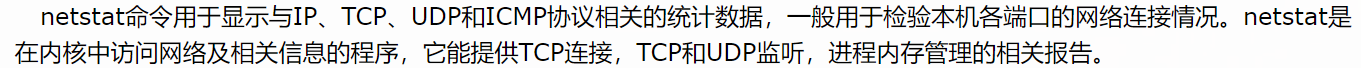
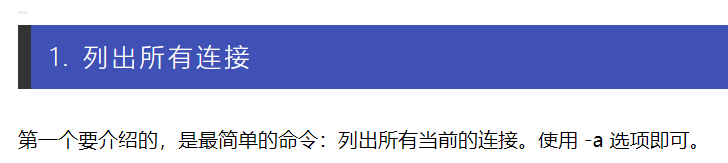
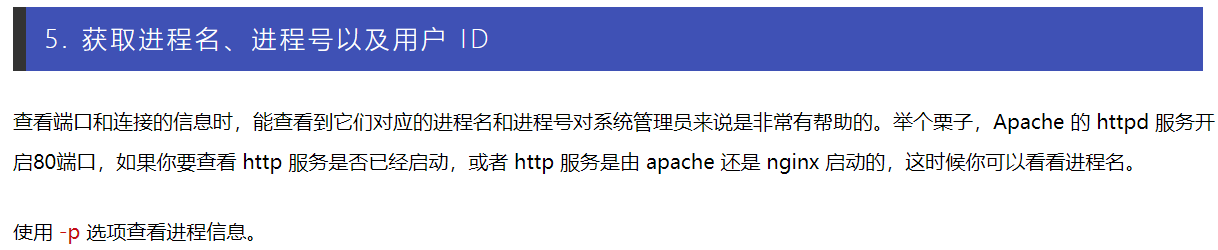
Netstat

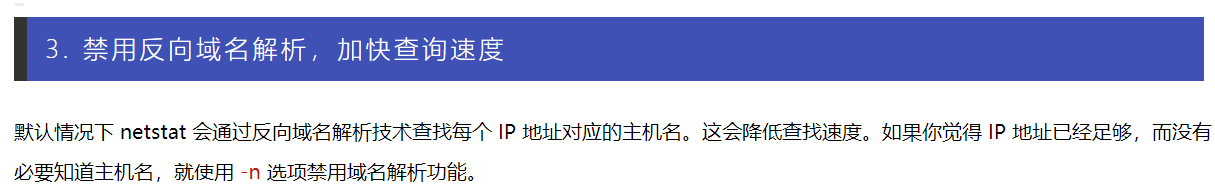








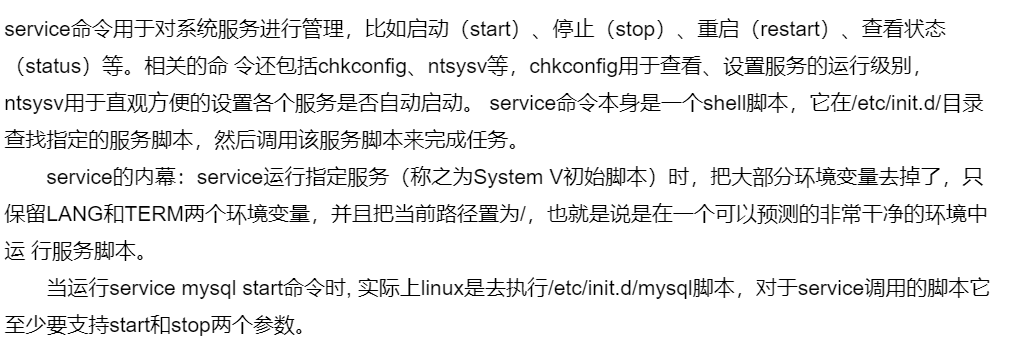




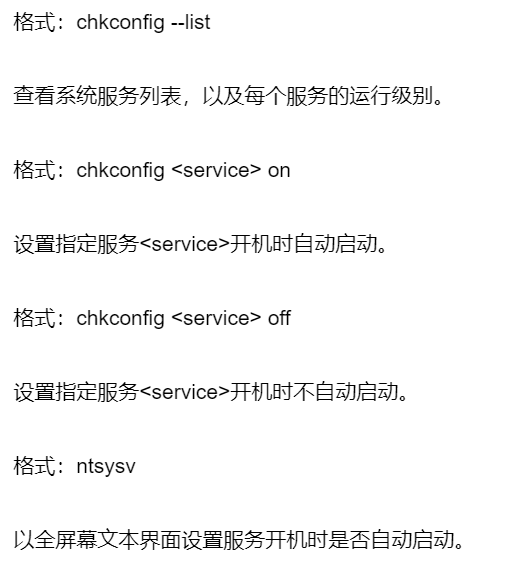
“用 “route -n” 和 “netstat -nr” 命令,我们可以查看默认网关。



