**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

BÁO CÁO BTL THUỘC HỌC PHẦN

CÔNG CỤ PHÁT TRIỂN PHẦN MỀM

**NGHIÊN CỨU CÔNG CỤ JENKINS VÀ ỨNG DỤNG**

|  |  |
| --- | --- |
| GVHD: | TS. Hà Mạnh Đào |
| Nhóm - Lớp: | 3 - 20222IT6003001-CCPTPM |
| Thành viên: | Lê Bá Thành - 2020602932 |
|  | Vũ Ngọc Thanh - 2020603347 |
|  | Dương Văn Thắng- 2020603234 |
|  | Lê Huy Toàn - 2020603651  Vũ Đức Triệu - 2020603821 |

Hà Nội, Năm 2023

[Chương 1: Tổng quan về các công cụ phần mềm 5](#_Toc136637669)

[1.1 Quality center 5](#_Toc136637670)

[1.2 Test monitor 7](#_Toc136637671)

[1.3 IronWASP 8](#_Toc136637672)

[1.4 LoadRunner 10](#_Toc136637673)

[1.5 Apache Jmeter 11](#_Toc136637674)

[1.6 JIRA 13](#_Toc136637675)

[1.7 Selenium 19](#_Toc136637676)

[Chương 2: Tìm hiểu về công cụ Jenkins 20](#_Toc136637677)

[2.1 JenKins 20](#_Toc136637678)

[2.1.1 Jenkins là gì? 20](#_Toc136637679)

[2.1.2 Một số tính năng quan trọng của Jenkins? 21](#_Toc136637680)

[2.1.3 Ưu và nhược điểm của việc sử dung Jenkins 22](#_Toc136637681)

[2.2 Pipeline 23](#_Toc136637682)

[2.2.1 Jenkins Pipeline là gì? 23](#_Toc136637683)

[2.2.2 Cách để tạo một Pipeline trong Jenkins 25](#_Toc136637684)

[2.3 JenKinsFile 26](#_Toc136637685)

[2.3.1 JenKinsFile là gì? 26](#_Toc136637686)

[2.3.2 Ưu và nhược điểm của JenkinsFile 27](#_Toc136637687)

[2.3.3 Cách để tạo 1 JenkinsFile mới 29](#_Toc136637688)

[2.4 Plugin 30](#_Toc136637689)

[2.4.1 Plugin là gì? 30](#_Toc136637690)

[2.4.2 Ưu nhược điểm của Plugin 31](#_Toc136637691)

[2.4.3 Các nhóm Plugin 32](#_Toc136637692)

[2.4.4 Bảy Plugin phổ biến 32](#_Toc136637693)

[2.4.5 Ba cách cài đặt Plugin 33](#_Toc136637694)

[2.5 Integration 37](#_Toc136637695)

[2.5.1 Integration là gì? 37](#_Toc136637696)

[2.5.2 Minh họa 38](#_Toc136637697)

[2.5.3 Các đặc điểm thực tiễn của CI 38](#_Toc136637698)

[2.5.4 Ưu điểm và nhược điểm của Continuous Integration 39](#_Toc136637699)

[2.5.5 Các công cụ sử dụng 39](#_Toc136637700)

[2.5.6 Cài đặt và cấu hình tích hợp (integration) trong Jenkins 40](#_Toc136637701)

[Chương 3: Ứng dụng của Jenkins 41](#_Toc136637702)

[3.1 Continuous Integration (CI) 41](#_Toc136637703)

[3.1.1 Khái niệm 41](#_Toc136637704)

[3.1.2 Các bước thực hiện CI trong Jenkins 41](#_Toc136637705)

[3.2 Automated Builds 42](#_Toc136637706)

[3.2.1 Khái niệm 42](#_Toc136637707)

[3.2.2 Tổng quan về quá trình Automated Builds 42](#_Toc136637708)

[3.3 Continuous Delivery/Deployment (CD) 43](#_Toc136637709)

[3.3.1 Khái niệm 43](#_Toc136637710)

[3.3.2 Tổng quan về Continuous Delivery/Deployment trong Jenkins 43](#_Toc136637711)

[3.4 Scheduled Jobs 45](#_Toc136637712)

[3.4.1 Khái niệm 45](#_Toc136637713)

[3.4.2 Tổng quan về Scheduled Jobs trong Jenkins 45](#_Toc136637714)

[3.5 Integration with Other Tools 46](#_Toc136637715)

[3.5.1 Khái niệm 46](#_Toc136637716)

[3.5.2 Tổng quan về Integration with Other Tools trong Jenkins 46](#_Toc136637717)

[3.6 Cách cài đặt Jenkins 47](#_Toc136637718)

[3.7 Ứng dụng Jenkins Pipeline vào tự động hóa với Github 57](#_Toc136637719)

[Tài liệu tham khảo 62](#_Toc136637720)

**Lời mở đầu**

Trong thế giới phát triển phần mềm đầy cạnh tranh ngày nay, việc tự động hóa quá trình xây dựng, kiểm thử và triển khai phần mềm là một yếu tố quan trọng để nâng cao chất lượng sản phẩm và tối ưu hóa hiệu suất làm việc của đội ngũ phát triển. Trong bộ môn công cụ phát triển phần mềm, chúng ta khám phá và nghiên cứu về các công cụ hỗ trợ phát triển phần mềm nhằm giúp tối đa hóa quá trình phát triển và quản lý dự án.

Chúng em, nhóm 3, dưới sự hướng dẫn của giảng viên Hà Mạnh Đào, đã chọn công cụ Jenkins để tìm hiểu và nghiên cứu trong bài tập lớn này. Jenkins là một công cụ mã nguồn mở rất phổ biến, được sử dụng để xây dựng và triển khai tự động phần mềm. Chúng em đã tiến hành tìm hiểu về kiến trúc, tính năng, và quy trình sử dụng Jenkins, cũng như thực hiện ứng dụng thực tế để đánh giá hiệu quả và lợi ích mà công cụ này mang lại.

Báo cáo này tập trung trình bày về quá trình nghiên cứu công cụ Jenkins và các khía cạnh quan trọng liên quan, bao gồm cài đặt, cấu hình, tích hợp và quản lý Jenkins.

Bằng việc thực hiện bài tập lớn này, chúng em mong muốn không chỉ củng cố kiến thức về công cụ Jenkins và các phương pháp liên quan trong phát triển phần mềm, mà còn nâng cao khả năng ứng dụng thực tế của chúng em. Bên cạnh đó, chúng em hy vọng rằng báo cáo này sẽ mang lại cái nhìn tổng quan về công cụ Jenkins và lợi ích mà nó mang lại trong quá trình phát triển phần mềm.

Chúng em xin chân thành cảm ơn

Nhóm 3

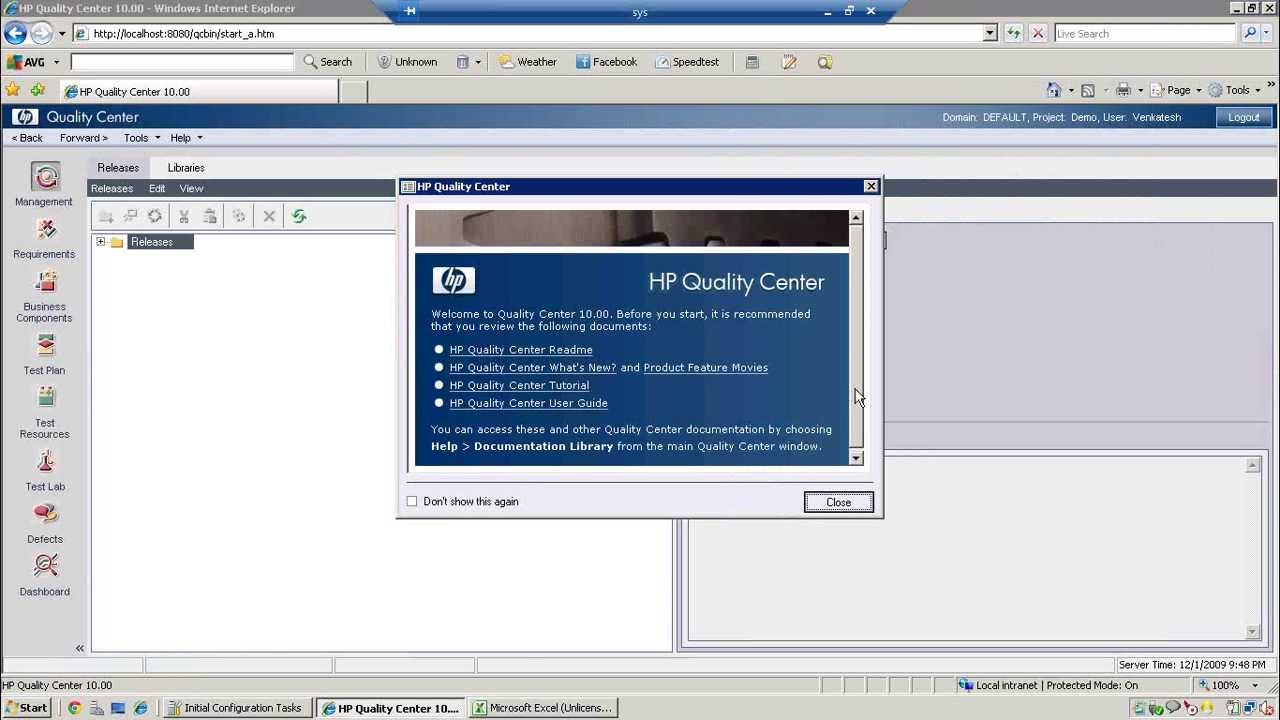
**PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Nội dung công việc*** | ***Kết quả đạt được*** | ***Thời gian bắt đầu, kết thúc*** | ***Người thực hiện*** |
|  | Tìm hiểu về đề tài lên kịch bản và kế hoạch | Hoàn Thành | 12/03/2023-16/03/2023 | Cả nhóm |
|  | Tìm hiểu cơ bản về Jenkins | Hoàn Thành | 17/03/2023-30/03/2023 | Dương Văn Thắng |
|  | Tìm hiểu về PipeLine | Hoàn Thành | 31/03/2023-6/04/2023 | Vũ Ngọc Thanh |
|  | Tìm hiểu về Jenkinsfile | Hoàn Thành | 07/04/2023-14/04/2023 | Lê Bá Thành |
|  | Tìm hiểu về Plugin | Hoàn Thành | 15/04/2023-  22/04/2023 | Lê Huy Toàn |
|  | Tìm hiểu về Integration | Hoàn Thành | 23/04/2023-  1/5/2023 | Vũ Đức Triệu |
|  | Tìm hiểu về Scale | Hoàn Thành | 2/5/2023-  8/5/2023 | Cả nhóm |
|  | Ứng dụng thực tế công cụ Jenkins vào quy trình tự động hóa với Github | Hoàn Thành | 9/5/2023-  16/5/2023 | Cả nhóm |
|  | Tổng kế và hoàn thiện báo cáo | Hoàn thành | 17/5/2023-  1/6/2023 | Cả nhóm |

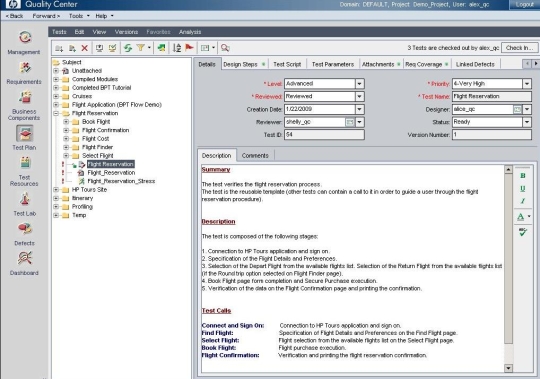
# Chương 1: Tổng quan về các công cụ phần mềm

## Quality center

* Quality Center (QC) là một công cụ quản lý kiểm thử phần mềm do công ty Hewlett-Packard (HP) phát triển. QC được sử dụng để quản lý quá trình kiểm thử phần mềm, từ việc lập kế hoạch kiểm thử, thiết kế ca kiểm thử, thực hiện kiểm thử, theo dõi tiến độ kiểm thử, đưa ra báo cáo và phân tích kết quả kiểm thử.
* Các chức năng chính của QC bao gồm:
* Quản lý yêu cầu và ca kiểm thử
* Thiết kế và quản lý kế hoạch kiểm thử
* Thiết kế và thực hiện kiểm thử tự động
* Quản lý dữ liệu kiểm thử
* Đánh giá kết quả kiểm thử
* Theo dõi tiến độ và báo cáo kết quả kiểm thử
* QC hỗ trợ nhiều ngôn ngữ lập trình và nền tảng khác nhau, cho phép các nhà phát triển phần mềm sử dụng các công cụ và ngôn ngữ lập trình phù hợp nhất để thực hiện kiểm thử phần mềm.
* QC cũng có tính năng tích hợp với các công cụ kiểm thử tự động như HP QuickTest Professional (QTP) và Selenium, giúp cho quá trình kiểm thử phần mềm được tự động hóa và hiệu quả hơn.
* Tổng thể, QC là một công cụ quản lý kiểm thử phần mềm đầy đủ tính năng và mạnh mẽ, giúp cho các nhà phát triển phần mềm quản lý quá trình kiểm thử phần mềm hiệu quả và tối ưu hóa quá trình phát triển phần mềm.



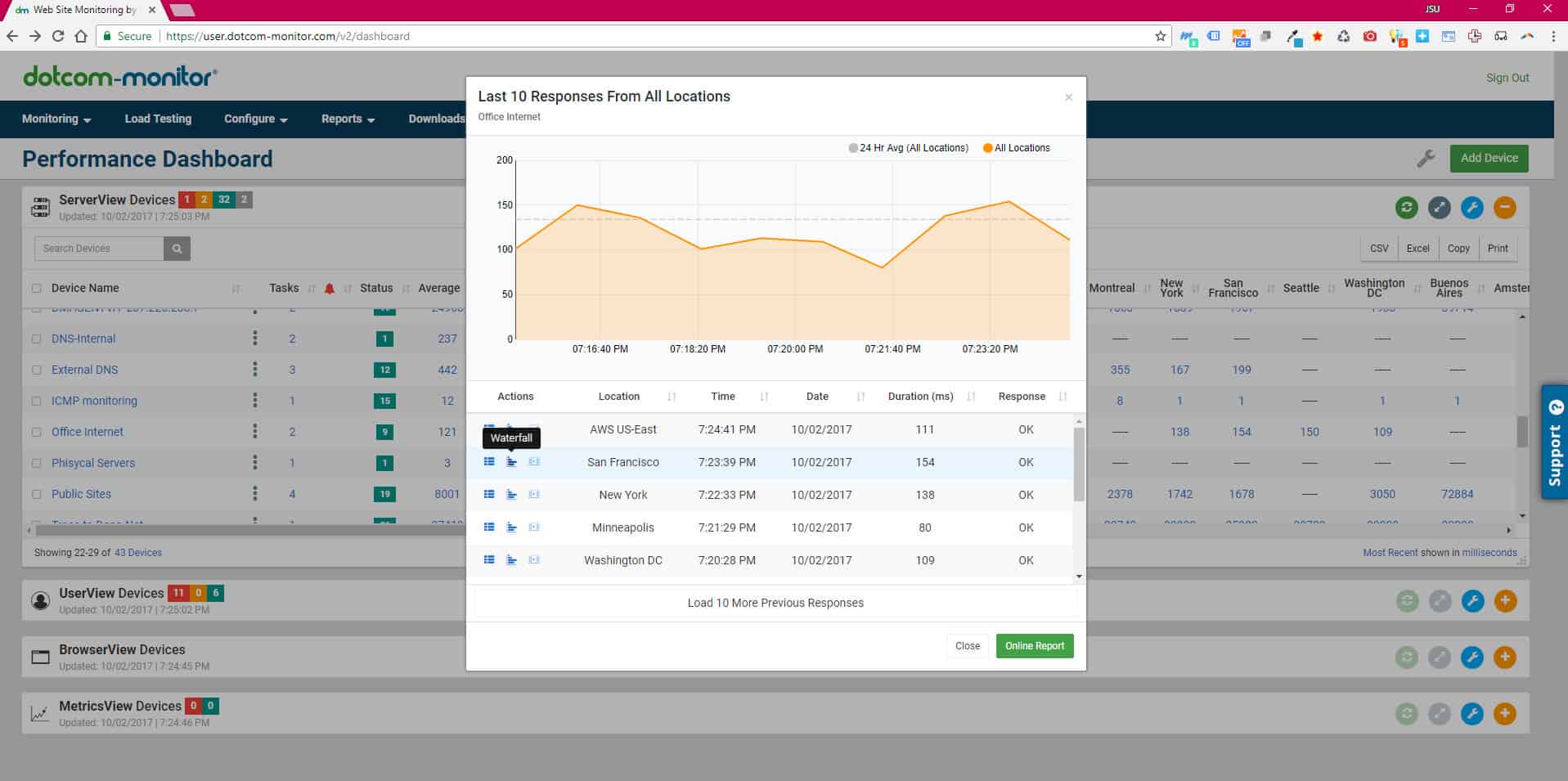
*Hình 1.1. Công cụ Quality center*



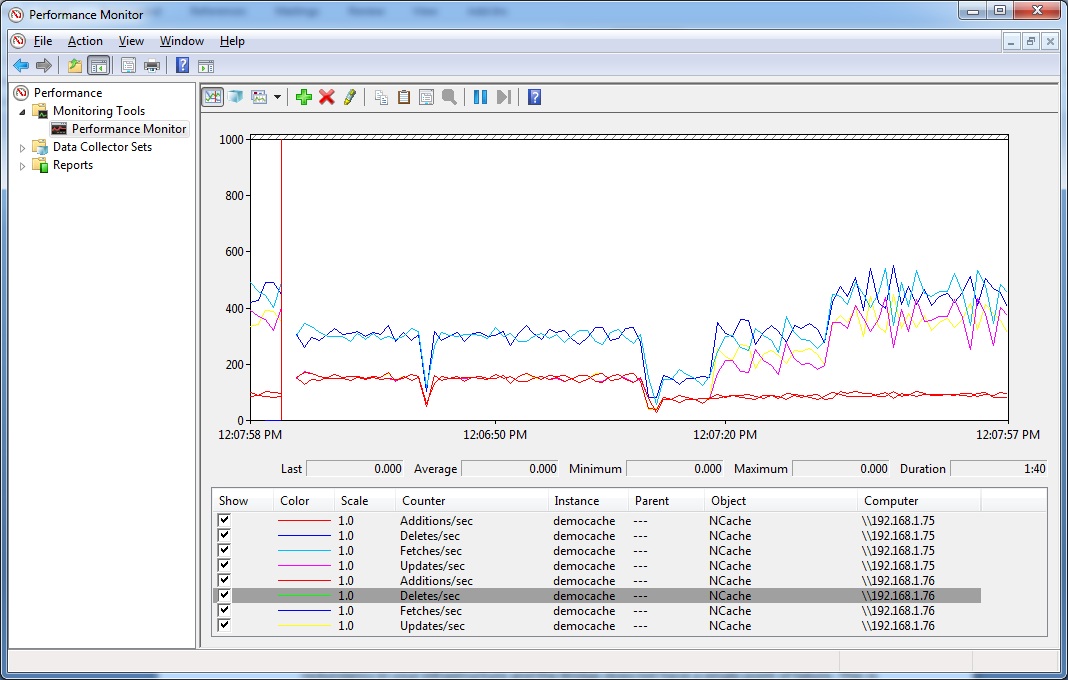
*Hình 1.2. Công cụ Quality center*

## Test monitor

* Test Monitor là một công cụ phần mềm kiểm thử tự động được sử dụng để giám sát các chương trình máy tính và các ứng dụng trên các thiết bị di động. Công cụ này cung cấp khả năng kiểm tra tích hợp, kiểm thử mô-đun và kiểm tra hệ thống tự động.
* Các tính năng chính của Test Monitor bao gồm:
* Giám sát các ứng dụng và trang web trên các thiết bị di động.
* Kiểm tra hiệu suất ứng dụng, đo lường thời gian phản hồi và thời gian tải.
* Thực hiện kiểm thử tự động và tự động hóa các bước kiểm thử.
* Tích hợp dữ liệu kiểm thử vào các công cụ quản lý kiểm thử khác, chẳng hạn như JIRA hoặc Quality Center…
* Theo dõi tiến độ kiểm thử và báo cáo kết quả kiểm thử.
* Test Monitor được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm kiểm thử phần mềm, kiểm thử ứng dụng di động và kiểm thử trang web. Công cụ này cũng được sử dụng để kiểm tra tính tương thích của các ứng dụng với các thiết bị di động và các trình duyệt web khác nhau.
* Với Test Monitor, các nhà phát triển và kiểm thử viên có thể tự động hóa quá trình kiểm thử và giảm thiểu thời gian và chi phí kiểm thử, đồng thời cải thiện chất lượng phần mềm và tăng hiệu quả của các dự án phần mềm.



