**CheungSSH Web3.0**

**前言**

**国产虽然不优秀，但请给予支持和理解**

在当今互联网的发展大势下，云计算和大数据已经成为主导。在两大趋势下，企业的服务器数量日益增多，如何有效的管理众多的服务器，已经成为了各大公司考虑的问题。在传统的人工管理模式下，已经很难做到高效的维护，是人管理服务器？还是服务器管理人？

为了解决这个问题，我痴迷于运维自动化研究，目前后端使用的Python语言+Django的Web框架，前端使用Bootstrap+Javascript+jQuery框架开发的CheungSSH Web3.0自动化运维系统，通信协议采用Linux系统自带的SSH协议，无Agent。并通过Paramiko库完成自动化登录，本系统可以基本满足日常的自动化运维需求。系统全程由我自主研发，不采用任何第三方工具和接口。

系统设计不足之处请各位朋友指出，也感谢大家的支持和信任，如果在使用过程中遇到任何问题，或者有任何的意见和建议，欢迎您与我交流学习。

**1.1 CheunSSH背景**

**Cheungssh 开源项目**

CheungSSH全程由张其川自主研发，后端采用的是Python语言，Django的web框架。前端采用Javascript、jQuery脚本语言，Bottstrap模板系统和div+css布局。前后端通信采用Restful API接口衔接。在2014年第一次推出shell版本，在2015年发布web2.0，目前的使用者和关注用户已经超过1000人。现在最新版本是cheungssh web3.0，前端和后端设计师都是张其川。

**1.2 CheungSSH特点**

* 无Agent，采用Linux系统自带的SSH通信协议，安全程度最高。
* 支持PC和手机终端，
* 支持任务流程编排
* 支持资产自动收集
* 支持主机状态自动检查
* RestFul API
* Web系统
* 自定义日志报表

**1.4 CheungSSH 功能清单**

|  |  |
| --- | --- |
| 功能 | 描述 |
| 自定义日志绘图 | 根据Nginx、Apache登录日志文件绘图，绘图数据包括：访问量走势，500错误，200正确百分比等 |
| 创建服务器 | 加入服务器 |
| 主机状态自动检查 | 加入主机到系统后，系统默认10分钟询问主机是否正常 |
| 系统网络拓扑 | 每个主机可以自动画出拓扑图 |
| 拓扑查看主机资产 | 在拓扑中双击该主机，就可以看到该主机的所有资源，比如CPU，状态，端口等。 |
| 批量命令 | 在网页中，模拟SecureCrt或者Xshell、Putty同时操作多个服务器执行命令，并且可以执行比如top、ping这样的动态命令。 |
| 批量命令计划任务 | 在批量执行命令出，可以把要执行的命令批量加入计划任务中，到了时间点再执行。 |
| 个人命令历史 | 在批量命令执行页面，显示自己的前5个命令历史记录。跟Linux的history一样。 |
| 批量文件下载 | 在Web上批量下载Linux服务器的文件，整个过程全自动 |
| 批量文件下载 | 从您的PC本地上传文件到各个远程服务器上，可以直接拖动鼠标上传 |
| 批量计划任务显示 | 显示各个主机上的计划任务列表 |
| 批量计划任务添加/删除/修改 | 批量操作远程服务器上的计划任务清单。可以删除/修改/添加 |
| 秘钥管理 | 如果您有Linux服务器是通过SSH-Key登录的，那么需要您上传秘钥文件，在这里进行管理。 |
| 批量脚本管理 | 您可以把您PC机器上的脚本拖动上传到各个远程服务器上，一键执行。 |
| 命令记录 | 您在web系统上一切的命令执行操作，都会被记录，并且在这里查看历史。 |
| 操作记录 | 记录那在web系统上，各个功能的操作记录，比如创建服务器，执行命令，修改计划任务等等。 |
| 登录记录 | 所有登录控制系统的用户均被记录。 |
| 命令黑名单 | 如果您的不愿意服务器执行某些特殊命令，比如：rm、shutdown、init等等，您可以把这些命令加入到这里，CheungSSH将会为您拦截这样的操作行为，但是如果您是超级管理员账号，系统会提示您是否强制性，如果是普通用户，直接拒绝，不会提示。 |
| 登录阈值 | 系统的登录安全，比如银行卡输入密码次数过多，就会锁定。这里是根据IP地址锁定的，如果超过系统默认设置的5次，那么就会锁定该IP地址，任何账户即便用正确账号密码也不可以通过该IP登录。 |
| 远程文件管理 | 直接在web上查看远程服务器的文件内容，无需cat、more、less命令 |
| 自定义资产 | 自定义手机条目，比如供应商手机，CPU和内存信息等。分为静态和动态资产。 |
| 资产展示 | 把所有的资产信息进行显示 |
| 资产数据画图 | 有历史数据的资产，均可以自行绘图，目前默认是折线图。 |
| 应用管理 | 直接才web界面操作linux服，或者 是自定义的程序等，比如tomcat，apache。 |
| 批量软件安装&应用部署 | 在cheungssh直接进行软件的安装和tomcat、weblogic等应用的代码部署 |
| 批量Docker镜像管理 | 自动采集远程主机上的docker镜像，并且支持镜像添加和创建容器 |
| 批量Docker容器管理 | 自动采集远程主机上的docker容器，支持删除、启动容器 |
| WebSSH | 在网页上直接登录SSH服务器 |

**1.4 CheungSSH API接口清单**

下列为每一项API的费用和功能明细

|  |
| --- |
| 功能 |
| 添加服务器 |
| 网络拓扑 |
| 命令历史 |
| 命令执行 |
| 命令定时执行 |
| 批量文件上传 |
| 批量文件下载 |
| 创建计划任务 |
| 脚本执行 |
| 脚本创建 |
| 新增拦截命令 |
| 远程文件/日志查看 |
| 自定义资产类型 |
| 资产信息 |
| 执行应用 |
| 执行部署任务 |
| Docker镜像下载 |
| Docker容器创建 |
| Docker容器启动 |
| Docker容器停止 |

**1.4 CheungSSH联系方式**

* QQ群：517241115

**1.2 CheungSSH安装方式**

* 一键安装

**1.3 一键安装**

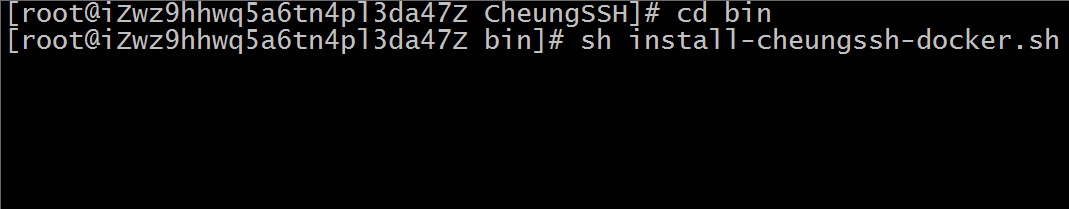
条件**：**

* 需要联网（已有Docker则无需联网）
* 系统为CentOS 6.5（以上版本或7都可以） 64bit
* 第一步 下载软件包（建议在开源中国下载，国内服务器网速更快）

http://git.oschina.net/CheungSSH\_OSC/CheungSSH

* 第二步

解压软件包，进入bin目录，然后执行安装脚本：



等待安装完毕后，即可打开网页。

* 第三步 访问页面

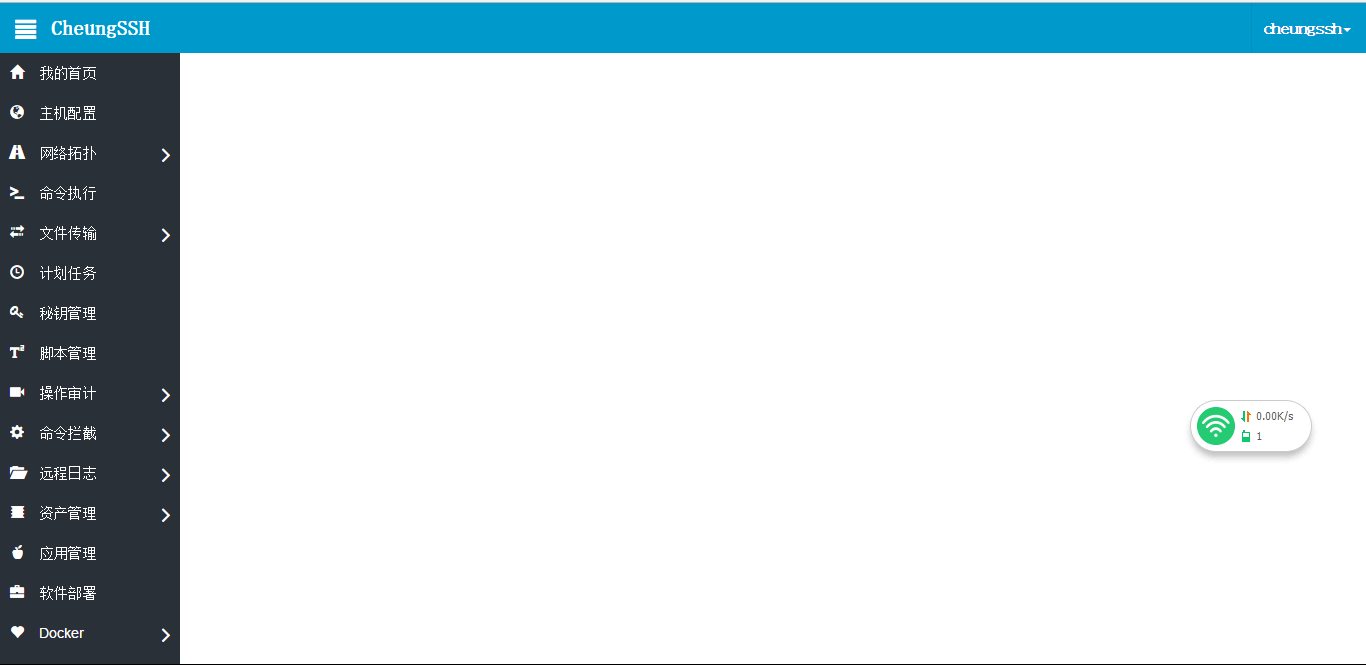
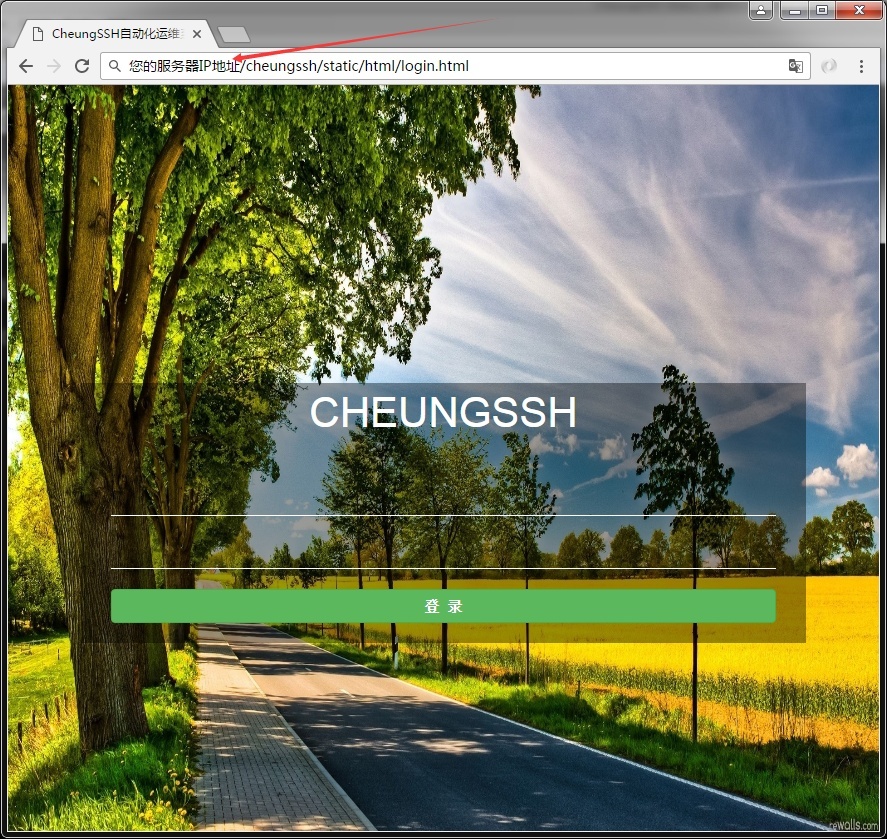
访问您安装CheungSSH系统的IP地址，在上面安装的时候指定端口

比如笔者的服务器地址是:

http://192.168.1.1

输入后，系统自动为您跳转到登陆页面，默认登录用户是：**cheungssh**默认登录密码是：**cheungssh123456**

**注意：**如果您登录的错误次数超过5次，系统会锁定您的IP地址，以后即便您使用正确的账号密码也不能正常登录系统，一天以后才能自动解锁，这是CheungSSH的安全机制。在此期间，您可以更换IP地址登录。



**1.4 启动CheungSSH服务**

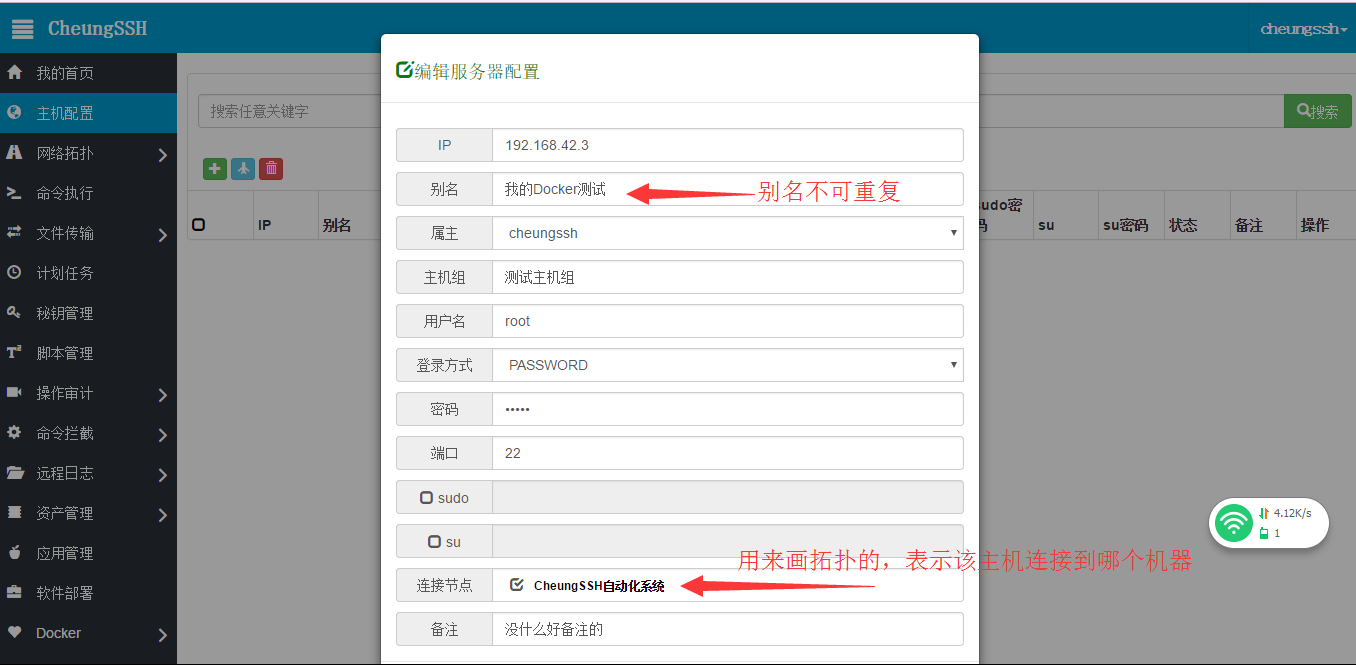
默认情况下，CheungSSH是随系统启动的，无需手工启动，如果您想人工干预，请访问：service cheungssh start

**2.1 服务器配置**

在使用CheungSSH自动化系统之前，您需要提前录入您的远程主机的信息，这样CheungSSH才能为您自动执行任务。否则，自动执行根本无从谈起。CheungSSH支持的登录方式有SSH密码登录、SSH-Key登录两种方式。

**2.1.1 单个添加被管理服务器**

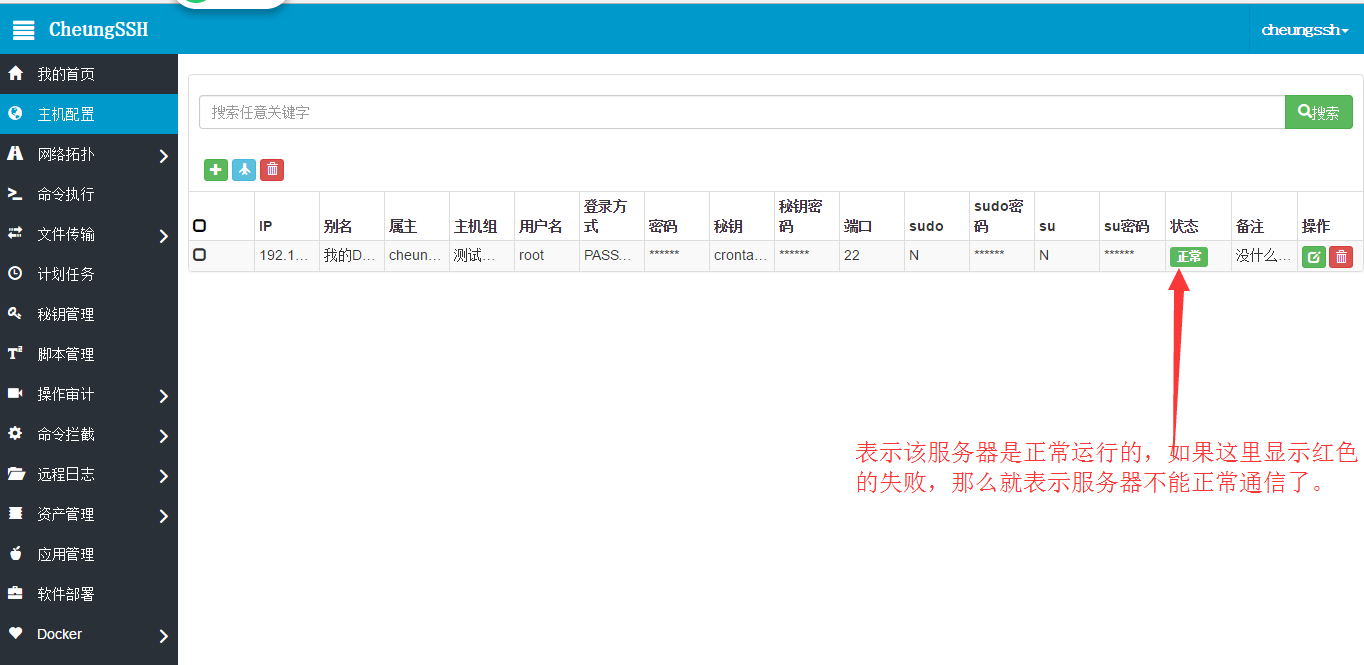




最后点击保存。



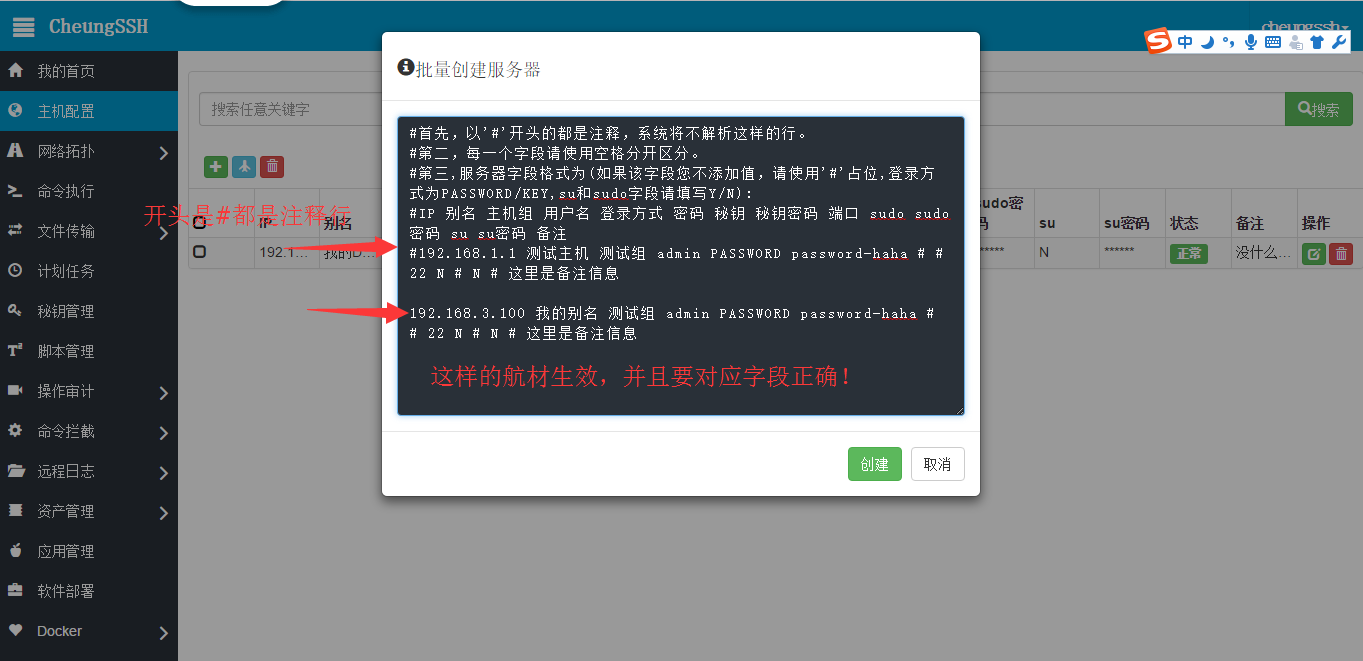
上面的刷新图标表示cheungssh在请求该服务器的状态。如果该服务器状态正常，则会显示如下图标：



**2.1.2 批量添加服务器**



您可以把您准备的txt文件，安装cheunggssh要求的格式，复制粘贴到里面就可以了。注意格式！ 如果是“#” 开头的，则表示注释！



**3. 1 网络拓扑**

到目前为止，您通过服务器配置添加了服务器信息，并且选择了主机连接的节点，那么到现在为止，您可以看到如下的拓扑情况：



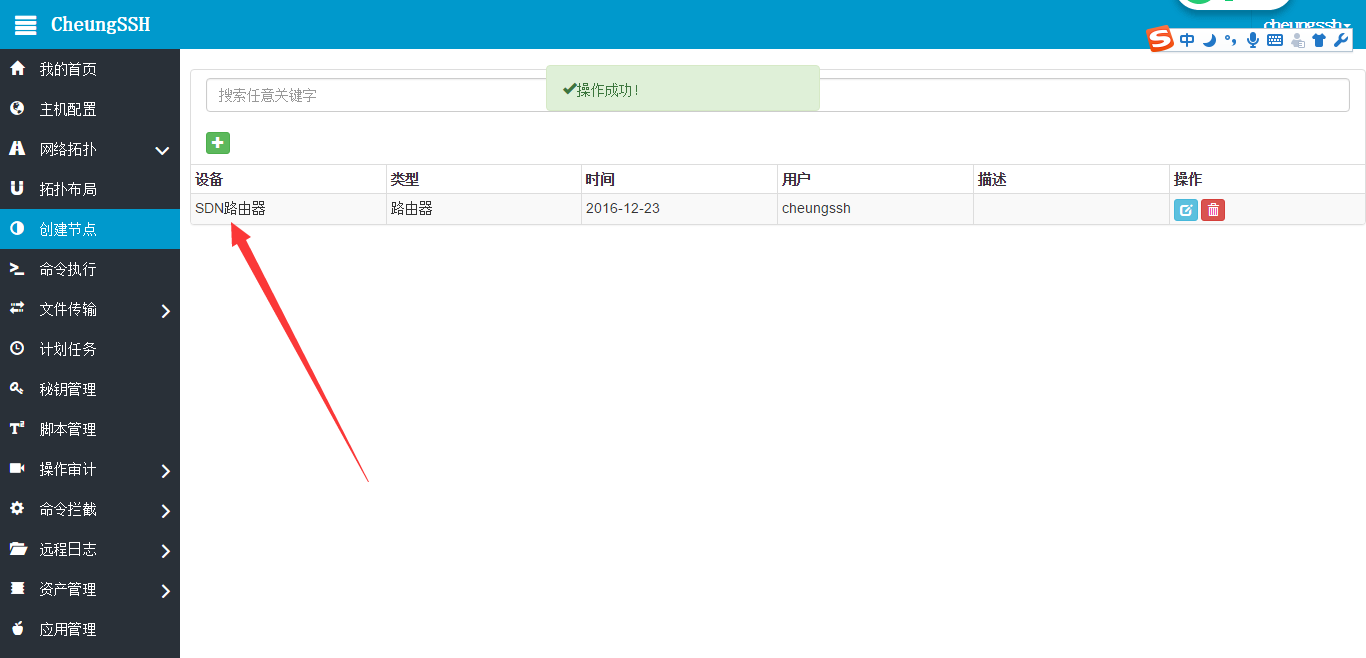
**2.2.1 添加拓扑节点**

到现在，您只看到了您的Linux服务器和CheungSSH自动化系统的拓扑图，但是，这可定不够用，比如，您的网络中，除了Linux服务器，还有路由器、交换机、防火墙等等网络设备，当然这些设备不归CheungSSH管理，CheungSSH也没有对这些设备做管理。但是，您却可以通过创建节点的方式，把他们加入网络拓扑中。

