

UNIVERSIDAD DE COSTA RICA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS DE LA  
COMPUTACIÓN E INFORMÁTICA

CI-0138 – Integración de Infraestructura de TI  
Prof. Ricardo Villalón

Servicio de TI con infraestructura de balanceo y HA

Elaborado por:

Hellen Fernández Jiménez B42525

I Semestre del 2021

<b>Infraestructura tecnológica</b>	<b>2</b>
Direccionamiento ip	2
<b>Imagen base para servidores</b>	<b>3</b>
<b>Sitio web</b>	<b>10</b>
<b>Base de Datos</b>	<b>10</b>
<b>Balanceo de Cargas con tolerancia a fallos en el sitio web</b>	<b>10</b>
<b>Balanceo de cargas de la base de datos</b>	<b>10</b>

# Servicio de TI con infraestructura de balanceo y HA

Esta documentación corresponde al aprovisionamiento de un servicio de TI de base de datos con su respectivo servidor web que aloje una página web Wordpress. Este servicio se trabaja sobre otro previamente implementado al cual se le añaden las características de tolerancia a fallos, balanceo de carga y alta disponibilidad en la Nube Académica Computacional.

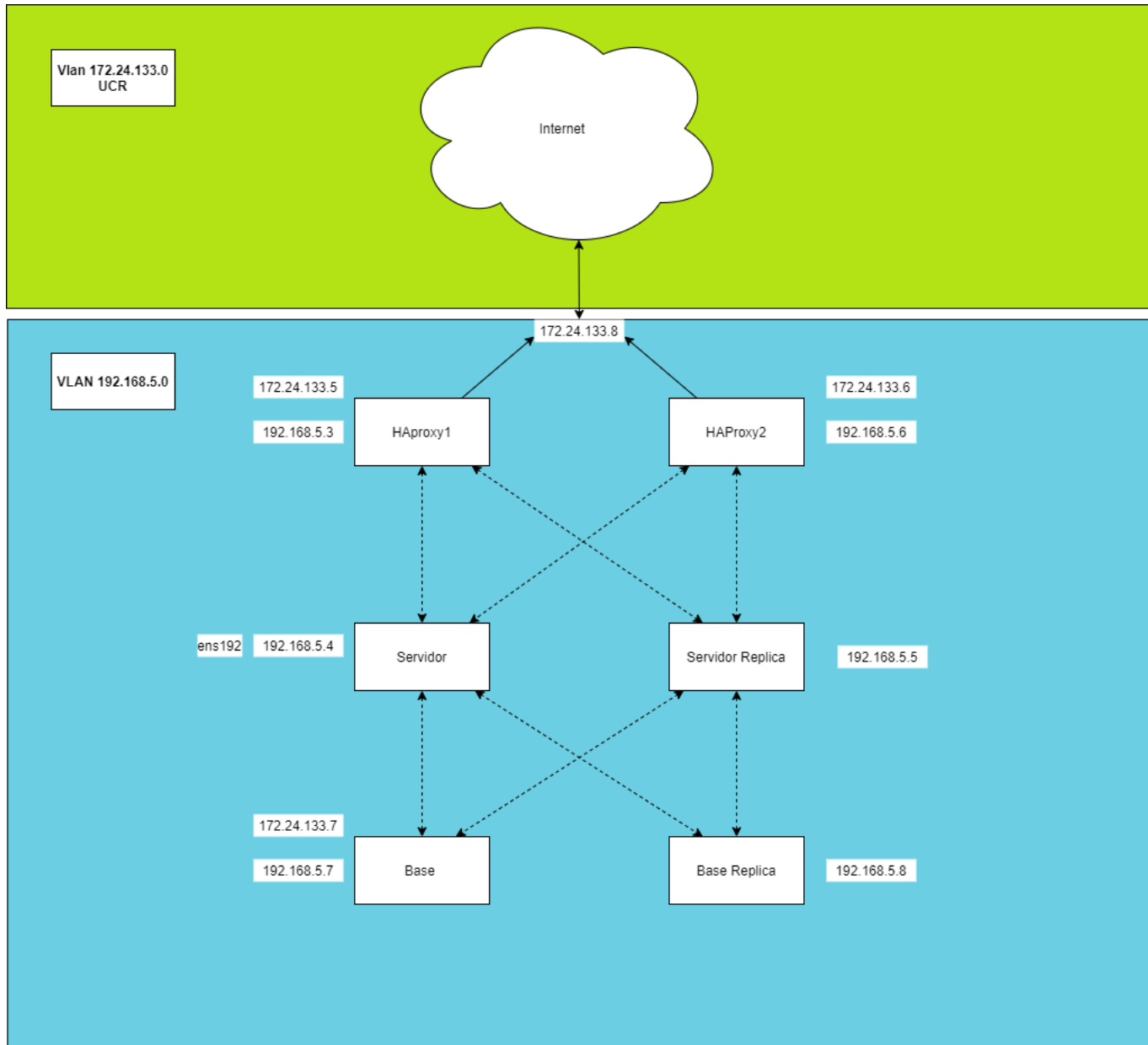
## 1. Infraestructura tecnológica

### Direccionamiento ip

A continuación se presenta el direccionamiento ip que se propone para la implementación de esta solución:

Equipo	Dirección	Red	Interfaz	Servicio
Base	192.168.5.7	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
	172.24.133.7		ens224	
Base réplica	192.168.5.8	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
Servidor web	192.168.5.4	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
Servidor web réplica	192.168.5.4	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
HAproxy1	192.168.5.3	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
	172.24.133.5	Vlan 172.24.133.0	ens224	Para ver las estadísticas de HAproxy y probar