*Hình 2.1. Ví dụ về công cụ Test monitor*



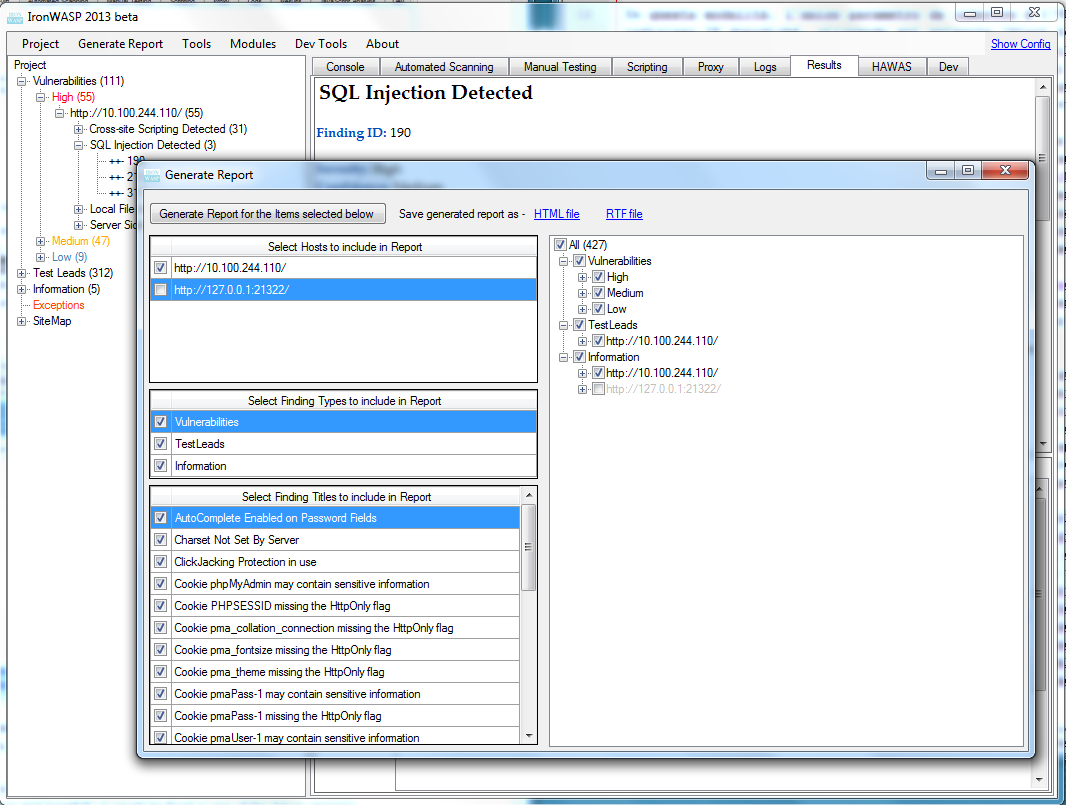
*Hình 2.2. Ví dụ về công cụ Test monitor*

## IronWASP

* IronWASP là một công cụ kiểm thử bảo mật ứng dụng web tự động. Công cụ này được phát triển bởi IronWASP Team và được cung cấp dưới dạng phần mềm miễn phí.
* IronWASP có các tính năng chính sau:
* Phát hiện các lỗ hổng bảo mật trong các ứng dụng web như SQL injection, cross-site scripting (XSS), file inclusion, local file inclusion (LFI), remote file inclusion (RFI), và nhiều hơn nữa.
* Hỗ trợ kiểm thử ứng dụng web đơn và đa trang.
* Cung cấp một giao diện đồ họa để tạo và quản lý các bài kiểm thử.
* Cung cấp khả năng tùy chỉnh các phương thức kiểm thử và kịch bản.
* Hỗ trợ tính năng tìm kiếm và phân tích mã nguồn.
* Cung cấp báo cáo chi tiết về các lỗ hổng bảo mật được tìm thấy, bao gồm các lỗ hổng cấp độ ưu tiên khác nhau.
* IronWASP có thể được sử dụng để kiểm thử bảo mật các ứng dụng web trên các hệ điều hành khác nhau như Windows, Linux và macOS. Công cụ này là một trong những công cụ kiểm thử bảo mật ứng dụng web phổ biến và được ưa chuộng trong cộng đồng phát triển và kiểm thử phần mềm.



*Hình 3.1. Công cụ IronWASP*

**

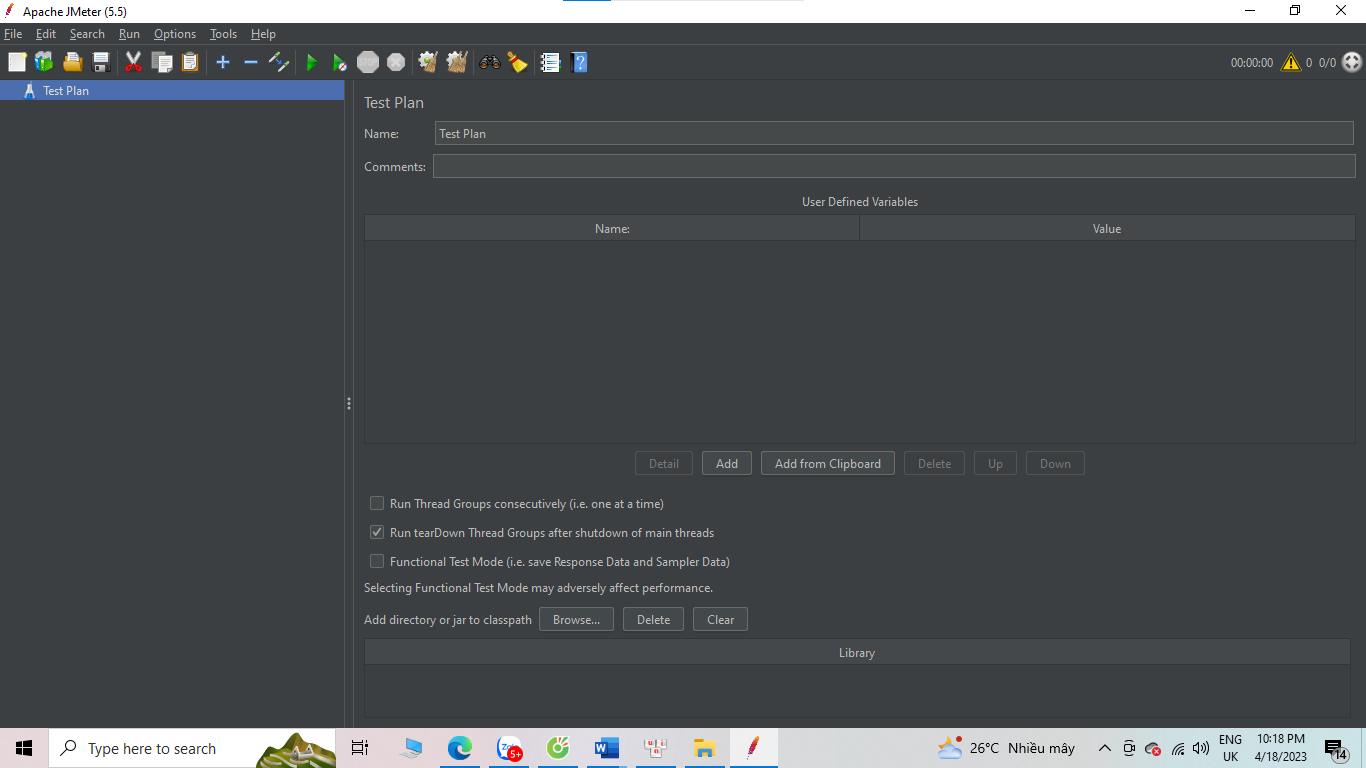
*Hình 3.2. Công cụ IronWASP*

## LoadRunner

* LoadRunner là một công cụ tự động hóa kiểm thử hiệu suất (performance testing) và tải (load testing) của ứng dụng máy tính, được phát triển bởi hãng Hewlett Packard (HP). Nó được sử dụng để đo đạc, kiểm tra và đánh giá khả năng chịu tải của ứng dụng máy tính trước khi triển khai chúng vào môi trường sản xuất.
* LoadRunner hỗ trợ nhiều giao thức khác nhau như HTTP, HTTPS, SOAP, REST, FTP, LDAP, JDBC, JMS, SMTP, POP3, và nhiều giao thức khác, cho phép người dùng mô phỏng và kiểm tra các hoạt động giao tiếp giữa các thành phần trong môi trường ứng dụng thực tế. Nó cũng cung cấp khả năng kiểm tra tải song song (concurrent users) trên nhiều nền tảng và thiết bị khác nhau, giúp đưa ra đánh giá chính xác về khả năng chịu tải của ứng dụng.
* LoadRunner cũng cung cấp khả năng theo dõi, thu thập và phân tích dữ liệu hiệu suất của ứng dụng trong quá trình kiểm thử, giúp phát hiện và giải quyết các vấn đề hiệu suất, tối ưu hóa hiệu suất ứng dụng, và đảm bảo ứng dụng hoạt động ổn định dưới tải cao.
* Một số tính năng chính của LoadRunner gồm:
  + Tạo kịch bản kiểm thử linh hoạt: LoadRunner cho phép người dùng tạo kịch bản kiểm thử bằng cách ghi lại hoạt động của người dùng trên ứng dụng hoặc tạo kịch bản tùy chỉnh sử dụng ngôn ngữ kịch bản của riêng nó.
  + Tạo tải cao đa dạng: LoadRunner cho phép mô phỏng tải cao từ nhiều nguồn khác nhau, đồng thời hỗ trợ kiểm thử đồng thời từ nhiều địa điểm khác nhau trên thế giới.
* Thu thập và phân tích dữ liệu hiệu suất: LoadRunner cung cấp khả năng thu thập, lưu trữ và phân tích dữ liệu hiệu suất của ứng dụng trong quá trình kiểm thử, giúp người dùng đưa ra đánh giá chính xác về hiệu suất của ứng dụng.
* Hỗ trợ nhiều giao thứckhác nhau như HTTP, HTTPS, SOAP, REST, FTP, LDAP, JDBC, JMS, SMTP, POP3, và nhiều giao thức khác, cho phép người dùng kiểm thử hiệu suất của các ứng dụng với nhiều giao thức khác nhau.
* Tích hợp với các công cụ và hệ thống khác: LoadRunner tích hợp tốt với các công cụ và hệ thống kiểm thử và quản lý dự án khác, cho phép tích hợp dễ dàng vào quy trình phát triển phổ biến như Agile hoặc DevOps.
* Đa nền tảng và đa hệ điều hành: LoadRunner hỗ trợ kiểm thử trên nhiều nền tảng và hệ điều hành khác nhau, bao gồm Windows, Linux, Unix, và Cloud.
* Giả lập môi trường thực tế: LoadRunner cho phép người dùng giả lập môi trường thực tế của người dùng cuối, bao gồm giả lập đa dạng địa điểm, thiết bị, trình duyệt, và băng thông mạng để đo đạc và kiểm thử hiệu suất của ứng dụng trong các điều kiện khác nhau.
* Báo cáo và phân tích kết quả: LoadRunner cung cấp các tính năng báo cáo và phân tích kết quả chi tiết, giúp người dùng đánh giá hiệu suất của ứng dụng, định vị và giải quyết các vấn đề hiệu suất, và cung cấp thông tin chi tiết để đưa ra quyết định về việc tối ưu hóa hiệu suất ứng dụng.

## Apache Jmeter

* Apache JMeter là một công cụ mã nguồn mở được sử dụng để kiểm thử hiệu suất và tải của ứng dụng web. Nó được phát triển bởi Apache Software Foundation và được sử dụng rộng rãi trong cộng đồng kiểm thử hiệu suất. Các tính năng chính của Apache JMeter bao gồm
* Hỗ trợ nhiều giao thức: Apache JMeter hỗ trợ nhiều giao thức web phổ biến như HTTP, HTTPS, SOAP, REST, FTP, JDBC, LDAP, JMS, SMTP, POP3, và nhiều giao thức khác. Điều này cho phép người dùng kiểm thử hiệu suất của các ứng dụng web với nhiều giao thức khác nhau.
* Đa nền tảng: Apache JMeter hoạt động trên nhiều nền tảng, bao gồm Windows, Linux, Unix, và macOS, cho phép người dùng kiểm thử hiệu suất của ứng dụng web trên nhiều môi trường khác nhau.
* Giao diện đồ họa thân thiện: Apache JMeter cung cấp giao diện đồ họa dễ sử dụng, cho phép người dùng tạo, cấu hình, và quản lý các kịch bản kiểm thử hiệu suất một cách trực quan.
* Tính năng ghi lại và phát lại: Apache JMeter cho phép người dùng ghi lại hoạt động của người dùng trên trình duyệt web, sau đó phát lại để kiểm thử hiệu suất của ứng dụng web dưới tải cao.
* Tích hợp với các công cụ và hệ thống khác: Apache JMeter tích hợp tốt với các công cụ và hệ thống kiểm thử và quản lý dự án khác, cho phép tích hợp dễ dàng vào quy trình phát triển phổ biến như Agile hoặc DevOps.
* Các tính năng mở rộng: Apache JMeter hỗ trợ các tính năng mở rộng như viết mã tùy chỉnh, kiểm thử đồng thời, kiểm thử phân tán, và nhiều tính năng mở rộng khác để đáp ứng các yêu cầu kiểm thử hiệu suất phức tạp.
* Báo cáo và phân tích kết quả: Apache JMeter cung cấp các tính năng báo cáo và phân tích kết quả chi tiết, giúp người dùng đánh giá hiệu suất của ứng dụng, định vị và giải quyết các vấn đề hiệu suất, vàtối ưu hóa ứng dụng web để cải thiện hiệu suất của nó.
* Hỗ trợ cho kiểm thử đa người dùng: Apache JMeter cho phép người dùng mô phỏng hoạt động của nhiều người dùng đồng thời, giúp đánh giá hiệu suất của ứng dụng web trong tình huống tải cao.
* Độ phổ biến và cộng đồng hỗ trợ: Apache JMeter là một công cụ phổ biến và có một cộng đồng lớn, cung cấp hỗ trợ, tài liệu, và các tài nguyên liên quan để giúp người dùng sử dụng công cụ một cách hiệu quả.



*Hình 5.1: Công cụ Apache Jmeter*

## JIRA

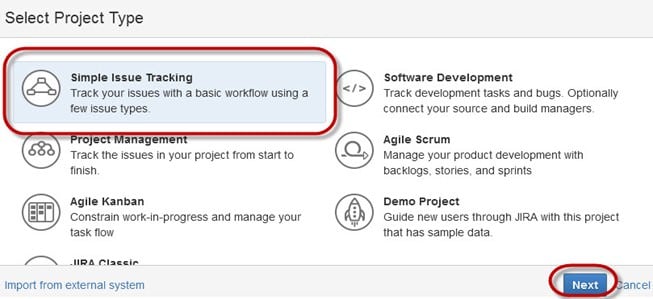
**JIRA** là Phần mềm được sử dụng để theo dõi lỗi, theo dõi vấn đề và quản lý dự án.

Phần mềm được phát triển để làm việc nhanh chóng và dễ dàng hơn, đem lại quy trình vô cùng khả quan cho tổ chức.

* **Điểm yếu của phần mềm Jira**
* Sử dụng Jira sẽ có chi phí cao, với những doanh nghiệp mà quy mô càng lớn thì số phí phải đóng để sử dụng ứng dụng sẽ càng lớn.
* Người sử dụng có thể tốn nhiều công sức và thời gian để có thể set up ứng dụng. Thế nên chúng chỉ có thể phát huy được tính siêu việt với những dự án có quy mô lớn. Ứng dụng này được nhận định không phù hợp với quy mô của những dự án nhỏ, có thời hạn hoàn thành dưới 3 tháng.
* Quy trình sử dụng Jira khá phức tạp, thế nên các lập trình viên phải đầu tư thời gian để sử dụng.
* Tiếng anh là ngôn ngữ chính của phần mềm, có nhiều thuật ngữ đòi hỏi sự tìm tòi cao
* **Cách tạo sự cố trong JIRA**

**Bước 1: Mở phần mềm Jira và đăng nhập bằng ID và mật khẩu của bạn**

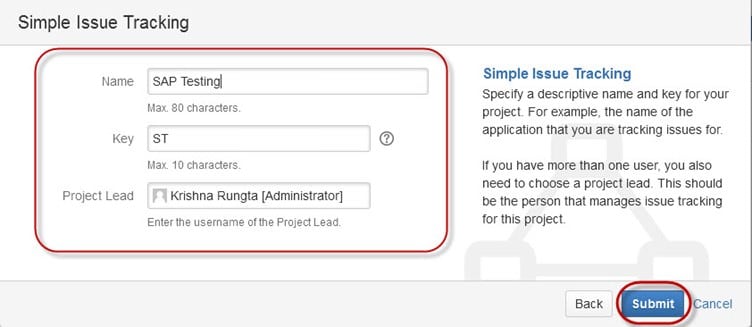
Bảng điều khiển JIRA sẽ mở khi bạn nhập ID người dùng và mật khẩu của mình. Trong bảng điều khiển JIRA miễn phí, bạn sẽ tìm thấy tùy chọn **Project,** khi bạn nhấp vào nó, nó sẽ mở ra một cửa sổ liệt kê các tùy chọn như **Simple Issue Tracking, Project Management, Agile Kanban, Jira Classic** , v.v. như minh họa trong ảnh chụp màn hình bên dưới.



*Hình 7.1: Công cụ JIRA*

**Bước 2: Cung cấp chi tiết vấn đề**

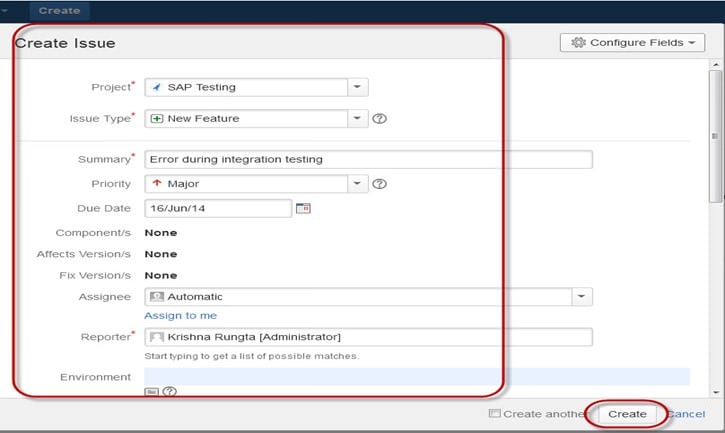
Khi bạn nhấp vào tùy chọn **Theo dõi vấn đề đơn giản,** một cửa sổ khác sẽ mở ra, nơi bạn có thể đề cập đến tất cả các chi tiết về vấn đề và cũng chỉ định vấn đề cho người chịu trách nhiệm.



