

Práctica: Redundancia en enrutadores

Objetivos

Configurar dos routers o balanceadores en alta disponibilidad con software libre

Preparación

Es necesario el siguiente material, como es habitual se recomienda el uso de GNS3:

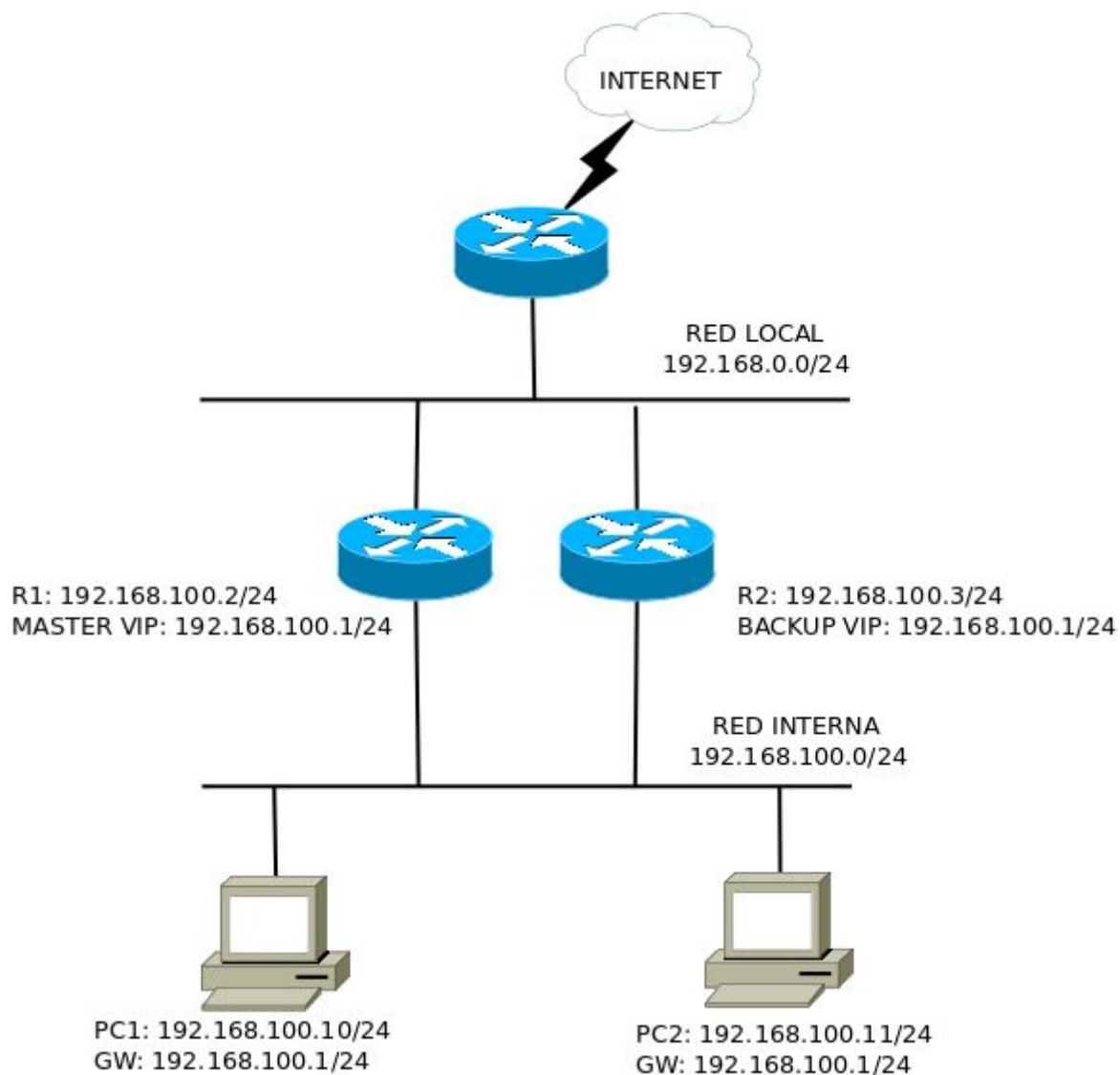
- Dos servidores reales o virtuales con GNU/Linux que harán las funciones de routers
- Un cliente en la red local
- Documentación de [keepalived](#)

Enunciado

Keepalived es un paquete software que implementa el protocolo VRRP de alta disponibilidad que habilita el enrutamiento tolerante a fallos para un par o más de routers que funcionan como balanceadores (mediante el software LVS: Linux Virtual Server) y el demonio keepalived que realiza el chequeo de los nodos de la granja de servidores para realizar el balanceo entre ellos según el algoritmo elegido. En esta práctica sólo usaremos el demonio VRRP que nos permite utilizarlo para montar un servicio de enrutamiento en alta disponibilidad en modo activo/pasivo.

