

DevOps实战

一、DevOps概念

1、DevOps是什么

Development和Operations的组合词；



无尽头的可能性：DevOps涵盖了代码、部署目标的发布和反馈等环节，闭合成一个无限大符号形状的DevOps能力闭环。

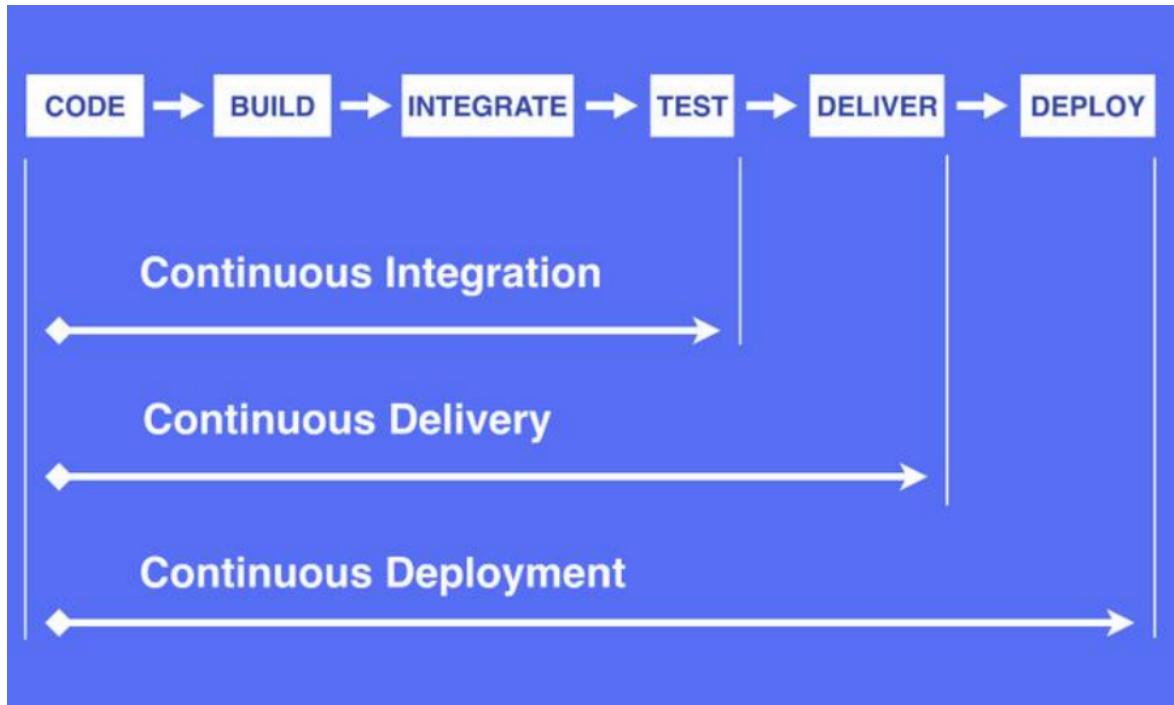
DevOps: Development 和 Operations 的组合

- DevOps 看作开发（软件工程）、技术运营和质量保障（QA）三者的交集。
- 突出重视软件开发人员和运维人员的沟通合作，通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠。
- DevOps 希望做到的是软件产品交付过程中 **IT 工具链的打通**，使得各个团队减少时间损耗，更加高效地协同工作。专家们总结出了下面这个 DevOps 能力图，良好的闭环可以大大增加整体的产出。

2、CICD是什么

持续集成 持续部署

1、基本理念



1、持续集成 (Continuous Integration)

持续集成是指软件个人研发的部分向软件整体部分交付，频繁进行集成以便更快地发现其中的错误。“持续集成”源自于极限编程（XP），是XP最初的12种实践之一。

CI 需要具备这些：

- **全面的自动化测试。**这是实践持续集成&持续部署的基础，同时，选择合适的自动化测试工具也极其重要；
- **灵活的基础设施。**容器，虚拟机的存在让开发人员和QA人员不必再大费周折；
- **版本控制工具。**如 Git, CVS, SVN 等；
- **自动化的构建和软件发布流程的工具，**如 Jenkins, flow.ci；
- **反馈机制。**如构建/测试的失败，可以快速地反馈到相关负责人，以尽快解决达到一个更稳定的版本。

2、持续交付 (Continuous Delivery)

持续交付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」（production-like environments）中。持续交付优先于整个产品生命周期的软件部署，建立在高水平自动化持续集成之上。

灰度发布。

持续交付和持续集成的优点非常相似：

- **快速发布。**能够应对业务需求，并更快地实现软件价值。
- **编码->测试->上线->交付的频繁迭代周期缩短，同时获得迅速反馈；**
- **高质量的软件发布标准。**整个交付过程标准化、可重复、可靠，
- **整个交付过程进度可视化，方便团队人员了解项目成熟度；**

- 更先进的团队协作方式。从需求分析、产品的用户体验到交互设计、开发、测试、运维等角色密切协作，相比于传统的瀑布式软件团队，更少浪费。

3. 持续部署 (Continuous Deployment)

持续部署是指当交付的代码通过评审之后，**自动部署到生产环境中**。持续部署是持续交付的最高阶段。这意味着，所有通过了一系列的自动化测试的改动都将自动部署到生产环境。它也可以被称为“Continuous Release”。

“开发人员**提交代码**，持续集成服务器获取代码，执行单元测试，根据测试结果决定是否部署到预演环境，如果成功部署到预演环境，进行整体验收测试，如果测试通过，**自动部署到产品环境**，**全程自动化高效运转**。”

