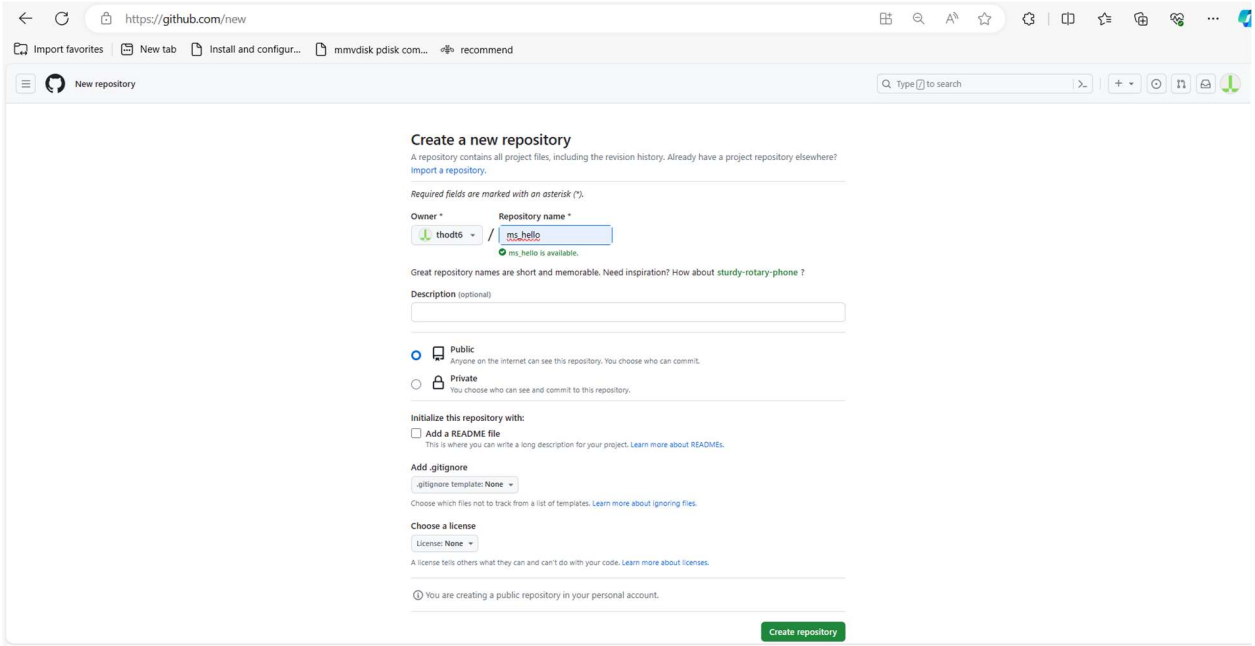
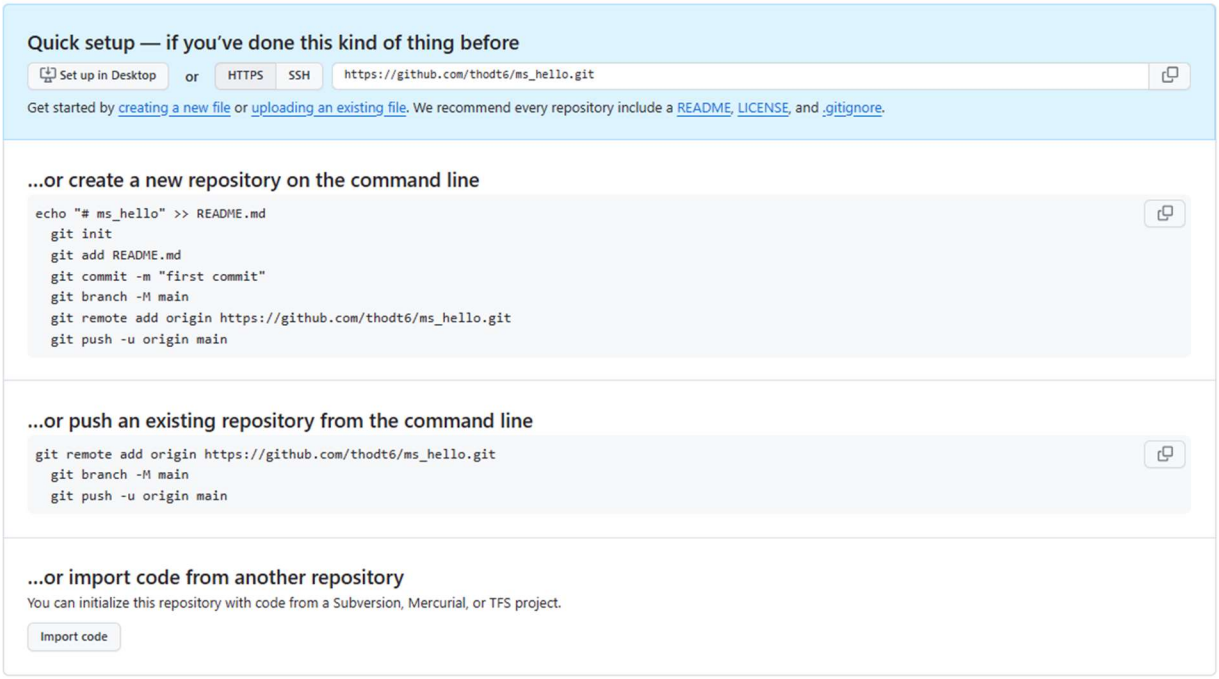
Trong bài viết này tôi sẽ mô tả việc sử dụng Jenskin để thực hiện tích hợp deploy source code là một ứng dụng Spring-boot trên Github về server test

Luồng thực hiện như sau:

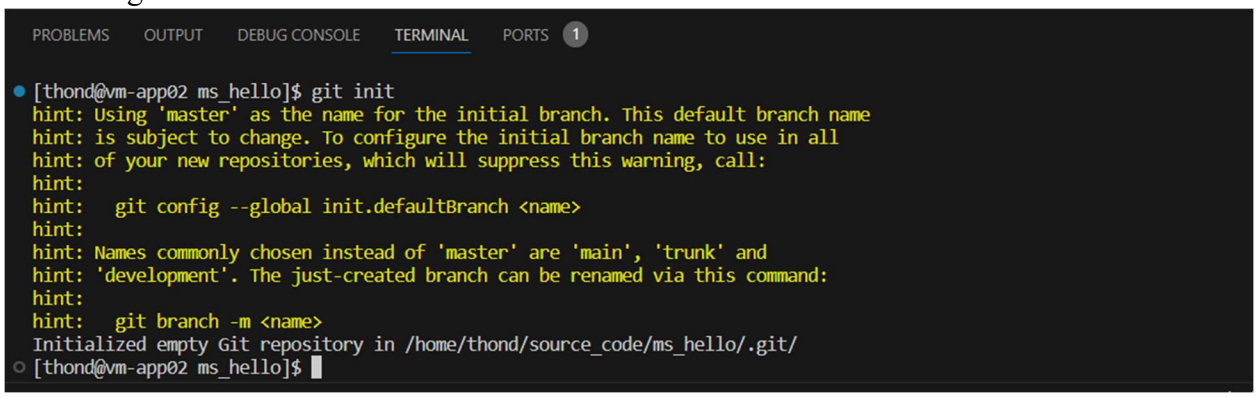
Bước 1: Chuẩn bị code Spring boot. Ở đây tôi sẽ xây dùng một ứng dụng spring-boot Hello world đơn giản theo các bước sau:

- Tạo project trên github

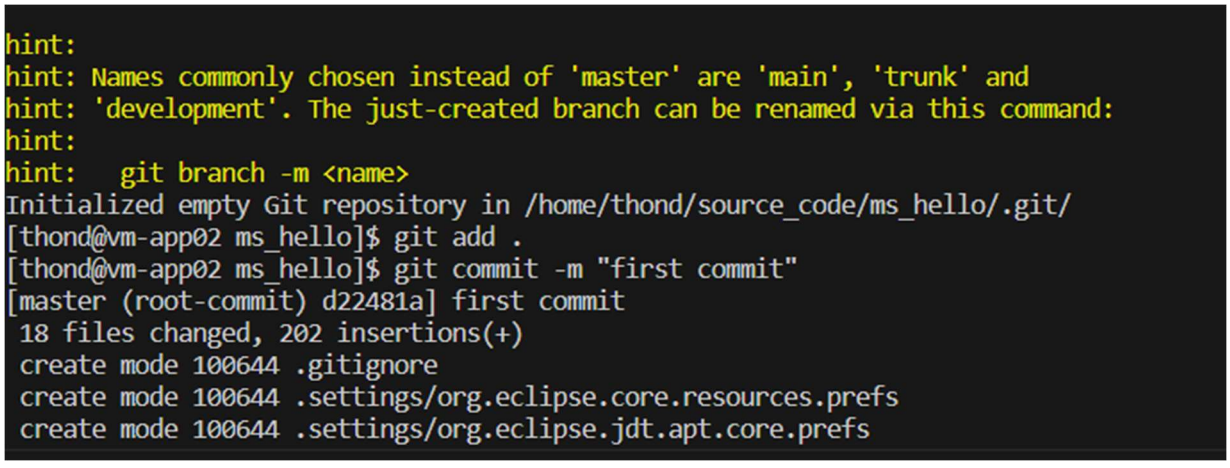




- Init local git source code:

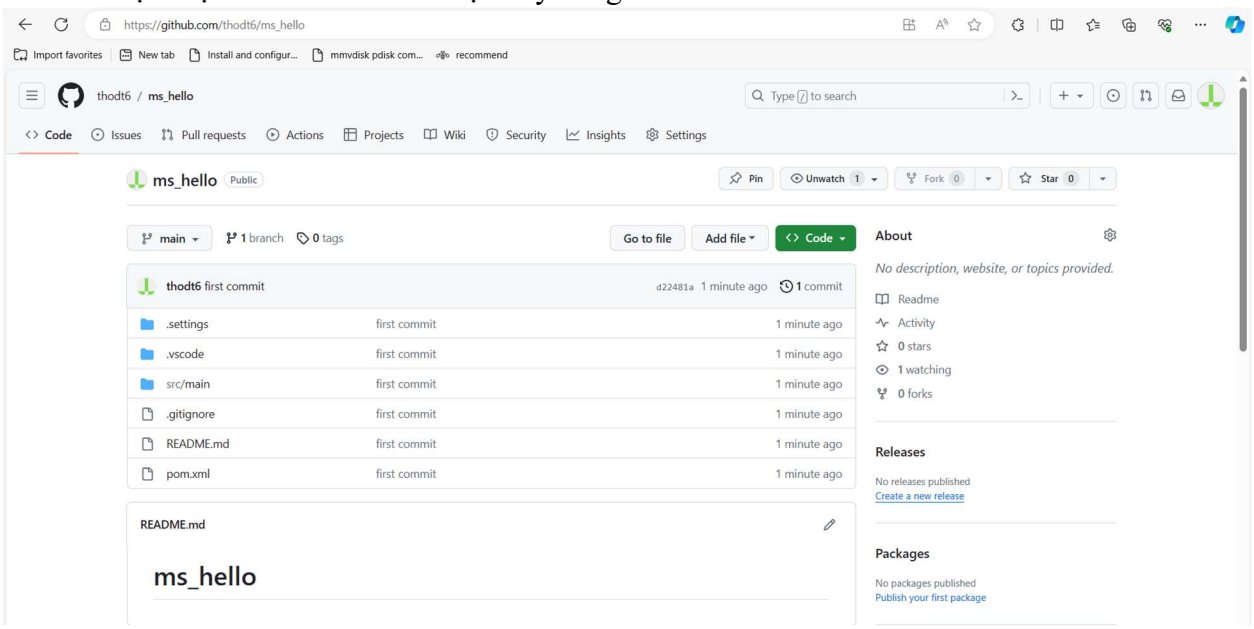


- Commit first helloworld version



```
[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 sampe source code: https://github.com/thodt6/ms_hello

