Linux

mthree

Introduction to Linux

Module 13: CRON

ALUMNI

BY MTHREE CONSULTING

Introduction to Linux

In this module we will go into the details of scheduling, its importance, and how to use the Linux-based scheduling tool cron.

Scheduling

By now you should be comfortable in a Linux environment.

When running large scale platforms made up of multiple servers and processes, it is apparent we need an automated way to start/stop these platforms and run any supporting scripts that are required to keep the systems running.

There are multiple tools to do this. Most companies will have some, if not all of their “Scheduling jobs” running on a centralised tool, such as Autosys or Control-M.

Unix provides a scheduling tool called Cron on the server. This module will teach you how to use and read cron.

Cron

Cron (named after the Greek god of time Kronos) is made up of the crontab which opens the cron table for editing and the cron daemon which executes the tasks at the scheduled time.

Companies are generally phasing out the use of cron for the following reasons:

De-centralised

Each server has its own crontab – which becomes challenging when you are running systems that need coordination between multiple jobs

Dependencies

Centralized systems allow you to specify to only start a job once a previous job has completed successfully

Alerting

Centralized systems such as Autosys and Control-M have good alerting mechanisms in place if a job is unsuccessful, which you do not get with cron

Crontab Syntax

As mentioned in the previous slide, crontab is the command to open the crontab table and view all the jobs.   Each user on the server can have their own crontab.

Crontabs can be found in /var/spool but the crontab command is the way to edit them.

crontab [-u user] file

-u user: this will open up the crontab of the user specified. If not included in the command, it will open the crontab for the user who ran the command

File: load the data from the specified file; if the file is a, it will read from standard input

crontab [-u user] [-l | -r | -e] [-i]

-l: display the current crontab
-r: remove the current crontab
-e: **e**dit the current crontab, using the editor specified in the environment variable VISUAL or EDITOR
-i: remove the crontab but ask for confirmation first (same as r but with a prompt)

What does a cron entry look like?

Below you can see an example of a crontab entry. The first 5 fields are to do with the scheduling of when to run – the table below explains each field. The 6th field specifies which command to run.

```
15 6 2 1 * /home/ec2-user/fixGenerator.sh
```

Field	Comments
Minute	Values allowed 0-59
Hour	Values allowed 0-23
Day of month	Values allowed 1-31
Month	Values allowed 1-12 (can use the first three letters of the name if you aren’t doing ranges)
Day of week	Values allowed 0-7 (0 or 7 is Sunday or you can use the first three letters of the name again)

Please note ranges can be included (8-11 means 8,9,10 and 11 inclusive).

Lists are also allowed, separated by commas: 1,2,5,9

Step values can also be used with ranges (0-23/2 for hours can mean every other hour)

Asterisks can also be used to symbolize running for every value in that field (*)

The crontab file

Each line of the file can be active or inactive. Blank lines, leading spaces, and tabs are ignored. Anything starting with # is assumed to be a comment and ignored.

Cron jobs can be allowed or disallowed for individual users - /etc/cron.allow and /etc/cron.deny.

If cron.allow exists, the user has to be present. If this file is not there but cron.deny is, the user must not be listed in that file if they want to run the command. If neither exists, only root can run the commands

Permissions can also be defined using PAM (pluggable authentication module) – located in /etc/cron.d/

Command	Comments
crontab –e	This is how you edit your crontab – it will open in your default editor so everything you learnt around using that editor will work here
crontab –l	This will list the contents of your crontab
crontab –r	This will remove your crontab – effectively removing all scheduled jobs
sudo crontab –l –u ec3-user	This (if you are permissioned to do so) will allow you to view the crontab of user ec3-user (sudo will give you elevated permissions)

Some cron examples

Here are some examples of crontabs.

01 00 * Jan Monday /home/ec2-user/fixGen	Run at 12:01AM every Monday in January
00 09-18 * * * /home/ec2-user/fixGen	Run every hour from 9am-6pm every day
*/20 09-18 * * * /home/ec2-user/fixGen	Run every 20 mins from 9am-6pm every day
0 9,18 * * Mon /home/ec2-user/fixGen	Run every Monday at 9am and 6pm
30 22 * * Mon,Tues,Wed,Thu,Fri /home/ec2-user/fixGen	Run at 10:30pm every weekday