

## centos控制台

### centos目录结构

/: 根目录  
bin  
root  
boot  
dev  
etc  
home  
lib  
sbin  
usr(一般软件安装的位置)  
var  
mnt  
temp  
注

### 清屏:

注销用户:

切换用户:

重启:

关机:

查看linux内核版本: 退出: q

CPU的使用情况和正在运行的进程情况的命令: top(类似任务管理器), 退出: q

显示月历或年历的命令: cal

显示或设置当前日期和时间的命令: date

远程连接: ifconfig: 用于显示或设置网络设备

### 基本命令

查看当前所在文件夹  
切换目录命令  
查看文件列表  
创建目录和移除目录 mkdir, rmdir  
touch创建一个空文件  
浏览文件  
clear清屏、也可以使用ctrl+I

#### 文件操作

rm: 删除文件  
cp、mv  
tar: 打包或解压  
find: 用于查找符合条件的文件  
grep: 查找文件里符合条件的字符串

### vi(无高亮显示)和vim(有高亮显示)编辑器

- 1、vim编辑器
- 2、重定向输出>和>>
- 3、系统管理命令
- 4、管道 |

### Linux权限命令

- 1、文件权限
- 2、Linux三种文件类型
- 3、文件类型标识
- 4、文件权限管理

### Linux上常用网络操作

- 1、主机配置
- 2、ip地址配置
- 3、域名映射
- 4、网络服务管理

## Linux上安装软件

### 上传方式

- 1、在windows上安装工具
- 2、在linux上安装
- 3、使用linux自带的sftp

### 在linux中安装jdk

- 1、上传jdk到linux服务器
- 2、在linux上安装jdk
- 3、配置JDK的环境变量

### 在linux上安装mysql

- 1、将mysql的安装文件上传到linux服务器
- 2、安装mysql服务器端(server)——>rpm -ivh mysql服务器端的名称
- 3、安装mysql客服端(client)
- 4、登录
- 5、windows端，mysql可视化工具远程登录

### 在linux上安装tomcat

- 1.Tomcat上传到linux 上
- 2.将上传的tomcat解压
- 3.在tomcat/bin目录下执行startup.sh (注意防火墙)
- 4.查看目标tomcat/logs/catalina.out

### 在linux上安装redis

- 1、安装gcc-c++(此步推荐在虚拟机里面操作)
- 2、安装redis(linux下输入)
  - 1、下载redis, 在/usr/local文件夹
  - 2、解压
  - 3、编译安装
  - 3、配置redis

## Nginx

- 1、在linux上安装nginx
  - 1.1、nginx的介绍
  - 1.2、nginx应用场景
  - 1.3、Nginx在linux下的安装
  - 1.4、准备环境
  - 1.5、Ngnix下载
  - 1.6、Nginx启动与访问
- 2、Nginx静态网页部署
  - 2.1、端口绑定
  - 2.2、域名绑定
- 3、反向代理
- 4、负载均衡

# centos控制台

虚拟文本控制台——>常见的就是在桌面打开终端；【用户名@主机名 所在目录】，普通用户是\$，root用户是#

又被称为虚拟终端(tty), CentOS操作系统有5个虚拟终端(tty1~ tty5)。

通过快捷键Ctrl+Alt+F2~ Ctrl+Alt+F6， 可以进行多个控制台之间的切换。

Ctrl+Alt+F1切换回到图形界面。

注：

- 1、在输入用户名后，输入密码时，密码是隐藏的，不可见
- 2、输入tty，可查看登录的虚拟终端层级

参数，可通过命令+空格+--help查看各个参数的信息；  
或man+空格+命令

# centos目录结构

---

## /：根目录

---

### bin

---

(binary)存放二进制的可运行文件

### root

---

超级用户目录

### boot

---

存放用于系统引导时使用的各种文件

### dev

---

(devices)用于存放设备文件

### etc

---

(etcetera)存放系统配置文件

### home

---

存放用户文件的根目录

### lib

---

(library)存放跟文件系统中的程序运行所需要的共享库及内核模块

### sbin

---

(super binary)存放二进制的可运行文件，只有root才能访问

## usr(一般软件安装的位置)

---

(unix shared resources)用于存放共享的系统资源

### var

---

(variable)用于存放运行时需要改变数据的文件

### mnt

---

(**mount**)系统管理员安装临时文件系统的安装点

## temp

---

(**temp**prary)用于存放各种临时文件

## 注

---

root管理员的home目录是root

其他用户的home目录是home目录

## 清屏：

---

ctrl+l或clear

## 注销用户：

---

logout或exit

## 切换用户：

---

su 用户名： su student

## 重启：

---

- 1、reboot 普通重启
- 2、shutdown -r now 立刻重启(root用户使用)
- 3、shutdown -r 10 过10分钟自动重启(root用户使用)
- 4、shutdown -r 20:35 在时间为20:35时候重启(root用户使用)

## 关机：

---

- 1、halt 立刻关机
- 2、poweroff 立刻关机
- 3、shutdown -h now 立刻关机(root用户使用)
- 4、shutdown -h 10 10分钟后自动关机

通过shutdown命令设置关机的话，可以用shutdown -c命令取消重启

## 查看linux内核版本：退出： q

---

- 1、uname -r
- 2、uname -a (较详细)
- 3、man uname (更详细)

查看已运行的时间：uptime (当前时间 已运行时长 用户个数 平均负载)； 1个cpu最大为4，正常情况不超过4

查看文件相信他的磁盘空间和剩余空间: df

当前系统内存使用情况: free

total(总大小) used(已用大小) free(剩余大小) shared(共享内存) buff/cache(使用的物理内存大小) availab(剩余物理内存大小)

Mem: 内存(物理内存)

Swap: 交换分区(虚拟内存)

查询spu的详细信息: cat /proc/cpuinfo(一起输入)

## CPU的使用情况和正在运行的进程情况的命令: top(类似任务管理器), 退出: q

查看登录日志信息的命令: last

查看登录用户信息: w 【用户名】 ----->pst/0: 表示在字符界面登录(输入tty有提示)

USER: 登录的用户名

TTY: 登录终端

FROM: 从哪个in地址登录

LOCIN@: 登录时间

IDLE: 用户闲置时间

JCPU:指的是和该终端连接的所有进程所占用的时间,这个时间并不包括过去后台的作业时间,但却包

括当前在运行的后台的作业所占用的时间

PCPU:指当前进程所占用的时间

WHAT:当前正在运行的命令

## 显示月历或年历的命令: cal

## 显示或设置当前日期和时间的命令: date

## 远程连接: ifconfig: 用于显示或设置网络设备

1、ping 连接的ip

- 2、使用ssh工具连接linux系统(可自选,如: putty、xshell、secureCRT、sublime text通过插件也可以)
- 3、Host Name (or IP address): 虚拟机的ip地址(addr)
- 4、采用ssh方式连接
- 5、登录(用户名和密码): root    root(此处输入的密码也是不可见的)

## 基本命令

### 查看当前所在文件夹

pwd

## 切换目录命令

进入根目录: `cd /`

进入用户目录: `cd`(也可以写`cd ~`), 普通用户是`home`内, `root`用户是`root`

返回上一级: `cd ..`

返回上一层的工作目录: `cd -`

当前目录: `./`

根目录: `/`

`cd` 后可以输入前几个字母, 使用`table`键补全

## 查看文件列表

`ls`: 列出所在目录下的文件夹/文件的名字 (/目录), 带有/: 代表从根目录起, 不带/: 代表当前目录起

参数: `-a`: 显示所有的文件夹/文件 (隐藏的)

- `-l`: 显示文件的详细信息(权限, 用户, 时间, 名称)(可简写为`ll`)

## 创建目录和移除目录 `mkdir, rmdir`

`mkdir` (make directory)命令可用来创建子目录, `mkdir app` 在当前目录下创建`app`目录`mkdir -p app2/test` 级联插件`app2`以及`test`

格式: `mkdir aaa`(在当前目录下, 创建`aaa`目录)

- `mkdir aaa/bbb`(在当前目录下, 创建`aaa`目录, 再次在`aaa`目录下创建`bbb`目录), 前提是有`aaa`目录才行

参数: `-p`: 表示(`aaa`)不存在目录时, 会创建; 存在也不会报错

`mkdir -p aaa/bbb`

`rmdir`(remove directory)命令可用来删除“空”的子目录

格式: `rmdir aaa`删除`aaa`目录

## touch创建一个空文件

`touch 文件名称.txt`

# 浏览文件

cat、more、less

**cat:** 用于显示文件的内容

格式: **cat** 参数 文件名---->**cat yum.conf**

**more:** 一般用于显示的内容会超过一个画面长度的情况, 按空格键显示下一个画面; 回车显示下一行内容; **q** 键退出或**ctrl+c**

格式: ---->**more yum.conf**

**less:** 用法和**more**用法类似, 而**less**也可以通过**pgUp**, **pgDn**键控制

格式: **less yum.conf**---->使用**pgUp**和**pgDn**进行上下翻页

**tail:** 命令是实际使用过程中使用非常多的一个命令, 其功能是: **用于显示文件后几行的内容**

格式:

查看后十行数据----->**tail -10 /etc/passwd**

动态查看日志----->**tail -f catalina.log**

结束查看----->**ctrl+c**

结束: **ctrl+c**

# clear清屏、也可以使用**ctrl+l**

## 文件操作

### **rm:** 删除文件

用法: **rm** 选项 文件...

