# Linux 的網路連線設定

Linux TCP/IP的網路連線設定最少需知道三項設定值分別是：IP、NETMASK、Default Gateway，網路的設定必須使用root 系統管理帳號透過兩個工具，分別是：ifconfig、route。而 Debian GNU/Linux 網路卡設定檔案也只允許root系統管理者帳號撰寫，檔案位於/etc/network/interfaces。Debian GNU/Linux 的網路卡設定檔為/etc/network/interfaces

|  |
| --- |
| ifconfig 語法為: ifconfig eth0 $IP netmask $NETMASK  如果IP = 10.1.1.30，NETMASK = 255.255.255.0  則命令為 $ ifconfig eth0 10.1.1.30 netmask 255.255.255.0  route設定default gateway語法為: route add default gw $GATEWAY  如果 GATEWAY = 10.1.1.1  則命令為: $ route add default gw 10.1.1.1  請試試看以下命令 :  $ ifconfig # 列出目前的網路卡設定狀態  $ route -n # 秀出目前的 route table 表 |

單純以手動的方式使用 ifconfig及route於系統重開機時就會還原為初始值，必須透過撰寫網路設定檔，於開機的時候初始化設定。不同公司出品的Linux都會有所不同，相同的都是透過ifconfig及route來設定網路，不過設定檔格式及位置會不同。Debian GNU/Linux網路卡設定檔位於「/etc/network/interfaces」。其格式如下。

|  |
| --- |
| $ nano /etc/network/interfaces  # A "#" character in the very first column makes the rest of the line be ignored. Blank lines are ignored.  # Lines may be indented freely. A "\" character at the very end of the line indicates the next line should be  # treated as a continuation of the current one.  # We always want the loopback interface.  *auto lo*  *iface lo inet loopback*  # To setup eth0 to dhcp, enter:  *auto eth0*  *iface eth0 inet dhcp*  # An example ethernet card setup: (broadcast and gateway are optional)  *auto eth0*  *iface eth0 inet static*  *address 192.168.0.42*  *network 192.168.0.0*  *netmask 255.255.255.0*  *broadcast 192.168.0.255*  *gateway 192.168.0.1*  # A more complicated ethernet setup with a single ethernet card with two interfaces.  *auto eth0 eth0:1*  *iface eth0 inet static*  *address 192.168.0.100*  *network 192.168.0.0*  *netmask 255.255.255.0*  *broadcast 192.168.0.255*  *gateway 192.168.0.1*  *iface eth0:1 inet static*  *address 192.168.0.200*  *network 192.168.0.0*  *netmask 255.255.255.0*  # Vlan interface definitions exist of the vlan interface name, and an optional ‘raw-device’ parameter.  # Vlan interfaces are numbered 1 to 4095. You have the option to have interface names zero-padded to  # 4 numbers, or just the plain digits without leading zero.  # 安裝套件 *#sudo apt-get install vlan*  *auto eth1 iface eth1 inet static address 192.168.0.123 netmask 255.255.255.0*  *auto eth1.50 iface eth1.50 inet static address 192.168.50.123 netmask 255.255.255.0 vlan-raw-device eth1*  *auto eth1.20 iface eth1.20 inet static address 192.168.20.123 netmask 255.255.255.0 vlan-raw-device eth1*  $ /etc/init.d/networking stop # 停掉網路裝置  $ ifconfig # 你會發現空空如也  $ /etc/init.d/networking start # 啟動網路裝置  $ ifconfig # 網路裝置回覆  $ route -n # 網路裝置回覆  Different command to start/stop the service:  $ *service networking stop* $ *service networking start* |

### Using *vconfig* command

The vconfig program allows you to create and remove vlan-devices on a vlan enabled kernel. Vlan-devices are virtual ethernet devices which represents the virtual Lans on the physical lan.

Add VLAN ID 5 with follwing command for eth0:

*# vconfig add eth0 5*

The **vconfig add** command creates a vlan-device on eth0 which result into eth0.5 interface. You can use normal ifconfig command to see device information:

*# ifconfig eth0.5*  
Use ifconfig to assign IP address to vlan interface :

*# ifconfig eth0.5 192.168.1.100 netmask 255.255.255.0 up*  
Get detailed information about VLAN interface:

*# cat /proc/net/vlan/eth0.5*  
If you wish to delete VLAN interface delete command:

*# ifconfig eth0.5 down*

*# vconfig rem eth0.5*  
Although the commands described above can configure VLAN, all settings are gone when rebooting. Write a script file and let the system configure the vlan parameters after rebooting.

