

# Nginx简介

Nginx (engine x) 是一个高性能的HTTP和反向代理服务，也是一个IMAP/POP3/SMTP服务。Nginx是由伊戈尔·赛索耶夫为俄罗斯访问量第二的Rambler.ru站点(俄文:Рамблер)开发的，第一个公开版本0.1.0发布于2004年10月4日。其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。2011年6月1日，nginx 1.0.4发布。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件(IMAP/POP3)代理服务器，并在一个BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有:百度、京东、新浪、网易、腾讯、淘宝等。

## 实验环境：

系统版本：centos7x3.10.0-514.el7.x86\_64

Nginx版本：nginx1.14.0

关闭防火墙并禁止开机自启

```
systemctl stop firewalld.service
```

```
systemctl disable firewalld
```

关闭selinux

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

修改主机名

```
vi /etc/hostname
```

```
nginx.wangfeiyu.com
```

域名绑定IP

```
vi /etc/hosts
```

```
[root@localhost ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.152.177 nginx.wangfeiyu.com
[root@localhost ~]#
[root@localhost ~]#
```

@51CTO博客

重启 reboot

## 安装nginx服务

## 1、安装nginx依赖环境包

```
yum install gcc-c++ pcre pcre-devel zlib zlib-devel openssl openssl-devel
```

## 2、官网下载nginx1.14.0压缩包

```
wget https://nginx.org/download/nginx-1.14.0.tar.gz
```

## 3、解压nginx

```
tar xzf nginx-1.14.0.tar.gz
```

## 4、进入解压目录

```
cd nginx-1.14.0
```

## 5、编译nginx

### 1) 默认编译方式

```
./configure
```

### 2) 自定义编译选项

```
./configure \  
--user=nginx \  
--group=nginx \  
--prefix=/usr/local/nginx \  
--conf-path=/usr/local/nginx/conf/nginx.conf \  
--pid-path=/usr/local/nginx/conf/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 \  
--with-http_ssl_module
```

注：以上为默认编译方式和具体指定的编译方式，任选以上这两种之一即可。--with-http\_ssl\_module这个选项是https的重要模块必须安装。

### 3) 本文中使用的编译安装方式

```
./configure
--prefix=/usr/local/nginx
--with-http_stub_status_module
--with-http_ssl_module
```

```
Configuration summary
+ using system PCRE library
+ using system OpenSSL library
+ using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/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"

[root@nginx nginx-1.14.0]#
```

@51CTO博客

注：以上--with-http\_ssl\_module这个模块是https的关键，必须安装！

## 6、安装nginx

```
make && make install
```

## 7、启动nginx

方式一

### 1) 启动nginx

```
/usr/local/nginx/sbin/nginx
```

```
[root@localhost ~]# /usr/local/nginx/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()
[root@localhost ~]#
```

@51CTO博客

### 2) 关闭nginx

```
/usr/local/nginx/sbin/nginx -s stop
```

### 3) 重启nginx

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

注：如果嫌以上方式太麻烦，可以做软连接 `ln -s /usr/local/nginx/sbin/nginx /usr/bin/nginx` 或者在全局环境变量里增加nginx环境变量，然后直接使用nginx即可！

## 方式二

### 1) 编辑nginx服务启动文件

`vi /etc/init.d/nginx`

```
#!/bin/bash
#chkconfig: - 85 15
PATH=/usr/local/nginx
NAME=nginx
DAEMON=$PATH/sbin/$NAME
CONFIGFILE=$PATH/conf/$NAME.conf
PIDFILE=$PATH/logs/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
set -e
[ -x "$DAEMON" ] || exit 0
do_start() {
$DAEMON -c $CONFIGFILE || echo -e "\033[32m nginx already running \033[0m"
}
do_stop() {
$DAEMON -s stop || echo -e "\033[31m nginx not running \033[0m"
}
do_reload() {
$DAEMON -s reload || echo -e "\033[31m nginx can't reload \033[0m"
}
case "$1" in
start)
echo -e "\033[32m $NAME running \033[0m"
do_start
;;
stop)
echo -e "\033[31m $NAME stoping \033[0m"
```

```
do_stop
;;
reload|graceful)
echo -e "\033[32m $NAME configuration...\033[0m"
do_reload
;;
restart)
echo -e "\033[32m Restarting : $NAME \033[0m"
do_stop
do_start
;;
*)
echo "Usage: $SCRIPTNAME {start|stop|reload|restart}" >&2
exit 3
;;
esac
exit 0
```

