**CheungSSH使用手册发布**

--原著：张其川

--版权所有：张其川

目录

[一、 CheungSSH产生的理由 2](#_Toc421483712)

[二、 下载CheungSSH并安装 5](#_Toc421483713)

[1. **使用平台介绍**： 5](#_Toc421483714)

[2. 安装环境要求: 5](#_Toc421483715)

[3. 模块支持： 5](#_Toc421483716)

[4. 下载CheungSSH： 5](#_Toc421483717)

[5. 安装CheungSSH： 7](#_Toc421483718)

[6. 初始化CheungSSH： 8](#_Toc421483719)

[7. 配置介绍： 9](#_Toc421483720)

[8. **配置样例**： 11](#_Toc421483721)

[**9.** **正式启动CheungSSH** 12](#_Toc421483722)

[10.让CheungSSH执行命令 12](#_Toc421483723)

[**1.初次执行命令**： 12](#_Toc421483724)

[**2.再试试执行命令**: 13](#_Toc421483725)

[三、 架构介绍： 14](#_Toc421483726)

[1． CheungSSH的总体架构设计 14](#_Toc421483727)

[2． CheungSSH的操作流程 15](#_Toc421483728)

[3． CheungSSH设计原理 15](#_Toc421483729)

[4. 功能表 16](#_Toc421483730)

[四、 功能介绍 20](#_Toc421483731)

[1. 命令行模式 20](#_Toc421483732)

[2. 命令行代码的实现： 22](#_Toc421483733)

[3. 快捷键的使用： 23](#_Toc421483734)

[**3.1** **cd切换目录** 23](#_Toc421483735)

[**3.2** **方向键** 24](#_Toc421483736)

[**3.3** 清屏操作 24](#_Toc421483737)

[**3.4** **History 命令记录** 25](#_Toc421483738)

[4. 日志审核 25](#_Toc421483739)

[5. 日志入库 27](#_Toc421483740)

[6. 原始日志（适合查看的日志格式） 29](#_Toc421483741)

# CheungSSH产生的理由

说起自动化运维工具，想必您或多或少听说过一些工具吧？如基于Ruby语言的Puppet、基于SSH协议的Saltstack、Ansible（Python开发）、以及Pssh，而目前市面上认为的是Ansible比Puppet好用

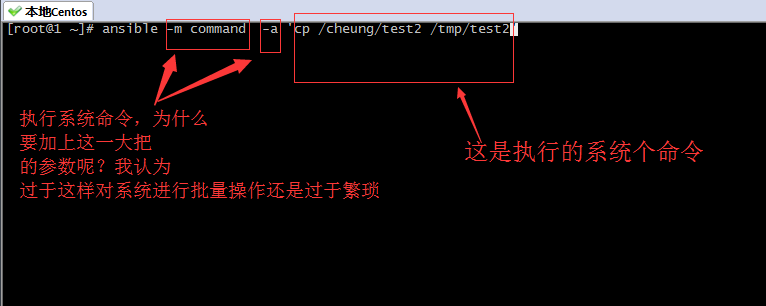
但是我认为：

* + - 1. Puppet的功能虽然强悍，但是我认为**操作过于繁琐，语法过于复杂** ，现在我们来简单的看看Puppet备份一个目录是如何配置的：

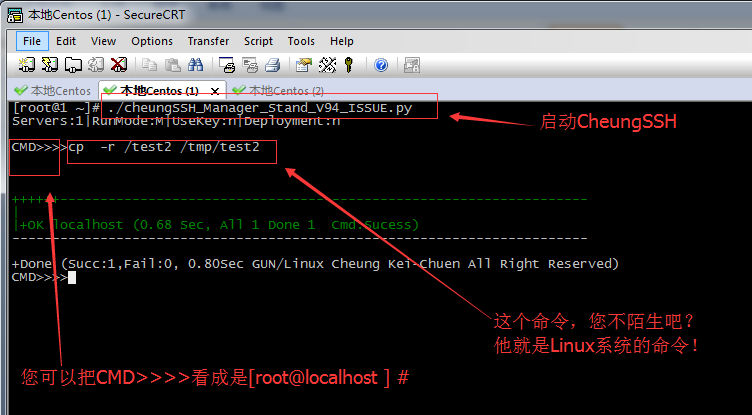


以上的配置，您看出是什么意思了吗？让我来告诉您吧，他其实就是把test2目录备份起来的一个操作，您觉得这个操作如何呢？

1. 现在让我们看看Ansible是如何对一个目录进行备份的配置和操作：



1. 现在让请您看下**CheungSSH**是如何执行系统命令进行备份的：



看完了以上Puppet、Ansible、**CheungSSH**的命令用法，您觉得哪个更适合您？哪个操作更简单，更能让您的双手快速适应？

4、就算您现在使用的是Puppet或者是Ansible，那么有朝一日，您觉得您的工作环境需要有所改变，而您自己又不懂Puppet和Ansible的设计，那怎么办？是不是就只能任由他们摆布？当然不是！假如，您选择了**CheungSSH**，那么您可以把您的需求告诉我，我来为您量身定做！当然，像这样的服务，您可能需要向我支付相关的服务费用。

5、**所以，CheungSSH产生的理由是：操作更简单(完全没有单独的语法)、技术支持更有保障（您可以把您在使用过程中的疑问与我交流）当然，重要的一点还有 CheungSSH是源码开发的，完全免费！任何人都可以下载自由使用。**

