

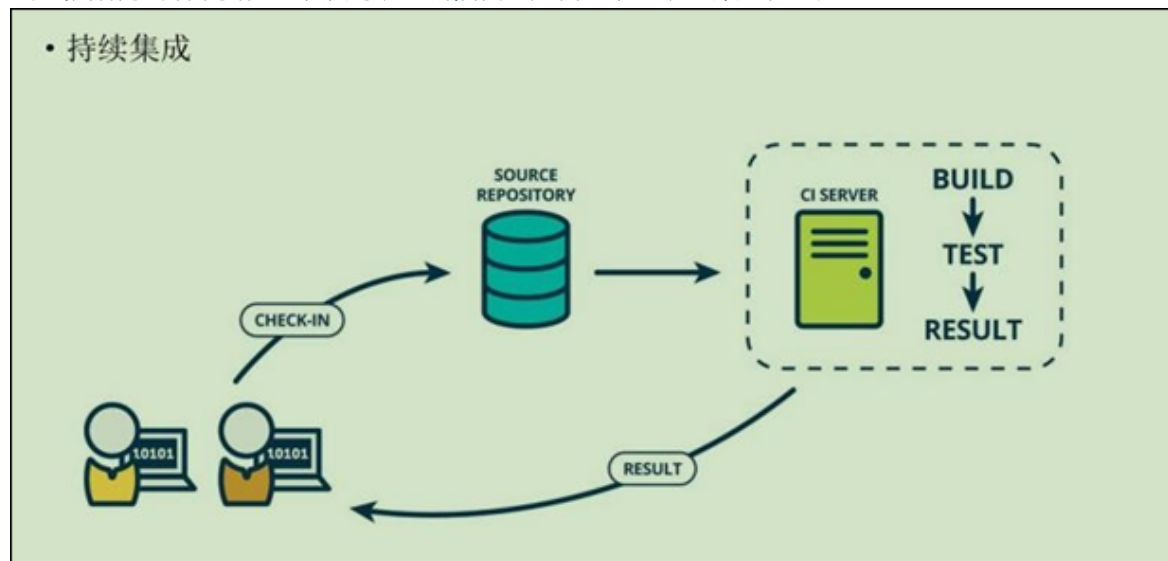
持续集成

持续集成、持续交付与持续部署

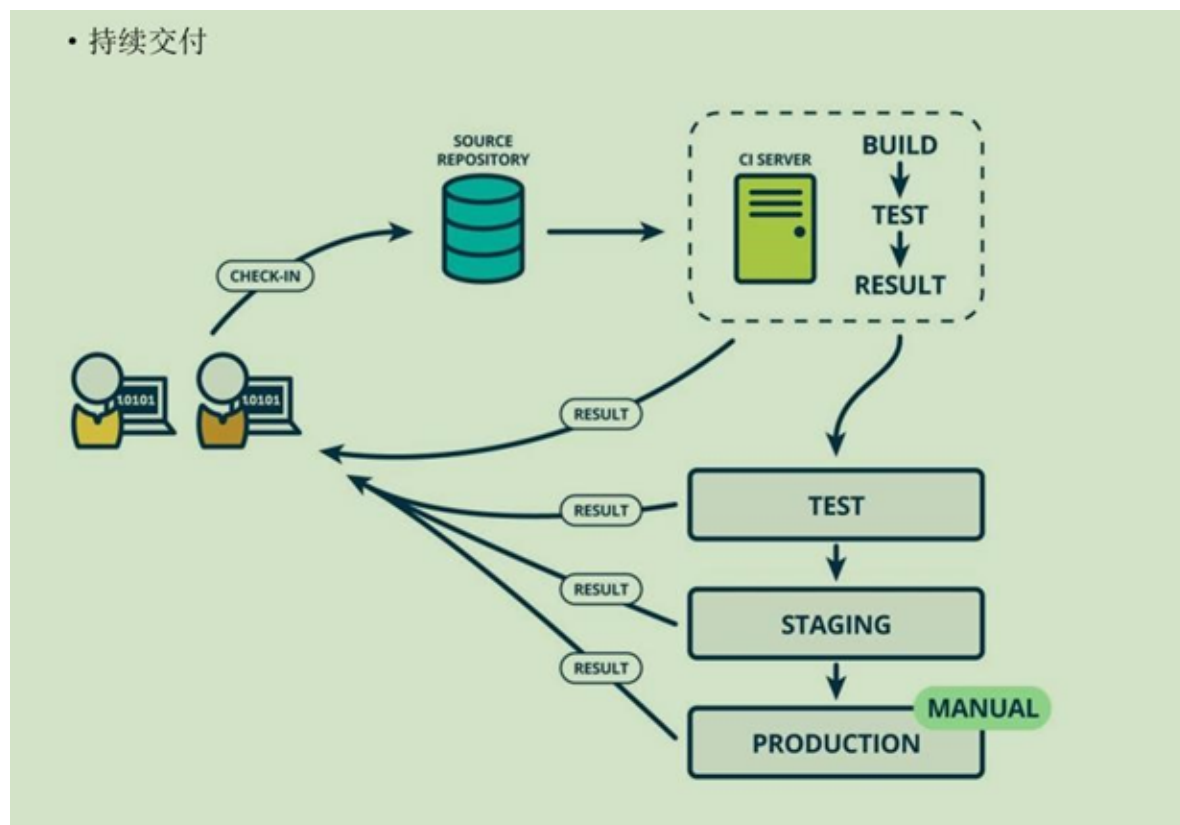
概念

#

持续集成 (Continuous Integration) 强调开发人员提交新代码后，系统立刻自动进行构建、测试。依据测试结果我们可以及时确定新旧代码是否可以正确的集成在一起。

