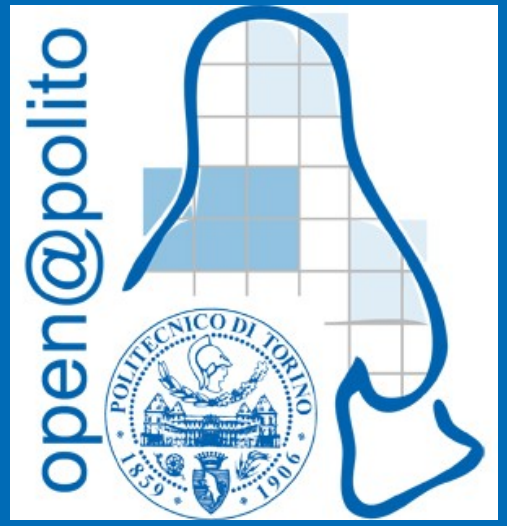
Introduzione al networking e SSH

Di Andrea Artuso

Con il supporto di:



Premessa

Il networking su sistemi GNU/Linux richiederebbe un intero corso in quanto è un argomento molto vasto.

Esistono numerosi metodi che permettono di ottenere la stessa configurazione, alcuni più semplici altri più complessi.

Data la natura di questo corso e il poco tempo a disposizione è stata fatta un'ampia selezione dei metodi e software che verranno presentati. L'obiettivo di questa lezione (come il titolo suggerisce) è quello di dare un'infarinatura sul vasto mondo del networking in GNU/Linux e dare alcuni semplici strumenti per iniziare a sperimentare.

Nella bibliografia di questa lezione sono elencate una serie di risorse per approfondire.

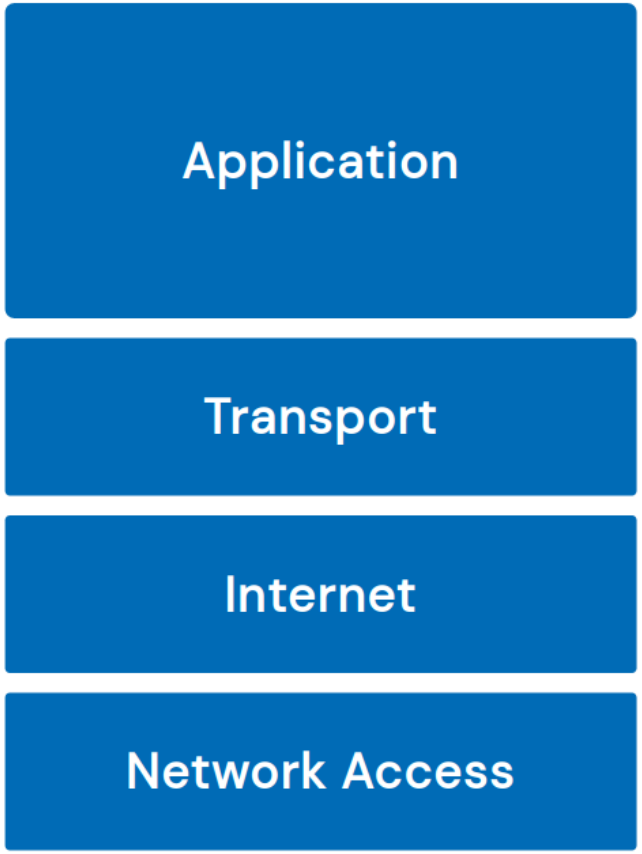
Fondamenti di reti IP

Indirizzo MAC:

- 3F:70:D6:10:7B:53
- Identifica **univocamente** una scheda di rete

Indirizzo IP (v4):

- 192.168.10.1/24
- Identifica un dispositivo all'interno di una rete IP



Stack TCP/IP

Interfacce di rete

In GNU/Linux *ogni* interfaccia di rete è un file (Lezione 02: filesystem), ma possono essere di due tipi:

1) Interfacce **fisiche**: rappresentano dei connettori fisici presenti sul dispositivo (es. porta RJ45).
2) Interfacce **virtuali**: sono delle interfacce completamente gestite da dei moduli software, possono essere utilizzate per far comunicare dei software virtualizzati (es. container Docker) con una rete fisica.

