* Gặp một vấn đề: build (run unit test, build installer) trên local machine, nhưng dev khác lấy code về ko build được (có thể commit thiếu file, môi trường build khác, …)
* Có thể sử dụng Docker (nếu có thời gian sẽ implement)

Implement CI/CD pipeline on SVN server:

* When have a commit to SVN server:
  + Auto build source code 🡪 Gen report.
  + Run unit test after build 🡪 Gen report
  + Create installer, checking installer 🡪 Gen report

1. Install Jenkins

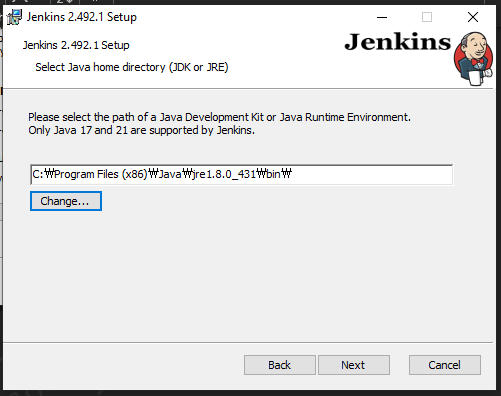
* Download link: <https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable/>
* Install:
  + By default, your Jenkins runs at https://localhost:8080/. This can be changed by editing jenkins.xml , which is located in your installation directory. This file is also the place to change other boot configuration parameters, such as JVM options, HTTPS setup, etc.
  + Jenkins is installed as a Windows service, and it is configured to start automatically upon boot. To start/stop them manually, use the service manager from the control panel, or the sc command line tool.
  + Create an account for running Jenkins service:

Tạo một tài khoản local (qua "Computer Management" > "Local Users and Groups" > "Users") như JenkinsUser với mật khẩu.

Gán quyền "Log on as a service" cho tài khoản này thông qua "Local Security Policy" (truy cập bằng secpol.msc > "Local Policies" > "User Rights Assignment" > "Log on as a service" > thêm tài khoản).

* + Service Jenkins User Logon Credentials: A screenshot of a computer

    AI-generated content may be incorrect.
  + Select Java home directory:



* Start Jenkins, access it via http://localhost:8080 (or your server’s IP), and complete the setup by installing recommended plugins. Ensure the Subversion Plugin is included (it’s usually part of the default set).

1. Set up Automatic Build Trigger when have 1 new commit change

There are two main ways to trigger a build on SVN commit:

Option 1: Polling SCM: Meaning create a schedule check SVN change every minute.

Option 2: Post-commit hook

* Configure Jenkins to Connect to Your SVN Repository:
  + Create a New Job: In Jenkins, click "New Item," select "Freestyle project," and name it (e.g., "SVN-Auto-Build").

A screenshot of a computer

AI-generated content may be incorrect.

* On your SVN server, locate the repository’s hooks directory (e.g., /path/to/repo/hooks). (Sample: D:\VisualSVN Server\Repositorie\DependencyManager\hooks)
* Windows: Create or edit post-commit.bat file
* Edit the above bat file, add:

**@echo off**

**setlocal enabledelayedexpansion**

**:: Jenkins credentials**

**set JENKINS\_USER=admin**

**set JENKINS\_TOKEN=1131d210a68f6cc7370baf10ada26d04df**

**set JENKINS\_URL=http://localhost:8080**

**set JOB\_NAME=Dependen-Auto-Build**

**::set TOKEN=AUTOBUILDTOKEN**

**:: Get Jenkins Crumb**

**for /f "tokens=\*" %%a in ('curl -s -u "%JENKINS\_USER%:%JENKINS\_TOKEN%" "%JENKINS\_URL%/crumbIssuer/api/xml?xpath=concat(//crumbRequestField,\":\",//crumb)"') do set JENKINS\_CRUMB=%%a**

**:: Check if the crumb was retrieved**

**if not defined JENKINS\_CRUMB (**

**echo Failed to retrieve Jenkins crumb. Exiting.**

**exit /b 1**

**)**

**:: Log the SVN commit revision (for debugging)**

**echo Commit detected: %1 > D:\VisualSVN Server\Repositorie\DependencyManager\hooks\post-commit.log**

**:: Trigger Jenkins job using curl or PowerShell**

**curl -X POST -u "%JENKINS\_USER%:%JENKINS\_TOKEN%" -H "!JENKINS\_CRUMB!" "%JENKINS\_URL%/job/%JOB\_NAME%/build"**

**:: Alternative using PowerShell if curl is unavailable**

**:: powershell -Command "Invoke-RestMethod -Uri '%JENKINS\_URL%/job/%JOB\_NAME%/build?token=%TOKEN%' -Method Post"**

**exit 0**

Enable Remote trigger in Jenkins:

* + File post-commit.bat at: D:\VisualSVN Server\Repositorie\DependencyManager\hooks
  + 

**How to Get Your Jenkins API Token**

Go to **Jenkins Dashboard** > Click on your username (top right).

Click **Configure**.

Scroll to **API Token** > Click **Generate New Token**.

Copy and use this token in your script (JENKINS\_TOKEN).

Every time commit a change, job: Dependen-Auto-Build will be run.

1. Define Build Steps

* Install **MSBuild** plugin:

A screenshot of a black screen

AI-generated content may be incorrect.

* Config multiple MSBuild version (2013, 2015, 2019, 2022)

A screenshot of a computer

AI-generated content may be incorrect.

* **Set Up Source Code Management (SVN Checkout),** go to: <http://localhost:8080/job/Dependen-Auto-Build/configure>

A screenshot of a chat

AI-generated content may be incorrect.

Note:

* Local module directory: **.** use default workspace, *C:\ProgramData\Jenkins\.jenkins\workspace\Dependen-Auto-Build*
* **Add a Build Step**: In the Jenkins job config, go to the "Build Steps" section and select "**Build a Visual Studio project or solution using MSBuild**"

A screenshot of a chat

AI-generated content may be incorrect.

Some note for Jenkins machine:

* Because include/link config use environment variables, so we have to sure the jenkins machine is set environment variables (sample: VcpkgRoot)

1. **View the Report**

* We can config to display build name of automative build job following SVN commit revision
  + Install the Build Name Setter Plugin: Go to "Manage Jenkins" > "Manage Plugins" > Search for "Build Name Setter" and install it.
  + Under "Build Environment," check "Set Build Name."
  + Set the build name pattern, e.g., DependencyManager/Trunk-r${SVN\_REVISION}.
  + Then, name of build job will be displayed following revision of commit changes.

A screenshot of a phone

AI-generated content may be incorrect.

* + We can to: [**http://localhost:8080/job/Dependen-Auto-Build/**](http://localhost:8080/job/Dependen-Auto-Build/)to track result build (Depend-Auto-Build: job name).