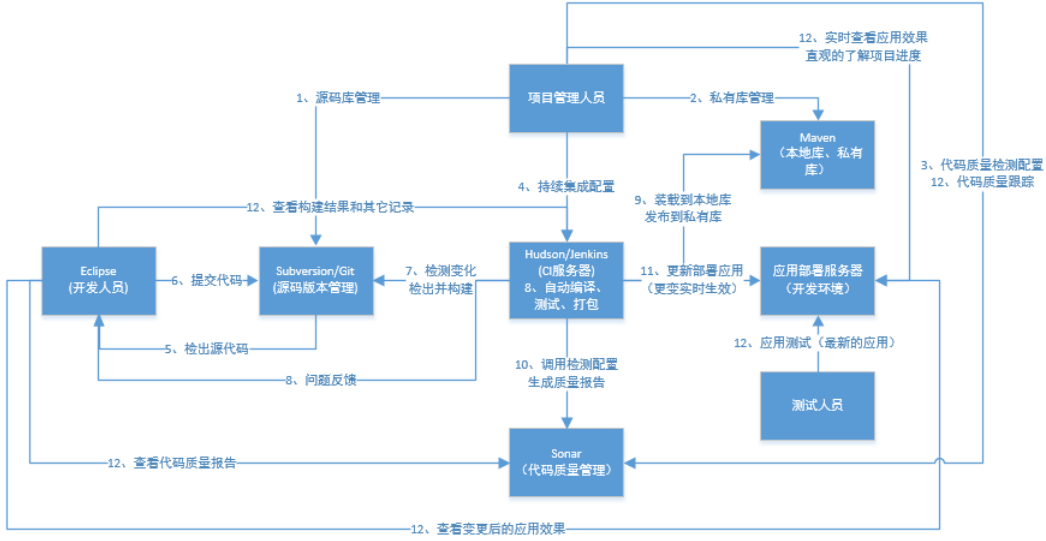
搭建敏捷高效的持续集成管理平台

1、持续集成是一种软件开发实践，团队开发成员经常需要集成他们的工作，每次集成都能方面高效的通过自动化的构建（包括自动化编译、测试、发布）来验证，从而尽快地发现集成错误。特别是在大规模的团队开发中，能让团队成员间进行高效的合作开发，同时高效的一键部署方式使得大量应用产生的不断的跟新部署变得简单,从而缩短开发周期提高了生产力。而且持续集成能与各种项目管理工具整合成一套使用灵活的开发管理平台，使得实时交付可用、可测试的软件产品成为可能。

2、持续集成管理平台的组成

持续集成平台由CI服务器加上一系列的软件开发管理工具以及插件一起组合形成。源码版本管理如：Subversion、Git，项目构建工具如：Maven、Ant，代码质量管理如：Sonar，持续集成引擎如：Hudson、Jenkins、Apache Continuum，应用部署环境如：操作系统、JDK、Tomcat、JBoss等等。



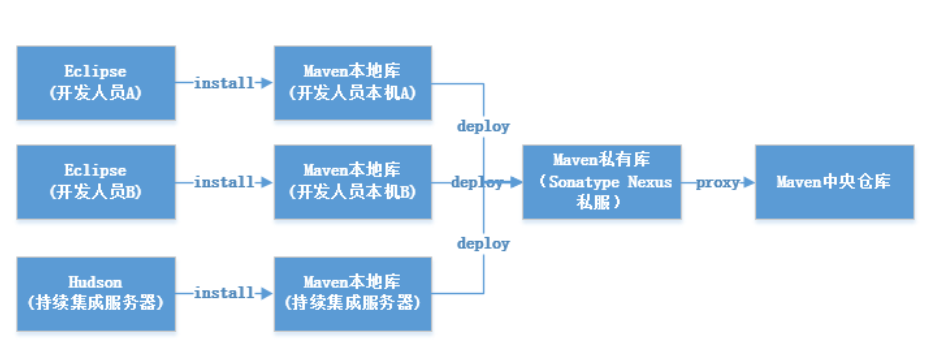
3、具体使用工具

3.1、Svn Admin

Svn admin是用java语言开发的用来管理svn服务器项目及其相关项目人员的web应用。首先安装好SVN的服务器端，可以采用多库模式，即用一份配置文件管理多个库，然后把Svn admin项目部署好，这样就可以在浏览器端很方便的管理项目和多个项目组，可依需求快速分配组成员和相应的操作权限。从此告别每次都要靠手工去服务器上修改配置文件。使用Svn admin优点还包括支持多种数据库(Mysql,Oracle,Sql Server)、多操作系统(windows,Linux)、安全，密码加密保存、支持多协议(svn协议、http协议)，在svn协议支持下，svn的配置信息都在仓库目录conf下的authz,passwd,svnserve.conf三个文件中，如果想要配置用户和权限的话都是通过修改passwd和authz文件内容并立刻生效的。其实Svn Admin的本质就是通过对这三个文件的修改进行管理的，而所有的成员、权限的相关数据都是保存在本地安装的数据库当中，所以当使用Svn Admin的浏览器页面进行相关的修改时，最终就是把信息输出配置文件和数据库当中。支持http协议，svn在安装完成后有两种配库方式，一种是多库方式，另一种是单库方式，每一个仓库需要单独配置各自的权限文件和密码。

