

Enxeñaría de Infraestruturas Informáticas

Grao en Enxeñaría Informática



UNIVERSIDADE DA CORUÑA

PRÁCTICAS SOBRE CLUSTERS DE ALTA DISPOÑIBILIDADE E BALANCEO DE CARGA

Xoán C. Pardo
xoan.pardo@udc.gal

PRACTICA 0

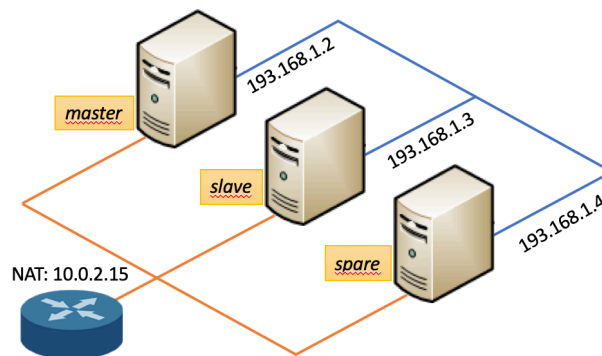
Configuración dun cluster de máquinas virtuais

Obxectivo

O obxectivo desta práctica é realizar a configuración inicial do clúster de máquinas virtuais (VMs) que se usará nas prácticas de clusters HA con Pacemaker/Corosync.

Configuración para as prácticas

O cluster que usaremos para as prácticas está formado por tres VMs (nós) coa configuración que pode verse na figura seguinte:



O habitual, nun caso real, é que os nós do cluster estean conectados como mínimo a dúas redes diferentes, unha privada para a comunicación entre eles e outra pública para o acceso dos clientes aos servizos aloxados nos nós. Ademais, aínda que nesta práctica por simplicidade non o imos facer así, é recomendábel que cada interface estea redundada para evitar puntos únicos de fallo (SPOF) no acceso ás redes ás que o nó estea conectado.

Nomenclatura a utilizar para nomear os recursos

Na realización das prácticas **É OBRIGATORIO** usar o seguinte formato para nomear os recursos:

<iniciais do nome e apelidos><curso>-<nome do recurso>

De non seguir esta nomenclatura **AS PRÁCTICAS ESTARÁN SUSPENSAS**

Por exemplo, o alumno Xoán Carlos Pardo Martínez, que fixo as prácticas no curso 2019/20 utilizará o seguinte prefixo: **xcpm1920-***<nome do recurso>*. Así os nós *master*, *slave* e *spare* da figura anterior, terían os seguintes nomes no seu caso: **xcpm1920-master**, **xcpm1920-slave** e **xcpm1920-spare**.

Nos enunciados das prácticas usárase o prefixo do exemplo anterior. Cada alumno terá que substituílo cos datos do curso e as iniciais do seu nome que correspondan en cada caso.

As ferramentas que usaremos para a realización desta práctica son:

- O software de virtualización **VirtualBox** (<https://www.virtualbox.org>) para executar as VMs
- A ferramenta **Vagrant** (<https://www.vagrantup.com>) para a automatización da creación e configuración das VMs.

Creación do proxecto Vagrant

Neste apartado vamos crear un proxecto Vagrant para configurar automaticamente o cluster coas 3 VMs necesarias para o resto das prácticas.

O requisito previo é que teñas instalado no teu equipo VirtualBox e Vagrant. **Se non os utilizaches nunca, comeza por facer o seguinte tutorial**, no que xa se fai a instalación dambas ferramentas: <https://learn.hashicorp.com/collections/vagrant/getting-started>

NOTA: podes saltarte os pasos "Share an Environment" e "Explore other providers" do tutorial.

Para configurar o cluster usando un proxecto Vagrant sigue os pasos seguintes:

1. Nun novo cartafol executa o comando

```
$ vagrant init hashicorp/bionic64
```

Isto creará un **Vagrantfile** que configura unha VM Ubuntu 18.04 para VirtualBox.

NOTA: Nun cluster real, o habitual é que as conexións do administrador aos nós do cluster se fagan usando unha rede privada dedicada só a iso. Por simplificar a práctica, nos vamos a conectarnos usando a NAT. Vagrant xa configura automaticamente, en cada VM que se defina no **Vagrantfile**, unha interface de rede NAT e o acceso ssh sen *password*, polo que o que necesitamos xa está configurado por defecto.

Edita o **Vagrantfile** creado no paso anterior da maneira seguinte:

NOTA: Para realizar os seguintes pasos necesitarás consultar a información que podes atopar aquí: <https://www.vagrantup.com/docs/multi-machine> e aquí: <https://www.vagrantup.com/docs/providers/virtualbox/configuration>

2. Define as 3 VMs que forman o cluster: **xcpm1920-master**, **xcpm1920-slave** e **xcpm1920-spare**.
3. Para optimizar o espazo ocupado polos discos das VMs, activa no **Vagrantfile** para o provedor VirtualBox, a opción **linked_clones**
4. En cada VM configura unha rede privada coa IP que lle corresponda a cada VM, segundo o indicado na figura da primeira páxina.