注：切记编辑完启动脚本以后一定要给予执行权限，不然启动无效！

## 2) 设置启动文件执行权限

```
chmod +x /etc/init.d/nginx
```

## 3) 启动nginx

//设置开机自启

```
chkconfig nginx on
```

//启动nginx

```
/etc/init.d/nginx start
```

//重启nginx

```
/etc/init.d/nginx restart
```

//查看nginx服务启动状态

```
chkconfig --list
```

//查看nginx服务是否开启

```
netstat -antupl | grep nginx
```

## 8、开机启动nginx

### 1) 编辑开机启动文件

```
vi /etc/rc.local
```

添加一行/usr/local/nginx/sbin/nginx

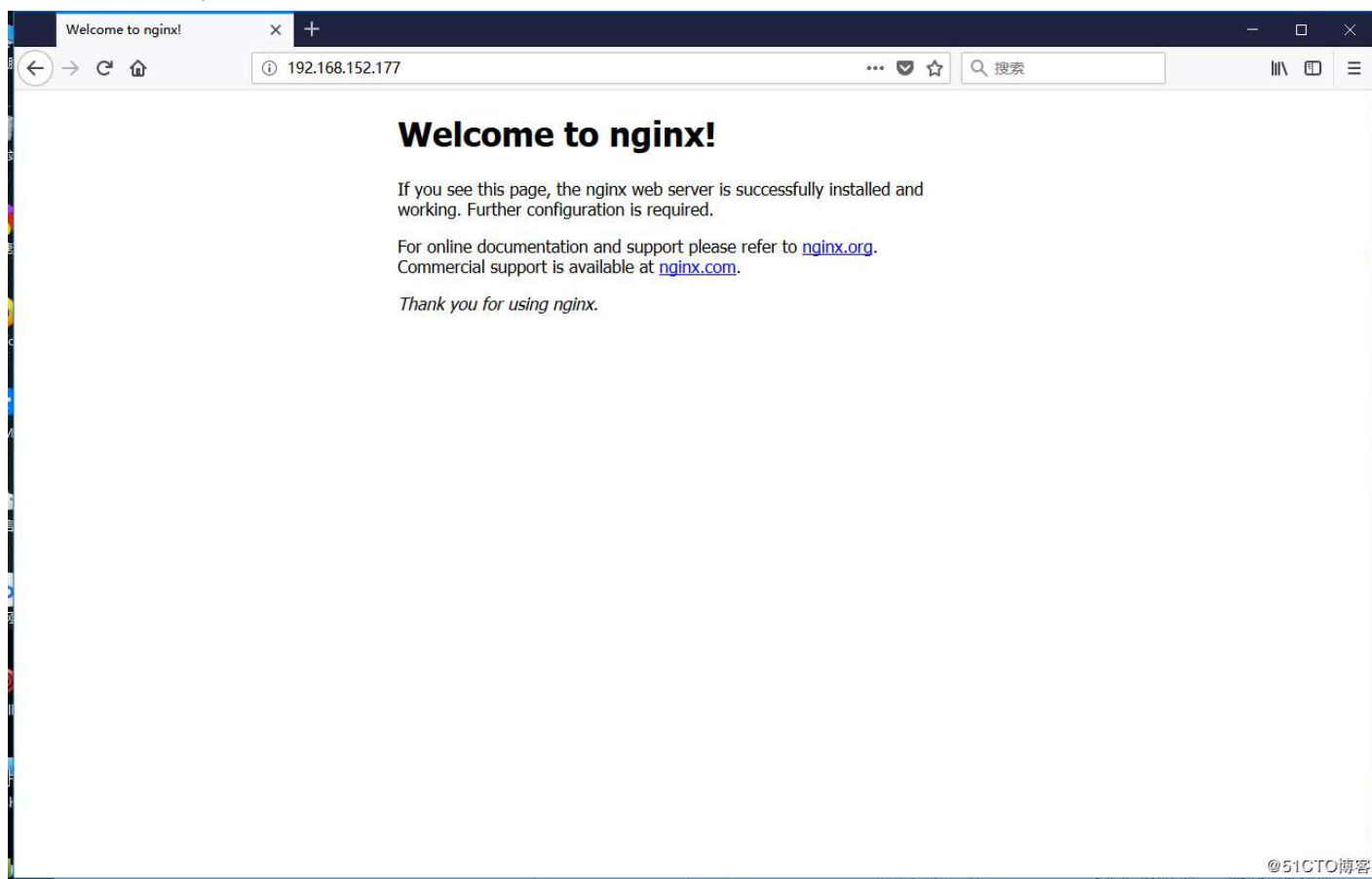
### 2) 设置启动文件权限

```
chmod 755 /etc/rc.local
```