*Hình 7.2: Công cụ JIRA*

**Bước 3: Cung cấp thông tin chi tiết để tạo một vấn đề**

Khi bạn nhấp vào nút “Gửi”, một cửa sổ sẽ mở ra nơi bạn có thể thực hiện danh sách công việc như tạo sự cố, chỉ định sự cố, kiểm tra trạng thái của sự cố như đã giải quyết, Đang xử lý hoặc đã đóng, v.v.



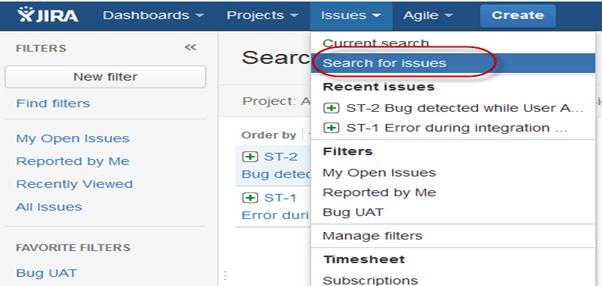
*Hình 7.3: Công cụ JIRA*

Sau khi sự cố được tạo, một cửa sổ bật lên sẽ xuất hiện trên màn hình của bạn cho biết sự cố của bạn đã được tạo thành công như trong ảnh chụp màn hình bên dưới

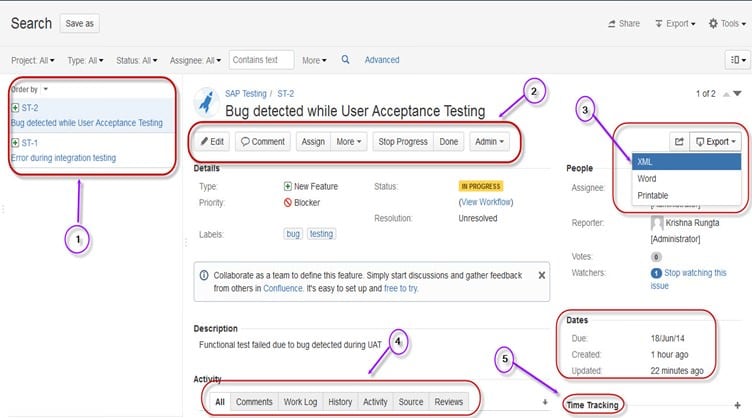
*Hình 7.4: Công cụ JIRA*

**Bước 4: Mở menu Sự cố để tìm kiếm và thực hiện nhiều chức năng đối với sự cố**

Bây giờ nếu bạn muốn chỉnh sửa một vấn đề hoặc bạn muốn xuất vấn đề sang tài liệu XML hoặc Word, thì bạn có thể di chuột trên bảng điều khiển chính và nhấp vào **Vấn đề** . Trong tùy chọn **Sự** cố, nhấp vào **tìm kiếm sự cố** sẽ mở ra một cửa sổ từ đó bạn có thể xác định vị trí sự cố của mình và thực hiện nhiều chức năng.



*Hình 7.5: Công cụ JIRA*

Khi bạn chọn **“tìm kiếm Sự cố”** trong Sự cố **,** một cửa sổ sẽ xuất hiện như minh họa trong ảnh chụp màn hình

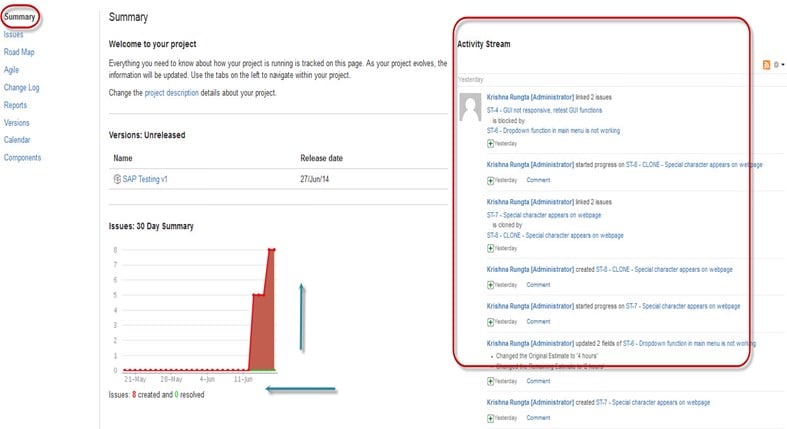
*Hình 7.6: Công cụ JIRA*

1. **Tùy chọn tìm kiếm sự cố** sẽ đưa bạn đến một cửa sổ nơi bạn có thể xem các sự cố do bạn tạo như ở đây, chúng tôi có sự cố ST1 và ST2
2. Ở đây trong ảnh chụp màn hình, bạn có thể thấy sự cố **“Đã phát hiện lỗi trong khi Kiểm tra mức độ chấp nhận của người dùng”** và tất cả các chi tiết liên quan đến sự cố đó. Từ đây, bạn có thể thực hiện nhiều tác vụ như bạn có thể **dừng tiến trình xử lý sự cố, chỉnh sửa sự cố, nhận xét về sự cố, chỉ định sự cố** , v.v.
3. Thậm chí, bạn có thể xuất chi tiết vấn đề sang tài liệu XML hoặc Word.
4. Ngoài ra, bạn có thể xem hoạt động đang diễn ra về sự cố, đánh giá về sự cố, nhật ký công việc, lịch sử của sự cố, v.v.
5. Trong tùy chọn theo dõi thời gian, bạn thậm chí có thể xem thời gian ước tính để giải quyết các sự cố

Trong cùng một cửa sổ, bạn có thể đặt bộ lọc cho vấn đề và lưu chúng trong **Bộ lọc yêu thích** , vì vậy khi muốn tìm kiếm hoặc xem một vấn đề cụ thể, bạn có thể định vị vấn đề đó bằng bộ lọc.

**Bước 5: Nhấp vào tùy chọn Tóm tắt để xem chi tiết về các vấn đề**

Để xem tóm tắt vấn đề, bạn có thể nhấp vào tùy chọn **tóm tắt, thao** tác này sẽ mở ra một cửa sổ hiển thị tất cả chi tiết về dự án và tiến độ của bạn trên biểu đồ này. Ở phía bên phải của cửa sổ tóm tắt, có một **Luồng hoạt động** cung cấp thông tin chi tiết về các vấn đề và nhận xét của người được giao về vấn đề đó.



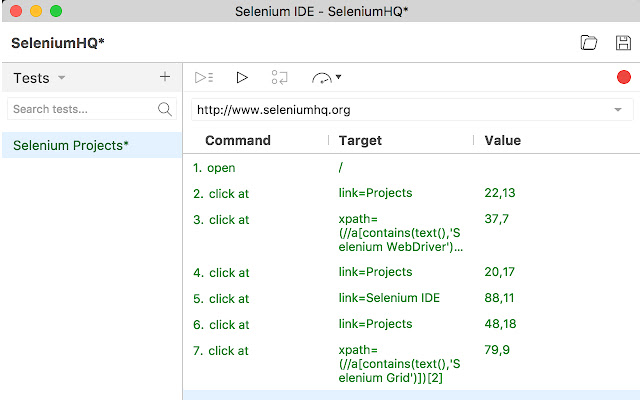
*Hình 7.7: Công cụ JIRA*

**Vai trò của JIRA**

* Ban đầu được thiết kế như một công cụ theo dõi lỗi và vấn đề, Jira đóng vai trò là công cụ quản lý công việc mạnh mẽ cho các trường hợp sử dụng khác nhau như:
* Quản lý yêu cầu và trường hợp kiểm tra – để quản lý các bài kiểm tra thủ công và tự động
* Các nhóm nhanh nhẹn – Phần mềm JIRA cung cấp bảng scrum và Kanban cho các nhóm thực hành các phương pháp nhanh nhẹn.
* Quản lý dự án – Phần mềm JIRA có thể được cấu hình để phù hợp với bất kỳ loại dự án nào ngay từ khi bắt đầu, cho đến khi thực hiện và kết thúc.
* Phát triển phần mềm – để phát triển phần mềm tốt hơn, nhanh hơn bằng cách kết hợp các công cụ Atlassian.
* DevOps – DevOps mở của Atlassian giúp các nhóm cung cấp phần mềm tốt hơn, nhấn mạnh vào các phương pháp hay nhất.
* Quản lý sản phẩm – JIRA giúp thiết kế lộ trình chi tiết, xử lý các phụ thuộc và chia sẻ kế hoạch cũng như tiến độ.
* Quản lý tác vụ –JIRA giúp dễ dàng tạo các tác vụ để thực hiện, với thông tin chi tiết, ngày đến hạn và lời nhắc.
* Theo dõi lỗi – công cụ xử lý công việc jira mạnh mẽ đảm bảo rằng các lỗi, sau khi đã bắt được, sẽ tự động được chỉ định và ưu tiên.

## Selenium

* Selenium là một công cụ tự động hóa trình duyệt web được sử dụng rộng rãi để kiểm thử ứng dụng web, tự động hóa các nhiệm vụ liên quan đến trình duyệt web và truy xuất dữ liệu từ các trang web.
* Công cụ này được xây dựng trên nền tảng Java và hỗ trợ nhiều ngôn ngữ lập trình khác nhau như Python, C#, Ruby, JavaScript, và PHP. Selenium cho phép bạn kiểm thử ứng dụng web trên nhiều trình duyệt khác nhau như Chrome, Firefox, Safari, và Edge.
* Selenium bao gồm ba thành phần chính:
* Selenium WebDriver: là thành phần chính của Selenium, cho phép bạn tương tác với trình duyệt web bằng cách sử dụng API của nó. WebDriver cho phép bạn tự động hóa các tác vụ như nhập liệu, nhấn nút, di chuyển trang, và lấy thông tin từ các trang web.
* Selenium IDE: là một công cụ ghi lại và chạy các kịch bản kiểm thử tự động trên trình duyệt web. Selenium IDE cho phép bạn ghi lại các thao tác trên trình duyệt và tạo các kịch bản kiểm thử tự động mà không cần viết mã.
* Selenium Grid: là một công cụ để phân phối các tác vụ kiểm thử trên nhiều trình duyệt và máy tính khác nhau. Selenium Grid cho phép bạn kiểm thử ứng dụng web trên nhiều môi trường khác nhau và đồng thời giảm thời gian kiểm thử.
* Selenium là một công cụ mạnh mẽ để tự động hóa kiểm thử ứng dụng web và tăng tốc độ phát triển. Nó cho phép bạn kiểm thử ứng dụng web trên nhiều trình duyệt và môi trường khác nhau và đồng thời tiết kiệm thời gian và công sức so với việc kiểm thử thủ công.

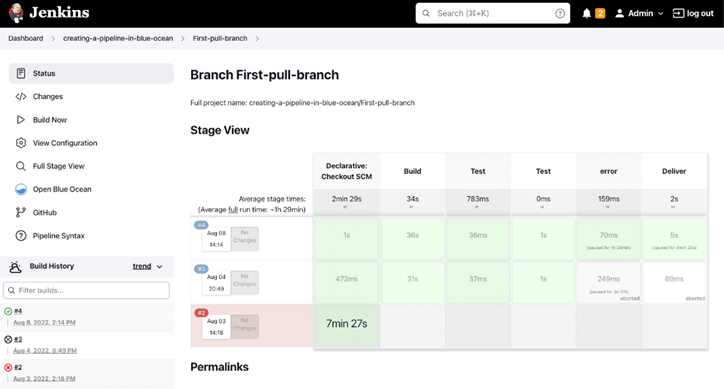


# Chương 2: Tìm hiểu về công cụ Jenkins

## JenKins

### 2.1.1 Jenkins là gì?

* Jenkins là một công cụ tự động mã nguồn mở được viết bằng ngôn ngữ lập trình Java cho phép tích hợp liên tục.
* Jenkins xây dựng và thử nghiệm các dự án phần mềm của người dùng, điều này liên tục giúp các nhà phát triển tích hợp các thay đổi vào dự án dễ dàng hơn và giúp người dùng dễ dàng có được bản dựng mới hơn.
* Jenkins cũng cho phép người dùng liên tục cung cấp phần mềm của mình bằng cách tích hợp với một số lượng lớn công nghệ thử nghiệm và triển khai.



*Hình 2.1.1 Jenkins*

### 2.1.2 Một số tính năng quan trọng của Jenkins?

* CI/CD: Jenkins cho phép tích hợp mã nguồn từ các kho chứa khác nhau như Git, Subversion và nhiều hơn nữa. Khi có thay đổi trong mã nguồn, Jenkins tự động kích hoạt các quy trình xây dựng, kiểm thử và triển khai để đảm bảo mã luôn ổn định và sẵn sàng cho triển khai.
* Quy trình xây dựng (Build): Jenkins cung cấp các công cụ để tạo và xây dựng dự án phần mềm từ mã nguồn. Nó hỗ trợ nhiều ngôn ngữ lập trình và công cụ xây dựng như Maven, Gradle, Ant và nhiều hơn nữa.
* Kiểm thử (Testing): Jenkins cho phép tích hợp các công cụ kiểm thử tự động, bao gồm kiểm thử đơn vị (unit testing), kiểm thử tích hợp (integration testing) và kiểm thử hệ thống (system testing). Bằng cách này, Jenkins giúp đảm bảo chất lượng mã nguồn và phát hiện các lỗi sớm nhất có thể.
* Triển khai (Deployment): Jenkins hỗ trợ việc triển khai ứng dụng lên các môi trường khác nhau như máy chủ thử nghiệm (staging server) hoặc môi trường sản phẩm (production environment). Bằng cách tự động hóa quy trình triển khai, Jenkins giúp giảm thiểu những sai sót do con người gây ra trong quá trình triển khai.
* Plugin: Jenkins có một hệ thống plugin phong phú cho phép mở rộng chức năng của nó. Các plugin cho phép tích hợp với các công cụ phát triển phổ biến như SonarQube, Jira, Docker, Kubernetes và nhiều công cụ khác.
* Giao diện người dùng: Jenkins cung cấp giao diện web đơn giản và dễ sử dụng để quản lý quy trình CI/CD. Người dùng có thể tạo, cấu hình và giám sát các công việc (jobs).

### 2.1.3 Ưu và nhược điểm của việc sử dung Jenkins

* + - Ưu điểm:
* Jenkins là một công cụ mã nguồn mở miễn phí.
* Jenkins không yêu cầu cài đặt hoặc thành phần bổ sung. Có nghĩa là nó rất dễ được cài đặt và cấu hình.
* Jenkins hỗ trọ 1000 plugin trở lên để giảm bớt công việc của bạn. Nếu một plugin không tồn tại, bạn có thể viết kịch bản cho nó và chia sẻ với cộng đồng.
* Jenkins là nền tảng độc lập. Nó có sẵn cho tất cả các nền tảng và hệ điều hành khác nhau. Giống như OS X, Windows hay Linux.
* Hỗ trợ dễ dàng, vì nó là mã nguồn mở và được sử dụng rộng rãi.
* Jenkins cũng hỗ trợ kiến trúc dựa trên đám mây để chúng tôi có thể triển khai Jenkins trên các nền tảng dựa trên đám mây.
  + - Nhược điểm:
* Giao diện của nó đã lỗi thời và không thân thiện với người dùng so với xu hướng giao diện người dùng hiện tại.
* Không dễ bảo trì vì nó chạy trên máy chủ và yêu cầu một số kỹ năng với tư cách là quản trị viên máy chủ để theo dõi hoạt động của nó.
* CI thường xuyên bị hỏng do một số thay đổi cài đặt nhỏ. CI sẽ bị tạm dừng và do đó cần có sự chú ý của nhóm nhà phát triển.

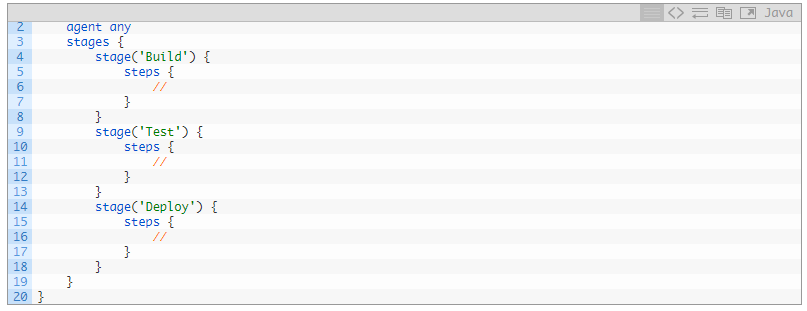
## Pipeline

### 2.2.1 Jenkins Pipeline là gì?

* Jenkins Pipeline được định nghĩa là một bộ plugin giúp hỗ trợ việc triển khai cũng như tích hợp CD dựa theo kiểu pipeline. Quá trình này bao gồm: Tập hợp các quy trình với những tác vụ có liên quan với nhau và được thực hiện theo một cách tuần tự dựa theo thứ tự đã được định sẵn. Khi đó, Jenkins Pipeline sẽ được khai báo ngay trên một tập tin được gọi với tên Jenkinsfile.
* Một pipeline trong Jenkins được viết dưới dạng mã hoặc tệp đặc tả (text-based script) được lưu trữ trong hệ thống quản lý phiên bản (VCS) như Git. Pipeline script có thể được viết bằng nhiều ngôn ngữ, như Groovy Script hoặc Declarative Pipeline Syntax. Các công việc trong pipeline được chia thành các bước (steps) và được sắp xếp thành các giai đoạn (stages).
* Một số khái niệm cơ bản về pipeline trong Jenkins:
* Node: Đại diện cho một máy chủ hoặc môi trường nơi các bước trong pipeline sẽ được thực thi. Jenkins hỗ trợ nhiều loại node như agent-based nodes, Docker containers, hoặc Kubernetes pods.
* Stage: Đại diện cho một phần của quy trình triển khai. Ví dụ: Build, Test, Deploy. Mỗi giai đoạn (stage) có thể có một hoặc nhiều bước (steps).
* Step: Đại diện cho một công việc cụ thể trong pipeline. Ví dụ: biên dịch mã nguồn, chạy các test case, triển khai ứng dụng. Mỗi bước (step) có thể là một lệnh shell, một plugin Jenkins, hoặc một lệnh bên ngoài khác.
* Declarative Pipeline: Là một cú pháp (syntax) đơn giản và có cấu trúc cho việc xây dựng pipeline. Nó sử dụng khối lệnh (block directive) để mô tả các giai đoạn và bước trong pipeline.
* Scripted Pipeline: Là một cú pháp mạnh mẽ hơn cho việc xây dựng pipeline, sử dụng ngôn ngữ Groovy Script. Nó cung cấp một số lợi ích như điều kiện, vòng lặp, và khả năng tái sử dụng mã.
* Triggers: Triggers là cơ chế để kích hoạt pipeline. Jenkins hỗ trợ nhiều loại trigger như Trigger theo lịch, trigger khi có sự thay đổi trên VCS (ví dụ: Git commit), trigger khi có yêu cầu từ người dùng, hoặc trigger dựa trên sự kiện từ hệ thống bên ngoài.
* Parameters: Pipeline có thể chứa các tham số (parameters) để cho phép người dùng điều chỉnh các cài đặt trong quy trình triển khai. Các tham số có thể là giá trị đầu vào như chuỗi, số, boolean hoặc danh sách các giá trị.
* Parallelization: Pipeline trong Jenkins cho phép thực hiện các bước trong giai đoạn (stage) cùng một lúc, đồng thời (parallel). Điều này giúp tăng hiệu suất và giảm thời gian chờ đợi trong quy trình triển khai.
* Error Handling: Pipeline hỗ trợ xử lý lỗi và khôi phục (error handling and recovery). Bằng cách sử dụng các câu lệnh điều kiện và khối try-catch, bạn có thể xử lý các tình huống lỗi, thực hiện các hành động phù hợp và tiếp tục quy trình triển khai.
* Integration and Extensibility: Jenkins pipeline có thể tích hợp với nhiều công cụ và dịch vụ khác nhau thông qua các plugin. Bằng cách sử dụng các plugin, bạn có thể tích hợp với hệ thống kiểm thử tự động, công cụ quản lý phiên bản, công cụ quản lý cấu hình, và nhiều hơn nữa.
* Visualization and Monitoring: Jenkins cung cấp giao diện đồ họa để theo dõi quy trình triển khai, hiển thị tiến độ, kết quả và lịch sử chạy của pipeline. Bạn có thể theo dõi và kiểm soát quy trình triển khai từ giao diện người dùng hoặc qua các trình duyệt web.

