****ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**BÁO CÁO ĐỀ TÀI**

**continuous integration**

JENKINS

Nhóm thực hiện: Họ tên MSSV

1. Nguyễn Thái Hòa 1642019
2. Hà Nguyễn Thái Học 1642021
3. Dương Tấn Huỳnh Phong 1642049
4. Nguyễn Xuân Phúc 1642051

Lớp: 16HCB

Môn học: Công cụ kiểm chứng phần mềm

Mục lục

[I. GIỚI THIỆU JENKINS 3](#_Toc510024709)

[II. PHẠM VI HỔ TRỢ 3](#_Toc510024710)

[1. Hệ điều hành và môi trường 3](#_Toc510024711)

[2. Ngôn ngữ lập trình 3](#_Toc510024712)

[III. CÀI ĐẶT 3](#_Toc510024713)

[1. Download: 3](#_Toc510024714)

[2. Cài đặt: 4](#_Toc510024715)

[IV. CHỨC NĂNG 5](#_Toc510024716)

[1. New Item: 6](#_Toc510024717)

[2. People: 7](#_Toc510024718)

[3. Build History: 7](#_Toc510024719)

[4. Project Relationship: 8](#_Toc510024720)

[5. Check File Fingerprint: 8](#_Toc510024721)

[6. Manage Jenkins: 9](#_Toc510024722)

[7. Credentials: 9](#_Toc510024723)

[8. New View: 10](#_Toc510024724)

[V. GIỚI THIỆU CÁCH HOẠT ĐỘNG 10](#_Toc510024725)

[1. Tạo job 10](#_Toc510024726)

[2. Build project 12](#_Toc510024727)

1. GIỚI THIỆU JENKINS

Jenkins là một CI (continuous integration – tích hợp liên tục), một máy chủ tự động giúp build, test, deploy lên server một cách tự động.

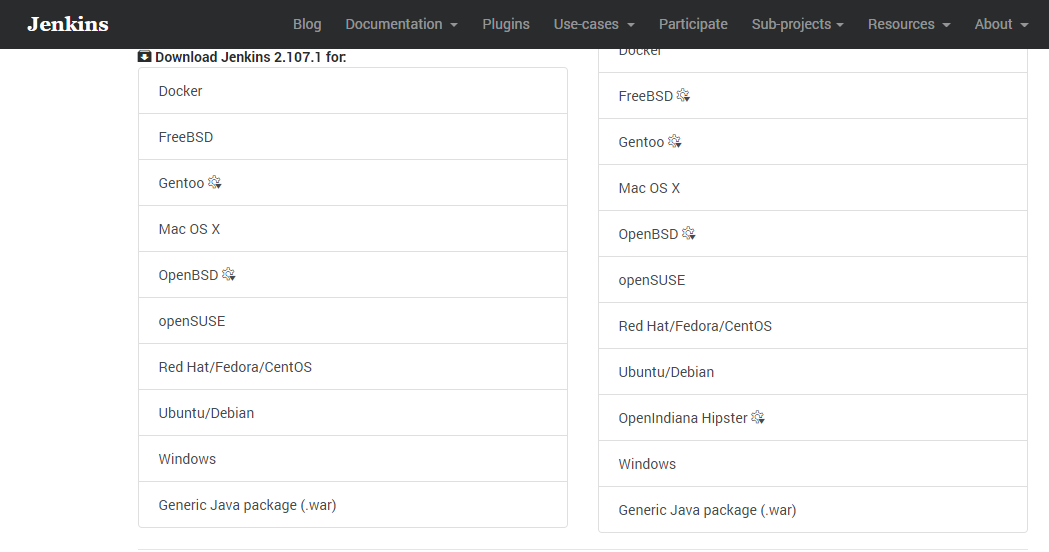
Jenkinds được viết bằng Java, có thể hoạt động trên hầu hết các công cụ khác nhau, từ một máy cục bộ, docker hay thậm chí là một server riêng, chỉ cần có JRE là có thể hoạt động được.