****



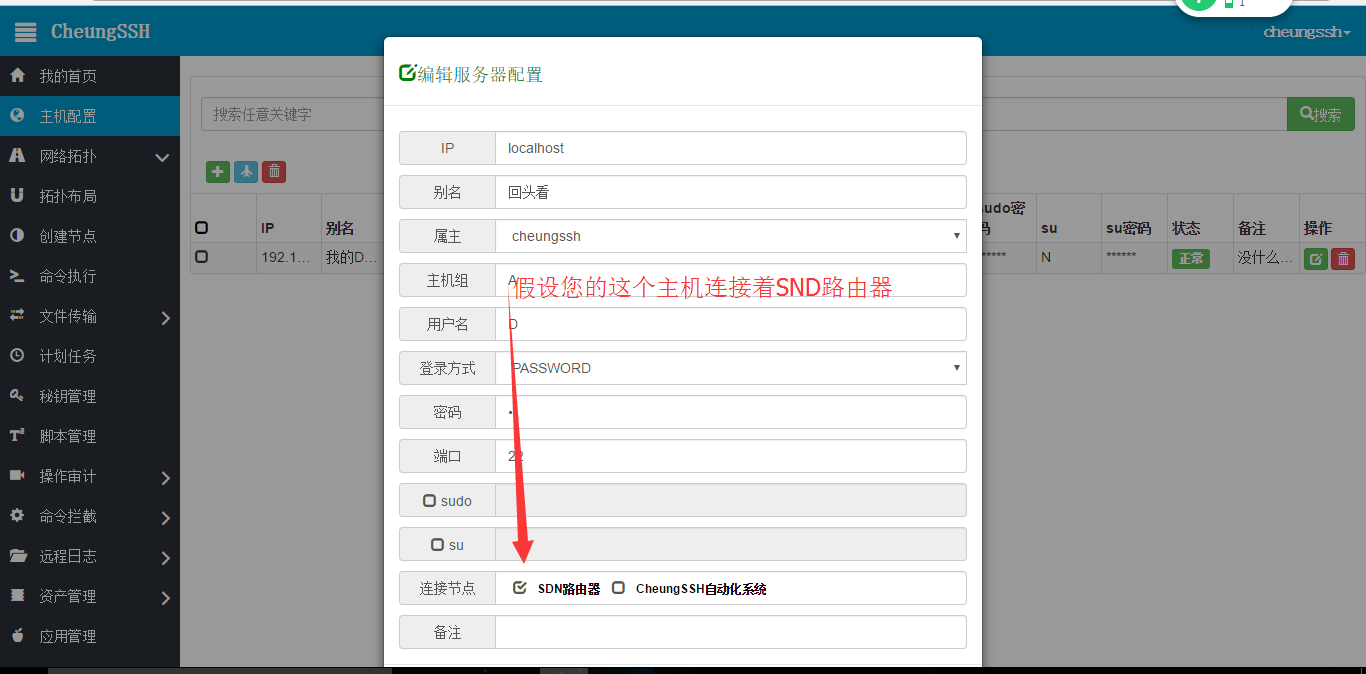
保存后看到如下列表：



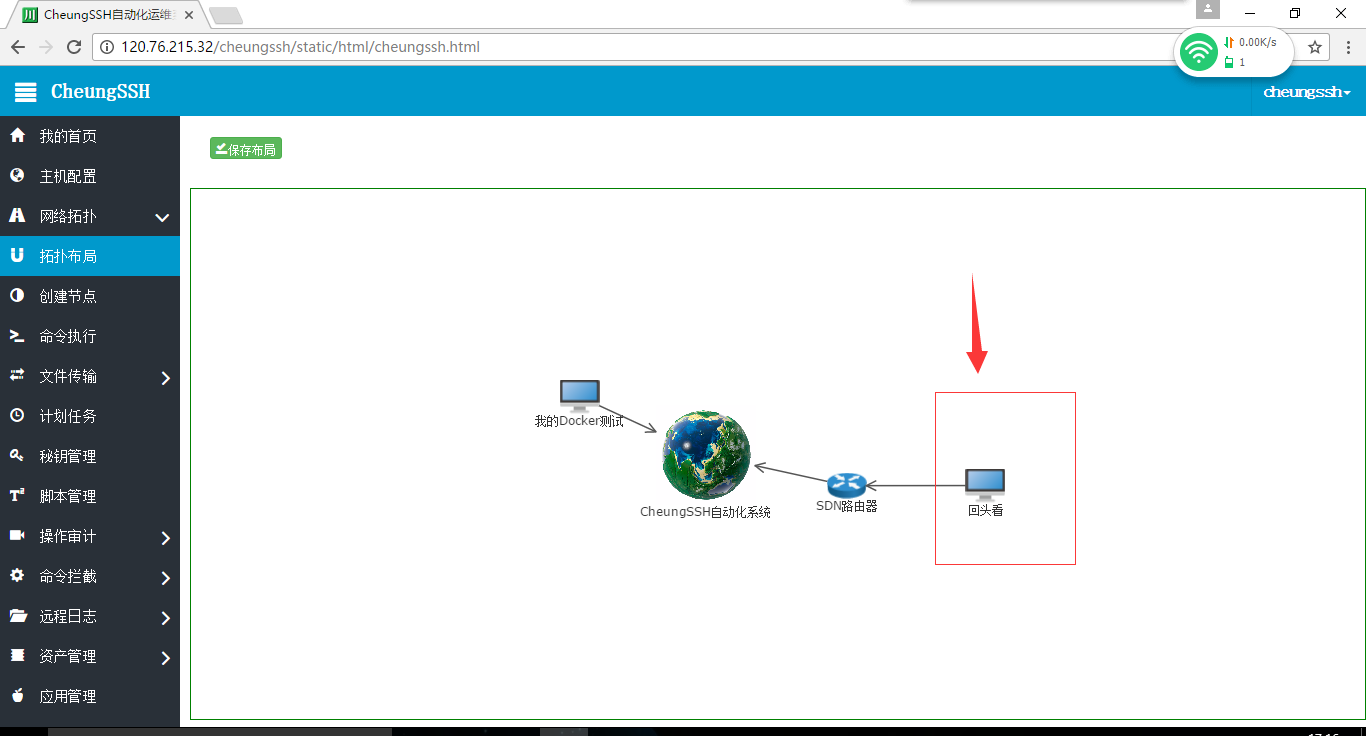
此时，您刷新web页面，然后定位到【拓扑布局】中，可以看到拓扑发生了变化



再回头，您再次添加一个服务器到系统中，并且选择该服务器的连接节点为您刚才创建的“SDN路由器”，保存后，您可以看到这个服务器连接着的就是SDN路由节点上了。

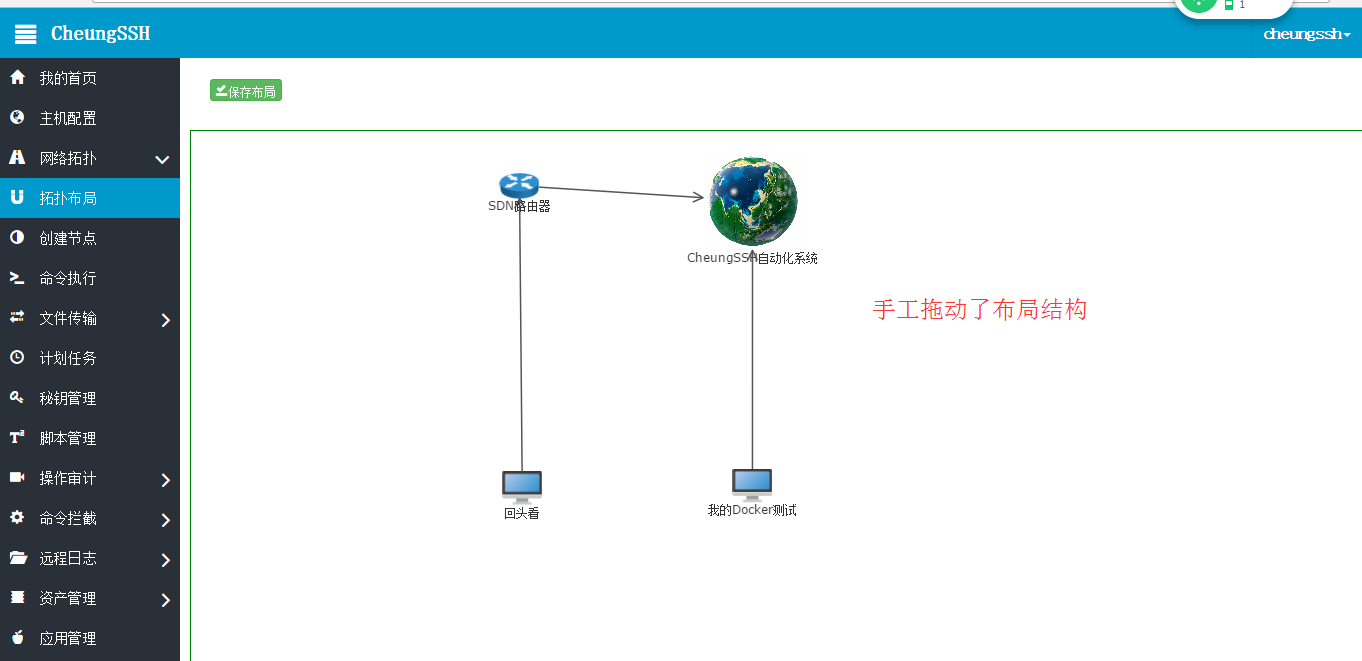


再次刷新页面，去看看网络拓扑：



**3.2 保存拓扑布局**

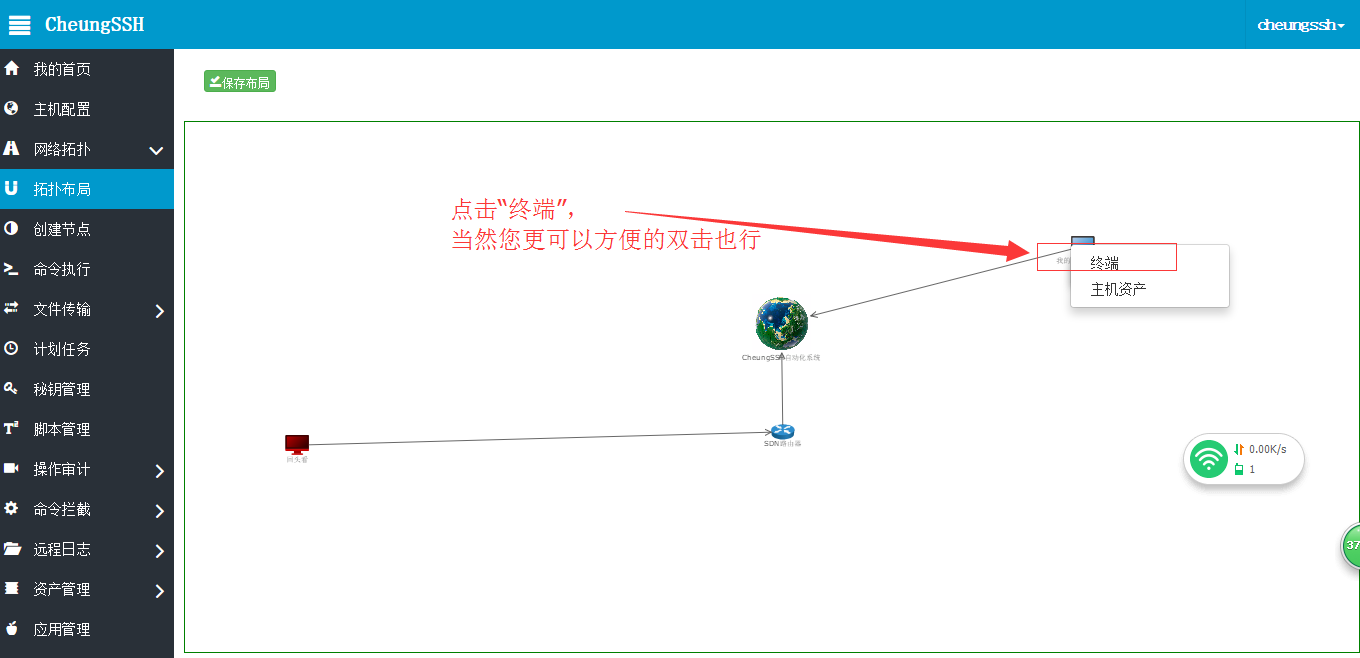
默认情况下，系统采用自适应布局。这是一个随机算法。您每一次打开网络拓扑的布局都不会相同。比如，您觉得系统的自动化布局不方便您观察，在您拖动了设备的布局后，比如这样：



但是，当您再次刷新页面，打开这个布局的时候，页面的布局又会回到原始的布局，系统并不会为您保存这个布局。

**3.2 执行设备属性（双击进入终端）**

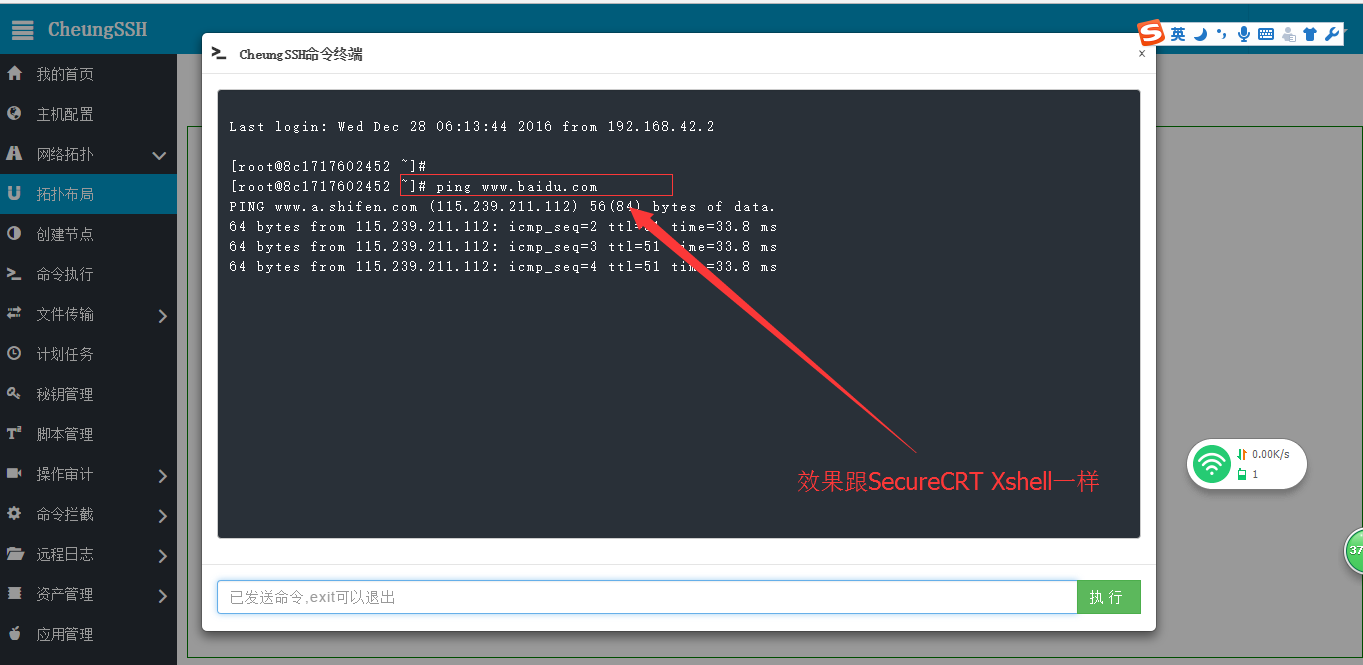
为了方便使用，作者特意为大家定制了双击网络设备图标，就进入交互式命令终端，在这里您可以执行如cd，top，ping这样的交互命令，但仍然不支持部分交互命令，比如vim等。



您可以双击设备图标，或者是右键，打开终端即可：



执行一个ping [www.baidu.com](http://www.baidu.com)的命令：



如果您想退出终端，您可以exit或者是logout，当然这些都是linux系统命令，再或者您可以直接关闭这个终端界面。

注意：在这个终端里面，您可以cd目录，ping、top等命令都可以，但是不支持比如vim、crontab这样的命令，如果您想要vim打开一个文件，请您到文件管理功能使用。

另外，在这里的属性功能【终端】和【资产管理】中，只有服务器才具备这样的功能，其他非服务器，比如防火墙，路由器，是不具备这样的功能的。

**3.2.2 资产属性功能**

当前开发中，还不支持点击拓扑图标打开资产的功能，如果您需要这样的功能，可以定制开发。

**4.1 命令执行**

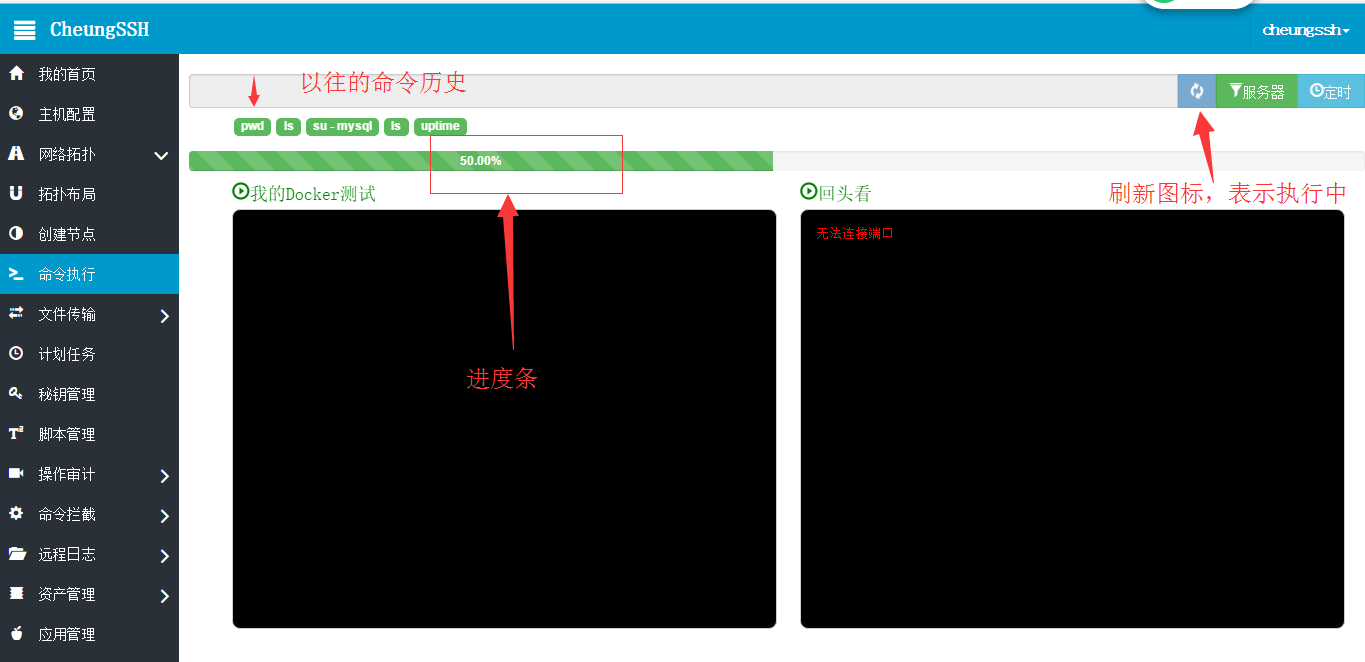
该功能如同Xshell、SecureCRT、Putty一样，可以批量的发送系统命令，让Linux服务器执行。

