

# Build VPN Server On Conoha VPS

随笔 conoha.jp vps ssh vpn pptp linux-dash

## Preface

提到VPS，就不得不想到云服务器业鼻祖的亚马逊公有云服务AWS（Amazon Web Services），Dell云，微软Windows Azure，传说中的阿里云ECS，以及VPS.me，老牌免费VPS—HapHost，爆发新秀Linode，小清新的台湾Micloud，非主流德国Host1Free。。

AWS的确牛逼，产品包括弹性计算网云 Amazon EC2、储存服务 Amazon S3、数据库服务 Amazon SimpleDB 等，只要有一个信用卡就可以免费使用一年AWS VPS主机。。。 （信用卡没有的说T\_T...）

好吧，对于上面提到的各种神器，对于一个没有信用卡没有美国手机号码翻不了墙不支持中国的学生来说太痛苦了，而且，生命的意义不在于浪费时间折腾在这些琐事上，所以才有了下文，就是想让各位看官免去各种不应该的折腾。

第一期送免费VPN，教你搭建私有VPN服务器，从此不用肉身翻墙。随后的系列里将带领你搭建私有Github和NPM。

## VPS

VPS（Virtual Private Server 虚拟专用服务器），是指通过虚拟化技术在独立服务器中运行的专用服务器。每个使用VPS技术的虚拟独立服务器拥有各自独立的公网IP地址、操作系统、硬盘空间、内存空间、CPU资源等，还可以进行安装程序、重启服务器等操作，与运行一台独立服务器完全相同。

这些VPS主机以最大化的效率共享硬件、软件许可证以及管理资源。每个VPS主机都可分配独立公网IP地址、独立操作系统、独立超大空间、独立内存、独立CPU资源、独立执行程序和独立系统配置等。VPS主机用户除了可以分配多个虚拟主机及无限企业邮箱外，更具有独立主机功能，可自行安装程序，单独重启主机。

废话这里就不多说了，对于VPS大家应该都了解，开始进入正题。

## Conoha

Conoha是啥，也是一家提供VPS服务的日本厂商。为啥用这家的？第一，毕竟是日本的，小清新，我喜欢；第二，没有各种烦恼，一步到位，简洁明了，再也不用担心没有信用卡了。

既然这样，小伙伴我们开始上路吧！

首先进入[官网](#)：



### ConoHa VPSの特長



#### 仮想サーバーの高い基本性能

ConoHaのVPSなら、これまでより少ないサーバー数あなたのシステムを支えられます。高いCPU性能やディスクI/O性能、大容量のメモリが実現するものは、驚くほど優れたコストパフォーマンスです。



#### 「安い」以上に「安心」

ConoHaのVPSは安心の月額固定・後払い。しかも、VPSを追加・削除した月の料金は日割りでの精算です。初期費用はもちろん、最低利用期間もありませんので、頻繁な追加・削除も気軽にできます。



#### 選べるOS

1分でサーバーが追加できる「テンプレートイメージ」、常に最新バージョンが体験できる「インストールイメージ」からOSを選べます。お持ちのISOイメージから再インストールすることも可能です。



#### 洗練されたコントロールパネル

サーバーやネットワークの管理に用いるコントロールパネルは、細部までこだわりぬきました。複数サーバーの一元管理はもちろん、ConoHaの先進的な機能を思うままに操ることができます。

[もっと見る](#)

[もっと見る](#)

[もっと見る](#)

[もっと見る](#)

## 最新の記事

**ConoHaの薄い本電子版 Vol.0 「クラウド?VPS? ConoHaがわかるはじめの一歩」**

「ConoHaの薄い本」電子版シリーズ、Vol.0を公開しました。ConoHaの薄い本は、ConoHaイベントなどで配布している「Co...」

> もっと見る

**このべん第5回 & 出張このべん in 大阪を開催しまし…**

こんにちは、ひろのぶです。3月19日に「このべん第5回」を東京で、そして3月21日に「出張このべん in 大阪」(以下出張このべん)を開…

> もっと見る

**オープンソースカンファレンス2015 Tokyo/Spr…**

2015年2月27(金)と28日(土)に「オープンソースカンファレンス2015 Tokyo/Spring」が開催されました。東京で開催される…

> もっと見る

**構成管理ツールChefを使ってみよう**

こんにちは、まさやです。普段はGMOインターネットのインフラエンジニアをしています。今回はとある縁でBlogを書かせていただくことになりました…

> もっと見る

## ConoHa支援プログラム



### ConoHa支援プログラム一覧

インターネットを楽しく、そして、より豊かにするために、もっとも優れたインフラと熱い想いを持った人が集まる空間を提供します。

> もっと見る



### プログラミングの楽しさを伝えたい！オンライン学…

23時過ぎまでマンツーマンレッスンを受けられる！2012年12月設立の株式会社トライブユニバ。代表取締役である池田洋…

> もっと見る



### 日本から世界へ！全文検索エンジン「Groonga…

さらなる進化を育む独自のコミュニティ活動とは？日本人が日本で手掛ける数少ない全文検索エンジンの1つ・Groonga。…

> もっと見る



### 「まずは何でもやってみる」がキーワード！情報通信…

現在、OpenStackの活用に挑戦中！情報通信に関する知識や技術を高めることを目的に、2014年、東海大学の学生が…

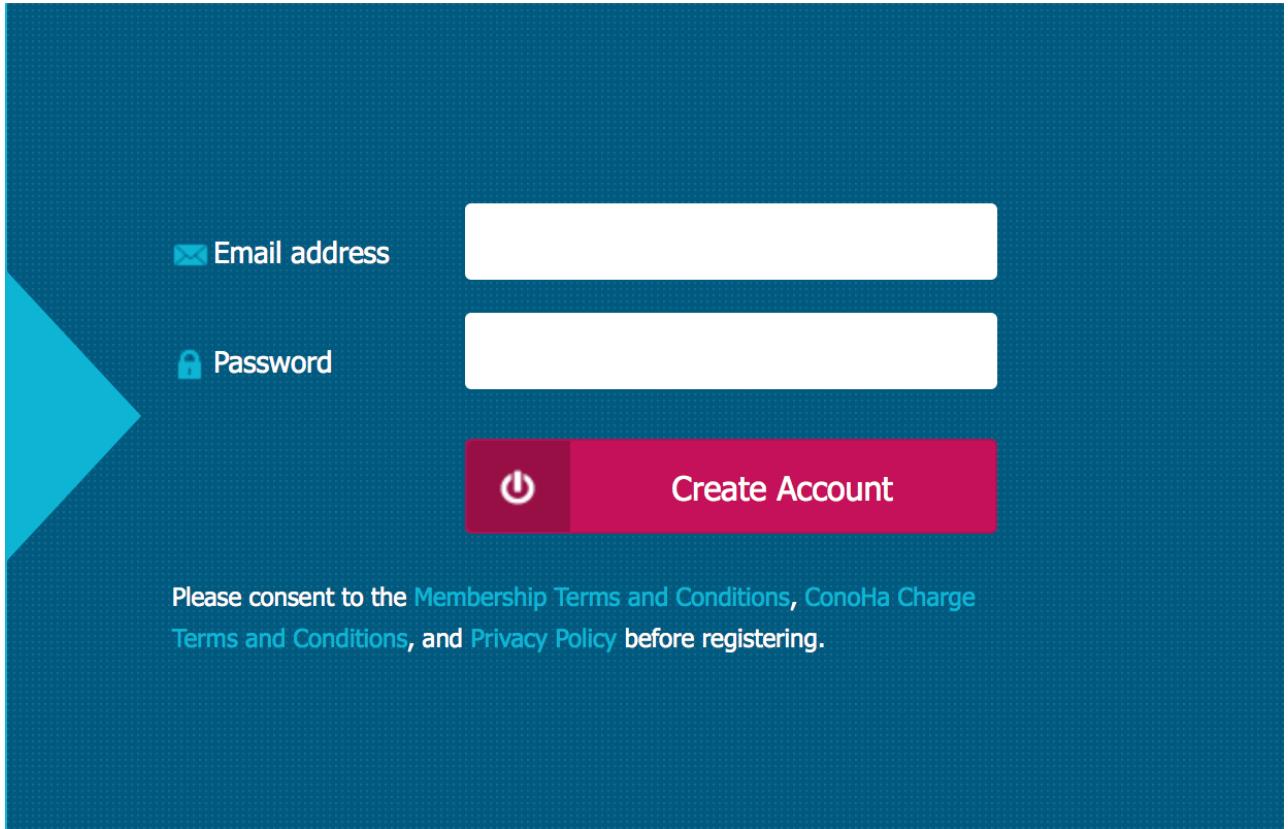
> もっと見る

[会員規約](#) | [ConoHaチャージ利用規約](#) | [特定商取引法に基づく表記](#) | [資金決済法に基づく表示](#) | [プライバシーポリシー](#) | [会社概要](#) | [サービス品質保証制度（SLA）](#)

Copyright (c) 2015 GMO Internet, Inc. All Rights Reserved.

