

# Proyecto Final CPD

Por: Rubén Calvo Villazán

# Índice:

Descripción de lo realizado .....	2
Pasos a seguir para la ejecución .....	2
Diagrama del cluster .....	3
Vagrantfile .....	4
Nodos activos (SSH) .....	5
Red de los nodos (IP) .....	5
Configuración Nginx LB .....	6
Prueba funcionamiento loadbalancer .....	6
Script de configuración rsync .....	7
Script de configuración Nginx como Load Balancer .....	8

# Realizado:

**Nodos:** 3 Servidores + 1 Balanceador + 1 Monitor = 5 Nodos

**Redundancia:** rsync entre los 3 servidores + rsync en el balanceador

**Monitorizacion:** Ansible en el nodo monitor

**Seguridad:** fail2ban, rkhunter, chrootkit, lynis

Despliegue realizado con **Vagrant** (<https://www.vagrantup.com/>)

## Pasos a seguir:

El total de ficheros incluye un Vagrantfile y una serie de scripts de configuración.

Crear un directorio, desplazar todos los ficheros dentro del directorio y lanzar Vagrant.

Ejecutar los scripts dentro de su nodo correspondiente para configurar el servicio.

```
$ mkdir ProyectoFinal
$ cd ProyectoFinal
$ ls ProyectoFinal
    Vagrantfile  scripts.sh
$ vagrant up
```

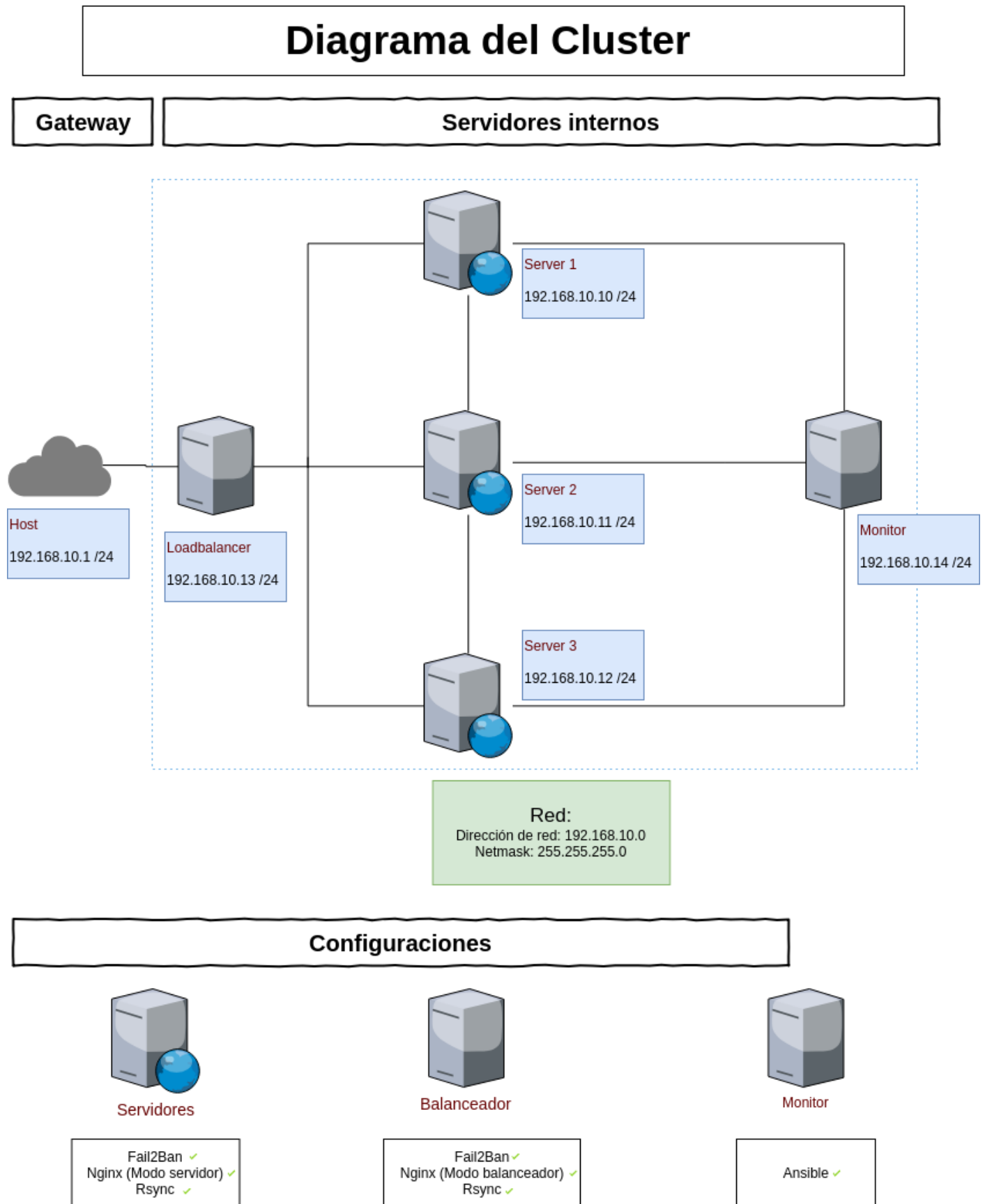
Dentro de cada máquina, los scripts se encuentran en el directorio vinculado del host

```
vagrant@server1 $ ls /vagrant/
    Vagrantfile  scripts.sh
```

