https://blog.csdn.net/qunnie\_yi/article/details/80129816

# 程序员必知的21个命令

2017年06月05日 15:34:41 [CSDN王艺](https://me.csdn.net/qunnie_yi) 阅读数：213 标签： [深度学习](http://so.csdn.net/so/search/s.do?q=%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0&t=blog)[人工智能](http://so.csdn.net/so/search/s.do?q=%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD&t=blog)[机器学习](http://so.csdn.net/so/search/s.do?q=%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0&t=blog)[神经网络](http://so.csdn.net/so/search/s.do?q=%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&t=blog)



在这篇文章里，我们将要一睹能快速分析文本数据（如日志，报告等）的最方便工具。很多时候，我们需要的数据并不存储在我们的本机上。所以首先，我们要知道如何链接到远程服务器上并使用它。为此，使用SSH最为合适。（SSH，即Secure Shell,是一个加密网络协议，它能让你在一个不安全的网络上安全地远程登陆和使用其他网络服务）

在Ubuntu的终端你可以用以下几个命令之一来登陆远程服务器。

$ ssh user@host

$ ssh -p port host

* 1
* 2

通过密钥链接：

$ ssh -i key.pem user@host

* 1

在远程主机上执行命令：

$ ssh -i key.pem user@host 'command'

* 1

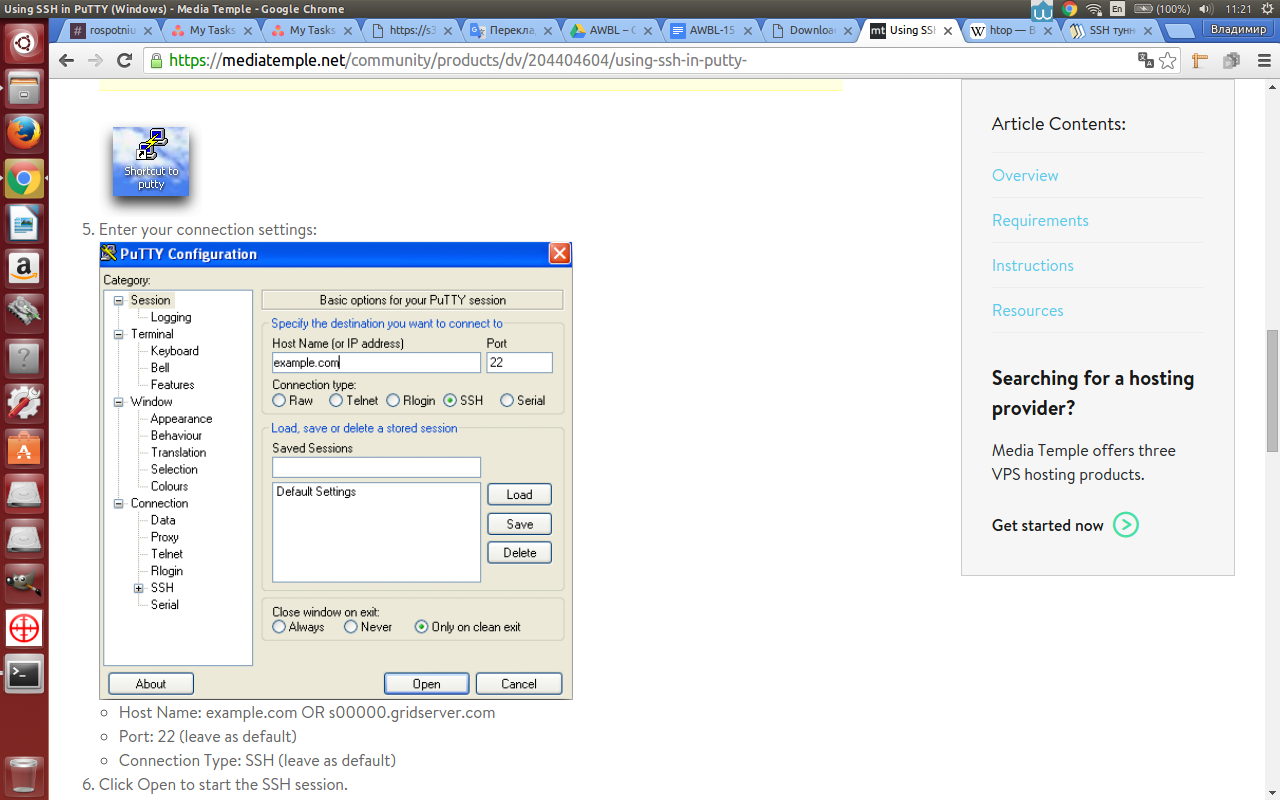
此时，SSH 客户端就已经在Ubuntu的默认程序列表中了。若没有，你可以在终端尝试以下命令。

command sudo apt-get install ssh

* 1

### On Windows

若你使用微软的Windows，你需要安装一款开源软件PuTTY（地址[http://www.putty.org/](http://www.putty.org/" \t "_blank)）来使用SHH。为了使用PuTTY链接，需要以下几步：

