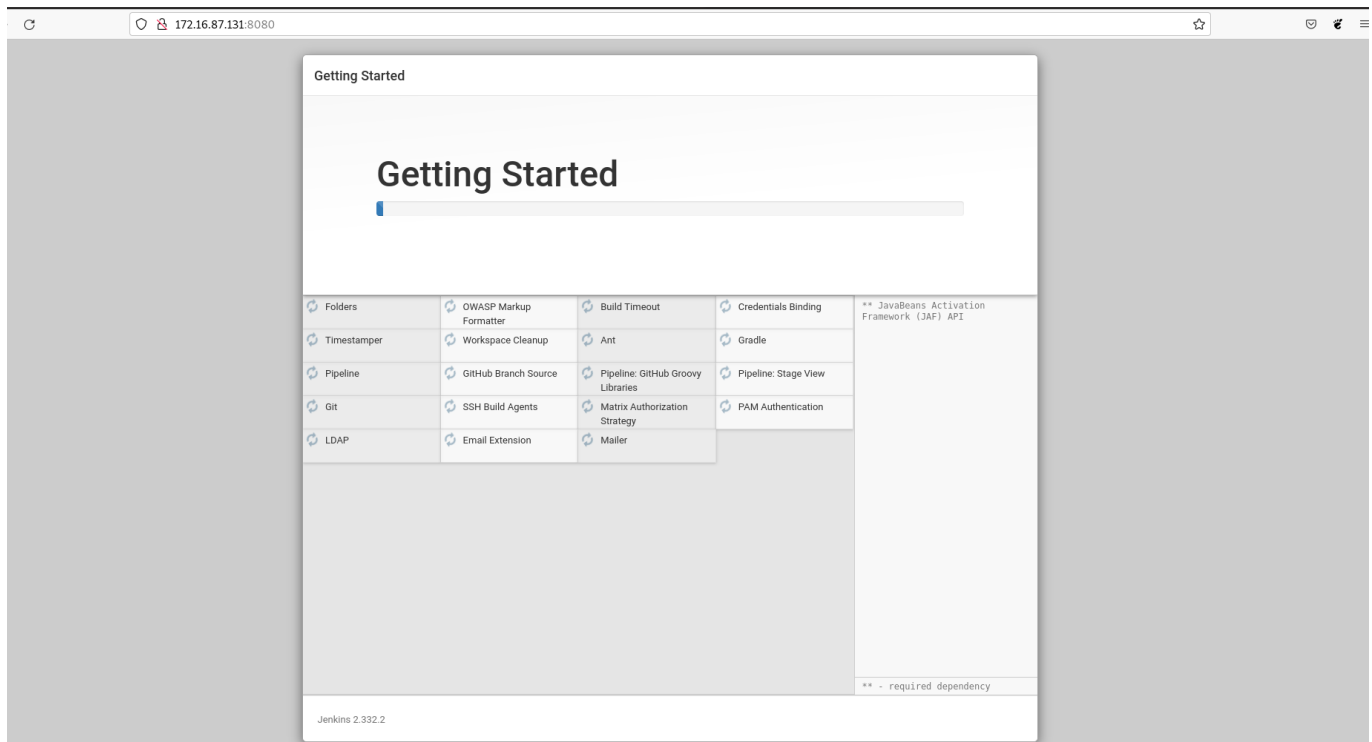
3. Hướng dẫn sử dụng

3.1. Cài đặt các plug-in

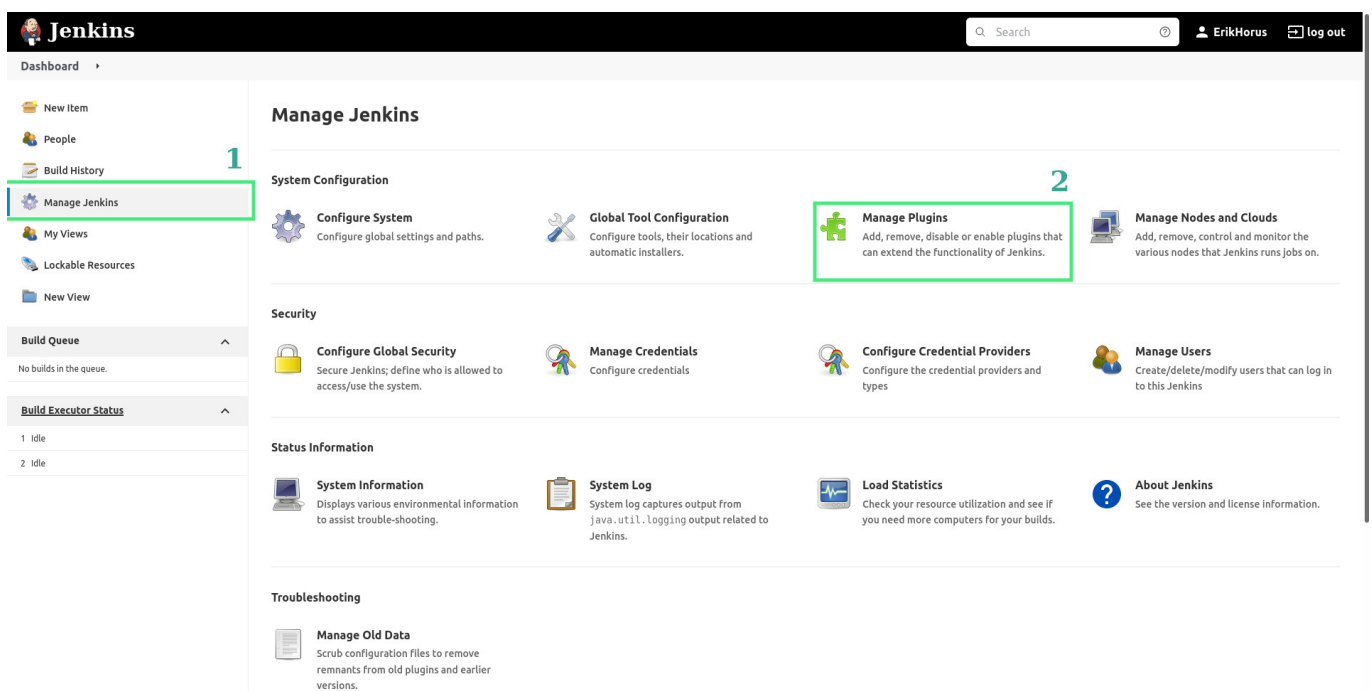
Bước 1: Nhập password đã lấy được vào Unlock Jenkins.