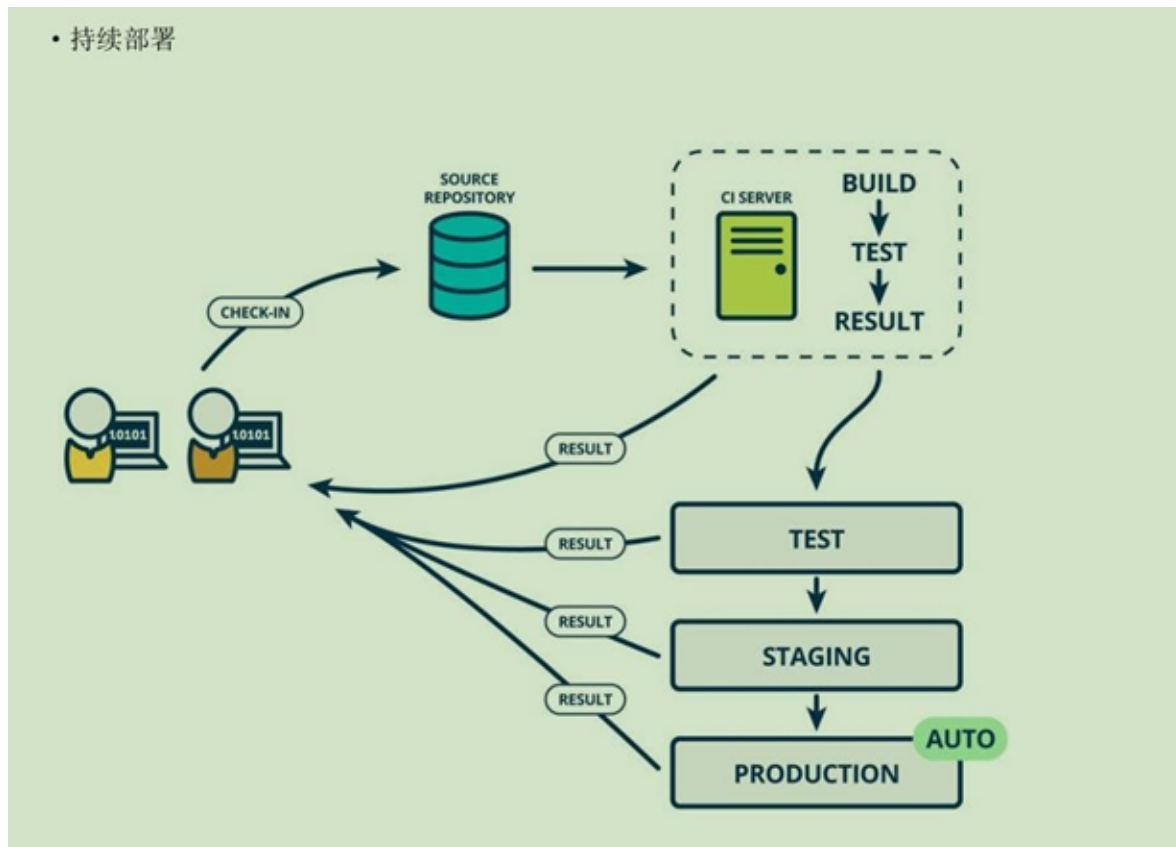


持续交付 (Continuous Delivery) 在持续集成的基础上，将集成后的代码部署到更贴近真实运行环境的“类生产环境”中。比如将测试完成的代码部署到STAGING环境中，再进行更多的测试。如果测试完成没有问题，再手动部署到生产环境中去。