**rm a.txt**删除a.txt文件

删除需要用户确认: **y/n**

**rm**删除不再询问

**rm -f a.txt** 不询问, 直接删除

**rm -r a** 递归删除, 不询问递归删除(慎用)

**rm -rf a** 不询问递归删除

**rm -rf \***删除所有文件

**rm -rf /\* 自杀**

## cp、 mv

**cp (copy)**命令可以将文件从一处复制到另一处。一般在使用**cp**命令时将一个文件复制成另个文件或复制到某目录时，需要指定源文件名与目标文件名或目录。

将a.txt复制为b.txt: **cp a.txt b.txt** (此处可指定文件夹)

将a.txt文件复制到上一层目录中: **cp a.txt ..**/

将a.txt复制到指定文件夹: **cp a.txt aaa** (复制到当前文件夹)

## mv移动或重命名

将a.txt复制到上一层目录中: **mv a.txt ..**/ (此处可指定文件夹)

将a.txt文件重命名为b.txt: **mv a.txt b.txt**

## tar: 打包或解压

tar命令位于/bin目录下，它能够将用户所指定的文件或目录**打包成一个文件，但不做压缩**。一般Linux上常用的压缩方式是选用tar

将许多文件打包成一个文件，再以gzip压缩命令压缩成xxx.tar.gz (或称为xxx.tgz)的文件。

### 常用参数:

-C: 创建一个新tar文件

-v: 显示运行过程的信息

-f: 指定文件名

-z: 调用gzip压缩命令进行压缩

-t: 查看压缩文件的内容

-x: 解开tar文件

打包: **tar -cvf xxx.tar .**(此处可指定文件夹， ./表示所在文件夹)

打包并压缩: **tar -zcvf xxx.tar.gz .**

### 解压:

**tar -xvf xxx.tar .**(解压到当前目录)

**tar -zxvf xxx.tar.gz -C /usr /aaa**

## find：用于查找符合条件的文件

格式：

查找指定文件： `find / -name 文件名称`

模糊查找文件： `find / -name 文件前几个字母*.文件格式`

`find / -name ins -ls`

查找用户是centos的文件： `find / -usr centos -ls`

查找用户centos的目录： `find / -usr centos -type d -ls`

查找权限是777的文件： `find / -perm -777 -type d -ls`

## grep：查找文件里符合条件的字符串

格式： `grep 关键字 路径/文件名称.文件格式`

在文件中查找lang： `grep lang anconda-ks.cfg`

高亮显示： `grep lang anconda-ks.cfg --color`

在关键词后再显示一行： `grep 关键字 路径 --color -A1`

在关键词前再显示一行： `grep 关键字 路径 --color -B1`

在关键词前和后再显示一行： `grep 关键字 路径 --color -A1 -B1`

## vi(无高亮显示)和vim(有高亮显示)编辑器

### 1、vim编辑器

在Linux下一般使用vi编辑器来编辑文件。 vi既可以查看文件也可以编辑文件。三种模式：命令行、插入、底行模式。

切换到命令行模式：按Esc键；

切换到插入模式：按i、o、a键；

i在当前位置前插入

I在当前行首插入

a在当前位置后插入

A在当前行尾插入

o在当前行之后插入一行

切换到底行模式：按: (冒号)；

打开文件: vim file

退出: esc + q

修改文件: 输入 i 进入插入模式

保存并退出: esc+wq

不保存退出: esc+q!

三种进入插入模式:

i : 在当前的光标所在处插入

o: 在当前光标所在的行的下一行插入

a : 在光标所在的下一个字符插入

快捷键:

dd - 快速删除一行

yy - 复制当前行

nyy - 从当前行向后复制几行

P - 粘贴

R - 替换

在 vim 查找关键字 -----> :/关键字

注: vim 不存在的文件.文件格式 -----> 进行编辑之后、会把数据保存且创建此文件

## 2、重定向输出>和>>

>重定向输出，覆盖原有内容；>>重定向输出，且追加功能

示例：

```
cat /etc/passwd/b.txt > a.txt 将输出定向到a. txt中(将左边的文件内容复制到右边文件中, 右侧文件不存在会自动创建)
```

```
cat /etc/pa3swd >> a.txt 输出并且追加  
ifconfig > ifconfig.txt
```

## 3、系统管理命令

ps 正在运行的某个进程的状态

```
ps -ef 查看所有(正在运行的)进程
```

```
ps -ef | grep java 查找某一(正在运行的)进程(查找包含java的进程)
```

```
kill 2868 杀掉2868编号的进程
```

```
kill -9 2868 强制杀死进程
```

## 4、管道 |

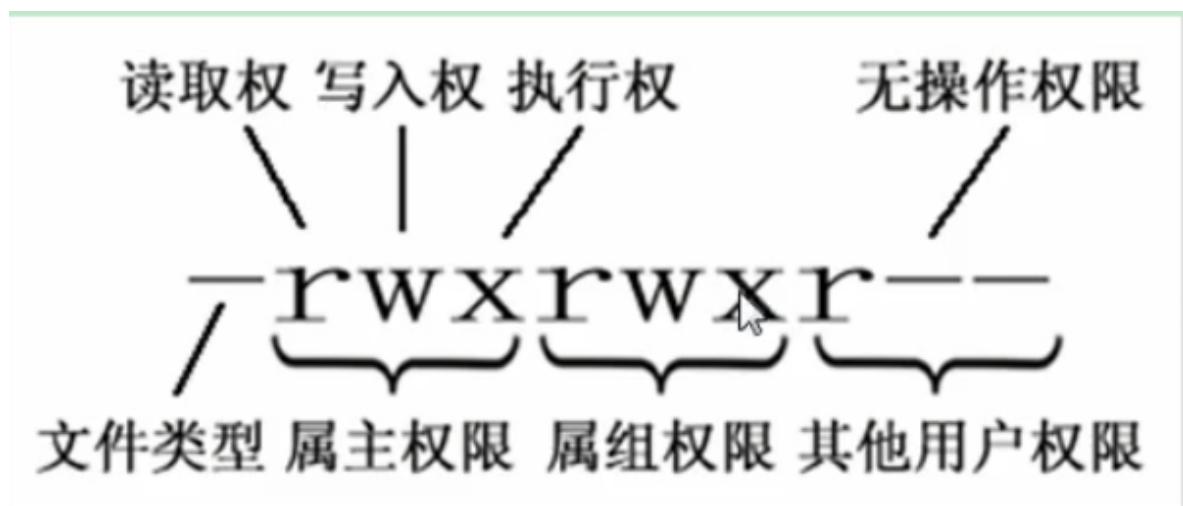
管道是Linux命令中重要的一个概念，其作用是将一个命令的输出用作另一个命令的输入。

示例

分页查询帮助信息: ls --help | more  
查询名称中包含java的进程: ps -ef | grep java  
ifconfig | more  
cat index.html | more  
ps -ef | grep aio

# Linux权限命令

## 1、文件权限



属主(user)			属组(group)			其他用户	
r	w	x	r	w	x	r	w
4	2	1	4	2	1	4	2

前面这串，代表权限

```

drwxr-xr-x. 4 root root      4096 Oct 30 19:02 aa
-rw-r--r--. 1 root root      964 Oct 30 20:23 aa.txt
-rw-----. 1 root root     1604 Oct 30 09:37 anaconda-ks.cfg
drwxr-xr-x. 9 root root     4096 Oct 30 18:09 apache-tomcat-7.0.51
-rw-r--r--. 1 root root 14254080 Oct 30 18:59 a.tar
-rw-r--r--. 1 root root      964 Oct 30 20:20 a.txt
-rw-r--r--. 1 root root      33 Oct 30 20:09 bb.txt
-rw-r--r--. 1 root root 8908965 Oct 30 19:00 b.tar.gz
-rw-r--r--. 1 root root      11 Oct 30 20:11 cc.txt
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Desktop
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Documents
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Downloads
-rw-r--r--. 1 root root      45 Oct 30 19:08 install.log
-rw-r--r--. 1 root root 10033 Oct 30 09:37 install.log.syslog
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Music
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Pictures
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Public
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Templates
drwxr-xr-x. 2 root root     4096 Oct 30 09:40 Videos

```

1、代表文件类型

-表示文件

d表示文件夹

l表示连接

2、当前用户具有该文件的权限

r: read 读

w: write 写

x: excute 可执行

3、当前组内其他用户具有该文件的权限

r: read 读

w: write 写

x: excute 可执行

4、其他组的用户具有该文件的权限

r: read 读

w: write 写

x: excute 可执行

r: 对文件定指可实取内容，对目录是可以ls

w: 对文件是指可修改文件内容，对目录是指可以在其中创建或删除子节点(目录或文件)

x :对文件是指是否可以运行这个文件，对目录是指是否可以cd进入这个目录

## 2、Linux三种文件类型

普通文件: 包括文本文件、数据文件、可执行的二进制程序文件等。

目录文件: Linux系统把目录看成是一种特殊的文件, 利用它构成文件系统的树型结构。

设备文件:Linux系统把每一个设备都看成是一个文件

## 3、文件类型标识

普通文件(-)目录(d)符号链接(1)

进入etc可以查看, 相当于快捷方式字符设备文件(c)块设备文件(a)套接字(3)命名管道(p)

## 4、文件权限管理

变更文件或目录的权限: chmod

chmod 755 a.txt

chmod u=rwx,g=rx,o=rx a. txt

简写: chome 755 a.txt(7: user的三个操作权限之和, 最后两个5: 组和其他用户的读取和运行权限),  
最大权限是777

注: 在linux终端内, 绿色代表可执行文件, 红色代表压缩文件, 白色代表可读写文件

# Linux上常用网络操作

## 1、主机配置

查看主机名: hostname———>安装时, 给linux取的名字

修改主机名 重启后无效: hostname xxx

如果想要永久生效, 可以修改 (cat) /etc/sysconfig/network文件

cat /etc/sysconfig/network 打开后, 可查看其内容

vim /etc/sysconfig/network 打开后, 进行修改HOSTNAME的值

## 2、ip地址配置

ifconfig查看(修改) ip地址(重启后无效)

ifconfig eth0 192.168.12.22 修改ip地址

如果想要永久生效

修改 */etc/sysconfig/network-scripts/ifcfg-eth0*文件

DEVICE=eth0 #网卡名称

BOOTPROTO=static #获取ip的方式(static/dhcp/bootcp/none)

再把下面的复制下去

```
HWADDR=00:0C:29:B5:B2:69 #MAC 地址  
IPADDR=12.168.177.129 #IP地址  
NETMASK=255.255.255.0 #子网掩码  
NETWORK=192.168.177.0 #网络地址  
  
BROADCAST=192.168.0.255 #广播地址  
  
NBOOT=yes #系统启动时是否设置此网络接口, 设置为yes时, 系统启动时激活此设备
```

### 3、域名映射

/etc/hosts文件用于在通过主机名进行访问时做ip地址解析之用

相当于Windows下的: C:\Windows\System32\drivers\etc\host

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6  
192.168.19.128 itcast  
192.168.19.129 itcast
```

### 4、网络服务管理

service network status 查看指定服务的状态

service network stop 停止指定服务

aervice network start 启动指定服务

service network restart 重启指定服务

service --status -all 查看系统中所有后台服务

netstat -nltp 查看系统中网络进程的端口监听情况

防火墙设置

防火墙根据配置文件/etc/sysconfig/iptables来控制本机的"出"、  
"入"网络访问行为

service iptables status 查看防火墙状态

**service iptables stop** 关闭防火墙

chkconfig iptables off 禁止防火墙自启

## Linux上安装软件

Linux上的软件安装有以下几种常见方式介绍。

### 1. 二进制发布包(缺点: 各个平台互不兼容)

软件已经针对具体平台编译打包发布, 只要解压, 修改配置即可。

## 2. RPM包(缺点：不会安装所依赖的软件包)

软件已经按照redhat的包管理工具规范RPM进行打包发布，需要获取到相应的软件RPM发布包，然后用RPM命令进行安装

## 3. Yum在线安装(缺点：速度慢)

软件已经以RPM规范打包，但发布在了网络上的一些服务器上，可用yum在线安装服务器上的rpm软件，并且会自动解决软件安装过程中的库依赖问题

## 4. 源码编译安装

软件以源码工程的形式发布，需要获取到源码工程后用相应开发工具进行编译打包部署。

# 上传方式

## 1、在windows上安装工具

例如：xftp、xshell、FileZilla

## 2、在linux上安装

1、安装(-y是免确认)：