Configurazione delle interfacce

Ogni interfaccia di rete può essere configurata:

- **Staticamente**: siamo noi a decidere i parametri dell'interfaccia: indirizzo IP, velocità del collegamento, rotte, TTL, ecc. Questa configurazione è spesso utilizzata per le interfacce dei server.
- **Dinamicamente**: se la rete a cui l'interfaccia (virtuale o fisica) è collegata implementa il *protocollo DHCP*, possiamo configurare l'interfaccia in modo dinamico, non siamo quindi noi a decidere i parametri, ma sono assegnati automaticamente da un altro dispositivo nella rete. Questa è la configurazione classica delle reti domestiche.

Configurazione statica

Per configurare staticamente un'interfaccia sono disponibili una serie di pacchetti:

- **net-tools**: collezione di binari e comandi utili per la configurazione e la diagnostica delle interfacce di rete.
- **iproute2**: è il pacchetto attualmente più utilizzato per la configurazione di interfacce.
- **Netplan**: è un tool moderno sviluppato da Canonical (i maintainer di Ubuntu) che sfrutta i concetti dell'*IaaC* per configurare le interfacce.

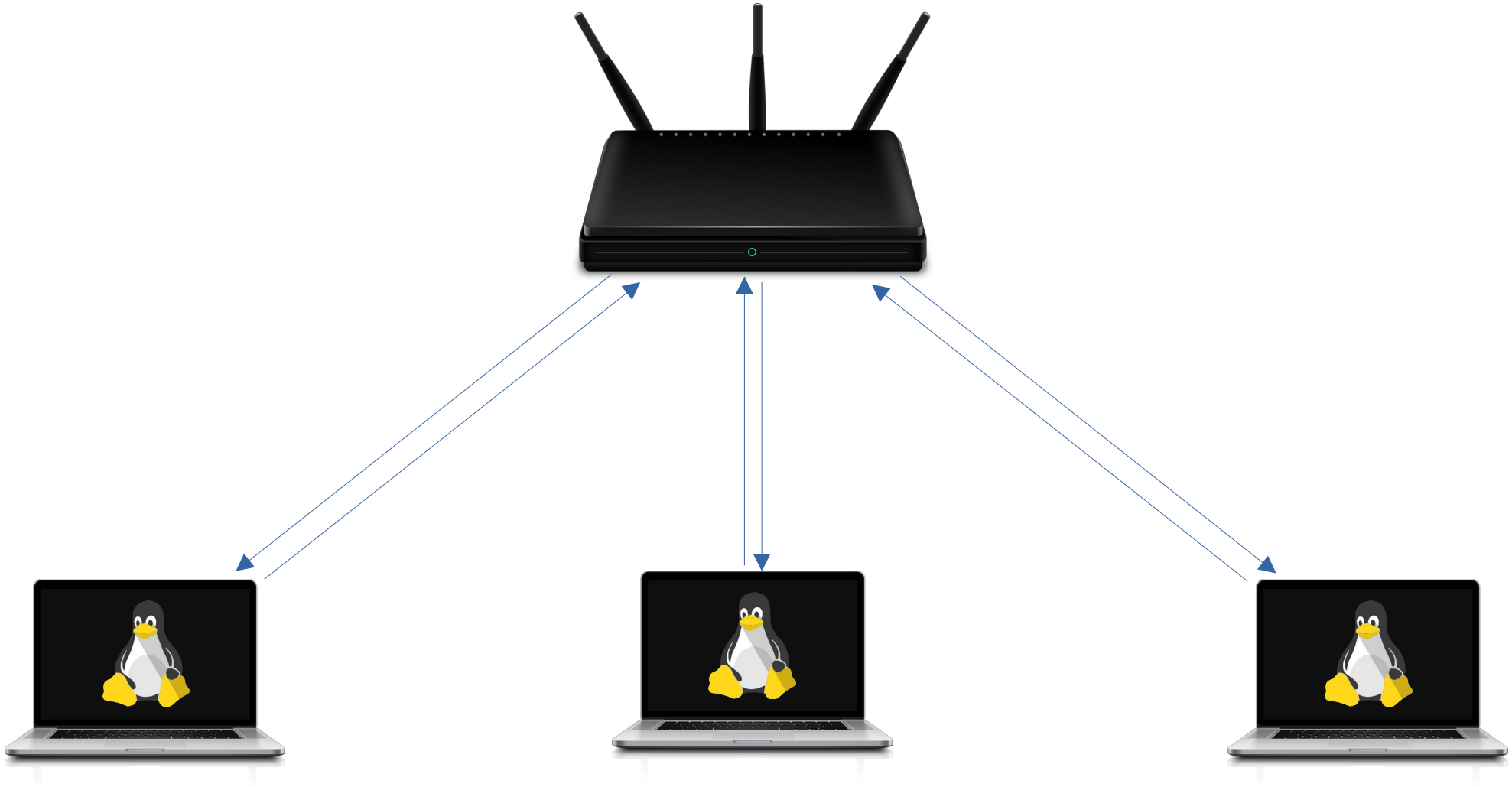
Noi vedremo solo i primi due in quanto Netplan richiede conoscenze più avanzate[1]

