

PRÁCTICAS SOBRE CLUSTERS DE ALTA DISPONIBILIDAD Y BALANCEO DE CARGA

Facultad Ingeniería Informática



UNIVERSIDADE DA CORUÑA

Curso 2024/2025 - Ingeniería de Computadores

Enxeñaria de Infraestruturas Informaticas

Presentado por:

Yeray Lois Sánchez

Fecha de Entrega: 30 de Octubre de 2024

Índice

Práctica 1: Clúster de Alta Disponibilidad Básico con Pacemaker/Corosync	4
1. Introducción	5
Parte 1: Configuración de un Clúster HA de 2 Nodos	6
1.1. Comandos	7
Parte 2: Configuración Activo/Pasivo	8
1.2. Comandos	9
1.2.1. Primeros Pasos	9
1.2.2. Detención del nodo poseedor de la IP virtual (yls2425-slave)	13
1.2.3. Reinicio del nodo parado	14
1.2.4. Detención del otro nodo (yls2425-master)	16
1.2.5. Reinicio del nodo parado	18
1.3. Justificación	20
Parte 3: Configuración Activo/Activo	21
1.4. Comandos	21
1.5. Justificación	29
Parte 4: Configuración N+1	31
1.6. Comandos	31
1.7. Justificación	42
Práctica 2: Cluster HA con Almacenamiento replicado usando DRBD	43
2. Introducción	44
2.1. Objetivo	44
Parte 1: Configuración de un servidor web en el clúster HA	46
2.2. Comandos	46

2.2.1.	Comprobación del estado de partida	47
2.2.2.	Parada del master y migración de recursos al slave . . .	49
2.2.3.	Reinicio del master	50
2.2.4.	Parada del slave y reasignación de recursos al master .	51
2.2.5.	Reinicio del slave y verificación del estado final	52
2.3.	Justificación	54
Parte 2: Configuración de un RAID 1 usando DRBD		55
2.3.1.	Creación de discos virtuales para RAID 1	55
2.3.2.	Instalación de DRBD	56
2.4.	Comandos	56
2.4.1.	Comprobación del estado actual	56
2.4.2.	Cambio de roles primario/secundario	57
2.4.3.	Montaje del volumen en el slave	58
2.4.4.	Desmontaje del volumen y cambio de roles	59
2.5.	Justificación	60
Parte 3: Configuración de DRBD como recurso del clúster HA		61
2.6.	Comandos	62
2.6.1.	Comprobación del estado de partida	62
2.6.2.	Parada y reinicio del master	64
2.6.3.	Comprobación del estado tras el reinicio del master . .	65
2.6.4.	Parada y reinicio del slave	66
2.6.5.	Comprobación del estado tras el reinicio del slave . . .	67
2.7.	Justificación	68
Práctica 3: Balanceo de carga y medida del rendimiento		68
3. Introducción		70
Parte 1: Configuración de un balanceador de carga		71
3.1.	Comandos y Justificación	71
Parte 2: Medida del rendimiento de la granja de servidores		75
3.2.	Justificación	75
3.3.	Gráficas	76
3.4.	Explicación de los resultados	82

3.4.1.	Solicitudes por segundo (valores medios)	82
3.4.2.	Tiempo total (valores medios)	82
3.4.3.	Velocidad de transferencia (valores medios)	83
3.4.4.	Tiempo de conexión total (valores medios)	83
3.4.5.	Porcentaje de solicitudes (Percentil 50)	83

Práctica 1: Clúster de Alta Disponibilidad Básico con Pacemaker/Corosync

1 Introducción

El objetivo de esta práctica es configurar un clúster de alta disponibilidad (HA) utilizando Pacemaker y Corosync en un entorno de máquinas virtuales (VM) previamente configurado en la Práctica 0. A lo largo de esta práctica, se llevarán a cabo una serie de configuraciones que abarcan distintos modos de funcionamiento del clúster HA, desde la configuración básica de un clúster de dos nodos hasta configuraciones más avanzadas, como los modos activo/-pasivo y activo/activo.

Primero, se implementará un clúster básico con dos nodos, **master** y **slave**, para entender los conceptos fundamentales de la alta disponibilidad y los mecanismos de gestión de recursos. Posteriormente, se configurará un clúster en modo activo/pasivo, donde el **master** ofrecerá servicios mientras esté disponible, y el **slave** asumirá el control si se produce un fallo. A continuación, se abordará la configuración en modo activo/activo, donde ambos nodos estarán activos y se añadirá un nodo **spare** como respaldo, permitiendo la migración automática de recursos entre los nodos según la disponibilidad. Finalmente, se configurará el clúster en un esquema N+1 para gestionar los recursos con mayor flexibilidad y redundancia.

Durante la práctica, se evaluará el comportamiento del clúster en diferentes escenarios, como la migración de recursos, la adherencia de servicios a nodos específicos y la respuesta ante fallos. Al finalizar, se espera comprender en detalle el funcionamiento de un clúster HA, las estrategias de tolerancia a fallos y cómo gestionar eficazmente los recursos en un entorno distribuido.

Parte 1: Configuración de un Clúster HA de 2 Nodos

En esta primera parte, se lleva a cabo la configuración de un clúster de alta disponibilidad (HA) con dos nodos, **master** y **slave**, utilizando Pacemaker y Corosync. A continuación, se resumen los pasos principales:

- **Inicio de Nodos:** Se levantan las máquinas virtuales (**master** y **slave**) y se inicia sesión en el nodo **master**:

```
# vagrant up yls2425-master yls2425-slave
# vagrant ssh yls2425-master
```

- **Creación del Clúster:** Se crea un nuevo clúster HA especificando la red de comunicación (interfaz **eth1**). Durante la configuración, se establecen opciones como:
 - Configuración del usuario **hacluster**.
 - Configuración de SSH sin contraseña para la conexión entre nodos.
 - Configuración de la red y otras opciones (sin IP virtual para administración y sin testigo de quórum).
- **Verificación del Estado:** Se verifica la correcta configuración del clúster, comprobando el estado de los nodos y recursos con los comandos **crm**.
- **Desactivar Quórum:** Se desactiva la política de quórum para evitar problemas en un clúster de solo dos nodos.

1.1 Comandos

Para la configuración y verificación del clúster, se utilizaron los siguientes comandos:

```
# sudo crm configure show
```

```
vagrant@yls2425-master:~> sudo crm configure show
node 1: yls2425-master
node 2: yls2425-slave
property cib-bootstrap-options: \
  have-watchdog=false \
  dc-version="2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312" \
  cluster-infrastructure=corosync \
  cluster-name=yls2425-clusterHA \
  stonith-enabled=false \
  no-quorum-policy=ignore
rsc_defaults build-resource-defaults: \
  resource-stickiness=1 \
  migration-threshold=3 \
  priority=1
op_defaults op-options: \
  timeout=600 \
  record-pending=true
vagrant@yls2425-master:~>
```

Figura 1: Resultado del comando `sudo crm configure show`.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-master (version 2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f8831
2) - partition with quorum
* Last updated: Tue Oct 1 21:57:34 2024 on yls2425-master
* Last change: Tue Oct 1 21:44:21 2024 by root via root on yls2425-master
* 2 nodes configured
* 0 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
* No resources

vagrant@yls2425-master:~> |
```

Figura 2: Resultado del comando `sudo crm status`.

Parte 2: Configuración Activo/Pasivo

En esta segunda parte, se configura un clúster HA en modo activo/pasivo con los nodos **master** y **slave**. El servicio proporcionado será una dirección IP virtual (192.168.56.10). El funcionamiento final será el siguiente:

- El nodo **master** sirve la IP virtual siempre que esté activo.
- Si el nodo **master** falla, el nodo **slave** ofrecerá el servicio en su lugar.

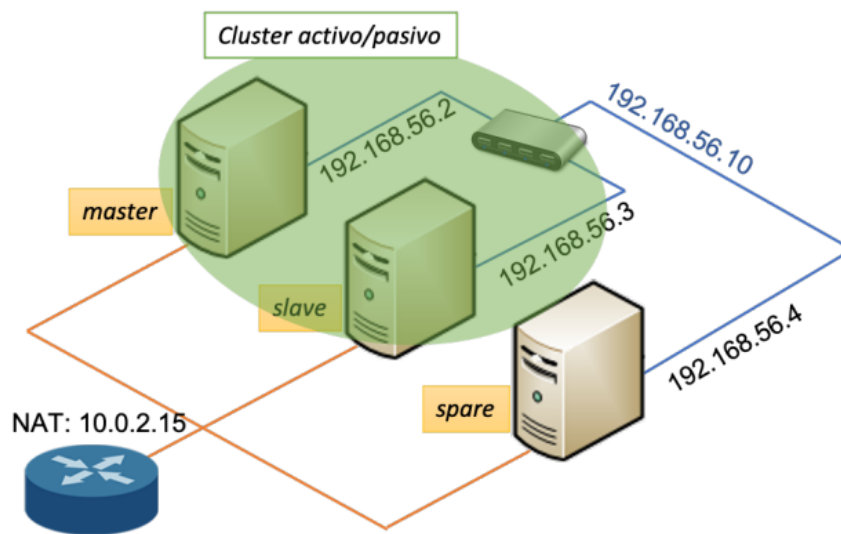


Figura 3: Esquema del clúster activo/pasivo con IP virtual 192.168.56.10.

La configuración de la IP virtual como recurso se realiza siguiendo estos pasos:

- **Configurar la IP virtual:** Crear el recurso de la IP virtual con los parámetros necesarios, como la dirección IP, la interfaz de red y las opciones de monitorización.
- **Verificar la configuración:** Comprobar el estado del clúster y la configuración del recurso para asegurar que se haya añadido correctamente.
- **Pruebas desde spare:** Iniciar el nodo **spare** y realizar una prueba de conexión (ping) para verificar la respuesta de la IP virtual.

- **Simular fallo del master:** Poner el nodo `master` en modo `standby` y comprobar que el recurso se migra automáticamente al nodo `slave`.
- **Configurar preferencia:** Establecer una preferencia para que la IP virtual se ejecute en el `master` siempre que esté disponible.
- **Configurar adherencia:** Ajustar el valor de adherencia para evitar migraciones innecesarias y mantener el recurso en su nodo actual, a menos que sea necesario.

1.2 Comandos

A continuación, se presenta la justificación de los resultados obtenidos al ejecutar los comandos para verificar y probar el comportamiento del clúster configurado:

1.2.1. Primeros Pasos

```
# sudo crm configure show
```

```
vagrant@yls2425-master:~> sudo crm configure show
node 1: yls2425-master
node 2: yls2425-slave
primitive yls2425-VIP IPAddr2 \
  params ip=192.168.56.10 nic=eth1 \
  op monitor interval=10s \
  meta is-managed=true
location prefer-slave yls2425-VIP 100: yls2425-slave
property cib-bootstrap-options: \
  have-watchdog=false \
  dc-version="2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312" \
  cluster-infrastructure=corosync \
  cluster-name=yls2425-clusterHA \
  stonith-enabled=false \
  no-quorum-policy=ignore
rsc_defaults build-resource-defaults: \
  resource-stickiness=100 \
  migration-threshold=3 \
  priority=1
op_defaults op-options: \
  timeout=600 \
  record-pending=true
vagrant@yls2425-master:~>
```

Figura 4: Resultado del comando `crm configure show` mostrando la configuración actual del clúster.

El comando anterior muestra que la configuración de la IP virtual y los nodos del clúster se ha realizado correctamente, incluyendo los parámetros definidos.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-master (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 14:13:09 2025 on yls2425-master
 * Last change: Thu Jan 30 14:11:54 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 1 resource instance configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-slave

vagrant@yls2425-master:~> |
```

Figura 5: Estado del clúster y los recursos después de la configuración inicial.

Este comando verifica que el recurso de la IP virtual está activo y actualmente asignado al nodo yls2425-slave.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-slave: 200
vagrant@yls2425-master:~> |
```

Figura 6: Puntuaciones de asignación de recursos, mostrando la preferencia actual del nodo yls2425-slave.

El estado de puntuación muestra que tanto nodo yls2425-slave y el

yls2425-master tienen la misma puntuación para el recurso, por lo que el recurso se encuentra en el **slave** por la adherencia.

```
spare:
# ping -c 1 192.168.56.10
```

```
vagrant@yls2425-spares:~> ping -c 1 192.168.56.10
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.72 ms

--- 192.168.56.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.716/1.716/1.716/0.000 ms
vagrant@yls2425-spares:~> |
```

Figura 7: Resultado del ping a la IP virtual desde el nodo spare.

El ping confirma que la IP virtual responde correctamente, indicando que el servicio está activo.

```
spare:
# ping -c 1 yls2425-master
```

```
vagrant@yls2425-spares:~> ping -c 1 yls2425-master
PING yls2425-master (192.168.56.2) 56(84) bytes of data.
64 bytes from yls2425-master (192.168.56.2): icmp_seq=1 ttl=64 time=1.59 ms

--- yls2425-master ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.590/1.590/1.590/0.000 ms
vagrant@yls2425-spares:~> |
```

Figura 8: Resultado del ping al nodo yls2425-master desde el nodo spare.

