



CI/CD với Jenkins

Nguyễn Hàn Duy

duy@techmaster.vn

Nội dung



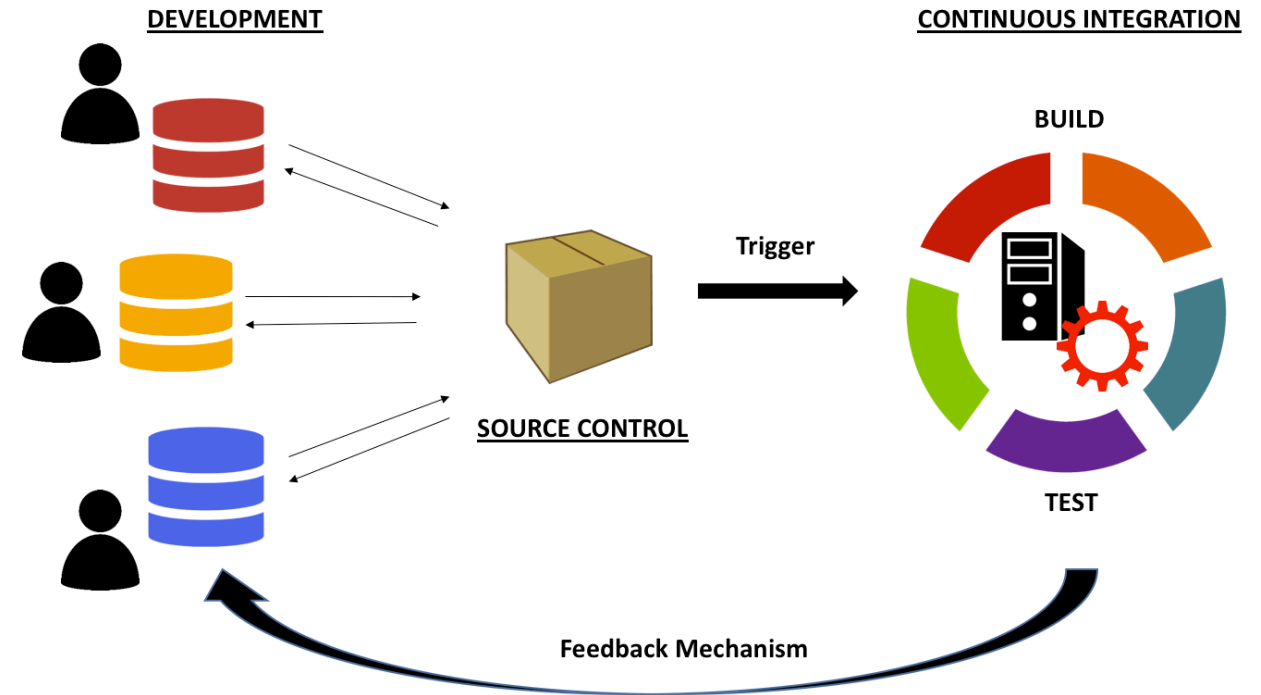
- Giới thiệu CI/CD và Jenkins
- Cài đặt Jenkins
- Kiến trúc Jenkins
- Jenkins pipeline - Jenkinsfile
- Tích hợp Jenkins với Git remote repo
- Tự động build và push Docker Image

Giới thiệu CI/CD và Jenkins

CI – Continuous Integration

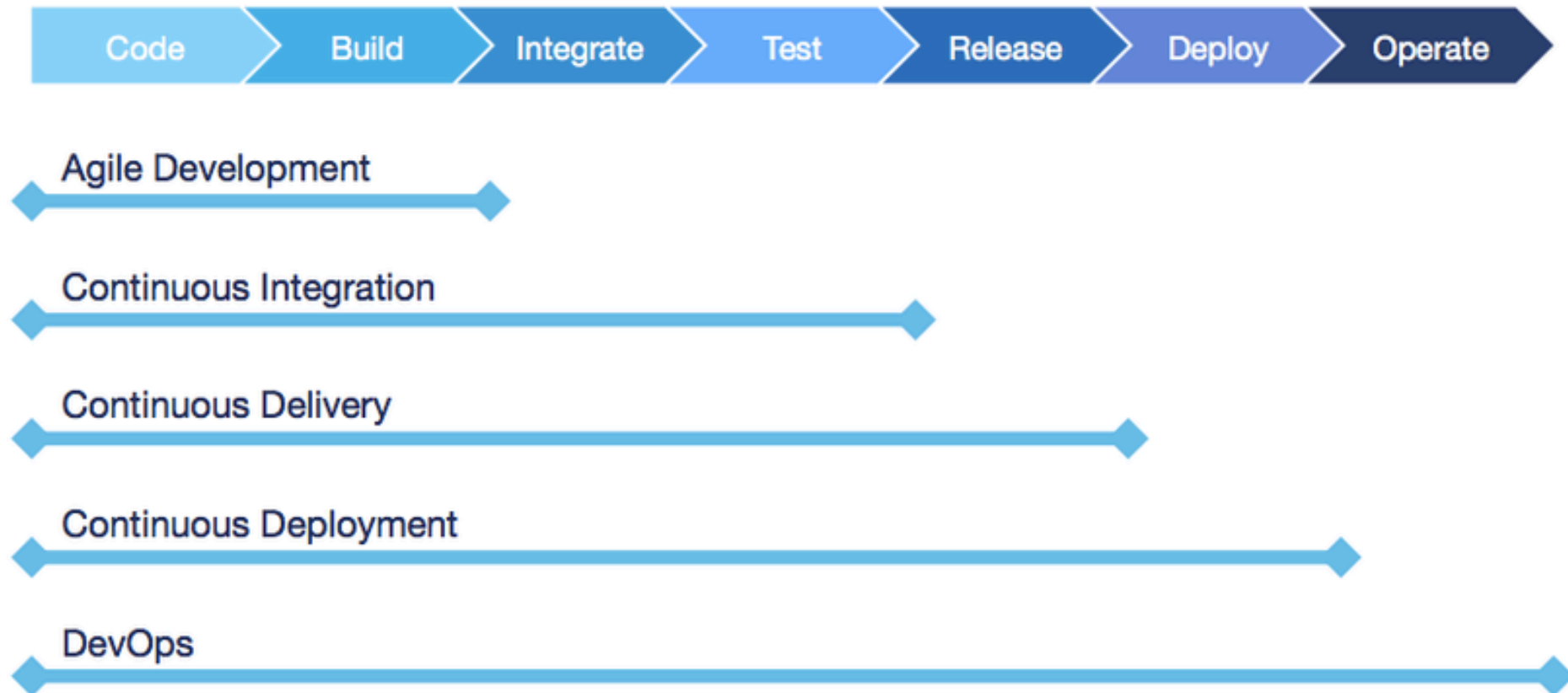
Tự động tích hợp công việc:

- Push code lên remote repo
- CI server phát hiện thay đổi
- Lấy code từ remote repo, build, test
- Gửi thông báo kết quả



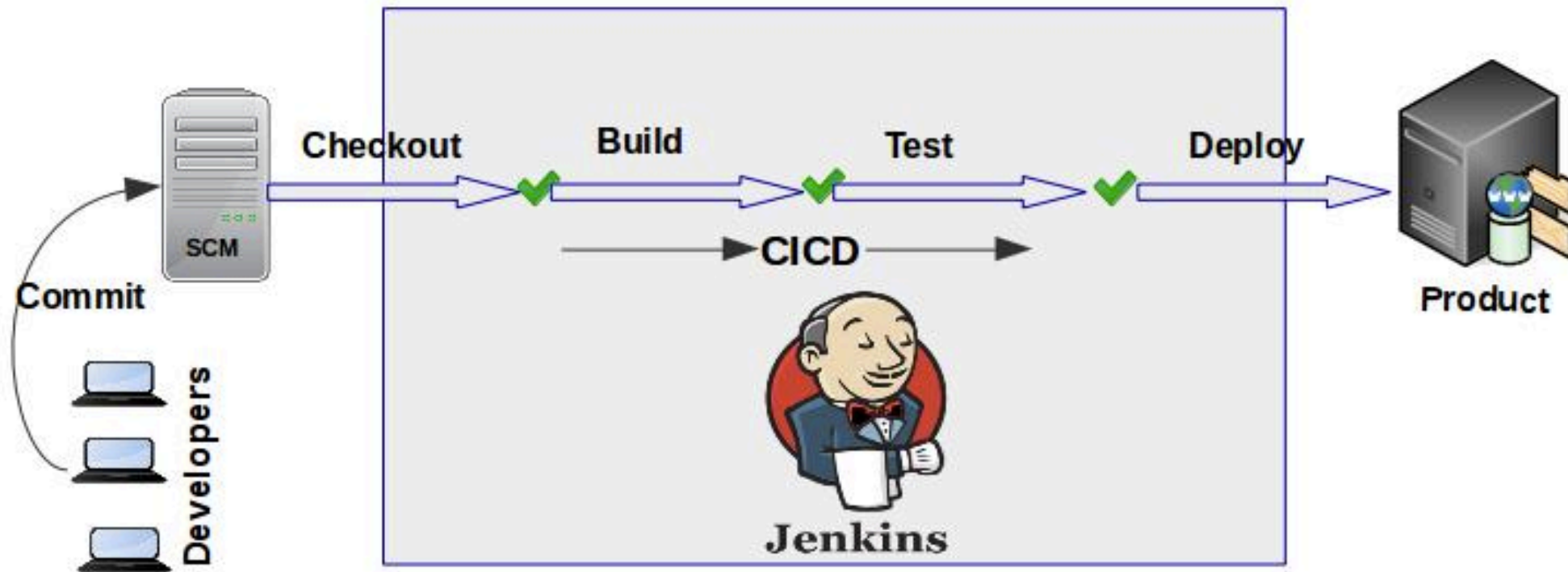
CD – Continuous Delivery

Tạo bản release sau quá trình chạy test, đảm bảo luôn có một *releasable product*



Jenkins

Phần mềm mã nguồn mở được viết bằng Java, cho phép thực hiện CI-CD tự động



Các công cụ CI/CD tự động khác

Continuous Integration & Delivery



Task #1: Cài đặt Jenkins



Cài đặt Jenkins với Docker



Dùng lệnh **docker run** để cài đặt Jenkins với các option sau:

1. Sử dụng Docker image: **jenkins/jenkins:its**
2. Mapping volume thư mục **/var/jenkins_home** của container chạy Jenkins ra một thư mục bên ngoài máy host
3. Expose 2 cổng **8080** và **50000** của container chạy Jenkins lần lượt ra 2 cổng **8080** và **50000** trên máy host

Lấy password đăng nhập Jenkins



1. Chạy lệnh: **docker logs** <ID-của-container-chạy-Jenkins>
2. Lấy password

Lấy password đăng nhập Jenkins

```
*****  
*****  
*****
```

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

```
4d6c1534f8fd48f09439d155dbfd8893
```

Copy đoạn mã này

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

```
*****  
*****  
*****
```