```
sudo apt-get install lrzs # ubuntu  
sudo yum -y install lrzs  
# centos(此处可没有sudo)  
然后在SecureCRT的选项-->会话选项-->X/Y/Zmodem中更改上传和下载目录
```

2、简单使用sz(发送)、rz(接收)

把需要下载的文件放在上传目录(windows中)，输入rz(linux)，会调用SecureCRT，选中，添加，确定，上传到当前目录

输入sz，后面跟上传的文件，下载到所配置的下载目录中

## 3、使用linux自带的sftp

使用alt+p，打开sftp窗口

下载(在linux中，用户的根目录，输入cd [~]，进入根目录)：put 盘符://路径//文件(windows的文件)

上传(下载到windows的文档内)：get 所需文件名称

# 在linux中安装jdk

准备操作：查看linux的版本64/32：getconf LONG\_BIT

## 1、上传jdk到linux服务器

上传jdk (/usr/local/自定义的jdk文件夹)

卸载自带的open-jdk；如下

查看jdk版本——>java -version

查看安装的jdk信息——>rpm -qa | grep java

卸载自带的jdk？ ——>rpm -e --nodeps 自带jdk的名称

rpm -e --nodeps java-1.6.0-openjdk-1.6.0.35-1.13.7.1.e16\_6.i686

rpm -e --nodeps java-1.7.0-openjdk-1.7.0.79-2.5.5.4.e16\_6.i686

## 2、在linux上安装jdk

通常将软件安装到/usr/local

直接解压就可以

解压：tar -Xvf jdk.tar.gz -C 目标路径

```
tar -xvf jdk-7u71-linux-i586.tar.gz -C ./jdk
```

## 3、配置JDK的环境变量

vim /etc/profile

在末尾添加(home右边是jdk所在()绝对路径)，输入i，进入添加模式

```
#set java environment
```

```
JAVA_HOME=/usr/local/jdk/jdk1.7.0_71
```

```
CLASSPATH=. :$JAVA_HOME/lib/tools.jar
```

```
PATH=&JAVA_HOME/bin:$PATH
```

```
export JAVA_HOME CLASSPATH PATH
```

保存退出， esc:wq

source/etc/profile 使更改的置立即生效

## 在linux上安装mysql

### 1、将mysql的安装文件上传到linux服务器

1.1、rz——>在SecureCRT选中所需文件，进行上传 (/usr/local/到自定义的mysql文件夹)

1.2、将mysql的tar解压

1.3、将系统自动=自带的mysql卸载

```
rpm -qa |grep mysql———>搜索自带的mysql，再删除
```

```
rpm -e --nodeps mysql-libs-5.1.73-5.e16_6.i686
```

#### 1.4、解压mysql

```
tar -xvf MySQL-5.6.22-1.e16.1686.rpm- bundle. tar(mysql压缩包名称，带.gz的，为 -zxvf)
```

## 2、安装mysql服务器端(server)——>rpm -ivh mysql服务器端的名称

```
rpm -ivh MySQL -server-5. 6.22-1.e16.i686. rpm
```

下面的提示是告诉我们root用户的密码第一次是随机生成的，它保存在/root/.mysql\_secret中，第一次登录需要修改root密码

```
A RANDOM PASSWORD HAS BEEN SET FOR THE MySQL root USER ! 对root用户设置了一个随机密码  
You will find that password in '/root/.mysql_secret'. 密码的路径  
  
You must change that password on your first connect,  
no other statement but 'SET PASSWORD' will be accepted.  
See the manual for the semantics of the 'password expired' flag.  
  
[root@itcast01 mysql]# cat /root/.mysql_secret      对应文件路径，进行查看  
# The random password set for the root user at Mon Nov  5 05:43:44 2018 (local time): _aBE6Mjknsbb4s1t    随机密码
```

## 3、安装mysql客服端(client)

```
rpm -ivh MySQL -client-5. 6.22-1.e16.i686. rpm
```

## 4、登录

启动mysql的服务——>service mysql start

首次登录使用随机生成的密码——>mysql -uroot -p随机密码

第一次操作mysql，必须修改root用户密码，否则报错

```
mysql> show databases;  
ERROR 1820 (HY000): You must SET PASSWORD before executing this statement
```

```
set password = password('root密码');
```

## 5、windows端，mysql可视化工具远程登录

首先，关掉linux的防火墙：service iptables stop

授权所有的权限给root用户(mysql的用户)

```
grant all privileges on 库名.表名 to '用户名'@'IP地址' identified by '密码' with  
grant option;远程登录时，需要虚拟机的mysql授权
```

增加用户root，密码为root。其对所有数据库所有权限。可以外网访问

刷新权限

```
grant all privileges on *.* to 'root' @'%' identified by 'root';
flush privileges;
```

在linux中很多软件的端口都被“防火墙”限止，我们需要将防火墙关闭

防火墙打开3306端口。

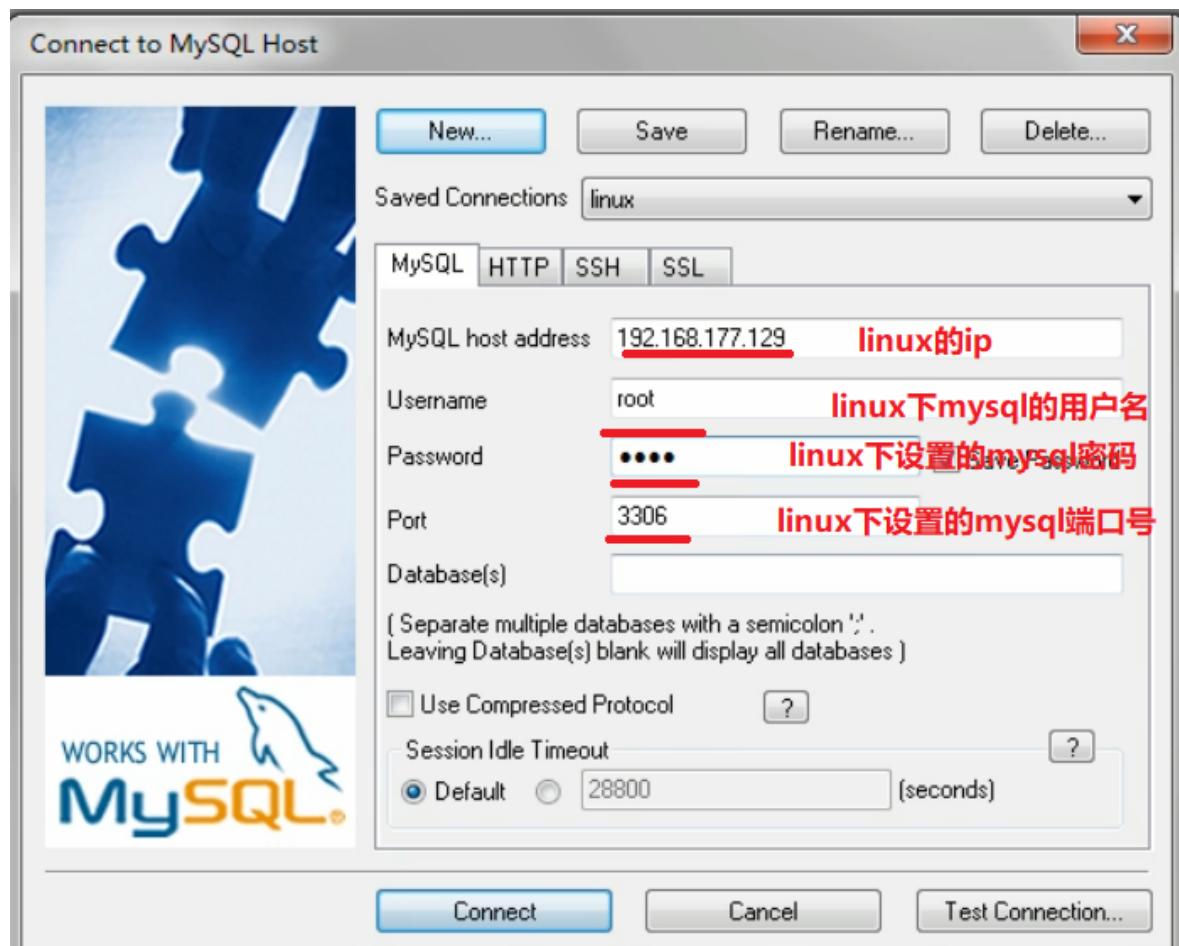
```
/sbin/iptables -I INPUT -P tcp --dport 3306 -j ACCEPT
```

```
/etc/rc.d/init.d/iptables save
```

```
/etc/init.d/iptables status
```

学习阶段我们也可以直接将防火墙关闭

```
service iptables stop;
```



## linux下Mysql命令

- 1, 进入mysql, 终端中输入 mysql -u 用户名 -p 。enter键后, 提示输入密码。
- 2, 执行grant all privileges on xxxdb.\* to user@ "%" identified by "passdtest"; 增加用户userstest, 密码为passdtest。其对数据库xxxdb有所有权限。可以外网访问。
- 3, show databases; 显示此用户下的数据库
- 4, select host, user from mysql.user; 查看用户的权限

### 1、创建新用户

通过root用户登录之后创建

```
>> grant all privileges on *.* to testuser@localhost identified by "123456"; // 创建新用户, 用户名为testuser, 密码为123456;  
>> grant all privileges on *.* to testuser@localhost identified by "123456"; // 设置用户testuser, 可以在本地访问mysql  
>> grant all privileges on *.* to testuser@ "%" identified by "123456"; // 设置用户testuser, 可以在远程访问mysql  
>> flush privileges; // mysql 新设置用户或更改密码后需用flush privileges刷新MySQL的系统权限相关表, 否则会出现拒绝访问, 还有一种方法, 就是重新启动mysql服务器, 来使新设置生效
```

### 2、设置用户访问数据库权限

```
>> grant all privileges on test_db.* to testuser@localhost identified by "123456"; // 设置用户testuser, 只能访问数据库test_db, 其他数据库均不能访问;  
>> grant all privileges on *.* to testuser@localhost identified by "123456"; // 设置用户testuser, 可以访问mysql上的所有数据库;  
>> grant all privileges on test_db.user_infor to testuser@localhost identified by "123456"; // 设置用户testuser, 只能访问数据库test_db的表user_infor, 数据库中的其他表均不能访问;
```

### 3、设置用户操作权限

