

# Jenkins介绍

- ① Jenkins简介
- ② Jenkins安装
- ③ Jenkins主要功能
- ④ Jenkins插件
- ⑤ Jenkins集成
- ⑥ Jenkins API

# Jenkins是什么

Jenkins是一个独立的开源自动化服务器，它可以用于自动化与构建、测试和交付或部署软件相关的所有任务，实现持续集成、持续交付项目。

Jenkins可以处理任何类型的持续构建、持续集成。集成Jenkins可以用于测试与部署技术。

目标：

- 1、持续、自动地构建、测试软件项目
- 2、监控软件开发流程、快速问题定位、处理，提示开发效率

# Jenkins特性

## 主要特性

- 开源的java语言开发持续集成工具，支持CI，CD。
- 易于安装部署配置：可通过yum安装,或下载war包以及通过docker容器等快速实现安装部署，可方便web界面配置管理。
- 消息通知及测试报告：集成RSS/E-mail通过RSS发布构建结果或当构建完成时通过e-mail通知，生成JUnit/TestNG测试报告。
- 分布式构建：支持Jenkins能够让多台计算机一起构建/测试。
- 文件识别:Jenkins能够跟踪哪次构建生成哪些jar，哪次构建使用哪个版本的jar等。
- 丰富的插件支持:支持扩展插件，你可以开发适合自己团队使用的工具，如git，svn，maven，docker等。

# Jenkins的版本

- **LTS ( Long-term Support ) 12weeks**
- **Weekly**
  
- Download
- <https://jenkins.io/download/>

# Jenkins的产品发布流程

1. 产品设计成型
2. 开发人员开发编码
3. 测试人员测试功能
4. 运维人员发布上线

# Jenkins-CI过程

1. 开发者检入代码到源代码仓库
2. CI系统会为每一个项目创建了一个单独的工作区。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件中执行定义的一套测试。完成后触发通知(Email,RSS等等)给相关的当事人
5. （配置如果存在）如果构建成功，这个构件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，诸如一个文件服务器或者像Java.net、 SourceForge之类的网站
6. CI系统通常会根据请求发起相应的操作，诸如即时构建、生成报告，或者检索一些构建好的构件。

- ① Jenkins简介
- ② **Jenkins安装**
- ③ Jenkins主要功能
- ④ Jenkins插件
- ⑤ Jenkins集成
- ⑥ jenkins API



# Jenkins安装

## 三种安装方式

- 本地安装
- 容器 ( docker )
- 单机模式运行于tomcat

## • 配置要求：

- RAM 256+ 推荐512MB
- 硬盘 10GB+
- 软件要求：
- JDK1.8

## • Docker 安装

- 安装Docker环境

# 本地安装

下载地址

Jenkins :

<https://jenkins.io/download/>

安装文档

```
Run java -jar jenkins.war --httpPort=8080
```

访问 :

<http://localhost:8080>

## 下载镜像

# To use the latest LTS: docker pull jenkins/jenkins:its

#To use the latest weekly: docker pull jenkins/Jenkins

## 启动

不挂在数据卷：

```
docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:its
```

挂在数据卷：

```
docker run -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home  
jenkins/jenkins:its
```

- ① Jenkins简介
- ② Jenkins安装
- ③ **Jenkins主要功能**
- ④ Jenkins插件
- ⑤ Jenkins集成
- ⑥ Jenkins API

# Jenkins主要功能

The screenshot displays the Jenkins web interface in a browser window. The address bar shows 'localhost:8080'. The page header includes the Jenkins logo, a search bar, and the user 'admin'. The main content area features a navigation sidebar on the left with links for '新建任务', '用户', '构建历史', '系统管理', '我的视图', and '新建视图'. The central area contains a '欢迎使用Jenkins!' message and a prominent button labeled '开始创建一个新任务'. Below this, there are two panels: '构建队列' (Build Queue) which is currently empty, and '构建执行状态' (Build Execution Status) which shows two idle build slots.

localhost:8080

Jenkins

1 查找 admin

Jenkins

- 新建任务
- 用户
- 构建历史
- 系统管理
- 我的视图
- 新建视图

构建队列

队列中没有构建任务

构建执行状态

- 1 空闲
- 2 空闲

欢迎使用Jenkins!