注：如果使用方式二脚本启动服务，那么以上启动方式可以省略！

## 9、访问测试

访问地址：<http://192.168.152.77>



## 升级nginx为https条件

## 1、查看nginx是否支持ssl

/usr/local/nginx/sbin/nginx -V

```
[root@nginx ~]# nginx -V
nginx version: nginx/1.14.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
built with OpenSSL 1.0.2k-fips 26 Jan 2017
TLS SNI support enabled
configure arguments: --prefix=/usr/local/nginx --with-http_stub_status_module --with-http_@51CTO博客
[root@nginx ~]#
```

注：查看 configure arguments 信息中是否包含 -with-http\_ssl\_module 字样，如果没有则需要重新编译。找到之前安装 Nginx 时的编译目录，配置ssl模块，因为这次是升级nginx，所以不需要执行 make install，执行命令如下：

./configure --with-http\_ssl\_module

make

## 2、查看openssl配置文件

vi /etc/pki/tls/openssl.cnf

```
36 [ ca ]
37 default_ca = CA_default # The default ca section
38
39 #####
40 [ CA_default ]
41
42 dir = /etc/pki/CA # Where everything is kept
43 certs = $dir/certs # Where the issued certs are kept
44 crl_dir = $dir/crl # Where the issued crl are kept
45 database = $dir/index.txt # database index file.
46 #unique_subject = no # Set to 'no' to allow creation of
47 # several certificates with same subject.
48 new_certs_dir = $dir/newcerts # default place for new certs.
49
50 certificate = $dir/cacert.pem # The CA certificate
51 serial = $dir/serial # The current serial number
52 crlnumber = $dir/crlnumber # the current crl number
53 # must be commented out to leave a V1 CRL
54 crl = $dir/crl.pem # The current CRL
55 private_key = $dir/private/akey.pem # The private key
56 RANDFILE = $dir/private/.rand # private random number file
57
58 x509_extensions = usr_cert # The extensions to add to the cert
59
```

注：以上截图默认就是这样的重要参数配置路径，如果你要配置修改路径，那么切记在后边签证书等等的操作都要按照这个配置路径去创建，不然当认证的时候会找不到证书！

## 3、创建生成证书需要的文件

### 1) 创建证书索引数据库文件

touch /etc/pki/CA/index.txt

### 2) 指定第一个颁发证书的序列号

echo 01 > /etc/pki/CA/serial

注：必须是两位十六进制数，99之后是9A！

## 4、CA自签证书

### 1) 生成CA私钥

```
cd /etc/pki/CA
```

```
umask 066
```

```
openssl genrsa -out /etc/pki/CA/private/cakey.pem 2048
```

```
[root@nginx ~]# cd /etc/pki/CA/
[root@nginx CA]#
[root@nginx CA]# ls
certs  crl  index.txt  newcerts  private  serial
[root@nginx CA]#
[root@nginx CA]# umask 066
[root@nginx CA]#
[root@nginx CA]# openssl genrsa -out /etc/pki/CA/private/cakey.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
[root@nginx CA]#
```

@51CTO博客

注：进入到/etc/pki/CA/目录下执行这两条命令！

### 2) 生成CA自签名证书

```
openssl req -new -x509 -key /etc/pki/CA/private/cakey.pem -days 7300 -out /etc/pki/CA/cacert.pem
```

```
[root@localhost ~]# openssl req -new -x509 -key /etc/pki/CA/private/cakey.pem -days 7300 -out /etc/pki/CA/cacert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:BJ
Locality Name (eg. city) [Default City]:BJ
Organization Name (eg. company) [Default Company Ltd]:WXYC
Organizational Unit Name (eg. section) []:JSB
Common Name (eg. your name or your server's hostname) []:wangfeiyu.com
Email Address []:wangfeiyu@xingyoucai.com
[root@localhost ~]#
```

@51CTO博客

注释：

-new: 生成新证书签署请求

-x509: 专用于 CA 生成自签证书

-key: 生成请求时用到的私钥文件

-days n: 证书的有效期限

-out: 证书的保存路径

提示输入国家，省，市，公司名称，部门名称，CA主机名（颁发者名称）

### 4) 查看生成的自签名证书

//linux系统下查看

```
openssl x509 -in /etc/pki/CA/cacert.pem -noout -text
```



//windows系统下查看

需要更改上述文件名后缀为.cer即可查看

## 5、颁发证书

### 1) 在当前创建/root/key/目录

mkdir key

### 2) 生成web服务器私钥

cd key/

umask 066

openssl genrsa -out key/service.key 2048

```
[root@nginx ~]# cd key/
[root@nginx key]#
[root@nginx key]# umask 066
[root@nginx key]#
[root@nginx key]# openssl genrsa -out service.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
[root@nginx key]#
[root@nginx key]#
```