**4.1.1 选择要执行命令的服务器**

默认情况下，系统已经选择全部主机，如果您要针对部分主机进行执行命令，请您首先选择要执行的服务器。



按下回车：

****

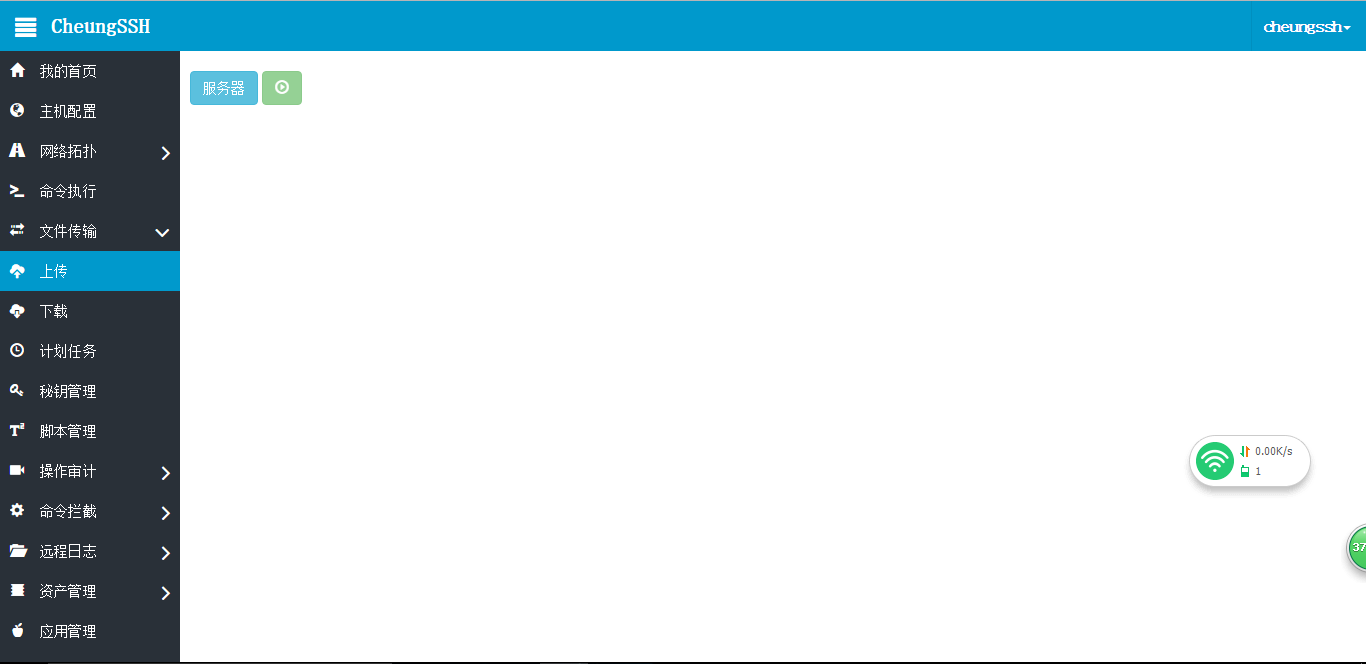
****

**4.1.2 把命令加入计划任务**

该功能暂时不支持免费版本，待日后开放。

**5.1 文件上传**

文件上传，是指您把您当前的PC电脑上的文件批量的上传到远程服务器上，在这里，您可以很轻松的通过拖动方式上传，只需要动动鼠标就搞定。



第一步，您应该选择要上传的服务器：

~~~~

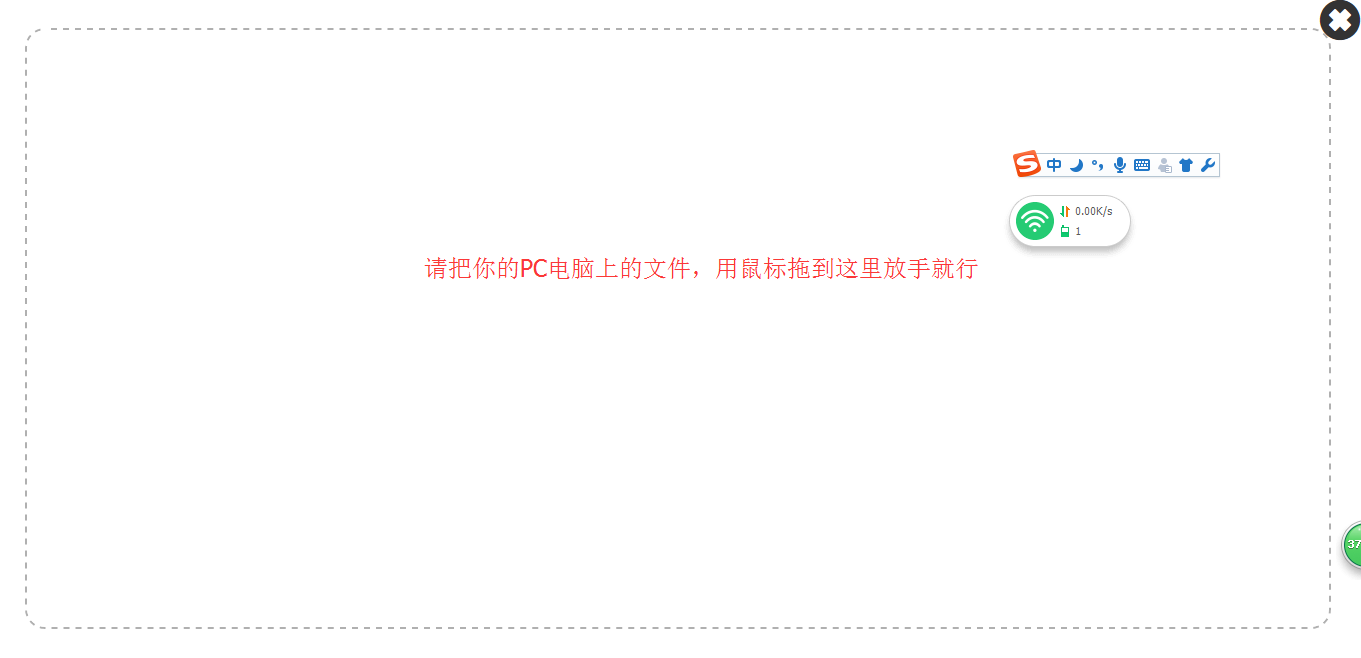
选择好了以后，点击【确定】，然后进入下一个页面。

**5.1.1 文件上传-快速上传**

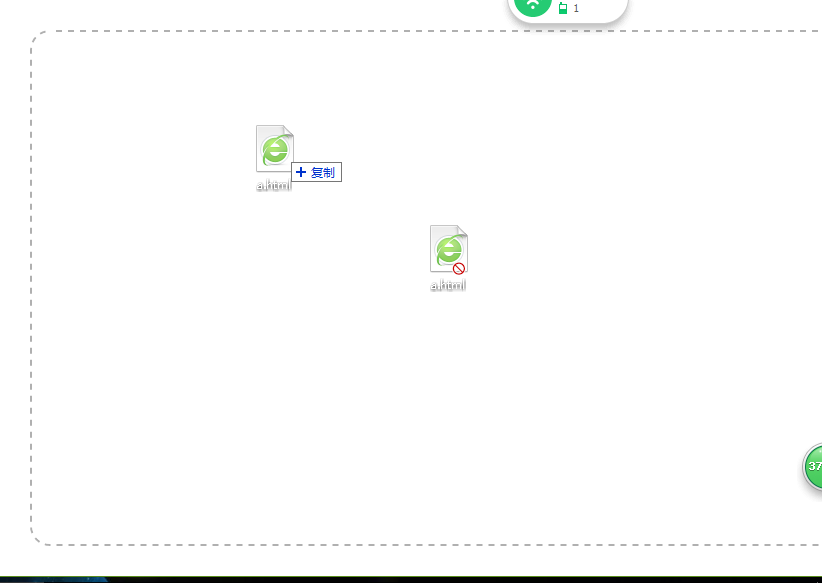
点击【快速】，意思是速度快，步骤简单

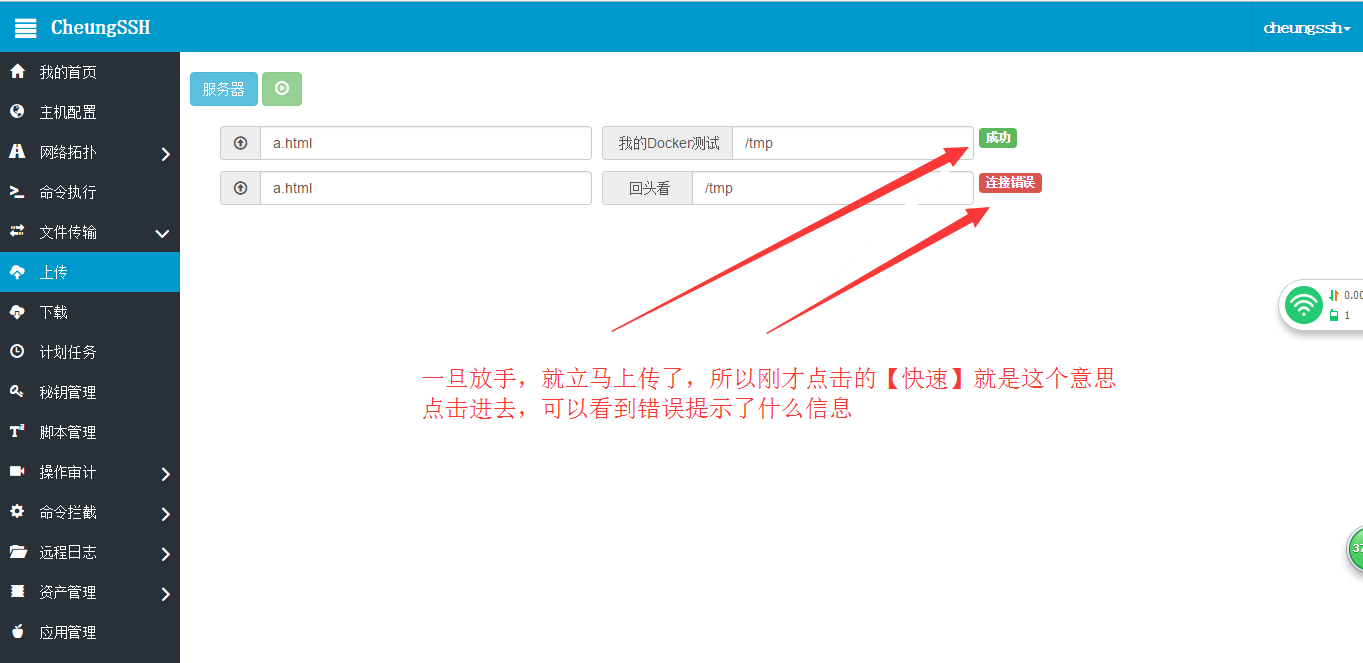


在上面的输入中，是输入您要把您的文件传到远程服务器的哪个目录下，注意，比如您选择了10个服务器，10个服务器的目标路径都一样，那么您在这里点击【快速】即可。但是如果您10服务器中，路径不是都一样，那么您必须选择【高级】，这样的动作，是为了区分路径而已，下面演示是【快速】方式。



在上面，请你把文件拖放就行，然后CheungSSH立马给您上传到远程服务器上去：



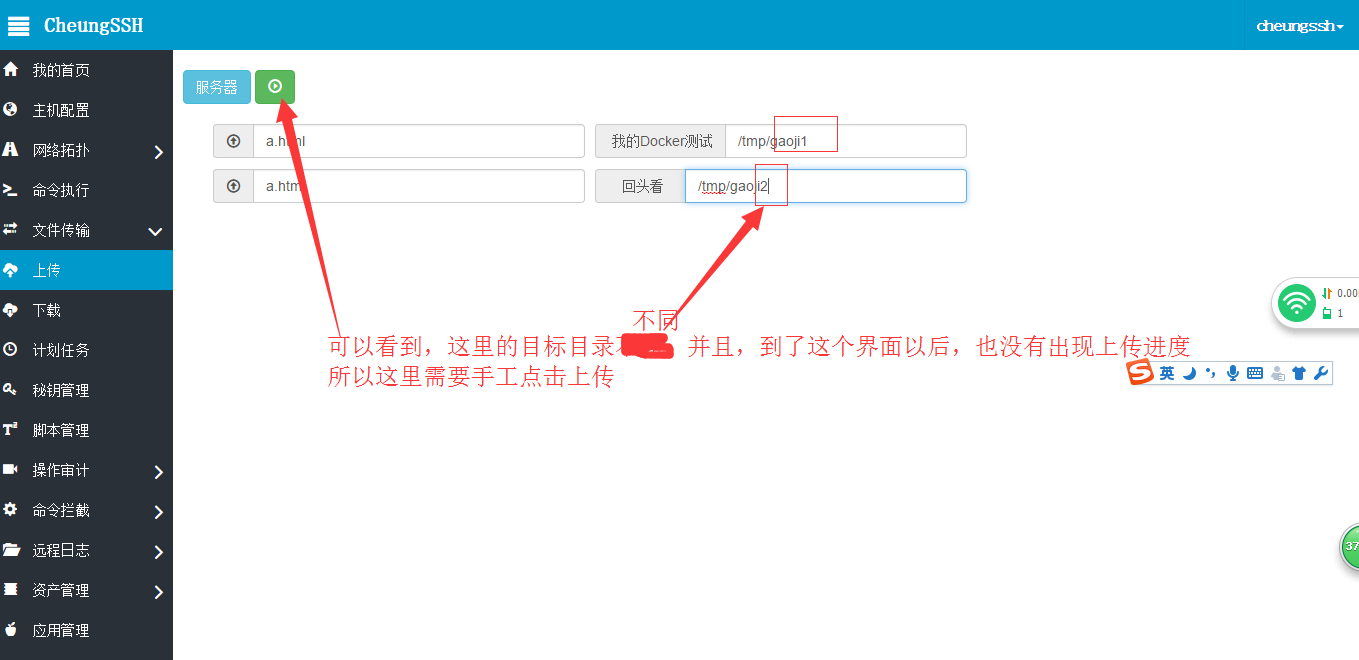


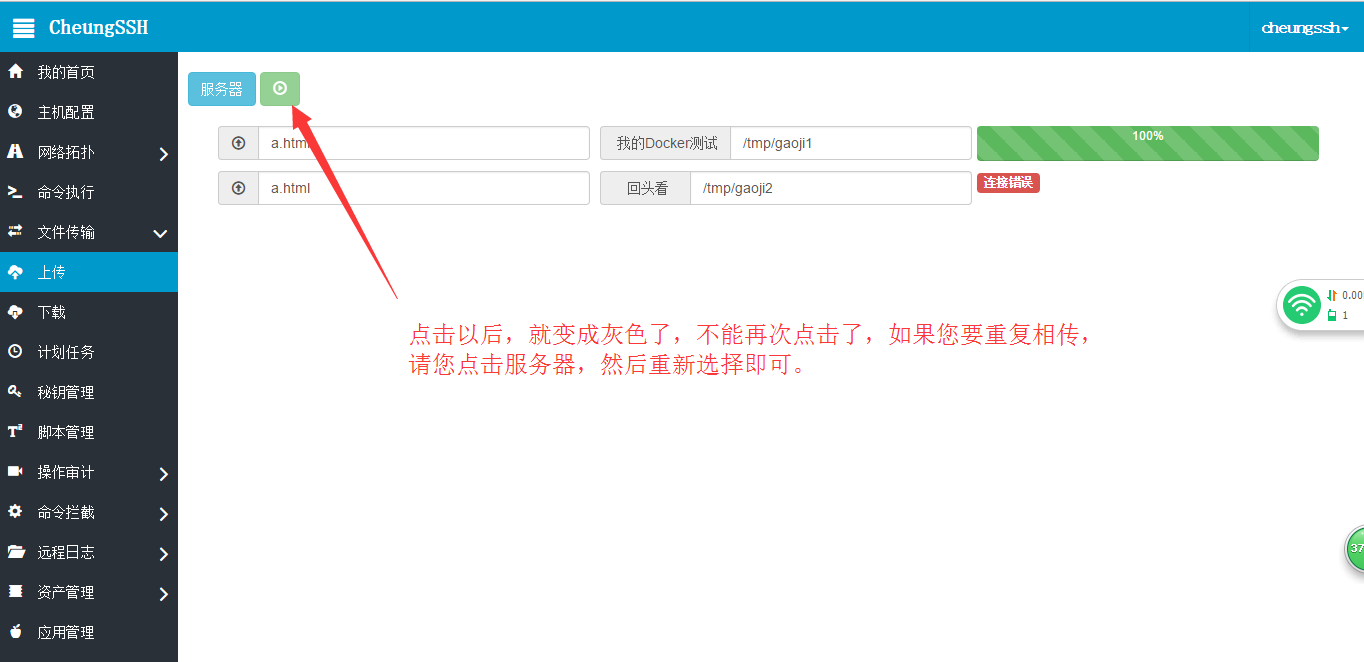
以上是快速上传。

**5.1.2 文件上传-高级上传**

意义在于，如果您要上传文件的10个服务器中，加入有3个服务器的路径不是一样的，那么在这种情况下您就可以使用本功能，单独执行不一样的路径

现在，点击【高级】上传：

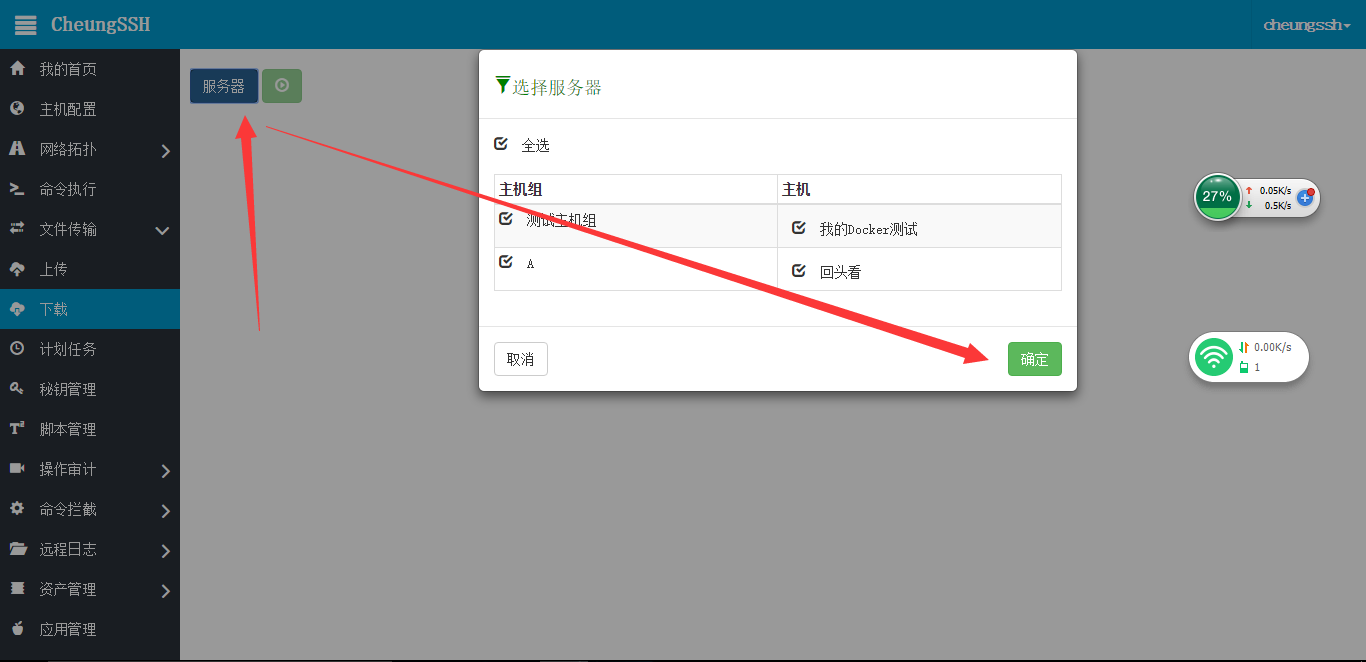




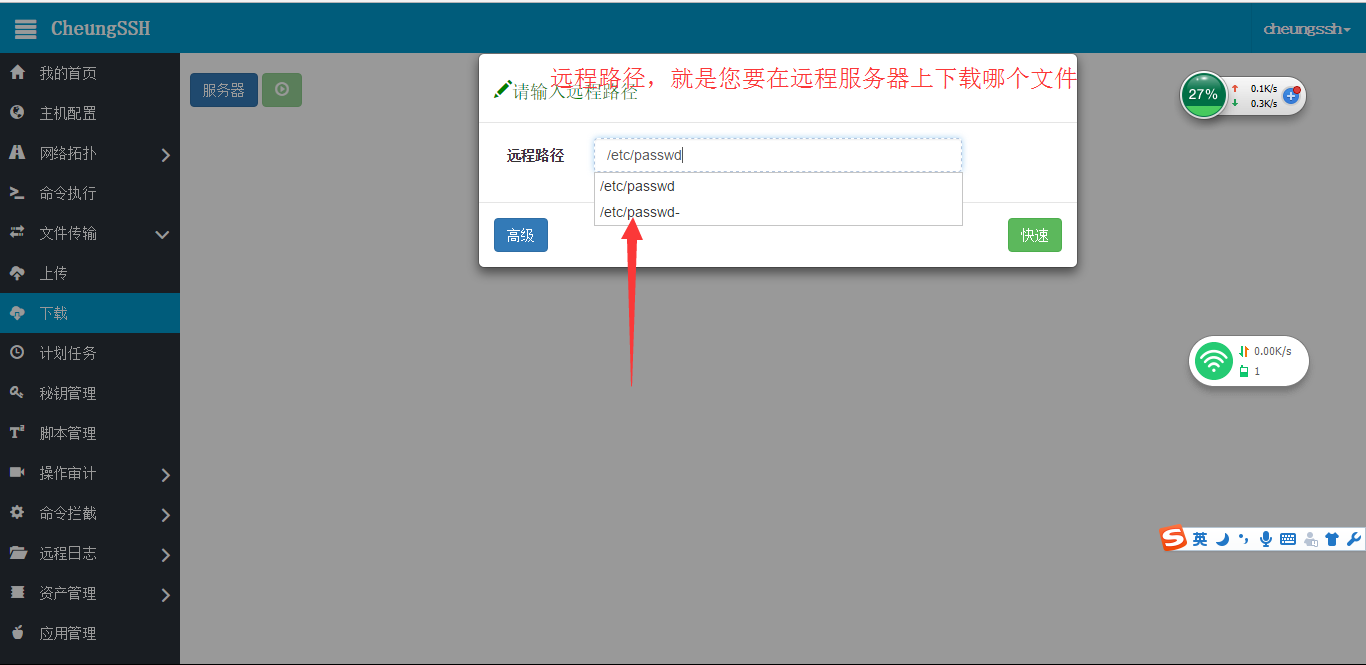
**5.2 文件下载**

文件下载，作者设计的比较先进，如果下载顺利，真的还需要点两次鼠标，所有的远程文件就自动的下载到了您的PC电脑上了。

现在，请您首选选择要在哪些服务器上下载文件：

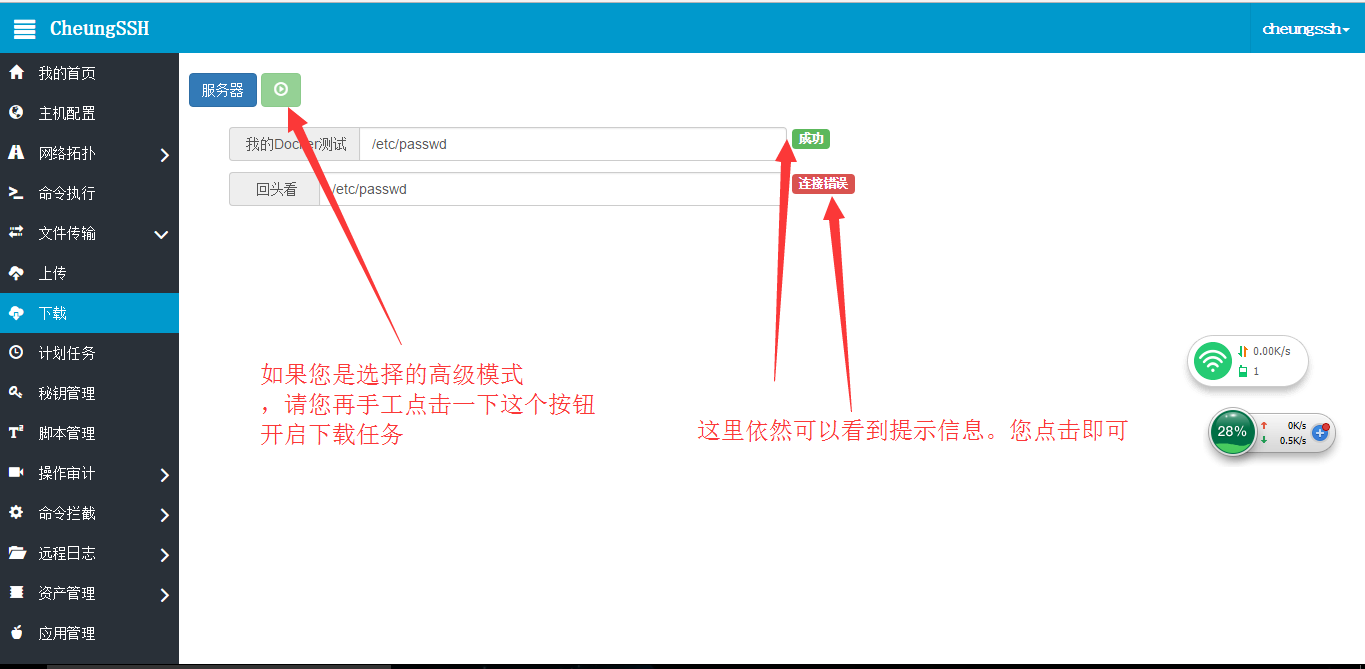


选择好了服务器以后，现在选择远程服务器的路径：

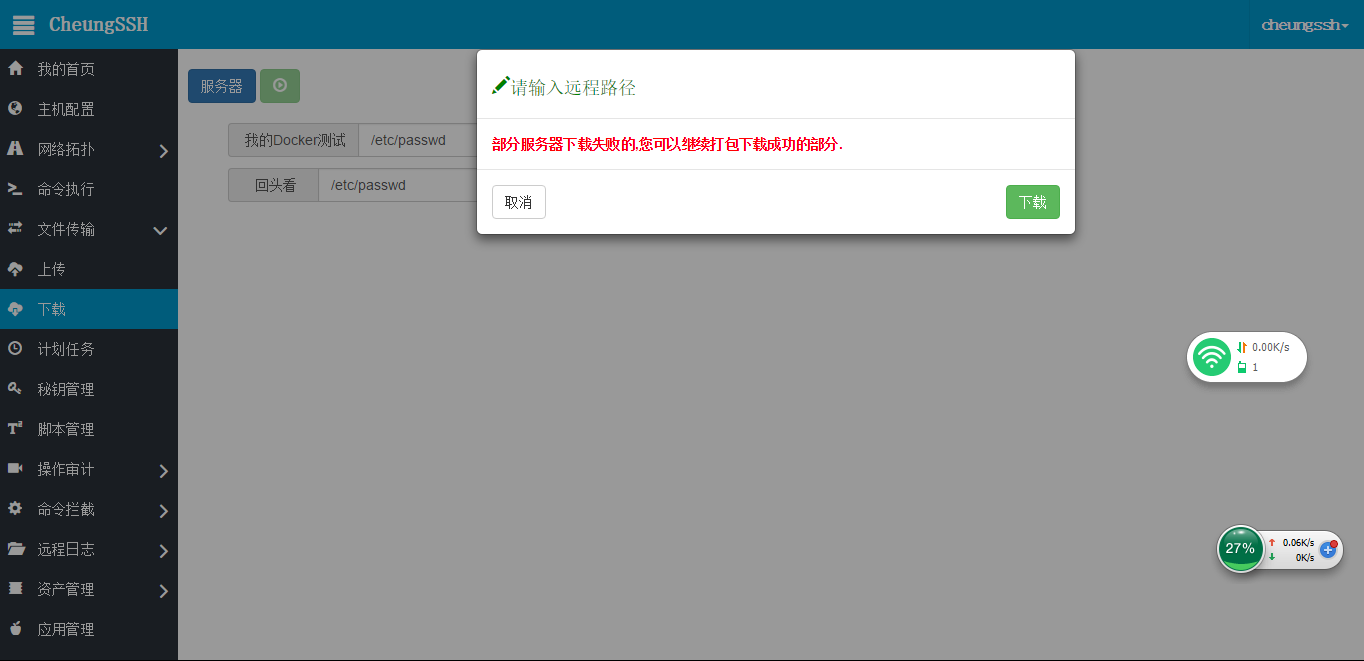


【高级】选项说明：比如您要下载10个服务器上的文件，但是这些文件的路径并不是都一样，那么您就可以通过这个选项来进行指定，指定好了以后，您需要手工启动下载。

【快速】选项说明：比如您要下载10个服务器上的文件，这些文件上的路径通通都一样，那么您只需要在这里输入路径，CheungSSH系统就替您一键下载，当然步骤更简洁一些。



如果在下载过程中，有部分服务器下载失败了，那么CheungSSH会弹出提示，告诉您是有部分失败了，然后您可以点击【下载】就能确认下载文件到您的PC电脑上了：



如果您要继续其他下载任务，请您重新选择一次【服务器】后开始即可。

**6.1 计划任务**

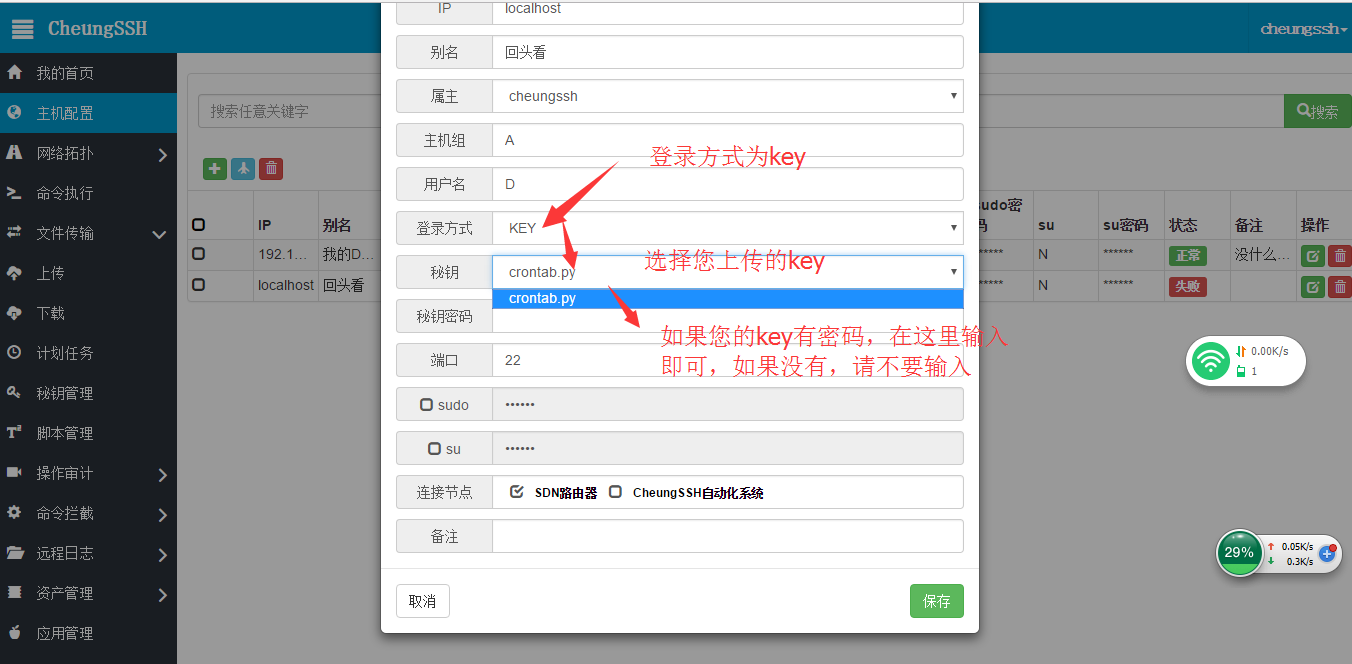
该功能暂时未开发。

7.1 上传秘钥

在CheungSSH所有的文件上传的功能中，都是通过拖放的方式上传的。



