



Jenkins là một công cụ tự động hóa mã nguồn mở viết bằng Java, hỗ trợ tích hợp liên tục (CI) và phân phối liên tục (CD) cho các dự án.

### CÀI ĐẶT JENKINS (INSTALL JENKINS)

#### Trên Windows

BƯỚC	MÔ TẢ
1	Cài đặt Java Development Kit (JDK)
2	Thiết lập biến môi trường (Path) cho JDK
3	Tải và cài đặt Jenkins
4	Chạy Jenkins trên <code>localhost:8080</code>

#### Trên Ubuntu

BƯỚC	MÔ TẢ
1	Cài đặt Java ( <code>sudo apt install -y openjdk-17-jdk openjdk-17-jre</code> )
2	Thêm kho lưu trữ Jenkins
3	Cài đặt Jenkins ( <code>sudo apt update</code> , <code>sudo apt install jenkins</code> )
4	Thiết lập Jenkins qua <code>http://localhost:8080</code>

## CÁC LOẠI JOB (JOB TYPES)

LOẠI JOB	MÔ TẢ
Freestyle	Job xây dựng đa năng, linh hoạt, phù hợp mọi dự án
Pipeline	Chạy toàn bộ quy trình phát triển dưới dạng mã nguồn
Multiconfiguration	Chạy cùng job trên nhiều môi trường khác nhau
Folder	Tạo thư mục để tổ chức và phân loại job
GitHub Organization	Quét toàn bộ tổ chức GitHub, tạo Pipeline cho repository có Jenkinsfile
Multibranch Pipeline	Hỗ trợ các Jenkinsfile khác nhau cho từng nhánh của dự án
Maven	Xây dựng dự án Maven, tận dụng file POM để đóng gói

## BUILD PIPELINE

Build Pipeline cho phép nối chuỗi nhiều job và chạy tuần tự.

BƯỚC	MÔ TẢ
1	Tạo 3 job Freestyle
2	Nối chuỗi 3 job: Job1 → Job2 → Job3 <b>Job1 → Configure → Post Build → Build other projects → Job2</b> <b>Job2 → Configure → Post Build → Build other projects → Job3</b>
3	Tạo Build Pipeline View <b>Jenkins Dashboard → Add view → Nhập tên → Build Pipeline View → OK → Configure → Pipeline Flow → Chọn Job ban đầu → Job1 → OK</b>
4	Chạy Build Pipeline

## JENKINS PIPELINE

Jenkins Pipeline chạy toàn bộ quy trình dưới dạng mã nguồn trong file Jenkinsfile, viết bằng Groovy DSL.

LOẠI	MÔ TẢ
Scripted Pipeline	Mã được viết trên giao diện Jenkins, nằm trong khối <code>node { ... }</code>
Declarative Pipeline	Mã được viết cục bộ trong file, đẩy lên SCM, nằm trong khối <code>pipeline { ... }</code>

## PLUGINS JENKINS (JENKINS PLUGINS)

Jenkins có hơn 2000 plugin. Một số plugin phổ biến: Maven, Git, Ant, Docker, Amazon EC2, HTML Publisher, Copy Artifact, v.v.

BƯỚC	MÔ TẢ
1	Jenkins Dashboard → Manage Jenkins → Manage Plugins → Available
2	Nhập tên plugin vào ô lọc và cài đặt

## DỰ ÁN MAVEN (MAVEN PROJECT)

TÍNH NĂNG	MÔ TẢ
Build Step	Viết mục tiêu Maven (CLEAN, INSTALL, COMPILE, TEST, PACKAGE) tác động lên file <code>POM.xml</code>
Post-build Actions	Triển khai file WAR/EAR lên container và lưu trữ
Pre-build Steps	Thực thi lệnh batch hoặc shell

**Ví dụ Dự án Maven:** Mở job Freestyle → Dán URL repository GitHub chứa POM.xml vào SCM → Thêm lệnh Maven (`mvn clean install`) → Tạo file WAR/JAR → Triển khai lên Tomcat.

## TRIỂN KHAI TỰ ĐỘNG (AUTOMATED DEPLOYMENT)

BƯỚC	MÔ TẢ
1	Vào <code>Manage Plugins</code> , cài plugin phù hợp
2	Lấy file WAR/EAR và triển khai lên ứng dụng remote đang chạy
3	Vào <code>Build</code> → <code>Configure</code> → Chọn " <code>Deploy WAR/EAR to container</code> "
4	Nhập thông tin server đích trong phần WAR container → Lưu

## QUẢN LÝ NGƯỜI DÙNG VÀ VAI TRÒ (MANAGE USERS & ROLES)

Jenkins hỗ trợ phân quyền cho từng job bằng "Project-based Matrix Authorization Strategy".

BƯỚC	MÔ TẢ
1	Đăng nhập bằng tài khoản Admin → <b>Manage Jenkins → Configure Global Security</b>
2	Trong "Authorization", chọn "Project-based Matrix Authorization Strategy"
3	Thêm user "Admin" và cấp toàn quyền bằng cách tích hết các ô

Xem trực tiếp tại: [Jenkins Cheat sheet - DevOps VietNam \(devops.vn\)](#)