



CLUSTER EN DEBIAN

C.E.S ACADEMIA LOPE DE VEGA

CFGS: 2º Administración de Sistemas Informáticos en Red

Curso: 2017/2018

Asignatura: SERVICIOS EN RED E INTERNET

Prof. Álvaro Márquez

Autor: Rafael Osuna Ventura

Antes de comenzar debemos de configurar las dos máquinas. La máquina de Debian1 estará en red NAT y la de Debian2 en red Interna. Una vez hecho esto vamos a configurar las interfaces de red. Tras esto reiniciamos las interfaces.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6      Fichero: /etc/network/interfaces      Modif

auto eth0
allow-hotplug eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.1.1
    netmask 255.255.255.0

manual eth2:0
iface eth2 inet static
    address 192.168.1.3
    netmask 255.255.255.0
```

```
GNU nano 2.2.6      Fichero: /etc/network/interfaces      Modificado

allow-hotplug eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.1.2
    netmask 255.255.255.0

manual eth2:0
iface eth2 inet static
    address 192.168.1.3
    netmask 255.255.255.0
```

Ahora vamos a modificar el archivo /etc/host de ambas maquinas, luego actualizamos repositorios.

```
GNU nano 2.2.6      Fichero: /etc/hosts      Modif:

127.0.0.1    localhost
127.0.1.1    cluster1.asir cluster1
192.168.1.1   cluster1
192.168.1.2   cluster2

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

```
GNU nano 2.2.6          Fichero: /etc/hosts          Modificado
127.0.0.1      localhost
127.0.1.1      cluster.asir    cluster1
192.168.1.1    cluster1
192.168.1.2    cluster2
# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Ahora vamos a instalar ssh en ambas máquinas, luego instalaremos el paquete corosync:

```
root@cluster1:/home/cluster1# apt-get install ssh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  ssh
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 6 no actualizados
Se necesita descargar 120 kB de archivos.
Se utilizarán 164 kB de espacio de disco adicional después de esta operación.
Des:1 http://ftp.fi.debian.org/debian/ jessie/main ssh all 1:6.7p1-5+deb8u1
[kB]
```

```
root@cluster1:/home/cluster1# apt-get install corosync
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libcfg4 libconfdb4 libcoroipec4 libcoroipecs4 libcpq4 libevs4 liblogsys4
  libpload4 libquorum4 libsam4 libtotem-pg4 libvotequorum4
Se instalarán los siguientes paquetes NUEVOS:
  corosync libcfg4 libconfdb4 libcoroipec4 libcoroipecs4 libcpq4 libevs4
```

Una vez instalados estos paquetes en ambas máquinas, instalamos el paquete “jessie-backports pacemaker” en la maquina Debian1. Y realizamos un ping para comprobar que hay conexión entre ambas maquinas.

Tras esto vamos a crear una clave de comunicación con corosync y la copiamos a la otra máquina mediante ssh:

```
root@cluster1:/home/cluster1# corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/random.
Press keys on your keyboard to generate entropy.
```

```
root@cluster1:/home/cluster1# scp /etc/corosync/authkey cluster1@192.168.1.2:/home/cluster1
The authenticity of host '192.168.1.2 (192.168.1.2)' can't be established.
```

Ahora vamos a modificar el archivo de configuración de corosync en ambas máquinas, indicando nuestra ip. También instalamos el siguiente paquete y modificamos su archivo de configuración en ambas máquinas.

```
GNU nano 2.2.6      Fichero: /etc/corosync/corosync.conf      Modific
# also set rrp_mode.
interface {
    # Rings must be consecutively numbered, starting at 0.
    ringnumber: 0
    # This is normally the *network* address of the
    # interface to bind to. This ensures that you can use
    # identical instances of this configuration file
    # across all your cluster nodes, without having to
    # modify this option.
    bindnetaddr: 192.168.1.0
    # However, if you have multiple physical network
    # interfaces configured for the same subnet, then the
    # network address alone is not sufficient to identify
    # the interface Corosync should bind to. In that case,
    # configure the *host* address of the interface
    # instead:
    # bindnetaddr: 192.168.1.1
    # When selecting a multicast address, consider RFC
    # 2365 (which, among other things, specifies that
```

```
root@cluster1:/home/cluster1# apt-get install crmsh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
```

```
GNU nano 2.2.6      Fichero: /etc/default/corosync

# Corosync runtime directory
#COROSYNC_RUN_DIR=/var/lib/corosync

# Path to corosync.conf
#COROSYNC_MAIN_CONFIG_FILE=/etc/corosync/corosync.conf

# Path to authfile
#COROSYNC_TOTEM_AUTHKEY_FILE=/etc/corosync/authkey

# Command line options
#OPTIONS=""
START=yes
```

Ahora desactivamos el mecanismo de stonith en el cluster 2 y configuramos la ip virtual:

```
root@cluster2:/etc/corosync# crm configure property stonith-enabled=false
root@cluster2:/etc/corosync#
```

```
root@cluster1:/home/cluster1# crm configure primitive FAILOVER-ADDR ocf:heartbeat:IPaddr2 params ip=192.168.1.3 nic=eth1:0 op monitor interval=10s meta is-managed=true
```

Para finalizar instalamos apache, al instalarlo nos sale un error el cual nos impide continuar:

```
root@cluster1:/home/cluster1# apt-get install apache2
```