### 2.2.2 Cách để tạo một Pipeline trong Jenkins

1. Mở giao diện Jenkins trên trình duyệt web bằng cách truy cập vào địa chỉ http://localhost:8080 (hoặc địa chỉ tương ứng của Jenkins server).
2. Trên giao diện chính của Jenkins, chọn "New Item" hoặc "Create new jobs" để tạo một pipeline mới.
3. Đặt tên cho pipeline và chọn kiểu pipeline mà bạn muốn tạo (Declarative hoặc Scripted). Sau đó, bấm "OK" để tiếp tục.
4. Trên trang cấu hình pipeline, bạn sẽ thấy các trường để cấu hình pipeline script. Tùy thuộc vào kiểu pipeline mà bạn chọn, bạn sẽ thấy các trường khác nhau để điền thông tin.
5. Nếu bạn chọn Declarative Pipeline, bạn sẽ thấy một trình chỉnh sửa văn bản để nhập mã pipeline. Bạn có thể sử dụng các khối lệnh như pipeline, stage, steps, và script để xác định các giai đoạn và bước trong pipeline.



1. Nếu bạn chọn Scripted Pipeline, bạn sẽ thấy một trình chỉnh sửa văn bản để nhập mã pipeline bằng ngôn ngữ Groovy Script.



1. Nhập mã pipeline vào trường cấu hình tương ứng. Bạn có thể sử dụng các bước có sẵn, như checkout, sh (shell script), docker, hoặc các plugin Jenkins khác để thực hiện các công việc trong pipeline.
2. Sau khi bạn hoàn thành cấu hình pipeline, bấm "Save" hoặc "Apply" để lưu cấu hình và tạo pipeline..

Lưu ý rằng để thực hiện các bước trong pipeline, cần có các agent (node) được cấu hình và sẵn sàng để thực thi công việc. Ta cũng có thể tùy chỉnh các trigger và tiến hành cấu hình bổ sung như xử lý lỗi, tích hợp với công cụ và dịch vụ khác, và theo dõi tiến trình pipeline trong Jenkins.

## 2.3 JenKinsFile

### 2.3.1 JenKinsFile là gì?

* + JenkinsFile là một phần quan trọng của Jenkins Pipeline, một công cụ quản lý dự án phổ biến trong quy trình phát triển phần mềm liên tục. Nó là một tệp script được viết bằng ngôn ngữ Groovy, cho phép bạn mô tả và tự động hóa các bước trong quy trình CI/CD của bạn.
  + JenkinsFile cho phép bạn xây dựng quy trình CI/CD linh hoạt và tuỳ chỉnh theo yêu cầu cụ thể của dự án. Bằng cách sử dụng ngôn ngữ Groovy, bạn có thể viết mã để định nghĩa các bước, giai đoạn và các yêu cầu của quy trình xây dựng và triển khai.
* Một JenkinsFile thường bao gồm các thành phần sau:
* Directive: Đây là khối đầu tiên trong JenkinsFile và được sử dụng để khai báo loại Pipeline mà bạn đang xây dựng. Có hai loại chính là "pipeline" và "multibranch pipeline".
* Environment: Khối này cho phép bạn khai báo các biến môi trường cần thiết cho quy trình xây dựng và triển khai. Điều này có thể bao gồm biến môi trường như đường dẫn thực thi, thông tin chứng chỉ hoặc bất kỳ biến nào khác cần thiết.
* Stages: Đây là phần quan trọng nhất của JenkinsFile. Bạn có thể định nghĩa các giai đoạn (stages) trong quy trình CI/CD của bạn và mô tả các bước cần thiết trong mỗi giai đoạn. Ví dụ, một giai đoạn có thể là "Build" và chứa các bước như kiểm tra mã nguồn, biên dịch và đóng gói ứng dụng.
* Steps: Trong mỗi giai đoạn, bạn có thể định nghĩa các bước cụ thể (steps) để thực hiện công việc. Các bước này có thể là các lệnh shell, gọi các công cụ xây dựng, chạy các bài kiểm tra hoặc triển khai ứng dụng.
* Post: Đây là khối được sử dụng để định nghĩa các bước được thực thi sau khi hoàn thành quy trình CI/CD. Ví dụ, bạn có thể thông báo kết quả, gửi email thông báo, lưu trữ bản sao kết quả xây dựng hoặc triển khai, và nhiều hơn nữa.

### 2.3.2 Ưu và nhược điểm của JenkinsFile

* **Ưu Điểm:**
* Quản lý linh hoạt: JenkinsFile cho phép bạn mô tả và quản lý các bước trong quy trình CI/CD của bạn theo cách tùy chỉnh và linh hoạt. Bạn có toàn quyền kiểm soát quy trình và có thể tùy chỉnh nó để phù hợp với yêu cầu cụ thể của dự án.
* Tích hợp dễ dàng: JenkinsFile tích hợp tốt với Jenkins và hệ sinh thái công cụ phát triển phần mềm khác. Nó cho phép bạn sử dụng các plugin và công cụ bổ sung để mở rộng tính năng và khả năng của quy trình CI/CD của bạn.
* Phiên bản hóa quy trình: JenkinsFile có thể lưu trữ trong kho mã nguồn và quản lý như một phần của mã nguồn. Điều này cho phép bạn duy trì lịch sử và phiên bản hóa quy trình xây dựng và triển khai, giúp kiểm soát và theo dõi các thay đổi theo thời gian.
* Tái sử dụng: Bạn có thể tái sử dụng JenkinsFile cho nhiều dự án khác nhau hoặc trong các môi trường khác nhau. Điều này giúp tiết kiệm thời gian và công sức trong việc xây dựng và duy trì quy trình CI/CD.
* Kiểm soát mã nguồn: JenkinsFile là mã nguồn được viết bằng ngôn ngữ Groovy, cho phép bạn kiểm soát mã nguồn và áp dụng các quy tắc kiểm tra mã tự động. Điều này giúp đảm bảo chất lượng mã và giúp phát hiện sớm các vấn đề tiềm ẩn trong quy trình CI/CD.
* **Nhược điểm:**
* Học phí: Sử dụng JenkinsFile đòi hỏi bạn có kiến thức về ngôn ngữ Groovy và cách Jenkins hoạt động. Điều này có thể đòi hỏi thời gian và nỗ lực để học và hiểu cú pháp và quy tắc của JenkinsFile.
* Phức tạp: Quy trình xây dựng và triển khai phức tạp có thể dẫn đến một JenkinsFile lớn và khó hiểu. Việc duy trì và quản lý JenkinsFile có thể trở nên phức tạp và tốn thời gian nếu không được tổ chức và cấu trúc tốt.
* Tích hợp hạn chế: Mặc dù JenkinsFile tích hợp tốt với Jenkins, việc tích hợp với các công cụ khác có thể hạn chế. Một số công cụ và hệ thống không cung cấp sẵn tích hợp cho JenkinsFile, và bạn có thể phải tự tìm cách tích hợp chúng.
* Sự phụ thuộc vào Jenkins: Sử dụng JenkinsFile đồng nghĩa với việc phụ thuộc vào Jenkins và hệ thống Jenkins. Nếu bạn quyết định chuyển đổi sang công cụ CI/CD khác, bạn có thể cần phải thay đổi và điều chỉnh lại quy trình CI/CD của mình.

### 2.3.3 Cách để tạo 1 JenkinsFile mới

* **Bước 1: Mở Jenkins và tạo một Job mới.**
* Đăng nhập vào giao diện Jenkins và chọn "New Item" để tạo một Job mới.
* **Bước 2: Chọn loại Job là "Pipeline".**
* Trong trang tạo mới Job, chọn "Pipeline" làm loại Job.
* **Bước 3: Xác định nguồn mã nguồn và quy trình Pipeline.**
* Trong phần "Pipeline", bạn có thể chọn "Pipeline script" để viết JenkinsFile trực tiếp hoặc chọn "Pipeline script from SCM" nếu bạn muốn lưu trữ JenkinsFile trong một kho mã nguồn và cấu hình Jenkins để nó tự động lấy JenkinsFile từ kho mã nguồn.
* **Bước 4: Viết JenkinsFile.**
* Nếu bạn chọn "Pipeline script", bạn có thể viết JenkinsFile trực tiếp trong trường "Script". Sử dụng cú pháp ngôn ngữ Groovy để định nghĩa các giai đoạn, bước và các yêu cầu của quy trình CI/CD. Ví dụ:



* Nếu bạn chọn "Pipeline script from SCM", bạn sẽ cần cấu hình kết nối với kho mã nguồn của bạn (ví dụ: Git) và chỉ định đường dẫn đến JenkinsFile trong kho mã nguồn.
* **Bước 5: Lưu và chạy Job.**
* Sau khi bạn đã hoàn thành viết JenkinsFile, lưu lại cấu hình Job và chạy Job để bắt đầu quy trình CI/CD của bạn dựa trên JenkinsFile.

## 2.4 Plugin

### 2.4.1 Plugin là gì?

* Plugin là một công cụ hỗ trợ cho sự phát triển website của [wordpress](https://news.timviec.com.vn/wordpress-la-gi-72217.html). Mỗi loại plugin sẽ có một công dụng cũng như chức năng riêng biệt. Có thể hiểu Plugin sẽ giống như một sự hỗ trợ tính năng cho website cài đặt với wordpress.
* Công cụ plugin là một thành phần không thể thiếu được lập trình riêng dựa trên các API và những hàm mở có sẵn của WordPress. Cứ một plugin sẽ được cài đặt thêm để có thể tạo ra một tính năng nào đó mà trên wordpress không sẵn có. Phần mềm Plugin được tạo ra bởi một hãng thứ 3 dựa theo tiêu chuẩn công nghệ thông tin. Mục đích nhằm giúp người dùng tạo ra được những kiểu hiệu ứng đặc biệt.

### 2.4.2 Ưu nhược điểm của Plugin

* Ưu điểm của việc sử dụng plugin trên website:
* Tiết kiệm thời gian: Sử dụng plugin giúp người dùng tiết kiệm thời gian vì không cần phải viết mã code từ đầu. Các plugin có thể được cài đặt và sử dụng ngay lập tức.
* Tránh sai sót: Plugin đã được phát triển và kiểm tra trước, giúp tránh các lỗi thường gặp trong việc viết code. Điều này đảm bảo tính ổn định và an toàn cho website.
* Dễ dàng cài đặt và gỡ bỏ: Người dùng có thể dễ dàng cài đặt và gỡ bỏ plugin theo nhu cầu của họ mà không ảnh hưởng đến hoạt động của website. Điều này mang lại tính linh hoạt và sự thay đổi nhanh chóng.
* Nhược điểm của việc sử dụng plugin trên website:
* Tốc độ tải trang chậm: Cài đặt quá nhiều plugin có thể làm tăng tải trọng và làm chậm tốc độ tải trang của website. Việc lựa chọn và sử dụng các plugin cần được cân nhắc cẩn thận để tránh ảnh hưởng đến trải nghiệm người dùng.
* Xung đột và không tương thích: Một số plugin có thể xung đột với nhau hoặc không tương thích với phiên bản WordPress hoặc các thành phần khác trên website. Điều này có thể gây ra lỗi và vấn đề hoạt động không đúng của website.
* Bảo mật và tin cậy: Việc sử dụng plugin từ các nguồn không đáng tin cậy hoặc không cập nhật thường xuyên có thể gây rủi ro bảo mật cho website. Người dùng cần chú ý đến tính tin cậy và bảo mật của plugin trước khi cài đặt và sử dụng.

### 2.4.3 Các nhóm Plugin

* Dựa trên tính năng, Plugin được phân làm 3 nhóm cơ bản:
* **Nhóm Plugin tối ưu:** Những khía cạnh được cải thiện bao gồm hình ảnh, SEO, bài viết. Bên cạnh đó, loại này hỗ trợ tốc độ tải trang của website.
* **Nhóm Plugin bảo mật:** Tăng tính an toàn cho website. Quản trị web tránh được tình trạng bị hack hoặc đánh sập bởi bất kỳ nhóm hacker nào.
* **Nhóm Plugin lưu trữ:** Có chức năng Data Backup. Cài đặt Plugin kiểu này giúp lưu trữ, khôi phục dữ liệu một cách dễ dàng hơn bao giờ hết.
* Có thể kết hợp cài đặt Plugin đến từ nhiều nhóm khác nhau để phục vụ cho công việc. Chúng vẫn đảm bảo hoạt động bình thường, không xảy ra sự mâu thuẫn nào.

### 2.4.4 Bảy Plugin phổ biến

* Hiện nay có hàng chục nghìn Plugin nhưng không phải loại nào cũng cần sử dụng. Dưới đây là các công cụ cần thiết nhất cho website WordPress.

|  |  |
| --- | --- |
| **Loại Plugin** | **Tính năng** |
| SEO by Yoast | Giải pháp dành cho việc tối ưu hóa website chuẩn SEO. |
| TinyMCE Advanced | Plugin tổng hợp các công cụ soạn thảo văn bản cho website, rất hữu ích cho sáng tạo nội dụng. |
| kk Star Ratings | **Đánh giá sao cho bài viết**, phục vụ cho mục đích tăng độ uy tín website với công cụ tìm kiếm Google, Gây dựng niềm tin. |
| Google XML Sitemaps | Plugin sơ đồ trang web. Việc sử dụng giúp công cụ tìm kiếm Indexing website nhanh chóng và tốt hơn. |
| RDFa Breadcrumb | Plugin hỗ trợ tạo thẻ điều hướng trong giao diện người dùng. |
| WooCommerce | Được xem là Plugin WordPress tối ưu nhất cho các website thương mại điện tử. Tính năng nổi bật bao gồm: Trang giỏ hàng, thanh toán an toàn với thẻ tín dụng, tùy chọn các phương án giao hàng,… |
| WP Rocket | Hỗ trợ tăng tốc website. Một trong những tính năng nổi bật là Lazy Load đem tới sự mượt mà hơn trong cách vận hành web. |

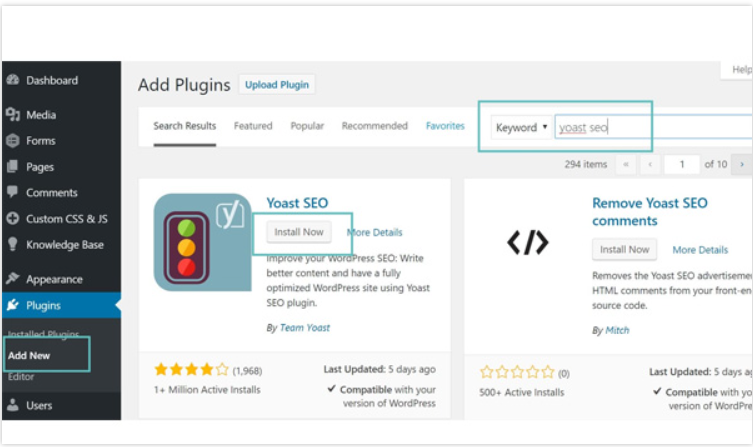
### 2.4.5 Ba cách cài đặt Plugin

* Cách 1: Cài đặt bằng Dashboard

Đây được đánh giá là**cách cài đặt Plugin** đơn giản nhất. Mục đích là **sử dụng công cụ miễn phí, có sẵn trong thư viện** của nền tảng này. Để thành công, bạn chỉ cần làm theo bốn bước như sau:

Bước 1: Đăng nhập vào trang **Dashboard** của WordPress. Sau đó chọn menu Plugin -> Add New.

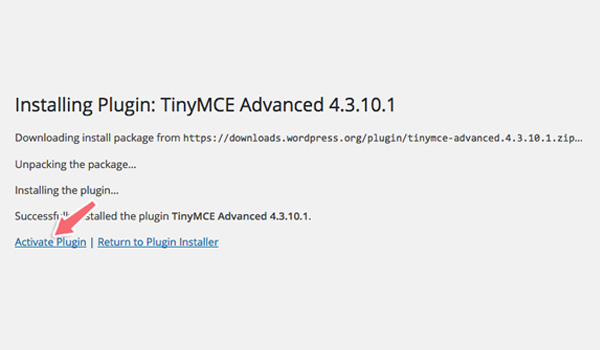
Bước 2: Gõ tên plugin cần tìm vào ô tìm kiếm rồi nhấn phím enter.



Hình 2.4.5.1: Gõ plugin muốn cài đặt vào ô tìm kiếm

Bước 3: Lúc này, hệ thống sẽ trả về một loạt danh sách các plugin liên quan. Chọn plugin bạn muốn và nhấn “Install Now” để cài đặt. Sau đó, WordPress sẽ tải plugin về và cài đặt tự động.

Bước 4: Sau khi hoàn thành giai đoạn cài đặt, bạn sẽ nhận được một thông báo yêu cầu kích hoạt plugin

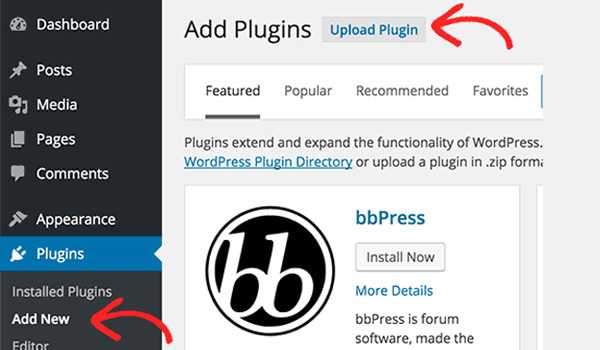


Hình 2.4.5.2: Thông báo yêu cầu kích hoạt plugin

Hãy click vào “**Activate Plugin**” để kích hoạt plugin.

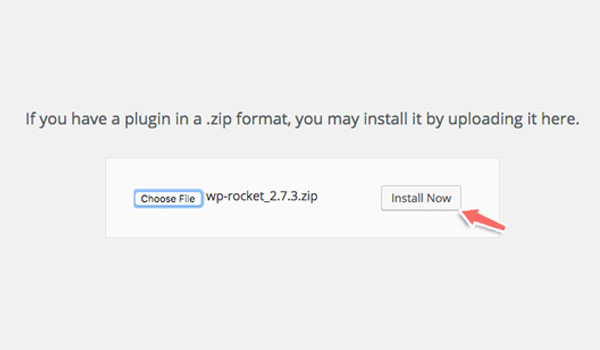
* **Cách 2: Cài bằng cách upload**
* Bước 1: Download plugin muốn cài đặt với định dạng file là .zip.
* Bước 2: Đăng nhập vào trang **Dashboard** của **WordPress**. Tiếp đến chọn **Plugin** -> Add New.
* Bước 3: Nhấn vào nút **Upload Plugin.**

Hệ thống sẽ hiển thị một mục thông báo để bạn upload file.



