

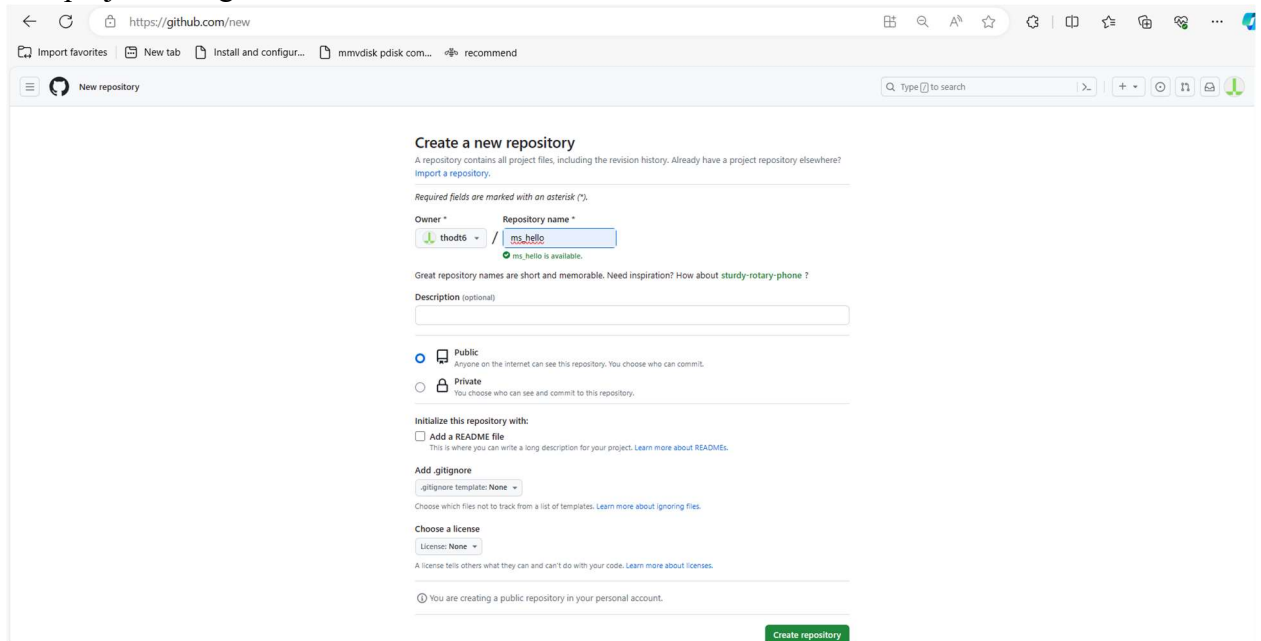
Trong bài viết này tôi sẽ mô tả việc sử dụng Jenkins để thực hiện tích hợp deploy source code là một ứng dụng Spring-boot trên Github về server test

Luồng thực hiện như sau:



Bước 1: Chuẩn bị code Spring boot. Ở đây tôi sẽ xây dựng một ứng dụng spring-boot Hello world đơn giản theo các bước sau:

- Tạo project trên github



Quick setup — if you've done this kind of thing before

or

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```

echo "# ms_hello" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/thodt6/ms_hello.git
git push -u origin main
    
```

...or push an existing repository from the command line

```

git remote add origin https://github.com/thodt6/ms_hello.git
git branch -M main
git push -u origin main
    
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

- Init local git source code:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  1
• [thond@vm-app02 ms_hello]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/thond/source_code/ms_hello/.git/
○ [thond@vm-app02 ms_hello]$
    
```

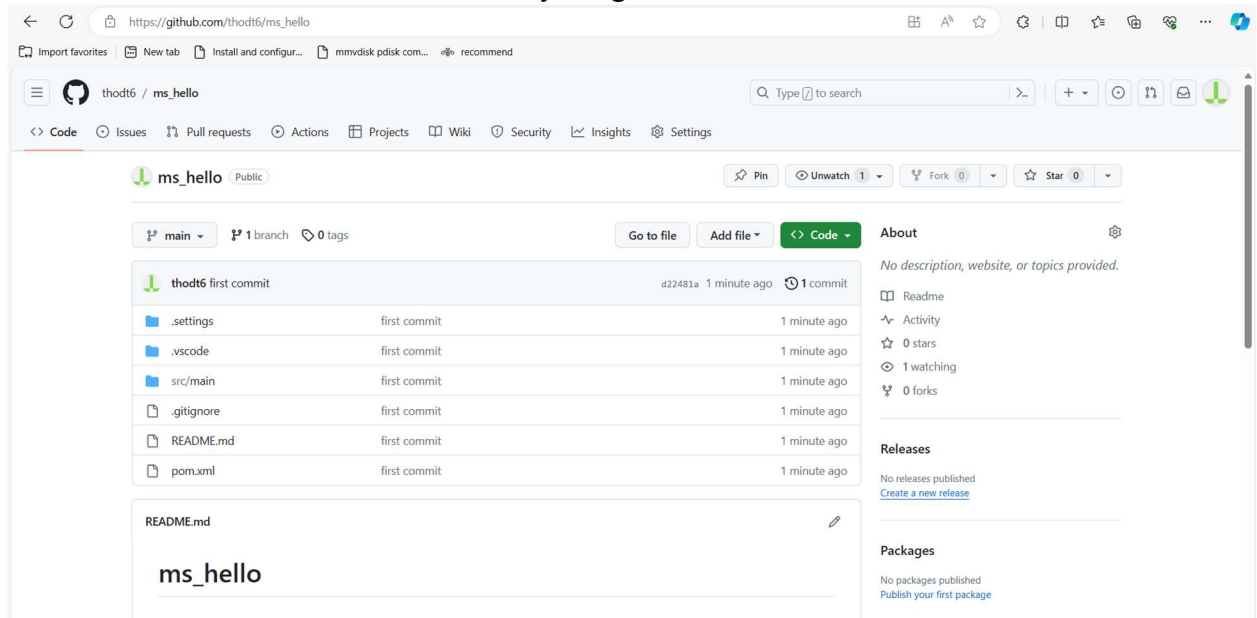
- Commit first helloworld version

```

hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/thond/source_code/ms_hello/.git/
[thond@vm-app02 ms_hello]$ git add .
[thond@vm-app02 ms_hello]$ git commit -m "first commit"
[master (root-commit) d22481a] first commit
18 files changed, 202 insertions(+)
create mode 100644 .gitignore
create mode 100644 .settings/org.eclipse.core.resources.prefs
create mode 100644 .settings/org.eclipse.jdt.apt.core.prefs
    
