

应用和 IPv6

前言

在数据中心网络IPv6协议改造时，我们通常更关注路由交换的部分。对于应用系统适配IPv6网络确实缺少关注，本文旨在更多的讨论应用和IPv6的关系，帮助个人、公司和组织能够更好的将他们的应用程序迁移至IPv6网络。

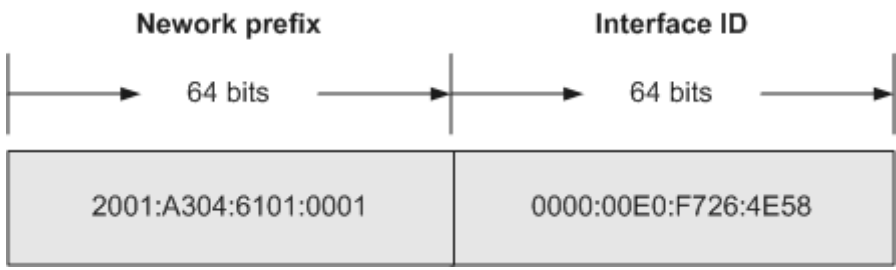
道阻且长，行则将至

第一章、用户和IPv6

IPv6（Internet Protocol Version 6）是网络层协议的第二代标准协议，也被称为IPng（IP Next Generation），它所在的网络层提供了无连接的数据传输服务。IPv6是IETF设计的一套规范，是IPv4的升级版本。它解决了目前IPv4存在的许多不足之处，IPv6和IPv4之间最显著的区别就是IP地址长度从原来的32位升级为128位。IPv6以其简化的报文头格式、充足的地址空间、层次化的地址结构、灵活的扩展头、增强的邻居发现机制将在未来的市场竞争中充满活力。

对于应用程序开发人员可能需要修改应用套接字接口(socket)以支持ipv6功能，但是对于绝大多数应用来说他们所依赖的框架、系统、web服务都已对IPv6有良好的支持，只需要在配置文件中开启对应功能即可。

IPv6地址由网络前缀和接口标识两个部分组成。网络前缀有n位，相当于IPv4地址中的网络ID；接口标识有（128-n）比特，相当于IPv4地址中的主机ID。



- 地址2001:A304:6101:1::E0:F726:4E58的构成示意图*

输入输出

IPv6 直观来看最大的变化是从IPv4 的点分十进制四个固定的地址块，变成了IPv6 冒号分隔8个大小不一的16进制地址块。

```
IPv4 address: 192.168.1.1
```

```
IPv6 address: 2001:A304:6101:1::E0:F726:4E58
```

当我们的应用系统需要输入输出IPv6 地址时，需要考虑显示区域是否能容纳IPv6 地址，功能组件是否能够同时处理IPv4 地址和IPv6 地址。还是使用单独控件(例如单选按钮)来进行切换。

可变长IPv6 地址

第二大变化是 IP 地址长度的可变性。使用 IPv4，地址具有四组十进制数的固定格式。但是，在 IPv6 中，地址可以用冒号分隔的完整八个十六进制数字块 - 或可以使用双冒号(“::”)表示法压缩全0的块，得到一个更短的地址。下列其中任何一个都是完全有效的 IPv6 地址：

```
2001:db8::1
```

```
2001:db8:10ff::ae:44f2
```

```
2001:db8:1212:3434:1212:5454:1afc:4001
```

您的应用程序能否应对地址格式的可变性？

现在不能再将 IP 地址视为四个十进制数字的情况进行处理，您必须能够处理使用分隔符解析文本字符串 - 基于冒号的情况。

考虑这些其他示例：

- 在 IPv4 中，本地环回地址为 127.0.0.1。在 IPv6 中它是 ::1。
- IPv4 地址198.51.20.1 内嵌IPv6 采用 ::198:51:20:1 形式。

您的应用程序是否能够处理这些变化？

端口号冒号问题

在 IPv4 中，您添加的 TCP 或 UDP 端口使用冒号将端口号添加到 IPv4 地址。例如，如果您想访问某个系统端口 8000 上运行的 Web 服务器，您将使用类似如下 URL：

