Dernière mise-à-jour : 2020/01/30 03:27

107.2 - Automatiser les tâches de l'administration avec la planification des jobs (4/60)

cron

> **Important** : Le service crond présume que la machine est allumée en permanence.

Le service crond est normalement lancé au démarrage de la machine. Ce service est chargé de faire exécuter des tâches (commandes et scripts) à des moments précis. Le service crond présume que la machine est allumée en permanence.

Le service crond lit toutes les minutes le fichier **/etc/crontab**, les crontabs se trouvant dans **/etc/cron.d** ainsi que les fichiers propres à chaque utilisateur.

Si une commande produit une sortie, celle-ci est dirigée vers la messagerie.

L'utilisation de crond est réservé à root. Cependant, vous pouvez établir une liste d'utilisateurs qui ont la permission d'utiliser crond en créant un fichier nommé **cron.allow** dans **/etc**. A l'inverse, un fichier **cron.deny** peut contenir une liste d'utilisateurs qui n'ont pas la permission d'utiliser crond.

Sous RHEL/CentOS, les crontabs des utilisateurs ont comme nom, le nom de l'utilisateur qui les crée et se trouvent dans le répertoire **/var/spool/cron/**. Par exemple, le crontab propre à l'utilisateur trainee est le fichier **/var/spool/cron/trainee**. Le service crond exécute des tâches en rajoutant une ligne dans son fichier de journalisation **/var/log/cron**.

Le Fichier /etc/crontab

Sous RHEL/CentOS le fichier /etc/crontab prend la forme suivante :

```
[root@centos7 ~]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

Dans le fichier /etc/crontab, on peut constater la définition de variables dont SHELL, MAILTO et PATH. Ces variables sont différentes des variables système. Notez que sous RHEL/CentOS, le fichier **/etc/crontab** ne contient aucune tâche à exécuter.

Les Champs Temporels

Chaque ligne dans un fichier /etc/crontab contient 5 champs temporels qui décrivent le périodicité de l'exécution de la tâche concernée.

Les 5 champs sont :

Minutes	Heures	Jour du mois	Mois de l'année	Jour de la sem.
(0-59)	(0-23)	(1-31)	(1-12)	(0-6)*

* le 0 correspond à dimanche.

Les champs temporels peuvent contenir des valeurs différentes :

Exemple	Description
Une valeur absolue telle 10	Dans le champs minutes = 10 minutes après l'heure
Une série de valeurs telle 2,6,8	Dans le champs mois = février, juin et août
Une intervalle telle 1-5	Dans le champs Jour de la Semaine = du lundi au vendredi
Le joker *	Dans le champs minutes = toutes les minutes
Une périodicité telle 0-23/2	Dans le champs heures = toutes les deux heures

Les crontabs Utilisateurs

Chaque utilisateur peut avoir sa propre version d'un fichier crontab. Pour vérifier s'il existe une version de crontab pour lui, l'utilisateur lancera la commande suivante :

```
[trainee@centos7 ~]$ crontab -l
no crontab for trainee
```

Afin de créer votre fichier crontab, lancez la commande crontab avec l'option -e en tant que votre nom d'utilisateur et non en tant que root :

```
$ crontab -e [Entrée]
```

Cette commande lance l'interface de l'éditeur **vi**. Tapez ensuite le texte suivant :

```
* * * * * /bin/pwd > pwd.txt
```

Sauvegardez et quittez vi.

Vous obtiendrez un résultat similaire à celui-ci :

```
[trainee@centos7 ~]$ crontab -e
no crontab for trainee - using an empty one
crontab: installing new crontab
```

Important : Ce fichier crontab va être exécuté toutes les minutes. La commande va inscrire le répertoire courant dans le fichier pwd.txt.

Le fichier crontab que vous venez de créer sous RHEL/CentOS, au nom de votre utilisateur, a été écrit sur le disque dans le répertoire **/var/spool/cron/**.

Malgré le fait que vous venez de l'éditer, vous ne pouvez pas visualiser votre fichier en tant que simple utilisateur avec la commande cat. Seule la commande crontab -e vous permettra de consulter ou modifier votre fichier crontab.

Passez donc en mode super-utilisateur et visualisez le fichier en utilisant la commande cat :