```

```
[thond@vm-app02 ms_hello]$ ^C
[thond@vm-app02 ms_hello]$ git branch -M main
[thond@vm-app02 ms_hello]$ git remote add origin https://github.com/thodt6/ms_hello.git
[thond@vm-app02 ms_hello]$ git push -u origin main
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
```

- Sau khi thực hiện source code đã được đẩy lên github vào nhánh main



Link sample source code: https://github.com/thodt6/ms_hello

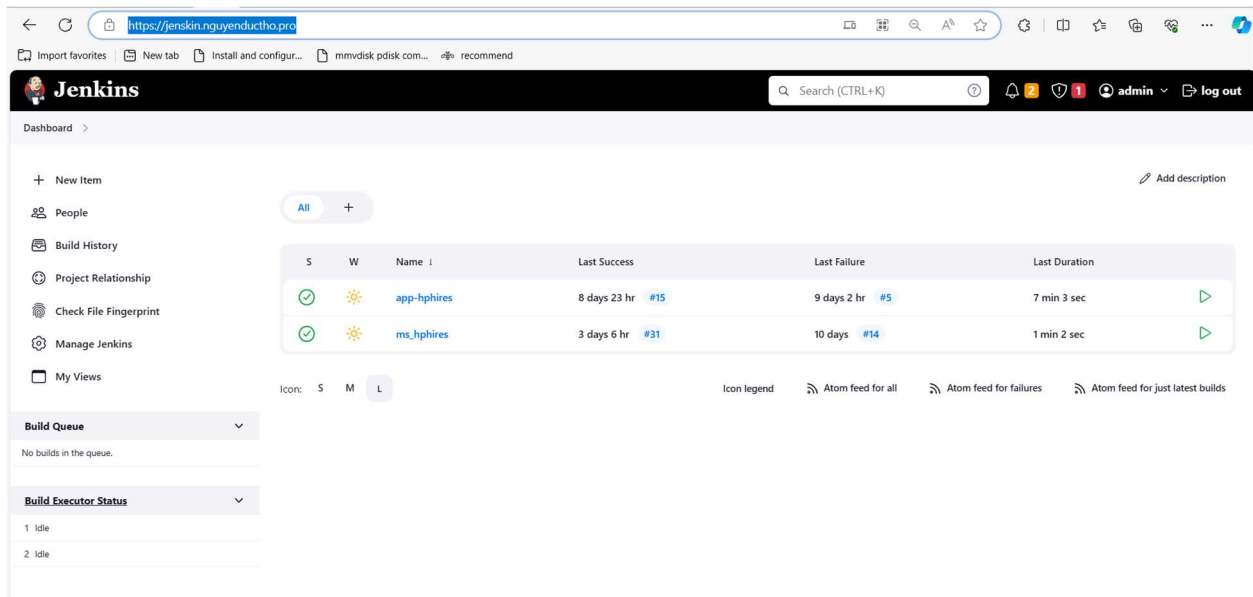
Như vậy bước một đã chuẩn bị xong một sample spring-boot code và có thể commit code đẩy lên github. Bước tiếp theo sẽ liên quan đến việc triển khai jenkins server

Bước 2: Chuẩn bị jenkins server + cài đặt plugin

Link hướng dẫn cài đặt jenkins sampel trên Linux

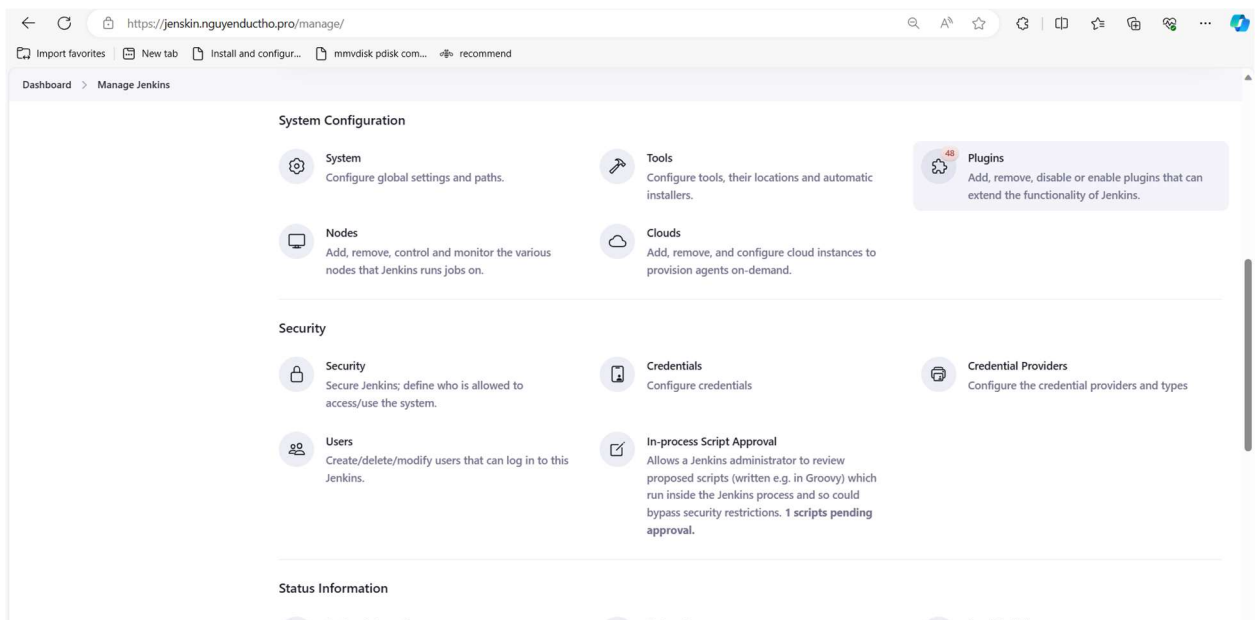
<https://linuxize.com/post/how-to-install-jenkins-on-centos-7/>

Trong ví dụ này tôi đã chuẩn bị một jenkins server theo địa chỉ <https://jenkin.nguyenductho.pro> và đây là màn hình đăng nhập



Ta có thể tiến hành login vào jenskin và chuẩn bị các bước khai báo build item. Tuy nhiên để thực hiện tích hợp với github ta cần cài đặt thêm github plugin theo các bước sau:

- Vào phần manage Jenkins -> Plugins



- Vào mục Available Plugins

Search (CTRL+K)

admin

log out

Dashboard > Manage Jenkins > Plugins

Updates

Available plugins

Installed plugins

Advanced settings

48

Plugins

Search available plugins

Install

Install	Name	Released
<input type="checkbox"/>	<div>JavaScript GUI Lib: ACE Editor bundle 1.1</div> <div>JavaScript GUI Lib: ACE Editor bundle plugin.</div> <div>This plugin is deprecated. In general, this means that it is either obsolete, no longer being developed, or may no longer work. Learn more.</div>	7 yr 8 mo ago
<input type="checkbox"/>	<div>Authentication Tokens API 1.53.v1c90fd9191a_b</div> <div>This plugin provides an API for converting credentials into authentication tokens in Jenkins.</div>	7 mo 27 days ago
<input type="checkbox"/>	<div>JavaScript GUI Lib: Moment.js bundle 1.1.1</div> <div>User Interface</div> <div>JavaScript GUI Lib: Moment.js bundle plugin.</div> <div>This plugin is deprecated. In general, this means that it is either obsolete, no longer being developed, or may no longer work. Learn more.</div>	7 yr 8 mo ago