持续部署（Continuous Deployment）是在持续交付的基础上，把部署到生产环境的过程也自动化。

• 持续部署



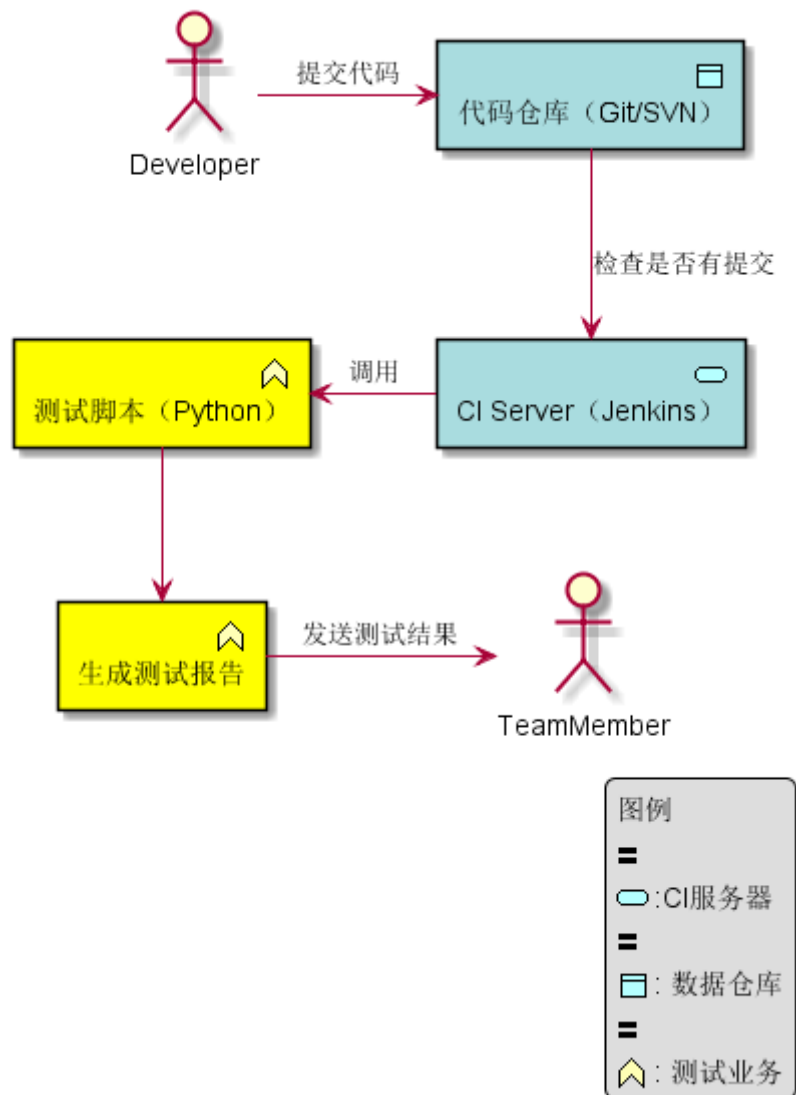
CI系统一般是按照以下几个 组成部分来发挥作用的

1. 开发者检入代码到源代码仓库
2. CI系统会为每个项目创建一个单独的工作区。当预设或请求一次新的构建时，它将把源代码仓库的源码存放到对应的工作区。
3. CI系统会在对应的工作区内执行构建过程。
4. （配置如果存在）构建完成后，CI系统会在一个新的构建中执行定义的一套测试。完成后触发通知给相关的当事人。
5. （配置如果存在）如果构建成功，这个构建会被打包并转移到一个部署目标（如应用服务器）或存储为软件仓库中的一个新版本。软件仓库可以是CI系统的一部分，也可以是一个外部的仓库，比如一个文件服务器或者象SourceForge、GitHub之类的网站。
6. CI系统通常会根据请求发起相应的操作，诸如及时构建、生成报告，或者检索一些构建好的构建。

Jenkins就是这样一个CI系统，它的前身是Hudson。

工作原理

#



安装

#

本地安装方法

1. 首先需要检查当前系统有无Java环境，如果没有，需要首先配置JDK或者JRE环境。
2. 打开 [Jenkins官网](#)，点击下载按钮然后选择对应的系统版本进行下载



Jenkins


Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Documentation](#)[Download](#)

Getting started with Jenkins

The Jenkins project produces two release lines, LTS and weekly. Depending on your organization's needs, one may be preferred over the other.

Both release lines are distributed as `.war` files, native packages, installers, and Docker containers. Packages with the  gear icon are maintained by third parties.

Long-term Support (LTS)



LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period. [Learn more...](#)

[Changelog](#) | [Upgrade Guide](#) | [Past Releases](#)

Deploy Jenkins 2.138.2



Download Jenkins 2.138.2 for:






Docker
FreeBSD
Gentoo 
Mac OS X
OpenBSD 
openSUSE
Red Hat/Fedora/CentOS
Ubuntu/Debian
Windows
Generic Java package (.war)

Weekly

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

[Changelog](#) | [Past Releases](#)

Download Jenkins 2.148 for:

Arch Linux 
Docker
FreeBSD 
Gentoo 
Mac OS X
OpenBSD 
openSUSE
Red Hat/Fedora/CentOS
Ubuntu/Debian
OpenIndiana Hipster 
Windows
Generic Java package (.war)

3. 下载执行下面的命令

```
1 curl -O https://prodjenkinsreleases.blob.core.windows.net/redhat-stable/jenkins-2.138.2-1.1.noarch.rpm
```

4. 下载完成后执行安装命令

```
1 rpm -ivh jenkins-2.138.1-1.1.noarch.rpm
```

5. 安装完执行启动jenkins命令

```
1 service jenkins start
```

6. 如果遇到下面的错误

```
1 Starting Jenkins bash: /usr/bin/java: No such file or directory
```