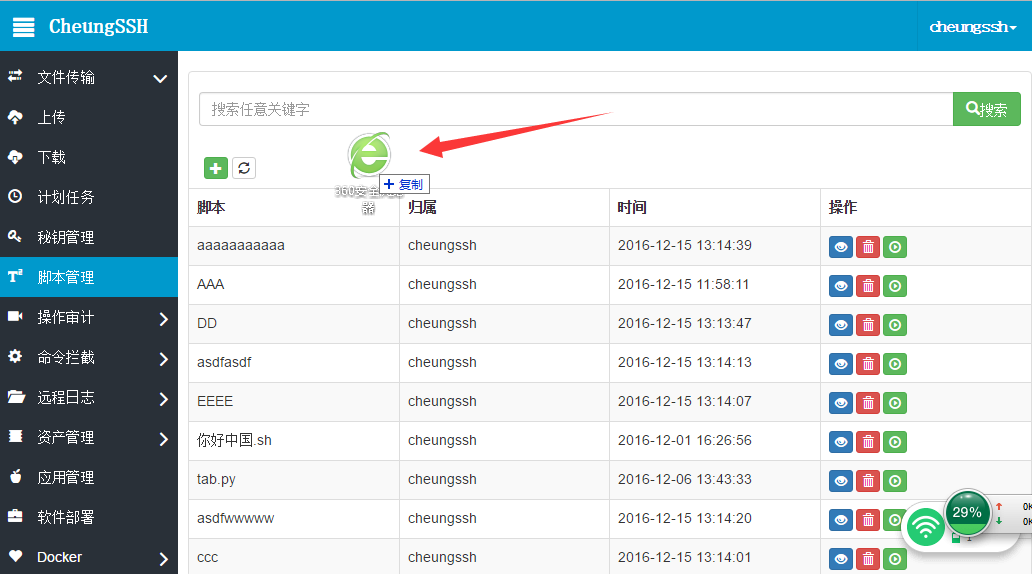
当然了，上面我只是演示行的放几个文件而已，不要在意文件名。此时，如果您有服务器是通过SSH-Key登录的，那么在服务器配置中，您可以做如下选择：



**8.1 上传脚本**

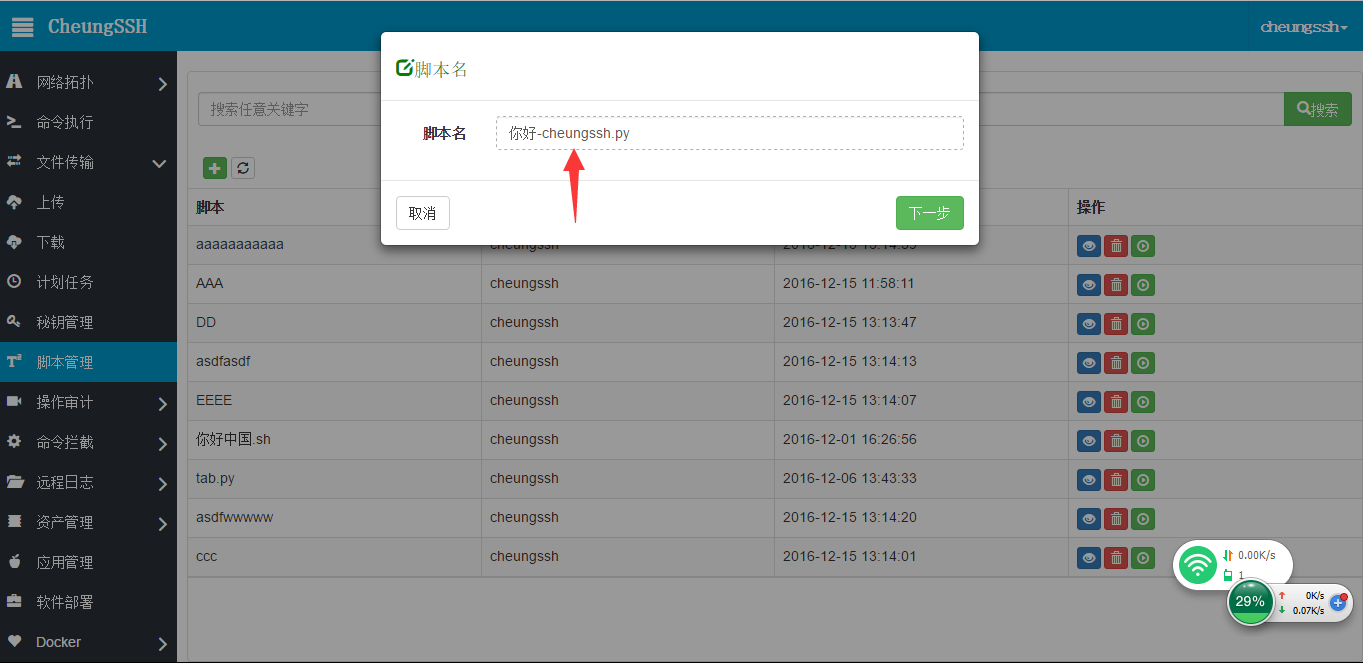
在cheungssh中，一样可以执行脚本，创建脚本的方式有两种。并且友好的支持中文的显示和脚本的中文名。

第一种是从您的PC电脑上传一个脚本文件，当然了，还是以老用法，直接拖动就行：

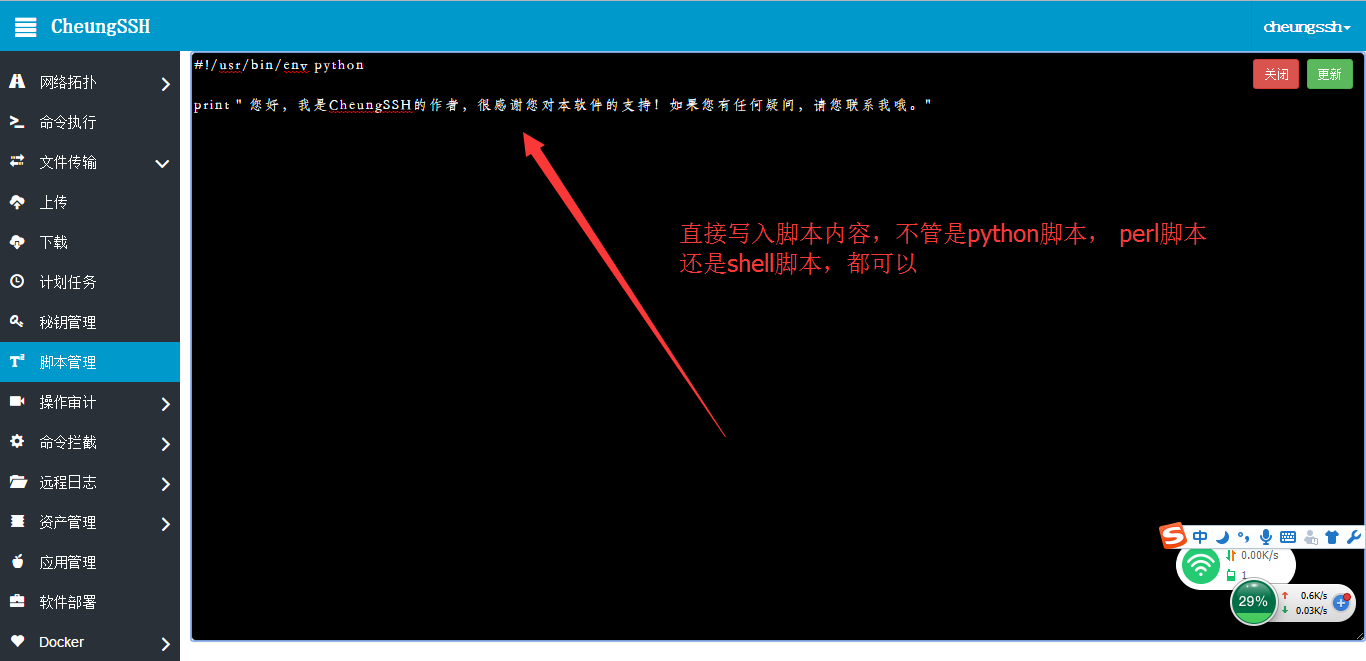
****

第二种上传方式，直接点击 在网页上直接写入脚本内容即可：

输入脚本名：

****

开始写入脚本内容：

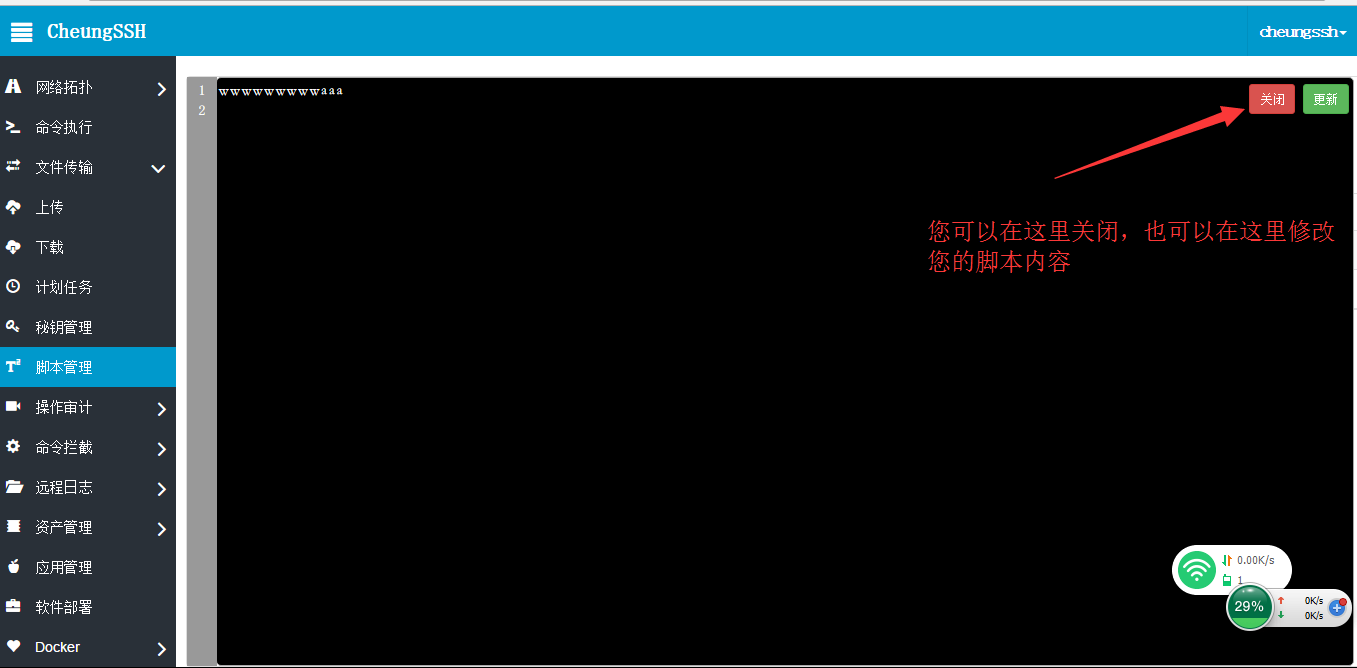
****

**写好了以后，请点击 即可。**

****

**8.2 查看脚本内容**

**查看脚本内容，只需要点击即可：**

****

**8.3 执行脚本**

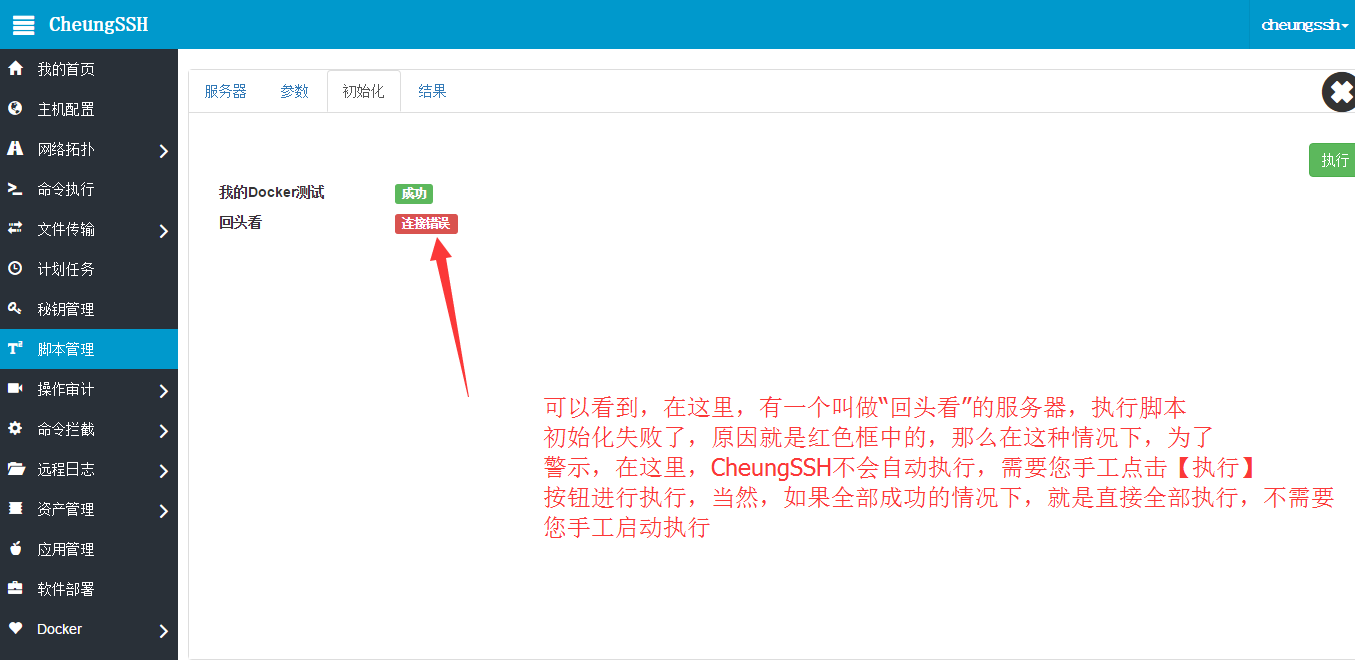
**执行脚本，请点击**

****

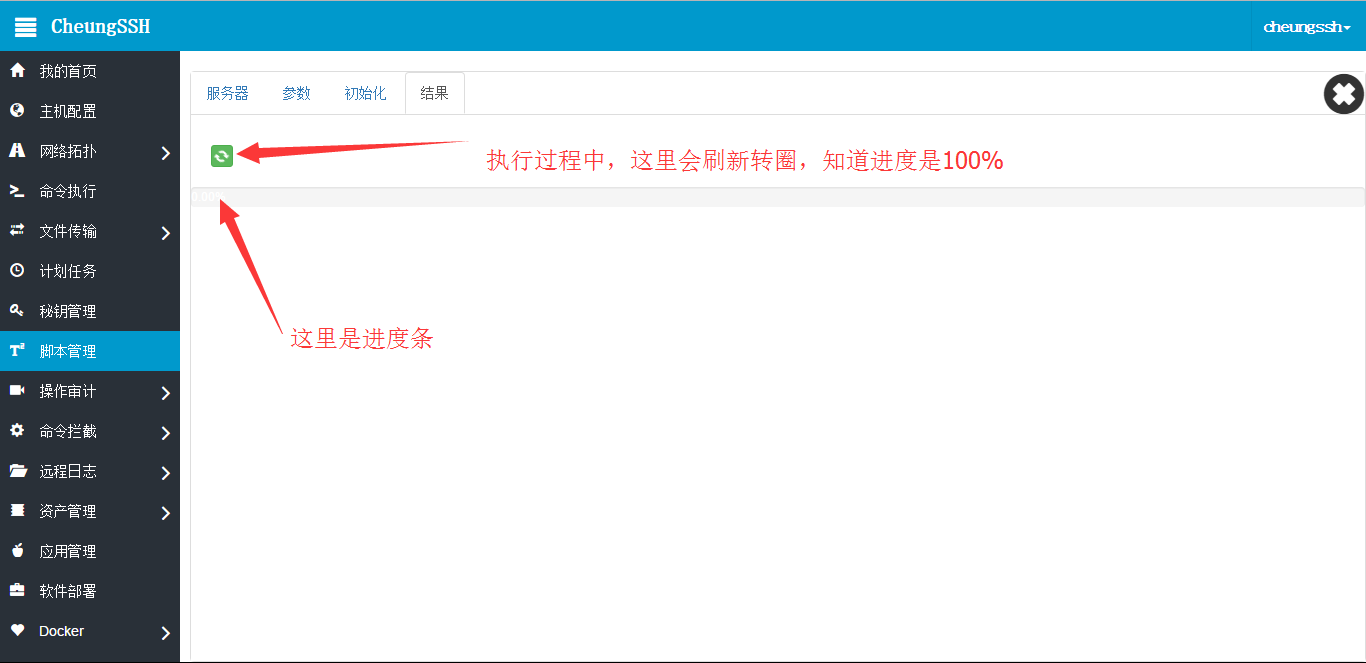
点击了下一步后，可以输出脚本参数，当然你可以可以不输入，直接下一步就行：



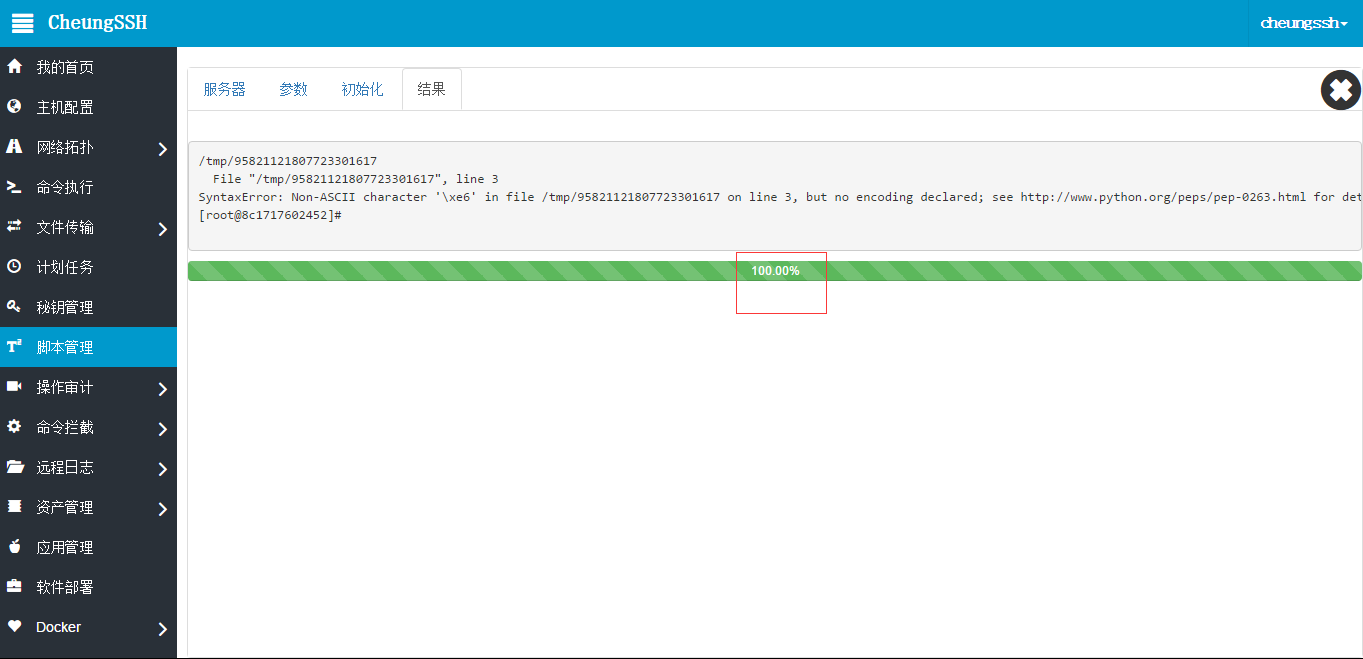
现在，点击【下一步】执行开始出初始化脚本：



现在，我手动点击【执行】按钮，那么久等待脚本执行结果了：



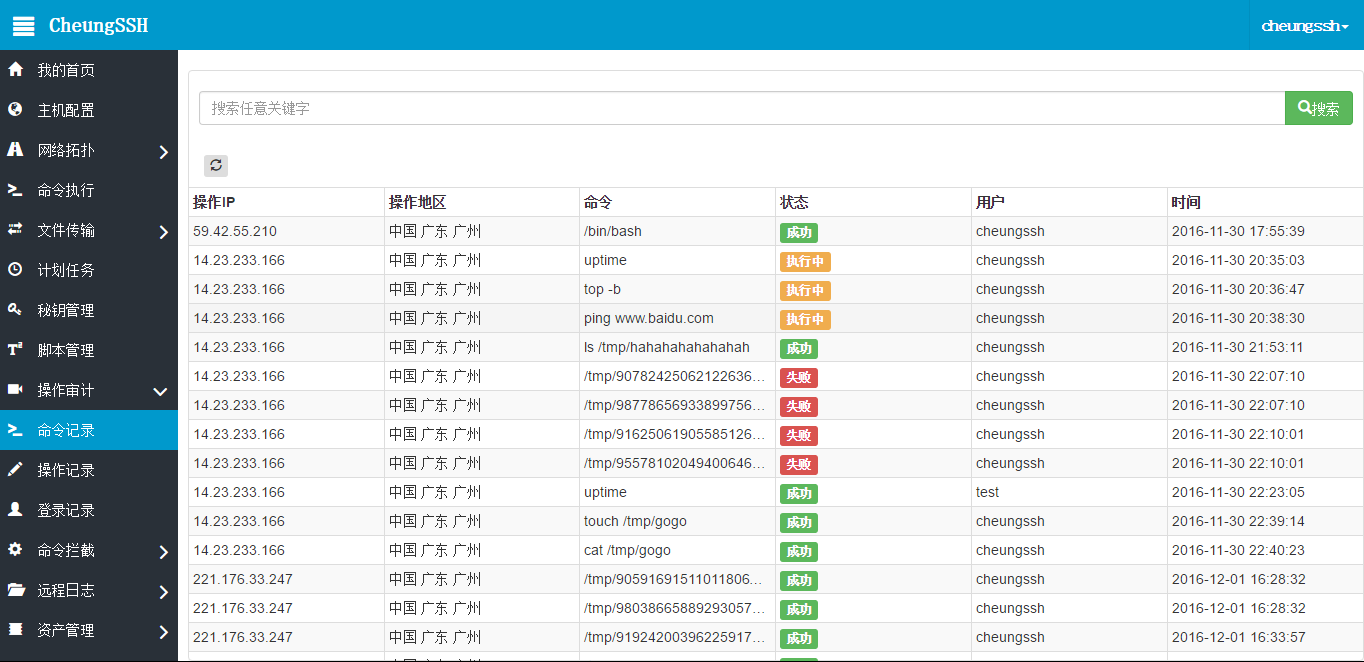
下面，就是显示脚本执行结果：



**注意：**每当您点击开启一个脚本执行流程后，如果需要再一次执行，那么请您务必关闭当前的执行窗口，重新点击执行。

**9.1 命令记录**

在【命令执行】功能中，所有的命令执行记录均被记录到了这里。



关于【状态】的解释：

* 表示执行命令是正确的，也就是说echo $?的值是0；
* 表示执行的命令可能是失败的，也可能是服务器的问题

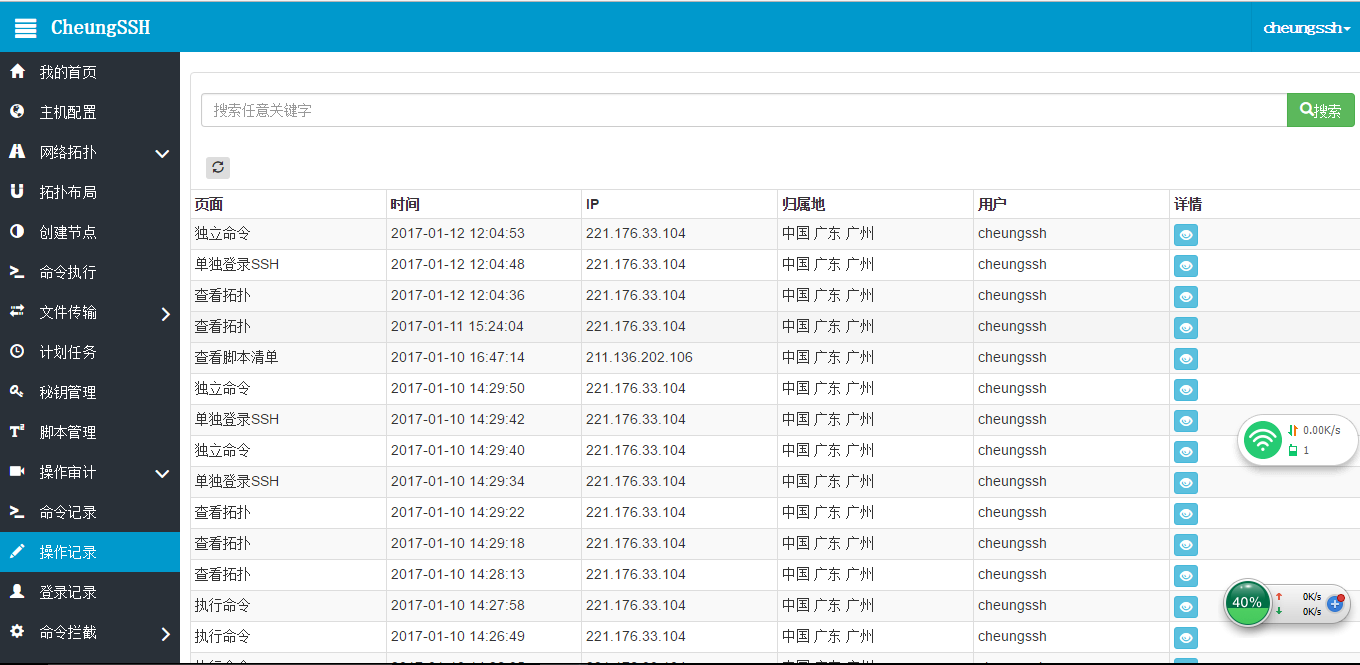
 表示这个命令还在执行过程中，没有结束，比如ping，top等这样的长执行命令

对于上述的三种状态，如果您要查看详情信息，可以直接点击它，然后查看详情，如下：



**9.2 操作记录**

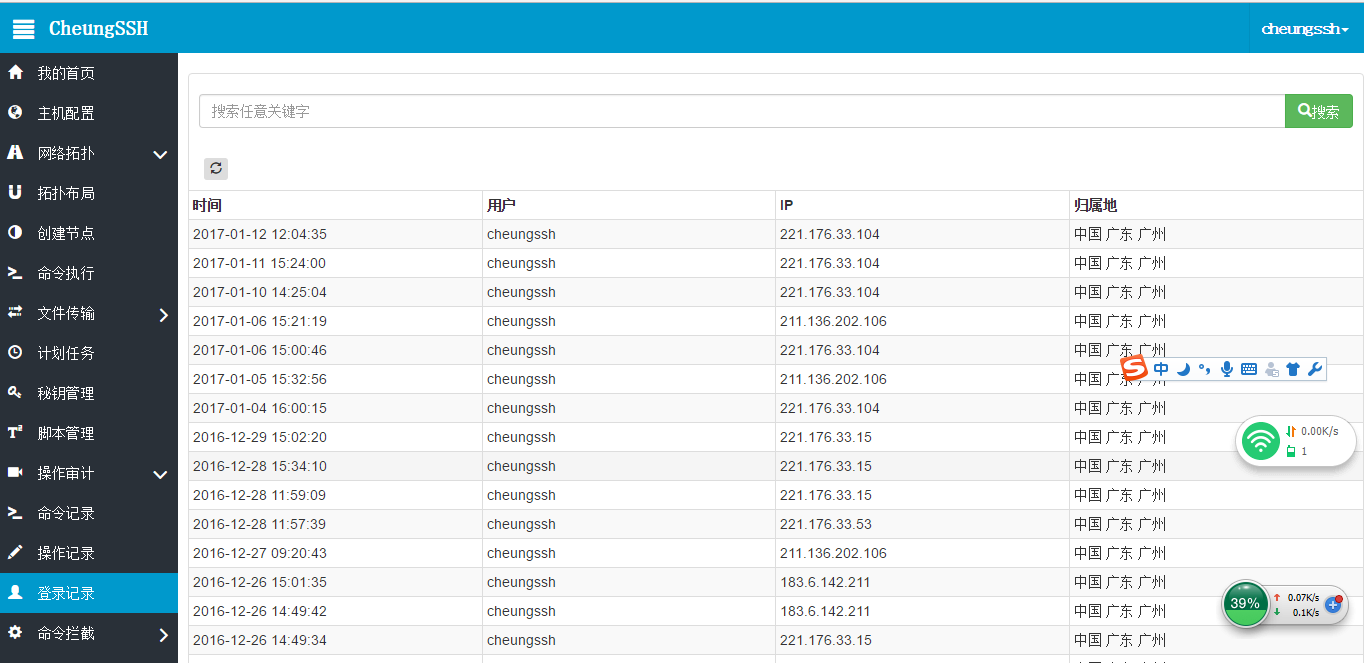
在CheungSSH系统中，所有的访问，比如执行命令，添加主机配置，删除资源等等，所有的操作都被审计记录在这里。



可以点击 查看用户操作的详细请求，比如添加的服务器，IP是多少，密码是多少等等，但是该功能目前仅仅服务于商业版本用户。

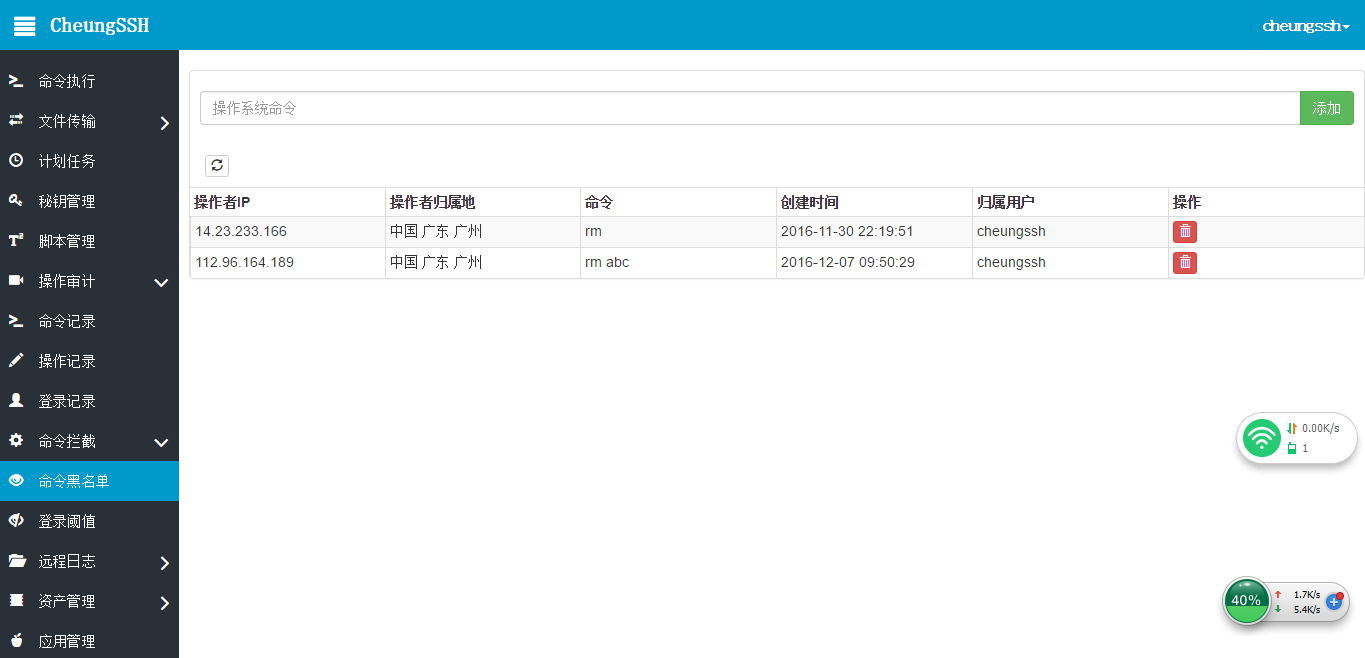
**9.3 登录记录**

凡是登录过CheungSSH系统的用户，都有记录什么时候登录，在哪里登录的。

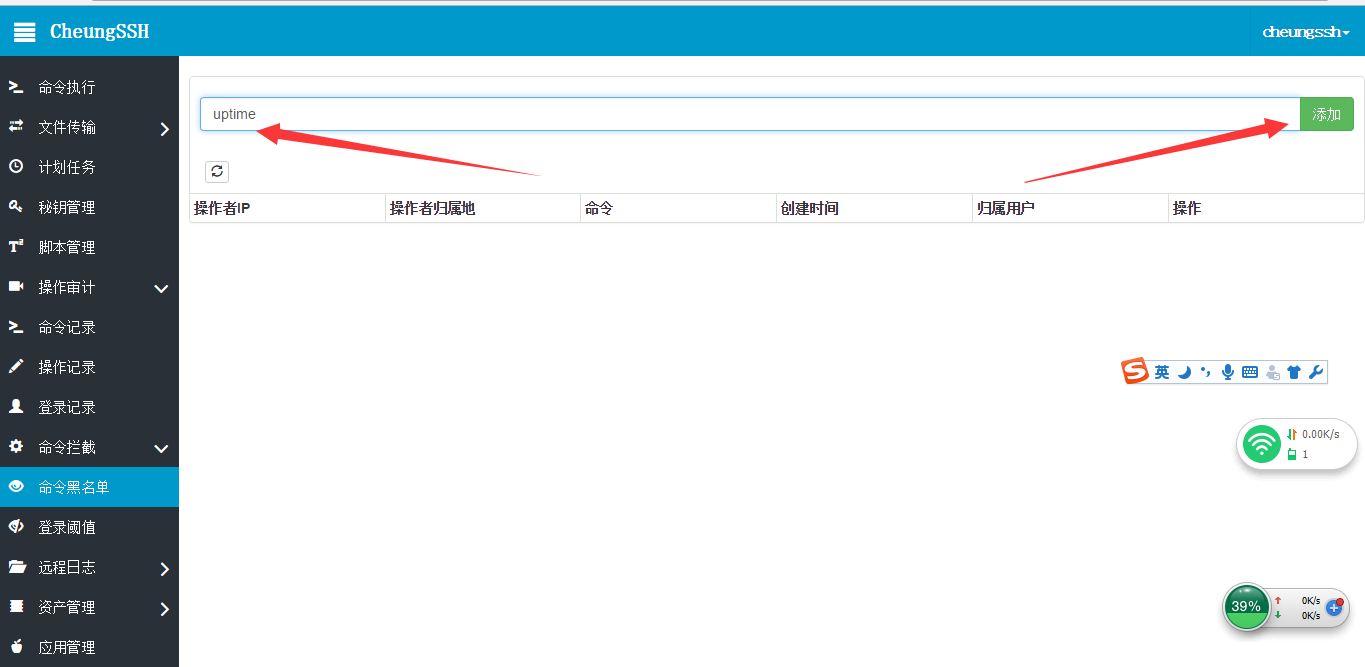
****

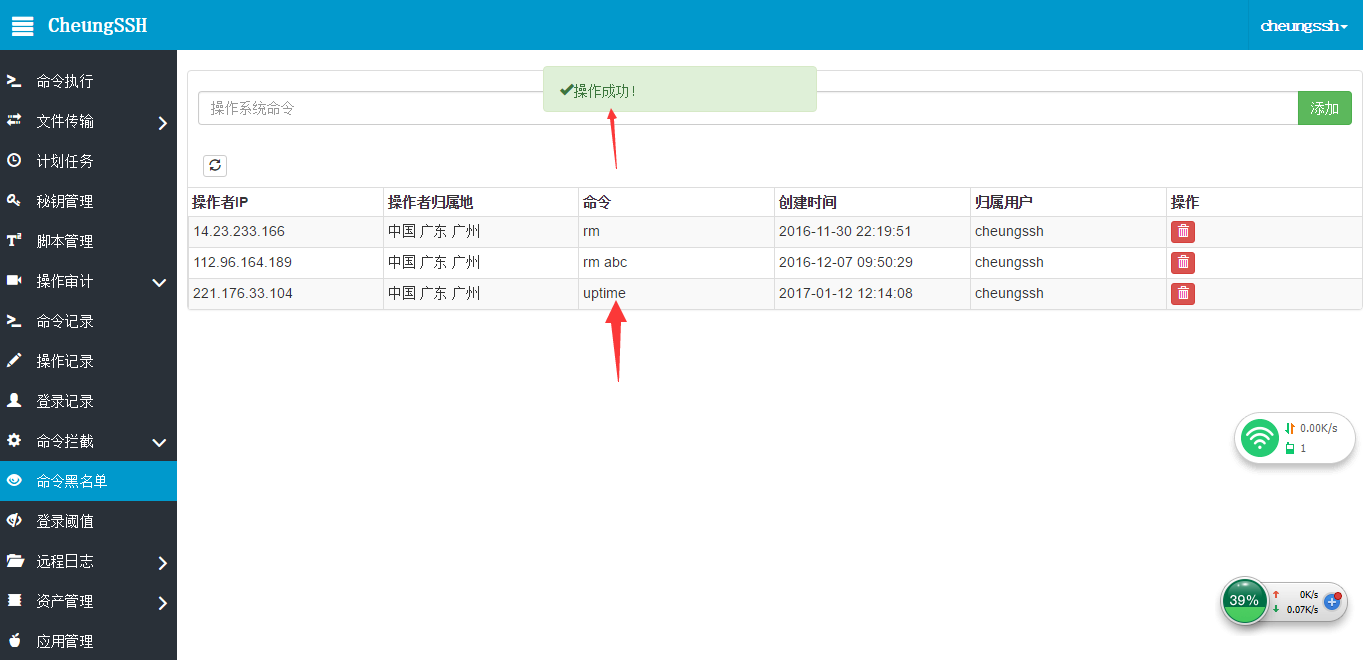
**10.1 命令黑名单**

在传统的SecureCRT、Putty、Xshell客户端中，均可以执行任何命令（除非Linux权限限制的），这样往往存在一些安全隐患，比如root错误的执行了 shutdown、init、rm等高危命令，而无法拦截导致系统崩溃。CheungSSH针对这个情况做了封装，您可以把您认为是不可以执行的命令加入系统中，那么用户在通过本系统执行命令的时候，如果发现是拒绝执行的命令，那么则会拦截。

