

## 目录

1	大概思路.....	1
2	Nginx 集群之 SSL 证书的 WebApi 微服务 .....	1
3	HTTP 与 HTTPS（SSL 协议） .....	1
4	Openssl 生成 SSL 证书 .....	2
5	编写.NET WebApi .....	3
6	部署 WebApi 到局域网内 3 台 PC 机.....	5
7	Nginx 集群配置搭建 .....	6
8	运行结果.....	8
9	总结.....	9

# 1 大概思路

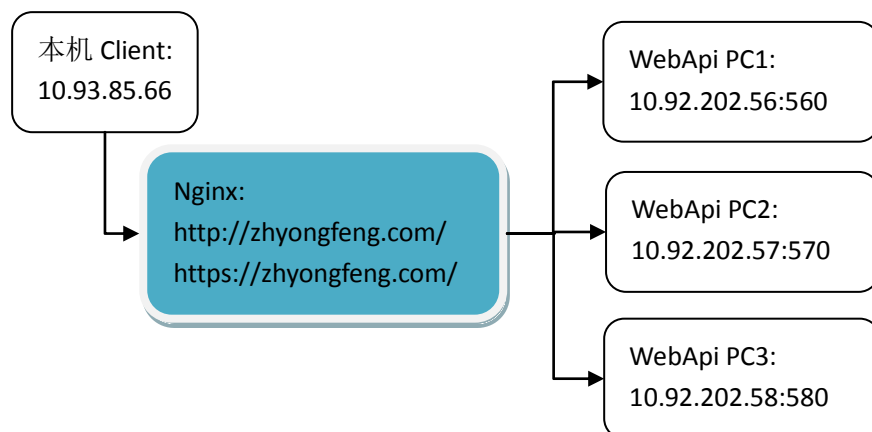
- Nginx 集群之 SSL 证书的 WebApi 微服务
- HTTP 与 HTTPS (SSL 协议)
- Openssl 生成 SSL 证书
- 编写 .NET WebApi
- 部署 WebApi 到局域网内 3 台 PC 机
- Nginx 集群配置搭建
- 运行结果
- 总结

## 2 Nginx 集群之 SSL 证书的 WebApi 微服务

Nginx 集群是 .NET WebApi 提供了负载均衡的其中一种实现方式,同时还增加了 SSL 认证,能够确保 WebApi 能够以加密形式进行响应。Nginx 使用其中的 SSL 模块,能够支持 HTTPS 的配置,当然也能够让 HTTP 与 HTTPS 并存(只需要增加 listen 80 监听端口则可),本文主要实现 HTTP 访问则跳转至以 HTTPS 方式。

以下是本文讲述的主要结构图:

客户端访问 Nginx 的域名 zhyongfeng.com, 然后 Nginx 进行负载均衡, 返回 https 的响应。Nginx 集群之 SSL 证书的 WepApi 微服务架构, 如下图所示:



## 3 HTTP 与 HTTPS (SSL 协议)

HTTP (HyperText Transfer Protocol)超文本传输协议是互联网上应用最为广泛的一种网络协议。所有的 WWW 文件都必须遵守这个标准。

HTTPS (Secure Hypertext Transfer Protocol) 安全超文本传输协议: 它是一个安全通信通道, 它基于 HTTP 开发, 用于在客户计算机和服务器之间交换信息, 它使用安全套接字层(SSL) 进行信息交换, 简单来说它是 HTTP 的安全版。

HTTPS 和 HTTP 的区别:

- https 协议需要到 ca 申请证书, 一般免费证书很少, 需要交费。
- http 是超文本传输协议, 信息是明文传输, https 则是具有安全性的 ssl 加密传输协议。
- http 和 https 使用的是完全不同的连接方式用的端口也不一样,前者是 80,后者是 443。
- http 的连接很简单,是无状态的。
- HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议 要比 http 协议安全。

## 4 Openssl 生成 SSL 证书

OpenSSL 是一个安全套接字层密码库, 囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议, 并提供丰富的应用程序供测试或其它目的使用。

本文安装了 Visualbox 虚拟机, 内装了 Linux Ubuntu 系统, 所以使用内置的 openssl, 具体生成 SSL 证书命令行如下:

(Windows 下可以到 <https://www.openssl.org/source/> 下载, 使用 ActivePerl 安装 Openssl): 自行颁发不受浏览器信任的 SSL 证书, 先生成一个 RSA 密钥

```
zhyongfeng@zhyongfeng-VirtualBox:~$ openssl genrsa -des3 -out server.key 1024
```

拷贝一个不需要输入密码的密钥文件

```
zhyongfeng@zhyongfeng-VirtualBox:~$ openssl rsa -in server.key -out server_nopass.key
```