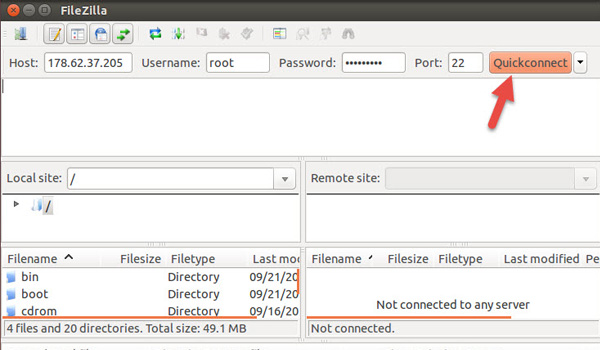
Hình 2.4.5.3: Nhấn vào nút Upload Plugin để tải file lên WordPress

Nhấn vào nút “**Choose File**” để tải plugin từ máy tính lên WordPress. Sau đó nhấn “**Install Now**” để cài đặt.



Hình 2.4.5.4: Click “Choose file” chọn file plugin từ máy tính

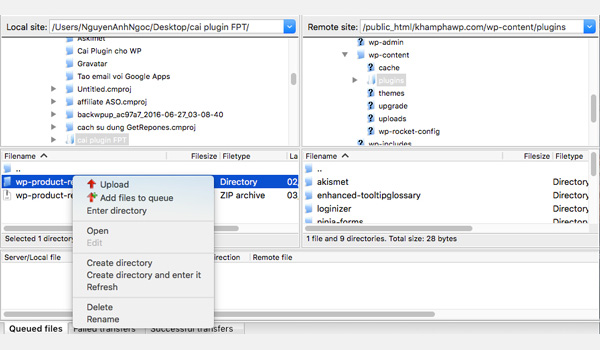
* Bước 4: Sau khi cài đặt xong, WordPress sẽ hiển thị một thông báo cho người dùng trên màn hình. Bấm “**Activate Plugin**” để kích hoạt plugin.
* **Cách 3: Cài đặt bằng giao thức FTP**
* Bước 1: [Tải phần mềm FileZilla](https://filezilla-project.org/download.php?platform=win64) về máy và tiến hành cài đặt.
* Bước 2: Giải nén file plugin .zip thành 1 thư mục cùng tên. Thư mục này sẽ là dữ liệu được bạn upload lên **hosting**.
* Bước 3: Mở phần mềm FileZilla, đăng nhập tên miền của hosting, user, password và chọn cổng kết nối 21.



Hình 2.4.5.5: Kết nối máy tính với hosting bằng cổng kết nối 21

* Bước 4: Khi đã kết nối máy tính với hosting thành công, phần mềm hiển thị 2 cửa sổ: Local site và Remote site.

Hãy tìm và chọn thư mục wp-content/plugins ở cửa sổ bên phải (Remote site).



Hình 2.4.5.6: Giao diện phần mềm hiển thị 2 cửa sổ: Local site và Remote site

* Bước 5: Tại cửa sổ bên trái (Local site), hãy tìm tới thư mục plugin cần tải lên. Sau đó, click chuột phải và chọn “Upload”.
* Bước 6:Hệ thống sẽ tự động tải plugin của bạn lên WordPress.

Sau khi quá tình này kết thúc, hãy vào trang quản trị của WordPress và tìm kiếm tên plugin mới được cài đặt.

Nhấn “Activate” để kích hoạt và bắt đầu sử dụng plugin.

## 2.5 Integration

### 2.5.1 Integration là gì?

* Tích hợp liên tục (Continuous Integration) là phương pháp phát triển phần mềm đòi hỏi các thành viên trong nhóm tích hợp công việc thường xuyên. Mỗi ngày, các thành viên đều phải theo dõi và phát triển công việc của họ ít nhất 1 lần. Việc này sẽ được một nhóm khác kiểm tra tự động, nhóm này sẽ tiền hành kiểm thử truy hồi để phát hiện lỗi nhanh nhất có thể. Cả nhóm thấy rằng phương pháp tiếp cận này giúp giảm bớt vấn đề về tích hợp hơn và cho phép phát triển phần mềm gắn kết nhanh hơn.

### 2.5.2 Minh họa

A picture containing text, screenshot, diagram

Description automatically generated

Hình 2.5.2: Ảnh minh họa

* Các developer thực hiện các ticket và đẩy code lên thư mực nơi sẽ lưu giữ code của Project.
* Hệ thống CI sẽ lắng nghe và thực hiện kéo code về (server CI), build rồi test code.
* Trong trường hợp build, test thất bại, các thành viên trong team sẽ nhìn thấy ngay được lỗi phát sinh và các developer sẽ chịu trách nhiệm sửa lỗi và đẩy code sửa lỗi đó lên repo.

### 2.5.3 Các đặc điểm thực tiễn của CI

* + Quản lý phiên bản (source control and version control).
  + Tự động build bao gồm test.
  + Đôi ngũ phát triển thường xuyên chuyển (commit) source về nơi lưu trữ chính (mainline).
  + Mỗi khi code có thay đổi sẽ build lại (mainline) thông qua build server.
  + Báo lỗi cho người lập trình gây lỗi và quản lý dự án.
  + Phát hành phiên bản hoàn chỉnh cho khách hàng khi không còn lỗi.
  + Tự động phân phối phiên bản mới đến khách hàng.
  + Mọi người có thể nhìn thấy những gì đã xảy ra (thay đổi, lỗi…) để xem xét và giải quyết kịp thời.

### 2.5.4 Ưu điểm và nhược điểm của Continuous Integration

* + Ưu điểm:
    - Giảm thiểu rủi ro do lỗi được phát hiện sớm.
    - Giảm thiểu sự lặp lại cho các quá trình.
    - Tạo phần mềm có giá trị sử dụng sớm nhất có thể và sẳn sàng triểnkhai mọi lúc mọi nơi.
    - Cung cấp cái nhìn xuyên suốt tổng quan và cụ thể cho từng giai đoạn.
    - Nâng cao kỹ năng của đội ngũ nhân viên phát triển phần mềm.
    - Cải thiện chất lượng phần mềm.
  + Nhược điểm:
    - Cần thời gian thiết lập hệ thống ban đầu.
    - Đòi hỏi quản lý dự án, người lập trình, người kiểm định phải am hiểu mô hình phát triển phần mềm Agile, hệ thống tích hợp CI, cách sử dụng các công cụ hỗ trợ cho Agile và CI.
    - Chi phí thiết bị phần cứng (các server cho CI).

### 2.5.5 Các công cụ sử dụng

* + Code Repositories: SVN, Mercurial, Git, Visual Source Safe…
  + Continuous Build Systems: Jenkins, Bamboo, Cruise, Hudson, …
  + Test Unit Frameworks: JUnit, Cucumber, CppUnit …
  + Test report website: Mantis, TestLink, Redmine…
  + Artifact Repositories: Nexus, Artifactory, Archiva …

### 2.5.6 Cài đặt và cấu hình tích hợp (integration) trong Jenkins

* + Cài đặt Jenkins: Đầu tiên, cài đặt Jenkins trên máy chủ hoặc hệ thống của bạn bằng cách tải xuống phiên bản phù hợp từ trang web chính thức của Jenkins và làm theo hướng dẫn cài đặt đi kèm.
  + Cài đặt plugin: Mở giao diện quản lý Jenkins và điều hướng đến phần "Manage Jenkins" và sau đó chọn "Manage Plugins". Trong tab "Available" (Có sẵn), tìm kiếm và chọn các plugin liên quan đến công cụ kiểm thử tích hợp mà bạn muốn tích hợp với Jenkins (ví dụ: Selenium, JUnit, TestNG, Cucumber, SoapUI). Sau đó, nhấn nút "Install" (Cài đặt) và chờ quá trình cài đặt hoàn thành.
  + Tạo một công việc (job) Jenkins: Quay trở lại trang chính của Jenkins và nhấp vào "New Item" (Mục mới) để tạo một công việc mới. Đặt tên cho công việc và chọn loại công việc phù hợp, ví dụ: "Freestyle project" (Dự án theo phong cách tự do) hoặc "Pipeline" (Luồng công việc). Nhấp vào "OK" để tiếp tục.
  + Cấu hình công việc Jenkins: Trong cấu hình công việc, bạn sẽ thấy các phần cấu hình khác nhau dựa trên loại công việc bạn chọn. Để tích hợp công cụ kiểm thử tích hợp, bạn cần cấu hình các phần liên quan, ví dụ: trong phần "Build" (Xây dựng), thêm các bước xây dựng kiểm thử tích hợp sử dụng các công cụ như Selenium, JUnit, Cucumber, hoặc SoapUI. Các cài đặt cụ thể sẽ phụ thuộc vào công cụ và plugin mà bạn đang sử dụng.
  + Lưu và chạy công việc: Sau khi hoàn thành cấu hình công việc, nhấp vào "Save" (Lưu) để lưu cấu hình và quay trở lại trang chính của Jenkins. Từ đó, bạn có thể nhấp vào công việc và chọn "Build Now" (Xây dựng ngay) để chạy công việc và thực hiện kiểm thử tích hợp.

# Chương 3: Ứng dụng của Jenkins

## 3.1 Continuous Integration (CI)

### 3.1.1 Khái niệm

Continuous Integration (CI) là một phương pháp phát triển phần mềm trong đó các thành viên trong nhóm phát triển tích hợp và kiểm tra mã nguồn của họ vào một kho mã chung thường xuyên, thường là hàng ngày. Mục tiêu của CI là giảm thiểu rủi ro và xung đột trong quá trình tích hợp mã nguồn, tạo điều kiện cho việc phát triển phần mềm nhanh chóng, đáng tin cậy và linh hoạt hơn.

### 3.1.2 Các bước thực hiện CI trong Jenkins

* Quản lý mã nguồn: Jenkins kết nối với hệ thống quản lý mã nguồn như Git, SVN, Mercurial để lấy mã nguồn từ kho mã.
* Xây dựng (Build): Jenkins thực hiện quá trình xây dựng tự động từ mã nguồn. Điều này có thể bao gồm biên dịch mã, tạo các gói cài đặt hoặc mã máy, và/hoặc thực hiện các bước tiền xử lý khác cần thiết.
* Kiểm thử (Testing): Sau khi quá trình xây dựng hoàn thành, Jenkins thực hiện các bước kiểm thử tự động để đảm bảo chất lượng của ứng dụng. Điều này có thể bao gồm kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử chức năng và kiểm thử hiệu năng.
* Báo cáo và phản hồi (Reporting and Feedback): Jenkins tạo ra báo cáo về kết quả xây dựng và kiểm thử. Nhóm phát triển có thể xem các báo cáo này để biết về trạng thái và chất lượng của ứng dụng. Nếu có lỗi hoặc vấn đề, Jenkins có thể thông báo cho nhóm phát triển để giải quyết kịp thời.
* Tích hợp liên tục (Continuous Integration): Jenkins thực hiện quá trình CI bằng cách tự động chạy các bước xây dựng và kiểm thử mỗi khi có sự thay đổi trong mã nguồn. Điều này giúp đảm bảo rằng mã nguồn luôn được kiểm tra và tích hợp một cách liên tục, giảm thiểu rủi ro và xung đột.
* Quy trình CI trong Jenkins tạo ra một luồng công việc tự động mạnh mẽ, giúp giảm thiểu sự phụ thuộc vào công việc thủ công và đảm bảo tính nhất quán và chất lượng của mã nguồn trong suốt quá trình phát triển phần mềm.

Quy trình CI trong Jenkins tạo ra một luồng công việc tự động mạnh mẽ, giúp giảm thiểu sự phụ thuộc vào công việc thủ công và đảm bảo tính nhất quán và chất lượng của mã nguồn trong suốt quá trình phát triển phần mềm.

## 3.2 Automated Builds

### 3.2.1 Khái niệm

Automated Builds là một phần quan trọng trong quy trình Continuous Integration (CI), và Jenkins cung cấp khả năng tự động hóa quá trình này. Trong Jenkins, Automated Builds được sử dụng để xây dựng ứng dụng tự động từ mã nguồn.

### 3.2.2 Tổng quan về quá trình Automated Builds

* Quản lý mã nguồn: Jenkins kết nối với hệ thống quản lý mã nguồn như Git, SVN, Mercurial để lấy mã nguồn từ kho mã.
* Cấu hình Build: Bạn cần cấu hình Jenkins để biết cách xây dựng ứng dụng từ mã nguồn. Điều này bao gồm các thông số như cách biên dịch, đóng gói, và tạo các gói cài đặt hoặc mã máy.
* Trigger Builds: Jenkins có thể được cấu hình để tự động kích hoạt quá trình xây dựng dựa trên các sự kiện như commit mới vào kho mã hoặc theo lịch trình được đặt trước.
* Xây dựng tự động: Khi quá trình xây dựng được kích hoạt, Jenkins tự động tải mã nguồn từ kho mã và thực hiện các bước xây dựng đã được cấu hình. Điều này có thể bao gồm biên dịch, gộp mã, tạo tệp tin thực thi, đóng gói ứng dụng và các quá trình tiền xử lý khác.
* Báo cáo và phản hồi: Sau khi xây dựng hoàn thành, Jenkins có thể tạo ra báo cáo về quá trình xây dựng và kết quả. Báo cáo này giúp nhóm phát triển đánh giá chất lượng và trạng thái của ứng dụng được xây dựng. Nếu có lỗi hoặc vấn đề, Jenkins có thể thông báo cho nhóm phát triển để họ có thể xử lý kịp thời.
* Quản lý lịch sử xây dựng: Jenkins lưu trữ lịch sử xây dựng, bao gồm thông tin về phiên bản mã nguồn, kết quả xây dựng và báo cáo. Điều này cho phép nhóm phát triển xem lại và kiểm tra các phiên bản xây dựng trước đó và phân tích các vấn đề nếu có.

Quá trình Automated Builds trong Jenkins giúp tự động hóa việc xây dựng ứng dụng từ mã nguồn, giảm thiểu sự phụ thuộc vào công việc thủ công và đảm bảo tính nhất quán và hiệu suất trong quá trình phát triển phần mềm.

## 3.3 Continuous Delivery/Deployment (CD)

### 3.3.1 Khái niệm

Continuous Delivery/Deployment (CD) là một phần quan trọng trong quy trình phát triển phần mềm, và Jenkins cung cấp khả năng tự động hóa quá trình này

### 3.3.2 Tổng quan về Continuous Delivery/Deployment trong Jenkins

* Xây dựng tự động: Jenkins hỗ trợ quá trình xây dựng tự động (Automated Builds) để tự động hóa việc xây dựng ứng dụng từ mã nguồn. Điều này đảm bảo rằng mã nguồn mới nhất được xây dựng và kiểm tra tự động trước khi triển khai.
* Tự động hóa kiểm thử: Jenkins cung cấp khả năng tự động hóa quá trình kiểm thử bằng cách chạy các bộ kiểm thử tự động sau quá trình xây dựng. Điều này giúp đảm bảo tính nhất quán và chất lượng của ứng dụng.
* Quản lý phiên bản: Jenkins tích hợp với hệ thống quản lý phiên bản như Git để quản lý và theo dõi các phiên bản mã nguồn. Điều này cho phép bạn quản lý lịch sử thay đổi, so sánh và chuyển đổi giữa các phiên bản khác nhau của ứng dụng.
* Tự động triển khai: Jenkins hỗ trợ tự động triển khai ứng dụng lên môi trường sản xuất hoặc môi trường thử nghiệm. Bằng cách cấu hình Jenkins để triển khai tự động, bạn có thể đảm bảo rằng phiên bản mới nhất của ứng dụng được triển khai một cách nhất quán và đáng tin cậy.
* Quản lý môi trường triển khai: Jenkins cho phép bạn quản lý các môi trường triển khai khác nhau, bao gồm môi trường thử nghiệm, môi trường staging và môi trường sản xuất. Bạn có thể cấu hình Jenkins để triển khai ứng dụng lên các môi trường này và kiểm soát quá trình triển khai.
* Báo cáo và theo dõi: Jenkins cung cấp các báo cáo và ghi lại quá trình Continuous Delivery/Deployment. Bạn có thể xem các báo cáo về quá trình xây dựng, kiểm thử và triển khai để đảm bảo rằng ứng dụng được phát triển và triển khai một cách thành công và đáng tin cậy.

Jenkins cung cấp khả năng tự động hóa quá trình Continuous Delivery/Deployment, từ xây dựng tự động, kiểm thử tự động đến triển khai tự động. Điều này giúp tăng tính nhất quán, hiệu suất và độ tin cậy trong quá trình phát triển và triển khai phần mềm.

## 3.4 Scheduled Jobs

### 3.4.1 Khái niệm

Scheduled Jobs là khả năng của Jenkins để thực thi công việc (jobs) theo một lịch trình được đặt trước. Đây là một tính năng quan trọng trong Jenkins cho phép bạn tự động thực hiện các tác vụ một cách định kỳ và theo lịch trình cụ thể

### 3.4.2 Tổng quan về Scheduled Jobs trong Jenkins

* Lập lịch công việc: Jenkins cho phép bạn cấu hình lịch trình thực thi cho các công việc. Bạn có thể đặt thời gian, ngày, tuần, tháng hoặc các lịch trình tùy chỉnh khác để chỉ định khi công việc sẽ được thực thi.
* Tự động thực thi: Các công việc được cấu hình với lịch trình sẽ tự động thực thi theo đúng thời gian đã được đặt. Jenkins sẽ kích hoạt công việc vào thời gian đã chỉ định mà không cần sự can thiệp thủ công.
* Cấu hình linh hoạt: Jenkins cung cấp nhiều tùy chọn cấu hình cho các công việc được lập lịch. Bạn có thể chỉ định các tham số đầu vào, kịch bản thực thi, môi trường, yêu cầu đồng thời và nhiều tùy chọn khác để điều chỉnh việc thực hiện công việc.
* Quản lý lịch sử thực thi: Jenkins ghi lại lịch sử thực thi của các công việc đã được lập lịch. Bạn có thể xem các thông tin về thời gian thực thi, kết quả, lỗi và thông báo từ các công việc đã được thực thi trong quá khứ.
* Đồng bộ hóa với các công việc khác: Scheduled Jobs trong Jenkins có thể được đồng bộ hóa với các công việc khác trong quy trình phát triển. Bạn có thể thiết lập các phụ thuộc và luồng công việc để đảm bảo rằng các công việc được thực thi đúng theo thứ tự và theo đúng thời gian đã định.

Tổng quan về Scheduled Jobs trong Jenkins cho phép tự động thực hiện các công việc theo lịch trình, giúp tăng tính đồng nhất, hiệu suất và độ tin cậy trong quy trình phát triển phần mềm.

## 3.5 Integration with Other Tools

### 3.5.1 Khái niệm

Jenkins tích hợp được với nhiều công cụ phát triển và quản lý dự án khác nhau như Git, SVN, Jira, Slack, Docker và nhiều hơn nữa. Điều này giúp tạo ra một quy trình làm việc mượt mà và tự động hơn, cho phép các công cụ khác làm việc cùng nhau để đạt được mục tiêu phát triển ứng dụng.