Service









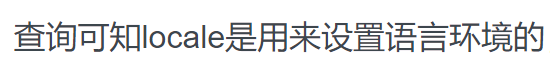
防火墙firewalld



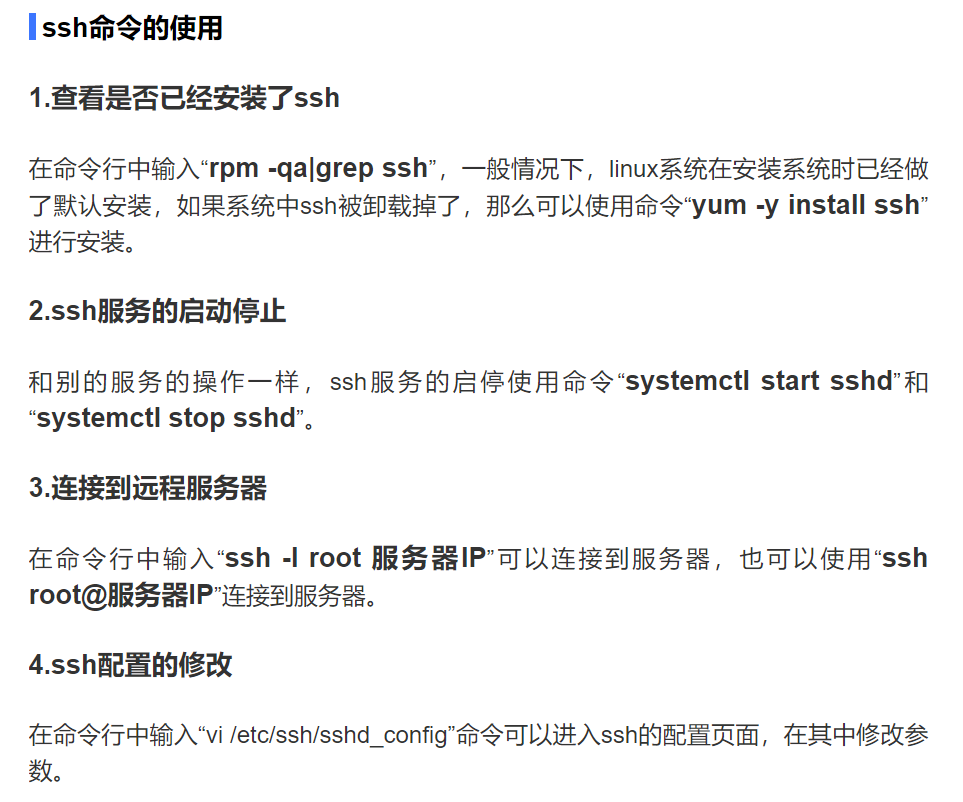
Wget



locale



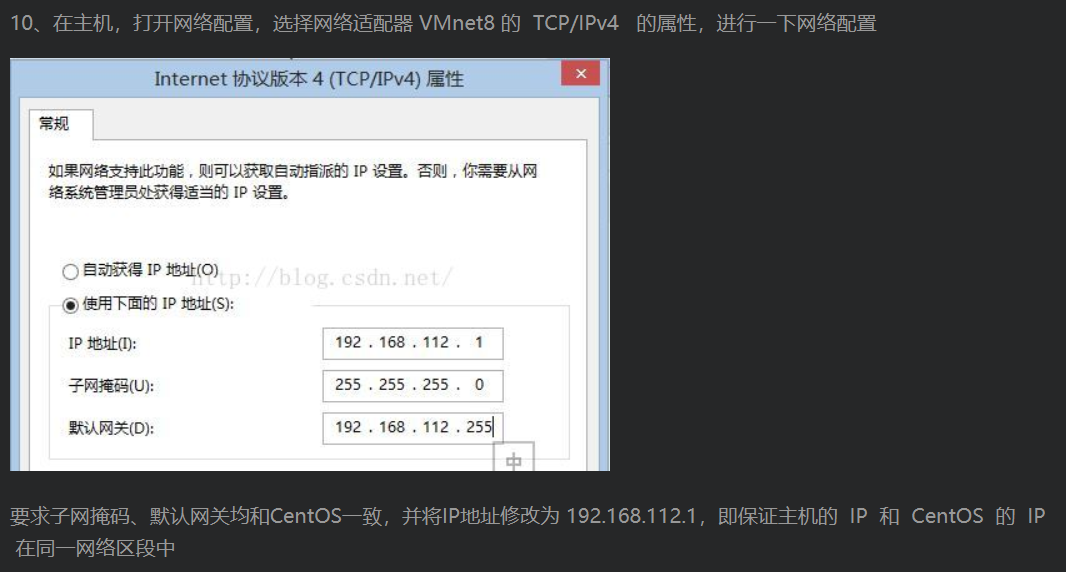
Ssh



Window主机中用ipconfig查看主机ip

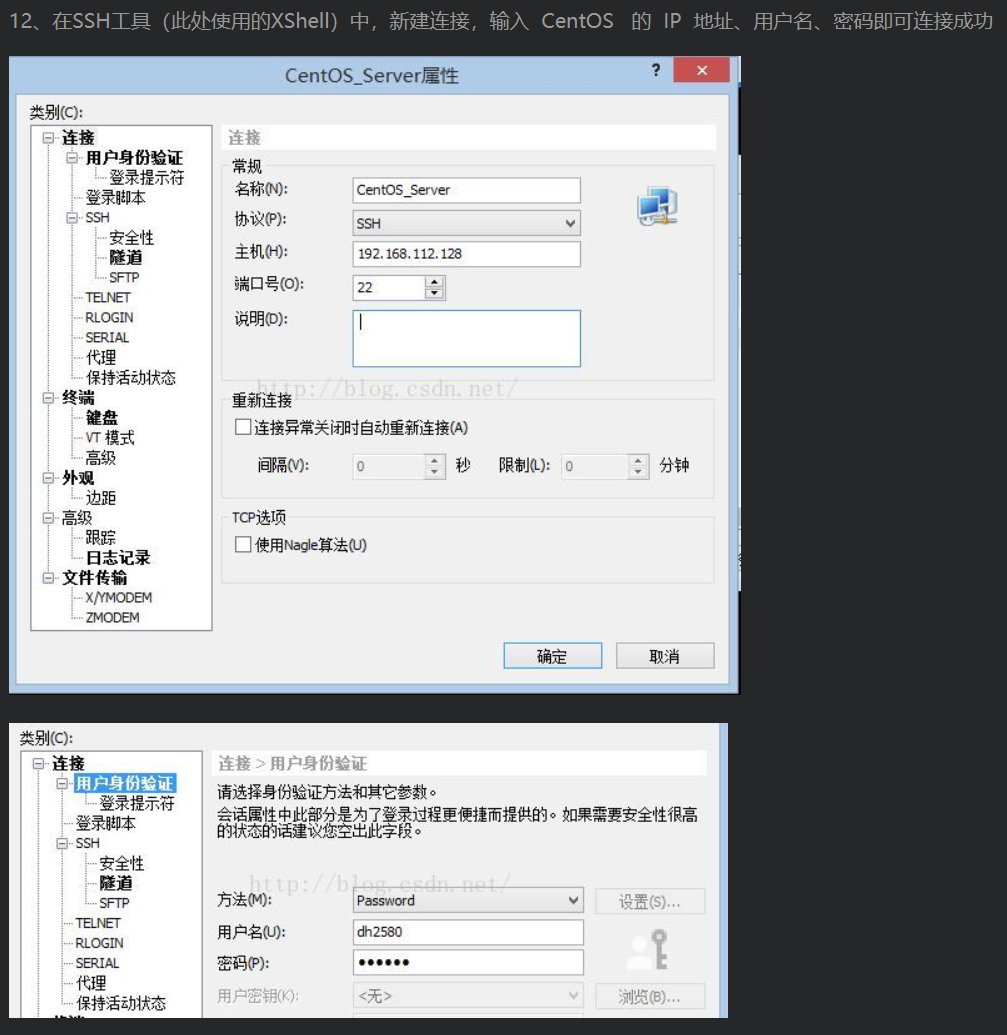
Centos中用ifconfig查看ip

用ping 使两个ip互通信成功



用docker作为作为centos的容器的话，要改tcp/ipv4属性的网络适配器是vEthernet (Default Switch

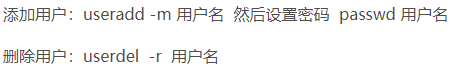
aaa.bbb.ccc.ddd的同一网络区段的ip是aaa.bbb.ccc.1

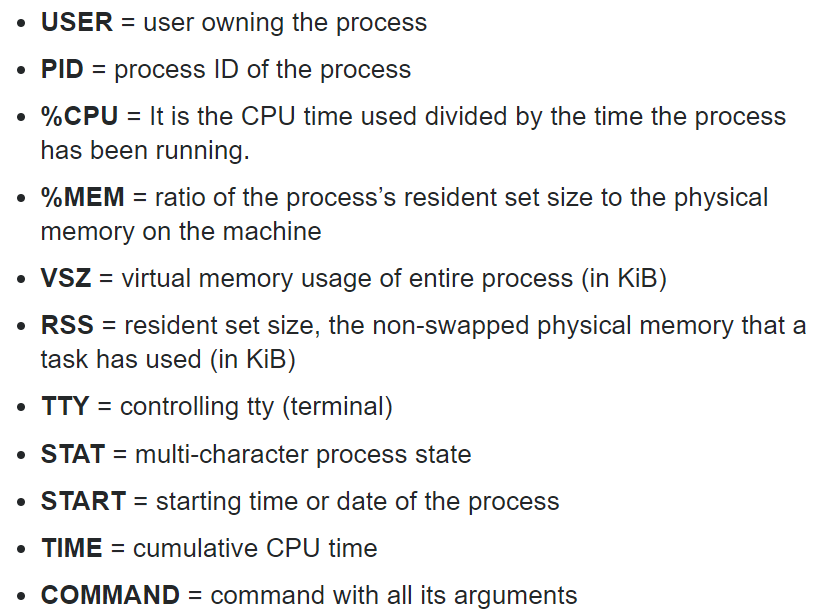


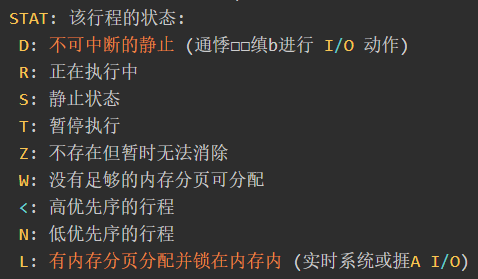
find

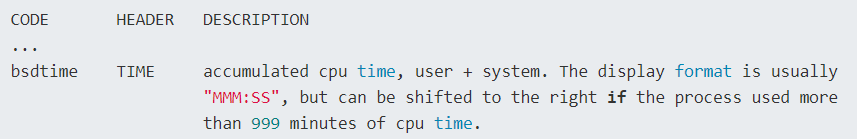
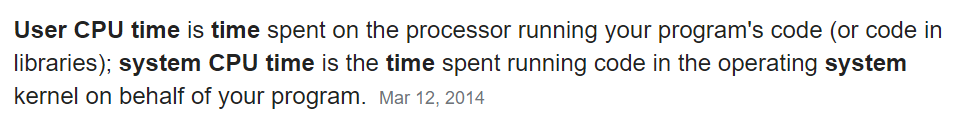
find ./ -name ‘xxx’

Useradd 增减用户



1. find  
   Run find to find file in dir:  $ find dir -name file -print   
     
   The find command accepts special pattern-matching characters such as *\*, but you must enclose them in single quotes ('\**')to protect the special characters from the shell's own globbing feature.   
     
   2.file   
   distinguish file's type  
     
   3. xargs  
   make the same command work on each arg passed to it individually, i.e. how many 'arg' passed by then how many 'command' would be execute  
     
   Many people use xargs with the find command. For example, the script below can help you verify that every file in the current directory tree that ends with .gif is actually a GIF (Graphic Interchange Format) image:   
   $ find . -name '\*.gif' -print | xargs file  
     
   In the example above, xargs runs the file command. However, this invocation can cause errors or leave your system open to security problems, because filenames can include spaces and newlines. When writing a script, use the following form instead, which changes the find output separator and the xargs argument delimiter from a newline to a NULL character:   
   $ find . -name '\*.gif' -print0 | xargs -0 file  
     
   xargs starts a lot of processes, so don’t expect great performance if you have a large list of files.    
     
