

# 基于Jenkins的DevOps实战

【入门实战篇】

讲师：赵班长



赵舜东

- 花名：“赵班长”，曾在武警某部负责指挥自动化的架构和运维工作，2008年退役后一直从事互联网运维工作，历任运维工程师、运维经理、运维架构师、运维总监，第四届北大互联网CIO班副班长。
- 中国SaltStack用户组发起人（<http://www.saltstack.cn/>）
- 运维社区创始人（<http://www.unixhot.com/>）
- DevOps学院创始人（<http://www.devopsedu.com/>）
- 著作：《SaltStack入门与实践》、《运维知识体系》、《缓存知识体系》
- 中国首批Exin DevOps Master 和Exin DevOps Professional认证讲师
- GOPS全球运维大会金牌讲师、金牌作者，SACC系统架构师大会专家顾问

#### 自动化运维工程师精英班【端到端自动化运维课程体系】

##### 课程特色

业内专家授课

生产案例实战

完整自动化体系

全程在线辅导

##### 1. 自动化安装

基于Cobbler的自动化系统安装实战



##### 2. 自动化配置管理

基于SaltStack的系统配置管理



##### 3. 自动化监控

基于Zabbix构建自动化多维监控体系



##### 4. 端到端持续交付流水线

全开源端到端持续交付流水线



##### 5. 自动化日志收集

基于ELKStack构建海量日志平台





1

持续交付之Jenkins实战

2

Jenkins Pipeline实战

3

Jenkins分布式构建

4

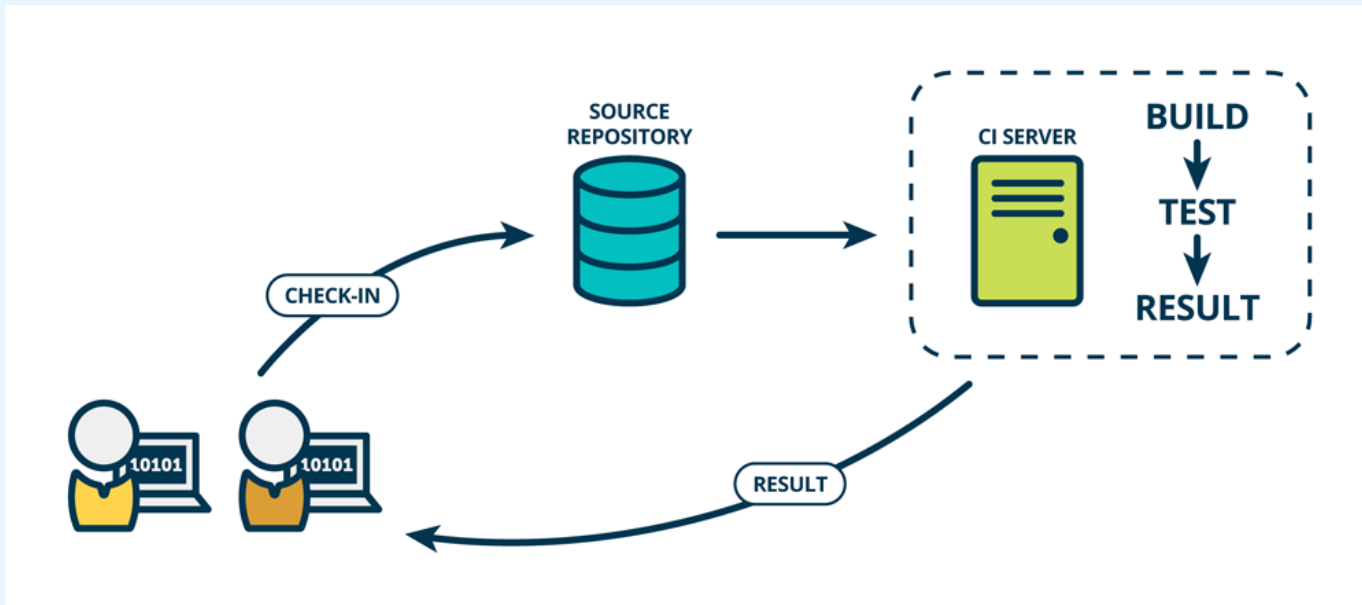
质量扫描SonarQube实战

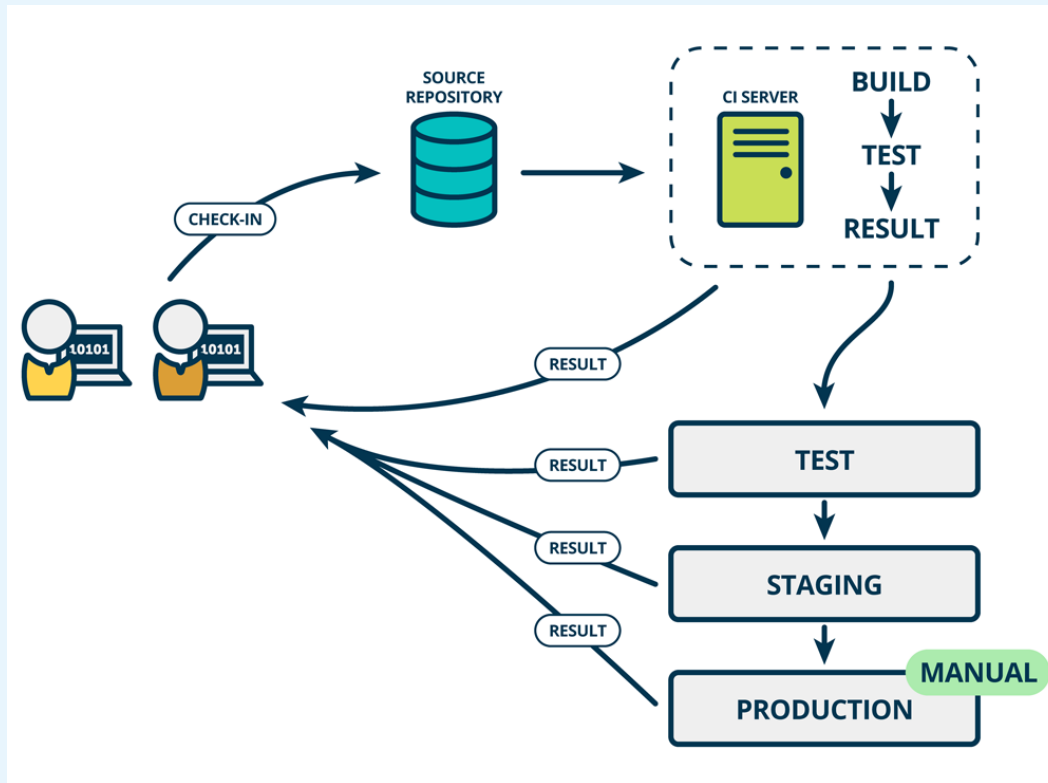
5

SonarQube和Jenkins集成

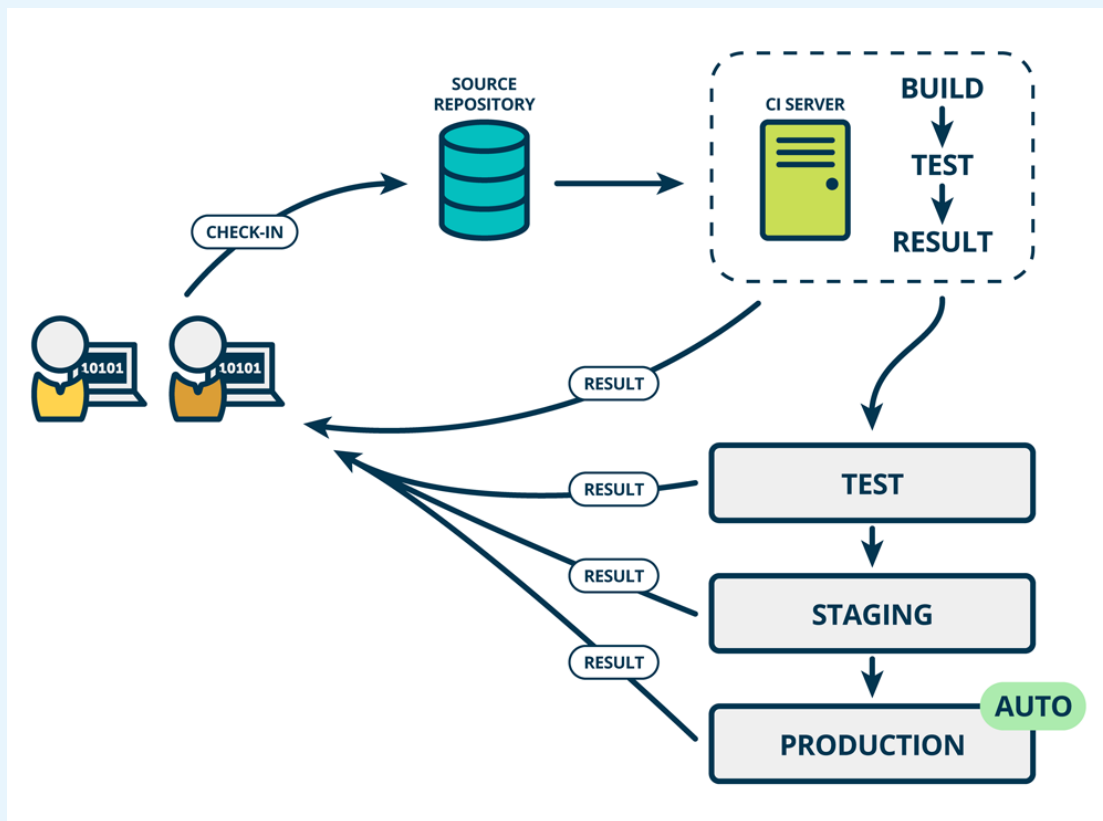
6

Gitlab和Jenkins集成

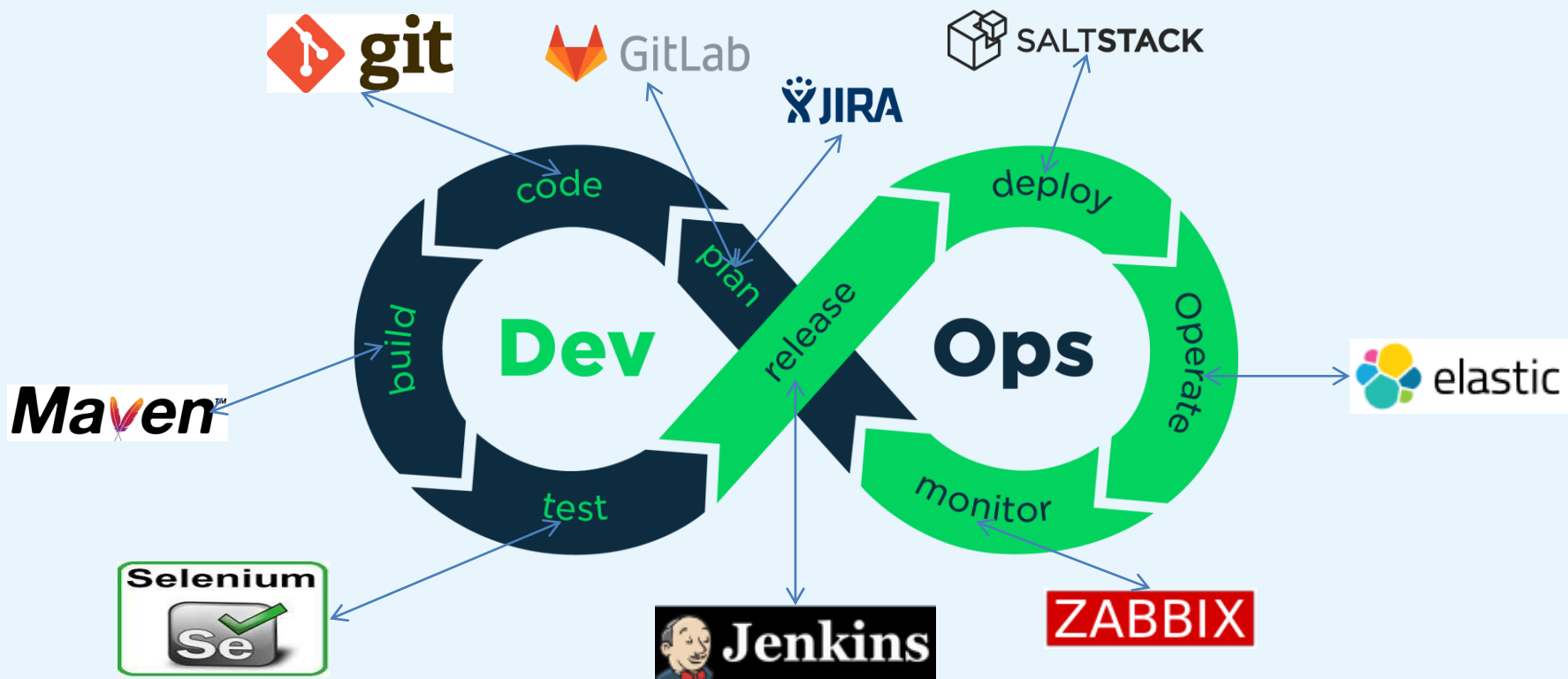




持续交付



持续部署





1 持续交付之Jenkins实战



2 Jenkins Pipeline实战

3 Jenkins分布式构建

4 质量扫描SonarQube实战

5 SonarQube和Jenkins集成

6 Gitlab和Jenkins集成

- Pipeline，简而言之，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流程**编排与可视化**。