# 下载CheungSSH并安装

1. **使用平台介绍**：

**CheungSSH**只适用于Linux系统和Unix系统，因为是基于SSH协议工作的。当前最好的支持Redhat、Cnetos、Ubuntu、Suse等，但是对被管理的远程服务器没有要求，只要支持SSH协议即可。

1. 安装环境要求:

使用CheungSSH的系统Python版本不低于2.4，需要有gcc、python-devel环境支持。由于CheungSSH与Ansible一样， 是基于SSH协议的，所以您的Linux/Unix只需要开启SSH即可，无需安装软件其他任何插件或者是Agent，管理相当简单。

1. 模块支持：

由于**CheungSSH**是基于SSH协议工作的，所以需要paramiko和pycrypto的支持。这些模块都会在**CheungSSH**的自动安装中集成安装。无需其他安装。

1. 下载CheungSSH：

由于**CheungSSH是开放源码且免费的，所以您可以到GitHub下载：**

第一种下载方式：

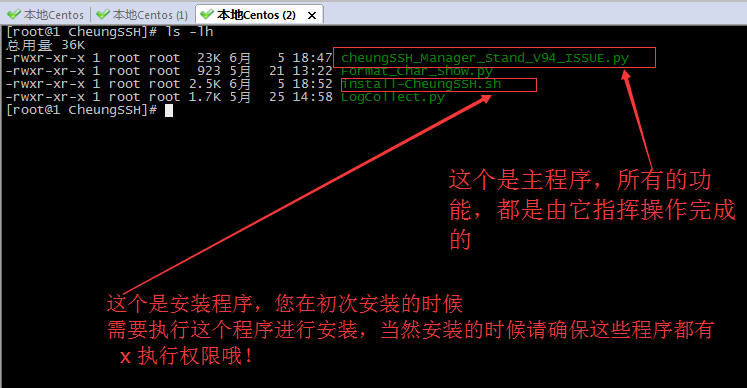
 [root@localhost ]#  git clone  <https://github.com/zhangqichuan/CheungSSH.git>下载成功后，会在您的当前目录有一个**CheungSSH**的目录，里面就是本软件的所有程序了。

第二种下载方式：

如果您的Linux上没有git命令，那么请使用以下方式进行下载：  
 (1) 在您的浏览器地址栏输入： [https://github.com/zhangqichuan/CheungSSH](https://github.com/zhangqichuan/CheungSSH.git)  
（2） 现在您将看到如下页面：  
               
（3） 点击Download ZIP就可以了，然后您可以unzip  CheungSSH-master.zip 解压即可！此时在CheungSSH-master中会有全部的程序。

1. 安装CheungSSH：

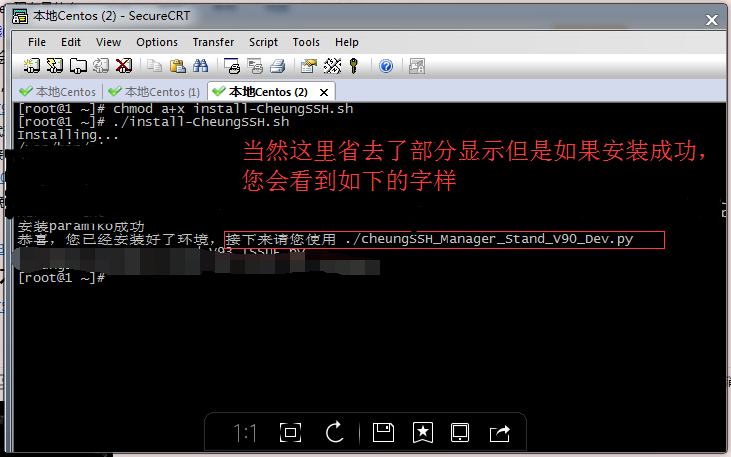
如果您当前已经正确下载了**CheungSSH**那么请执行如下操作进行自动安装**CheungSSH：**

****

**[root@1 CheungSSH] # cd CheungSSH #如果您是浏览器下载的则执行cd CheungSSH-master**

**[root@1 CheungSSH]# chmod a+x install-CheungSSH.sh**

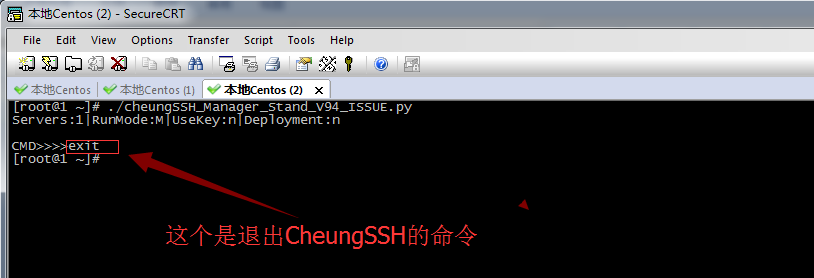
**[root@1 CheungSSH]# ./install-CheungSSH.sh**