Bước 2: Chọn cài đặt các plug-in theo đề xuất của Jenkins.



Bước 3: Cài đặt các Plug-in quan trọng.



- Tại giao diện dashboard của Jenkins chọn mục **Manage Jenkins** sau đó chọn tab **Manage plugins** -> chọn **Available** tag -> chọn **Search** và tìm với các keyword sau :
 - Docker plugin
 - Pipeline
 - github intergration
 - Github pull request builder
- Check vào các ô trước tên các plugin trên -> chọn **Install without restart**.

Jenkins Dashboard > Plugin Manager

Back to Dashboard Manage Jenkins

Plugin Manager

Updates **Available** Installed Advanced

Install	Name	Released
<input type="checkbox"/>	Javadoc 217.v905b_86277a_2a This plugin adds Javadoc support to Jenkins.	2 mo 13 days ago
<input type="checkbox"/>	Maven Integration 3.18 Build Tools This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (Junit, ...).	1 mo 23 days ago
<input type="checkbox"/>	WMI Windows Agents 1.8 Agent Management windows Allows you to setup agents on Windows machines over Windows Management Instrumentation (WMI) This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	11 mo ago
<input type="checkbox"/>	MapDB API 1.0.9.0 Library plugins (for use by other plugins) This plugin provides a shared dependency on the MapDB library so that other plugins can co-operate when using this library.	5 yr 11 mo ago
<input type="checkbox"/>	Subversion 2.15.4 Source Code Management The Jenkins Plugins Parent POM Project	15 days ago
<input type="checkbox"/>	External Monitor Job Type 191.v363d0d1efdf8 External Site/Tool Integrations Miscellaneous	2 mo 27 days ago

Install without restart Download now and install after restart Update information obtained: 14 hr ago Check now

- Các plug-in đang được cài đặt.

Jenkins Dashboard > Update Center

Back to Dashboard Manage Jenkins Manage Plugins

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Plugin	Status
Authentication Tokens API	Success
Docker Commons	Success
Docker API	Success
Docker	Pending
Docker Pipeline	Pending
JavaScript GUI Lib: JQuery bundles (jQuery and JQuery UI)	Pending
GitHub Integration	Pending
Loading plugin extensions	Pending

