

Grado en Ingeniería Informática

Administración de Servidores

Práctica 3

José Manuel Morales García
Gabriel Fernando Sánchez Reina

15 de abril de 2019

Índice

1. Ejercicio 2	3
2. Ejercicio 3 DHCP	7
3. Ejercicio 4 DNS	9

1. Ejercicio 2

En la topología a desarrollar, usaremos 5 máquinas, (de vm1 a vm5), donde vm1 y vm2 pertenecerán a la red “Red2” (192.168.2.0/24), vm3 y vm4 a la “Red3” (192.168.3.0/24) y vm5 actuará de router, con 2 tarjetas de red.

El Vagrantfile será el siguiente:

```
Vagrant.configure("2") do |config|

  config.vm.define "vm1" do |vm1|
    vm1.vm.box="hashicorp/precise64"
    vm1.vm.hostname="vm1"
    vm1.vm.network "private_network", ip: "192.168.2.2"
  end

  config.vm.define "vm2" do |vm2|
    vm2.vm.box="hashicorp/precise64"
    vm2.vm.hostname="vm2"
    vm2.vm.network "private_network", ip: "192.168.2.3"
  end

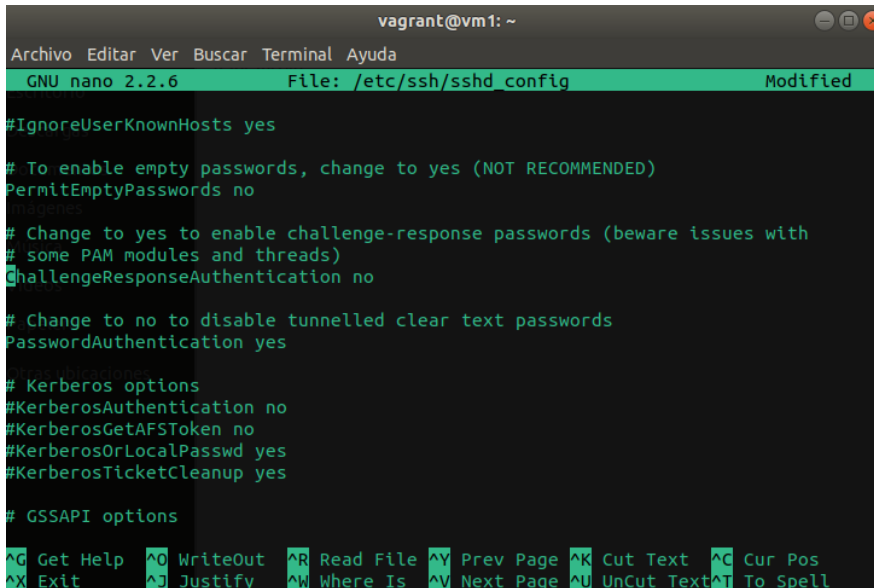
  config.vm.define "vm3" do |vm3|
    vm3.vm.box="hashicorp/precise64"
    vm3.vm.hostname="vm3"
    vm3.vm.network "private_network", ip: "192.168.3.2"
  end

  config.vm.define "vm4" do |vm4|
    vm4.vm.box="hashicorp/precise64"
    vm4.vm.hostname="vm4"
    vm4.vm.network "private_network", ip: "192.168.3.3"
  end

  config.vm.define "vm5" do |vm5|
    vm5.vm.box="hashicorp/precise64"
    vm5.vm.hostname="vm5"
    vm5.vm.network "private_network", ip: "192.168.2.1"
    vm5.vm.network "private_network", ip: "192.168.3.1"
  end

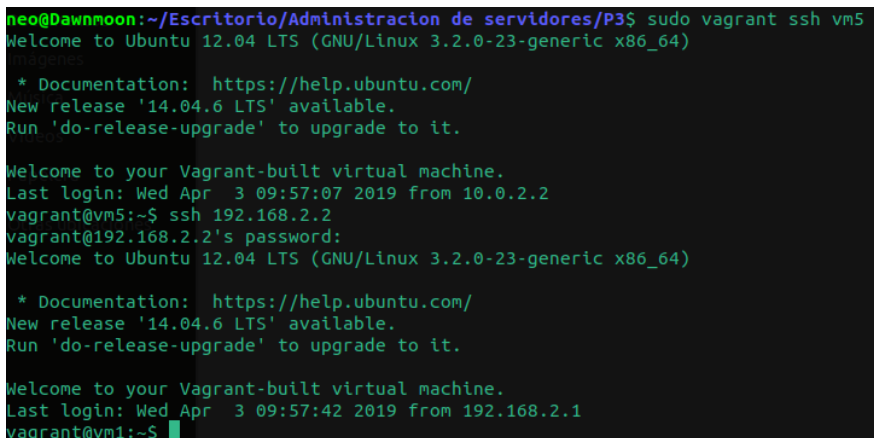
end
```