<input type="checkbox"/>	<div>Javadoc 243.vb_b_503b_b_45537</div> <div>This plugin adds Javadoc support to Jenkins.</div>	

- Search github và tiến hành cài đặt plugin

Search (CTRL+K)

admin

log out

Dashboard > Manage Jenkins > Plugins

Updates

Available plugins

Installed plugins

Advanced settings

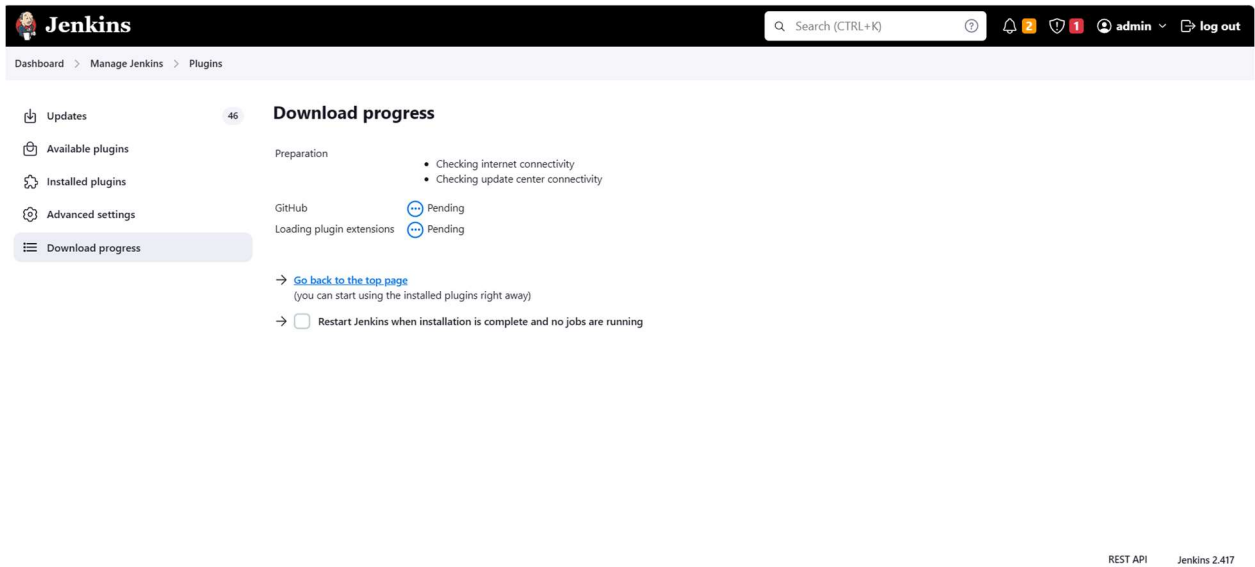
46

Plugins

github

Install

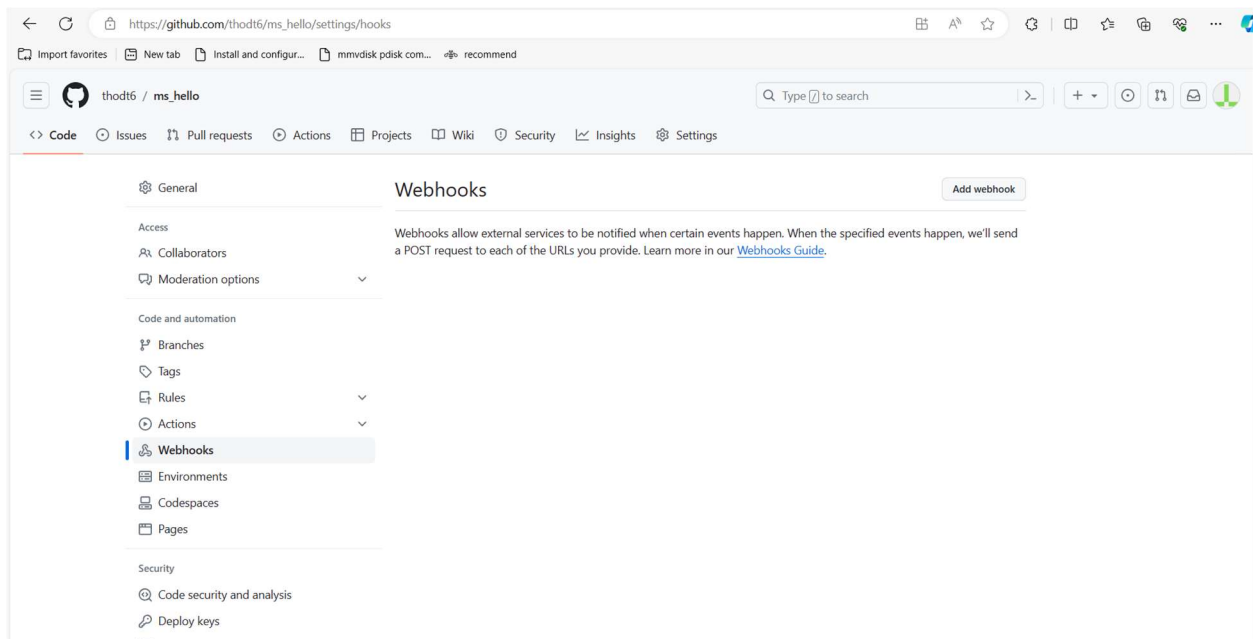
Install	Name	Released
<input type="checkbox"/>	<div>GitHub 1.37.3.1</div> <div>External Site/Tool Integrations github</div> <div>This plugin integrates GitHub to Jenkins.</div>	1 mo 6 days ago
<input type="checkbox"/>	<div>GitHub Branch Source 1751.v90e17c48a_6a_c</div> <div>pipeline github Source Code Management</div> <div>Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.</div>	7 days 20 hr ago
<input type="checkbox"/>	<div>GitHub Pipeline for Blue Ocean 1.27.9</div> <div>External Site/Tool Integrations User Interface</div> <div>BlueOcean GitHub organization pipeline creator</div>	22 days ago
<input type="checkbox"/>	<div>Docker API 3.3.1-79.v20b_53427e041</div> <div>Library plugins (for use by other plugins) docker</div> <div>This plugin provides docker-java API for other plugins.</div> <div>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</div>	5 mo 21 days ago



Click vào ô Restart Jenkins

Bước 3: Tạo webhook cho Github

- Login vào github vào ms_hello repo và vào setting/webhooks



General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

https://jenskin.nguyenductho.pro/github-webhook/

Content type

application/json

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification

Disable (not recommended)

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

Let me select individual events.

Active

We will deliver event details when this hook is triggered.


Add webhook

Bước 4: Tiến hành thực hiện cấu hình item mới cho Jenkins


Enter an item name

ms_helo


» Required field


Freestyle project


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


Pipeline


Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.


Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.


Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.


Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

- Cập nhật các tham số của ms_hello item như bên dưới

Configure

General

Source Code Management


Build Triggers

Build Environment

Build Steps

Post-build Actions

General

Enabled 

Description

Build for Hello

Plain text [Preview](#)


☐ Discard old builds [?](#)

☐ GitHub project

☐ This project is parameterized [?](#)

☐ Throttle builds [?](#)

☐ Execute concurrent builds if necessary [?](#)

Advanced 

- Source Code Management: Chú ý Branch to build chọn đúng branch trên github

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git

Repositories

Repository URL

https://github.com/thodt6/ms_hello.git

Credentials

- none -

Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

Save

Apply

- Build trigger:

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☒ Build after other projects are built

☐ Build periodically

☐ Build when a change is pushed to BitBucket

☒ GitHub hook trigger for GITScm polling

☐ Poll SCM

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Send files or execute commands over SSH before the build starts

☐ Send files or execute commands over SSH after the build runs

☐ Add timestamps to the Console Output

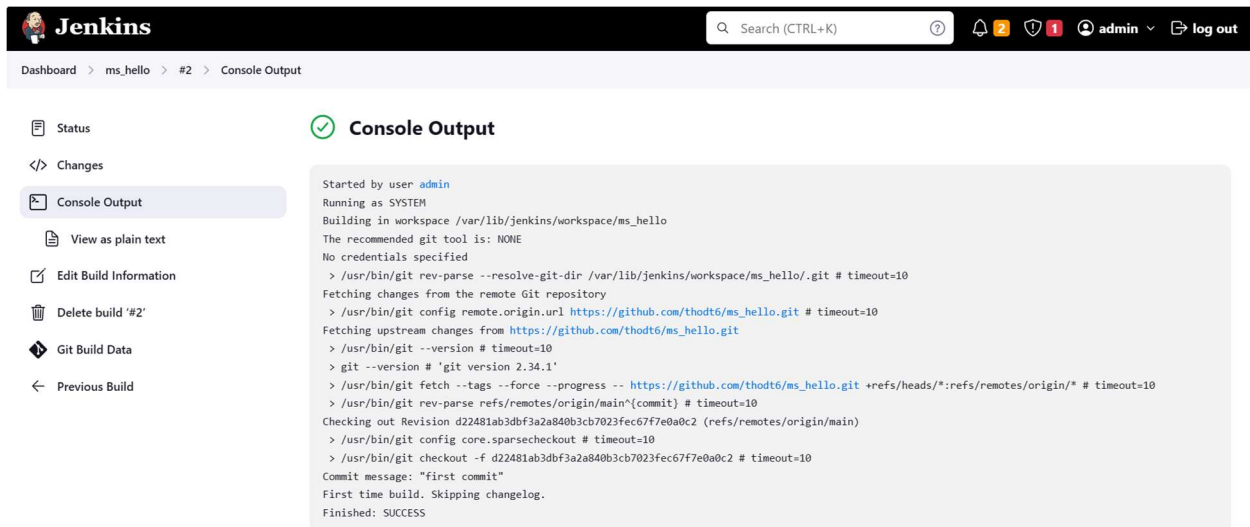
☐ Execute shell script on remote host using ssh

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

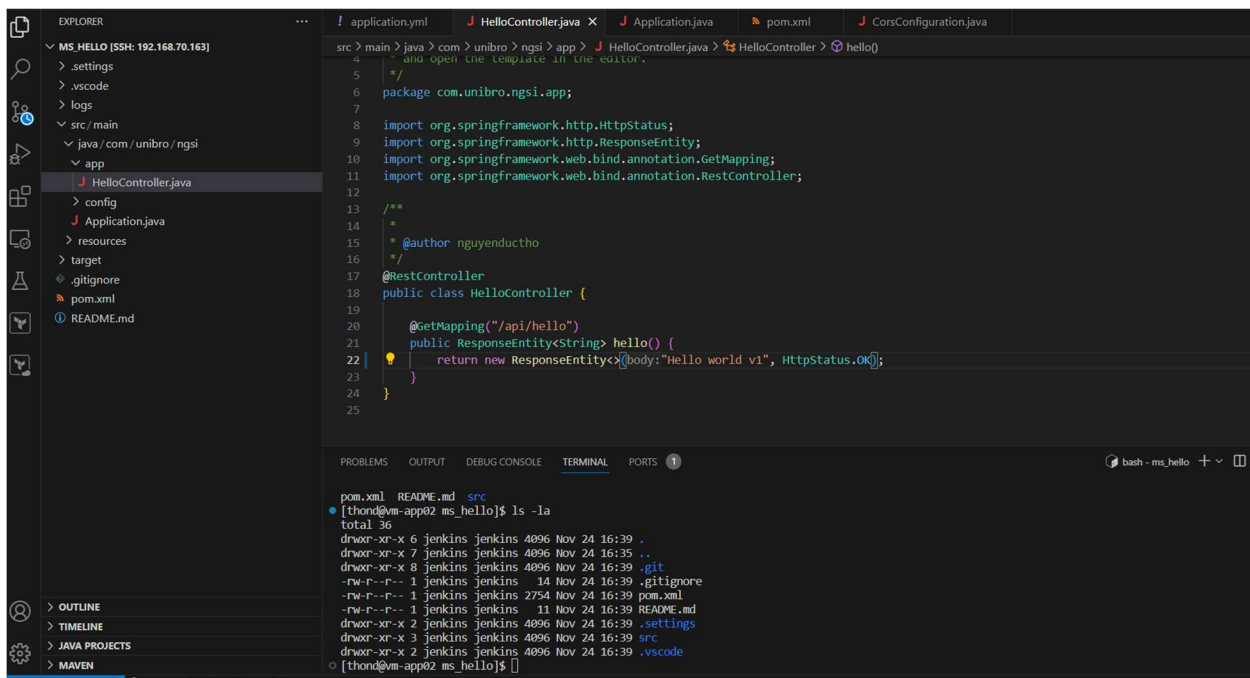
☐ With Ant

Tiến hành save item và tiến hành build thử xem jenkins có khả năng kết nối và tải source code về



Như vậy các bước khai báo đã trên jenkins đã xong bây giờ thử test commit code mới xem jenkins sẽ tự động tải code mới về.

Update source code trong file hello world thành Hello world v1








Commit code:


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1


[main 4ffc555] update Hello world v1
1 file changed, 1 insertion(+), 1 deletion(-)
• [thond@vm-app02 ms_hello]$ git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 789 bytes | 394.00 KiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/thodt6/ms_hello.git
d22481a..4ffc555 main -> main
○ [thond@vm-app02 ms_hello]$
```


