仅仅为了测试，中间可能包含了许多的非安全配置

目录准备

**新增用户nginxer，并设置密码**

[root@master src]# useradd nginxer

[root@master src]# passwd nginxer

**将nginxer添加sudo**

[root@master src]# vim /etc/sudoers

**切换到nginxer用户并进入到nginxer的家目录**

[root@master src]# su nginxer

[nginxer@master src]$ cd

**创建src目录和app目录**

[nginxer@master ~]$ mkdir src

[nginxer@master ~]$ mkdir app

文件准备

**进入到src目录，下载nginx源码包**

[nginxer@master src]$ wget <http://nginx.org/download/nginx-1.12.1.tar.gz>

**将nginx源码压缩文件解压到app目录**

[nginxer@master src]$ tar -xzvf nginx-1.12.1.tar.gz -C ../app

预编译、编译和安装

注意：nginx的url rewrite需要pcre库，因此如果希望将重写模块编译进nginx则需要先安装pcre和pcre-devel,或者在编译nginx的时候执行pcre的源码包。这里采用yum 安装pcre 和 pcre-devel

[nginxer@master src]$ sudo yum install -y pcre

[nginxer@master src]$ sudo yum install -y pcre-devel

如果使用gzip模块的话需要先安装zlib和zlib-devel，与pcre一样这里采用yum安装：

[nginxer@master nginx-1.12.1]$ sudo yum install -y zlib

[nginxer@master nginx-1.12.1]$ sudo yum install -y zlib-devel.x86\_64

如果不编译rewrite和gzip模块则不需要安装以上两项，但在预编译的时候需要指定不编译这些模块，在预编译的时候会有提示。

**预编译nginx，指定安装目录为/home/nginxer/app/nginx-1.12.1/bin**

[nginxer@master nginx-1.12.1]$ sudo ./configure --prefix=/home/nginxer/app/nginx-1.12.1/bin

Configuration summary

+ using system PCRE library

+ OpenSSL library is not used

+ using system zlib library

nginx path prefix: "/home/nginxer/app/nginx-1.12.1/bin"

nginx binary file: "/home/nginxer/app/nginx-1.12.1/bin/sbin/nginx"

nginx modules path: "/home/nginxer/app/nginx-1.12.1/bin/modules"

nginx configuration prefix: "/home/nginxer/app/nginx-1.12.1/bin/conf"

nginx configuration file: "/home/nginxer/app/nginx-1.12.1/bin/conf/nginx.conf"

nginx pid file: "/home/nginxer/app/nginx-1.12.1/bin/logs/nginx.pid"

nginx error log file: "/home/nginxer/app/nginx-1.12.1/bin/logs/error.log"

nginx http access log file: "/home/nginxer/app/nginx-1.12.1/bin/logs/access.log"

nginx http client request body temporary files: "client\_body\_temp"

nginx http proxy temporary files: "proxy\_temp"

nginx http fastcgi temporary files: "fastcgi\_temp"

nginx http uwsgi temporary files: "uwsgi\_temp"

nginx http scgi temporary files: "scgi\_temp"

**编译**

[nginxer@master nginx-1.12.1]$ sudo make

如果没有Error提示的话则代表编译成功

**安装**

[nginxer@master nginx-1.12.1]$ sudo make install

安装成功后就会在/home/nginxer/app/nginx-1.12.1/bin目录中生成如下文件：

[nginxer@master nginx-1.12.1]$ ls ./bin/

conf html logs sbin

配置和启动服务

**启动nginx服务**

[nginxer@master bin]$ sudo ./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)

此处出错说明80端口被占用，我们需要修改配置文件更改任意未被占用的端口

[nginxer@master bin]$ sudo vim ./conf/nginx.conf #修改端口为88端口

**查看服务**

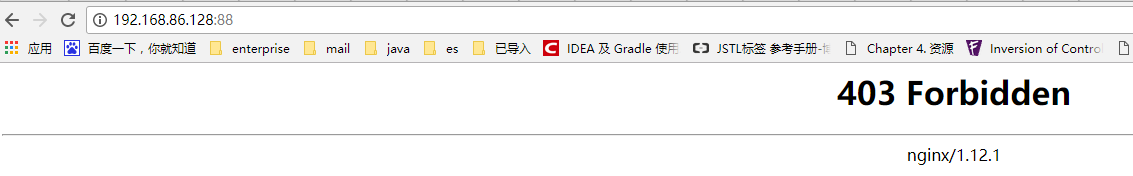
root 82490 0.0 0.1 20008 1304 ? Ss 19:27 0:00 nginx: master process ./sbin/nginx

root 82547 0.0 0.1 20460 1612 ? S 19:38 0:00 nginx: worker process

关闭防火墙服务一遍外部能够访问88端口（为了安全也可单独配置入站出站规则，这里为了简便直接关闭防火墙）

[nginxer@master bin]$ sudo service iptables stop

在外部浏览器访问



之所以说forbidden，是因为默认nginx的运行用户为nobody，它没有权限读取html目录

为了测试简单的配置为：修改配置文件，将nginx服务运行用户该为root

[nginxer@master bin]$ sudo vim ./conf/nginx.conf

修改了配置文件后需要重启nginx或者使用信号

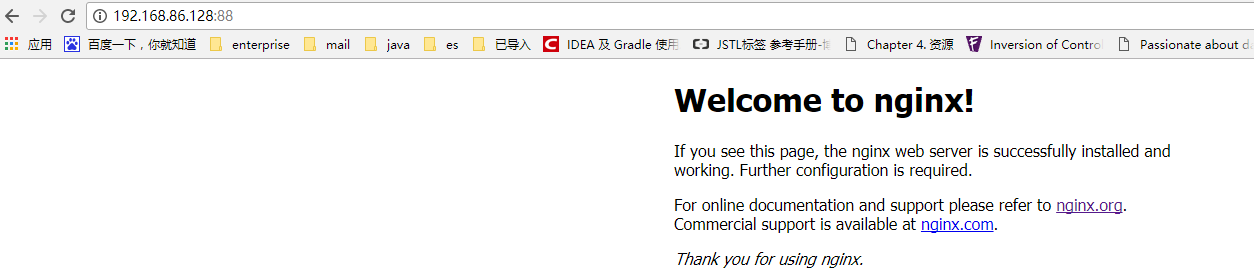
重启：[nginxer@master bin]$ sudo ./sbin/nginx -s reload

或者

发送信号重读配置文件：

[nginxer@master bin]$ sudo kill -HUP `cat ./logs/nginx.pid`

在外部浏览器再次访问：



安装配置完成