**Linux**

**一、查询Centos版本信息**

查看已经安装的CentOS版本信息

cat /etc/redhat-release 查看CentOS版本

|  |
| --- |
|  |

cat /proc/version

|  |
| --- |
|  |

proc 为process的缩写，里面存放与内核相关的文件。

uname -a 显示如下

|  |
| --- |
|  |

uname -r

|  |
| --- |
|  |

查看系统是32位或者64位的方法

1. getconf LONG\_BIT or getconf WORD\_BIT

输入：getconf LONG\_BIT

返回结果：64

输入：getconf WORD\_BIT

返回结果：32

分析：32位的系统中int类型和long类型一般都是4字节，64位的系统中int类型还是4字节的，但是long已变成了8字节inux系统中可用”getconf WORD\_BIT”和”getconf LONG\_BIT”获得word和long的位数。64位系统中应该分别得到32和64。

所以该系统为64为Linux系统。

2. file /bin/ls

/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, stripped

可以看到 ELF 64-bit LSB 所以该系统为64位

**一、常用命令**

**1、tar命令**

tar命令是Linux/Unix系统下的打包压缩工具，可以将多个文件合并为一个文件，打包后的文件后缀为“.tar”。

这里首先需要明确两个概念：打包和压缩。打包是指将一大堆文件或目录合并成一个总的文件；压缩则是将一个大容量的文件通过一些压缩算法（gzip、bzip2等命令）变成一个小容量的文件。Linux/Unix下的压缩命令通常都只能对一个文件进行压缩操作，所以通常需要使用tar命令对文件进行打包，然后进行压缩操作。

①tar命令的使用格式（语法）

tar 功能/参数 文件

②tar命令常用的功能

-c：创建新的tar文件 (--create 新建一个压缩文档，即打包)

-x：解开tar文件 (--extract,--get解压文件)

-t：列出tar文件中包含的文件的信息 (--list,查看压缩文档里的所有内容)

-r：附加新的文件到tar文件中 (--append 向压缩文档里追加文件)

-u：-update 更新原压缩包中的文件

注意：以上功能是独立的，一次只能使用一个。

③tar命令常用的参数

-C <目录>：这个选项用在解压缩，若要在特定目录解压缩，可以使用这个选项。

-z：使用gzip进行解压缩

-j：使用bzip2进行解压缩

-Z：使用compress进行解压缩

-v：显示解压缩执行过程

-f：指定要处理的文件名,注意，在f之后要立即接文档名，不要再加其他参数！

注意：如果需要使用-f参数，需要将f参数放在所有参数最后面，在f之后要立即接文件名，不能有其他参数。

例如：tar -zxvf /tmp/etc.tar.gz是正确的， tar -zxfv /tmp/etc.tar.gz则是错误的。

④tar命令使用实例

1.打包/etc目录下所有文件

tar -cvf /tmp/etc.tar /etc

2.打包并压缩/etc目录下所有文件

tar -zcvf /tmp/etc.tar.gz /etc //以gzip压缩

tar -jcvf /tmp/etc.tar.bz2 /etc //以bzip2压缩

tar -Zcvf /tmp/etc.tar.Z /etc //以compress压缩

3.查看打包压缩文件的文件内容

tar -tvf /tmp/etc.tar //查看打包文件

tar -ztvf /tmp/etc.tar.gz //查看以gzip压缩的文件

tar -jtvf /tmp/etc.tar.bz2 //查看以bzip2压缩的文件

tar -Ztvf /tmp/etc.tar.Z //查看以compress压缩的文件

4.解压打包压缩文件

tar -xvf /tmp/etc.tar //打开tar打包文件

tar -zxvf /tmp/etc.tar.gz //解压以gzip压缩的文件

tar -jxvf /tmp/etc.tar.bz2 //解压以bzip2压缩的文件

tar -Zxvf /tmp/etc.tar.Z //解压以compress压缩的文件

tar -zxvf apache-activemq-5.15.2-bin.tar.gz -C /usr/local/activeMQ/ //把指定的压缩文件解压到/usr/local/activeMQ/ 目录下。

**2、yum命令**

yum命令是在Fedora和RedHat以及SUSE中基于rpm的软件包管理器，它可以使系统管理人员交互和自动化地更细与管理RPM软件包，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载、安装。

|  |
| --- |
| install：安装rpm软件包；  update：更新rpm软件包；  check-update：检查是否有可用的更新rpm软件包；  remove：删除指定的rpm软件包；  list：显示软件包的信息；  search：检查软件包的信息；  info：显示指定的rpm软件包的描述信息和概要信息；  clean：清理yum过期的缓存；  shell：进入yum的shell提示符；  resolvedep：显示rpm软件包的依赖关系；  localinstall：安装本地的rpm软件包；  localupdate：显示本地rpm软件包进行更新；  deplist：显示rpm软件包的所有依赖关系。 |

**①使用yum安装和卸载软件，有个前提是yum安装的软件包都是rpm格式的。**

1.用YUM安装软件包命令：yum install ~

2.用YUM删除软件包命令：yum remove ~

其中~ 代表软件名

安装的命令是，yum install ~，yum会查询数据库，有无这一软件包，如果有，则检查其依赖冲突关系，如果没有依赖冲突，那么最好，下载安装;如果有，则会给出提示，询问是否要同时安装依赖，或删除冲突的包，你可以自己作出判断；

删除的命令是，yum remove ~，同安装一样，yum也会查询数据库，给出解决依赖关系的提示。

**②用yum查询想安装的软件**

我们常会碰到这样的情况，想安装一个软件，只知道它和某方面有关，但又不能确切知道它的名字。这时yum的查询功能就起作用了。我们可以用 yum search keyword这样的命令来进行搜索，比如我们要则安装一个Instant Messenger，但又不知到底有哪些，这时不妨用 yum search messenger这样的指令进行搜索，yum会搜索所有可用rpm的描述，列出所有描述中和messeger有关的rpm包，于是我们可能得到 gaim，kopete等等，并从中选择。

有时我们还会碰到安装了一个包，但又不知道其用途，我们可以用yum info packagename这个指令来获取信息。

|  |
| --- |
| 1.使用YUM查找软件包  命令：yum search ~  2.列出所有可安装的软件包  命令：yum list  3.列出所有可更新的软件包  命令：yum list updates  4.列出所有已安装的软件包  命令：yum list installed  5.列出所有已安装但不在Yum Repository 內的软件包  命令：yum list extras  6.列出所指定软件包  命令：yum list ～  7.使用YUM获取软件包信息  命令：yum info ～  8.列出所有软件包的信息  命令：yum info  9.列出所有可更新的软件包信息  命令：yum info updates  10.列出所有已安裝的软件包信息  命令：yum info installed  11.列出所有已安裝但不在Yum Repository 內的软件包信息  命令：yum info extras  12.列出软件包提供哪些文件  命令：yum provides~ |

**③清除YUM缓存**

yum 会把下载的软件包和header存储在cache中，而不会自动删除。如果我们觉得它们占用了磁盘空间，可以使用yum clean指令进行清除，更精确的用法是yum clean headers清除header，yum clean packages清除下载的rpm包，yum clean all 清除所有。

1.清除缓存目录(/var/cache/yum)下的软件包

命令：yum clean packages

2.清除缓存目录(/var/cache/yum)下的 headers

命令：yum clean headers

3.清除缓存目录(/var/cache/yum)下旧的 headers

命令：yum clean oldheaders

4.清除缓存目录(/var/cache/yum)下的软件包及旧的headers

命令：yum clean, yum clean all (= yum clean packages; yum clean oldheaders)

**④yum命令工具使用举例**

yum update 升级系统

yum install ～ 安装指定软件包

yum update ～ 升级指定软件包

yum remove ～ 卸载指定软件

yum grouplist 查看系统中已经安装的和可用的软件组，可用的可以安装

yum grooupinstall ～安装上一个命令显示的可用的软件组中的一个

yum grooupupdate ～更新指定软件组的软件包

yum grooupremove ～ 卸载指定软件组中的软件包

yum deplist ～ 查询指定软件包的依赖关系

yum list yum\\* 列出所有以yum开头的软件包

yum localinstall ～ 从硬盘安装rpm包并使用yum解决依赖

**3、vim命令**

vim三种模式：命令模式、插入模式、编辑模式。使用ESC或i或：来切换模式。

命令模式下：

:q 退出

:q! 强制退出

:wq 保存并退出

:set number 显示行号

:set nonumber 隐藏行号

/apache 在文档中查找apache 按n跳到下一个，shift+n上一个

yyp 复制光标所在行，并粘贴

h(左移一个字符←)、j(下一行↓)、k(上一行↑)、l(右移一个字符→)

**4、SCP命令**

scp命令用于在Linux下进行远程拷贝文件的命令，和它类似的命令有cp，不过cp只是在本机进行拷贝不能跨服务器，而且scp传输是加密的。可能会稍微影响一下速度。当你服务器硬盘变为只读read only system时，用scp可以帮你把文件移出来。另外，scp还非常不占资源，不会提高多少系统负荷，在这一点上，rsync就远远不及它了。虽然 rsync比scp会快一点，但当小文件众多的情况下，rsync会导致硬盘I/O非常高，而scp基本不影响系统正常使用。

**①语法：**

**scp(选项)(参数)**

**②选项：**

|  |
| --- |
| -1：使用ssh协议版本1；  -2：使用ssh协议版本2；  -4：使用ipv4；  -6：使用ipv6；  -B：以批处理模式运行；  -C：使用压缩；  -F：指定ssh配置文件；  -l：指定宽带限制；  -o：指定使用的ssh选项；  -P：指定远程主机的端口号；  -p：保留文件的最后修改时间，最后访问时间和权限模式；  -q：不显示复制进度；  -r：以递归方式复制。 |

**③参数：**

源文件：指定要复制的源文件。

目标文件：目标文件。格式为user@host：filename（文件名为目标文件的名称）。

**④例子：**

**从远处复制文件到本地目录：**

|  |
| --- |
| 例一：scp root@10.10.10.10:/opt/soft/nginx-0.5.38.tar.gz /opt/soft/  解释：从10.10.10.10机器上的/opt/soft/的目录中下载nginx-0.5.38.tar.gz 文件到本地/opt/soft/目录中。 |

**从远处复制到本地**

|  |
| --- |
| 例二：scp -r root@10.10.10.10:/opt/soft/mongodb /opt/soft/  从10.10.10.10机器上的/opt/soft/中下载mongodb目录到本地的/opt/soft/目录来。 |

**上传本地文件到远程机器指定目录**

|  |
| --- |
| 例三：scp /opt/soft/nginx-0.5.38.tar.gz [root@10.10.10.10:/opt/soft/scptest](mailto:root@10.10.10.10:/opt/soft/scptest)  复制本地/opt/soft/目录下的文件nginx-0.5.38.tar.gz到远程机器10.10.10.10的opt/soft/scptest目录。 |