```
spare:
# ping -c 1 yls2425-slave
```

```
vagrant@yls2425-spares:~$ ping -c 1 yls2425-slave
PING yls2425-slave (192.168.56.3) 56(84) bytes of data:
64 bytes from yls2425-slave (192.168.56.3): icmp_seq=1 ttl=64 time=1.94 ms

--- yls2425-slave ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.939/1.939/1.939/0.000 ms
vagrant@yls2425-spares:~$
```

Figura 9: Resultado del ping al nodo yls2425-slave desde el nodo spare.

Estos ping confirma que el nodo yls2425-master y yls2425-slave están en línea y operativos.

```
spare:
# cat /proc/net/arp
```

```
vagrant@yls2425-spares:~$ cat /proc/net/arp
IP address      HW type    Flags     HW address    Mask     Device
10.0.2.3         0x1        0x2       52:54:00:12:35:03    *        eth0
10.0.2.2         0x1        0x2       52:54:00:12:35:02    *        eth0
192.168.56.3     0x1        0x2       08:00:27:c7:41:6d    *        eth1
192.168.56.2     0x1        0x2       08:00:27:0f:92:4c    *        eth1
192.168.56.10    0x1        0x2       08:00:27:c7:41:6d    *        eth1
vagrant@yls2425-spares:~$
```

Figura 10: Tabla ARP, mostrando la asociación de la IP virtual con la dirección MAC del nodo yls2425-slave.

La tabla ARP muestra que la IP virtual está asociada a la dirección MAC del nodo yls2425-slave, confirmando su asignación correcta.

1.2.2. Detención del nodo poseedor de la IP virtual (yls2425-slave)

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave  
INFO: standby node yls2425-slave
```

Figura 11: Nodo yls2425-master en modo standby, migrando el recurso al nodo yls2425-slave.

Al poner el nodo yls2425-slave en modo standby, el clúster migra automáticamente el recurso al nodo yls2425-master, tal como se espera.

```
spare:  
# ping -c 1 192.168.56.10
```

```
vagrant@yls2425-spares:~> ping -c 1 192.168.56.10  
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.  
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.58 ms  
  
--- 192.168.56.10 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.584/1.584/1.584/0.000 ms  
vagrant@yls2425-spares:~> |
```

Figura 12: Ping a la IP virtual después de migrar al nodo yls2425-master.

El ping confirma que la IP virtual sigue siendo accesible tras la migración al nodo yls2425-slave.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 14:24:13 2025 on yls2425-master
 * Last change: Thu Jan 30 14:22:01 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 1 resource instance configured

Node List:
 * Node yls2425-slave: standby
 * Online: [ yls2425-master ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master

vagrant@yls2425-master:~> |
```

Figura 13: Estado del clúster mostrando la IP virtual asignada al nodo yls2425-master.

Se observa que el recurso migró al único nodo online yls2425-master correctamente.

1.2.3. Reinicio del nodo parado

```
# sudo crm node online yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-slave
INFO: online node yls2425-slave
```

Figura 14: Reincorporación del nodo yls2425-slave al clúster.

Una vez que el nodo yls2425-slave vuelve a estar en línea, se observa que la IP virtual no migra de vuelta a él, debido a la configuración de la adherencia de nodo.

```
spare:
# ping -c 1 192.168.56.10
```

```
vagrant@yls2425-spare:~> ping -c 1 192.168.56.10
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.54 ms

--- 192.168.56.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.541/1.541/1.541/0.000 ms
vagrant@yls2425-spare:~> |
```

Figura 15: Ping a la IP virtual desde el nodo `spare`, confirmando que el servicio está activo.

El ping confirma que la IP virtual (192.168.56.10) responde correctamente desde el nodo `spare`, indicando que el servicio está operativo.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 14:26:20 2025 on yls2425-master
 * Last change: Thu Jan 30 14:23:25 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 1 resource instance configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master
vagrant@yls2425-master:~> |
```

Figura 16: Estado del clúster mostrando la IP virtual asignada al nodo `yls2425-master`.

El estado del clúster indica que el recurso de la IP virtual está activo y asignado al nodo **yls2425-master**.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-master: 100
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-slave: 100
vagrant@yls2425-master:~> |
```

Figura 17: Puntuaciones de asignación de recursos, mostrando la preferencia actual del nodo **yls2425-master** gracias a la adherencia.

En este momento, la puntuación refleja que gracias a la **adherencia**, tanto el nodo **yls2425-slave** como el **yls2425-master** presentan la misma puntuación, por tanto no es necesaria la migración del recurso y por eso permanece en **yls2425-master**

1.2.4. Detención del otro nodo (yls2425-master)

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 18: El nodo **yls2425-master** se pone en modo standby, migrando el recurso al nodo **yls2425-slave**.

Al poner el nodo **yls2425-master** en modo **standby**, el clúster migra automáticamente el recurso al nodo **yls2425-slave**.

```
spare:
# ping -c 1 192.168.56.10
```

```
vagrant@yls2425-spares:~$ ping -c 1 192.168.56.10
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.20 ms

--- 192.168.56.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.197/1.197/1.197/0.000 ms
vagrant@yls2425-spares:~$
```

Figura 19: Ping a la IP virtual después de la migración al nodo yls2425-slave.

El ping confirma que la IP virtual sigue siendo accesible después de la migración al nodo yls2425-slave.

```
# sudo crm status
```

```
vagrant@yls2425-master:~$ sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 14:28:17 2025 on yls2425-master
 * Last change: Thu Jan 30 14:26:14 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 1 resource instance configured

Node List:
 * Node yls2425-master: standby
 * Online: [ yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-slave
vagrant@yls2425-master:~$
```

Figura 20: Estado del clúster mostrando el recurso asignado al nodo yls2425-slave.

1.2.5. Reinicio del nodo parado

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-master  
INFO: online node yls2425-master
```

Figura 21: Reincorporación del nodo yls2425-master al clúster.

El nodo yls2425-master se reincorpora al clúster. Debido a la configuración de la adherencia, la IP virtual permanece en el nodo yls2425-slave.

```
spare:  
# ping -c 1 192.168.56.10
```

```
vagrant@yls2425-spares:~> ping -c 1 192.168.56.10  
PING 192.168.56.10 (192.168.56.10) 56(84) bytes of data.  
64 bytes from 192.168.56.10: icmp_seq=1 ttl=64 time=1.55 ms  
  
--- 192.168.56.10 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.548/1.548/1.548/0.000 ms  
vagrant@yls2425-spares:~> |
```

Figura 22: Ping a la IP virtual después de la reincorporación del nodo yls2425-master.

El ping muestra que la IP virtual sigue accesible y permanece en el nodo yls2425-slave tras la reincorporación del master debido a la adherencia.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 14:29:21 2025 on yls2425-master
 * Last change: Thu Jan 30 14:27:33 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 1 resource instance configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-slave
```

Figura 23: Estado del clúster mostrando el recurso asignado al nodo yls2425-slave.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP allocation score on yls2425-slave: 200
vagrant@yls2425-master:~> |
```

Figura 24: Puntuaciones de asignación tras la reincorporación del nodo yls2425-master.

Las puntuaciones de asignación reflejan la preferencia y adherencia configuradas, indicando que el recurso permanecerá en el nodo yls2425-slave hasta que haya un motivo fuerte para migrarlo.

1.3 Justificación

Se observó que, inicialmente, el recurso `yls2425-VIP` estaba asignado al nodo `slave`, debido a una prueba previa en la que se puso en modo *standby* al nodo `master`. Durante las pruebas, al poner los nodos en *standby*, los **resource scores** variaron para evitar la migración constante del recurso, gracias al valor de adherencia configurado.

Cuando se apagó el nodo `slave`, el recurso migró automáticamente al `master`, lo cual es el comportamiento esperado, ya que la IP virtual debe garantizar la integridad del servicio.

Tras reiniciar el nodo `slave`, la IP virtual se mantuvo en el `master`. Esto se debe a que, al consultar los **resource scores** con:

```
# sudo crm resource scores | grep allocation
```

ambos nodos tenían una puntuación de **100**. La igualdad de puntos impide la migración, a menos que haya una razón de fuerza mayor (por ejemplo, si la puntuación del `slave` fuese superior a 100).

Finalmente, los pings a la IP virtual siempre devolvieron respuestas exitosas, confirmando que la IP virtual se gestiona dinámicamente de forma correcta.

Parte 3: Configuración Activo/Activo

En esta tercera parte de la práctica, se utilizarán los mismos comandos empleados en la parte anterior para configurar un nuevo servicio en el clúster.

En esta ocasión, se añadirá una nueva IP virtual y se configurará el clúster de manera que ambos nodos, **master** y **slave**, estén activos simultáneamente. Cada nodo servirá una de las IPs virtuales y actuará como pasivo del otro en caso de fallo.

Los pasos a seguir en esta configuración son:

- Primero, se detiene y elimina el recurso `yls2425-VIP` configurado en la parte anterior.
- Luego, se crean dos nuevas IPs virtuales: `yls2425-VIP1` y `yls2425-VIP2`.
- La configuración del clúster se realiza de la siguiente manera:
 - El nodo **master** ofrecerá el servicio `yls2425-VIP1` siempre que esté disponible.
 - El nodo **slave** ofrecerá el servicio `yls2425-VIP2` siempre que esté disponible.
 - Si el nodo **master** falla, el nodo **slave** tomará el control de `yls2425-VIP1`.
 - Si el nodo **slave** falla, el nodo **master** se hará cargo de `yls2425-VIP2`.
- Finalmente, desde el nodo **spare**, se probará la conexión con ambas IPs virtuales utilizando `ping`, para comprobar que el funcionamiento del clúster es el esperado.

1.4 Comandos

A continuación, se muestran los comandos ejecutados y las salidas obtenidas:

```
# sudo crm configure show
```

```
vagrant@yls2425-master:~> sudo crm configure show
node 1: yls2425-master \
  attributes standby=off
node 2: yls2425-slave \
  attributes standby=off
primitive yls2425-VIP1 IPAddr2 \
  params ip=192.168.56.11 nic=eth1 \
  op monitor interval=10s \
  meta is-managed=true
primitive yls2425-VIP2 IPAddr2 \
  params ip=192.168.56.12 nic=eth1 \
  op monitor interval=10s \
  meta is-managed=true
location prefer-master yls2425-VIP1 100: yls2425-master
location prefer-slave yls2425-VIP2 100: yls2425-slave
property cib-bootstrap-options: \
  have-watchdog=false \
  dc-version="2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312" \
  cluster-infrastructure=corosync \
  cluster-name=yls2425-clusterHA \
  stonith-enabled=false \
  no-quorum-policy=ignore
rsc_defaults build-resource-defaults: \
  resource-stickiness=1 \
  migration-threshold=3 \
  priority=1
op_defaults op-options: \
  timeout=600 \
  record-pending=true
vagrant@yls2425-master:~> |
```

Figura 25: Configuración actual del clúster.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
* Last updated: Thu Jan 30 11:44:42 2025 on yls2425-master
* Last change: Thu Jan 30 11:42:01 2025 by root via root on yls2425-master
* 2 nodes configured
* 2 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
* yls2425-VIP1 (ocf::heartbeat:IPAddr2): Started yls2425-master
* yls2425-VIP2 (ocf::heartbeat:IPAddr2): Started yls2425-slave
vagrant@yls2425-master:~> |
```

Figura 26: Estado actual del clúster mostrando los recursos y su ubicación.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 101
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 101
```

Figura 27: Puntuaciones de asignación de recursos en los nodos.

```
spare:
# ping -c 1 192.168.56.11
# ping -c 1 192.168.56.12
# ping -c 1 yls2425-master
# ping -c 1 yls2425-slave
```

```
vagrant@yls2425-spares:~> ping -c 1 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=8.18 ms

--- 192.168.56.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 8.181/8.181/8.181/0.000 ms
vagrant@yls2425-spares:~> ping -c 1 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=5.72 ms

--- 192.168.56.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.717/5.717/5.717/0.000 ms
vagrant@yls2425-spares:~> ping -c 1 yls2425-master
PING yls2425-master (192.168.56.2) 56(84) bytes of data.
64 bytes from yls2425-master (192.168.56.2): icmp_seq=1 ttl=64 time=4.08 ms

--- yls2425-master ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.084/4.084/4.084/0.000 ms
vagrant@yls2425-spares:~> ping -c 1 yls2425-slave
PING yls2425-slave (192.168.56.3) 56(84) bytes of data.
64 bytes from yls2425-slave (192.168.56.3): icmp_seq=1 ttl=64 time=2.89 ms

--- yls2425-slave ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.886/2.886/2.886/0.000 ms
vagrant@yls2425-spares:~>
```

Figura 28: Pings a las IPs virtuales y nodos desde spare.


```
spare:
# cat /proc/net/arp
```

```
vagrant@yls2425-spare:~> cat /proc/net/arp
IP address      HW type        Flags          HW address     Mask           Device
192.168.56.11   0x1            0x2           08:00:27:f5:ce:b6  *             eth1
192.168.56.12   0x1            0x2           08:00:27:96:35:4c  *             eth1
192.168.56.3    0x1            0x2           08:00:27:96:35:4c  *             eth1
192.168.56.2    0x1            0x2           08:00:27:f5:ce:b6  *             eth1
10.0.2.2        0x1            0x2           52:54:00:12:35:02  *             eth0
10.0.2.3        0x1            0x2           52:54:00:12:35:03  *             eth0
vagrant@yls2425-spare:~>
```

Figura 29: Tabla ARP mostrando las direcciones IP y sus asociaciones con las direcciones MAC.