1. 运行PuTTY并输入你的链接设置：   
     
   a. Host Name: example.com   
   b. Port: 22 (leave as default)   
   c. Connection Type: SSH (leave as default)
2. 点击Open开启一个SSH回话。
3. 当打开SSH连接时，你应该会看到一个终端提示让你输入用户名。
4. 接下来，输入密码。需要注意的是，当输入密码的时，你不会看到游标移动或任何输入字符（如\*\*）。这是一个标准的PuTTY安全特性。点击enter。
5. 现在你已经可以在终端提示符下使用远程主机了。

SSH的一个好的特性是可以创建secure tunnels。

### On Ubuntu

**1. 将远程主机连接到本地**

进入主机1.1.1.1

ssh -f -N -R 22:192.168.0.1:22 username@1.1.1.1

* 1

获得主机192.168.0.1的权限

$ ssh localhost

* 1

操作：   
- -R 将远程主机(服务器)的某个端口转发到本地端指定机器的指定端口。执行远程重定向。当你访问远程主机的这个端口，就会建立一个SSH隧道，并建立了通向这个主机端口的链接。   
- -N不执行远程脚本或命令，只在前向端口时使用。   
- -f 经过远程系统认证后立即进入后台模式。

使用ssh隧道一个非常有用的方式就是用于这种通信加密。比如，你在一个开放的网络中使用，但是你不想别人能够拦截你的数据。

**2. 将本地机(客户机)的某个端口转发到远端指定机器的指定端口**

在本机1.1.1.1输入[http://localhost:80](http://localhost/)就会得到访问远程主机192.168.0.1的页面。

ssh -f -N -L 80:192.168.0.1:80 username@1.1.1.1

* 1

操作：   
- -L 将本地机(客户机)的某个端口转发到远端指定机器的指定端口。

反向隧道用于这种情况，你需要访问一台被防火墙保护或处于NAT之后的主机。

### 使用PuTTY的SSH隧道功能

* 在Connection -> SSH -> Tunnels，输入
* Source port: 22

Destination: localhost:22

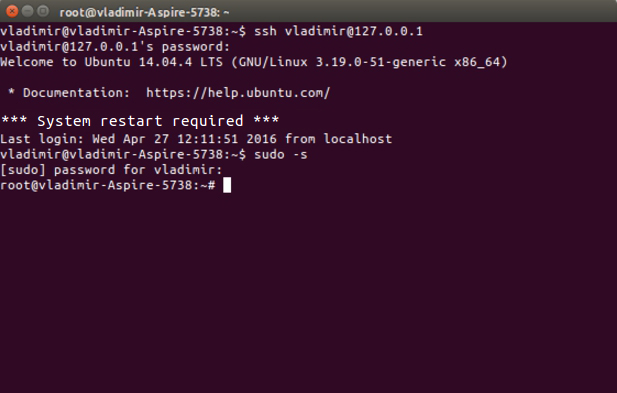
* + 1
  + 2
* 选择”Dynamic”，点击“Add”
* 在会话中输入目标主机地址并选择SSH协议。通过手动输入已保存的会话名来保存你的选择。例如：“HostName with ssh tunnel on 22”，点击“保存”，双击会话名，连接成功。
* 当SSH会话建立成功，你可以通过浏览器验证链接是否成功。