1. PHẠM VI HỔ TRỢ
   1. Hệ điều hành và môi trường

* Linux: hỗ trợ hầu hết các distro phổ biến như Ubuntu, Debian, RedHat, Fedora, CentOS, …
* Windows
* MacOS
* FreeBSD
* Docker
  1. Ngôn ngữ lập trình
* Android
* C/C++
* Java
* Node.js/Javascript
* PHP
* Python
* Ruby

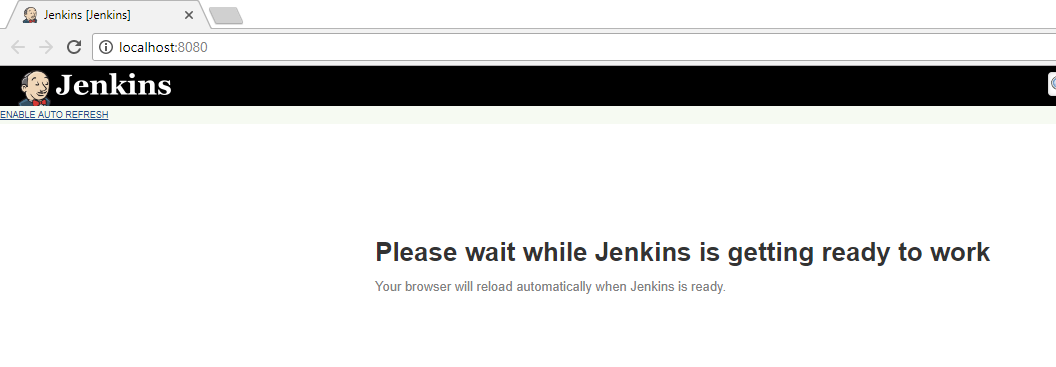
1. CÀI ĐẶT
   1. Download:

Tải Jenkins từ trang chủ: <https://jenkins.io/download/>. Tùy nền tảng sử dụng mà chọn phiên bản tương ứng.



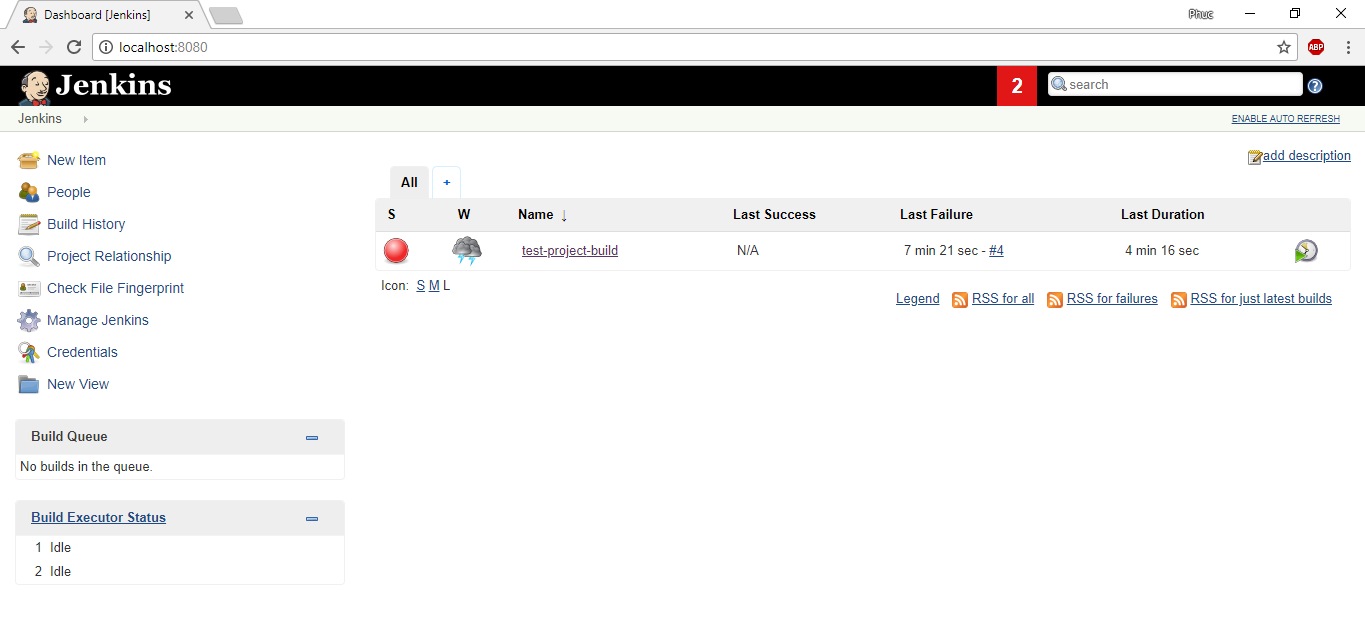
* 1. Cài đặt:

Sau khi download về, cài đặt theo hướng dẫn của chương trình. Sau khi cài đặt hoàn tất, Jenkins sẽ tạo một server ngay trong máy cục bộ và có địa chỉ là <http://localhost:8080>