- `http://198.51.100.22:8000/`

您看到这将在 IPv6 中引起的问题吗？如果在 IPv6 地址中使用冒号作为分隔符，如何表示端口数字？这是一个普遍的问题。你需要考虑一下。

解决端口号冒号问题的主要方法是在 RFC 3986 中规定的 IPv6 地址周围使用方括号。考虑到在 IPv4 下，如果你想连接到一个系统上运行的服务，很常见的做法是直接在 URL 中输入 IPv4 地址。例如，在一个网络浏览器中，你可以输入这些 IP 地址中的任何一个：

- `http://198.51.100.22/`
- `http://198.51.100.22:8000/`

使用 IPv6 访问 URL 可能如下所示：

- `http://[2001:db8:10ff::ae:44f2]/`
- `http://[2001:db8:10ff::ae:44f2]:8000/`

请注意，在第二种情况下，端口号在方括号之外，因此才能解决端口号冒号问题。

这种方括号表示法不仅适用于 Web URL。它可以也可用于其他协议的 URI。例如，URI 与 IP 语音的会话发起协议 (SIP) 可能看起来像这个：

- `sip:admin@[2001:db8:34a5::1]`
- `sip:admin@[2001:db8:34a5::1]:5060`

当然，理想情况下，您使用的是域名并且只需键入这些地址用于测试目的。如果您的应用程序需要使用 URL 或 URI，您需要考虑是否可以解析方括号。

(CIDR) 表示法与子网掩码

IPv6 的另一个变化是区分网络部分和主机部分的。使用 IPv4，如果您要向某人提供您的 IP 地址，您通常会像下面这样给他们：

- IP 地址 198.51.100.22，子网掩码为 255.255.255.0

从应用程序的角度来看，您通常有两个单独的字段来输入 IPv4 地址和相应的子网。IPv4 使用中一些人会使用无类域间路由 (CIDR) RFC 4632 中指定的表示法。您可能还听说过这种称为斜线表示法或类似的东西。这个想法是，您无需提供特定的子网掩码，而是指定对地址的网络部分重要的位数。在上例中，子网掩码 255.255.255.0 表示前 24 位（三个 8 位的块）地址代表网络部分和后面 8 位代表主机部分。这用 CIDR 表示法写成：

- IP 地址 198.51.100.22/24

IPv4 通常使用这种表示法，但是对于 IPv6 此表示法是强制性的。所有地址都使用斜线符号。IPv6 地址可以写成：

- IP 地址 2001:db8:1234:5a:f3a3::22/48

前 48 位（地址的前三个块）表示网络部分。同样，如果给定一个 IPv6 地址块，它通常会写成：

- 2001:db8:1234:56::/64

前 64 位（或地址的前四个块）代表网络部分。从应用程序开发人员的角度来看，这里的主要问题是：

- 如何显示 IPv6 地址和网络掩码？在 IPv4 中您使用两个输入框那么在 IPv6 中您需要合并。
- 同样，您需要考虑如何处理 IPv6 地址和相应的输入。

大小写敏感问题

在 IPv6 网络中下面两种地址写法都是相同的。

```
2001:db8:10ff::ae:44f2
2001:DB8:10FF::AE:44F2
```

与IPv4相比，在应用程序中需要考虑大小写问题，或者大小写混和的问题。应用代码是否对大小写敏感。

RFC 5952建议所有的IPv6地址都应该用小写书写，RFC 5952还包含其他一些关于IPv6地址书写方式的建议。例如，它建议使用 "::" 对IPv6地址中连续的零字符串进行压缩，可以作为参考。

输入字段的有效性检查

当考虑到地址格式的变化时，你需要考虑表格标签输入内容的有效性检查。通常情况下，你要对任何形式的输入进行某种格式的有效性检查，以确保输入的IP地址是正确的并防止安全问题，如跨站脚本、注入攻击等。诸如此类的问题。对于IPv4，这是相对简单的。一个IPv4地址有一个固定的格式，并且只使用十进制数字。

IPv6地址可以是一个可变的格式，并使用十六进制数字，开发人员需要考虑如何处理这种情况。

第二章、DNS 和IPv6

理想情况下我们使用域名而不是IPv6地址访问业务系统，如果使用域名访问那么就需要正确的域名配置。以使用户能够正确的查询到对应的地址。在IPv4中，将主机名www指向IP地址的DNS记录是一个A记录，在DNS区域文件中是这样的

```
www 3600 A 198.168.1.1
```