萌萌哒的官网！看不懂日文的同学不要着急，首先我们有谷歌翻译，再不济官网也提供了英文页面，我们可以换成英文的，一下子熟悉多了。

第一步嘛，当然是注册啦。



注册成功后会发送一封邮件到你的邮箱，里面会告诉你登录用的ID（不使用邮箱登陆哟）。

この度はConoHaのアカウント登録のお申込みありがとうございます。  
お客様のConoHaアカウントをご案内します。

---

ConoHaアカウント : 7374072

---

上のConoHaアカウントと、登録時にご入力されたパスワードを使って  
コントロールパネルにログインしてください。

URL: <https://cp.conoha.jp/Login.aspx>

ログイン後は、画面の指示にしたがつて

- ・アカウント情報の登録
- ・お支払い情報の登録
- ・電話番号の認証

を済ますだけで、VPSのご利用が可能になります。  
ご不明な点は下記ヘルプページをご参照ください。

然后登录即可，接下来才是完善各种信息。

第一次登陆时会提示你添加一台VPS，选默认的，免费一个月，创建过程中会一步一步引导你完善各种信息，记住，电话号码请填写自己的，需要打电话验证哟，在日语说完后输入验证码就可以了（完全听不懂说什么啊）。

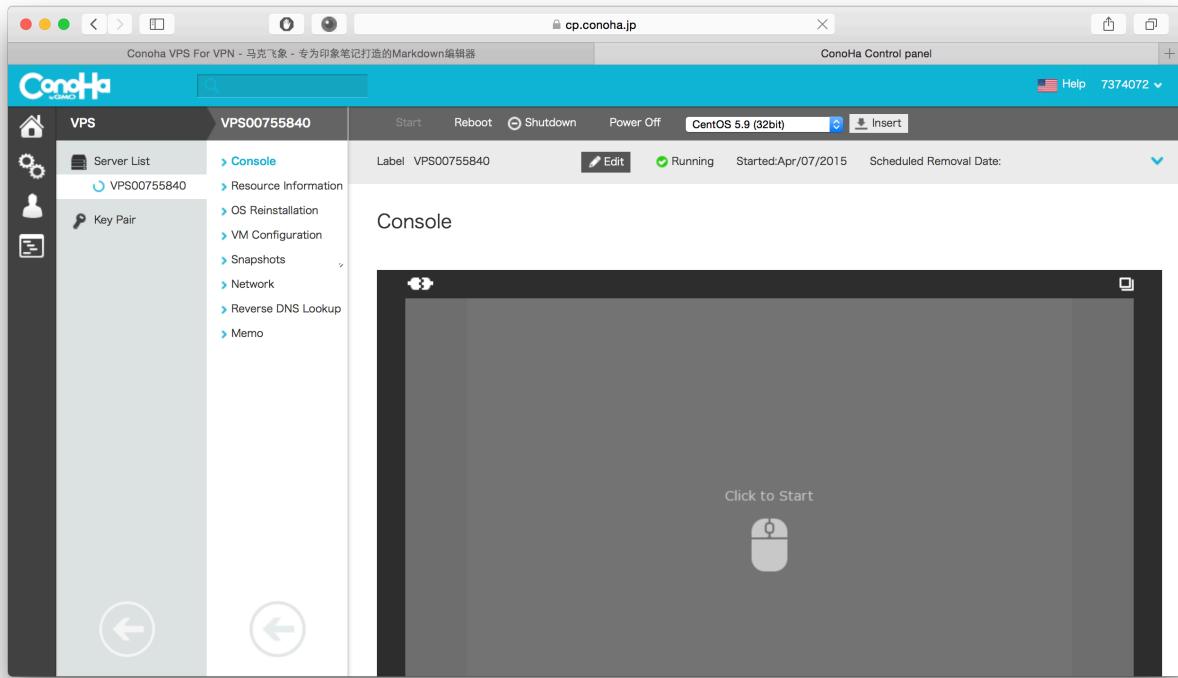
这是Dashboard界面：

The screenshot shows the Conoha Control Panel dashboard. At the top, there's a header bar with the URL "cp.conoha.jp", the title "Conoha Control panel", and a user ID "7374072". Below the header is a sidebar with icons for Home, Settings, User, and Help. The main content area has tabs for Notices, Failure, Maintenance, and History. Under Notices, there's a list of recent events in Japanese. Below that is a section titled "Services" with four categories: VPS Hosting (1), Additional IP addresses (0), Local Network (0), and Object Storage (0). A large blue button at the bottom right says "Create New VPS".

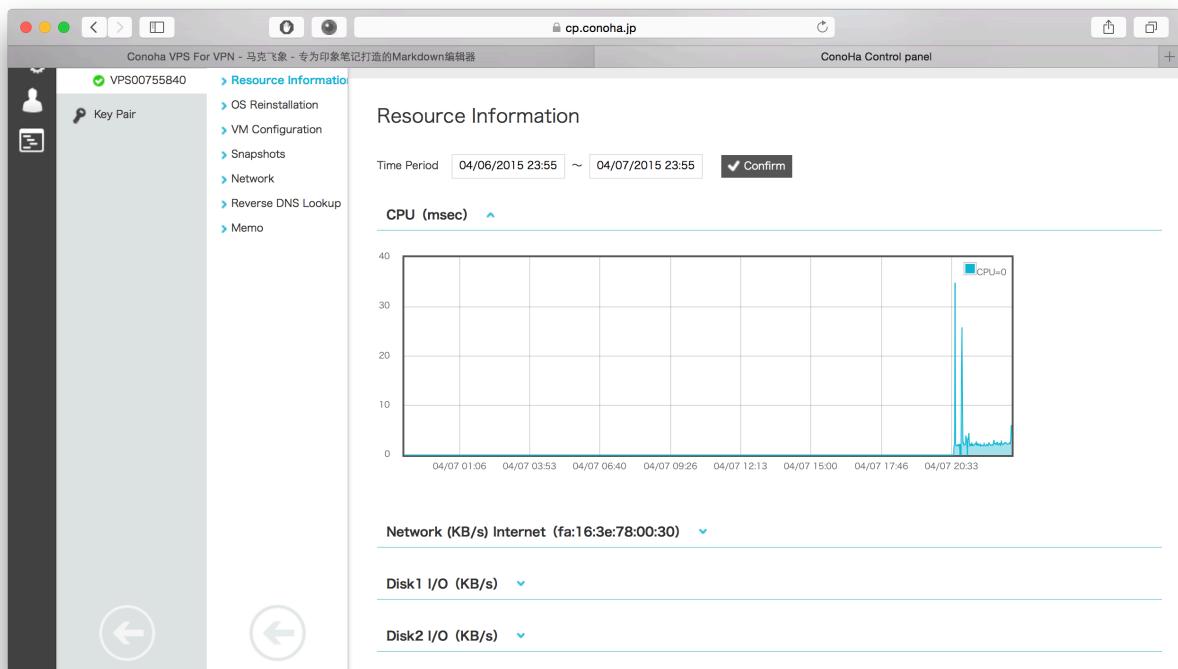
小齿轮是Services，可以看到有关VPS的相关信息：

The screenshot shows the Conoha Control Panel VPS service list. The sidebar is open with "VPS" selected. The main area shows a table for "Server List" with one entry: "VPS00755840". The table columns are: Label, Service status, Service ID, Plan, Created, Scheduled Removal Date, and Payment Interval. The data for the entry is: VPS00755840, In operation, VPS00755840, 1GB Memory, Apr/07/2015 21:37, and 1month. There are buttons for Start, Reboot, Shutdown, Power Off, Delete, and Undo Delete.

Key Pair里面是ssh登陆要用到的密匙，需要下下来。我们先点击Label栏里的才建好的VPS主机，会看到下图：



一个强大的web端管理界面，展示了非常详细的信息，你可以看到内存状态，网络状况，CPU，I/O读取，快照，在线终端，重装系统等等。



在最下面，则告诉了我们如何连接ssh，以及如何上传iso文件装自定义的系统。



由于一开始是默认关闭使用用户名密码ssh登录的，所以我们只用使用密匙登陆。

最后要说的就是侧边栏第四个是在线log日志，可以看到最近的登陆信息：

The screenshot shows the "Operation Logs" section of the Conoha Control panel. The logs table has columns for Date, Access point, Category, Label, and Processed. The logs entries are:

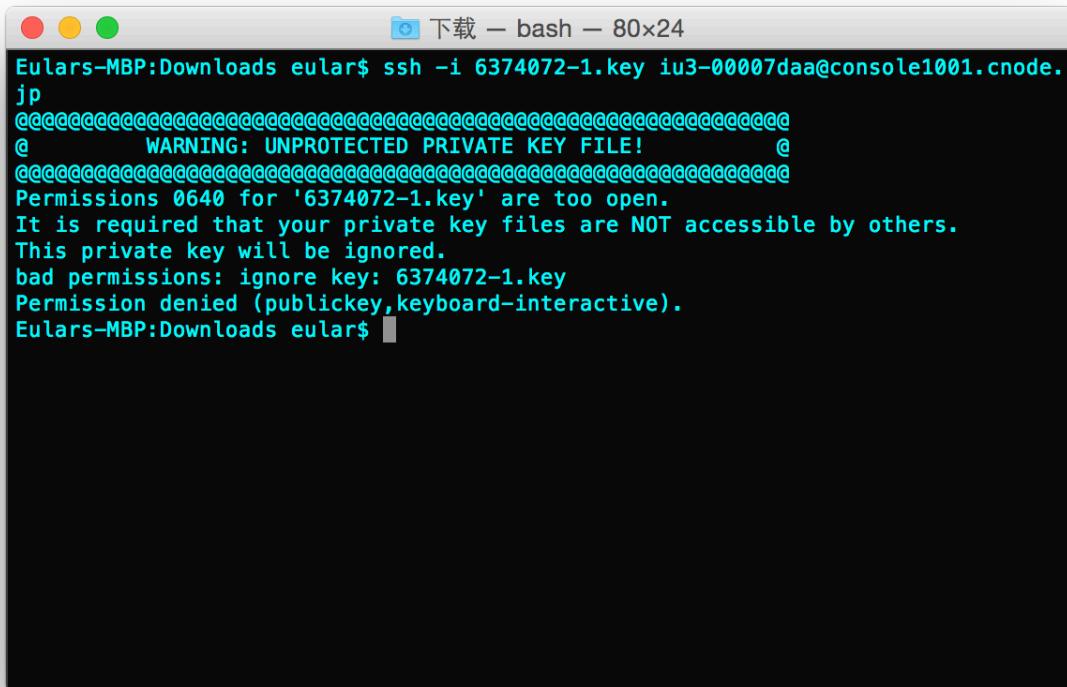
Date	Access point	Category	Label	Processed
Apr/07/2015 23:32	218.197.226.134	Control panel	Account	Logged in with ConoHa account
Apr/07/2015 21:55	218.197.226.134	Control panel	Account	SSH Key Pair downloaded [KeyName=6374072-1]
Apr/07/2015 21:35	218.197.226.134	Control panel	Account	VPS has been added[ServiceID=VPS00755840]
Apr/07/2015 21:25	218.197.226.134	Control panel	Account	Account information has been changed
Apr/07/2015 21:25	218.197.226.134	Control panel	Account	Updated payment method
Apr/07/2015 21:21	218.197.226.134	Control panel	Account	Logged in with ConoHa account