```
[trainee@centos7 ~]$ su -
Password: fenestros
Last login: Sat Oct 24 16:22:56 CEST 2015 on pts/0
[root@centos7 ~]# cat /var/spool/cron/trainee
* * * * * /bin/pwd > pwd.txt
```

Afin d'accorder ou non le droit d'éditer son propre fichier crontab, root a la possibilité d'inscrire les noms d'utilisateurs dans un des deux fichiers :

- **/etc/cron.allow** pour accorder le droit d'éditer son propre fichier crontab
- **/etc/cron.deny** pour interdire le droit d'éditer son propre fichier crontab

Par contre, si root inscrit le nom d'un utilisateur dans /etc/cron.deny et cet utilisateur a déjà planifié des tâches avec son propre fichier crontab, les tâches seront exécutées. Pour détruire les tâches planifiées de l'utilisateur, il faut détruire le fichier crontab de l'utilisateur.

anacron

L'inconvénient avec crond est qu'afin que celui-ci puisse exécuter des tâches automatiquement, la machine doit rester allumée en permanence.

Il existe un alternatif très efficace pour des machines qui ne sont pas constamment allumée. Cette solution est le logiciel Anacron. Sous RHEL/CentOS,

l'utilisation d'anacron **remplace** l'utilisation de cron pour les tâches système.

Ouvrez donc le fichier /etc/anacrontab :

```
[root@centos7 ~]# cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1   5    cron.daily        nice run-parts /etc/cron.daily
7   25    cron.weekly        nice run-parts /etc/cron.weekly
@monthly 45 cron.monthly        nice run-parts /etc/cron.monthly
```

Vous noterez que ce fichier fait référence aux répertoires cron.daily, cron.weekly et cron.monthly.

Les deux premiers champs sont des champs temporels. Dans le cas d'anacron ceux-ci correspondent à:

Période	**Délai**
La périodicité en jours	Le délai en minutes, après le démarrage d'anacron et avant l'exécution des tâches dans le répertoire concerné

Anacron mémorise dans les fichiers qui se trouvent dans **/var/spool/anacron/** la date à laquelle il a exécuté les commandes qui se trouvent dans anacrontab pour la dernière fois. Ces fichiers sont appelés cron.daily, cron.weekly et cron.monthly.

```
[root@centos7 ~]# ls /var/spool/anacron/
cron.daily  cron.monthly  cron.weekly
```

```
[root@centos7 ~]# cat /var/spool/anacron/cron.daily
20150604
[root@centos7 ~]# cat /var/spool/anacron/cron.weekly
20150601
[root@centos7 ~]# cat /var/spool/anacron/cron.monthly
20150601
```

Sous RHEL/CentOS, anacron consulte le fichier **/var/spool/anacron/cron.daily** chaque heure grâce à l'exécution du script **/etc/cron.hourly/0anacron** et, en fonction de la date inscrite dans le fichier et la date du jour, décide si oui ou non s'il doit exécuter la ligne de commande.

Voici le script **/etc/cron.hourly/0anacron** :

```
[root@centos7 ~]# cat /etc/cron.hourly/0anacron
#!/bin/sh
# Check whether 0anacron was run today already
if test -r /var/spool/anacron/cron.daily; then
    day=`cat /var/spool/anacron/cron.daily`
fi
if [ `date +%Y%m%d` = "$day" ]; then
    exit 0;
fi

# Do not run jobs when on battery power
if test -x /usr/bin/on_ac_power; then
    /usr/bin/on_ac_power >/dev/null 2>&1
    if test $? -eq 1; then
    exit 0
    fi
fi
/usr/sbin/anacron -s
```

Ce script a pour but d'exécuter la commande **/usr/sbin/anacron -s**. L'option **-s** indique à anacron d'attendre la fin d'exécution d'un job avant d'exécuter le suivant.

at

Tout comme avec la commande cron, root a la possibilité d'accorder ou d'interdire aux utilisateurs le droit d'exécuter des tâches avec at en utilisant les deux fichiers suivants :

- **/etc/at.allow**
- **/etc/at.deny**

Si le fichier at.allow existe, seuls les utilisateurs dans ce fichier pourront exécuter at.

En tant que root, mettez en place maintenant deux tâches pour le 31/12/2016 à 13h00 et 14h00 respectivement :