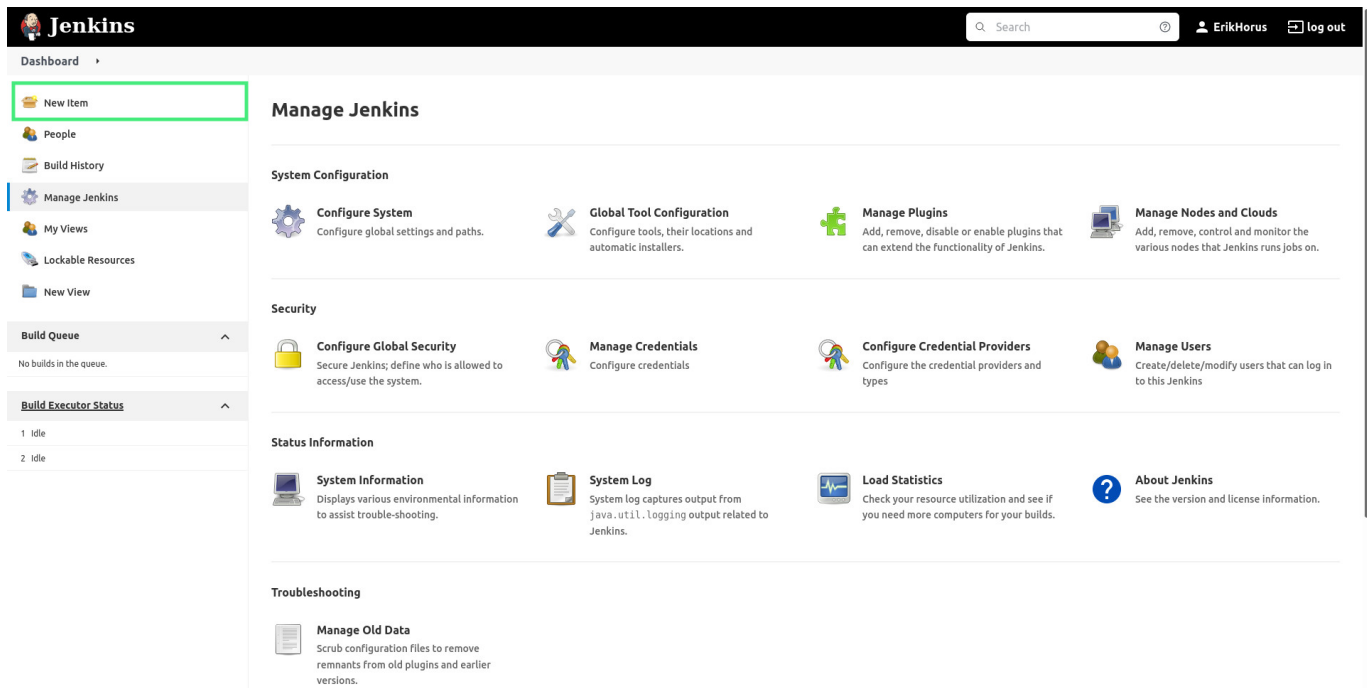
[Go back to the top page](#)
(you can start using the installed plugins right away)

☐ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.332.2

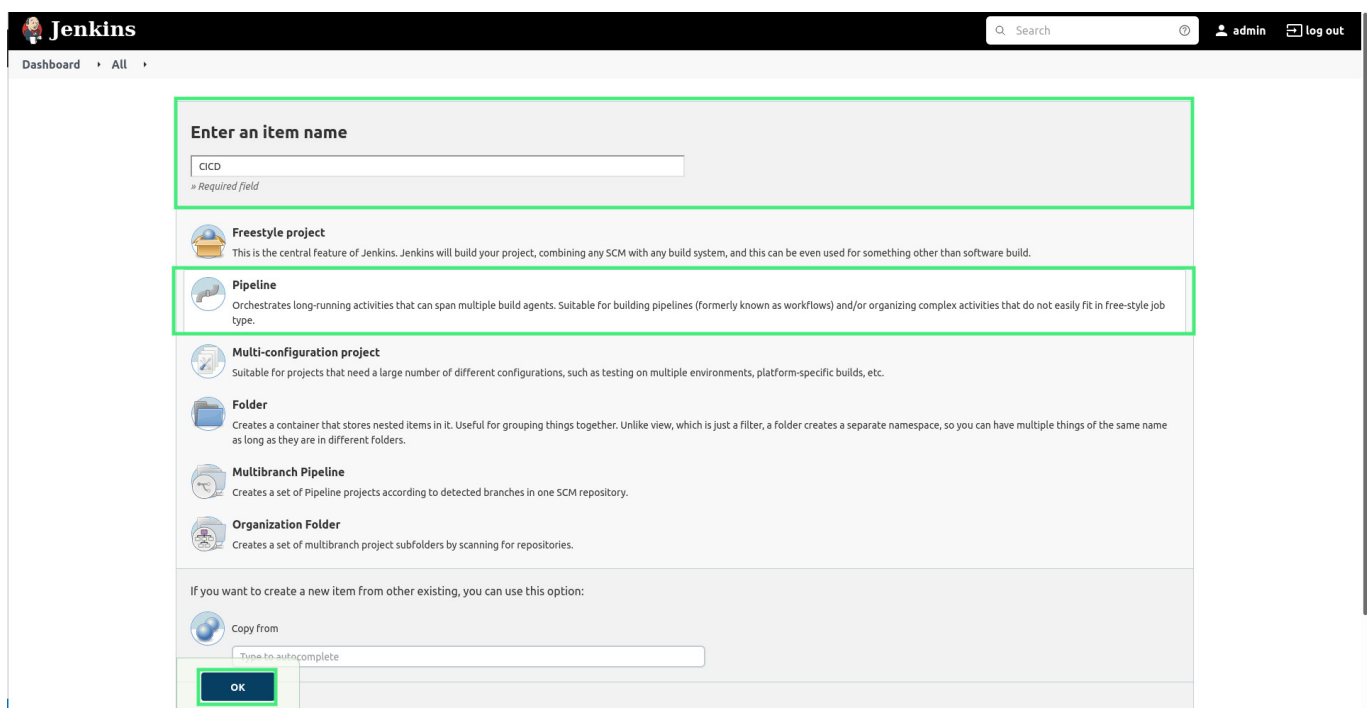
Bước 4: Tạo Job mới và cấu hình github.

- Chọn **New item**.



