Best practices for maintaining cronjobs and shell scripts?

Asked 13 years, 4 months ago Modified 13 years, 4 months ago Viewed 9k times

14

I have inherited a sprawling crontab that I need to maintain and update. I don't have much experience with it or bash scripting (I think I've got a decent grip on the basics) and I want to do a good job. Short request: Any guidelines for 'refactoring' a messy crontab and set of bash scripts

Long request: I've run into a number of issues, but are so many people using cron files etc that I feel like I must be missing some large repository of information, best practices and tools - or is this just a stylistic difference for this kind of programming? (My bias: why do something manually if I can use a tool to do it faster, consistently and well?).

Examples of issues so far:

1. Due to an external event, the crontab didn't run for a couple of days. Along with someone else, we manually went through the list, trying to figure out what didn't run, what we needed to rerun, and what scripts we needed to edit and run with earlier dates etc. What I can't find:
    - There are plenty of (slightly pointless) 'cron generators' online. Where are the reverse? Something I can feed in a long crontab, two dates, and have it output which processes should have run when, or just how many times total? This seems within my meager scripting capabilities, so shouldn't it exist already? ;)
    - Alternatively, if I ever have to do that again, is there some way of calling a bashscript so that any instances of date() are pre-set to an earlier time, rather than changing every date call within the script? (e.g. for all the missed reports and billing invoices)
2. It turns out a particular report hadn't been running for two years. It was just requested again, and lo, there it was in the crontab! The bash script just had broken path references to the relevant files. What I can't find: some kind of path checker for bash files? Like a website link checker. Yes I'll be going through these all manually eventually, but it'd show up some at least some of the problem areas.
3. It sounds like some times, there has either been too long or short a gap between dependent processes, so updates have happened after the first has been run, or the first hasn't finished running before the second has been called. I've seen a few possible options for this (eg anacron runs in sequential order), but what would you recommend?
4. There are also a large number of essentially meaningless emails generated from the crontab (scripts throwing errors but running 'correctly', failing mostly silently, or just printing everystep of non-essential scripts). I'll be manually going through scripts and trying to get them to provide more useful data, or 'succeed quietly', but y'know - any guidelines?

If my understanding or layout of the issue is confused, then I apologize, but hey - you see my problem then! I need to go from newbie, to knowing what to do to get this *right*, and not screw up a touchy system further. Thanks!

bash cron file-organization

Share Follow

asked Apr 13, 2011 at 10:55
Azazo
311 ● 2 ● 8

2 Answers

Sorted by: Highest score (default)

4

Not a full answer, but more resources that have been helpful: http://blog.endpoint.com/2008/12/best-practices-for-cron.html

I am slowly going through this, and trying to implement each of the points. I hadn't thought to google 'best practices cron' til after my post. :P

For version control, I'm just going to use RCS in the meantime, as I edit scripts on a file-by-file basis, but I've been advised to get Git set up (or Mercurial if I was on a Windows system).

This actually sounds great: http://everythingsysadmin.com/2010/09/xed-202-released.html "xed is a perl script that locks a file, runs $EDITOR on the file, then unlocks it."...and puts it in RCS if it wasn't already. Completely brainless version control. If I get my head around bash, I'd like to create an editing shortcut that automatically commits to whichever version control system I use.

Other tips I received from an System Admin, Dates: Rather than using say, date, or --date="last monday", use a fixed date and add a day/week etc to it each time it runs (if not more than current day obviously), because then if the script doesn't run, I can just re-run the script repeatedly until it catches up. Ah! (And, this might sound obvious, but heaps of the reports I'll be eventually edit, don't say *prominently* what dates the report is running for. Will fix.)

And was reassured I should try and get the cron emails as quiet as possible, so that I actually notice if there's an error email. There are wrappers for better cron error reporting that I have not yet investigated, linked here: http://habilis.net/cronic/

Share Follow

answered Apr 18, 2011 at 22:02
Azazo
311 ● 2 ● 8

3

Herculean task ahead of you, best of luck. :)

I'd suggest finding all the tasks that run daily and shove them into their own scripts in `/etc/cron.daily/`. Same for weekly into `/etc/cron.weekly`, hourly, and monthly.

You might want to investigate use of `anacron(8)` for scheduling your jobs, if the machine won't always be online, but you still need some level of control over when the jobs are run. It's been the default cron-helper-tool for multiple distributions for a few years, so hopefully it's stable enough to rely on for your own tasks; but I could easily imagine that it might not perfectly meet your needs.

Faking the dates to scripts can be done with at least two packages on Ubuntu: `datefudge` and `faketime`. I have no experience with either, but both sound like they should be able to help. I hope you won't need it in the future. :)

Sorry, I know of no path-checker for bash scripts. It seems unlikely, since simple scripts are simple and easy to check by eye :) and complex scripts will be generating their pathnames at runtime anyhow. Maybe you could keep a database of pathnames used by each script and write a new script to verify that database regularly.

You *could* disable the cron email by setting `MAILTO=""`. I'm not sure I like this. Maybe setting `MAILTO` to a logging-only account would help the deluge. Another option is getting really good at your `procmail(1)` rules so you can stuff them in another mailbox completely.

Getting good at `mutt` `color` or `score` controls can help you spot the wheat amongst the chaff. (`color index red black ERROR` or similar commands might help you spot the problems more quickly.)

Share Follow

answered Apr 13, 2011 at 11:16
sarnold
104k ● 22 ● 185 ● 243

Thanks for the advice. Reassuring at least! "Sorry, I know of no path-checker for bash scripts. It seems unlikely, since simple scripts are simple and easy to check by eye :)" Hahahahaha! Oh well. ;) Mutt is currently driving me nuts (HTML email? Easy. Csv attached to an email? Easy. Both? Ridiculous!) so I need to learn my way around it better anyway. – Azazo Apr 18, 2011 at 22:07