Configurazione statica

Principali comandi di configurazione:

net-tools	iproute2	
ifconfig	ip addr/ip link	Config. indirizzi
route	ip route	Config. rotte
arp	ip neigh	”Neighbors”
ifup/ifdown	ip link set <interface> up/down	Attiva/disattiva interf.
iptunnel	ip tunnel	Gestisce i tunnel
nameif/ifrename	ip link set name	Nome dell’interfaccia
netstat	ss, ip route	Mostrano statistiche

Configurazione dinamica: DHCP



Configurazione dinamica

Ci sono diversi modi per gestire la configurazione dinamica su GNU/Linux. Nei sistemi moderni spesso è già tutto preconfigurato (es. utilizzando NetworkManager). Se si ha necessità di operare su un'interfaccia configurata dinamicamente è possibile utilizzare il comando `dhclient` (per installarlo è necessario installare *isc-dhcp-client*, su Debian-based: `sudo apt install isc-dhcp-client`).

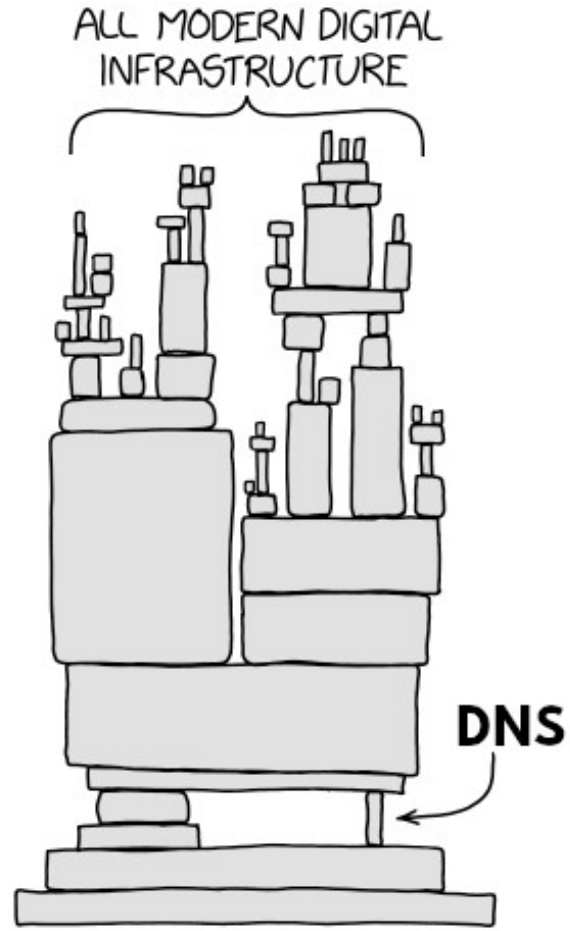
`dhclient <interfaccia>`	Avvia il processo di richiesta di un IP tramite DHCP
`dhclient -r <interfaccia>`	Forza la rimozione della configurazione dinamica