The screenshot shows the Jenkins Dashboard. On the left sidebar, the 'New Item' button is highlighted with a green box. The main area is titled 'Manage Jenkins' and contains several sections: 'System Configuration' (Configure System, Global Tool Configuration, Manage Plugins, Manage Nodes and Clouds), 'Security' (Configure Global Security, Manage Credentials, Configure Credential Providers, Manage Users), 'Status Information' (System Information, System Log, Load Statistics, About Jenkins), and 'Troubleshooting' (Manage Old Data).

- Đặt tên cho Job và chọn **Pipeline** -> Chọn **Ok**.



The screenshot shows the 'Enter an item name' dialog in Jenkins. The 'Name' field is filled with 'CICD'. Below the field, there is a list of item types: 'Freestyle project', 'Pipeline' (highlighted with a green box), 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. At the bottom, there is a 'Copy from' field and an 'OK' button, which is also highlighted with a green box.

- Hoàn tất tạo Job.

Dashboard

New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, New View

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle

S	W	Name	Last Success	Last Failure	Last Duration
✓	🐳	FastAPI-Docker	13 hr #2	N/A	1 min 27 sec

Icon: S M L

Icon legend, Atom feed for all, Atom feed for failures, Atom feed for just latest builds

REST API Jenkins 2.332.2

Bước 5: Tạo Github Personal Access Token cho Jenkins. Chú ý thêm đầy đủ các Role như trong hình. Sau đó copy Token để sử dụng.

- Tham khảo hướng dẫn tạo GPAT tại đây: [Github](#).

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Jenkins — *admin:repo_hook, notifications, read:user, repo:status, user:email, write:discussion*

Last used within the last week

Delete

Expires on Sun, May 22 2022.

Bước 6: Cấu hình kết nối **Github**.

- Tại **Manage Jenkins** -> **Configure System**.

Jenkins

Dashboard

Manage Jenkins

System Configuration

- Configure System**
Configure global settings and paths.
- Global Tool Configuration**
Configure tools, their locations and automatic installers.
- Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials**
Configure credentials
- Configure Credential Providers**
Configure the credential providers and types
- Manage Users**
Create/delete/modify users that can log in to this Jenkins

Status Information

- System Information**
Displays various environmental information to assist trouble-shooting.
- System Log**
System log captures output from java.util.logging output related to Jenkins.
- Load Statistics**
Check your resource utilization and see if you need more computers for your builds.
- About Jenkins**
See the version and license information.

Troubleshooting

- Manage Old Data**
Scrub configuration files to remove remnants from old plugins and earlier versions.

- Tại cấu hình Github thêm **Name** cho github server. VD: **Github-server**.
- Tại mục **Credentials** -> chọn **Add** -> chọn **Jenkins**.

Dashboard > configuration

Registry credentials

- none - Add

GitHub

GitHub Servers ?

GitHub Server ?

Name ?

Github E

API URL ?

https://api.github.com

Credentials ?

- none - Add

Jenkins

Test connection

Advanced...

Delete

Add GitHub Server

Save Apply

- Thêm các thông tin sau vào Form:
 - Kind:** Secret text
 - Secret:** paste Token tạo tại bước 5
 - ID:** Github-secretext

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' dialog box. It has a title bar with a home icon and the text 'Jenkins Credentials Provider: Jenkins'. Below the title bar is a section titled 'Add Credentials'. The form contains the following fields: 'Domain' (a dropdown menu with 'Global credentials (unrestricted)' selected), 'Kind' (a dropdown menu with 'Secret text' selected), 'Scope' (a dropdown menu with 'Global (Jenkins, nodes, items, all child items, etc)' selected), 'Secret' (a text input field with masked characters), 'ID' (a text input field), and 'Description' (a text input field). At the bottom of the dialog are 'Add' and 'Cancel' buttons. Below the dialog, in the background, are 'Add GitHub Server', 'Save', and 'Apply' buttons.

- Sau khi tạo xong thì thêm vào mục bên trái và chọn **test connection** để kiểm tra.

The screenshot shows the Jenkins configuration page for GitHub. It has a title bar with 'Dashboard' and 'configuration'. Below the title bar is a dropdown menu with '- none -' and an 'Add' button. The main section is titled 'GitHub'. It contains a 'GitHub Servers' section with a table of servers. The first server is 'Github-server' with 'API URL' 'https://api.github.com'. Below the table is a 'Credentials' dropdown menu with 'Github-secrettext' selected and an 'Add' button. Below the credentials is a message 'Credentials verified for user ErikHorus1249, rate limit: 4999'. There is a 'Test connection' button. Below that is a 'Manage hooks' checkbox which is checked. At the bottom of the 'GitHub Servers' section is an 'Add GitHub Server' button. Below the 'Manage hooks' section is a 'GitHub API usage' section with a 'Save' button and an 'Apply' button. On the right side of the page, there are 'Advanced...' buttons for the 'Test connection' and 'Manage hooks' sections, and a 'Delete' button for the 'Test connection' section.

- Tiếp tục thêm Webhook cho Github phục vụ cho việc commit code thì Jenkins sẽ tự động build -> chọn **advanced** -> **Re-register hooks for all jobs**.

Dashboard » configuration

Credentials ?

ErikHorus1249_pat Add

Test connection

☒ Manage hooks

Advanced...