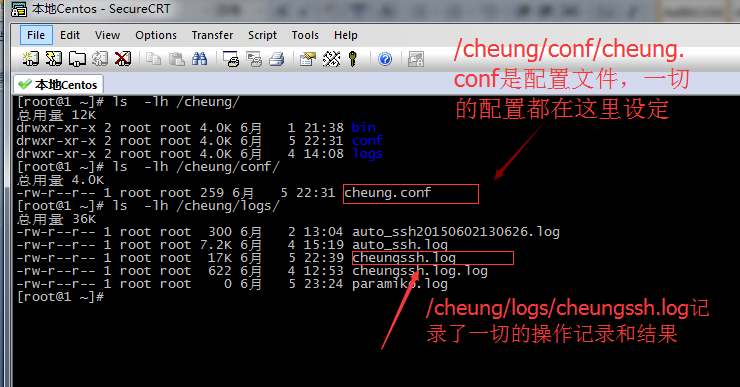
1. 初始化CheungSSH：

如果您已经成功安装了CheungSSH环境，接下来请简单的执行一下

[root@CheungSSH]# ./cheungSSH\_Manager\_Stand\_V94\_ISSUE.py：



然后输入exit退出程序。因为这个时候，就生成了/cheung这个目录，并生成了初始化的配置文件**/cheung/conf/cheung.conf** 您可以使用您的Linux系统命令ls -lh /cheung进行查看



1. 配置介绍：

[CheungSSH]

Servers=192.168.1.1,10.0.0.1,172.16.1.1,www.baidu.com #在这里，您可以定义一个或者多个IP地址，让CheungSSH对这些主机进行操作,多个IP请用 "," 一个逗号分开

Username=root #这里定义您远程服务器上的登陆用户名，就像您用SecureCRT登陆Linux服务器一样指定一个用户名,假如您的所有服务器的登陆账户都一样，那么您只需要指定一个Username就可以了

192.168.1.1\_Username=root123 #假如您定义的N多个IP中，有些服务器的登陆用户名都不一样，那么您可以使用这个配置进行特别的指定一个IP的用户名

10.0.0.1\_Username=root456 #定义同上

Password=123456 #这里定义您的远程服务器上登陆用户的密码，也就是对应的Username账户的密码

Useroot=N #假如，您在您的单位里面，不能直接使用root登陆服务器，必须通过一个普通账户登陆，那么您可以使用这个配置 Useroot=Y 自动切换到root,也就相当于使用su - root命令

Port=22 #默认的SSH端口是22

##################以上是最基本的配置(配置好Servers ，Username，Password即可)，如果您是初次使用，那么建议您先熟悉以上配置，如果您有更高需求，那么久使用以下配置

192.168.1.1\_Port=222 #假如您的某些服务器中SSH端口不是22，比如说是222,333

10.0.0.1\_Port=333

Passwordroot=123456789 #如果您的开启了切换到root账户的配置Useroot=Y ，那么此处您需要指定su - root的密码

Timeout=10 #这个是您登陆您定义的那些IP中，超时的时间，比如在指定的10秒钟内，无法连接上服务器，那么就停止登陆

RunMode=M #使用的线程模式，M表示多线程，D表示单线程，默认是多线程的，为的是提高批量执行的效率。

UseKey=n #是否启用KEY秘钥登陆服务器，如果您担心密码安全问题，那么您可以使用这个来登陆服务器，当然前提是您已经配置好了KEY

Deployment=n #是否启用部署模式，假如您需要批量重启Jboss Tomcat等，那么可以选用这个模式

#ListenFile=/var/log/messages #这个是开启Deployment=y后使用的功能，就是制定一个日志文件，让程序去扫描的文件

#ListenTime=60 #这个是开启Deployment=y后使用的功能，就是定义扫描日志的时间规定，超过这个时间，程序就不会扫描了。

#ListenChar=Server startup #这个是开启Deployment=y后使用的功能，就是定义一个关键字幅信息，假如您操作的对象是tomcat，那么遇到这个信息，就表示启动成功。