				su funcionamiento
HAproxy2	192.168.5.3	Vlan 192.168.5.0	ens192	Comunicación interna con otros servidores.
	172.24.133.6	Vlan 172.24.133.0	ens224	Para ver las estadísticas de HAproxy y probar su funcionamiento.



## 2. Imagen base para servidores

La imagen base utilizada para estos servidores (web y base de datos), proviene de las máquinas virtuales que ya están corriendo los servicios de Wordpress sobre servidor web de Apache realizados en la tarea pasada sobre CentOS7 minimal. Es decir, se va a realizar una copia del

disco de la máquina de base de datos, para la réplica de la base de datos, y el mismo procedimiento para la máquina que aloja el servidor apache.

Para esto, se parte del hecho que ya se sabe cómo exportar imágenes de una máquina, y que hemos exportado los discos de las máquinas que queremos tomar como base. Estos son aquellos que tienen extensión `.vmdk`. Además que las máquinas tienen direcciones ips de acuerdo al direccionamiento ip presentado anteriormente. Además, para los servidores HAProxy se va a partir de la imagen base en la cual se tienen instalados varios servicios básicos para CentOS.

Usuario de las máquinas: hellen

Contraseña: maguie

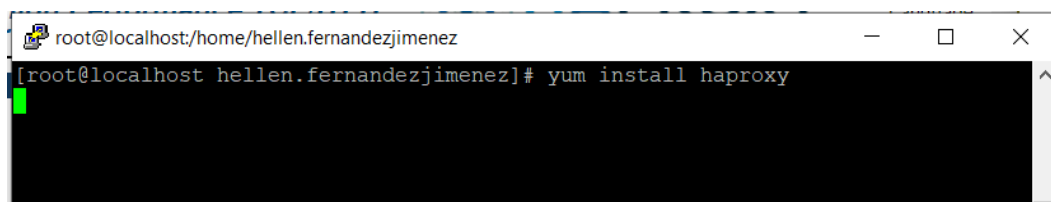
### 3. Balanceo de Cargas con tolerancia a fallos en el sitio web

#### Instalación de HAProxy

Para brindar el servicio de balanceo de cargas se va a implementar el servicio de HAProxy en dos servidores virtuales, los cuales gestionan mediante round robin el acceso de los clientes a los dos servidores web que alojan el sitio de wordpress. El método de round robin en este caso como algoritmo de balanceo va a alternar estableciendo conexiones hacia el servidor web y el servidor web de respaldo.