```
[root@centos7 ~]# at 13:00 12/31/2016
at> pwd > /tmp/test13.atd
at> [^D]
at> <EOT>
job 1 at Sat Dec 31 13:00:00 2016

[root@centos7 ~]# at 14:00 12/31/2016
at> free > /tmp/test14.atd
at> [^D]
at> <EOT>
job 2 at Sat Dec 31 14:00:00 2016
```

Sous RHEL/CentOS, les fichiers concernant ces deux tâches sont stockés dans le répertoire **/var/spool/at/** :

```
[root@centos7 ~]# ls /var/spool/at
a0000101793190  a00002017931cc  spool
```

A l'examen du deuxième fichier, vous constaterez un résultat similaire à celui-ci :

```
[root@centos7 ~]# cat /var/spool/at/a0000101793190
#!/bin/sh
```

```
# atrun uid=0 gid=0
# mail trainee 0
umask 22
XDG_VTNR=1; export XDG_VTNR
XDG_SESSION_ID=1; export XDG_SESSION_ID
HOSTNAME=centos7.fenestros.loc; export HOSTNAME
SHELL=/bin/bash; export SHELL
HISTSIZE=1000; export HISTSIZE
USER=root; export USER
LS_COLORS=rs=0:di=38\;5\;27:ln=38\;5\;51:mh=44\;38\;5\;15:pi=40\;38\;5\;11:so=38\;5\;13:do=38\;5\;5:bd=48\;5\;232
\;38\;5\;11:cd=48\;5\;232\;38\;5\;3:or=48\;5\;232\;38\;5\;9:mi=05\;48\;5\;232\;38\;5\;15:su=48\;5\;196\;38\;5\;15
:sg=48\;5\;11\;38\;5\;16:ca=48\;5\;196\;38\;5\;226:tw=48\;5\;10\;38\;5\;16:ow=48\;5\;10\;38\;5\;21:st=48\;5\;21\;
38\;5\;15:ex=38\;5\;34:\*.tar=38\;5\;9:\*.tgz=38\;5\;9:\*.arc=38\;5\;9:\*.arj=38\;5\;9:\*.taz=38\;5\;9:\*.lha=38\
;5\;9:\*.lz4=38\;5\;9:\*.lzh=38\;5\;9:\*.lzma=38\;5\;9:\*.tlz=38\;5\;9:\*.txz=38\;5\;9:\*.tzo=38\;5\;9:\*.t7z=38\
;5\;9:\*.zip=38\;5\;9:\*.z=38\;5\;9:\*.Z=38\;5\;9:\*.dz=38\;5\;9:\*.gz=38\;5\;9:\*.lrz=38\;5\;9:\*.lz=38\;5\;9:\*
.lzo=38\;5\;9:\*.xz=38\;5\;9:\*.bz2=38\;5\;9:\*.bz=38\;5\;9:\*.tbz=38\;5\;9:\*.tbz2=38\;5\;9:\*.tz=38\;5\;9:\*.de
b=38\;5\;9:\*.rpm=38\;5\;9:\*.jar=38\;5\;9:\*.war=38\;5\;9:\*.ear=38\;5\;9:\*.sar=38\;5\;9:\*.rar=38\;5\;9:\*.alz
=38\;5\;9:\*.ace=38\;5\;9:\*.zoo=38\;5\;9:\*.cpio=38\;5\;9:\*.7z=38\;5\;9:\*.rz=38\;5\;9:\*.cab=38\;5\;9:\*.jpg=3
8\;5\;13:\*.jpeg=38\;5\;13:\*.gif=38\;5\;13:\*.bmp=38\;5\;13:\*.pbm=38\;5\;13:\*.pgm=38\;5\;13:\*.ppm=38\;5\;13:\
*.tga=38\;5\;13:\*.xbm=38\;5\;13:\*.xpm=38\;5\;13:\*.tif=38\;5\;13:\*.tiff=38\;5\;13:\*.png=38\;5\;13:\*.svg=38\;
5\;13:\*.svgz=38\;5\;13:\*.mng=38\;5\;13:\*.pcx=38\;5\;13:\*.mov=38\;5\;13:\*.mpg=38\;5\;13:\*.mpeg=38\;5\;13:\*.
m2v=38\;5\;13:\*.mkv=38\;5\;13:\*.webm=38\;5\;13:\*.ogm=38\;5\;13:\*.mp4=38\;5\;13:\*.m4v=38\;5\;13:\*.mp4v=38\;5
\;13:\*.vob=38\;5\;13:\*.qt=38\;5\;13:\*.nuv=38\;5\;13:\*.wmv=38\;5\;13:\*.asf=38\;5\;13:\*.rm=38\;5\;13:\*.rmvb=
38\;5\;13:\*.flc=38\;5\;13:\*.avi=38\;5\;13:\*.fli=38\;5\;13:\*.flv=38\;5\;13:\*.gl=38\;5\;13:\*.dl=38\;5\;13:\*.
xcf=38\;5\;13:\*.xwd=38\;5\;13:\*.yuv=38\;5\;13:\*.cgm=38\;5\;13:\*.emf=38\;5\;13:\*.axv=38\;5\;13:\*.anx=38\;5\;
13:\*.ogv=38\;5\;13:\*.ogx=38\;5\;13:\*.aac=38\;5\;45:\*.au=38\;5\;45:\*.flac=38\;5\;45:\*.mid=38\;5\;45:\*.midi=
38\;5\;45:\*.mka=38\;5\;45:\*.mp3=38\;5\;45:\*.mpc=38\;5\;45:\*.ogg=38\;5\;45:\*.ra=38\;5\;45:\*.wav=38\;5\;45:\*
.axa=38\;5\;45:\*.oga=38\;5\;45:\*.spx=38\;5\;45:\*.xspf=38\;5\;45:; export LS_COLORS
MAIL=/var/spool/mail/root; export MAIL
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin; export PATH
PWD=/root; export PWD
LANG=en_US.UTF-8; export LANG
HISTCONTROL=ignoredups; export HISTCONTROL
SHLVL=1; export SHLVL
```