开始创建一个新任务

# 功能介绍

新建任务：新建编译打包任务

用户：Jenkins用户管理

构建历史：构建历史记录信息

系统管理：Jenkins系统管理信息，包括系统设置、全局安全配置、工具配置、读取控制、管理插件等。

我的视图：当前用户建立视图

新建视图：新建用户视图

# 新建任务

输入一个任务名称

» 该字段不能为空, 请输入一个合法的名称



构建一个自由风格的软件项目

这是Jenkins的主要功能,Jenkins将会结合任何SCM和任何构建系统来构建你的项目,甚至可以构建软件以外的系统.

确定

# 用户

- 新建任务
- 用户
- 构建历史
- 系统管理
- 我的视图
- 新建视图

**构建队列** -

队列中没有构建任务

**构建执行状态** -

- 1 空闲
- 2 空闲

## 用户

包含所有已知“用户”，包括当前安全域中的登录ID和在变更记录的提交者的信息的人

用户标识	名称	Last Commit Activity ↑	在线
 <a href="#">admin</a>	<a href="#">admin</a>	N/A	

图标: [S](#) [M](#) [L](#)



# 构建历史

- 新建任务
- 用户
- 构建历史
- 系统管理
- 我的视图
- 新建视图

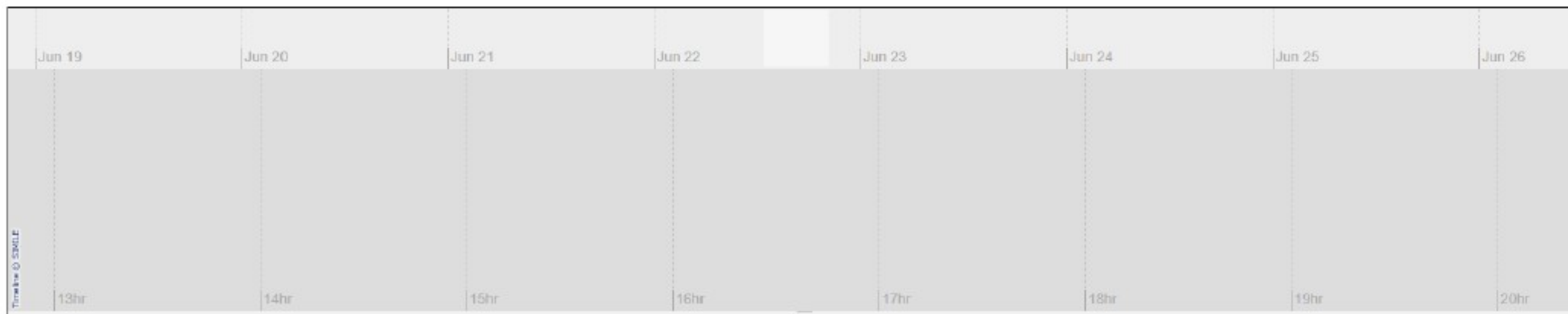
#### 构建队列

队列中没有构建任务

#### 构建执行状态

- 1 空闲
- 2 空闲

## Jenkins 的构建历史



[导出为XML \(Recurse in subfolders\)](#)

构建	日期 ↑	状态
----	------	----










图标: S M L

[图标](#) [RSS 全部](#) [RSS 失败](#) [RSS 最新的构建](#)

# 系统管理

localhost:8080/manage

Jenkins

-  **系统设置**  
全局设置&路径
-  **全局安全配置**  
Jenkins安全，定义谁可以访问或使用系统。
-  **全局工具配置**  
工具配置，包括它们的位置和自动安装器
-  **读取设置**  
放弃当前内存中所有的设置信息并从配置文件中重新读取 仅用于当您手动修改配置文件时重新读取设置。
-  **管理插件**  
添加、删除、禁用或启用Jenkins功能扩展插件。
-  **系统信息**  
显示系统环境信息以帮助解决问题。
-  **系统日志**  
系统日志从java.util.logging捕获Jenkins相关的日志信息。
-  **负载统计**  
检查您的资源利用情况，看看是否需要更多的计算机来帮助您构建。
-  **Jenkins CLI**  
从您命令行或脚本访问或管理您的Jenkins。
-  **脚本命令行**  
执行用于管理或故障探测或诊断的任意脚本命令。
-  **管理节点**  
添加、删除、控制和监视系统运行任务的节点。
-  **关于Jenkins**  
查看版本以及证书信息。
-  **Manage Old Data**  
Scrub configuration files to remove remnants from old plugins and earlier versions.