现在我们连上了主机，已经可以操作了。但是一些命令需要root权限。如果主机允许使用root权限，你可以在终端中执行：

$ sudo -s

* 1

在输入root用户的密码后，通过终端向远程主机的链接就会被确认，你也获得了root权限：



我们知道了如何链接到主机和获取必要权限的所有要点，但这还不足以用来分析主机上的数据。接下来还要学习在远程主机上控制、数据获取和获取的基本命令。

### 使用文件系统

现在，我们开始学习如何使用文件系统。以下是相应的命令用来：

**1. 处理文件和文件夹**

新建一个文件夹 dir\_name

mkdir dir\_name

* 1

将dir\_name重命名为dir\_name2

mv dir\_name dir\_name2

* 1

删除文件夹/文件

rm -rf dir\_name

* 1

操作：

* -r, -R 处理所有嵌套的子目录。
* -i 每一个传输链接移除之前需确认
* -f 若错误是因文件不存在导致，则不以false结束，且不询问是否建立连接

删除所有以file2015开头的文件；\*表示任意字符，你可以把它用在任何表示缺失字符的地方。

rm -rf file2015\*

* 1

创建文件 file\_name.txt

touch file\_name.txt

* 1

重命名文件

mv file\_name.txt file\_name2.txt

* 1

**2. 权限**

* ls -la 查看目录下所有文件、文件夹的连接情况及所有权
* -l 关键的扩展输出

执行file文件的权限

chmod 777 file.sh

* 1

R 代表dir\_name文件夹所有的文件

chmod -R 777 dir\_name - set access 777 recursively

* 1

更多信息请参考： <https://ru.wikipedia.org/wiki/Chmod>

**3. 所有者(Owner)/所属群组(Group)**

为file.txt文件设置所有者和群组

chown ubuntu:ubuntu file.txt

* 1

递归设置所有者和群组

chown -R ubuntu:ubuntu dir\_name

* 1

**4. 查看磁盘空间**

显示目前在Linux系统上的文件系统的磁盘使用情况

df -h

* 1

获取文件夹大小

du -sh dir\_name/

* 1

获取文件夹和子目录的大小

du -h dir\_name/\*

* 1

获取文件大小

du -h filename

* 1

下面的命令可以用来获取主机系统的状态：

* 查看内存和交换空间的使用情况

free

* + 1
* 实时显示系统中各个进程的资源占用状况

top/htop

* + 1

现在我们已经知道如何使用文件系统并获得我们想要信息。不幸的是，在大部分情况下，这些信息都不便于人们理解。所以要考虑能方便数据分析的工具。

分析信息的第一步，就是找到信息。

find / var -name search \_name

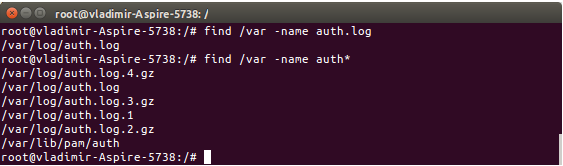
* 1

通过这个命令可以用\*符号，如果全名太长或者不知道，这时用

find / var -name search\_name\*

* 1

这个例子查询主机上的认证日志：



一旦找到了文件，通常情况下我们希望查看他们的内容。

* 输出文件内容

cat file\_name

* + 1
* 将文件1和文件2合并，存入文件3

cat file\_name1 file\_name2 > file\_name3

* + 1
* 将多个文件合并存入文件的开始

cat file\_name\* > file\_name\_end

* + 1
* 将多个文件合并到制定目录下的制定文件

cat some\_dir\*/file\_name\* > file\_name\_end

* + 1

除了对文件重定向，也可以对脚本和命令重定向

* 重定向输出到文件（覆盖文件）

./some\_script.sh > /path/to/file.txt

* + 1
* 重定向输出到文件尾部

./some\_script.sh >> /path/to/file.txt

* + 1
* 将标准输出和错误输出重定向到/dev/null

some command > /dev/null 2>&1

* + 1

下一步保存文件内容

将

/var/log/auth.log

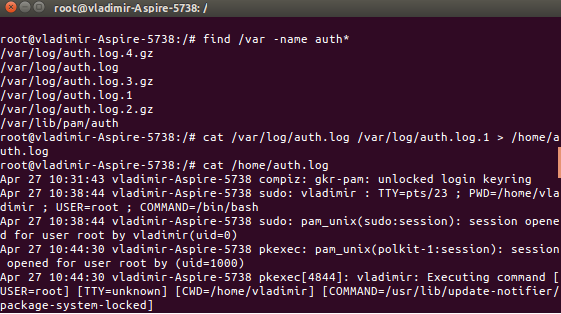
* 1

和

/var/log/auth.log.1

* 1

的文件内容保存到/home/auth.log



从上面的输出，我们可以看到，有许多冗余的信息。我们可以通过过滤来得到想要的信息。有一个非常有用的用法grep来处理这种情况。语法如下，

* 将cat的output信息过滤，这得到包含some\_line的行

cat file\_name | grep some\_line

* + 1

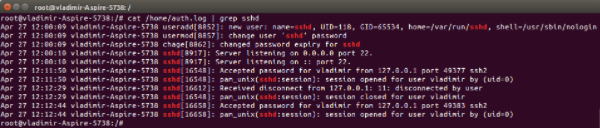
Grep也可以应用到命令和脚本中

* 只输出包含errors的字符串