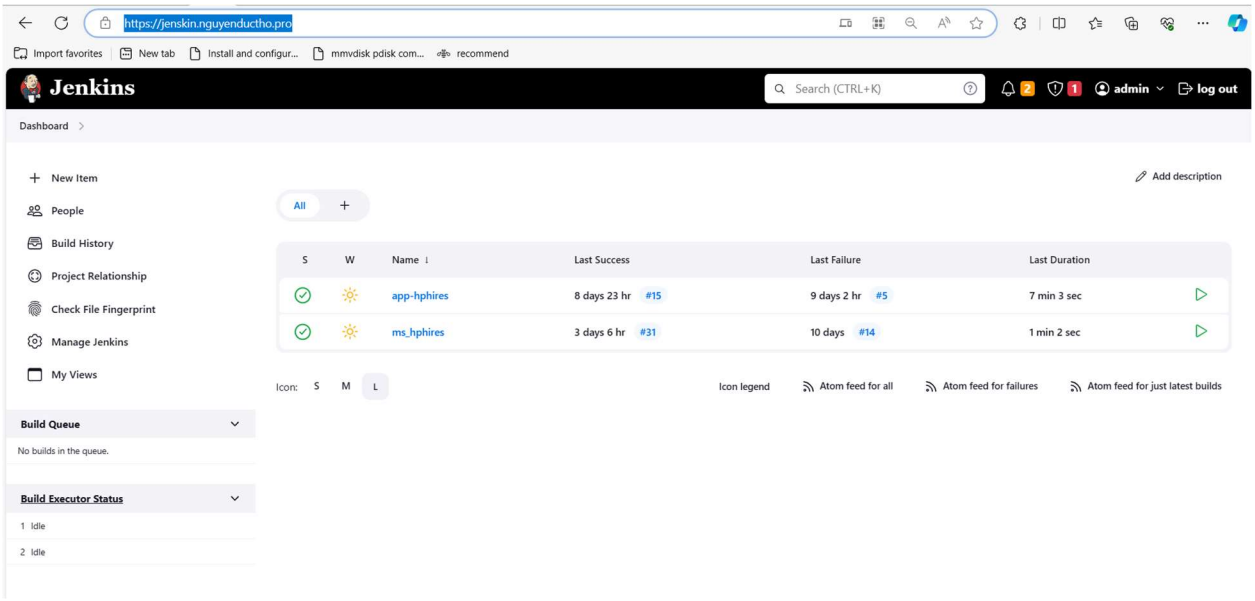
Như vậy bước một đã chuẩn bị xong một sample spring-boot code và có thể commit code đẩy lên github. Bước tiếp theo sẽ liên quan đến việc triển khai jenskin server

Bước 2: Chuẩn bị jenkins server + cài đặt plugin

Link hướng dẫn cài đặt jenskin sampel trên Linux

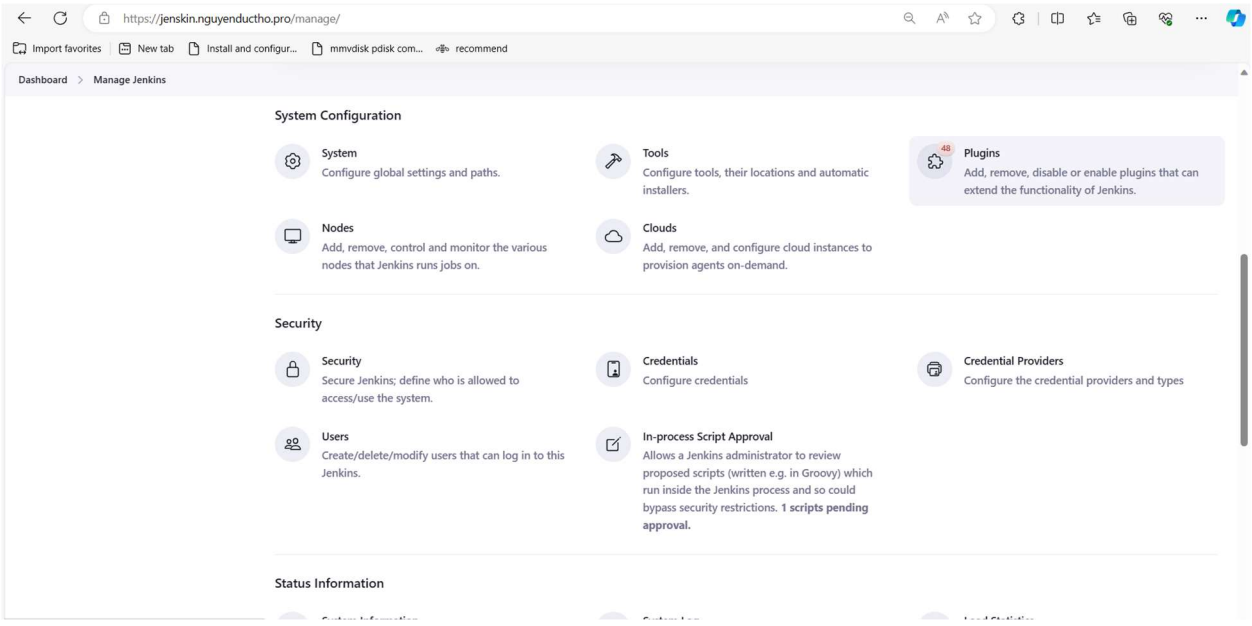
https://linuxize.com/post/how-to-install-jenkins-on-centos-7/

Trong ví dụ này tôi đã chuẩn bị môt jenskin server theo địa chỉ https://jenskin.nguyenductho.pro và đây là màn hình đăng nhập

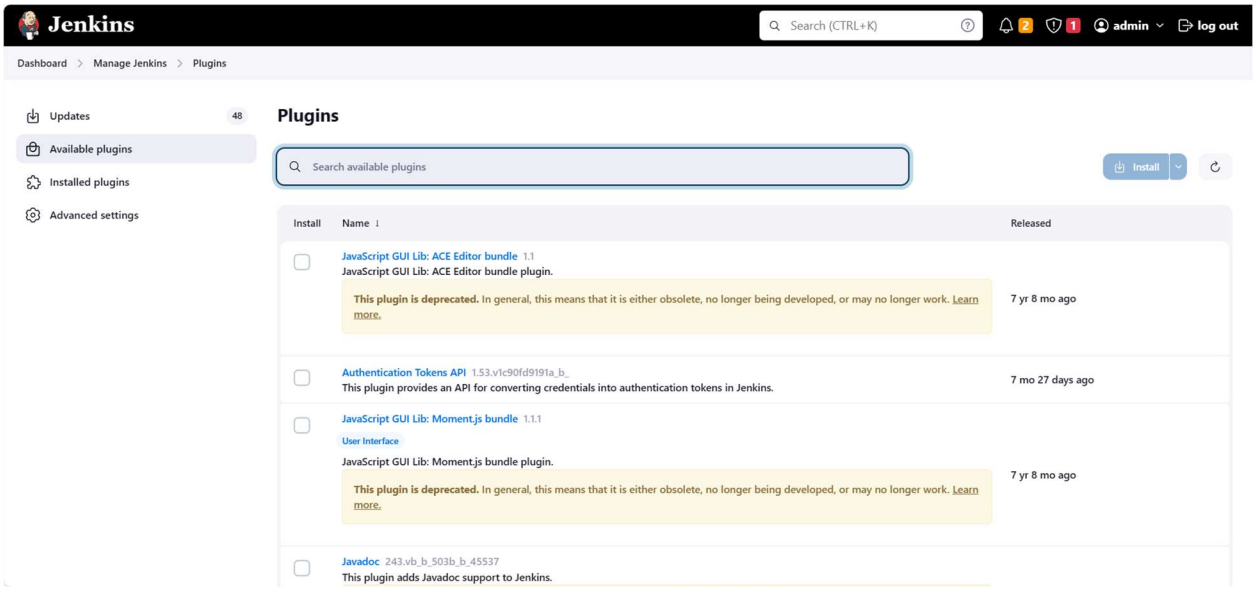


