entOS / RHEL : Beginners guide to cron

By admin

Cron is a time-based job scheduler, it is configured it to run commands at given times or intervals. Each User has a cron table which defines what to execute and at what interval. crontab command is used to create, modify and view cron jobs.

Configuration files and directories

– cron is controlled by a set of files called crontabs.

– There is the master file in /etc/crontab, along with crontab files for the users in **/var/spool/cron/**. In the latter directory, the files are given the same name as a user’s username.

– The **/etc/crontab** file automatically executes items in several subdirectories at regular periods. The scripts places in various directories – /etc/cron.\* are run as per the time interval given below. All the scripts in these directories are run with the root privilege.

Directory Time

/etc/cron.hourly First minute of every hour

/etc/cron.daily Between 3:05 AM to 10.55 PM each day

/etc/cron.weekly Between 3:25 AM and 11:10 PM after 7 days since last execution

/etc/cron.monthly Between 3:45 AM and 11:30 PM after a month since last execution

– All the sysadmin needs to do is to place a shell script or a link to an executable in one of the directories and it will automatically be run at the appropriate time.

crontab syntax

Setting up a user-level crontab is somewhat different. The files in /var/spool/cron are not edited directly. Instead, a program called crontab is used to manipulate them. The syntax of the crontab command is :

Usage:

crontab [options] file

crontab [options]

crontab -n [hostname]

Options:

-u [user] define user

-e edit user's crontab

-l list user's crontab

-r delete user's crontab

-i prompt before deleting

-n [host] set host in cluster to run users' crontabs

-c get host in cluster to run users' crontabs

-s selinux context

-x [mask] enable debugging

How to edit a crontab

The best way to edit a crontab is using the command **crontab -e**. Another way of doing it is:

1. su to the user whose cron you want to change

2. crontab -l > file [ copy the crontab to a file ].

3. vi file [ make changes to the file as per your need ]

4. crontab file [ this makes the "file" as new crontab ]

There is no need to restart the cron daemon after this.

Interpreting the time and date fields

Each cron command has 5 time and date fields, followed by a user name [optional], and if this is the system crontab file, it will be followed by a command. Commands are executed when the time specified by the time/date fields matches the current time.

field allowed values

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

minute 0-59

hour 0-23

day of month 0-31

month 0-12 (or names, see below)

day of week 0-7 (0 or 7 is Sun, or use names)

A field may be an asterisk (\*), which always stands for first to last. So when used in the month field, it means every month from 0 (Jan) to 12 (Dec).

Example Cron job:

# 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

| | | | |

0 1 \* \* \* [user-name] [command to be executed]

Examples of setting cron jobs

Below are few examples of crontab usages to understand how to schedule a task :

Example : runing a job five minutes after midnight, every day :

5 0 \* \* \* /home/oracle/scan\_asm\_devices.sh

Example : running a job at 5:30pm on the 1st of every month :

30 17 1 \* \* mail -s "It's 5:30pm"

Example : Running a job at 4:05 every Monday.

5 4 \* \* mon echo "run at 5 after 4 every monday"

User access control

– To allow users to access the crontabs, /etc/cron.allow and /etc/cron.deny files can be used to allow or deny access respectively. Simple put one username in either of the 2 files to allow or deny the access to the crontab.

– If the /etc/cron.allow file exists then the /etc/cron.deny file will not be used.

– In the default installation only an empty file /etc/cron.deny will exist.

– If neither of the files exists then only root user will be given access to schedule a job through cron.