##### 1. Instalación de los paquetes de HAproxy

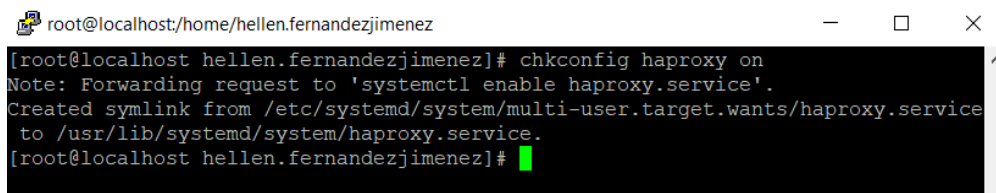
```
yum install haproxy -y
```



```
root@localhost/home/hellen.fernandezjimenez
[root@localhost hellen.fernandezjimenez]# yum install haproxy
```

##### 2. Verifique que HAProxy se inicie cada vez que reiniciamos nuestro servidor:

```
chkconfig haproxy on
```



```
root@localhost/home/hellen.fernandezjimenez
[root@localhost hellen.fernandezjimenez]# chkconfig haproxy on
Note: Forwarding request to 'systemctl enable haproxy.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/haproxy.service
to /usr/lib/systemd/system/haproxy.service.
[root@localhost hellen.fernandezjimenez]#
```

#### Configuración de HAProxy

Realice la siguiente configuración en una de las máquinas que será el HAProxy. Cuando la configuración total del HAProxy esté correcta, entonces realice una copia del disco para que este ya contenga lo que a continuación se va a realizar.

##### 1. Vaya a la siguiente ruta donde se encuentra la configuración del haproxy.

```
cd /etc/haproxy
```

2. Haga un respaldo de la configuración que está por defecto con el comando:

```
mv haproxy.cfg haproxy.cfg_bac
```

3. Modifique el archivo de configuración de haproxy con la siguiente información:

```
nano haproxy.cfg
```

```
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    stats timeout 30s
    user haproxy
    group haproxy
    daemon
defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000

#frontend
frontend http_front
    bind *:80
    stats uri /haproxy?stats
    default_backend http_back

#round robin balancing backend http
backend http_back
    balance roundrobin
#balance leastconn
mode http

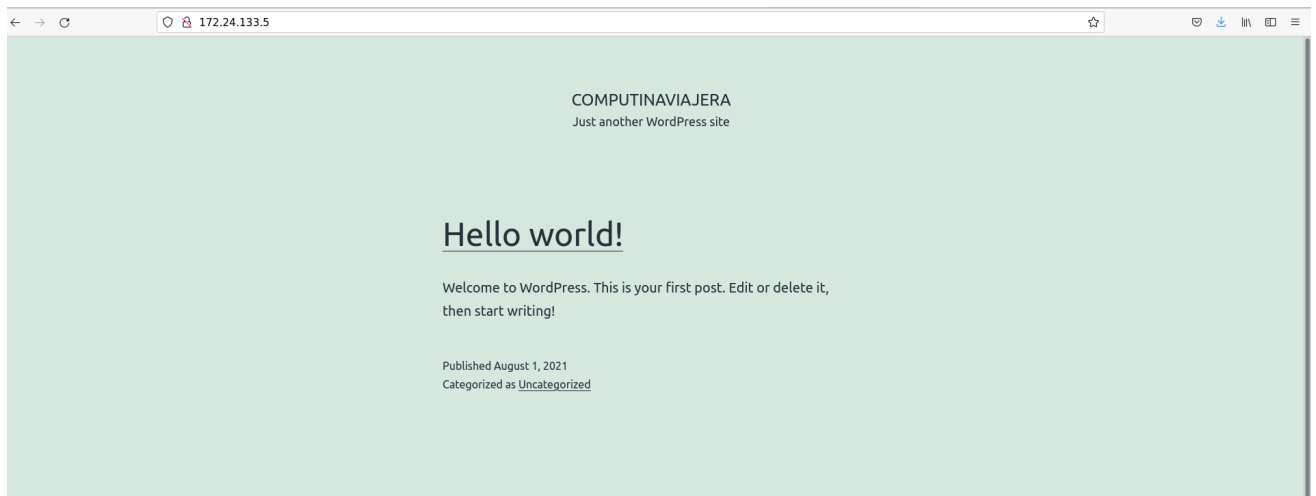
#ip address of 1st centos webserver
server webserver1 192.168.5.4:80 check

#ip address of 2nd centos webserver
server webserver2 192.168.5.5:80 check
```