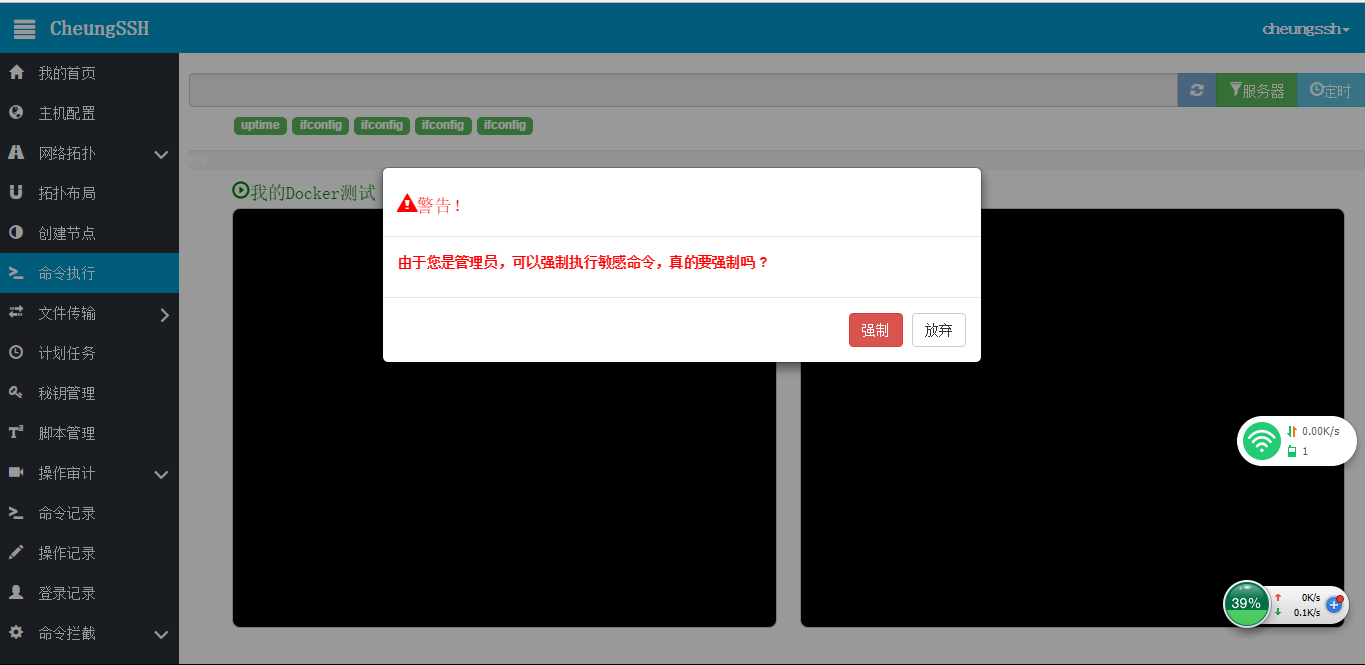
****

添加命令黑名单，比如 uptime

****

****

这样，在批量命令的时候，如果执行 这个“uptime”命令，就会被拦截下来，而不会被执行：

****

上面的显示很清楚，软件作者在开发的时候，考虑了如果您是超级管理员，依然是可以执行任何命令的，但是需要确认执行。假如您是普通用户，而不是超级管理员，那么系统根本不会提示您是否需要强制，而是直接拒绝！

**注意：**超级管理员会提示是否强制，而普通用户则不会提示，直接拒绝。

**10.2 登录阈值**

为了登录安全，CheungSSH在登录验证的时候，如果一个IP连续登录的次数超过5次，那么CheungSSH系统则会拒绝这个IP的所有登录，而不管密码是否正确，就像是银行卡一样。

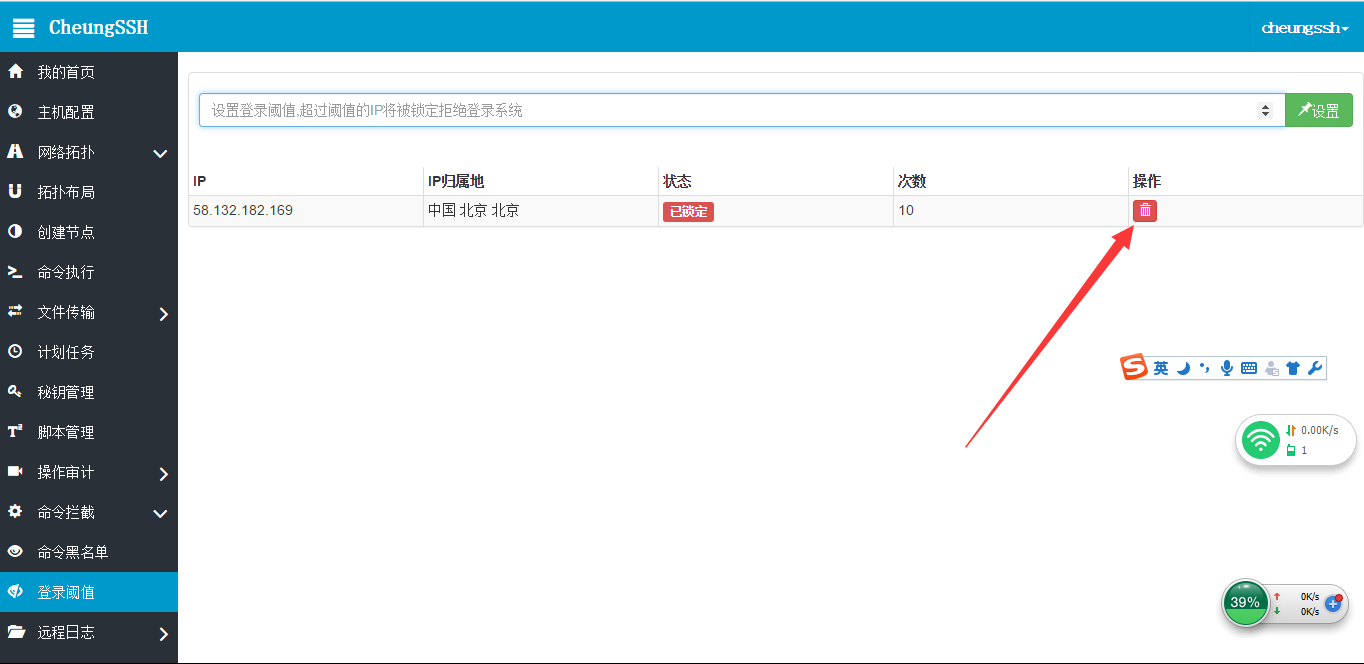
****

设置登录阈值：

****

**注意：**如果您的IP不幸被系统锁定，请您换一个IP重新登录，登陆后可以解锁该IP

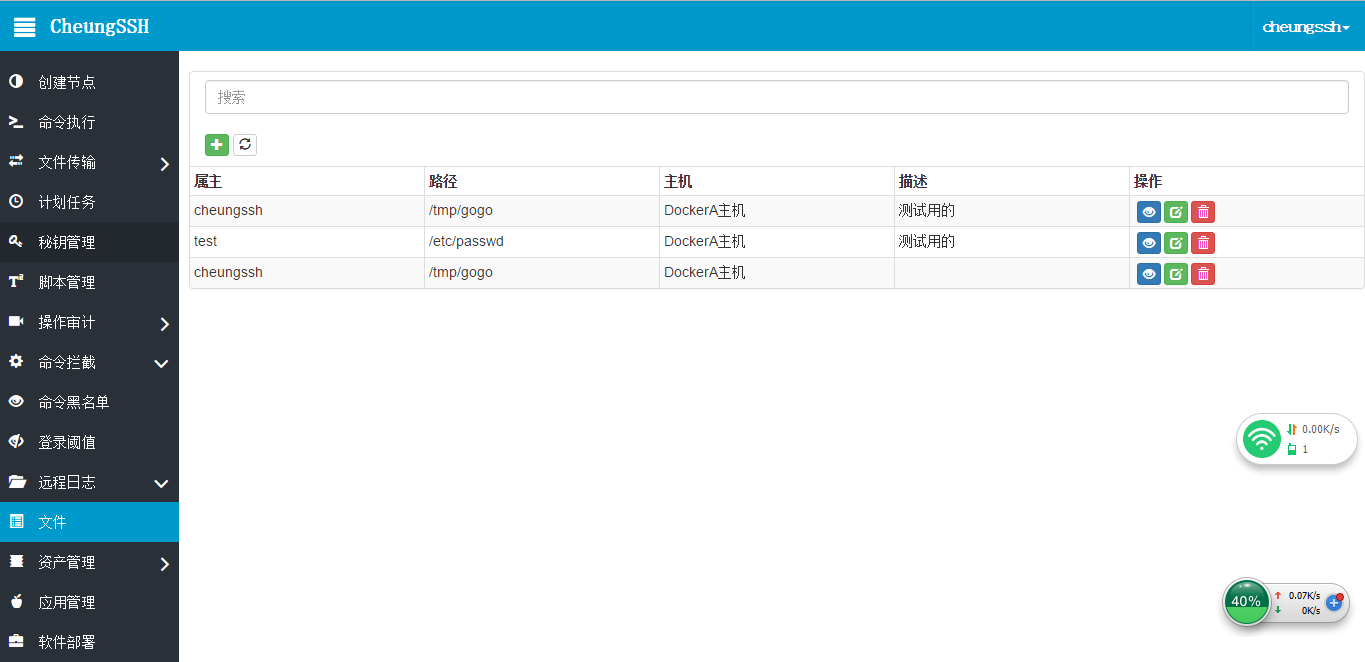
解锁IP

****

删除该IP即可登录。

**12.1 远程文件**

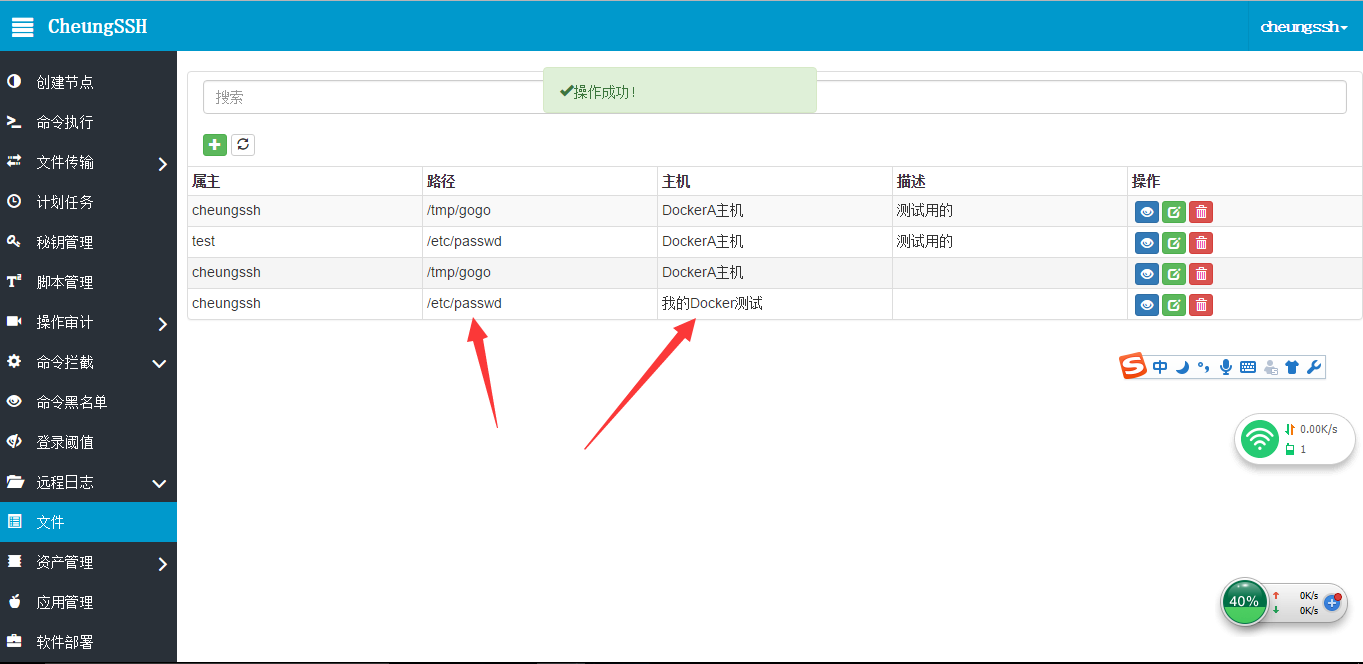
作者的设计初衷是为了让所有人都能方便的打开网页，就能查看Linux系统文件内容，而不再用cat、vim、more等命令去打开文件，这对于我们一个普通的研发或者是财务人员来说，显得太繁琐，因为他们很可能不会这些命令，其次还要有Linux密码等等条件，有了这个功能，您甚至可以用iPhone或者是SAMSUNG、HUAWEI等等手机的浏览器，直接看文件内容，当然不管是文件还是日志，只要是系统存在对的一个文件，都能看。



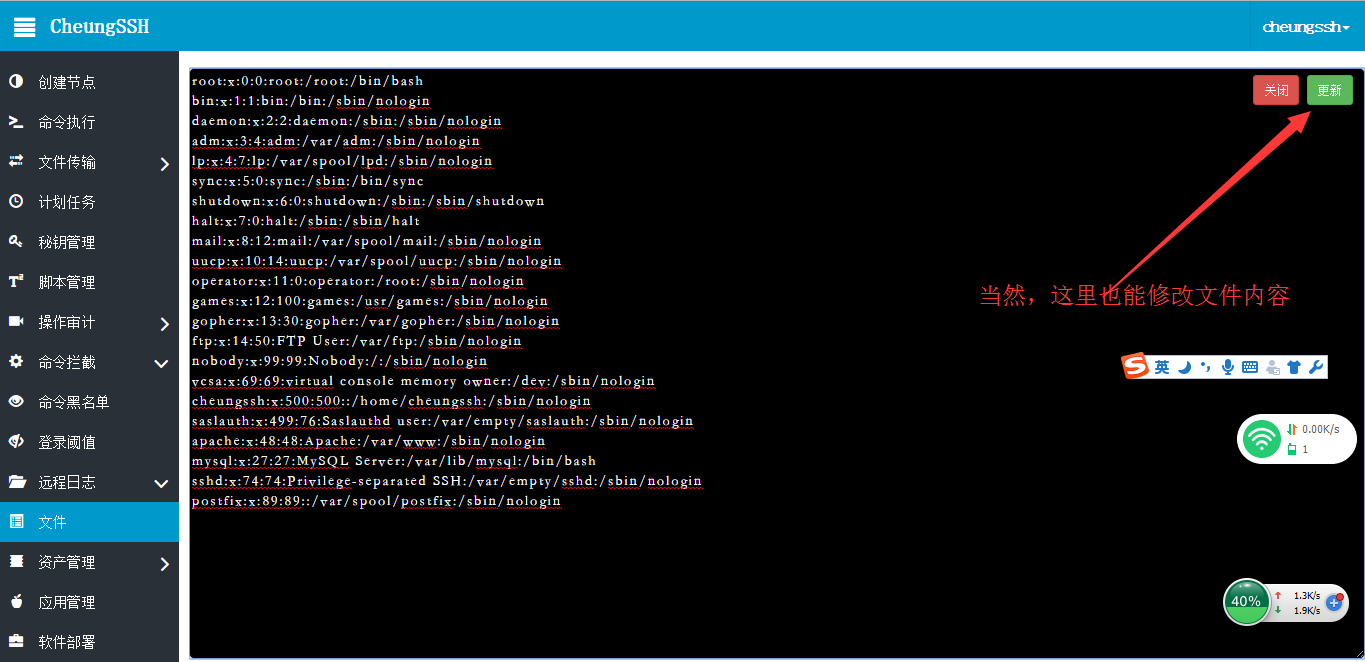
首先，您需要添加一个您要在什么服务器上查看什么路径文件的条目：



保存好了以后，就能看到如下条目：



现在，您可以点击进行查看服务器“我的Docker测试”上的“/etc/passwd”的文件内容了：

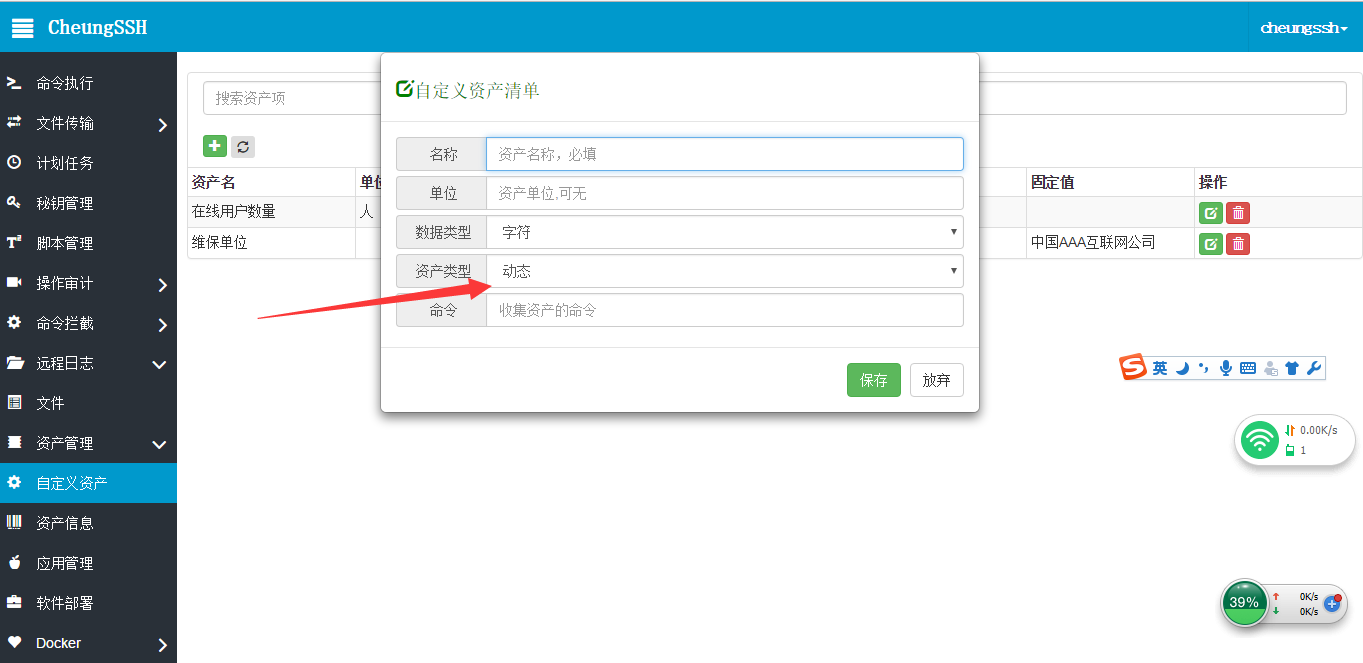


**注意：**这里之所以能选择服务器，是因为在【主机配置】功能中，已经添加过这个服务器

**13.1 自定义资产**

为什么有自定义资产呢？是这样的，CheungSSH作者在设计的时候，只是象征性的加入了某些资产要素进行采集，比如磁盘、CPU、内存等，但这些往往不能适用于每一个用户的需求，比如您可能需要采集Tocmat的连接数、在线用户数量等等不尽相同的需求。那么在这里，为了能让您的体验更好，CheungSSH作者加入了【自定义资产】功能，也就是说，您可以通过自行定义一个命令，比如您要统计Tomcat的连接数，它是去扫描一个文件内容然后进行各种计算，那么对于系统来说，它就是一个命令去完成的。所以，您只需要把您进行统计的命令加入即可，这就是自定义资产功能。然后，系统最终得到一个数字，录入系统。





输入项解释：

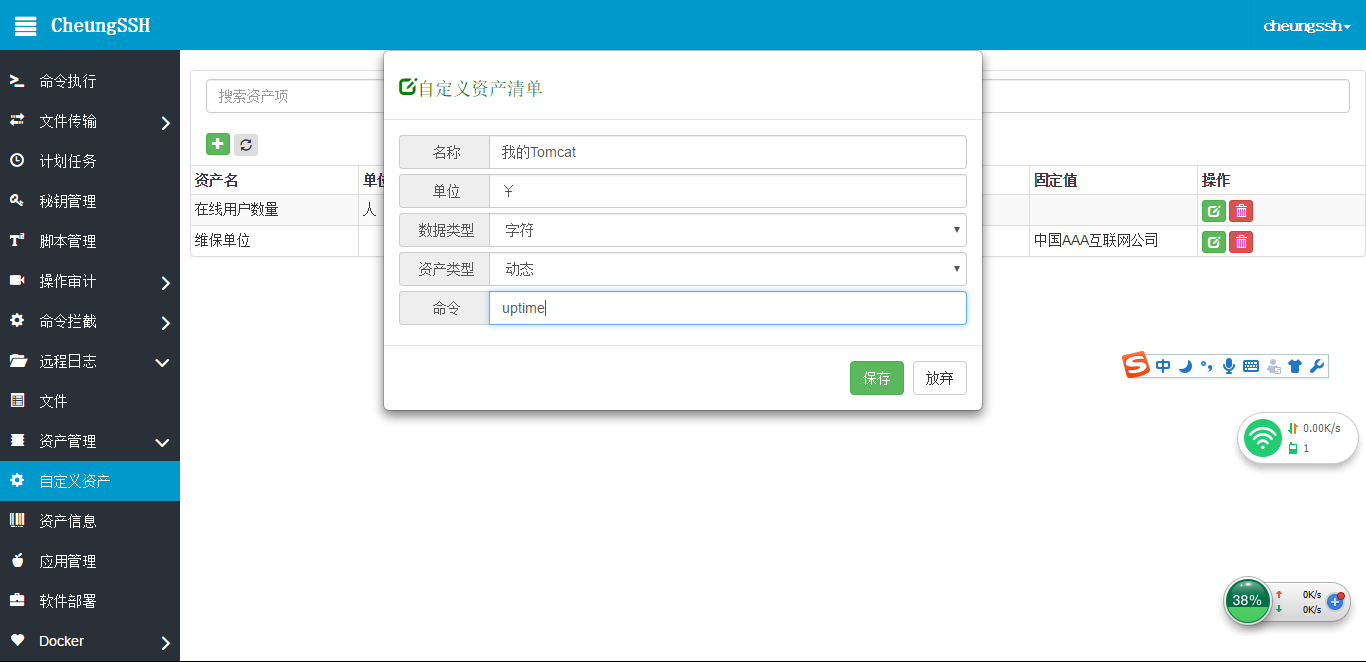
【名称】就是您定义一个资产的名称，比如“我的Tocmat”；

【单位】您随意取一个单位，这个单位会在【资产信息】的表头中显示，如果没有，您就不写；