Para ejecutar un script por ejemplo el que configura nginx en el balanceador:

```
vagrant@loadbalancer $ cp /vagrant/script.sh .
vagrant@loadbalancer $ ./script.sh
```

# Diagrama del Cluster:



# Vagrantfile:

```
14 ## Cluster de servidores
15
16 identificadores = [ 'server1', 'server2', 'server3', 'loadbalancer', 'monitor' ] # Añadir/Quitar nombres
17 servidores = []
18 num_servers = 5 # Cambiar si se añaden/quitan nodos (de forma dinamica)
19
20 dir_red = '192.168.10.' # 192.168.2. Para que no colapse con ninguna otra red WLAN
21 ip_inicial = 10 # Primera IP: 192.168.2.10
22
23
24 for i in 0..num_servers-1 do
25   name = identificadores[i]
26   ip = dir_red + (ip_inicial + i).to_s
27   servidores << { 'name' => name,
28                 'ip' => ip,
29                 }
30 end
31
32 Vagrant::Config.run do |config|
33   servidores.each do |server|
34     config.vm.define server['name'] do |config_server|
35       config_server.vm.box = "ubuntu/trusty64"
36       config_server.vm.host_name = server['name']
37       config_server.vm.network :hostonly, server['ip']
38
39       if server['name'] == 'monitor' # Si el nodo es monitor, instalamos ansible (monitorizar)
40         config_server.vm.provision :shell, :inline => "
41         echo 'SOY EL MONITOR'"
42         # apt install software-properties-common
43         # apt-add-repository --yes --update ppa:ansible/ansible
44         # apt install ansible"
45       else
46
47         # DESCOMENTAR ESTAS LINEAS PARA PROVISIONAR LAS MAQUINAS (TARDA MAS)
48         config_server.vm.provision :shell, :inline => "
49
50         echo 'INSTALANDO: FAIL2BAN, NGINX, RSYNC'"
51
52
53         # echo vagrant | sudo su -S ;
54         # apt update; apt install fail2ban -y ;
55         # apt install nginx -y;
56         # apt install rsync -y;
57
58
59         # apt install rkhunter -y; # Descomentar para añadir mas funcionalidad
60         # apt install chrootkit -y;
61         # apt install lynis -y;
62         #"
63       end
64
65
66
67       # NOTA:
68       # Aqui solo realizamos las instalaciones, no configuramos nada porque un solo fallo en configuracion de
69       # una maquina ya evita que se lancen y configuren las demas
70
71       # Ejecutar scripts o configurar manualmente cada maquina
72
73     end
74   end
75 end
```

# Nodos activos (SSH)

```
root@valkyrie: ~/CPD/ProyectoFinal

vagrant@server1: ~
t: https://landscape.canonical.com/
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
0 packages can be updated.
0 updates are security updates.
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Thu Dec 13 15:50:53 2018 from 10.0.2.2
vagrant@server1:~$ |

vagrant@server2: ~
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
0 packages can be updated.
0 updates are security updates.
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Thu Dec 13 15:51:02 2018 from 10.0.2.2
vagrant@server2:~$ |

vagrant@server3: ~
neric 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
0 packages can be updated.
0 updates are security updates.
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
vagrant@server3:~$ |

vagrant@monitor: ~
t: https://landscape.canonical.com/
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
0 packages can be updated.
0 updates are security updates.
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Thu Dec 13 15:50:49 2018 from 10.0.2.2
vagrant@monitor:~$ |

vagrant@loadbalancer: ~
eth0: 10.0.2.15
Swap usage: 0%
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
0 packages can be updated.
0 updates are security updates.
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
vagrant@loadbalancer:~$ |

root@valkyrie: ~/CPD/ProyectoFinal
root@valkyrie:ProyectoFinal# ls
T0-D0.txt Vagrantfile
root@valkyrie:ProyectoFinal#
```