Sau khi commit code lập tức trên jenkins đã tự động tải code mới về như bên dưới


 **Jenkins**  2  1  admin  log out


Dashboard > ms_hello > #3


 Status


 Changes


 Console Output


 Edit Build Information


 Delete build '#3'

 Polling Log


 Git Build Data

 Previous Build


 **#3 (Nov 24, 2023, 4:48:54 PM)**


 Add description

Started 1 min 50 sec ago
Took 1.6 sec

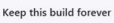
 Changes

1. update Hello world v1 ([details](#) / [githubweb](#))

 Started by [GitHub push by thodt6](#)

 Revision: 4ffc5551214311bb0df7cfd6c4e815d2700b752
Repository: https://github.com/thodt6/ms_hello.git

• [refs/remotes/origin/main](#)

 Keep this build forever

Dashboard > ms_hello > #3 > Changes

 Status

 Changes

 Console Output

 Edit Build Information

 Delete build '#3'

 Polling Log

 Git Build Data

 Previous Build

 **Changes**

Summary

1. update Hello world v1 ([details](#))

Commit 4ffc5551214311bb0df7cfd6c4e815d2700b752 **by** [Nguyen Duc Tho](#)
update Hello world v1

 [src/main/java/com/unibro/ngsi/app/HelloController.java \(diff\)](#)

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a search bar. Below the navigation bar, the breadcrumb trail is 'Dashboard > ms_hello > #3 > Console Output'. On the left sidebar, there are links for 'Status', 'Changes', 'Console Output' (which is selected), 'View as plain text', 'Edit Build Information', 'Delete build '#3'', 'Polling Log', 'Git Build Data', and 'Previous Build'. The main area is titled 'Console Output' with a green checkmark icon. It displays the following log text:

```
Started by GitHub push by thodt6
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/ms_hello
The recommended git tool is: NONE
No credentials specified
> /usr/bin/git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/ms_hello/.git # timeout=10
Fetching changes from the remote Git repository
> /usr/bin/git config remote.origin.url https://github.com/thodt6/ms_hello.git # timeout=10
Fetching upstream changes from https://github.com/thodt6/ms_hello.git
> /usr/bin/git --version # timeout=10
> git --version # 'git version 2.34.1'
> /usr/bin/git fetch --tags --force --progress -- https://github.com/thodt6/ms_hello.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> /usr/bin/git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4ffc5551214311bb0df7cfff6c4e815d2700b752 (refs/remotes/origin/main)
> /usr/bin/git config core.sparsecheckout # timeout=10
> /usr/bin/git checkout -f 4ffc5551214311bb0df7cfff6c4e815d2700b752 # timeout=10
Commit message: "update Hello world v1"
> /usr/bin/git rev-list --no-walk d22481ab3dbf3a2a840b3cb7023fec67f7e0a0c2 # timeout=10
Finished: SUCCESS
```

Như vậy quy trình thực hiện commit code -> github -(webhook)-> Jenkins đã thực hiện xong. Mỗi khi bạn commit code lập tức webhook sẽ gọi Jenkins để thực hiện update source code và thực hiện các bước build như bên dưới.