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 30: El nodo yls2425-master en modo *standby*, migrando los recursos.

```
spare:
# ping -c 1 192.168.56.11
# ping -c 1 192.168.56.12
```

```
vagrant@yls2425-spare:~> ping -c 1 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=2.89 ms

--- 192.168.56.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 2.888/2.888/2.888/0.000 ms
vagrant@yls2425-spare:~> ping -c 1 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=2.60 ms

--- 192.168.56.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.604/2.604/2.604/0.000 ms
vagrant@yls2425-spare:~> |
```

Figura 31: Pings a las IPs virtuales después de migrar al nodo *slave*.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 11:56:30 2025 on yls2425-master
 * Last change: Thu Jan 30 12:13:43 2025 by root via root on yls2425-slave
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-master: standby
 * Online: [ yls2425-slave ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-slave
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
vagrant@yls2425-master:~> |
```

Figura 32: Estado del clúster después de poner el nodo master en modo *standby*.

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node online yls2425-master
INFO: online node yls2425-master
```

Figura 33: Reincorporación del nodo yls2425-master al clúster.

```
spare:
# ping -c 1 192.168.56.11
# ping -c 1 192.168.56.12
```

```
vagrant@yls2425-spares:~> ping -c 1 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=2.87 ms

--- 192.168.56.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.869/2.869/2.869/0.000 ms
vagrant@yls2425-spares:~> ping -c 1 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=2.30 ms

--- 192.168.56.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.295/2.295/2.295/0.000 ms
vagrant@yls2425-spares:~> |
```

Figura 34: Pings a las IPs virtuales después de reincorporar el nodo yls2425-master.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 11:57:46 2025 on yls2425-master
 * Last change: Thu Jan 30 12:01:42 2025 by root via root on yls2425-slave
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave

vagrant@yls2425-master:~> |
```

Figura 35: Estado del clúster tras la reincorporación del nodo master.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 101
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 101
```

Figura 36: Puntuaciones de asignación de recursos tras la reincorporación del nodo master.

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave
INFO: standby node yls2425-slave
```

Figura 37: El nodo yls2425-slave en modo *standby*, migrando los recursos.

```
spare:
# ping -c 1 192.168.56.11
# ping -c 1 192.168.56.12
```

```
vagrant@yls2425-spare:~> ping -c 1 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=3.57 ms

--- 192.168.56.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.566/3.566/3.566/0.000 ms
vagrant@yls2425-spare:~> ping -c 1 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=2.75 ms

--- 192.168.56.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.747/2.747/2.747/0.000 ms
vagrant@yls2425-spare:~> |
```

Figura 38: Pings a las IPs virtuales tras poner en standby el nodo slave.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave
INFO: standby node yls2425-slave
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 11:59:23 2025 on yls2425-master
 * Last change: Thu Jan 30 12:10:21 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-slave: standby
 * Online: [ yls2425-master ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-master
vagrant@yls2425-master:~> |
```

Figura 39: Estado del clúster tras poner en standby el nodo slave.

```
# sudo crm node online yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-slave  
INFO: online node yls2425-slave
```

Figura 40: Reincorporación del nodo yls2425-slave al clúster.

```
spare:  
# ping -c 1 192.168.56.11  
# ping -c 1 192.168.56.12
```

```
vagrant@yls2425-spare:~> ping -c 1 192.168.56.11  
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.  
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=2.66 ms  
  
--- 192.168.56.11 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 2.658/2.658/2.658/0.000 ms  
vagrant@yls2425-spare:~> ping -c 1 192.168.56.12  
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.  
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=2.94 ms  
  
--- 192.168.56.12 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 2.937/2.937/2.937/0.000 ms  
vagrant@yls2425-spare:~> |
```

Figura 41: Pings a las IPs virtuales tras reincorporar el nodo yls2425-slave.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 12:00:57 2025 on yls2425-master
 * Last change: Thu Jan 30 12:14:44 2025 by root via root on yls2425-master
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP1      (ocf::heartbeat:IPaddr2):      Started yls2425-master
 * yls2425-VIP2      (ocf::heartbeat:IPaddr2):      Started yls2425-slave

vagrant@yls2425-master:~> |
```

Figura 42: Estado del clúster tras la reincorporación del nodo slave.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 101
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 101
vagrant@yls2425-master:~> |
```

Figura 43: Puntuaciones de asignación de recursos tras la reincorporación del nodo slave.

1.5 Justificación

Al ejecutar la secuencia de comandos, se observó que inicialmente las IPs virtuales yls2425-VIP1 y yls2425-VIP2 estaban en los nodos preferidos: yls2425-VIP1 en yls2425-master y yls2425-VIP2 en yls2425-slave. Al realizar pings desde el nodo spare, ambas IPs respondieron correctamente, lo que indica un funcionamiento correcto del clúster.

Después, al poner en modo *standby* al nodo **yls2425-master**, la IP **yls2425-VIP1** migró al nodo **yls2425-slave**, permitiendo que ambos recursos siguieran accesibles. Al devolver el nodo **yls2425-master** a su estado activo, se verificó que la IP asociada al recurso **yls2425-VIP1** regresó al nodo preferido (**yls2425-master**) concordando con lo indicado en la configuración requerida, que establece una preferencia sobre este recurso a dicho nodo.

Al repetir la prueba con el nodo **yls2425-slave**, la IP **yls2425-VIP2** migró al nodo **yls2425-master** cuando el **slave** entró en modo *standby*.

Al devolver el nodo **yls2425-slave** a su estado activo, se verificó que la IP **yls2425-VIP2** regresó automáticamente a su nodo preferido (**yls2425-slave**), siguiendo la configuración definida. Este comportamiento valida que la configuración implementada garantiza que cada nodo gestione su IP principal siempre que esté disponible y que, en caso de fallo, el otro nodo asuma temporalmente la dirección correspondiente hasta que se restablezca la normalidad.

Parte 4: Configuración N+1

En esta cuarta parte de la práctica, se añadirá el nodo **spare** al clúster HA y se configurará para que el **master** y el **slave** sean nodos activos, mientras que el **spare** actuará como un nodo pasivo de respaldo para ambos.

Utilizaremos los mismos servicios (las dos direcciones IP virtuales) que se configuraron en la segunda parte.

Los pasos a seguir son:

- Iniciar el nodo **spare** y unirlo al clúster.
- Configurar el funcionamiento del clúster con la siguiente lógica:
 - El nodo **master** ofrecerá el servicio **yls2425-VIP1** siempre que esté disponible.
 - El nodo **slave** ofrecerá el servicio **yls2425-VIP2** siempre que esté disponible.
 - Si el nodo **master** falla, el **spare** asumirá el control de **yls2425-VIP1**. Si el **spare** falla, el **slave** tomará el control. Se evitarán migraciones innecesarias del **slave** al **spare**, pero no se impedirá la migración de vuelta al **master** cuando esté disponible nuevamente.
 - Si el nodo **slave** falla, el **spare** tomará el control de **yls2425-VIP2**. Si el **spare** falla, el **master** se encargará del servicio. De igual forma, se evitarán migraciones innecesarias del **master** al **spare**, pero se permitirá la migración de vuelta al **slave** cuando esté disponible.
- Finalmente, verificar que el comportamiento del clúster es el esperado según estas configuraciones.

1.6 Comandos

A continuación, se muestran los comandos ejecutados y las salidas obtenidas:


```
# sudo crm configure show
```

```
vagrant@yls2425-slave:~> sudo crm configure show
node 1: yls2425-master \
  attributes standby=off
node 2: yls2425-slave \
  attributes standby=off
node 3: yls2425-spare \
  attributes standby=off
primitive yls2425-VIP1 IPAddr2 \
  params ip=192.168.56.11 nic=eth1 \
  op monitor interval=10s \
  meta is-managed=true
primitive yls2425-VIP2 IPAddr2 \
  params ip=192.168.56.12 nic=eth1 \
  op monitor interval=10s \
  meta is-managed=true
location loc_vip1_master yls2425-VIP1 200: yls2425-master
location loc_vip1_slave yls2425-VIP1 0: yls2425-slave
location loc_vip1_spare yls2425-VIP1 50: yls2425-spare
location loc_vip2_master yls2425-VIP2 0: yls2425-master
location loc_vip2_slave yls2425-VIP2 200: yls2425-slave
location loc_vip2_spare yls2425-VIP2 50: yls2425-spare
property cib-bootstrap-options: \
  have-watchdog=false \
  dc-version="2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312" \
  cluster-infrastructure=corosync \
  cluster-name=yls2425-clusterHA \
  stonith-enabled=false \
  no-quorum-policy=ignore
rsc_defaults build-resource-defaults: \
  resource-stickiness=100 \
  migration-threshold=3 \
  priority=0
op_defaults op-options: \
  timeout=600 \
  record-pending=true
vagrant@yls2425-slave:~>
```

Figura 44: Configuración actual del clúster.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 12:53:45 2025 on yls2425-slave
 * Last change: Thu Jan 30 12:48:11 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1      (ocf::heartbeat:IPaddr2):      Started yls2425-master
 * yls2425-VIP2      (ocf::heartbeat:IPaddr2):      Started yls2425-slave

vagrant@yls2425-slave:~> |
```

Figura 45: Estado actual del clúster mostrando los recursos y su ubicación.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-slave:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 300
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-spare: 50
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 300
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-spare: 50
```

Figura 46: Puntuaciones de asignación de recursos en los nodos.

```
spare:
# ping -c 1 192.168.56.11
# ping -c 1 192.168.56.12
```

```
vagrant@yls2425-spare:~> ping -c 1 192.168.56.11
PING 192.168.56.11 (192.168.56.11) 56(84) bytes of data.
64 bytes from 192.168.56.11: icmp_seq=1 ttl=64 time=1.49 ms

--- 192.168.56.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.489/1.489/1.489/0.000 ms
vagrant@yls2425-spare:~> ping -c 1 192.168.56.12
PING 192.168.56.12 (192.168.56.12) 56(84) bytes of data.
64 bytes from 192.168.56.12: icmp_seq=1 ttl=64 time=0.846 ms

--- 192.168.56.12 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.846/0.846/0.846/0.000 ms
vagrant@yls2425-spare:~> |
```

Figura 47: Pings a las IPs virtuales desde el nodo spare.

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 48: El nodo yls2425-master en modo *standby*.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 12:59:30 2025 on yls2425-slave
 * Last change: Thu Jan 30 12:58:26 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-master: standby
 * Online: [ yls2425-slave yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-spare
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
```

Figura 49: Estado del clúster tras poner en standby el nodo master.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-slave:~> sudo crm resource scores | grep allocation
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 200
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-spare: 150
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 300
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-spare: 50
```

Figura 50: Puntuaciones de asignación de recursos tras la migración.

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node online yls2425-master
INFO: online node yls2425-master
```

Figura 51: Reincorporación del nodo yls2425-master al clúster.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 13:06:48 2025 on yls2425-slave
 * Last change: Thu Jan 30 13:05:40 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1      (ocf::heartbeat:IPaddr2):      Started yls2425-master
 * yls2425-VIP2      (ocf::heartbeat:IPaddr2):      Started yls2425-slave
```

Figura 52: Estado del clúster tras reincorporar el nodo master.

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 53: Nodo yls2425-master puesto en modo *standby*.

```
# sudo crm node standby yls2425-spare
```

```
vagrant@yls2425-slave:~> sudo crm node standby yls2425-spare
INFO: standby node yls2425-spare
```

Figura 54: Nodo yls2425-spare puesto en modo *standby*.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 13:01:48 2025 on yls2425-slave
 * Last change: Thu Jan 30 13:09:00 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-master: standby
 * Node yls2425-spare: standby
 * Online: [ yls2425-slave ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-slave
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave

vagrant@yls2425-slave:~> |
```

Figura 55: Estado del clúster con los nodos master y spare en modo *standby*.

```
# sudo crm node online yls2425-spare
```

```
vagrant@yls2425-slave:~> sudo crm node online yls2425-spare
INFO: online node yls2425-spare
```

Figura 56: Reincorporación del nodo yls2425-spare al clúster.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 13:04:18 2025 on yls2425-slave
 * Last change: Thu Jan 30 13:03:15 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-master: standby
 * Online: [ yls2425-slave yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-slave
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
```

Figura 57: Estado del clúster tras la reincorporación del nodo spare.

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-slave:~> sudo crm node online yls2425-master
INFO: online node yls2425-master
```

Figura 58: Reincorporación del nodo yls2425-master al clúster.

