Prácticas COS – Curso 2022-23 – Práctica 2

Infraestructura del clúster

Trabajaremos desde un ordenador del laboratorio, arrancando en Linux (64 bits). Tras iniciar sesión en la máquina del seminario, iniciaremos sesión remota ssh en el clúster *cac*:

```
ssh -l cosxx cac1.disca.upv.es –X
Password: xx202223cos
```

- En primer lugar obtendremos información relativa al hardware, tipo de procesador, número de cores, frecuencia actual de trabajo, frecuencia máxima, tamaño de RAM, etc.
- Averiguaremos la distribución Linux instalada en los nodos y la versión del kernel.
- Analizaremos la configuración de la red:

```
/sbin/ifconfig
```

- La tabla de encaminamiento:

```
/sbin/route
```

- La tabla ARP:

```
/sbin/arp
```

¿Qué significan las siglas ARP?

- Realizaremos algunas consultas al DNS:

```
host www.upv.es
host 158.42.4.23
nslookup 8.8.8.8
dig www.upv.es
dig –x 158.42.4.23
dig @8.8.8.8 –x 8.8.4.4
```

  Explicar brevemente la información que se obtiene con los diferentes comandos. Consultar la página del manual para conocer las opciones disponibles.

- Analizar los archivos de configuración de las interfaces de red (puede emplearse cat, more, less):

```
cd /etc/sysconfig/network
cat ifcfg-XXX
```

- Analizar algunos archivos de configuración del sistema:

```
cat /etc/passwd
cat /etc/nsswitch.conf
cat /etc/hosts
cat /etc/resolv.conf
cat /etc/dhcpd.conf
```

- Analizar los scripts de los niveles de ejecución. Por ejemplo, para el nivel 5:

```
cd /etc/init.d/rc5.d
ls –la
```

  Observar los enlaces simbólicos.

- Listar los scripts de configuración

```
cd /etc/init.d
ls
```

Dar un vistazo a alguno de ellos. Por ejemplo:

```
more network
more sshd
more single
```

- Cambiar al directorio *home* del usuario:

```
cd $HOME
```

- Configurar ssh para que no nos pida contraseña (acceso mediante clave pública) al iniciar sesión en el resto de máquinas del clúster:
  En *cac1*, crearemos las claves, pública y privada, sin contraseña (*passphrase*)

```
ssh-keygen
```

Como en el cluster *cac* el directorio *home* es compartido por todas las máquinas, en este caso basta con copiar la clave pública al archivo `authorized_keys` del directorio `.ssh` local:

```
cp .ssh/id_rsa.pub .ssh/authorized_keys
```

Si el directorio home no estuviera compartido por todas las máquinas (por ejemplo, esto ocurre con el usuario *root*), habría que copiarla o añadirla a cada una de ellas:

```
scp .ssh/id_rsa.pub cac2:.ssh/nueva_clave
ssh cac2
cat .ssh/nueva_clave >> .ssh/authorized_keys
exit
…
```

También puede utilizarse la orden `ssh-copy-id`, que copia la clave pública sobre el archivo authorized_keys de la máquina destino:

```
ssh-copy-id cac2
ssh-copy-id cac3
…
```

Conectarse a alguna máquina del clúster, por ejemplo, *cac2*. Ya no debería pedir contraseña:

```
ssh cac2
```

Sin embargo, la primera vez que nos conectamos, la máquina local puede no conocer la máquina remota, solicitando su inclusión en el archivo `known_hosts.` Para que lo incluya sin preguntar:

```
ssh cac3 –o StrictHostKeyChecking=no
```

También podemos obtener las claves y añadirlas al archivo `known_hosts:`

```
ssh-keyscan –t rsa cac4 >> .ssh/known_hosts
ssh cac4
```

- Ejecutar algunas órdenes remotas con ssh. Desde *cac1:*
  - Lista de archivos en nuestro directorio *home* de *cac2:*

```
ssh cac2 ls
```

- Listado extendido de archivos en nuestro directorio *home* de *cac2:*

```
ssh cac2 ls –la
```

- Obtener los procesos en ejecución en *cac2:*

```
ssh cac2 ps aux
```

- Obtener la configuración de la red de *cac2:*

```
ssh cac2 /sbin/ifconfig
```

- Obtener el contenido del archivo `/etc/hosts` de *cac2:*

```
ssh cac2 cat /etc/hosts
```

- Procesos en ejecución en todos los servidores del cluster (*cac2 a cac8):*

```
for i in 2 3 4 5 6 7 8; do echo cac$i ; ssh cac$i ps ; done
for i in `seq 2 8`; do echo cac$i ; ssh cac$i ps ; done
```

- Para evitar tener que emplear bucles de bash cada vez que queremos enviar una orden a todos los nodos, **prepararemos un script para lanzar órdenes a todos los nodos**. El archivo se llamará **`psh`**. Algunos ejemplos de utilización:

```
./psh ls
./psh “ls –la“
./psh ps aux
./psh “ps aux”
./psh /sbin/ifconfig
./psh “cat /etc/hosts”
```

Hay alguna diferencia al ejecutar *psh ps aux* y *psh “ps aux”*?

El script mostrará a qué máquina corresponde la salida mostrada. Por ejemplo:

```
psh ls
==========
cac1
----------
id_rsa
id_rsa.pub
==========
cac2
----------
id_rsa
id_rsa.pub
…
```

Puede emplearse cualquier editor (vi, nano, jpico, kate)
Dar derechos de ejecución:

```
chmod +x psh
```

Qué pasaría si hacemos como root?

```
./psh poweroff
```

- Modificar el script `psh` para que las órdenes se lancen a todos los nodos simultáneamente. El nuevo script se llamará `ppsh` y le pasaremos la orden a ejecutar. Por ejemplo:

```
./ppsh “dd if=/dev/zero of=prueba bs=1K count=10”
```

```
./ppsh “rm prueba”
```

- Preparar un script para lanzar órdenes a un grupo de nodos. El archivo se llamará `pgsh` y le pasaremos tres argumentos: los números de los nodos inicial y final, así como la orden a ejecutar. Por ejemplo:

```
./pgsh 2 5 ps
```

- Preparar un script para copiar un mismo archivo a todos los nodos en una ubicación determinada. El script se llamará `pcp` e incluirá como argumentos el archivo a copiar y el archivo de destino. Por ejemplo:

```
./pcp /etc/hosts /tmp/cosXX_hosts
```

  Una vez comprobado que el fichero se ha copiado en cada uno de los nodos se debe eliminar.

- Probar el paquete pdsh

```
pdsh –w cac[2-4] –R ssh date
pdsh –w cac[2-4] –R ssh date | dshbak
pdsh –w cac[1-6] –x cac[3-6] –R ssh date | dshbak
```

  Acceder a la página de manual y analizar sus opciones.
  Comparar el comportamiento de `pdsh` y `ssh` (scripts `psh` y `ppsh`). Por ejemplo:

```
./psh df
./ppsh df
pdsh –w cac[2-8] –R ssh df
```

  Probar también con las órdenes `ps, uptime, w, /sbin/ifconfig`