1. CHỨC NĂNG

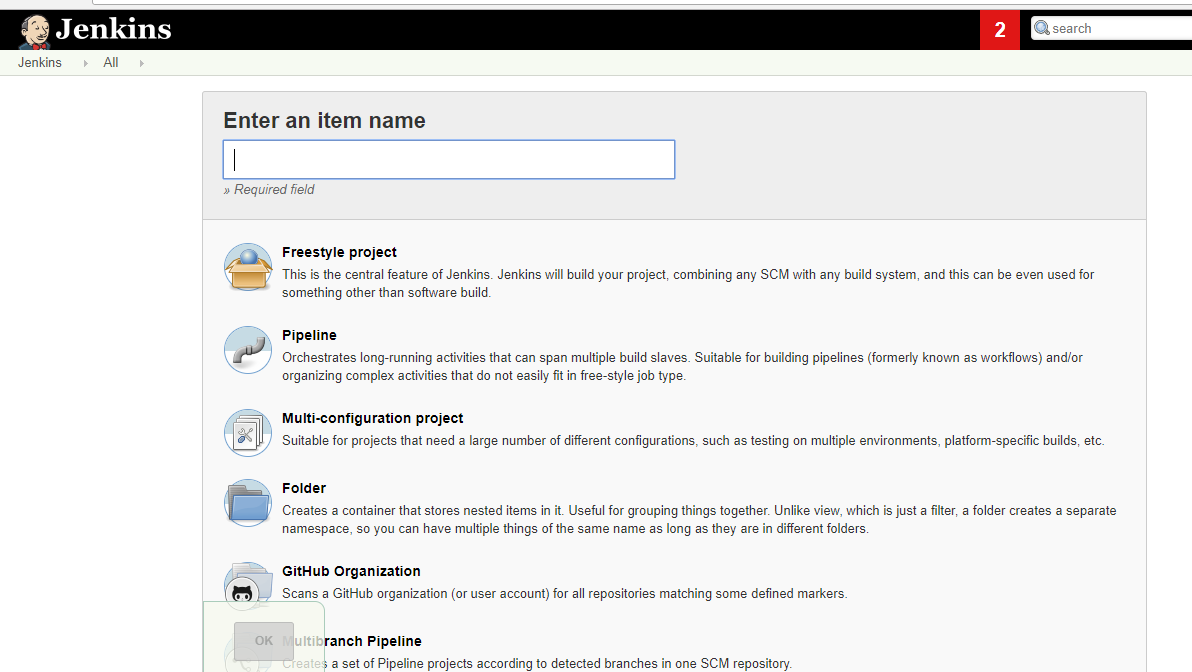
Giao diện chính của Jenkins



* 1. New Item:

Tạo một job/project mới trong jenkins

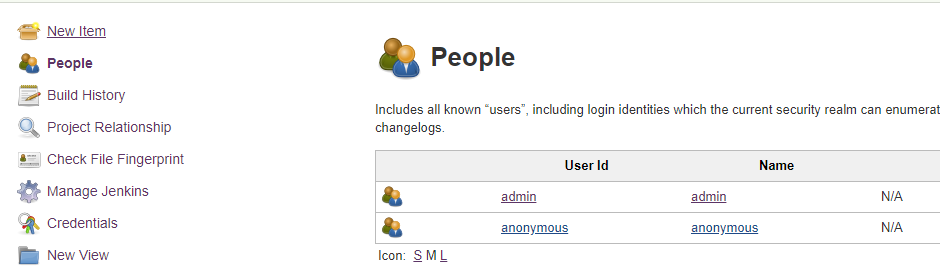
* *Freestyle project:* cho người dùng lựa chọn tất cả các loại thiết lập, là loại thiết lập linh hoạt nhất.
* *Maven project:* Maven 2/3 project thiết lập cho maven project, có thiết lập sẵn nhiều tính năng như là tự tích hợp tích hợp JUnit, findbugs, Incremental build (chỉ build phần module thay đổi thôi),Build modules in parallel,...
* *Pipeline:* project build theo 1 chuỗi (chain) các bước, ta sẽ tìm hiểu kỹ hơn ở phần Pipeline.
* *Eternal Job:* những job chạy ngoài Jenkins (có thể là ở remote host).
* *Folder:* thư mục chung chứa nhiều job lồng nhau.
* *GitHub Organization:* thiết lập cho tất cả project trong account github (chỉ dùng được cho github premium).
* *Multibranch Pipeline:* thiết lập pipeline cho project trên nhiều branch khác nhau.