Showing 1 to 6 of 6 entries  
Previous < Next >

## SSH登陆

因为密匙已经下载好了，我们先用密匙登陆。

```
ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
```



```
Eulars-MBP:Downloads eular$ ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
@ WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0640 for '6374072-1.key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: 6374072-1.key
Permission denied (publickey,keyboard-interactive).
Eulars-MBP:Downloads eular$
```

你发现会出一个错误，是因为ssh的安全机制导致的。所以我们先要

```
chmod 600 6374072-1.key
```

然后再登陆就可以了：

```
ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
```

```
Eulars-MBP:Downloads eular$ ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
@@@@@WARNING: UNPROTECTED PRIVATE KEY FILE! @@@
Permissions 0640 for '6374072-1.key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: 6374072-1.key
Permission denied (publickey,keyboard-interactive).
Eulars-MBP:Downloads eular$ chmod 600 6374072-1.key
Eulars-MBP:Downloads eular$ ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
[Enter '^Ec?' for help]

CentOS release 6.5 (Final)
Kernel 2.6.32-431.17.1.el6.x86_64 on an x86_64

v157-7-50-137.z1d19.static.cnode.jp login: root
Password:
Last login: Tue Apr  7 22:13:41 on ttys0
[root@v157-7-50-137 ~]#
```

登陆进去之后你就可以随便逛逛了，反正也没什么好玩的。还有更重要的事等着我们去做呢！

要干啥呢，当务之急是改成能用密码登录呀，用密匙多不方便啊，而且其他小伙伴要想登陆还要去生成新的密匙你看多麻烦啊，所以嘛，我们要开启密码登录。

首先，打开ssh登陆配置文件

```
vi /etc/ssh/sshd_config
```

然后，设置为密码登陆方式

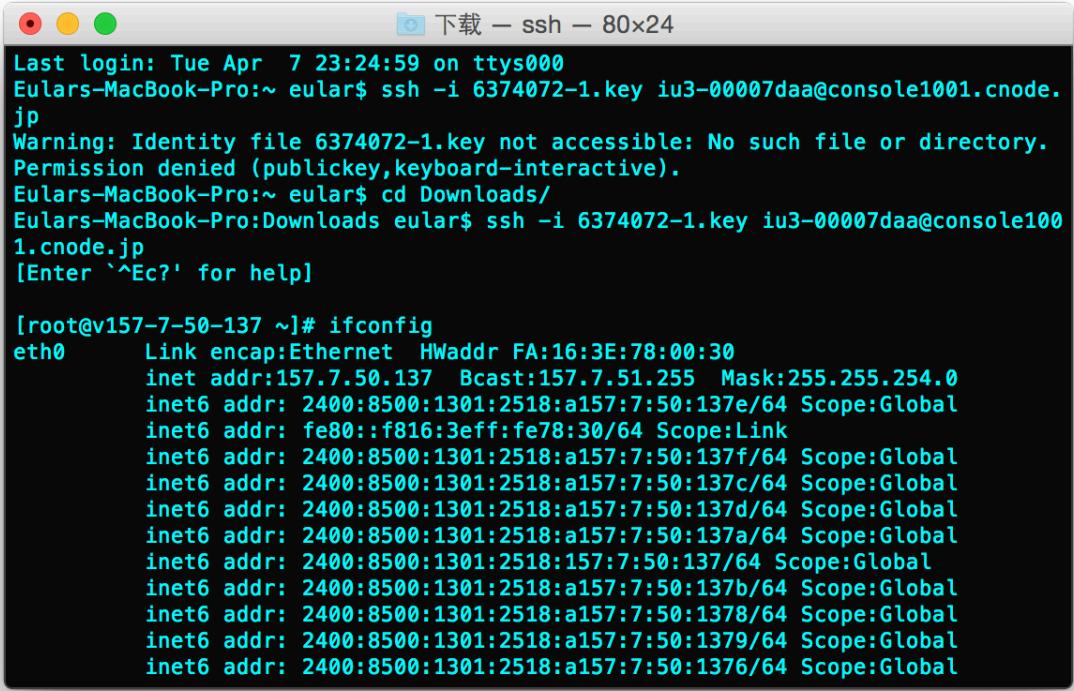
```
# 删除前面的#注释
PermitRootLogin yes

# no 改为 yes
PasswordAuthentication yes
```

最后，重启ssh服务或重启服务器

```
service sshd restart
```

让我们记住ip地址

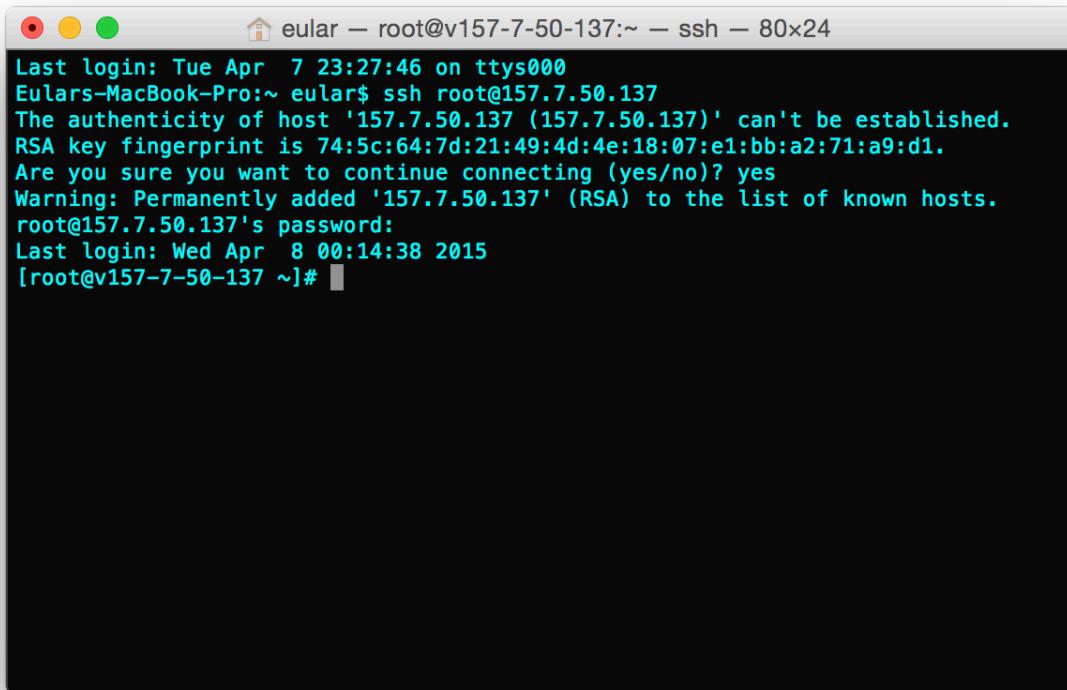


The screenshot shows a terminal window titled "下载 - ssh - 80x24". It displays the following text:

```
Last login: Tue Apr  7 23:24:59 on ttys000
Eulars-MacBook-Pro:~ eular$ ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
Warning: Identity file 6374072-1.key not accessible: No such file or directory.
Permission denied (publickey,keyboard-interactive).
Eulars-MacBook-Pro:~ eular$ cd Downloads/
Eulars-MacBook-Pro:Downloads eular$ ssh -i 6374072-1.key iu3-00007daa@console1001.cnode.jp
[Enter '^Ec?' for help]

[root@v157-7-50-137 ~]# ifconfig
eth0      Link encap:Ethernet HWaddr FA:16:3E:78:00:30
          inet addr:157.7.50.137 Bcast:157.7.51.255 Mask:255.255.254.0
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137e/64 Scope:Global
          inet6 addr: fe80::f816:3eff:fe78:30/64 Scope:Link
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137f/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137c/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137d/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137a/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:157:7:50:137/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:137b/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:1378/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:1379/64 Scope:Global
          inet6 addr: 2400:8500:1301:2518:a157:7:50:1376/64 Scope:Global
```

然后登陆上去



```
>Last login: Tue Apr  7 23:27:46 on ttys000
Eulars-MacBook-Pro:~ eular$ ssh root@157.7.50.137
The authenticity of host '157.7.50.137 (157.7.50.137)' can't be established.
RSA key fingerprint is 74:5c:64:7d:21:49:4d:4e:18:07:e1:bb:a2:71:a9:d1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '157.7.50.137' (RSA) to the list of known hosts.
root@157.7.50.137's password:
Last login: Wed Apr  8 00:14:38 2015
[root@v157-7-50-137 ~]#
```

