# 一、Java

# 二、LinuxCentOS 7.0

## 1、CentOS 7.0默认使用的是firewall作为防火墙

### 1.1、关闭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）

### 1.2、iptables防火墙（这里iptables已经安装，下面进行配置）

vi/etc/sysconfig/iptables #编辑防火墙配置文件

# sampleconfiguration for iptables service

# you can edit thismanually or use system-config-firewall

# please do not askus to add additional ports/services to this default configuration

\*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT[0:0]

:OUTPUT ACCEPT[0:0]

-A INPUT -m state--state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -p icmp -jACCEPT

-A INPUT -i lo -jACCEPT

-A INPUT -p tcp -mstate --state NEW -m tcp --dport 22 -j ACCEPT

-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -jACCEPT

-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080-j ACCEPT

-A INPUT -j REJECT--reject-with icmp-host-prohibited

-A FORWARD -jREJECT --reject-with icmp-host-prohibited

COMMIT

:wq! #保存退出

备注：这里使用80和8080端口为例。\*\*\*部分一般添加到“-A INPUT -p tcp -m state --state NEW -m tcp--dport 22 -j ACCEPT”行的上面或者下面，切记不要添加到最后一行，否则防火墙重启后不生效。

systemctlrestart iptables.service #最后重启防火墙使配置生效

systemctlenable iptables.service #设置防火墙开机启动

## 2、解压文件

tar -cvxf XXXX

## 3、设置权限

chmod 777 aaa

## 4、设置不要图形界面的linux

修改 /etc/inittab 文件，将“id:5:initdefault:”这一行的"id:"后的数字（默认为5）改为 3即可。

# 三、Nginx

## 1、CentOS7.0安装Nginx 1.7.4 安装准备

首先由于nginx的一些模块依赖一些lib库，所以在安装nginx之前，必须先安装这些lib库，这些依赖库主要有g++、gcc、openssl-devel、pcre-devel和zlib-devel 所以执行如下命令安装

1. $   yum install gcc-c++
2. $   yum install pcre pcre-devel
3. $   yum install zlib zlib-devel
4. $   yum install openssl openssl-devel

## 2、安装Nginx

### 2.1、安装之前，最好检查一下是否已经安装有nginx

1. $   find -name nginx

### 2.2、如果系统已经安装了nginx，那么就先卸载

1. $   yum remove nginx

### 2.3、首先进入/usr/local目录

1. $   cd /usr/local

### 2.4、从官网下载最新版的nginx

1. $   wget http://nginx.org/download/nginx-1.7.4.tar.gz

### 2.5、解压nginx压缩包

1. $   tar -zxvf nginx-1.7.4.tar.gz

### 2.6、会产生一个nginx-1.7.4 目录，这时进入nginx-1.7.4目录

1. $   cd  nginx-1.7.4

### 2.7、接下来安装，使用--prefix参数指定nginx安装的目录,make、make install安装y

1. $   ./configure  $默认安装在/usr/local/nginx
2. $   make
3. $   make install

### 2.8、如果没有报错，顺利完成后，最好看一下nginx的安装目录 默认的安装目录/usr/local

1. $   whereis nginx
2. $ cd/usr/local/nginx
3. conf html logs sbin
4. cd sbin
5. Nginx
6. #查看所有的命令
7. [root@localhost conf]# cd ../conf
8. [root@localhost conf]# ls
9. fastcgi.conf koi-win scgi\_params
10. fastcgi.conf.default mime.types scgi\_params.default
11. fastcgi\_params mime.types.default uwsgi\_params
12. fastcgi\_params.default nginx.conf uwsgi\_params.default
13. koi-utf nginx.conf.default win-utf

## 3、测试nginx 的启动

1. $ [root@localhost nginx]# cd sbin/
2. [root@localhost sbin]# ls
3. nginx
4. [root@localhost sbin]# ./nginx
5. [root@localhost sbin]# ./nginx -t
6. nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
7. nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