Ta có thể tiến hành login vào jenskin và chuẩn bị các bước khai báo build item. Tuy nhiên để thực hiện tích hợp với github ta cần cài đặt thêm github plugin theo các bước sau:

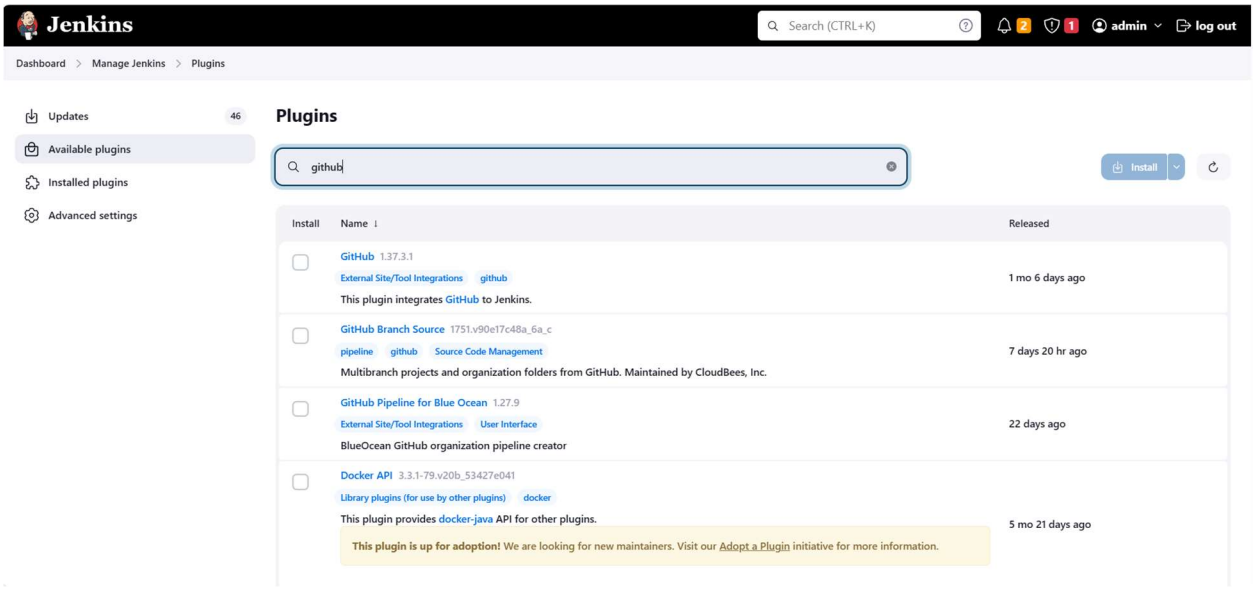
- Vào phần manage Jenskin -> Plugins

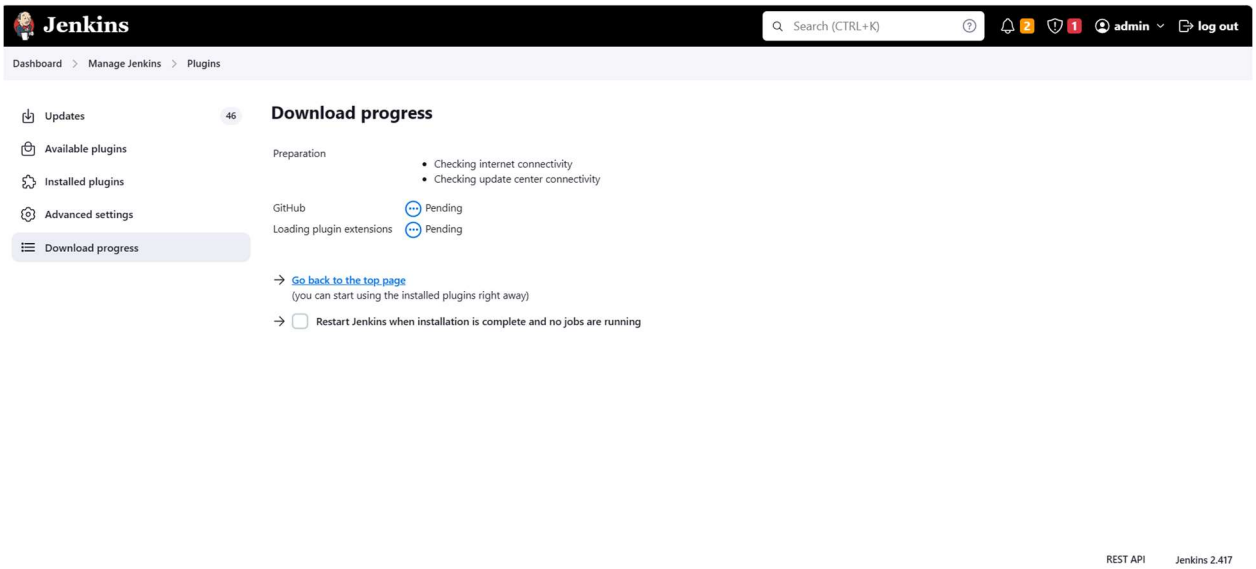


- Vào mục Available Plugins



- Search github và tiến hành cài đặt plugin

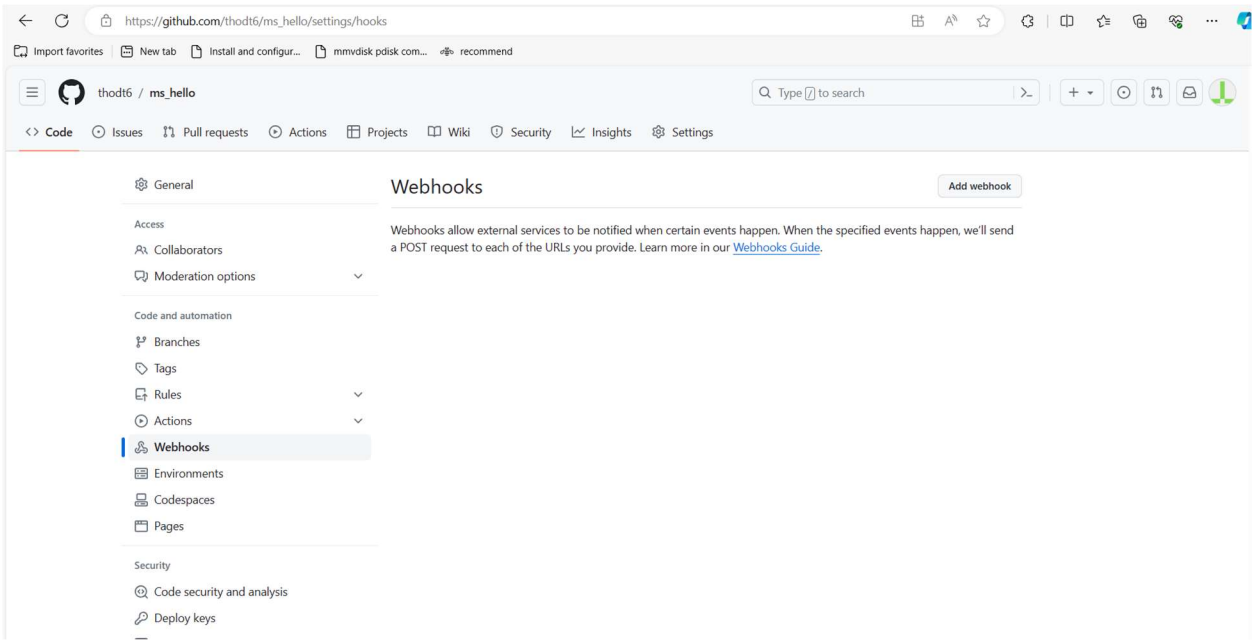


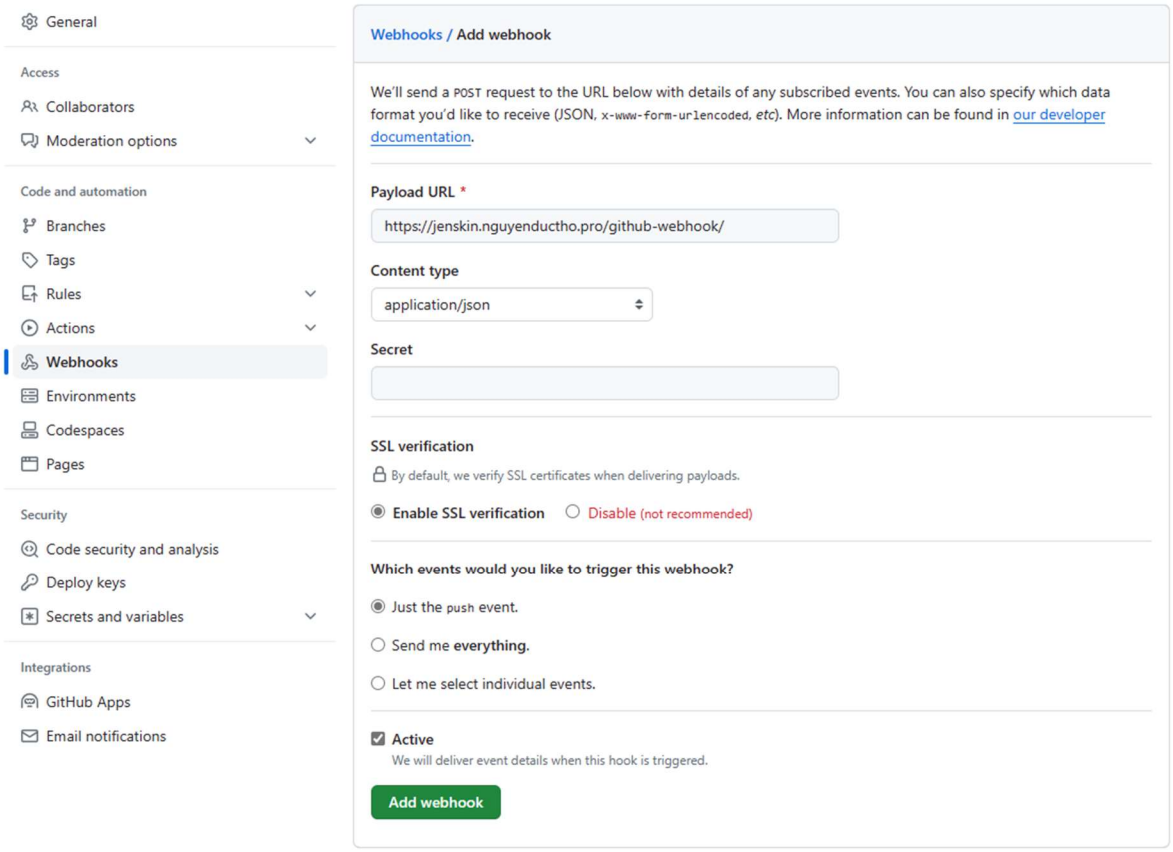


Click vào ô Restart Jenkins

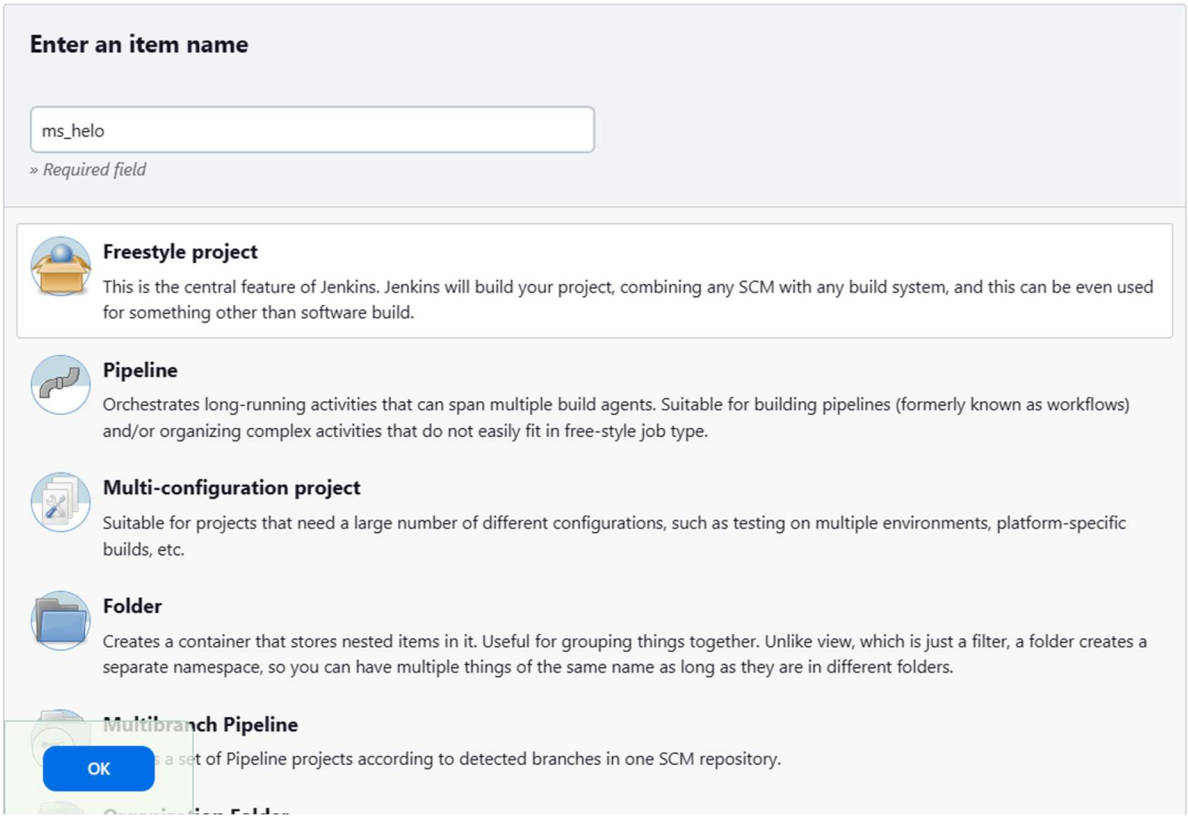
Bước 3: Tạo webhook cho Github

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

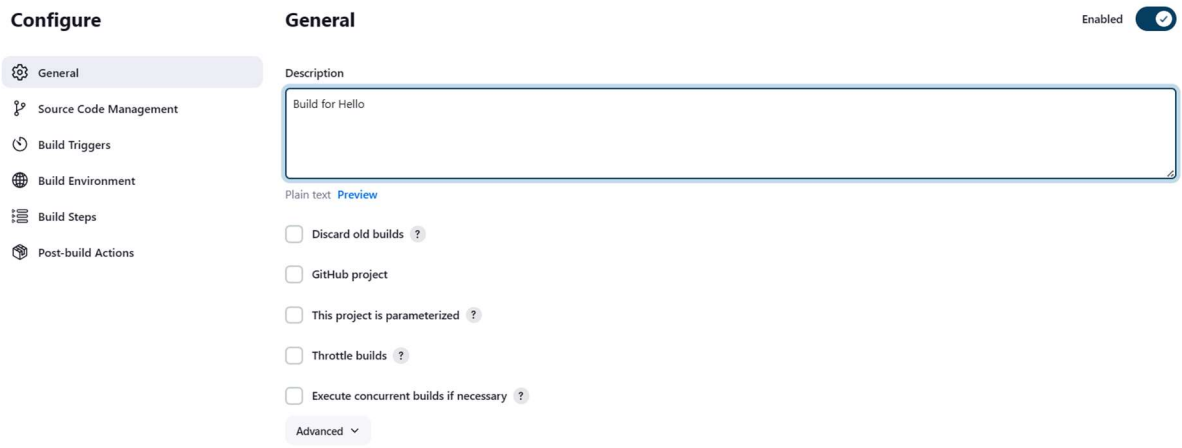




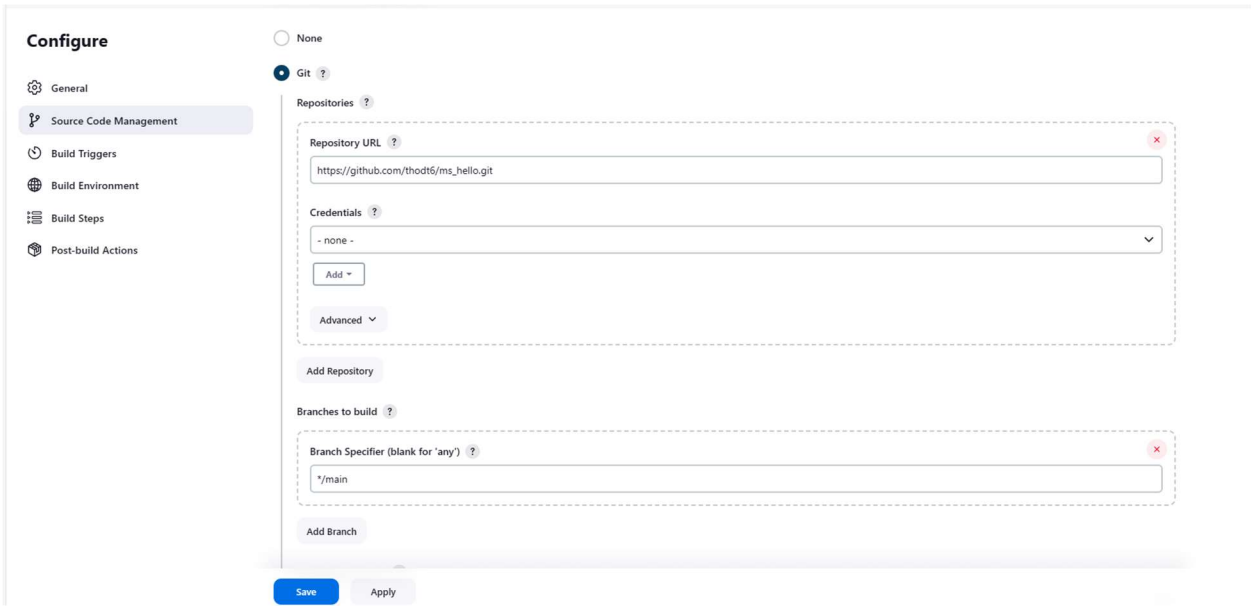
