Universidad Politécnica de Madrid

Escuela Técnica Superior
de Ingeniería Informática

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Departamento de Lenguajes, Sistemas Informáticos e
Ingeniería de Software

Administración de Sistemas Informáticos

Memoria de Proyecto Práctico

Script maestro para la configuración de un cluster Linux

4º Curso — 7º Semestre

Autores

Ramírez Solans, Antonio
Nº Matrícula: 170035

Miguel Alonso, Carlos
Nº Matrícula: 170243

Aouad Idrissi Boulid, Younes
Nº Matrícula: 170155

22 de diciembre de 2020

Índice de Contenidos

1. Introducción del proyecto **1**

2. Desarrollo del proyecto **2**
2.1. Decisiones de diseño 2

3. Servicios implementados **3**
3.1. Mount 3
3.2. Raid 3
3.3. LVM 4
3.4. NIS 4
3.4.1. Servidor 4
3.4.2. Cliente 4
3.5. NFS 4
3.5.1. Servidor 4
3.5.2. Cliente 4
3.6. Backup 4
3.6.1. Servidor 4
3.6.2. Cliente 4

4. Códigos de error **5**
4.1. Comúnes a todos los servicios 5
4.2. Mount 5
4.3. Raid 5
4.4. LVM 5
4.5. NIS 6
4.5.1. Servidor 6
4.5.2. Cliente 6
4.6. NFS 6
4.6.1. Servidor 6
4.6.2. Cliente 6
4.7. Backup 6
4.7.1. Servidor 6
4.7.2. Cliente 6

5. Testing **7**

1. Introducción del proyecto

Este proyecto se basa en el desarrollo de un *script* (o conjunto de ellos) en Bash que permitan la configuración de diferentes servicios en un *cluster* formado por múltiples máquinas ejecutando un sistema Linux.

De esta forma, el administrador puede configurar diferentes servicios en diferentes máquinas de forma simultánea, teniendo que escribir solo los archivos de configuración de cada servicio y el archivo de configuración general.

Los servicios que soportados por el *script* de configuración son:

- Mount
- Raid
- LVM
- NIS (cliente + servidor)
- NFS (cliente + servidor)
- Backup (cliente + servidor)

2. Desarrollo del proyecto

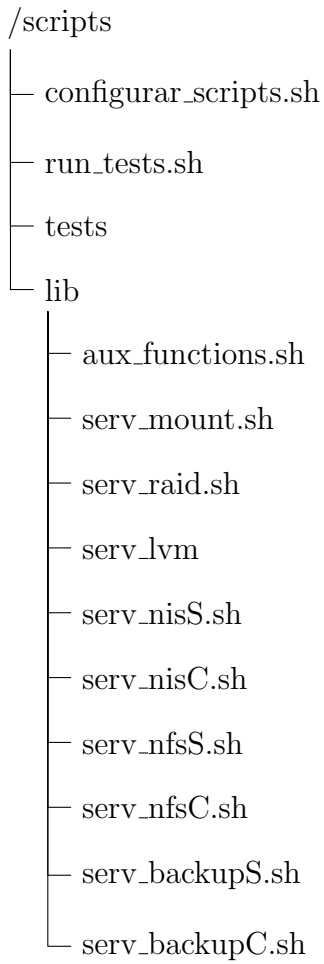
2.1. Decisiones de diseño

Durante el desarrollo de la práctica, vimos que había ciertos aspectos de ella que podían realizarse de diferentes maneras, tales como la ejecución de los comandos en la máquina destino, el formato del/los *script/s*, etc...

Por ello, se han tomado la siguientes decisiones:

- Tener un *script* central que ejecutará el administrador (`configurar_cluster.sh`) que leerá línea a línea el fichero de configuración que se proporciona como argumento, y llamará a la función perteneciente al servicio leído, que está localizada en la carpeta `lib`, cada servicio en un fichero Bash diferente. Además, ciertas funciones se han puesto en común en el archivo `aux_functions.sh`, de forma que todos los ficheros que lo requieran puedan acceder a estas funciones de forma sencilla y sin tener código repetido multitud de veces.