**上传本地目录到远程机器指定目录**

|  |
| --- |
| 例四：scp -r /opt/soft/mongodb [root@10.10.10.10:/opt/soft/scptest](mailto:root@10.10.10.10:/opt/soft/scptest)  上传本地目录/opt/soft/mongodb到远程机器10.10.10.10上/opt/soft/scptest的目录中去。 |

**5、rm命令**

rm命令可以删除一个目录中的一个或多个文件或目录，也可以将某个目录及其下属的所有文件及其子目录均删除掉。对于链接文件，只是删除整个链接文件，而原有文件保持不变。

注意：使用rm命令要格外小心。因为一旦删除了一个文件，就无法再恢复它。所以，在删除文件之前，最好再看一下文件的内容，确定是否真要删除。rm命令可以用-i选项，这个选项在使用文件扩展名字符删除多个文件时特别有用。使用这个选项，系统会要求你逐一确定是否要删除。这时，必须输入y并按Enter键，才能删除文件。如果仅按Enter键或其他字符，文件不会被删除。

**①语法：**

**rm (选项)(参数)**

**②选项：**

|  |
| --- |
| -d：直接把欲删除的目录的硬连接数据删除成0，删除该目录；  -f：强制删除文件或目录；  -i：删除已有文件或目录之前先询问用户；  -r或-R：递归处理，将指定目录下的所有文件与子目录一并处理；  --preserve-root：不对根目录进行递归操作；  -v：显示指令的详细执行过程。 |

**③参数：**

|  |
| --- |
| 文件：指定被删除的文件列表，如果参数中含有目录，则必须加上-r或者-R选项。 |

**④例子：**

|  |  |
| --- | --- |
| 实例一：删除文件file，系统会先询问是否删除。  命令：  rm 文件名  输出：   |  | | --- | | [root@localhost test1]# ll  总计 4  -rw-r--r-- 1 root root 56 10-26 14:31 log.log  root@localhost test1]# rm log.log  rm：是否删除 一般文件 “log.log”? y  root@localhost test1]# ll  总计 0[root@localhost test1]# |   说明：  输入rm log.log命令后，系统会询问是否删除，输入y后就会删除文件，不想删除则数据n。 |

|  |  |
| --- | --- |
| 实例二：强行删除file，系统不再提示。  命令：  rm -f log1.log  输出：   |  | | --- | | [root@localhost test1]# ll  总计 4  -rw-r--r-- 1 root root 23 10-26 14:40 log1.log  [root@localhost test1]# rm -f log1.log  [root@localhost test1]# ll  总计 0[root@localhost test1]# | |

|  |  |
| --- | --- |
| 实例三：删除任何.log文件；删除前逐一询问确认  命令：  rm -i \*.log  输出：   |  | | --- | | [root@localhost test1]# ll  总计 8  -rw-r--r-- 1 root root 11 10-26 14:45 log1.log  -rw-r--r-- 1 root root 24 10-26 14:45 log2.log  [root@localhost test1]# rm -i \*.log  rm：是否删除 一般文件 “log1.log”? y  rm：是否删除 一般文件 “log2.log”? y  [root@localhost test1]# ll  总计 0[root@localhost test1]# | |

|  |  |
| --- | --- |
| 实例四：将 test1子目录及子目录中所有档案删除  命令：  rm -r test1  输出：   |  | | --- | | [root@localhost test]# ll  总计 24drwxr-xr-x 7 root root 4096 10-25 18:07 scf  drwxr-xr-x 2 root root 4096 10-26 14:51 test1  drwxr-xr-x 3 root root 4096 10-25 17:44 test2  drwxrwxrwx 2 root root 4096 10-25 17:46 test3  drwxr-xr-x 2 root root 4096 10-25 17:56 test4  drwxr-xr-x 3 root root 4096 10-25 17:56 test5  [root@localhost test]# rm -r test1  rm：是否进入目录 “test1”? y  rm：是否删除 一般文件 “test1/log3.log”? y  rm：是否删除 目录 “test1”? y  [root@localhost test]# ll  总计 20drwxr-xr-x 7 root root 4096 10-25 18:07 scf  drwxr-xr-x 3 root root 4096 10-25 17:44 test2  drwxrwxrwx 2 root root 4096 10-25 17:46 test3  drwxr-xr-x 2 root root 4096 10-25 17:56 test4  drwxr-xr-x 3 root root 4096 10-25 17:56 test5  [root@localhost test]# | |

|  |  |
| --- | --- |
| 实例五：rm -rf test2命令会将 test2 子目录及子目录中所有档案删除,并且不用一一确认  命令：  rm -rf test2  输出：   |  | | --- | | [root@localhost test]# rm -rf test2  [root@localhost test]# ll  总计 16drwxr-xr-x 7 root root 4096 10-25 18:07 scf  drwxrwxrwx 2 root root 4096 10-25 17:46 test3  drwxr-xr-x 2 root root 4096 10-25 17:56 test4  drwxr-xr-x 3 root root 4096 10-25 17:56 test5  [root@localhost test]# | |

|  |  |
| --- | --- |
| 实例六：删除以 -f 开头的文件  命令：  rm -- -f  输出：   |  | | --- | | [root@localhost test]# touch -- -f  [root@localhost test]# ls -- -f  -f[root@localhost test]# rm -- -f  rm：是否删除 一般空文件 “-f”? y  [root@localhost test]# ls -- -f  ls: -f: 没有那个文件或目录  [root@localhost test]#  也可以使用下面的操作步骤:  [root@localhost test]# touch ./-f  [root@localhost test]# ls ./-f  ./-f[root@localhost test]# rm ./-f  rm：是否删除 一般空文件 “./-f”? y  [root@localhost test]# | |

**6、rpm命令**

rpm命令是RPM软件包的管理工具。rpm原本是Red Hat Linux发行版专门用来管理Linux各项套件的程序，由于它遵循GPL规则且功能强大方便，因而广受欢迎。逐渐受到其他发行版的采用。RPM套件管理方式的出现，让Linux易于安装，升级，间接提升了Linux的适用度。

**①语法：**

**rpm(选项)(参数)**

**②选项：**

|  |
| --- |
| -a：查询所有套件；  -b<完成阶段><套件档>+或-t <完成阶段><套件档>+：设置包装套件的完成阶段，并指定套件档的文件名称；  -c：只列出组态配置文件，本参数需配合"-l"参数使用；  -d：只列出文本文件，本参数需配合"-l"参数使用；  -e<套件档>或--erase<套件档>：删除指定的套件；  -f<文件>+：查询拥有指定文件的套件；  -h或--hash：套件安装时列出标记；  -i：显示套件的相关信息；  -i<套件档>或--install<套件档>：安装指定的套件档；  -l：显示套件的文件列表；  -p<套件档>+：查询指定的RPM套件档；  -q：使用询问模式，当遇到任何问题时，rpm指令会先询问用户；  -R：显示套件的关联性信息；  -s：显示文件状态，本参数需配合"-l"参数使用；  -U<套件档>或--upgrade<套件档>：升级指定的套件档；  -v：显示指令执行过程；  -vv：详细显示指令执行过程，便于排错。 |

**③参数：**

软件包：指定要操纵的rpm软件包。

**④例子：**

**7、mv命令**

mv命令用来对文件或目录重新命名，或者将文件从一个目录移到另一个目录中。source表示源文件或目录，target表示目标文件或目录。如果将一个文件移到一个已经存在的目标文件中，则目标文件的内容将被覆盖。

mv命令可以用来将源文件移至一个目标文件中，或将一组文件移至一个目标目录中。源文件被移至目标文件有两种不同的结果：

①如果目标文件是到某一目录文件的路径，源文件会被移到此目录下，且文件名不变。

②如果目标文件不是目录文件，则源文件名（只能有一个）会变为此目标文件名，并覆盖己存在的同名文件。如果源文件和目标文件在同一个目录下，mv的作用就是改文件名。当目标文件是目录文件时，源文件或目录参数可以有多个，则所有的源文件都会被移至目标文件中。所有移到该目录下的文件都将保留以前的文件名。

注意事项：mv与cp的结果不同，mv好像文件“搬家”，文件个数并未增加。而cp对文件进行复制，文件个数增加了。

**①语法：**

**mv(选项)(参数)**

**②选项：**

|  |
| --- |
| --backup=<备份模式>：若需覆盖文件，则覆盖前先行备份；  -b：当文件存在时，覆盖前，为其创建一个备份；  -f：若目标文件或目录与现有的文件或目录重复，则直接覆盖现有的文件或目录；  -i：交互式操作，覆盖前先行询问用户，如果源文件与目标文件或目标目录中的文件同名，则询问用户是否覆盖目标文件。用户输入”y”，表示将覆盖目标文件；输入”n”，表示取消对源文件的移动。这样可以避免误将文件覆盖。  --strip-trailing-slashes：删除源文件中的斜杠“/”；  -S<后缀>：为备份文件指定后缀，而不使用默认的后缀；  --target-directory=<目录>：指定源文件要移动到目标目录；  -u：当源文件比目标文件新或者目标文件不存在时，才执行移动操作。 |

**③参数：**

--源文件：源文件列表。

--目标文件：如果“目标文件”是文件名则在移动文件的同时，将其改名为“目标文件”；如果“目标文件”是目录名则将源文件移动到“目标文件”下。

**④例子：**

|  |
| --- |
| 将文件ex3改名为new1 :  mv ex3 new1 |
| 将目录/usr/men中的所有文件移到当前目录（用.表示）中：  mv /usr/men/\* .  将多个文件移到指定目录，文件之间用空格隔开：  mv mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-dump redis-cli redis-server /usr/local/redis/bin |

**8、cp命令**

cp命令用来将一个或多个源文件或者目录复制到指定的目的文件或目录。它可以将单个源文件复制成一个指定文件名的具体的文件或一个已经存在的目录下。cp命令还支持同时复制多个文件，当一次复制多个文件时，目标文件参数必须是一个已经存在的目录，否则将出现错误。

**①语法：**

**cp(选项)(参数)**

**②选项：**

|  |
| --- |
| -a：此参数的效果和同时指定"-dpR"参数相同；  -d：当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录；  -f：强行复制文件或目录，不论目标文件或目录是否已存在；  -i：覆盖既有文件之前先询问用户；  -l：对源文件建立硬连接，而非复制文件；  -p：保留源文件或目录的属性；  -R/r：递归处理，将指定目录下的所有文件与子目录一并处理；  -s：对源文件建立符号连接，而非复制文件；  -u：使用这项参数后只会在源文件的更改时间较目标文件更新时或是名称相互对应的目标文件并不存在时，才复制文件；  -S：在备份文件时，用指定的后缀“SUFFIX”代替文件的默认后缀；  -b：覆盖已存在的文件目标前将目标文件备份；  -v：详细显示命令执行的操作。 |