Bước 5: Cấu hình buildsteps

Trong phần này sẽ hướng dẫn các bước buildsteps sau:

- Step 1: Build lại code sử dụng mvn
- Step 2: Thực hiện lệnh stop dịch vụ ms_hello tại máy chủ cần deploy thông qua ssh remote bash call
- Step 3: Copy code build tới thư mục máy chủ deploy
- Step 4: Start lại dịch vụ ms_hello tại máy chủ deploy

Tiến hành test kết quả trên máy deploy bằng cách cập nhật code lên Hello world v3.

Step1: Tiến hành build code ra file chạy ms_hello.jar bằng maven

Chọn BuildStep và tiến hành Add Build Step

← ↻ https://jenskin.nguyenductho.pro/job/ms_hello/configure

Import favorites | New tab | Install and configur... | mmvdisk pdisk com... | recommend

Dashboard > ms_hello > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps
- Post-build Actions

- ☐ Send files or execute commands over SSH after the build runs
- ☐ Add timestamps to the Console Output
- ☐ Execute shell script on remote host using ssh ?
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Add build step ^

Filter

- Execute Windows batch command
- Execute shell
- Execute shell script on remote host using ssh
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets**
- Run with timeout
- Send files or execute commands over SSH
- Set build status to "pending" on GitHub commit

Build Steps

≡

Invoke top-level Maven targets ?

×

Maven Version

mvn

Goals

clean install

Advanced ▾

Add build step ▾

Step này sẽ thực hiện lệnh mvn clean install và build ra file ms_hello.jar trong thư mục target. Để xem step này có hoạt động không ta tiến hành build thử

Kết quả:

```
Dashboard > ms_hello > #4 > Console Output

[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ ngisi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/ms_hello/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ ngisi ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ ngisi ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ ngisi ---
[INFO] Building jar: /var/lib/jenkins/workspace/ms_hello/target/ngisi-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.1.RELEASE:repackage (repackage) @ ngisi ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ ngisi ---
[INFO] Installing /var/lib/jenkins/workspace/ms_hello/target/ms_hello.jar to /var/lib/jenkins/.m2/repository/com/unibro/ngisi/0.0.1-SNAPSHOT/ngisi-0.0.1-SNAPSHOT.jar
[INFO] Installing /var/lib/jenkins/workspace/ms_hello/pom.xml to /var/lib/jenkins/.m2/repository/com/unibro/ngisi/0.0.1-SNAPSHOT/ngisi-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.109s
[INFO] Finished at: Fri Nov 24 17:04:30 ICT 2023
[INFO] Final Memory: 23M/90M
[INFO] -----
Finished: SUCCESS
```

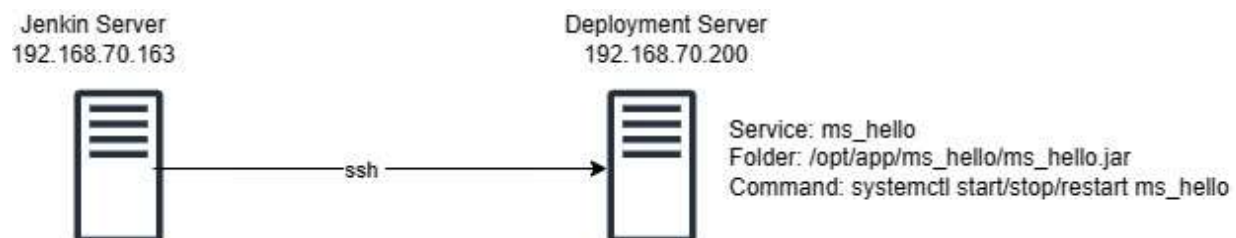
Thư mục trên workspace của jenkins đã build xong:

```
[thond@vm-app02 ms_hello]$ cd /var/lib/jenkins/workspace/ms_hello/
[thond@vm-app02 ms_hello]$ ls
pom.xml  README.md  src  target
[thond@vm-app02 ms_hello]$ cd target/
[thond@vm-app02 target]$ ls
classes  generated-sources  maven-archiver  maven-status  ms_hello.jar  ngisi-0.0.1-SNAPSHOT.jar
[thond@vm-app02 target]$ pwd
/var/lib/jenkins/workspace/ms_hello/target
[thond@vm-app02 target]$
```

Trước khi bước sang bước 2 ta sẽ chuẩn bị một máy chủ deploy để triển khai code ms-hello.jar. Trên máy chủ này tôi sẽ khai báo một service có tên là ms_hello và thực hiện chạy lệnh sau để thực hiện stop/start dịch vụ:

- systemctl stop ms_hello
- systemctl start ms_hello
- systemctl restart ms_hello

Mô hình kết nối:



Thư mục lưu đường file chạy ms_hello: /opt/app/ms_hello

Tạo một file /etc/systemd/system/ms_hello.service với nội dung như sau:

```
[Unit]
Description=ms_hello
After=syslog.target

[Service]
User=thond
ExecStart=/opt/app/ms_hello/ms_hello.jar
SuccessExitStatus=162

[Install]
WantedBy=multi-user.target
```

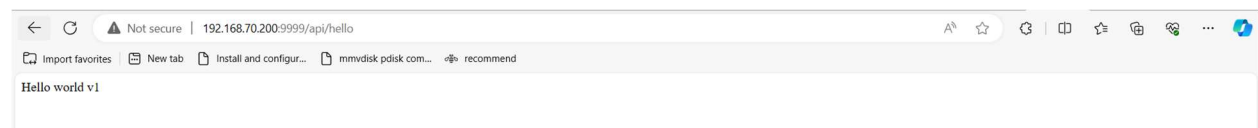
Copy file ms_hello.jar đã biên dịch sang thư mục /opt/app/ms_hello trên máy chủ deployment thông qua lệnh

scp /var/lib/jenkins/ms_hello/ms_hello.jar thond@192.168.70.200:/opt/app/ms_hello