@51CTO博客

### 3) 生成CA证书申请文件

openssl req -new -key service.key -out service.csr

```
[root@nginx key]# openssl req -new -key service.key -out service.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value.
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:BJ
Locality Name (eg. city) [Default City]:BJ
Organization Name (eg. company) [Default Company Ltd]:WXYC
Organizational Unit Name (eg. section) []:JSB
Common Name (eg. your name or your server's hostname) []:wangfeiyu.com
Email Address []:wangfeiyu@xingyoucai.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@nginx key]#
[root@nginx key]#
```

为了方便连接证书（除特殊情况）不建议设置密码

@51CTO博客

注：同样提示输入国家、省、市、公司等信息。切记：国家，省，公司名称三项必须和CA一致。主机名称必须和网站域名相同，如www.centos73.com。或者使用泛域名，即\*.centos73.com，匹配所有。

### 4) 将证书文件移动到CA服务器/etc/pki/CA/csr目录下

```
mv service.csr /etc/pki/CA/csr/
```

```
[root@nginx key]# mkdir /etc/pki/CA/csr
[root@nginx key]# ls /etc/pki/CA/
cacert.pem  certs  crl  csr  index.txt  newcerts  private  serial
[root@nginx key]#
[root@nginx key]# mv service.csr /etc/pki/CA/csr/
[root@nginx key]#
```

@51CTO博客

注：默认好像是没有这个csr目录，那么就手动创建一个！

## 5) CA签署证书，并将证书颁发给请求者

```
openssl ca -in /etc/pki/CA/crl/service.csr -out /etc/pki/CA/certs/service.crt -days 365
```

```
[root@nginx key]# openssl ca -in /etc/pki/CA/csr/service.csr -out /etc/pki/CA/certs/service.crt -days 365
Using configuration from /etc/pki/tls/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: Oct 10 06:43:29 2018 GMT
    Not After : Oct 10 06:43:29 2019 GMT
  Subject:
    countryName           = CN
    stateOrProvinceName   = BJ
    organizationName      = WXYC
    organizationalUnitName = JSB
    commonName            = wangfeiyu.com
    emailAddress          = wangfeiyu@xingyoucai.com
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
      BA:A1:5A:53:55:B6:35:EE:2A:46:28:86:98:E9:D9:76:58:F1:4E:3C
    X509v3 Authority Key Identifier:
      keyid:63:4E:D9:88:EC:4F:CB:7E:5D:16:3F:D8:E4:B2:F6:F6:7C:83:47:DE

Certificate is to be certified until Oct 10 06:43:29 2019 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
[root@nginx key]#
```