**③参数：**

-- 源文件：制定源文件列表。默认情况下，cp命令不能复制目录，如果要复制目录，则必须使用-R选项；

-- 目标文件：指定目标文件。当“源文件”为多个文件时，要求“目标文件”为指定的目录。

**④例子：**

|  |
| --- |
| 如果把一个文件复制到一个目标文件中，而目标文件已经存在，那么，该目标文件的内容将被破坏。此命令中所有参数既可以是绝对路径名，也可以是相对路径名。通常会用到点.或点点..的形式。例如，下面的命令将指定文件复制到当前目录下：  cp ../mary/homework/assign . |
| 所有目标文件指定的目录必须是己经存在的，cp命令不能创建目录。如果没有文件复制的权限，则系统会显示出错信息。  将文件file复制到目录/usr/men/tmp下，并改名为file1  cp file /usr/men/tmp/file1 |
| 将目录/usr/men下的所有文件及其子目录复制到目录/usr/zh中:  cp -r /usr/men /usr/zh |
| 交互式地将目录/usr/men中的以m打头的所有.c文件复制到目录/usr/zh中:  cp -i /usr/men m\*.c /usr/zh |
| 我们在Linux下使用cp命令复制文件时候，有时候会需要覆盖一些同名文件，覆盖文件的时候都会有提示：需要不停的按Y来确定执行覆盖。文件数量不多还好，但是要是几百个估计按Y都要吐血了，于是折腾来半天总结了一个方法：  cp aaa/\* /bbb  复制目录aaa下所有到/bbb目录下，这时如果/bbb目录下有和aaa同名的文件，需要按Y来确认并且会略过aaa目录下的子目录。  cp -r aaa/\* /bbb  这次依然需要按Y来确认操作，但是没有忽略子目录。  cp -r -a aaa/\* /bbb  依然需要按Y来确认操作，并且把aaa目录以及子目录和文件属性也传递到了/bbb。  \cp -r -a aaa/\* /bbb  成功，没有提示按Y、传递了目录属性、没有略过目录。 |

9、unzip命令

unzip命令用于解压缩由zip命令压缩的“.zip”压缩包。

**①语法：**

**unzip(选项)(参数)**

**②选项：**

|  |
| --- |
| -c：将解压缩的结果显示到屏幕上，并对字符做适当的转换；  -f：更新现有的文件；  -l：显示压缩文件内所包含的文件；  -p：与-c参数类似，会将解压缩的结果显示到屏幕上，但不会执行任何的转换；  -t：检查压缩文件是否正确；  -u：与-f参数类似，但是除了更新现有的文件外，也会将压缩文件中的其他文件解压缩到目录中；  -v：执行时显示详细的信息；  -z：仅显示压缩文件的备注文字；  -a：对文本文件进行必要的字符转换；  -b：不要对文本文件进行字符转换；  -C：压缩文件中的文件名称区分大小写；  -j：不处理压缩文件中原有的目录路径；  -L：将压缩文件中的全部文件名改为小写；  -M：将输出结果送到more程序处理；  -n：解压缩时不要覆盖原有的文件；  -o：不必先询问用户，unzip执行后覆盖原有的文件；  -P<密码>：使用zip的密码选项；  -q：执行时不显示任何信息；  -s：将文件名中的空白字符转换为底线字符；  -V：保留VMS的文件版本信息；  -X：解压缩时同时回存文件原来的UID/GID；  -d<目录>：指定文件解压缩后所要存储的目录；  -x<文件>：指定不要处理.zip压缩文件中的哪些文件；  -Z：unzip-Z等于执行zipinfo指令。 |

10、tail命令

**tail**命令用于输入文件中的尾部内容。tail命令默认在屏幕上显示指定文件的末尾10行。如果给定的文件不止一个，则在显示的每个文件前面加一个文件名标题。如果没有指定文件或者文件名为“-”，则读取标准输入。

**①语法：**

tail [ -f ] [ -c Number | -n Number | -m Number | -b Number | -k Number ] [ File ]

**②选项：**

参数解释：

|  |
| --- |
| -f 该参数用于监视File文件增长。  -c Number 从 Number 字节位置读取指定文件  -n Number 从 Number 行位置读取指定文件。  -m Number 从 Number 多字节字符位置读取指定文件，比方你的文件假设包括中文字，假设指定-c参数，可能导致截断，但使用-m则会避免该问题。  -b Number 从 Number 表示的512字节块位置读取指定文件。  -k Number 从 Number 表示的1KB块位置读取指定文件。  File 指定操作的目标文件名称  上述命令中，都涉及到number，假设不指定，默认显示10行。Number前面可使用正负号，表示该偏移从顶部还是从尾部開始计算。  tail 可运行文件一般在/usr/bin/以下。 |

**④例子：**

|  |
| --- |
| tail file （显示文件file的最后10行）  tail +20 file （显示文件file的内容，从第20行至文件末尾）  tail -c 10 file （显示文件file的最后10个字符）  tail -f -n 100 file （动态显示文件file的100行） |

11、netstat命令

查看所有端口号占用信息：

|  |
| --- |
|  |

12、centos7防火墙

查看已经开放的端口：

firewall-cmd --list-ports

|  |
| --- |
|  |

查看端口开启状态：

firewall-cmd --query-port=9998/tcp

开启端口：

firewall-cmd --add-port=9998/tcp --permanent

|  |
| --- |
| 开启端口  firewall-cmd --zone=public --add-port=80/tcp --permanent  命令含义：  –zone #作用域  –add-port=80/tcp #添加端口，格式为：端口/通讯协议  –permanent #永久生效，没有此参数重启后失效 |

关闭端口：

firewall-cmd --remove-port=9998/udp --permanent

重新开启防火墙：

systemctl start firewalld （重启后还是没用）

chkconfig firewalld on （开机就启动防火墙）

|  |
| --- |
|  |

重新加载配置：

firewall-cmd --reload

|  |
| --- |
| firewall-cmd --reload #重启firewall  systemctl stop firewalld.service #停止firewall  systemctl disable firewalld.service #禁止firewall开机启动  firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running） |
|  |

13、centos7系统命令

Linux 的五个重启命令

　　1、shutdown 2、poweroff

　　3、init 4、reboot 5、halt

14、crontab命令

Linux crontab是用来定期执行程序的命令。

crond命令每分锺会定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作。

而linux任务调度的工作主要分为以下两类：

1、系统执行的工作：系统周期性所要执行的工作，如备份系统数据、清理缓存

2、个人执行的工作：某个用户定期要做的工作，例如每隔10分钟检查邮件服务器是否有新信，这些工作可由每个用户自行设置

语法：

|  |
| --- |
| crontab [ -u user ] { -l | -r | -e } |

**说明：**

crontab 是用来让使用者在固定时间或固定间隔执行程序之用，换句话说，也就是类似使用者的时程表。

-u user 是指设定指定 user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 -u user 的话，就是表示设定自己的时程表。

参数说明：

-e : 执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数来指定使用那个文字编辑器(比如说 setenv VISUAL joe)

-r : 删除目前的时程表

-l : 列出目前的时程表

时程表的格式如下：

|  |
| --- |
| f1 f2 f3 f4 f5 program |

>> 其中 f1 是表示分钟，f2 表示小时，f3 表示一个月份中的第几日，f4 表示月份，f5 表示一个星期中的第几天。program 表示要执行的程序。

>> 当 f1 为 \* 时表示每分钟都要执行 program，f2 为 \* 时表示每小时都要执行程序，其余类推。

>> 当 f1 为 a-b 时表示从第 a 分钟到第 b 分钟这段时间内要执行，f2 为 a-b 时表示从第 a 到第 b 小时都要执行，其余类推。

>> 当 f1 为 \*/n 时表示每 n 分钟个时间间隔执行一次，f2 为 \*/n 表示每 n 小时个时间间隔执行一次，其余类推。

>> 当 f1 为 a, b, c,... 时表示第 a, b, c,... 分钟要执行，f2 为 a, b, c,... 时表示第 a, b, c...个小时要执行，其余类推。

如果crontab程序没有启动，则需要手工启动：

ps -el | grep crond

|  |
| --- |
|  |

/etc/rc.d/init.d/crond start

|  |
| --- |
| crontab -e  这样可以已编辑模式打开个人的crontab配置文件，然后加入一下这行：  0 0 \* \* \* /home/linrui/XXXXXXXX.sh  这将会在每天凌晨运行 指定的.sh文件 |
|  |

|  |
| --- |
| 0 \*/2 \* \* \* /sbin/service httpd restart 意思是每两个小时重启一次apache  50 7 \* \* \* /sbin/service sshd start 意思是每天7：50开启ssh服务  50 22 \* \* \* /sbin/service sshd stop 意思是每天22：50关闭ssh服务  0 0 1,15 \* \* fsck /home 每月1号和15号检查/home 磁盘  1 \* \* \* \* /home/bruce/backup 每小时的第一分执行 /home/bruce/backup这个文件  00 03 \* \* 1-5 find /home "\*.xxx" -mtime +4 -exec rm {} \; 每周一至周五3点钟，在目录/home中，查找文件名为\*.xxx的文件，并删除4天前的文件。  30 6 \*/10 \* \* ls 意思是每月的1、11、21、31日是的6：30执行一次ls命令 |

15、chmod命令

chmod命令用来变更文件或目录的权限。在UNIX系统家族里，文件或目录权限的控制分别以读取、写入、执行3种一般权限来区分，另有3种特殊权限可供运用。用户可以使用chmod指令去变更文件与目录的权限，设置方式采用文字或数字代号皆可。符号连接的权限无法变更，如果用户对符号连接修改权限，其改变会作用在被连接的原始文件。

语法：

|  |
| --- |
| chmod [-cfvR] [--help] [--version] mode file... |

权限范围的表示法如下：

u 表示该文件的拥有者，g 表示与该文件的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示这三者皆是。

+ 表示增加权限、- 表示取消权限、= 表示唯一设定权限。

r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

其他参数说明：

-c : 若该文件权限确实已经更改，才显示其更改动作

-f : 若该文件权限无法被更改也不要显示错误讯息

-v : 显示权限变更的详细资料

-R : 对目前目录下的所有文件与子目录进行相同的权限变更(即以递回的方式逐个变更)

--help : 显示辅助说明

--version : 显示版本

|  |
| --- |
| 将文件 file1.txt 设为所有人皆可读取 :  chmod ugo+r file1.txt  将文件 file1.txt 设为所有人皆可读取 :  chmod a+r file1.txt  将文件 file1.txt 与 file2.txt 设为该文件拥有者，与其所属同一个群体者可写入，但其他以外的人则不可写入 :  chmod ug+w,o-w file1.txt file2.txt  将 ex1.py 设定为只有该文件拥有者可以执行 :  chmod u+x ex1.py  将目前目录下的所有文件与子目录皆设为任何人可读取 :  chmod -R a+r \*  此外chmod也可以用数字来表示权限如 :  chmod 777 file |

16、chown命令

利用 chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。

一般来说，这个指令只有是由系统管理者(root)所使用，一般使用者没有权限可以改变别人的文件拥有者，也没有权限可以自己的文件拥有者改设为别人。只有系统管理者(root)才有这样的权限。

语法：

|  |
| --- |
| chown [-cfhvR] [--help] [--version] user[:group] file... |

参数 :

user : 新的文件拥有者的使用者 ID

group : 新的文件拥有者的使用者群体(group)