Ahora, nos conectaremos a cada máquina host (vm1 a vm4) con “`vagrant ssh vmX`”, y una vez dentro, vamos a permitir el acceso por SSH con contraseña, y reiniciar el servicio. Para ello, editaremos el fichero de configuración `/etc/ssh/sshd_config`, y dentro escribiremos (o en nuestro caso descomentaremos) la línea “`PasswordAuthentication yes`”. Para reiniciar el servicio, escribiremos en la consola “`sudo /etc/init.d/ssh restart`”.



```
vagrant@vm1: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
GNU nano 2.2.6 File: /etc/ssh/sshd_config Modified  
  
#IgnoreUserKnownHosts yes  
  
# To enable empty passwords, change to yes (NOT RECOMMENDED)  
PermitEmptyPasswords no  
  
# Change to yes to enable challenge-response passwords (beware issues with  
# some PAM modules and threads)  
ChallengeResponseAuthentication no  
  
# Change to no to disable tunnelled clear text passwords  
PasswordAuthentication yes  
  
# Kerberos options  
#KerberosAuthentication no  
#KerberosGetAFSToken no  
#KerberosOrLocalPasswd yes  
#KerberosTicketCleanup yes  
  
# GSSAPI options  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

A continuación, nos conectaremos desde la máquina vm5 a las demás mediante SSH con el comando “`ssh 192.168.X.X`” (contraseña vagrant en nuestro caso).



```
neo@Dawnmoon:~/Escritorio/Administracion de servidores/P3$ sudo vagrant ssh vm5  
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)  
  
* Documentation: https://help.ubuntu.com/  
New release '14.04.6 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Welcome to your Vagrant-built virtual machine.  
Last login: Wed Apr 3 09:57:07 2019 from 10.0.2.2  
vagrant@vm5:~$ ssh 192.168.2.2  
vagrant@192.168.2.2's password:  
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)  
  
* Documentation: https://help.ubuntu.com/  
New release '14.04.6 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Welcome to your Vagrant-built virtual machine.  
Last login: Wed Apr 3 09:57:42 2019 from 192.168.2.1  
vagrant@vm1:~$
```

Una vez dentro, apagaremos la interfaz de acceso al exterior que vagrant crea por defecto con “sudo ifconfig <nombre_interfaz>down” (la interfaz al exterior en nuestro caso es eth0). Tras esto, la máquina perderá la conexión al exterior hasta que configuremos todo el sistema de enrutamiento correctamente. Esto implica también que no se podrán establecer conexiones SSH directamente desde nuestra máquina, deberán hacerse desde el router.

```
vagrant@vm1:~$ ifconfig
eth0: Desc: Link encap:Ethernet HWaddr 08:00:27:88:0c:a6
      inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe88:ca6/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:1516 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1168 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:143254 (143.2 KB) TX bytes:122682 (122.6 KB)

eth1: Paper: Link encap:Ethernet HWaddr 08:00:27:62:42:ea
      inet addr:192.168.2.2 Bcast:192.168.2.255 Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe62:42ea/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:84 errors:0 dropped:0 overruns:0 frame:0
      TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:10976 (10.9 KB) TX bytes:8668 (8.6 KB)

lo: Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

vagrant@vm1:~$ ifconfig eth0 down
SIOCSIFFLAGS: Permission denied
vagrant@vm1:~$ sudo ifconfig eth0 down
vagrant@vm1:~$ ping 8.8.8.8
connect: Network is unreachable
```

Aprovecharemos dentro de cada máquina para cambiar la puerta de enlace predeterminada (default gateway) con “sudo route add default gw <IP_puertaEnlace><interfaz_hacia_puertaEnlace>” (en nuestro caso 192.168.2.1 y eth1 respectivamente).