在IPv6中，新的DNS记录是AAAA（也被称为“四A”）记录，例如

```
www 3600 AAAA 2001:abcd:12af::53
```

通常情况下为了同时满足IPv4 和IPv6 的使用我们需要同时配置如下记录

```
www 3600 A 198.168.1.1  
www 3600 AAAA 2001:abcd:12af::53
```

应用如何处理AAAA 记录

假如您的应用是一个APP,您的客户端需要考虑如何处理响应的AAAA 记录,当您能获取到AAAA 记录时也并不代表您具备访问IPv6 业务的条件。

“dig”是 BIND 开发的一个强大的命令行工具,用于查询 DNS 名称服务器。它可以识别 IP 地址记录,从权威名称服务器获得响应时记录查询路由,诊断其他 DNS 问题。在您的 IPv4 网络终端中可以进行如下测试:

```
选择 C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19044.1889]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\zy>dig @114.114.114.114 www.sina.com AAAA

; <<>> DiG 9.14.4 <<>> @114.114.114.114 www.sina.com AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48371
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;, udp: 512
;; QUESTION SECTION:
;www.sina.com.                IN      AAAA

;; ANSWER SECTION:
www.sina.com.                76      IN      CNAME   spool.grid.sinaedge.com.
spool.grid.sinaedge.com.    76      IN      AAAA    2400:89c0:1023:1::23:216

;; Query time: 26 msec
;; SERVER: 114.114.114.114#53(114.114.114.114)
;; WHEN: Mon Aug 22 16:43:47 ;; MSG SIZE  rcvd: 103

C:\Users\zy>
```

AAAA和A记录的优先级

一个更大的问题是,当你的应用程序与主机通信时,会默认使用哪个地址?主机时默认使用哪个地址?AAAA记录所提供的IPv6地址?还是A记录提供的IPv4地址?当有A 记录和AAAA 记录时如何选择合适的访问地址,以及当其中一个记录响应的地址无法访问时,如何切换至另一个记录。这是一个很关键的问题,也是导致业务IPv6 部署缓慢的因素之一。

在一些早期的实验中,网站同时列出A和AAAA记录对应的服务地址,浏览器会在不具备IPv6 通信条件的终端上优先和IPv6地址的服务器通信。最终它将超时并切换到使用IPv4地址,但超时的时间太长,以至于用户体验很差。由于这个原因,一些网站已经转向拥有一个IPv6标签的网站版本。例如:

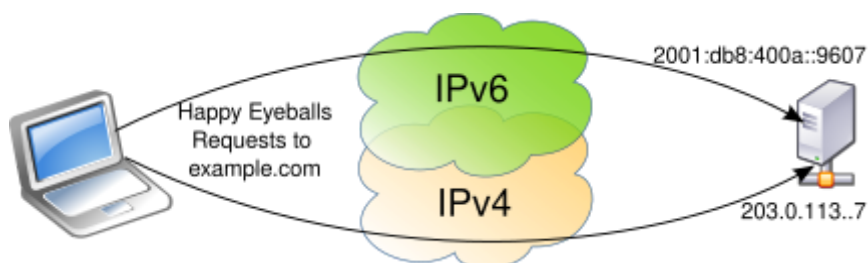
```
ipv6.google.com
www.v6.facebook.com
```

显然网站使用独立的IPv4和IPv6的DNS名称并不理想,应用程序能够同时正确处理这两种类型的记录才是最理想的方式。

Happy Eyeballs

Happy Eyeballs（也称为 Fast Fallback）是 IETF 发布的一种算法，它通过尝试同时使用 IPv4 和 IPv6 进行连接（更喜欢 IPv6），从而最大限度地减少 IPv6 连接或设置不完善的用户遇到的常见问题。“快乐的眼球”这个名字来源于术语“眼球”来描述代表互联网最终用户的端点，而不是服务器。

Happy Eyeballs 旨在解决许多 IPv6 网络无法从 Internet 的某些区域访问的问题。Happy Eyeballs 通过并行尝试访问 IPv4/IPv6 来确定哪种传输更适合从而解决了这个问题。使用 Happy Eyeballs 算法的应用程序会检查 IPv4 和 IPv6 连接（首选 IPv6）并使用返回的第一个连接。这些地址通常是使用循环算法从 DNS 中选择的。