-c : 若该文件拥有者确实已经更改，才显示其更改动作

-f : 若该文件拥有者无法被更改也不要显示错误讯息

-h : 只对于连结(link)进行变更，而非该 link 真正指向的文件

-v : 显示拥有者变更的详细资料

-R : 对目前目录下的所有文件与子目录进行相同的拥有者变更(即以递回的方式逐个变更)

--help : 显示辅助说明

--version : 显示版本

|  |
| --- |
| 将文件 file1.txt 的拥有者设为 users 群体的使用者 jessie :  chown users:jessie file1.txt  将目前目录下的所有文件与子目录的拥有者皆设为 users 群体的使用者 lamport :  chown -R lamport:users \* |

**二、Linux权限**

1、权限介绍

首先，我们通过linux的ls命令操作获得每个文件的权限，如下图：

|  |
| --- |
|  |

1 表示连接的文件数

chengxp表示用户

root表示用户所在的组

1127 表示文件大小（字节）

6月 4 21:45 表示最后修改日期

authz 表示文件名

由上图可以看出：最左边由十个字母或者短杠组成，其中：

**第一位表示Linux的文件类型**：

文件的类型可以分为以下几类：

>> 普通文件（regular file）第一个字符为[ - ] ,包括：

①纯文本档(ASCII)：这是Linux系统中最多的一种文件类 型，称为纯文本档。是因为内容为我们⼈人类可以直接读到的数据，

例如数字、字母等等。

②二进制文件(binary)： 你的Linux中的 可执行文件(scripts, 文字型批处理文件不算)就是这种格式的。

③数据格式文件(data)：有些程序在运作的过程当中会读取某些特 定格式的档案，那些特定格式的档案可以被称为数据⽂文件 (data file)。一般用cat看到的都是乱码。

>> 目录(directory)： 第一个字符为[ d ]

>> 连接文件（link）第⼀一个字符为[ l ]

>> 设备和设备文件 包括：

①块设备文件，即一些存储文件，如硬盘、软盘等，第一个字符为 [ b ］

②字符设备文件，即一些串行端口的接口文件，如键盘、鼠标等， 第一个字符为[ c ]

>> 套接字（sockets）第一个字符为[ s ] 。

还被称为数据接口文件，这种类型的文件通常被用在网络上的数 据连接。我们可以启动一个程序来监听客户端的请求，而客户端就 可以通过数据接口文件来进行数据通信。

>> 管道（FIFO，pipe）第一个字符为[ p ］

**第二位开始，三位为一组**：

可以将上述分为三组，分别是第一组：rwx或者rw-，第二组：rwx或者rw-，第三组：rwx或者r--。

在linux中，第一组称为拥有者，第二组称为所属组，第三组称为其他用户。即就是：

文件访问者的分类（人）

a)文件和文件目录的所有者：u---User

一般为文件的创建者，谁创建了该文件，就天然的成为该文件的所有者，用ls ‐ahl命令可以看到文件的所有者,也可以使用chown 用户名 文件名来修改文件的所有者

b)文件和文件目录的所有者所在的组的用户：g---Group

当某个用户创建了一个文件后，这个文件的所在组就是该用户所在的组，用ls ‐ahl命令可以看到文件的所有组，也可以使用chgrp 组名 文件名来修改文件所在的组

c)其它用户：o---Others

除开文件的所有者和所在组的用户外，系统的其它用户都是文件的其它组。

每个权限部分可选字符选项如下：

r： read对文件而言，具有读取文件内容的权限；对目录来说，具有浏览该目录信息的权限。

w：write对文件而言，具有修改文件内容的权限；对目录来说具有删除移动目录内文件的权限。

S：set如果在所有者权限部分，表示该文件不能被执行和设置为set-user-ID模式。如果在组权限部分，表示该文件不能被执行和设置为set-group-ID模式。

s：如果该字符在所有者权限部分，表示该文件可执行和设置为set-user-ID模式。如果设置在组权限部分，表示该文件可执行和设置为set-group-ID模式。

x：execute对⽂文件⽽而⾔言，具有执⾏行⽂文件的权限； 对⺫⽬目录来说，具有进⼊入⺫⽬目录的权限

-：无某（读，写，执行，setuid等）权限

以下两个字符只能用在第三部分其他人权限部分：

T 沾滞位设置（1000模式），但不能执行或者搜索。

t 沾滞位设置（1000模式），并可以执行或搜索。（该字符较常见一般用于授予目录，作用为任何人可以在该目录中创建文件，但是该文件只有创建者可删除。见/tmp目录权限）。

2、文件访问权限的相关设置方法

⑴chmod

功能：设置文件的访问权限

格式：chmod [参数] 权限文件名

参数： R -> 递归修改目录文件的权限

V -> 显示指令执行过程

说明：只有文件的拥有者和root才可以改变文件的权限

chmod命令权限值的格式：① 用户表示符+/-=权限字符

+ 向权限范围增加权限代号所表⽰示的权限

- 向权限范围取消权限代号所表⽰示的权限

= 向权限范围赋予权限代号所表⽰示的权限

用户符号：

u 拥有者

g 拥有者同组用户

o 其它⽤用户

a 所有用户

|  |
| --- |
|  |

⑵chown

功能：修改文件的拥有者

格式：chown [参数] 用户名 文件名

⑶chgrp

功能：修改文件或目录的所属组

格式：chgrp [参数] 用户组名 文件名

参数：-R 递归修改文件或目录的所属组

⑷umask

功能：查看或修改文件掩码

新建文件夹默认权限=0666-权限掩码

新建目录默认权限=0777-权限掩码

格式：umask 权限值

说明：将现有的存取权限减去权限掩码后，即可产生建立文件时预设权限。

超级用户默认掩码值为0022，普通用户默认为0002。

3、linux系统644、755、777权限详解

常用的linux文件权限：

444 r--r--r--

600 rw-------

644 rw-r--r--

666 rw-rw-rw-

700 rwx------

744 rwxr--r--

755 rwxr-xr-x

777 rwxrwxrwx

从左至右，1-3位数字代表文件所有者的权限，4-6位数字代表同组用户的权限，7-9数字代表其他用户的权限。

而具体的权限是由数字来表示的，读取的权限等于4，用r表示；写入的权限等于2，用w表示；执行的权限等于1，用x表示；

通过4、2、1的组合，得到以下几种权限：0（没有权限）；4（读取权限）；5（4+1 | 读取+执行）；6（4+2 | 读取+写入）；7（4+2+1 | 读取+写入+执行）

以755为例：

1-3位7等于4+2+1，rwx，所有者具有读取、写入、执行权限；

4-6位5等于4+1+0，r-x，同组用户具有读取、执行权限但没有写入权限；

7-9位5，同上，也是r-x，其他用户具有读取、执行权限但没有写入权限。

**二、使用CentOS7卸载自带jdk安装自己的JDK1.8**

Centos7安装：

<http://blog.csdn.net/hui_2016/article/details/68927487>

JDK下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

****通过xshell工具成功连接安装好的虚拟机之后可通过** rpm -qa|grep java**命令来查询出系统自带的jdk（带箭头的四个就是系统自带的）**注：不带箭头的那三个不要删掉**

|  |
| --- |
|  |

****然后通过    rpm -e --nodeps   后面跟系统自带的jdk名    这个命令来删除系统自带的jdk，****

****例如：**rpm -e --nodeps java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86\_64  
           rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86\_64  
           rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.111-2.6.7.8.el7.x86\_64  
           rpm -e --nodeps java-1.7.0-openjdk-1.7.0.111-2.6.7.8.el7.x86\_64**

|  |
| --- |
| **rpm -e --nodeps java-1.8.0-openjdk-1.8.0.65-3.b17.el7.x86\_64**  **rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86\_64**  **rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.65-3.b17.el7.x86\_64**  **rpm -e --nodeps java-1.7.0-openjdk-1.7.0.91-2.6.2.3.el7.x86\_6** |

****接下来就是要安装自己的jdk了，通过命令**** **cd  /usr/local/**进入local目录，并通过**ll（两个小写的L）命令或者** **ls** **命令（ ll 本身不是命令，只是 ls -l 命令的一个别名，不过无所谓了，都是一样查看的）列出当前目录下得所有非隐含的文件，如果想要看到隐含（以  . 开头的，如  .test.txt）文件信息可通过  ll -a** **来查看**

|  |
| --- |
|  |

****进入local目录之后 通过** mkdir  java  **命令来创建java目录存放自己的jdk****

****（扩展：如果你想一次性在同一级目录下创建多个平级的目录可以通过**mkdir  brother1  brother2 (如要创建更多就在后面加上去就可以了，中间用空格隔开)这样的命令来创建**，如果要一次创建父子目录（parent/child）可以通过****

**mkdir -p parent/child/grandson** **来创建）**

通过 tar -zxvf {安装包名} 命令解压jdk 如：tar -zxvf jdk-8u11-linux-x64.tar.gz 我个人比较喜欢把 zxvf 转成中文念，叫“至小威风” 这样我觉得方便记偷笑，不喜勿喷哦

通过 vim /etc/profile 命令打开 profile 文件盘配置环境变量。

|  |
| --- |
|  |

**打开之后按 i   进入 insert （插入）模式，在文件末尾添加上环境变量**

|  |
| --- |
| **export JAVA\_HOME=/usr/local/java/jdk1.8.0\_151**  **export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar**  **export PATH=$PATH:$JAVA\_HOME/bin** |

**添加完之后保存并退出，保存并退出的命令有两种 第一种是：  按住shift** **键然后连按两次z（这是我常用的，因为它方便快速），第二种是：wq 命令，有一种是不保存退出的命令：q   ，注意：以上三种命令都是在非插入模式（插入模式下按键盘左上角的ESC键退出插入模式就是非插入模式了）下操作。**

**保存完之后输入：source /etc/profile  命令使刚才配置的环境变量生效**

|  |
| --- |
|  |

**接下来我们通过** **java -version命令来查看你安装的jdk信息**

|  |
| --- |
|  |

**三、Linux上安装Tomcat**

用xftp上传tomcat的tar包到Linux上或从另一个Linux上远程复制一份过来，

|  |
| --- |
|  |

解压缩到 /usr/local/下，

|  |
| --- |
|  |

把解压后的文件夹重命名为tomcat：

|  |
| --- |
|  |

进入到 /usr/local/tomcat/conf 文件夹下：

|  |
| --- |
|  |

编辑tomcat-users.xml文件：

|  |
| --- |
|  |

在最底部加入访问tomcat主页的配置，然后保存：

|  |
| --- |
| <role rolename="admin-gui"/>  <role rolename="admin-script"/>  <role rolename="manager-gui"/>  <role rolename="manager-script"/>  <role rolename="manager-jmx"/>  <role rolename="manager-status"/>  <user password="admin" roles=" admin-gui ,admin-script, manager-gui,manager-script,manager-jmx,manager-status " username="admin"/> |
|  |

最后开放8080端口：

|  |
| --- |
|  |
| firewall-cmd --zone=public --add-port=8080/tcp --permanent  firewall-cmd --reload  firewall-cmd --query-port=8080/tcp |