  A continuación se muestra el árbol de directorios de los *scripts* del proyecto):



- Para facilitar la ejecución e implementación, se ha decidido que los comandos se ejecutarán a través de `SSH`, en vez de enviar un fichero de configuración y ejecutarlo. Esta decisión aumenta el tráfico de la red, ya que se tienen que enviar los comandos uno a uno, tanto para comprobaciones como para configuraciones, pero permite un control más preciso de la ejecución.

3. Servicios implementados

A continuación se explicarán los diferentes servicio que se han implementado y su funcionamiento general, incluyendo cualquier detalle relevante a la hora de entender su funcionamiento.

Las funciones de cada servicio reciben 4 argumentos, que son, el fichero de configuración (`fichero_configuración`), la dirección del host (ya sea una dirección IP o un nombre), el fichero de configuración del servicio (e.g. `raid.conf`) y el número de línea del fichero de configuración en el que se han leído los datos anteriores.

De igual forma, en todas las funciones se ha re-direccionado la lectura del fichero de configuración de cada servicio por el **descriptor de fichero 3**.

Todos los *scripts* de los servicios retornarán 0 si se ha realizado satisfactoriamente la operación, y, si se ha producido algún error, un valor específico representando ese error (ver *Códigos de Error*).

3.1. Mount

El servicio *mount* leerá de su fichero de configuración dos campos, el dispositivo a montar y el punto en el que montar dicho dispositivo.

El *script* montará el dispositivo en el punto de montaje tras comprobar que se han leído todos los campos necesarios, que dicho dispositivo existe en la máquina, y que el punto de montaje es un directorio vacío. Si el punto de montaje no existe, se crea.

3.2. Raid

El servicio *raid* leerá 3 campos de su fichero de configuración, el nombre del nuevo dispositivo RAID, el nivel de RAID, y los dispositivos que compondrán el RAID.

Lo primero que hace el *script* es comprobar si la herramienta *mdadm*, que usará para la instalación del RAID, está instalada, si no, la instalará.

Tras esto, comprueba la correcta lectura de los campos, tras lo que se comprueba que el nivel de RAID leído es uno de los niveles soportados (0, 1, 5, 6 o 10). Tras esto, se comprueba que cada dispositivo leído que compondrá el RAID no tiene un sistema de ficheros previo y, si todos estos requisitos se cumplen, el *script* creará de forma correcta el RAID.

3.3. LVM

El servicio *lvm* lee las dos primeras líneas, que contienen el nombre del grupo del RAID y los volúmenes físicos que lo compondrán. Tras esto, leerá, línea a línea, cada uno de los volúmenes lógicos a crear en el RAID.

Primero comprueba que la herramienta a usar, en este caso, *lvm2*, y todos sus componentes, estén instalados. Tras esto, verifica la correcta lectura de las dos primeras líneas, comprueba que cada volumen físico exista, crea el grupo y añade los volúmenes a él.

Si no han surgido errores, el *script* leerá cada uno de los volúmenes lógicos, comprobando que no se exceda el tamaño máximo ni la capacidad de almacenamiento.

3.4. NIS

3.4.1. Servidor

3.4.2. Cliente

3.5. NFS

3.5.1. Servidor

3.5.2. Cliente

3.6. Backup

Para la implementación del servicio *backup*, se ha elegido la herramienta *rsync*, principalmente por su sencillez. Como *rsync* no requiere de servidor dedicado, lo único que tendría que hacer el servicio *backup* en su parte de servidor es comprobar que exista y esté vacío el directorio que se va a usar para almacenar los backups.

