# 简介

## 为什么资源划分，为什么CI

在自动化测试中，尤其是在有限的资源情况，合理的利用资源，保障资源的能动性，并且不受其他资源的影响，保持独立性。 在持续集成中，我们不需要关心什么时候来处理资源，不需要关心被搁置的任务列表到底该怎么继续执行，不需要关心手机是否需要充电，这一切都由交由我们的的管家Jenkins来全方位的管理，我们只需要在必要的时候告诉我们的管家， 我们新到的资源是否归类于某一个项目，我们的某一个资源是不是由于服役期限到了，或者使用寿命到了，或者其他原因必须移除掉了。

## 为什么可以实现CI

也可以说，在兼容性测试的案例中，能够将资源划分的概念引入的CI的一个重要的原因就是CI的[Node and Label parameter plugin](https://wiki.jenkins-ci.org/display/JENKINS/NodeLabel+Parameter+Plugin) 的引入，解决了在CI中Node不能参数化的问题。

# 工作流程

## 准备工作

1. 创建对应PC的基准Node, 并且配置这台PC可能会用到的一些环境，例如Gradle的其他版本，Java的其他版本，以便在环境变量之外的可控性，并且更新PC的信息到数据库的pcInfo表内，并且添加这个Node到HandleDevice的job里面，使job可以在这台PC上面进行操作。
2. 如果需要接入到特定的项目，例如兼容性测试。找到挂入到对应PC的设备的设备编号，通过ADB命令的方式获取，然后写入到projectMapping表内，格式为{"project" : "monkey", "deviceID" : "HZ703FZC0G5O00O"}。
3. 接入设备的对应的PC，并确保adb下面可以识别到设备。

## 执行过程

1. HandleDevice会间隔10分钟定时触发，在所有配置的Node上面（也就是PC上面）自动扫描当前的设备接入情况。
2. 然后会根据HandleDevice.ddd的流程图上面所显示的流程执行下去。
3. 如果有新接入的设备，会邮件通知是否当前设备还没有来的接加入到特定的项目内。

## 服务端的操作

1. Jenkins服务端：创建Node, 删除Node, Offline Node, Online Node, Chang TCP port。
2. Nginx服务端：端口转发。
3. 数据库服务端：获取信息，插入信息， 更新信。.

## 客户端操作

1. PC端：创建定时任务，过时强制启动任务。
2. 设备端：获取deviceID, PlatformInfo, ProductInfo, Battery Level

# TroubleShooting

## Jenkins

1. 参数化node的过程中，通过调研发现执行运行的Node是内核的内容，不能做为参数，只能在配置job的过程中指定。最终找到的解决方案是[Node and Label parameter plugin](https://wiki.jenkins-ci.org/display/JENKINS/NodeLabel+Parameter+Plugin)， 这也是资源划分方案得以实现的最重要的一点。
2. jenkinsApi没有提供更改TCP port的方法，通过手动更改抓取post json数据的方式，保存为urllib2 post的参数模板，然后通过更改里面的port参数的值，来达到更改TCP port的目的。 缺点为只要Jenkins的全局安全配置发生的更改，post的模板就要更新，不过从目前的情况来看，这个模块需要更改的时候非常少。
3. JenkinsApi里面提供的offline Node的方法无法对无连接状态的Node进行操作，解决办法为采用urllib2 post的方式实现。

## Nginx

1. 对Nginx的操作采用SSH的方式，并且在端口转发的时候需要采用sudo的方式，但是第一次需要输入密码。但是python处理paramiko的时候无法继承当前的会话的状态， 可以采用telnet的方式进行操作，但是不是很便利，所以最终通过更改Linux的配置达到sudo的时候可以不输入密码。

## PC

1. 创建好的Node没办法远程在PC端运行。最终解决办法为，在PC上面自动创建计划任务，然后执行计划任务。

## Others

1. 僵尸Node或者不可用设备的处理，删除在Jenkins以及数据库的任务，并且在对应的项目表中更新deleted的状态。当设备再一次可用时，会做为第一次接入的流程进行。
2. 在Nginx服务接入的jenkins服务中，Slave Node与TCP port有一一对应的关系，所以当端口一旦被成功用过，如果再被其他Node使用时，是会失败的。所以当遇到这种情况，最好能找到原始的设备信息，更新回去，或者直接采用最新的TCP端口。