8. TAREAS PERIÓDICAS EN EL SISTEMA

8.1.- Introducción

A menudo un administrador necesita dejar programado la ejecución de distintos comandos en el sistema, bien sean de forma puntual o periódica; estos comandos se agrupan en tareas o trabajos.

Para las tareas puntuales tenemos el comando *at* y para las tareas que se tienen que repetir cada cierto tiempo, usaremos el comando *cron*.

8.2.- Tareas at

Por defecto, en una instalación estándar de CentOS, el servicio *atd* está instalado. Este servicio permite programar tareas puntuales en el futuro, por defecto, a todos los usuarios del sistema. Si es una instalación muy básica habrá que instalar el paquete *at*, arrancar y activar el servicio *atd*.

El servicio *atd* proporciona 26 colas, de la a-z, donde almacenar las tareas, siendo tareas menos prioritarias las que están en las colas con letra más cercana al final del abecedario.

Para interactuar con el servicio *atd*, tenemos los comandos *at*,*atrm* y *atq*.

8.2.1.- Programar tareas at

Para programar una tarea, usaremos el comando *at* con la sintaxis:

at -q <cola> <tiempo> y en las siguientes líneas introduciremos los comandos a ejecutar, uno por línea, hasta que pulsemos CRTL+D.

donde:

<*cola*>: es una letra a-z, donde la cola *a* es una cola más prioritaria que la *z*. Se puede omitir la opción -*q* y entonces se asignará a la cola *a*.

<*tiempo*>: es una especificación de tiempo, como por ejemplo *now + 5 days, tomorrow, ...*

Otras opciones interesantes son:

- -*m* <*mail*>*:* para enviar al e-mail dado, la salida los comandos ejecutados en la tarea una vez realizados. Envía el e-mail aunque no haya salida de los comandos.
- -*f* <*archivo*>: para proporcionar un archivo donde estén los comandos a ejecutar.

Las tareas programadas se almacenan en el directorio */var/spool/at*.

Se pueden declarar variables de entorno antes de escribir los comandos de una tarea, con el formato <*variable*>=<*valor*>. Dichas variables afectarán a los comandos subsiguientes. Por ejemplo, suele ser habitual declarar la variable *MAILTO*=<*direccion_mail*>, para que cualquier salida que produzca la tarea programada, la envíe a dicho correo.

→ Especificar el tiempo: hay múltiples combinaciones:

- *at <hh:mm>*: si la hora ya ha pasado, se hará en esa hora del día siguiente.
- *at <hh:mm> <mes dia>*: el día y mes próximo, a esa hora.
- *at <hh:mm> <mes dia año>*: el día, mes y año dado, a la hora dada.
- *at now +5min, at noon +4 days, at teatime tomorrow, at midnight +5 weeks,….*
  siendo *teatime* las 4 de la tarde, *noon* las 12 de la mañana y *midnight* las 12 de la noche.

Se puede encontrar documentación de las definiciones de tiempo que se pueden usar en */usr/share/doc/at-*/timespec.*

8.2.2.-Ver las tareas at programadas

Utilizaremos el comando *at -l* o *atq*. La salida es una lista de las tareas programadas donde cada línea tiene varios campos: número de tarea, fecha (día de la semana, mes, día, hora y año) para la que está programada, la cola en la que está la tarea y el usuario que programó la tarea.
Para ver los detalles de la tarea, usaremos el comando *at -c <número_tarea>*.
Un usuario normal, sólo verá sus tareas, el usuario *root* verá las de todos los usuarios.

8.2.3.- Eliminar tareas at programadas

Para eliminar una tarea ya programada, primero tenemos que saber cuál es su número de tarea y entonces utilizaremos *atrm <número_tarea>* o *at -r <número_tarea>* o *at -d <número_tarea>*.

8.2.4.- Restringir el uso de at

Se puede restringir la programación de tareas usando *at* con el uso de dos archivos que contienen un usuario por línea:

- */etc/at.deny:* listado de los usuarios que no pueden programar tareas con *at*. Por defecto este archivo existe y está vacío; permitiendo que todos los usuarios puedan utilizar *at*.
- */etc/at.allow*: listado de usuarios que tienen permitido usar *at*. Este archivo por defecto no existe.

El proceso para decidir si un usuario puede usar *at* o no es el siguiente:
Primero, se busca el archivo */etc/at.allow*:

- si existe: se mira si el usuario está en la lista, si lo está, podrá utilizar *at* y si no, no.
- si no existe: se busca el archivo */etc/at.deny* y se mira si el usuario está en la lista, si lo está, no se le permite usar *at* y si no está, si.
- Si el archivo */etc/at.deny* tampoco existe, sólo el usuario *root* podrá utilizar *at*.

8.3.- Tareas cron

El servicio *crond* permite en un sistema programar tareas periódicas tanto a usuarios regulares como a *root*. Normalmente, por muy mínima que sea una instalación, está ya en el sistema instalado el paquete *cronie*, el cual nos proporciona *crond*, y arrancado y activado el servicio.