```
vagrant@vm1:~$ sudo route add default gw 192.168.2.1 eth1
vagrant@vm1:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_req=1 ttl=64 time=0.362 ms
64 bytes from 192.168.2.1: icmp_req=2 ttl=64 time=0.391 ms
64 bytes from 192.168.2.1: icmp_req=3 ttl=64 time=0.339 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.339/0.364/0.391/0.021 ms
```

En el router, deberemos activar el IP forwarding para que pueda realizar su función. Para ello, modificaremos el archivo /proc/sys/net/ipv4/ip_forward, el cual contendrá un ‘0’, y lo cambiaremos por ‘1’.

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 File: /proc/sys/net/ipv4/ip_forward
1 0
documentos
```

Ahora, las máquinas de dos redes distintas deberían ser capaces de comunicarse.

```
vagrant@vm1:~$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_req=1 ttl=63 time=1.35 ms
64 bytes from 192.168.3.2: icmp_req=2 ttl=63 time=1.14 ms
64 bytes from 192.168.3.2: icmp_req=3 ttl=63 time=1.18 ms
^C
--- 192.168.3.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.143/1.228/1.354/0.095 ms
```

Ahora, vamos a configurar el router de forma que los paquetes de las demás máquinas puedan salir al exterior a través de él. Para ello, escribiremos el siguiente comando: “sudo iptables -t nat -A POSTROUTING -o <interfaz_al_exterior> -j MASQUERADE” (en nuestro caso la interfaz al exterior es eth0). Ahora las máquinas host pueden comunicarse con el exterior.

```
vagrant@vm5:~$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
vagrant@vm5:~$ ssh 192.168.2.2
vagrant@192.168.2.2's password:
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vm5:~$ ssh 192.168.2.2
vagrant@192.168.2.2's password:
Welcome to your Vagrant-built virtual machine.
Last login: Wed Apr  3 10:22:33 2019 from 192.168.2.1
vagrant@vm1:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=61 time=26.7 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=61 time=23.2 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=61 time=23.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 23.251/24.604/26.792/1.561 ms
```

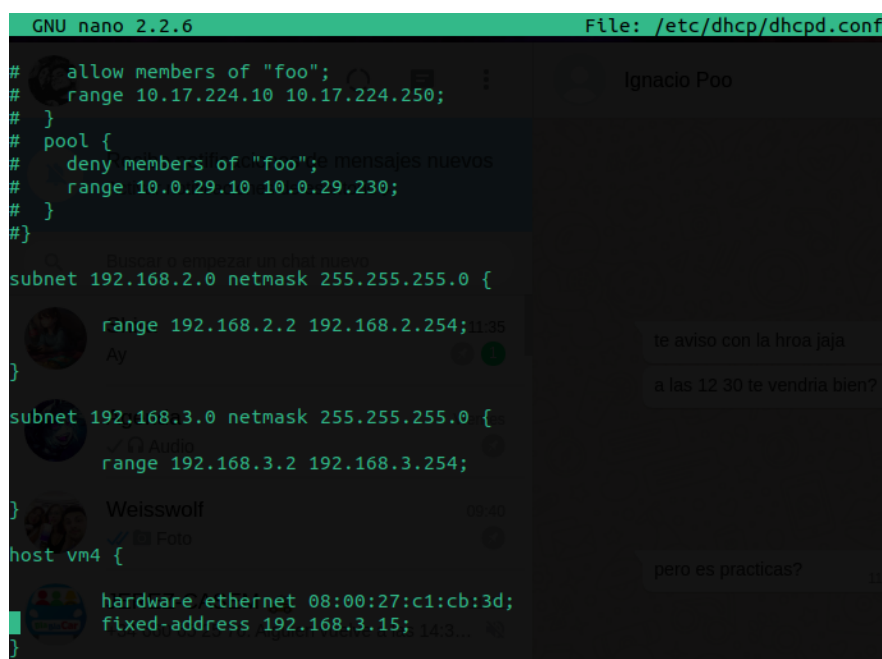
2. Ejercicio 3 DHCP

Entraremos en la maquina vm5, e instalaremos el servidor dhcp de la siguiente forma:

```
“sudo apt-get update”
```

```
“sudo apt-get install isc-dhcp-server”
```

Ahora se modificará el fichero de configuración con “sudo nano /etc/dhcp/dhcpd.conf”. Una vez dentro, estableceremos 2 áreas de subred, de forma que se le puedan prestar direcciones a máquinas de ambas redes. Además, estableceremos una dirección fija para la máquina vm4 de la segunda red, de forma que siempre le otorgue esa. Para ello, primero deberemos conectarnos a vm4 (se puede con vagrant ssh si no se conserva la configuración del ejercicio anterior, o bien conectarse por ssh a través de vm5 como antes). Ejecutaremos “ifconfig” para conocer la dirección MAC de la interfaz de red interna, la cual usaremos para identificar a quién hay que asignarle esa dirección. El fichero quedará de la siguiente forma (los intervalos y la ip concreta pueden ser los que se deseen):



```
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf
# allow members of "foo";
# range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo"; mensajes nuevos
#   range 10.0.29.10 10.0.29.230;
# }
#}
subnet 192.168.2.0 netmask 255.255.255.0 {
    range 192.168.2.2 192.168.2.254;
}
subnet 192.168.3.0 netmask 255.255.255.0 {
    range 192.168.3.2 192.168.3.254;
}
host vm4 {
    hardware ethernet 08:00:27:c1:cb:3d;
    fixed-address 192.168.3.15;
```