@51CTO博客

## 6) 查看证书中的信息

//查看自签证书

openssl x509 -in 绝对路径 -noout -text | issuer | subject | serial | dates

```
[root@nginx ~]# openssl x509 -in /etc/pki/CA/certs/service.crt -noout -text
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=CN, ST=BJ, L=BJ, O=WXYC, OU=JSB, CN=wangfeiyu.com/emailAddress=wangfeiyu@xingyoucai.com
    Validity
      Not Before: Oct 10 06:43:29 2018 GMT
      Not After : Oct 10 06:43:29 2019 GMT
    Subject: C=CN, ST=BJ, O=WXYC, OU=JSB, CN=wangfeiyu.com/emailAddress=wangfeiyu@xingyoucai.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:ad:ab:17:ce:f2:e3:e7:c7:83:b0:15:93:ba:3f:
        1d:71:fd:5f:41:b4:23:b4:11:63:9c:c6:f0:56:67:
        9c:22:28:53:d1:75:40:e7:3f:fb:ef:4a:1c:16:0e:
        ce:9f:fc:35:b0:79:35:a3:b3:71:bf:d4:92:68:ef:
        04:e4:b5:2f:28:4e:67:a0:42:be:b6:46:a0:58:c0:
        77:1d:e0:e0:22:f8:c4:66:fe:24:f0:6a:f3:c5:5e:
        85:fa:db:76:7d:eb:a1:84:86:8e:f8:8b:c3:9f:3e:
        70:61:49:5a:81:2c:ea:7a:c1:2d:e3:27:1d:56:3f:
        09:16:12:b2:2f:de:03:bb:e4:60:b9:52:56:2e:b8:
        68:92:12:c7:8d:41:67:d8:6d:57:58:b0:67:c1:3a:
        db:20:b6:e6:80:bc:9b:a9:da:0c:6f:6f:bf:f5:69:
        48:cb:92:30:45:fd:4f:e6:70:47:85:61:02:53:94:
        0b:fa:4b:ef:d0:5f:de:a5:60:97:5c:22:82:e3:85:
        79:c5:fa:e0:3b:1d:33:35:32:15:43:19:91:46:e7:
        a9:7a:05:8d:4d:da:c1:b4:17:08:a7:90:0b:bf:79:
        ad:a4:6f:ee:0d:01:a4:09:c8:24:8c:17:a6:73:9f:
        cd:4f:93:b4:84:9b:a0:a0:ea:c2:5a:c7:9e:dc:fe:
        3b:a7
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      Netscape Comment:
        OpenSSL Generated Certificate
      X509v3 Subject Key Identifier:
        BA:A1:5A:53:55:B6:35:EE:2A:46:28:86:98:E9:D9:76:58:F1:4E:3C
      X509v3 Authority Key Identifier:
        keyid:63:4E:D9:88:EC:4F:CB:7E:5D:16:3F:D8:E4:B2:F6:F6:7C:83:47:DE

    Signature Algorithm: sha256WithRSAEncryption
    01:47:f4:a5:0e:42:50:fd:5a:42:b8:92:38:da:e6:b8:08:ee:
    97:0c:97:eb:13:49:3d:36:7f:2c:39:57:a6:93:d5:6a:03:01:
    7f:cb:cd:12:a6:52:ac:ff:95:a6:d9:ad:a9:47:e6:c7:ea:af:
    97:7b:8f:e3:90:10:eb:a5:5e:a1:a3:cb:54:53:f6:4c:a6:52:
    29:70:5a:98:31:98:fa:06:f9:98:c3:4d:44:a0:a9:25:ac:67:
    bc:fb:6b:ec:1f:e7:5a:88:2e:16:7d:f6:ef:56:28:fd:90:ab:
    bd:fd:f4:fb:9d:3f:d8:ac:61:c1:2c:54:76:fa:3e:fd:5a:87:
    66:0b:e3:f8:1c:46:69:76:f7:8a:87:8a:0d:65:39:1c:ef:b8:
    0b:79:a4:e6:b2:b2:15:e9:9a:bc:e6:8e:16:3a:53:53:0e:85:
    e3:8c:46:cc:2b:77:0b:99:d4:d4:44:93:46:19:fd:f4:9d:99:
    19:0c:99:90:c2:ce:a6:35:8e:cc:e1:7f:ac:6b:54:2a:dd:56:
    2d:5d:14:f4:be:79:20:3f:ed:0a:ff:d2:46:01:ea:18:7d:08:
    04:30:b9:08:7d:a8:24:08:93:e8:2d:f2:38:42:cf:ba:c3:27:
    80:b6:e9:ce:28:e5:e6:d9:46:b6:66:93:e1:26:b4:4a:a3:74:
    01:ef:b0:69
```

@51CTO博客

```
cd:4f:93:b4:84:9b:a0:a0:ea:c2:5a:c7:9e:dc:fe:
3b:a7
Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  Netscape Comment:
    OpenSSL Generated Certificate
  X509v3 Subject Key Identifier:
    BA:A1:5A:53:55:B6:35:EE:2A:46:28:86:98:E9:D9:76:58:F1:4E:3C
  X509v3 Authority Key Identifier:
    keyid:63:4E:D9:88:EC:4F:CB:7E:5D:16:3F:D8:E4:B2:F6:F6:7C:83:47:DE

Signature Algorithm: sha256WithRSAEncryption
01:47:f4:a5:0e:42:50:fd:5a:42:b8:92:38:da:e6:b8:08:ee:
97:0c:97:eb:13:49:3d:36:7f:2c:39:57:a6:93:d5:6a:03:01:
7f:cb:cd:12:a6:52:ac:ff:95:a6:d9:ad:a9:47:e6:c7:ea:af:
97:7b:8f:e3:90:10:eb:a5:5e:a1:a3:cb:54:53:f6:4c:a6:52:
29:70:5a:98:31:98:fa:06:f9:98:c3:4d:44:a0:a9:25:ac:67:
bc:fb:6b:ec:1f:e7:5a:88:2e:16:7d:f6:ef:56:28:fd:90:ab:
bd:fd:f4:fb:9d:3f:d8:ac:61:c1:2c:54:76:fa:3e:fd:5a:87:
66:0b:e3:f8:1c:46:69:76:f7:8a:87:8a:0d:65:39:1c:ef:b8:
0b:79:a4:e6:b2:b2:15:e9:9a:bc:e6:8e:16:3a:53:53:0e:85:
e3:8c:46:cc:2b:77:0b:99:d4:d4:44:93:46:19:fd:f4:9d:99:
19:0c:99:90:c2:ce:a6:35:8e:cc:e1:7f:ac:6b:54:2a:dd:56:
2d:5d:14:f4:be:79:20:3f:ed:0a:ff:d2:46:01:ea:18:7d:08:
04:30:b9:08:7d:a8:24:08:93:e8:2d:f2:38:42:cf:ba:c3:27:
80:b6:e9:ce:28:e5:e6:d9:46:b6:66:93:e1:26:b4:4a:a3:74:
01:ef:b0:69
```