```
>> grant all privileges on *.* to testuser@localhost identified by "123456" WITH GRANT OPTION; //设置用户testuser, 拥有所有的操作权限, 也就是管理员;  
>> grant select on *.* to testuser@localhost identified by "123456" WITH GRANT OPTION; //设置用户testuser, 只拥有【查询】操作权限;  
>> grant select,insert on *.* to testuser@localhost identified by "123456"; //设置用户testuser, 只拥有【查询插入】操作权限;  
>> grant select,insert,update,delete on *.* to testuser@localhost identified by "123456"; //设置用户testuser, 只拥有【查询插入】操作权限;  
>> REVOKE select,insert ON what FROM testuser //取消用户testuser的【查询插入】操作权限;
```

### 4、设置用户远程访问权限

```
>> grant all privileges on *.* to testuser@"192.168.1.100" identified by "123456"; //设置用户testuser, 只能在客户端IP为192.168.1.100上才能远程访问mysql;
```

### 5、关于root用户的访问设置

设置所有用户可以远程访问mysql, 修改my.cnf配置文件, 将bind-address = 127.0.0.1前面加#"注释掉, 这样就可以允许其他机器远程访问本机mysql了;

```
>> grant all privileges on *.* to root@ "%" identified by "123456"; // 设置用户root, 可以在远程访问mysql
```

```
>> select host,user from user; //查询mysql中所有用户权限
```

关闭root用户远程访问权限

```
>> delete from user where user="root" and host="%" ; //禁止root用户在远程机器上访问mysql
```

```
>> flush privileges; //修改权限之后, 刷新MySQL的系统权限相关表方可生效
```

# 在linux上安装tomcat

## 1.Tomcat上传到linux 上

在windows上, 将需要的tomcat文件, 放在SecureCRT的上传文件夹内

/usr/local/到自定义的tomcat文件夹

linux下输入: rz

选择, 添加, 确定

## 2.将.上传的tomcat解压

tar -zxvf tomcat名称(带gz后缀, 是-zxvf, 没有gz, -xvf)

### 3.在tomcat/bin目录下执行startup.sh (注意防火墙)

进入tomcat的bin目录： cd bin

启动tomcat: startup.sh(停止: shutdown.sh)

### 4.查看目标tomcat/logs/catalina.out

## 在linux上安装redis

注：博客上手动安装详见<https://www.cnblogs.com/zdd-java/p/10288734.html> (网页全屏: f12  
——>ctrl+shift+p——>输入Capture)

The screenshot shows a blog post titled "Linux安装redis和部署". The main content area displays the Redis website's download page for version 4.0.9. Below the website screenshot, there is a "Download" section with a red-bordered "Download 4.0.9" button. To the right of the main content, there is a sidebar with a calendar for May 2020, a search bar, and a "我的标签" (My Tags) section listing "Filter(1)", "Java代码重构简化(1)", "Java跨域请求(1)", "Maven(1)", and "Redis(1)". The bottom of the sidebar shows a portion of a terminal window with command-line logs related to the Redis installation process.

**注：如果在编译过程中出现问题，有可能是安装包下载的有误，这里可以尝试的用别人下载的安装包或者直接从官网下载。**

```
wget http://download.redis.io/releases/redis-4.0.9.tar.gz
```

如果发现上述读不能解决问题,请参照该链接: <https://www.cnblogs.com/liu2-/p/6914159.html>

编译完成之后，可以看到解压文件redis-3.0.7 中会有对应的src、conf等文件夹

```
[root@ouyuli redis-4.0.9]# ll
total 304
-rw-rw-r-- 1 root root 157632 Mar 27 00:04 00-RELEASENOTES
-rw-rw-r-- 1 root root 53 Mar 27 00:04 BUGS
-rw-rw-r-- 1 root root 1815 Mar 27 00:04 CONTRIBUTING
-rw-rw-r-- 1 root root 1487 Mar 27 00:04 COPYING
drwxrwxr-x 6 root root 4096 Apr 10 00:04 deps
drwxrwxr-x 2 root root 4096 Mar 27 00:04 include
drwxrwxr-x 1 root root 151 Mar 27 00:04 Makefile
-rw-rw-r-- 1 root root 4223 Mar 27 00:04 MANIFESTO
-rw-rw-r-- 1 root root 20543 Mar 27 00:04 README.md
-rw-rw-r-- 1 root root 58766 Mar 27 00:04 redis.conf
-rwxrwxr-x 1 root root 271 Mar 27 00:04 runtest
-rwxrwxr-x 1 root root 280 Mar 27 00:04 runtest-cluster
-rwxrwxr-x 1 root root 287 Mar 27 00:04 runtest-sentinel
-rwxrwxr-x 1 root root 7565 Mar 27 00:04 runtest-timed.conf
-rwxrwxr-x 3 root root 4096 Apr 10 00:03 src
drwxrwxr-x 10 root root 4096 Mar 27 00:04 tests
drwxrwxr-x 8 root root 4096 Mar 27 00:04 utils
[root@ouyuli redis-4.0.9]#
```

3. 编译成功后，进入src文件夹，执行make install进行Redis安装

如下图示安装完成，界面如下

```
[root@ouyu redis-4.0.0]# cd src/
[root@ouyu src]# make install

Hint: It's a good idea to run 'make test' ;)

INSTALL install
INSTALL install
INSTALL install
INSTALL install
INSTALL install
[root@ouyu src] ll
total 2016
-rw-rw-r-- 1 root root 10291 Mar 27 00:04 adlist.c
-rw-rw-r-- 1 root root 3512 Mar 27 00:04 adlist.h
-rw-rw-r-- 1 root root 17016 Apr  4 10:02 adlist.o
-rw-rw-r-- 1 root root 1973 Mar 27 00:04 ae.c
-rw-rw-r-- 1 root root 4846 Mar 27 00:04 ae epoll.c
-rw-rw-r-- 1 root root 10939 Mar 27 00:04 ae_export.c
-rw-rw-r-- 1 root root 5388 Mar 27 00:04 ae.h
-rw-rw-r-- 1 root root 4567 Mar 27 00:04 ae_kqueue.c
-rw-rw-r-- 1 root root 35312 Apr  4 10:02 ae.o
-rw-rw-r-- 1 root root 20701 Mar 27 00:04 ae_select.c
-rw-rw-r-- 1 root root 3562 Mar 27 00:04 anet.h
-rw-rw-r-- 1 root root 49320 Apr  4 10:02 anet.o
-rw-rw-r-- 1 root root 64450 Mar 27 00:04 aof.c
-rw-rw-r-- 1 root root 141976 Apr  4 10:02 aof.o
-rw-rw-r-- 1 root root 2837 Mar 27 00:04 aollo.h
-rw-rw-r-- 1 root root 557 Mar 27 00:04 atomcvar.h
-rw-rw-r-- 1 root root 10822 Mar 27 00:04 bio.c
-rw-rw-r-- 1 root root 2120 Mar 27 00:04 bio.h
```

第三步：部署

1.为了方便管理，将Redis文件中的conf配置文件和常用命令移动到统一文件中

## 1) 创建hi

```
[root@yolo redis]# ls  
00-RELEASENOTES CONTRIBUTING deps Makefile README.md runtest runtest-sentinel src utils  
COPYING INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests  
[root@yolo redis]# cd src  
[root@yolo redis]# ./configure --prefix=/usr/local/redis  
[root@yolo redis]# ls
```

3) 回到刚刚安装目录，找到 `radio.conf`，将其复制移动到 `/usr/local/radio/etc/` 目录下。

七

#### REFERENCES

*Journal of Health Politics, Policy and Law*, Vol. 33, No. 3, June 2008  
DOI 10.1215/03616878-33-3-601 © 2008 by The University of Chicago

进入src目录，移动mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server

执行命令：mv mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server /usr/local/redis/bin/

```
[root@yulu redis]# mv Redis.com /usr/local/redis/etc/
[root@yulu redis]# cd src
[root@yulu redis]# ./configure --prefix=/usr/local/redis
[root@yulu redis]# make
[root@yulu redis]# make check
[root@yulu redis]# make install
[root@yulu redis]# cp ./redis.conf /usr/local/redis/etc/
[root@yulu redis]# ./redis-server /usr/local/redis/etc/redis.conf
```

```
bitops.c dict.h latency.c object.c redis-check-aof.c sds.o t_hash.c
bitops.o dict.o latency.h object.o redis-check-aof.o sentinel.c t_hash.o
blocked.c endianconv.c latency.o pgsort.c redis-check-rdb sentinel.o t_list.c
childinfo.c endianconv.h lazyfree.c pgsort.o redis-check-server sentinel.o t_set.c
childinfo.o endianconv.o lazyfree.o pgsort.o redis-check-rdb sentinel.h t_set.o
redis.conf
[root@yuli src]# ./mreleaserdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server /usr/local/redis/5.0.7
```

依次查看移动后文件

```
[root@yuli src]# cd /usr/local/redis/bin/
[root@yuli bin]# ls
mreleaserdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server
[root@yuli bin]# cd ..
[root@yuli redis]# cd ect/
[root@yuli ect]# ls
redis.conf
[root@yuli ect]#
```

## 2. 执行redis-server 启动redis

```
[root@yuli bin]# ./redis-server
5884c: 04 Apr 11:35:30.516 # o0000000000 Redis is starting o0000000000
5884c: 04 Apr 11:35:30.516 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=5884, just started
5884c: 04 Apr 11:35:30.516 # Warning: no config file specified, using the default config. In order to specify a config file use ./redis-server /path/to/redis.conf
5884c: 04 Apr 11:35:30.516 * Increased maximum number of open files to 10232 (it was originally set to 1024).
Redis 4.0.9 (0000000000) 64 bit
Running in standalone mode
Port: 6379
PID: 5884
http://redis.io
```



```
5884:M 04 Apr 11:35:30.539 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
5884:M 04 Apr 11:35:30.539 # Server initialized
5884:M 04 Apr 11:35:30.539 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this to take effect.
5884:M 04 Apr 11:35:30.540 # WARNING overcommit_memory '1' to '/etc/sysctl.conf' and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
5884:M 04 Apr 11:35:30.540 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
5884:M 04 Apr 11:35:30.540 * Ready to accept connections
```

## 3. 设置绑定ip (注: 该步骤如果不需要可省略)

```
# include /path/to/local.conf
# include /path/to/other.conf
#####
##### MODULES #####
#####
# Load modules at startup. If the server is not able to load modules
# it will abort. It is possible to use multiple loadmodule directives.
# loadmodule /path/to/my_module.so
# loadmodule /path/to/other_module.so
#####
##### NETWORK #####
#####
# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
# bind 192.168.1.100 0.0.0.1
# bind 127.0.0.1 ::1
#
# ---- WARNING ---- If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 loopback interface address (this means Redis will be able to
# receive connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
bind 127.0.0.1
#
# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited
```

如需要, 可将上图绑定ip改为指定ip。

## 4. 设置后台启动redis

### 1) 首先编辑conf文件, 将daemonize属性改为yes (表明需要在后台运行)