## 启动 nginx

### 4.1、启动命令

1.[root@localhost sbin]# ./nginx

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

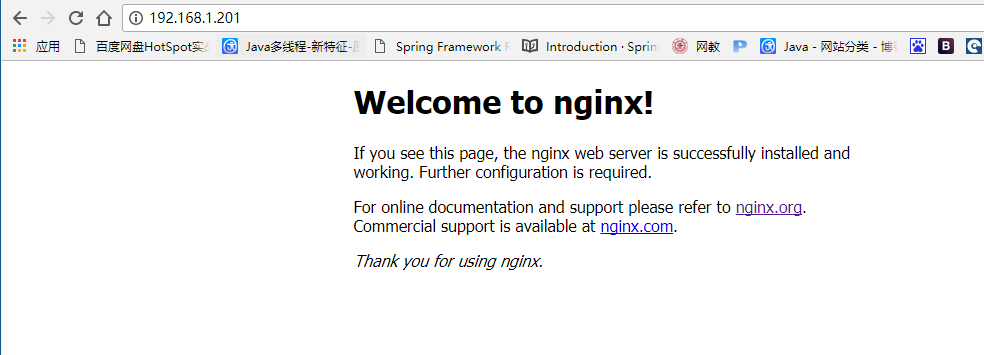
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

nginx: [emerg] still could not bind()

### 4.2、通过远程访问ip地址

防火墙开启80端口 firewall-cmd --zone=public --add-port=80/tcp --permanent

重启防火墙 firewall-cmd --reload



### 4.3、停止nginx

1.[root@localhost sbin]# ./nginx -s stop 或 ./nginx -s quit

### 4.4、重启nginx

1. [root@localhost sbin]# ./nginx -s stop 或 ./nginx -s reload

## 5、查看进程

1.[root@localhost sbin]# ps -ef | grep nginx(java|tomcat)

## 6、Nginx Http模块

### 6.1、Nginx.conf文件配置结构

http{ //协议级别

.....

server{//服务器级别

......

location /{ //请求级别

}

}

}

### 6.2、 HTTP Core 核心模块

### 6.3、 Location 配置

location匹配命令

~      #波浪线表示执行一个正则匹配，区分大小写  
~\*    #表示执行一个正则匹配，不区分大小写  
^~    #^~表示普通字符匹配，如果该选项匹配，只匹配该选项，不匹配别的选项，一般用来匹配目录  
=      #进行普通字符精确匹配  
@     #"@" 定义一个命名的 location，使用在内部定向时，例如 error\_page, try\_files

location 匹配的优先级(与location在配置文件中的顺序无关)  
= 精确匹配会第一个被处理。如果发现精确匹配，nginx停止搜索其他匹配。  
普通字符匹配，正则表达式规则和长的块规则将被优先和查询匹配，也就是说如果该项匹配还需去看有没有正则表达式匹配和更长的匹配。  
^~ 则只匹配该规则，nginx停止搜索其他匹配，否则nginx会继续处理其他location指令。  
最后匹配理带有"~"和"~\*"的指令，如果找到相应的匹配，则nginx停止搜索其他匹配；当没有正则表达式或者没有正则表达式被匹配的情况下，那么匹配程度最高的逐字匹配指令会被使用。

location 优先级官方文档

Directives with the = prefix that match the query exactly. If found, searching stops.

All remaining directives with conventional strings, longest match first. If this match used the ^~ prefix, searching stops.

Regular expressions, in order of definition in the configuration file.

If #3 yielded a match, that result is used. Else the match from #2 is used.

=前缀的指令严格匹配这个查询。如果找到，停止搜索。

所有剩下的常规字符串，最长的匹配。如果这个匹配使用^〜前缀，搜索停止。

正则表达式，在配置文件中定义的顺序。

如果第3条规则产生匹配的话，结果被使用。否则，使用第2条规则的结果。

例如

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | location  = / {    # 只匹配"/".    [ configuration A ]  }  location  / {    # 匹配任何请求，因为所有请求都是以"/"开始    # 但是更长字符匹配或者正则表达式匹配会优先匹配    [ configuration B ]  }  location ^~ /images/ {    # 匹配任何以 /images/ 开始的请求，并停止匹配 其它location    [ configuration C ]  }  location ~\* .(gif|jpg|jpeg)$ {    # 匹配以 gif, jpg, or jpeg结尾的请求.    # 但是所有 /images/ 目录的请求将由 [Configuration C]处理.    [ configuration D ]  } |

请求URI例子:

/ -> 符合configuration A

/documents/document.html -> 符合configuration B

/images/1.gif -> 符合configuration C

/documents/1.jpg ->符合 configuration D

@location 例子  
error\_page 404 = @fetch;

location @fetch(  
proxy\_pass http://fetch;  
)

