
MONITORING VIRTUALIZED NETWORKS

Design of Networks and Communication Systems

a.a. 2020/2021

Calore Chiara
Cesaro Pasquale Alan
Miotto Sara

Obiettivi del Progetto:



- Sviluppare una rete di virtual machines/containers su uno o più computers;
- Le connessioni devono essere a banda limitata;
- La rete deve essere basata su OpenVSwitch;
- Lo stato del sistema deve essere analizzato raccogliendo informazioni riguardo l'utilizzo delle risorse delle macchine virtuali/container e dei link.

Strumenti utilizzati (1/7)



VAGRANT

VAGRANT

- 📍 E' uno strumento per costruire e gestire ambienti di macchine virtuali in modo semplice
- 📍 Si focalizza sull'automatizzazione
- 📍 Permette di creare e configurare ambienti facilmente riproducibili

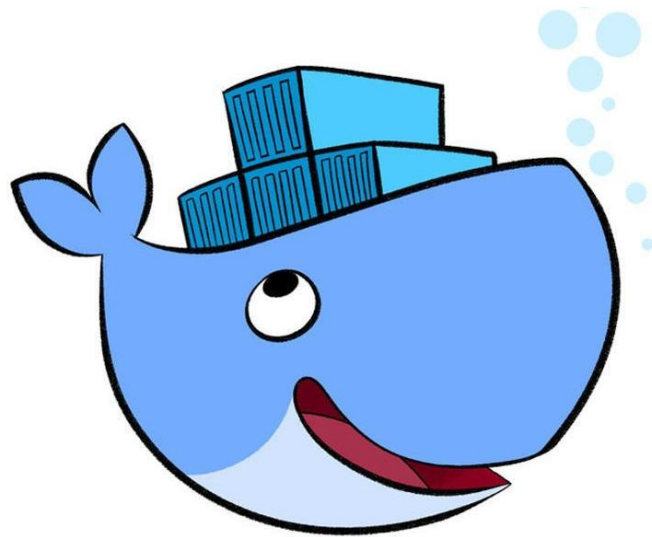
Strumenti utilizzati (2/7)



VIRTUALBOX

- 📍 E' un software gratuito e open source per l'esecuzione di macchine virtuali
- 📍 Supporta Windows, GNU/Linux e macOS come sistemi operativi host
- 📍 Permette la configurazione, la creazione e l'utilizzo di più sistemi operativi nel proprio PC usando un solo sistema operativo

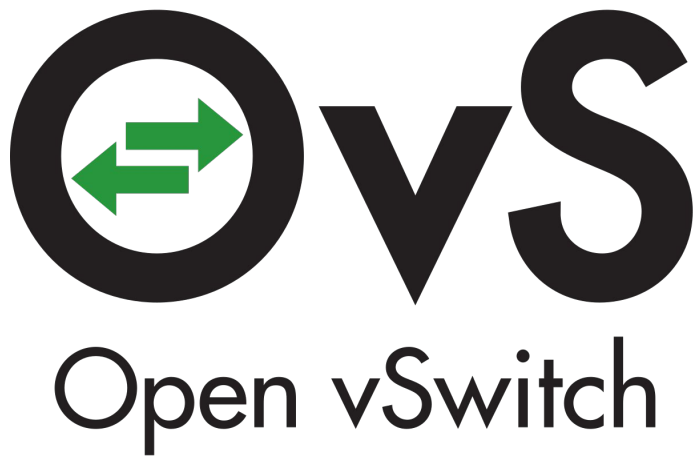
Strumenti utilizzati (3/7)



DOCKER

- 📍 Progetto open-source che automatizza lo sviluppo di applicazioni all'interno di contenitori software
- 📍 Fornisce un'astrazione a livello di sistema operativo di Linux tramite la virtualizzazione
- 📍 Implementa API di alto livello per gestire container che eseguono processi in ambienti isolati, utilizzando le funzionalità del kernel e sfruttando l'isolamento delle risorse (per mezzo di containers);
- 📍 Semplifica la creazione di sistemi distribuiti, permettendo a diverse applicazioni o processi di lavorare in modo autonomo sulla stessa macchina fisica o su diverse macchine virtuali

Strumenti utilizzati (4/7)



OPENVSWITCH

- 📍 E' un software open source che implementa uno switch virtuale a più livelli
- 📍 Supporta vari protocolli e standards utilizzati nelle reti di computer virtuali

Strumenti utilizzati (5/7)