thực hiện enable service, start service

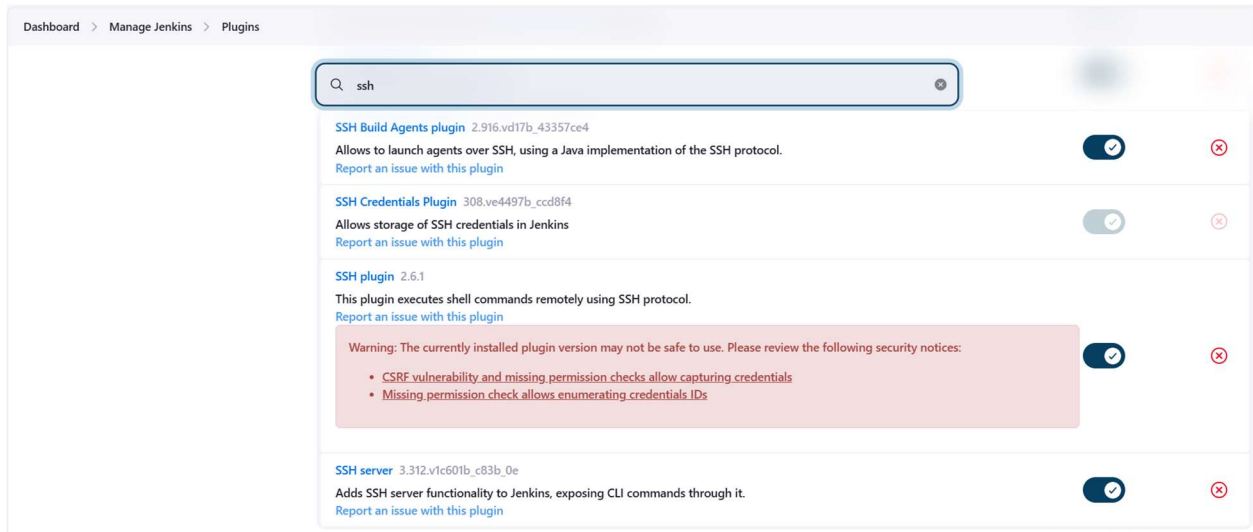
```
root@VM-Elastic thond]# systemctl enable ms_hello
Created symlink from /etc/systemd/system/multi-user.target.wants/ms_hello.service to /etc/systemd/system/ms_hello.service.
root@VM-Elastic thond]# systemctl start ms_hello
root@VM-Elastic thond]# systemctl status ms_hello
● ms_hello.service - ms hello
   Loaded: loaded (/etc/systemd/system/ms_hello.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-11-24 17:24:23 +07; 6s ago
     Main PID: 24402 (ms_hello.jar)
    Tasks: 36
```

Test thử dịch vụ đã chạy:



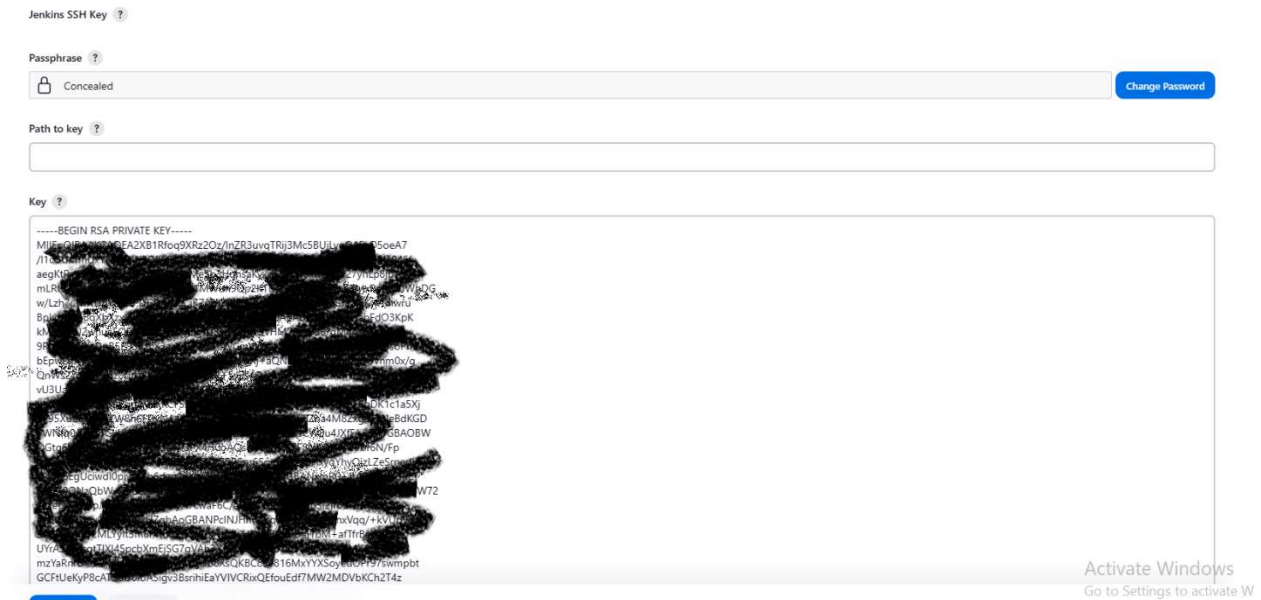
Như vậy máy chủ Deployment đã có thể chạy dịch vụ khi đẩy file đã build lên. Tiếp theo ta sẽ thực hiện các bước buildstep để có thể tự động build source code khi commit version mới và stop dịch vụ trên máy chủ Deployment thông qua ssh remote call plugin(dùng lệnh systemctl stop ms_hello) trước khi đẩy bản build mới sang, copy bản build mới thông qua remote copy ssh plugin và sau khi copy xong sẽ tiến hành start lại dịch vụ thông qua lệnh systemctl start ms_hello)

Step2: Tiến hành thực hiện lệnh systemctl stop ms_hello trên máy chủ Deployment thông qua 2 plugin cài đặt như bên dưới là SSH Plugin và SSH Server plugin



Để thực hiện remote call hoặc remote copy ta tiến hành cấu hình một số tham số sau:

- Cấu hình private key trong mục Manage Jenkins => System => Publish over SSH.
Private key này giúp máy chủ Jenkins có thể thực hiện ssh hoặc scp thông qua private key này.



- Cấu hình SSH server:

SSH Servers

SSH Server

Name ?

production-server

Hostname ?

192.168.70.200

Username ?

thond

Remote Directory ?

/opt/app/

☐ Avoid sending files that have not changed ?

