# Nginx笔记

## 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)

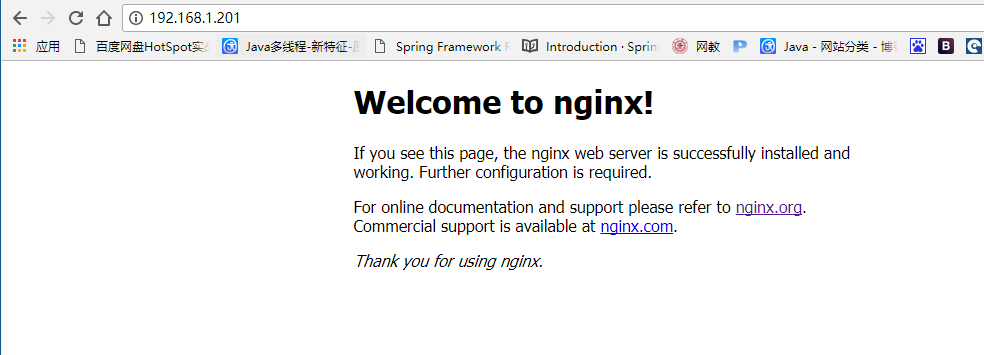
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)

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;  
)

7、静态代理