Si un comando ejecutado por una tarea cron produce una salida que no se ha redirigido, ya sea la salida estándar o la de error estándar, el servicio *crond* intentará enviar dicha salida por e-mail al usuario que creo la tarea cron.

Los usuarios que tengan definida como shell */dev/null* no podrán utilizar *crontab*.

Y para aquello que tengan la shell */sbin/nologin*, hay una forma de que puedan programar tareas periódicas. Deben definir la variable *SHELL=/bin/bash* o *SHELL=/bin/sh* al principio del archivo y para poder ejecutar *crontab -e*, deben ejecutar:

```
su -l <usuario> -s /bin/bash -c “crontab -e”
```

8.3.1.- Programar tareas periódicas de usuarios regulares

Los usuarios regulares usarán el comando *crontab* para gestionar tareas periódicas con las opciones:

- *-l*: lista las tareas periódicas del usuario.
- *-r*: borra todas las tareas periódicas del usuario.
- *-e*: Edita las tareas periódicas y permite crear nuevas.

Proporcionando un archivo al comando *crontab*, se borran todas las tareas periódicas del usuario y se sustituyen por las que existen en el archivo.

Si se ejecuta sin opciones ni archivo, se introducen las tareas por la línea de comandos.

→ Formato de una tarea periódica

Es una línea con 5 campos que detallan cuándo se ejecutará la tarea y el último campo que detalla cual será el comando/s a ejecutar. Tiene el formato:

```
<minutos> <hora> <dia_mes> <mes> <dia_semana> <comando>
```

donde:

<*minutos*>: 0-59.

<*hora*>: 0-23.

<*dia_mes*>: 1-31 (ojo que no todos los meses tienen 31 días).

<*mes*>*:* 1-12. También se admite jan, feb, mar, ….

<*dia_semana*>: 0-7, siendo el 0 y el 7 el domingo, 1 lunes, …. También se admite *mon, tue, wed, thu, fri, sat, sun*.

<*comando*>: comando/s a ejecutarse con la shell /bin/sh, si se desea otra, hay que declarar una variable SHELL=<otra_shell>. Si se pone un carácter %, se entiende que es otra línea.

Hay que tener cuidado con los campos <*dia_mes*> y <*dia_semana*> ya que si tienen otros valor que no sea *, la tarea se ejecutará cuando uno de los dos campos ocurra. No son excluyentes.

Para los 5 campos que detallan cuándo se realiza la tarea, existen además las reglas de sintaxis:

- *: significa “siempre”.
- *n-m*: rango de números, ambos incluidos.
- *n,m*: para listas que pueden incluir rangos.
- */n*: para indicar un intervalo, p.e., si se pone en el campo <*hora*>, sera cada n-horas.

Al igual que en con AT, se pueden declarar variables de entorno antes de escribir las líneas del *crontab*, con el formato <*variable*>=<*valor*>. Dichas variables afectarán a las líneas subsiguientes.

NOTA: El usuario *root* puede gestionar tareas periódicas para otros usuarios usando: *crontab -u* <*usuario*> y a continuación el resto de opciones que se quieran poner.

8.3.2.- Programar tareas periódicas de sistema

Para realizar tareas periódicas de administración del sistema, no se utiliza el comando *crontab*, se configuran en una serie de archivos cuyas líneas tendrán el mismo formato que las líneas del comando crontab pero con un campo extra antes del comando/s a ejecutar; el campo del usuario que ejecutará dicha tarea periódica. Estos archivos son */etc/crontab* y los archivos del directorio */etc/cron.d.*

Además en estos directorios:

- */etc/cron.hourly*: ejecución cada hora.
- */etc/cron.daily*: ejecución diaria.
- */etc/cron.weekly*: ejecución semanal.
- */etc/cron.monthly*: ejecución mensual.

se almacenan unos scripts ejecutables de bash o de otros lenguajes.

8.3.4.- Restringir el uso de cron

Al igual que con AT, se puede restringir la creación de tareas periódicas con *cron* con el uso de dos archivos que contienen un usuario por línea:

- */etc/cron.deny:* listado de los usuarios que no pueden utilizar *crontab*. Por defecto este archivo existe y está vacío; permitiendo que todos los usuarios puedan utilizar *crontab*.

- */etc/cron.allow*: listado de usuarios que tienen permitido usar *crontab*. Este archivo por defecto no existe.

La lógica de funcionamiento de los archivos *allow* y *deny* es la misma que en AT. Si existe el archivo */etc/cron.allow* se mira se mira si el usuario está en la lista, si lo está, podrá utilizar *crontab* y si no, no. Si este archivo no existe, se busca el archivo */etc/cron.deny* y se mira si el usuario está en la lista, si lo está, no se le permite usar *crontab* y si no está, si.

En el caso de que no exista ninguno de los dos archivos, sólo el usuario *root* podrá utilizar *crontab*.

8.4.- Anacron

Anacron se encarga de ejecutar las tareas pendientes que no se hayan procesado con el servicio crond por estar el sistema apagado.

Anacron lee la lista de trabajos del archivo */etc/anacrontab*. En el pasado este archivo se gestionaba con el demonio *anacron* pero en CentOS 7, también lo gestiona el servicio *crond*.