- Pipeline 是Jenkins2.X最核心的特性，帮助Jenkins实现从CI到CD与DevOps的转变
- <https://jenkins.io/2.0/>

- Jenkins Pipeline是一组**插件**，让Jenkins可以**实现持续交付管道**的落地和实施。
- 持续交付管道(CD Pipeline)是将软件从**版本控制阶段**到**交付给用户或客户的完整过程的自动化表现**。
  - 软件的每一次更改（提交到源代码管理系统）都要经过一个复杂的过程才能被发布。
- Pipeline提供了一组可扩展的工具，通过Pipeline Domain Specific Language (DSL) syntax可以达到Pipeline as Code的目的
- Pipeline as Code : Jenkinsfile 存储在项目的源代码库

- **代码**：Pipeline以代码的形式实现，通常被检入源代码控制，使团队能够编辑，审查和迭代其CD流程。
- **可持续性**：Jenkins重启或者中断后都不会影响Pipeline Job。
- **停顿**：Pipeline可以选择停止并等待人工输入或批准，然后再继续Pipeline运行。
- **多功能**：Pipeline支持现实世界的复杂CD要求，包括fork/join子进程，循环和并行执行工作的能力。
- **可扩展**：Pipeline插件支持其DSL的自定义扩展以及与其他插件集成的多个选项。

- **Stage**

- **阶段**，一个Pipeline可以划分为若干个Stage，每个Stage代表一组操作，例如："Build", "Test", "Deploy"。
- 注意，Stage是一个逻辑分组的概念，可以跨多个Node。

- **Node**

- **节点**，一个Node就是一个Jenkins节点，或者是Master，或者是Agent，是执行Step的具体运行环境。

- **Step**

- **步骤**，Step是最基本的操作单元，小到创建一个目录，大到构建一个Docker镜像，由各类Jenkins Plugin提供，例如：`sh 'make'`

- Pipeline脚本是由Groovy语言实现
  - **无需专门学习Groovy**
- Pipeline支持两种语法
  - Declarative 声明式 ( 在Pipeline plugin 2.5中引入 )
  - Scripted Pipeline 脚本式
- 如何创建基本的Pipeline
  - 直接在Jenkins Web UI 网页界面中输入脚本
  - 通过创建一个Jenkinsfile可以检入项目的源代码管理库
- 最佳实践
  - 通常推荐在 Jenkins中直接从源代码控制(SCM)中载入Jenkinsfile Pipeline