* 1. People:

Quản lý người dùng trong dự án

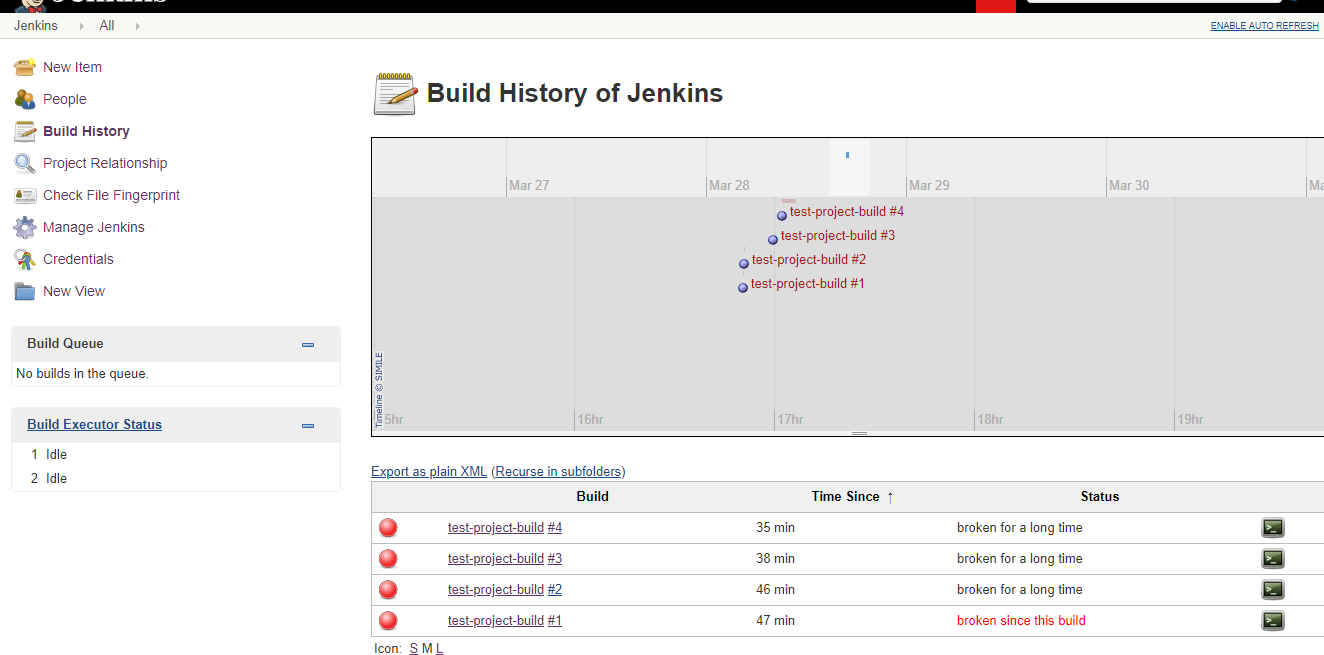
* Quản lý danh sách người dùng
* Quảy lý tùy chỉnh đối với từng chức năng người dùng.



* 1. Build History:

