[关于不停机部署方案的选择](https://www.cnblogs.com/wangliaohan/p/10438840.html)

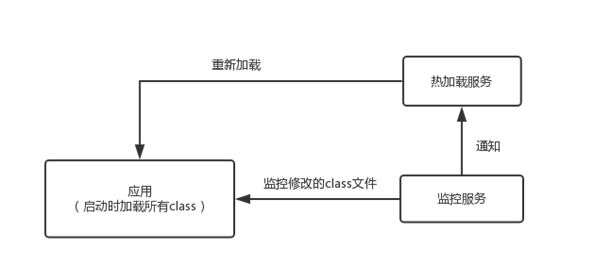
概述：鄙人才疏学浅，了解到不停机部署方案有热加载、热部署、集群分批部署三种方式，这里只针对这三种方式进行讨论。

一、热加载

**原理：**热加载的实现原理主要依赖jvm的类加载机制，在运行时对被修改过的类进行重新载入。

**实现方式：**在容器启动的时候起一条后台线程，定时的检测类文件的时间戳变化，如果类的时间戳变掉了，则将类重新载入。

**适用场景：**热加载会直接修改jvm中的字节码，这种方式对既有代码行为的影响难以控制，风险较大，几乎不会在生产环境使用该方案。



 二、热部署

**原理：**热部署是指在服务器运行时重新部署项目，它是直接重新加载整个应用，这种方式会释放内存，比热加载更加干净彻底，但同时也更费时间。

**实现方式：**Weblogic、tomcat都可以通过配置支持热部署，但主要用于开发测试环境，较少应用于生产环境。

**适用场景：**热部署大大简化了发布操作，在开发和测试环境使用很普遍；但由于开通热部署后会增大系统开销，因此较少在生产环境使用。

三、集群分批部署

**原理：**利用nginx、apache等代理服务器自动对后端服务器的有效性进行判断，转发请求到有效的服务器上。在此基础上对后端服务器先后逐个更新，达到不停机部署的目的。

**实现方式：**在集群的前端代理服务器上配置有效性检测和自动剔除失效的后端服务器。

**适用场景：**适用于多台后端服务器的集群环境，该方案普遍应用于生产环境。

方案对比：

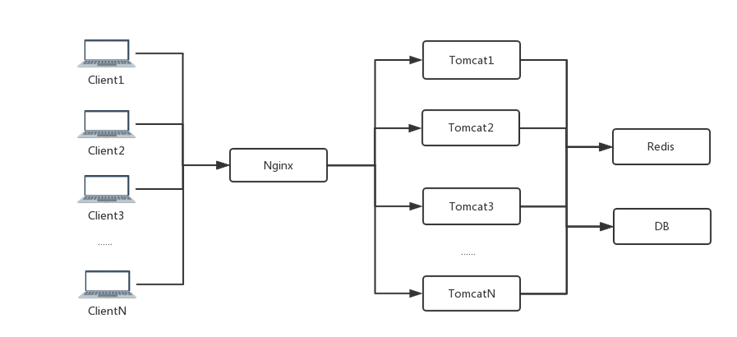
|  |  |  |  |
| --- | --- | --- | --- |
| 方案名称 | 优点 | 缺点 | 适用的环境 |
| 热加载 | 1. 每次只重新加载变化的文件，发布速度较快。 | 1．  直接修改jvm中的字节码，风险较高。  2．  需要开发监控和加载服务。 | 一般用于重启速度较慢的测试环境，基本不会应用于生产环境。 |
| 热部署 | 1. 中间件支持，风险相对较低。 2. 内存释放重建，更加干净彻底。 3. 重新加载所有的文件，耗时较长。 4. 增大了中间件的系统开销。 |  | 一般用于开发和测试环境，较少应用于生产环境。 |
| 集群分批部署 | 1. 服务不中断，用户无感知。 2. 无法支持一个后端服务器的情况。 3. 手动操作，相对繁琐 |  | 普遍应用于生产环境 |

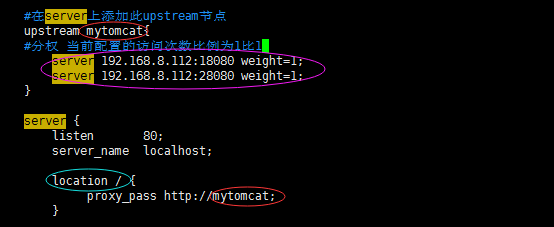
方案选择：

　　基于安全方面的考虑，我们建议选择“集群分批部署”的方案。

集群分批部署的原型

　　架构图如下：



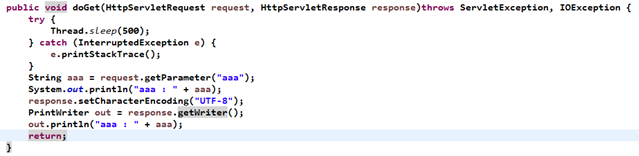
1. Nginx上配置上游服务器，示例如下：
2. 

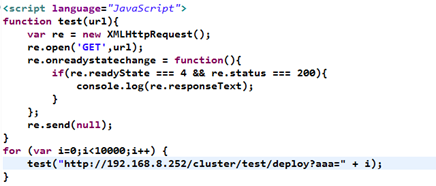
原型验证

1. 目的：验证并发访问时，剔除或停止某个应用服务器，请求会不会丢失。
2. 过程：

　　　　a)         编写应用程序，应用的服务端即时打印接收到的请求，并返回给前端jsp页面，前端jsp页面收到后也打印出请求信息。

　　　　b)         关键代码：

　　　　c)        

　　　　d)        

　　　　e)         部署该应用到两个tomcat：192.168.8.112:1880 和 192.168.8.112:28080。

　　　　f)          模拟3个用户分别不间断地发送1万个请求给服务器。

　　　　g)         在请求发送的过程中，修改nginx的配置文件ngxin.conf，剔除其中一个服务器，然后重新加载配置文件。

　　　　h)         在请求发送的过程中，对其中一个tomcat进行关闭，1分钟后重启。

　　　　i)          统计服务端的接收到的请求总数，统计值为30000。

　　　　j)          统计前端jsp页面接收到请求总数，统计值为30000。

　　3.结论：在集群运行过程中，主动剔除，或直接停止某个应用节点做发布，请求不会丢失。