Bước 4: Tiến hành thực hiện cấu hình item mới cho Jenkins



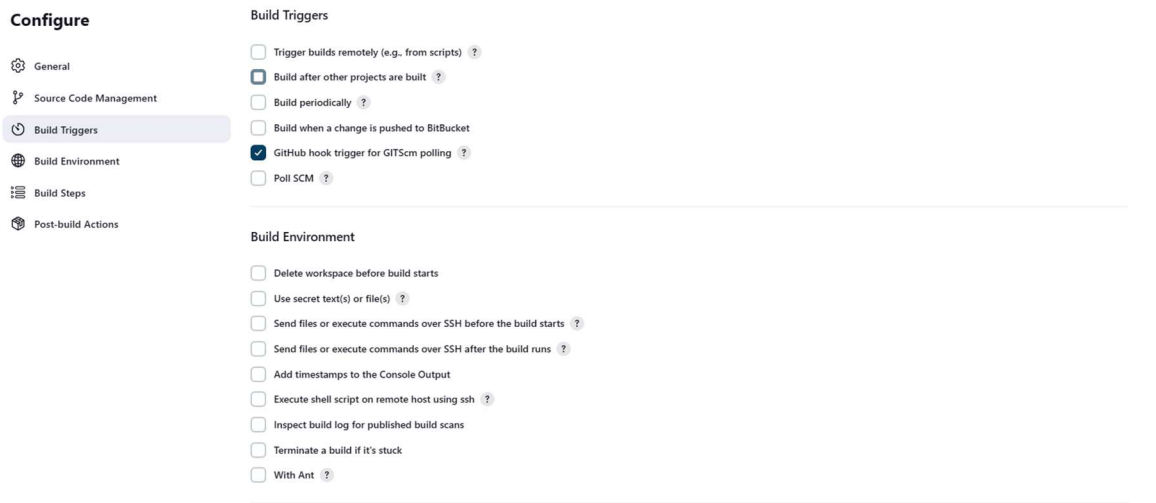
- Cập nhật các tham số của ms_hello item như bên dưới



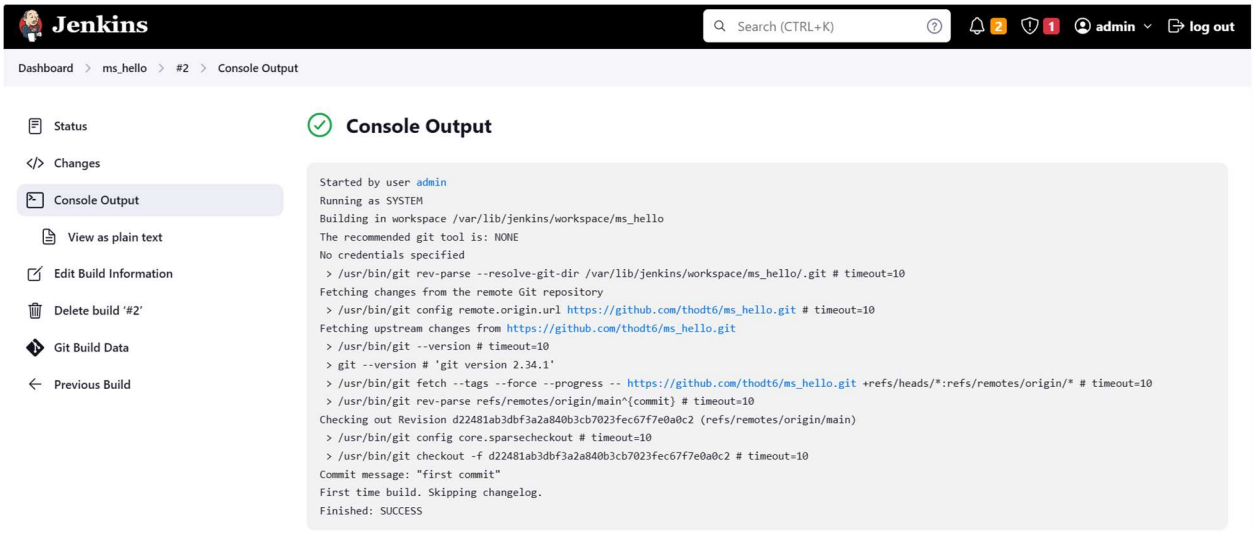
- Source Code Management: Chú ý Branch to build chọn đúng branch trên github



- Build trigger:

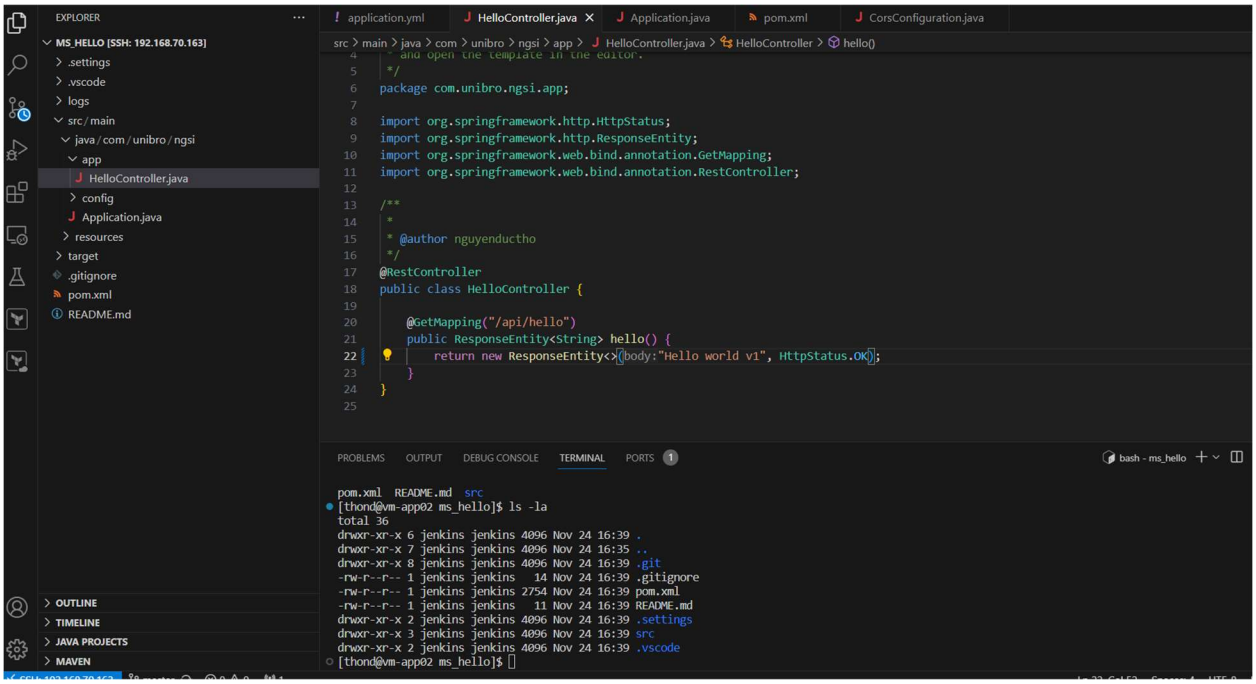


Tiến hành save item và tiến hành build thử xem jenkins có khả năng kết nối và tải source code về



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

Update source code trong file hello world thành Hello world v1

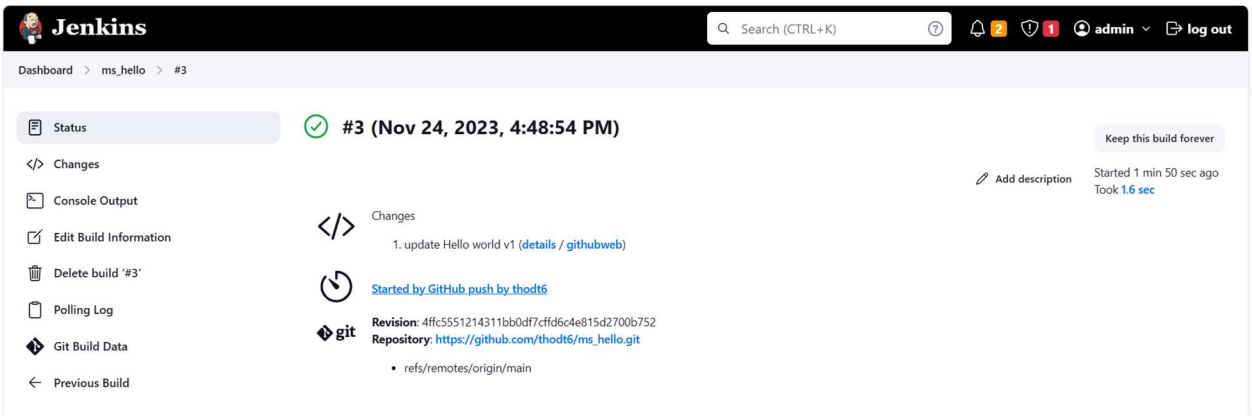


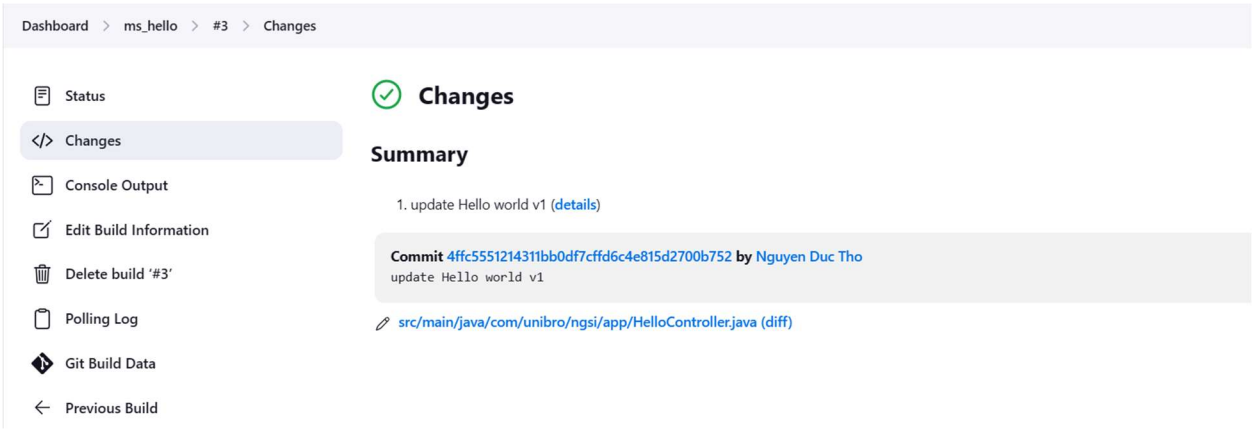
Commit code:

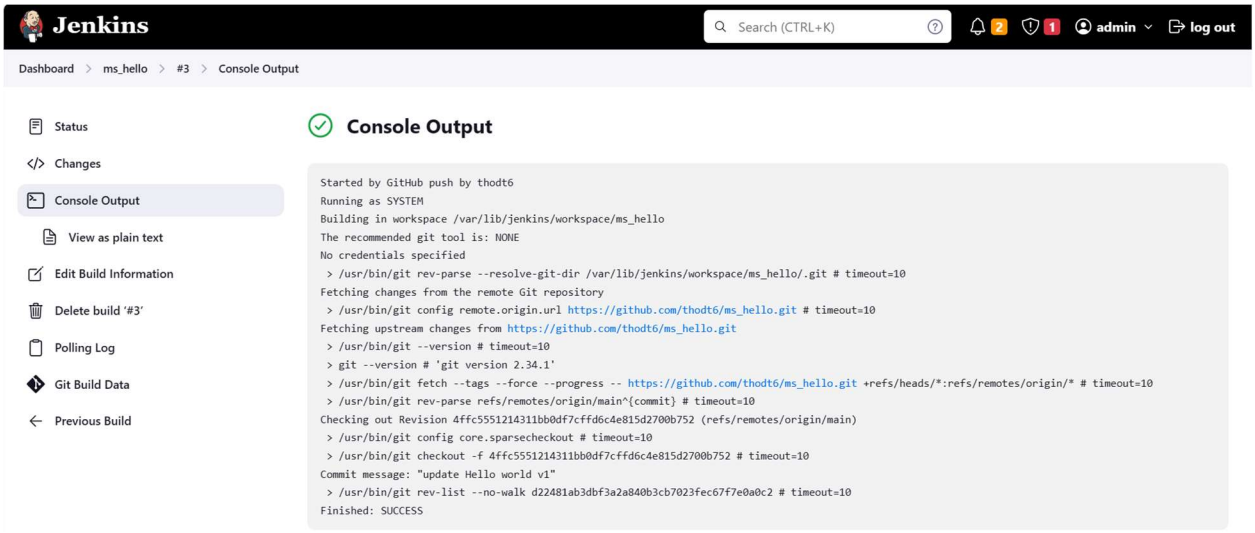
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 1

[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







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

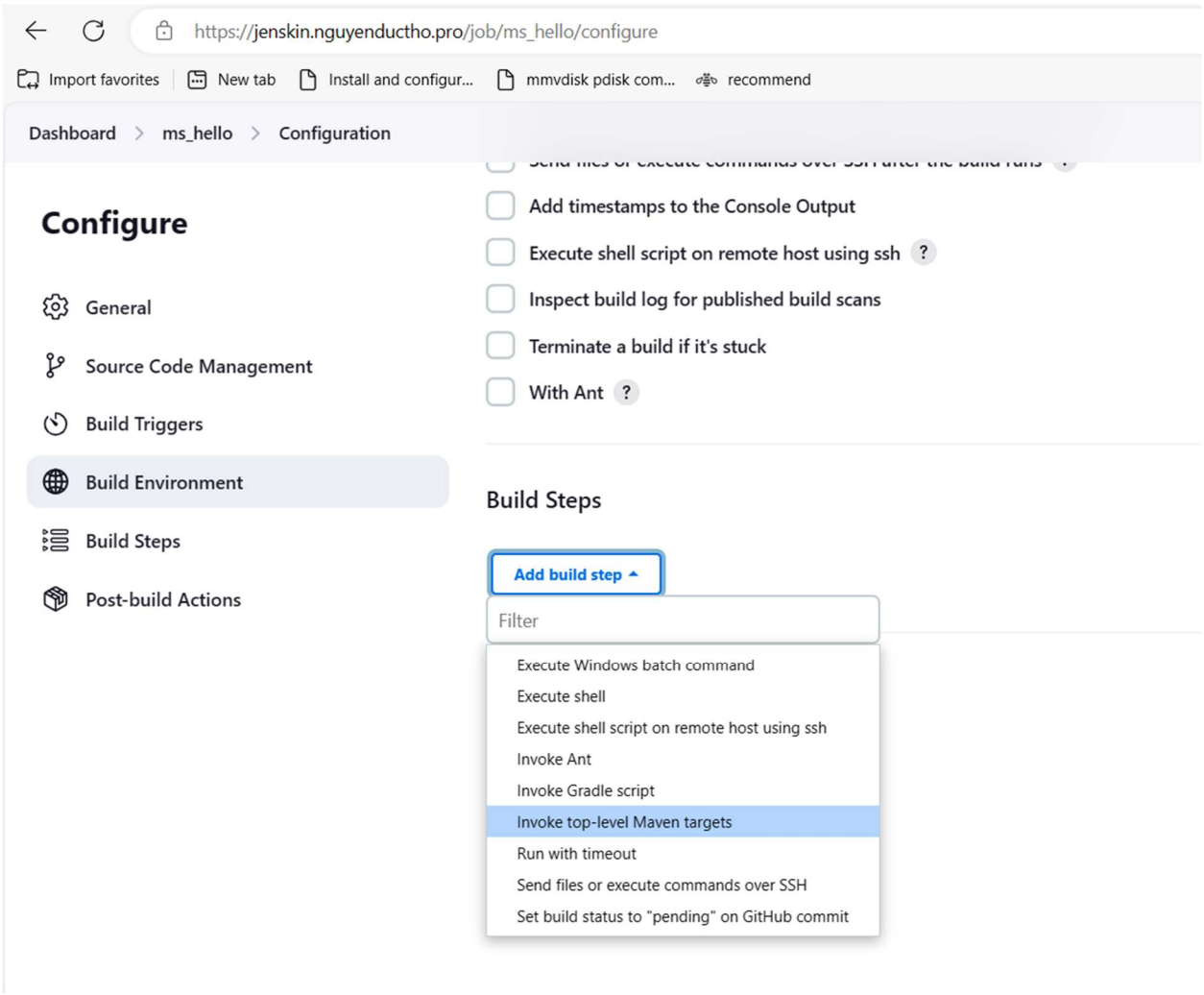
Trong phần này sẽ hướng dẫn các bước buildsteps sau:

- 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 kế quả trên máy deploy bằng cách cập nhật code lên Hello world v3.

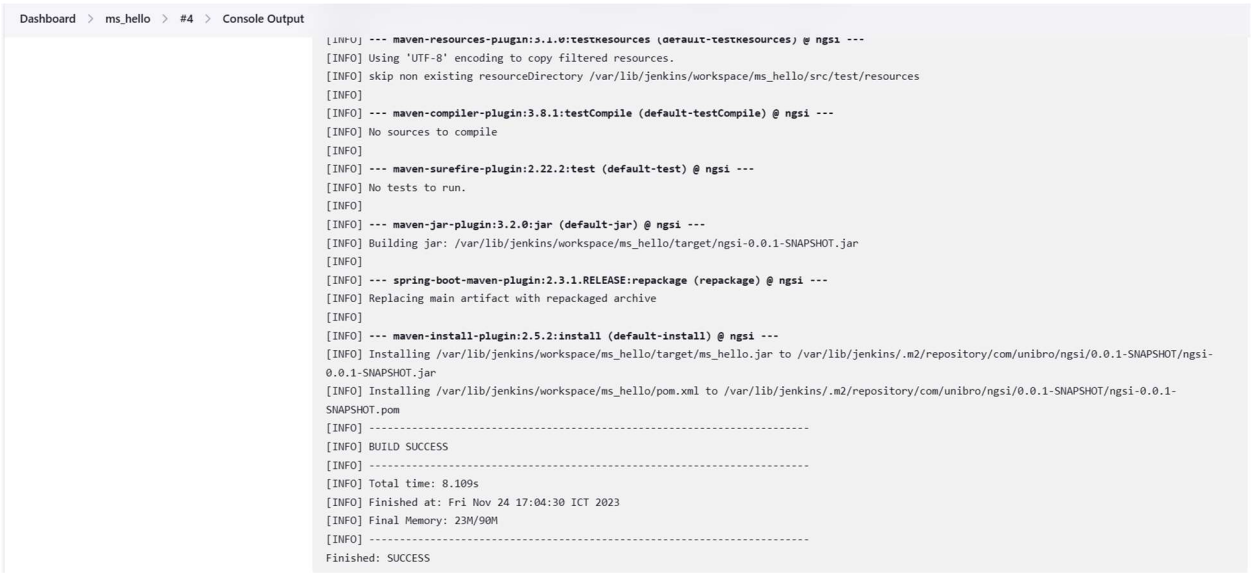
Step1: Tiến hành build code ra file chạy ms_hello.jar bằng maven

Chọn BuildStep và tiến hành Add Build Step



Step này sẽ thực hiên lện 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:



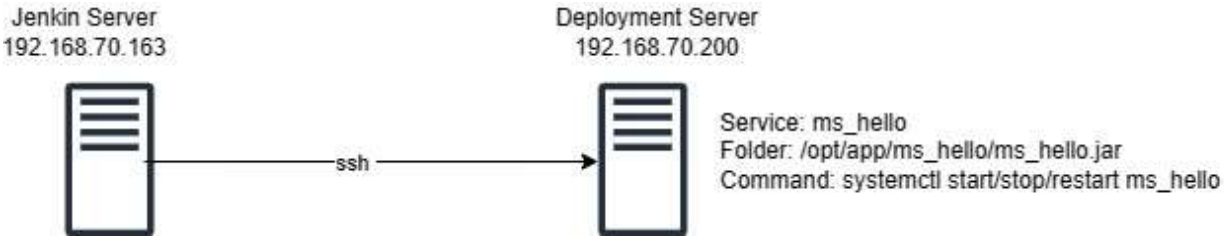
Thư mục trên workspace của jenkins đã build xong:

```
[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  ngsi-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
```

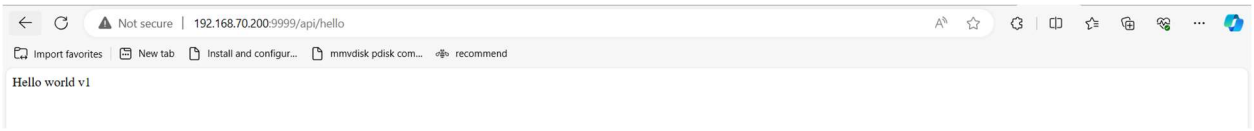
Copy file ms_hello.jar đã biên dịch sang thư mục /opt/app/ms_hello trên máy chủ deployment thông qua lệnh

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

thực hiện enable service, start service

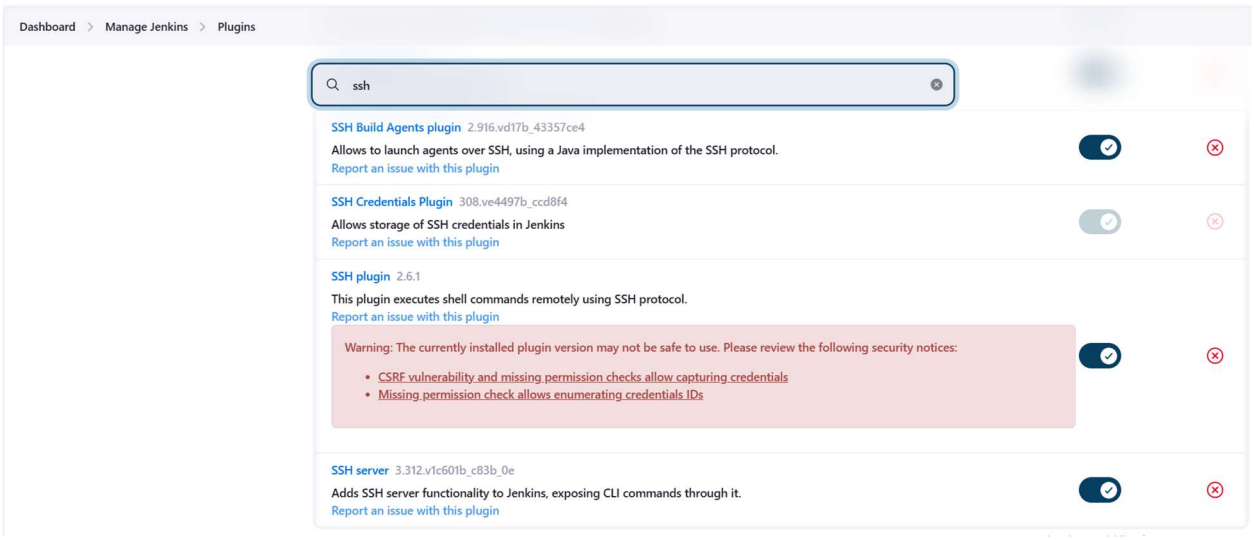
```
[root@VM-Elastic thond]# systemctl enable ms_hello
Created symlink from /etc/systemd/system/multi-user.target.wants/ms_hello.service to /etc/systemd/system/ms_hello.service.