```
# sudo crm status
```

```
vagrant@yls2425-slave:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 13:07:12 2025 on yls2425-slave
 * Last change: Thu Jan 30 13:06:18 2025 by root via root on yls2425-slave
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
```

Figura 59: Estado final del clúster tras la reincorporación del nodo master.

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave  
INFO: standby node yls2425-slave
```

Figura 60: Nodo yls2425-slave puesto en modo *standby*.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status  
Cluster Summary:  
 * Stack: corosync (Pacemaker is running)  
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7  
+20231219.0f7f88312) - partition with quorum  
 * Last updated: Thu Jan 30 12:57:33 2025 on yls2425-master  
 * Last change: Thu Jan 30 13:12:54 2025 by root via root on yls2425-master  
 * 3 nodes configured  
 * 2 resource instances configured  
  
Node List:  
 * Node yls2425-slave: standby  
 * Online: [ yls2425-master yls2425-spare ]  
  
Full List of Resources:  
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master  
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-spare
```

Figura 61: Estado del clúster tras poner el nodo yls2425-slave en modo *standby*.

```
# sudo crm resource scores | grep allocation
```

```
vagrant@yls2425-master:~> sudo crm resource scores | grep allocation  
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-master: 300  
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-slave: 0  
pcmk__primitive_assign: yls2425-VIP1 allocation score on yls2425-spare: 50  
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-master: 0  
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-slave: 200  
pcmk__primitive_assign: yls2425-VIP2 allocation score on yls2425-spare: 150  
vagrant@yls2425-master:~> |
```

Figura 62: Puntuaciones de asignación de recursos tras la migración.


```
# sudo crm node online yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-slave  
INFO: online node yls2425-slave
```

Figura 63: Reincorporación del nodo yls2425-slave al clúster.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status  
Cluster Summary:  
  * Stack: corosync (Pacemaker is running)  
  * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7  
+20231219.0f7f88312) - partition with quorum  
  * Last updated: Thu Jan 30 12:58:52 2025 on yls2425-master  
  * Last change: Thu Jan 30 13:14:03 2025 by root via root on yls2425-master  
  * 3 nodes configured  
  * 2 resource instances configured  
  
Node List:  
  * Online: [ yls2425-master yls2425-slave yls2425-spare ]  
  
Full List of Resources:  
  * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master  
  * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
```

Figura 64: Estado final del clúster tras la reincorporación del nodo slave.

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave  
INFO: standby node yls2425-slave
```

Figura 65: Nodo yls2425-slave puesto en modo *standby*.

```
# sudo crm node standby yls2425-spare
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-spare
INFO: standby node yls2425-spare
```

Figura 66: Nodo yls2425-spare puesto en modo *standby*.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 12:59:56 2025 on yls2425-master
 * Last change: Thu Jan 30 13:15:01 2025 by root via root on yls2425-master
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-slave: standby
 * Node yls2425-spare: standby
 * Online: [ yls2425-master ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-master
```

Figura 67: Estado del clúster con los nodos slave y spare en modo *standby*.

```
# sudo crm node online yls2425-spare
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-spare
INFO: online node yls2425-spare
sudvagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
 * Last updated: Thu Jan 30 13:00:26 2025 on yls2425-master
 * Last change: Thu Jan 30 13:15:35 2025 by root via root on yls2425-master
 * 3 nodes configured
 * 2 resource instances configured

Node List:
 * Node yls2425-slave: standby
 * Online: [ yls2425-master yls2425-spare ]

Full List of Resources:
 * yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-master
```

Figura 68: Estado del clúster tras la reincorporación del nodo spare.

```
# sudo crm node online yls2425-slave
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-slave
INFO: online node yls2425-slave
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.6.1-2.1.7
+20231219.0f7f88312) - partition with quorum
* Last updated: Thu Jan 30 13:02:03 2025 on yls2425-master
* Last change: Thu Jan 30 13:17:04 2025 by root via root on yls2425-master
* 3 nodes configured
* 2 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave yls2425-spare ]

Full List of Resources:
* yls2425-VIP1 (ocf::heartbeat:IPaddr2): Started yls2425-master
* yls2425-VIP2 (ocf::heartbeat:IPaddr2): Started yls2425-slave
vagrant@yls2425-master:~> |
```

Figura 69: Estado final del clúster tras la reincorporación del nodo `slave`.

1.7 Justificación

Al ejecutar la secuencia de comandos, el clúster se comportó conforme a los requisitos establecidos: mientras `yls2425-master` y `yls2425-slave` estuvieron activos, los recursos `yls2425-VIP1` y `yls2425-VIP2` permanecieron en sus nodos preferidos respectivamente.

Al poner `yls2425-master` en modo *standby*, `yls2425-VIP1` migró adecuadamente a `spare`, y de forma análoga, al hacer *standby* sobre `yls2425-slave`, la IP `yls2425-VIP2` también se reasignó a `spare`.

Una vez se reactivaron `yls2425-master` y `yls2425-slave`, se observó que los recursos retornaban a sus nodos preferidos. Esto se debe a que se configuró la **adherencia (resource-stickiness)** en **100**, lo que garantiza que cada IP virtual vuelva automáticamente a su nodo principal sin incurrir en migraciones innecesarias, tal y como se había previsto en la configuración del clúster.

Práctica 2: Cluster HA con Almacenamiento Replicado Usando DRBD

2 Introducción

2.1 Objetivo

El objetivo de esta práctica es configurar un clúster HA que tenga almacenamiento replicado. Para replicar el almacenamiento, utilizaremos DRBD (<https://www.linbit.com/en/drbd-oss-distribution>).

El núcleo de la funcionalidad de DRBD está implementado en un controlador del kernel que realiza las lecturas y escrituras en discos locales y, de manera transparente, la replicación en red en tiempo real en una o más ubicaciones de respaldo.

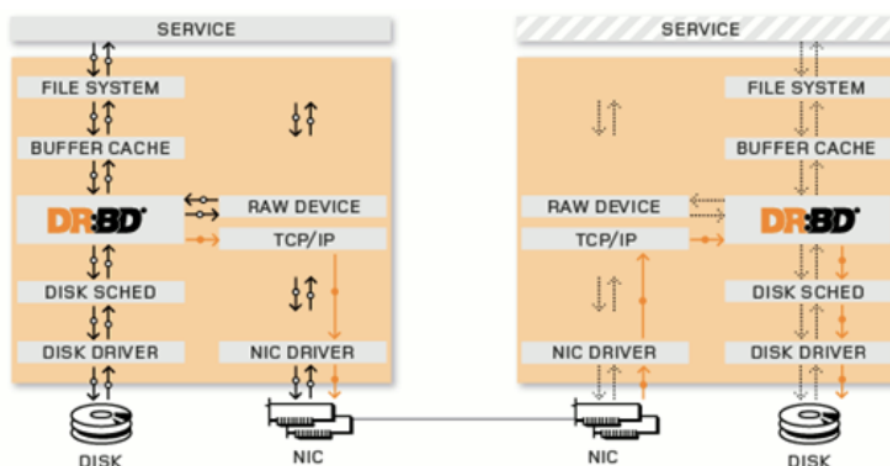


Figura 70: Esquema de funcionamiento de DRBD.

DRBD soporta casi cualquier dispositivo de almacenamiento:

- una partición de disco o un disco completo.
- un dispositivo RAID por software.
- un volumen lógico LVM (Linux).
- cualquier dispositivo de bloques que nuestro sistema reconozca (por ejemplo, un disco USB).

Esto proporciona mucha flexibilidad, ya que se podría crear un RAID 1 utilizando, por ejemplo, una partición en el disco local de un ordenador y un disco USB externo conectado en otro ordenador remoto accesible por red. Y todo en tiempo real, los datos que se escriban en el disco local se replicarán automáticamente en el disco USB remoto mediante la red. Además, se podrían replicar tantos dispositivos como se desee, sincronizándose todos simultáneamente.

Para esta práctica, configuraremos un **RAID 1** utilizando el almacenamiento disponible en 2 nodos del clúster HA. Además, se aprovechará la práctica para ver **opciones más avanzadas de Pacemaker**, como el **emparejamiento de recursos** y las **restricciones de orden a la hora de detener/iniciar servicios**.

Parte 1: Configuración de un servidor web en el clúster HA

En esta primera parte, se procederá a la instalación de un servidor web Apache en los dos nodos del clúster y se configurará como recurso dentro de Pacemaker. El objetivo es garantizar que uno de los nodos siempre pueda servir la página web en caso de fallo del otro nodo, simulando un entorno de alta disponibilidad.

Los pasos principales a seguir son:

- **Instalación de Apache en ambos nodos:** Se instalará el servidor web Apache en los dos nodos para habilitar el servicio web en ambos.
- **Creación de páginas web diferenciadas:** Se crearán páginas web en ambos nodos con contenido distinto para poder identificar de manera visual qué nodo está sirviendo la página.
- **Conmutación de nodos:** Se pondrá uno de los nodos en modo standby para verificar que el otro nodo asume los recursos y comienza a servir la página web.
- **Configuración de Apache como recurso de Pacemaker:** Finalmente, se configurará Apache como un recurso del clúster, asegurando que Pacemaker gestione automáticamente el servicio en caso de caída de un nodo.

2.2 Comandos

En esta sección se ejecutarán los comandos necesarios para verificar el correcto funcionamiento del clúster HA configurado con Apache como recurso gestionado por Pacemaker.

2.2.1. Comprobación del estado de partida

```
# sudo crm configure show
```

```
vagrant@yls2425-master:~> sudo crm configure show
node 1: yls2425-master \
    attributes standby=off
node 2: yls2425-slave \
    attributes standby=off
primitive yls2425-VIP IPAddr2 \
    params ip=192.168.56.10 nic=eth1 \
    meta is-managed=true \
    op monitor interval=10s timeout=20s \
    op start timeout=20s interval=0s \
    op stop timeout=20s interval=0s
primitive yls2425-webserver apache \
    op monitor interval=1min timeout=20s \
    op start timeout=40s interval=0s \
    op stop timeout=60s interval=0s
location prefer-master yls2425-VIP 100: yls2425-master
colocation yls2425-website inf: yls2425-VIP yls2425-webserver
order yls2425-websiteorder Mandatory: yls2425-VIP yls2425-webserver
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version="2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312" \
    cluster-infrastructure=corosync \
    cluster-name=yls2425-clusterHA \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults build-resource-defaults: \
    resource-stickiness=100 \
    migration-threshold=3 \
    priority=1
op_defaults op-options: \
    timeout=600 \
    record-pending=true
```

Figura 71: Configuración del clúster al inicio.

El comando muestra la configuración inicial del clúster, asegurando que todos los recursos están configurados correctamente.


```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status

Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-master (version 2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Fri Oct 11 16:59:40 2024 on yls2425-master
 * Last change: Fri Oct 11 16:57:46 2024 by root via root on yls2425-master
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-webserver (ocf::heartbeat:apache): Started yls2425-master
```

Figura 72: Estado del clúster antes de realizar cambios.

Se observa que el nodo **master** está gestionando la IP virtual y los servicios están funcionando correctamente.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:~> curl 192.168.56.2
Esta é a páxina de yls2425-master
```

Figura 73: Respuesta del nodo yls2425-master.

Este comando confirma que el nodo **master** está sirviendo la página web correctamente.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:~> curl 192.168.56.3
curl: (7) Failed to connect to 192.168.56.3 port 80 after 1 ms: Couldn't connect to server
```

Figura 74: Respuesta del nodo yls2425-slave.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:~> curl 192.168.56.10
Esta é a páxina de yls2425-master
```

Figura 75: Respuesta de la IP virtual antes de realizar cambios.

La IP virtual responde correctamente, confirmando que está asignada al nodo master.

2.2.2. Parada del master y migración de recursos al slave

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 76: Poniendo el nodo yls2425-master en standby.

Este comando pone al nodo **master** en **standby**, lo que fuerza la migración de los recursos al nodo **slave**.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:~> curl 192.168.56.3  
Esta é a páxina de yls2425-slave
```

Figura 77: El nodo yls2425-slave responde correctamente tras la migración de recursos.

Se confirma que el nodo **slave** ha asumido correctamente los recursos tras poner el **master** en standby.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:~> curl 192.168.56.10  
Esta é a páxina de yls2425-slave
```

Figura 78: La IP virtual responde desde yls2425-slave.

La IP virtual sigue siendo accesible, pero ahora está siendo gestionada por el nodo **slave**.

2.2.3. Reinicio del master

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-master  
INFO: online node yls2425-master
```

Figura 79: Reinicio del nodo yls2425-master.

El nodo **master** vuelve a estar online, y el clúster lo incorpora de nuevo al grupo de nodos activos.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-master (version 2.1.7+20231219.0f7f88312-150600.6.3.
1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Fri Oct 11 17:05:01 2024 on yls2425-master
 * Last change: Fri Oct 11 17:04:41 2024 by root via root on yls2425-master
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-slave
 * yls2425-webserver (ocf::heartbeat:apache): Started yls2425-slave
```

Figura 80: Estado del clúster tras el reinicio del master.

El estado del clúster muestra que ambos nodos están ahora online y el slave sigue gestionando la IP virtual.

2.2.4. Parada del slave y reasignación de recursos al master

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node standby yls2425-slave
INFO: standby node yls2425-slave
```

Figura 81: Poniendo el nodo yls2425-slave en standby.

Se pone el slave en standby, lo que fuerza la reasignación de los recursos al nodo master.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:~> curl 192.168.56.2
Esta é a páxina de yls2425-master
```

Figura 82: El nodo yls2425-master responde correctamente tras asumir los recursos.

Ahora el nodo master ha reasumido los recursos y sirve la página web.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:~> curl 192.168.56.10
Esta é a páxina de yls2425-master
```

Figura 83: La IP virtual está nuevamente gestionada por yls2425-master.