成功了不是吗，嘻嘻！

## VPN隧道协议

到了这一步，接下来要干什么也就不言而喻了，不过首先需要了解一些基础知道。

VPN虚拟专用网络发展至今已经不在是一个单纯的经过加密的访问隧道了，它已经融合了访问控制、传输管理、加密、路由选择、可用性管理等多种功能，并在全球的信息安全体系中发挥着重要的作用。使用VPN代理时我们常常会碰到PPTP、L2TP、IPSec和SSLVPN等选项，到底这些词汇的意思是什么，它们之间有何区别呢？

首先，VPN的连接的认证和加密通常是基于PPP协议的。

### PPP (Point-to-Point Protocol) 协议

ppp协议是数据链路层的协议，与以太网（Ethernet）协议处于同一层。IP层的数据可以通过ppp链路传输。

一般如果电脑直接通过串口或并口或者其它方式连接在一起时，可以使用ppp协议形成一个数据链路层，供上层数据协议比如IP协议使用。

在现在这种直接连接的方式很少见了。但如果两台连接到互联网的计算机，将ppp数据包包裹到IP或其它协议中通过互联网发送时，就可以在两台计算机直接形成一条虚拟的连接，在这条连接上又可以发送其它协议的数据，如果发送的是IP协议，那就形成了一个虚拟的网络。

如果这个ppp连接两边是局域网，通过这条连接就将两个物理分隔的局域网连接在了一起，这应该就是虚拟局域网（VPN, Virtual Private Network）名称的由来。

由于PPP连接是不能自动完成的，需要在其它方式的控制下完成，这就是PPTP和L2TP协议的作用了。

## PPTP(Point-to-Point Tunneling Protocol)协议

是由包括微软和3Com等公司组成的PPTP论坛开发的一种点对点隧道协议，基于拨号使用的PPP协议使用PAP或CHAP之类的加密算法，或者使用Microsoft的点对点加密算法MPPE。其通过跨越基于TCP/IP的数据网络创建VPN实现了从远程客户端到专用企业服务器之间数据的安全传输。PPTP允许加密IP通讯，然后在要跨越公司IP网络或公共IP网络发送的IP头中对其进行封装。

PPTP客户端使用TCP协议向服务器发起连接请求，PPTP服务器默认使用TCP端口1723。连接建立后使用GRE协议发送数据，但PPTP使用的GRE协议对标准的GRE协议有改动。

为了保证传输中数据的保密性，需要加密，一般使用MPPE协议加密，而这个加密要求认证方式为MS-CHAP，所以pptp方式一般使用MS-CHAPv2认证和MPPE128加密。

但MPPE加密协议已经被发现有漏洞，可以被破解（见文后连接）。所以在需要数据绝对安全的情况下，不推荐使用PPTP方式的VPN。但目前破解对一般普通人还是不易完成，所以如果只是用来突破防火墙的包过滤，那还是可以使用的。

## L2TP (Layer 2 Tunneling Protocol) 协议

L2TP第 2 层隧道协议 (L2TP) 是IETF基于L2F (Cisco的第二层转发协议)开发的PPTP的后续版本。是一种工业标准 Internet 隧道协议，其可以为跨越面向数据包的媒体发送点到点协议 (PPP) 框架提供封装。PPTP和L2TP都使用PPP协议对数据进行封装，然后添加附加包头用于数据在互联网络上的传输。PPTP只能在两端点间建立单一隧道。L2TP支持在两端点间使用多隧道，用户可以针对不同的服务质量创建不同的隧道。L2TP可以提供隧道验证，而PPTP则不支持隧道验证。但是当L2TP 或PPTP与IPSEC共同使用时，可以由IPSEC提供隧道验证，不需要在第2层协议上验证隧道使用L2TP。PPTP要求互联网络为IP网络。L2TP只要求隧道媒介提供面向数据包的点对点的连接，L2TP可以在IP(使用UDP)，桢中继永久虚拟电路 (PVCs),X.25虚拟电路(VCs)或ATM VCs网络上使用。

L2TP使用UDP发送数据，默认使用端口1701。xl2tpd软件是L2TP的实现，xl2tpd在使用L2TP建立隧道后，通过隧道中的数据再使用PPP协议传输。虽然理论上L2TP隧道可以传输其它协议的数据，但目前xl2tpd只支持PPP协议。L2TP虽然有自己的认证方式，但方式有限，只有CHAP形式，不方便使用。PPP协议认证方式支持较多，比如PAP、CHAP、MS-CHAP、EAP等。所以xl2tpd中，推荐使用PPP协议完成认证。

## IPSEC (Internet Protocol Security) 协议

IPSec 隧道模式是封装、路由与解封装的整个过程。隧道将原始数据包隐藏(或封装)在新的数据包内部。该新的数据包可能会有新的寻址与路由信息，从而使其能够通过网络传播。隧道与数据保密性结合使用时，在网络上窃听通讯的人将无法获取原始数据包数据(以及原始的源和目标)。封装的数据包到达目的地后，会删除封装，原始数据包头用于将数据包路由到最终目的地。

ipsec首先通过ISAKMP (internet security association and key management protocol) 协议完成安全通路的建立。ISAKMP使用UDP协议进行，UDP端口号为500。安全通道建立后，加密后的数据通过ESP (Encapsulating Security Payload) 协议发送。

## SSLVPN

SSL协议提供了数据私密性、端点验证、信息完整性等特性。SSL协议由许多子协议组成，其中两个主要的子协议是握手协议和记录协议。握手协议允许服务器和客户端在应用协议传输第一个数据字节以前，彼此确认，协商一种加密算法和密码钥匙。在数据传输期间，记录协议利用握手协议生成的密钥加密和解密后来交换的数据。

SSL独立于应用，因此任何一个应用程序都可以享受它的安全性而不必理会执行细节。SSL置身于网络结构体系的传输层和应用层之间。此外，SSL本身就被几乎所有的Web浏览器支持。这意味着客户端不需要为了支持SSL连接安装额外的软件。这两个特征就是SSL能应用于VPN的关键点。

## 搭建VPN

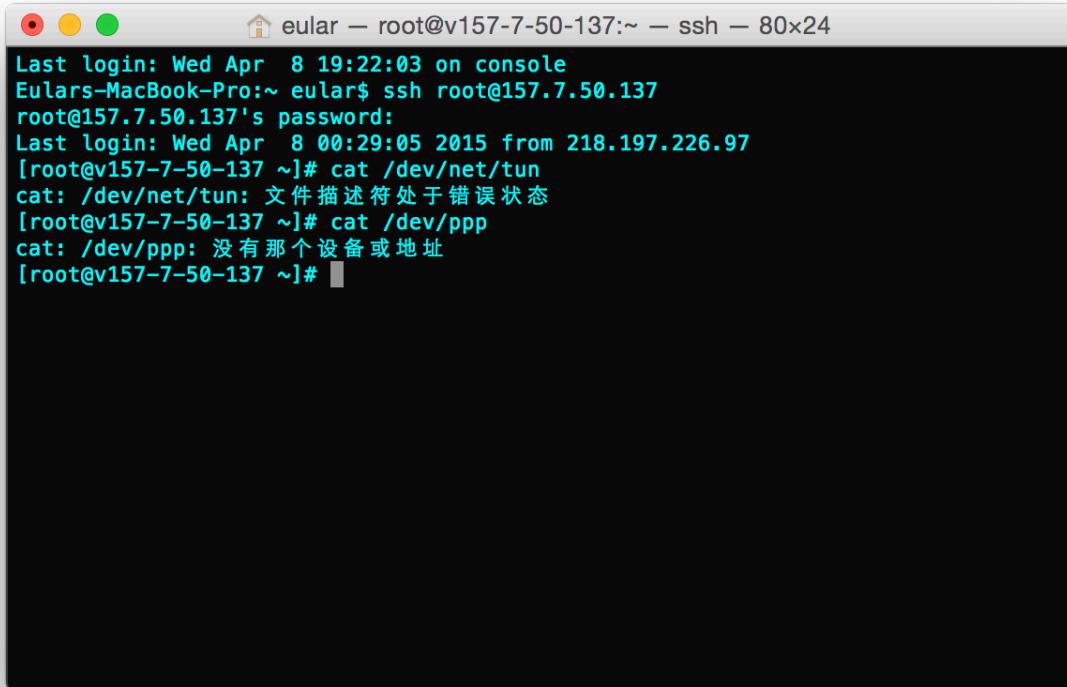
在Linux系统下PPTP形式的VPN通常使用pptpd软件实现，IPSEC/L2TP形式的VPN通常使用openSwan或者strongSwan软件配合xl2tpd实现。

下面，我们将在CentOS6.5 64bit上手把手带领大家搭建PPTP服务。

### 1. 检查一下该VPS可不可以做VPN

```
# cat /dev/net/tun
File descriptor in bad state
# cat /dev/ppp
No such device or address
```

出现如图所示的，就是可以的。



The screenshot shows a terminal window titled 'eular — root@v157-7-50-137:~ — ssh — 80x24'. The window contains the following text:

```
Last login: Wed Apr  8 19:22:03 on console
Eulars-MacBook-Pro:~ eular$ ssh root@157.7.50.137
root@157.7.50.137's password:
Last login: Wed Apr  8 00:29:05 2015 from 218.197.226.97
[root@v157-7-50-137 ~]# cat /dev/net/tun
cat: /dev/net/tun: 文件描述符处于错误状态
[root@v157-7-50-137 ~]# cat /dev/ppp
cat: /dev/ppp: 没有那个设备或地址
[root@v157-7-50-137 ~]#
```