**查询tomcat启动日志**

tail -f -n /opt/software/tomcat/logs/catalina.out

1、tomcat管理页面403 Access Denied的解决方法

安装tomcat，配置好tomcat环境变量以后，访问manager app页面，出现403 Access Denied错误，解决的方法如下：

首先在conf/tomcat-users.xml文件里面，在</tomcat-users>前面添加如下代码：

|  |
| --- |
| <role rolename="admin-gui"/>  <role rolename="admin-script"/>  <role rolename="manager-gui"/>  <role rolename="manager-script"/>  <role rolename="manager-jmx"/>  <role rolename="manager-status"/>  <user password="admin" roles=" admin-gui ,admin-script, manager-gui,manager-script,manager-jmx,manager-status " username="admin"/> |

然后重启tomcat，重新访问manager app页面，如果此时仍然出现403 Access Denied错误，那么就是访问的ip地受到了限制，我们打开/webapps/manager/META-INF/目录下context.xml文件，不是conf/目录下的context.xml文件，一定不要搞错了

vim ./webapps/manager/META-INF/context.xml

|  |
| --- |
| 我们将里面的内容<Valve className="org.apache.catalina.valves.RemoteAddrValve"  allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|\d+\.\d+\.\d+\.\d+" /> 注释掉或者修改为: |

**四、Linux上安装SVN**

1、下载安装SVN版本 yum install subversion

|  |
| --- |
|  |

2、查看安装版本 svnserve --version

|  |
| --- |
| **rpm -qa|grep subversion** |

3、创建SVN版本库目录 mkdir -p /usr/local/svn/springcloud

|  |
| --- |
|  |

4、创建版本库 svnadmin create /usr/local/svn/springcloud

|  |
| --- |
|  |

执行了这个命令之后会在/var/svn/svnrepos目录下生成如下这些文件：conf db format hooks locks README.txt

|  |
| --- |
| cd /usr/local/svn/springcloud/ |

5、进入conf目录（进行版本库配置） cd conf

authz：文件是权限控制文件

passwd：是帐号密码文件

svnserve.conf ：SVN服务配置文件

|  |
| --- |
| IMG_256 |

6、设置账号密码 vim passwd

在[users]块中添加用户和密码，格式：帐号=密码，如helijie=123456

|  |
| --- |
| IMG_256 |

7、设置权限 vim authz

在末尾添加如下代码：

[/]

helijie=rw

意思是helijie对版本库有读写权限

|  |
| --- |
| IMG_256 |

8、修改svnserve.conf文件 vim svnserve.conf

打开下面的几个注释：

anon-access = read #匿名用户可读

auth-access = write #授权用户可写

password-db = passwd #使用哪个文件作为账号文件

authz-db = authz #使用哪个文件作为权限文件

realm = svnrepos # 认证空间名，版本库所在目录

|  |
| --- |
| IMG_256 |

9、启动svn版本库 svnserve -d -r /usr/local/svn/ （停止SVN命令 killall svnserve）

|  |
| --- |
|  |

10、开放3690端口

SVN默认的打开端口是3690

netstat -antp | grep svn

|  |
| --- |
|  |

centos7 打开防火墙端口

|  |
| --- |
| firewall-cmd --permanent --add-port=3690/tcp  firewall-cmd --reload  firewall-cmd --query-port=3690/tcp |
|  |

11、window真机测试 svn://192.168.0.134/svnTest

|  |
| --- |
|  |

12、linux svn 使用

|  |
| --- |
|  |

13、设置Linux启动时启动SVN

方式一：直接添加启动命令：

给/etc/rc.d/rc.local文件可执行权限:

chmod +x /etc/rc.d/rc.local

|  |
| --- |
|  |

在rc.local中添加svn启动命令。

|  |
| --- |
| vim /etc/rc.d/rc.local  svnserve -d -r /usr/local/svn/ |

方式二：编写shell脚本

shell脚本

|  |
| --- |
|  |

在rc.local添加shell脚本执行命令 : sh /usr/local/svn/runsvn.sh

|  |
| --- |
|  |

14、常见问题

|  |
| --- |
|  |

14、安装支持Http协议

现有的svn已经支持svn协议，并且版本库、账户和权限已设置好。

yum install httpd

yum install mod\_dav\_svn

|  |
| --- |
|  |

设置开机启动：chkconfig httpd on

启动Apache：service httpd start

查看Apache版本: httpd -version

|  |
| --- |
|  |

cd /etc/httpd/modules/

|  |
| --- |
| #查看是否有mod\_dav\_svn.so和mod\_authz\_svn.so模块,如果有，说明mod\_dav\_svn安装成功！ |

如果启动Apache提示错误：

httpd:httpd: Could not reliably determine the server's fully qualif domain name, using ::1 for ServerName

解决办法：

vi /etc/httpd/conf/httpd.conf #编辑

ServerName www.example.com:80 #去掉前面的注释

:wq! #保存退出

启动Apache http服务报错:

|  |
| --- |
|  |
| 解决办法:  SElinux未禁用。  临时禁用SELinux : setenforce 0 |

配置svn支持http访问：

创建账号密码认证文件

htpasswd -cm /usr/local/svn/springcloud/conf/http\_passwd chengxp

|  |
| --- |
|  |
| 注意：  /home/svn/conf/目录下面passwd文件是svnserve独立服务器使用的认证文件，密码没有加密，明文显示。  /home/svn/conf/目录下面http\_passwd文件是Apache的http模式使用的认证文件，密码使用MD5加密。  passwd和http\_passwd文件中，账号密码必须设置相同。 |

设置Apache配置文件

vim /etc/httpd/conf/httpd.conf

|  |
| --- |
| ServerName www.example.com:80 #去掉前面的注释 |

新建subversion.con文件

vim /etc/httpd/conf.d/subversion.conf

|  |
| --- |
| LoadModule dav\_svn\_module modules/mod\_dav\_svn.so  LoadModule authz\_svn\_module modules/mod\_authz\_svn.so  <location /springcloud>  DAV svn  SVNPath /usr/local/svn/springcloud/  #SVNParentPath /usr/local/svn  # # Limit write permission to list of valid users.  # <LimitExcept GET PROPFIND OPTIONS REPORT>  # # Require SSL connection for password protection.  # # SSLRequireSSL  #  AuthType Basic  AuthName "Authorization SVN"  AuthzSVNAccessFile /usr/local/svn/springcloud/conf/authz  AuthUserFile /usr/local/svn/springcloud/conf/http\_passwd  Require valid-user  # </LimitExcept>  </Location> |
| 说明：DAV为svn支持文件，svnpath为svn版本库目录，只有一个版本库时使用svnpath，如果有多个版本时使用 svnparentpath和svnlistparentpath配合使用；AuthaSVNAccessFile为权限文件目录，跟svn协议共同使用一个权限文件即可；AuthUserFile为用户认证文件，填入第3步配置的认证文件目录；Require为用户是否需要验证。 |

为Apache设置目录权限

chown apache:apache /usr/local/svn/springcloud/ -R

为Apache设置数据仓库的所有者权限:

重启Apache服务

service httpd restart

**Linux下修改Apache即httpd的端口**

停止httpd服务

service httpd stop

查看httpd的状态，确认httpd服务已经停止

service httpd status

查找httpd.conf配置文件的存放路径

find /etc/httpd/ -name \*conf

编辑httpd.conf配置文件

vim /etc/httpd/conf/httpd.conf

查找Listen配置段,修改Listen后面的端口，默认是80，这里修改为90

|  |
| --- |
|  |

修改后需要再次启动httpd服务

service httpd start

**四、Linux安装SVN +http+多版本库**

<https://www.jb51.net/article/69232.htm>

**1、安装Apache**

yum install httpd

yum install mod\_dav\_svn

启动Apache：service httpd start

设置开机启动：chkcofing httpd on

查看Apache版本: httpd -version

|  |
| --- |
|  |

cd /etc/httpd/modules/

|  |
| --- |
| #查看是否有mod\_dav\_svn.so和mod\_authz\_svn.so模块,如果有，说明mod\_dav\_svn安装成功！ |

2、安装SVN

yum install subversion #使用yum命令在线安装

svnserve --version #查看svn版本信息

|  |
| --- |
|  |

3、配置SVN

#创建svn版本库存放目录

mkdir -p /usr/local/svn

#进入目录

cd /usr/local/svn

#创建svn版本库springboot、springcloud、gradle-springboot、gradle-springcloud。

svnadmin create /usr/local/svn/springboot

svnadmin create /usr/local/svn/springcloud

svnadmin create /usr/local/svn/gradle-springboot

svnadmin create /usr/local/svn/gradle-springcloud

设置配置文件

mkdir -p /usr/local/svn/conf #创建配置文件目录

#拷贝账号密码、权限、全局 配置文件模板

cp /usr/local/svn/springboot/conf/passwd /usr/local/svn/conf/

cp /usr/local/svn/springboot/conf/authz /usr/local/svn/conf/

cp /usr/local/svn/springboot/conf/svnserve.conf /usr/local/svn/conf/

|  |
| --- |
|  |

vim /usr/local/svn/conf/passwd

|  |
| --- |
|  |

vim /usr/local/svn/conf/authz

|  |
| --- |
|  |

vim /usr/local/svn/conf/svnserve.conf

|  |
| --- |
|  |

启动SVN

#--config-file后面跟全局配置参数文件

svnserve -d -r /usr/local/svn --config-file /usr/local/svn/conf/svnserve.conf --listen-port 3690

ps -ef|grep svn|grep -v grep #查看进程

netstat -ln |grep 3690 #检查端口

|  |
| --- |
|  |

设置svn服务开机启动

vim /etc/init.d/svn

|  |
| --- |
| #!/bin/sh  #chkconfig: 2345 85 85  # processname: svn  svn\_bin=/usr/bin  svn\_port=3690  svn\_home=/usr/local/svn  svn\_config=/usr/local/svn/conf/svnserve.conf  if [ ! -f "$svn\_bin/svnserve" ]  then  echo "svnserver startup: cannot start"  exit  fi  case "$1" in  start)  echo "Starting svnserve..."  $svn\_bin/svnserve -d -r $svn\_home --config-file $svn\_config --listen-port $svn\_port  echo "Successfully!"  ;;  stop)  echo "Stoping svnserve..."  killall svnserve  echo "Successfully!"  ;;  restart)  $0 stop  $0 start  ;;  \*)  echo "Usage: svn { start | stop | restart } "  exit 1  esac |
|  |

#添加执行权限

chmod +x /etc/init.d/svn

#开机自启动

chkconfig --add svn

chkconfig svn on

#启动

service svn start

配置svn支持http访问

创建账号密码认证文件

htpasswd -cm /usr/local/svn/conf/http\_passwd chengxp

|  |
| --- |
| 注意：  /usr/local/svn/conf/目录下面passwd文件是svnserve独立服务器使用的认证文件，密码没有加密，明文显示。  /usr/local/svn/conf/目录下面http\_passwd文件是Apache的http模式使用的认证文件，密码使用MD5加密。  passwd和http\_passwd文件中，账号密码必须设置相同。 |