图片来源:http://www.makikiweb.com/ipv6/_images/happy_eyeballs.png

Happy Eyeballs 是2011年世界IPv6日的一部分。

Happy Eyeballs算法还可以用于在其他类型的传输协议之间进行选择，例如在TCP和SCTP之间进行选择。

来自:https://en.wikipedia.org/wiki/Happy_Eyeballs

RFC:<https://tools.ietf.org/html/rfc6724#section-6>

第三章、Application Programming Interfaces (APIs)

在当前世界API 纷繁多样的情况下您需要考虑您的应用程序是否开放了需要关注IP 地址格式的API，对于IP地址是否有依赖关系。还需要考虑相关的功能、模块、库的支持。

API 通信

当通过IPv6 访问您的API 时，您需要确保您的系统、apache、nginx、数据库、SLB、应用能够进行IPv6 通信。

apache 举例：

```
Listen [2400:6180:0:d0::1f33:d001]:80
```

nginx 举例：

```
服务监听  
listen [::]:8000;  
listen [::1];  
ipv6only=on|off
```

参考:http://nginx.org/en/docs/http/nginx_http_core_module.html

第四章、IP 地址存储

对于需要通过对源IP地址进行分析展示模块，您需要考虑在双栈情况下如何存储对应的地址。

数据库

针对下面两种写法，您需要在数据库中选择合适的数据类型和长度。

```
::1
```

```
2001:db8:3f4d:ccdd:4179:880a:f00d:5511
```

文件

当在文件中写入IPv6 地址时一定请不要忘记添加[] .以及注意文件大小写及使用[::] 进行压缩。

第五章、传输

技术名称	双栈技术	隧道技术					转换/翻译技术
		GRE隧道	6over4手工配置隧道	6PE/6VPE隧道	6to4隧道	ISATAP隧道	
适用阶段	初期或后期	过渡全程	初期	初、中期	初、中期	初期	过渡全程
应用场合	IPv6/IPv4孤岛互联	点对点连接孤立IPv6（或IPv4）网络	点对点连接孤立IPv6网络	多点对多点连接孤立IPv6（或IPv4）网络	点对点连接孤立IPv6网络	IPv4网络中零星双栈主机接入IPv6网络	IPv6与IPv4网络边界实现协议及应用的转换
封装方式		GRE封装IPv6（或IPv4）报文	IPv4封装IPv6报文	MPLS封装IPv6报文	IPv4封装IPv6报文	IPv4封装IPv6报文	
建立方式		手工配置	手工配置	手工配置	自动生成	自动生成	
技术特点	技术成熟，主流厂商设备均支持；在实际应用中往往需与其它隧道技术结合。	技术成熟；仅对入口和出口路由器有双栈要求；安全性较好；技术寿命长。但仅能提供点对点连接。	简单、成熟、稳定。但可扩展性差，安全性不如GRE隧道。	多点对多点互联；6VPE可提供IPv6 VPN。技术成熟；只需对PE设备做升级，成本低；适合大规模部署。	自动配置；中继能力；平滑升级；NAT替代；成本低廉。但地址空间有限，不适用于新建IPv6网络之间互通。	技术简单；扩展性好；很容易与其它隧道技术结合使用；可跨越NAT。	NAT-PT作为主要的翻译技术其发展较成熟，基本实现协议层的翻译，但对于含IPv6地址的部分应用翻译效果较差。目前NAT64技术较为成熟。