NETEM

- 📍 E' un potenziamento delle funzioni di controllo del traffico della rete
- 📍 Permette di aggiungere ritardi alla rete, perdita e duplicazioni di pacchetti
- 📍 Si utilizza in ambienti Linux-based

Strumenti utilizzati (6/7)



VISUAL STUDIO CODE

- 📍 E' un editor di codice sorgente sviluppato da Microsoft per Windows, Linux e macOS
- 📍 È un software libero e gratuito, anche se la versione ufficiale è sotto una licenza proprietaria
- 📍 Può essere usato con vari linguaggi di programmazione

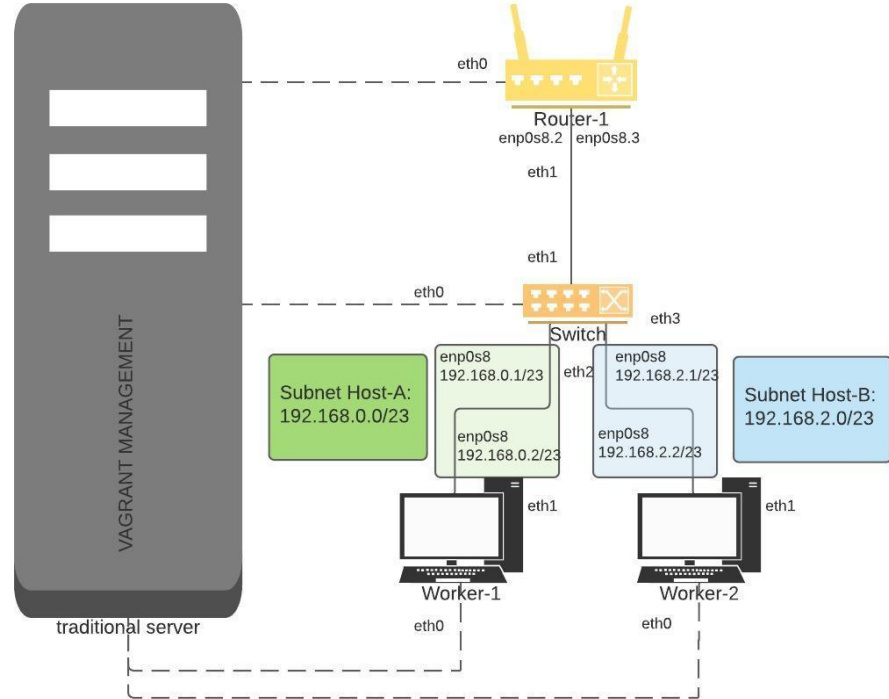
Strumenti utilizzati (7/7)



GIT BASH

- 📍 Applicazione che fornisce un'emulazione dei comandi su Bash presenti in Linux e MacOS
- 📍 Da utilizzare in ambiente Windows

Network Topology



Vagrantfile

Creazione del Vagrantfile

Contiene le macchine virtuali necessarie per il progetto e la loro configurazione

```
11 Vagrant.configure("2") do |config|
12   config.vm.box_check_update = false
13   config.vm.provider "virtualbox" do |vb|
14     vb.customize ["modifyvm", :id, "--usb", "on"]
15     vb.customize ["modifyvm", :id, "--usbhci", "off"]
16     vb.customize ["modifyvm", :id, "--nicpromisc2", "allow-all"]
17     vb.customize ["modifyvm", :id, "--nicpromisc3", "allow-all"]
18     vb.customize ["modifyvm", :id, "--nicpromisc4", "allow-all"]
19     vb.customize ["modifyvm", :id, "--nicpromisc5", "allow-all"]
20     vb.cpus = 1
21   end
22   config.vm.define "router-1" do |router1|
23     router1.vm.box = "ubuntu/bionic64"
24     router1.vm.hostname = "router-1"
25     router1.vm.network "private_network", virtualbox__intnet: "broadcast
26     router1.vm.network "private_network", virtualbox__intnet: "broadcast
27     router1.provision "shell", path: "router-1.sh"
28     router1.vm.provider "virtualbox" do |vb|
29       vb.name = "Router"
30       vb.memory = 256
31     end
32   end
33
34   config.vm.define "switch" do |switch|
35     switch.vm.box = "ubuntu/bionic64"
36     switch.vm.hostname = "switch"
37     switch.vm.network "private_network", virtualbox__intnet: "broadcast
38     switch.vm.network "private_network", virtualbox__intnet: "broadcast
39     switch.vm.network "private_network", virtualbox__intnet: "broadcast
40     switch.provision "shell", path: "switch.sh"
41     switch.vm.provider "virtualbox" do |vb|
42       vb.name = "Switch"
```