持续部署主要好处是，可以相对独立地部署新的功能，并能快速地收集真实用户的反馈。

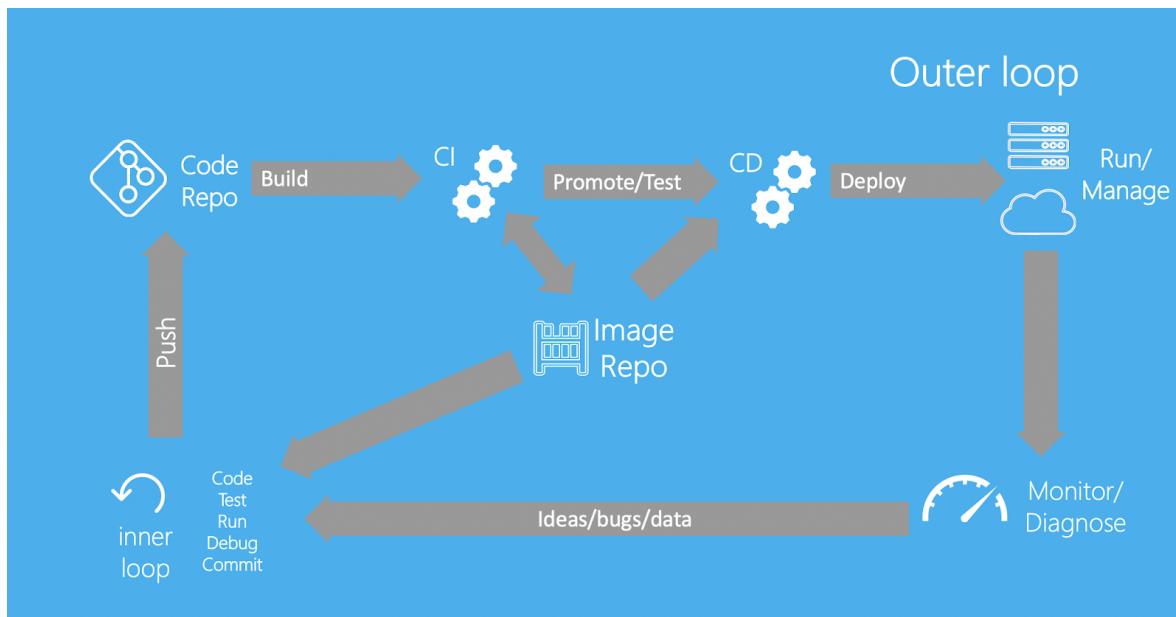
“You build it, you run it”，这是 Amazon 一年可以完成 5000 万次部署，平均每个工程师每天部署超过 50 次的核心秘籍。

$$5000/365 = 15 \text{ 万次}$$

开发人员代码敲完。可以release的时候，提交代码，剩下的全部一站式自动搞定

2. 最佳实践

1. 内循环与外循环

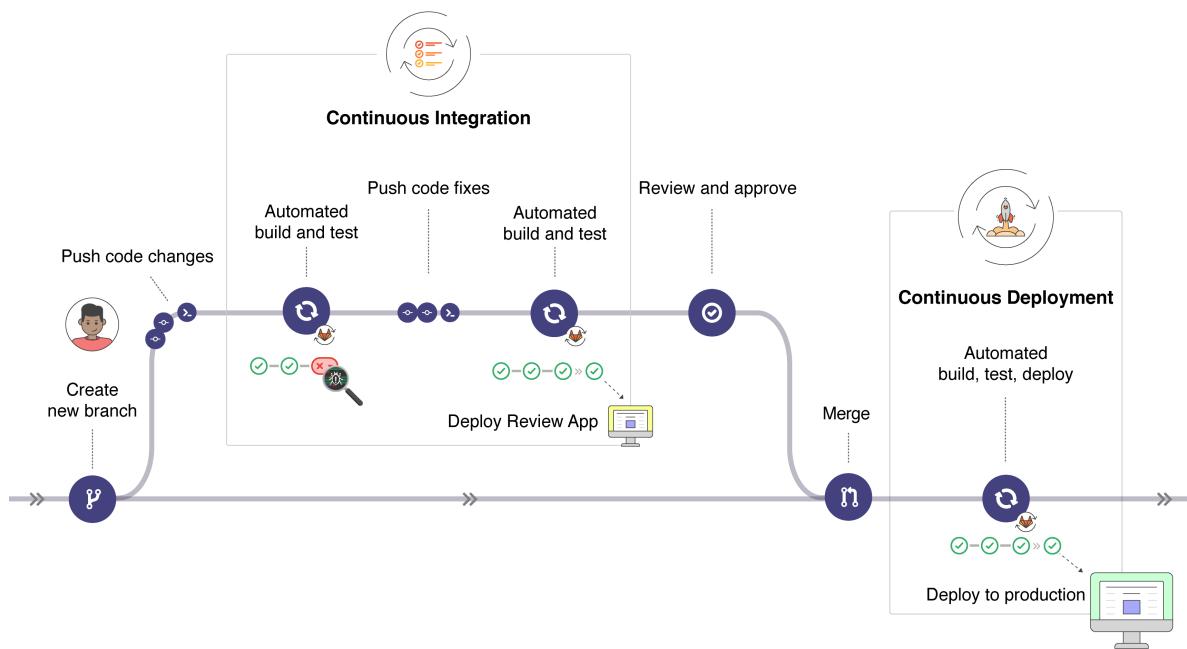


- 内循环（开发要做的事情）：
 - 编码、测试、运行、debug、提交
- 代码推送到代码仓库 (svn, git) 【代码回滚】
- 进行CI过程（持续集成），万物皆可容器化。打包成一个Docker镜像
- 镜像推送到镜像仓库
- 测试
- 持续部署流程（CD），拿到之前的镜像，进行CD。怎么放到各种环境。uat、test、prod
- 外循环（）
 - 运行时监控