El formato del archivo es diferente del de *cron*. Tiene cuatro campos por línea:

<periodo_días> <espera_minutos> <id_tarea> <comando>

Para cada trabajo, Anacron comprueba si el trabajo se ha iniciado en los últimos <periodo_días>. Si la última vez que se inició el trabajo fue hace más de esos días, se vuelve a lanzar de nuevo, tras esperar el número de minutos de <espera_minutos>. Tras ejecutar el comando, Anacron guarda la información de la ejecución del comando para saber cuándo tiene que lanzarlo de nuevo.

Si los comandos a ejecutar generan salida o errores, se envían por correo al usuario con el que se ejecuta *anacron*, o a la dirección especificado por la variable *MAILTO* de */etc/anacrontab.*

La configuración por defecto que hay en el archivo */etc/anacrontab* hace que los scripts de */etc/cron.daily*, */etc/cron.weekly* y */etc/cron.monthly* se ejecuten con el comando *run-parts*.

8.5.- Caso práctico

En **server1**, como usuario *admin*, se va a programar una tarea el día 1 de enero que cree un archivo llamado README en su home que contenga la frase *"Feliz año"* y otra tarea en día 15 de enero que crea un archivo en /tmp llamado *prueba.txt* con el contenido *"Prueba"*. Una vez creada, se eliminará esta última tarea.

Los usuarios del grupo de *redes* no pueden programar tareas con *at.*

En **central** se va a realizar mensualmente, un backup de la base de datos de *employees* en */backup/<aaaa-mm>backup-employees.dump* donde *<aaaa-mm>* es el año y el mes del backup.

Además, en **central** se va a restringir el uso de cron a todos los usuarios del sistema.

RESOLUCIÓN

- Como en **server1** se realizó una instalación básica, no está instalado el paquete *at*; lo instalamos y arrancamos el servicio *atd*:

```
[root@server1 ~]# yum install -y at
[root@server1 ~]# systemctl start atd && systemctl enable atd
```

- Programo las tareas del día 1 y 15 como usuario *admin*:

```
[root@server1 ~]# su - admin
[admin@server1 ~]$ echo "echo 'Feliz Año' > ~/README" | at 8:00am january 1
job 1 at Mon Jan  1 08:00:00 2018
[admin@server1 ~]$ echo "touch /tmp/prueba" | at january 15
job 2 at Mon Jan 15 17:49:00 2018
```

- Miro las tareas programadas y borro la del día 15:

```
[admin@server1 ~]$ atq
1       Mon Jan  1 08:00:00 2018 a admin
2       Mon Jan 15 17:49:00 2018 a admin
[admin@server1 ~]$ at -c 1; at -c 2
[admin@server1 ~]$ atrm 2
[admin@server1 ~]$ atq
1       Mon Jan  1 08:00:00 2018 a admin
```

- Restringimos a los usuarios del grupo de *redes* el uso de *at*:

```
[root@server1 ~]# getent group redes | cut -d: -f4 | tr ',' '\n' >> /etc/at.deny
ó
[root@server1 ~]# for usu in $(groupmems -g redes -l);do echo $usu>> /etc/at.deny;done
[root@server1 ~]# cat /etc/at.deny
hlamarr
tberners
rkahn
```

- Verificamos que con uno de estos usuarios no se puede hacer uso de *at*:

```
[hlamarr@server1 ~]$ atq
You do not have permission to use atq.
```

- En **central**, vamos a realizar un *backup* mensual de la base de datos de *employees* con un archivo en el directorio */etc/cron.d*:

```
[root@central ~]# echo "01 0 1 * * root /usr/bin/mysqldump -u root -pmariadb employees
        > /backup/$(date +%y-%m)-backup-employees.dump" > /etc/cron.d/backupBBDD
```

- También se puede realizar el *backup* mensual, creando un script bash que realiza el *backup* en el directorio */etc/cron.monthly*:

```
[root@central ~]# cat > /etc/cron.monthly/backupBBDD.sh <<EOF
#!/bin/bash
/usr/bin/mysqldump -u root -pmariadb employees > /backup/$(date +%y-%m)-backup-employees.dump
logger "Backup de la BBDD employees realizado $date"
EOF
chmod +x /etc/cron.monthly/backupBBDD.sh
```

- En **central**, no permitimos que ningún usuario ejecute *crontab*, para esto, basta eliminar */etc/cron.deny* y */etc/cron.allow* si existiese.

```
[root@central ~]#  ls /etc/cron.*
/etc/cron.deny
/etc/cron.d:
0hourly
/etc/cron.daily:
logrotate  man-db.cron
/etc/cron.hourly:
0anacron
/etc/cron.monthly:
/etc/cron.weekly:
[root@central ~]# rm /etc/cron.deny
rm: remove regular empty file ‘/etc/cron.deny’? y
```

- Compruebamos que con el usuario *admin* no se puede usar *crontab*:

```
[root@central ~]# su - admin
[admin@central ~]$ crontab -l
You (admin) are not allowed to use this program (crontab)
See crontab(1) for more information
```