```
cd etc/
vim redis.conf
```

```
# unixsocket /tmp/redis.sock
# unixsocketperm 700
# Close the connection after a client is idle for N seconds (0 to disable)
timeout 0
# TCP keepalive.
# If non-zero, use SO_KEEPALIVE to send TCP ACKs to clients in absence
# of communication. This is useful for two reasons:
# 1) Detect dead peers.
# 2) Take the connection alive from the point of view of network
#   equipment in the middle.
# On Linux, the specified value (in seconds) is the period used to send ACKs.
# Note that to close the connection the double of the time is needed.
# On other kernels the period depends on the kernel configuration.
# A reasonable value for this option is 300 seconds, which is the new
# Redis default starting with Redis 3.2.1.
tcp-keepalive 300
#####
##### GENERAL #####
#####
# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
# daemonize no
#
# If you run Redis from upstart or systemd, Redis can interact with your
# supervisor tree. Options:
# supervised none - no supervision interaction
# supervised upstart - signal upstart by putting Redis into SIGSTOP mode
# supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
# supervised auto - detect upstart or systemd method based on
#   environment variables $SYSTEMD and $UPSTART. $NOTIFY_SOCKET is a required
#   variable.
# Note: these supervision methods only signal "process is ready".
# They do not enable continuous liveness pings back to your supervisor.
supervised no
```

将no修改为yes

### 2)、再次启动redis服务, 并指定启动服务配置文件

```
redis-server /usr/local/redis/etc/redis.conf
```

```
[root@yuli bin]# ./redis-server /usr/local/redis/etc/redis.conf → 启动redis
5671c: 04 Apr 12:58:40.250 # o0000000000 Redis is starting o0000000000
5671c: 04 Apr 12:58:40.250 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=5671, just started
[root@yuli bin]# ./redis-cli → 链接客户端
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> exit
[root@yuli bin]# netstat -ntulnp | grep 5279 → 查看进程
tcp        0      0 192.168.210.108:6379          0.0.0.0:*        LISTEN      2268/./redis-server
tcp        0      0 127.0.0.1:6379            0.0.0.0:*        LISTEN      2244/./redis-server
[root@yuli bin]#
```

配置后台运行成功。

不经一番寒彻骨,怎得梅花扑鼻香

分类: Redis



2020年1月(1)

2019年12月(1)

2019年11月(2)

2019年10月(1)

2019年9月(1)

2019年7月(5)

2019年6月(1)

2019年5月(1)

2019年4月(7)

2019年1月(2)

2018年11月(12)

2018年10月(1)

2018年9月(3)

2018年8月(3)

2018年7月(1)

2018年6月(1)

2018年5月(3)

2018年4月(7)

2018年3月(7)

2018年2月(1)

2018年1月(5)

2017年12月(1)

2017年11月(5)

2017年10月(4)

2017年9月(12)

2017年8月(3)

2017年7月(2)

2017年5月(2)

2017年4月(2)

2017年3月(4)

2017年2月(8)

2017年1月(1)

2016年12月(3)

2016年11月(3)

2016年10月(6)

2016年9月(1)

## 最新评论

1. Re:Linux安装redis和部署

有个问题, (使用的xshell) 在执行“第三步: 部署 2. 执行redis-server 启动redis”后, 退不去了 后来, 在vmware上面redis目录下面输入:redis-cli shutdown...  
--31楼下小黑

2. Re:【转】小程序实现单选多选功能

效果图呢?

--楼成家、

3. Re:【转】使用Freemarker实现网页静态化

----

达免哥 关注: 14 粉丝: 36 +加关注

« 上一篇: 【原】DjianGo Windows7下的安装  
» 下一篇: 【原】通过多线程分批处理派发任务

posted @ 2019-01-18 17:12 达免哥 阅读(38583) 评论(1) 编辑 收藏

5 0 推荐 反对

6666 --规格严格-功夫到家

4. Re: 【原】Redis实现生成自增流水号  
5555 --规格严格-功夫到家

5. Re: 【转】Java策略消除if else  
@ liusen1也可以采用spi的方式...  
-达免哥

评论列表  
#1楼 2020-04-21 15:44 31楼下小黑

有一个问题，（使用的xshell）在执行“第三步：部署 2. 执行redis-server 启动redis”后，退不去了

后来，在vmware上面redis目录下面输入redis-cli shutdown，就好多了。

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问 网站首页。

【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库  
【推荐】阿里宝盒16篇文章，深度讲解Java开发、系统设计、职业发展  
【推荐】斩获阿里offer的必看12篇面试合辑

Kubernetes 集群规划、部署及运营平台  
CNCF 认证，100% 开源、简单易用、一键部署、离线安装  
FIT2CLOUD 飞致云

相关博文：  
Linux安装redis和部署  
Linux上安装Redis教程  
Redis学习-----单节点安装部署  
LinuxCentOS7.2下安装Redis&配置Redis开机启动  
Ubuntu中安装部署IntelCSWebRTC  
» 更多推荐...  
精品问答：Java 技术 1000 问

最新 IT 新闻：  
研究人员发现细菌构建细胞壁关键机制，新药开发如虎添翼  
为什么朋友圈里转发“后浪”的大多是领导和老板、叔叔阿姨？  
为什么有些病毒感染人类细胞，有些却不？  
直降15万！拼多多五五折购车在即：又追加1万张消费券  
巴菲特罕见大翻车！抄底航空公司失利 股神也得割肉清仓  
» 更多新闻...

Copyright © 2020 达免哥  
Powered by .NET Core on Kubernetes

阅读排行榜

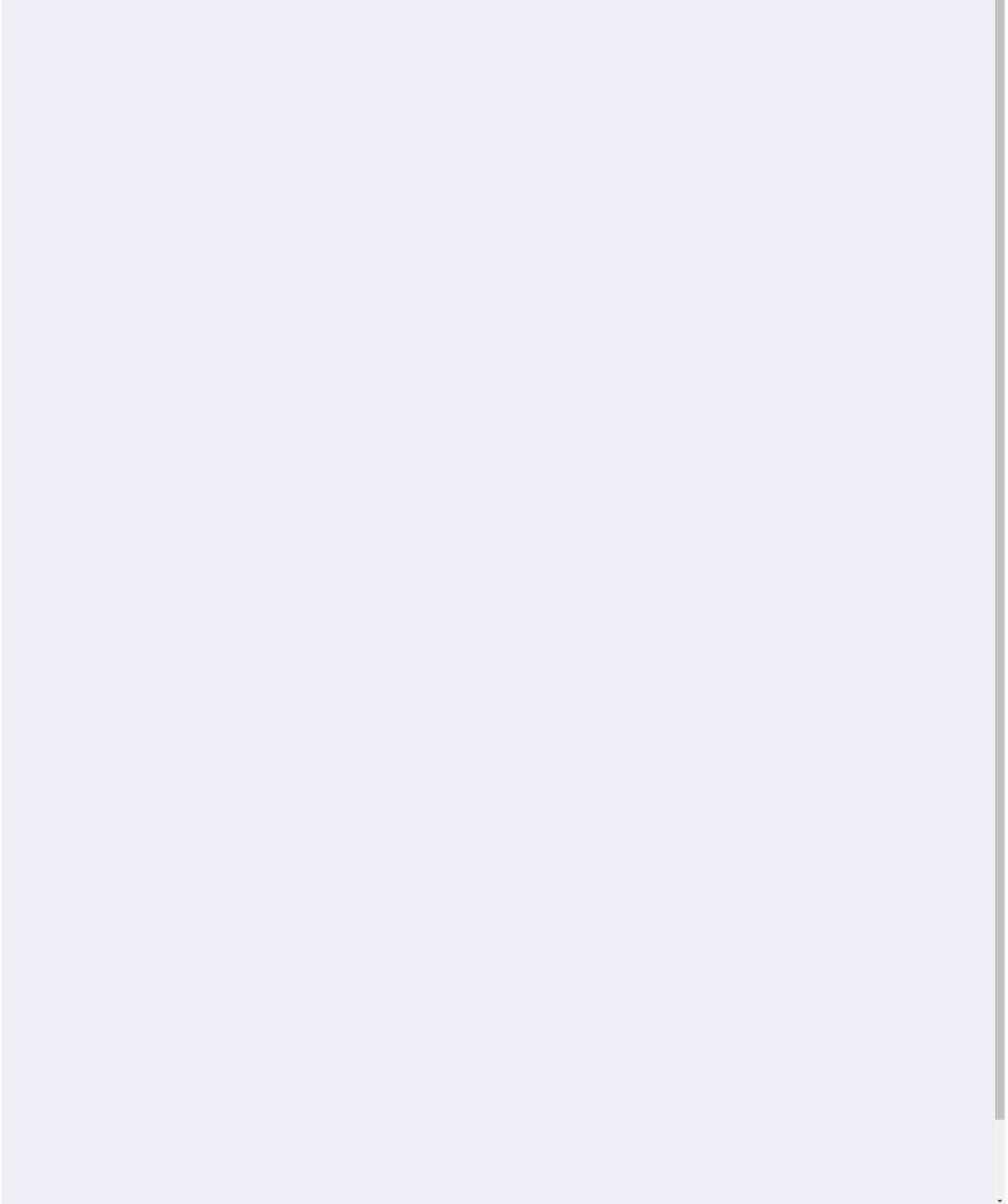
1. Linux安装redis和部署(38582)
2. 【原】Spring和Dubbo基于XML配置整合过程(16846)
3. 【原】通过Dubbo注解实现RPC调用(16515)
4. 【原】Nginx搭建FTP服务器的细节问题(13059)
5. 【原】ActiveMq实现分布式事务一致性(12306)

评论排行榜

1. 【原】继承AbstractRoutingDataSource再通过AOP实现动态数据源切换(24)
2. 【原】ActiveMq实现分布式事务一致性(5)
3. 【原】通过Dubbo注解实现RPC调用(5)
4. 【转】Java策略消除if else(4)
5. 【原】Spring AOP实现对Redis的缓存同步(4)

推荐排行榜

1. Linux安装redis和部署(5)
2. 【转】浅谈Nginx负载均衡与F5的区别(4)
3. 【转】Cookie深度解析(3)
4. 【原】通过Dubbo注解实现RPC调用(2)
5. 【原】Spring和Dubbo基于XML配置整合过程(2)



[https://blog.csdn.net/qq\\_30764991/article/details/81564652](https://blog.csdn.net/qq_30764991/article/details/81564652)

## Redis简介

Redis官网: <http://redis.io/> Redis是一个开源的高性能键值对数据库。它通过提供多种键值数据类型来适应不同场景下的存储需求，并且借助许多高级别的接口使其可以胜任，如缓存、队列系统的不同角色。

源码托管在github <https://github.com/antirez/redis> 下载linux的版本的

### 1. Redis的安装

#### 1. 1. Redis的安装

Redis是c语言开发的。

安装redis需要c语言的编译环境。如果没有gcc需要在线安装。