python run\_sum\_script.py | grep error

* + 1

通过sshd过滤，我们可以打印只关于ssh的认证记录



我们也可以跟踪文件更改记录。例如，当你需要测试网页时，这将变得非常有用。所以当你查询日志或者查看新的认证记录时，你可以获取到这样的信息：

打印制定文件的第三方应用的新记录

tail -f some\_web\_server\_log

* 1

也可以配合grep来使用

打印包含error的新记录

tail -f some\_web\_server\_log | grep error

* 1

接下来让我们看看tail做了什么。首先设置跟踪文件

/var/log/auth.log

* 1

然后利用tail命令

-f /var/log/auth.log | grep sshd

* 1

在每个面向主机ssh的新连接建立以后，我们显示相关信息和认证记录。

另一个用于分析文件的有用工具是awk. 利用它，我们可以很容易地处理任何一个文件结构。Awk是用于语义搜索和编辑的工具。它可以看做成一种shell形式的awk。运行awk程序有很多种方式。对于一个很短的程序，很容易将其加入运行awk的命令，如下：

awk 'program' input-file1 input-file2

* 1

更多信息，请参考手册： <http://www.gnu.org/software/gawk/manual/gawk.html>

我们可以利用awk查看用户何时连接或加入ssh

$ awk '/sshd/ && /pam\_unix/ {print ($1,$2,$3,$8,$11)}' /var/log/auth.log

* 1



另外一个有用的命令是sed. 该命令将文件（默认为标准输入）复制到标准输出，利用包含这些命令的脚本编辑这些文件。利用标识符-f命令可以得到sfile文件的镜像。如果只有-e script选项，-e标识符可以省略。默认情况下-n标识符无效。Script包含编辑命令，每个占一行，格式如下：

[addr [, addr]] cmd [args]

* 1

Sed命令循环将输入字符串转化，并输出。

**例子：**

* 将report.txt文件中所有的John替换为Nick

sed 's/Nick/John/g' report.txt

* + 1
* 删除那些包含如下信息的行：

who | sed '2,4d'

* + 1
* 等等。

下面介绍一下一些便于远程主机操作的有用的命令。Tar和zip对于文档的操作会节省空间和流量。常用的命令如下：

**对于tar：**

* 压缩阶段忽略读取错误

tar czf new-tar-file-name.tar.gz file-or-folder-to-archive

* + 1

--ignore-failed-read

* + 1

其中，

* + c 用于从创建新的文档
  + z 设置用gzip压缩
  + f 设置文档文件。
* 对于多级目录:
* tar -czf new-tar-file-name.tar.gz file1 file2 folder1 folder2

tar -xzf tar-file-name.tar.gz

* + 1
  + 2
  + x设置提取文档
  + z设置利用gzip解压
  + f设置文档文件

**对zip：**

* 压缩文件命令：

zip file-or-folder-to-archive new-zip-file-name.zip

* + 1
* 解压文件命令：

unzip -d new-zip-file-name.zip

* + 1

其中，-d指明目标目录

当然，远程主机的相关任务还需要本地电脑和远程主机之间的文件传输。下面的命令将帮助我们完成这些任务：

* 从本机下载tar

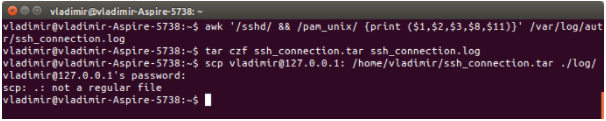
scp -r (recursive) username@server:(remote location) (local location

* + 1
* 向远程服务器上传文件

scp -r (local location) username@server:(remote location)

* + 1

利用awk命令我们已经获得用户连接时间和断开时间。现在我们将这些保存在一个文件中，将其压缩成tar文件并上传：



### 总结

通过这篇文章，我们可以看多CLI的多种用途和它是如何帮助我们进行数据科学探究的。多数读者肯定在搜索命令行上花了不少的时间。本文旨在帮助理解这些常用的基础命令。

每一个指令都很有用，它们的合理组合将会形成一个非常强大指令。现在你可以熟练运用grep，sed和awk等命令。它们将会给你带来很多机会。

我们希望了解更多读者平时常用的指令，请在下面留言，继续学习这些指令，提升自我。好运！

原文：[21 essential command line interface tools for Data Scientists](https://activewizards.com/blog/21-essential-command-line-interface-tools-for-data-scientists/)   
译者：李乐 王晨

**2017中国人工智能大会（CCAI 2017）| 7月22日-23日 杭州**