## 2. 获取pptp的安装脚本，并执行安装

```
 wget http://www.auvps.com/wp-
content/uploads/files/centos6_pptpd.sh
sh ./centos6_pptpd.sh
```

```
s6_pptpd.sh
--2015-04-08 23:18:39-- http://www.auvps.com/wp-content/uploads/files/centos6_pptpd.sh
正在解析主机 www.auvps.com... 133.242.132.240
正在连接 www.auvps.com|133.242.132.240|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：1335 (1.3K) [application/octet-stream]
正在保存至：“centos6_pptpd.sh”

100%[=====] 1,335 --.-K/s in 0s

2015-04-08 23:18:41 (127 MB/s) - 已保存 “centos6_pptpd.sh” [1335/1335]

[root@v157-7-50-137 ~]# sh ./centos6_pptpd.sh
Loaded plugins: fastestmirror, security
Setting up Remove Process
No Match for argument: pptpd
Determining fastest mirrors
epel/metalink | 5.8 kB     00:00
 * base: ftp.tsukuba.wide.ad.jp
 * epel: ftp.kddilabs.jp
 * extras: ftp.tsukuba.wide.ad.jp
 * updates: ftp.tsukuba.wide.ad.jp
base | 3.7 kB     00:00
```

安装速度那是相当的快，不到一分钟就好了。

```
正在连接 www.huzs.net|107.161.24.30|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：76180 (74K) [application/x-redhat-package-manager]
正在保存至：“pptpd-1.4.0-1.el6.x86_64.rpm”

100%[=====] 76,180 362K/s in 0.2s

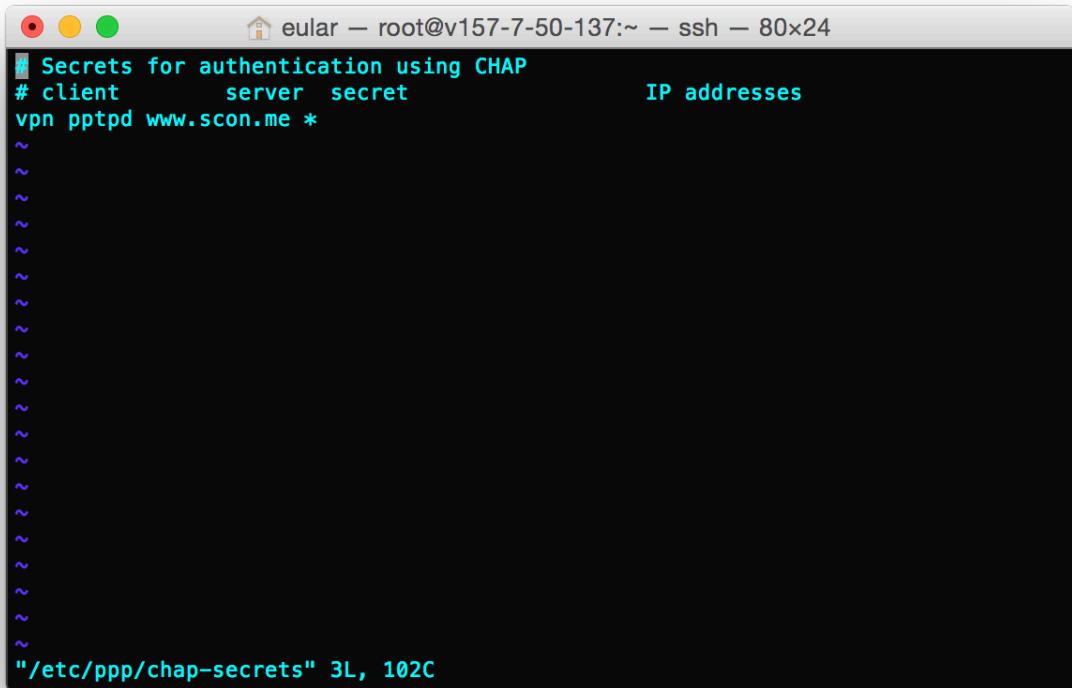
2015-04-08 23:21:03 (362 KB/s) - 已保存 “pptpd-1.4.0-1.el6.x86_64.rpm” [76180/76180]

warning: pptpd-1.4.0-1.el6.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 862a
cc42: NOKEY
Preparing... ################################################ [100%]
 1:pptpd ################################################ [100%]
mknod: "/dev/ppp": 文件已存在
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
iptables: Setting chains to policy ACCEPT: nat filter [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Unloading modules: [ OK ]
iptables: Applying firewall rules: [ OK ]
Starting pptpd: [ OK ]
??c????VPN PPTP?ÛR??n???, ?Û?????vpn ??www.scon.me
?Û?????l?????r????Û??? vi /etc/ppp/chap-secrets
[root@v157-7-50-137 ~]#
```

### 3. 查看和修改VPN帐号密码

```
vi /etc/ppp/chap-secrets
```

可以看到， 默认给你添加了一个 `vpn` 的用户， 密码为 `www.scon.me`。



```
# Secrets for authentication using CHAP
# client server secret           IP addresses
vpn pptpd www.scon.me *
```

这里你按照自己的意愿改就是了。

#### 4.开启IP转发

```
vi /etc/sysctl.conf
```

修改 `net.ipv4.ip_forward = 0` 为 `1`。

```
sysctl -p
```

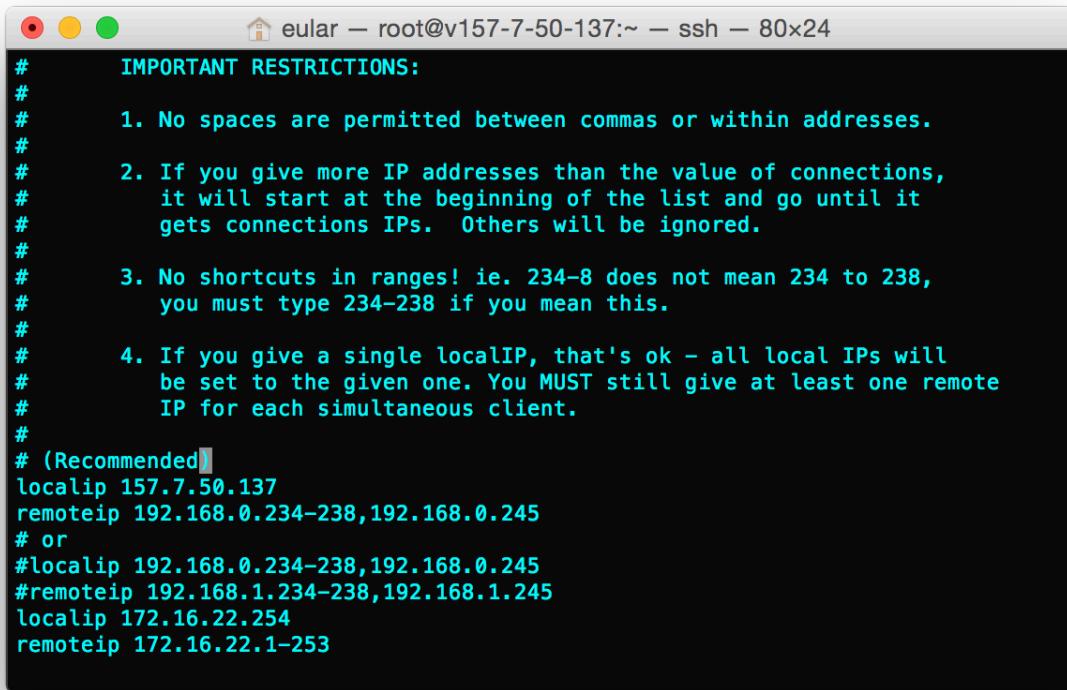
使之生效执行。

```
iptables: Setting chains to policy ACCEPT: nat filter      [  OK  ]
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Applying firewall rules:                         [  OK  ]
Starting pptpd:                                           [  OK  ]
??c?????VPN PPTP?Ù?n???, ?Ù?????vpn    ???最 www.scon.me
??l?????????z???vi /etc/ppp/chap-secrets
[root@v157-7-50-137 ~]# vi /etc/ppp/chap-secrets
[root@v157-7-50-137 ~]# vi /etc/sysctl.conf
[root@v157-7-50-137 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmmax = 68719476736
kernel.shmall = 4294967296
[root@v157-7-50-137 ~]# $
```