```
XDG_SEAT=seat0; export XDG_SEAT
HOME=/root; export HOME
LOGNAME=root; export LOGNAME
LESSOPEN=\|\|/usr/bin/lesspipe.sh\ %s; export LESSOPEN
XAUTHORITY=/root/.xauthlCR8rx; export XAUTHORITY
cd /root || {
     echo 'Execution directory inaccessible' >&2
     exit 1
}
${SHELL:-/bin/sh} << 'marcinDELIMITER45e6452d'
pwd > /tmp/test13.atd

marcinDELIMITER45e6452d
```

Pour supprimer cette tâche il convient d'utiliser la commande suivante :

```
# at -d N° de tâche [Entrée]
```

Par exemple :

```
[root@centos7 ~]# at -l
1   Sat Dec 31 13:00:00 2016 a root
2   Sat Dec 31 14:00:00 2016 a root
[root@centos7 ~]# at -d 2
[root@centos7 ~]# at -l
1   Sat Dec 31 13:00:00 2016 a root
```

Finalement, pour exécuter plusieurs commandes à la même heure d'une manière séquentielle, vous pouvez les insérer dans un fichier :

```
# at 14:30 12/31/2016 < todo.list [Entrée]
```

Par exemple :

```
[root@centos7 ~]# touch todo.list
```

```
[root@centos7 ~]# echo pwd > todo.list
[root@centos7 ~]# echo free >> todo.list
[root@centos7 ~]# echo who >> todo.list
[root@centos7 ~]# cat todo.list
pwd
free
who
[root@centos7 ~]# at 14:30 12/31/2016 < todo.list
job 3 at Thu Dec 31 14:30:00 2016
```

<html>

Copyright © 2004-2017 Hugh Norris.

Ce(tte) oeuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.

</html>