UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS INFORMÁTICOS



Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

Administración de Sistemas Informáticos

**Memoria**

SEGUNDO CURSO CUARTO SEMESTRE

*Autores:*

ILUT, CIPRIAN  
N.º de matrícula: 160348

MATEOS MARTÍN, CARLOS  
N.º de matrícula: 160185

DEL RÍO GARCÍA, JAVIER  
N.º de matrícula:160451

Contenido

Tabla de contenido

[Configurar\_cluster 2](#_Toc27881462)

[LVM 3](#_Toc27881463)

[MOUNT 4](#_Toc27881464)

[NFS 5](#_Toc27881465)

[Cliente 5](#_Toc27881466)

[Servidor 5](#_Toc27881467)

[NIS 7](#_Toc27881468)

[Cliente 7](#_Toc27881469)

[Servidor 7](#_Toc27881470)

[RAID 8](#_Toc27881471)

[BACKUP 9](#_Toc27881472)

[ANEXOS 10](#_Toc27881473)

[Códigos de error: 10](#_Toc27881474)

# Configurar\_cluster

Configurar clouster es el script principal. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que el número de argumentos esperados (1) sea el recibido, si no lo es se responde con error código 200.
* Por cada línea que contenga el fichero de contaminación:
  + Se comprueba el formato, si es erróneo se responde con exit status 201.
  + Si el formato es correcto se copia el script correspondiente al servicio solicitado y su fichero de configuración a la maquina indicada en el primer parámetro.
  + A continuación, se ejecuta el script en la maquina remota y se eliminan los ficheros anteriormente transferidos para no dejar archivos innecesarios en la máquina remota.

\*Si cualquiera de los comandos que termina de forma errónea hace terminar la ejecución del script. Sucederá de forma análoga en los scripts asociados a cada servicio.

# LVM

Lvm.sh es el script que se encarga de la configuración de volúmenes lógicos. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio de lvm.
* Se comprueba que el número de líneas es al menos 3.
* Se comprueba que la primera línea contiene una cadena de caracteres, el nombre del grupo de volúmenes.
* Se obtiene la segunda línea y por cada uno de los dispositivos que aparecen en ella:
  + Se obtiene el nombre del dispositivo.
  + Se comprueba si el dispositivo está en el sistema.
  + Se comprueba si el dispositivo ya perteneces a un volumen lógico.
  + Se comprueba que tenga formato y si no lo tiene se formatea.
  + Se comprueba que esté montado y si no lo está se monta.
* Se inicializan los volúmenes físicos.
* Se comprueba la presencia de lvm2 y si no está, se instala.
* Se crea el volumen lógico solicitado previa comprobación de que el tamaño del lvm no excede al espacio disponible.

# MOUNT

Mount.sh es el script que se encarga del montaje de dispositivos en un punto del árbol de directorios determinado. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio de mount. Si no es así, el programa terminará con un exit status 110.
* Se comprueba que el número de líneas del fichero de perfil de servicio es dos. Si no es así, el programa termina con un exit status 111.
* Se comprueba que el dispositivo está en el sistema. Si no es así, el programa termina con un exit status 112.
* Se comprueba que el dispositivo tiene un sistema de ficheros. Si no, se formatea.
* Se comprueba si el dispositivo ya está montado en el sistema. Si lo está. el programa termina con un exit status 113.
* Si no lo está se monta en el punto de montaje especificado, que será creado previamente si no estuviera en el sistema con anterioridad. Además se añade una nueva línea en el fichero /etc/fstab para que se monte automáticamente al arrancar el sistema.

# NFS

## Cliente

Nfs\_client.sh es el script que se encarga de la creación del cliente NFS. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio del cliente NIS. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el fichero pasado como parámetro existe. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el número de líneas sea al menos 2.
* Se comprueba que esté presente en el equipo el paquete nfs-common y si no, se instala.
* Por cada una de las líneas del fichero:
  + Se comprueba que tenga 3 argumentos. Si no es así, el programa terminará con un exit status #.
  + Se comprueba si el servidor es alcanzable. Si no es así, el programa terminará con un exit status #.
  + Se comprueba si el directorio solicitado se encuentra en la lista de exportación del servidor. Si no es así, el programa terminará con un exit status #.
  + Se comprueba que exista el punto de montaje y si no, se crea. A continuación, se monta y se añade una nueva entrada a la tabla del archivo /etc/fstab para que se realice el montaje al inicio del sistema.