   You may need to add two dashes (--) to the end of your xargs command if there is a chance that any of the target files start with a single dash (-). The double dash (--) can be used to tell a program that any arguments that follow the double dash are filenames, not options. However, keep in mind that not all programs support the use of a double dash.   
     
   There’s an alternative to xargs when using find: the -exec option. However, the syntax is somewhat tricky because you need to supply a {} to substitute the filename and a literal ; to indicate the end of the command. Here’s how to perform the preceding task using only find:   
   $ find . -name '\*.gif' -exec file {} \;   
     
   4. pgrep  
   pgrep looks through  the  currently  running  processes and lists  the process IDs which match the selection criteria to stdout.  All the criteria have to match.  
   $ pgrep svr02  
     
   5. kill  
   When you run kill, you’re asking the kernel to send a signal to another process.  
   if kill not specify which signal to be sent, then would send TERM signal by default.  
   -s signal  
   send specify signal to process, the signal could be represented by number or signal name. signal 9 equal to SIGKILL, force to kill process.  
   $ kill -s 9 pid  
     
   6. free  
   $ free -m  
   show RAM's usage in Unit of MByte.  
     
   关于Buffers和Cached的区别，来自 http://www.taobaotesting.com/blogs/qa?bid=2265 的一段解释如下：  
     
   buffers是指用来给块设备做的缓冲大小，他只记录文件系统的metadata以及 tracking in-flight pages.  
     
   cached是用来给文件做缓冲。  
     
   那就是说：buffers是用来存储，目录里面有什么内容，权限等等。而cached直接用来记忆我们打开的文件  
     
   7. du  
   $ du -m filename  
   show disk space occupied by file in unit of MByte.  
     
   8. ps - report a snapshot of the current processes  
   To see every process on the system using standard syntax:  
   $ ps -ef  
     
   To see every process on the system using BSD syntax:  
   $ ps axu  
     
   查看进程：  
   1) ps 命令用于查看当前正在运行的进程。  
   grep 是搜索  
   例如： ps -ef | grep java  
   表示查看所有进程里 CMD 是 java 的进程信息  
   2) ps -aux | grep java  
   -aux 显示所有状态  
   ps  
   3) kill 命令用于终止进程  
   例如： kill -9 [PID]  
   -9 表示强迫进程立即停止  
   通常用 ps 查看进程 PID ，用 kill 命令终止进程  
     
   $ ps -eo
2. 



1. 
2. 
3. 9. strings  
   显示文件中的可打印字符  
   $ strings /usr/lib64/libstdc++.so.6 | grep 'CXXABI'  
     
   10. ln  
   在文件之间建立连接  
   $ ln -s libstdc++.so.6.0.19 libstdc++.so.6  
   above command create a soft link.  
     
   11. linux 添加开机启动项  
   将写好的脚本（.sh文件）放到目录 /etc/profile.d/ 下，系统启动后就会自动执行该目录下的所有shell脚本。  
     
   12. pgrep  
   looks through the currently running processes and lists the process IDs which match the selection criteia to stdout.  
   $ pgrep processName | xargs kill -s 9 //this command used to kill the process represented by processName  
     
   13.  
   grep  
   print lines matching a pattern  
     
   -n, --line-number  
   Prefix each line of output with the 1-based (which means, the line number count from 1) line number within its input file.  
     
   -E, --extended-regexp  
   Interpret PATTERN as an extended regular expression, or use egrep command has the same effect with grep -E , coz grep only support basic RE by default  
     
   -q, --quiet, --silent  
   suppress all normal output, do not write anything to standard output  
     
   -l, --files-with-matches  
   Suppress normal output; instead print the name of each input file form which output would normally have been printed (which mean, print the matched filename only, which contained the matched contents). Then scanning will stop on the first match.  
     
   -Z, --null  
   Output a zero byte (the AXCII NUL character) instead of the character that normally follows a file name. For example, grep -lz outputs a zero byte after each file name instead of the usual newline. This option makes the output unambiguous, even in the presence o file names containing unusual characters like newlines. This iption can be used with commands like find -print0, perl -0, sort -z, and xargs -0 to process arbitrary file names, even those that contain newline characters.  
     
   -i, --ignore-case  
   Ignore case distinctions in both the PATTERN and the input files (which mean, a letter in upper-case or lower-case are the same).  
     
