

Standalone Sysadmin

GENERAL

THE FUTURE OF SYSTEM ADMINISTRATION

2009/08/19 | MATT SIMMONS | 37 COMMENTS

Here's a hint...it's not sitting at a shell prompt typing out commands.

I'm sitting here on hold, waiting for Dell support to come online and tell me that my laptop is out of warranty, so I've got some time to think, and I'm reflecting on some of the sysadmins I've met lately, as well as some of the technologies I've seen for enabling administration, and it all seems pointed in one direction. Sysadmins are becoming developers.

If you think about it, in a sense, we've always been developers...developers of systems, that is. Infrastructuresmiths, you might even call us. In my mind, though, there's always been a delineation between the admins who run the systems and the programmers who program them... at least there has been since the early sages who wrote the software and the operating system. Maybe that's just my own inexperience talking (if so, please, comment and let me know!).

Administration is definitely getting away from the method that I "grew up" learning, which was *"fix it right the first time so you don't screw it up, then document it when you get the chance"*, where "fix it" meant log in to the machine (as root, of course), edit the configuration by hand, test the syntax (if the service supports that sort of thing), then restart the service and check to see if the edit worked. If what you "fixed" broke something else, then you'd scramble to fix it, and restart the service again. Lather, rinse, repeat.

It's probably an understatement to say that there are better ways of going about this.

For a while, best practices flirted with development by storing machine configurations in version control repositories like subversion. You would make a change to the config, commit the file to the subversion repository (along with notes regarding whatever you did), and if your change broke the service, you just rolled back the change. It was a big improvement, and similar in methodology to how developers manage source code. This was really the first step for us.

Now, a few years later, we're being introduced to another paradigm of system administration: framework-based configuration management AKA [cfengine](#), [puppet](#), and [chef](#).

These configuration management packages offer exciting new time-saving abilities. Not only do you get configuration management, you get it centralized and programatized. I can't say I have any experience with cfengine or chef, but what little experience I have with puppet has taught me that it's more programming language than configuration file, and best practices are to treat it as such. Store it in a subversion repository to keep it safe, and to roll back changes when needed. You can even use subversion to create development testing branches and the like.

If we're not becoming developers, we're blurring the line quite a bit. This isn't necessarily a bad thing, but it does require us to update our skillset a bit. CS degrees which teach proper development techniques are going to become more valuable. Those of us without those particular skills may want to keep them in mind for the next time we get some free time to work on improving skills.

[Michael Halligan](#) said the other day [on twitter](#), "I feel that any sysadmin today who isn't learning Ruby and either Chef or Puppet will be unemployed in 5 years".

That might be overstating it a bit, but knowing up-to-date system management solutions will never be a hindrance, in my opinion.

Are you using anything remotely modern on your systems, or do you still use the "change and scramble" method? And don't be afraid to share, we won't judge. Chances are really good that we've all seen (and done) worse!

Please share this post on...

37 Comments

Standalone SysAdmin

 Login ▾

 Recommend

 Share

Sort by Best ▾



Join the discussion...



MERLE CERVANTES · 5 years ago

I've been looking around www.standalone-sysadmin.com and actually am impressed by the great content material here. I work the nightshift at my job and it is so boring. I've been coming