DNS e nameserver

Nei sistemi Debian-based è possibile indicare i resolver DNS nel file /etc/resolv.conf:

```
nameserver 1.2.3.4
nameserver 5.6.7.8
```

È anche possibile installare un resolver locale come *bind9*, che permette una configurazione più avanzata.



Inspired by xkcd #2347, https://xkcd.com/2347/

Comandi utili per la diagnostica

arp: visualizza la tabella ARP

netstat: visualizza informazioni sulle connessioni aperte

ping: verifica la connettività tra due host di rete

route: mostra la Routing Table

traceroute: mostra la “rotta” che i pacchetti seguono tra due host

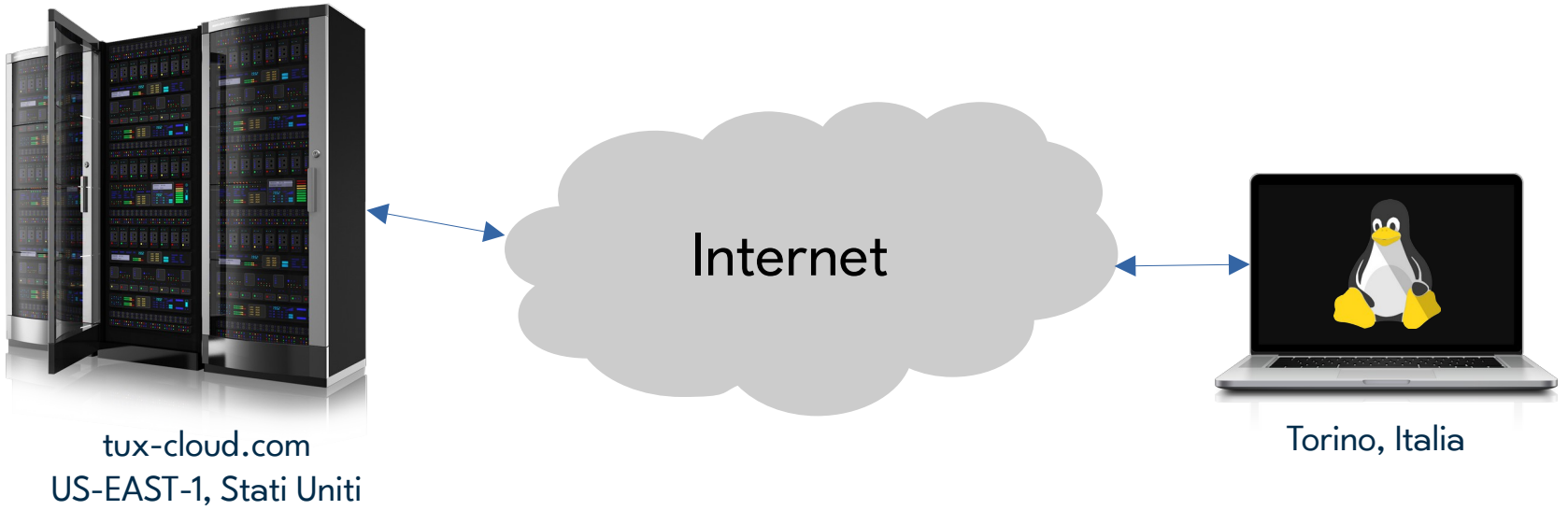
nslookup / dig: risolve nomi di dominio nel corrispettivo indirizzo IP

nc / netcat / ncat: permette di leggere/scrivere dati su una connessione di rete

Accesso remoto: SSH

Secure **SH**ell è un protocollo per gestire l'accesso remoto in modo sicuro ad una shell, ovvero ad un altro sistema raggiungibile attraverso una rete IP.