3.2、Maven

1、maven工作原理图



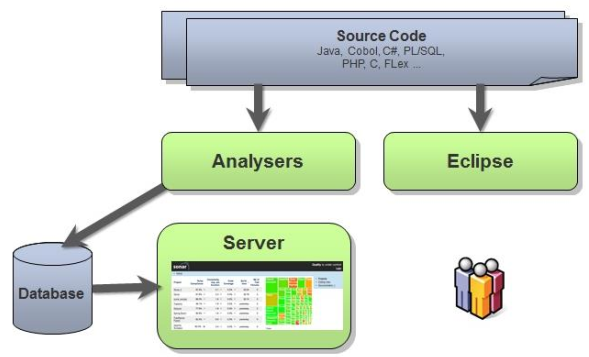
1. maven是目前公司项目开发所使用的项目构建主流工具，也是采用纯Java编写的开源项目管理工具。在持续集成平台上中，就是通过使用maven提供的一系列插件，对从版本库中拉下来的源代码进行编译，构建，打包，测试，发布。Maven采用了project object model（POM）概念来管理项目配置和依赖的包。Maven项目通过pom.xml文件配置的依赖包的坐标先去本地仓库寻找对应的包，如果本地没有，那么就会去Nexus私服去下载对应的包，如果私服也没有就会去远程中央仓库下载依赖，由私服下载好后再下载到本地Maven库供项目引用。

3.3、SonarQube

SonarQube是一个管理代码质量的开放平台,通过使用不同的插件算法来对项目开发人员上传到SonarQube服务器的代码进行检测和对其结果再加工处理，最终以量化可视化的方式如图表来展示代码质量，从而实现对各种规模和种类的工程项目进行代码质量的管理和分析，实现了对代码的质量的全面自动化分析和管理。Sonarqube可以从七个维度对代码质量进行检测。

1. 复杂度分布： 代码复杂度过高将难以理解、维护
2. 重复代码：程序中所包含大量重复的代码
3. 单元测试：统计并得到单元测试覆盖率
4. 编码规范：通过使用Findbugs、PMD、CheckStyle等安装的插件规范代码
5. 注释：代码的重要组成部分，增强程序的可读性和降低程序维护的难度
6. 潜在的Bug：通过使用Findbugs、PMD、CheckStyle等安装的插件检测bug
7. 结构与设计：程序之间的依赖、耦合

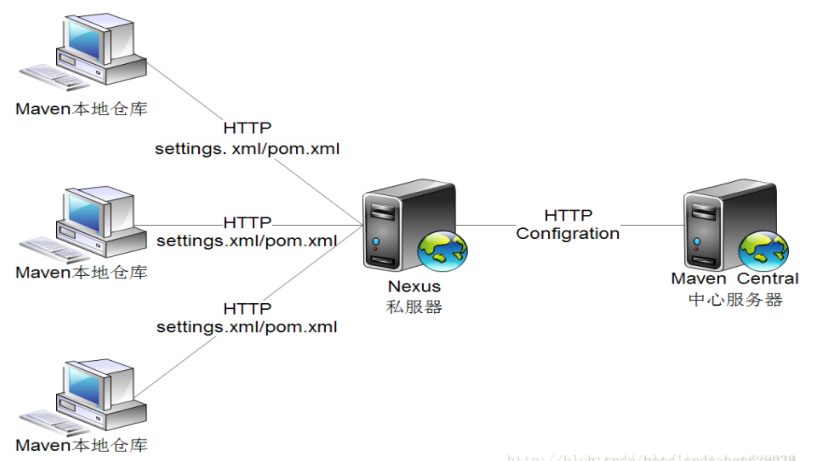
SonarQube的平台由数据库、web服务、分析器组成，架构图如下：



其中，数据库用于存放SonarQube的各种配置数据以及代码质量的快照，web服务用于方便查看SonarQube的配置数据、代码质量的快照数据，对项目代码进行分析，生成质量结果数据并存入数据库中。

3.4、Nexus私服

上面简单的介绍了maven的作用，maven提供了本地仓库，项目可以很容易从其中引用所需的jar包。如果项目所需的jar包maven本地仓库中没有的话，maven默认连接的中央仓库是Apache提供的远程仓库，这对于企业开发是非常不合理的。而Nexus可以部署在局域网中扮演中央仓库的角色，maven只需去Nexus私服中去下载所需的jar包，nuxus可以在项目前期准备时期就连接中央服务器下载项目的所需的jar包或项目开发间隙下载所需依赖包。这样一来不紧节省了网络带宽并缩短了项目搭建所需的周期，而且项目团队可以利用nexus私服方便的集中管理自己项目所需的全部依赖文件和项目部署包，包括一些重要的自己上传、部署的第三方依赖包和自己的开发项目。因此Nexus使得maven变得更强大，是企业项目开发不可缺少的开发软件。其组成如下图：



3.5、Hudson

Hudson是一款基于java开发的开源的持续集成引擎，其被设计用于项目的一键部署（自动化的编译、分析、打包、发布、部署项目）。使得项目开发人员从低效、繁琐的手工部署的工作中解脱出来，提供开发效率。尤其对于开发模块的众多的大型项目来说，其优势凸显，再加上Hudson支持分布式构建，多台计算机一起构建、测试使得分工开发更高效。如果要想搭建一套完整的持续集成平台，首先就要在服务器上安装、部署Hudson，再在Hudson平台上按照自己的需求整合上面提到的一系列软件SVN、Maven、Sonar、Nexus即可。Hudson提供了友好的web界面方面项目开发人员使用。开发人员在web界面进行插件安装、安全配置、账号注册、系统设置、添加配置项目以及项目的一键全自动部署。其大致流程如下：

从SVN中拉取提交代码

使用集成的maven插件编译、打包源码

从Hudson的jobs目录下拷贝对应的部署文件到对应的服务器目录

SSH连接部署项目的服务器

执行.sh启动脚本文件

显示项目构建、最终执行部署结构

Hudson提供大量的插件供开发人员安装使用，开发人员可以根据自己的需求选择安装、配置使用。