# LinuxCentOS 7.0安装jdk

## 1、上传jdk文件

将jdk-8u144-linux-x64.tar.gz上传至/usr/java 目录下

## 2、执行解压命令

输入 tar -zxvf jdk-8u144-linux-x64.tar.gz

## 执行vim ~/.bashrc

## 4、输入环境变量配置信息

export JAVA\_HOME=/usr/java/jdk1.8.0\_144

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

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

## 保存并退出 WQ

## 运行bashrc文件

输入source ~/.bashrc文件

## 7、验证是否配置成功 输入javac java

# Maven

## 1. maven 插件

### 1.1 eclipse 用maven添加Tomcat插件

<build>

<!-- 配置插件 -->

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<configuration>

<port>8080</port>

<path>/</path>

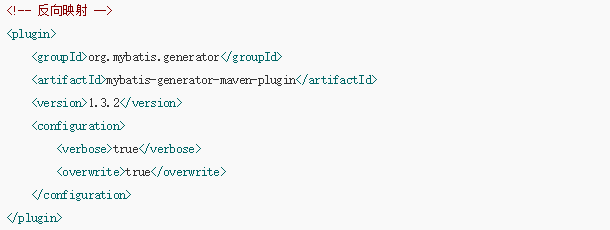
</configuration>

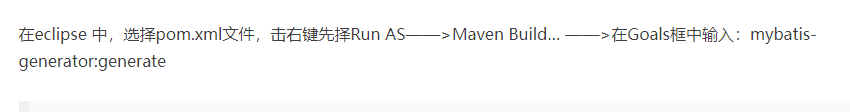
</plugin>

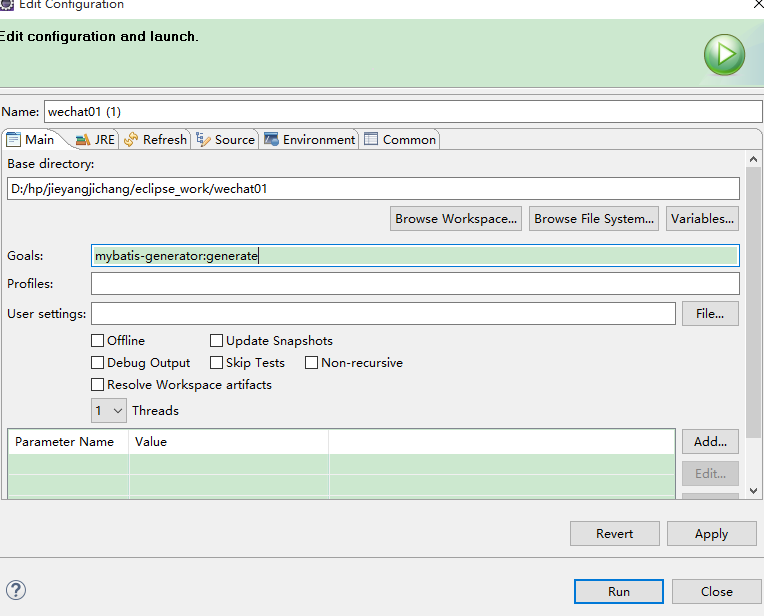
</plugins>

</build>

### 1.2 使用maven插件反向映射generatorConfig.xml生成代码







# 连接池配置

## 1.配置\_DruidDataSource参考配置

|  |
| --- |
| <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">  <!-- 数据库驱动 -->  <property name="driverClassName" value="${jdbc.driverClassName}" />  <!-- 基本属性 url、user、password -->  <property name="url" value="${jdbc\_url}" />  <property name="username" value="${jdbc\_user}" />  <property name="password" value="${jdbc\_password}" />  <!-- 配置初始化大小、最小、最大 -->  <property name="initialSize" value="1" />  <property name="minIdle" value="1" />  <property name="maxActive" value="20" />  <!-- 配置获取连接等待超时的时间 -->  <property name="maxWait" value="60000" />  <!-- 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->  <property name="timeBetweenEvictionRunsMillis" value="60000" />  <!-- 配置一个连接在池中最小生存的时间，单位是毫秒 -->  <property name="minEvictableIdleTimeMillis" value="300000" />  <property name="validationQuery" value="SELECT 'x'" />  <property name="testWhileIdle" value="true" />  <property name="testOnBorrow" value="false" />  <property name="testOnReturn" value="false" />  <!-- 打开PSCache，并且指定每个连接上PSCache的大小 -->  <property name="poolPreparedStatements" value="true" />  <property name="maxPoolPreparedStatementPerConnectionSize" value="20" />  <!-- 配置监控统计拦截的filters -->  <property name="filters" value="stat" />  </bean> |