- 生产环境的管理
- 监控
- 线上反馈到开发
- 来到内循环

MVC: Model (bean,entity,to,po.....) View(thymeleaf、前后分离....) Controller (xxxxx)

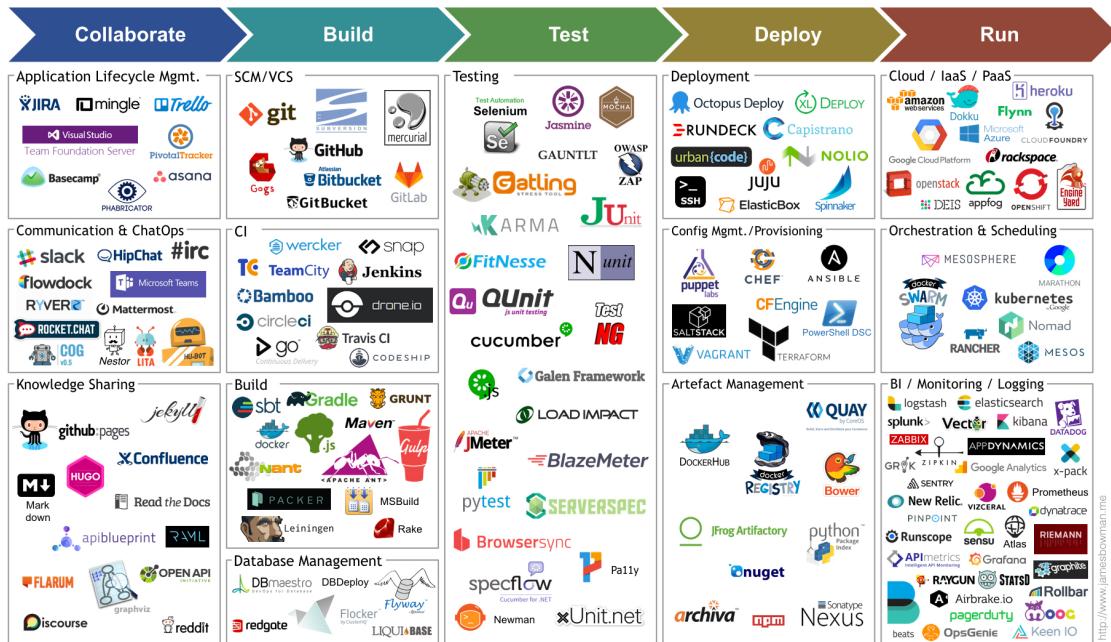
2、实践流程



新功能，bug修复。

- 创建分支来做这个事情（开发功能）
- 提交分支的代码改变
- **进入持续集成流程**
 - 当前分支代码功能性自动化构建和测试
 - 自动工具推送这次提交
 - 自动化集成测试
 - 可以看到效果
 - 人工确认此次功能是否发布到生产环境
- 代码合并。
- **进入持续部署流程**
 - 构建、测试、发布.....

3、CICD LandSpace



二、Jenkins

```

1 /var/jenkins_home jenkins的家目录
2 包含了jenkins的所有配置。
3
4
5 以后要注意备份 /var/jenkins_home (以文件的方式固化的)

```

Java

Jenkins镜像用 <https://hub.docker.com/r/jenkinsci/jenkins/>

驱动我们整个CI&CD过程的很多工具



1、Jenkins安装

<https://www.jenkins.io/zh/doc/book/installing/>

```
1 docker run \
2   -u root \
3   -d \
4   -p 8080:8080 \
5   -p 50000:50000 \
6   -v jenkins-data:/var/jenkins_home \
7   -v /etc/localtime:/etc/localtime:ro \
8   -v /var/run/docker.sock:/var/run/docker.sock \
9   --restart=always \
10  jenkinsci/blueocean
11
12
13 #自己构建镜像 RUN的时候就把时区设置好
14 #如果是别人的镜像, docker hub, UTC: 容器运行时, -v
15 /etc/localtime:/etc/localtime:ro
16
17
18 jenkinsci/jenkins 是没有 blueocean插件的, 得自己装
19 jenkinsci/blueocean: 带了的
20
21 #/var/run/docker.sock 表示Docker守护程序通过其监听的基于Unix的套接字。 该映射允许
jenkinsci/blueocean 容器与Docker守护进程通信, 如果 jenkinsci/blueocean 容器需要实例化
其他Docker容器, 则该守护进程是必需的。 如果运行声明式管道, 其语法包含agent部分用 docker; 例
如, agent { docker { ... } } 此选项是必需的。
22
23 #如果你的jenkins 安装插件装不上。使用这个镜像【 registry.cn-
qingdao.aliyuncs.com/lfy/jenkins:plugins-blueocean 】默认访问账号/密码是
【admin/admin】
```

安装插件，并配置用户

创建第一个管理员用户

用户名:	<input type="text" value="lfy"/>
密码:	<input type="password" value="....."/>
确认密码:	<input type="password" value="....."/>
全名:	<input type="text" value="leifengyang"/>
电子邮件地址:	<input type="text" value="534096094@qq.com"/>

```
1 #备份jenkins
2 tar -cvf jenkins_data.tar /var/lib/docker/volumes/jenkins-data/_data/
3
4 #恢复jenkins
5 tar -xvf jenkins_data.tar /var/lib/docker/volumes/jenkins-data/_data/
```