Reiniciaremos el servicio para que se cargue la nueva configuración con el comando:

```
“sudo /etc/init.d/isc-dhcp-server restart” o bien “sudo service isc-dhcp-server restart”.
```

Pasamos a conectarnos a las máquinas. En ellas, ejecutaremos “sudo dhclient -v”. Veremos la solicitud y respuesta del servidor con el préstamo, y con ifconfig se puede comprobar si la IP quedó asignada. Vm4 deberá haber quedado configurada con la IP fija que se estableció.

```
vagrant@vm3:~$ sudo dhclient -v
Internet Systems Consortium DHCP Client 4.1-ESV-R4
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/08:00:27:04:ec:f9
Sending on   LPF/eth1/08:00:27:04:ec:f9
Sending on   Socket/fallback
DHCPREQUEST of 192.168.3.4 on eth1 to 255.255.255.255 port 67
DHCPACK of 192.168.3.4 from 192.168.3.1
RTNETLINK answers: File exists
bound to 192.168.3.4 -- renewal in 296 seconds.
vagrant@vm3:~$
```

En el servidor pueden verse los préstamos con “sudo cat /var/lib/dhcp/dhcpd.leases”:

```
lease 192.168.3.4 {
  starts 3 2019/04/10 10:56:07;
  ends 3 2019/04/10 11:06:07;
  cltt 3 2019/04/10 10:56:07;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:04:ec:f9;
  client-hostname "vm3";
}
vagrant@vm5:~$
```


3. Ejercicio 4 DNS

En vm5 (máquina router), instalaremos el servidor DNS con “sudo apt-get install bind9”. A continuación nos movemos al directorio “/etc/bind”, y editaremos el fichero “named.conf.default-zones”, y crearemos una zona con el nombre que queramos, como la siguiente:

```
GNU nano 2.2.6 File: named.conf.default-zones

    type master;
    file "/etc/bind/db.0";
}; hay:
    pues con sudo
zone "255.in-addr.arpa" {
    con el type master;
    e no si file "/etc/bind/db.255";
}; enemos que incrementar en 1 el Serial.
    lo al." //Importante el punto al final otra vez
zone "as.uca.es" {
    type master;
    file "/etc/bind/db.as.uca.es";
};
```

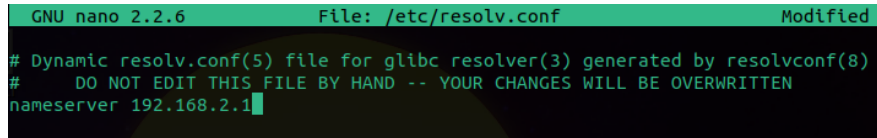
Ahora toca modificar el archivo de configuración; para no escribirlo desde cero, haremos una copia de uno existente como base, que tenga el nombre db. Seguido del nombre del area, con “sudo cp db.local db.as.uca.es”, y luego lo abrimos. Una vez dentro, escribiremos la configuración de forma similar a la imagen. Es importante aumentar el número designado como “Serial”, ahora y cada vez que hagamos cambios al archivo de forma que reconozca que han habido cambios y los tenga en cuenta.

```
GNU nano 2.2.6 File: db.as.uca.es

;
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA as.uca.es. root.as.uca.es. (
    4 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS main.as.uca.es.
main IN A 192.168.2.1
vm1.net1 IN A 192.168.2.2
vm2.net1 IN A 192.168.2.3
www IN A 192.168.3.2
ftp IN A 192.168.3.2
bd IN A 192.168.3.3
```

Ahora, ejecutaremos el comando “sudo named-checkconf”, si no muestra ningún resultado, la configuración es correcta. Luego ejecutaremos “sudo named-checkzone as.uca.es db.as.uca.es”, o el equivalente con nuestros nombres de zona. Deberá darnos un OK si todo ha ido bien. Ahora reiniciaremos el servicio con “sudo /etc/init.d/bind9 restart”.

Ahora nos conectaremos a las demás máquinas y abrimos el fichero “/etc/resolv.conf”, y cambiamos la ip que aparece por la que actúa como servidor DNS.



```
GNU nano 2.2.6      File: /etc/resolv.conf      Modified
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.2.1
```

Las máquinas donde cambiemos esto ya podrán referirse a las demás con estos nombres.