La IP virtual vuelve a estar gestionada por el nodo master, como se esperaba.

2.2.5. Reinicio del slave y verificación del estado final

```
# sudo crm node online yls2425-slave
```

```
vagrant@yls2425-master:~> sudo crm node online yls2425-slave
INFO: online node yls2425-slave
```

Figura 84: Reinicio del nodo yls2425-slave.

El nodo slave vuelve a estar online y es reincorporado al clúster.

```
# sudo crm status
```

```
vagrant@yls2425-master:~> sudo crm status
Cluster Summary:
 * Stack: corosync (Pacemaker is running)
 * Current DC: yls2425-master (version 2.1.7+20231219.0f7f88312-150600.6.3.
1-2.1.7+20231219.0f7f88312) - partition with quorum
 * Last updated: Fri Oct 11 17:07:33 2024 on yls2425-master
 * Last change: Fri Oct 11 17:07:08 2024 by root via root on yls2425-master
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
 * yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master
 * yls2425-webserver (ocf::heartbeat:apache): Started yls2425-master
```

Figura 85: Estado final del clúster tras reiniciar ambos nodos.

El estado final muestra que ambos nodos están online y el sistema ha vuelto a la normalidad, con el master gestionando la IP virtual.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:~> curl 192.168.56.2
Esta é a páxina de yls2425-master
```

Figura 86: El nodo yls2425-master sigue sirviendo la página.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:~> curl 192.168.56.3  
curl: (7) Failed to connect to 192.168.56.3 port 80 after 1 ms: Couldn't connect to server
```

Figura 87: El nodo yls2425-slave no responde comprobando la correcta configuración.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:~> curl 192.168.56.10  
Esta é a páxina de yls2425-master
```

Figura 88: La IP virtual sigue respondiendo correctamente.

2.3 Justificación

Durante la ejecución de los comandos, se confirmó el correcto funcionamiento del clúster. Inicialmente, el nodo **yls2425-master** estaba activo y sirviendo la IP virtual, mientras que el **slave** permanecía en standby. Al poner el **master** en standby, el recurso migró al **slave**, quien asumió correctamente la IP virtual. Tras volver a activar el **master**, el clúster restauró su configuración inicial.

De igual manera, cuando se puso el **slave** en standby, el **master** reasumió los recursos y la IP virtual. Finalmente, tras reactivar el **slave**, ambos nodos quedaron en su estado normal, confirmando la alta disponibilidad del clúster y su capacidad de migrar los recursos de manera eficiente.

Parte 2: Configuración de un RAID 1 usando DRBD

En esta segunda parte de la práctica, se procederá a instalar y configurar DRBD en ambos nodos del clúster. DRBD permite replicar datos en tiempo real entre dos nodos mediante una red, proporcionando alta disponibilidad y redundancia de datos.

El objetivo de esta sección es establecer un RAID 1 usando los discos de 1GB en los nodos y preparar los contenidos para ser servidos como parte del clúster HA.

A continuación, se detallan los pasos para la instalación de DRBD, la configuración de un volumen replicado, y la comprobación del correcto funcionamiento del sistema:

- **Instalación de DRBD:** Se instalará el módulo y las herramientas de DRBD en ambos nodos.
- **Configuración del volumen replicado:** Configuración del recurso DRBD para replicar el disco en ambos nodos.
- **Preparación del contenido del disco replicado:** Formateo y montaje del volumen replicado para almacenar y servir contenido web.
- **Comprobación de la configuración:** Verificación del correcto funcionamiento de la replicación mediante la conmutación entre los nodos.

Los pasos detallados a continuación describen cómo llevar a cabo esta configuración en ambos nodos.

2.3.1. Creación de discos virtuales para RAID 1

- **Paso 1:** Detener las VMs usando `vagrant halt`.
- **Paso 2:** Modificar el archivo Vagrantfile para agregar un disco adicional de 1GB en ambos nodos (`master` y `slave`).
- **Paso 3:** Levantar nuevamente las VMs y comprobar que el nuevo disco esté disponible.

2.3.2. Instalación de DRBD

- **Paso 4:** Comprobar si el kernel instalado en las VMs es el más reciente.
- **Paso 5:** Instalar las dependencias necesarias para la compilación de DRBD.
- **Paso 6:** Descargar y compilar DRBD en ambos nodos.

2.4 Comandos

2.4.1. Comprobación del estado actual

```
master:
# sudo drbdadm dump yls2425-drbd1
```

```
vagrant@yls2425-master:/usr/src/linux-obj/x86_64/default> sudo drbdadm dump
yls2425-drbd1
# resource yls2425-drbd1 on yls2425-master: not ignored, not stacked
# defined at /etc/drbd.d/yls2425-drbd1.res:1
resource yls2425-drbd1 {
    device                minor 1;
    disk                  /dev/sdb;
    meta-disk             internal;
    on yls2425-master {
        node-id 0;
    }
    on yls2425-slave {
        node-id 1;
    }
    connection {
        host yls2425-master      address      ipv4 192.168.56.2:7789;
        host yls2425-slave      address      ipv4 192.168.56.3:7789;
        net {
            _name                yls2425-slave;
        }
    }
    net {
        verify-alg              sha1;
    }
}
```

Figura 89: Resultado del comando `drbdadm dump`, mostrando el estado actual del volumen `yls2425-drbd1`.

```
master:
# sudo drbdadm status
```

```
vagrant@yls2425-master:/usr/src/linux-obj/x86_64/default> sudo drbdadm status
yls2425-drbd1 role:Primary
disk:UpToDate
yls2425-slave role:Secondary
peer-disk:UpToDate
```

Figura 90: Estado actual del DRBD en el nodo yls2425-master, mostrando el rol primario del nodo.

2.4.2. Cambio de roles primario/secundario

```
master:
# sudo drbdadm secondary yls2425-drbd1
```

```
vagrant@yls2425-master:/usr/src/linux-obj/x86_64/default> sudo drbdadm secondary yls2425-drbd1
```

Figura 91: El nodo yls2425-master cambia su rol a secundario.

```
slave:
# sudo drbdadm primary yls2425-drbd1
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> sudo drbdadm primary yls2425-drbd1
```

Figura 92: El nodo yls2425-slave asume el rol primario.

```
slave:
# sudo drbdadm status
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> sudo drbdadm status
yls2425-drbd1 role:Primary
  disk:UpToDate
  yls2425-master role:Secondary
  peer-disk:UpToDate
```

Figura 93: El nodo yls2425-slave ahora muestra su rol como primario después del cambio de rol.

2.4.3. Montaje del volumen en el slave

```
slave:
# sudo mount /dev/drbd1 /mnt
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> sudo mount /dev/drbd1 /mnt
```

Figura 94: Montaje exitoso del volumen DRBD en el nodo yls2425-slave.

```
slave:
# sudo cat /mnt/index.html
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> cat /mnt/index.html
Esta é a páxina do volume DRBD
```

Figura 95: Visualización del archivo index.html en el volumen replicado, confirmando la correcta replicación de datos.

2.4.4. Desmontaje del volumen y cambio de roles

```
slave:  
# sudo umount /mnt
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> sudo umount /mnt
```

Figura 96: Desmontaje exitoso del volumen en el nodo yls2425-slave.

```
slave:  
# sudo drbdadm secondary yls2425-drbd1
```

```
vagrant@yls2425-slave:/usr/src/linux-obj/x86_64/default> sudo drbdadm second  
ary yls2425-drbd1
```

Figura 97: El nodo yls2425-slave cambia su rol a secundario.

```
master:  
# sudo drbdadm primary yls2425-drbd1
```

```
vagrant@yls2425-master:/usr/src/linux-obj/x86_64/default> sudo drbdadm prima  
ry yls2425-drbd1
```

Figura 98: El nodo yls2425-master retoma el rol primario.

```
master:  
# sudo drbdadm status
```

```
vagrant@yls2425-master:/usr/src/linux-obj/x86_64/default> sudo drbdadm statu  
s  
yls2425-drbd1 role:Primary  
  disk:UpToDate  
  yls2425-slave role:Secondary  
  peer-disk:UpToDate
```

Figura 99: Estado final del DRBD, con el nodo yls2425-master nuevamente como primario.

2.5 Justificación

Durante la ejecución de la secuencia de comandos, se observó que al cambiar los roles de **primario** a **secundario** y viceversa, el clúster DRBD migró correctamente el control del volumen entre los nodos.

Al montar el volumen en el **slave**, se accedió exitosamente a los datos replicados, y tras desmontarlo, se restauraron los roles originales.

Esto confirma que el sistema está replicando y sincronizando los datos de manera correcta, y que el cambio de roles y acceso a los datos funciona de acuerdo con lo esperado en un entorno de alta disponibilidad.

Parte 3: Configuración de DRBD como recurso del clúster HA

En esta tercera parte, configuraremos DRBD como un recurso del clúster HA utilizando Pacemaker. DRBD será responsable de replicar los datos en tiempo real entre los dos nodos, asegurando así la alta disponibilidad del almacenamiento. El clúster Pacemaker gestionará tanto el servicio web como el recurso DRBD.

Los pasos principales incluyen:

- Desactivar y deshabilitar el servicio DRBD en ambos nodos para permitir que Pacemaker tome control exclusivo de DRBD.
- Configurar DRBD como un recurso del clúster a través de Pacemaker.
- Configurar un clon de DRBD para que pueda ejecutarse en ambos nodos del clúster, permitiendo roles de primario y secundario.
- Configurar el sistema de archivos asociado al recurso DRBD y emparejarlo con el servidor web Apache.

Además, verificaremos el estado del clúster y realizaremos pruebas de funcionamiento para asegurarnos de que el volumen DRBD replica correctamente los datos y los recursos son gestionados de manera adecuada entre los nodos.

2.6 Comandos

2.6.1. Comprobación del estado de partida

```
# sudo crm configure show
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm configure show
node 1: yls2425-master \
  attributes standby=off
node 2: yls2425-slave \
  attributes standby=on
primitive yls2425-VIP IPAddr2 \
  params ip=192.168.56.10 nic=eth1 \
  meta is-managed=true \
  op monitor interval=10s timeout=20s \
  op start timeout=20s interval=0s \
  op stop timeout=20s interval=0s
primitive yls2425-webdata ocf:linbit:drbd \
  params drbd_resource=yls2425-drbd1 \
  op monitor timeout=20s interval=20s role=Slave \
  op monitor timeout=20s interval=10s role=Master \
  op start timeout=240s interval=0s \
  op promote timeout=90s interval=0s \
  op demote timeout=90s interval=0s \
  op stop timeout=100s interval=0s
primitive yls2425-webfs Filesystem \
  params device="/dev/drbd1" directory="/srv/www/htdocs" fstype=ext4 \
  op monitor timeout=40s interval=20s \
  op start timeout=60s interval=0s \
  op stop timeout=60s interval=0s
primitive yls2425-webserver apache \
  op monitor interval=1min timeout=20s \
  op start timeout=40s interval=0s \
  op stop timeout=60s interval=0s
clone yls2425-webdataclone yls2425-webdata \
  meta clone-max=2 clone-node-max=1 notify=true promotable=true interleave=true
location prefer-master yls2425-VIP 100: yls2425-master
order yls2425-fs-after-drbd Mandatory: yls2425-webdataclone:promote yls2425-webfs:start
colocation yls2425-fs-on-drbd inf: yls2425-webfs yls2425-webdataclone:Master
order yls2425-web-after-fs Mandatory: yls2425-webfs yls2425-webserver
colocation yls2425-web-with-fs inf: yls2425-webserver yls2425-webfs
colocation yls2425-website inf: yls2425-VIP yls2425-webserver
order yls2425-websiteorder Mandatory: yls2425-VIP yls2425-webserver
property cib-bootstrap-options: \
  have-watchdog=false \
  dc-version="2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312" \
  cluster-infrastructure=corosync \
  cluster-name=yls2425-clusterHA \
  stonith-enabled=false \
  no-quorum-policy=ignore \
  last-lrm-refresh=1728757043
rsc_defaults build-resource-defaults: \
  resource-stickiness=100 \
  migration-threshold=3 \
  priority=1
op_defaults op-options: \
  timeout=600 \
  record-pending=true
```

Figura 100: Visualización de la configuración inicial del clúster.

```
# sudo crm status
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312) - partition with quorum
* Last updated: Sat Oct 12 19:11:55 2024 on yls2425-master
* Last change: Sat Oct 12 19:11:19 2024 by root via root on yls2425-master
* 2 nodes configured
* 5 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
* yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master
* yls2425-webserver (ocf::heartbeat:apache): Started yls2425-master
* Clone Set: yls2425-webdataclone [yls2425-webdata] (promotable):
  * Masters: [ yls2425-master ]
  * Slaves: [ yls2425-slave ]
* yls2425-webfs (ocf::heartbeat:Filesystem): Started yls2425-master
```

Figura 101: Estado inicial del clúster antes de realizar cualquier modificación.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.2
Esta é a páxina do volume DRBD
```

Figura 102: Comprobación de la página servida por el nodo yls2425-master.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.3
curl: (7) Failed to connect to 192.168.56.3 port 80 after 2 ms: Couldn't connect to server
```

Figura 103: Comprobación de la página servida por el nodo yls2425-slave.