Preparación del escenario

El primer paso es preparar el escenario que se va a usar para configurar un servicio de enrutamiento en alta disponibilidad. El escenario se muestra en la siguiente figura:



En el escenario se distinguen dos redes: la **red interna** y la **red local** con conexión a internet. La red local de la figura será la red local de nuestro domicilio o centro y la red interna es un nuevo direccionamiento que utilizaremos en la práctica como red intermedia. La razón de este escenario es para evitar interferir en la infraestructura de red que tengamos disponible para realizar la práctica.

Los dos routers del escenario tienen una IP en ambas redes. En la red interna además tendrán una dirección IP virtual, que únicamente tendrá el router maestro (R1) y que será la que usen los clientes de la red interna como puerta de enlace. De esta forma si el router maestro cae, el otro router (R2) asumirá las funciones de enrutamiento.

Los routers tienen que ser por tanto máquinas GNU/Linux con la distribución que desee el alumno, pero es necesario habilitar el **ip forwarding** y el **NAT**, para que los clientes de la red interna puedan llegar a Internet. Puedes consultar el punto 2.3 de la UD4 que describe el escenario de prácticas de VPN para ver cómo hacerlo. También es necesario que deshabilites cualquier regla de firewall que pueda interferir con la práctica.

Una vez configurado el escenario, deberías comprobar que el cliente o clientes pueden navegar por Internet. Para realizar la práctica es suficiente con disponer únicamente de un cliente en la red, por ejemplo PC1, ya sea real o virtualizado.

Se deja libertad al alumno para que, en función de los recursos de que disponga, monte el escenario con las máquinas reales o virtuales que quiera. Algunas posibles opciones son:

1. Todos los ordenadores del escenario son máquinas reales. Entendemos que ésta no va a ser la situación habitual, además de requerir dos tarjetas en los equipos que hacen de routers y que deben tener instalado GNU/Linux. También es posible hacerlo con una sola tarjeta en cada router, pero en este caso, ambas redes deben superponerse en el mismo segmento LAN, teniendo cada router doble configuración IP en la tarjeta de red.
2. Teniendo un anfitrión con suficiente RAM, simular los dos routers con máquinas virtuales usando una distribución con pocos recursos, como CentOS minimal. Cada router tendrá dos tarjetas: la local estará en modo puente con la red del anfitrión (conectándose a Internet de esta manera) y la red interna que será una red sólo anfitrión o host-only. El cliente puede ser cualquier otra máquina virtual en modo sólo anfitrión o incluso el mismo anfitrión utilizando la red sólo-anfitrión con puerta de enlace la VIP de los routers.
3. Cualquier otra que se le ocurra al alumno y sirva para simular el escenario.

Para comenzar con la práctica, realiza los siguientes pasos:

1. Instala el paquete **keepalived** utilizando el sistema de gestión de paquetes de tu distribución. En algunas distribuciones como CentOS o Fedora, no se encuentra en los repositorios oficiales, sino en epel.
2. Como root, crea en tu router el fichero **/etc/keepalived/keepalived.conf** con el siguiente contenido en R1:

```
global_defs {
    router_id R1
}
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 1
    priority 100
    virtual_ipaddress {
        192.168.100.1/24 brd 192.168.100.255 dev eth0
    }
    authentication {
        auth_type PASS
        auth_pass SeguridadAD
    }
}
```

```

        track_interface {
            eth1 weight -10
            eth1 weight +10
        }
    }
}

```

Estamos suponiendo que el nombre de la interfaz interna del router en el sistema es **eth0** y que la conexión con internet es la **eth1**. Consúltense la documentación de keepalived y los ejemplos (normalmente en `/usr/share/doc/keepalived-version/samples`) para entender las funciones de cada directiva. A destacar la directiva **track_interface** que monitoriza la interfaz de conexión a Internet (eth1) de forma que si ésta cae, disminuye la prioridad en un valor de 10 y el router pasa a modo backup. Si se levanta de nuevo, aumenta en 10 y volvería a ser el maestro.

