layout: post tags: [etc] ---

Jenkins ( )1

Chapter 1. Introduction

Continuous Integration( ) & Continuous Deliver( ) Jenkins .
, 2.46.2
.

1.1. Continuous Integration

> In software engineering, continuous integration (CI) is the practice of merging all developer working copies to a shared mainline several times a day. Grady Booch first named and proposed CI in his 1991 method, although he did not advocate integrating several times a day. Extreme programming (XP) adopted the concept of CI and did advocate integrating more than once per day - perhaps as many as tens of times per day.
>
> — Wikipedia

(CI)

.

,

. , CI ,

.

CI

-
-
-
-

1.2. Continuous Delivery

CI , , (CD)
. ,
. CD Continuous Deployment( ) ,
( ) ,
( ) .

CD

-
-
-
-

1.3. Jenkins

Jenkins Sun Microsystem Hudson , Sun Microsystem Oracle Oracle Jenkins . Jenkins Java CI CD . CI , Blue Ocean , CD .

Jenkins

-
- Reporting
-
-
-
-
-
- Groovy script Job Scheduling
-
-
-
-

Jenkins Jenkins . , Jenkins .

Jenkins .

, UI .

.

Build Pipeline

2 .

Chapter 2. Installing Jenkin

Jenkins .

2.1. Pre-install

2.1.1. System Requirements

:

- Java 7
- 256MB free memory
- 1GB+ free disk space

:

- Java 8
- 1GB+ free memory
- 50GB+ free disk space

2.2. Installation

2.2.1. Unix/Linux

Ubuntu Debian apt .
, LTS .

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```

/etc/default/jenkins ,
.

2.2.2. OS X

- jenkins .
- .

brew .

-

```
brew install jenkins
```

- LTS

```
brew install jenkins-lts
```

2.2.3. Windows

- jenkins .
- .

2.2.4. Docker

- Docker jenkins pull .

```
docker pull jenkins
```

- Docker .

```
docker run -d -p 9000:8080 -v $PWD/jenkins:/var/jenkins_home -t jenkins
```

- 9000 8080 .
- `-d` , .

2.2.5. Other

war `Tomcat` `Jetty` .

2.3. Post-installation(Setup Wizard)

2.3.1. Create Admin User and Password for Jenkins

Jenkins , .
Jenkins , .

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

Jenkins initial setup is required. A security token is required to proceed.
Please use the following security token to proceed to installation:

41d2b60b0e4cb5bf2025d33b21cb

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

[administrator password] | *jenkins/installing-jenkins/post-installation/administrator-password.png*

Figure 1.

2.3.2. Initial Plugin Installation

Plugin ,
.

[initial plugin installation] | *jenkins/installing-jenkins/post-installation/initial-plugin-installation.png*

Figure 2.

[install suggested plugins] | *jenkins/installing-jenkins/post-installation/install-suggested-plugins.png*

Figure 3.

Chapter 3. System Configuration

Jenkins .

Jenkins executor , , VCS , . .

Chapter 4. Managing Security

Jenkins .

Jenkins . Jenkins

.

4.1. Enabling Security

Enable Security .

.

[enable security] | *jenkins/managing-jenkins/managing-security/enable-security.png*

Figure 4.

4.1.1. JNLP TCP Port

Jenkins JNLP Agent TCP .

JNLP Agent .

Random

JNLP .

Fixed

JNLP .

4.1.2. Access Control

Access Control Jenkins . Access Control

.

1. **Security Realm** .
   " "
2. **Authorization** .
   " "

Security Realm Authorization .

, Role-based Authorization Strategy Access Control

.

Security Realm

Jenkins Security Realm .

Delegate to servlet container

Jenkins Servlet Container . , Servlet

Container .

Jenkins' own user database

Jenkins .
2.0 , .

LDAP

LDAP . LDAP .

LDAP , plugin:ldap[LDAP plugin]
.

Unix user/group database

Unix . Unix .

Security Realm

- plugin:active-directory[Active Directory]
- plugin:github-oauth[GitHub Authentication]
- plugin:crowd2[Atlassian Crowd 2]

Authorization

Security Realm Jenkins . Authorization
. Jenkins Authorization .

Anyone can do anything

Jenkins .
.

Legacy mode

"admin" ,
. .

Logged in users can do anything

.
. .

Matrix-based security

.

Project-based Matrix Authorization Strategy

Matrix-based security .

4.1.3. Markup Formatter

Jenkins HTML Javascript . < &
Plain Text . **Safe HTML**
HTML .

4.2. Cross Site Request Forgery

CSRF .

4.3. Agent/Master Access Control

, Jenkins master agent
. agent master
.

For larger or mature Jenkins environments where a Jenkins administrator might enable agents provided by other teams or organizations, a flat agent/master trust model is insufficient. The Agent/Master Access Control system was introduced [2: Starting with 1.587, and 1.580.1, releases] to allow Jenkins administrators to add more granular access control definitions between the Jenkins master and the connected agents. As of Jenkins 2.0, this subsystem has been turned on by default.

4.3.1. Customizing Access

For advanced users who may wish to allow certain access patterns from the agents to the Jenkins master, Jenkins allows administrators to create specific exemptions from the built-in access control rules. By following the link highlighted above, an administrator may edit Commands and File Access Agent/Master access control rules.