【数据类型】就是您这个采集的数据是什么类型的，一共三种选项：数字，表示一个数字，比如链接数量，就是一个数字；日期：就是日期；字符：就比如ABCD这样的数据就是字符，您在这里准确的选择，讲有利于系统自动计算。比如数字类型的资产，系统**自动进行折现绘图。**

【资产类型】分为静态和动态，CheungSSH作者在设计的时候曾考虑过这样的情况，比如“Tocmat的连接数量”，这一定是一个命令查询的结果，也就是说是执行命令得来的；再比如“维保厂家”，这是一个固定的标记，而不是命令得来的。所以作者把需要执行命令得来的数据称为“**动态资产**”，把不用命令执行得来的数据称为“**静态资产**”。

【命令】也就是用来采集数据的命令，当然，如果您的是静态资产，这里就不要写命令了。



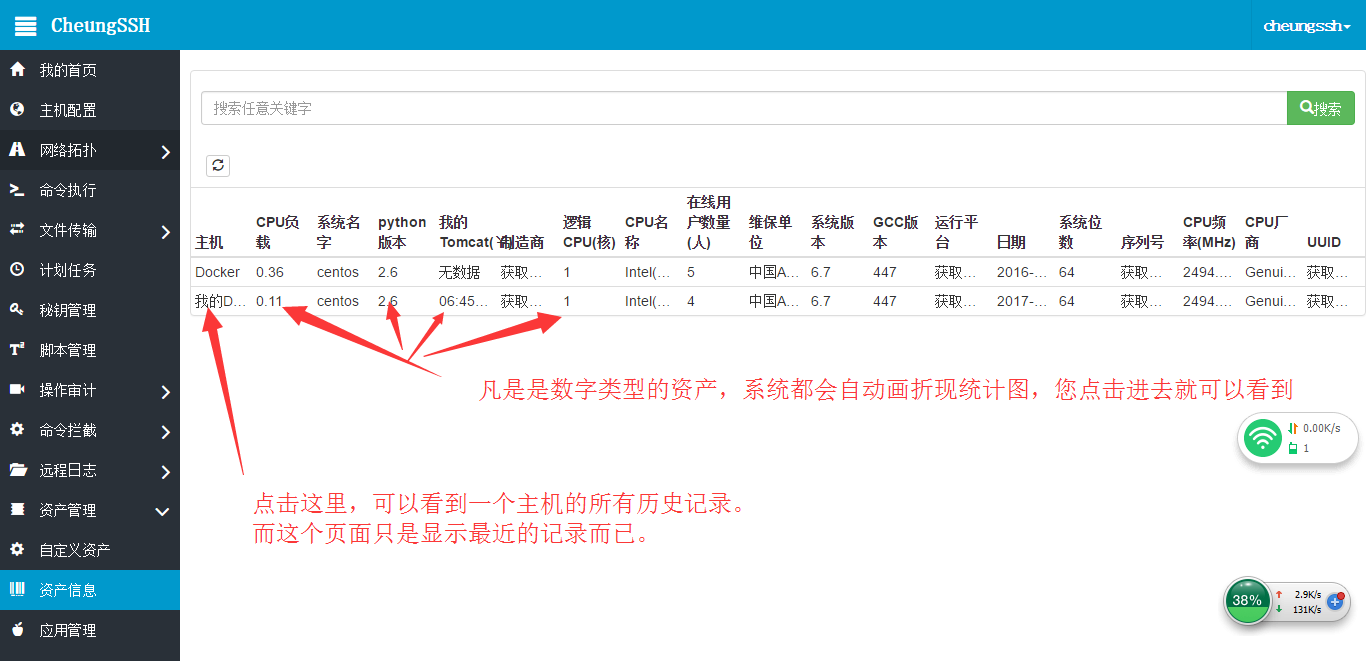
保存即可：



**注意：**CheungSSH系统后台自动采集资产，间隔时间是30分钟一次，您也可以自行修改

**13.2 资产信息**

上面讲述了自定义资产，当然。不管您是否需要定义自定义资产，那么系统都会每间隔30分钟后去自动采集数据，采集的数据就会显示到这里来。那么被采集的服务器是从哪里来的呢？当然是您在【主机配置】的时候，输入了主机的账号密码等信息，CheungSSH系统默认就替您采集了。



**注意：**被采集资产的服务器，账号必须是root权限，否则无法采集，因为有些信息不是root账号就无法采集，至于为什么，这个我想您应该明白。另外，如果您发现采集的时间不是最近的，那么很可能是被采集的服务器出现了什么问题，比如账号密码不对了，网络不通了等等。

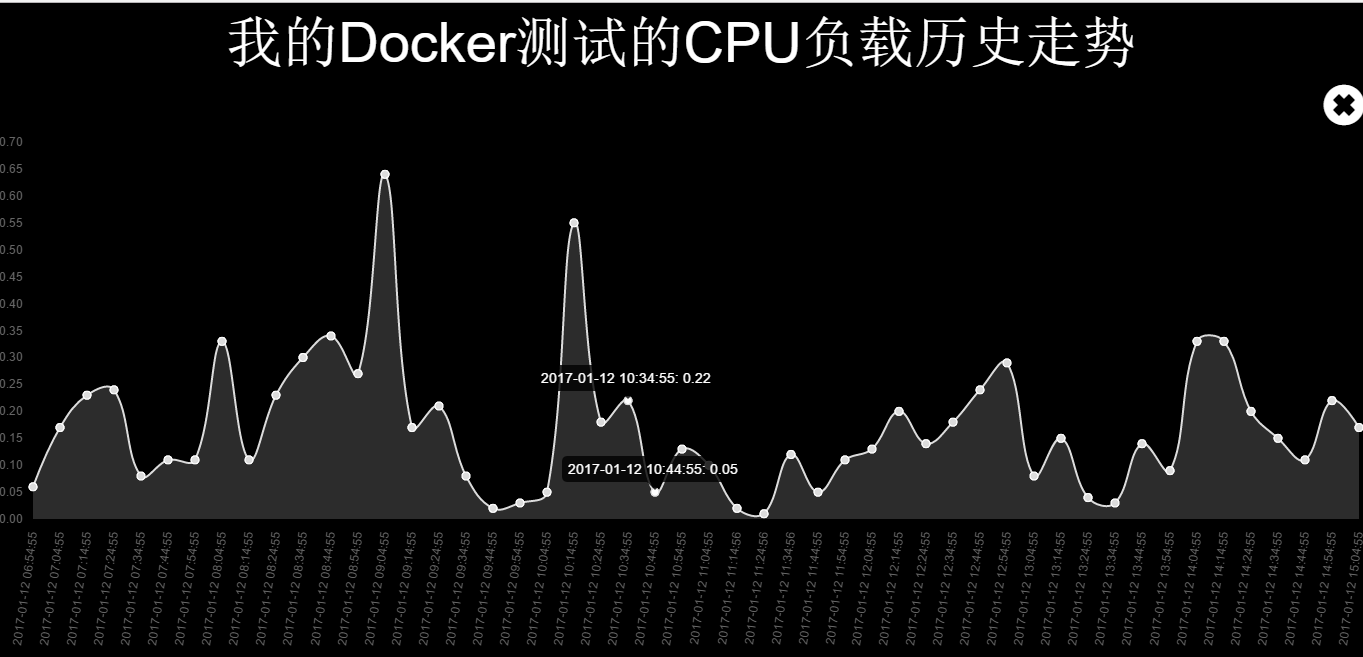
**13.2.1 每一个资产的历史记录**

在第一个资产页面中，那里看到的是每一个资产的最近统计数据，如果您要开一个资产的历史统计数据，那么请您点击它的主机名，然后进入另一个页面中：



**13.2.2 查看数字类型的资产折线图：**

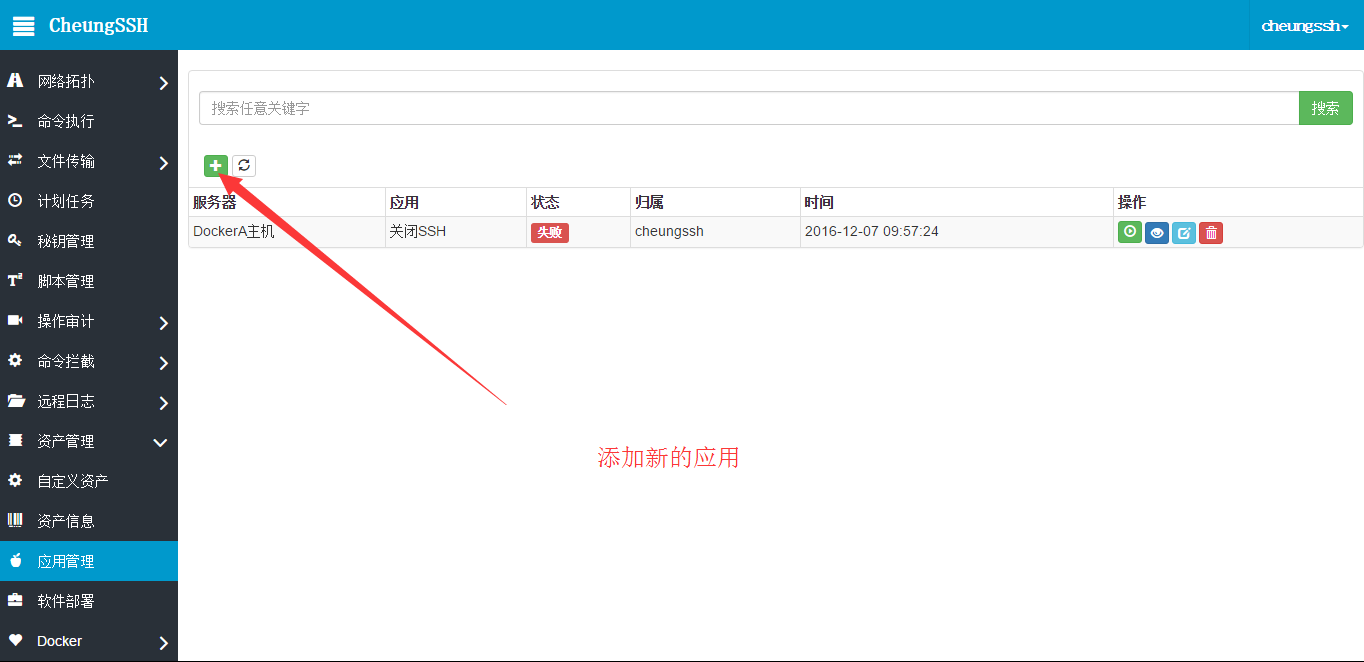
点击是数字类型的项，就可以进入如下页面：



**注意：**历史资产数据点，默认最多是50个

**14.1 应用管理**

还记得您常用的 service httpd start 、service sshd stop这样的命令吗？这些都是在停止启动服务，为了让研发人员，或者是快速操作服务，CheungSSH作者在这里给大家制定了简单的应用管理。在这里，您可以把上面的命令加入进来，然后平时只需要点击一下按钮就能执行了，是不是方便了很多呢？





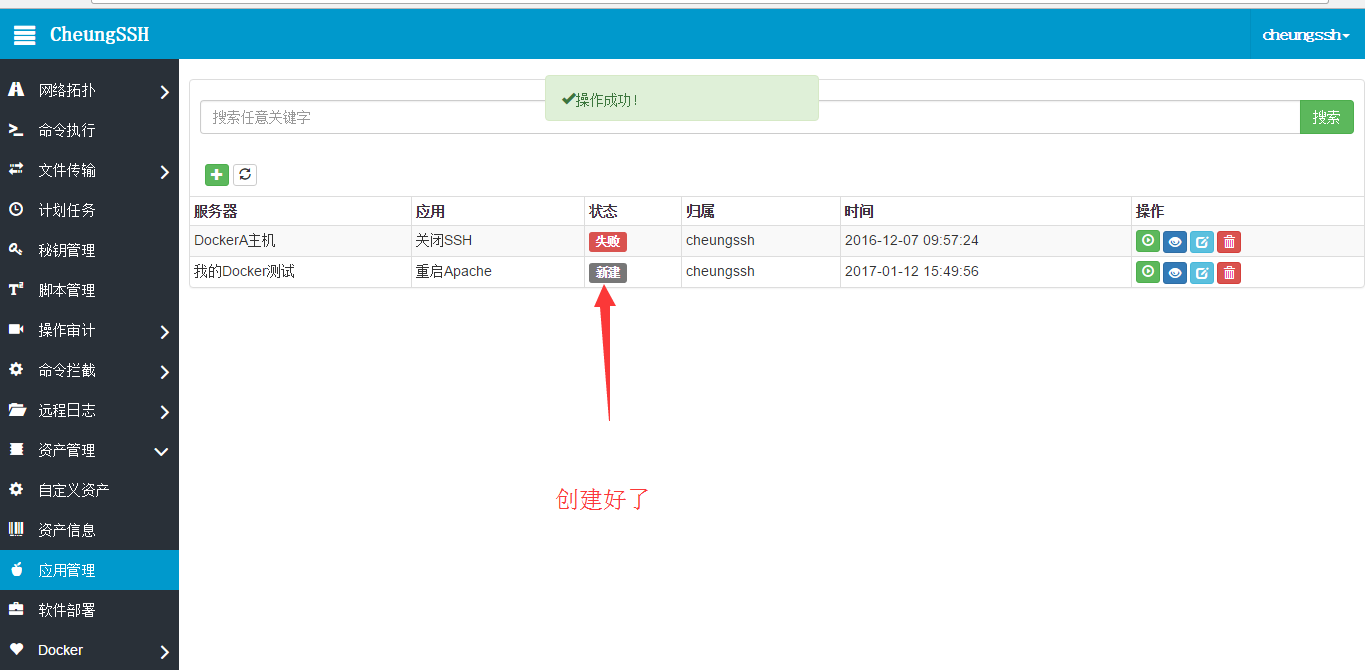
选项解释：

【服务器】就是您要在哪个服务器上执行这个命令；

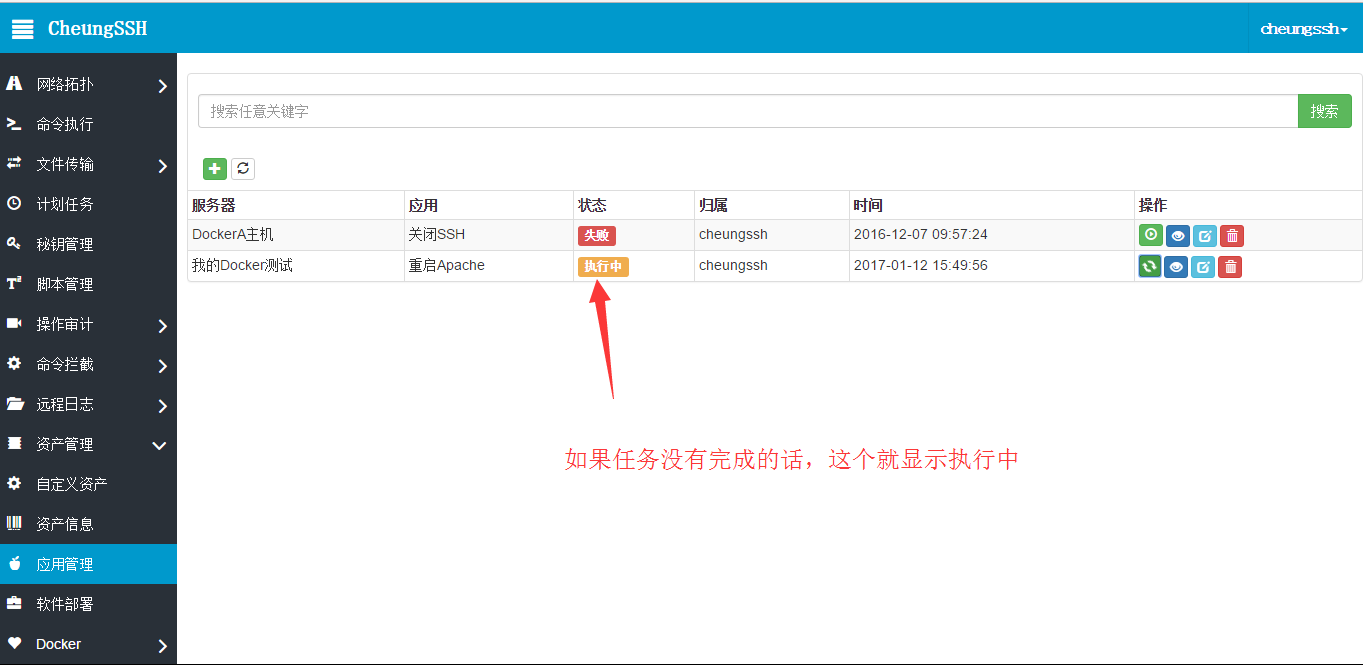
【应用名】给这个应用起一个名字；

【检查命令】默认是echo $? 比如启动Tocmat的时候，可能需要检查网页才能判定，那么您就在这里输入您的命令就行了。系统根据执行的命令echo $?是否返回0进行判断是成功还是失败；

【归属用户】哪个用户可以操作，当然了超级管理员无限制；



然后点操作栏的 就能执行这个命令了，当然不要 误会这个操作按钮是“启动”的意思，它表示的意思是执行这个应用的命而已。

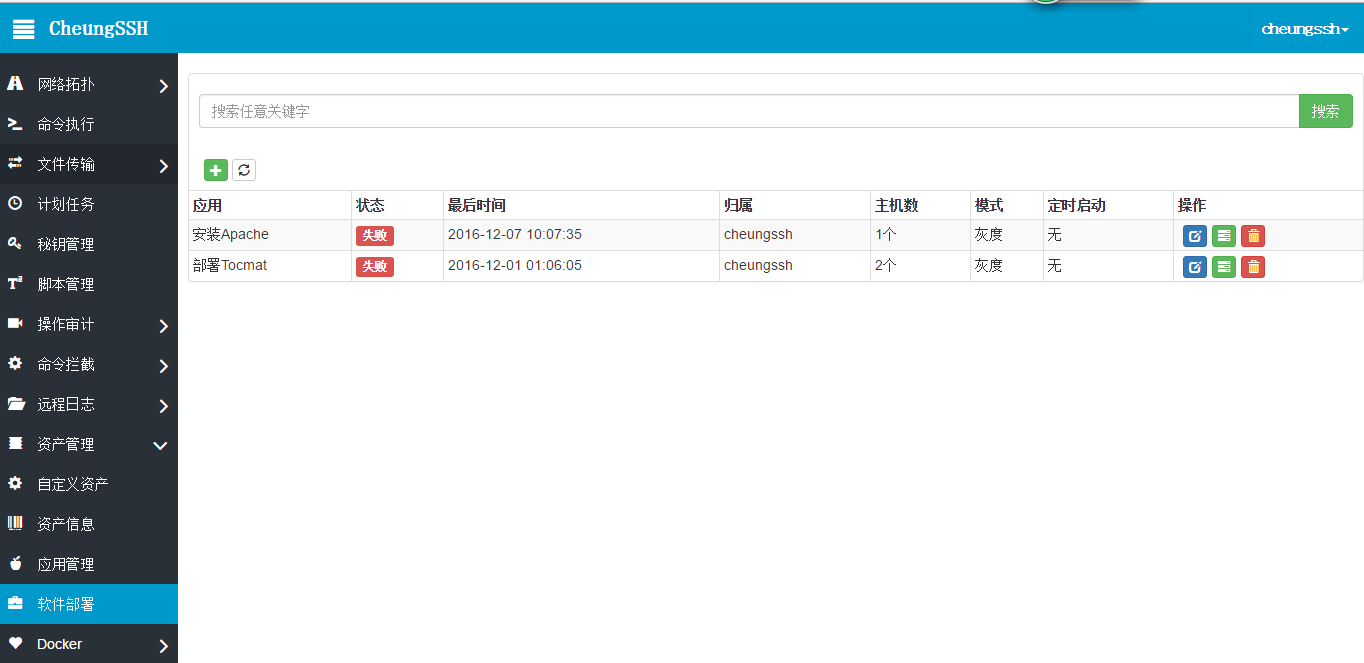


执行完毕后，我们看下结果：