File .sh (¼)

router-1.sh

Comandi per gestire la comunicazione tra il router e lo switch e, tramite questo, tra il router e i due host della rete, attraverso le due corrispondenti vlan

```
router-1.sh
1  export DEBIAN_FRONTEND=noninteractive
2
3  #Startup commands go here
4  #Enable routing
5  sudo sysctl -w net.ipv4.ip_forward=1
6  #Network and VLAN interface config
7  sudo ip link add link enp0s8 name enp0s8.2 type vlan id 2
8  sudo ip link add link enp0s8 name enp0s8.3 type vlan id 3
9  sudo ip addr add 192.168.0.1/23 dev enp0s8.2
10 sudo ip addr add 192.168.2.1/23 dev enp0s8.3
11 sudo ip link set dev enp0s8 up
```

File .sh (2/4)

switch.sh

Comandi per gestire la comunicazione tra il router, lo switch e i due host della rete.

Crea una rete basata su OpenVSwitch, un Bridge per il controllo del traffico e il limite della banda della rete (comando `tc` di Netem)

```
switch.sh
1 export DEBIAN_FRONTEND=noninteractive
2 #Startup commands for switch go here
3 apt-get update
4 apt-get install -y tcpdump
5 apt-get install -y openvswitch-common openvswitch-switch apt-transport-https ca-certificates curl software-properties-common
6 #Switch ports config
7 sudo ovs-vsctl add-br switch
8 sudo ovs-vsctl add-port switch enp0s8
9 sudo ovs-vsctl add-port switch enp0s9 tag="2"
10 sudo ovs-vsctl add-port switch enp0s10 tag="3"
11 sudo apt install iperf
12 sudo tc qdisc add dev enp0s3 root tbf rate 1mbit burst 32kbit latency 400ms
13 sudo tc qdisc add dev enp0s8 root tbf rate 1mbit burst 32kbit latency 400ms
14 sudo tc qdisc add dev enp0s9 root tbf rate 1mbit burst 32kbit latency 400ms
15 sudo tc qdisc add dev enp0s10 root tbf rate 1mbit burst 32kbit latency 400ms
16 #Setting up links
17 sudo ip link set dev enp0s8 up
18 sudo ip link set dev enp0s9 up
19 sudo ip link set dev enp0s10 up
20
```

File .sh (3/4)

host-a.sh

Comandi per gestire il worker-1 e la sua
connessione allo switch.

La presenza dell'immagine docker è stata
utilizzata per rendere più realistico l'ambiente di
lavoro

host-a.sh

```
1  export DEBIAN_FRONTEND=noninteractive
2
3  #Startup commands go here
4  sudo ip addr add 192.168.0.2/23 dev enp0s8
5  #Network interface config
6  sudo ip link set dev enp0s8 up
7  #Default gateway set up
8  sudo ip route add 192.168.0.0/21 via 192.168.0.1
9  sudo apt-get update
10 sudo apt -y install docker.io
11 sudo systemctl start docker
12 sudo docker pull mlabbe/iperf
13
14
```

File .sh (4/4)

host-b.sh

Comandi per gestire il worker-2 e la sua
connessione allo switch.

La presenza dell'immagine docker è stata
utilizzata per rendere più realistico l'ambiente di
lavoro

```
host-b.sh
1  export DEBIAN_FRONTEND=noninteractive
2
3  #Startup commands go here
4  #Download package information from all configured sources
5  sudo apt-get update
6  #Install and run Docker.io
7  sudo apt -y install docker.io
8  sudo systemctl start docker
9  sudo systemctl enable docker
10 sudo docker pull dustnic82/nginx-test
11 sudo docker run --name nginx -p 80:80 -d dustnic82/nginx-test
12 #Network interface config
13 sudo ip addr add 192.168.2.2/23 dev enp0s8
14 sudo ip link set dev enp0s8 up
15 #Default gateway set up
16 sudo ip route add 192.168.0.0/21 via 192.168.2.1
17
```