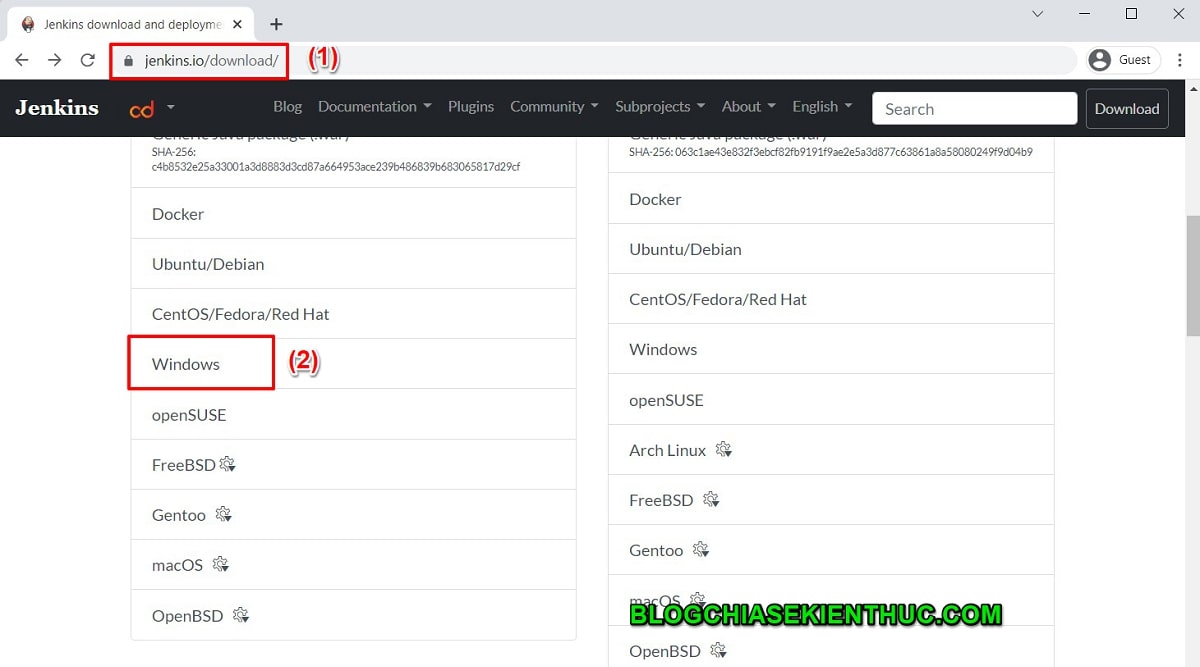
### 3.5.2 Tổng quan về Integration with Other Tools trong Jenkins

* Hệ thống quản lý mã nguồn: Jenkins có thể tích hợp với các hệ thống quản lý mã nguồn phổ biến như Git, SVN, Mercurial. Điều này cho phép Jenkins lấy mã nguồn từ kho mã và thực hiện các công việc xây dựng, kiểm thử và triển khai tự động từ mã nguồn.
* Công cụ xây dựng và quản lý phụ thuộc: Jenkins có thể tích hợp với các công cụ xây dựng như Apache Maven, Gradle, Ant để thực hiện quá trình biên dịch, đóng gói và quản lý phụ thuộc của ứng dụng.
* Công cụ kiểm thử và phân tích: Jenkins có thể tích hợp với các công cụ kiểm thử tự động như JUnit, NUnit, Selenium để chạy các bộ kiểm thử và thu thập kết quả. Ngoài ra, Jenkins cũng có thể tích hợp với các công cụ phân tích mã nguồn như SonarQube để đánh giá chất lượng mã nguồn.
* Công cụ triển khai và quản lý môi trường: Jenkins có thể tích hợp với các công cụ triển khai và quản lý môi trường như Docker, Kubernetes, AWS, Azure để triển khai ứng dụng lên các môi trường sản xuất và quản lý cơ sở hạ tầng.
* Công cụ thông báo và quản lý dự án: Jenkins có thể tích hợp với các công cụ thông báo như Slack, Email để gửi thông báo kết quả xây dựng và các thông báo quan trọng khác cho các thành viên trong nhóm phát triển. Ngoài ra, Jenkins cũng có khả năng tích hợp với các công cụ quản lý dự án như Jira, Trello để ghi lại và theo dõi tiến độ công việc.

Tích hợp với các công cụ khác trong Jenkins giúp tạo ra một quy trình phát triển phần mềm toàn diện và tự động hóa, tăng tính nhất quán và hiệu suất của quá trình phát triển phần mềm.

## 3.6 Cách cài đặt Jenkins

**Bước 1:** Truy cập: [https://www.jenkins.io/download/](https://www.jenkins.io/download/%20). Chọn hệ điều hành Window

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-1.jpg)

*Hình ảnh 3.6.1*

**Bước 2:** Bấm Next để bắt đầu cài đặt.



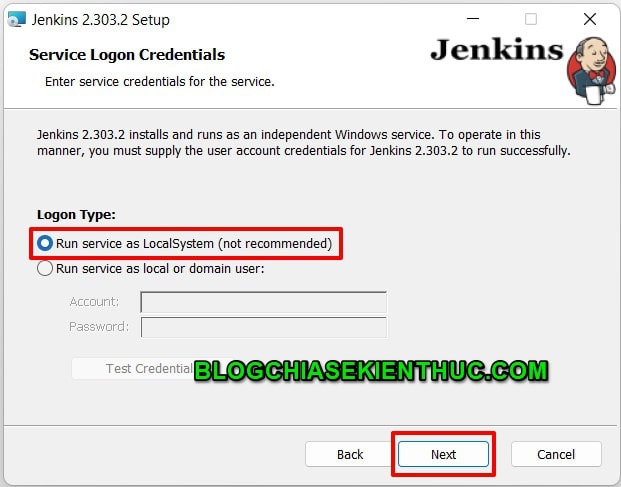
*Hình ảnh 3.6.2*

**Bước 3:** Mặc định Jenkins sẽ được cài đặt tại C:\Program Files\Jenkins\ như hình bên dưới. Nếu muốn thay đổi thì bấm vào Change…



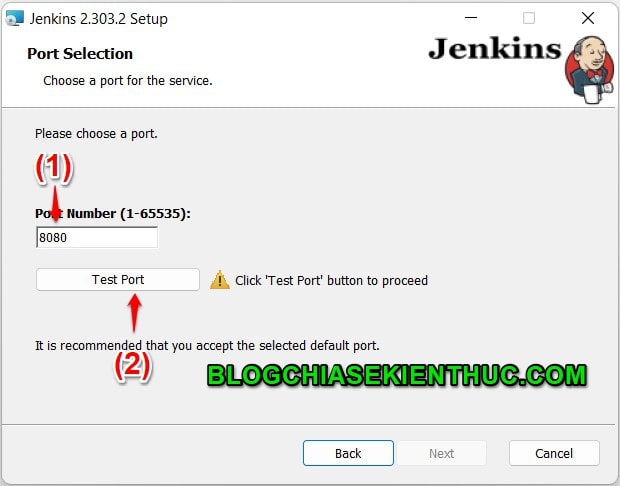
*Hình ảnh 3.6.3*

**Bước 4:** Cài đặt Jenkins trên local



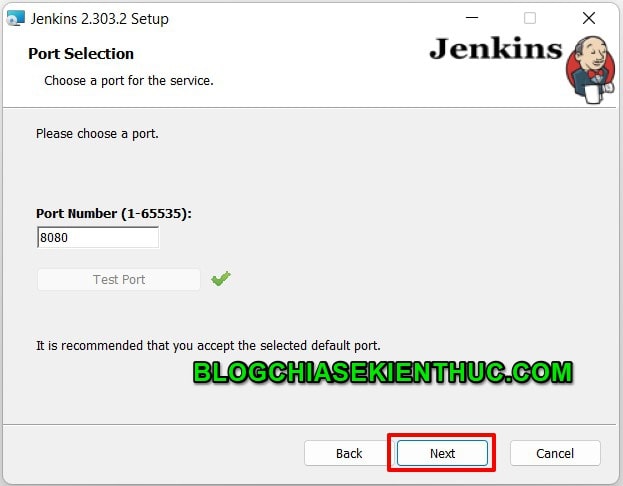
*Hình ảnh 3.6.4*

**Bước 5:** Tiếp theo, chúng ta sẽ chọn Port (mặc định sẽ là 8080).

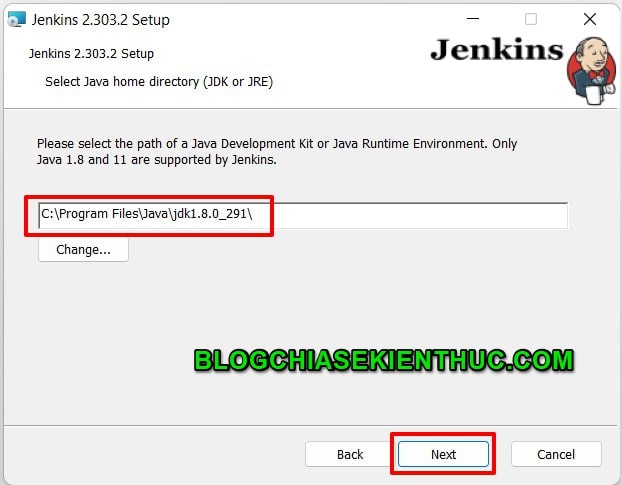


*Hình ảnh 3.6.5*

Nếu port đó sẵn sàng thì kết quả sẽ như hình bên dưới, các bạn bấm Next để tiếp tục.



*Hình ảnh 3.6.6*

**Bước 6:** Do Jenkins được phát triển bằng ngôn ngữ lập trình Java. Nên phải cài đặt Java và trỏ vào vị trí như hình bên dưới

*Hình ảnh 3.6.7*

Bấm Next để tiếp tục.

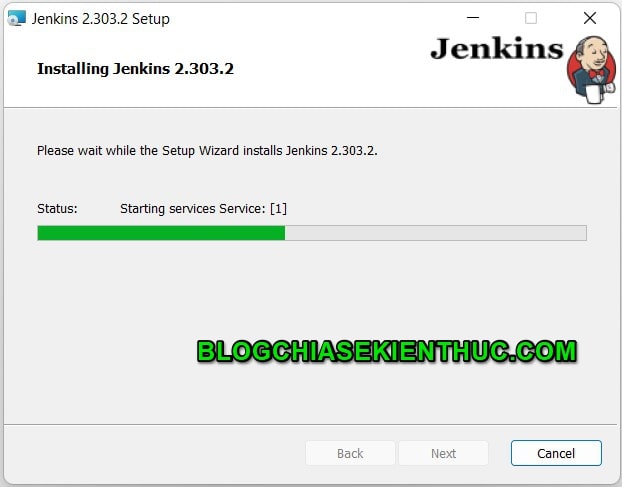


*Hình ảnh 3.6.8*

Bấm Install để bắt đầu quá trình cài đặt.

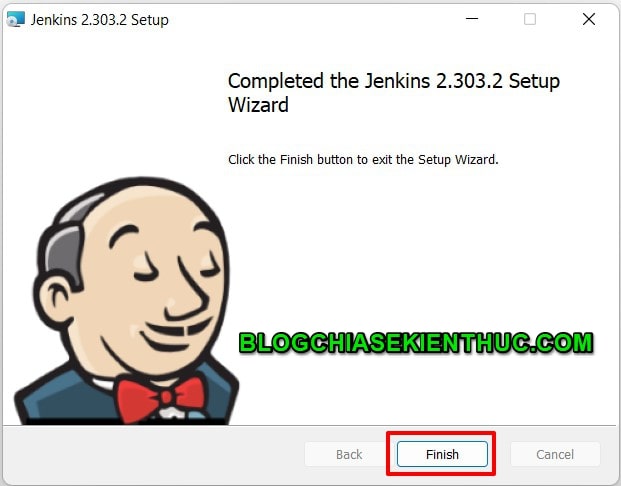


*Hình ảnh 3.6.9*



*Hình ảnh 3.6.10*

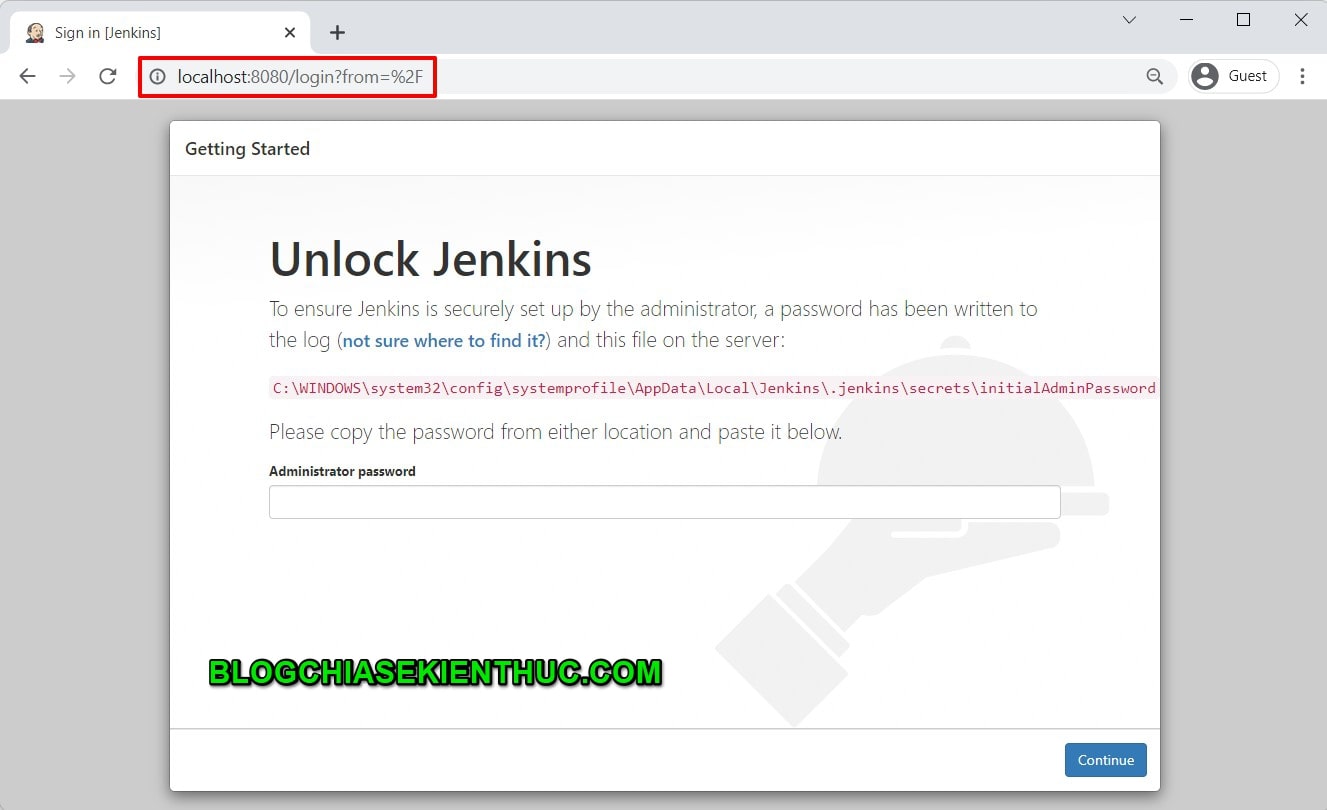
Bấm Finsish để hoàn tất.



*Hình ảnh 3.6.11*

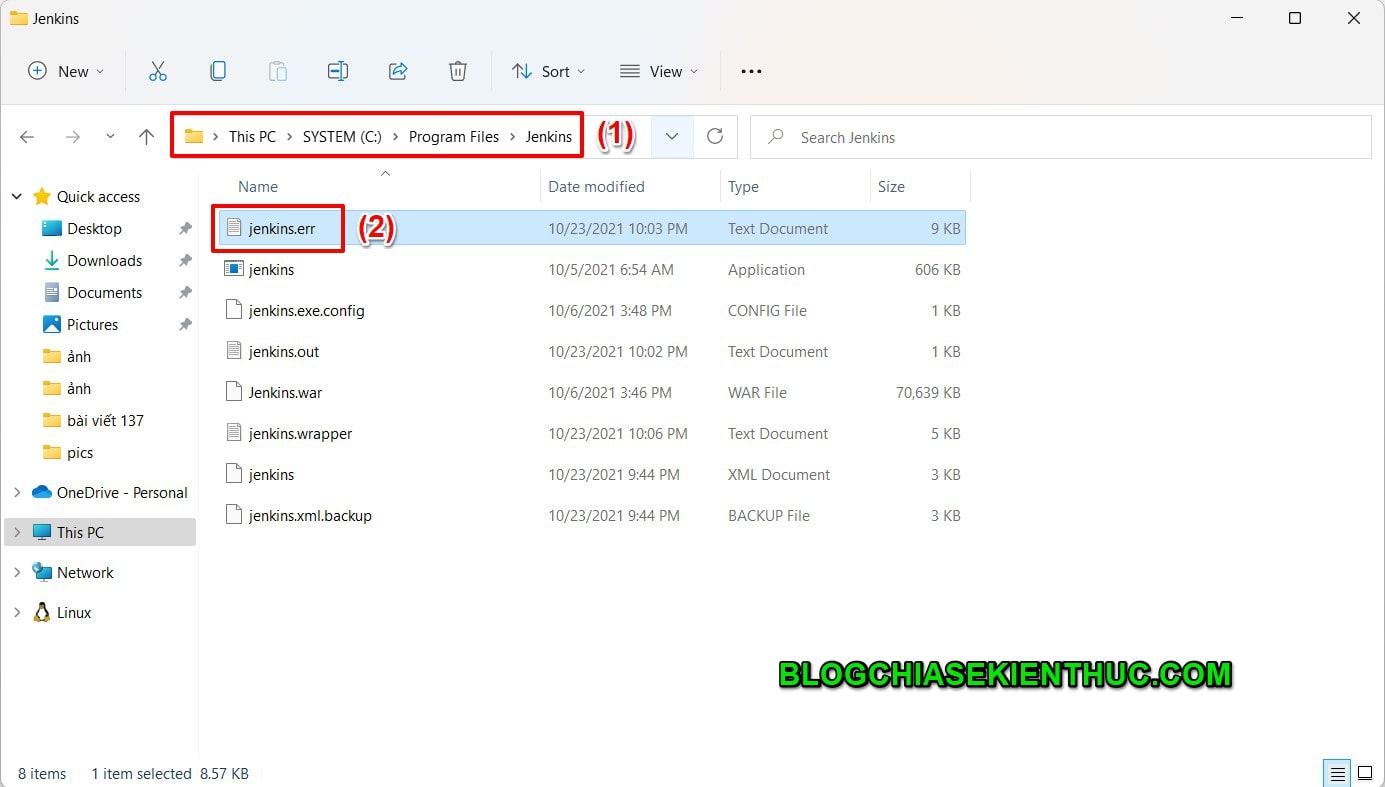
**Bước 7:** Jenkins sau khi cài đặt có thể truy cập vào địa chỉ [http://localhost:8080/login](http://localhost:8080/login )để đăng nhập.