Đăng nhập vào Jenkins



1. Truy cập vào: **localhost:8080**
2. Paste đoạn mã vừa copy từ docker logs của Jenkins container vào ô nhập password

Đăng nhập vào Jenkins

localhost:8090/login?from=%2F

8090 là cổng được expose ra bên ngoài máy host của Jenkins container

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

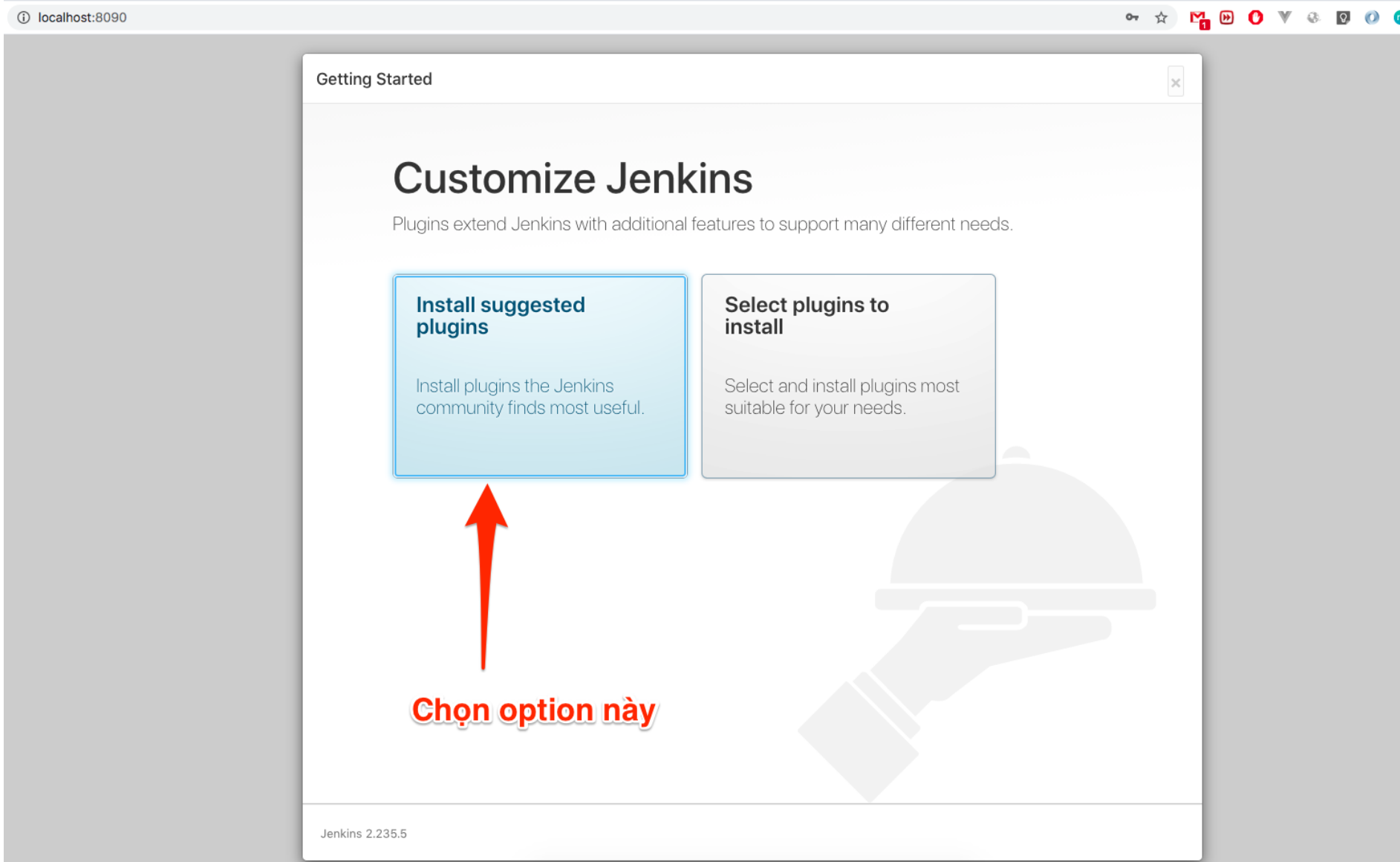
Please copy the password from either location and paste it below.

Administrator password

Paste đoạn mã vừa lấy ra từ docker logs của Jenkins container vào đây

Continue

Install suggested plugins



localhost:8090

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.

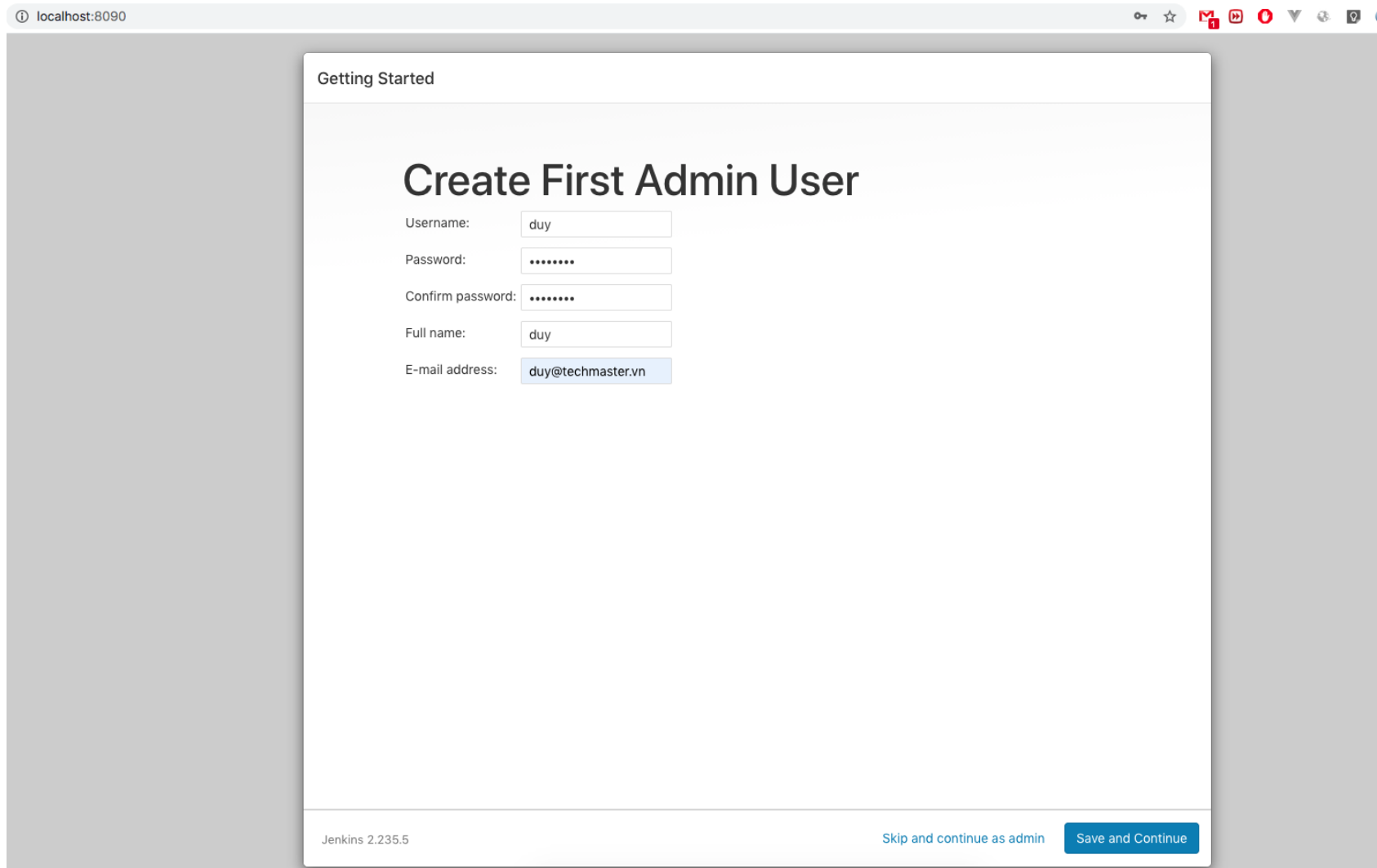
Select plugins to install
Select and install plugins most suitable for your needs.