1. **配置样例**：

假如您现在有4个服务器需要被批量管理，其中：

192.168.1.1 的账户是root 密码是 123456

192.168.1.2 的账户是admin 密码是 654321

10.0.0.1 的账户是test 密码是test123

10.0.0.2 的账户是test 密码是test123

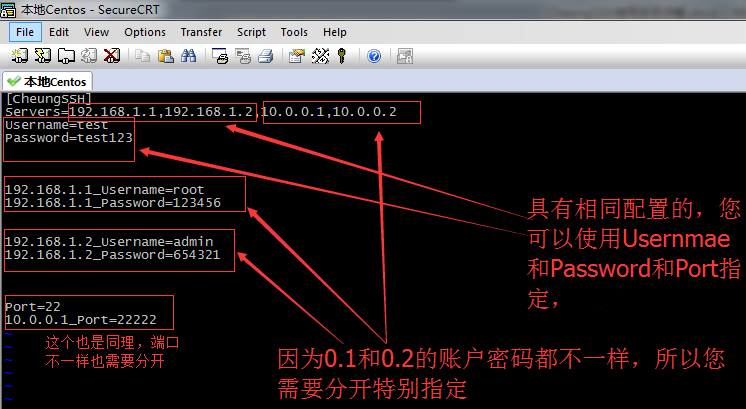
**提示：**

**10.0.0.1与10.0.0.2是一组的，因为他们的Usernmae和密码都一样**

**192.168.1.1是单独的一组，因为没有主机的Usernmae或者是密码与这个IP相同,需要特别指定**

**192.168.1.2是单独的一组，因为没有主机的Usernmae或者是密码与这个IP相同，需要特别指定**

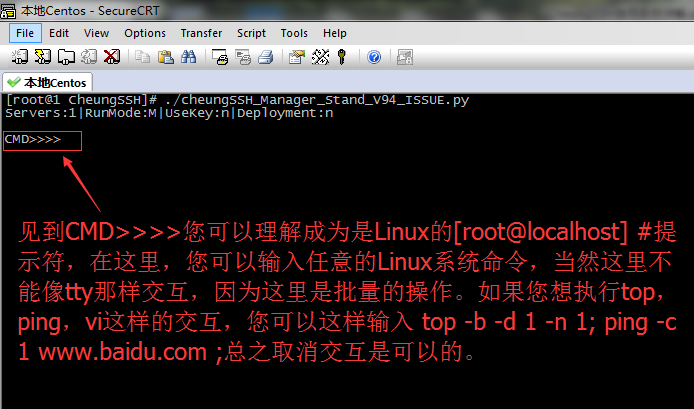
那么你需要这样配置：



1. **正式启动CheungSSH**

如果，通过上面的阅读，您已经配置好了/cheung/conf/cheung.conf这个配置文件，正确的指定了您要被管理的主机，那么您可以使用命令正式启动进入CheungSSH：

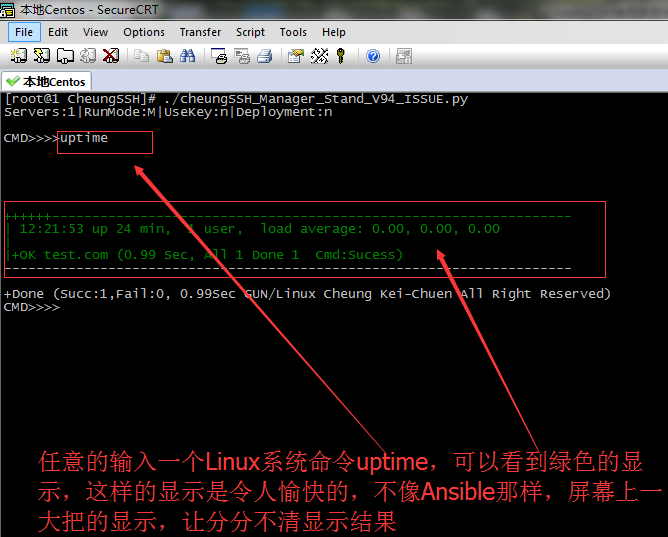
**[root@1 CheungSSH]# ./cheungSSH\_Manager\_Stand\_V94\_ISSUE.py**



## 10.让CheungSSH执行命令

**1.初次执行命令**：

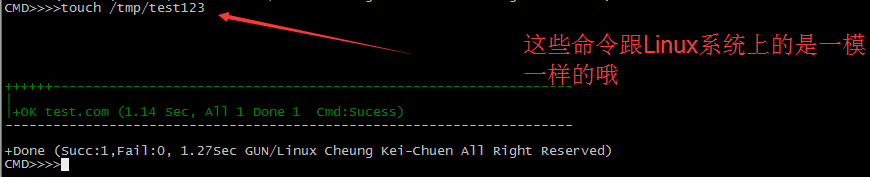
现在您可以输入任何的Linux系统命令，来简单的看看：



当然uptime这是一个远程服务器上的系统命令。

**2.再试试执行命令**:

不仅仅是uptime，您可以输入touch /tmp/test123 在远程服务器上创建一个文件（这些命令仅仅是示例，不要被这些示例混淆，认为只能输入这些命令哦！而是可以输入任何的Linux系统命令）:



**小提示：**

**在CheungSSH里面输入的命令，都是在您的远程服务器上执行的哦！CehungSSH做的就是让所有的服务器都执行您输入的命令，而省去的是您每台服务器逐个进行操作，并且对每个服务器的执行结果进行汇总统计，比如成功了多少，失败了多少？这些都会有统计显示，而这些都在后面的文章中介绍。假如您的服务器有100000台，那么这个效率将是很惊人的！**

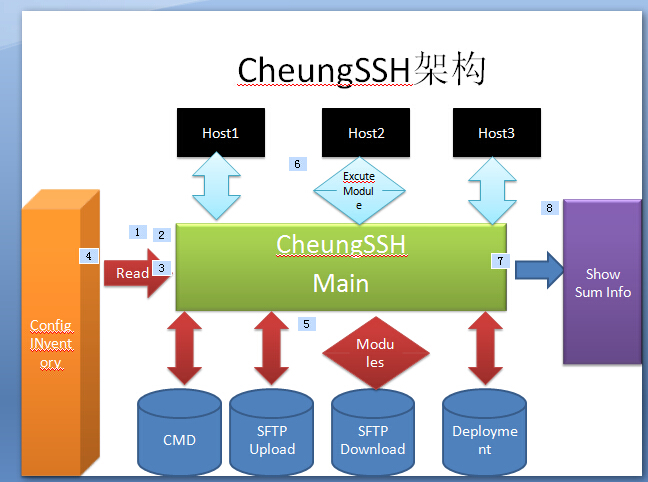
现在，您可以自行登陆test.com(我自己配置的是test.com，当然您配置的服务器可能是其他比如192.168.1.1)上看看 ls –lh /tmp/test123这个文件是否创建成功:



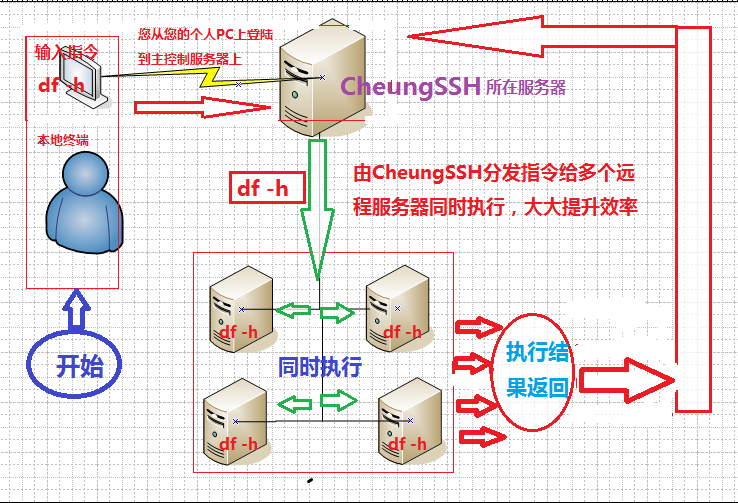
可以看到，的确是创建成功了！

# 架构介绍：

1. CheungSSH的总体架构设计



1. CheungSSH的操作流程



1. CheungSSH设计原理

在运维工作中，我们往往是管理着众多的服务器，如果某个时刻您需要对这些服务器进行管理，比如突然创建一个文件， 突然要多这些服务器进行某种操作，那么此时您是如何应付的？(当然您可能正在使用比如Ansible，Puppet等工具，而我在开篇就说过这些工具的特点是繁琐的)传统的对服务器进行逐个操作，或者是试用SecurCRT进行批量的操作，那将是不可取的，因为效率一样低下。所以，CheungSSH要做的是，把在您的PC上输入的指令，或者是上传下载文件的操作，或者是更新上线代码动作，批量的让被管理的服务器同时执行提高效率，而且对执行过后的结果有序的进行统一汇总显示。就算您要让服务器执行N多次命令或者操作，那么您只需要在您的显示器上看一个汇总结果就能知道所执行的情况了，除了提高批量管理的效率，还提高了人工分析的效率，这就是ChuengSSH的原理。

1. 功能表

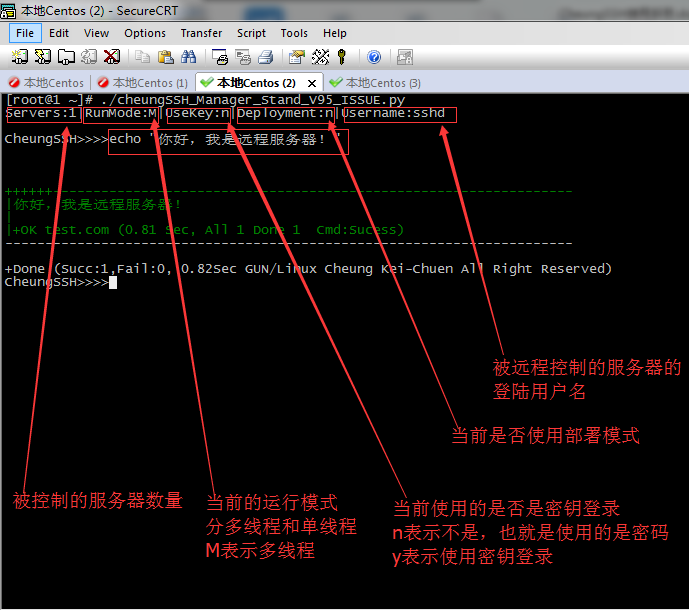
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 序号 | 功能 | 功能解释 | 本软件CheungSSH是否支持 | Ansible是否支持 |
| 2 | 批量分发命令 | 在您的本地输入一条命令，程序将会自动把这些命令发送给远程多个服务器执行 | 支持 | 支持 |
| 3 | 批量上传文件 | 在您的本地执行获取远程服务器的文件到本地，原理跟SFTP一样，但是这里是多个服务器同时执行SFTP | 支持 | 支持 |
| 4 | 批量下载文件(正则匹配) | 在您的本地执行上传远程服务器的文件到远程服务器，原理跟SFTP一样，但是这里是同时为多个服务器执行SFTP，并且支持正则表达式的匹配，比如,您想下载服务器上Tomcat在5月~8月的日志到本地查看: tomcat/logs2015-0[5-8].\*.log | 支持 | 不支持 |
| 5 | 多线程 | 为多个服务器同时执行，这样能减少执行的时间，加快执行效率，尤其是在服务器较多的情况下 | 支持 | 支持 |
| 6 | 自定义输出格式 | 为了更人性化的显示执行结果， 您可以自行编写一个脚本对结果内容进行重新格式化输出，只为了方便您的查看 | 支持 | 不支持 |
| 7 | 格式化日志入库 | 程序执行的命令和结果，都将写入日志文件中，加入您在每天早上要执行巡检CPU MEM  IO  DISK 的信息，比如命令：df -h|grep '/' 得到结果 96% ,那么这些结果都将会格式化的写入日志中，同事也省去了您整理输出信息的时间。提高工作效率 | 支持 | 不支持 |
| 8 | 命令结果返回状态判断 | 每个命令执行是否成功，成功了多少， 用时多少，都会显示 | 支持 | 不支持 |
| 9 | su root权限 | 假如您的服务器只能以普通账户登录，而您平时又是用普通用户登录后su  - root执行命令，这里也支持。 | 支持 | 不支持 |
| 10 | 结果统计 | 执行的命令条数，成功数，用时等人性化统计。 | 支持 | 不支持 |
| 11 | 自助区分 | 每个服务器的返回结果，都以特定格式区分开来，方便查看 | 支持 | 不支持 |
| 12 | 批量发送命令界面 | 可以在程序里面进行批量发送命令，就像是Linux的shell终端一样，但是不支持实时交互 | 支持 | 不支持 |
| 13 | 后台运行 | 以命令的方式运行，比如您把命令写入一个脚本中，然后用crond每天早上为您自动为多个服务器巡检 .Auto\_SSH\_Manager\_Stand\_V71.0.py  <<EOF \n free -m | 支持 | 支持 |
| 14 | 模块化接口 | 每个命令功能，都是一样以模块执行，所以跟Ansible是一样的原理，但是我认为比Ansibel更人性化 | 支持 | 支持 |
| 15 | 二次开发 | 有API接口， 如果您想要针对您的工作进行进一步的开发，那么也是可以的 | 支持 | 支持 |
| 16 | 本地脚本在远程服务器运行（远程服务器没有脚本） | 您在本地编写一个脚本，或者是程序，或者是可执行文件都行，在远程服务器上事先不需要存在该脚本，而您可以直接在远程服务器上运行 | 支持 | 支持 |
| 17 | 自动化巡检 | 比如您可以在crontd中编写一条命令， 定时为您的多个服务器进行巡检操作，巡检的结果，将会格式化的写入本地日志，提 | 支持 | 支持 |
| 18 | 不同文件上传到不同服务器上 | 比如您部署Tomcat  Weblogic集群的时候，有可能每一个Tomcat节点的配置文件都不同，那么可以采用此模式，进行上传配置文件 | 开发中 | 不支持 |
| 19 | 部署模式类似 | 比如您时常部署Tomcat程序 Weblogic程序等，部署的服务器需要部署完成一台才能进行下一台的操作。很多企业都是这样 | 支持 | 支持 |
| 20 | 密码方式登录 | 密码登陆，与Shell一样 | 支持 | 支持 |
| 21 | 密钥方式登录 | PublicKey登陆，与Shell一样 | 支持 | 支持 |