2、Jenkins实战

代码在本地修改----提交到远程gitee----触发jenkins整个自动化构建流程（打包，测试，发布，部署）

1、准备一个git项目进行测试

我们以gitee为例，github可能太慢了。需要idea安装gitee插件。或者自己熟悉手动命令也行。

步骤：

- 1、idea创建Spring Boot项目
- 2、VCS - 创建git仓库
- 3、gitee创建一个空仓库，示例为public
- 4、idea提交内容到gitee
- 5、开发项目基本功能，并在项目中创建一个Jenkinsfile文件
- 6、创建一个名为 devops-java-demo的流水线项目，使用项目自己的流水线

Jenkins的工作流程

1、先定义一个流水线项目，指定项目的git位置

- 流水线启动
 - 1、先去git位置自动拉取代码
 - 2、解析拉取代码里面的Jenkinsfile文件
 - 3、按照Jenkinsfile指定的流水线开始加工项目

Jenkins重要的点

0、jenkins的家目录 /var/jenkins_home 已经被我们docker外部挂载了；

/var/lib/docker/volumes/jenkins-data/_data

1、WORKSPACE (工作空间) =/var/jenkins_home/workspace/java-devops-demo

- 每一个流水线项目，占用一个文件夹位置
- BUILD_NUMBER=5; 当前第几次构建
- WORKSPACE_TMP (临时目录) =/var/jenkins_home/workspace/java-devops-demo@tmp

```
1   JOB_URL=http://139.198.9.163:8080/job/java-devops-demo/
```

1

1

1

2、常用的环境如果没有，jenkins配置环境一大堆操作

3、jenkins_url：<http://139.198.27.103:8080/>

1

2、远程构建触发

期望效果：远程的github代码提交了，jenkins流水线自动触发构建。

实现流程：

- 1、保证jenkins所在主机能被远程访问
- 2、jenkins中远程触发需要权限，我们应该使用用户进行授权
- 3、配置gitee/github， webhook进行触发

```
1 #远程构建即使配置了github 的webhook，默认会403.我们应该使用用户进行授权
2 1、创建一个用户
3 2、一定随便登陆激活一次
4 3、生成一个apitoken
5 http://leifengyang:113620edce6200b9c78ecadb26e9cf122e@139.198.186.134:8080/job/dev
  ops-java-demo/build?token=leifengyang
6
```

触发远程构建 (例如, 使用脚本)

身份验证令牌

Use the following URL to trigger build remotely: JENKINS_URL/job/simple-java-maven-app/build?token=TOKEN_NAME 或者
/buildWithParameters?token=TOKEN_NAME
 Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

远程触发：[JENKINS_URL /job/simple-java-maven-app/build?token=TOKEN_NAME](http://JENKINS_URL/job/simple-java-maven-app/build?token=TOKEN_NAME) 请求即可

6、流水线语法

1、基础格式

```
1 pipeline {
2     agent any
3     environment {
4         CC = 'clang'
5     }
6     stages {
7         stage('Example') {
8             steps {
9                 sh 'printenv'
10                sh 'echo $CC'
11            }
12        }
13    }
14 }
```

2、环境变量

3、密钥管理

```
// environment aa = credxxx("xxxxx")
```

4、自定义agent

The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text "docker". Below it, there are four tabs: "可更新" (Updatable), "可选插件" (Optional Plugins) which is currently selected, "已安装" (Installed), and "高级" (Advanced). Under the "Name" column, the "Docker Pipeline" plugin is listed. It has a checkbox next to its name, which is checked. Below the checkbox are several labels: "云提供商" (Cloud Provider), "集群管理和分布式构建" (Cluster Management and Distributed Build), and "docker". A descriptive text below the labels states: "This plugin integrates Jenkins with Docker". In the "Docker Commons" section, another plugin named "Docker API" is listed with a checked checkbox, and its labels include "api-plugin", "docker", and "插件库 (被其他插件使用)" (Plugin Library (Used by other plugins)). A brief description for this section is: "Provides the common shared functionality for various Docker-related plugins." A red box highlights the "Docker Pipeline" section. At the bottom of the page, a code snippet is shown:

```
1 //需要安装docker、docker pipeline插件
2
3 pipeline {
4     agent none
```

```

5      stages {
6          stage('Example Build') {
7              agent {
8                  docker 'maven:3-alpine'
9                  //args 是指定 docker run 的所有指令
10                 args '-v /var/jenkins_home/maven/.m2:/root/.m2'
11             }
12             steps {
13                 echo 'Hello, Maven'
14                 sh 'mvn --version'
15             }
16         }
17         stage('Example Test') {
18             agent { docker 'openjdk:8-jre' }
19             steps {
20                 echo 'Hello, JDK'
21                 sh 'java -version'
22             }
23         }
24     }
25 }
```

- 配置maven加速
 - 把Maven的配置文件放在jenkins-data里面的某个位置

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5      xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
6      http://maven.apache.org/xsd/settings-1.0.0.xsd">
7      <!-- localRepository
8          | The path to the local repository maven will use to store artifacts.
9          |
10         | Default: ${user.home}/.m2/repository
11     <localRepository>H:\Devsoft\apache-maven-3.6.1\repository</localRepository>
12     用户目录下的.m2是所有jar包的地方； maven容器内jar包的位置
13     -->
14     <localRepository>/root/.m2</localRepository>
15
16
17     <pluginGroups>
18
19     </pluginGroups>
20
21     <proxies>
22
23     </proxies>
24
25     <servers>
26
27     </servers>
28
29
30     <mirrors>
```