## 5. 设置pptp

```
vi /etc/pptpd.conf
```

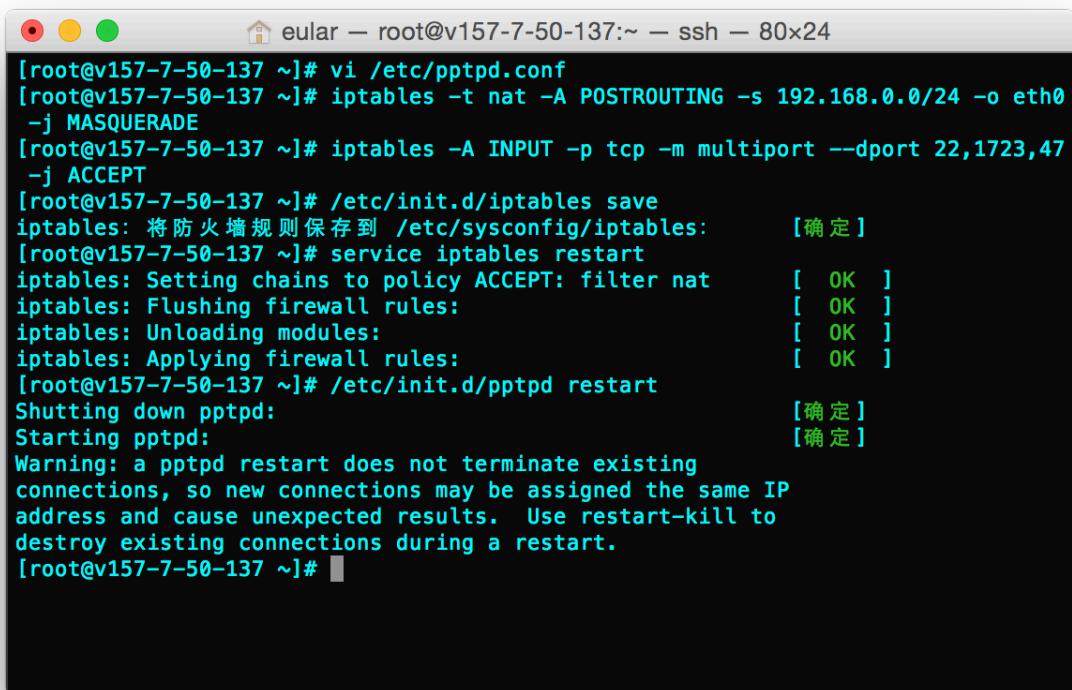
取消注释 `localip` 和 `remoteip` 并编辑，其中 `localip` 表示服务器的IP地址，`remoteip` 表示客户端连到服务器上将会被分配的IP地址范围。如图所示：



```
# IMPORTANT RESTRICTIONS:  
#  
# 1. No spaces are permitted between commas or within addresses.  
#  
# 2. If you give more IP addresses than the value of connections,  
#     it will start at the beginning of the list and go until it  
#     gets connections IPs. Others will be ignored.  
#  
# 3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,  
#     you must type 234-238 if you mean this.  
#  
# 4. If you give a single localIP, that's ok - all local IPs will  
#     be set to the given one. You MUST still give at least one remote  
#     IP for each simultaneous client.  
#  
# (Recommended)  
localip 157.7.50.137  
remoteip 192.168.0.234-238,192.168.0.245  
# or  
#localip 192.168.0.234-238,192.168.0.245  
#remoteip 192.168.1.234-238,192.168.1.245  
localip 172.16.22.254  
remoteip 172.16.22.1-253
```

## 6.最后几步

```
# 设置网段  
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth0 -j  
MASQUERADE  
# 设置开放端口  
iptables -A INPUT -p tcp -m multiport --dport 1723,47 -j ACCEPT  
# 保存配置  
/etc/init.d/iptables save  
# 重启防火墙  
service iptables restart  
# 重启PPTP服务  
/etc/init.d/pptpd restart  
# 注意查水表
```

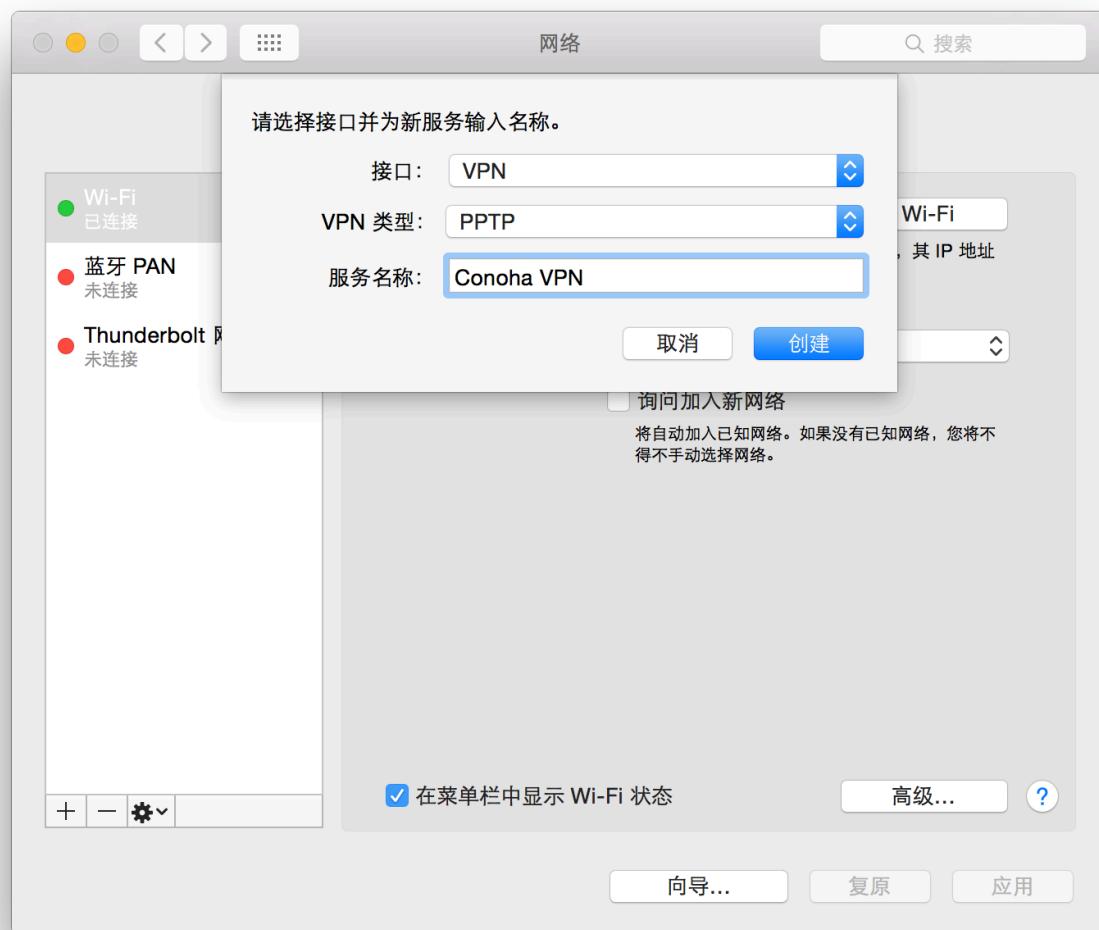


```
eular — root@v157-7-50-137:~ — ssh — 80x24
[root@v157-7-50-137 ~]# vi /etc/pptpd.conf
[root@v157-7-50-137 ~]# iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth0
-j MASQUERADE
[root@v157-7-50-137 ~]# iptables -A INPUT -p tcp -m multiport --dport 22,1723,47
-j ACCEPT
[root@v157-7-50-137 ~]# /etc/init.d/iptables save
iptables: 将防火墙规则保存到 /etc/sysconfig/iptables: [确定]
[root@v157-7-50-137 ~]# service iptables restart
iptables: Setting chains to policy ACCEPT: filter nat [OK]
iptables: Flushing firewall rules: [OK]
iptables: Unloading modules: [OK]
iptables: Applying firewall rules: [OK]
[root@v157-7-50-137 ~]# /etc/init.d/pptpd restart
Shutting down pptpd: [确定]
Starting pptpd: [确定]
Warning: a pptpd restart does not terminate existing
connections, so new connections may be assigned the same IP
address and cause unexpected results. Use restart-kill to
destroy existing connections during a restart.
[root@v157-7-50-137 ~]#
```