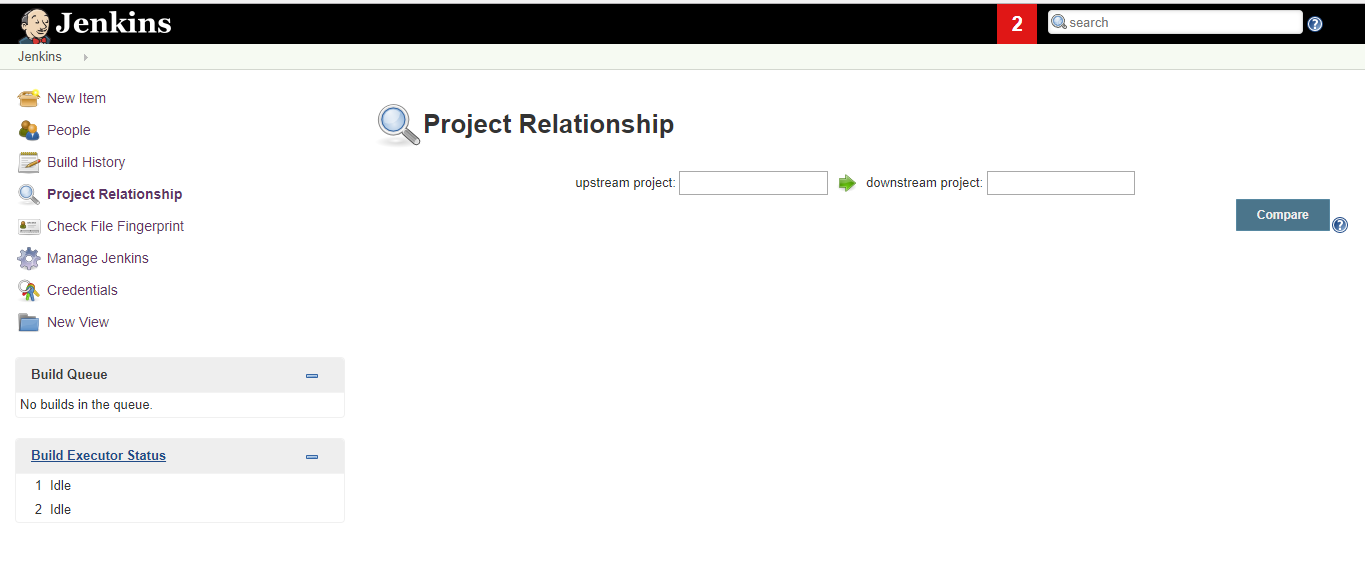
Xem lịch sử thực thi

* Hiển thị lịch sử thực thi của các job.
* Có thể xem chi tiết của các job bằng cách click vào tên job.



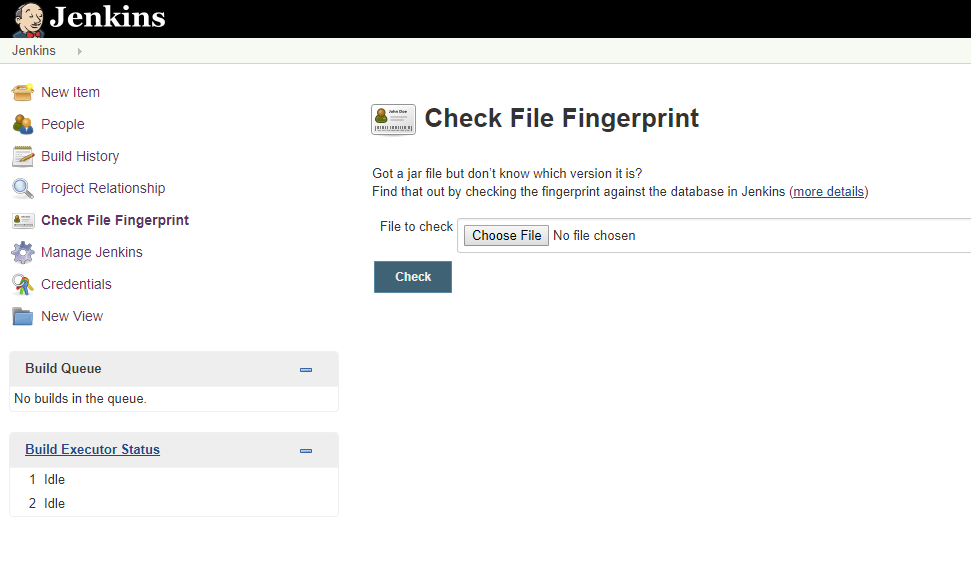
* 1. Project Relationship:

Xem các mối quan hệ giữa các project trong jenkins.



* 1. Check File Fingerprint:

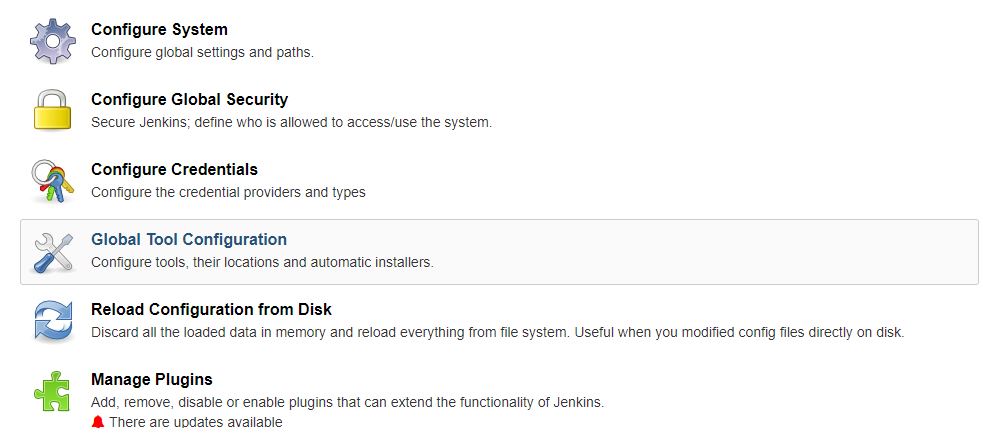
Dùng để kiểm tra phiên bản của các file được sử dụng trong project.