```
yum install gcc-c++
```

安装步骤：

第一步：redis的源码包上传到linux系统。

第二步：解压缩redis。tar zxvf redis-3.0.0.tar.gz

第三步：编译。cd redis-3.0.0

```
[root@admin redis-3.0.0]# make
```

进入redis源码目录。make

第四步：安装。make install PREFIX=/usr/local/mysoft/redis 后面是你自己的安装路径。

PREFIX参数指定redis的安装目录。一般软件安装到/usr目录下

```
make[1]: Entering directory '/usr/local/mysoft/redis/redis-3.0.0/src'
Hint: It's a good idea to run 'make test' ;)

INSTALL install      说明make编译成功
INSTALL install
INSTALL install
INSTALL install
INSTALL install
make[1]: Leaving directory '/usr/local/mysoft/redis/redis-3.0.0/src'
[root@admin redis-3.0.0]# cd /usr/local/mysoft/redis
[root@admin redis]# ls
bin  redis-3.0.0  redis-3.0.0.tar.gz    编译之后所得到的redis bin
[root@admin redis]# cd bin
[root@admin bin]# ls
redis-benchmark  redis-check-aof  redis-check-dump  redis-cli  redis-sentinel  redis-server
[root@admin bin]# ./redis-server          启动redis-server服务端
https://blog.csdn.net/qq_30764991
```

#### 1. 1. 连接redis

1. redis的启动：

前端启动：在redis的安装目录下直接启动redis-server

```
[root@localhost bin]# ./redis-server , 成功之后得到如下图
```



退出：Ctrl+c 即可退出。

后台启动：

把/root/redis-3.0.0/redis.conf复制到/usr/local/redis/bin目录下

```
[root@localhost redis-3.0.0]# cp redis.conf /usr/local/redis/bin/
```

```
[root@admin mysoft]# cd redis      找到redis解压后的redis-3.0.0
[root@admin redis]# ls
bin  redis-3.0.0  redis-3.0.0.tar.gz
[root@admin redis]# cd redis-3.0.0
[root@admin redis-3.0.0]# ll          把redis-3.0.0中的redis.conf复制到安装的bin中
总用量 144
-rw-rw-r--  1 root root 25890 4月   1 2015 00-RELEASENOTES
-rw-rw-r--  1 root root     52 4月   1 2015 LICENSE
```

```
-rw-rw-r-- 1 root root 35 4月 1 2015 BUGS
-rw-rw-r-- 1 root root 1439 4月 1 2015 CONTRIBUTING
-rw-rw-r-- 1 root root 1487 4月 1 2015 COPYING
drwxrwxr-x 6 root root 4096 8月 11 00:47 deps
-rw-rw-r-- 1 root root 11 4月 1 2015 INSTALL
-rw-rw-r-- 1 root root 151 4月 1 2015 Makefile
-rw-rw-r-- 1 root root 4223 4月 1 2015 MANIFESTO
-rw-rw-r-- 1 root root 5261 4月 1 2015 README
-rw-rw-r-- 1 root root 41403 4月 1 2015 redis.conf
-rwxrwxr-x 1 root root 271 4月 1 2015 runtest
-rwxrwxr-x 1 root root 280 4月 1 2015 runtest-cluster
-rwxrwxr-x 1 root root 281 4月 1 2015 runtest-sentinel
-rw-rw-r-- 1 root root 7109 4月 1 2015 sentinel.conf
drwxrwxr-x 2 root root 4096 8月 11 00:48 src
drwxrwxr-x 10 root root 4096 4月 1 2015 tests
drwxrwxr-x 5 root root 4096 4月 1 2015 utils
[root@admin redis-3.0.0]# cp redis.conf /usr/local/mysoft/redis/bin
```

```
[root@admin redis-3.0.0]# cd /usr/local/mysoft/redis/bin
[root@admin bin]# ll
总用量 13896
-rw-r--r-- 1 root root 18 8月 11 01:01 dump.rdb
-rwxr-xr-x 1 root root 4169558 8月 11 00:49 redis-benchmark
-rwxr-xr-x 1 root root 16475 8月 11 00:49 redis-check-aof
-rwxr-xr-x 1 root root 37711 8月 11 00:49 redis-check-dump
-rwxr-xr-x 1 root root 4258344 8月 11 00:49 redis-cli bin中复制得到的redis.conf
-rw-r--r-- 1 root root 41403 8月 11 01:16 redis.conf
lrwxrwxrwx 1 root root 12 8月 11 00:49 redis-sentinel ln -s redis.conf redis-server
-rwxr-xr-x 1 root root 5688317 8月 11 00:49 redis-server
```

编辑修改配置文件：vim redis.conf

```
#####
##### GENERAL #####
# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
daemonize yes 把原来的no修改为yes
# When running daemonized, Redis writes a pid file in /var/run/redis.pid by
# default. You can specify a custom pid file location here.
pidfile /var/run/redis.pid
:wq 按Esc之后，输入:wq ---这个是保存退出
```

[root@admin bin]# ./redis-server redis.conf 这样就后台启动了

检查后台到底是否启动呢？那么我们可以查看进程： ps aux|grep redis

[root@admin bin]# ps aux|grep redis，看到说明成功了！

```
[root@admin bin]# ps aux|grep redis
root 9727 0.1 0.1 35552 1724 ? https Ssl 01:23 0:00 ./redis-server *:6379
root 9734 0.0 0.0 5980 744 pts/0 S+ 01:25 0:00 grep redis
```

结束进程：[root@admin bin]# kill 9727 这个是正常结束，kill-9强制结束！

通过客户端关闭如下：

关闭redis：[root@localhost bin]# ./redis-cli shutdown

至此：linux安装redis 就好了。再次提醒启动方式：./redis-server redis.conf 。结束两种方式：1.kill +加redis的端口号

客户端：2，./redis-cli shutdown

接下来稍稍演示redis的使用：

开启redis服务：[root@admin bin]# ./redis-server redis.conf

查看redis服务是否启动：

```
[root@admin bin]# ps aux|grep redis
root 9788 0.0 0.1 35552 1724 ? Ssl 01:38 0:00 ./redis-server *:6379
root 9792 0.0 0.0 5980 740 pts/0 S+ 01:38 0:00 grep redis
```

启动redis客户端：[root@admin bin]# ./redis-cli

默认得到的是本机的ip:127.0.0.1:6379>

测试：ping

会得到：PONG

quit,退出进程

切换redis的ip:

```
[root@admin bin]# ./redis-cli -h 192.168.60.130 -p 6379
```

ok,以上就演示简单的redis命令

## 1、安装gcc-c++(此步推荐在虚拟机里面操作)

redis是C语言开发，安装redis需要先将官网下载的源码进行编译，编译依赖gcc环境。

进入： /usr/local

输入命令: yum install gcc-c++

确认下载： Is this ok [y/N]: y

确认安装输入： y

显示Complete，则安装成功

## 2、安装redis(linux下输入)

### 1、下载redis，在/usr/local文件夹

wget <http://download.redis.io/releases/redis-3.0.4.tar.gz>

### 2、解压

tar -zxvf redis-3.0.4.tar.gz

### 3、编译安装

切换至程序目录，并执行make命令编译

cd redis-3.0.4

make

执行安装命令(=右边是指定安装的目录，install是安装)

make PREFIX=/usr/local/redis install

make install 安装完成后，会在/usr/local/bin 目录下生成下面几个可执行文件，它们的作用分别是：

redis-server: Redis 服务器端启动程序

redis-cli: Redis 客户端操作工具。也可以用telnet根据其纯文本协议来操作

redis-benchmark: Redis 性能测试工具

redis-check-aof:数据修复工具

redis-check-dump:检查导出工具

## 3、配置redis

1、复制配置文件到/usr/local/redis/bin目录

cd redis-3.0.4

cp redis.conf /usr/local/redis/bin

2、启动redis

当前目录下的文件夹内的文件： ./redis - server redis.conf

也可以在其目录下启动： cd /usr/local/redis/bin—>./redis-cli

# Nginx

## 1、在linux上安装nginx

### 1.1、nginx的介绍

Nginx是一款高性能的http服务器/反向代理服务器及电子邮件( IMAP/POP3)代理服务器。由俄罗斯的程序设计师伊戈尔·西索夫(Igor Sysoev)所开发，官方测试nginx能够支撑5万并发连接，并且cpu、内存等资源消耗却非常低，运行非常稳定。



Igor Sysoev , Nginx的创始人

### 1.2、nginx应用场景

- 1、http服务器。Nginx是一个http服务可以独立提供http服务。可以做[网页静态服务器](#)。
- 2、虚拟主机。可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。
- 3、反向代理,负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用nginx做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

### 1.3、Nginx在linux下的安装

重新准备一台虚拟机作为服务器。比如IP地址为192.168.25.141

### 1.4、准备环境

- 1、需要安装gcc的环境： yum install gcc-c++()

## 2、第三方的开发包：PCRE、zlib、OpenSSL

```
yum install -y pcre pcre-devel
```

PCRE(Perl Compatible Regular Expressions)是一个Perl 库，包括perl 兼容的正则表达式库。nginx 的http 模块使用pcre 来解析正则表达式，所以需要在linux 上安装pcre库。

注：pcre-devel是使用pcre 开发的一个二次开发库。nginx 也需要此库。

```
zlib: yum install -y zlib zlib-devel
```

zlib库提供了很多种压缩和解压缩的方式，nginx 使用zlib 对http 包的内容进行gzip,所以需要在linux 上安装zlib 库。

```
OpenSSL: yum install -y openssl-devel
```

OpenSSL是一个强大的安全套接字层密码库,囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL 协议，并提供丰富的应用程序供测试或其它目的使用。nginx不仅支持http协议，还支持https (即在ssl 协议上上传输http)，所以需要在linux 安装openssl 库。

## 1.5、Ngnix下载

官方网站下载nginx: <http://nginx.org/>

我们课程中使用的版本是1.8.0 版本。

### 1.6、Ngnix的安装

第一步:把nginx的源码包nginx-1.8.0.tar.gz.上传到linux 系统

```
cd /usr/local
```

```
rz, 选中上传的文件
```

第二步:解压缩： tar zxvf nginx-1.8.0.tar.gz

第三步:进入nginx-1.8.0目录使用configure 命令创建一  
makeFile文件。