## Servidor

Nfs\_server.sh es el script que se encarga de la creación del servidor NFS. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio del servidor NIS. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el fichero pasado como parámetro existe. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el número de líneas del fichero de perfil de servicio es una o más. Si no es así, el programa termina con un exit status #.
* Se comprueba que esté presente en el equipo el paquete nfs-kernel-server y si no, se instala.
* Por cada una de las líneas del fichero:
  + Se comprueba que el directorio indicado en la línea es de verdad un directorio. Si no es así, el programa termina con un exit status #. Si es así se añade al fichero /etc/exports para incluirlo en la lista de exportación del servidor.

# NIS

## Cliente

Nis\_client.sh es el script que se encarga de la creación del cliente NIS. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio del cliente NIS. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el número de líneas del fichero de perfil de servicio es dos. Si no es así, el programa termina con un exit status #.
* Se instala el programa NIS y se arranca el cliente definiendo dominio NIS y servidor al que queremos conectarnos.

## Servidor

Nis\_server.sh es el script que se encarga de la creación del servidor NIS. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio del servidor NIS. Si no es así, el programa terminará con un exit status #.
* Se comprueba que el número de líneas del fichero de perfil de servicio es una. Si no es así, el programa termina con un exit status #.
* Se instala el paquete nis y se arranca el servidor solicitado con los parámetros indicados.

# RAID

Raid.sh es el script que se encarga de la creación del RAID solicitado. Para conseguir que realice la función esperada se procede de la siguiente manera:

* Se comprueba que se recibe como parámetro un único argumento, el fichero de perfil de servicio de raid. Si no es así, el programa terminará con un exit status #. También se comprueba que exista, si no es así, el programa terminará con un exit status #.
* Se comprueba que el número de líneas del fichero de perfil de servicio es tres. Si no es así, el programa termina con un exit status #.
* Se comprueba que los dispositivos que van a formar parte del RAID no pertenezcan a un sistema de ficheros. Si no es así, el programa termina con un exit status #.
* Se comprueba la presencia de mdadm y si no está, se instala.
* Se crea el RAID con el nivel, el nombre y los dispositivos especificados. El nivel se habrá comprobado previamente que está entre 0 y 5.

# BACKUP

# ANEXOS

## Códigos de error:

Codigos de error:

------------------------------------------------------------------------------------

10x: CODIGO DE ERROR DE SCRIPT PRINCIPAL:

100: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA A SCRIPT PRINCIPAL

------------------------------------------------------------------------------------

11x: CODIGO DE ERROR DE SCRIPT AUX MOUNT:

110: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA A SCRIPT AUXILIAR MOUNT

------------------------------------------------------------------------------------

12x: CODIGO DE ERROR DE SCRIPT AUX RAID:

120: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

------------------------------------------------------------------------------------

13x: CODIGO DE ERROR DE SCRIPT AUX LVM:

130: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

------------------------------------------------------------------------------------

14x: CODIGO DE ERROR DE SCRIPT AUX NIS\_SERVER:

140: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

--------------------------------------------------------------------------------

15x: CODIGO DE ERROR DE SCRIPT AUX NIS\_CLIENT:

150: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

------------------------------------------------------------------------------------

16x: CODIGO DE ERROR DE SCRIPT AUX NFS\_SERVER:

160: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

--------------------------------------------------------------------------------

17x: CODIGO DE ERROR DE SCRIPT AUX NFS\_CLIENT:

170: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

------------------------------------------------------------------------------------

18x: CODIGO DE ERROR DE SCRIPT AUX BACKUP\_SERVER:

180: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA

--------------------------------------------------------------------------------

19x: CODIGO DE ERROR DE SCRIPT AUX BACKUP\_CLIENTE:

190: NUMERO DE PARAMETROS INCORRECTO EN LLAMADA