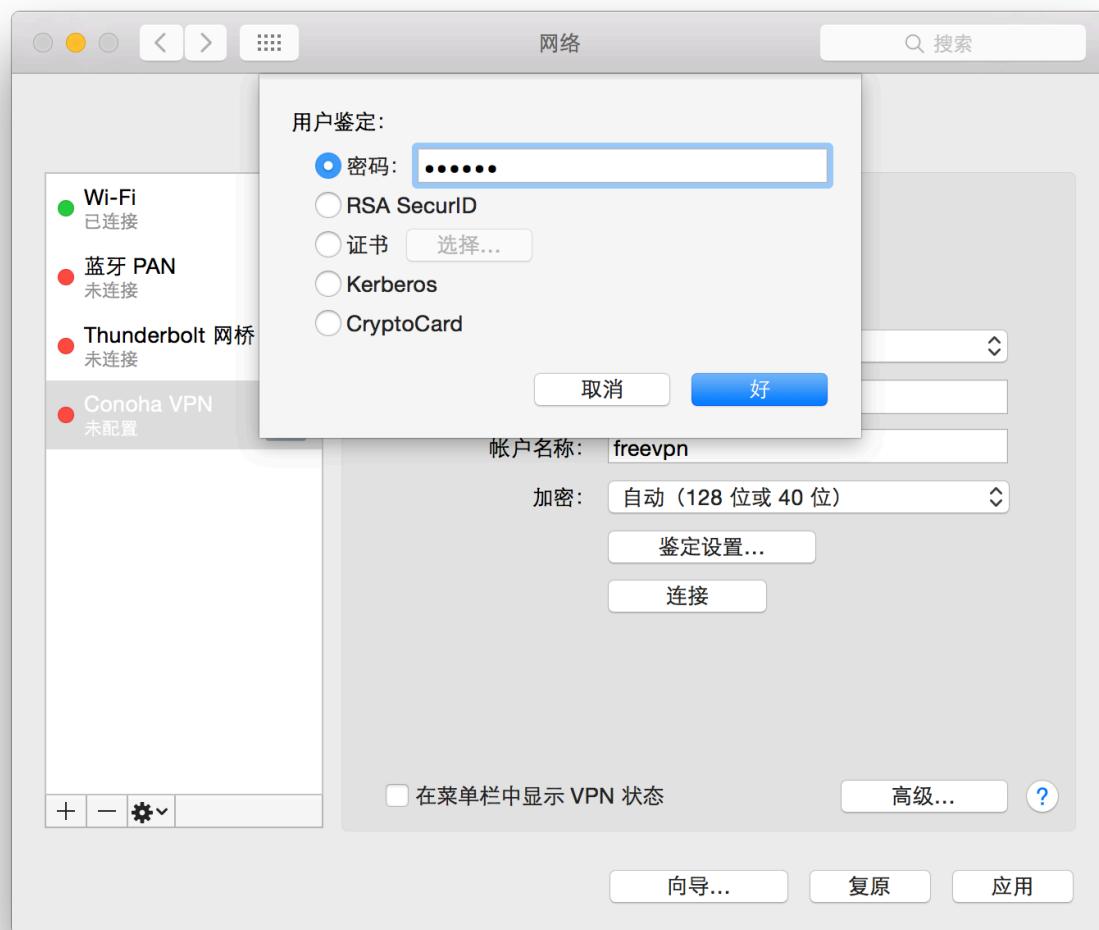
接下来连上去测试一把。

## Connect To VPN

打开设置->网路->点击+号新建一个VPN连接：



填好服务器地址和账户名称后点击鉴定设置，然后输入VPN密码：



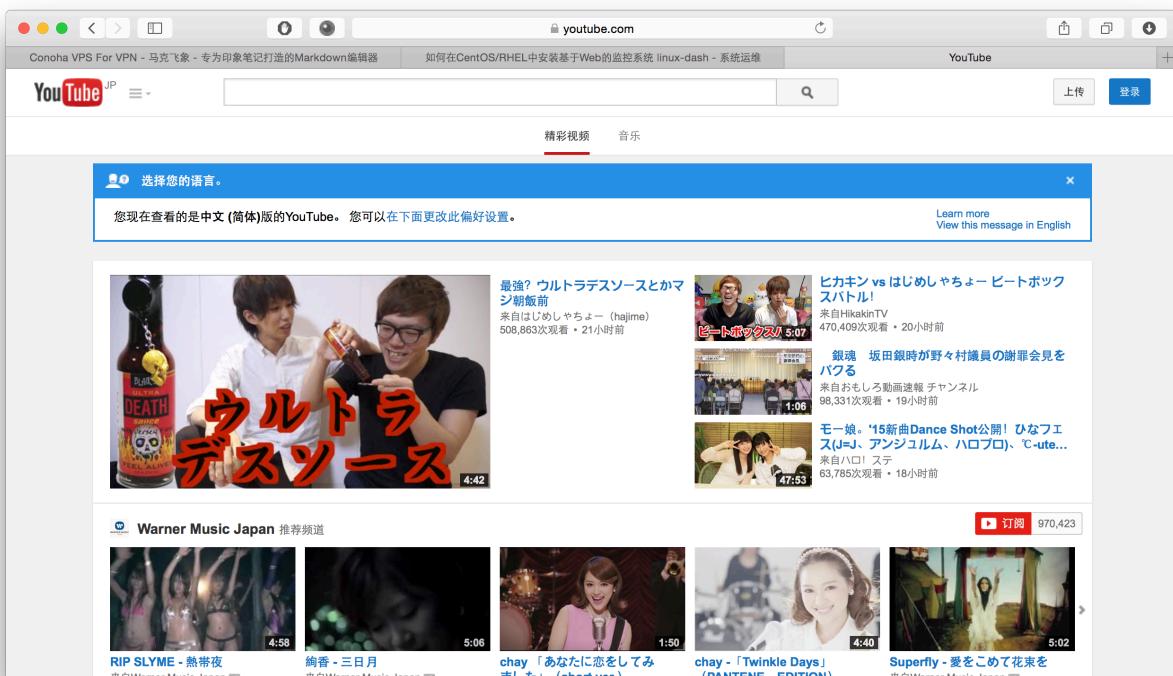
接着点击高级，勾选 [通过VPN连接发送所有流量](#) 选项：



最后就开始连接了！



不错，上youtube看视频速度那是杠杠的！



本着分享的精神，在这里我提供免费的账号密码，欢迎大家使用：

```
user: freevpn  
password: 123456
```

## Linux-dash

VPN都有了，接下来你还想干啥？鄙人表示Conoha官网连接速度简直是慢到不能忍，而且上面的dashborad丑到了极致。所以，接下来我们将要安装自己的dashborad——Linux-dash。

Linux-dash是一款为Linux设计的基于web的轻量级监控面板。这个程序会实时显示各种不同的系统属性，比如CPU负载、RAM使用率、磁盘使用率、网速、网络连接、RX/TX带宽、登录用户、运行的进程等等。它不会存储长期的统计。因为它没有后端数据库。

由于机器上自带了Apache，感觉不习惯，而这里所使用的web服务器是Nginx。

### 1. 安装Nginx和php-fpm组件

```
yum install nginx  
yum install git php-common php-fpm
```

```
eular — root@v157-7-50-137:~ — ssh — 80x24
Verifying : php-cli-5.3.3-40.el6_6.x86_64 4/13
Verifying : php-5.3.3-40.el6_6.x86_64 5/13
Verifying : php-gd-5.3.3-40.el6_6.x86_64 6/13
Verifying : php-common-5.3.3-40.el6_6.x86_64 7/13
Verifying : php-xml-5.3.3-27.el6_5.x86_64 8/13
Verifying : php-cli-5.3.3-27.el6_5.x86_64 9/13
Verifying : php-5.3.3-27.el6_5.x86_64 10/13
Verifying : php-gd-5.3.3-27.el6_5.x86_64 11/13
Verifying : php-common-5.3.3-27.el6_5.x86_64 12/13
Verifying : php-pdo-5.3.3-27.el6_5.x86_64 13/13

Installed:
  php-fpm.x86_64 0:5.3.3-40.el6_6

Updated:
  php-common.x86_64 0:5.3.3-40.el6_6

Dependency Updated:
  php.x86_64 0:5.3.3-40.el6_6           php-cli.x86_64 0:5.3.3-40.el6_6
  php-gd.x86_64 0:5.3.3-40.el6_6         php-pdo.x86_64 0:5.3.3-40.el6_6
  php-xml.x86_64 0:5.3.3-40.el6_6

Complete!
[root@v157-7-50-137 ~]#
```

## 2.在nginx中配置Linux-dash

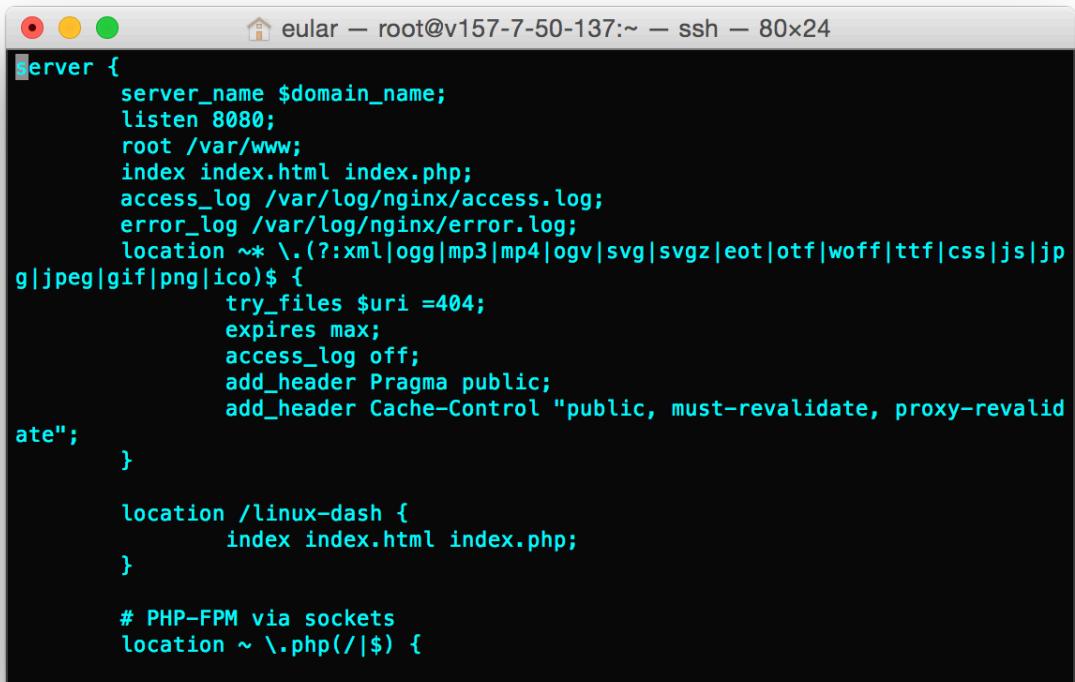
```
vi /etc/nginx/conf.d/linuxdash.conf
```

在这个空的文件中填入下面内容：

```
server {
    server_name $domain_name;
    listen 8080;
    root /var/www;
    index index.html index.php;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    location ~* \.
    (?::xml|ogg|mp3|mp4|ogv|svg|svgz|eot|otf|woff|ttf|css|js|jpg|jpeg
    |gif|png|ico)$ {
        try_files $uri =404;
        expires max;
        access_log off;
        add_header Pragma public;
        add_header Cache-Control "public, must-revalidate,
proxy-revalidate";
    }

    location /linux-dash {
        index index.html index.php;
    }

    # PHP-FPM via sockets
    location ~ \.php(/|$) {
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_split_path_info ^(.+?\.\php)(/.*)$;
        fastcgi_pass unix:/var/run/php-fpm.sock;
        if (!-f $document_root$fastcgi_script_name) {
            return 404;
        }
        try_files $uri $uri/ /index.php?$args;
        include fastcgi_params;
    }
}
```



```
server {
    server_name $domain_name;
    listen 8080;
    root /var/www;
    index index.html index.php;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    location ~* \.(?:xml|ogg|mp3|mp4|ogv|svg|svgz|eot|otf|woff|ttf|css|js|jpeg|gif|png|ico)$ {
        try_files $uri =404;
        expires max;
        access_log off;
        add_header Pragma public;
        add_header Cache-Control "public, must-revalidate, proxy-revalidate";
    }
    location /linux-dash {
        index index.html index.php;
    }
    # PHP-FPM via sockets
    location ~ \.php(/|$) {
```