Delete

Add GitHub Server

Called re-register hooks for 1 items

Re-register hooks for all Jobs

Override Hook URL

☐ Specify another hook URL for GitHub configuration

Shared secrets

Add shared secret

Additional actions ?

Manage additional GitHub actions

GitHub API usage

GitHub API usage rate limiting strategy ?

Normalize API requests

Save Apply

Bước 7: Cấu hình Github pull request builder.

- Tại Github pull request builder -> điền thông tin repo (chú ý chỉ điền github-user/name-of-repo).

Dashboard » configuration

GitHub Pull Request Builder

GitHub Auth

GitHub Server API URL ?

https://api.github.com

Jenkins URL override ?

Shared secret ?

Credentials ?

ErikHorus1249/***** Add

☐ Test basic connection to GitHub

Repository owner/name

https://github.com/ErikHorus1249/CICD

☐ Test Permissions to a Repository

☐ Test adding comment to Pull Request

☐ Test updating commit status

Create API Token...

Description ?

Anonymous connection

Save Apply

pul ^ v Highlight All Match Case Match Diacritics Whole Words 1 of 3 matches X

- Chọn Credentials như khi làm ở bước cấu hình github.

Dashboard > configuration

Delete Server

Add

- ☒ Auto-manage webhooks
- ☒ Use comments to report results when updating commit status fails ?
- ☐ Use comments to report intermediate phases: triggered et al

Admin list

ErikHorus1249

Advanced...

Application Setup

Update commit status during build

Commit Status Context ?

Jenkins Test & build

Commit Status URL ?

Commit Status Build Triggered ?

Save Apply

pul ^ v Highlight All Match Case Match Diacritics Whole Words 1 of 3 matches X

- Chọn 3 mục như hình -> thêm tên của admin (chính là github-account) -> thêm thông điệp trong context -> Chọn **Ok** để lưu.
- Sau khi cấu hình hoàn tất mỗi khi code mới được push lên Jenkins sẽ thực hiện quá trình build phiên bản mới.

Bước 8: Tạo Personal Access Token cho **Dockerhub**.

- Tạo tài khoản tại [Hub.Docker.com](https://hub.docker.com) nếu chưa có tài khoản.
- Lưu lại thông tin username/password sử dụng khi tạo Access Token.

dockerhub Search for great content (e.g., mysql) Explore Repositories Organizations Help Upgrade erikhorus1249