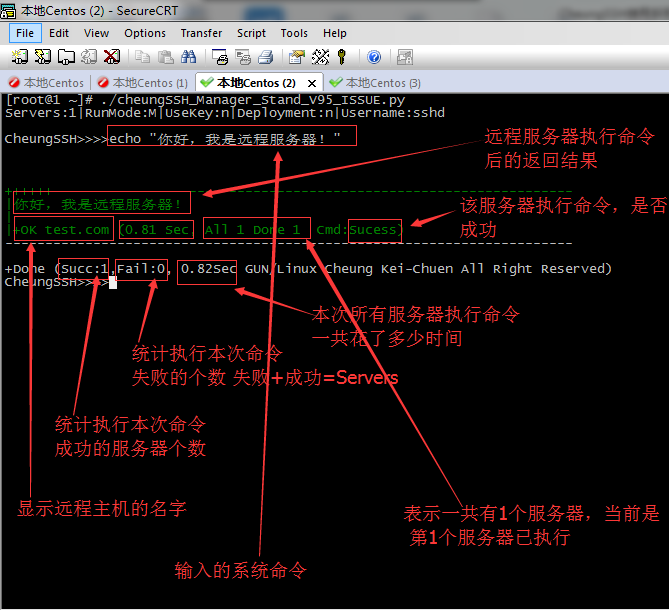
# 功能介绍

1. 命令行模式

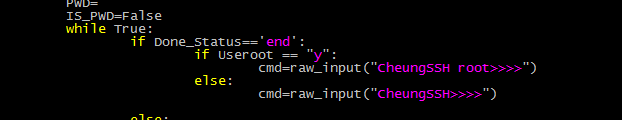
首先在您的终端里面执行命令启动

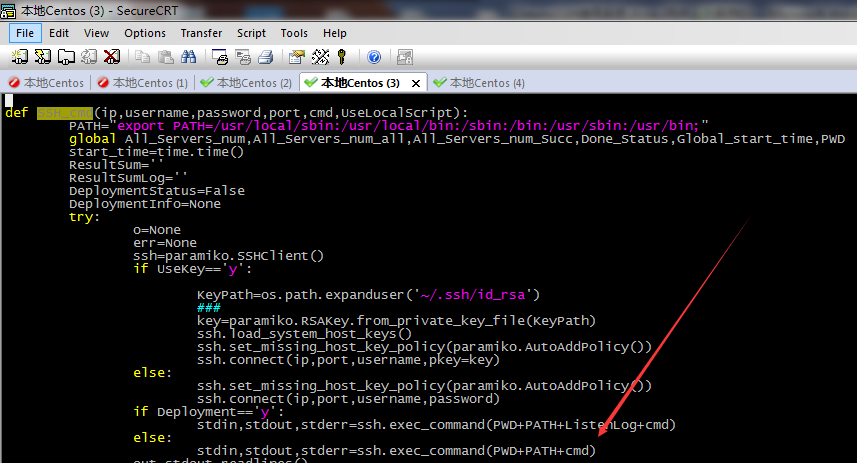
[root@1 CheungSSH]# ./cheungSSH\_Manager\_Stand\_V94\_ISSUE.py