### 3.配置php-fpm

```
vi /etc/php-fpm.d/www.conf
```

找到 `listen` , `user` 和 `group` 字段, 并修改为下:

```
...
listen = /var/run/php-fpm.sock
...
user = nginx
group = nginx
...
```

```

; Valid syntaxes are:
; 'ip.add.re.ss:port'      - to listen on a TCP socket to a specific address on
;                           a specific port;
; 'port'                  - to listen on a TCP socket to all addresses on a
;                           specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
;listen = 127.0.0.1:9000
listen = /var/run/php-fpm.sock

; Set listen(2) backlog. A value of '-1' means unlimited.
; Default Value: -1
;listen.backlog = -1

; List of ipv4 addresses of FastCGI clients which are allowed to connect.
; Equivalent to the FCGI_WEB_SERVER_ADDRS environment variable in the original
; PHP FCGI (5.2.2+). Makes sense only with a tcp listening socket. Each address
; must be separated by a comma. If this value is left blank, connections will be
; accepted from any ip address.
; Default Value: any
listen.allowed_clients = 127.0.0.1

; Set permissions for unix socket, if one is used. In Linux, read/write
-- INSERT --

```

```

; permissions must be set in order to allow connections from a web server. Many
; BSD-derived systems allow connections regardless of permissions.
; Default Values: user and group are set as the running user
;                   mode is set to 0666
;listen.owner = nobody
;listen.group = nobody
;listen.mode = 0666

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;       will be used.
; RPM: apache Choosed to be able to access some dir as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = nginix

; Choose how the process manager will control the number of child processes.
; Possible Values:
;   static  - a fixed number (pm.max_children) of child processes;
;   dynamic - the number of child processes are set dynamically based on the
;             following directives:
;               pm.max_children    - the maximum number of children that can
;                                 be alive at the same time.
;
```

#### 4. 下载并安装linux-dash

```
git clone https://github.com/afaqurk/linux-dash.git  
cp -r linux-dash/ /var/www/  
chown -R nginx:nginx /var/www
```

## 5.开启允许对外访问的 80 和 8080 端口

```
/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT  
/etc/rc.d/init.d/iptables save  
/etc/rc.d/init.d/iptables restart
```

查看端口是否已经开放，确认一下

```
/etc/init.d/iptables status
```

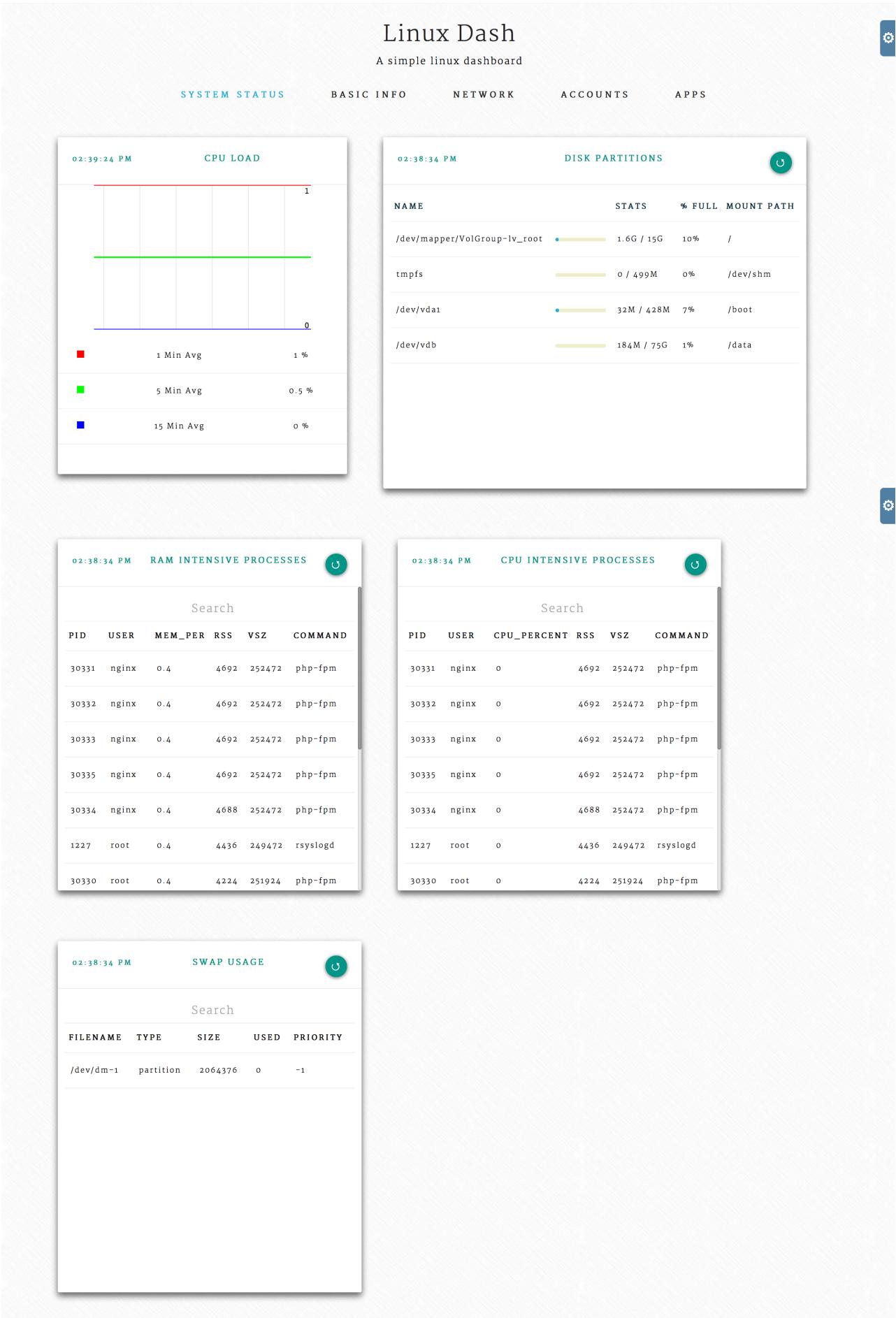
```
Eulars-MacBook-Pro:~ eular$ nmap -Pn 157.7.50.137  
Starting Nmap 6.47 ( http://nmap.org ) at 2015-04-09 14:40 CST  
Nmap scan report for v157-7-50-137.z1d19.static.cnode.jp (157.7.50.137)  
Host is up (0.15s latency).  
Not shown: 996 filtered ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
1723/tcp  open  pptp  
8080/tcp  open  http-proxy
```

(请无视我用nmap在外网扫下)

## 6.重启nginx和php-fpm并设为开机自动启动

```
service nginx restart  
service php-fpm restart  
chkconfig nginx on  
chkconfig php-fpm on
```

至此已经大功告成，在浏览器里输入 <http://<IP地址>:8080/linux-dash/> 来访问 Linux-dash。



界面一目了然，简洁清新，十分耐看。

# Last

Conoha的VPS只是免费一个月，一个月后你可以考虑考虑下图的价格，土豪请任性。

The screenshot shows a web browser window for the Conoha Control panel. The URL is https://www.conoha.jp/pricing. The main title is "VPS". Below it is a table comparing "スペック" (Specs) with "ひと月あたりの料金" (Monthly price). The table has two columns: "標準プラン" (Standard Plan) and "Windowsプラン" (Windows Plan). The rows represent different memory configurations: 1GB, 2GB, 4GB, 8GB, and 16GB. The prices range from 930円 to 17,400円. Below the table is a section titled "その他" (Others) with a table for additional services: "追加IPアドレス" (Additional IP Address) at 200円/1IP and "オブジェクトストレージ" (Object Storage) at 450円/100GB.

スペック			ひと月あたりの料金	
メモリ	CPU	HDD	標準プラン	Windowsプラン
1GB	2コア	100GB	930円	1,930円
2GB	3コア	200GB	1,400円	3,700円
4GB	4コア	400GB	3,780円	5,780円
8GB	6コア	800GB	7,590円	9,590円
16GB	10コア	1TB	15,400円	17,400円

サービス	ひと月あたりの料金
追加IPアドレス ※1	200円/1IP
オブジェクトストレージ ※2	450円/100GB

※1 2、4、8、16個単位でお申込みいただけます。  
※2 REST API (OpenStack Swift) 経由でご利用いただける容量無制限のファイル保存サービスで

总之，兄弟我只能帮到这里了，接下来就是广告时间了。

=====

欢迎关注我的微信公众号



微信资助

微信扫一扫 支付



微信号: Urinxs

支付宝打赏



好人一生平安

## Reference

- [0]. [虚拟专用服务器 - 维基百科, 自由的百科全书](#)
- [1]. [亚马逊AWS中国版,Dell云,微软Windows Azure,阿里云ECS免费VPS主机试用](#)
- [2]. [CentOS SSH密钥登陆改为密码登陆 \(Conoha\)](#)
- [3]. [conoha vps搭建pptp vpn](#)
- [4]. [利用Conoha.jp的VPS主机安装PPTP](#)
- [5]. [PPTP和IPSEC/L2TP的VPN笔记](#)
- [6]. [VPN隧道协议PPTP、L2TP、IPSec和SSLVPN介绍及区别](#)
- [7]. [如何在CentOS/RHEL中安装基于Web的监控系统 linux-dash](#)