```

31      <mirror>
32          <id>nexus-aliyun</id>
33          <mirrorOf>central</mirrorOf>
34          <name>Nexus aliyun</name>
35          <url>http://maven.aliyun.com/nexus/content/groups/public</url>
36      </mirror>
37  </mirrors>
38
39  <profiles>
40      <profile>
41          <id>jdk-1.8</id>
42          <activation>
43              <activeByDefault>true</activeByDefault>
44              <jdk>1.8</jdk>
45          </activation>
46          <properties>
47              <maven.compiler.source>1.8</maven.compiler.source>
48              <maven.compiler.target>1.8</maven.compiler.target>
49
50          <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>
51          </properties>
52      </profile>
53  </profiles>
54 </settings>

```

默认所有的可变配置项都推荐放在jenkins-home的位置，增强移植性

如何自定义使用maven的配置文件

```

1  //流水线这样
2  stage('编译'){
3      agent {
4          docker {
5              image 'maven:3-alpine' //用完就会杀掉
6              args '-v maven-repo:/root/.m2'
7              //docker run -v repo:/root/.m2 这个容器是直接在linux上的,docker
8              volume也很灵活
9          }
10     }
11     steps {
12         sh 'pwd && ls -alh'
13         sh 'mvn -v'
14         sh 'mvn clean package -s
"var/jenkins_home/appconfig/maven/settings.xml" -Dmaven.test.skip=true '
15         //残留的问题，下一次运行这个命令，已经之前下载的jar还会下载。
16     }
17 }
18
19 //再给 linux主机挂载的jenkins_home位置自定义一个 settings.xml 即可
20 //原理： jenkins在解析流水线期间，可以任意访问jenkins家目录的位置和相关环境信息
21
22 //一个配置文件+docker maven agent = maven加速

```

- 临时容器导致的问题
 - 1、第一次检出代码， 默认在 /var/jenkins_home/workspace/【java-devops-demo】
 - 2、使用docker临时agent的时候，每一个临时容器运行又分配临时目录
/var/jenkins_home/workspace/java-devops-demo@2；默认就是workspace/java-devops-demo 的内容
 - 3、在临时容器里面 运行的mvn package命令，会在 /var/jenkins_home/workspace/java-devops-demo@2 进行工作
 - 4、package到了 /var/jenkins_home/workspace/java-devops-demo@2 位置
 - 5、进入下一步进行打包镜像，又会回到 /var/jenkins_home/workspace/【java-devops-demo】这个默认位置
 - 6、这个位置没有运行过 mvn clean package , 没有target。默认的工作目录没有 target
 -

5、参数化

6、条件表达式