1. Write a shell script:

*# touch /opt/script/set\_vlan.sh*

*# nano set\_vlan.sh*

*#!/bin/sh*

*vconfig add eth1 50*

*ifconfig eth1.50 192.168.50.133 netmask 255.255.255.0*

1. Modify the */etc/rc.local* so that system can run the script after rebooting:

*# nano /etc/rc.local*

Add the */opt/script/set\_vlan.sh* above the ‘*exit 0*’ command.

Caution: Although modifying *rc.local* or */etc/network/interfaces* can both configure VLAN, it is more stable to setup VLAN by writing the parameters in */etc/network/interfaces*.

**19.2 Client 端的設定**

**19.2.1 相關設定檔**

主機名稱對應到IP是透過DNS架構！那麼這兩種方法分別使用什麼設定檔？先來談一談幾個設定檔吧！

* /etc/hosts：這個是最早的 hostname 對應 IP 的檔案；
* /etc/resolv.conf：這個重要！就是ISP的DNS伺服器IP記錄處；
* /etc/nsswitch.conf：這個檔案則是在『決定』先要使用/etc/hosts還是/etc/resolv.conf的設定！

Linux的預設主機名稱與 IP的對應搜尋都以 /etc/hosts 為優先，查看/etc/nsswitch.conf，找到hosts的項目：

|  |
| --- |
| [root@www ~]# **vim /etc/nsswitch.conf**  hosts: files dns |

上面的files是使用/etc/hosts，而最後的dns則是使用/etc/resolv.conf的DNS伺服器來進行搜尋！可先以/etc/hosts設定IP對應！當然也可以調換過來，不過總是/etc/hosts比較簡單，所以將他擺在前面比較好！假設在台灣使用hinet的168.95.1.1這部DNS伺服器，所以應該這樣寫：

|  |
| --- |
| [root@www ~]# **vim /etc/resolv.conf**  nameserver 168.95.1.1  nameserver 139.175.10.20 |

DNS伺服器的IP可設多個，當第一部(照設定順序)DNS掛點時，用戶端可用第二部(139.175.10.20)查詢。建議至少填兩部DNS伺服器，在網路正常使用情況下，**只有第一部DNS伺服器會用來查詢**，其他只在第一部出問題時才會被使用。

**DNS的正、反解查詢指令： nslookup**

|  |
| --- |
| [root@www ~]# **nslookup [FQDN] [server]**  [root@www ~]# **nslookup**  選項與參數：  1. 可以直接在 nslookup 加上待查詢的主機名稱或者是 IP ，[server] 可有可無；  2. 如果在 nslookup 後面沒有加上任何主機名稱或 IP，那將進入 nslookup的查詢功能在nslookup 的查詢功能當中，可以輸入其他參數來進行特殊查詢，例如：  set type=any ：列出所有的資訊『正解方面設定檔』  set type=mx ：列出與 mx 相關的資訊！  # 1. 直接搜尋 mail.ksu.edu.tw 的 IP 資訊  [root@www ~]# **nslookup mail.ksu.edu.tw**  Server: 168.95.1.1  Address: 168.95.1.1#53 <==還是請特別注意 DNS 的 IP 是否正確！  Non-authoritative answer:  Name: mail.ksu.edu.tw  Address: 120.114.100.20 <==回報 IP 給你囉！ |

nslookup可單純的將hostname與IP對應列出，不過還是會將查詢的DNS主機的IP列出來的！

# Linux 如何指定 DNS server

所有的Linux包含FreeBSD指定DNS server的方法均相同，全部都是透過撰寫「/etc/resolv.conf」，來指定DNS server 以及設定search domain。(設定 search domain 的格式 FreeBSD 與 Linux 有些微差異)

|  |
| --- |
| $ edit /etc/resolv.conf  search kh.coventive.com coventive.com # 設定 search domain  nameserver 211.20.240.115 # 設定第一台 DNS  nameserver 192.168.22.4 # 設定第二台 DNS |