Lần đầu truy cập, phải Unlock Jenkins. Như ảnh bên dưới cần một thông tin gọi là Administrator password.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-13.jpg)

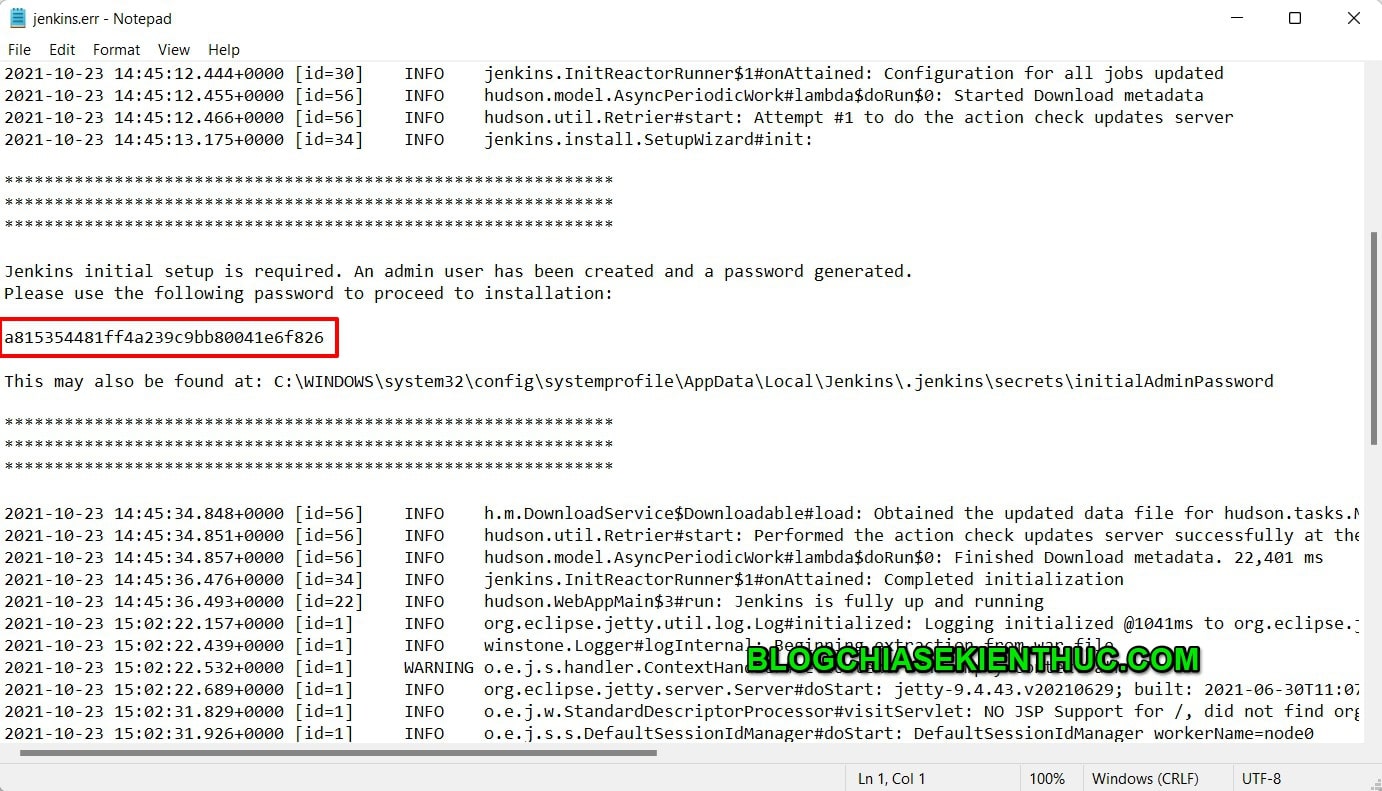
*Hình ảnh 3.6.12*

**Bước 8:** Để lấy được thông tin, truy cập vào thư mục, nơi cài đặt Jenkin sau đó mở file jenkins.err như hình bên dưới.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-14.jpg)

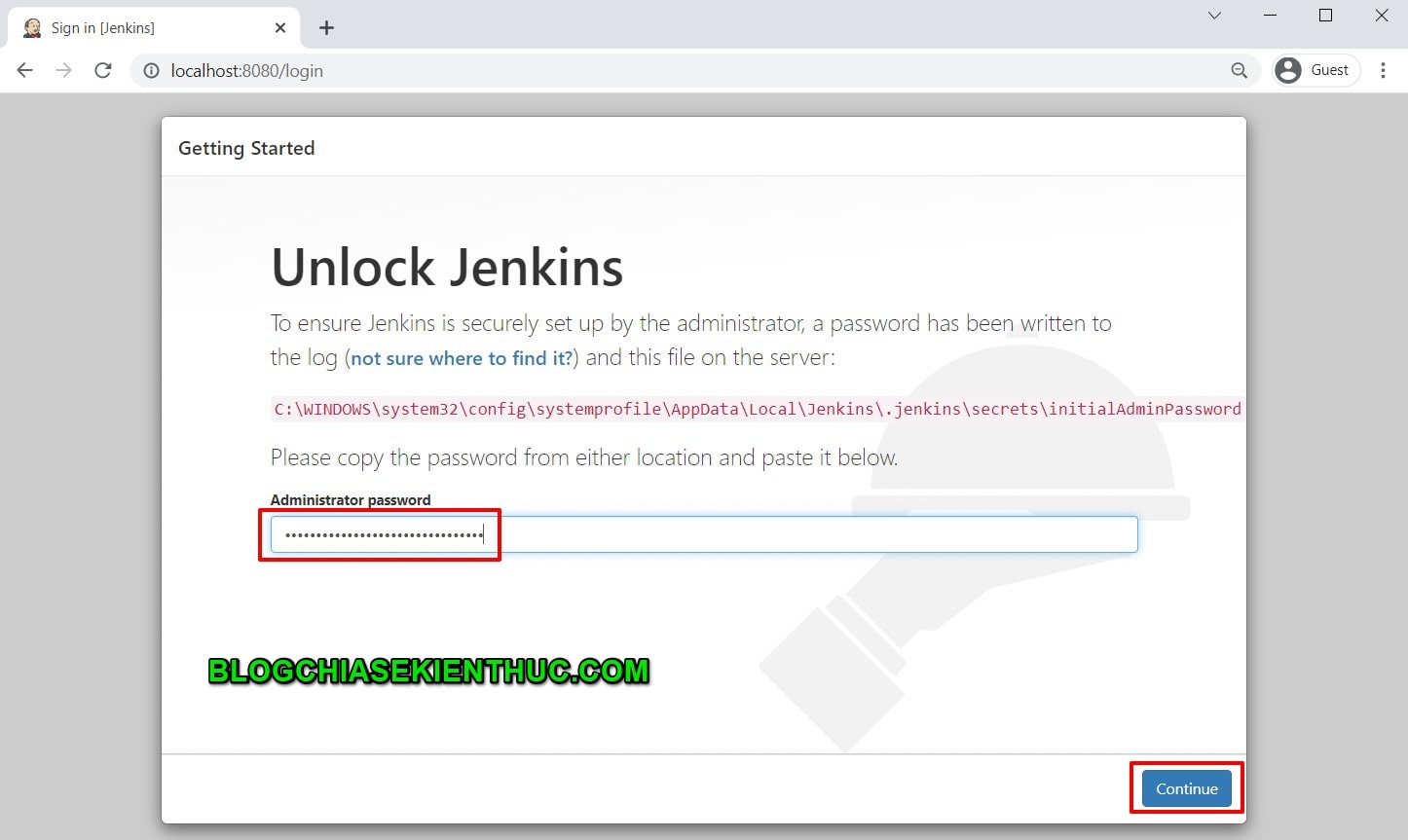
*Hình ảnh 3.6.13*

Đây chính là thông tin Administrator password mà chúng ta phải điền vào form bên trên.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-15.jpg)

*Hình ảnh 3.6.14*

**Bước 9:** Copy sau đó điền vào form => và bấm Continue như hình bên dưới.

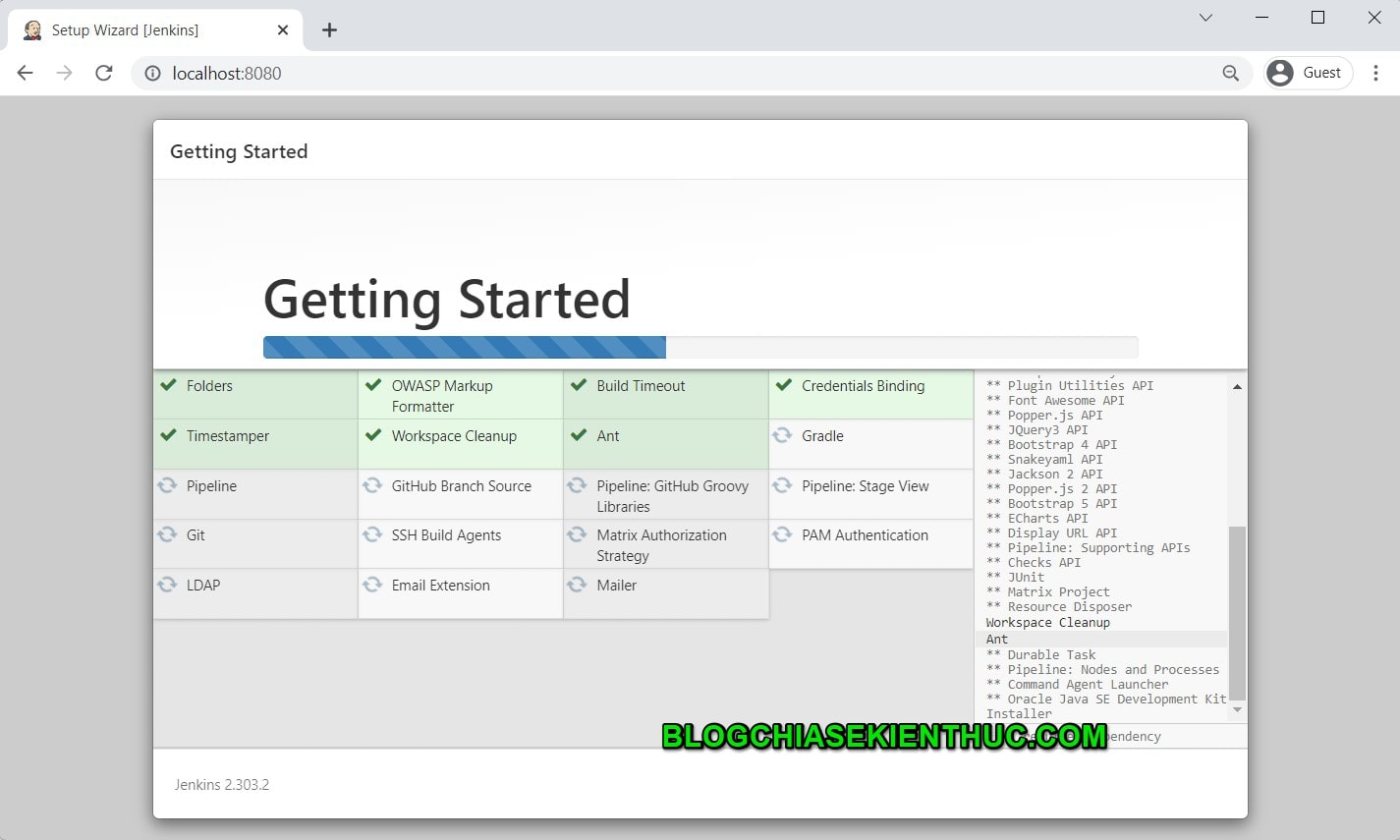
[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-16.jpg)

*Hình ảnh 3.6.15*

**Bước 10:** Login thành công. Giờ chúng ta sẽ cài các tiện ích, ở đây có hai option. Nếu mới bắt đầu tìm hiểu về Jenkins các bạn nên chọn: Install suggested plugins.

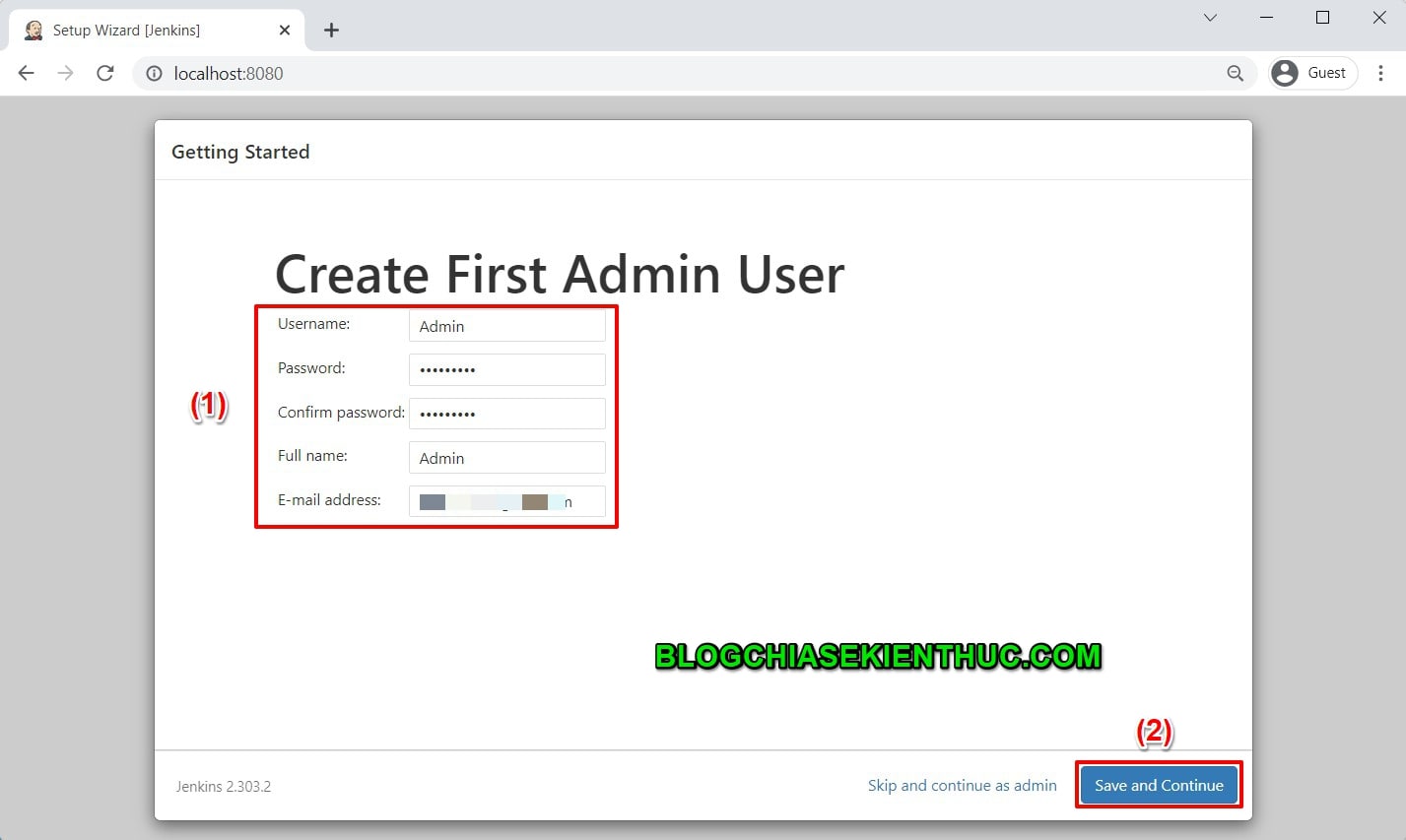
[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-17.jpg)

*Hình ảnh 3.6.16*

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-18.jpg)

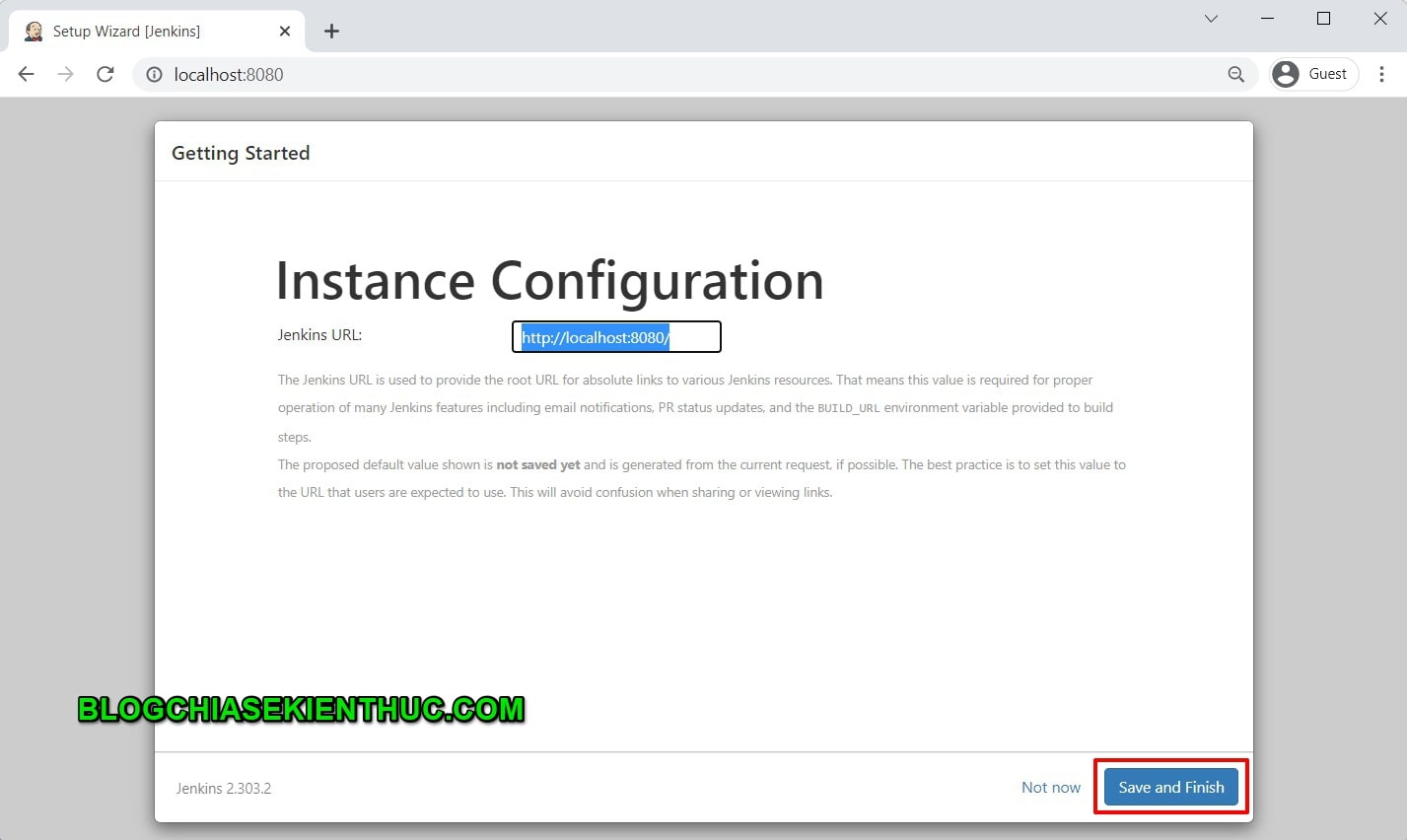
*Hình ảnh 3.6.17*

**Bước 11:** Sau khi cài đặt, sẽ phải cấu hình thông tin cho người dùng quản trị như hình bên dưới sau đó bấm Save and Continue.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-19.jpg)

*Hình ảnh 3.6.18*

**Bước 12:** Bước tiếp theo bấm Save and Finish.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-20.jpg)

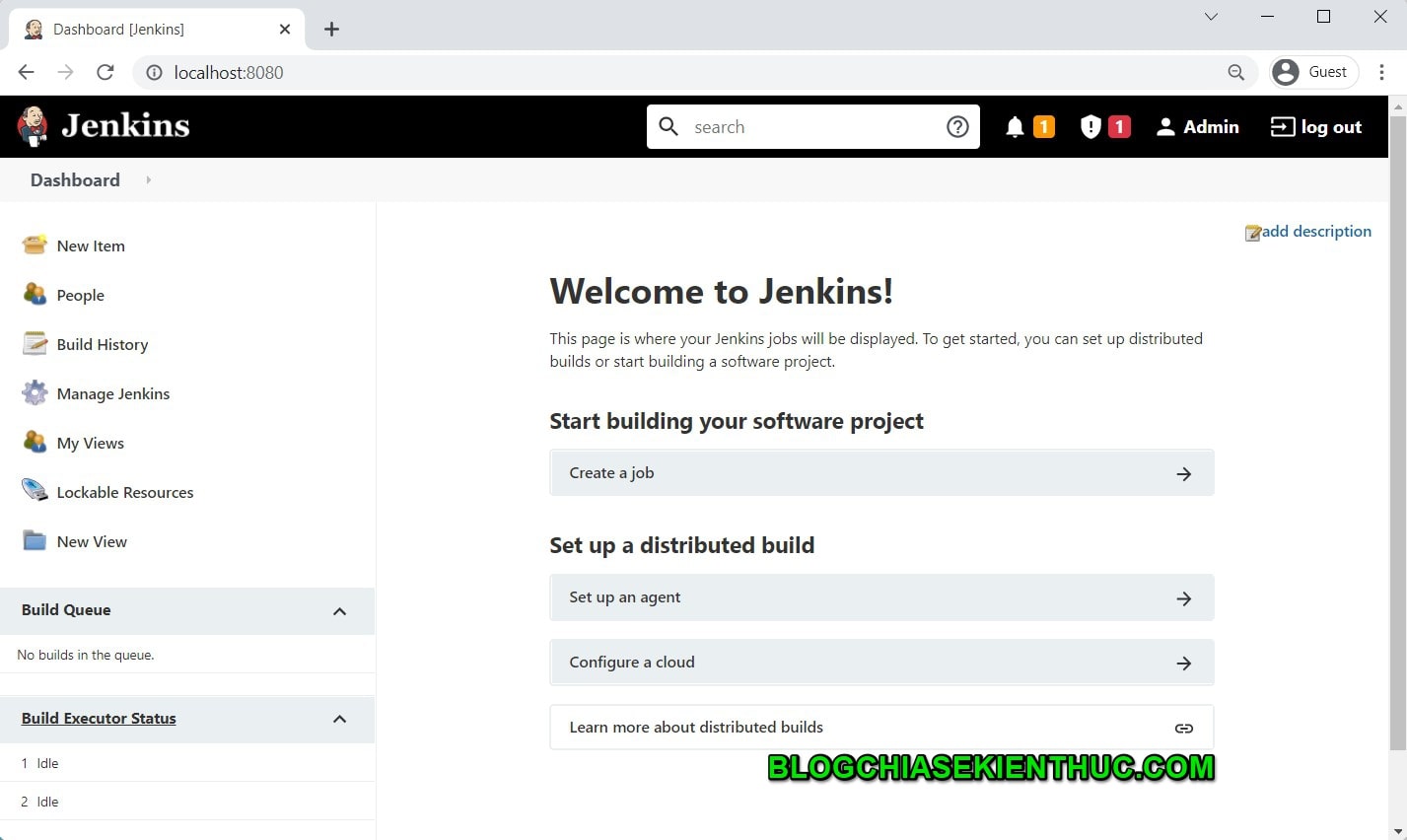
*Hình ảnh 3.6.19*

Jenkins đã được cài đặt và cấu hình xong. Bấm Start using Jenkins để bắt đầu sử dụng.

