



**运用 Jenkins 前后对比**

|  |  |
| --- | --- |
| 运用 Jenkins 前 | 运用 Jenkins 后 |
| 构建并测试了整个源代码。在构建和测试失败的情况下查找和修复错误既困难又耗时，这反过来又减慢了软件交付过程的速度。 | 在源代码中进行的每个提交都将被构建和测试。因此，开发人员无需检查整个源代码，而只需要专注于特定的提交即可。可频繁发布新软件。 |
| 开发人员必须等待测试结果 | 开发人员知道源代码中每次提交的测试结果。 |
| 整个过程是手工的 | 只需要提交对源代码的更改，Jenkins 将为您自动完成其余过程。 |

3.部署方式：

3.1 jenkins触发式构建：用于开发环境部署，开发人员push代码或者合并代码到gitlab项目的master分支，jenkins就部署代码到对应服务器。

3.2 jenkins参数化构建：用于测试环境预上线环境部署，开发push代码或者合并代码到gitlab项目的master分支之后，并不会部署代码，而是需要登录到jenkins的web界面，点击构建按钮，传入对应的参数（比如参数需要构建的tag，需要部署的分支）然后才会部署。

3.3 jenkins定时构建：用于APP自动打包，定时构建是在参数化构建的基础上添加的，开发人员可以登录jenkins手动传入tag进行打包，如果不手动打包，那么jenkins就每天凌晨从gitlab拉取最新的APP代码打包。