erikhorus1249 Edit profile

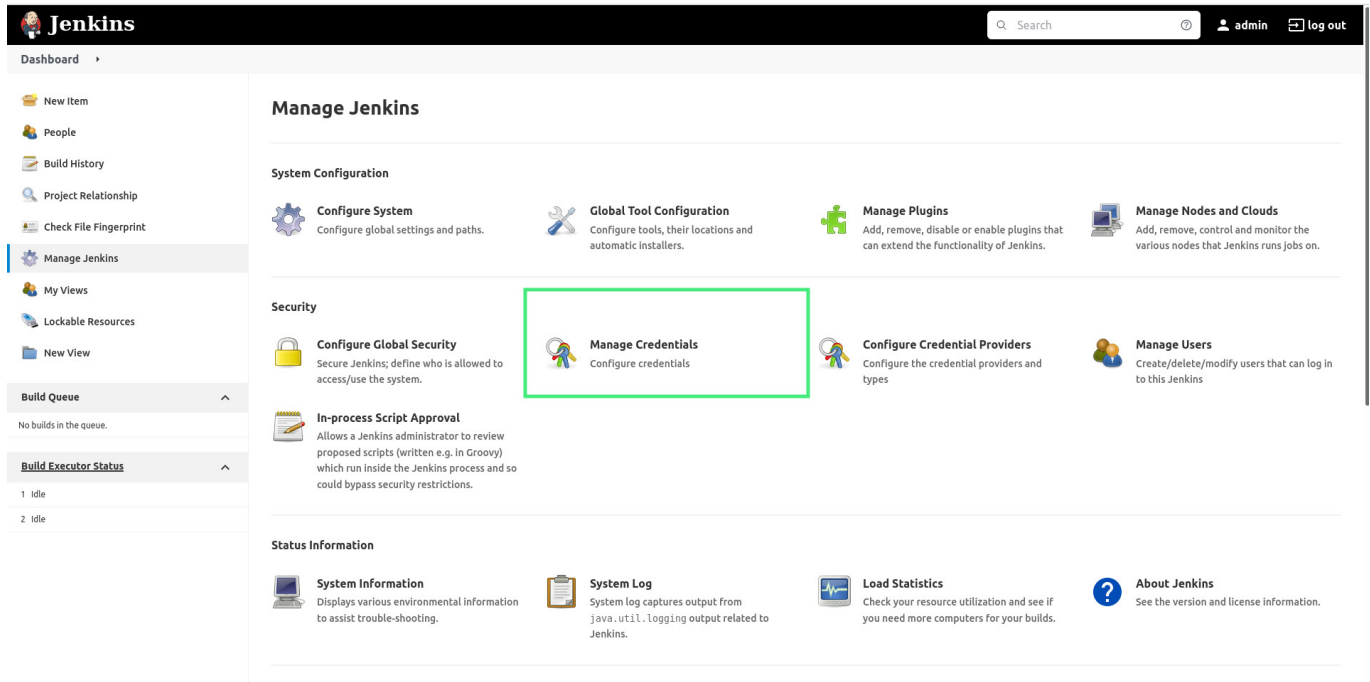
Community User joined January 19, 2021

Repositories Starred Contributed

Displaying 2 of 2 repositories

erikhorus1249/fastapi-docker	0 Stars
By erikhorus1249 • Updated 13 hours ago	
Container	
erikhorus1249/elasticsearch_cr_7.17.1	6 Downloads 0 Stars
By erikhorus1249 • Updated a month ago	
Cracked elasticsearch	
Container	

- Chọn **Manage Jenkins** -> **Manage Credentials**.



The screenshot shows the Jenkins Dashboard. The left sidebar contains navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (highlighted), My Views, Lockable Resources, New View, Build Queue, and Build Executor Status. The main content area is titled 'Manage Jenkins' and is divided into three sections: System Configuration, Security, and Status Information. In the Security section, the 'Manage Credentials' link is highlighted with a green box. Below this, the 'Stores scoped to Jenkins' section shows a table with one entry: Jenkins (global).

Manage Jenkins

System Configuration

- Configure System**: Configure global settings and paths.
- Global Tool Configuration**: Configure tools, their locations and automatic installers.
- Manage Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Manage Nodes and Clouds**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Configure Global Security**: Secure Jenkins; define who is allowed to access/use the system.
- Manage Credentials**: Configure credentials (highlighted with a green box).
- Configure Credential Providers**: Configure the credential providers and types.
- Manage Users**: Create/delete/modify users that can log in to this Jenkins.

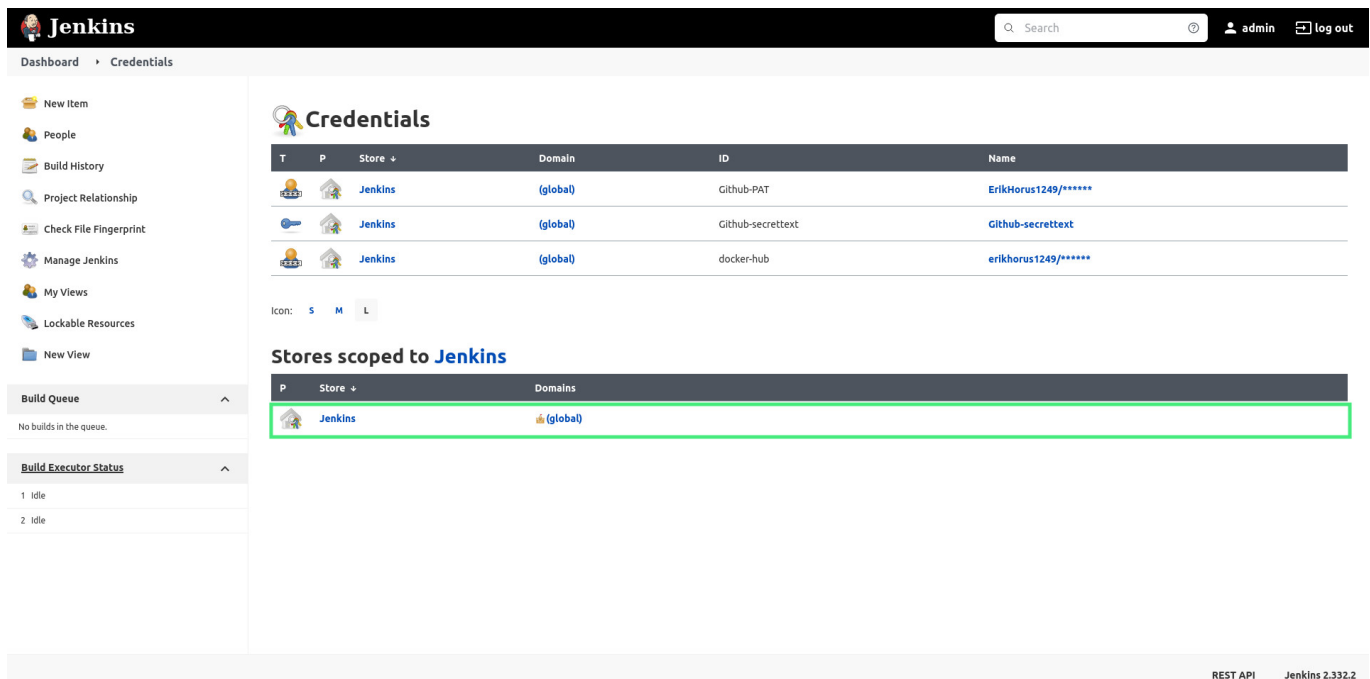
In-process Script Approval

Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions.

Status Information

- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from java.util.logging output related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- About Jenkins**: See the version and license information.

- Chọn như hình sau.