4. Habilite el servicio de HAProxy

```
systemctl enable haproxy
systemctl start haproxy
```

5. Ingrese a la ip pública del HAProxy, en esta se deberá mostrar su página:



6. Observe que se muestran las estadísticas del HAProxy al ingresar a la siguiente página:

ipPublicaHAProxy/haproxy?stats

<http://172.24.133.5/haproxy?stats>

Apache HTTP Server Test x Log in - VMware ESXi x Log in - VMware ESXi x Telegram x Q How to configure HAProxy x Statistics Report for HAProxy +

172.24.133.5/haproxy?stats

HAProxy version 1.5.18, released 2016/05/10

Statistics Report for pid 13180

> General process information

pid = 13180 (process #1, nbproc = 1)  
uptime = 0d 0h0m0s  
system limits: memmax = unlimited; ulimit = 4023  
maxsock = 4023; maxconn = 2000; maxpipes = 0  
current conn = 1; current pipes = 0; conn rate = 0/sec  
Running tasks: 1/7; idle = 100 %

active UP  
active UP, going down  
active DOWN, going up  
active or backup DOWN  
active or backup DOWN for maintenance (MAINT)  
active or backup SOFT STOPPED for maintenance  
Note: "NOLB/DRAIN" = UP with load-balancing disabled.

Display option:  
Scope:   
Hide DOWN servers  
Refresh now  
CSV export

External resources:  
Primary site  
Updates v1.5  
Online manual

http front

	Queue			Session rate			Sessions			Total	LbTot	Last	In	Out	Denied	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dm	Downtime	Thrt
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit																						
Frontend	0	0	1	0	1	1	0	1	1	2000	2	1	3178	112549	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

http back

	Queue			Session rate			Sessions			Total	LbTot	Last	In	Out	Denied	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dm	Downtime	Thrt
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit																						
webserver1	0	0	-	0	1	1	0	1	1	1	1	7m6s	331	11514	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
webserver2	0	0	-	0	1	1	0	1	1	1	1	7m6s	283	380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Backend	0	0	-	0	1	1	0	1	1	200	2	2	7m6s	614	11894	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

7. Una vez que esto funcione en uno de los HAProxys, copie el disco para crear otro HAProxy con esta configuración, asegurándose de cambiar correctamente las direcciones ip de las máquinas. Nuevamente ahora muestre las estadísticas del haproxy entrando desde la máquina cliente:

<http://172.24.133.6/haproxy?stats>

Apache HTTP Server Test x

Log in - VMware ESXI x

Log in - VMware ESXI x

Telegram x

How to configure Wor x

Statistics Report for HAP x

+

←

→

↺

🔍

🔒

172.24.133.6/haproxy?stats

📄

📄

📄

📄

📄

☰

# HAProxy version 1.5.18, released 2016/05/10

## Statistics Report for pid 1045

> General process information

pid = 1045 (process #1, nproc = 1)  
uptime = 50 Dn(13m7s)  
system limits: memmax = unlimited; ulimit-n = 4023  
maxsock = 4023; maxconn = 2000; maxpipes = 0  
current conn = 1; current pipes = 100; conn rate = 1/sec  
Running tasks: 1/7; idle = 100 %

active UP  
active UP, going down  
active DOWN, going up  
active or backup DOWN  
not checked  
active or backup DOWN for maintenance (MAINT)  
active or backup SOFT STOPPED for maintenance  
Note: "NOLEAF/DRAIN" = UP with load-balancing disabled

Display option:

Scope:

• [Proxyman's life](#)

• [Updates \(v1.5\)](#)

• [haproxy DOWN servers](#)

• [StatsOnly](#)

• [CSV export](#)

External resources:

• [Proxyman's life](#)

• [Updates \(v1.5\)](#)

• [Online manual](#)

http front														http back														Server													
Queue				Session rate				Sessions						Bytes		Denied		Errors		Warnings		Status		LastChk		Weight		Server													
Cur	Max	Limit		Cur	Max	Limit		Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Conn	Resp	Retr	Redis			Act	Bck	Chk	Dwn	Downtime	Thrtle												
Frontend				1	1	-		1	1	-	2 000	3		152	23 028	0	0	0	0			OPEN																			
http back				Session rate				Sessions						Bytes		Denied		Errors		Warnings		Status		LastChk		Weight		Server													
Cur	Max	Limit		Cur	Max	Limit		Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Conn	Resp	Retr	Redis			Act	Bck	Chk	Dwn	Downtime	Thrtle												
webserver1	0	0	-	0	1	-		0	1	-	1	1	8m3s	76	11 514	0	0	0	0	0	0	13m7s UP	L4OK in 5ms	1	Y	-	0	0	-												
webserver2	0	0	-	0	1	-		0	1	-	1	1	7m47s	76	11 514	0	0	0	0	0	0	13m7s UP	L4OK in 1ms	1	Y	-	0	0	-												
Backend	0	0	-	0	1	-		0	1	-	200	2	7m47s	152	23 028	0	0	0	0	0	0	13m7s UP		2	2	0	0	0	-												