@51CTO博客

//查看颁发证书的序列号

cat /etc/pki/CA/serial

```
[root@nginx ~]# cat /etc/pki/CA/serial
02
```

@51CTO博客

//查看指定编号的证书状态

openssl ca -status 1

```
[root@nginx CA]# openssl ca -status 1
Using configuration from /etc/pki/tls/openssl.cnf
01=Valid (V)
[root@nginx CA]#
[root@nginx CA]#
```

@51CTO博客

注：这个编号是颁发的第几个证书，当前就一个所以是1！

//查看证书详细信息

cat /etc/pki/CA/index.txt

```
[root@nginx CA]# cat /etc/pki/CA/index.txt
V          191010024933Z      01      unknown /C=CN/ST=BJ/O=WXYC/OU=JSB/CN=nihao.com/emailAddress=wangfeiyu@xingyou
cai.com
[root@nginx CA]#
[root@nginx CA]#
```

@51CTO博客

注：开头V表示当前证书的状态正常！

//查看subjects信息

```
[root@nginx CA]# cat /etc/pki/CA/index.txt.attr
unique_subject = yes
[root@nginx CA]#
[root@nginx CA]#
```

@51CTO博客

注：yes表示subjects信息必须是唯一的，不能重复申请！

## 6、修改nginx配置文件

vi /usr/local/nginx/conf/nginx.conf

```
95
96
97     server {
98         listen      443 ssl;
99         server_name  nginx.nihao.com;
100
101         ssl_certificate      /etc/pki/CA/cacert.pem;
102         ssl_certificate_key  /etc/pki/CA/private/cakey.pem;
103
104         ssl_session_cache    shared:SSL:1m;
105         ssl_session_timeout  5m;
106
107         ssl_ciphers  HIGH:!aNULL:!MD5;
108         ssl_prefer_server_ciphers  on;
109
110         location / {
111             root    html;
112             index   index.html index.htm;
113         }
114     }
115
116 }
```

