

GESTIONE DELLA RETE SU DEBIAN E DERIVATE

Piccola promemoria su come configurare la rete su debian e derivate. Non vi e' alcuna completezza, si tratta solo di una somma di appunti, per risalire rapidamente alle informazioni utili. Non verra' analizzato l'indirizzamento IPV6 . Inoltre si fa riferimento solamente alla connessione via cavo. Non viene analizzata la connessione tramite onde radio (wireless).

- I nomi dei dispositivi vengono risolti in base alle priorit  dettate dal file /etc/nsswitch.conf ecco tale file dopo un'installazione standard di debian 7

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:    compat
group:     compat
shadow:    compat

hosts:     files mdns4_minimal [NOTFOUND=return] dns mdns4
networks:  files

protocols: db files
services:  db files
ethers:    db files
rpc:       db files

netgroup:  nis
```

- il file /etc/resolv.conf si occupa di indicare quali server dns contattare, ed in quale ordine

- se servono delle mappature statiche tra nome-host ed indirizzo ip allora e' possibile utilizzare il file /etc/hosts. Anzi, e' meglio controllarne il contenuto prima di iniziare la configurazione della rete. Ecco tale file dopo un'installazione standard di debian 7

```
127.0.0.1    localhost
127.0.1.1    debian7.veronese.lan  debian7

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

- le schede di rete vengono rinominate in base alle direttive del pacchetto udev. Appena esse vengono rilevate , il pacchetto udev esegue le direttive del file /etc/udev/rules.d/70-persistent-net.rules. Solitamente la scheda di rete prende il nome " eth0 " , e la seconda scheda di rete -se presente- prende il nome di " eth1 "
- [Quando si modifica il file delle regole "/etc/udev/rules.d/70-persistent-net.rules", si deve mantenere ciascuna regola su una singola riga e l'indirizzo MAC in lettere minuscole. Per esempio, se si trova in questo file una voce "FireWire device" e una "PCI device", probabilmente si vorr  chiamare "PCI

device "eth0" e configurarla come interfaccia di rete primaria]

- Nei sistemi desktop (non server) ci sono software di gestione delle impostazioni di rete come ad esempio NetworkManager o WICD od altri. Tali software hanno un proprio demone e lavorano in maniera indipendente dal vecchio sistema ifupdown . Questi moderni metodi di gestione delle reti sono pensati per elaboratori "home" , non son pensati in ambienti server . Questi moderni strumenti di gestione della rete devono essere configurati accuratamente al fine di evitare conflitti col vecchio metodo di gestione "ifupdown" , e relativo file di configurazione /etc/network/interfaces. SOLO LE INTERFACCHE CHE NON SONO PRESENTI NEL FILE /etc/network/interfaces SONO GESTITE DA NETWORKMANAGER , al fine di evitare conflitti.
- le schede di rete vengono configurate in base alle direttive presenti nel file /etc/network/interfaces Ecco tale file dopo un'installazione standard di debian 7

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

passiamo ora a spiegare il significato delle varie direttive

auto NOME_INTERFACCIA
attiva NOME_INTERFACCIA all'avvio del sistema

allow-auto NOME_INTERFACCIA
come sopra

allow-hotplug NOME_INTERFACCIA
attiva l'interfaccia all'avvio del sistema, inoltre ogni volta che il cavo di rete viene inserito o disinserito ecco che viene riconfigurata l'interfaccia

le righe che iniziano con iface NOME_CONFIG definiscono la configurazione di rete NOME_CONFIG . Le seguenti righe esiste sempre nel file /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback
```

gli indirizzi ip statici vengono configurati nel modo seguente

```
allow-hotplug eth0
iface eth0 inet static
address 192.168.11.100
```

```
netmask 255.255.255.0
gateway 192.168.11.1
dns-nameservers 192.168.11.1 # per questa direttiva occorre installare il pacchetto resolvconf
```

inet

indica che si intende utilizzare ipv4 , se intendiamo utilizzare ipv6 allora la direttiva diventa inet6 .

inet manual

non assegna nessun indirizzo all'interfaccia. Usato quando dobbiamo utilizzare la scheda di rete per compiti particolari, esempio un bridge di rete.

Inet dhcp

riceve l'indirizzamento tramite un server dhcp

pre-up COMANDO

esegue il COMANDO prima di alzare l'interfaccia di rete

post-up COMANDO

esegue il COMANDO dopo aver alzato l'interfaccia di rete

pre-down COMANDO

esegue il COMANDO prima di abbattere l'interfaccia di rete

post-down COMANDO

esegue il COMANDO dopo aver abbattuto l'interfaccia di rete

address

indirizzo ip assegnato staticamente

netmask

maschera di rete

gateway

gateway per la rete indicata

broadcast

indirizzo di broadcast

network

indica la rete corrente

up ip address add INDIRIZZO dev eth0

assegna un ulteriore indirizzo a eth0

nell'esempio seguente configureremo l'interfaccia di rete eth0 con un indirizzo statico

```
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
```

```
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
dns-nameserver 8.8.8.8      # la direttiva dns-nameserver e' valida a condizione di installare
dns-nameserver 8.8.4.4      # il pacchetto resolvconf
```

Se volessimo dare all'elaboratore anche l'indirizzo ip 192.168.1.110 dovremmo aggiungere la direttiva

```
up ip address add 192.168.1.110 dev eth0
```

un esempio avanzato

These options are pretty much "layer 2" options, setting up link states on interfaces, and are not related with "layer 3"(routing and addressing). As an example you could have a link aggregation where the bond0 interface needs to be up whatever the link state is, and it's members could be up after a link state event:

```
auto bond0
iface bond0 inet manual
    down ip link set $IFACE down
    post-down rmmmod bonding
    pre-up modprobe bonding mode=4 miimon=200
    up ip link set $IFACE up mtu 9000
    up udevadm trigger

allow-hotplug eth0
iface eth0 inet manual
    up ifenslave bond0 $IFACE
    down ifenslave -d bond0 $IFACE 2> /dev/null

allow-hotplug eth1
iface eth1 inet manual
    up ifenslave bond0 $IFACE
    down ifenslave -d bond0 $IFACE 2> /dev/null
```

So, this way i create a link aggregation and the interfaces will be added to it and removed on cable link states.

Fonte : <http://unix.stackexchange.com/questions/128439/good-detailed-explanation-of-etc-network-interfaces-syntax>

un altro esempio avanzato, qui definiamo una vlan

```
# Eth0
auto eth0
iface eth0 inet manual
    pre-up modprobe 8021q
    pre-up ifconfig eth0 up
    post-down ifconfig eth0 down

# Vlan Interface
auto vlan00
iface vlan00 inet static
address 10.0.0.1
netmask 255.255.255.0
gateway 10.0.0.254
vlan-raw-device eth0
ip_rp_filter 0
```

Fonte : <http://unix.stackexchange.com/questions/128439/good-detailed-explanation-of-etc-network-interfaces-syntax>

**IMPORTANTE!! NON DIMENTICARSI DI CONSULTARE LA DOCUMENTAZIONE
LOCATA PRESSO /usr/share/doc/ifupdon !**

http://www.linuxtopia.org/online_books/linux_system_administration/debian_linux_guides/debian_linux_reference_guide/ch-gateway.en_016.html

<http://manpages.ubuntu.com/manpages/lucid/man5/interfaces.5.html>