Nasce come sostituto a TelNet e viene principalmente utilizzato per accesso a server.



SSH: configurazione server

Uno dei software più utilizzati è OpenSSH: `sudo apt install openssh-server`

I file di configurazione si trovano nella directory `/etc/ssh`

Possiamo configurare:

- IP/Porta
- Metodi di autenticazione e utenti abilitati
- Jails (SFTP)[2]
- Molto altro...

SSH: configurazione client

Esistono diversi client, il più diffuso è OpenSSH: `sudo apt install openssh-client`

Possiamo connetterci ad un server direttamente da linea di comando:

```
ssh <utente>@<ip/hostname> -p <porta> -o <opzioni>
```

Oppure possiamo configurare degli host in:

`$HOME/.ssh/config`

e collegarci usando semplicemente: `ssh <nome-host>`

SSH: metodi di autenticazione

SSH supporta diversi tipi di autenticazione, i principali sono:

- Autenticazione basata su password: meno sicura, più semplice, adatta per utilizzi “non pubblici” (es. homeserver non esposto su Internet)
- Autenticazione basata su chiave pubblica: metodo molto sicuro (attenzione agli algoritmi usati!) e *deve* essere utilizzato per server esposti su Internet.
- Autenticazione basata su certificati (argomento avanzato)[3]

La configurazione di questi metodi deve essere fatta nel file `/etc/ssh/sshd_config` sul server. SSH permette l’abilitazione di più metodi di autenticazione insieme, anche se non è una pratica consigliata.

SSH: autenticazione basata su chiave pubblica

Generazione di una chiave (client): `ssh-keygen -t <algoritmo> -C <nome chiave>`

Verranno generati due file: `<nome chiave>` e `<nome chiave>.pub`

Il primo contiene la chiave privata e **deve** rimanere sul client, il secondo contiene la chiave pubblica che deve essere importata nel file `$HOME/.ssh/authorized_keys` sul server.

Per connettersi al server autenticandosi con la chiave è necessario utilizzare l'opzione `-i`:

```
ssh user@server -p <porta> -i /percorso/chiave/privata
```

SSH: best-practices

Best practices per la configurazione di SSH:

- Usare l'autenticazione basata su chiave pubblica con algoritmi di cifratura sicuri (es. ed25519) o basata su certificati digitali[3].
- Cambiare le chiavi frequentemente.
- **DISABILITARE** l'autenticazione a root tramite SSH.
- Disabilitare il login basato su password e selezionare gli utenti che possono accedere tramite SSH.
- Configurare `fail2ban`[4]

Bibliografia

Fonti e riferimenti

[1] https://netplan.readthedocs.io/en/stable/
[2] https://www.redhat.com/en/blog/set-linux-chroot-jails
[3] https://goteleport.com/blog/how-to-configure-ssh-certificate-based-authentication/
[4] https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-debian-11

Risorse per approfondire

James F. Kurose and Keith Ross. 2020. Computer Networking: A Top-Down Approach (8th Edition). ISBN-13: 978-0135928615
Olaf Kirch and Terry Dawson. 2000. Linux Network Administrator's Guide (2nd Edition). ISBN-13: 978-1565924000. URL: https://tldp.org/LDP/nag2/nag2.pdf
Dynamic Host Configuration Protocol. wikipedia.org. URL: https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
Domain Name System. wikipedia.org. URL: https://en.wikipedia.org/wiki/Domain_Name_System
What is SSH? cloudflare.com. URL: https://www.cloudflare.com/learning/access-management/what-is-ssh/
Secure Shell. wikipedia.org. URL: https://en.wikipedia.org/wiki/Secure_Shell

Copyleft