* 1. Manage Jenkins:

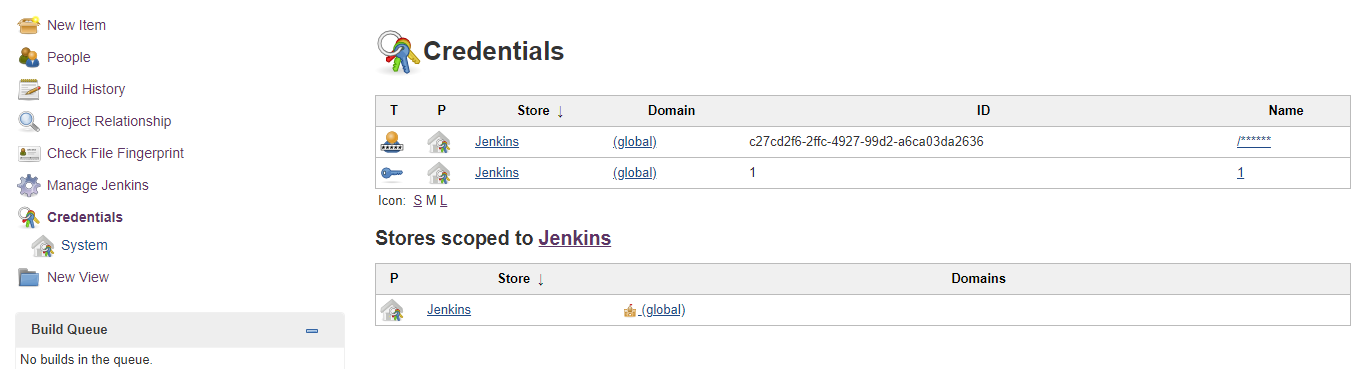
Quản lý cài đặt jenkins, bao gồm một số chức năng chính như:

* Quản lý cài đặt hệ thống.
* Quản lý bảo mật
* Quản lý tài khoản.
* Quản lý các plugin
* …



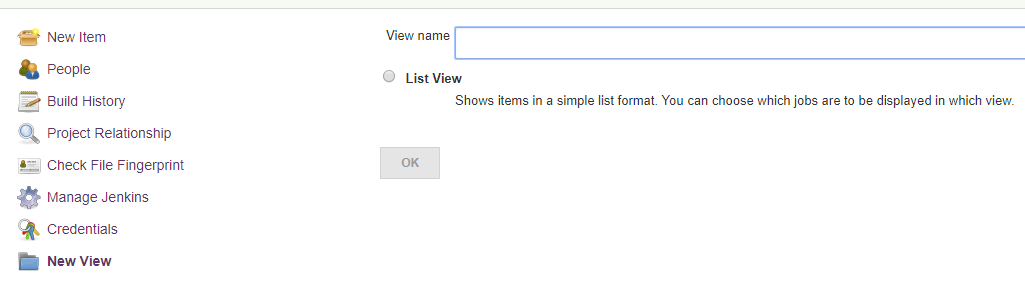
* 1. Credentials:

Quản lý các thông tin đăng nhập của jenkins.



* 1. New View:

Quản lý view, quản lý những chức năng và tùy chọn giới hạn cho người dùng.



