

# 高可用及动态扩缩容agent的jenkins CI构建系统

## 项目背景

在当前的CI/CD系统中，各个项目基本都是使用jenkins来做构建，CI/CD系统通过对接不同项目的jenkins，来触发构建行为。随着接入的项目越来越多，遇到了如下的问题：

- jenkins难以管理。不同项目有不同的账号和不同的构建机器和环境。
- 不同项目的jenkins使用者对jenkins的理解能力不同，往往只是使用一个自由风格，并且是基于shell脚本的方式来编写CI过程。在接入的时候，需要查看具体的shell脚本，还要适配CI/CD系统来进行改造，增加了接入成本。
- 每个项目都要部署一台jenkins服务器，如果构建频繁，还需要部署构建节点，增加了服务器的开支，并且，部署节点的构建环境初始化还需要花费研发人员的时间。

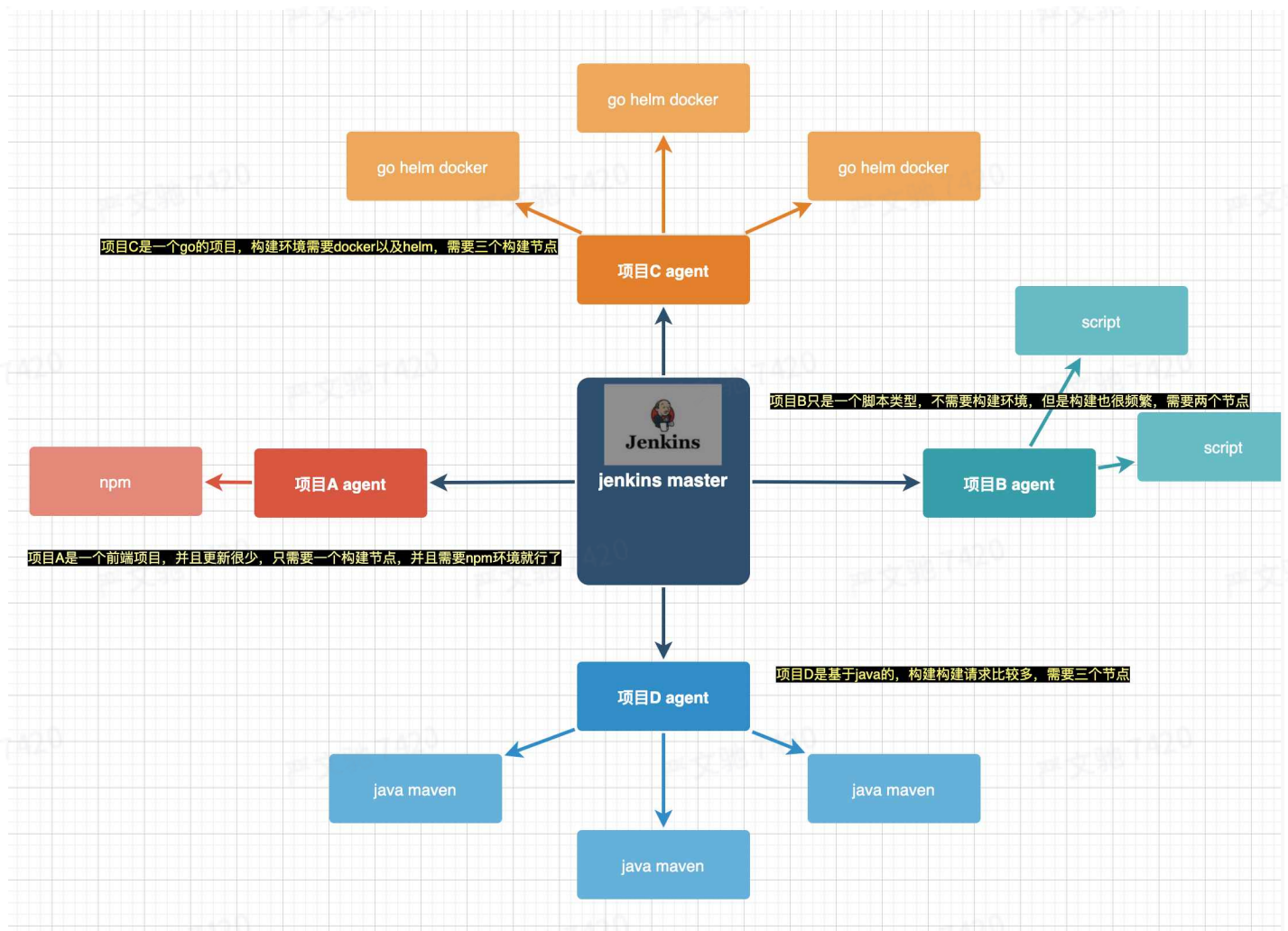
基于当前公司各个项目的CI情况，CI/CD系统在接入项目时，会使用统一的jenkins。这样，项目研发人员不必关心jenkins的构建环境和构建机器问题，CI/CD系统直接和研发人员对接构建方式，节省双方时间。

但是，这样又会引入以下问题：

- 构建节点的构建环境。不同的项目和开发语言往往需要不同的构建环境，例如npm，java，go，python，他们都有自己的构建环境要求。
- 频繁的构建会造成jenkins master压力过大，触发构建以后，还需要等待之前的还没有完成的构建。造成构建时间过长。
- 每次接入一个项目，都要准备相应的构建节点，有一些项目构建频率很高，可能需要更多的构建节点；而有一些项目构建频率低，往往只需要一个构建节点，造成了难以评估不同的项目需要配置多少个构建节点。
- Jenkins master是个单节点，节点宕机以后，会影响所有的构建。