**15.1 应用和软件部署**

首先，本功能不仅仅是应用部署，更可以软件安装、应用部署，它就相当于是某软件的YML功能一般，可以自行定义要操作的步骤。所以，不管您是Tocmat应用代码更新上线，还是Apache软件安装，等等功能操作，都可以完成。



不知您是否做过Tomcat 代码更新上线，安装过Apache、MySQL等软件，DB备份。像这些操作往往都不是一条命令可以完成的，都是多个命令完成，并且有的时候还有必要判断上一个命令是否执行成功，如果失败了的话，还需要立即终止并且需要回退。像这样的操作，CheungSSH完美的解决！

首先，在这里假设第一个实际案例，比如Apache软件安装。现在，粗略的梳理一下安装软件的准备条件和过程。

**Apache自动安装实例**

**准备软件包**

* http-2.4.25.tar.gz 一个
* httd.conf 配置好的一份
* 假设无依赖，并且已经安装好了gcc编译环境（万一没有gcc，就yum install –y gcc\*）

**被执行任务的服务器**

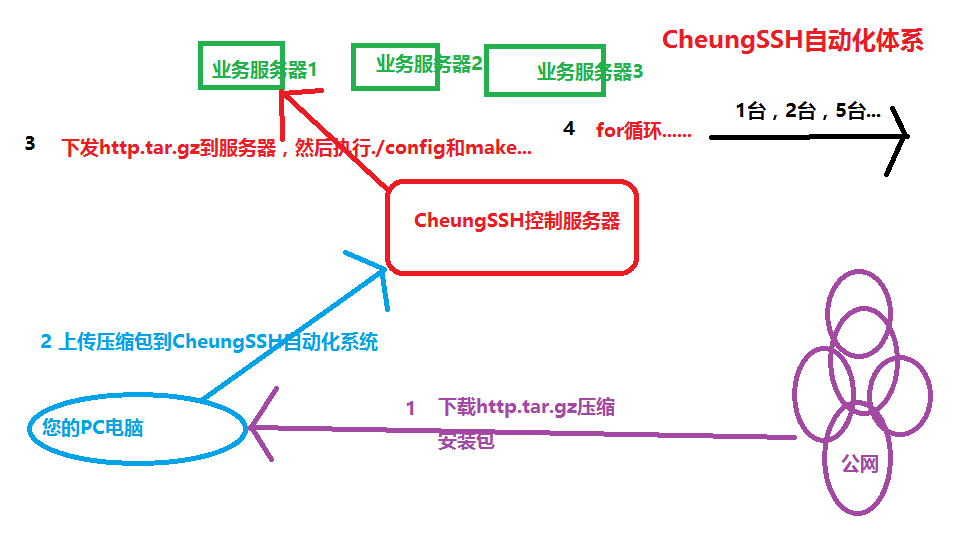
* 数量自定义，可以多个（灰度发布模式，也就是逐个服务执行）

**操作命令步骤梳理（**每一个步骤执行必须成功，否则终止**）**

1. 从您的PC上传软件包http-2.4.25.tar.gz到CheungSSH服务器上（当然，您得先在互联网上下载这个软件到您的PC电脑上）
2. 从CheungSSH服务器上传http-2.4.25.tar.gz软件包到被安装软件的服务器上
3. 在CheungSSH服务器上，解压那个软件包
4. 在CheungSSH服务器上，进入解压包目录，执行./config程序开始安装
5. make && make install
6. 移动准备好的配置文件httpd.conf到指定位置
7. 执行启动命令
8. 完成

**注意：**作者只是为了演示使用方法，您没必要根据作者的步骤来安装apache，请您务必根据您当前服务器环境决定您的安装流程。

**安装流程图**

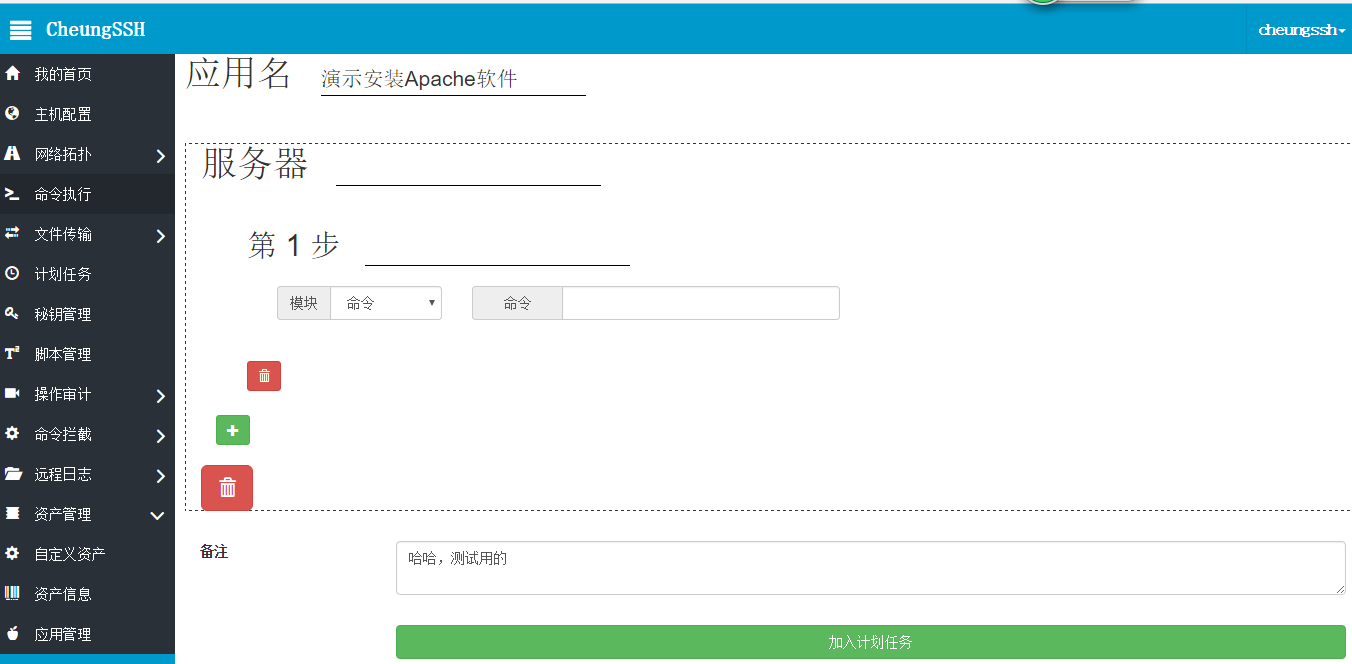
****

**注意：**上面只是作者本人安装Apache的举例，如果您的安装步骤不相同都没关系，您完全可以自行定义。在这里只是为了演示而已。

现在，开始创建任务：



点击添加步骤任务：

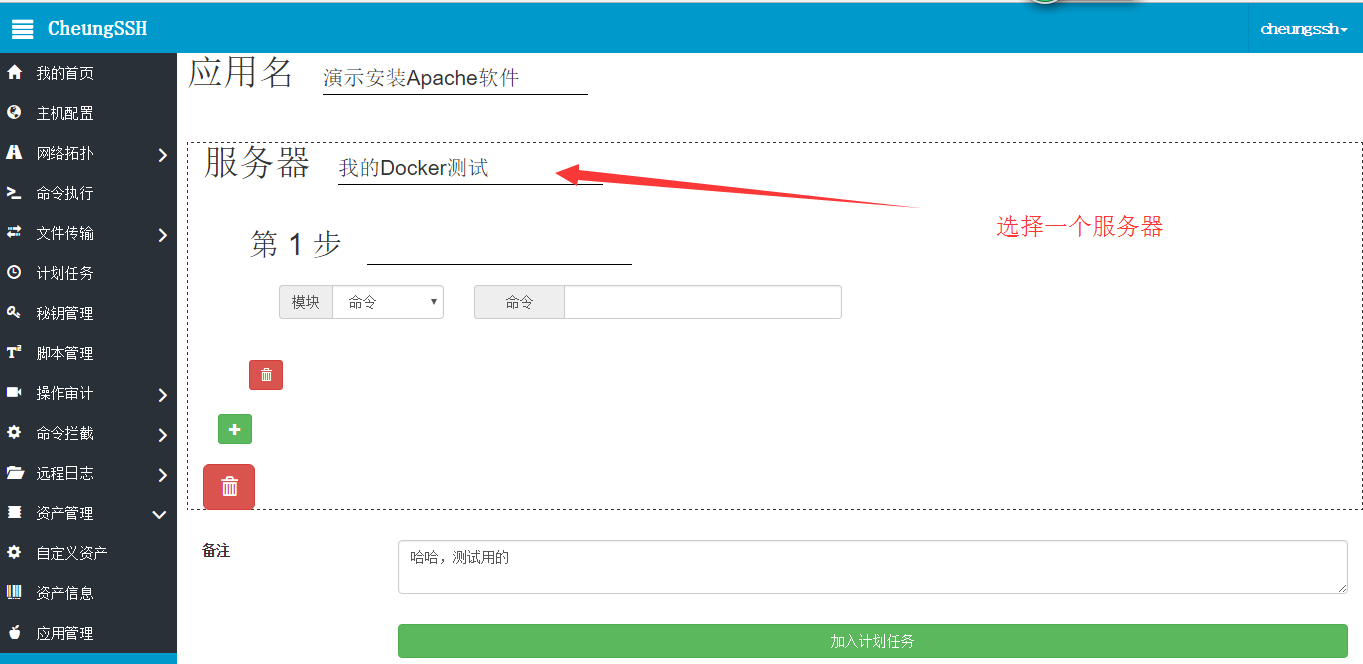


现在，可以看到页面中出现了一个服务器，还有第一个步骤，那么先选择一个服务器吧，什么意思呢？就是要选择一个服务器，然后在这个服务器上面执行我在上面梳理的那些步骤。

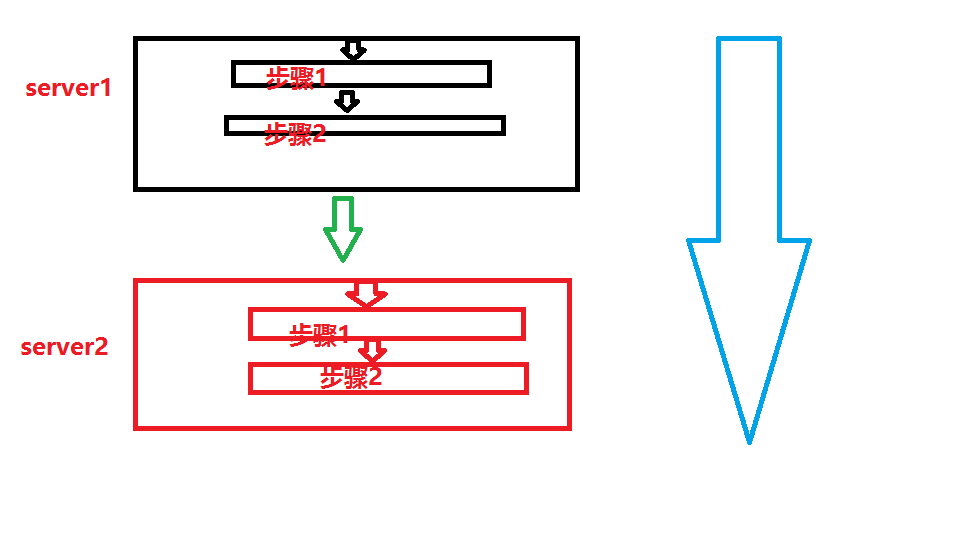
点击“服务器”旁边的下划线，然后弹出服务器选择界面，选择一个服务器



**注意：**复选框只能选一个服务器，不能选多个。如果要选多个，就需要添加多个服务器步骤。



现在已经选择好了一个服务器了，那么就应该给这个服务器设定执行任务的过程图（梳理一下过程）

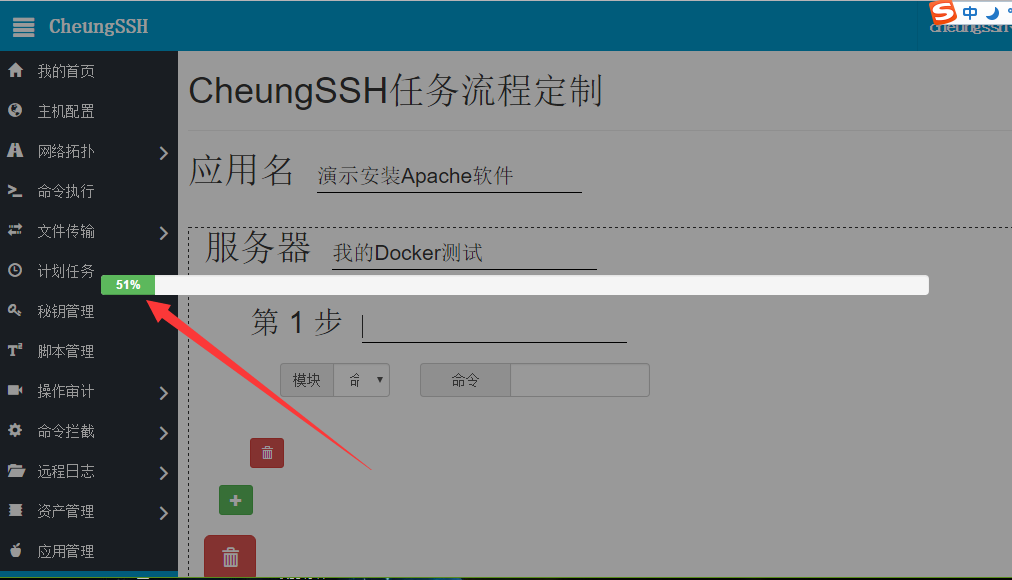


可以看到，是分了服务器的，而且每一个服务器中有N个步骤，执行完了一个服务器的N个步骤以后，再继续执行下一个服务器的N个步骤，直到执行完全部服务器的步骤。为什么要这样设计呢？就是考虑到，您的一组业务服务器，比如Tomcat的安装目录不是完全相同等因素。CheungSSH作者只是为了您能灵活使用。当然，这也会带来一个问题，假如有100个服务器需要操作，那么就要添加100次服务器了。这个比较繁琐，后续会升级，增加一个复制步骤的按钮和调整步骤顺序的按钮。