生成一个证书请求, 会提示输入省份、城市、域名信息等, 重要的是, email 一定要是你的域名后缀的。这样就有一个 csr 文件了, 提交给 ssl 提供商的时候就是这个 csr 文件。

```
zhyongfeng@zhyongfeng-VirtualBox:~$ openssl req -new -key server.key -out server.csr
```

自己签发证书

```
zhyongfeng@zhyongfeng-VirtualBox:~$ openssl x509 -req -days 365 -in server.csr -signkey server.key -out
```

如下图所示:

```

zyhongfeng@zyhongfeng-VirtualBox:~$ openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
zyhongfeng@zyhongfeng-VirtualBox:~$ openssl rsa -in server.key -out server_nopass.key
Enter pass phrase for server.key:
writing RSA key
zyhongfeng@zyhongfeng-VirtualBox:~$ openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
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) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
zyhongfeng@zyhongfeng-VirtualBox:~$ openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

```

server	2017/11/29 10:55	安全证书	1 KB
server.csr	2017/11/29 10:55	CSR 文件	1 KB
server.key	2017/11/29 10:54	KEY 文件	1 KB
server_nopass.key	2017/11/29 10:54	KEY 文件	1 KB

## 5 编写 .NET WebApi

UserController.cs

```

using System.Net;
using System.Web.Http;

namespace SSLWebApi.Controllers
{
    [RoutePrefix("api/User")]
    public class UserController : ApiController
    {
        /// <summary>
        /// 获取当前用户信息
        /// </summary>
        /// <param name="msg"></param>
        /// <returns></returns>
        [HttpPost]
        [Route("PostMessage")]
        public string PostMessage([FromBody]string msg)
        {
            return string.Format("当前输入的消息是:{0}", msg);
        }
    }
}

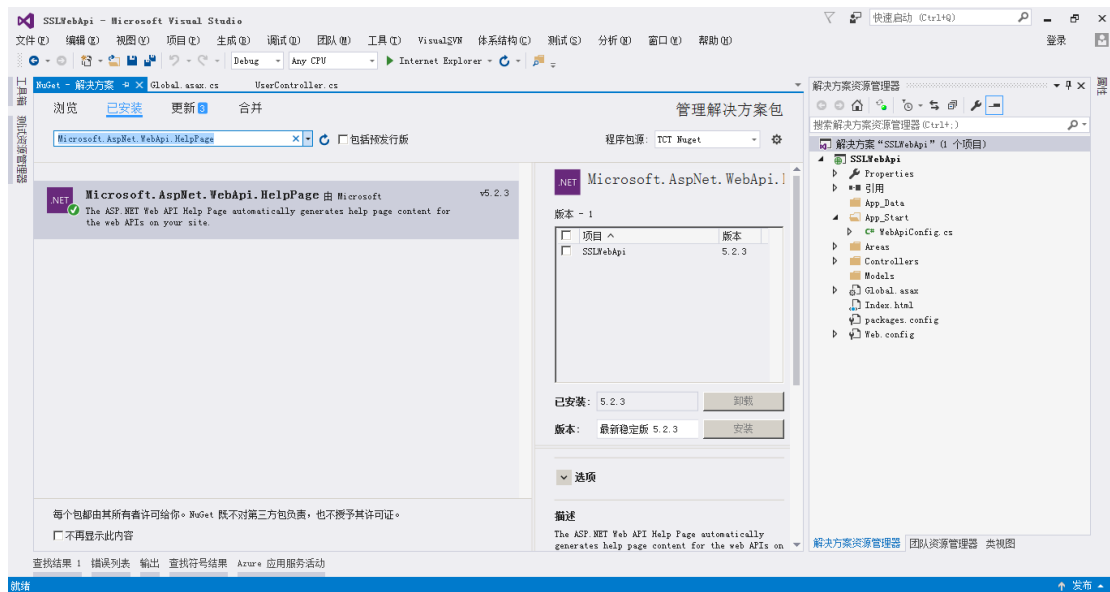
```

```

[Route("GetMachine")]
public string GetMachine()
{
    string AddressIP = string.Empty;
    foreach (IPAddress _IPAddress in
        Dns.GetHostEntry(Dns.GetHostName()).AddressList)
    {
        if (_IPAddress.AddressFamily.ToString() == "InterNetwork")
        {
            AddressIP = _IPAddress.ToString();
        }
    }
    return string.Format("当前系统的 IP 是: {0}", AddressIP);
}
}
}

```

### 安装 Microsoft.AspNet.WebApi.HelpPage



注册 HelpPage 页面:

Global.asax

```
using System.Web.Http;
```

```
using System.Web.Mvc;
```

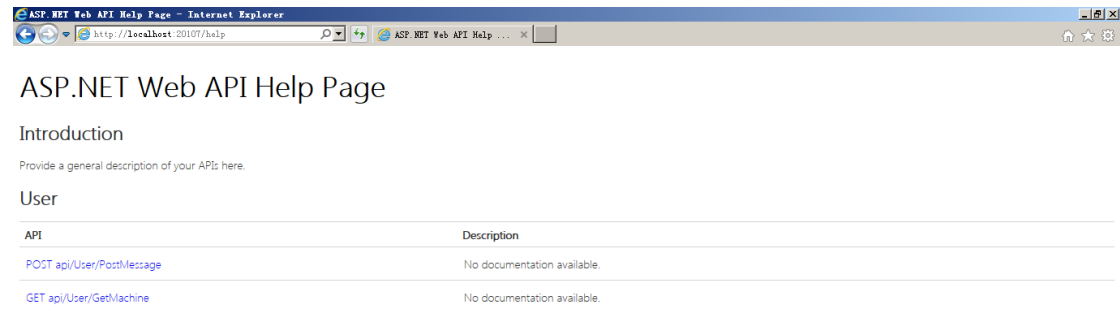
```
namespace SSLWebApi
```

```
{
    public class WebApiApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {

```

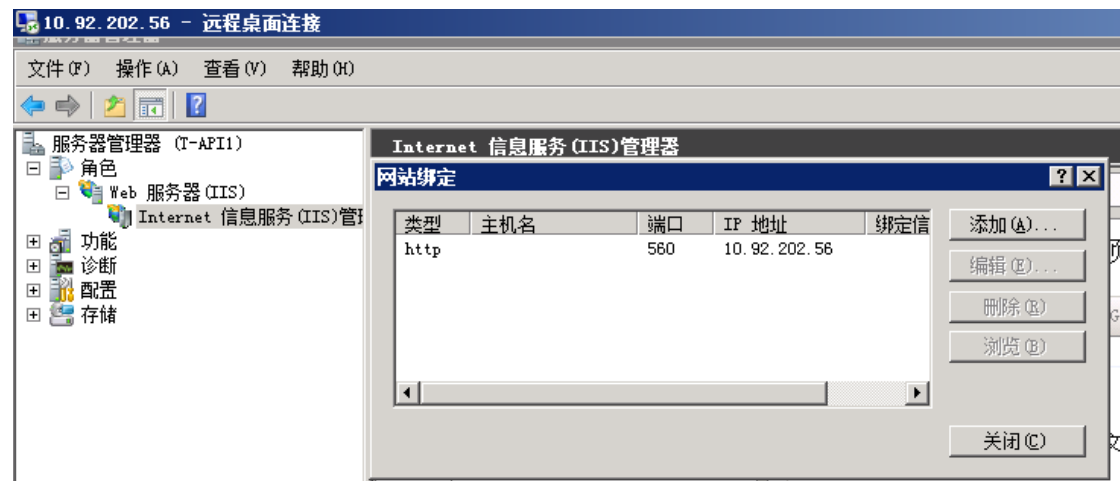
```
//注册 HelpPage 页面
AreaRegistration.RegisterAllAreas();
GlobalConfiguration.Configure(WebApiConfig.Register);
}
}
```

编译后如下:



## 6 部署 WebApi 到局域网内 3 台 PC 机

将 WebApi 部署到以下 10.92.202.56 的 3 台 PC 机



## 7 Nginx 集群配置搭建

通过自定义域名 `zhyongfeng.com` : 80 端口进行负载均衡集群访问, 则访问 `C:\Windows\System32\drivers\etc\hosts`, 添加下列“本机 IP 自定义的域名”:

```
10.93.85.66 zhyongfeng.com
```

Nginx 的 localhost 配置如下:

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    # HTTPS server
    server {
        listen 443 ssl;
        server_name localhost;
        ssl_certificate server.crt;
        ssl_certificate_key server_nopass.key;
        location / {
            root html;
            index index.html index.htm;
        }
    }
}
```

Nginx 的集群配置:

```
#user nobody;
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    #server {
    #    listen 80;
    #    server_name localhost;
```

```
# location / {
#     root    html;
#     index  index.html index.htm;
# }
# error_page 500 502 503 504 /50x.html;
# location = /50x.html {
#     root    html;
# }
#}

upstream zhyongfeng.com {
    server 10.92.202.56:560;
    server 10.92.202.57:570;
    server 10.92.202.58:580;
}

server {
    listen      80;
    server_name zhyongfeng.com;
    rewrite ^(.*)$ https://$host$1 permanent;
}