设置Apache配置文件

vim /etc/httpd/conf.d/subversion.conf

|  |
| --- |
| LoadModule dav\_svn\_module modules/mod\_dav\_svn.so  LoadModule authz\_svn\_module modules/mod\_authz\_svn.so  <location /svn>  DAV svn  #SVNPath  SVNParentPath /usr/local/svn/  SVNListParentPath on  # # Limit write permission to list of valid users.  # <LimitExcept GET PROPFIND OPTIONS REPORT>  # # Require SSL connection for password protection.  # # SSLRequireSSL  #  AuthType Basic  AuthName "Authorization SVN"  AuthzSVNAccessFile /usr/local/svn/conf/authz  AuthUserFile /usr/local/svn/conf/http\_passwd  Require valid-user  # </LimitExcept>  </Location> |

#设置svn目录所有者为Apache服务运行账号apache

chown apache:apache /usr/local/svn -R

重启Apache服务

service httpd restart

项目目录访问:

svn://192.168.0.151/springcloud

http://192.168.0.151/svn/springcloud

|  |
| --- |
|  |

|  |
| --- |
| svnserve -d -r /usr/local/svn/ --config-file /usr/local/svn/conf/svnserve.conf --listen-port 3690 |
| chown apache:apache /usr/local/svn -R  service httpd restart |

**四、SVN仓库备份**

**1、windows下备份**

⑴备份：

使用svnadmin dump命令，备份版本库

|  |
| --- |
| svnadmin dump G:\SVN\Repositories\spring > J:\Java学习资料\SVN备份\7-21\spring.dump |

⑵恢复：

先创建版本库

|  |
| --- |
| svnadmin create G:\SVN\Repositories\spring |

恢复版本库。

|  |
| --- |
| svnadmin load G:\SVN\Repositories\spring < J:\Java学习资料\SVN备份\7-21\spring.dump |

⑶增量备份及恢复

增量备份：

-r 代表版本修订号(revision)， 12:20表示增量备份12至20版本修订号， --incremental表示增量备份。

注意：增量备份会产生一个全量dump和若干个增量dump文件。

|  |
| --- |
|  |

增量恢复：

先恢复全量dump，然后在逐个恢复增量dump。

完整备份及增量备份的比较

|  |
| --- |
|  |

**四、 Linux下安装rabbitmq**

**1、安装Erlang环境**

1)安装GCC GCC-C++ Openssl等模块,安装过就不需要安装了

yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel openssl-devel

|  |
| --- |
|  |

2)安装ncurses

yum -y install ncurses-devel

|  |
| --- |
|  |

3)安装erlang环境

wget http://erlang.org/download/otp\_src\_20.2.tar.gz

|  |
| --- |
|  |

tar xvfz otp\_src\_18.2.1.tar.gz

cd /usr/local/rabbitMQ/otp\_src\_20.2/

./configure

|  |
| --- |
|  |

make install

|  |
| --- |
|  |

**2、安装RabbitMQ**

1)下载rabbitmq-server-3.6.9.tar.xz

|  |
| --- |
| wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.15/rabbitmq-server-generic-unix-3.6.15.tar.xz |

2)对于下载xz包进行解压，首先先下载xz压缩工具：

yum install xz

|  |
| --- |
|  |

3)对rabbitmq包进行解压：

xz -d rabbitmq-server-generic-unix-3.6.15.tar.xz

tar -xvf rabbitmq-server-generic-unix-3.6.15.tar.xz

|  |
| --- |
|  |
|  |

4)随后移动至/usr/local/下 改名rabbitmq：

cp -r rabbitmq\_server-3.6.9 /usr/local/rabbitmq

|  |
| --- |
|  |

5)这种下载的方式解压后直接可以使用，无需再编译安装；

进入到rabbit文件内，其命令文件存在于sbin文件夹下，因此需要将sbin文件夹的路径添加到PATH中：修改/etc/profile

export PATH=/usr/local/rabbitmq/sbin:$PATH

执行source /etc/profile使得PATH路径更新，rabbitMQ安装成功。

|  |
| --- |
|  |

6)随后启用MQ管理方式：

rabbitmq-plugins enable rabbitmq\_management #启动后台管理

rabbitmq-server -detached #后台运行rabbitmq

|  |
| --- |
|  |

7)设置端口号，可供外部访问：

firewall-cmd --permanent --add-port=15672/tcp

firewall-cmd --reload

firewall-cmd --query-port=15672/tcp

|  |
| --- |
|  |

8)添加用户和权限

默认网页guest用户是不允许访问的，需要增加一个用户修改一下权限，代码如下：

添加用户: rabbitmqctl add\_user admin admin

添加权限: rabbitmqctl set\_permissions -p "/" admin ".\*" ".\*" ".\*"

修改用户角色:rabbitmqctl set\_user\_tags admin administrator

|  |
| --- |
|  |

然后就可以远程访问了，然后可直接配置用户权限等信息。

<http://192.168.0.150:15672/#/>

|  |
| --- |
|  |

**五、 Linux下安装mysql**

1、下载MySQL压缩包

wget <http://dev.MySQL.com/get/Downloads/MySQL-5.7/mysql-5.7.11-Linux-glibc2.5-x86_64.tar.gz>

|  |
| --- |
|  |

// 如果上边的命令不行的话 可以使用下边的命令

curl -O -L <http://dev.MySQL.com/get/Downloads/MySQL-5.7/mysql-5.7.11-Linux-glibc2.5-x86_64.tar.gz>

2、解压并复制

tar -zxvf mysql-5.7.11-Linux-glibc2.5-x86\_64.tar.gz

mv mysql-5.7.11-Linux-glibc2.5-x86\_64/\* /usr/local/mysql/

|  |
| --- |
|  |

tar -zxvf /usr/local/software/mysql-5.7.11-Linux-glibc2.5-x86\_64.tar.gz -C

/usr/local/mysql/

|  |
| --- |
|  |

3、创建data目录

mkdir /usr/local/mysql/data

4、创建mysql用户组及其用户

#添加用户组

groupadd mysql

#添加用户mysql 到用户组mysql

useradd -r -g mysql mysql

|  |
| --- |
|  |

5、初始化数据

[root@localhost mysql] ./bin/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data/

2016-01-20 02:47:35 [WARNING] mysql\_install\_db is deprecated. Please consider switching to mysqld --initialize

2016-01-20 02:47:45 [WARNING] The bootstrap log isn‘t empty:

2016-01-20 02:47:45 [WARNING] 2016-01-19T18:47:36.732678Z 0 [Warning] --bootstrap is deprecated. Please consider using --initialize instead

2016-01-19T18:47:36.750527Z 0 [Warning] Changed limits: max\_open\_files: 1024 (requested 5000)

2016-01-19T18:47:36.750560Z 0 [Warning] Changed limits: table\_open\_cache: 431 (requested 2000)

|  |
| --- |
|  |

6、复制配置文件到 /etc/my.cnf

cp -a ./support-files/my-default.cnf /etc/my.cnf (选择y)

|  |
| --- |
|  |

7、MySQL的服务脚本放到系统服务中

cp -a ./support-files/mysql.server /etc/init.d/mysqld

|  |
| --- |
|  |

修改my.cnf文件 vim /etc/my.cnf

|  |
| --- |
| basedir = /usr/local/mysql  datadir = /usr/local/mysql/data  port = 3306  server\_id = mysql134  socket = /tmp/mysql.sock  # Remove leading # to set options mainly useful for reporting servers.  # The server defaults are faster for transactions and fast SELECTs.  # Adjust sizes as needed, experiment to find the optimal values.  join\_buffer\_size = 128M  sort\_buffer\_size = 2M  read\_rnd\_buffer\_size = 2M |

8、创建ln软件链接并开放端口

ln -s /usr/local/mysql/bin/mysql /usr/bin/

|  |
| --- |
| ln -s /usr/local/mysql/bin/mysql /usr/bin/ |
| 开放3306端口:  firewall-cmd --add-port=3306/tcp --permanent  firewall-cmd --reload  firewall-cmd --query-port=3306/tcp |

9、启动服务

service mysqld start

|  |
| --- |
| 启动前：mysql目录改为mysql读写权限  chown -R mysql:mysql /usr/local/mysql |

10、初始化密码

mysql5.7会生成一个初始化密码，在root中.mysql\_secret文件中。

|  |
| --- |
| cat /root/.mysql\_secret |

12、登录并修改密码

|  |
| --- |
| mysql -u root -h 127.0.0.1 -p |

登录后：

alter user root@localhost identified by '123456';

flush privileges;

|  |
| --- |
|  |

13、退出重新登录，完成

|  |
| --- |
|  |

14、window上访问Linux上的Mysql

授权MySQL，允许远程用户登录访问MySQL：

方法一：

本地登入mysql，更改 "mysql" 数据库里的 "user" 表里的 "host" 项，将"localhost"改为"%" 。

#mysql -u root -proot

　　mysql>use mysql;

　　mysql>update user set host = '%' where user = 'root';

　　mysql>select host, user from user;

方法二、直接授权(推荐)

从任何主机上使用root用户，密码：youpassword（你的root密码）连接到mysql服务器：

　　 mysql -u root -proot

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

操作完后切记执行以下命令刷新权限

　　 FLUSH PRIVILEGES

|  |
| --- |
|  |

15、设置Linux启动时启动mysql

方式一：

vim /etc/rc.d/rc.local

chmod +x /etc/rc.d/rc.local

|  |
| --- |
| 在rc.local中加入service mysqld start |

方式二：（引用 <https://www.cnblogs.com/sunny3096/p/7954146.html> ）

①将服务文件拷贝到init.d下，并重命名为mysql：

cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld

②赋予可执行权限：

chmod +x /etc/init.d/mysqld

③添加服务

chkconfig --add mysqld

④显示服务列表

chkconfig --list

|  |
| --- |
|  |

如果看到mysql的服务，并且3,4,5都是on的话则成功，如果是off，则键入

chkconfig --level 345 mysql on

⑤重启电脑：reboot

⑥验证：netstat -na | grep 3306

|  |
| --- |
| chkconfig mysqld on  chkconfig mysqld --list |

**六、 Linux下安装Postgresql(yum安装)**

1、Yum安装

yum install https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86\_64/pgdg-centos10-10-2.noarch.rpm

|  |
| --- |
|  |

2、查看postgresql源

yum list | grep postgresql

|  |
| --- |
|  |

我们需要安装的是这两个。postgresql10-contrib postgresql10-server

|  |
| --- |
|  |

yum install postgresql10-contrib postgresql10-server -y

|  |
| --- |
|  |

3、初始化数据库

Postgresql安装目录是/usr/pgsql-10,而Postgresql的数据目录是/var/lib/pgsql/版本号/data目录。

在这里，如果在装系统开始分配var空间足够大则可以继续，如果分配var空间不够，我们需要更改数据目录，在这里，我们假设var空间足够大。直接开始初始化。

/usr/pgsql-10/bin/postgresql-10-setup initdb

|  |
| --- |
|  |

显示这个代表初始化成功

4、开放端口

firewall-cmd --add-port=5432/tcp --permanent

firewall-cmd --reload

firewall-cmd --query-port=5432/tcp

|  |
| --- |
|  |

5、启动数据库并设置开机启动

sudo systemctl start postgresql-10

|  |
| --- |
|  |

sudo systemctl enable postgresql-10.service

|  |
| --- |
|  |

6、登录postgresql并设置密码

postgresql在安装时默认添加用户postgres

输入： su - postgres

psql

|  |
| --- |
|  |

进入数据库，样子长这样

|  |
| --- |
|  |

我们来设置密码：

postgres=# ALTER USER postgres WITH PASSWORD '123456';