```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.10
Esta é a páxina do volume DRBD
```

Figura 104: Comprobación de la página servida por la IP virtual.

2.6.2. Parada y reinicio del master

```
# sudo crm node standby yls2425-master
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm node standby yls2425-master
INFO: standby node yls2425-master
```

Figura 105: Parada del nodo yls2425-master.

```
# sudo crm node online yls2425-master
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm node online yls2425-master
INFO: online node yls2425-master
```

Figura 106: Reinicio del nodo yls2425-master.

2.6.3. Comprobación del estado tras el reinicio del master

```
# sudo crm status
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312) - partition with quorum
* Last updated: Sat Oct 12 19:14:16 2024 on yls2425-master
* Last change: Sat Oct 12 19:13:58 2024 by root via root on yls2425-master
* 2 nodes configured
* 5 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
* yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-slave
* yls2425-webserver (ocf::heartbeat:apache): Started yls2425-slave
* Clone Set: yls2425-webdataclone [yls2425-webdata] (promotable):
* Masters: [ yls2425-slave ]
* Slaves: [ yls2425-master ]
* yls2425-webfs (ocf::heartbeat:Filesystem): Started yls2425-slave
```

Figura 107: Estado del clúster tras el reinicio del nodo yls2425-master.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.2
curl: (7) Failed to connect to 192.168.56.2 port 80 after 0 ms: Couldn't connect to server
```

Figura 108: Comprobación de la página del nodo yls2425-master tras el reinicio.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.3
Esta é a páxina do volume DRBD
```

Figura 109: Comprobación de la página del nodo yls2425-slave tras el reinicio del master.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.10  
Esta é a páxina do volume DRBD
```

Figura 110: Comprobación de la página servida por la IP virtual tras el reinicio del master.

2.6.4. Parada y reinicio del slave

```
# sudo crm node standby yls2425-slave
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm node standby yls2425-slave  
INFO: standby node yls2425-slave
```

Figura 111: Parada del nodo yls2425-slave.

```
# sudo crm node online yls2425-slave
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm node online yls2425-slave  
INFO: online node yls2425-slave
```

Figura 112: Reinicio del nodo yls2425-slave.

2.6.5. Comprobación del estado tras el reinicio del slave

```
# sudo crm status
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> sudo crm status
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: yls2425-slave (version 2.1.7+20231219.0f7f88312-150600.6.3.1-2.1.7+20231219.0f7f88312) - partition with quorum
* Last updated: Sat Oct 12 19:16:59 2024 on yls2425-master
* Last change: Sat Oct 12 19:16:38 2024 by root via root on yls2425-master
* 2 nodes configured
* 5 resource instances configured

Node List:
* Online: [ yls2425-master yls2425-slave ]

Full List of Resources:
* yls2425-VIP (ocf::heartbeat:IPaddr2): Started yls2425-master
* yls2425-webserver (ocf::heartbeat:apache): Started yls2425-master
* Clone Set: yls2425-webdataclone [yls2425-webdata] (promotable):
* Masters: [ yls2425-master ]
* Slaves: [ yls2425-slave ]
* yls2425-webfs (ocf::heartbeat:Filesystem): Started yls2425-master
```

Figura 113: Estado del clúster tras el reinicio del nodo yls2425-slave.

```
# curl 192.168.56.2
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.2
Esta é a páxina do volume DRBD
```

Figura 114: Comprobación de la página del nodo yls2425-master tras el reinicio del slave.

```
# curl 192.168.56.3
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.3
curl: (7) Failed to connect to 192.168.56.3 port 80 after 1 ms: Couldn't connect to server
```

Figura 115: Comprobación de la página del nodo yls2425-slave tras el reinicio.

```
# curl 192.168.56.10
```

```
vagrant@yls2425-master:/usr/src/drbd-9.2.10> curl 192.168.56.10  
Esta é a páxina do volume DRBD
```

Figura 116: Comprobación de la página servida por la IP virtual tras el reinicio del slave.

2.7 Justificación

Durante la ejecución de los comandos, se observó que el **master** sirvió inicialmente la página desde la IP **virtual**, como se esperaba. Al poner el **master** en *standby*, el **slave** asumió los recursos correctamente y comenzó a servir la página desde la IP **virtual**.

Tras volver a activar el **master**, los recursos permanecieron en el **slave** hasta que se puso en *standby* al **slave**, momento en el cual el **master** retomó los servicios.

Esto demuestra que el clúster gestiona correctamente la migración de recursos y garantiza la alta disponibilidad de los servicios, manteniendo la replicación del almacenamiento y la gestión de los recursos de manera eficiente entre los nodos del clúster.

Práctica 3: Balanceo de carga y medida del rendimiento

3 Introducción

En esta práctica, el objetivo principal es realizar dos actividades:

1. Configuración de un balanceador de carga entre varias máquinas virtuales (VMs) para simular un clúster de balanceo de carga.
2. Análisis del rendimiento del clúster mediante el uso de un benchmark, centrándonos en las métricas de rendimiento más relevantes.

En la primera parte de la práctica, configuramos el balanceador de carga utilizando **Nginx** como software para distribuir las solicitudes entre dos VMs que actúan como servidores web utilizando *Apache*. La configuración de cada máquina en el clúster se muestra en la siguiente figura:

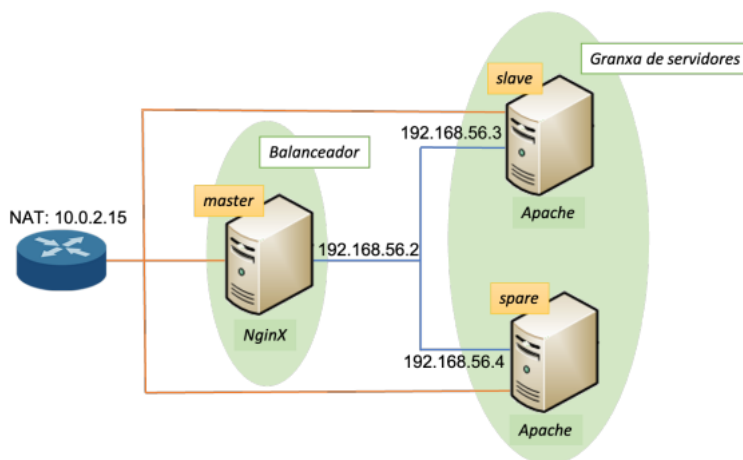


Figura 117: Configuración del balanceador de carga y la granja de servidores

En la segunda parte de la práctica, realizaremos un análisis del rendimiento de este clúster, empleando herramientas de benchmark que nos permitirán medir la capacidad del balanceador de carga y la eficiencia de los servidores web al gestionar solicitudes concurrentes.

Parte 1: Configuración de un balanceador de carga

En esta primera parte de la práctica, configuramos un balanceador de carga utilizando Nginx para repartir solicitudes entre dos máquinas virtuales (VMs) que funcionan como servidores web. El balanceador de carga distribuye el tráfico entre las VMs con Apache.

3.1 Comandos y Justificación

```
yls2425-master$:  
# cat /etc/nginx/conf.d/*.conf
```

```
vagrant@yls2425-master:~> cat /etc/nginx/conf.d/*.conf  
upstream yls2425-webfarm {  
    server yls2425-slave weight=3;  
    server yls2425-spare weight=2;  
}  
  
upstream yls2425-slave {  
    server yls2425-slave;  
}  
  
upstream yls2425-spare {  
    server yls2425-spare;  
}  
  
server {  
    listen 80;  
    server_name yls2425-lb;  
    root /srv/www/htdocs/;  
    error_log /var/log/nginx/yls2425-lb.error.log;  
  
    # Balanceo de carga 3:2 entre slave y spare  
    location / {  
        proxy_pass http://yls2425-webfarm;  
    }  
  
    # Redirigir todas las solicitudes a yls2425-slave  
    location /slave {  
        proxy_pass http://yls2425-slave/;  
    }  
  
    # Redirigir todas las solicitudes a yls2425-spare  
    location /spare {  
        proxy_pass http://yls2425-spare/;  
    }  
}  
  
upstream yls2425-rr {  
    server yls2425-slave;  
    server yls2425-spare;  
}  
  
server {  
    listen 80;  
    server_name yls2425-rr;  
    root /srv/www/htdocs/;  
    error_log /var/log/nginx/yls2425-rr.error.log;  
  
    location / {  
        proxy_pass http://yls2425-rr;  
    }  
}
```

Figura 118: Contenido del archivo de configuración de Nginx

En esta fase, se verifica el contenido del archivo de configuración de Nginx, asegurando que los parámetros y la estructura de los bloques **upstream** y **server** sean correctos para el balanceo de carga.

```
yls2425-master$:  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb
```

```
vagrant@yls2425-master:~> curl http://yls2425-lb  
Este es el servidor SLAVE  
vagrant@yls2425-master:~> curl http://yls2425-lb  
Este es el servidor SPARE  
vagrant@yls2425-master:~> curl http://yls2425-lb  
Este es el servidor SLAVE  
vagrant@yls2425-master:~> curl http://yls2425-lb  
Este es el servidor SPARE  
vagrant@yls2425-master:~> curl http://yls2425-lb  
Este es el servidor SLAVE
```

Figura 119: Resultados de los primeros 5 accesos al balanceador de carga

Aquí se observa cómo el balanceador distribuye las primeras cinco solicitudes de manera adecuada entre los servidores de la granja, alternando entre el servidor principal (slave) y el secundario (spare) según la configuración de **weight**.

```
yls2425-master$:  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb  
# curl http://yls2425-lb
```

```
vagrant@yls2425-master:~> curl http://yls2425-lb
Este es el servidor SLAVE
vagrant@yls2425-master:~> curl http://yls2425-lb
Este es el servidor SPARE
vagrant@yls2425-master:~> curl http://yls2425-lb
Este es el servidor SLAVE
vagrant@yls2425-master:~> curl http://yls2425-lb
Este es el servidor SPARE
vagrant@yls2425-master:~> curl http://yls2425-lb
Este es el servidor SLAVE
```

Figura 120: Resultados de los segundos 5 accesos al balanceador de carga

En los accesos adicionales, se observa un comportamiento consistente, donde el balanceador sigue distribuyendo las solicitudes de acuerdo con la ponderación asignada a cada servidor.

```
yls2425-master$:
# curl http://yls2425-lb/slave
```

```
vagrant@yls2425-master:~> curl http://yls2425-lb/slave
Este es el servidor SLAVE
```

Figura 121: Resultados del acceso al servidor slave

Aquí se confirma que las solicitudes dirigidas específicamente al servidor **slave** son procesadas correctamente, accediendo de forma directa al servidor correspondiente.

```
yls2425-master$:
# curl http://yls2425-lb/spare
```

```
vagrant@yls2425-master:~> curl http://yls2425-lb/spare
Este es el servidor SPARE
```

Figura 122: Resultados del acceso al servidor spare

De manera similar, se verifica que el acceso directo al servidor **spare** funciona

correctamente, mostrando el contenido proporcionado por este servidor en particular.

```
yls2425-master$:  
# curl http://yls2425-rr  
# curl http://yls2425-rr  
# curl http://yls2425-rr  
# curl http://yls2425-rr  
# curl http://yls2425-rr  
# curl http://yls2425-rr
```

```
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SLAVE  
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SPARE  
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SLAVE  
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SPARE  
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SLAVE  
vagrant@yls2425-master:~> curl http://yls2425-rr  
Este es el servidor SPARE
```

Figura 123: Resultados del acceso al balanceador con algoritmo round-robin

Finalmente, al utilizar el algoritmo **round-robin (rr)**, las solicitudes se distribuyen de forma alterna entre los servidores **slave** y **spare**, sin aplicar pesos adicionales, lo que permite un reparto equitativo de las solicitudes.

Parte 2: Medida del rendimiento de la granja de servidores

En esta segunda parte de la práctica, se analizará el rendimiento de la granja de servidores configurada en la primera parte utilizando la herramienta **Apache Benchmark (ab)**. El objetivo es medir las métricas de rendimiento en diferentes condiciones de concurrencia.

3.2 Justificación

```
yls2425-master$:
# sudo zypper install apache2-utils

# Realizamos 10,000 solicitudes con 100 solicitudes
simultáneas
yls2425-master$:
# ab -k -c 100 -n 10000 http://yls2425-lb/
```

Las siguientes tablas muestran los resultados del benchmark para cada conjunto de pruebas. Se repitieron 10 veces por cada nivel de concurrencia ($c=1$, $c=10$, $c=100$) y se calcularon las medias y desviaciones estándar para cada métrica.