```

1  ##默认的when
2
3  ##还可以写脚本哦。。。。
4

```

7、修改jenkins插件源

推荐白天装

```

1  http://mirror.xmission.com/jenkins/updates/update-center.json
2

```

jenkins插件中心的插件；

The screenshot shows the Jenkins 'Advanced Settings' page. At the top, there's a navigation bar with 'Dashboard' and a red box around 'Plugins'. Below it are links: 'Return to Workbench', 'Manage Jenkins', and 'Update Center'. On the right, there are tabs: 'Updatable', 'Optional Plugins', 'Installed', and a red box around 'Advanced'. The main content area is titled 'Proxy Settings' with a 'Servers' tab. At the bottom left is a 'Upload' button, and at the bottom right is a message '1 day 10 hours ago updated information' with a 'Get Now' button.

8、推荐安装的插件

- Docker Pipeline && Docker
 - 安装Docker Pipeline会自动安装docker相关的
 - 这个允许我们自定义agent使用docker环境
- Git Parameter
 - 解析git参数，允许我们选择分支进行构建
- Active Choices
 - 可以做到参数的级联选择
- Generic Webhook Trigger
 - 通用的webhook触发器，构建更强大的webhook功能
- Role-based Authorization Strategy
 - RBAC权限指定，给一个用户精确指定权限
- List Git Branches Parameter
 - 列出分支参数
- Build With Parameters
 - 基于自定义参数构建

我的插件列表

返回工作台

管理 Jenkins

更新中心

 filter

可更新 可选插件 已安装 高级

启用	名称	版本	上一个安装的版本	卸载
Active Choices Plug-in				
<input checked="" type="checkbox"/>	This plug-in provides additional parameter types for jobs, that allow you to cascade changes and render images or other HTML elements instead of the traditional parameter.	2.5.6		
<input checked="" type="checkbox"/>	Ant Plugin Adds Apache Ant support to Jenkins	1.11		
Apache HttpComponents Client 4.x API Plugin				
Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.				
<input checked="" type="checkbox"/>	This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	4.5.13-1.0		
Authentication Tokens API Plugin				
<input checked="" type="checkbox"/>	This plugin provides an API for converting credentials into authentication tokens in Jenkins.	1.4		
Autofavorite for Blue Ocean				
<input checked="" type="checkbox"/>	Automatically favorites multibranch pipeline jobs when user is the author	1.2.4		
Bitbucket Branch Source Plugin				
Allows to use Bitbucket Cloud and Bitbucket Server as sources for multi-branch projects. It also provides the required connectors for Bitbucket Cloud Team and Bitbucket Server Project folder (also known as repositories auto-discovering).				
<input checked="" type="checkbox"/>	This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	2.9.8		
Bitbucket Pipeline for Blue Ocean				
<input checked="" type="checkbox"/>	BlueOcean Bitbucket pipeline creator	1.24.6		
Blue Ocean				
<input checked="" type="checkbox"/>	BlueOcean Aggregator	1.24.6		
Blue Ocean Core JS				
<input checked="" type="checkbox"/>	The Jenkins Plugins Parent POM Project	1.24.6		
Blue Ocean Pipeline Editor				
<input checked="" type="checkbox"/>	The Blue Ocean Pipeline Editor is the simplest way for anyone wanting to get started with creating Pipelines in Jenkins	1.24.6		
Bootstrap 4 API Plugin				
<input checked="" type="checkbox"/>	Provides Bootstrap 4 for Jenkins plugins.	4.6.0-3		
bouncycastle API				
<input checked="" type="checkbox"/>	This plugin provides an stable API to Bouncy Castle related tasks.	2.16.0		
Branch API Plugin				
<input checked="" type="checkbox"/>	This plugin provides an API for multiple branch based projects.	2.6.3		
Build Timeout				
This plugin allows builds to be automatically terminated after the specified amount of time has elapsed.				
<input checked="" type="checkbox"/>	This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	1.20		
Build With Parameters				
<input checked="" type="checkbox"/>	Allows the user to provide parameters for a build in the url (similar to /job/JOBNAME/buildWithParameters), prompting for confirmation before triggering the job.	1.5.1		
Checks API plugin				
<input checked="" type="checkbox"/>	This plugin defines an API for Jenkins to publish checks to SCM platforms.	1.7.0		
Command Agent Launcher				
<input checked="" type="checkbox"/>	Allows agents to be launched using a specified command.	1.2		
Common API for Blue Ocean				
<input checked="" type="checkbox"/>	This plugin is a part of Blue Ocean UI	1.24.6		
Config API for Blue Ocean				
<input checked="" type="checkbox"/>	BlueOcean Analytics Tools plugin	1.24.6		
Credentials Binding Plugin				
<input checked="" type="checkbox"/>	Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.24		
Credentials Plugin				
<input checked="" type="checkbox"/>		2 3 17		

	This plugin allows you to store credentials in Jenkins.	View
<input checked="" type="checkbox"/>	Dashboard for Blue Ocean Blue Ocean Dashboard	1.24.6 卸载
<input checked="" type="checkbox"/>	Design Language The Jenkins Plugins Parent POM Project	1.24.6 卸载
<input checked="" type="checkbox"/>	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.3.4 卸载
<input checked="" type="checkbox"/>	Display URL for Blue Ocean This plugin generates BlueOcean specific URLs for the Display URL plugin.	2.4.1 卸载
<input checked="" type="checkbox"/>	Docker Commons Plugin Provides the common shared functionality for various Docker-related plugins.	1.17 卸载
<input checked="" type="checkbox"/>	Docker Pipeline Build and use Docker containers from pipelines.	1.26 卸载
<input checked="" type="checkbox"/>	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.35 卸载
<input checked="" type="checkbox"/>	ECharts API Plugin Provides ECharts for Jenkins plugins.	5.0.2-1 卸载
<input checked="" type="checkbox"/>	Email Extension Plugin This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says	2.82 卸载
<input checked="" type="checkbox"/>	Events API for Blue Ocean Blue Ocean Events	1.24.6 卸载
<input checked="" type="checkbox"/>	Favorite This plugin allows users to favorite a job.	2.3.3 卸载
<input checked="" type="checkbox"/>	Folders Plugin This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.	6.15 卸载
<input checked="" type="checkbox"/>	Font Awesome API Plugin Provides the free fonts of Font Awesome for Jenkins plugins.	5.15.2-2 卸载
<input checked="" type="checkbox"/>	Generic Webhook Trigger Plugin Can receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. Works with GitHub, GitLab, Bitbucket, Jira and many more.	1.72 卸载
<input checked="" type="checkbox"/>	Git client plugin Utility plugin for Git support in Jenkins	3.7.1 卸载
<input checked="" type="checkbox"/>	Git Parameter Plug-In Adds ability to choose branches, tags or revisions from git repositories configured in project.	0.9.13 卸载
<input checked="" type="checkbox"/>	Git Pipeline for Blue Ocean BlueOcean Git SCM pipeline creator	1.24.6 卸载
<input checked="" type="checkbox"/>	Git plugin This plugin integrates Git with Jenkins.	4.7.1 卸载
<input checked="" type="checkbox"/>	GIT server Plugin Allows Jenkins to act as a Git server.	1.9 卸载
<input checked="" type="checkbox"/>	GitHub API Plugin This plugin provides GitHub API for other plugins.	1.123 卸载
<input checked="" type="checkbox"/>	GitHub Branch Source Plugin Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.10.2 卸载
<input checked="" type="checkbox"/>	GitHub Pipeline for Blue Ocean BlueOcean GitHub organization pipeline creator	1.24.6 卸载
<input checked="" type="checkbox"/>	GitHub plugin This plugin integrates GitHub to Jenkins.	1.33.1 卸载
<input checked="" type="checkbox"/>	Gradle Plugin This plugin allows Jenkins to invoke Gradle build scripts directly.	1.36 卸载
<input checked="" type="checkbox"/>	Handy Uri Templates 2.x API Plugin Bundles Handy Uri Templates 2.x and allows it to be used by Jenkins plugins	2.1.8-1.0 卸载
<input checked="" type="checkbox"/>	HTML Publisher plugin This plugin publishes HTML reports.	1.25 卸载
<input checked="" type="checkbox"/>	i18n for Blue Ocean Blue Ocean Internationalization (i18n) Plugin. This plugin is a part of the Blue Ocean Plugin set.	1.24.6 卸载
<input checked="" type="checkbox"/>	Jackson 2 API Plugin This plugin exposes the Jackson 2 JSON APIs to other Jenkins plugins.	2.12.2 卸载
<input checked="" type="checkbox"/>	Java JSON Web Token (JWT) Plugin Bundles the Java JSON Web Token (JWT) library.	0.11.2-9.c8b45b8bb173 卸载