   -w, --word-regexp  
   Select only those lines containing matches that form whole words (which means, exactly match the search string in the entir fully-single form). The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a no-word constituent character. Word-constituent characters are letters, digits and the underscore.  
   i.e.  
   grep -w 'is' ./\*  
   it would not print the line contain "this", "his".. but would display the line containing the fully-single word "is"  
     
   -R, -r, --recursive  
   recursive call, searches files in the current working directory and all subdirectories below it.  
     
   EXAMPLES  
   `find / -type f -print0 | xargs -0 grep -liwZ GUI | xargs -0 rm -f`  
   `grep -rliwZ GUI / | xargs -0 rm -f`  
   Either of the above will remove any file containing "GUI".  
     
   -v, --invert-atch  
   Invert the sense of matching, to select non-matching lines.  
   e.g.  
   grep pattern1 \*.txt | grep -v pattern2  
   # Matches all lines in "\*.txt" files containing "pattern1",  
   # but \*\*\*not\*\*\* "pattern2".  
     
   -c, --count  
   Suppress normal output; instead print a count of matching lines for each input file. With the -v, --invert-match option, would count non-matching lines.  
     
   -z, --null-data  
   Treat the input as a set of lines, each terminated by a zero byte (the ASCII NUL character) instead of a newline. Like the -Z or --null option, this option can be used with commands like sort -z to process arbitrary file names.  
     
   --color[=WHEN], --colour[=WHEN]  
    Surround the matched (non-empty) strings, matching lines, context lines, file names, line numbers, byte offsets, and separators (for fields and groups of context lines) with escape sequences to display them in color on the terminal. The colors are defined by the environment variable GREP\_COLORS. The deprecated environment variable GREP\_COLOR is still supported, but its setting does not have priority. WHEN is never, always, or auto.  
     
    The --color (or --colour) option marks the matching string in color (on the console or in an  
    xterm window). Since grep prints out each entire line containing the matching pattern, this lets you  
    see exactly what is being matched.  
     
   -H, --with-filename  
   Print the file name for each match. This is the default when there is more than one file to search.  
     
   -A NUM, --after-context=NUM  
   Print NUM lines of trailing context after matching lines (including the matched line). Places a line containing a group separator (--) between contiguous groups of matches. With the -o or --only-matching option, this has no effect and a warning is given  
     
   -o, --only-matching  
   Print only the matched (non-empty) parts of a matching line, with each such part on a separate output line.  
     
   -F, --basic-regexp  
   Interpret PATTERN as a list of ixed strings (instead of regular expressions), separated by newlines, any of which is to be matched.  
   fgrep -- fast grep -- is the same as grep -F. It does a literal string search (no Regular Expressions), which generally speeds things up a bit  
     
   -m NUM, --max-count=NUM  
   Stop reading a file after NUM matching lines.  
     
   14. down things --- wget  
   wget url  
     
   15. install Qt5  
   download qt-opensource-linux-x64-5.12.4.run  
     
   yum install gcc-c++  
   yum -y install mesa-libGL-devel  
     
   16. accquire execute permission  
   chmod +x filename  
   chmod 777 filename  
     
   17. setting environment variable  
   append the /etc/profile with:  
   export PATH="$PATH:/opt/au1200\_rm/build\_tools/bin"  
   source /etc/profile  
     
   18. uname -a  
     
   19. wget url  
     
   20. firewall relative cmds  
   $ systemctl status firewalld  
   查看防火墙状态  
     
   开启防火墙  
   systemctl start firewalld  
     
   关闭防火墙  
   systemctl stop firewalld  
     
   查看当前firewall状态  
   firewall-cmd --state  
     
   重启firewall  
   firewall-cmd --reload  
     
   禁止开机启动  
   systemctl disable firewalld.service  
     
   开启端口  
   查看已经开放的端口：  
   firewall-cmd --list-ports  
     
   开启端口  
   firewall-cmd --zone=public --add-port=80/tcp --permanent  
     
   21. linux查看进程内存占用  
   ps -aux | grep xxx  
     
   22. decompress  
   tar -xvf filename  
     
   -j：有bz2属性的  
     
   23. error while loading shared libraries: libtest.so: cannot open shared object file: No such file or directory  
   1) 如果共享库文件安装到了/lib或/usr/lib目录下, 那么需执行一下ldconfig命令  
     
   ldconfig命令的用途, 主要是在默认搜寻目录(/lib和/usr/lib)以及动态库配置文件/etc/ld.so.conf内所列的目录下, 搜索出可共享的动态链接库(格式如lib*\*.so\**), 进而创建出动态装入程序(ld.so)所需的连接和缓存文件. 缓存文件默认为/etc/ld.so.cache, 此文件保存已排好序的动态链接库名字列表.  
     
