

# Jenkins 面试题 (Beta)

以下面试题，基于网络整理，和自己编辑。具体参考的文章，会在文末给出所有的链接。

如果胖友有自己的疑问，欢迎在星球提问，我们一起整理吊吊的 Jenkins 面试题的大保健。

而题目的难度，尽量按照从容易到困难的顺序，逐步下去。

：实际上，面试中基本不会问 Jenkins 。因为，也没啥好问的。所以，本文胖友可以作为啥呢？我也不造，就当简单过过一些知识点吧。

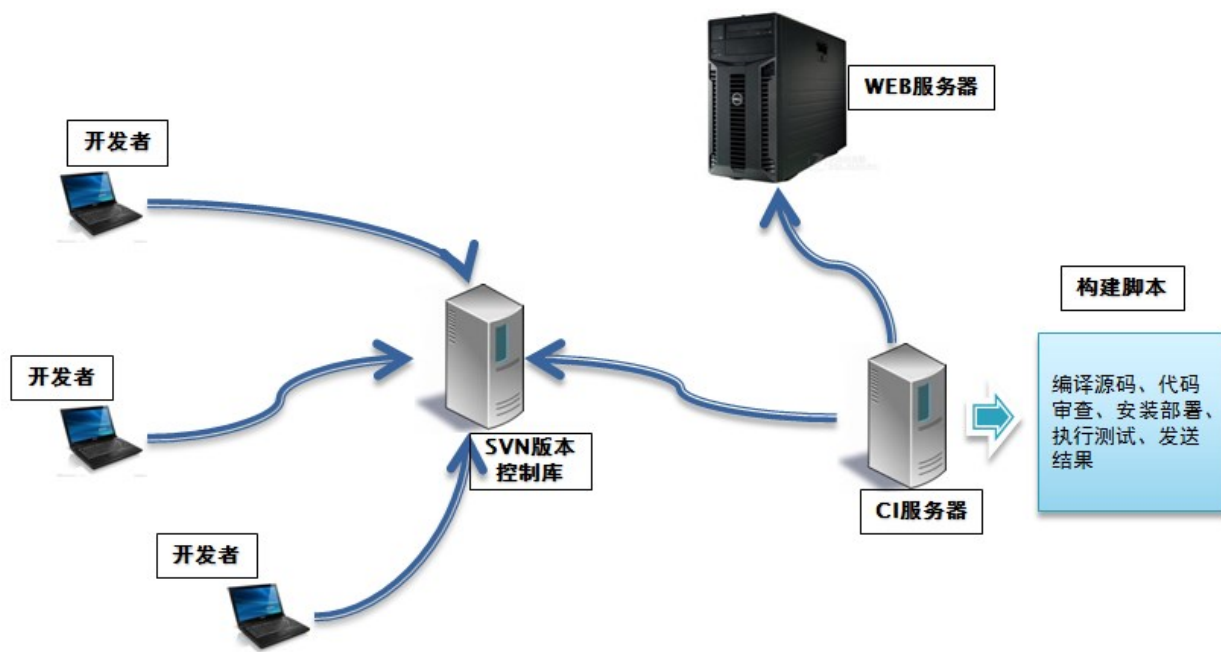
## 持续集成是什么？

持续集成，源于极限编程(XP)，是一种软件实践，软件开发过程中集成步骤是一个漫长并且无法预测的过程。集成过程中可能会爆发大量的问题，因此集成过程需要尽可能小而多，实际上持续集成讲的是不断的去做软件的集成工作。

### 🔗 持续集成有什么作用？

- 场景一、某项目最后做模块集成的时候，发现很多接口都不通，甚至有的模块连安装包都没有。
- 场景二、没有可用的软件包，需要人手动去编译打包最新的代码。
- 场景三、搭建测试环境的时候，需要手动去解压包，然后一系列拷贝修改配置等等。
- 场景四、团队成员或者 teamleader 想了解当前项目的状态，该如何去展示这些信息。

持续集成就是用来解决以上问题，它的价值主要在于减少重复的步骤，降低项目的风险，任何时间任何地点生成可用的软件，增强项目的可见性等。



：胖友，有没发现前后端分离之后，集成越来越是一个问题，特别是项目越到后期，越多问题。尽早集成，即使前期进度可能会略有滞后，大家需要经常加班。但是呢，前紧后松，一定能让项目更加可控。

### 🔧 持续集成怎么做？

持续集成，最简单的形式是包括一个监控版本控制（SVN、Git 等等）变化的工具。当变化被发觉时，这个工具可以自动的编译并测试你的应用。

当然，目前更多的是，使用 Jenkins 来实现持续集成。

：我记得我 11 年工作的时候，公司内部做了一个简单的发布系统。这个系统会告诉我们哪些类发生了变化，然后我们选择哪几个类需要发布到测试或生产环境上。😓 实在简陋，而且常常会少发，痛苦不堪。感谢 Jenkins，嘿嘿。