The screenshot shows the Jenkins Credentials page. The left sidebar is the same as the dashboard. The main content area is titled 'Credentials' and shows a table of credentials. Below the table, there is a section 'Stores scoped to Jenkins' which shows a table with one entry: Jenkins (global). This entry is highlighted with a green box.

Credentials

T	P	Store +	Domain	ID	Name
		Jenkins	(global)	Github-PAT	ErikHorus1249/*****
		Jenkins	(global)	Github-secrettext	Github-secrettext
		Jenkins	(global)	docker-hub	erikhorus1249/*****

Icon: S M L

Stores scoped to Jenkins

P	Store +	Domains
	Jenkins	(global)

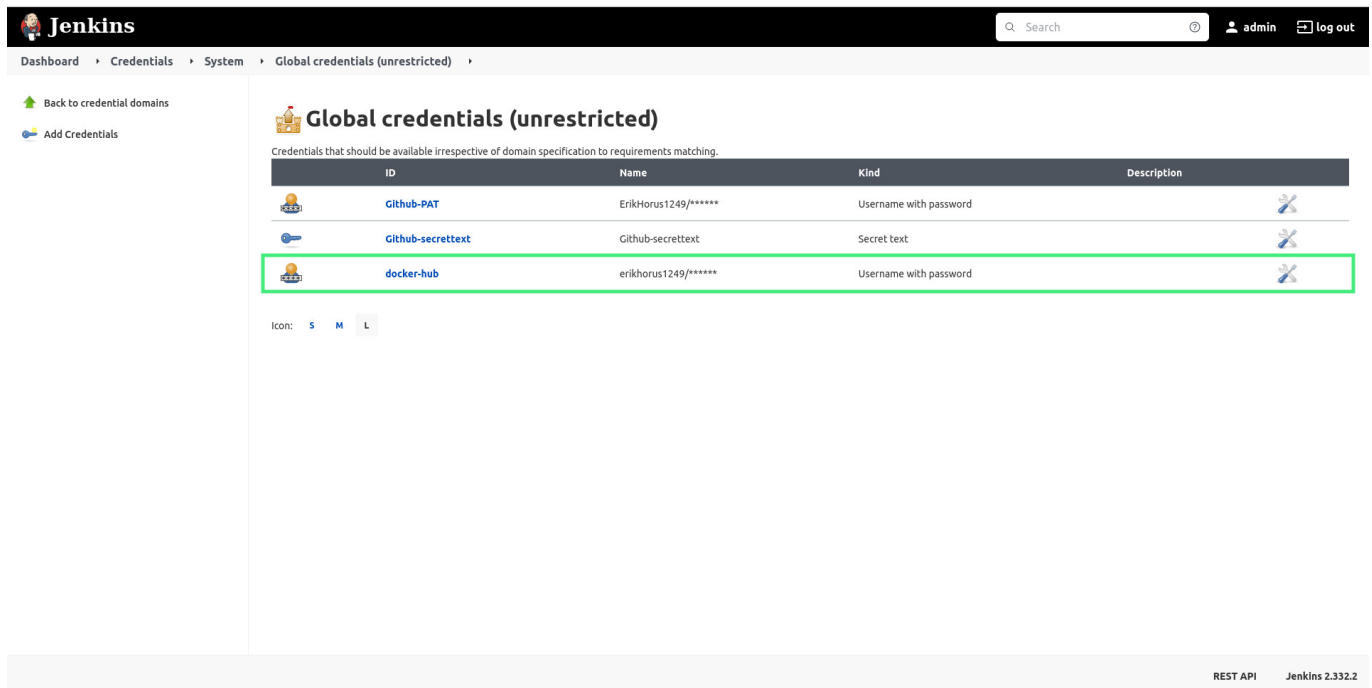
- Chọn **Global credentials**.

The screenshot shows the Jenkins 'System' page. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area is titled 'System' and contains a table with two columns: 'Domain' and 'Description'. A single row is highlighted with a green border, showing 'Global credentials (unrestricted)' in the Domain column and 'Credentials that should be available irrespective of domain specification to requirements matching.' in the Description column. Below the table, there are icons for 'S', 'M', and 'L'. At the bottom right, it says 'REST API' and 'Jenkins 2.332.2'.

- Chọn **Add credentials**
- Điền các thông tin cần thiết theo mẫu sau:
 - **Username:** <dockerhub-username>
 - **Password:** <dockerhub-password>
 - **ID:** docker-hub
 - **Descriptions:** Tùy chọn khi nhập
- Sau khi điền chọn **Ok** để hoàn tất.

The screenshot shows the Jenkins 'Add Credentials' form. On the left sidebar, there is a link 'Back to credential domains' and a link 'Add Credentials' which is highlighted with a green border. The main content area is titled 'Global credentials (unrestricted)' and contains a form with the following fields: 'Kind' (Username with password), 'Scope' (Global (Jenkins, nodes, items, all child items, etc)), 'Username' (erikhorus1249), 'Treat username as secret' (checkbox), 'Password' (masked with dots), 'ID' (docker-hub), and 'Description' (add dockerhub PAT). An 'OK' button is at the bottom. At the bottom right, it says 'REST API' and 'Jenkins 2.332.2'.

- Sau khi tạo **Credential**.



Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Github-PAT	ErikHorus1249/*****	Username with password	
Github-secrettext	Github-secrettext	Secret text	
docker-hub	erikhorus1249/*****	Username with password	

Icon: S M L

REST API Jenkins 2.332.2

3.2. Chuẩn bị source code dự án

3.2.1. Lưu ý về source code

- Dự án sử dụng trong demo là một project FastAPI triển khai một số API đã có Dockerfile và docker-compose.yml.
- Dự án sử dụng poetry để quản lý dependency và package. Chi tiết sử dụng poetry tham khảo tại : [Python-poetry](#).
- Quan trọng nhất là file pipeline : **Jenkinsfile** chứa thông tin test và build của dự án. Tham khảo chi tiết về Jenkinsfile tại: [Github/ErikHorus1249](#).

```

app
├── __init__.py
├── main.py
└── tests
    ├── __init__.py
    ├── test_server.py
    ├── .gitignore
    ├── Dockerfile
    ├── Jenkinsfile
    ├── README.md
    ├── docker-compose.yml
    ├── poetry.lock
    └── pyproject.toml
  
```

- Đường dẫn tham khảo source code dự án: [Github](#)

3.2.1. Kiểm tra webhook đã được thêm vào Github repository chưa.

- Tại chọn **Settings**.

Search or jump to... Pull requests Issues Marketplace Explore

ErikHorus1249 / CICD (Public)

<> Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

develop 2 branches 0 tags

This branch is 19 commits ahead of master.

ErikHorus1249 update 0eab84a 14 hours ago 21 commits

File	Commit Message	Time
Documents	update	14 hours ago
app	create develop branch	2 days ago
tests	update	yesterday
.gitignore	cannot connect to server!	7 days ago
Dockerfile	edit jenkinsfile	yesterday
Jenkinsfile	edit jenkinsfile	yesterday
README.md	create develop branch	2 days ago

About: No description, website, or topics provided.

Releases: No releases published. [Create a new release](#)

Packages: No packages published

- Chọn **Webhooks** và kiểm tra xem đã Webhook được thêm vào chưa.
- Nếu chưa có thực hiện lại phần cấu Github **Bước 6**.

Search or jump to... Pull requests Issues Marketplace Explore

ErikHorus1249 / CICD (Public)

<> Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

Security

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

URL	Events	Status
http://ec2-18-140-68-161.ap-south... (push)	push	✓

Edit Delete

3.3. Build dự án

Bước 1: Chọn vào tên của Job -> chọn Build now.

The screenshot shows the Jenkins Dashboard. On the left sidebar, there are various navigation options. The main area displays a table of jobs. The job 'FastAPI-Docker' is highlighted with a green border. Below the table, there are links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	⚙️	FastAPI-Docker	13 hr #3	N/A	5 min 14 sec

- Chọn vào mục Build History để xem thông tin build chi tiết qua console.

The screenshot shows the Jenkins Pipeline FastAPI-Docker page. On the left sidebar, the 'Build Now' button is highlighted in green. The main area displays the 'Pipeline FastAPI-Docker' page. Below the 'Recent Changes' section, there is a 'Stage View' table showing the average stage times for the pipeline. The table has columns for 'Declarative: Checkout SCM', 'Test', 'Build', and 'Declarative: Post Actions'. The 'Build' column shows the average stage time for the 'Build' stage. The 'Build History' section shows a list of builds, with build #3 highlighted in green.

Declarative: Checkout SCM	Test	Build	Declarative: Post Actions
1s	29s	2min 9s	167ms
1s	53s	3min 50s	136ms
1s	11s	1min 6s	153ms
1s	24s	1min 31s	214ms

- Quan sát và kiểm tra khi có lỗi phát sinh trong quá trình test và build.

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (admin, log out). The breadcrumb trail is Dashboard > FastAPI-Docker > #3. The left sidebar contains a list of links: Back to Project, Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#3', Polling Log, Git Build Data, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Console Output' and displays the following log:

```

Started by GitHub push by ErikHorus1249
Obtained Jenkinsfile from git https://github.com/ErikHorus1249/CICD.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/FastAPI-Docker
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential Github-PAT
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/FastAPI-Docker/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ErikHorus1249/CICD.git # timeout=10
Fetching upstream changes from https://github.com/ErikHorus1249/CICD.git
> git --version # timeout=10
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/ErikHorus1249/CICD.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision 0eab84a22af46ef4356f6747ebbb03e1f22930e (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0eab84a22af46ef4356f6747ebbb03e1f22930e # timeout=10
Commit message: "update"
> git rev-list --no-walk 92c2cf5f376a202003e3613a3be19f89926ec2b7 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker inspect -f . python:3.8-slim-buster
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withDockerContainer

```

- Quá trình build sẽ diễn ra tự động khi có code mới được push lên.

[*]Chúc ace thường đọc vui vẻ. Mọi ý kiến góp ý, chỉnh sửa xin để lại tại phần **Issues** . Chân thành cảm ơn !

Nguồn tham khảo:

- [poetry](#)
- [Topdev](#)
- [File-tree-extension](#)
- [Imgur](#)
- [Stackedit](#)