请执行下面的命令

```
1 vi /etc/init.d/jenkins
```

在该文件中找到下面的部分修改添加。

```
1 candidates="
2 /usr/local/jdk1.8.0_181/bin/java # 添加本地的java安装路径
3 /etc/alternatives/java
4 /usr/lib/jvm/java-1.8.0/bin/java
5 /usr/lib/jvm/jre-1.8.0/bin/java
6 /usr/lib/jvm/java-1.7.0/bin/java
7 /usr/lib/jvm/jre-1.7.0/bin/java
8 /usr/bin/java
9 "
```

保存这个文件后重新启动jenkins服务即可

- jenkins的默认访问端口是8080，这与tomcat等许多软件会有冲突，所以如果我们要同时使用这些软件的话，就必须修改端口。jenkins端口的修改需首先执行下面的命令打开配置文件。

```
1 vi /etc/sysconfig/jenkins
```

修改该文件中的如下端口

```
1 ## Type: integer(0:65535)
2 ## Default: 8080
3 ## ServiceRestart: jenkins
4 #
5 # Port Jenkins is listening on.
6 # Set to -1 to disable
7 #
8 JENKINS_PORT="8080" #修改默认端口为其他未使用的端口，比如8090。
```

- 修改完端口后切记需要在防火墙中开放该端口并重启防火墙

```
1 iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
2 service iptables save
3 service iptables restart
```

- 现在在浏览器中输入 <http://localhost:8080> 即可访问配置好的jenkins了。

Tomcat下配置war包的安装方法

- 首先需要检查当前系统有无Java环境，如果没有，需要首先配置JDK或者JRE环境。
- 然后下载配置Tomcat环境
- 打开 [Jenkins官网](#)，点击下载按钮然后选择war包进行下载



Jenkins


Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Documentation](#)[Download](#)

Getting started with Jenkins

The Jenkins project produces two release lines, LTS and weekly. Depending on your organization's needs, one may be preferred over the other.

Both release lines are distributed as `.war` files, native packages, installers, and Docker containers. Packages with the  gear icon are maintained by third parties.

Long-term Support (LTS)



LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period. [Learn more...](#)

[Changelog](#) | [Upgrade Guide](#) | [Past Releases](#)

Deploy Jenkins 2.138.2



Download Jenkins 2.138.2 for:






Docker
FreeBSD
Gentoo 
Mac OS X
OpenBSD 
openSUSE
Red Hat/Fedora/CentOS
Ubuntu/Debian
Windows
Generic Java package (.war)

Weekly

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

[Changelog](#) | [Past Releases](#)

Download Jenkins 2.148 for:

Arch Linux 
Docker
FreeBSD 
Gentoo 
Mac OS X
OpenBSD 
openSUSE
Red Hat/Fedora/CentOS
Ubuntu/Debian
OpenIndiana Hipster 
Windows
Generic Java package (.war)

3. 切换系统当前目录到Tomcat的webapps目录下
4. 下载执行下面的命令

```
1 curl -O https://mirrors.tuna.tsinghua.edu.cn/jenkins/war-stable/2.138.2/jenkins.war
```

5. 重启Tomcat服务
6. 现在在浏览器中输入 <http://localhost:8080/jenkins> 即可访问配置好的jenkins了。

初始化配置

#

1. 首次打开jenkins，系统将会提示输入密码，这个密码具体获取方法请看类似下面的网页说明，到相关路径下打开initialAdminPassword文件即可，该文件为普通的文本文件。

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:

```
/root/.jenkins/secrets/initialAdminPassword
```

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

Administrator password

Continue

```
1 cat /root/.jenkins/secrets/initialAdminPassword # linux操作系统查看密钥的方法
2
3 d75f6b837a774723844c05fa10ec1161
```