[root@VM-Elastic thond]# systemctl start ms_hello
[root@VM-Elastic thond]# systemctl status ms_hello
● ms_hello.service - ms_hello
   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: 26
```

Test thử dịch vụ đã chạy:



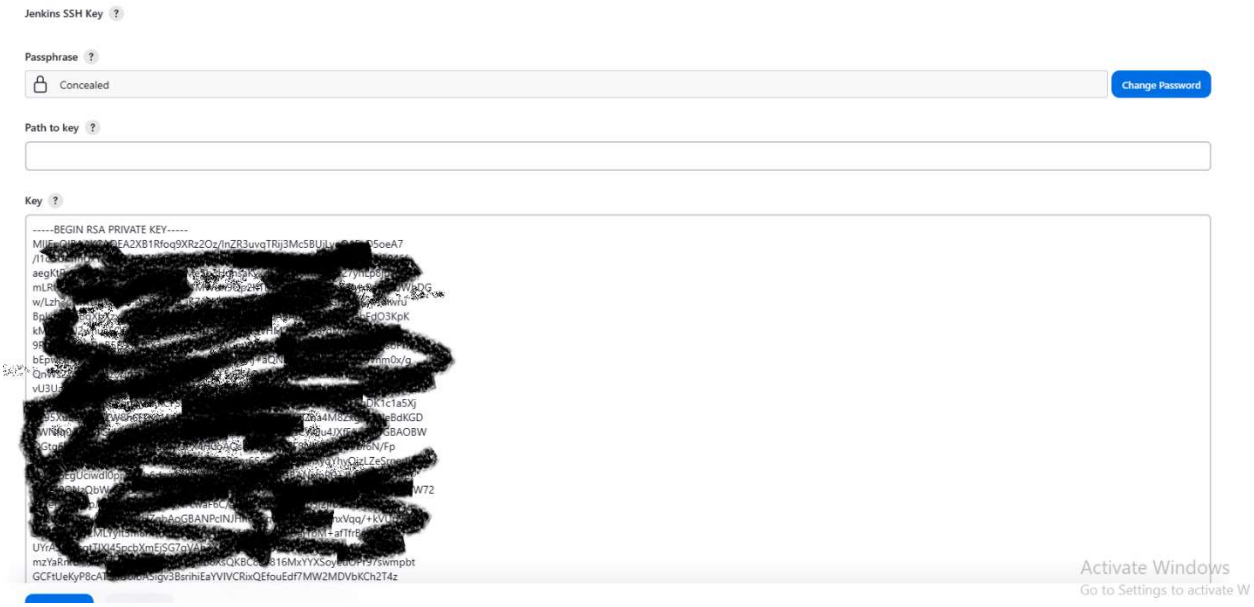
Như vậy máy chủ Deployment đã có thể chạy dịch vụ khi đẩy file đã build lên. Tiếp theo ta sẽ thực hiện các bước buildstep để có thể tự động build source code khi commit version mới và stop dịch vụ trên máy chủ Deployment thông qua ssh remote call plugin(dùng lệnh systemctl stop ms_hello) trước khi đẩy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



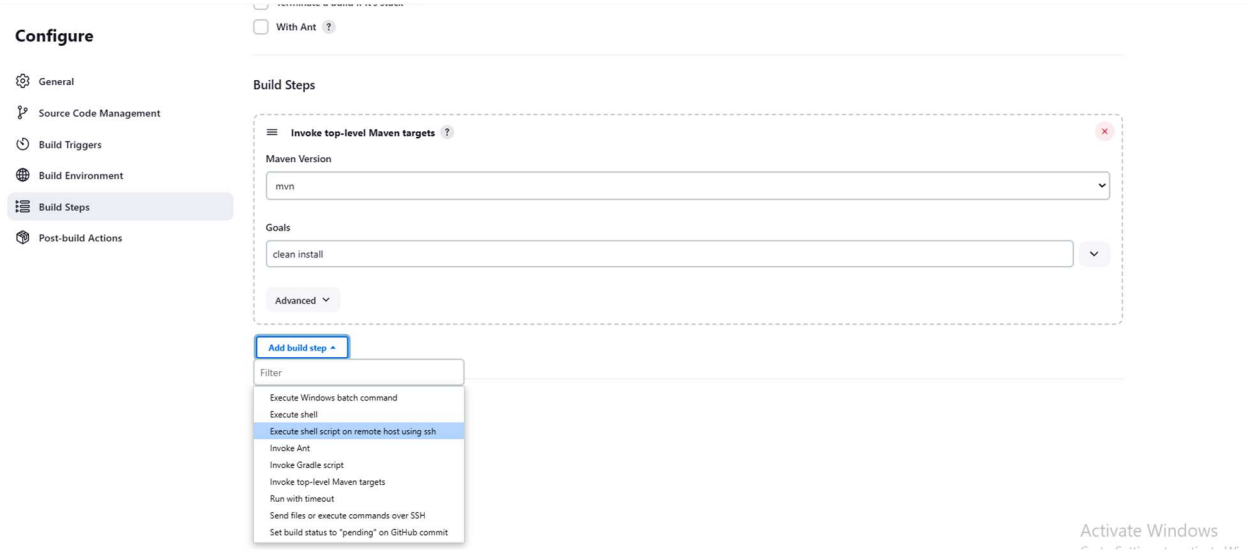
Để thực hiện remote call hoặc remote copy ta tiến hình cấu hình một số tham số sau:

- 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.



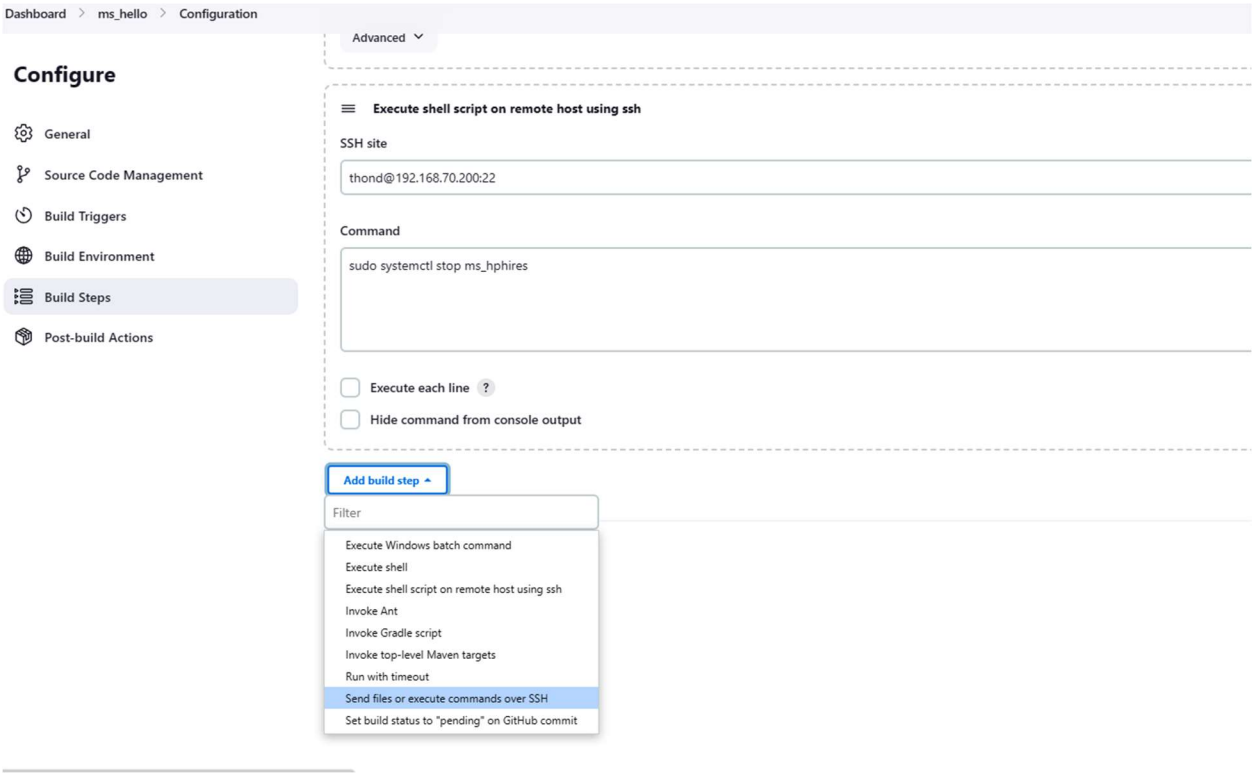
