

El presente artículo es una reedición de un artículo y un post publicado en 2008 por cristiansan. Debido a la naturaleza de XenServer, la experiencia en entornos Linux es aplicable a XenServer ampliando sus funcionalidades con características expresas de sistemas Linux. En este artículo, vamos a explicar de manera sencilla el funcionamiento del CRON de Linux y cómo podemos aplicarlo sobre nuestros XenServer para automatizar tareas.

El servicio cron nos permite definir tareas automatizadas para que nuestro sistema Linux las ejecute sin necesidad de que tecleemos los comandos continuamente. c

El Cron nos permite establecer la hora, la fecha y la periodicidad de la ejecución del comando, permitiendo automatizar tareas repetitivas y facilitando nuestra labor diaria.

Cron puede ser interpretado como un servicio de Linux que se carga durante el proceso de arranque de nuestro sistema.

Para ejecutar tareas, cron usa un archivo conocido como crontab. El archivo crontab esta, generalmente, localizado en el directorio /etc. en /var/spool/cron/ localizamos también ficheros de cron que hacen relación a un usuario en concreto. (ver directorio)

Como usar el Crontab

El primer paso es abrir crontab. Para ello, podemos utilizar cualquier editor de textos, como vi, joe o emacs. También podemos editar crontab tecleando “crontab -e” utilizando de este modo el editor vi para su edición. (ello editara por defecto el crontab del usuario que invoca el comando)

El crontab utiliza el siguiente formato.

[minutos] [horas] [días] [mes] [días de la semana] [usuario] [comando]

minutos: 0-59

horas: 0-23

días: 1-31

mes: 1-12

Días de la semana: 0-6 (de domingo a sábado)

Usuario: usuario con el que se lanza el comando

Comando: comando a lanzar

El orden de los valores debe ser mantenido. En lugar de un valor, podemos usar también el carácter “*” el cual será utilizado como comodín, es decir, si ponemos en día de la semana un *, significará que se lanzará cada día, si ponemos en el de los minutos, se lanzará el comando a cada minuto.

También es posible especificar rangos con “-“, si ponemos en días de la semana, 0-3, el comando se ejecutará, domingo, lunes, martes, miércoles. Igualmente podemos indicar días o horas separadas, utilizando la ““, así pues usando 0,3 el comando solo será lanzado los días domingo y miércoles.

Ejemplo: Teniendo en cuenta que disponemos de un script en nuestro directorio personal llamado borrado.sh con sus respectivos permisos)
Editamos el crontab y añadimos:

```
$ crontab -e
```

```
# comentario sobre la tarea
```

```
30 22 1,15 * * cristiansan /home/cristiansan/borrado.sh
```

Esto ejecutaria el script borrado.sh a las 22:30 de cada día 1 y 15 de cada mes, sin tener en cuenta el día de la semana.

Cabe tener en cuenta además los ficheros `/etc/cron.deny` y `/etc/cron.allow`. `cron.deny` nos permite definir usuarios los cuales no tendrán acceso a `cron`. Su formato es muy sencillo, un usuario por línea. El espacio en blanco no está permitido, tampoco es necesario reiniciar el demonio `cron` tras el cambio realizado, estos ficheros son leídos por `cron` cuando se intenta acceder a él.

`root` siempre debe disponer de acceso a `cron`. Si el fichero `cron.allow` existe, solo los nombres que hay en dicho fichero tendrán acceso a `cron`, y `cron.deny` será ignorado.

Si `cron.allow` no existe, `cron` consultará `cron.deny` y denegará el acceso a los usuarios que aparezcan en este.

Aplicando CRON en XenServer

`Crontab` nos ofrece el poder para automatizar muchas tareas dentro de nuestro XenServer tales como borrado de logs, update programados, reinicios y todo aquello susceptible de realizar en línea de comandos (CLI).

Un ejemplo práctico podría ser el programar un pequeño script de pausado o suspendido de VM en ejecución y uno de arranque. Utilizando CRON podríamos establecer que a cierta hora, en la cual no son necesarias ciertas VM, suspender las mismas y configurar el mismo `cron` para que a cierta hora (principio de la jornada) las arranque.

Los siguientes son dos scripts muy sencillos (podrían mejorarse mucho) que pausan el estado de la máquina arrancadas y arrancan las máquinas pausadas anteriormente respectivamente.

Escrito por cristiansan
Jueves, 20 de Mayo de 2010 07:40 -

Los scripts son llamados pause.sh y unpause.sh. Una vez editados podemos añadir las siguientes líneas en Crontab:

```
00 20 * * [1-5] ctxdom /home/ctxdom/pause.sh
00 07 * * [1-5] ctxdom /hom/ctxdom/unpause.sh
```

Ello ejecutara el script de pausado a las 20:00h de Lunes a Viernes y arrancara las máquinas igualmente de Lunes a viernes a las 7 AM.

A continuación dejo los dos scripts realizados a modo orientativo:

Pause.sh (Script de pausado de máquinas en ejecución)

```
#!/bin/bash
echo "Listando VMs en ejecucion..."
sleep 2
xe vm-list params=uuid power-state=running | grep uuid | cut -c 17-63 > /tmp/listado.vm
declare -a VM
exec</tmp/listado.vm
let count=0
while read line; do
VM[$count]="$line"
count=$((count+1))
done
COUNTVM=`wc -l /tmp/listado.vm | cut -c1-2`
VMPI=0
echo "VM totales en ejecucion...: $COUNTVM"
sleep 3
while [ $VMPI -lt $COUNTVM ]
do
    xe vm-pause uuid=${VM[$VMPI]}
    echo "Pausando maquina: $VMPI"
    sleep 2
    2>/dev/null
    VMPI=$((VMPI+1))
done
echo "Finalizado"
```

unpause.sh (Script de arrancado de máquinas pausadas)

```
#!/bin/bash
echo "Listando VMs en ejecucion..."
sleep 2
declare -a VM
exec</tmp/listado.vm
let count=0
while read line; do
    VM[$count]="$line"
    count=$((count+1))
done

COUNTVM=`wc -l /tmp/listado.vm | cut -c1-2`
VMPI=0
echo "VM totales pausadas...: $COUNTVM-1"
sleep 3
while [ $VMPI -lt $COUNTVM ]
do
    xe vm-unpause uuid=${VM[$VMPI]}
    echo "Arrancando maquina: $VMPI"
    sleep 2
    2>/dev/null
    VMPI=$((VMPI+1))
done
echo "Finalizado"
```