   2) 如果共享库文件安装到了/usr/local/lib(很多开源的共享库都会安装到该目录下)或其它"非/lib或/usr/lib"目录下, 那么在执行ldconfig命令前, 还要把新共享库目录加入到共享库配置文件/etc/ld.so.conf中, 如下:  
     
   # cat /etc/ld.so.conf  
   include ld.so.conf.d/\*.conf  
   # echo "/usr/local/lib" >> /etc/ld.so.conf  
   # ldconfig  
     
   3) 如果共享库文件安装到了其它"非/lib或/usr/lib" 目录下,  但是又不想在/etc/ld.so.conf中加路径(或者是没有权限加路径). 那可以export一个全局变量LD\_LIBRARY\_PATH, 然后运行程序的时候就会去这个目录中找共享库.  
     
   LD\_LIBRARY\_PATH的意思是告诉loader在哪些目录中可以找到共享库. 可以设置多个搜索目录, 这些目录之间用冒号分隔开. 比如安装了一个mysql到/usr/local/mysql目录下, 其中有一大堆库文件在/usr/local/mysql/lib下面, 则可以在.bashrc或.bash\_profile或shell里加入以下语句即可:  
     
   export LD\_LIBRARY\_PATH=/usr/local/mysql/lib:$LD\_LIBRARY\_PATH  
     
   一般来讲这只是一种临时的解决方案, 在没有权限或临时需要的时候使用.  
     
   24. host  
   host -t A www.baidu.com  
     
   25. tcpdump  
   用于抓包  
   tcpdump输出的字节码中，每2个字符等于1字节，每4个字符组代表2字节：  
   0x0000: 4510 003c a5da 4000 4006 96cf 7f00 0001  
     
   26. route  
   $ route add -host 192.168.1.109 dev(?) th0(?)  
   $ route del -net 192.168.1.0 netmask(?) 255.255.255.0  
   $ route del default  
   $ route add default gw(?) 192.168.1.109 dev(?) eth0(?)  
     
   $ route -Cn 查看路由表缓：冲  
     
   27. iptable  
   用于过滤数据包，可以用来丢弃所有接收到的连接请求  
     
   28. netstat  
   是一个功能很强大的网络信息统计工具，它可以打印本地网卡接口上的全部连接，路由表信息，网卡接口信息等，我们主要利用显示TCP连接及其状态信息的功能  
     
   29. nc  
   用于创建网络连接  
     
   30. iperf  
   是一个测量网络状况的工具，-s选项表示将其作为服务器运行。iperf默认监听5001端口，并丢弃端口上接收到的所有数据，相当于一个discard服务器。  
     
   31. data  
   可以测试网络超时重传坚持的时间  
     
   32. wget  
   下载器  
     
   33. export  
     
   34. lsof - list open files  
   查看服务器程序监听的端口号  
     
   35. service  
   service是一个脚本程序(/usr/sbin/service)，它为/etc/init.d/目录下的众多服务器程序(比如httpd，vsftpd，sshd和mysqld等）的启动（start），停止（stop），重启（restart）等动作提供了一个统一的管理。  
     
   $ service squid3 restart  
     
   36. journalctl -xe  
   查看网络日志  
     
   37. ip addr  
   查看网卡信息  
     
   38. getconf  
   get configuration values  
     
   39. ulimit  
     
   40. sysctl  
     
   41. strace - trace system calls and signals  
   strace命令使我们能够清楚地查看每次系统调用发生地时机，以及相关参数的值，这比用gdb调试更方便  
     
   42. ifstat  
     
   43. ll  
   ll是ls -l的一个别名  
     
   44. rpm  
   rpm -ql redis  
   查询redis安装的所有文件  
     
   45. systemctl list-unit-files | grep redis  
   查看redis的自启动状态  
     
   46. systemctl enable redis  
   改为开机自启动，这个命令会复制一份redis.service文件到/usr/lib/systemd/system/目录下，这个目录中是所有开机自启动软件对应的service文件。  
     
   47. chown  
     
   48. chgrp  
     
   49. yum install  
   yum remove  
   yum update  
     
   50.