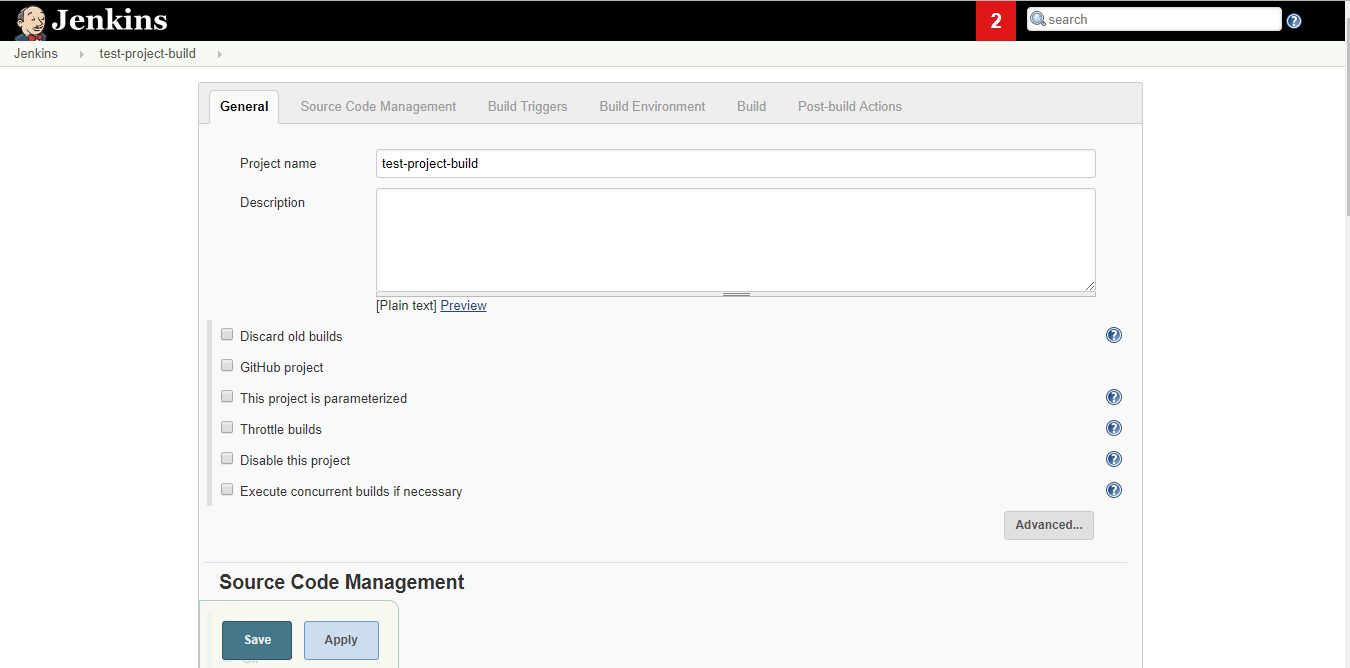
1. GIỚI THIỆU CÁCH HOẠT ĐỘNG

Các hoạt động trong jenkins đều thực thi dưới dạng các job.

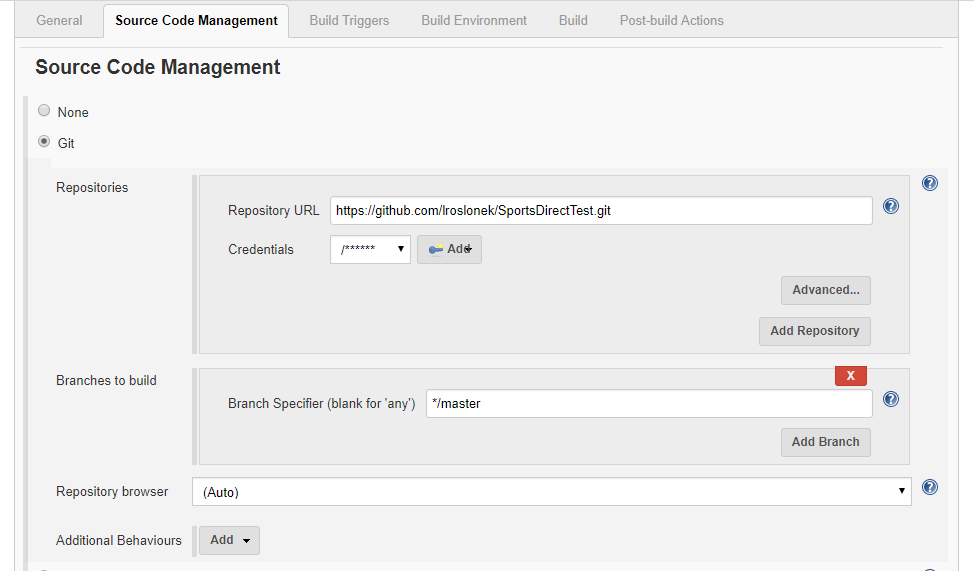
* 1. Tạo job

Để tạo các job thì ta vào chọn New Item sau đó chọn loại job tùy vào nhu cầu như giới thiệu ở trên. Ví dụ chọn ở đây demo chọn là *Freestyle project*. Nhập tên job (tên các job phải phân biệt với nhau).

