日志

Jdk安装

export JAVA\_HOME=/usr/java/jdk1.7.0\_07

export PATH=$JAVA\_HOME/bin:$PATH

export

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

//配置文件生效

source /etc/profile

netstat -ano|grep 80

## Nginx的安装

模块依赖性Nginx需要依赖下面3个包

1. gzip 模块需要 zlib 库 ( 下载: http://www.zlib.net/ )

2. rewrite 模块需要 pcre 库 ( 下载: http://www.pcre.org/ )

3. ssl 功能需要 openssl 库 ( 下载: http://www.openssl.org/ )

依赖包安装顺序依次为:openssl、zlib、pcre, 然后安装Nginx包.

### 第一步：

下载安装所需包

openssl-fips-2.0.2.tar.gz

zlib-1.2.7.tar.gz

pcre-8.21.tar.gz

nginx-1.2.6.tar.gz

### 第二步:

依次安装openssl-fips-2.0.2.tar.gz, zlib-1.2.7.tar.gz, pcre-8.21.tar.gz, nginx-1.2.6.tar.gz

#### 安装openssl-fips-2.0.2.tar.gz

[root@localhost mrms]# tar -zxvf openssl-fips-2.0.2.tar.gz

[root@localhost mrms]# cd openssl-fips-2.0.2

[root@localhost openssl-fips-2.0.2]# ./config

[root@localhost openssl-fips-2.0.2]# make

[root@localhost openssl-fips-2.0.2]# make install

#### 2.安装zlib-1.2.7.tar.gz

[root@localhost mrms]# tar -zxvf zlib-1.2.7.tar.gz

[root@localhost mrms]# cd zlib-1.2.7

[root@localhost zlib-1.2.7]# ./configure

[root@localhost zlib-1.2.7]# make

[root@localhost zlib-1.2.7]# make install

#### 3.安装pcre-8.21.tar.gz

[root@localhost mrms]# tar -zxvf pcre-8.21.tar.gz

[root@localhost mrms]# cd pcre-8.21

[root@localhost pcre-8.21]# ./configure

[root@localhost pcre-8.21]# make

[root@localhost pcre-8.21]# make install

#### 4.安装 nginx-1.2.6.tar.gz

[root@localhost mrms]# tar -zxvf nginx-1.2.6.tar.gz

[root@localhost mrms]# cd nginx-1.2.6

[root@localhost nginx-1.2.6]# ./configure --with-pcre=../pcre-8.21 --with-zlib=../zlib-1.2.7 --with-openssl=../openssl-fips-2.0.2

[root@localhost nginx-1.2.6]# make

[root@localhost nginx-1.2.6]# make install

至此Nginx的安装完成!

### 第三步:检测是否安装成功

[root@localhost nginx-1.2.6]# cd /usr/local/nginx/sbin

[root@localhost sbin]# ./nginx -t

启动nginx

[root@localhost sbin]# ./nginx

查看端口

[root@localhost sbin]# netstat –ntlp

#### 1.安装 nginx-1.2.6.tar.gz

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 16k;

#gzip\_http\_version 1.0;

gzip\_comp\_level 2;

gzip\_types text/plain application/x-javascript text/css application/xml text/javascript application/x-httpd-php image/jpeg image/gif image/png;

gzip\_vary off;

gzip\_disable "MSIE [1-6]\.";

说明

第1行：开启Gzip

第2行：不压缩临界值，大于1K的才压缩，一般不用改

第3行：buffer，就是，嗯，算了不解释了，不用改

第4行：用了反向代理的话，末端通信是HTTP/1.0，有需求的应该也不用看我这科普文了；有这句的话注释了就行了，默认是HTTP/1.1

第5行：压缩级别，1-10，数字越大压缩的越好，时间也越长，看心情随便改吧

第6行：进行压缩的文件类型，缺啥补啥就行了，JavaScript有两种写法，最好都写上吧，总有人抱怨js文件没有压缩，其实多写一种格式就行了

第7行：跟Squid等缓存服务有关，on的话会在Header里增加"Vary: Accept-Encoding"，我不需要这玩意，自己对照情况看着办吧

第8行：IE6对Gzip不怎么友好，不给它Gzip了

4、:wq保存退出，重新加载Nginx

/usr/local/nginx/sbin/nginx -s reload

## root 授权用户操作制定文件夹

=========root 授权用户操作制定文件夹===========

root@whqbd:/opt# chown -R whq:whq ./tomcat8

root@whqbd:/opt# chmod 760 ./tomcat8/

## maven 导入外部jar包

C:\Users\Administrator>mvn install:install-file -DgroupId=org.json -DartifactId=

json-baidu -Dversion=1.0.0 -Dpackaging=jar -Dfile=D:/json.jar

## Git 常用命令

$ git status

$ git add test.txt

$ git commit -m '文件增加ddddddd'

$ git log

$ git log --pretty=oneline

$ git reset --hard head^ //回退到上个版本

$ git reflog //查看已提交文件版本号

$ git reset --hard 38eb94d //回退的指定版本 最新版本

$ git checkout -- test.txt //撤销对文件修改

查看被修改内容

$ git diff text.txt

查看提交日志

$ git log