# 我的视图

The screenshot shows the Jenkins web interface. The browser address bar displays `localhost:8080/me/my-views/view/all/`. The page header includes the Jenkins logo, a search bar, and the user name `admin` with a `注销` (Logout) link. The breadcrumb navigation shows `Jenkins > admin > 我的视图 > 所有`. On the right side of the header, there are links for `关闭自动刷新` and `添加说明`. The main content area features a sidebar with navigation options: `新建任务`, `用户`, `构建历史`, and `新建视图`. The central part of the page displays a welcome message: `欢迎使用Jenkins!` with a button to `开始 创建一个新任务`. Below this, there are two expandable sections: `构建队列` (Build Queue) and `构建执行状态` (Build Execution Status). The `构建队列` section shows `队列中没有构建任务`. The `构建执行状态` section shows two entries, both labeled `空闲` (Idle).

localhost:8080/me/my-views/view/all/

Jenkins

admin | 注销

Jenkins > admin > 我的视图 > 所有

关闭自动刷新

添加说明

新建任务

用户

构建历史

新建视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

欢迎使用Jenkins!

开始 创建一个新任务

# 新建视图

The screenshot shows the Jenkins web interface at localhost:8080/newView. The page title is 'Jenkins' and the user is logged in as 'admin'. The left sidebar contains navigation links: '新建任务', '用户', '构建历史', '系统管理', '我的视图', and '新建视图'. The main content area is a dialog for creating a new view. It features a text input field for '视图名称' (View Name) and two radio button options: '列表视图' (List View) and '我的视图' (My View). Below the options is an 'OK' button. The '构建队列' (Build Queue) section shows '队列中没有构建任务' (No build tasks in queue). The '构建执行状态' (Build Execution Status) section shows two '空闲' (Idle) jobs.

localhost:8080/newView

Jenkins

admin | 注销

新建任务

用户

构建历史

系统管理

我的视图

新建视图

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

视图名称

列表视图  
显示简单列表。你可以从中选择任务来显示在某个视图中

我的视图  
该视图自动显示当前用户有权限访问的任务

OK

# 功能-Pipeline

Jenkins Pipeline是一套插件，支持将连续输送Pipeline实施和整合到Jenkins。Pipeline提供了一组可扩展的工具，用于通过PipelineDSL为代码创建简单到复杂的传送Pipeline

# 功能-Pipeline

新建pipeline,构建三个状态，运行

三个状态分别输出消息，后面要添加构建脚本



The screenshot displays the Jenkins configuration page for a pipeline. At the top, there is a section titled "高级项目选项" (Advanced Project Options) with a search icon and a "高级..." (Advanced...) button. Below this is the "流水线" (Pipeline) section, where the definition is set to "Pipeline script". The script content is as follows:

```
1 node {  
2   stage('Build') {  
3     echo 'Building....'  
4   }  
5   stage('Test') {  
6     echo 'Building....'  
7   }  
8   stage('Deploy') {  
9     echo 'Deploying....'  
10  }  
11 }
```

At the bottom left, there are buttons for "保存" (Save) and "应用" (Apply), along with the text "使用 Groovy 沙盒" (Use Groovy Sandbox). A "try sample Pipeline..." button is also visible on the right side of the script editor.

# 功能-Pipeline

The screenshot shows a Jenkins pipeline execution page for 'my-pipeline-three 1'. The browser address bar indicates the URL is localhost:8081/blue/organizations/jenkins/my-pipeline-three/detail/my-pipeline-three/1/pipeline. The page header includes a green bar with the pipeline name and several action buttons: '流水线' (Pipeline), '改变' (Change), '测试' (Test), 'Artifacts', a refresh icon, a settings icon, a share icon, and a close icon. Below the header, the pipeline's branch and commit information are displayed: '分支: -' (Branch: -), '提交: -' (Commit: -), '2s' (2s), 'a few seconds ago', '没有修改' (No changes), and '由用户 admin 启动' (Started by user admin). The pipeline progress is shown as a horizontal line with five stages: 'Start', 'Build', 'Test', 'Deploy', and 'End'. The 'Build', 'Test', and 'Deploy' stages are marked with green checkmarks, indicating successful completion. The 'Deploy' stage is currently active, as indicated by a blue circle around its checkmark. Below the progress bar, the 'Deploy' stage details are shown: 'Deploy - <1s' and a log entry: 'Deploying... - Print Message' with a duration of '<1s'. The log entry is preceded by a green checkmark and a right-pointing arrow.