@51CTO博客

注：这里有一个坑就是默认的HTTPS SERVER这行必须删除，要不然一直报错！

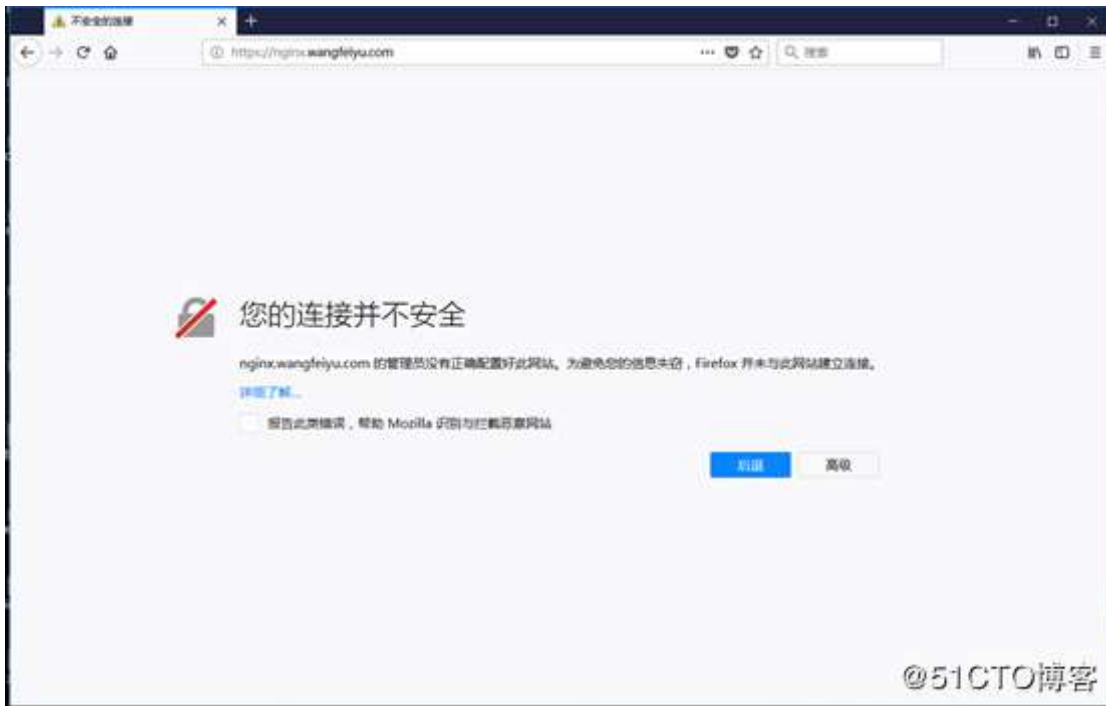
## 7、重启nginx服务

/etc/init.d/nginx restart

## 8、测试（建议使用Firefox浏览器测试）

### 访问网页测试

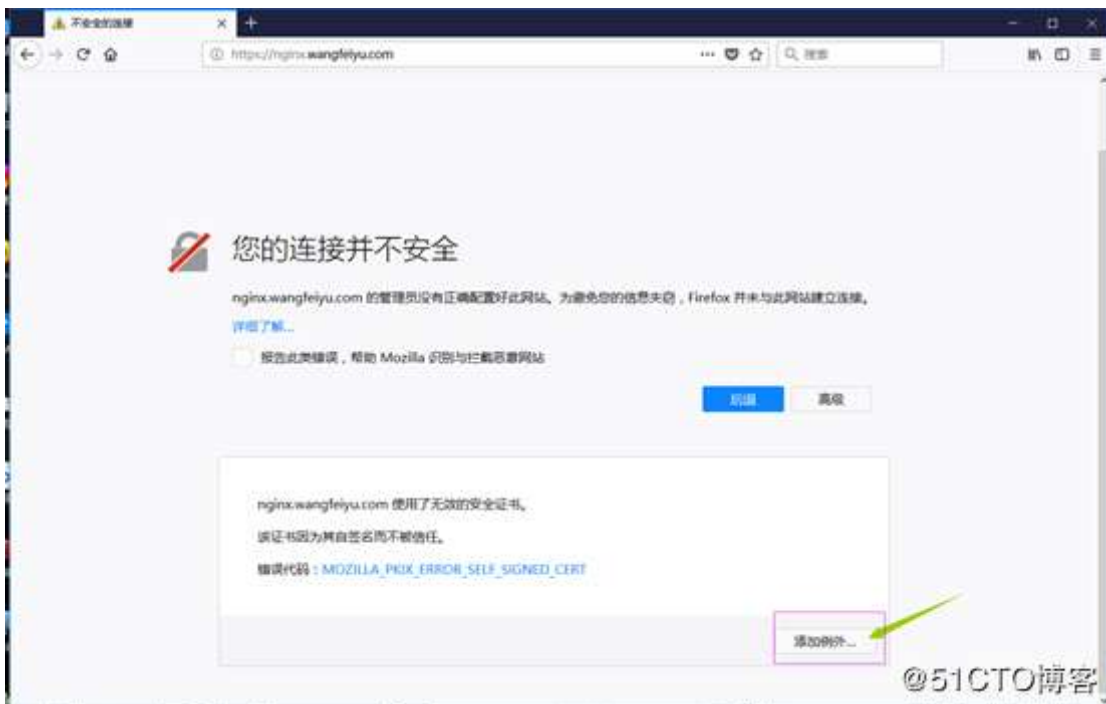
1) 域名访问地址：<https://nginx.wangfeiyu.com/>