3.6.1. Servidor

3.6.2. Cliente

En el lado del cliente del servicio *backup*, el *script* comprobará que la herramienta *rsync* está instalada en la máquina host, y no si, lo instala. Luego, comprueba que todos los campos se hayan leído correctamente y que el directorio del que hacer backup, y en el que se van a guardar los backups (cada uno en su respectivo host) existan. Por último, comprueba que la frecuencia de los backups dada es mayor o igual a 1 (e.g. realizar el backup cada 1 hora).

Si se cumplen todos los anteriores requisitos/comprobaciones, el *script* creará el servicio *backup* en el host dado de forma satisfactoria.

4. Códigos de error

Para facilitar el desarrollo y la implementación de todas las comprobaciones, errores y mensajes, se ha asignado a cada servicio un rango de 10 posibles códigos de error, los cuales están especificados a continuación:

4.1. Comúnes a todos los servicios

Rango de códigos de error: 1 - 9 y 255

- **1**: No se ha proporcionado fichero de configuración a `configurar_cluster.sh`
- **2**: El fichero de configuración no existe o es un directorio
- **3**: Error en el formato del fichero de configuración
- **4**: Un fichero de configuración de un servicio no existe
- **5**: Servicio desconocido en el fichero de configuración
- **6**: Error en el formato del fichero de configuración de un servicio
- **255**: Error del servicio SSH

4.2. Mount

Rango de códigos de error: 10 - 19

- **10**: El dispositivo a montar no existe
- **11**: El punto de montaje no es un directorio vacío
- **12**: Error inesperado durante el montaje
- **13**: Error inesperado al crear el directorio

4.3. Raid

Rango de códigos de error: 20 - 29

- **20**: Error inesperado al configurar el RAID
- **21**: El nivel de RAID no es válido (no soportado)

4.4. LVM

Rango de códigos de error: 30 - 39.

- **30**: Uno de los dispositivos especificados no existe
- **31**: Error inesperado al inicializar los volúmenes físicos
- **32**: Error inesperado al crear el grupo de volúmenes físicos
- **33**: Se ha excedido el tamaño del grupo al crear los volúmenes lógicos
- **34**: Error inesperado al crear uno de los volúmenes lógicos

4.5. NIS

4.5.1. Servidor

Rango de códigos de error: 40 - 49.

- **40**: Error al configurar el rol del servidor NIS
- **41**: Error al configurar el rango de direcciones IP que tienen acceso al servidor NIS
- **42**: Error al configurar la variable MERGE_PASSWD en el fichero /var/yp/Makefile
- **43**: Error al configurar la variable MERGE_GROUP en el fichero /var/yp/Makefile
- **44**: Error al añadir el servicio NIS al fichero /etc/hosts
- **45**: Error al configurar la base de datos del servicio NIS
- **46**: Error al reiniciar el servicio NIS

4.5.2. Cliente

Rango de códigos de error: 50 - 59.

4.6. NFS

4.6.1. Servidor

Rango de códigos de error: 60 - 69.

4.6.2. Cliente

Rango de códigos de error: 70 - 79.

4.7. Backup

4.7.1. Servidor

Rango de códigos de error: 80 - 89.

4.7.2. Cliente

Rango de códigos de error: 90 - 99.

5. Testing

Para testear de forma rápida y sencilla cada uno de los servicio y probar que funcionan de la forma esperada, se ha creado el fichero `run_tests.sh`, el cual contiene un array de los nombres de cada test, un array con los resultados esperados al ejecutar cada test, y un bucle que ejecuta cada uno de ellos e informa por la salida estándar de si el test pasa o no.

Los tests se han emplazado en el directorio `tests/` con cada servicio en su propio directorio, en el que aparecen el fichero de configuración general y el fichero de configuración del servicio.

De esta forma, escribiendo tests para comprobar la detección de errores de cada *script* y la correcta ejecución del servicio si no se encuentran errores, podemos probar cada uno de los servicios de forma instantánea.

A continuación se muestra la estructura del directorio `tests/`