2. 将密码复制粘贴到密码栏后，点击继续按钮。

Getting Started

Customize Jenkins

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

Install suggested plugins

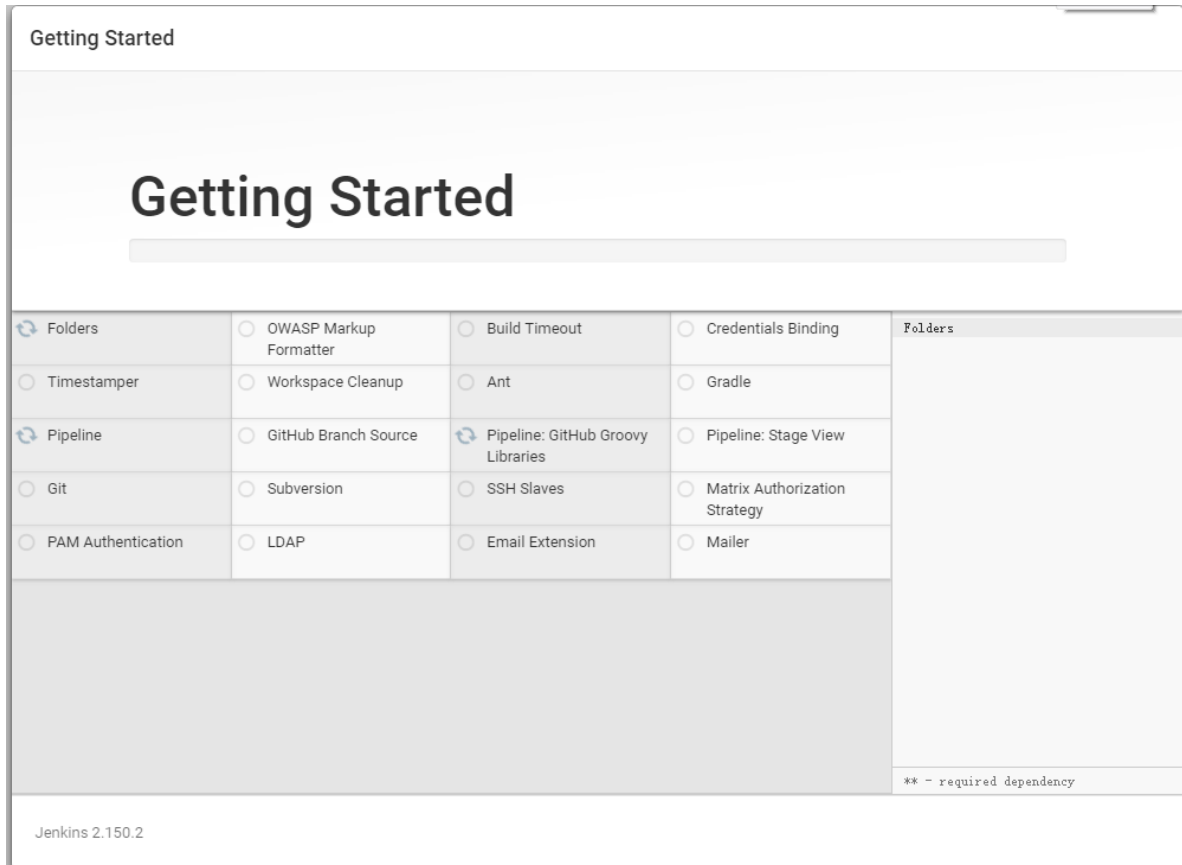
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