表 1 各种过渡技术的比较

	6rd	DS-lite	A+P	NAT64	PNAT	IVI
种类	隧道	隧道	隧道	翻译	翻译	翻译
技术特点	以 IPv4 为隧道，承载 IPv6 流量	以 IPv6 为隧道，解决 IPv4 地址不足问题	以 IPv4 地址加端口方式，解决 IPv4 地址不足问题	支持 IPv6 应用程序访问 IPv4 网络中的资源	主机侧 46 翻译+网络侧 64 翻译(可选)，无需修改应用程序	无状态 IP 地址族翻译
技术定位	互通方案	互通方案	互通方案	IPv6 访问 IPv4	IPv4/IPv6 互访	IPv4/IPv6 互访
CPE 设备地址分配	获得 ISP IPv6 前缀，同时获得私有/公有 IPv4 地址	获得全球 IPv6 地址	获得端口受限的全球 IPv4 地址，同时获得全球 IPv6 地址	获得全球 IPv6 地址	全球 IPv6 地址，双栈环境还可获得私有/公有 IPv4 地址	获得 IVI6 地址
是否需要 NAT	可选 NAT	网络侧 AFTR 进行 NAT	CPE 进行 NAT	需要	需要	1:1 映射时不需要
用户 IPv6 终端	支持，通过 IPv4 隧道访问 IPv6 资源	支持，可以直接访问 IPv6 资源	支持，可以直接访问 IPv6 资源	支持，可直接访问 IPv6 资源，通过网关翻译可访问 IPv4 资源	支持，可直接访问 IPv6 资源，通过 PNAT64 网关可访问 IPv4 资源	支持，可以直接访问 IPv6 资源，通过 IVI 网关翻译可访问 IPv4 资源
用户 IPv4 终端	支持，可以直接访问 IPv4 资源	支持，通过隧道访问 IPv4 资源	支持，通过隧道访问 IPv4 资源	不支持	支持，通过主机侧 46 翻译和网络侧 64 翻译访问 IPv4 资源	不支持，用户终端只能是得到 IVI6 地址的 IPv6 终端
用户侧的改变	CPE 需升级支持 6rd DHCP 选项及隧道模块	相关设备需增加 B4 模块且支持 DNS proxy 功能	CPE 需支持分配端口范围 DHCP 协议和 NAT 模块	用户终端必须支持 IPv6 协议栈	用户终端需升级 PNAT 模块	用户终端必须支持 IPv6 协议栈
网络侧的改变	需部署 6rd BR 设备	需部署 AFTR 设备	需部署 PRR 设备	需部署 NAT64 与 DNS64 设备	需部署 PNAT64 网关设备	需部署 IVI 网关和 IVI DNS

Path MTU Discovery

在互联网数据传输中，通常我们认为数据是由路由器一跳一跳进行数据转发，而路由器之间通过物理链路进行连接。而在物理链路中传输的数据报都被编码在数据链路层帧中进行传输。

为了使数据报成功地沿路径传送，它的大小必须足够小，以适应沿途每一跳的较小帧传输。最大传输单元 (MTU) 定义了物理网络的大小限制。如果一个数据报超过 MTU，它必须被分成多个片，这个过程称为分片，然后在目标设备上重新组合这些片段。这是自 IPv4 的一项机制。

在 IPv4 中，路由器和物理链路需要处理的最小 MTU 为 576 字节。在 IPv6 中，所有链路必须能够处理至少 1280 字节的数据报文，不允许数据进行分片。通过增加最大有效载荷与报头长度的比率，这种传输速率对于 IPv4 几乎由一倍的提升，并降低了需要分段的频率。以上内容对于业务出现缓慢问题提供参考。

RFC: <https://www.rfc-editor.org/rfc/rfc2460#section-5>

RFC: <https://www.rfc-editor.org/rfc/rfc1981>

第六章、测试及说明

在您对应用系统能否支持IPv6 进行评估时您需要考虑下面这些问题：

部署场景

网络环境是否支持IPv6 ,如果不支持是否由其它的过渡机制来实现，例如NAT64 或者 tunnel

1. 仅使用 IPv4 网络：一些网络仍然仅使用 IPv4，并且会在很长一段时期内保持这种状态。您需要确认应用程序能否正常工作。
2. 仅使用 IPv6 网络：未来IPv6 绝对会完全替代IPv4,为啥？猜的。您需要考虑是否只支持IPv6.
3. 双栈网络: 这是目前使用最多、最广泛的技术。配合其它过渡机制可以灵活部署。对于双栈的选择您可以参考前面的Happy Eyeballs

4. **ISP和DNS:** 当使用IPv6 时我建议您尽可能的让整个系统的通信使用域名，可以避免很多配置查看的错误。还有很重要的一点是您需要向您的网关咨询数据中心的ISP 是否接入了IPv6 ,这是业务IPv6 发布的关键。

日志文件、数据库、正则表达式和操作系统

对于 Web 服务器，例如 IIS、nginx、Apache 等..... 您将开始在日志文件、**referrers**、源 IP 等中看到 IPv6 地址的打印输出。对于常见业务统计和问题排查您需要对过滤脚本进行修改以支持IPv6 地址,建议您更多的使用各种库.例如 .NET 中的 **IPAddress** 和 Java 中的 **InetAddress**。还需要确认您的数据库能否支持IPv6.