1 持续交付之Jenkins实战

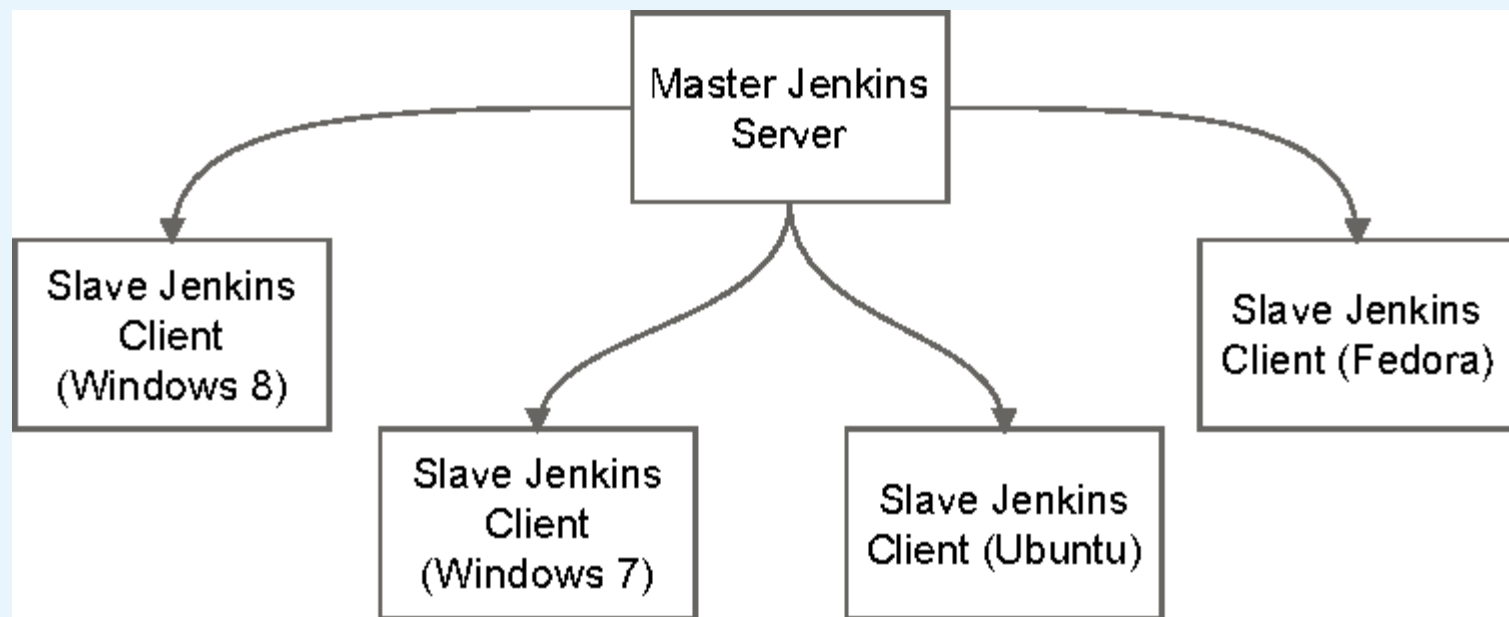
2 Jenkins Pipeline实战

➔ 3 Jenkins分布式构建

4 质量扫描SonarQube实战

5 SonarQube和Jenkins集成

6 Gitlab和Jenkins集成





1 持续交付之Jenkins实战

2 Jenkins Pipeline实战

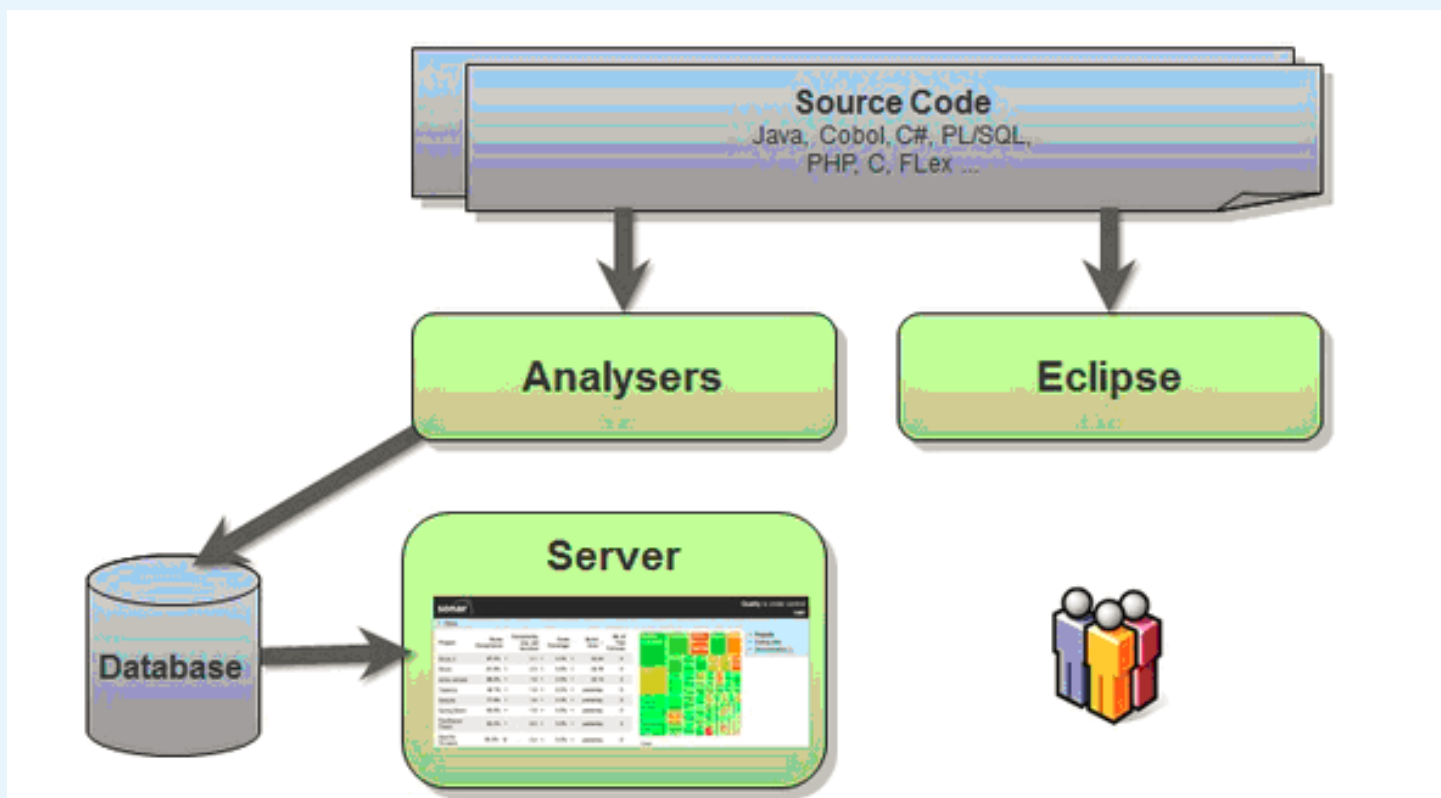
3 Jenkins分布式构建

➔ 4 质量扫描SonarQube实战

5 SonarQube和Jenkins集成

6 Gitlab和Jenkins集成

1. 编码规范：是否遵守了编码规范，遵循了最佳实践。
2. 潜在的BUG：可能在最坏情况下出现问题的代码，以及存在安全漏洞的代码。
3. 文档和注释：过少（缺少必要信息）、过多（没有信息量）、过时的文档或注释。
4. 重复代码：违反了Don't Repeat Yourself原则。
5. 复杂度：代码结构太复杂（如圈复杂度高），难以理解、测试和维护。
6. 测试覆盖率：编写单元测试，特别是针对复杂代码的测试覆盖是否足够。
7. 设计与架构：是否高内聚、低耦合，依赖最少。