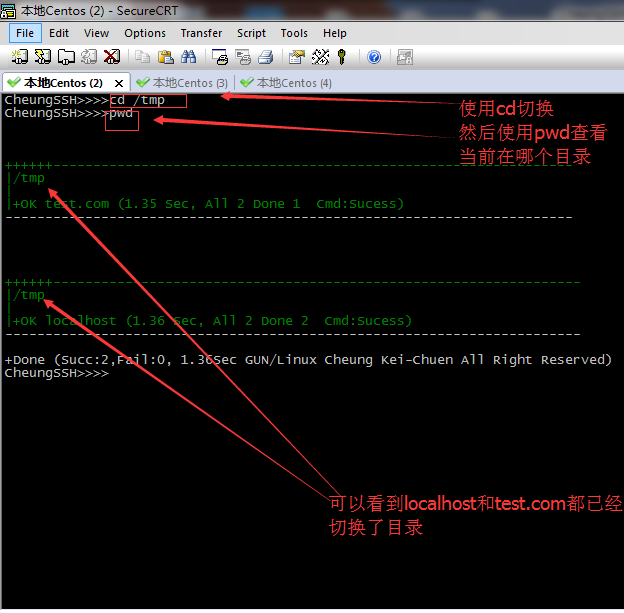


1. 命令行代码的实现：

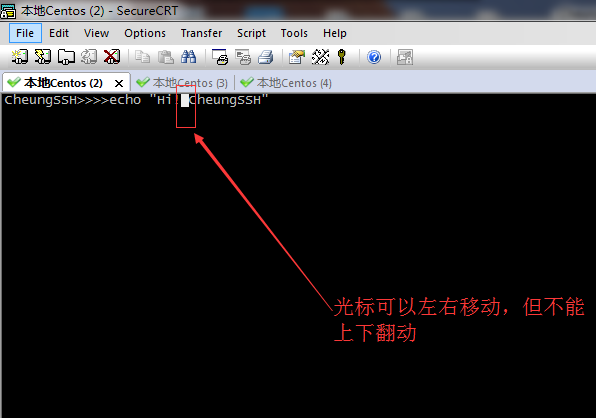




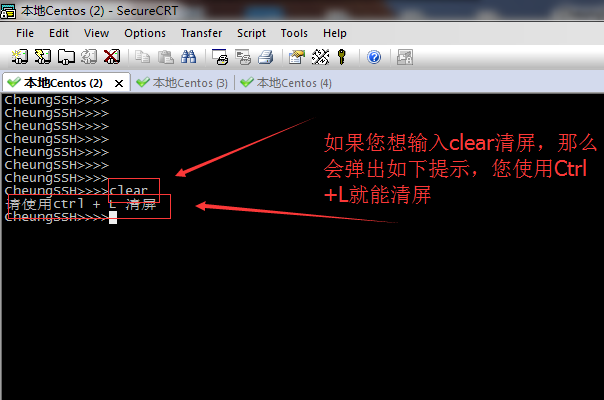
1. 快捷键的使用：
   1. **cd切换目录**



* 1. **方向键**



* 1. 清屏操作

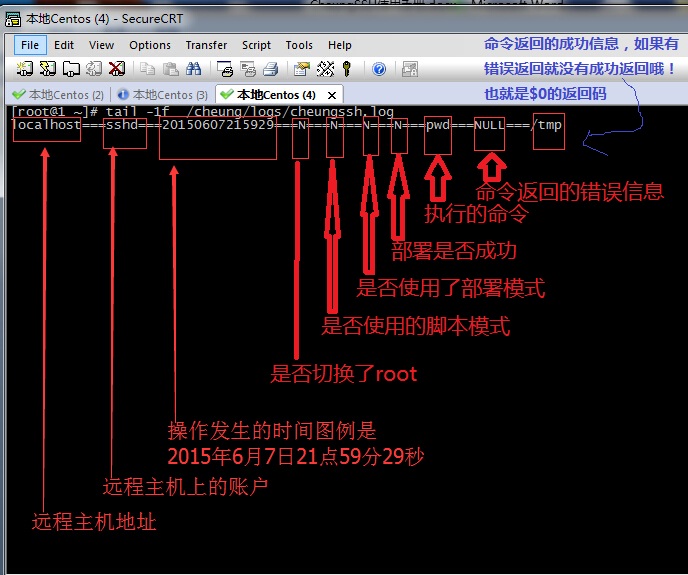


* 1. **History 命令记录**

在CheungSSH中，一样支持了在TTY中的方向键，您可以使用上下方向键翻出您曾经输入过的命令

1. 日志审核

4.1日志格式介绍：



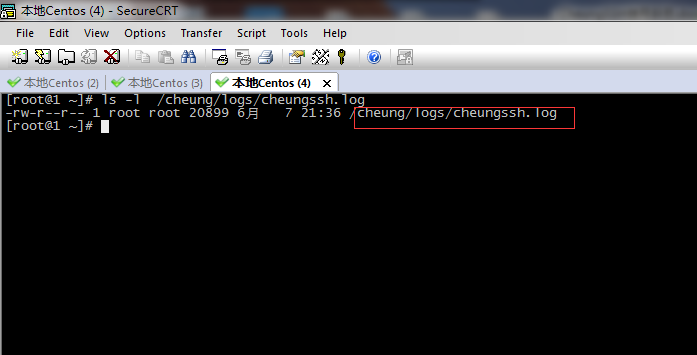
提示：

倒数第二列如果为NULL则表示没有**错误**的返回，表示命令执行是结果成功的

倒数第一列如果为NuLL则表示没有**成功**的返回，表示命令执行的结果失败的

4.2日志路径：

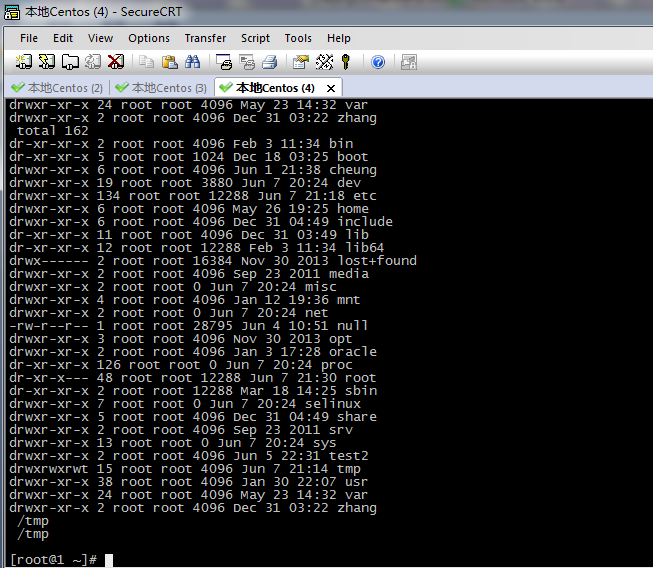
对于您所有的操作，我都为您做了日志记录，这样以便于您在事后对操作进行安全审计，日志文件在/cheung/logs/cheungssh.log



4.3 查看格式化日志：

假如您使用CheungSSH执行了命令 ls –lh / ，那么会生成如下的日志记录,返回的信息不止一行，而我为了能格式化的记录日志，已经合并成了一行，所以，您需要使用命令