Unha vez definidas as VMs, vamos configurar o aprovisionamento. Este terá dúas partes, unha que é común a todas as VMs, e outra que é específica de cada VM. Para a primeira usaremos un *script* `bootstrap.sh` almacenado nun arquivo e, para as segundas, un *script* `INLINE` na definición de cada VM. Os pasos a seguir son os seguintes:

5. Para a configuración común das VMs, crea un *script* `bootstrap.sh` no cartafol do proxecto que faga o seguinte:

- Actualice os repositorios e paquetes instalados

```
apt-get update
apt-get -y upgrade
```

NOTA: Vagrant executa os *scripts* de aprovisionamento con permisos de `root`, polo que non é necesario indicar `sudo` diante dos comandos.

- Cambie a configuración do `LOCALE`

```
locale-gen es_ES.UTF-8
update-locale LANG=es_ES.UTF-8 LC_MESSAGES=POSIX
```

- Engada a IP e o nome de cada VM do cluster no ficheiro `/etc/hosts`

NOTA: Isto é necesario facelo porque na rede privada non dispoñemos dun servidor DNS, e queremos poder usar os nomes das VMs nos comandos.

```
echo "#xcpm1920-cluster" >> /etc/hosts
echo "193.168.1.2 xcpm1920-master" >> /etc/hosts
echo "193.168.1.3 xcpm1920-slave" >> /etc/hosts
echo "193.168.1.4 xcpm1920-spare" >> /etc/hosts
```

- Engade no `Vagrantfile` unha configuración `provision` que use o *script* `bootscript.sh` para aprovisionar todas as VMs.

NOTA: ollo que a posición que ocupe esta configuración no `Vagrantfile` é importante. Ten que estar situada de maneira que se execute antes que o aprovisionamento específico de cada VM. Consulta a documentación.

6. Para a configuración individual de cada VM, engade no `Vagrantfile`, na definición de cada VM, unha configuración `provision` que use un *script* `INLINE` para facer o seguinte:

- Configurar o nome da VM, p.e. o comando para a VM *master* sería

```
hostnamectl set-hostname xcpm1920-master
```

- Instalar Corosync e Pacemaker

```
apt-get install -y crmsh csync2
```

NOTA: aínda que este comando é o mesmo para todas as VMs, non o podemos incluír no script `bootstrap.sh`, porque para que a instalación do Corosync e o Pacemaker sexa correcta, é necesario que cada VM teña un nome único configurado previamente.

Comprobación do proxecto Vagrant

Para comprobar que o proxecto está configurado correctamente sigue os pasos seguintes:

7. Inicia o cluster. Comproba que na saída non haxa ningunha mensaxe de erro.

```
$ vagrant up
```

8. Comproba que as 3 VMs aparecen en estado `running`.

```
$ vagrant status
```

Os seguintes pasos (8-13) mostran un exemplo para o nó *master*. Repíteos en todas as VMs do cluster:

9. Conéctate ao *master*.

```
$ vagrant ssh xcpm1920-master
```

10. Comproba que o nome do nó está correctamente configurado¹.

```
master$ hostname  
xcpm1920-master
```

11. Comproba que as interfaces de rede están correctamente configuradas. Ao executar o comando `ifconfig` debería aparecer unha interface `eth0` con IP 10.0.2.15, que é a da rede NAT, e unha interface `eth1` con IP 193.168.1.2, que é a da rede privada entre os nós.

```
master$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
[...]  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 193.168.1.2 netmask 255.255.255.0 broadcast 193.168.1.255  
[...]
```

¹ Cando sexa necesario distinguir en que nó se introducen os comandos usaremos nos exemplos os *prompts* `master$`, `slave$` e `spare$`.

12. Comproba que podes facer ping aos outros nós usando as súas IPs estáticas.

```
master$ ping -c 3 193.168.1.3
master$ ping -c 3 193.168.1.4
```

13. Comproba que podes facer ping aos outros nós usando os seus nomes.

```
master$ ping -c 3 xcpm1920-slave
master$ ping -c 3 xcpm1920-spare
```

14. Comproba que Corosync/Pacemaker están instalados e funcionando.

```
master$ sudo crm status
Stack: corosync
Current DC: xcpm1920-master (version 1.1.18-2b07d5c5a9) - partition
WITHOUT quorum
Last updated: Mon Sep 14 11:58:18 2020
Last change: Thu Aug 13 20:41:10 2020 by hacluster via crmd on xcpm1920-
master

1 node configured
0 resources configured

Online: [ xcpm1920-master ]

No resources
```

NOTA: de momento cada nó "vese" a si mesmo como único membro do cluster HA.

Que facer se algo falla?

Se algunha das configuracións non che funciona, tes dúas opcións:

- Apaga a VM (`vagrant halt`), edita o `Vagrantfile` e/ou o `script bootstrap.sh` e reinicia e reaproviona a VM (`vagrant up --provision`)
- Elimina a VM (`vagrant destroy`), edita o `Vagrantfile` e/ou o `script bootstrap.sh` e crea a VM de novo (`vagrant up`). Esta opción é máis drástica que a anterior.

Solución da práctica

Para evitar problemas posteriores debidos a diferencias e erros de configuración, ao comezo da práctica seguinte daranse as instrucións oportunas para descargar un proxecto Vagrant coa solución desta práctica.