## 4. Establezca un servicio de alta disponibilidad con Keepalived

Para establecer el servicio de alta disponibilidad se implementará el servicio de keepalived. La función de Keepalived es detectar el estado del servidor. Si un servidor web falla o no funciona, Keepalived lo detectará y eliminará el servidor fallido del sistema. Al mismo tiempo, se utilizarán otros servidores para reemplazar el trabajo del servidor. Esto utiliza una dirección ip virtual que será la entrada a nuestro servicio de la página web. En este caso, esta dirección será la 172.24.133.8.

Realice los siguientes pasos para ambos HAProxys:

1. Instale keepalived con el siguiente comando:

```
yum install -y keepalived
```

2. Realice un respaldo de la configuración por defecto:

```
cd /etc/keepalived
mv keepalived.conf keepalived.conf.bak
touch keepalived.conf
nano keepalived.conf
```

3. Escriba las siguientes líneas en el archivo de configuración para el nodo maestro, en nuestro caso HAProxy1:

```
vrrp_instance VI_1 {
    state MASTER
    interface ens224 #put your interface name here.
virtual_router_id 51
    priority 255
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111 #password
    }
    virtual_ipaddress {
        172.24.133.8/24 # use the virtual ip address.
    }
}
```



```
}
```

4. Para la configuración del nodo secundario, en nuestro caso HAProxy2, se debe especificar que es un backup y tener una prioridad más baja que el nodo principal.

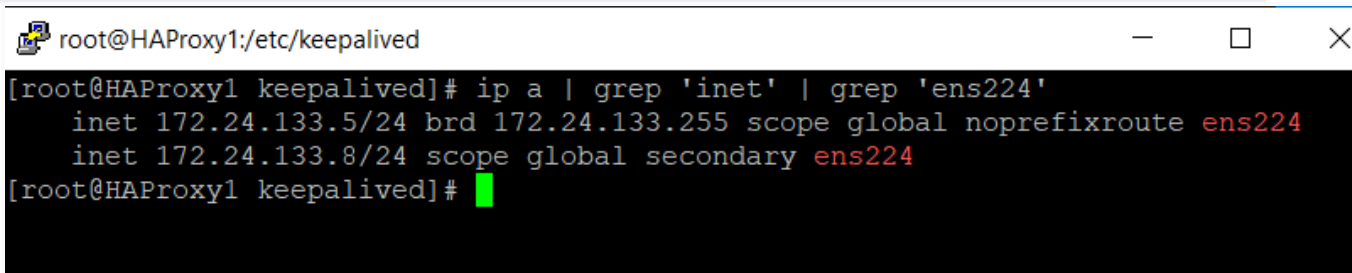
```
vrrp_instance VI_1 {
    state BACKUP
    interface ens224
    virtual_router_id 51
    priority 254
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111 #password
    }
    virtual_ipaddress {
        172.24.133.8/24 # use the virtual ip address.
    }
}
```

4. Inicie y habilite el servicio:

```
systemctl enable keepalived
systemctl start keepalived
```

5. Revise mediante el siguiente comando que el HAProxy maestro en la interfaz indicada se apodere de la ip virtual mediante el siguiente comando:

```
ip a | grep 'inet' | grep 'ens224'
```



```
root@HAProxy1:/etc/keepalived
[root@HAProxy1 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.5/24 brd 172.24.133.255 scope global noprefixroute ens224
    inet 172.24.133.8/24 scope global secondary ens224
[root@HAProxy1 keepalived]#
```

Como se puede notar, en la interfaz ens224, ahora existen 2 direcciones ips (172.24.133.8), la virtual y la del haproxy convencional.