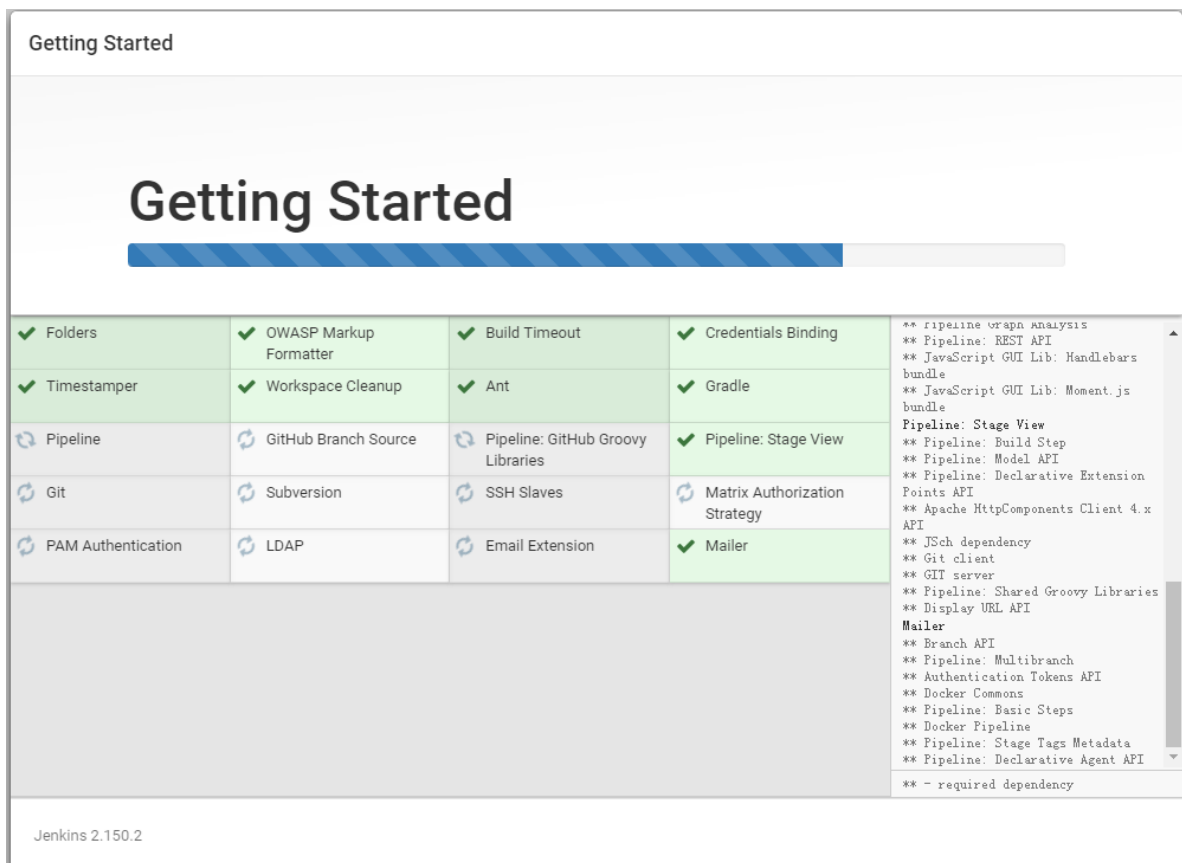
Jenkins 2.150.2

3. 会提示用户安装jenkins插件，我们选择安装推荐的插件即可。



若出现不了上面画面，需要按下面步骤来做

- 检查jenkins所在服务器是否可以访问外网，如果可以正常访问
- 刷新当前网页，会发现已经进入jenkins或者提示要登录，提示登录的话可以使用admin用户名，刚才initialAdminPassword文件显示的密码登录。登录后进入 系统设置 → 插件管理 → 高级，滚动到画面最下方，将其中的网址https修改为http



- 等待一段时间插件安装完毕会要求创建管理员用户，我们学习阶段可以使用root用户名和root密码，方便记忆。最后点击保存并完成按钮。

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.150.2

Continue as admin

Save and Continue

- 画面将显示jenkins服务的访问url，再次点击保存并完成。我们的初始化工作就完成了。

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.150.2

Not now

Save and Finish

使用方法

#

- 创建任务

- 任务类型
 - Freestyle Project

General 源码管理 构建触发器 构建环境 构建 构建后操作

描述

[纯文本] 预览

☐ GitHub 项目

☐ This build requires lockable resources

☐ Throttle builds

☐ 丢弃旧的构建

☐ 参数化构建过程

☐ 关闭构建

☐ 在必要的时候并发构建

高级...

源码管理

☒ 无

☐ Git

☐ Subversion

构建触发器

☐ 触发远程构建 (例如,使用脚本)

☐ 其他工程构建后触发

☐ 定时构建

☐ GitHub hook trigger for GITScm polling

☐ 轮询 SCM

构建环境

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ With Ant

构建

增加构建步骤

构建后操作

增加构建后操作步骤

- General 基本的Job信息
- Source Code Management 代码版本控制，其中Credentials可以用于添加用户名、口令或SSH Key
- Build Trigger 构建触发器，指定触发一次构建的条件。推荐使用（轮询PCM）Poll SCM，它的作用是定时检查版本仓库，如果发现有更新就触发，否则则不会动作

日程表语法规则：

这里利用了unix的任务调度工具cron的规则。采用5段式规则代表5个不同的时间单位，中间可以用空格或者制表符隔开。

1 MINUTE HOUR DOM MONTH DOW 分 时 日 月 星期

字段	*	*	*	*	*
含义	分钟	小时	日期	月份	星期
取值范围	0-59	0-23	1-31	1-12	0-7
示例					
每隔15分钟执行一次	H/15	*	*	*	*
每隔2个小时执行一次	H	H/2	*	*	*
每隔3天执行一次	H	H	H/3	*	*
每隔3天执行一次(每月的1-15号)	H	H	1-15/3	*	*
每周1,3,5执行一次	H	H	*	*	1,3,5
规则					
匹配范围内所有值	*				
匹配M~N范围内的所有值	M-N				
匹配范围内能被x整除的值	M-N/X或*/X				
匹配多个值	A,B,...,Z				
表示随机 (hash)	H				
表示M~N范围内随机	H(M-N)				

- Build Environment 构建环境，构建的环境配置
 - Build 构建
 - Post-build Actions 构建后操作，构建后需要执行的动作，一般是发送邮件等等
- Pipeline
 - 所谓Pipeline，简单来说，就是一套运行于Jenkins上的工作流框架，将原本独立运行于单个或者多个节点的任务连接起来，实现单个任务难以完成的复杂发布流程
 - Pipeline的实现方式是一套Groovy DSL(类似Gradle)，任何发布流程都可以表述为一段Groovy脚本，并且Jenkins支持从代码库直接读取脚本，从而实现了Pipeline as Code的理念
 - Pipelined 的产生背景：

第一层面，与不断增长的发布复杂度有关，其中一个典型场景就是灰度发布。原本只有大公司才有的灰度发布，随着敏捷开发实践的广泛采用、产品迭代周期的不断缩短、数据增长理念的深入人心，越来越多的中小公司也开始这一方面的探索，这对发布的需求也从点状的 CI 升级到线状的 CD。

第二层面，与应用架构的模块化演变有关，以微服务为代表，一次应用升级往往涉及到多个模块的协同发布，单个 CI 显然无法满足此类需求。

第三层面，与日益失控的CI数量有关。一方面，类似于Maven、pip、RubyGems这样的包管理工具使得有CI需求的应用呈爆发性增长，另一方面，受益于便捷的Git分支特性，即便对于同一个应用，往往也需要配置多个CI。随着CI数量的不断增长，集中式的任务配置就会成为一个瓶颈，这就需把任务配置的职责下放到应用团队。