3. Para el equipo que hace las funciones de R2, la configuración es similar pero indicando que es **BACKUP** y que inicia el servicio VRRP con menor prioridad:

```

global_defs {
    router_id R1
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 1
    priority 99
    virtual_ipaddress {
        192.168.100.1/24 brd 192.168.100.255 dev eth0
    }
    authentication {
        auth_type PASS
        auth_pass SeguridadAD
    }
    track_interface {
        eth1 weight -10
        eth1 weight +10
    }
}

```

4. Una vez realizada la configuración, **arranca** el servicio keepalived (service keepalived start o con systemctl start keepalived.service, depende de la distribución) en ambos routers y observa si el equipo R1 ha adquirido la ip en su sistema con el comando `ip address list | grep secondary` que devolvería algo como esto:

```
inet 192.168.100.1/24 brd 192.168.100.255 scope global  
secondary eth0
```

5. Puedes monitorizar el estado del demonio keepalived así como las transiciones entre estados ante fallos en los routers, mediante el comando:

```
tail -f /var/log/messages | grep keepalived
```

o su versión más actual (en distribuciones basadas en systemd):

```
journalctl -u keepalived.service --follow
```

Verificación del servicio de enrutamiento en alta disponibilidad con VRRP

Una vez configurados los clientes con direcciones IP de la red interna, con DNS y con la VIP como puerta de enlace, intenta salir a Internet desde cualquiera de ellos y observa como el tráfico pasa por R1 que es el router activo. Puedes consultar la tabla arp de ambos PC y apuntar la MAC de la puerta de enlace, que es la de R1.

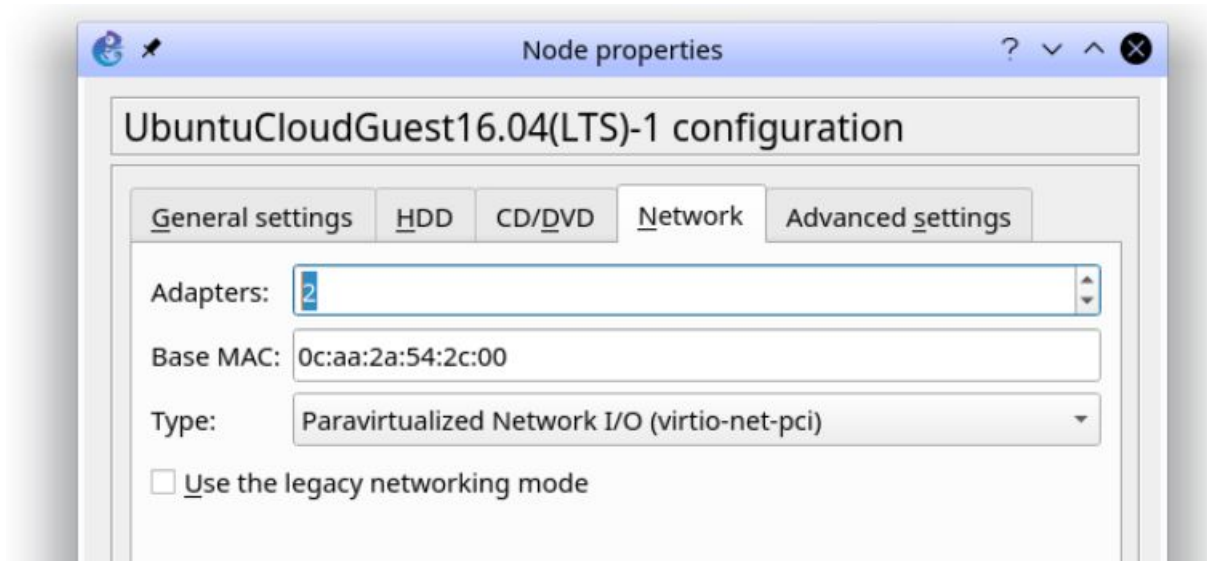
Ahora provoca un fallo en la conexión a Internet de R1 (por ejemplo tirando la interfaz eth1) o apagar la máquina R1. Observa si ahora R2 ha asumido el estado de activo pasando al estado MASTER (desde /var/log/messages) y que tiene la ip virtual ahora con el comando `ip a | grep secondary`. Si observas la tabla arp del PC2, ahora se ha actualizado a la MAC de R2 y el tráfico de Internet para por R2, que es ahora el router activo de la red, sin que los equipos internos de la red hayan notado el fallo. Si recuperas el fallo que has provocado, observarás que R1 vuelve a ser el router maestro.