Advanced ▾

Test Configuration

Sau đó ta sẽ tiến hành tạo thêm build step bằng cách thêm build step “Execute shell script on remote host using ssh”

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Invoke top-level Maven targets ?

Maven Version

mvn

Goals

clean install

Advanced ▾

Add build step ▾

Filter

Execute Windows batch command

Execute shell

Execute shell script on remote host using ssh

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Send files or execute commands over SSH

Set build status to "pending" on GitHub commit

Execute shell script on remote host using ssh

SSH site

thond@192.168.70.200:22

Command

sudo systemctl stop ms_hello

☐ Execute each line ?

☐ Hide command from console output

Step này sẽ tiến hành dừng dịch vụ ở máy chủ Deployment trước khi copy bản build mới

Step3: Tiến hành copy bản build tới máy chủ Deployment thông qua remote ssh copy plugin

Dashboard > ms_hello > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Advanced

Execute shell script on remote host using ssh

SSH site

thond@192.168.70.200:22

Command

sudo systemctl stop ms_hpires

☐ Execute each line ?

☐ Hide command from console output

Add build step

Filter

Execute Windows batch command

Execute shell

Execute shell script on remote host using ssh

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Send files or execute commands over SSH

Set build status to "pending" on GitHub commit

Send files or execute commands over SSH ?

SSH Publishers

SSH Server

Name ?

production-server [No Exec]

Advanced

Transfers

Transfer Set

Source files ?

target/ms_hello.jar

Remove prefix ?

target

Remote directory ?

ms_hello

Exec command ?

Trong đó:

- SSH Server: Là server đã khai báo ở bước trên
- Source files: là thư mục lưu bản build. ở đây bản build sẽ lưu ra thư mục target/ms_hello.jar
- Remote prefix: là thư mục target do ta chỉ copy file chứ ko copy thư mục target
- Remote directory: Thư mục lưu ở máy chủ Deployment: Ở đây thư mục base của ssh server là /opt/app do vậy thư mục tuyệt đối là /opt/app/ms_hello. Do vậy ta cần điền tên ms_hello vào phần này

Như vậy bước này đã tiến hành copy bản build mới đè lên bản build cũ trong máy chủ Deployment

Step4: Tiến hành thực hiện lệnh `systemctl start ms_hello` trên máy chủ Deployment sau khi đã copy bản build mới

The screenshot shows the Jenkins configuration interface. At the top, there is a section for 'Exec command' with a text input field and a help icon. Below this, a note states: 'All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)'. There are two expandable sections: 'Advanced' (collapsed) and 'Add Transfer Set' (collapsed). Below these are 'Add Server' and 'Advanced' (collapsed) buttons. At the bottom, the 'Add build step' dropdown menu is open, showing a list of build steps. The 'Execute shell script on remote host using ssh' option is highlighted. The list of build steps includes: Execute Windows batch command, Execute shell, Execute shell script on remote host using ssh, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, Send files or execute commands over SSH, and Set build status to 'pending' on GitHub commit.

Exec command ?

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Advanced ▾

Add Transfer Set

Add Server

Advanced ▾

Add build step ▾

Filter

- Execute Windows batch command
- Execute shell
- Execute shell script on remote host using ssh
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Send files or execute commands over SSH
- Set build status to "pending" on GitHub commit

Execute shell script on remote host using ssh

SSH site

thond@192.168.70.200:22

SSH Site not specified

Command

```
sudo systemctl start ms_hello
```

☐ Execute each line ?

☐ Hide command from console output

Add build step

Như vậy cơ bản các bước build đã hoàn thành. Sau khi commit code lập tức bản build này sẽ được triển khai trên máy chủ Deployment.

Bước 6: Test commit code sang Hello world v2.

EXPLORER

MS_HELLO [SSH: 192.168.70.163]

> .settings

> .vscode

> logs

> src/main

> java/com/unibro/ngsi

app

HelloController.java

> config

Application.java

> resources

> target

> .gitignore

pom.xml

README.md

application.yml

HelloController.java

Application.java

pom.xml

CorsConfiguration.java

```

src > main > java > com > unibro > ngsi > app > HelloController.java > HelloController > hello()
4  * and open the template in the editor.
5  */
6  package com.unibro.ngsi.app;
7
8  import org.springframework.http.HttpStatus;
9  import org.springframework.http.ResponseEntity;
10 import org.springframework.web.bind.annotation.GetMapping;
11 import org.springframework.web.bind.annotation.RestController;
12
13 /**
14  *
15  * @author nguyenductho
16  */
17 @RestController
18 public class HelloController {
19
20     @GetMapping("/api/hello")
21     public ResponseEntity<String> hello() {
22         return new ResponseEntity<>(body:"Hello world v2", HttpStatus.OK);
23     }
24 }
25

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

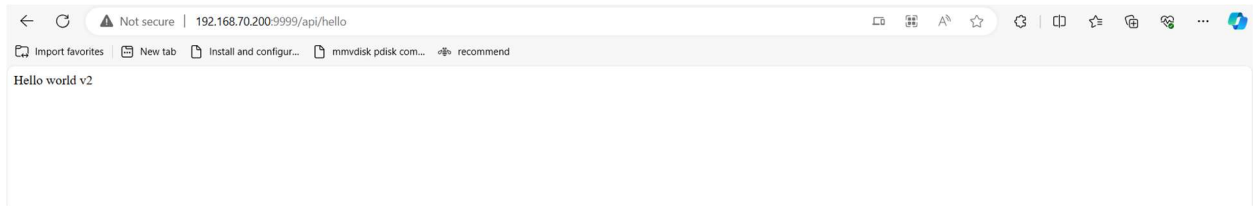
1

```

[main S1cacef] Hello world v2
1 file changed, 1 insertion(+), 1 deletion(-)
[thond@vm-app02 ms_hello]$ git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 786 bytes | 393.00 KiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/thodt6/ms_hello.git
4ffc555..51cacef main -> main
[thond@vm-app02 ms_hello]$

```

Sau khi commit code chờ một lúc lập tức trên server đã được deploy code mới



Như vậy toàn bộ quy trình tích hợp CI/CD cho ứng dụng Spring-boot sử dụng Jenkins đã thành công. Chúc các bạn vui vẻ