



PRACTICA SCRIPT MAESTRO PARA LA CONFIGURACION DE UN CLUSTER LINUX

ADMINISTRACION DE SISTEMAS INFORMATICOS

CURSO 2017/2018

INTEGRANTES:

Ismael Ortega Sánchez – 140006

Daniel Sánchez Muñoz - 130338

David Amador Señorís – 130030

GRUPO: 9

Contenido

1. INTRODUCCION	2
2. SCRIPT CONFIGURACION CLUSTER.....	3
3. SERVICIOS	4
3.1 SERVICIO DE MONTAJE	4
3.2 SERVICIO RAID.....	4
3.3 SERVICIO LVM	4
3.4 SERVICIO NIS.....	4
3.4.1 SERVIDOR NIS.....	4
3.4.2 CLIENTE NIS.....	5
3.5 SERVICIO NFS	5
3.5.1 SERVIDOR NFS.....	5
3.5.2 CLIENTE NFS	5
3.6 SERVICIO BACKUP	5
3.6.1 SERVIDOR BACKUP.....	5
3.6.2 CLIENTE BACKUP	5
4. CONCLUSIONES.....	6

1. INTRODUCCION

El objetivo del proyecto era programar un script bash que sea capaz de instalar y configurar de forma totalmente automática un conjunto de servicios en un grupo de máquinas conectadas a una red local para formar un cluster.

Para la resolución de la práctica hemos decidido utilizar varios scripts auxiliares, de forma que cada servicio tiene su propio script que son llamados por el script principal `configuracion_cluster`.

Script Principal: `configuracion_cluster`

Tenemos dos carpetas:

- Carpeta Config: Donde se encuentran los ficheros de configuración
- Carpeta Servicios: Donde se encuentran los scripts de los servicios

Contenido de la carpeta Config:

Nombre fichero	Descripción
<code>cluster.cfg</code>	Fichero de configuración cluster
<code>mount.cfg</code>	Fichero de configuración mount
<code>raid.cfg</code>	Fichero de configuración raid
<code>lvm.cfg</code>	Fichero de configuración lvm
<code>snfs.cfg</code>	Fichero de configuración snfs
<code>cnfs.cfg</code>	Fichero de configuración cnfs
<code>snis.cfg</code>	Fichero de configuración snis
<code>cnis.cfg</code>	Fichero de configuración cnis
<code>backup_server.cfg</code>	Fichero de configuración backup_server
<code>backup_client.cfg</code>	Fichero de configuración backup_client

Contenido de la carpeta Servicios:

Nombre fichero	Descripción
<code>mount.sh</code>	Script Mount.sh
<code>raid.sh</code>	Script raid.sh
<code>lvm.sh</code>	Script lvm.sh
<code>snfs.sh</code>	Script snfs.sh
<code>cnfs.sh</code>	Script cnfs.sh
<code>snis.sh</code>	Script snis.sh
<code>cnis.sh</code>	Script cnis.sh
<code>backup_server.sh</code>	Script backup_server.sh
<code>backup_client.sh</code>	Script backup_client.sh

2. SCRIPT CONFIGURACION CLUSTER

Según el enunciado de la práctica, el script configurar cluster.sh debería ser capaz de instalar y configurar una serie de servicios en las máquinas que se soliciten. Cada servicio tendrá unos parámetros asociados, dependiendo de su funcionalidad. Además, debe realizar de forma general las siguientes tareas, cuando sea necesario:

- Instalación de software necesario para cada servicio, si no está presente en la máquina, asegurándose de que no es necesaria ninguna intervención manual.
- Creación de directorios necesarios, como puntos de montaje, etc.

Para elaboración de nuestra práctica, hemos desarrollado un Script configuracion_cluster que realiza los siguientes pasos en su ejecución:

1. Comprobamos al utilizar **\$./configurar_cluster.sh fichero_configuracion** si el número de argumentos es correcto, de no serlo provoca un error:

Uso: \$0 <fichero_configuracion>

2. Comprobamos si se encuentran disponibles todos los ficheros, en caso de que alguno no esté disponible provoca la salida:

CLUSTER: El fichero ./conf/\$fich no está disponible. Abortando ejecución.

3. La siguiente función de configuracion_cluster es, mediante un switch, configurar el servicio auxiliar a utilizar. En caso de fallo, provoca el error:

CLUSTER: Error en el servicio indicado (\$SERVICIO). Abortando...

4. Si se puede establecer conexión, creamos la carpeta del proyecto, en caso de no poder, se produce el siguiente error:

CLUSTER: No es posible establecer conexión con la máquina. Abortando...

5. Copiamos los archivos necesarios.
6. Ejecutamos el servicio correspondiente.
7. Se eliminan los ficheros de configuración temporales que hemos utilizado.
8. Comprobamos que existe el fichero de configuración, si no existe, produce error:

CLUSTER: No existe el fichero \$1

9. Comprobamos que el fichero de configuración cumple con estas propiedades:

- Las líneas en blanco se ignorarán.
- Las líneas que empiecen por # (comentarios) se ignorarán.
- Las líneas de configuración deberán cumplir el siguiente formato:
máquina-destino nombre-del-servicio fichero-de-perfil-de-servicio

Cabe destacar que hemos decidido estructurar la práctica de esta forma (script principal mas los servicios adicionales) debido a que proporciona modularidad y facilita la adición o edición de los mismos.