Chọn option này

Jenkins 2.235.5

Tạo first admin user

User này sẽ được sử dụng để kết nối Git remote repo với Jenkins



The screenshot shows the Jenkins web interface in a browser window. The address bar indicates the URL is localhost:8090. The page title is 'Getting Started'. The main heading is 'Create First Admin User'. Below the heading are five input fields: 'Username' (filled with 'duy'), 'Password' (filled with seven dots), 'Confirm password' (filled with seven dots), 'Full name' (filled with 'duy'), and 'E-mail address' (filled with 'duy@techmaster.vn'). At the bottom of the form, there are two links: 'Skip and continue as admin' and 'Save and Continue'.

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.235.5

[Skip and continue as admin](#) [Save and Continue](#)

Task #2: Tạo public URL cho Jenkins localhost bằng ngrok

Tạo public URL bằng ngrok



- Hiện tại Jenkins đang chạy ở localhost
- Muốn kết nối Git remote repo (Github, Gitlab) với Jenkins, cần một địa chỉ public URL cho Jenkins
- Do đó, cần một công cụ để tạo public URL cho Jenkins localhost

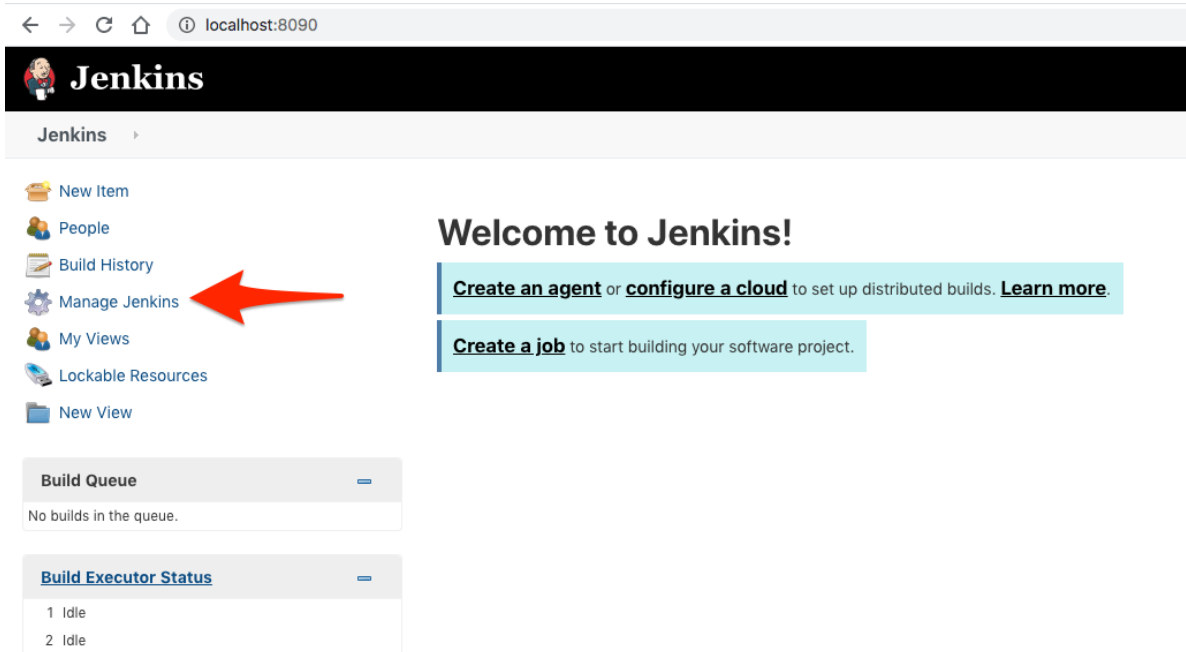
Cài đặt và sử dụng ngrok



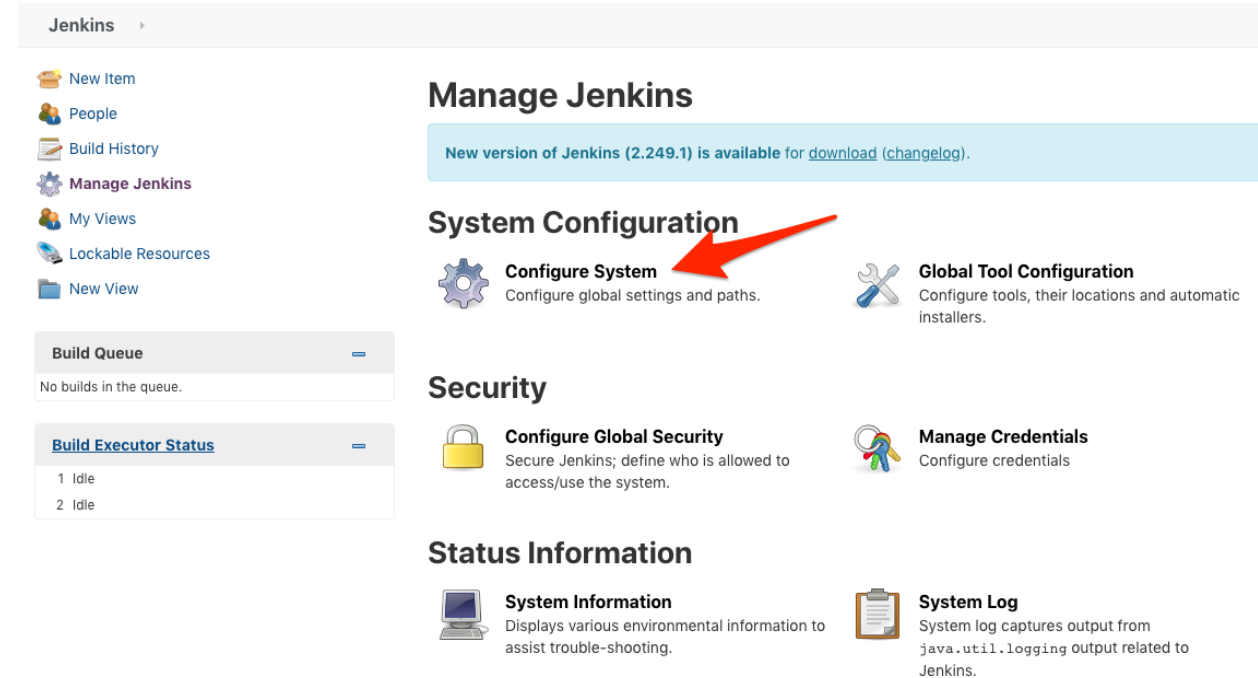
- Cài đặt ngrok: <https://ngrok.com/>
- Cấu hình ngrok: <https://xuanthulab.net/su-dung-ngrok-de-truy-cap-tu-internet-vao-may-ca-nhan.html>
- Tạo public URL cho Jenkins trên localhost:
ngrok http 8080
- Truy cập vào public URL mà ngrok vừa tạo