# HTTPS server
#
server {
    listen      443 ssl;
    server_name zhyongfeng.com;
    ssl_certificate      server.crt;
    ssl_certificate_key  server_nopass.key;
#    ssl_session_cache  shared:SSL:1m;
#    ssl_session_timeout 5m;
#    ssl_ciphers HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers on;
    location / {
        proxy_pass http://zhyongfeng.com;
    }
}
}
```

运行 CMD:

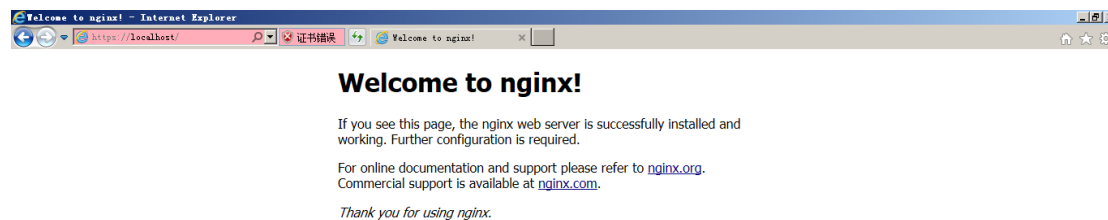
```
D:\DTLDownloads\nginx-1.10.2>start nginx
```

```
D:\DTLDownloads\nginx-1.10.2>nginx -s reload
```

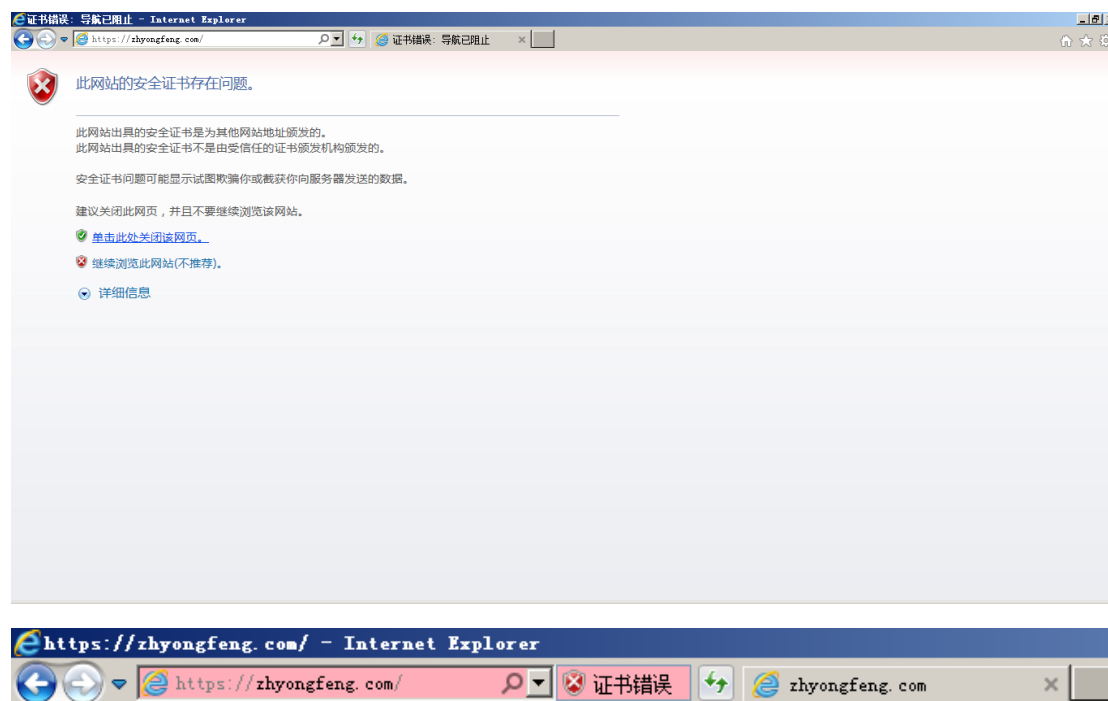


## 8 运行结果

Nginx 的 localhost 配置运行结果:



Nginx 的集群配置运行结果:



**Hello, This Computer IP Address is 10.92.202.57 !**



**Hello, This Computer IP Address is 10.92.202.58 !**



## ASP.NET Web API Help Page

### Introduction

Provide a general description of your APIs here.

### User

API	Description
<a href="#">POST api/User/PostMessage</a>	No documentation available.
<a href="#">GET api/User/GetMachine</a>	No documentation available.

## 9 总结

Nginx 使用 SSL 模块可以支持 WebApi 的 https 访问，增加了访问的安全性。SSL 模块，具体可以参考 Nginx 中文文档 <http://www.nginx.cn/doc/> 的 SSL\* 模块。WebApi 基于 SSL 协议数据传输的加密，保证了通信的安全性。SSL 的功能包含了建立服务器与客户之间安全的数据通道、服务器对客户的身分认证（如公钥、私钥）等。

源代码下载：

[http://download.csdn.net/download/ruby\\_matlab/10138057](http://download.csdn.net/download/ruby_matlab/10138057)

PDF 下载：

[Nginx 集群之 SSL 证书的 WebApi 微服务.pdf](#)