

Virtual Networking Laboratory



*Laboratorio di Reti
2014/2015
Prof. Laura Ricci*



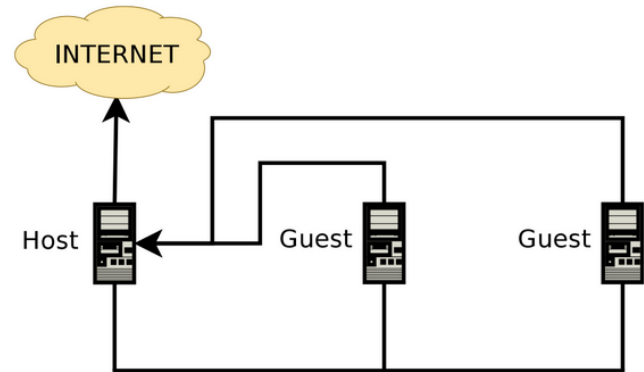
Speaker: Alessandro Lulli - lulli@di.unipi.it

Reference

- le slide sono una sintesi delle slide che potete trovare al seguente indirizzo (from: Augusto Ciuffoletti):
 - <https://sites.google.com/site/laboratoriodireti/materiale-didattico/appunti-del-laboratorio>
- per ulteriori informazioni:
 - <https://sites.google.com/site/virtlabreti/howto>

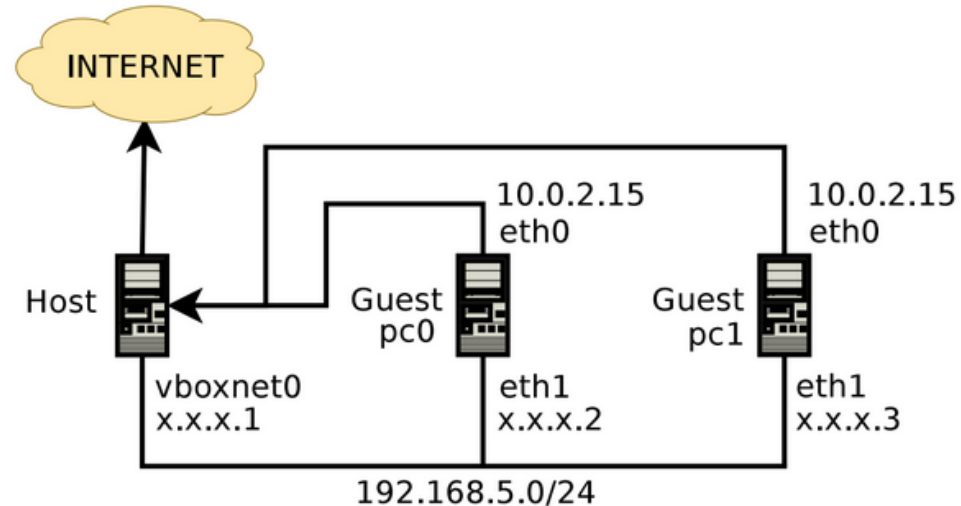
Cosa vogliamo fare

- problema:
 - per il nostro laboratorio serve una piccola rete in cui provare i nostri programmi
 - almeno 2 o 3 nodi per poter fare comunicazione tra di loro
 - nodi collegati tra loro e con accesso internet
- soluzione: **rete virtuale**



La rete virtuale

- le tre interfacce sulla rete virtuale Ethernet hanno indirizzi:
 - 192.168.5.1 per il nodo **Host**
 - 192.168.5.2 per il primo guest (**pc0**)
 - 192.168.5.3 per il secondo guest (**pc1**)



Realizzazione rete virtuale

- *virtual box*
 - prodotto per la virtualizzazione di risorse di calcolo
 - <https://www.virtualbox.org/>
 - versione ≥ 4.3
- *vagrant*
 - per il provisioning dell'ambiente
 - <https://www.vagrantup.com/>
 - versione $\geq 1.6.5$



Virtual Box

- riproduce funzionamento di una piattaforma hardware all'interno di un'altra (scatole cinesi):
 - sistema host (che ospita)
 - guest (che viene ospitato)
- il sistema guest:
 - può accedere a buona parte delle periferiche del sistema host
 - il sistema guest può utilizzare file del sistema host come se fossero hard disk



Vagrant

- creare una directory di lavoro
- nella directory salvate il file *Vagrantfile* del seguente indirizzo:
 - <https://sites.google.com/site/laboratoriodireti/il-laboratorio-virtuale>