之前的构建架构是这样的





## 项目目的

CI/CD系统需要确保CI的高可用以及构建环境的管理，要求能做到：

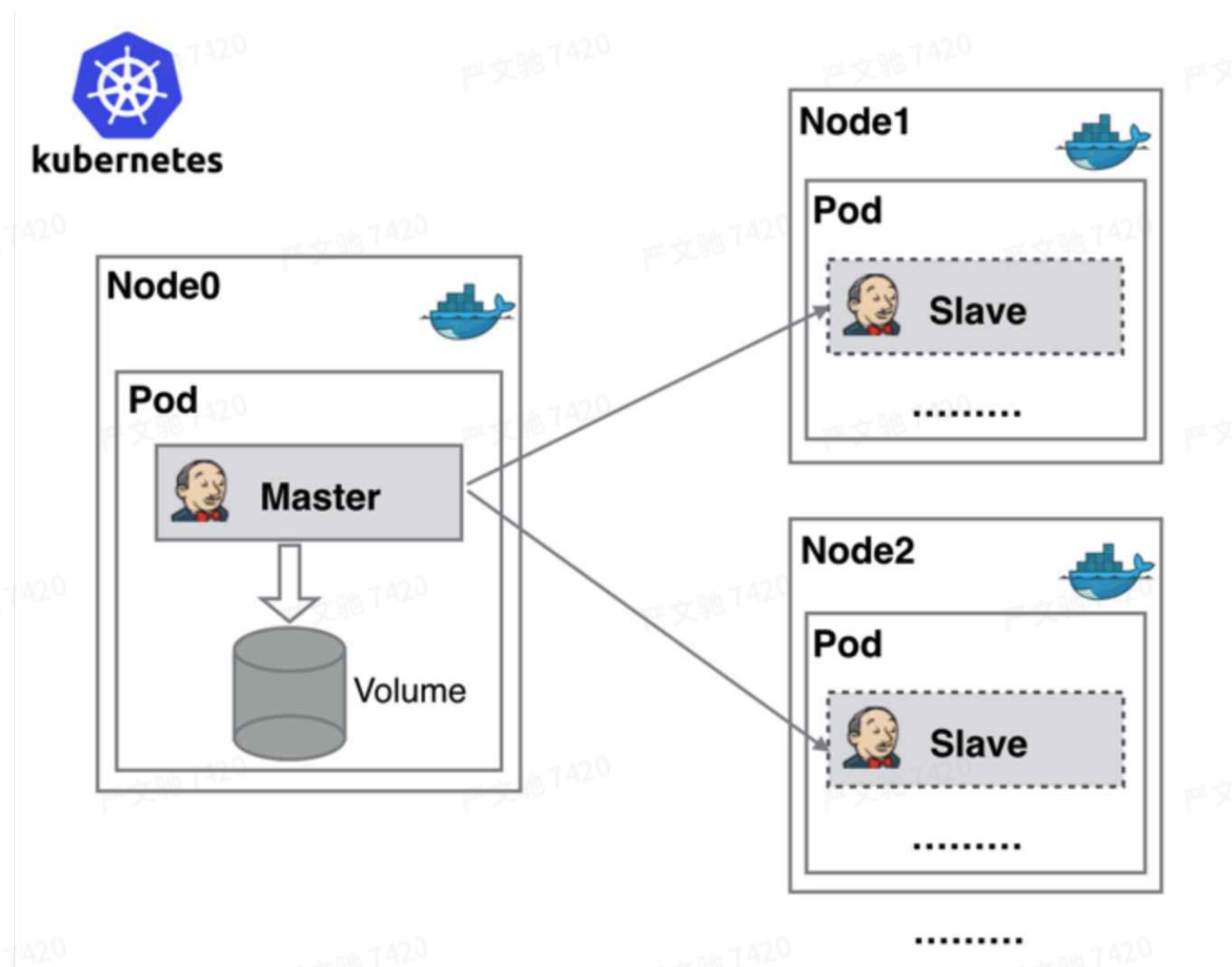
- Jenkins master节点挂了以后，还有其他节点能马上顶上立即可用。
- 构建环境的管理问题。
- 构建节点的管理问题。
- pipeline即代码。流水线也是CI/CD代码的一部分，溯源流水线的创造者，找出有问题的构建。

## 项目实现

- 在k8s集群中部署jenkins高可用，通过gfs持久化存储保证一个jenkins pod挂了或者迁移到其他的节点以后，能继续使用之前的数据，**实现了高可用**。
- 通过docker来管理构建环境的问题，和研发沟通需要项目所需的构建环境，生成一个构建镜像，上传到harbor，**免去了构建环境的部署**。

- 通过jenkins的k8s插件，实现构建节点的动态生成。每次有了一个构建请求，就会在k8s集群中生成配置好的pod，这个pod中包含了项目所需要的构建docker镜像。构建完成后，k8s集群会删除这个pod。因此，不需要额外的agent节点来配置好构建环境，**节省了服务器资源**。
- 通过把pipeline放在gitlab中，每个项目的CI构建，都能通过代码体现，**省去了配置jenkins**。（配置一个项目，需要在jenkins上配置的太多了，导致在测试环境使用同样的配置，需要点击页面很多次）。

改造后的jenkins架构是这样的



## 项目周期

从调研k8s部署jenkins高可用以及动态扩缩容agent花了两周时间。

## 项目效果

当前已经把tt 运营后台的前端npm构建迁移到了新的架构上，已经无缝对接，节省了构建时间和构建服务器的资源。

后期计划把声洞运营活动cdn发布，大数据开发平台的脚本，声洞脚本，这些纯文件形式的构建迁移到新的架构中，节省构建服务器资源。

## 项目影响范围

该项目适用于所有需要对接到CI/CD的项目。对于后续新项目的接入，优先使用这个构建来触发构建，提高构建效率，节省构建资源。