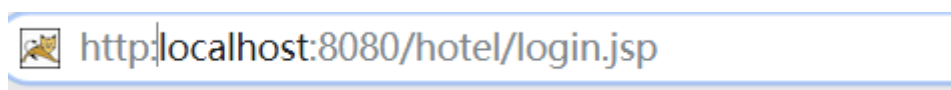
Libraries

IPv6 地址与 IPv4 地址的不同之处在于它们的长度是不可预测的。对于您引入的处理IP 地址的库您需要确认是否支持IPv6.

URLs

在WWW上，每一信息资源都有统一的且在网上唯一的地址，该地址就叫**URL（Uniform Resource Locator**,统一资源定位符），它是WWW的统一资源定位标志，就是网络资源位置。

简单来说就是你打开一个浏览器，地址栏上显示的网址，例如：



对于IPv6 地址访问如下：

```
https://[2001:500:4:13::125]:443/
```

同样也适用与curl 命令。

DNS

在前面我们已经说了DNS 对于IPv6 的重要性。您需要让您的网络部门确认DNS 对于IPv6 的支持情况。并学会适用dig 和nslookup 来辅助进行日常的问题排查。

IP Geolocation

应用程序根据客户端的位置定制服务是很常见的，其中一种使用的方法是使用客户端的IP 地址（IP 地理定位）在地理上定位客户端。在这种情况下，将地址类型从 IPv4 更改为 IPv6 可能会产生一些影响。

IP 地理定位的工作原理是获取客户端所在地理位置的 IP 地址并在数据库中查找。数据库的创建方式和准确性取决于数据库的提供者。根据数据库进行 IP 地理位置定位，您需要问下能否支持IPv6 ，还有IPv6 的地址信息是否准确。

IP Geolocation 根据我日常的工作在SLB、DNS 、NGINX 中都有相关的功能，并且我们也有使用。通常IP Geolocation 根据我的了解都是购买的专业服务商的信息。

Loopback and localhost

一些客户端应用程序使用 IP 套接字专门向同一终端上的其他进程发送数据或创建连接。一些应用程序具有特殊的处理规则，例如授权策略，如果软件识别出连接来自同一终端上的另一个进程，这些规则就会发生变化。通常称为回环地址通信。

在 IPv4 中，回环地址可以是 127.0.0.0 和 127.255.255.255 之间的任何地址，尽管 127.0.0.1 默认使用的地址。对于 IPv6，只有一个回环地址 ::1。对于客户端，解决本地进程通信最好的方法是解析 localhost 主机名，而不是使用 IP 地址。

测试

下面是我平时收集整理的IPv6 测试工具：

IPv6 地址计算：http://www.gestioip.net/cgi-bin/subnet_calculator.cgi

上海交通大学IPv6 网站测试:<https://ipv6.ustc.edu.cn/onlinecheck.php?hostname=&cmd=%E9%87%8D%E6%96%B0%E6%B5%8B%E8%AF%95> (推荐使用速度很快)

正在测试 www.sina.com, 请等待测试完成

测试项目	结果
DNSSEC	Failed!
IPv4解析	www.sina.com is an alias for spool.grid.sinaedge.com. spool.grid.sinaedge.com has address 121.194.5.14
IPv4 HTTP	✓
IPv4 HTTPS	✓
IPv4 HTTP/2	✓
IPv6解析	www.sina.com is an alias for spool.grid.sinaedge.com. spool.grid.sinaedge.com has IPv6 address 2400:89c0:1133:5::22:77
IPv6 AAAA	✓
IPv6 HTTP	✓
IPv6 HTTPS	✓
IPv6 HTTP/2	Failed!

测试完成

主机名:

[重新测试](#)

注意事项:当我们的域名使用CNAME 后, 有的测试网站会报告错误。具体原因在于测试工具对于DNS CNAME 没有进一步处理。

DNS 在线测试:<https://www.gdnspc.com/dns-propagation-checker/cn.html#MX&cge.cl>

API 测试: <https://www.postman.com/> (postman 支持ipv6 请求,如果测试报错, 请选择官方支持版本)

测试终端是否支持IPv6: <https://test-ipv6.com/>