Come iniziamo?



```
PS C:\Users\hp\Desktop\2_prog_Vagrant> vagrant up
Bringing machine 'router-1' up with 'virtualbox' provider...
Bringing machine 'switch' up with 'virtualbox' provider...
Bringing machine 'host-a' up with 'virtualbox' provider...
Bringing machine 'host-b' up with 'virtualbox' provider...
=> router-1: Importing base box 'ubuntu/bionic64'...
=> router-1: Matching MAC address for NAT networking...
=> router-1: Setting the name of the VM: Router
=> router-1: Fixed port collision for 22 => 2222. Now on port 2205.
=> router-1: Clearing any previously set network interfaces...
=> router-1: Preparing network interfaces based on configuration...
router-1: Adapter 1: nat
router-1: Adapter 2: intnet
router-1: Adapter 3: intnet
=> router-1: Forwarding ports...
router-1: 22 (guest) => 2205 (host) (adapter 1)
=> router-1: Running 'pre-boot' VM customizations...
=> router-1: Booting VM...
=> router-1: Waiting for machine to boot. This may take a few minutes...
router-1: SSH address: 127.0.0.1:2205
router-1: SSH username: vagrant
router-1: SSH auth method: private key
router-1: Warning: Connection reset. Retrying...
router-1: Warning: Connection aborted. Retrying...
router-1: Warning: Connection reset. Retrying...
router-1: Warning: Connection aborted. Retrying...
router-1:
router-1: Vagrant insecure key detected. Vagrant will automatically replace
router-1: this with a newly generated keypair for better security.
router-1:
router-1: Inserting generated public key within guest...
router-1: Removing insecure key from the guest if it's present...
router-1: Key inserted! Disconnecting and reconnecting using new SSH key...
=> router-1: Machine booted and ready!
=> router-1: Checking for guest additions in VM...
router-1: The guest additions on this VM do not match the installed version of
router-1: VirtualBox! In most cases this is fine, but in rare cases it can
router-1: prevent things such as shared folders from working properly. If you see
router-1: shared folder errors, please make sure the guest additions within the
router-1: virtual machine match the version of VirtualBox you have installed on
router-1: your host and reload your VM.
router-1:
router-1: Guest Additions Version: 5.2.42
router-1: VirtualBox Version: 6.1
```

vagrant up

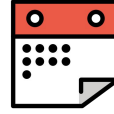


Comando da inserire da terminale



Avvia la creazione dell'ambiente di sviluppo
(genera le diverse virtual machines)

Menu Bash



```
/usr/bin/bash --login -i C:\Users\Windows\OneDrive\Desktop\2_prog_Vagrant\te...  
1) Monitor worker-1      6) Visualize information worker-2  
2) Monitor worker-2      7) Visualize information router  
3) Monitor router        8) Visualize information switch  
4) Monitor switch        9) Visualize information bandwidth  
5) Visualize information worker-1 10) Quit  
Please enter your choice:
```