```
cd nginx-1.8.0
```

```
./configure \
--prefix=/usr/local/nginx\
--pid-path=/var/run/nginx/nginx.pid \
--lock-path=/var/lock/nginx.lock \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-http_gzip_static_ module \
--http-client-body-temp-path=/var/temp/nginx/client \
--http-proxy-temp-path=/var/temp/nginx/proxy \
--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \
--http-uwsgi-temp-path=/var/temp/nginx/uwsgi\
--http-scgi-temp-path=/var/temp/nginx/scgi
```

执行后可以看到Makefile文件(nginx-1.8.0下)

```
[root@itcast01 nginx-1.8.0]# ll
total 660
drwxr-xr-x. 6 1001 1001 4096 Nov  6 21:50 auto
-rw-r--r--. 1 1001 1001 249124 Apr 21 2015 CHANGES
-rw-r--r--. 1 1001 1001 379021 Apr 21 2015 CHANGES.ru
drwxr-xr-x. 2 1001 1001 4096 Nov  6 21:50 conf
-rw xr-xr-x. 1 1001 1001 2478 Apr 21 2015 configure
drwxr-xr-x. 4 1001 1001 4096 Nov  6 21:50 contrib
drwxr-xr-x. 2 1001 1001 4096 Nov  6 21:50 html
-rw-r--r--. 1 1001 1001 1397 Apr 21 2015 LICENSE
-rw-r--r--. 1 root root 345 Nov  6 21:51 Makefile
drwxr-xr-x. 2 1001 1001 4096 Nov  6 21:50 man
drwxr-xr-x. 3 root root 4096 Nov  6 21:51 objs
-rw-r--r--. 1 1001 1001 49 Apr 21 2015 README
drwxr-xr-x. 8 1001 1001 4096 Nov  6 21:50 src
[root@itcast01 nginx-1.8.0]#
```

小提示：

**Makefile**是一种配置文件，**Makefile**一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，**makefile** 定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为**makefile** 就像一个**Shell** 脚本一样，其中也可以执行操作系统的命令。。

```
configure参数
./configure \
--prefix=/usr\指向安装目录
--sbin-path=/usr/sbin/nginx\指向(执行)程序文件(nginx)中
--conf-path=/etc/nginx/nginx.conf\指向配置文件
--error-log-path=/var/log/nginx/error.log\指向log
--http-log-path=/var/log/nginx/access.log\指向http-log
--pid-path=/var/run/nginx/nginx.pid\指拘pidw
--lock-path=/var/ock/nginx.lock\ (安装文件锁定,防止安装文件被别人利用,或误操作。)
--user=nginx
--group=ngnix
--with-http_ssl_module\启用ngx_http_ssl_module支持(使支持https请求,需已安openssl)
--with-http_flv_module\启用ngx_http_flv_module支持(提供寻求内存使用基于时间的偏移量文件)
--with-http_stub_status_module\启用ngx_http_stub_status_module 支持(获取nginx自上次启动以来的工作状态)
--with-http_gzip_static_module\启用ngx_http_gzip_static_module支持(在线实时压缩输出数据流)
--http-client-body-temp-path=/var/tmp/nginx/client/\设定http客户端清求临时文件路径
--http-proxy-temp-path=/var/tmp/nginx/proxy/\设定http代理临时文件路径
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/\ 设定http fastcgi临时文件路径
--http uwsgi-temp-path=/var/tmp/nginx/uwsgi\设定http uwsgi临时文件路径
```

```
--http-scgi-temp-path=/var/tmp/nginx/scgi\设定http scgi临时文件路径
```

编译输入: make

```
-lthread -lcrypt -lpcre -lcrypto -lcrypto -lz
make[1]: Leaving directory `/usr/local/nginx-1.8.0'
make -f objs/Makefile manpage
make[1]: Entering directory `/usr/local/nginx-1.8.0'
sed -e "s|%%PREFIX%%|/usr/local/nginx|" \
      -e "s|%%PID_PATH%%|/var/run/nginx.pid|" \
      -e "s|%%CONF_PATH%%|/usr/local/nginx/conf/nginx.conf|" \
      -e "s|%%ERROR_LOG_PATH%%|/var/log/nginx/error.log|" \
      < man/nginx.8 > objs/nginx.8
make[1]: Leaving directory `/usr/local/nginx-1.8.0'
[root@itcast01 nginx-1.8.0]#
```

安装完成后安装, 输入: make install

```
cp conf/koi-win '/usr/local/nginx/conf'
cp conf/koi-utf '/usr/local/nginx/conf'
cp conf/win-utf '/usr/local/nginx/conf'
test -f '/usr/local/nginx/conf/mime.types'           || cp conf/mime.types '/usr/local/nginx/conf'
cp conf/mime.types '/usr/local/nginx/conf/mime.types.default'
test -f '/usr/local/nginx/conf/fastcgi_params'        || cp conf/fastcgi_params '/usr/local/nginx/conf'
cp conf/fastcgi_params '/usr/local/nginx/conf/fastcgi_params.default'
test -f '/usr/local/nginx/conf/fastcgi.conf'          || cp conf/fastcgi.conf '/usr/local/nginx/conf'
cp conf/fastcgi.conf '/usr/local/nginx/conf/fastcgi.conf.default'
test -f '/usr/local/nginx/conf/uwsgi_params'          || cp conf/uwsgi_params '/usr/local/nginx/conf'
cp conf/uwsgi_params '/usr/local/nginx/conf/uwsgi_params.default'
test -f '/usr/local/nginx/conf/scgi_params'          || cp conf/scgi_params '/usr/local/nginx/conf'
cp conf/scgi_params '/usr/local/nginx/conf/scgi_params.default'
test -f '/usr/local/nginx/conf/nginx.conf'            || cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf'
cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'
test -d '/var/run/nginx'                             || mkdir -p '/var/run/nginx'
test -d '/var/log/nginx' ||                           mkdir -p '/var/log/nginx'
test -d '/usr/local/nginx/html'                     || cp -R html '/usr/local/nginx'
test -d '/var/log/nginx' ||                           mkdir -p '/var/log/nginx'
make[1]: Leaving directory `/usr/local/nginx-1.8.0'
[root@itcast01 nginx-1.8.0]#
```

## 1.6、Nginx启动与访问

注意:启动nginx之前, 上边将临时文件目录指定为/var/temp/nginx/client, 需要在/var下创建此目录

mkdir var/temp/nginx/client -p(在nginx-1.8.0目录下)

进入local目录: cd ..

进入nginx目录: cd nginx: conf、html、sbin

(分别是配置文件目录, 静态资源(网页)目录、启动文件放置的目录)

输入./nginx

或

进入到Nginx目录下的sbin目录

cd /usr/local/nginx/sbin

输入启动命令Nginx

**./nginx**

启动后查看进程

ps aux|grep nginx

地址输入虚拟机的ip即可访问(默认端口为80)

关闭nginx: ./nginx -s stop(类似非正常退出)

或者: ./nginx -s quit(推荐这个(正常退出), 保存配置, 然后退出)

重启: nginx

1、先关闭后启动

2、刷新配置文件: ./nginx -s reload

## 2、Nginx静态网页部署

将/资料/静态页面/index 目录下的所有内容上传到服务器的/usr/local/nginx/html 下即可访问

进入conf配置文件夹: cd /usr/local/nginx/conf

打开nginx.conf配置文件: vim nginx.conf (带#蓝色的, 是被注释过的)

将做好的项目上传至linux的nginx中, 把location内改成项目的首页

```
location / {  
    root index;  
    index index.html index.htm;  
}
```

重启nginx: ./nginx -s reload

再次访问到linux的ip, 就变成了项目的首页面

### 2.1、端口绑定

可使用红本(EditPlus), 进行连接修改, 和传文件之类的

可修改Nginx的配置文件: /usr/local/nginx/conf/nginx.conf

```
server {  
    listen 80;                      #默认端口号  
    server_name localhost;          #域名或ip  
  
    location / {  
        root index;                #默认访问资源的目录  
        index index.html index.htm; #默认访问资源名称
```

```
}

error_page 500 502 503 504 / 50x.html;#错误页面
location = /50x.html {
    root html;
}