Execución	Tempo total	Solicitudes/s	Velocidade de transferencia	Tempo de conexión total			Porcentaxe de solicitudes					
				min	med	max	50 %	75 %	90 %	95 %	98 %	100 %
1	52.366	190.96	49.79	3	5	22	5	5	6	6	7	22
2	53.179	188.05	49.03	4	5	27	5	5	6	6	7	27
3	52.302	191.20	49.85	3	5	10	5	5	6	6	7	10
4	53.712	186.18	48.54	4	5	117	5	6	6	6	7	117
5	54.692	182.84	47.67	3	5	1034	5	6	6	6	7	1034
6	68.200	146.63	38.23	3	7	1035	5	6	7	8	9	1035
7	61.323	163.07	42.52	4	6	1040	5	6	6	6	7	1040
8	60.971	164.01	42.76	3	6	1040	5	5	6	6	6	1040
9	58.006	172.39	44.95	3	6	1038	5	5	6	6	6	1038
10	63.282	158.02	41.20	4	6	1042	5	5	6	6	6	1042
Media	57.803	170.63	45.80	3.4	5.6	642.5	5	5.5	6	6	6.8	640.7
Desviación	5.443	14.89	3.71	0.52	0.73	467.7	0	0.5	0.4	0.5	1.2	475.1

Tabla 1: Resultados para $n = 10000$, $c = 1$

Execución	Tempo total	Solicitudes/s	Velocidade de transferencia	Tempo de conexión total			Porcentaxe de solicitudes						
				min	med	max	50 %	75 %	90 %	95 %	98 %	100 %	
1	34.041	293.76	76.60	14	34	504	33	35	37	38	39	504	
2	35.454	282.05	73.54	14	35	562	35	37	38	39	41	562	
3	39.166	255.32	66.57	11	39	94	39	43	48	51	56	94	
4	45.479	219.88	57.33	10	45	137	42	53	59	64	71	137	
5	56.230	177.84	46.37	15	56	148	54	64	82	98	111	148	
6	51.491	194.21	50.64	17	51	125	52	57	61	65	70	125	
7	55.557	180.00	46.93	18	55	204	54	62	70	78	95	204	
8	41.396	241.57	62.99	10	41	106	40	44	48	50	54	106	
9	57.080	175.19	45.68	10	57	160	52	66	84	96	109	160	
10	43.279	231.06	60.25	18	43	244	41	45	55	61	69	244	
Media	45.217	215.89	58.99	13.1	44.0	234.4	43.7	51.8	58.2	64.4	71.5	234.4	
Desviación	8.003	38.77	10.60	3.1	9.1	164.6	7.2	10.9	15.5	19.3	23.3	156.1	

Tabla 2: Resultados para $n = 10000$, $c = 10$

Execución	Tempo total	Solicitudes/s	Velocidade de transferencia	Tempo de conexión total			Porcentaxe de solicitudes					
				min	med	max	50 %	75 %	90 %	95 %	98 %	100 %
1	47.128	212.19	55.33	128	469	977	484	561	643	664	700	977
2	41.573	240.54	62.72	86	414	825	434	510	585	649	686	825
3	46.193	216.48	56.45	79	460	841	460	516	578	615	692	841
4	40.239	248.51	64.80	76	401	722	408	456	512	536	596	722
5	40.651	246.00	64.14	71	405	809	409	448	597	651	718	809
6	45.344	220.54	57.50	91	451	832	446	554	636	683	724	832
7	44.066	226.93	59.17	60	439	858	436	471	521	564	596	858
8	41.038	243.68	63.54	75	409	864	417	466	519	552	613	864
9	40.239	248.51	64.80	96	401	678	413	453	506	538	573	678
10	44.766	223.38	58.25	76	446	689	434	468	541	582	643	689
Media	43.524	232.38	60.29	83.8	436.5	809.5	434.1	491.3	574.2	603.4	666.1	809.5
Desviación	2.914	14.38	3.73	20.2	30.2	99.4	23.6	43.3	44.7	52.4	48.0	94.3

Tabla 3: Resultados para $n = 10000$, $c = 100$

3.3 Gráficas

Las gráficas a continuación ilustran los valores medios de cada métrica obtenidos a partir de las pruebas realizadas. Se puede observar cómo la concurrencia afecta el rendimiento del clúster, destacando la relación entre el número de solicitudes por segundo, el tempo total de respuesta y la velocidade de transferencia.

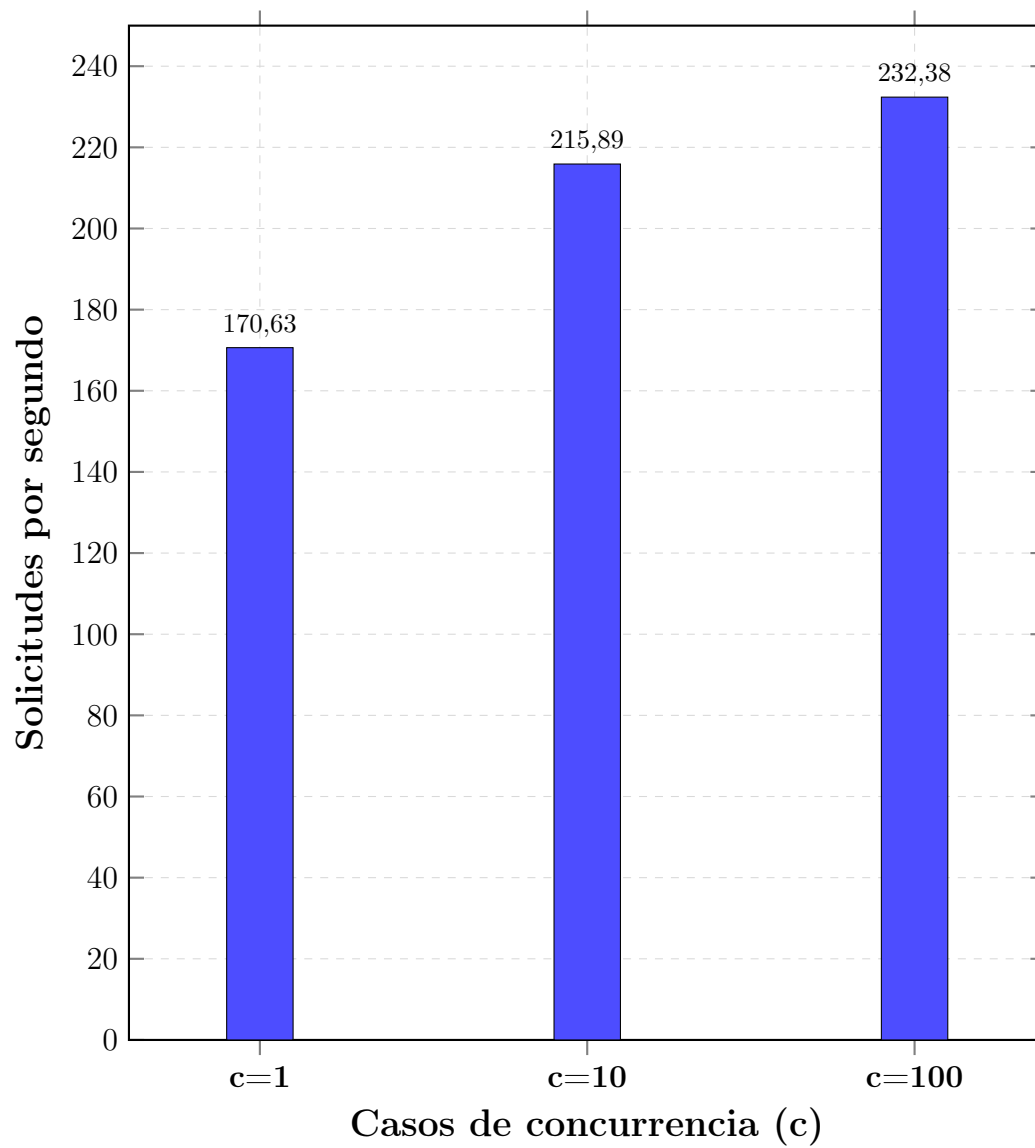


Figura 124: Solicitudes por segundo (valores medios)

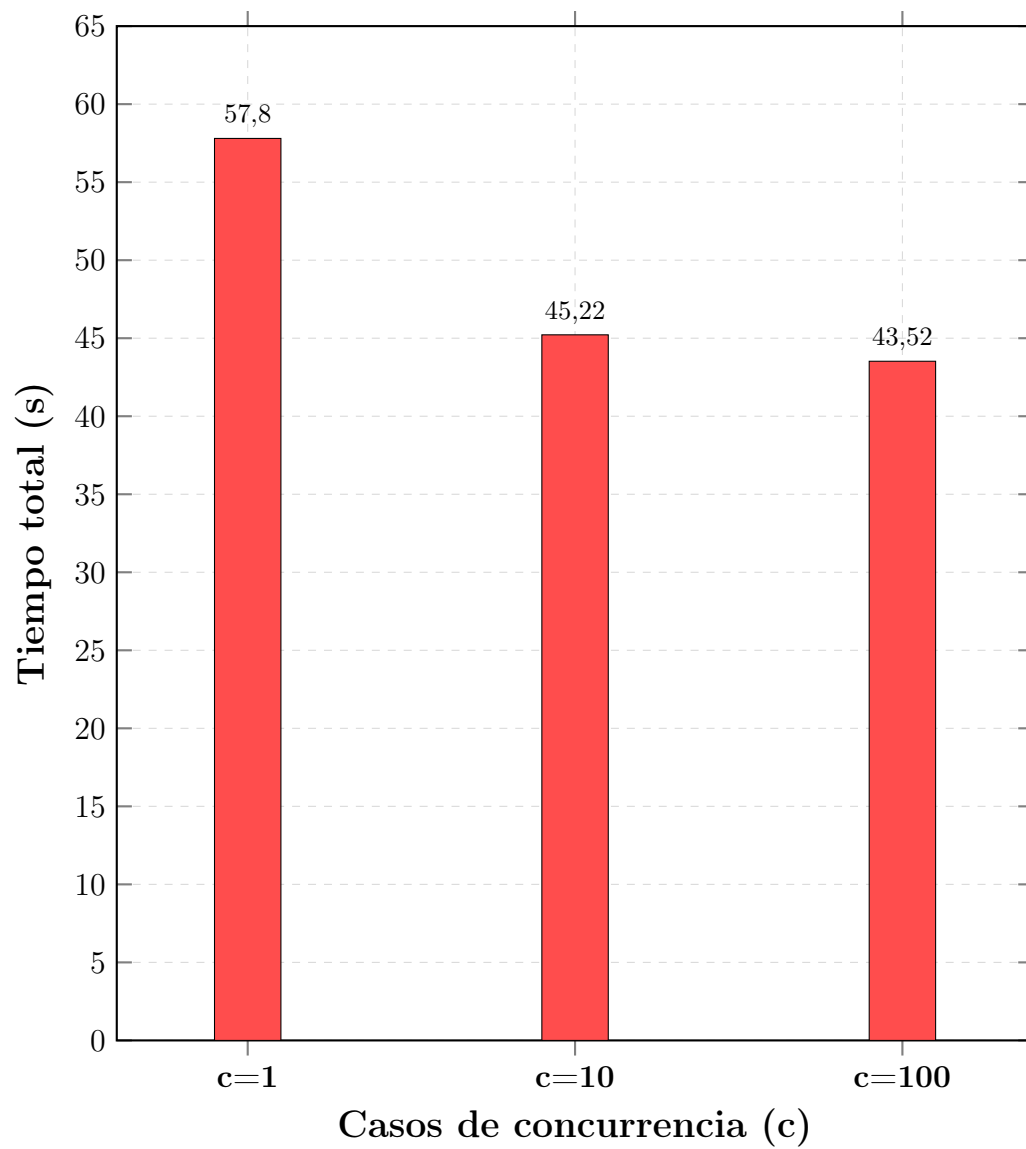


Figura 125: Tiempo total (valores medios)