ALTER USER postgres WITH PASSWORD '密码';

|  |
| --- |
|  |

退出：\q ctrl+d

|  |
| --- |
|  |

备注其他：列出所有库\l 列出所有用户\du 列出库下所有表\d

|  |
| --- |
|  |

7、默认情况下postgresql是不用密码不支持远程登录的。我们需要修改配置文件

vim /var/lib/pgsql/10/data/pg\_hba.conf

原本长成这样

|  |
| --- |
|  |

修改客户端认证配置文件pg\_hba.conf，在IPV4下面添加一条记录”host all all 10.1.5.0/24 md5”，其中10.1.5.0代表网段’10.1.5.\*’, 24是子网掩码，代表10.1.5.0-10.1.5.255, md5代表使用md5加密

我们需要改成：

|  |
| --- |
|  |

保存退出，我们改远程访问：vim /var/lib/pgsql/10/data/postgresql.conf

往下拉我们会看到：

|  |
| --- |
|  |

需要改成：

|  |
| --- |
|  |

重启postgresql： systemctl restart postgresql-10

|  |
| --- |
|  |

8、登录postgresql

postgresql在安装时默认添加用户postgres

输入： su - postgres

psql

9、远程连接数据库

我们用pgadmin进行验证连接:

|  |
| --- |
|  |

10、我们测试创建数据库及创建用户

因为postgres属于superuser，我们需要创建部分低权限用户。

创建数据库：CREATE DATABASE testdb;

创建用户：CREATE USER testuser CREATEDB LOGIN PASSWORD 'testpassword';

将testdb所有权限赋给用户testuser：GRANT ALL ON DATABASE testdb TO testuser;

|  |
| --- |
|  |

11、删除数据库及测试用户

删除数据库：drop database testdb;

删除用户：drop role testuser;

**六、 Linux下安装Postgresql(tar包安装)**

1、tar包安装

在postgresql的官方即可找到源码文件目录，地址如下：https://www.postgresql.org/ftp/source/，在下载列表中根据需求选择版本，如下图：

|  |
| --- |
|  |
|  |

如上图，可以看到提供了两种压缩格式，此处我们选择postgresql-9.5.5.tar.gz，下载完成后上传至CentOS服务器的指定目录即可。

2、配置编译安装

首先进入pg压缩包目录通过tar -zxvf ./postgresql-9.5.5.tar.gz进行解压，然后就可以开始编译安装了，进入解压目录，通过./configure --help可以看到编译相关的帮助信息，如下图：

|  |
| --- |
|  |

如上图，--prefix=dir可以指定安装目录，还有比如--with-python则可以使用python语法的PL/Python过程语言自定义函数，根据需求我们暂且用不到，所以编译时仅指定一个安装目录即可：

./configure --prefix=/usr/local/postgresql

运行编译后发现会提示没有c编译器，是因为新系统的原因，所以安装一个gcc编译器即可：

yum -y install gcc

yum -y install readline-devel

yum -y install zlib-devel

安装完成后再次编译postgresql，并没有再报error，可以看到提示创建了config.status配置文件：

|  |
| --- |
|  |

配置完成了，接下来就可以编译安装了，依次执行：

make

make install

看到如下提示即可说明编译安装成功：

|  |
| --- |
|  |

3、用户权限与环境变量

编译安装成功后，接下来要做的就是创建一个普通用户，因为默认超级用户（root）不能启动postgresql，所以需要创建一个普通用户来启动数据库，执行以下命令创建用户：

useradd postgres

接下来需要设置权限，将postgres的数据目录全部赋权给postgres用户（此处我将postgres的数据目录指定在在/usr/local/postgresql/data目录下）：

chown -R postgres:postgres /usr/local/postgresql/

最后为了方便起见设置一下相关的环境变量，此处仅仅设置postgres用户的环境变量，所以首先通过su - postgres切换到postgres用户，打开.bash\_profile文件并追加以下内容：

su - postgres

vim /home/postgres/.bash\_profile

|  |
| --- |
| # User specific environment and startup programs  PGHOME=/usr/local/postgresql  export PGHOME  PGDATA=/usr/local/postgresql/data  export PGDATA  PATH=$PATH:$HOME/.local/bin:$HOME/bin:$PGHOME/bin  export PATH |
|  |

修改完成后可以通过source /home/postgres/.bash\_profile使其立即生效，接下来检验一下环境变量是否设置正确，切换任意目录输入which psql以及psql -V即可分别查看psql客户端的路径以及postgresql的数据库版本，如下图：

|  |
| --- |
|  |

一切准备就绪之后接下来就可以初始化数据库了。

4、初始化数据库

由于配置了环境变量，所以此处我们直接执行initdb即可完成db初始化，但在这之前我们可以通过initdb --help看一下初始化相关的帮助信息：

|  |
| --- |
|  |

如上图，可以看到在使用initdb进行初始化的同时我们可以指定参数来同时进行一些初始化工作，例如指定pgdata（postgresql数据目录）、指定encoding（编码）、指定数据库超级用户的用户名和密码等等，在最后面我标记出的这段话指出了如果data目录没有指定，则会默认使用环境变量中的PGDATA，由于之前我们刚刚设置了PGDATA环境变量，所以此处我们也就无需再额外指定，最后执行初始化命令即可：

initdb

看到如下信息就说明初始化成功了：

|  |
| --- |
|  |

同时在postgresql的目录可以看到生成的数据目录data以及该目录的相关数据和配置文件：

|  |
| --- |
|  |

如上图，base目录是表空间目录，global目录是相关全局变量的目录，pg\_hba.conf和postgresql.conf在之前的博客也都提及了，一个是访问控制配置（127.0.0.1改为信任的客户端ip网段使其可以远程访问），一个是postgresql主配置文件（listen\_address=localhost改为星号使其监听整个网络），方便起见我这里将pg\_hba.conf的ip地址修改为0.0.0.0/0，而加密方式改为md5，就表示需要密码访问，算是提供一个最低级的安全防护：

|  |
| --- |
|  |

最后别忘记开放pg的5432端口，否则即使做了上面两处修改客户端依然无法连接postgresql，所以将5432端口加入到zone即可，依次运行以下命令（注意此处需要切回root用户，否则没有权限）：

|  |
| --- |
| firewall-cmd --zone=public --add-port=5432/tcp --permanent  firewall-cmd --reload |

至此配置相关的内容就全部完成了，最后就是启动并连接数据库了。

5、启动和连接

在初始化数据库结束时我们已经看到了启动命令，如下图：

|  |
| --- |
|  |

由于我们设置了环境变量，所以已经指定了数据目录PGDATA，-l表示日志文件目录，通常需要指定，所以我们在/usr/local/postgresql根目录下用postgresql用户再创建一个log目录用来存放日志文件（注意别忘记赋予可写的权限），

启动数据库:

/usr/local/postgresql/bin/pg\_ctl start -D /usr/local/postgresql/data/ -l /usr/local/postgresql/log/log\_server.log

|  |
| --- |
|  |

最后运行pg\_ctl start -l /usr/local/postgresql/log/pg\_server.log即可启动数据库，看到如下提示就说明启动成功了：

|  |
| --- |
|  |

启动成功后我们就可以通过postgresql自带的客户端工具psql来进行连接，直接输入psql看到版本信息则说明连接成功：

|  |
| --- |
|  |

而在日志文件目录中可以看到数据库日志文件以及刚才的启动日志：

|  |
| --- |
|  |

接下来要做的第一件事就是设置postgres用户的密码（默认为空），用psql连接成功后直接输入\password即会提示输入两次密码，如下图：

|  |
| --- |
|  |

此处我们暂且将密码修改为111111，通过\l命令即可查看数据库列表，关于psql工具的使用在此处就不再做过多赘述。最后也是最重要的一点就是验证非本地客户端工具的连接了，毕竟我们是要做数据库服务器的，这里我选择的工具是Navicat Premium，在主机（Windows7）打开Navicat与虚拟机中的postgresql服务器进行连接测试：

停止postgresql数据库

/usr/local/postgresql/bin/pg\_ctl -D /usr/local/postgresql/data/ stop

|  |
| --- |
|  |

**七、 Linux下安装PgAdmin**

1、下载安装包

查看Linux的Python版本:

|  |
| --- |
|  |

首先要保证linux下要有pip/pip3

打开pgAdmin官网，找到Download->pgAdmin4->Python Wheel ，

|  |
| --- |
|  |

只需要下载.whl的文件即可。

|  |
| --- |
|  |

等下好了.whl文件，为了方找，把文件拖到Linux的/usr/local/software文件夹下。

2、CentOS7下安装python-pip

首先检查linux有没有安装python-pip包,直接执行

yum install python-pip

|  |
| --- |
|  |

没有python-pip包就执行命令:

yum -y install epel-release

|  |
| --- |
|  |

执行成功之后，再次执行yum install python-pip

|  |
| --- |
|  |

对安装好的pip进行升级 pip install --upgrade pip

|  |
| --- |
|  |

3、error: command 'gcc' failed with exit status 1

由于没有正确安装Python开发环境导致

sudo yum install python-devel

|  |
| --- |
|  |

pip install gevent

|  |
| --- |
|  |

把环境更新下 sudo yum install groupinstall 'development tools'

|  |
| --- |
|  |

4、安装

打开终端(如果是python3，将pip换成pip3)，输入：

pip install pgadmin4-3.0-py2.py3-none-any.whl

|  |
| --- |
|  |

这一步要花费很长时间。

|  |
| --- |
|  |

5、进入安装位置

cd /lib/python2.7/site-packages/pgadmin4/

|  |
| --- |
|  |

6、运行pgadmin4.py

如果是python3，把python换成python3：

python pgadmin4.py

第一次会提示让你输入用作登陆的邮箱和密码：

|  |
| --- |
|  |

7、开放端口5050

firewall-cmd --add-port=5050/tcp --permanent

firewall-cmd --reload

firewall-cmd --query-port=5050/tcp

|  |
| --- |
|  |

**八、 Linux下搭建Git服务器**

环境：

服务器 CentOS7.2 + git（version 1.8.3.1）

客户端 Windows10 + git（version 2.8.4.windows.1）

1、下载安装包

Linux 做为服务器端系统，Windows 作为客户端系统，分别安装 Git。

**服务器端：**

yum install -y git

|  |
| --- |
|  |

安装完后，查看 Git 版本

|  |
| --- |
|  |

**客户端：**

第一步：下载 Git for Windows，地址：<https://git-for-windows.github.io/>

第二步：双击下载好的git安装包，弹出提示框，如下图：

|  |
| --- |
|  |

第三步： 直接点击“next”进入下一步，选择安装路径，如下图：

|  |
| --- |
|  |

第四步：选择好安装路径后，点击“next”进入下一步，弹出安装配置窗口，包括git命令行、git图形窗口等，如下图所示：

|  |
| --- |
|  |

第五步：按照上述默认配置，直接点击“next”进入下一步，弹出“选择菜单开始文件”的窗口，如下图所示：

|  |
| --- |
|  |

第六步：按照默认路径即可，直接点击“next”，进入下一步，进入“调整路径环境”窗口，如下图所示：

|  |
| --- |
| 注：该窗口中，各项选项的意思为：  第一项：直接安装，不会配置git命令的环境变量。  第二项：会自动配置好git命令的环境变量。  第三项：git命令和unix工具命令都会添加到环境变量。 |