# 功能-Blue Ocean

BlueOcean重新考虑了Jenkins的用户体验。BlueOcean由Jenkins Pipeline设计，但仍然兼容自由式工作，减少了团队成员的混乱，增加了清晰度

连续交付（CD）Pipeline的复杂可视化，允许快速和直观地了解Pipeline的状态。

Pipeline编辑器通过引导用户直观和可视化的过程创建Pipeline，使创建Pipeline平易近人。

个性化，以适应团队每个成员的角色需求。

需要干预和/或出现问题时确定精度。BlueOcean显示了Pipeline需要注意的地方，便于异常处理和提高了生产率。

用于分支和拉取请求的本地集成可以在GitHub和Bitbucket中与其他人进行代码协作时最大限度提高开发人员的生产力。



# 功能-Blue Ocean-pipeline

✓ 选择代码仓库

Bitbucket Cloud    Bitbucket Server

GitHub    GitHub Enterprise

Git

✓ Connect to GitHub

Jenkins needs an access token to authorize itself with GitHub.  
[Create an access token here.](#)

..... ✓

✓ Which organization does the repository belong to?

leebean123

✓ Choose a repository

Loaded 2 repositories

Search...

Redis

simple-java-maven-app

创建流水线

○ Creating Pipeline...

# 功能-Blue Ocean-pipeline

✓ demo-pipeline-01 1

分支: master 32s 没有修改  
提交: 3de4a21 a minute ago Branch indexing

流水线 改变 测试 Artifacts 刷新 设置 分享 退出

Start build test deploy End

deploy - <1s

✓ > deploy success - Print Message <1s

The screenshot displays a Blue Ocean pipeline interface. At the top, a green header bar shows the pipeline name 'demo-pipeline-01 1' and a toolbar with icons for '流水线' (Pipeline), '改变' (Change), '测试' (Test), 'Artifacts', a refresh icon, a settings icon, a share icon, and an '退出' (Exit) button. Below the header, a green bar provides details: '分支: master' (Branch: master) with a commit hash '3de4a21' and a duration of '32s', and '提交: 3de4a21' (Commit: 3de4a21) with a timestamp 'a minute ago' and the message 'Branch indexing'. The main area features a horizontal pipeline diagram with nodes 'Start', 'build', 'test', 'deploy', and 'End'. The 'build', 'test', and 'deploy' nodes are marked with green checkmarks, indicating success. The 'deploy' node is highlighted with a blue circle. At the bottom, a log entry for the 'deploy' step is shown, indicating 'deploy success' with a duration of '<1s' and a 'Print Message' option.

- ① SonarQube简介
- ② SonarQube安装
- ③ SonarQube主要功能
- ④ **SonarQube插件**
- ⑤ SonarQube集成
- ⑥ SonarQube API

# Jenkins插件

Jenkins强大的插件扩展机制使得Jenkins的功能非常强大。

从Jenkins现有的功能扩展或开发者们为Jenkins提供的新功能都可以称之为Jenkins插件。有些插件可以无缝添加到您的构建过程，而其它，诸如除CVS和Subversion的SCM插件则需要源代码控制系统的支持