Start Virtual Lab

- digitate il comando “*vagrant up*” nella directory di lavoro
- attendere che il provisioning sia completo
- **ATTENZIONE** l’operazione è lunga, e scarica circa 500MByte da Internet

```
File Edit View Search Terminal Help
alemare@bho: ~/mio/uni/virtualLab
alemare@bho:~/mio/uni/virtualLab$ vagrant up
Bringing machine 'pc0' up with 'virtualbox' provider...
Bringing machine 'pc1' up with 'virtualbox' provider...
==> pc0: Checking if box 'labrettl/labvm' is up to date...
==> pc0: Clearing any previously set forwarded ports...
==> pc0: Clearing any previously set network interfaces...
==> pc0: Preparing network interfaces based on configuration...
    pc0: Adapter 1: nat
    pc0: Adapter 2: hostonly
==> pc0: Forwarding ports...
    pc0: 22 => 2222 (adapter 1)
==> pc0: Booting VM...
==> pc0: Waiting for machine to boot. This may take a few minutes...
    pc0: SSH address: 127.0.0.1:2222
    pc0: SSH username: vagrant
    pc0: SSH auth method: private key
    pc0: Warning: Connection timeout. Retrying...
    pc0: Warning: Connection timeout. Retrying...
    pc0: Warning: Connection timeout. Retrying...
    pc0: Warning: Remote connection disconnect. Retrying...
    pc0: Warning: Remote connection disconnect. Retrying...
==> pc0: Machine booted and ready!
==> pc0: Checking for guest additions in VM...
==> pc0: Setting hostname...
==> pc0: Configuring and enabling network interfaces...
==> pc0: Mounting shared folders...
    pc0: /vagrant => /home/alemare/mio/uni/virtualLab
==> pc0: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
```


Accesso ad una macchina guest

- nella directory di lavoro dare il comando
 - *vagrant@pc0:~\$ vagrant ssh pc0*
- non è necessaria una password, esiste un utente preconfigurato
 - username → vagrant, password → vagrant
- possono essere utilizzati anche comandi che presentano una interfaccia grafica
 - *vagrant@pc0:~\$ wireshark &*

Networking da una macchina guest

- per verificare la connettività della rete virtuale
 - `vagrant@pc0:~$ ping 192.168.5.1`
- per verificare la connettività Internet
 - `vagrant@pc0:~$ ping www.google.com`

La directory condivisa

- le macchine guest condividono la directory **/vagrant**
 - il contenuto di questa directory è in comune
 - il contenuto di questa directory è identico a quello della directory in cui è installato il Vagrantfile sull'host
- esecuzione di un jar su una delle macchine guest: esportare il jar eseguibile nella directory di lavoro
 - `vagrant@pco:~$ cd /vagrant`
 - `vagrant@pco:~$ java -jar /vagrant/Hallo.jar`

Esempio: Echo

- un semplice server che fa l'eco
- in **pc1** lanciamo il server

```
alemare@bho:~/mio/uni/virtualLab$ vagrant ssh pc1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information disabled due to load higher than 1.0

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

Last login: Fri Feb 13 14:24:10 2015 from 10.0.2.2
vagrant@pc1:~$ ls
vagrant@pc1:~$ cd /
bin/          dev/          home/         lib64/        media/        opt/          root/        sbin/         sys/          usr/          var/
boot/         etc/          lib/          lost+found/   mnt/          proc/         run/         srv/          tmp/          vagrant/
vagrant@pc1:~$ cd /vagrant/
vagrant@pc1:/vagrant$ ls
tcpClient.jar  tcpClient_lib  tcpServer.jar  tcpServer_lib  Vagrantfile
vagrant@pc1:/vagrant$ java -jar tcpServer.jar
```

Echo Client

```
vagrant@pc0: /vagrant
File Edit View Search Terminal Tabs Help
vagrant@pc0: /vagrant x vagrant@pc1: /vagrant x
alemare@bho:~/nio/uni/virtuaLLab$ vagrant ssh pc0
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Mar 20 10:00:47 UTC 2015

System load:  0.08          Processes:      73
Usage of /:   3.8% of 39.34GB Users logged in:  0
Memory usage: 25%         IP address for eth0: 10.0.2.15
Swap usage:   0%           IP address for eth1: 192.168.5.2

Graph this data and manage this system at:
  https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

Last login: Fri Mar 20 10:00:47 2015 from 10.0.2.2
vagrant@pc0:~$ cd /vagrant/
vagrant@pc0:~/vagrant$ ls
tcpClient.jar  tcpClient_lib  tcpServer.jar  tcpServer_lib  Vagrantfile
vagrant@pc0:~/vagrant$ java -jar tcpClient.jar 192.168.5.1
Exception in thread "main" java.net.ConnectException: Connection refused
    at java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:339)
    at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:200)
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:182)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
    at java.net.Socket.connect(Socket.java:579)
    at java.net.Socket.connect(Socket.java:528)
    at java.net.Socket.<init>(Socket.java:425)
    at java.net.Socket.<init>(Socket.java:208)
    at lpr.tcpSocket.TCPClient.main(TCPClient.java:15)
vagrant@pc0:~/vagrant$ java -jar tcpClient.jar 192.168.5.3
ciao
FROM SERVER: CIAO
```

```
vagrant@pc1:~/vagrant$ java -jar tcpServer.jar
Received: ciao
```

SERVER