5. También revise como el HAProxy 2 no tiene la ip virtual en este momento ya que es un Backup:

```
ip a | grep 'inet' | grep 'ens224'
```

```
root@HAProxy2:/etc/keepalived
[root@HAProxy2 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.6/24 brd 172.24.133.255 scope global noprefixroute ens224
[root@HAProxy2 keepalived]#
```

6. Realice la prueba de que el HAProxy1 deja de funcionar, para asegurarse de que el nodo Backup toma la ip virtual. Para esto, detenga el servicio de keepalived en el HAProxy1:

```
systemctl stop keepalived
ip a | grep 'inet' | grep 'ens224'
```

```
root@HAProxy1:/etc/keepalived
[root@HAProxy1 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.5/24 brd 172.24.133.255 scope global noprefixroute ens224
    inet 172.24.133.8/24 scope global secondary ens224
[root@HAProxy1 keepalived]# systemctl stop keepalived
[root@HAProxy1 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.5/24 brd 172.24.133.255 scope global noprefixroute ens224
[root@HAProxy1 keepalived]#
```

7. Ahora revise nuevamente las direcciones ip asignadas a la interfaz:

```
ip a | grep 'inet' | grep 'ens224'
```

```
root@HAProxy2:/etc/keepalived
[root@HAProxy2 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.6/24 brd 172.24.133.255 scope global noprefixroute ens224
[root@HAProxy2 keepalived]# ip a | grep 'inet' | grep 'ens224'
    inet 172.24.133.6/24 brd 172.24.133.255 scope global noprefixroute ens224
    inet 172.24.133.8/24 scope global secondary ens224
[root@HAProxy2 keepalived]#
```

8. Ahora detenga e inicie nuevamente los servicios para que estén funcionando correctamente:

```
systemctl stop keepalived
systemctl start keepalived
```

Guía para realizar la configuración tomada de:

[https://sysdaemons.com/implementation-of-php-based-site-on-a-web-server-nginx-along-with-setting-up-of-load-balancer-using-nginx-haproxy-and-keepalived/?cf\\_chl\\_jschl\\_tk=pmd\\_baf992a491c44df0f79c703](https://sysdaemons.com/implementation-of-php-based-site-on-a-web-server-nginx-along-with-setting-up-of-load-balancer-using-nginx-haproxy-and-keepalived/?cf_chl_jschl_tk=pmd_baf992a491c44df0f79c703)

## 9.Cluster de base de datos

En los servidores de bases de datos que tienen instalada MariaDB versión 10.4.20, que tienen la imagen utilizada para la tarea anterior, asegúrese de que en ambas base de datos se les garanticen los permisos suficientes a las direcciones ip de ambos servidores web para que puedan acceder a la base de datos destinada para wordpress. Para esto cree los usuarios y garantice los permisos:

```
mysql -u root -p
```

Digite la contraseña: 1UcHiT44

Cree los usuarios para que puedan ser accedidos de forma remota:

```
CREATE USER compuUsuaría@192.168.5.4 IDENTIFIED BY 'm4.GgUi3';  
CREATE USER compuUsuaría@192.168.5.5 IDENTIFIED BY 'm4.GgUi3';
```

Brinde los permisos para que estos usuarios puedan acceder a la base de datos:

```
GRANT ALL PRIVILEGES ON BaseViajera.* TO compuUsuaría@192.168.5.5 IDENTIFIED BY  
'm4.GgUi3';  
GRANT ALL PRIVILEGES ON BaseViajera.* TO compuUsuaría@192.168.5.4 IDENTIFIED BY  
'm4.GgUi3';
```

Para realizar un cluster de bases de datos con Galera se deben seguir los siguientes pasos:

1. Para sincronizar los servidores y controlar Linux con seguridad mejorada (SELinux), asegúrese de que estén instalados antes de continuar.

```
yum install rsync xinetd -y
```

2. Modifique el archivo de hosts de los nodos que van a conformar el cluster de galera con sus respectivos nombres y sus ips de red privada.

```
nano /etc/hosts
```

Los respectivos nombres serán:

Para Base: db1

Para Base Replica: db2

```
192.168.5.7 db1  
192.168.5.8 db2
```

```
root@BaseDatos:/etc/my.cnf.d
GNU nano 2.3.1 File: /etc/hosts Modified
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.5.7 db1
192.168.5.8 db2
```