[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-21.jpg)

*Hình ảnh 3.6.20*

Đây chính là giao diện mở đầu khi đăng nhập thành công vào Jenkins.

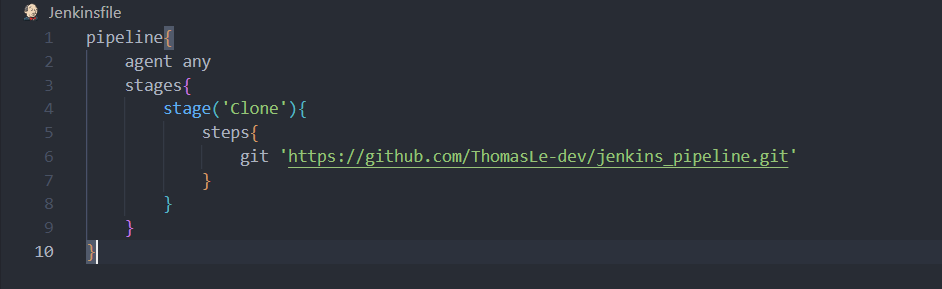
[](https://blogchiasekienthuc.com/wp-content/uploads/2021/11/cach-su-dung-jenkins-22.jpg)

*Hình ảnh 3.6.21*

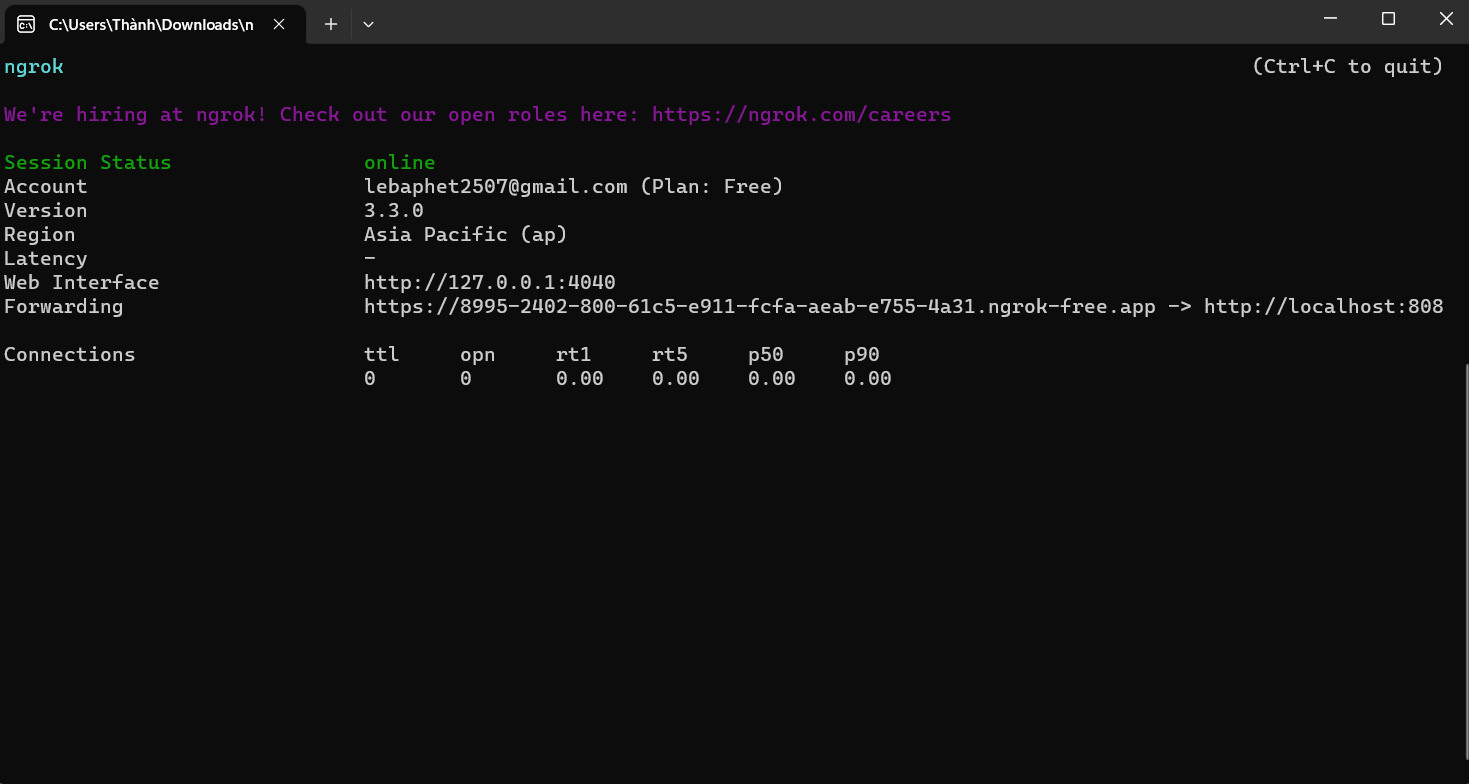
## 3.7 Ứng dụng Jenkins Pipeline vào tự động hóa với Github

Link repository: <https://github.com/ThomasLe-dev/jenkins_pipeline.git>

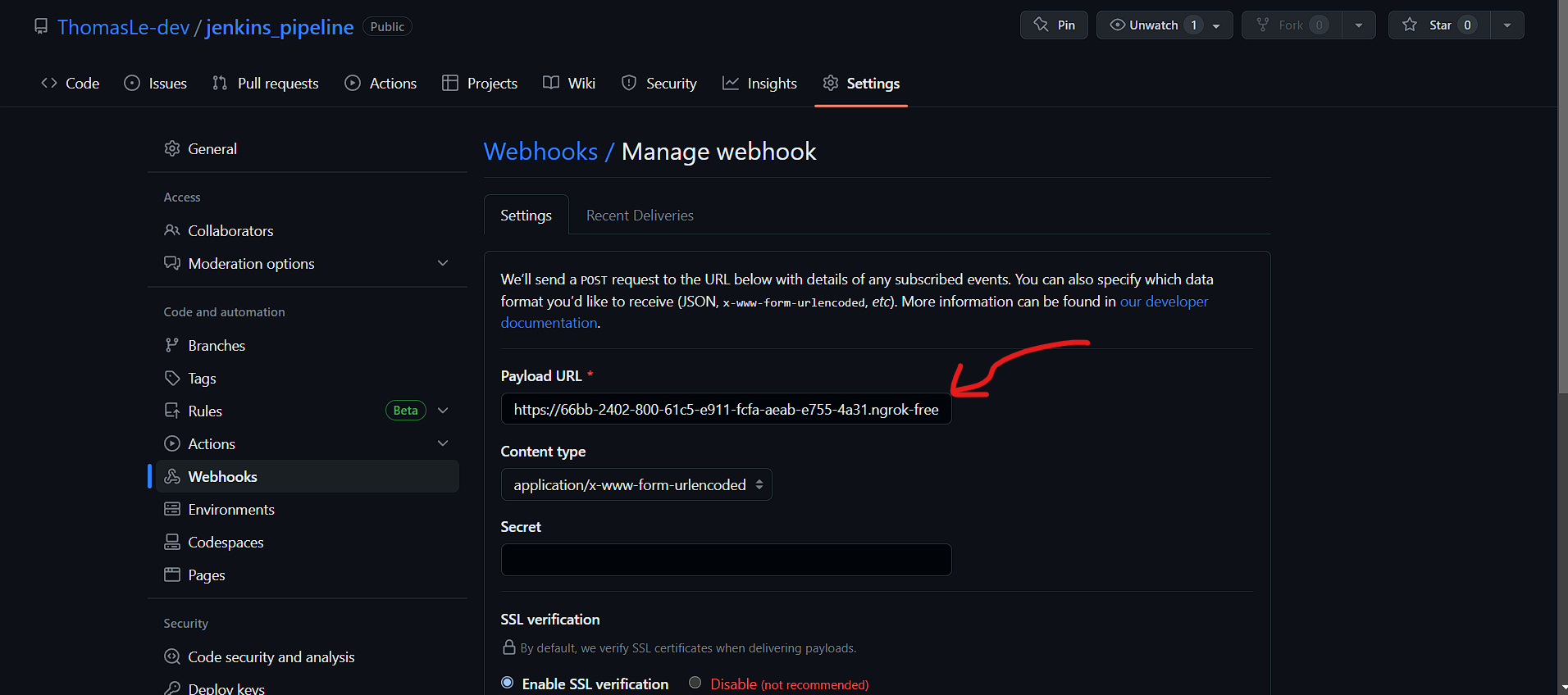
* Bước 1: Tạo 1 Jenkinsfile config và push lên 1 repository có sẵn



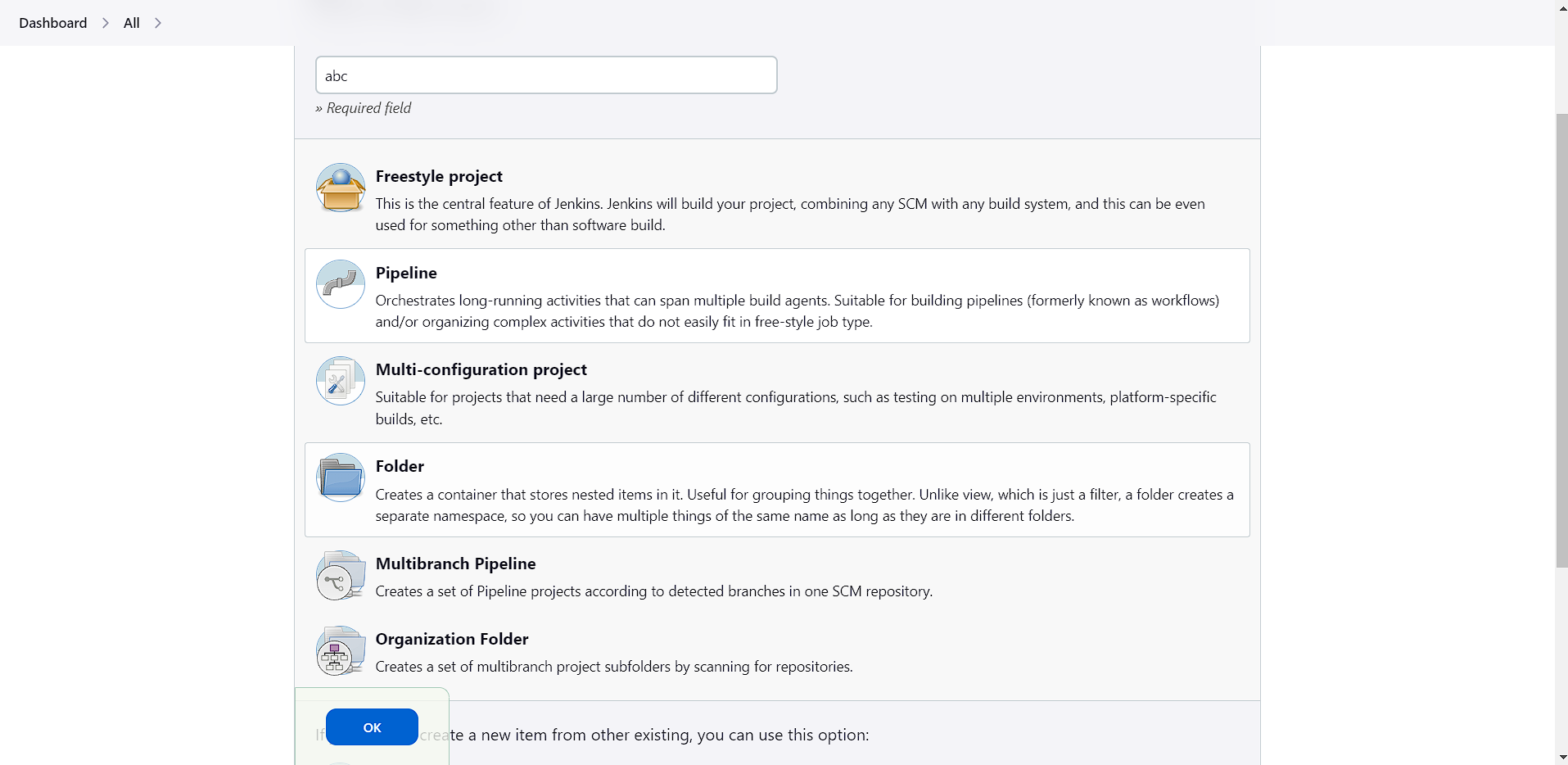
* Bước 2: Triển khai Jenkins localhost lên online thông qua công cụ ngrok



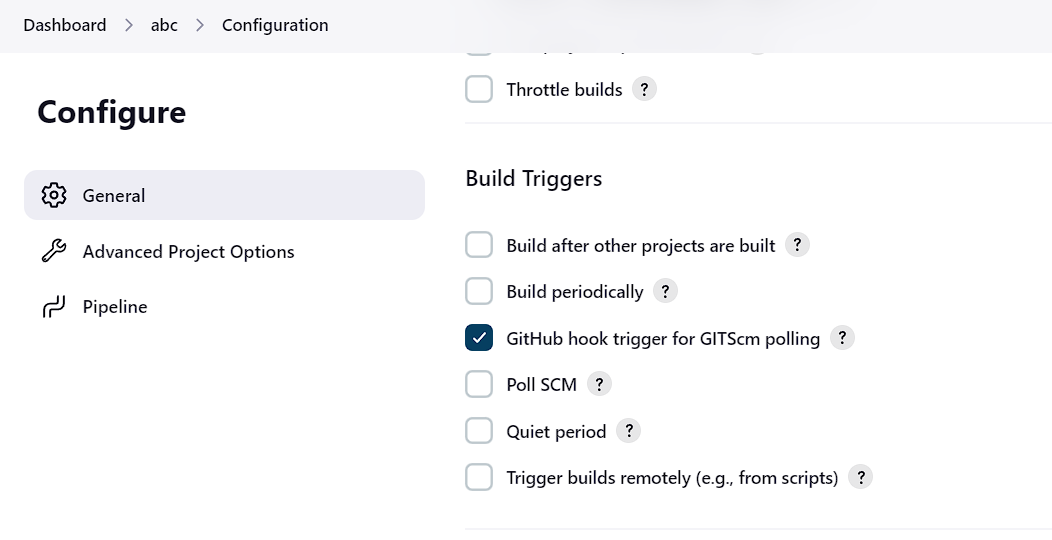
* Bước 3: Tạo webhook giữa Jenkins server và repository đã tạo trên Github bằng cách nhập URL vừa public của jenkins server vào tùy chọn Payload URL



* Bước 4: Truy cập Jenkins server và tạo item pipeline mới bằng cách click vào new Item và điền thông tin pipelien item muốn tạo



* Bước 5: Tiến hành config pipeline item với các tùy chọn sau



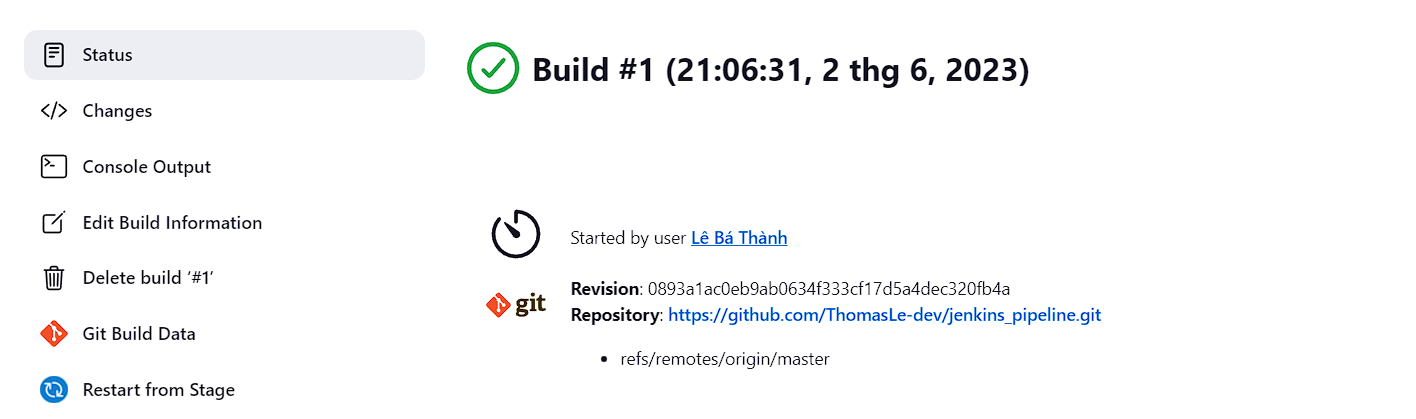
* Bước 6: Kết nối với repo đã tạo và nhấn save



* Bước 7: Nhấn vào tùy chọn Build Now để tiến hành build



* Bước 8: Kiểm tra xem đã build thành công hay chưa bằng cách nhấn vào tùy chọn Status



* Bước 9 tạo 1 commit mới và push lên repo để kiểm tra sự tự động hóa quy trình đã config trong Jenkinsfile



Trạng thái trả về success nếu như Jenkins thực hiện step thành công



# Tài liệu tham khảo

[1]. Cách cài đặt công cụ Jenkins

<https://blogchiasekienthuc.com/lap-trinh/tai-va-cai-dat-jenkins.html>

[2]. Tài liệu tìm hiểu tổng quan về các công cụ

* Quality center: <https://doc.qt.io/squish/quality-center-integration.html>
* Test moniter:<http://tryqa.com/what-is-fundamental-test-process-in-software-testing/>
* IronWASP: <http://blog.ironwasp.org/>
* LoadRunner: <https://docplayer.net/3449009-Comparative-study-of-testing-tools-apache-jmeter-and-load-runner.html>
* Apache Jmeter: <https://jmeter.apache.org/>
* Jira: <https://confluence.atlassian.com/jira>
* Selenium : <https://www.selenium.dev/documentation/>