# Red de los nodos (IP)

```
root@valkyrie: ~/CPD/ProyectoFinal

vagrant@server1: ~
vagrant@server1:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:85:1b:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.10/24 brd 192.168.10.255 scope global eth1
        valid_lft forever preferred_lft forever
vagrant@server1:~$ |

vagrant@server2: ~
vagrant@server2:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:bc:96:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.11/24 brd 192.168.10.255 scope global eth1
        valid_lft forever preferred_lft forever
vagrant@server2:~$ |

vagrant@server3: ~
vagrant@server3:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:20:24:e5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.12/24 brd 192.168.10.255 scope global eth1
        valid_lft forever preferred_lft forever
vagrant@server3:~$ |

vagrant@monitor: ~
vagrant@monitor:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:01:f8:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.14/24 brd 192.168.10.255 scope global eth1
        valid_lft forever preferred_lft forever
vagrant@monitor:~$ |

vagrant@loadbalancer: ~
vagrant@loadbalancer:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:e7:bc:b1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.13/24 brd 192.168.10.255 scope global eth1
        valid_lft forever preferred_lft forever
vagrant@loadbalancer:~$ |

root@valkyrie: ~/CPD/ProyectoFinal
root@valkyrie:ProyectoFinal#
```

# Configuración Nginx Load Balancer

```
root@server2: ~/home/vagrant/
vagrant@loadbalancer: ~
user www-data;
worker_processes 4;
pid /run/nginx.pid;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    upstream servidores {
        server 192.168.10.10;
        server 192.168.10.11;
        server 192.168.10.12;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://servidores;
        }

    }
}
```

## Prueba de funcionamiento del Load Balancer

```
root@valkyrie: ~/CPD/ProyectoFinal
t SERVER 1
root@valkyrie:ProyectoFinal# curl 192.168.10.13
SERVER 2
root@valkyrie:ProyectoFinal# curl 192.168.10.13
SERVER 3
root@valkyrie:ProyectoFinal# curl 192.168.10.13
SERVER 1
root@valkyrie:ProyectoFinal# curl 192.168.10.13
0 SERVER 2
0 root@valkyrie:ProyectoFinal# curl 192.168.10.13
f SERVER 3
root@valkyrie:ProyectoFinal# curl 192.168.10.13
e SERVER 1
root@valkyrie:ProyectoFinal# curl 192.168.10.13
SERVER 2
root@valkyrie:ProyectoFinal# curl 192.168.10.13
SERVER 3
root@valkyrie:ProyectoFinal# |
```

# Script de configuración de rsync (redundancia)

```
#!/bin/bash

# Script para configurar rsync

# Ejecutar en un nodo y pasarle la IP del otro nodo a vincular como argumento

sudo su
mkdir /var/www/

exec rsync -avzhie "ssh" vagrant@$1:/var/www/ /var/www/

# Crontab que lo ejecuta cada minuto

echo "* * * * *    root    rsync -avzhie 'ssh' vagrant@$1:/var/www/ /var/www/" >> /etc/crontab
```



# Script de configuración de Nginx como Load Balancer

```
1  #!/bin/bash
2
3
4  # Script para configurar nginx como un balanceador
5
6  # Ejecutar solo en el nodo loadbalancer
7
8
9  sudo su
10 cp /etc/nginx/nginx.conf /etc/nginx/nginxBK.conf
11
12 echo '
13     user www-data;
14     worker_processes 4;
15     pid /run/nginx.pid;
16
17     events {
18         worker_connections 768;
19         # multi_accept on;
20     }
21
22     http {
23
24         upstream servidores {
25             server 192.168.10.10;
26             server 192.168.10.11;
27             server 192.168.10.12;
28         }
29
30
31         server {
32             listen 80;
33
34             location / {
35                 proxy_pass http://servidores;
36             }
37         }
38     }
39 }' > /etc/nginx/nginx.conf
40
41 service nginx restart
```

