

Keeping testing and production server environments clean, in sync, and consistent

Asked 11 years, 2 months ago Active 8 years, 11 months ago Viewed 9k times



16



7



It seems that the company that I work for is always **struggling with our customers' server environments**.

Specifically, we almost always encounter problems with testing servers and production servers, and the fact that they always seem to be configured differently. When we test the applications that we develop, the testing servers behave in one way, and thus we tweak and configure our applications to fit that particular behavior. But when we install the same application on the production servers we observe another behavior that is not consistent with the testing servers, thus rendering our tweaks and configurations useless. The most frustrating part is that this happens all the time and that no one seem to know what to do about it.

Of course we have a general idea of why this happens. Every cloned environment starts out the same and works the same the first couple of days, but sooner or later someone reconfigure something in only one of the server environments (be it a database update, an update of a component library, a web file update, or other configurations), thereby leading to discrepancy. And as time goes by, more and more discrepancies builds up. But the question is: what can we do about it?

I've tried searching the web but can't find any good answers on what to do. I've also tried to figure out some solutions on my own, but most of my ideas seem to be problematic in some way. New routines, no matter how rigorous, can be circumvented. Regular cloning of the production servers to create testing servers is a tedious and often very slow process. Automatic replication is not always reliable or even possible. So what on Earth should we do about this problem? How can we guarantee that the experience when testing will match the experience when going live?

I imagine that others have this very problem as well. Or do they? Maybe it's just my particular company that is incompetent? Have any of you encountered the problem? If so, what did you do about it?

Sincerely,

Linus, Swedish systems developer

synchronization

production-environment

consistency

test-environments

edited Mar 12 '09 at 21:23



Jeff Atwood

59.1k ● 45 ● 145 ● 149

asked Mar 12 '09 at 17:33



Linus

Linus, this refers *specifically* to shipping code to your customers' environments, right? How much control can you realistically exert over those environments. – [Jeff Atwood](#) Mar 12 '09 at 21:21

5 Answers

Active	Oldest	Votes
--------	--------	-------



6



You need to start keeping track of every change that you make to the testing environment and provide a way for propagating this to the production environment.

For code, this means a versioning systems such as CVS, Subversion or GIT.

For the database, it means a structure comparison tool or deploy scripts that update the production database.

For configuration, the two systems should be exactly the same and any 'tweaking' or changing needs to first be applied to the testing server then later applied to the production server during a deploy.

Until you have a process that works, you will continue to have problems.

answered Mar 12 '09 at 17:40



jonstjohn

53.9k ● 8 ● 39 ● 54

for config management, if you're feeling really ambitious you can look at cfengine or puppet to define your system config (installed packages, config file contents, etc.) with a declarative language. – [Peter Cordes](#)
Dec 9 '09 at 5:11



0



You need to make sure that any changes to the environments are done in a consistent manner.

I'd consider either starting with fresh images and enforcing a strict modification log policy, or using something like Capistrano to execute remote commands on and deploy code to all machines simultaneously.

Ideally, all requirements should be checked into your version control system (along the lines of how Rails lets you store gems in the /vendor directory and preferentially loads those at runtime), along with a readme file that describes exactly how to set up the environment (required libraries, etc). The readme file needs to be rigorously updated by anyone who makes changes to the environment.

answered Mar 12 '09 at 17:37



Jarin Udom

1,789 ● 3 ● 18 ● 23



0



Your problems are quite normal. There are at least two strategies I know work fairly well:

If you're distributing on linux, you can build rpms/debs from your development process and use the package management function. I know lots of projects do this with great success for inhouse projects.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



maintained by development, and this script overwrites *any* modifications that have been made manually. A script like this is usually maintained by development, kept under version control and sent to deployment as a full distribution. We use cygwin for this. Normally the script reads some kind of configuration that may be administered by operations. I've had scripts that'd actually set up the *entire* system from scratch as if installing on a totally blank, freshly installed machine.

Both of these strategies should preferably include automated production of these artifacts all the way from your build script/build system. The smoother this process runs, the better for all involved parties.

edited Mar 12 '09 at 18:06

answered Mar 12 '09 at 17:41



krosenvold

67.6k ● 26 ● 135 ● 199



0

Of course we have a general idea of why this happens. Every cloned environment starts out the same and works the same the first couple of days, but sooner or later someone reconfigure something in only one of the server environments



I think you're being generous, or you've been very lucky. As often as not shops that need to contract out for development work don't really understand the development process. If they provide you with a test environment at all what you'll get is a their production system from before the last server refresh.

answered Mar 13 '09 at 15:10



Joel Coehoorn

343k ● 101 ● 515 ● 751



0

For keeping MySQL databases in sync, I just ran into this tool:

<http://schemasync.org/>



I haven't actually used it yet so I can't speak to whether it's any good, but we need some way to control schema drift between test and prod so we're going to give it a shot.

answered Jun 1 '11 at 23:05



user41871