[root@1 ~]# awk -F '===' '{print $NF "\\n"}' /cheung/logs/cheungssh.log >/tmp/tmp.txt;echo -e `cat /tmp/tmp.txt`



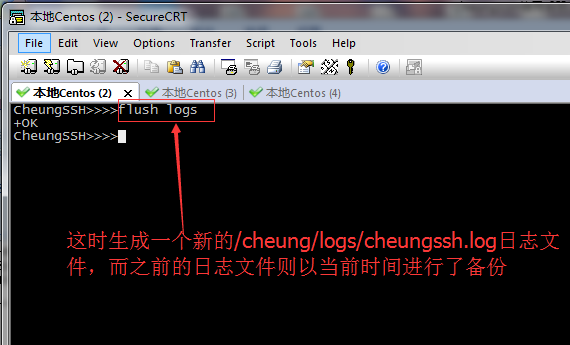
提示：

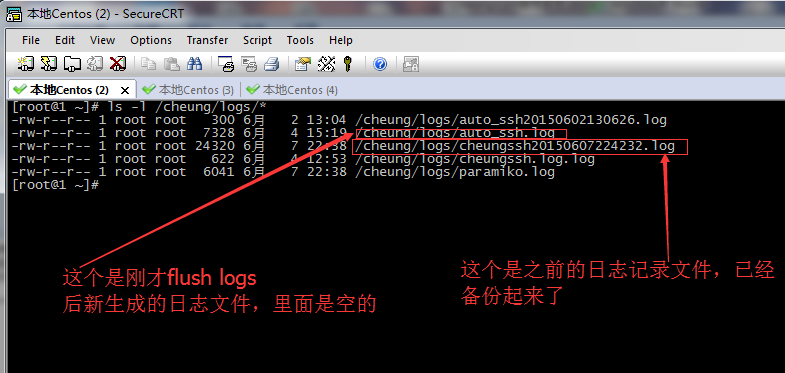
该日志格式，只是为了方便入库而设计的，所以不建议您查看该日志的内容，而查看/cheung/logs/cheungssh.source.log 这个日志文件

1. 日志入库

5.1假如每天要对各个服务器进行Disk、CPU、Mem等的巡检，而要对这些数据进行处理，比如写入Excel、Mysql数据库等等

5.2如果您要对日志进行入库的操作，建议首先生成一个全新的日志文件，在CheungSSH命令行中，执行 flush logs将生成一个新的/cheung/logs/cheungssh.log日志文件





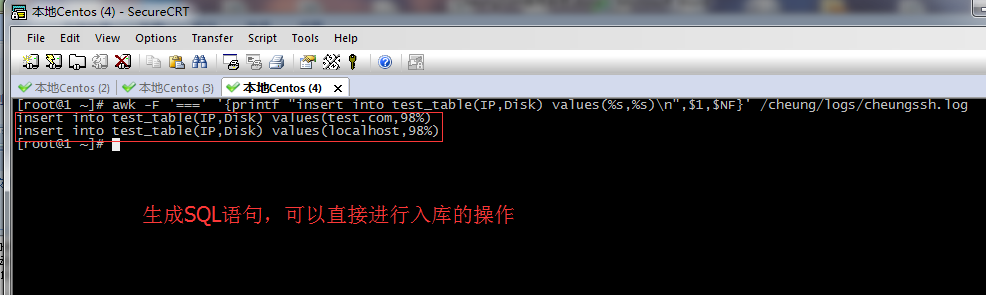
这样做的目的，只是为了能更方便的入库操作

5.3 对磁盘进行巡检

df -h|awk '$NF=="/"{print $(NF-1)}'



* 1. 对执行的结果进行提取，并写入mysql数据库：

awk -F '===' '{printf "insert into test\_table(IP,Disk) values(%s,%s)\n",$1,$NF}' /cheung/logs/cheungssh.log 

1. 原始日志（适合查看的日志格式）

6.1 [root@1 ~]# tail -f /cheung/logs/cheungssh.source.log