<input checked="" type="checkbox"/>	JavaScript GUI Lib: ACE Editor bundle plugin	1.1	<button>卸载</button>
<input checked="" type="checkbox"/>	JavaScript GUI Lib: Handlebars bundle plugin	3.0.8	<button>卸载</button>
<input checked="" type="checkbox"/>	JavaScript GUI Lib: Moment.js bundle plugin	1.1.1	<button>卸载</button>
<input checked="" type="checkbox"/>	jQuery plugin	1.12.4-1	<button>卸载</button>
<input checked="" type="checkbox"/>	JQuery3 API Plugin	3.6.0-1	<button>卸载</button>
<input checked="" type="checkbox"/>	JSch dependency plugin	0.1.55.2	<button>卸载</button>
<input checked="" type="checkbox"/>	JUnit Plugin	1.49	<button>卸载</button>
<input checked="" type="checkbox"/>	JWT for Blue Ocean	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	LDAP Plugin	2.5	<button>卸载</button>
<input checked="" type="checkbox"/>	List Git Branches Parameter Plugin	0.0.9	<button>卸载</button>
<input checked="" type="checkbox"/>	Localization Support Plugin	1.1	<button>卸载</button>
<input checked="" type="checkbox"/>	Localization: Chinese (Simplified)	1.0.24	<button>卸载</button>
<input checked="" type="checkbox"/>	Lockable Resources plugin	2.10	<button>卸载</button>
<input checked="" type="checkbox"/>	Mailer Plugin	1.34	<button>卸载</button>
<input checked="" type="checkbox"/>	Matrix Authorization Strategy Plugin	2.6.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Matrix Project Plugin	1.18	<button>卸载</button>
<input checked="" type="checkbox"/>	OkHttp Plugin	3.14.9	<button>卸载</button>
<input checked="" type="checkbox"/>	Oracle Java SE Development Kit Installer	1.0	<button>卸载</button>
<input checked="" type="checkbox"/>	OWASP Markup Formatter Plugin	2.1	<button>卸载</button>
<input checked="" type="checkbox"/>	PAM Authentication plugin	1.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Personalization for Blue Ocean	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline	2.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline Graph Analysis Plugin	1.10	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline implementation for Blue Ocean	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline SCM API for Blue Ocean	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: API	2.42	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Basic Steps	2.23	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Build Step	2.13	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Declarative	1.8.4	<button>卸载</button>

<input checked="" type="checkbox"/>	Pipeline: Declarative Extension Points API APIs for extension points used in Declarative Pipelines.	1.8.4	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: GitHub Groovy Libraries Allows Pipeline Groovy libraries to be loaded on the fly from GitHub.	1.0	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Groovy Pipeline execution engine based on continuation passing style transformation of Groovy scripts.	2.90	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Input Step Adds the Pipeline step <code>input</code> to wait for human input or approval.	2.12	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Job Defines a new job type for pipelines and provides their generic user interface.	2.40	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Milestone Step Plugin that provides the milestone step	1.3.2	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Model API Model API for Declarative Pipeline.	1.8.4	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Multibranch Enhances Pipeline plugin to handle branches better by automatically grouping builds from different branches.	2.23	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Nodes and Processes Pipeline steps locking agents and workspaces, and running external processes that may survive a Jenkins restart or agent reconnection.	2.38	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: REST API Plugin Provides a REST API to access pipeline and pipeline run data.	2.19	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: SCM Step Adds a Pipeline step to check out or update working sources from various SCMs (version control).	2.12	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Shared Groovy Libraries Shared libraries for Pipeline scripts.	2.18	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Stage Step Adds the Pipeline step <code>stage</code> to delineate portions of a build.	2.5	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Stage Tags Metadata Library plugin for Pipeline stage tag metadata.	1.8.4	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Stage View Plugin Pipeline Stage View Plugin.	2.19	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Step API API for asynchronous build step primitive.	2.23	<button>卸载</button>
<input checked="" type="checkbox"/>	Pipeline: Supporting APIs Common utility implementations to build Pipeline Plugin	3.8	<button>卸载</button>
<input checked="" type="checkbox"/>	Plain Credentials Plugin Allows use of plain strings and files as credentials.	1.7	<button>卸载</button>
<input checked="" type="checkbox"/>	Plugin Utilities API Plugin Provides several utility classes that can be used to accelerate plugin development.	2.1.0	<button>卸载</button>
<input checked="" type="checkbox"/>	Popper.js API Plugin Provides Popper.js for Jenkins plugins.	1.16.1-2	<button>卸载</button>
<input checked="" type="checkbox"/>	Pub-Sub "light" Bus A simple Publish-Subscribe light-weight event bus for Jenkins	1.13	<button>卸载</button>
<input checked="" type="checkbox"/>	Resource Disposer Plugin Dispose resources asynchronously. Utility plugin for resources that require more retries or take a long time to delete.	0.15	<button>卸载</button>
<input checked="" type="checkbox"/>	REST API for Blue Ocean This plugin is a part of Blue Ocean UI	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	REST Implementation for Blue Ocean This plugin is a part of Blue Ocean UI	1.24.6	<button>卸载</button>
<input checked="" type="checkbox"/>	Role-based Authorization Strategy Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.	3.1.1	<button>卸载</button>
<input checked="" type="checkbox"/>	SCM API Plugin This plugin provides a new enhanced API for interacting with SCM systems.	2.6.4	<button>卸载</button>
<input checked="" type="checkbox"/>	Script Security Plugin Allows Jenkins administrators to control what in-process scripts can be run by less-privileged users.	1.76	<button>卸载</button>
<input checked="" type="checkbox"/>	Server Sent Events (SSE) Gateway Plugin Server Sent Events (SSE) Gateway.	1.24	<button>卸载</button>
<input checked="" type="checkbox"/>	Snakeyaml API Plugin This plugin provides Snakeyaml for other plugins.	1.27.0	<button>卸载</button>
<input checked="" type="checkbox"/>	SSH Build Agents Allows to launch agents over SSH, using a Java implementation	1.31.5	<button>卸载</button>

Plugin	Description	Version	Action
SSH Credentials	Allows storage of SSH credentials in Jenkins	1.18.1	<button>卸载</button>
Structs Plugin	Library plugin for DSL plugins that need names for Jenkins objects.	1.22	<button>卸载</button>
Timestamper	Adds timestamps to the Console Output	1.12	<button>卸载</button>
Token Macro Plugin	This plug-in adds reusable macro expansion capability for other plug-ins to use.	2.15	<button>卸载</button>
Trilead API Plugin	Trilead API Plugin provides the Trilead library to any dependent plugins in an easily update-able manner.	1.0.13	<button>卸载</button>
Variant Plugin	This user-invisible library plugin allows other multi-modal plugins to behave differently depending on where they run.	1.4	<button>卸载</button>
Web for Blue Ocean	Blue Ocean core	1.24.6	<button>卸载</button>
Workspace Cleanup Plugin	This plugin deletes the project workspace when invoked.	0.39	<button>卸载</button>

Jenkins 中文社区 REST API

9、邮件发送

- 系统配置 === 配置管理员邮箱

系统管理员邮件地址

◦ 534096094@qq.com

- 配置邮件发送的认证权限信息

- 登录自己邮箱，开启POP3/SMTP邮件服务
- 获取到自己的授权码 rkffwdqdyprccbcf
- 配置并测试好邮件发送即可

- 邮件模板内容