1 持续交付之Jenkins实战

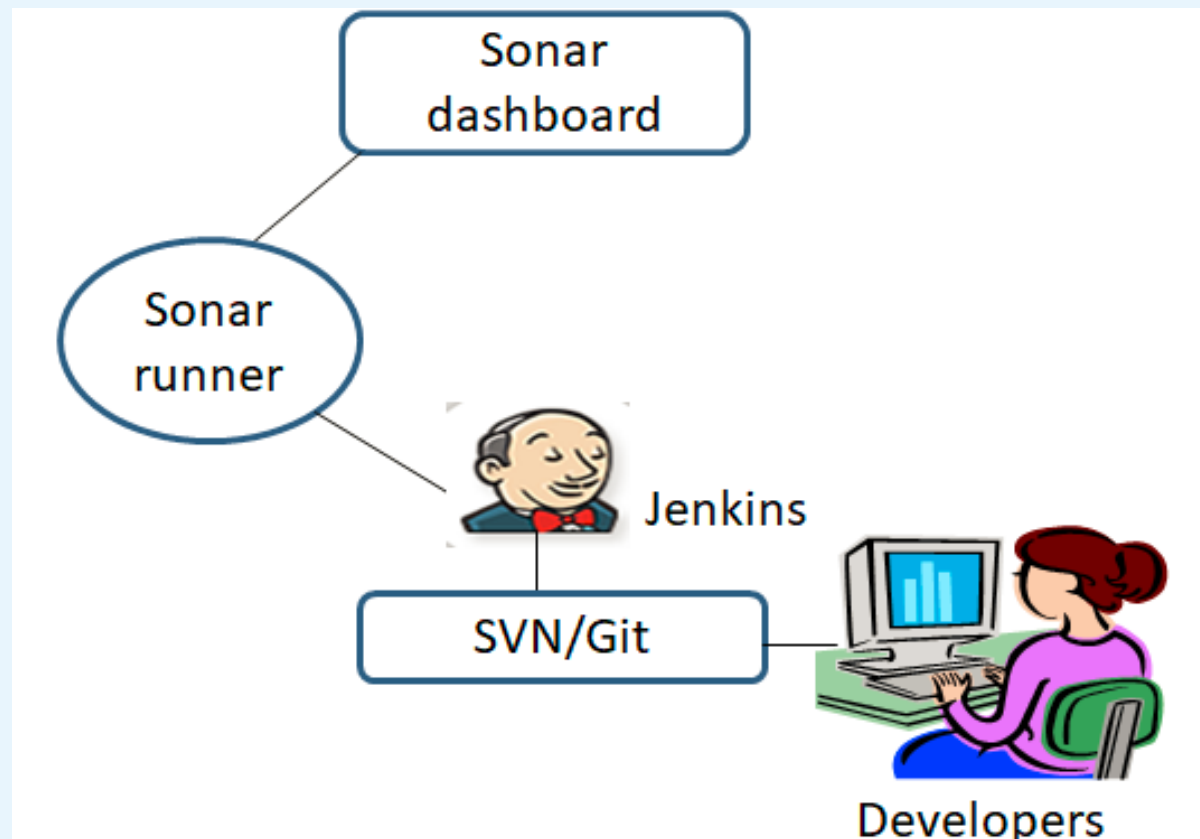
2 Jenkins Pipeline实战

3 Jenkins分布式构建

4 质量扫描SonarQube实战

➔ 5 SonarQube和Jenkins集成

6 Gitlab和Jenkins集成



1 持续交付之Jenkins实战

2 Jenkins Pipeline实战

3 Jenkins分布式构建

4 质量扫描SonarQube实战

5 SonarQube和Jenkins集成

➔ 6 Gitlab和Jenkins集成

# 开发Push代码后 自动触发流水线执行

运维社区



运维人的学习社区

速云科技



企业DevOps运维专家