通常来说，只需要修改initialSize、minIdle、maxActive。

如果用Oracle，则把poolPreparedStatements配置为true，mysql可以配置为false。分库分表较多的数据库，建议配置为false。

druid常见问题汇总：

<https://github.com/alibaba/druid/wiki/%E5%B8%B8%E8%A7%81%E9%97%AE%E9%A2%98>

# 七、配置ftp服务

## 1.安装vsftpd组件

安装完后，有/etc/vsftpd/vsftpd.conf 文件，是vsftp的配置文件。

[root@bogon ~]# yum -y install vsftpd

## 2.添加一个ftp用户

此用户就是用来登录ftp服务器用的。

[root@bogon ~]# useradd ftpuser

这样一个用户建完，可以用这个登录，记得用普通登录不要用匿名了。登录后默认的路径为 /home/ftpuser.

## 3.给ftp用户添加密码。

[root@bogon ~]# passwd ftpuser

输入两次密码后修改密码。

## 4.修改配置文件

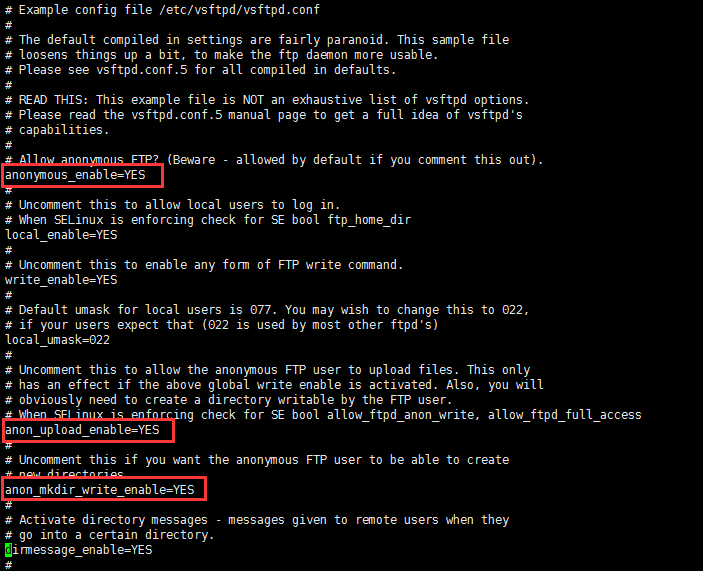
vi /etc/vsftpd/vsftpd.conf

    保证下面3项为YES

    anonymous\_enable=YES

    anon\_upload\_enable=YES

    anon\_mkdir\_write\_enable=YES

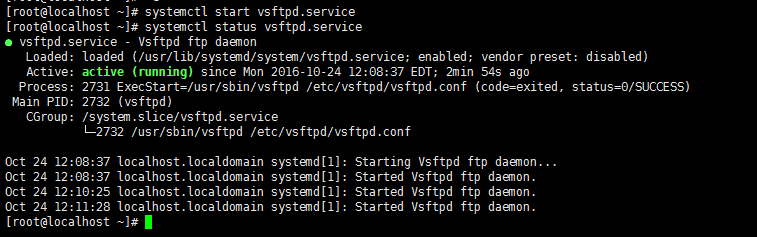


## 设置vsftpd开机启动

systemctl enable vsftpd.service

1. 启动并查看vsftpd服务状态，systemctl启动服务成功不会有任何提示，绿色的active表示服务正在运行

systemctl start vsftpd.service  
systemctl status vsftpd.service



## 6.文件读写

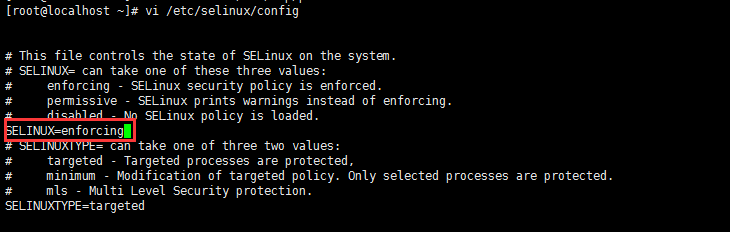
设置文件夹权限，将pub文件夹的权限设置为777

chmod 777 -R /var/ftp/pub

## 6.关闭selinux服务

vi /etc/selinux/config

  将SELINUX=enforcing改为：SELINUX=disabled



## 6.重启ftp服务

service vsftpd restart

## 7.设置开机启动vsftpd ftp服务

[root@bogon ~]# chkconfig vsftpd on