3. SERVICIOS

En este apartado explicaremos el funcionamiento de cada uno de los servicios que compone el programa. Todos estos comprueban a su vez de ficheros de configuración y que cumplen los parámetros especificados para dichos ficheros.

Antes de mostrar lo que hace el script auxiliar de cada servicio hay que destacar que cada uno de los scripts comprueba que existe el fichero de perfil de servicio y si hay en el algún error en su contenido como por ejemplo que falte alguno de los parámetros o que no haya sido pasado correctamente. Y además obtienen los parámetros de su fichero de perfil de servicio.

3.1 SERVICIO DE MONTAJE

Montaje de un dispositivo en un punto del árbol de directorios. El servicio `Mount.sh` intenta montar el directorio en un punto de montaje, según los parámetros que se han pasado en el fichero de perfil de servicio (`mount.cfg`). Si el montaje es correcto, la configuración se guarda en el fichero `/etc/fstab`, en caso contrario da error.

3.2 SERVICIO RAID

Configuración de un RAID software, usando `mdadm`. El servicio **`raid.sh`** lo primero que hace es la instalación de la herramienta `mdadm` sin que necesite input de parte del usuario y finalmente monta el RAID con los parámetros que se han pasado en el fichero de perfil de servicio (`raid.cfg`) y guarda la configuración.

3.3 SERVICIO LVM

Configuración de un grupo de volúmenes lógicos. El servicio **`lvm.sh`** lo primero que hace es la instalación del servicio LVM. A continuación, se inicializan los volúmenes físicos que se han pasado como parámetro en el fichero de perfil de servicio (`lvm.cfg`). Posteriormente se crea el grupo de volúmenes y por último se crean volúmenes lógicos.

3.4 SERVICIO NIS

- 3.4.1 SERVIDOR NIS

Configuración de un servidor de NIS. El servicio **`snis.sh`** comprueba que los parámetros pasados son correctos a través del fichero de perfil de servicio (`snis.cfg`) y realiza una instalación de los paquetes necesarios NIS y se añade una línea de configuración al fichero de configuración `/etc/yp.conf` y se realiza un reinicio del servicio.

- [3.4.2 CLIENTE NIS](#)

Configuración de un cliente de NIS. El servicio **cnis.sh** comprueba que los parámetros pasados a través del fichero de perfil de servicio (cnis.cfg) y realiza una instalación de los paquetes necesarios NIS. Finalmente se comprueba la configuración y se realiza un reinicio del servicio.

[3.5 SERVICIO NFS](#)

- [3.5.1 SERVIDOR NFS](#)

Configuración de un servidor de NFS con una serie de directorios que exportar. El servicio **snfs.sh** obtiene sus parámetros del fichero de perfil de servicio (snfs.cfg) y realiza la instalación de los paquetes necesarios nfs-common y nfs-kernel-server sin que necesite input de parte del usuario. Posteriormente añade los directorios exportados al final del fichero /etc/exports y se produce un reinicio del servicio.

- [3.5.2 CLIENTE NFS](#)

Configuración de un cliente de NFS. El servicio **cnfs.sh** obtiene sus parámetros del fichero de perfil de servicio (cnfs.cfg) y realiza el montaje de los directorios. Posteriormente realiza la instalación del paquete nfs-common sin que necesite input de parte del usuario y comprueba la IP del servidor.

[3.6 SERVICIO BACKUP](#)

- [3.6.1 SERVIDOR BACKUP](#)

Configuración de un cliente Backup. El servicio Backup_server.sh crea el directorio obtenido del fichero de perfil de servicio (backup_server.cfg) donde se va a realizar el Backup

- [3.6.2 CLIENTE BACKUP](#)

Configuración de un cliente de Backup. Hemos decidido utilizar la herramienta rsync porque es una herramienta potente de Linux para transferir y sincronizar archivos y directorios en el mismo ordenador o entre varios a través de la red, también es fácil de utilizar. El servicio **backup_client.sh**, obteniendo los parámetros del fichero de perfil de servicio (backup_client.cfg), comprueba si existe el directorio del que se quiere hacer Backup y el directorio de la máquina destino donde se va a realizar el Backup y si la dirección IP del server corresponde a una máquina. Posteriormente, se instala la herramienta rsync y se comprueba si proceso demonio que hará el Backup ya existe o hay que añadirlo al fichero /etc/crontab y estableciendo la periodicidad con la que se va a realizar el backup.

4. CONCLUSIONES

Como conclusión de esta práctica queremos exponer algunos de los problemas que hemos encontrado durante la realización de la misma:

1. El primer problema encontrado fue a la hora de configurar las máquinas virtuales y en realizar configuraciones para poder trabajar con otros sistemas operativos que nos llevó un tiempo.
2. Problemas al trabajar con diferentes editores de código y sus fines de línea que producían errores cuando ejecutábamos los scripts.
3. Aunque pocos, también hemos tenido algún problema relacionado con el lenguaje bash a la hora de realizar algún mandato.
4. La realización de ciertas pruebas nos ha resultado costosa debido a que los scripts de los servicios realizan modificaciones en la máquina virtual y en algunos casos pueden producir fallos que hay que volver a configurar.

Sin embargo, gracias a este proyecto, una vez solventados los problemas comentados anteriormente, podemos decir que hemos ampliado nuestro conocimiento sobre bash, UNIX y en la administración de sistemas informáticos.