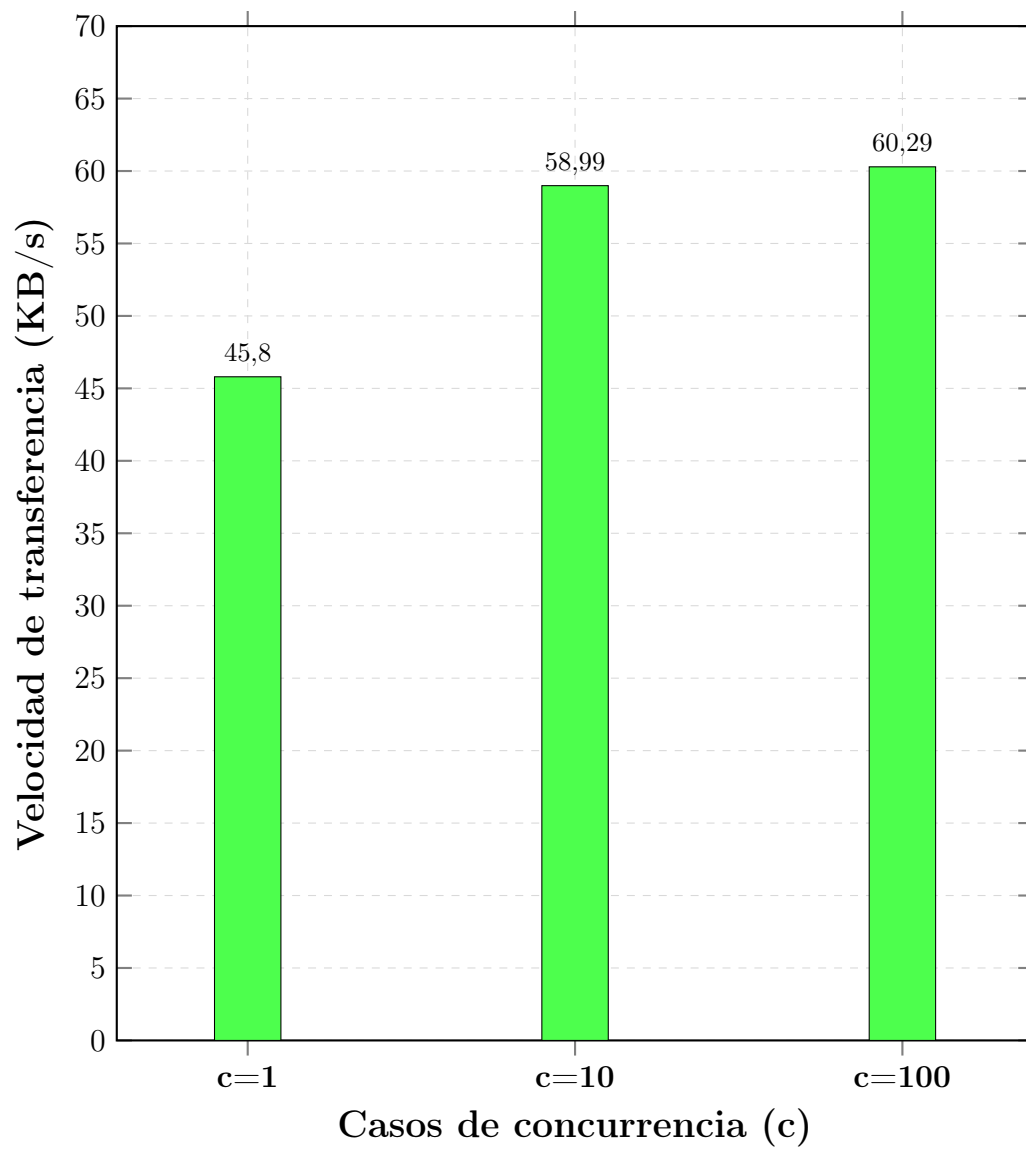


Figura 126: Velocidad de transferencia (valores medios)

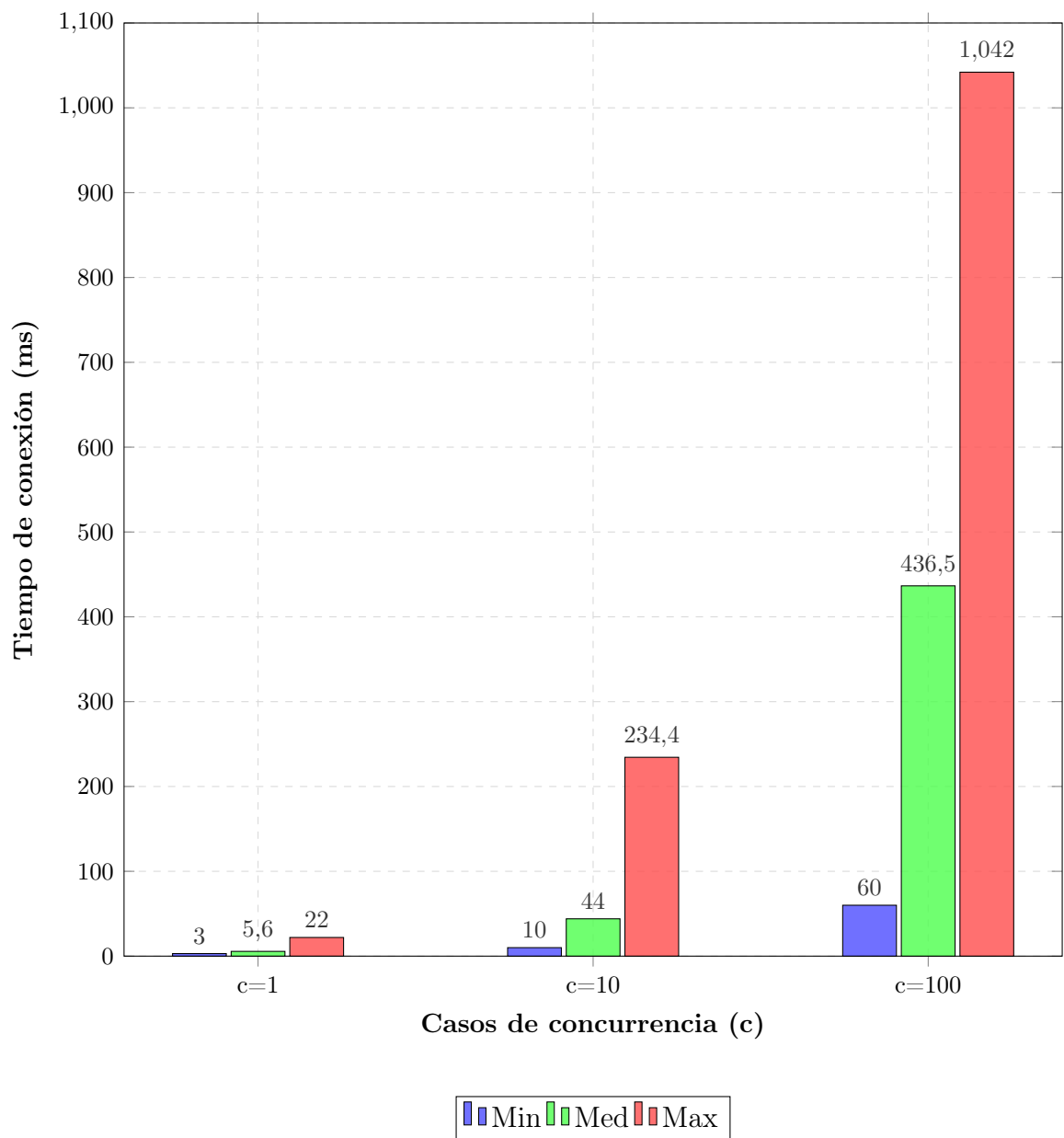


Figura 127: Tiempo de conexión total (valores mínimo, medio, máximo)

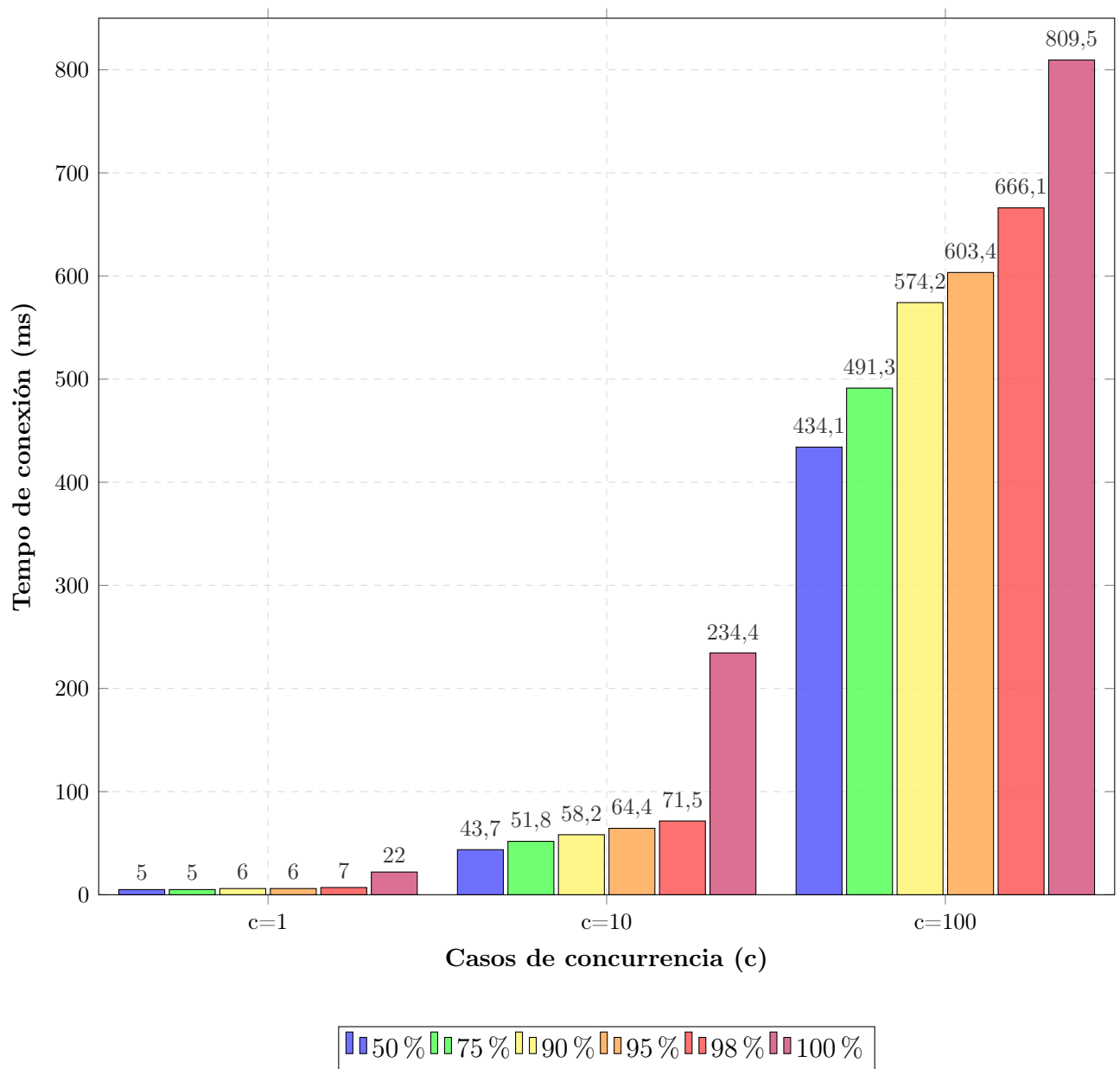


Figura 128: Porcentaxe de solicitudes (Percentiles 50, 75, 90, 95, 98, 100)

3.4 Explicación de los resultados

3.4.1. Solicitudes por segundo (valores medios)

La gráfica de **Solicitudes por segundo** (ver Gráfica 1) muestra el rendimiento del sistema en términos de las solicitudes procesadas por segundo a diferentes niveles de concurrencia. Los resultados observados son los siguientes:

- **c=1**: Procesa en promedio 170.63 solicitudes por segundo.
- **c=10**: Aumenta significativamente a 215.89 solicitudes por segundo.
- **c=100**: Alcanza su máximo rendimiento con 232.38 solicitudes por segundo.

Conclusión: A medida que aumentamos el número de solicitudes concurrentes, el servidor maneja más solicitudes por segundo. Sin embargo, a partir de $c=100$, el aumento es marginal, lo que indica que estamos alcanzando el límite de capacidad del sistema.

3.4.2. Tiempo total (valores medios)

La gráfica de **Tiempo total** (ver Gráfica 2) refleja el tiempo total que toma completar todas las solicitudes. Los resultados son los siguientes:

- **c=1**: El tiempo total es el más alto, con 57.803 segundos, debido a la baja concurrencia.
- **c=10**: El tiempo total disminuye significativamente a 45.217 segundos, lo que indica que el sistema se desempeña mejor con mayor concurrencia.
- **c=100**: Se alcanza un tiempo total de 43.524 segundos, una ligera mejora respecto a $c=10$.

Conclusión: A medida que aumenta la concurrencia, el tiempo total disminuye, lo que indica que el sistema se beneficia de la capacidad de manejar múltiples solicitudes de forma paralela.

3.4.3. Velocidad de transferencia (valores medios)

La gráfica de **Velocidad de transferencia** (ver Gráfica 3) indica cómo varía la cantidad de datos transferidos por segundo:

- **c=1**: La velocidad es de 45.80 KB/s.
- **c=10**: Aumenta a 58.99 KB/s.
- **c=100**: Se alcanza una velocidad máxima de 60.29 KB/s.

Conclusión: La velocidad de transferencia también mejora con mayor concurrencia, aunque de manera menos pronunciada que las solicitudes por segundo, lo que sugiere un límite en la capacidad de red o procesamiento.

3.4.4. Tiempo de conexión total (valores medios)

La gráfica de **Tiempo de conexión total medio** (ver Gráfica 4) mide el tiempo que toma procesar una solicitud:

- **c=1**: El tiempo medio es de 5.6 ms.
- **c=10**: El tiempo de conexión medio aumenta a 44.0 ms.
- **c=100**: El tiempo de conexión sube significativamente a 436.5 ms.

Conclusión: El tiempo de conexión aumenta considerablemente con mayor concurrencia, lo que sugiere que el sistema tiene dificultades para manejar solicitudes concurrentes sin aumentar el tiempo de espera para cada una.

3.4.5. Porcentaje de solicitudes (Percentil 50)

Finalmente, la gráfica de **Porcentaje de solicitudes** en el percentil 50 (ver Gráfica 5) muestra la distribución del tiempo de conexión:

- **c=1**: El 50 % de las solicitudes se completan en 5 ms.
- **c=10**: El tiempo aumenta a 43.7 ms.
- **c=100**: Aumenta de forma significativa a 434.1 ms.

Conclusión: A medida que la concurrencia aumenta, el tiempo que toma completar el 50 % de las solicitudes también crece significativamente, lo que demuestra el impacto negativo de la concurrencia en los tiempos de respuesta.