Sử dụng public URL cho Jenkins

Truy cập Manage Jenkins → Configure system



The screenshot shows the Jenkins homepage at localhost:8090. The left sidebar contains a menu with items: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. A red arrow points to the 'Manage Jenkins' item. The main content area displays 'Welcome to Jenkins!' with two light blue boxes: 'Create an agent or configure a cloud to set up distributed builds. Learn more.' and 'Create a job to start building your software project.' Below the welcome message are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two 'Idle' executors).



The screenshot shows the 'Manage Jenkins' page. The left sidebar lists: New Item, People, Build History, Manage Jenkins (highlighted), My Views, Lockable Resources, and New View. The main content area has a light blue banner at the top: 'New version of Jenkins (2.249.1) is available for download (changelog)'. Below the banner is the 'System Configuration' section, which includes 'Configure System' (with a red arrow pointing to it) and 'Global Tool Configuration'. The 'Security' section includes 'Configure Global Security' and 'Manage Credentials'. The 'Status Information' section includes 'System Information' and 'System Log'.

Nhập public URL mà ngrok tạo rồi bấm Apply và Save

configuration

Home directory /var/jenkins_home

System Message

[Plain text] [Preview](#)

of executors 2

Labels

Usage Use this node as much as possible

Quiet period 5

SCM checkout retry count 0

☐ Restrict project naming

Jenkins Location

Jenkins URL http://localhost:8090/ **Nhập public URL mà ngrok đã tạo**

⚠ Please set a valid host name, instead of localhost

System Admin e-mail address address not configured yet <nobody@nowhere>

Serve resource files from another domain

Resource root URL

Without a resource root URL, resources will be served from the main domain with Content-Security-Policy set.

Global properties

☐ Disable deferred wipeout on this node

☐ Environment variables

☐ Tool Locations

Pipeline Speed/Durability Settings

Save Apply

Kiến trúc Jenkins



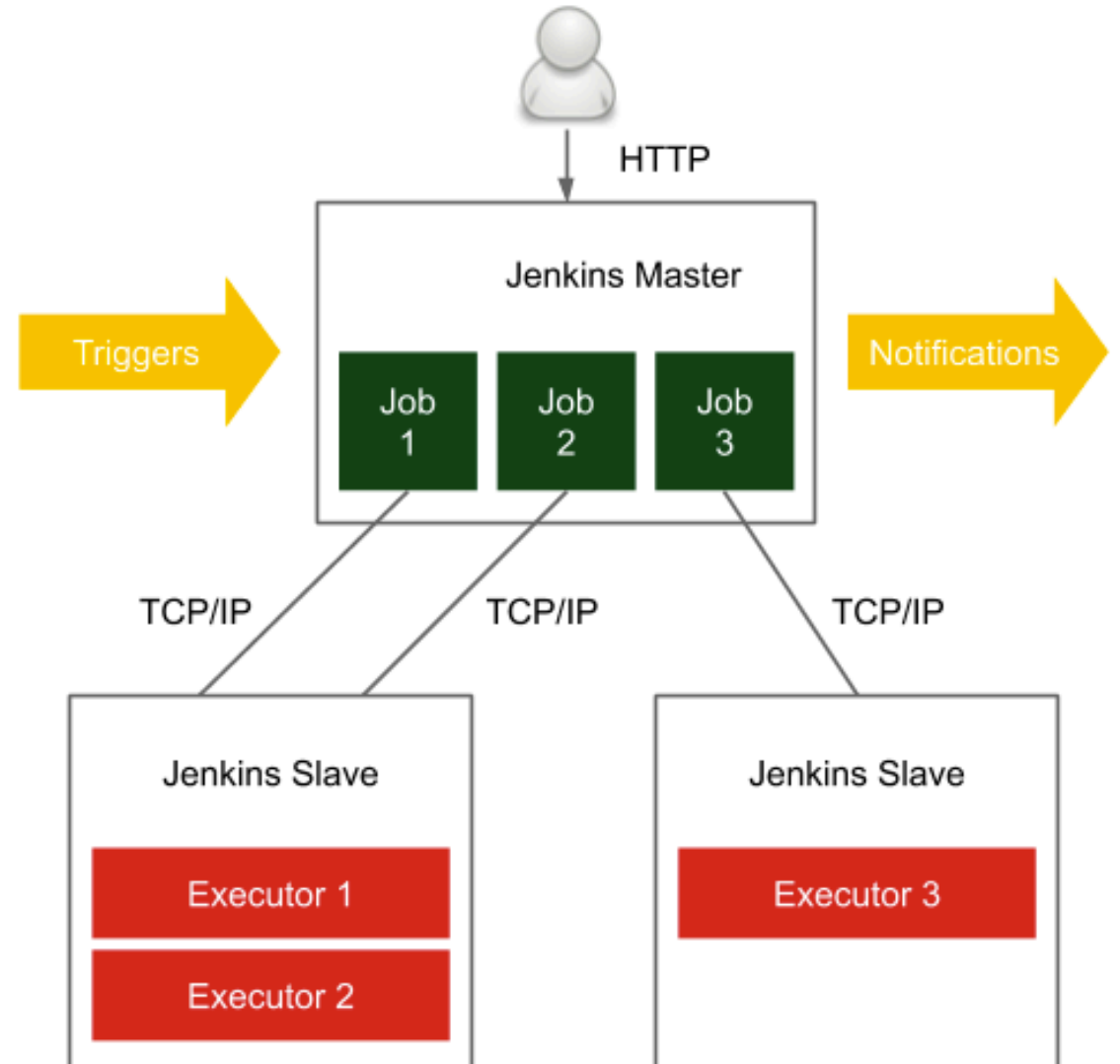
Master - slave

Master:

- Phát hiện các sự kiện kích hoạt chu trình CI (push code lên Github/Gitlab)
- Gửi yêu cầu chạy các task trong chu trình CI đến các node slave
- Nhận kết quả chạy chu trình CI được gửi đến từ các node slave
- Gửi thông báo kết quả chạy CI (qua email, Slack)

Slave:

- Chạy các task được gửi đến từ Jenkins master.



Task #3: Thêm 1 Jenkins slave



Chuẩn bị



- 1 máy cloud VM
- Cấu hình public – private SSH key. Tham khảo:
<https://xuanthulab.net/tao-ssh-key-va-xac-thuc-ket-noi-ssh-bang-public-private-key.html>
- Cài Java 8, Docker trên máy cloud VM

Thực hành



Tham khảo: <http://shorturl.at/kCQY1>

Jenkins pipeline - Jenkinsfile



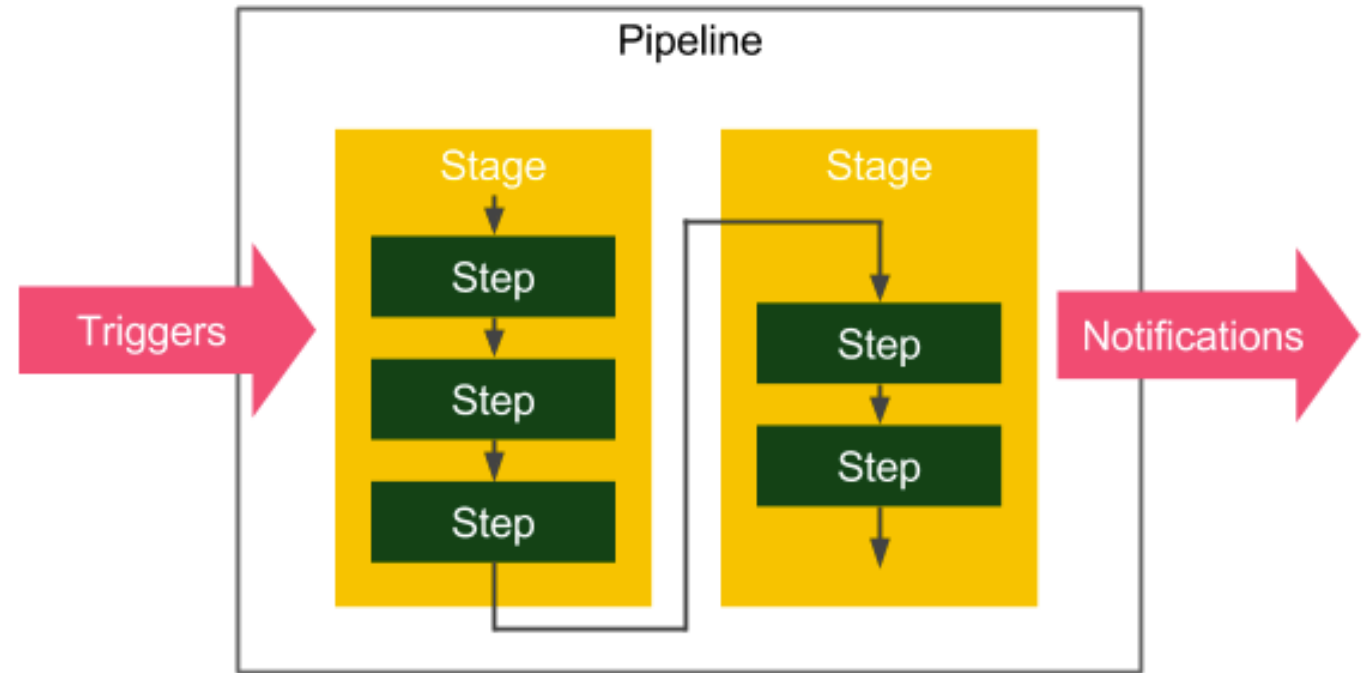
Jenkins pipeline

Step

- Tác vụ cụ thể mà Jenkins cần phải thực hiện

Stage

- Một nhóm các step được gom lại với nhau, đại diện cho một khâu trong chu trình CI/CD



Task #4: Hello pipeline



Yêu cầu



- Tạo 1 item mới, đặt tên là **hello-pipeline**, có kiểu là **Pipeline**
- Ở phần Pipeline, để definition là **Pipeline script**, nội dung script xem ở slide sau
- Bấm **Save** để tạo item
- Ở giao diện chi tiết item, bấm **Build Now** để kích hoạt pipeline