第七步：由于第一项不会配置环境变量，第三项会添加可选unix工具，基本没用，所以选第二项，然后点击“next”进入下一步，如下图所示：

|  |
| --- |
|  |

配置

用户名和邮箱地址的作用：

用户名和邮箱地址是本地Git客户端的一个变量，不随Git库而改变。

每次Commit都会用用户名和邮箱地址记录。

|  |
| --- |
| git config --global user.name "chengxp"  git config --global user.email "cheng3\_shan3@163.com" |

安装完之后，可以使用 Git Bash 作为命令行客户端。

安装完之后，查看 Git 版本

|  |
| --- |
|  |

2、服务器端创建 git 用户，用来管理 Git 服务，并为 git 用户设置密码

|  |
| --- |
| [root@localhost home]# id git  [root@localhost home]# useradd git  [root@localhost home]# passwd git |

3、服务器端创建 Git 仓库

设置 /home/data/git/gittest.git 为 Git 仓库

然后把 Git 仓库的 owner 修改为 chengxp

|  |
| --- |
| mkdir -p /home/data/git/gittest.git  git init --bare /home/data/git/gittest.git/  cd /home/data/git/  chown -R git:git gittest.git/ |

4、客户端 clone 远程仓库

进入 Git Bash 命令行客户端，创建项目地址（设置在 E:\Eclipse workspace\Idea 2017 workspace\git）并进入:

|  |  |
| --- | --- |
| git clone git@192.168.0.150:/home/data/git/gittest.git    当第一次连接到目标 Git 服务器时会得到一个提示：   |  | | --- | | The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.  RSA key fingerprint is SHA256:Ve6WV/SCA059EqoUOzbFoZdfmMh3B259nigfmvdadqQ.  Are you sure you want to continue connecting (yes/no)? |   选择 yes：  Warning: Permanently added '192.168.56.101' (RSA) to the list of known hosts. |

此时 C:\Users\用户名\.ssh 下会多出一个文件 known\_hosts，以后在这台电脑上再次连接目标 Git 服务器时不会再提示上面的语句。

5、客户端创建 SSH 公钥和私钥

ssh-keygen -t rsa -C "cheng3\_shan3@163.com"

|  |
| --- |
|  |

此时 C:\Users\用户名\.ssh 下会多出两个文件 id\_rsa 和 id\_rsa.pub

id\_rsa 是私钥

id\_rsa.pub 是公钥

|  |
| --- |
|  |

6、服务器端 Git 打开 RSA 认证

进入 /etc/ssh 目录，编辑 sshd\_config，打开以下三个配置的注释：

|  |
| --- |
| cd /etc/ssh/ |

|  |
| --- |
| vim sshd\_config  RSAAuthentication yes  PubkeyAuthentication yes  AuthorizedKeysFile .ssh/authorized\_keys |

保存并重启 sshd 服务：

|  |
| --- |
| service sshd restart |

由 AuthorizedKeysFile 得知公钥的存放路径是 .ssh/authorized\_keys，实际上是 $Home/.ssh/authorized\_keys，由于管理 Git 服务的用户是 git，所以实际存放公钥的路径是 /home/git/.ssh/authorized\_keys 。

在 /home/git/ 下创建目录 .ssh

|  |
| --- |
| cd /home/git  pwd  mkdir .ssh  ls -a |

然后把 .ssh 文件夹的 owner 修改为 git

|  |
| --- |
| chown -R git:git .ssh |

7、将客户端公钥导入服务器端 /home/git/.ssh/authorized\_keys 文件

回到 Git Bash 下，导入文件：

|  |
| --- |
| ssh git@192.168.0.150 'cat >> .ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub |

回到服务器端，查看 .ssh 下是否存在 authorized\_keys 文件：

|  |
| --- |
| cd .ssh  ll |

可以查看一下是否是客户端生成的公钥。

重要:

修改 .ssh 目录的权限为 700

修改 .ssh/authorized\_keys 文件的权限为 600

|  |
| --- |
| chmod 700 .ssh  chmod 600 authorized\_keys |

8、客户端再次 clone 远程仓库

git clone git@192.168.0.150:/home/data/git/gittest.git

|  |
| --- |
|  |

查看客户端项目目录：

|  |
| --- |
|  |

项目已经 clone 了。

**七、 Linux上安装samba服务器**

我们知道linux与linux之间进行文件共享的是nfs服务器。那么linux与其他类型系统进行文件共享的服务器是什么呢。就是samba。当然你也可以用vsftp。这两者各有优缺点，samba处理内部事务比较好，配置也比较简单。而vsftp的安全性更高，权限设置的全面、并且传输效率上也要比samba高很多。samba是单线程的，并且效率不太高，容易造成阻塞的现象。

首先查看，当前系统有没有安装samba：

rpm -qa |grep samba

即使安装了也可以卸掉，自己重新安装：

卸载命令：yum remove samba -y

(同时把/etc/samba文件夹下的文件也删除了。注意：不要把smb.conf删了，这个是samba的核心配置文件。)

安装命令：yum install samba

安装完之后：rpm -qa| grep samba，看一下是不是装全了。

|  |
| --- |
|  |

看一下/etc/samba文件夹下，应该确保有smb.conf,跟sambauser这两个文件。

|  |
| --- |
|  |

剩下的就是配置了，代码如下。一步步配就能成功。

|  |  |
| --- | --- |
| service smb start #首先启动smb服务  cd / # 进入到/目录  #这一步是添加用户的操作（查看当前所有用户：cat /etc/passwd）  useradd tom  useradd mary  #这一步是为用户设置密码  passwd tom  passwd mary  smbpasswd -a tom  smbpasswd -a mary  #创建要共享文件夹（这个文件夹是可读可写，名字自己起）  mkdir readwrite  #创建要共享文件夹（这个文件夹是可读可写，名字自己起）  mkdir readonly  #可以在这个文件夹下，先创建一个文件  vim readwrite/write  vim readonly/only  #设置文件的访问权限（如果想在windows映射成硬盘的话，最好使用777）  chmod 555 readonly  chmod 700 readwrite  #设置文件的拥有者  chown -R tom readwrite  #在配置文件中设置samba的用户  vim /etc/samba/smbusers   |  | | --- | |  |     #前面是你刚才设置的用户，后面是在windows中，想使用的用户名  添加：tom=wintom  mary=winmary  vim /etc/samba/smb.conf  #设置用户列表  在golbal下添加username map=/etc/samba/smbusers |

|  |
| --- |
| 在share下添加下列代码：  [share]  #局部配置，[]里面是段名，也就是我们的共享名  #该共享的说明  comment = This is my shared folder.  #共享路径  path = /home/samba  #共享文件夹是否可写  writeable = yes  #是否可浏览  browseable = yes  #跟public一样，是否允许guest用户  guest ok = no  #创建文件权限定制  create mode =0664  #创建文件夹权限  directory mode=0775  #允许访问该共享的用户  valid users=chengxp  #禁止访问的用户，root，@group,@表示组  #invalid usrs=  #该共享的管理者  admin users=root  #有写权限的用户  write list=chengxp  # #这三个是避免乱码的，dispaly是显示目录时的编码  display charset=UTF-8  unix charset=UTF-8  dos charset=UTF-8 |
|  |

service smb restart 保存后，重启服务

service iptables stop 关闭防火墙，或者开启端口

setenforce 0 设置selinux为disabled

备注：

**1、CentOS7设置网络自动启动**

切换到网卡配置文件的目录，找到网卡的配置文件。

cd /etc/sysconfig/network-scripts/

|  |
| --- |
|  |

vim ifcfg-eno16777736

|  |
| --- |
| 把ONBOOT对应的选项改为: ONBOOT=yes |

**2、CentOS7无桌面版设置网卡固定IP**

cd /etc/sysconfig/network-scripts/

|  |
| --- |
|  |

vim ifcfg-eth0 修改局域网连接地址

|  |
| --- |
| DEVICE="eth0"  ONBOOT="yes"  TYPE="Ethernet"  PERSISTENT\_DHCLIENT="yes"  BOOTPROTO=none  DNS1=192.168.0.1  IPADDR=192.168.0.136  GATEWAY=192.168.0.1  NETMASK=255.255.255.0  DEFROUTE=yes  IPV4\_FAILURE\_FATAL=no  IPV6INIT=yes  IPV6\_AUTOCONF=yes  IPV6\_DEFROUTE=yes  IPV6\_FAILURE\_FATAL=no  NAME=eth0  UUID=a4d89b94-54f2-4060-b579-652b7b0e69e2  PREFIX=24  IPV6\_PEERDNS=yes  IPV6\_PEERROUTES=yes |
|  |

vim ifcfg-eth1 修改仅主机网络地址

|  |
| --- |
| DEVICE=eth1  HWADDR=08:00:27:12:ad:ba  TYPE=Ethernet  BOOTPROTO=none  IPADDR=192.168.56.101  PREFIX=24  GATEWAY=192.168.56.1  DNS1=192.168.56.1  DEFROUTE=yes  IPV4\_FAILURE\_FATAL=no  IPV6INIT=yes  IPV6\_AUTOCONF=yes  IPV6\_DEFROUTE=yes  IPV6\_PEERDNS=yes  IPV6\_PEERROUTES=yes  IPV6\_FAILURE\_FATAL=no  NAME=eth1  UUID=fbe91746-c465-4bcd-ad35-4395849e2efd  ONBOOT=yes |

service network restart

|  |
| --- |
|  |

**3、最小安装CentOS7**

⑴安装netstat、ifconfig命令

centos7 最小化安装 无 ifconfig,netstat 的安装

centos7 最小化安装之后，默认是没有 ifconfig,netstat命令的；

我们可以直接使用 yum -y install net-tools 即可；

|  |
| --- |
| yum info net-tools |

|  |
| --- |
| yum -y install net-tools |

|  |
| --- |
| ifconfig  netstat -ntlp |

⑵安装vim命令

yum install vim

|  |
| --- |
|  |

⑶安装wget命令

yum install wget

|  |
| --- |
|  |