```

1 emailext body: '''<!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" style="font-size: 11pt;
11      font-family: Tahoma, Arial, Helvetica, sans-serif">
12      <tr>
13        <br/>
14        各位同事，大家好，以下为${PROJECT_NAME}项目构建信息<br>
15        <td><font color="#CC0000">构建结果 - ${BUILD_STATUS}</font></td>
16    </tr>

```

```

17      <tr>
18          <td><br />
19          <b><font color="#0B610B">构建信息</font></b>
20          <hr size="2" width="100%" align="center" /></td>
21      </tr>
22      <tr>
23          <td>
24              <ul>
25                  <li>项目名称 : ${PROJECT_NAME}</li>
26                  <li>构建编号 : 第${BUILD_NUMBER}次构建</li>
27                  <li>触发原因: ${CAUSE}</li>
28                  <li>构建状态: ${BUILD_STATUS}</li>
29                  <li>构建日志: <a href="${BUILD_URL}console">${BUILD_URL}console</a></li>
30                      <li>构建 Url : <a href="${BUILD_URL}">${BUILD_URL}</a></li>
31
32                      <li>工作目录 : <a href="${PROJECT_URL}ws">${PROJECT_URL}ws</a></li>
33                      <li>项目 Url : <a href="${PROJECT_URL}">${PROJECT_URL}</a>
34                  </li>
35
36          <h4><font color="#0B610B">最近提交</font></h4>
37          <ul>
38              <hr size="2" width="100%" />
39              ${CHANGES_SINCE_LAST_SUCCESS, reverse=true, format="%c", changesFormat=<li>%d
[%a] %m</li>"}
40          </ul>
41  详细提交: <a href="${PROJECT_URL}changes">${PROJECT_URL}changes</a><br />
42
43          </td>
44      </tr>
45  </table>
46  </body>
47  </html>''' , subject: '${ENV, var="JOB_NAME"}-第${BUILD_NUMBER}次构建日志' , to:
'17512080612@163.com'
48
49
50 大家自己修改 to 的位置

```

实现 短信通知，钉钉通知，微信通知，xxxxx

3、高级使用

1、级联变量

参照插件介绍即可

1、基本选择 Active Choices Parameter

```
1 // groovy脚本
2 return ["hello", "world", "abc:selected"]
3
4
5 //fallback 定义错误返回
6 return "error"
7
```

2、引用级联 Active Choices Reactive Parameter

要指定之前参考参数

Referenced parameters 

selected_branch

```
1 //如下
2 if ( selected_branch.equals("hello") ) {
3     return ["1", "2"]
4 }
5 if( selected_branch.equals("world") ){
6     return ["11", "22"]
7 }else{
8     return ["111", "222"]
9 }
10 //也可以执行shell
11
12
```

2、分支选择

3、自定义远程触发

Generic Webhook Trigger：参照官网进行使用

三、 Jenkins On Docker

1

四、 Jenkins On Kubernetes

1