- Cấu hình SSH server:

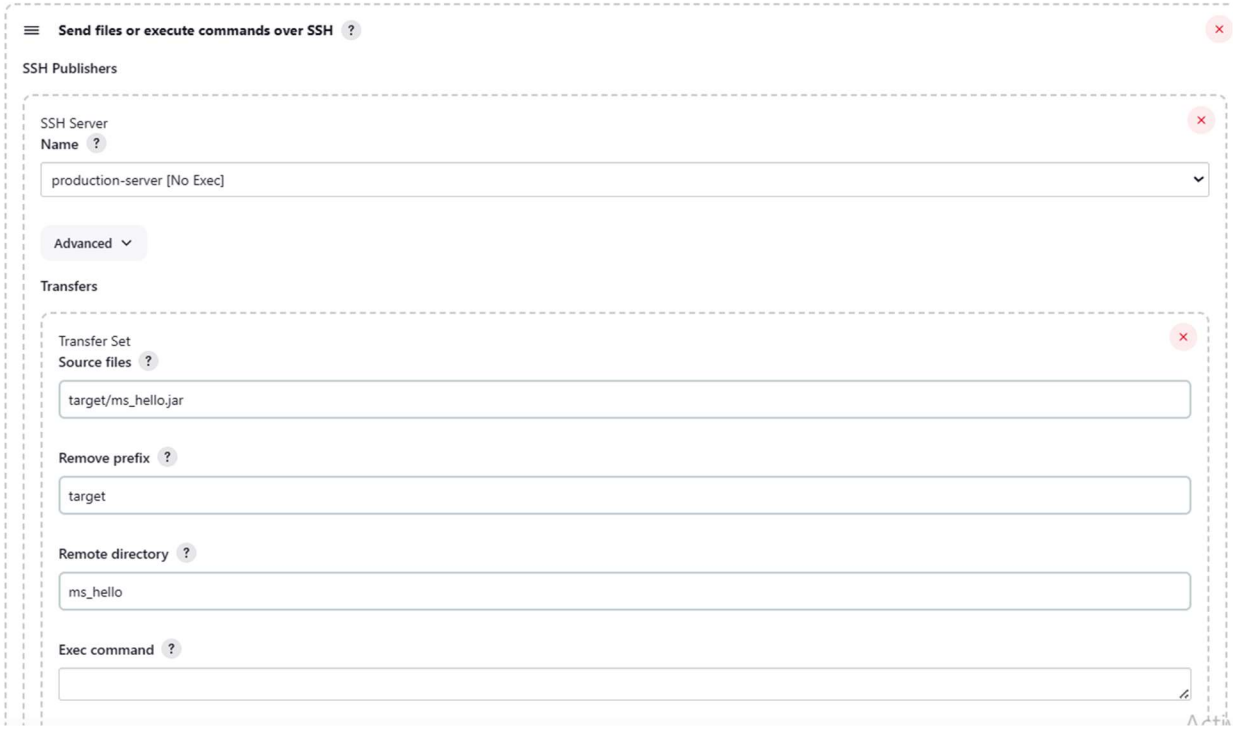
Sau đó ta sẽ tiến hành tạo thêm build step bằng cách thêm build step “Execute shell script on remote host using ssh”



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



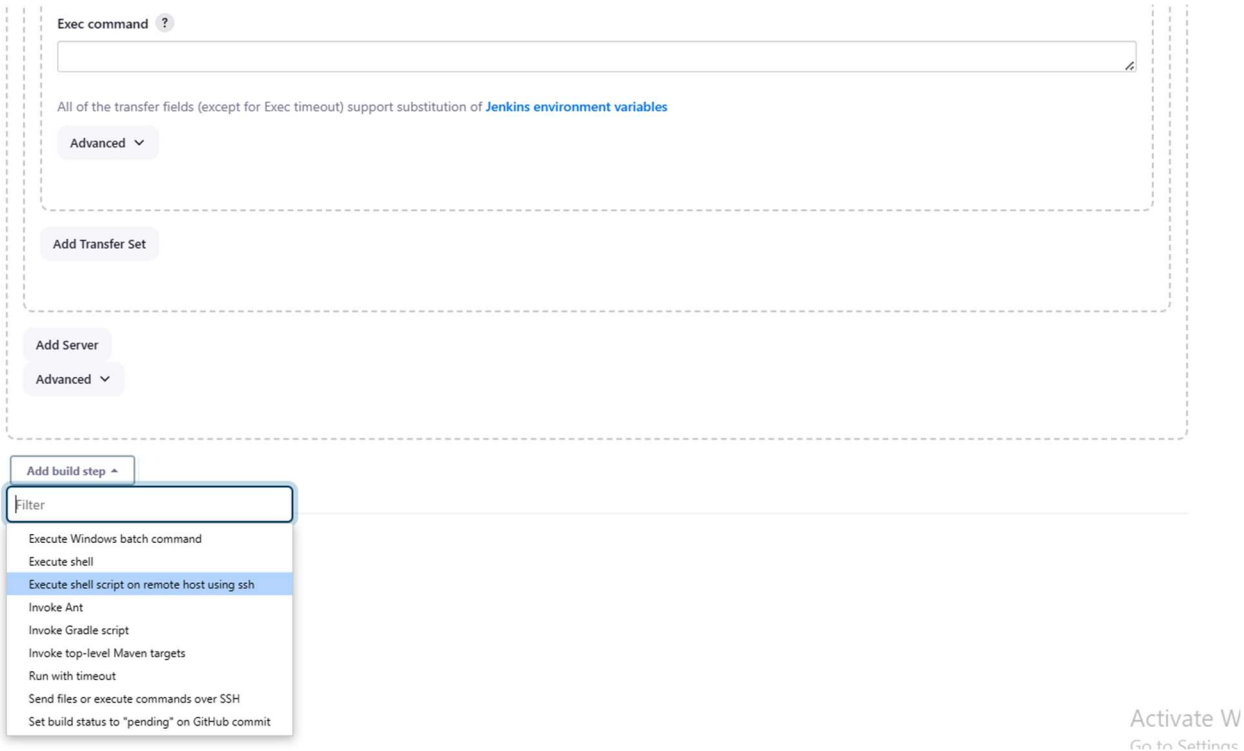


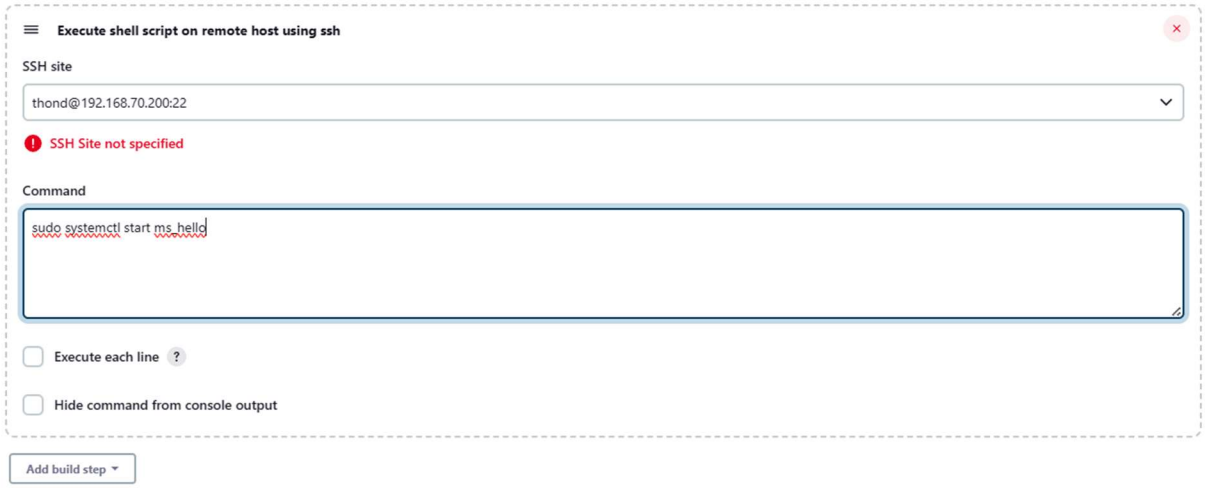
Trong đó:

- 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

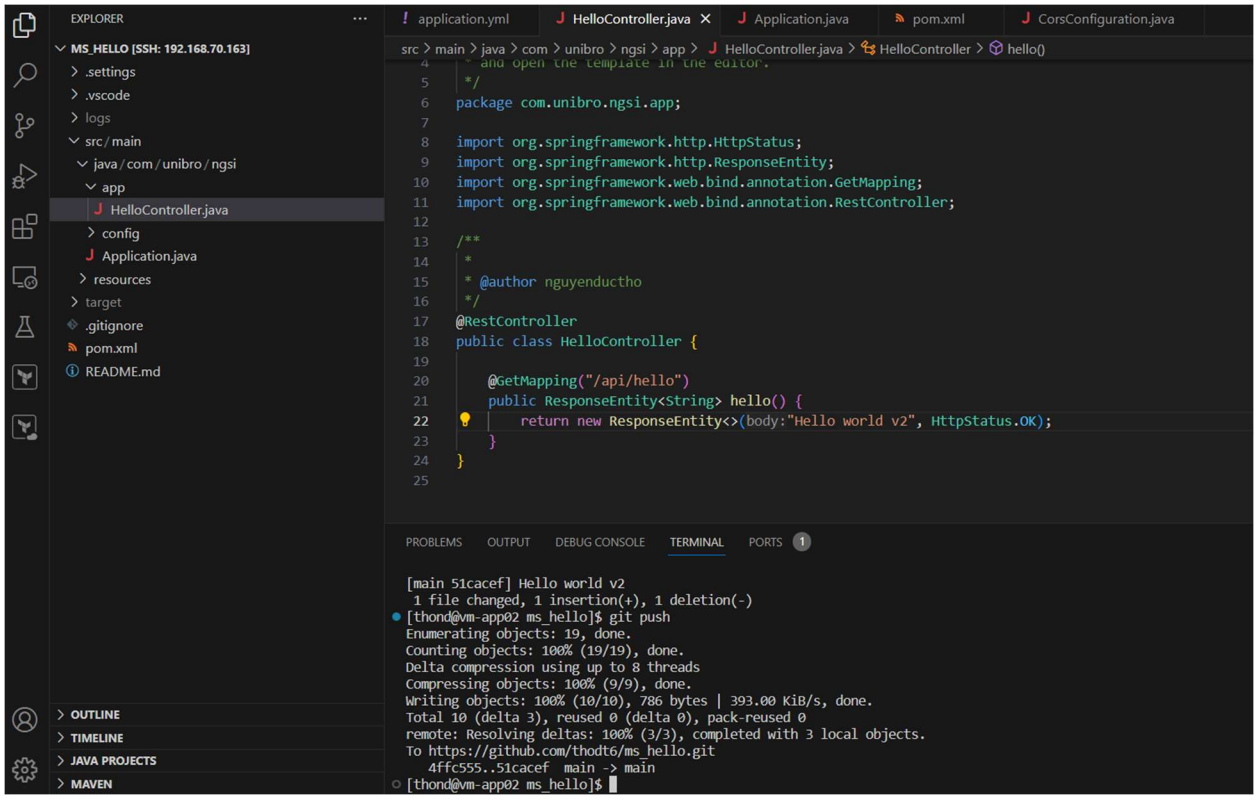
Step4: Tiến hành thực hiện lệnh systemctl start ms_hello trên máy chủ Deployment sau khi đã copy bản build mới



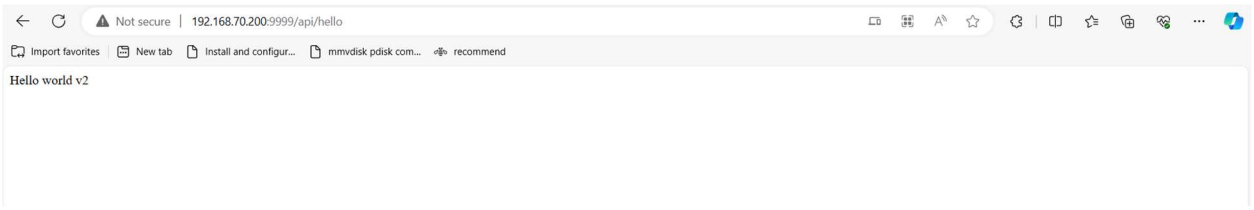


Như vậy cơ bản các bước build đã hoàn thành. Sau khi commit code lập tức bản build này sẽ được triển khai trên máy chủ Deployment.

Bước 6: Test commit code sang Hello world v2.



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