支持包括Maven、Git、PipeLine等

# Jenkins插件-更新中心

localhost:8080/pluginManager/available

Jenkins > 插件管理

<input type="checkbox"/>	<a href="#">Violations</a>	0.7.11
<input type="checkbox"/>	<a href="#">Visual Studio Code Metrics</a>	1.7
<input type="checkbox"/>	<a href="#">VSTest Runner</a>	1.0.7
<input type="checkbox"/>	<a href="#">Start Windocks Containers</a> Allows for the creation and start of Windocks Containers	1.4
<input type="checkbox"/>	<a href="#">change-assembly-version-plugin</a>	1.10
DevOps		
<input type="checkbox"/>	<a href="#">Ansible</a> Ansible support in Jenkins	1.0
<input type="checkbox"/>	<a href="#">Fortify on Demand Uploader</a> Gathers build artifacts and uploads them to the Fortify on Demand service with user-specified options.	3.0.8
<input type="checkbox"/>	<a href="#">Hubot Pipeline Steps</a> Hubot Pipeline Steps	2.0.0
<input type="checkbox"/>	<a href="#">OctoPerf Load Testing Plugin</a> Run load tests on OctoPerf.com from Jenkins.	4.0.1
<input type="checkbox"/>	<a href="#">OpenShift Client</a> A Jenkins plugin which leverages the OpenShift client binary (oc) to enable powerful Pipeline DSL based interactions with OpenShift clusters.	1.0.13
<input type="checkbox"/>	<a href="#">Puppet Enterprise Pipeline</a>	1.3.1
<input type="checkbox"/>	<a href="#">Serena SRA Deploy</a> Serena SRA Deploy for Jenkins	1.4.2.4
<input type="checkbox"/>	<a href="#">CodeSonar</a>	2.0.7
<input type="checkbox"/>	<a href="#">Docker Pipeline</a> Build and use Docker containers from pipelines	1.17
Groovy相关		
<input type="checkbox"/>	<a href="#">Global Post Script</a>	1.1.4
<input type="checkbox"/>	<a href="#">Groovy Remote Control</a> Groovy Remote Control Plugin	0.2
<input type="checkbox"/>	<a href="#">Matrix Groovy Execution Strategy</a> Matrix Groovy Execution Strategy Plugin	1.0.7
<input type="checkbox"/>	<a href="#">Groovy</a>	2.0
<input type="checkbox"/>	<a href="#">Config File Provider</a> Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace	2.18
<input type="checkbox"/>	<a href="#">Groovy Doctool</a>	...

[直接安装](#) [下载并稍后安装](#) 3 小时 30 分 之前获取了更新信息 [立即获取](#)

- ① Jenkins简介
- ② Jenkins安装
- ③ Jenkins主要功能
- ④ Jenkins插件
- ⑤ **Jenkins集成**
- ⑥ Jenkins API

# Jenkins集成

- Maven
- GitLab
- Nexus

# Jenkins集成Maven

## Jenkins安装Maven插件支持

### Maven

Maven 安装

新增 Maven

Maven

Name

E:\hx\_code\maven

MAVEN\_HOME

自动安装



删除 Maven

新增 Maven

系统下Maven 安装列表

Save

Apply



# Jenkins集成GitLab

安装插件，支持git，安装GIT plugin

## Git

Git installations

### Git

Name

Path to Git executable

自动安装

Delete Git

(2) Git 的配置



## Git

Git installations

### Git

Name

Path to Git executable

自动安装

Add Git

description

# Jenkins集成nexus

## Jenkins安装拆建--Maven Integration plugin

高级...

---

### JDK

JDK 安装

新增 JDK

JDK

别名

自动安装

新增安装

新增 JDK

系统下JDK 安装列表

删除 JDK

---

系统下Ant 安装列表

### Maven

Maven 安装

新增 Maven

Maven

Name

MAVEN\_HOME

自动安装

新增 Maven

系统下Maven 安装列表

删除 Maven

- ① Jenkins简介
- ② Jenkins安装
- ③ Jenkins主要功能
- ④ Jenkins插件
- ⑤ Jenkins集成
- ⑥ **Jenkins API**

# Jenkins API

Jenkins提供了一些RESTful API，实现系统集成，包括查询Job信息、启动Job等

<https://wiki.jenkins.io/display/JENKINS/Remote+access+API>

## jenkins-rest

---

Java client, built on top of jclouds, for working with Jenkins REST API.

### Setup

---

Client's can be built like so:

```
JenkinsClient client = JenkinsClient.builder()
    .endPoint("http://127.0.0.1:8080") // Optional. Defaults to http://127.0.0.1:8080
    .credentials("admin:password") // Optional.
    .build();

SystemInfo systemInfo = client.api().systemApi().systemInfo();
assertTrue(systemInfo.jenkinsVersion().equals("1.642.4"));
```

### Latest release

---

Can be found in jcenter like so:

```
<dependency>
  <groupId>com.cdancy</groupId>
  <artifactId>jenkins-rest</artifactId>
  <version>X.Y.Z</version>
</dependency>
```