Configuración de dos balanceadores en alta disponibilidad

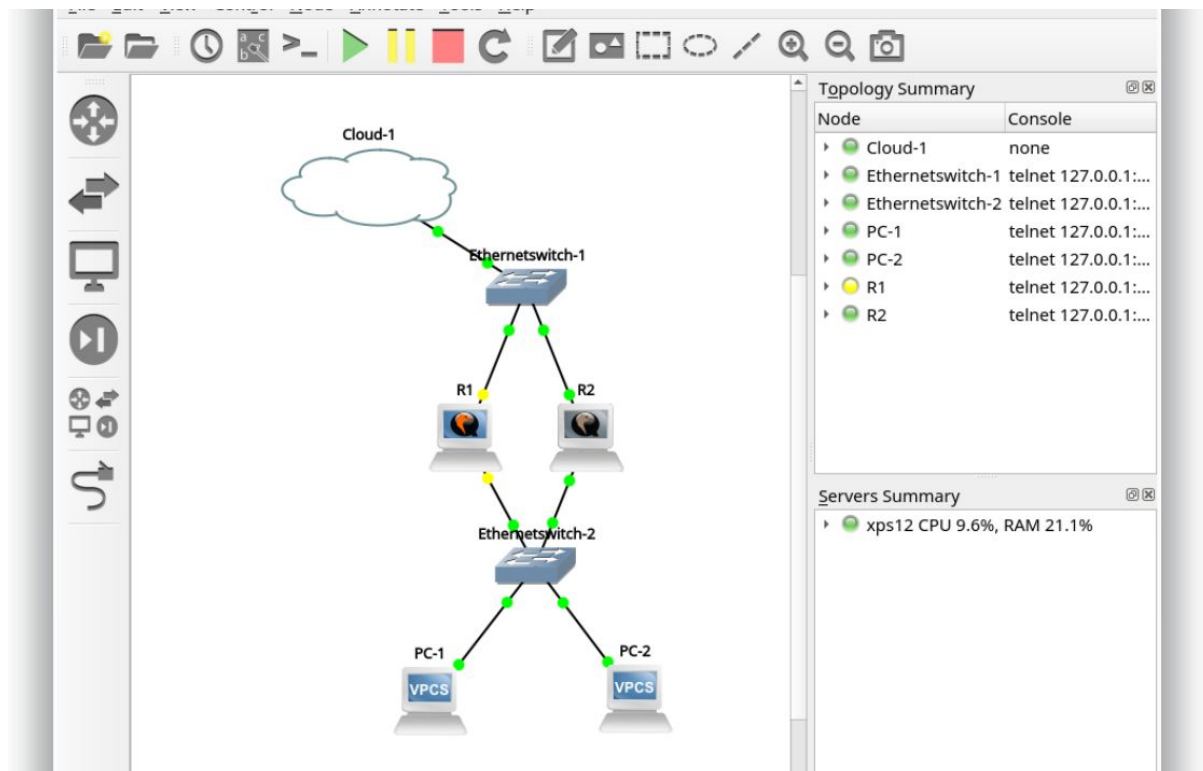
Si se combina la configuración de esta práctica, con la de la práctica anterior (balanceo con keepalived), se puede montar un sistema de alta disponibilidad con una granja de servidores balanceados por dos balanceadores en alta disponibilidad. En la [documentación](#) de keepalived hay ejemplos de este tipo de configuración.