注：以上截图访问方式使用的是https加密访问但是需要我们将证书导入浏览器才行！

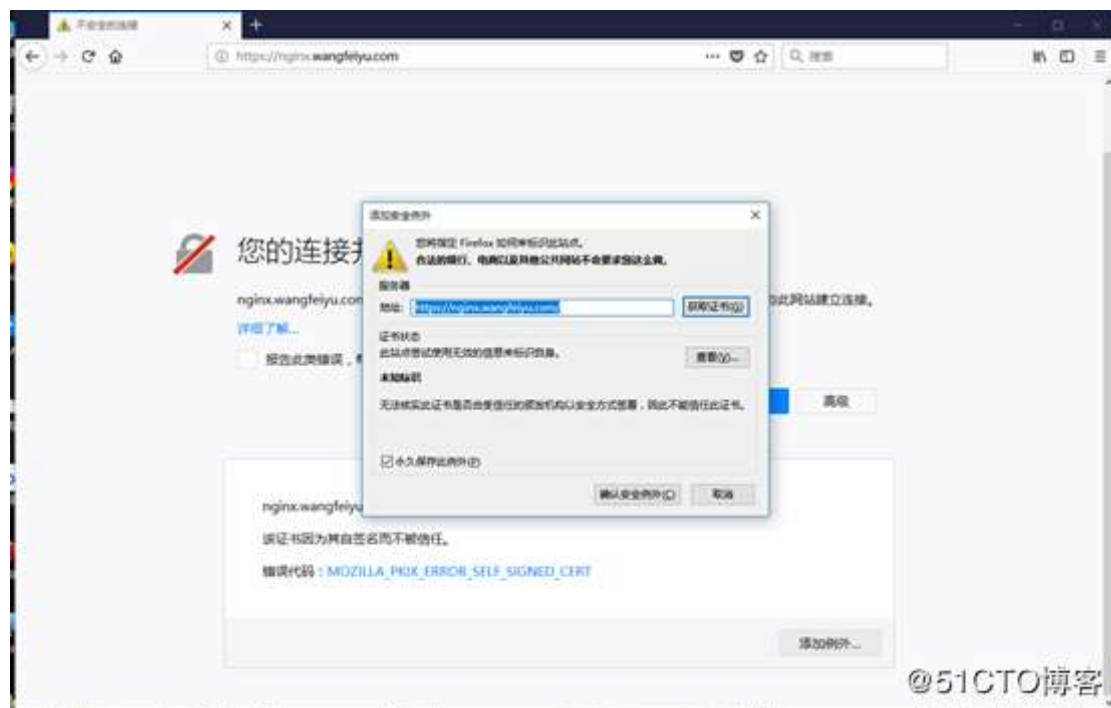
### 导入方式：

//点击高级



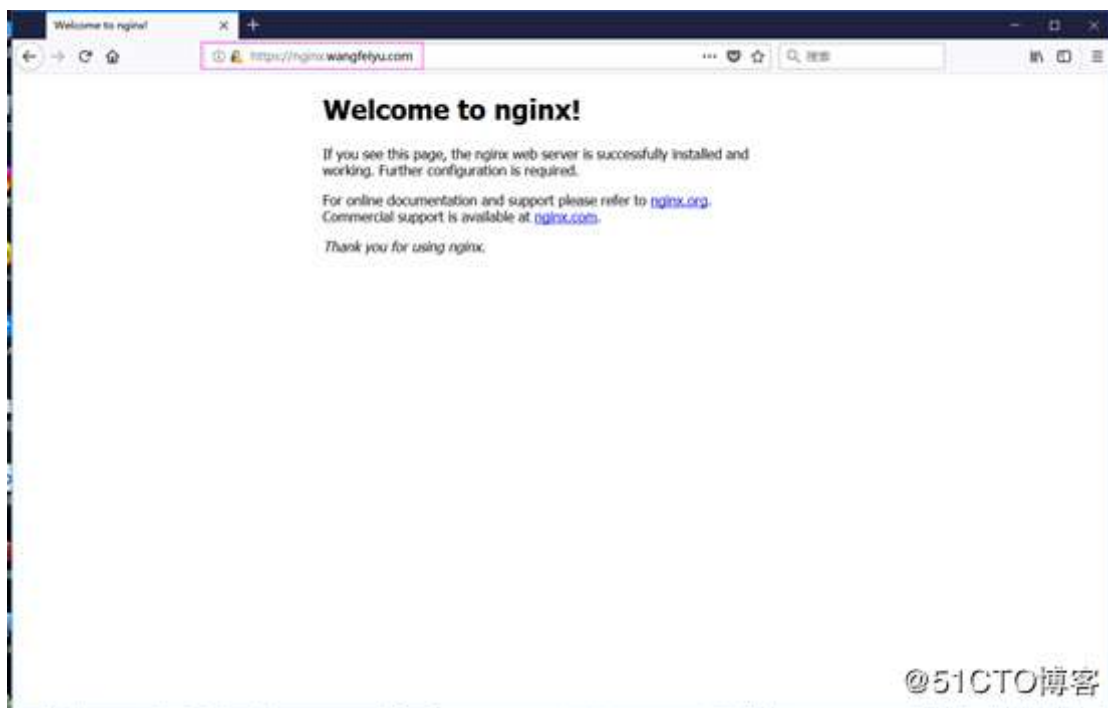


//点击添加例外



//点击确认安全例外





注：以上截图已经可以访问到网页，说明nginx加密成功或者证书导入成功！其他的浏览器导入证书方式不一样，但是超级简单，自行百度即可！

## 2) IP访问地址: <http://192.168.152.177/>



注：这种方式默认使用的还是http协议！也可以设置为通过http跳转到https！

## 3) IP地址https访问: <https://192.168.152.177/>