### 🔧 持续集成有哪些良好的实践？

- 维护一个单一的代码库
- 使构建自动化
- 使构建自测试

自测试，相对来说难落地，主要原因：

- 大多数公司，开发很少写单元测试。
- 大多数公司，没有自动化测试工程师。

- 每人每天都向主线提交代码

因为采用 GitFlow 工作流，所以不能向 master 分支提交代码，更多的是，向主仓库对应的功能分支提交。

如果真的要向 master 提交，需要配合 [特性开关](#)。

- 每次提交都应在集成机上进行构建
- 快速构建
- 使任何人都能轻易获得可执行文件
- 人人都能看到正在发生什么
- 自动化部署

## 简单介绍 Jenkins 是什么？

- 持续集成是一种实践，而 Jenkins 可以帮助团队去尽量好的去完成这种实践。

即，Jenkins 是一个持续集成的工具。

- Jenkins 是基于 Java 语言的开源持续集成工具，提供了一套非常易用的用户界面，用以自动化构建、测试、部署等功能。
- Jenkins 类似于 Eclipse，基于插件化的架构，方便功能的扩展，目前有几个百个现成插件可以使用，这些插件涵盖从版本控制、构建工具、代码质量、构建通知、集成外部系统、UI 定制、游戏等等各个方面。

😺 只要是个工具，基本是 **插件化** 的架构。

历史小故事：

伴随着 Jenkins，有时人们还可能看到它与 Hudson 关联。Hudson 是由 Sun Microsystems 开发的一个非常流行的开源，基于 Java 的持续集成工具，后来被 Oracle 收购。Sun 被 Oracle 收购之后，一个从 Hudson 源代码的分支由 Jenkins 创建出台。

## Jenkins 你都用了哪些插件？

- 最常用
  - [SSH Plugin](#)：这个可以登陆远程服务器，然后上面执行脚本。
  - [Role Strategy Plugin](#)：用来精细化管理权限，基于角色维度。
  - [Git plugin](#)：对 Git 的支持。
  - [SCM](#)：除 CVS 和 Subversion 外，需要实现与源代码控制系统支持的插件。
- 根据情况
  - Triggers：事件监听并触发构建的插件。例如，URL 改变触发器将监控一个 URL，当地址内容发生改变，这个触发器就将执行一次作业。
  - Build tools：实现额外构建工具的插件，如 MSBuild 和 Rake。如果您想在 Hudson 中构建非 Java 的软件时这些就特别有用。
  - Build wrappers：通常涉及时执行在受控制的构建过程本身之前和之后事件的插件。例如，VMware 插件将在构建之前启动一个客户虚拟机，建立和然后在构建完成后关闭它。这在您可能需要访问 VM 以执行单元测试的情况下是非常有用的。

## Jenkins 如何实现发布和回滚？

- 发布：Jenkins 配置好代码路径(SVN 或 Git)，然后拉代码，打tag。需要编译就编译，编译之后推送到发布服务器（Jenkins 里面可以调脚本），然后从分发服务器往下分发到业务服务器上。
- 回滚：按照版本号到发布服务器找到对应的版本推送。

一定要打 Tag 噢，不然回滚会比较麻烦。

## Jenkins 怎么做备份与恢复？

目前有三种方式：

- 1、使用插件备份。
  - 例如 ThinBackup 插件。
- 2、使用 Rsync 异地备份。
- 3、使用版本控制工具进行备份。

具体的，可以参考文章 [《Jenkins 系列：（五）Jenkins 数据备份与恢复》](#)。

相对来说，比较推荐使用 ThinBackup 插件，简单方便，能够满足需求。

## Jenkins 如何删除历史构建数据？

目前来说，有三种方式：

- 1、手工删除构建记录。
- 2、转移磁盘空间。
- 3、自动丢弃构建历史数据。

具体的，可以参考文章 [《Jenkins 服务器磁盘空间管理策略》](#)。

如果说，我们希望多保持一些构建历史数据，那么可以设置较大的“发布包最大保留”，同时我们需要提供相对大的磁盘空间。当然，即使再大的磁盘空间，也可能被撑爆，所以出问题时，我们可以手工删除构建记录。也就会说，三种方式，一起配合。

另外，Jenkins 的日志文件也挺占用内存的，可以参考 [《Linux 中 Jenkins 日志记录占满磁盘问题》](#) 文章，进行处理。

## 彩蛋

---

发现，真的没什么好写的。当然，如果没有自己配置过 Jenkins 的胖友，建议尝试自己搭建一次，然后至少部署几个 Spring Boot 的项目，再然后，思考下需要划分几套环境，每套环境的定义。一般来说：

```
Feature 环境 => 测试环境 => 预发布环境 => 生产环境
```

参考与推荐如下文章：

- [《Jenkins 面试题》](#)
- [《持续集成以及 Jenkins 的知识介绍》](#)
- [《如何在 CentOS 下安装部署 Jenkins 持续集成环境》](#)