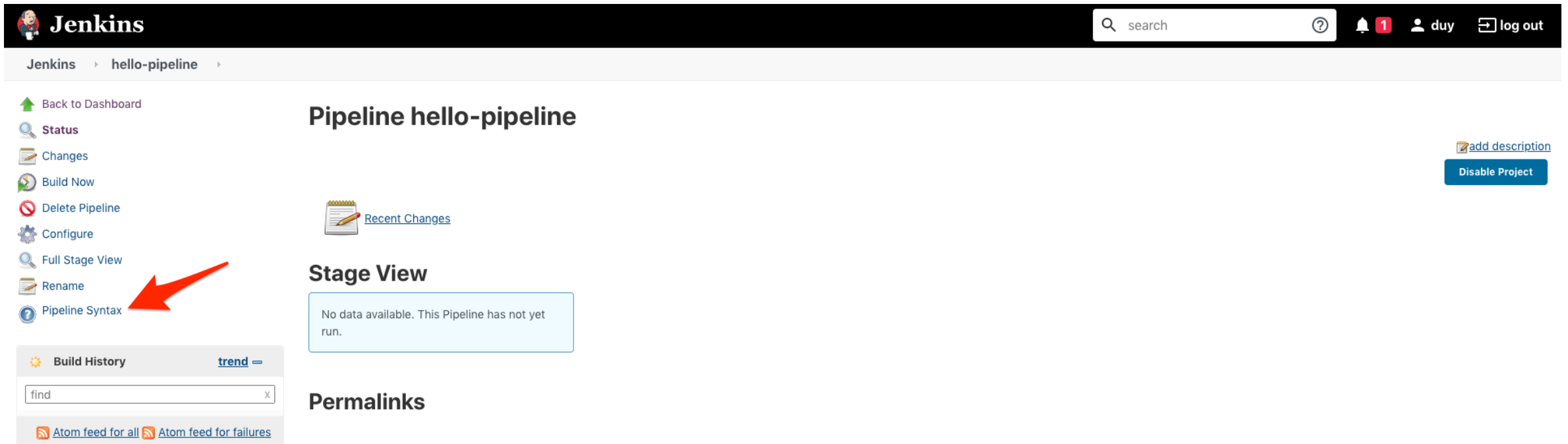
Nội dung Pipeline script



```
pipeline {  
    agent any  
    stages {  
        stage("Hello") {  
            steps {  
                echo 'Hello World'  
            }  
        }  
    }  
}
```

Viết step theo đúng Pipeline syntax

- Từ giao diện trang chủ Jenkins, chọn pipeline hello-pipeline vừa tạo
- Chọn **Pipeline Syntax**



The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, and user information. Below the header is a breadcrumb trail: Jenkins > hello-pipeline. On the left sidebar, a list of actions includes 'Back to Dashboard', 'Status', 'Changes', 'Build Now', 'Delete Pipeline', 'Configure', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. A red arrow points to 'Pipeline Syntax'. Below the sidebar is a 'Build History' section with a search bar and atom feed links. The main content area is titled 'Pipeline hello-pipeline' and contains a 'Recent Changes' section with a 'Recent Changes' link. Below that is a 'Stage View' section with a message: 'No data available. This Pipeline has not yet run.' At the bottom is a 'Permalinks' section. On the right side of the main content area, there are links for 'add description' and a 'Disable Project' button.

Jenkins

search

Jenkins > hello-pipeline

Back to Dashboard

Status

Changes

Build Now

Delete Pipeline

Configure

Full Stage View

Rename

Pipeline Syntax

Build History trend

find

Atom feed for all Atom feed for failures

Pipeline hello-pipeline

Recent Changes

Stage View

No data available. This Pipeline has not yet run.

Permalinks

add description

Disable Project

Viết step theo đúng Pipeline syntax

- Ở phần **Sample Step**, chọn step cần thực hiện (ví dụ, sh step là step chạy các lệnh dạng bash shell trên Linux)
- Nhập thao tác cần thực hiện, sau đó bấm **Generate Pipeline Script**

[Back](#)
[Snippet Generator](#)
[Declarative Directive Generator](#)
[Declarative Online Documentation](#)
[Steps Reference](#)
[Global Variables Reference](#)
[Online Documentation](#)
[Examples Reference](#)
[IntelliJ IDEA GDSL](#)

Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step sh: Shell Script

Shell Script pwd

Advanced...

Generate Pipeline Script

sh label: "", script: 'pwd'

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

1. Nhập lệnh cần chạy

2. Tạo pipeline script

Task #5: Bổ sung stage và step

Yêu cầu



- Bổ sung thêm 1 stage có tên là "**Create new file**". Trong stage gồm 2 step
- Step 1: Tạo một file có tên test.txt với lệnh **touch**
- Step 2: Kiểm tra nội dung thư mục với lệnh **ls**

Task #6: Bổ sung 2 stage lấy code từ Github và build Docker image



Yêu cầu



Bổ sung 2 stage

Stage 1:

- Gồm 1 step thực hiện việc lấy source code từ Github repo:
<https://github.com/handuy/nodejs-todolist>

Stage 2:

- Gồm 1 step thực hiện build Dockerfile thành docker image

Jenkinsfile

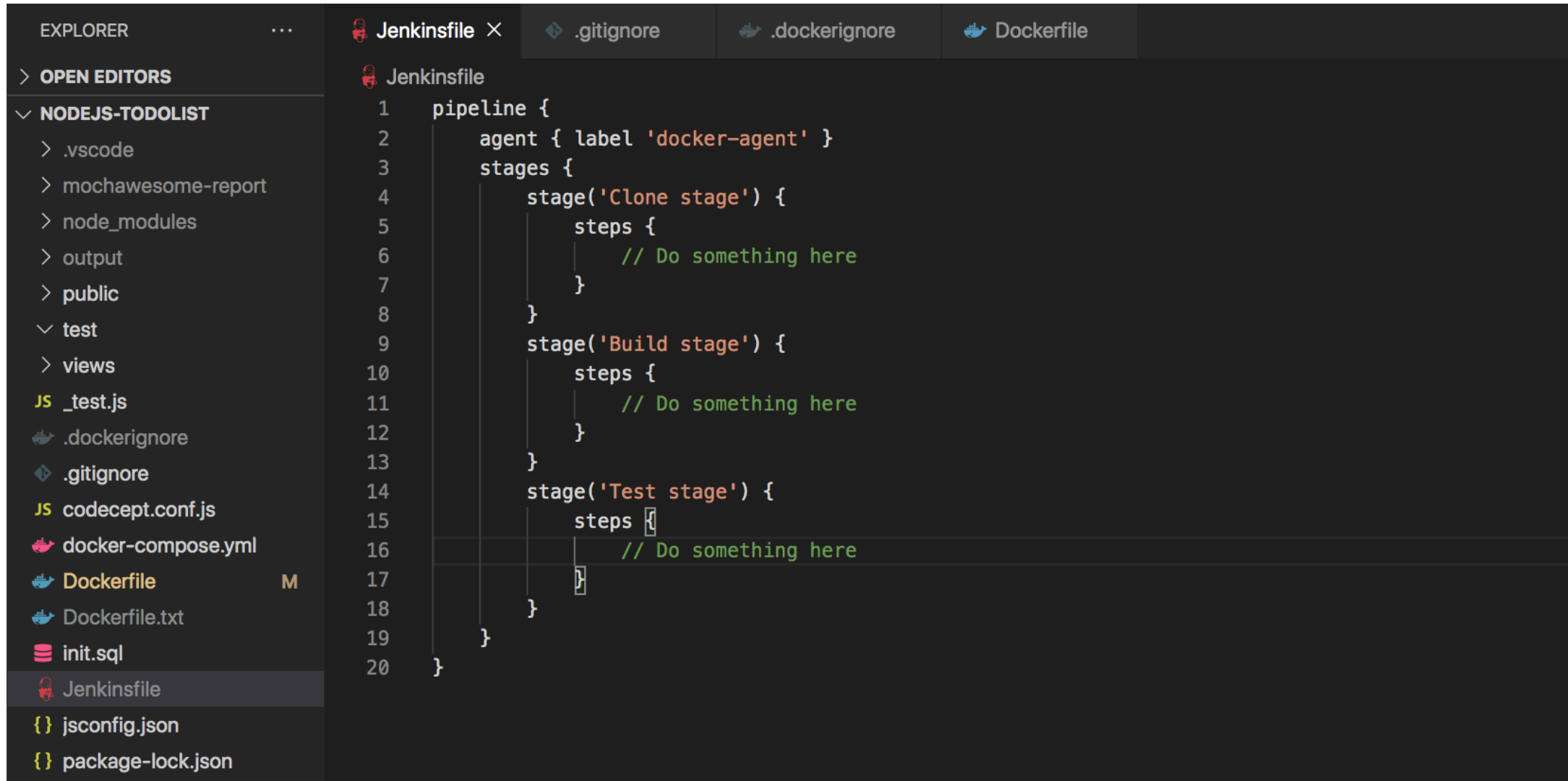


Jenkinsfile



- File text chứa cấu hình của toàn bộ quy trình Jenkins Pipeline dưới dạng code.
- Đặt trong source code được quản lý bởi Git

Jenkinsfile



The image shows a code editor interface with a sidebar on the left and a main editor area on the right. The sidebar, titled "EXPLORER", contains a list of files and folders. The main editor area displays the content of the "Jenkinsfile".

EXPLORER

- > OPEN EDITORS
- ▼ NODEJS-TODOLIST
 - > .vscode
 - > mochawesome-report
 - > node_modules
 - > output
 - > public
 - ▼ test
 - > views
- JS _test.js
- .dockerignore
- .gitignore
- JS codecept.conf.js
- docker-compose.yml
- Dockerfile **M**
- Dockerfile.txt
- init.sql
- Jenkinsfile**
- { } jsconfig.json
- { } package-lock.json

Jenkinsfile

```
1 pipeline {
2   agent { label 'docker-agent' }
3   stages {
4     stage('Clone stage') {
5       steps {
6         // Do something here
7       }
8     }
9     stage('Build stage') {
10      steps {
11        // Do something here
12      }
13    }
14    stage('Test stage') {
15      steps {
16        // Do something here
17      }
18    }
19  }
20 }
```

Cú pháp Jenkinsfile

<https://www.jenkins.io/doc/book/pipeline/>

Jenkinsfile (Declarative Pipeline)

```
pipeline {  
  agent any ❶  
  stages {  
    stage('Build') { ❷  
      steps {  
        // ❸  
      }  
    }  
    stage('Test') { ❹  
      steps {  
        // ❺  
      }  
    }  
    stage('Deploy') { ❻  
      steps {  
        // ❼  
      }  
    }  
  }  
}
```

- ❶ Execute this Pipeline or any of its stages, on any available agent.
- ❷ Defines the "Build" stage.
- ❸ Perform some steps related to the "Build" stage.
- ❹ Defines the "Test" stage.
- ❺ Perform some steps related to the "Test" stage.
- ❻ Defines the "Deploy" stage.
- ❼ Perform some steps related to the "Deploy" stage.

Task #7: Tích hợp Jenkins với Github repo



Tích hợp Jenkins với Github repo



Các bước tiến hành:

- Tạo Github repo chứa source code của ứng dụng NodeJS:
<https://github.com/handuy/nodejs-todolist>
- Tạo Github webhook. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/znfrsmtw/step/2pj9pl2k>
- Tạo file Jenkinsfile đặt vào trong source code. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/znfrsmtw/step/jf7u2tlj>
- Tạo Jenkins pipeline item kết nối với Github repo. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/znfrsmtw/step/iizb2l3v>

Task #8: Tích hợp Jenkins với Gitlab repo



Tích hợp Jenkins với Gitlab repo



Các bước tiến hành:

- Tạo Gitlab repo chứa source code ứng dụng NodeJS:
<https://github.com/handuy/nodejs-todolist>
- Cài Gitlab plugin cho Jenkins. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/gidzdz9r/step/mbcj11eu>
- Kết nối Gitlab repo đến Jenkins thông qua webhook. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/gidzdz9r/step/m1b6hc3p>
- Cấu hình cho Jenkins pipeline item. Tham khảo:
<https://techmaster.vn/user/learn/course/uod/lesson/gidzdz9r/step/k969vj0j>

Task #9: Build Docker image



Build Docker image



Yêu cầu:

- Thêm 1 stage trong Jenkinsfile. Stage này gồm 1 step: Build Docker image
- Build Docker Image:

<https://techmaster.vn/user/learn/course/uod/lesson/m2kc8j7x/step/beaonfs8>

Task #10: Push image lên Docker Hub



Yêu cầu



- Thêm 1 stage để push image vừa build lên Docker Hub
- Sử dụng Pipeline Syntax là **withDockerRegistry** để cấu hình registry URL và credential cho Docker Hub

Task #11: Push image lên Gitlab private registry



Yêu cầu



- Ở Gitlab repo: thêm 1 deploy token có quyền read và write đối với Container Registry
- Ở Jenkins: Thêm 1 credential với gitlab deploy token vừa tạo
- Trong file Jenkinsfile: Cập nhật 2 stage build và push docker image trong Jen, sử dụng lệnh build và push theo hướng dẫn của Gitlab Container Registry
- Push code lên Gitlab và kiểm tra