

# Chương 5: Hướng dẫn Cài đặt & Triển khai

Chương này hướng dẫn chi tiết cách thức thiết lập môi trường, đóng gói và triển khai ứng dụng cho cả 2 khối cấu thành nên Signing Service Tân Cảng: Web Service (chạy trên Server) và Desktop App (chạy trên máy trạm của người dùng).

- [5.1. Yêu cầu Hệ thống \(Prerequisites\)](#)
- [5.2. Hướng dẫn Triển khai Signing Web Service \(Server\)](#)
- [5.3. Đóng gói & Cài đặt Signing Desktop App \(Client\)](#)

# 5.1. Yêu cầu Hệ thống (Prerequisites)

Để đảm bảo hệ thống có thể biên dịch và chạy trơn tru, máy chủ / máy lập trình viên cần đáp ứng các điều kiện sau:

- **Môi trường Java:** Khuyến nghị sử dụng cài đặt **JDK 17**.
- **Trình quản lý gói: Maven** `3.8+` (Hoặc dùng file mvnw cung cấp sẵn trong source code).
- **Môi trường Máy chủ (Web Service):** Linux/Windows Server đã cài đặt sẵn **Docker** và **Docker Compose**.
- **Môi trường Đóng gói (Desktop App):** Máy tính Windows để đóng gói JRE custom. Yêu cầu có cài đặt **WiX Toolset (v3.x)** nếu muốn build file cài dạng `.msi`.
- **Database & Cache:** MSSQL Server (2016 trở lên) và Redis Server (Khuyến nghị Redis v7+).
- **Source code:** [http://192.168.0.95/nam/signing\\_service\\_tan\\_cang.git](http://192.168.0.95/nam/signing_service_tan_cang.git) nhánh [dev\\_namdh\\_app\\_signing\\_desktop](#)

# 5.2. Hướng dẫn Triển khai Signing Web Service (Server)

Ứng dụng Web Service được tối ưu để hoạt động trong Docker Container, cung cấp khả năng khởi tạo nhanh và độc lập môi trường.

## Cách 1: Chạy trực tiếp qua Docker Compose (Khuyến nghị)

Hệ thống đã cung cấp sẵn file docker-compose.yml. Khi gọi lệnh `docker-compose up`, nó sẽ tự động triển khai 2 service: `redis` và ứng dụng `kyso-service`. (Lưu ý: Bạn cũng có thể kéo Image từ Harbor nếu có tài khoản, ví dụ: `harbor.lifetex.vn/kysotaptrung/kyso-service`)

**Các biến môi trường (Environment Variables) cần thay đổi trước khi chạy:** Mở file docker-compose.yml và kiểm tra phần `environment` của service `kyso-service`:

```
environment:
  # Kết nối Cơ sở dữ liệu MS SQL Server
  SPRING_DATASOURCE_URL:
  jdbc:sqlserver://<DB_HOST>:<PORT>;databaseName=<DB_NAME>;encrypt=true;trustServerCertificate=true
  SPRING_DATASOURCE_USERNAME: <User_DB>
  SPRING_DATASOURCE_PASSWORD: <Pass_DB>

  # Cấu hình Redis (Thông qua container nội bộ)
  SPRING_DATA_REDIS_HOST: kyso-redis
  SPRING_DATA_REDIS_PORT: 6379
```

### Lệnh khởi động dịch vụ:

```
# Đi vào thư mục chứa source signing_service_tan_cang
# Build image và chạy ngầm (Detached mode)
docker-compose up -d --build
```

Log hệ thống có thể được xem thông qua lệnh: `docker logs -f kyso-service`. API cổng chính thức hoạt động trên port `6868`.

## Cách 2: Triển khai trực tiếp bằng file JAR

Nếu không dùng Docker, bạn có thể build file `.jar`:

```
mvn clean package -pl signing-common,signing-core,signing-scanner-sdk,signing-web-service -am -DskipTests
```

Sau đó chạy file JAR kèm theo file `application.yml`:

```
java -jar signing-web-service/target/signing-web-service-0.0.1-SNAPSHOT.jar
```

# 5.3. Đóng gói & Cài đặt Signing Desktop App (Client)

Client App được tối ưu hóa đặc biệt. Thay vì phải yêu cầu hàng trăm nhân sự dùng máy tính văn phòng tự tải JDK/JRE về để chạy lệnh `java -jar`, ứng dụng Desktop được **nhúng (bundle) thẳng JRE** (Custom JRE) vào bên trong thông qua công cụ `jlink` và `jpackage`. Nó biến Web Service local thành một file `<Tên-App>.exe` hoặc `<Tên-App>.msi` độc lập.

Quy trình đóng gói đã được gom gọn tự động hóa toàn bộ trong script **package\_desktop.bat**.

## Hướng dẫn chạy script đóng gói:

1. Đảm bảo biến môi trường máy tính của bạn đã trở chuẩn xác `%JAVA_HOME%` (JDK 17 của thư mục bin). Nếu quên, script sẽ báo lỗi không tìm thấy `jlink` hay `jpackage`.
2. (Tùy chọn) Cài đặt phần mềm **WiX Toolset (v3.x)** tại <https://wixtoolset.org/releases/>. Đây là thư viện gốc do Microsoft hỗ trợ để đóng gói hệ thống dưới định dạng Cài đặt chuẩn `.msi` (Dễ dàng để IT căng push policy GPO tự động cài cho toàn bộ máy nhánh).
3. Mở cửa sổ Command Prompt (cmd) hoặc Terminal, chạy lệnh:

```
package_desktop.bat
```

## Diễn giải 5 bước cấu thành của Script:

- **[1/5] Build Source Code:** Script tự động gọi `mvn clean package -DskipTests` riêng cho module `signing-desktop` (và các module con tham chiếu).
- **[2/5] Custom JRE (jlink):** Loại bỏ bớt các thư viện thừa (như javafx, jdk.compiler không cần thiết), tạo ra một máy ảo Java thu nhỏ (giảm dung lượng) vào thư mục `custom-jre`. Tính năng `java.smartcardio` được thêm vào bắt buộc để tương tác được USB Token lỗi.
- **[3/5] Portable Image (jpackage):** Gom mã nguồn Desktop (file `.jar`) làm nhân của file EXE (App Image). Kết quả cho ra ở thư mục `dist/signing-desktop`. Ai copy thư mục này ném sang máy tính khác chạy luôn file `.exe` bên trong là hệ thống sẽ bật.
- **[4/5] Check WiX Toolset:** Kiểm tra tool `candle` của WiX. Nếu không có ở máy build, sẽ skip Bước 5.
- **[5/5] Build Windows Installer (.msi):** Gom cả Bundle thư mục Portable thành 1 file Install System duy nhất. Khi người dùng chạy, nó sẽ tự cài vào `C:\Program Files\`, tạo lối tắt đề ngoài màn hình Desktop (Shortcut) và trong thanh Start Menu.

Kể từ nay, bạn chỉ việc gửi file `.msi` hoặc file `.exe portable` đã sinh ở thư mục `dist/` cho kế toán/nhân sự để cài đặt sử dụng ký bằng Token một cách mượt mà.