- 基本概念

- Stage

- 一个Pipeline可以划分为多个Stage，每个Stage代表一个组操作。Stage是一个逻辑分组的概念，可以跨多个Node。

- Node

- 一个Node就是一个Jenkins节点，可以是master，也可以是agent。它是执行Step的具体运行环境

- Step

- Step是一个最基本的操作单元，小到创建一个目录，大到构建一个Docker镜像，由Jenkins的各类插件提供。

- 简单来讲，Pipeline(管道)由多个步骤组成，允许您构建、测试和部署应用程序。

- Pipeline Script

```
1  node {
2      // Mark the code checkout 'stage'....
3      stage 'Checkout'
4
5      // Get some code from a GitLib repository
6      git([url: 'http://jacky-vpc64:8000/root/jenkins-demo.git',
7          branch: 'master'])
8
9      // Mark the code run 'stage'....
10     stage 'Run'
11     // Run the program
12     sh "python requests_unittest_agileone.py"
```

2. 控制Jenkins

- 重启

```
1 http://localhost:8080/jenkins/restart
```

- 重新加载配置信息

```
1 http://localhost:8080/jenkins/reload
```

- 关闭

```
1 http://localhost:8080/jenkins/exit
```

构建自动化测试并发送邮件

#

邮件通知在Jenkins中有两种方式：

1. 默认的邮件通知方式

本身有很多局限性，比如它的邮件通知无法提供详细的邮件内容、无法定义发送邮件的格式、无法定义灵活的邮件接收配置等等。

2. Email Extension Plugin 插件的通知方式

不论哪种通知方式都拥有 **全局配置** 和 **项目配置**，全局配置在系统设置中进行，而项目配置在项目中的设置中进行。

- **全局配置**：
 - Jenkins URL:
 - System Admin e-mail address: **name@woniuxy.com**
 - 默认邮件通知配置演示
 - SMTP Server: smtp.woniuxy.com
 - Advance
 - SMTP Server: smtp.woniuxy.com
 - Use SMTP Authentication
 - User Name: **name@woniuxy.com**
 - Password: 123456
 - SMTP port: 25
 - Test configuration by sending Test e-mail recipient
 - Test e-mail recipient: **name@woniuxy.com**
 - 为了在邮件中收到job运行的log以及以附件发送的测试报告等信息，我们还需要再配置扩展邮件插件
 - SMTP Server: smtp.woniuxy.com
 - Default Content Type: HTML(text/html)
 - Default Recipients: **name@woniuxy.com** 如果有多个收件地址可以用 **,** 分隔
 - Advance
 - SMTP Server: smtp.woniuxy.com
 - Use SMTP Authentication
 - User Name: **name@woniuxy.com**
 - Password: 123456
 - SMTP port: 25
 - Default Content