Digitare il comando *bash test.sh* per aprire il menù bash



Seguire le istruzioni

Come monitoriamo la rete?



```
PROBLEME  BUTYPER  DEBUZ CONSOLE  TERMINAL
54 Bytes from 192.168.2.2: icmp_seq=6 ttl=63 time=5.00 ms
54 Bytes from 192.168.2.2: icmp_seq=7 ttl=63 time=4.12 ms
54 Bytes from 192.168.2.2: icmp_seq=8 ttl=63 time=4.76 ms
54 Bytes from 192.168.2.2: icmp_seq=9 ttl=63 time=7.26 ms
54 Bytes from 192.168.2.2: icmp_seq=10 ttl=63 time=10.7 ms
54 Bytes from 192.168.2.2: icmp_seq=11 ttl=63 time=11.5 ms
54 Bytes from 192.168.2.2: icmp_seq=12 ttl=63 time=2.37 ms
54 Bytes from 192.168.2.2: icmp_seq=13 ttl=63 time=3.52 ms
54 Bytes from 192.168.2.2: icmp_seq=14 ttl=63 time=3.78 ms
54 Bytes from 192.168.2.2: icmp_seq=15 ttl=63 time=1.68 ms
54 Bytes from 192.168.2.2: icmp_seq=16 ttl=63 time=1.45 ms
54 Bytes from 192.168.2.2: icmp_seq=17 ttl=63 time=3.20 ms
54 Bytes from 192.168.2.2: icmp_seq=18 ttl=63 time=1.29 ms
54 Bytes from 192.168.2.2: icmp_seq=19 ttl=63 time=1.76 ms
54 Bytes from 192.168.2.2: icmp_seq=20 ttl=63 time=1.75 ms
54 Bytes from 192.168.2.2: icmp_seq=21 ttl=63 time=3.22 ms
54 Bytes from 192.168.2.2: icmp_seq=22 ttl=63 time=5.23 ms
54 Bytes from 192.168.2.2: icmp_seq=23 ttl=63 time=3.18 ms
54 Bytes from 192.168.2.2: icmp_seq=24 ttl=63 time=1.59 ms
54 Bytes from 192.168.2.2: icmp_seq=25 ttl=63 time=2.94 ms
54 Bytes from 192.168.2.2: icmp_seq=26 ttl=63 time=1.28 ms
54 Bytes from 192.168.2.2: icmp_seq=27 ttl=63 time=1.86 ms
54 Bytes from 192.168.2.2: icmp_seq=28 ttl=63 time=1.29 ms
54 Bytes from 192.168.2.2: icmp_seq=29 ttl=63 time=1.70 ms
```

```
MarBOM-A1r-02-Alex:2 vagrant ~$ sudo ./test.sh
1) Monitor worker-1          5) Visualize information worker-2
2) Monitor worker-2          7) Visualize information router
3) Monitor router            8) Visualize information switch
4) Monitor switch            9) Visualize information bandwidth
5) Visualize information worker-1  0) Quit
Please enter your choice: 4
you chose choice 4 which is Monitor switch
Collecting information...
press Ctrl+C when you want to stop the monitoring of the net and wait
Time stamp  Object      Metric      Value
-----
09:50:54.783 Switch    CPU/Load/Kernel  0.57%
09:50:54.783 Switch    RAM/Usage/Used    380888 kb
09:50:54.783 Switch    Disk/Usage/Used    1862 MB
09:50:54.783 Switch    Net/Rate/Rx        489 B/s
09:50:54.783 Switch    Net/Rate/Tx        401 B/s
09:50:54.783 Switch    Guest/RAM/Usage/Total 224632 kb
09:50:54.783 Switch    Guest/RAM/Usage/Free 116664 kb
```

La banda è limitata?



```
vagrant@host-a: ~  
1) Monitor worker-1      6) Visualize information worker-2  
2) Monitor worker-2      7) Visualize information router  
3) Monitor router        8) Visualize information switch  
4) Monitor switch        9) Visualize information bandwidth  
5) Visualize information worker-1  10) Quit  
Please enter your choice: 9  
you chose choice 9 which is Visualize information bandwidth  
From which device you want to measure it?  
Enter the number: worker-1 (1), worker-2 (2), switch (3), router (4)  
Instead if you want to exit digit exit  
1  
please wait until the program will place you inside worker-1.  
Then enter the command:  
sudo tc qdisc show dev [name of the interface you want do display]  
You can select one of the ensuing interfaces:  
enp0s3, enp0s8  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-121-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sat Feb 13 14:05:13 UTC 2021
```

Scegliere l'opzione 9 e seguire le istruzioni proposte

Problematiche



Approccio a software utilizzati per la prima volta in questo corso

Situazione sanitaria che ci ha impedito di lavorare fisicamente insieme



Grazie per l'attenzione

```

1 bash
2 #!/bin/bash
3
4 PS1="Please enter your choice: "
5
6 options="Monitor host-a " "Monitor host-b " "Monitor router " "Monitor switch " "Visualize information host-a "
7 select opt in "${options[@]}"
8 do
9     case $opt in
10         "Monitor host-a")
11             echo "You chose choice 1"
12             WbomNage metrics collect --period 10 --samples 1 worker-1 CPU/Loader/User,CPU/Load/Kernel,RAM
13             ;;
14         "Monitor host-b")
15             echo "You chose choice 2"
16             WbomNage metrics collect --period 10 --samples 1 worker-2 CPU/Loader/User,CPU/Load/Kernel,RAM
17             ;;
18         "Monitor router")
19             echo "You chose choice $REPLY which is $opt"
20             WbomNage metrics collect --period 10 --samples 1 Router CPU/Loader/User,CPU/Load/Kernel,RAM
21             ;;
22         "Monitor switch")
23             echo "You chose choice $REPLY which is $opt"
24             WbomNage metrics collect --period 10 --samples 1 Switch CPU/Loader/User,CPU/Load/Kernel,RAM
25             ;;
26         "Visualize information host-a")
27             echo "You chose choice $REPLY which is $opt"
28             cat worker-1.txt
29             ;;
30         "Visualize information host-b")
31             echo "You chose choice $REPLY which is $opt"
32             cat worker-2.txt
33             ;;
34     esac
35 done

```

[illegible]