Commands

"Commands" in Jenkins and its plugins are identified by their fully-qualified class names. The majority of these commands are intended to be executed on agents by a request of a master, but some of them are intended to be executed on a master by a request of an agent. Plugins not yet updated for this subsystem may not classify which category each command falls into, such that when an agent requests that the master execute a command which is not explicitly allowed, Jenkins will err on the side of caution and refuse to execute the command. In such cases, Jenkins administrators may "whitelist" [3: en.wikipedia.org/wiki/Whitelist] certain commands as acceptable for execution on the master. 20

Advanced

Administrators may also whitelist classes by creating files with the .conf extension in the directory JENKINS_HOME/secrets/whitelisted-callables.d/. The contents of these .conf files should list command names on separate lines. The contents of all the .conf files in the directory will be read by Jenkins and combined to create a default.conf file in the directory which lists all known safe command. The default.conf file will be re-written each time Jenkins boots. Jenkins also manages a file named gui.conf, in the whitelisted-callables.d directory, where commands added via the web UI are written. In order to disable the ability of administrators to change whitelisted commands from the web UI, place an empty gui.conf file in the directory and change its permissions such that is not writeable by the operating system user Jenkins run as.

File Access Rules

The File Access Rules are used to validate file access requests made from agents to the master. Each File Access Rule is a triplet which must contain each of the following elements: 1. allow / deny: if the following two parameters match the current request being considered, an allow entry would allow the request to be carried out and a deny entry would deny the request to be rejected, regardless of what later rules might say. 2. operation: Type of the operation requested. The following 6 values

exist. The operations can also be combined by comma-separating the values. The value of all indicates all the listed operations are allowed or denied. ◦ read: read file content or list directory entries ◦ write: write file content ◦ mkdirs: create a new directory ◦ create: create a file in an existing directory ◦ delete: delete a file or directory

◦ stat: read metadata of a file/directory, such as timestamp, length, file access modes. 3. file path: regular expression that specifies file paths that matches this rule. In addition to the base regexp syntax, it supports the following tokens: ◦ <JENKINS_HOME> can be used as a prefix to match the master's JENKINS_HOME directory. ◦ <BUILDDIR> can be used as a prefix to match the build record directory, such as /var/lib/jenkins/job/foo/builds/2014-10-17_12-34-56. ◦ <BUILDID> matches the timestamp-formatted build IDs, like 2014-10-17_12-34-56. The rules are ordered, and applied in that order. The earliest match wins. For example, the following rules allow access to all files in JENKINS_HOME except the secrets folders: # To avoid hassle of escaping every '\' on Windows, you can use / even on Windows. deny all <JENKINS_HOME>/secrets/.* allow all <JENKINS_HOME>/.* Ordering is very important! The following rules are incorrectly written because the 2nd rule will never match, and allow all agents to access all files and folders under JENKINS_HOME: allow all <JENKINS_HOME>/.* deny all <JENKINS_HOME>/secrets/.*

Advanced

Administrators may also add File Access Rules by creating files with the .conf. extension in the directory JENKINS_HOME/secrets/filepath-filters.d/. Jenkins itself generates the 30-default.conf file on boot in this directory which contains defaults considered the best balance between compatibility and security by the Jenkins project. In order to disable these built-in defaults, replace 30-default.conf with an empty file which is not writable by the operating system user Jenkins run as. On each boot, Jenkins will read all .conf files in the filepath-filters.d directory in alphabetical order, therefore it is good practice to name files in a manner which indicates their load order. Jenkins also manages 50-gui.conf, in the filepath-filters/ directory, where File Access Rules added via the web UI are written. In order to disable the ability of administrators to change the File Access Rules from the web UI, place an empty 50-gui.conf file in the directory and change its permissions such that is not writeable by the operating system user Jenkins run as.

4.3.2. Disabling

While it is not recommended, if all agents in a Jenkins environment can be considered "trusted" to the same degree that the master is trusted, the Agent/Master Access Control feature may be disabled. Additionally, all the users in the Jenkins environment should have the same level of access to all configured projects. 22 An administrator can disable Agent/Master Access Control in the web UI by un-checking the box on the Configure Global Security page. Alternatively an administrator may create a file in JENKINS_HOME/secrets named slave-to-master-security-kill-switch with the contents of true and restart Jenkins. Most Jenkins environments grow over time requiring their trust models to CAUTION evolve as the environment grows. Please consider scheduling regular "check- ups" to review whether any disabled security settings should be re-enabled.

Chapter 5. Managing Users

Chapter 6. Managing Plugins

Chapter 7. Managing Nodes

Chapter 8. Freestyle Project

Chapter 9. Pipeline Project

Chapter 10. Multi-branch Pipeline Project

Chapter 11. Blue Ocean

Chapter 12. Pipeline Syntax

Chapter 13. Jenkins with Java

Chapter 14. Jenkins with PHP

Chapter 15. Jenkins with Javascript

Chapter 16. Jenkins with .NET

Chapter 17. Conclusions

Chapter 18. Terms

Chapter 19. References

- http://www.moreagile.net/2014/01/jenkins-cicontinuous-integration-2.html
- https://jenkins.io/doc/