}
```

在复制一份server的配置，则会多一个访问ip或域名(其中的配置也可以更改)

## 2.2、域名绑定

### 1、顶级域名

顶级域名 又分为两类：

一是国家顶级域名 (nationaltop-leveldomainnames，简称nTLDs) ， 200多个国家都按照ISO3166国家代码分配了顶级域名，例如中国是cn，美国是us，日本是jp等；

二是国际顶级域名 (internationaltop-leveldomainnames，简称iTLDs) ， 例如表示工商企业的.Com，表示网络提供商的.net，表示非盈利组织的.org等。大多数域名争议都发生在com的顶级域名下，因为多数公司上网的目的都是为了赢利。为加强域名管理，解决域名资源的紧张，Internet协会、Internet分址机构及世界知识产权组织 (WIPO) 等国际组织经过广泛协商，在原来三个国际通用顶级域名：(com) 的基础上，新增加了7个国际通用顶级域名：firm (公司企业) 、store (销售公司或企业) 、Web (突出WWW活动的单位) 、arts (突出文化、娱乐活动的单位) 、rec(突出消遣、娱乐活动的单位) 、info(提供信息服务的单位) 、nom(个人) ，并在世界范围内选择新的注册机构来受理域名注册申请

例如：baidu.com

### 2、二级域名

二级域名 是指顶级域名之下的域名，在国际顶级域名下，它是指域名注册人的网上名称，例如ibm， yahoo， microsoft等；

在国家顶级域名下，它是表示注册企业类别的符号，例如com， edu， gov， net等。 中国在国际互联网络信息中心 (InterNIC) 正式注册并运行的顶级域名是CN，这也是中国的一级域名。在顶级域名之下，中国的二级域名又分为类别域名和行政区域名两类。类别域名共6个，包括用于科研机构的ac；用于工商金融企业的com；用于教育机构的edu；用于政府部门的gov；用于互联网络信息中心和运行中心的net；用于非盈利组织的org。而行政区域名有34个，分别对应于中国各省、自治区和直辖市

例如：map.baidu.com

### 3、三级域名

三级域名 用字母 (A~Z, a~z, 大小写等) 、数字 (0~9) 和连接符 (-) 组成，各级域名之间用实点 (.) 连接，三级域名的长度不能超过20个字符。如无特殊原因，建议采用申请人的英文名(或者缩写) 或者汉语拼音名(或者缩写) 作为三级域名，以保持域名的清晰性和简洁性

例如：item.map.baidu.com

域名和ip绑定:

一个域名对应一个ip地址，一个ip地址可以被多个域名绑定。

本地测试可以修改hosts文件(`C:\Windows\System32\drivers\etc`)

可以配置域名和ip的映射关系，如果hosts文件中配置了域名和ip的对应关系，不需要走dns服务器。

192.168.177.129      www.hmtravel1.com  
192.168.177.129      regist.hmtravel1.com

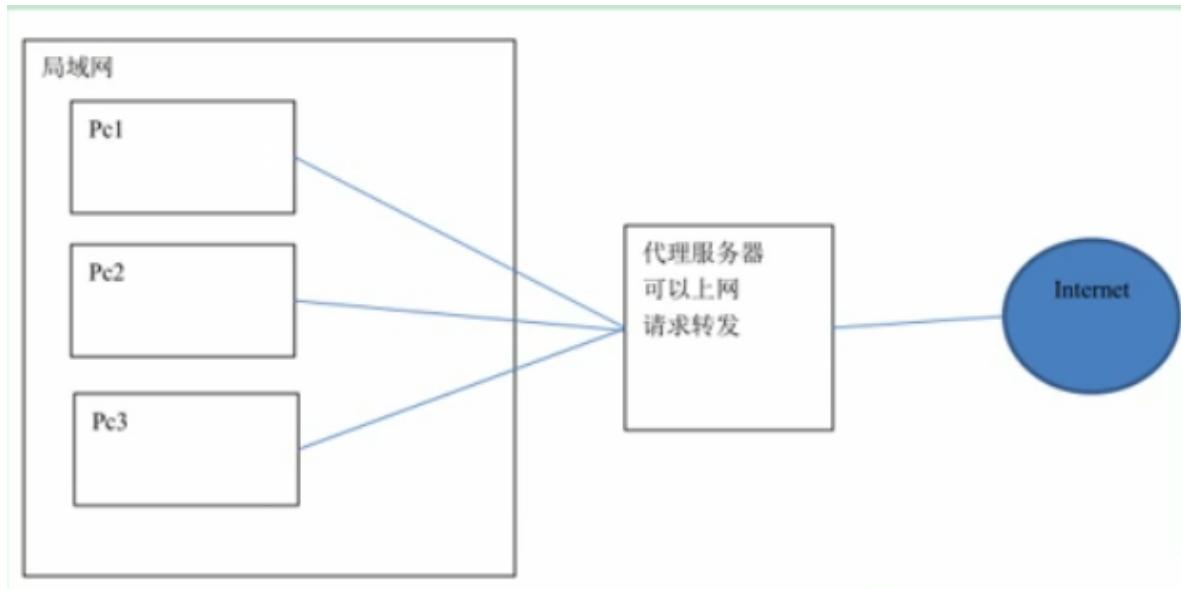
做好域名指向后，修改nginx配置文件

如果有多个server配置时，且端口都是80，那么把多个域名放进去；保存，重启nginx，即可使用域名访问

### 3、反向代理

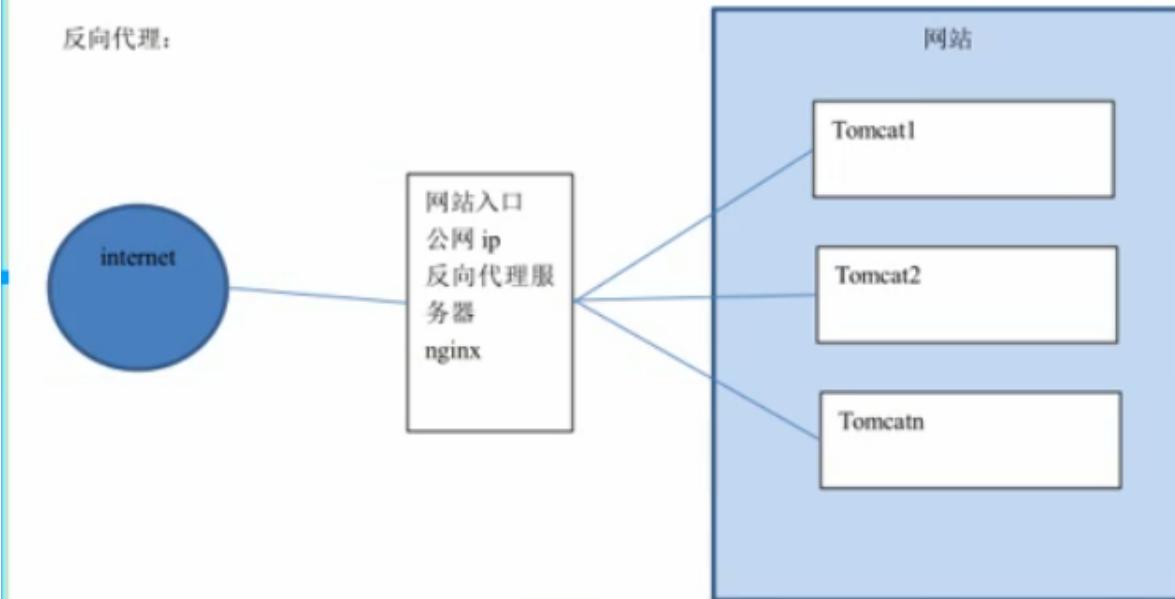
反向代理(Reverse Proxy)方式是指以代理服务器来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给internet上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。

首先我们先理解正向代理，如下图：



正向代理是针对你的客户端，而反向代理是针对服务器的，如下图

反向代理：



代理服务器使用正则映射将客户机请求转发给内部内容服务器



### 3.1、准备配置方向代理

1、将travel案例部署到tomcat中(ROOT目录)，上传到服务器。

进入到tomcat文件夹： cd .tomcat——>cd apache-tomcat-7.0.57/webapps/

删除webapps下的所有文件/文件夹： rm -rf /\*

把项目打包为ROOT.war文件

2、启动TOMCAT,输入网址<http://192.168.177.129:8080>可以看到网站首页

启动tomcat： cd .. /bin————>./startup.sh

### 3.2、配置反向代理

1、在Nginx主机修改Nginx配置文件

把server上的代码复制到server上

```

upstream tomcat-travel{
    server 192.168.177.129:8080;
}

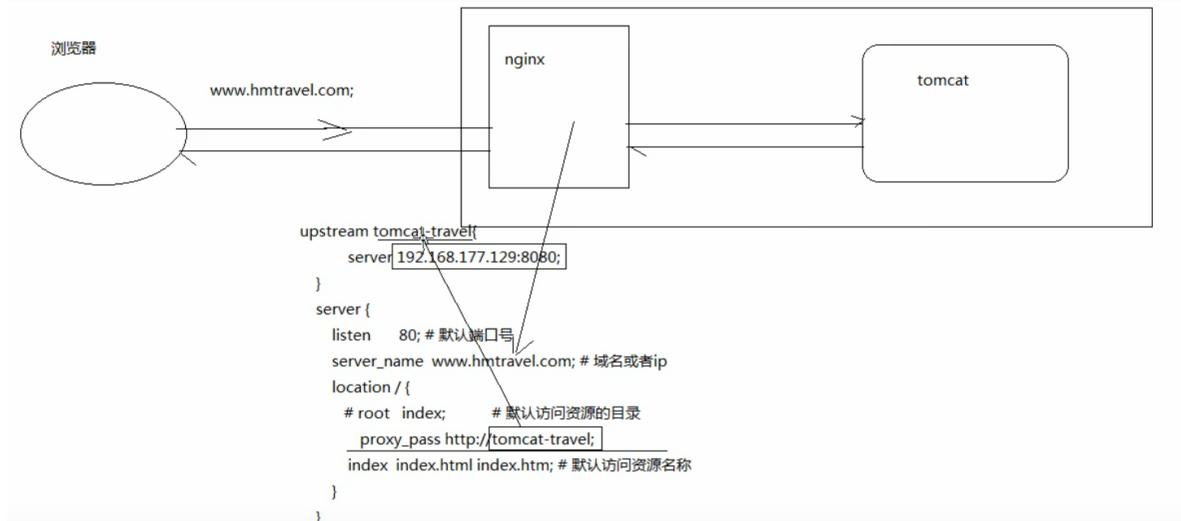
server {
    listen 80; #默认端口号
    server_name localhost; #域名或ip

    location / {
        #root index; #默认访问资源的目录
        proxy_pass http://tomcat-travel; #对应上面方法块的名字
        index index.html index.htm; #默认访问资源名称
    }

    error_page 500 502 503 504 / 50x.html;#错误页面
    location = /50x.html {
        root html;
    }
}

```

重启Nginx: ./nginx -s reload



## 4、负载均衡

当吞吐量较大时，需要多个tomcat时，则达到每一个tomcat的吞吐量平均

负载均衡建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡（Load Balance）其意思就是分摊到多个操作单元上进行执行，例如web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务

## 1、准备工作

停止tomcat：进入tomcat的bin文件夹： ./shutdown.sh；返回到tomcat文件夹

将刚才的存放工程的tomcat复制三份，修改端口分别为8080, 8081, 8082

```
cp -r apache-tomcat-7.0.57 tomcat1  
cp -r apache-tomcat-7.0.57 tomcat1  
cp -r apache-tomcat-7.0.57 tomcat1
```

三个tomcat下的conf内的server.xml的以下配置都不能一样(一般第一个可不用更改，但是要保证三个以下数据不一致)

```
22 <Server port="8005" shutdown="SHUTDOWN">  
23   <Listener className="org.apache.catalina.startup.VersionLoggerListener" />  
24   <!-- Security listener. Documentation at /docs/config/listeners.html  
25   <Listener className="org.apache.catalina.security.SecurityListener" />  
  
71 <Connector port="8080" protocol="HTTP/1.1"  
72   connectionTimeout="20000" 超时时间，可不更改  
73   redirectPort="8443" />  
  
93 <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />  
94
```

分别启动这三个tomcat服务。

tomcat1/bin/startup.sh

tomcat2/bin/startup.sh

tomcat3/bin/startup.sh

在nginx文件夹中conf配置文件夹内的nginx.conf文件

```
upstream tomcat-travel{  
    server 192.168.177.129:8080;      #此处的三个端口和上面第二张图片的端口一直  
    server 192.168.177.129:8081;  
    server 192.168.177.129:8082;  
}
```

重启nginx： ./nginx -s reload

为了能够区分是访问哪个服务器的网站，可以在首页标题加上标记以便区分

进入tomcat1目录的webapps的ROOT目录的  
index.html(/usr/local/tomcat/tomcat1/webapps/ROOT)

把title后加一个端口号

后两个也是一样的操作

## 2、配置负载均衡

修改Nginx配置文件/usr/local/nginx/conf/nginx.conf)

```
upstream tomcat-travel{  
    #哪个服务器性能高，则权重就大；按照比例进行给值  
    server 192.168.177.129:8080 weight=2;  
    server 192.168.177.129:8081;  
    server 192.168.177.129:8082;  
}
```

重启nginx: ./nginx -s reload