Sau khi tạo job sẽ bắt đầu tiến hành tùy chỉnh job:

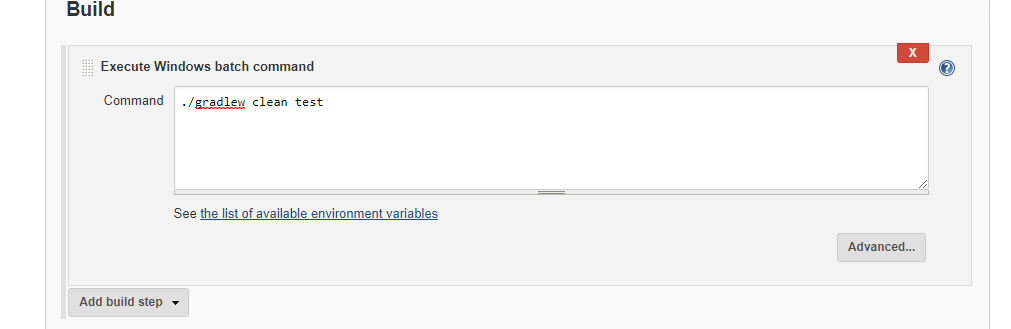


* General: các thông số chung, trong ví dụ demo thì ta không cần phải chỉnh gì.
* Source Code Management: nguồn quản lý source: có 3 lựa chọn là None (không có), Git và Subversion. Ở đây demo ta sẽ chọn Git. Sử dụng một project demo có sẵn cho jenkins: <https://github.com/lroslonek/SportsDirectTest.git>



* Build Trigger: cho phép tùy chỉnh việc build (thực hiện từ động) job. Có một số tùy chọn như:
  + *Build after other projects are built* (Build sau khi một job khác được build).
  + *Build periodically*: build định kì.
  + *GitHub hook trigger for GITScm polling*: build khi có sự thay đổi trên git.
  + *Poll SCM*: một dạng build định kỳ, nhưng trước khi build sẽ kiểm tra nếu có sự thay đổi git kể từ lần build trước đó thì mới thực hiện việc build (build periodically sẽ build theo định kì, dù code không thay đổi).
* Build Environment: build job trên môi trường nào.
* Build: cách thức build job thông qua các bước lệnh nào. Các lệnh này có thể là batch, shell command, Ant, Gradle script, …

Ở ví dụ demo, ta thực hiện build bằng batch command với lệnh là *./gradlew clean test*



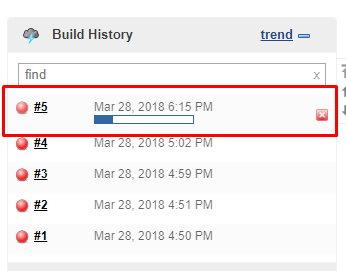
* Post-build Actions: các việc sẽ làm sau khi build xong job.

Sau khi cài đặt xong thì ta nhấn Save, các cài đặt sẽ được lưu lại và chuyển sang trang Project.

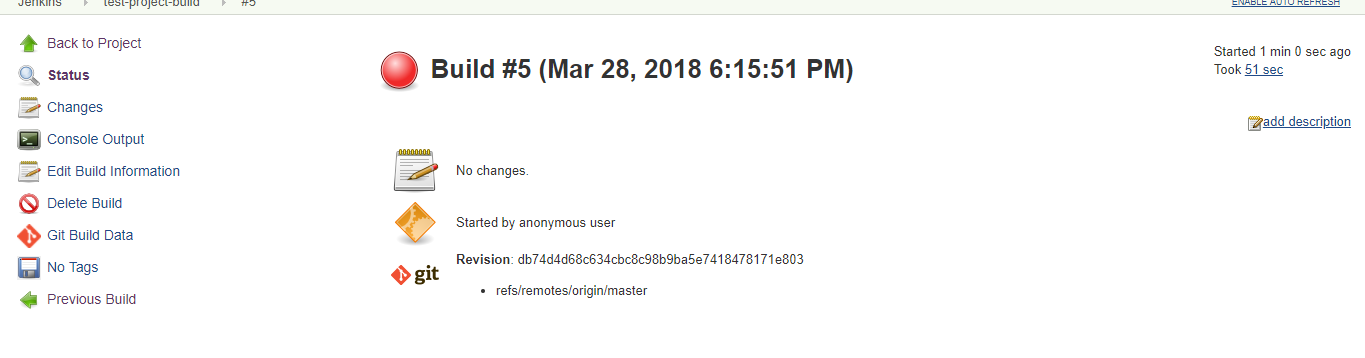
* 1. Build project



Để build project, ta trong trang project ta click *Build Now*, project sẽ bắt đầu được chạy



Để xem chi tiết kết quả build, click vào lịch sử build tương ứng.



Để xem kết quả thực thi, click vào Console Output để xem kết quả