3. Modifique el archivo de server.cnf para habilitar un cluster, guardando un respaldo de la configuración anterior:

```
cd /etc/my.cnf.d/
cp server.cnf server.cnf_backup
nano /etc/my.cnf.d/server.cnf
```

Complete con la siguiente configuración  
Para la máquina BaseDatos que tiene a db1

```
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="cluster_viajero"
wsrep_cluster_address="gcomm://192.168.5.7,192.168.5.8"
# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="192.168.5.7"
wsrep_node_name="db1"
```

Para la maquina ReplicaBaseDatos que tiene a db2

```
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Configuration
```

```
wsrep_on=ON
wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so

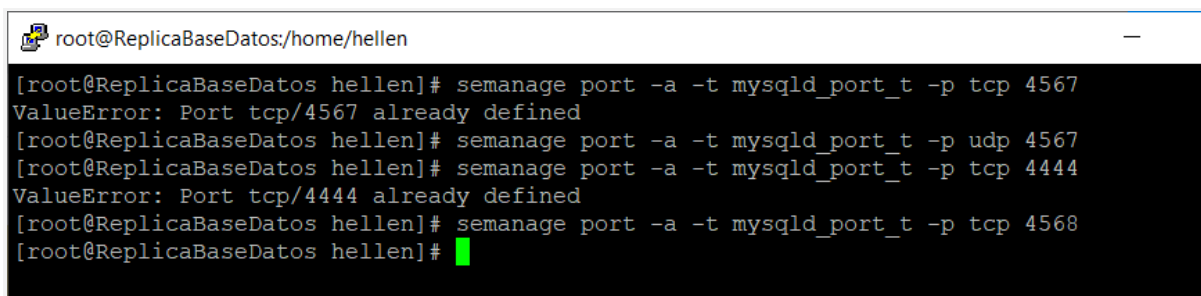
# Galera Cluster Configuration
wsrep_cluster_name="cluster_viajero"
wsrep_cluster_address="gcomm://192.168.5.7,192.168.5.8"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="192.168.5.8"
wsrep_node_name="db2"
```

4. Cree una política de linux que permita que los nodos del cluster puedan comunicarse entre sí. Permita el acceso a puertos relevantes ejecutando los siguientes comandos en los 2 servidores de base de datos:

```
sudo semanage port -a -t mysqld_port_t -p tcp 4567
sudo semanage port -a -t mysqld_port_t -p udp 4567
sudo semanage port -a -t mysqld_port_t -p tcp 4568
sudo semanage port -a -t mysqld_port_t -p tcp 4444
```



```
root@ReplicaBaseDatos:/home/hellen
[root@ReplicaBaseDatos hellen]# semanage port -a -t mysqld_port_t -p tcp 4567
ValueError: Port tcp/4567 already defined
[root@ReplicaBaseDatos hellen]# semanage port -a -t mysqld_port_t -p udp 4567
[root@ReplicaBaseDatos hellen]# semanage port -a -t mysqld_port_t -p tcp 4444
ValueError: Port tcp/4444 already defined
[root@ReplicaBaseDatos hellen]# semanage port -a -t mysqld_port_t -p tcp 4568
[root@ReplicaBaseDatos hellen]#
```

5. Configure el dominio MySQL SELinux en modo permisivo temporalmente en los dos servidores:

```
semanage permissive -a mysqld_t
```

6. Detenga el servidor de la base de datos en todos los nodos para que pueda arrancar el clúster de la base de datos con políticas SELinux compartidas.

```
systemctl stop mariadb
```

7. Ahora, inicie el clúster para generar eventos de comunicación entre nodos que se agregarán a la política de SELinux. En el primer nodo, inicie el clúster ejecutando:

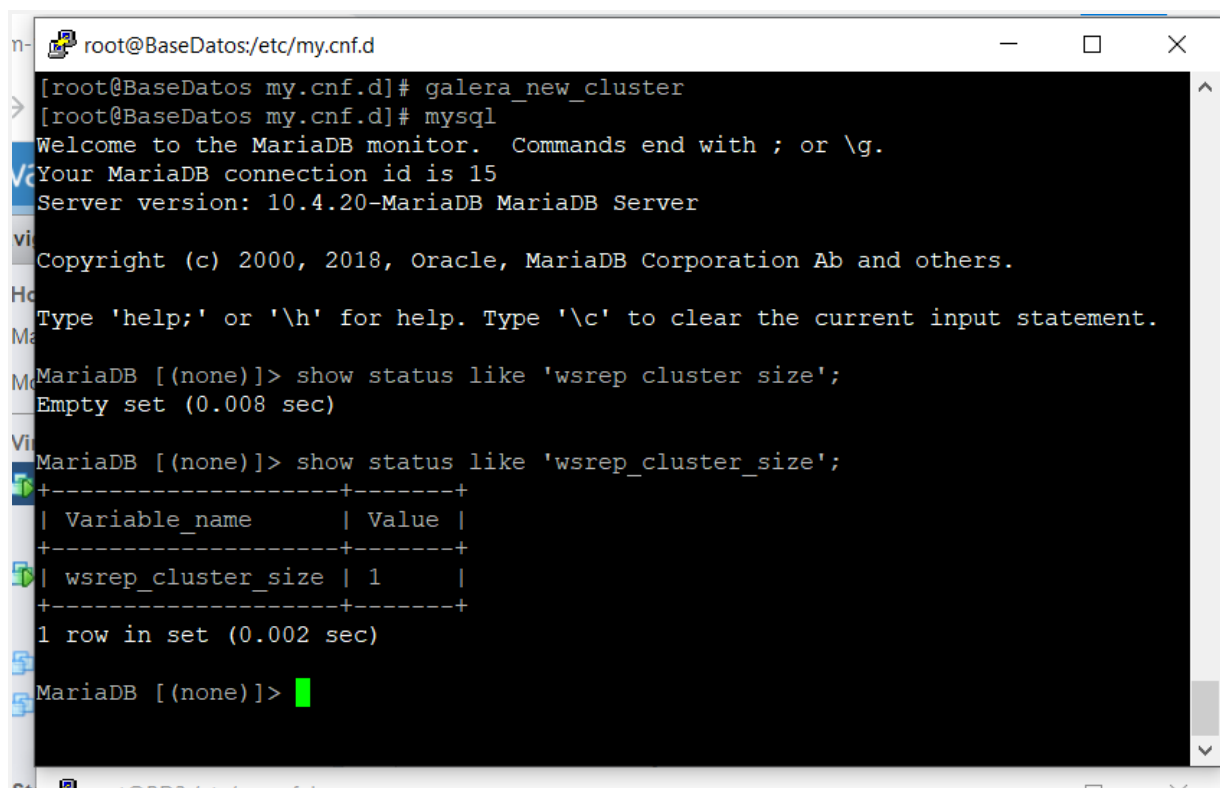
```
galera_new_cluster
```

Para asegurarse de que funciona, verifique el tamaño del cluster con los comandos:

mysql

```
show status like 'wsrep cluster size';
```

En este momento se debería indicar que sólo 1 nodo conforma el cluster, ya que el nodo de réplica aún no ha sido iniciado como se muestra en la siguiente figura:



```
root@BaseDatos:/etc/my.cnf.d
[root@BaseDatos my.cnf.d]# galera_new_cluster
[root@BaseDatos my.cnf.d]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.4.20-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show status like 'wsrep cluster size';
Empty set (0.008 sec)

MariaDB [(none)]> show status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 1      |
+-----+-----+
1 row in set (0.002 sec)

MariaDB [(none)]>
```

8. Ahora inicie el servidor en el segundo nodo:

```
systemctl start mariadb
```

Ahora, revise nuevamente en el primer nodo, como el valor del tamaño del cluster ahora es de 2.

```
root@BaseDatos:/etc/my.cnf.d
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show status like 'wsrep_cluster_size';
Empty set (0.008 sec)

MariaDB [(none)]> show status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 1 |
+-----+-----+
1 row in set (0.002 sec)

MariaDB [(none)]> show status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 2 |
+-----+-----+
1 row in set (0.003 sec)

MariaDB [(none)]> 
```

```
root@BD2:/etc/my.cnf.d
[root@BD2 my.cnf.d]# systemctl start mariadb
[root@BD2 my.cnf.d]# 
```

Ahora bien. Si se realiza un curl para verificar que la página esté funcionando a través de la ip virtual se realiza:

```
curl 172.24.133.8
```