**注意：**执行过程中，如果遇到任意一个步骤是失败的，那么都会终止后续步骤的执行，为了保证安全嘛。

我们按照上面的执行步骤逐个创建：

* 第一步 从您的PC上传软件包http-2.4.25.tar.gz到CheungSSH服务器上（事前已下载好该软件），您可以直接拖动文件到网页中，直接上传



* 第二步 在CheungSSH系统上，上传刚才的软件到远程“我的Docker测试”服务器上去（也就是相当于我们常用的scp命令一样）



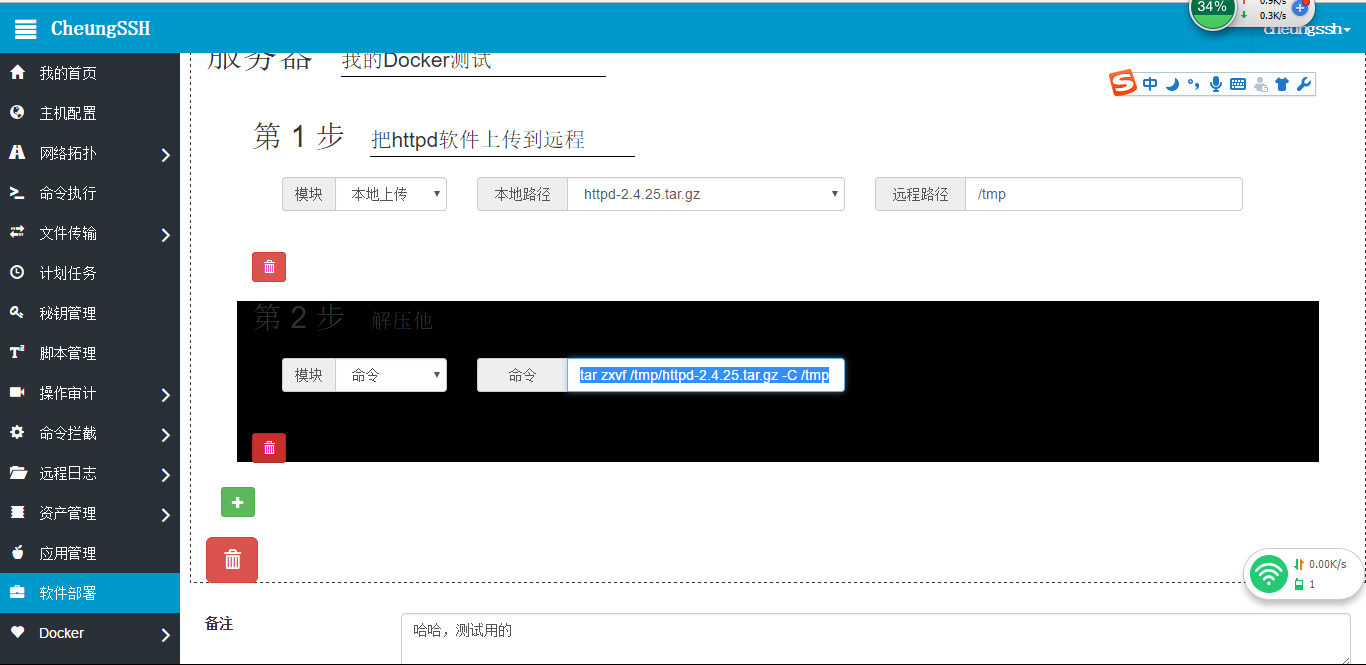
创建完毕第一个步骤，看看效果：



* 第三步 解压前面上传好的httpd文件（当然了，此时还没开始执行任务，只是预先写好而已）



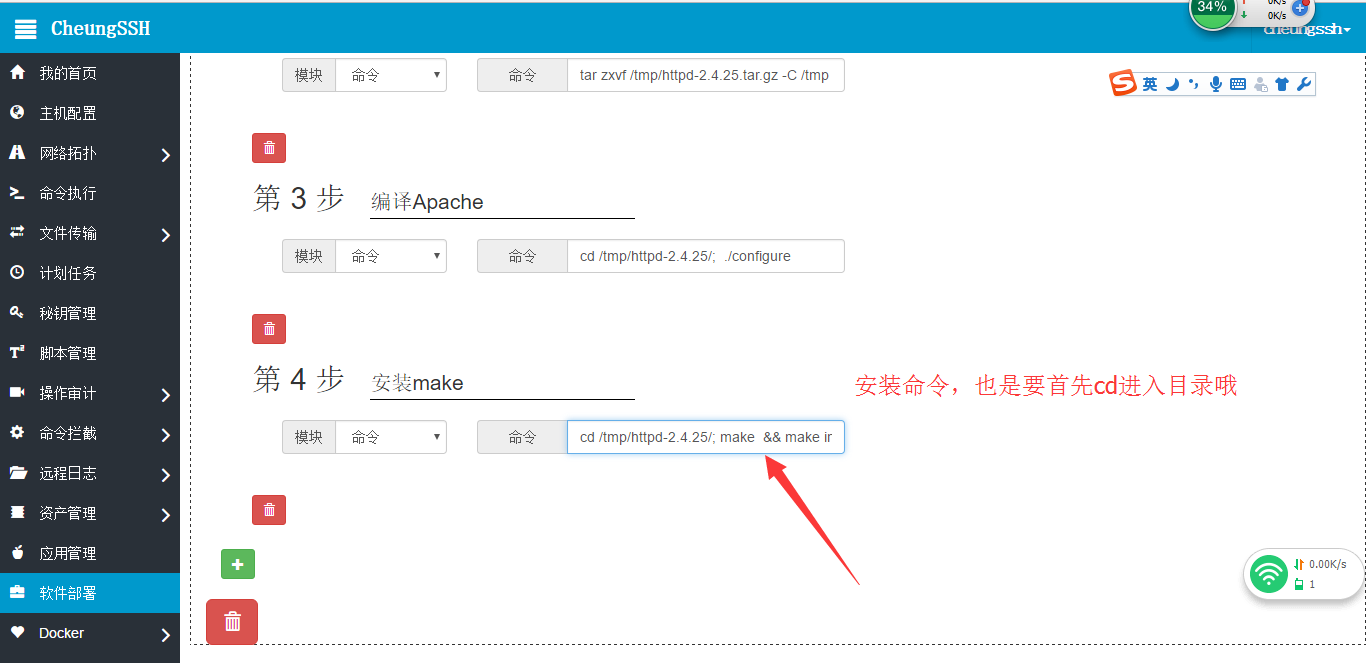
另外，如果您想删除某一个步骤，可以点击，如图，变黑的部分就是要删除的部分（当然，这里是不删除的）：



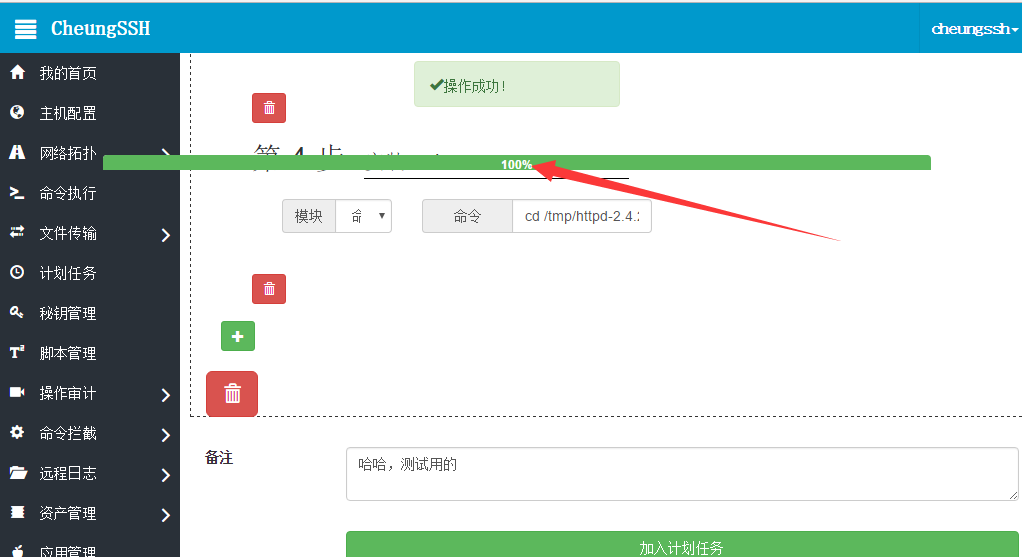
* 第四步 开始编译Apache（至于编译的命令，根据您个人的需求而定，作者为了在这里演示，写的简单）



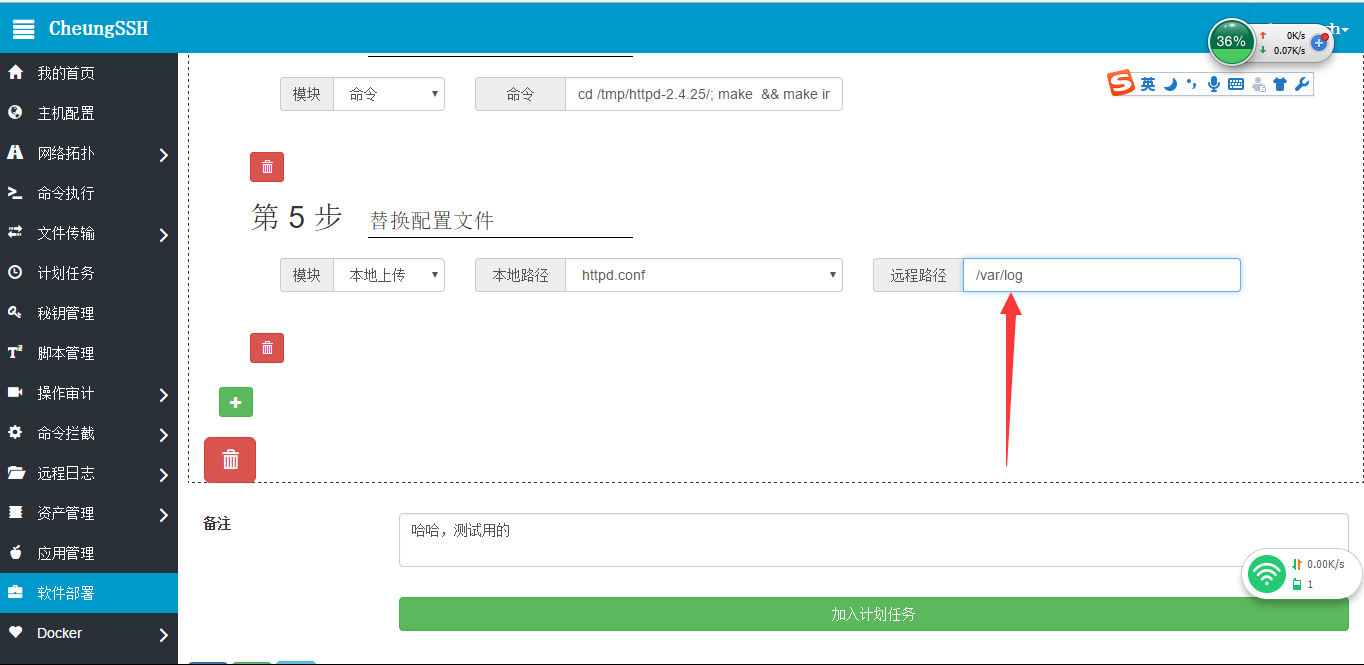
* 第五步 执行安装make 和make install



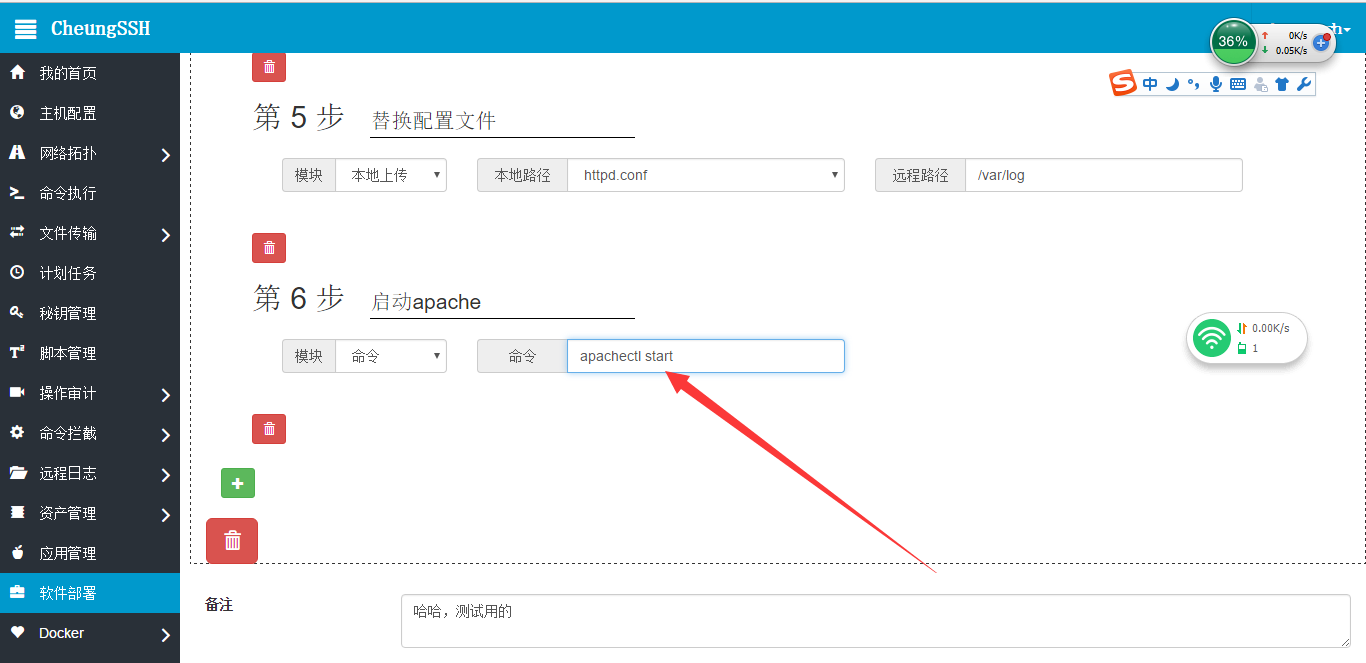
* 第六步 上传准备好的httpd.conf配置文件（也是拖动上传就行了）



* 第七步 从cheungssh上传刚才的httpd.conf文件到远程服务器上

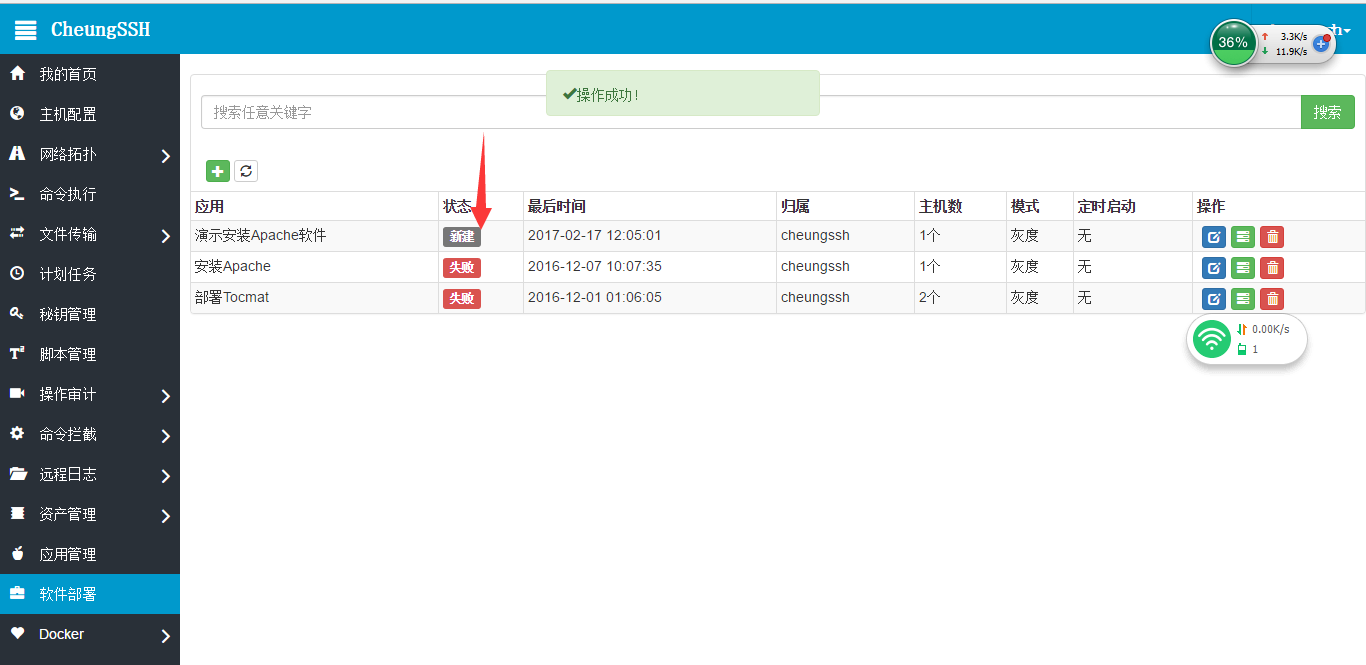


* 第八步 执行apache的启动命令（apachectl start）

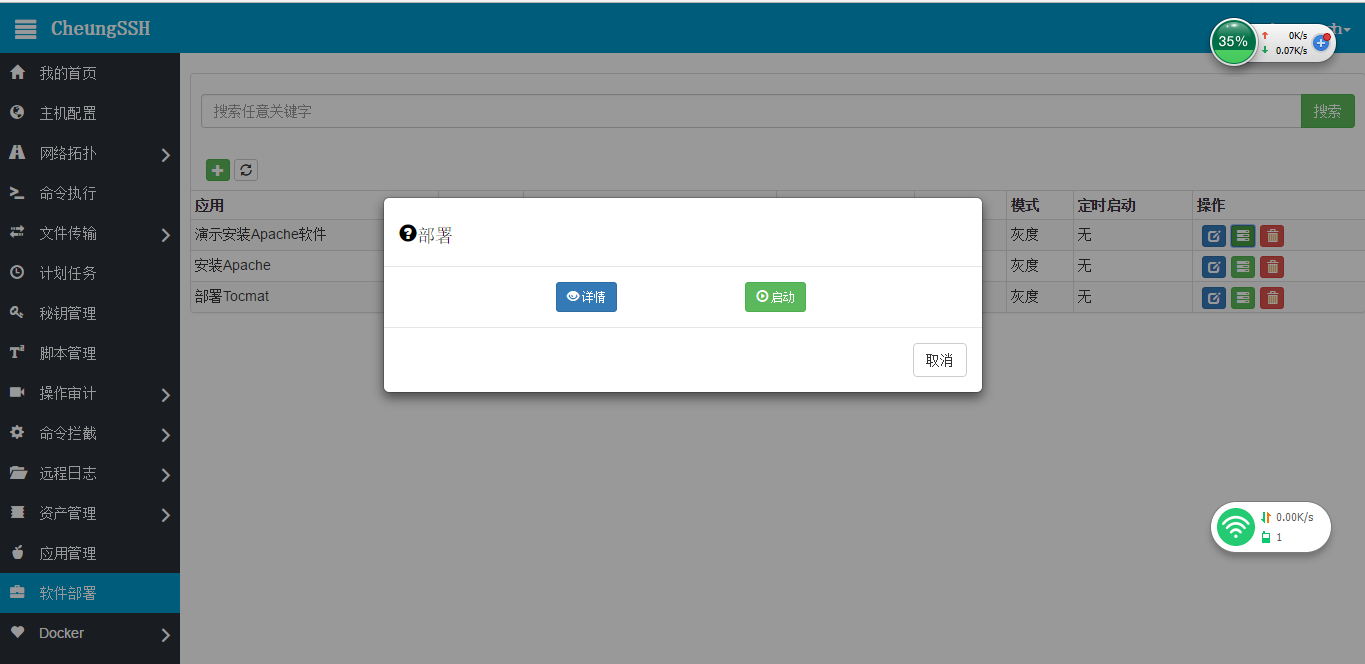


* 第九步 如果您有N个服务器，那么请您循环上面的步骤流程即可。

现在，请点击  保存这个任务。



现在，已经创建好了任务，可以点击执行任务了。请点击，如图所示：



**关于【详情】的功能解释：**

该功能是指在任务执行前，任务执行中，任务执行后均可查看任务状态的功能，他不会启动任务，只是看任务的状态而已。

**关于【启动】的功能解释：**

该功能是用来启动任务的，启动任务后，您可以在线等待任务的执行，当然也可以关闭网页，以后再回来重新打开网页，通过【详情】功能来查看任务的执行状态，该功能是后台执行的，所以不必关心页面是否关闭。

现在，我们要启动这个任务，请点击 进行启动：

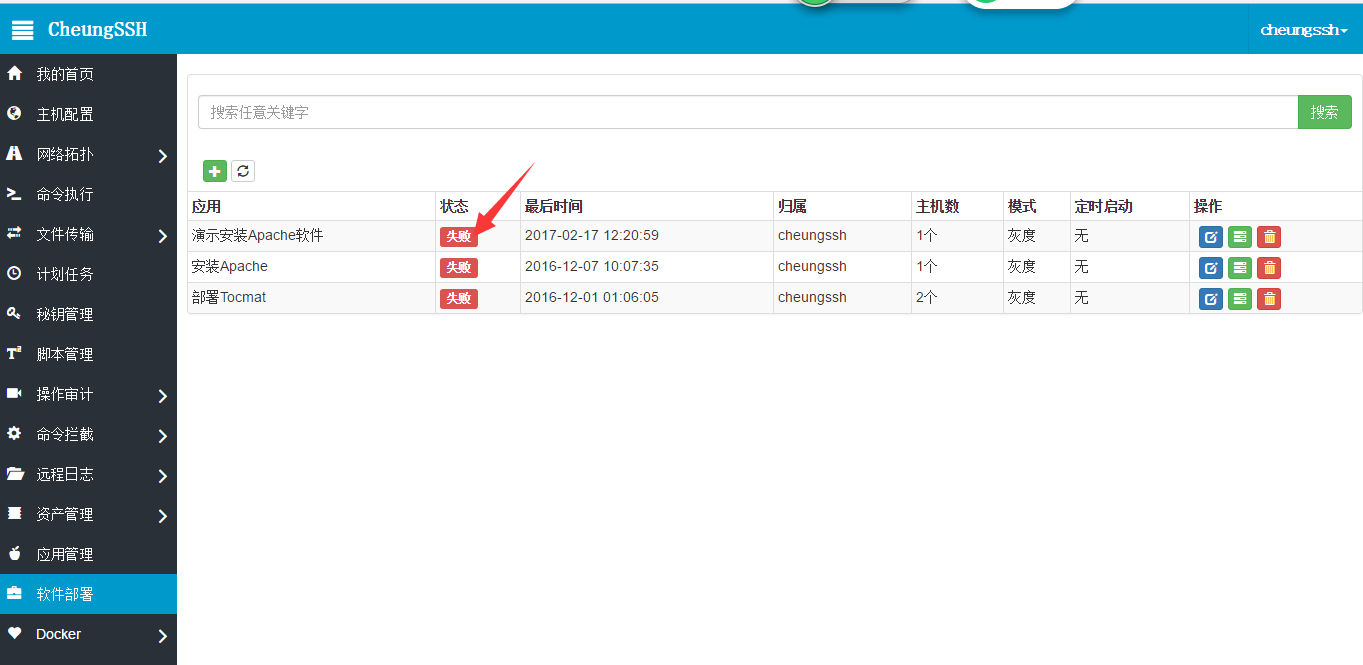


上面有出错了，可以点击它，看看详情是什么：



看到错误消息，因为APR的问题，这里就不在继续解决这个依赖的问题了。我想，我已经完成了演示的目的了。

最后，返回主页面，可以看到任务的状态已经标记为了【失败】状态了：



**16.1 Docker镜像自发现**

如果您在【主机配置】中，添加主机的时候使用的是root超级管理员，而且该主机中确实有docker镜像存在，那么CheungSSH系统会自动发现该列表，并且展现在网页中



自动发现的时间，间隔是30分钟一次。在该页面，您可以自行操作docker镜像功能。

**16.2 Docker容器启动**

选中容器，请点击 即可启动。

**16.2 Docker容器停止**

选中容器，请点击 即可停止容器。