Realización con GNS3

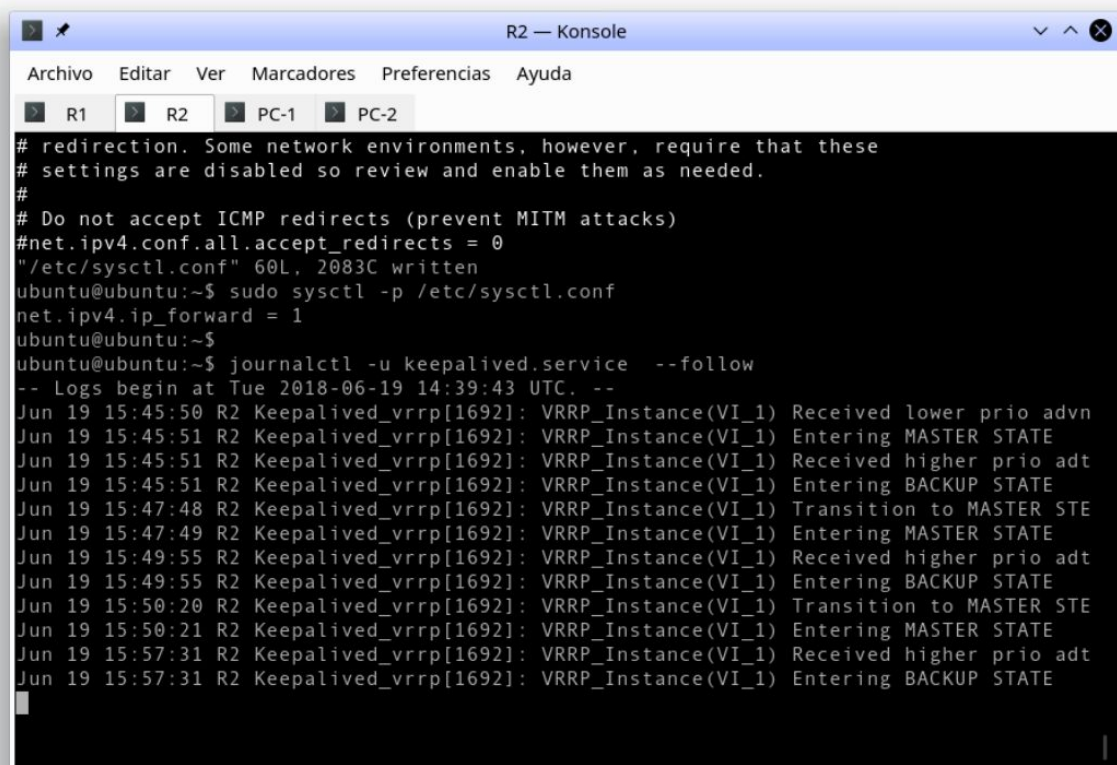
Para configurar cualquier Linux que uséis como router, necesitaréis configurarle dos tarjetas de red:



Aquí tenéis un ejemplo de cómo le he probado yo:



En particular, ahí estoy probando cómo deteniendo uno de los routers el tráfico se redirige por el otro. Para ver esto, podéis ver el *log* relacionado con *keepalived* de los routers:



```
R2 — Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
R1  R2  PC-1  PC-2
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
"/etc/sysctl.conf" 60L, 2083C written
ubuntu@ubuntu:~$ sudo sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ journalctl -u keepalived.service --follow
-- Logs begin at Tue 2018-06-19 14:39:43 UTC. --
Jun 19 15:45:50 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Received lower prio advn
Jun 19 15:45:51 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering MASTER STATE
Jun 19 15:45:51 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Received higher prio adt
Jun 19 15:45:51 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering BACKUP STATE
Jun 19 15:47:48 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Transition to MASTER STE
Jun 19 15:47:49 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering MASTER STATE
Jun 19 15:49:55 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Received higher prio adt
Jun 19 15:49:55 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering BACKUP STATE
Jun 19 15:50:20 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Transition to MASTER STE
Jun 19 15:50:21 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering MASTER STATE
Jun 19 15:57:31 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Received higher prio adt
Jun 19 15:57:31 R2 Keepalived_vrrp[1692]: VRRP_Instance(VI_1) Entering BACKUP STATE
```

Además, en los VPCSs podéis ver la tabla arp con el comando `show arp`. Y podéis forzar la renovación de la tabla arp con `clear arp`.

Recuerda subir a Classroom en la actividad correspondiente, una memoria con capturas describiendo el proceso realizado. Sube también el proyecto GNS3 exportado si lo has hecho con él.