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>${ENV, var="JOB_NAME"}-第${BUILD_NUMBER}次构建日志</title>
6 </head>
7
8 <body leftmargin="8" marginwidth="0" topmargin="8" marginheight="4"
9 offset="0">
10 <table width="95%" cellpadding="0" cellspacing="0"
11 style="font-size: 11pt; font-family: Tahoma, Arial, Helvetica,
12 sans-serif">
13 <tr>
14 <td>
15 <h2>
16 <font>来自Mr.Jenkins的邮件通知</font>
```

[illegible]

- 在系统全局工具配置git工具
- Path to Git executable: 配置当前系统git的路径
- Freestyle Project
 - General
 - 丢弃旧的构建
 - 在必要的时候并发构建
 - 源码管理
 - 配置git
 - 添加远程仓库的URL
 - 构建触发器
 - 轮询SCM
 - 构建环境

- Delete workspace before build starts
 - Add timestamps to the Console Output
- 构建
 - 添加Windows批处理命令或执行shell
- 构建后操作
 - Editable Email Notification

General

源码管理构建触发器构建环境构建构建后操作

描述

针对Agileone的接口自动化测试

[纯文本] 预览

☐ GitHub 项目

☐ This build requires lockable resources

☐ Throttle builds

☒ 丢弃旧的构建

策略

Log Rotation

保持构建的天数5

如果非空，构建记录将保存此天数

保持构建的最大个数50

如果非空，最多此数目的构建记录将被保存

高级...

☐ 参数化构建过程

☐ 关闭构建

☒ 在必要的时候并发构建

高级...

源码管理

无

☒ Git

Repositories

Repository URLhttp://jacky-vpc64:8000/root/jenkins-demo.git

Credentialsroot/*****Ad*

高级...

Add Repository

Branches to build

Branch Specifier (blank for 'any')*/master

Add Branch

源码库浏览器

(自动)

Additional Behaviours

新增

☐ Subversion

构建触发器

☐ 触发远程构建 (例如,使用脚本)

☐ 其他工程构建后触发

☐ 定时构建

☐ GitHub hook trigger for GITScm polling

☒ 轮询 SCM

日程表

H/5 * * * *

上次运行的时间 Saturday, October 27, 2018 8:02:12 PM CST; 下次运行的时间 Saturday, October 27, 2018 8:07:12 PM CST.

忽略钩子 post-commit

构建环境

☒ Delete workspace before build starts

高级...

☐ Use secret text(s) or file(s)

- ☐ Abort the build if it's stuck
- ☒ Add timestamps to the Console Output
- ☐ With Ant



构建

执行 shell



命令 `python requests_unittest_agileone.py`

[查看 可用的环境变量列表](#)

高级...

增加构建步骤

构建后操作

Editable Email Notification



Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Project From

Project Recipient List

\$DEFAULT_RECIPIENTS



Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List

\$DEFAULT_REPLYTO



Comma-separated list of email address that should be in the Reply-To header for this project.

Content Type

Default Content Type



Default Subject

\$DEFAULT_SUBJECT



Default Content

\$DEFAULT_CONTENT



Attachments

report/agileone_test_report_*.html



Can use wildcards like 'module/dist/**/*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log

Do Not Attach Build Log



Content Token Reference



Pre-send Script

\$DEFAULT_PRESEND_SCRIPT



Post-send Script

\$DEFAULT_POSTSEND_SCRIPT



Additional groovy classpath

新增



Save to Workspace

☐



Triggers

Always



Send To

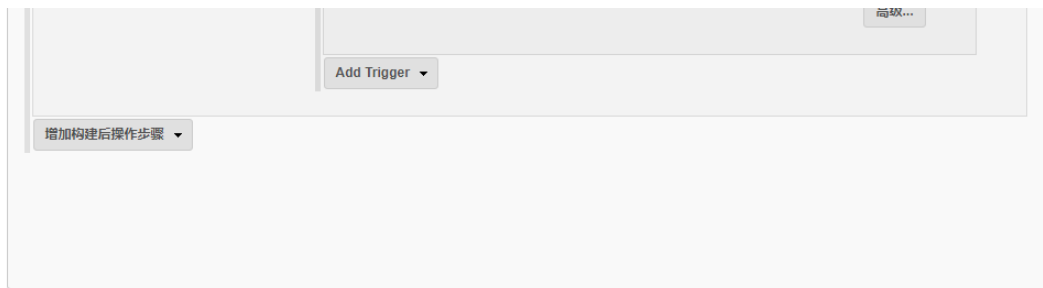
Developers



Recipient List



新增



###Jenkins+Jmeter+Ant 自动化集成环境搭建

- 配置Jmeter和Ant环境

1. Ant官网[下载](#)最新版

```
1 curl -O http://mirrors.shu.edu.cn/apache//ant/binaries/apache-ant-1.10.5-bin.tar.gz
```

2. 解压

```
1 tar -zxvf apache-ant-1.10.5-bin.tar.gz -C /usr/local/
```

3. 配置

```
1 vi /etc/profile
```

增加如下内容，然后保存

```
1 ANT_HOME=/usr/local/apache-ant-1.10.5
2 PATH=$PATH:$ANT_HOME/bin
3 export ANT_HOME PATH
```

```
1 source /etc/profile
```

4. 检查

```
1 ant -version
```

- Jenkins系统设置的全局工具配置
 - 配置JDK
 - 配置Ant
- 安装HTML Publisher插件
- 配置任务

General

[源码管理](#)[构建触发器](#)[构建环境](#)[构建](#)[构建后操作](#)

描述

[纯文本] [预览](#)

☐ GitHub 项目

☐ This build requires lockable resources

☐ Throttle builds

☒ 丢弃旧的构建



策略

Log Rotation

保持构建的天数

5

如果非空，构建记录将保存此天数

保持构建的最大个数

20

如果非空，最多此数目的构建记录将被保存

高级...

☐ 参数化构建过程

☐ 关闭构建

☒ 在必要的时候开发构建



高级...

源码管理

☐ 无

☒ Git

Repositories

Repository URL

Credentials

root/*****



高级...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

Add Branch

源码库浏览器

(自动)

Additional Behaviours

新增

☐ Subversion

构建触发器

☐ 触发远程构建 (例如,使用脚本)

☐ 其他工程构建后触发

☐ 定时构建

☐ GitHub hook trigger for GITScm polling

☒ 轮询 SCM

日程表

H/5 ****

上次运行的时间 Saturday, October 27, 2018 11:44:42 PM CST; 下次运行的时间 Saturday, October 27, 2018 11:49:42 PM CST.

构建环境

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☒ Add timestamps to the Console Output
- ☒ With Ant

Ant Version

ant

JDK

jdk

构建

执行 shell

命令 `ant -lib /usr/local/apache-jmeter-4.0/extras/ant-jmeter-1.1.1.jar -f build.xml`

查看 [可用的环境变量列表](#)

高级...

增加构建步骤

构建后操作

Publish HTML reports

Reports

HTML directory to archive

results/html

Index page[s]

agileone.html

Index page title[s] (Optional)

Report title

HTML Report

Publishing options...

新增

增加构建后操作步骤