DevCentral upgrade&hellip; with a few hitches

Jeff Browning, 2010-01-12

With the rapid growth of DevCentral, we continue to get great suggestions for how we can improve the site (BTW – you too can provide suggestions via the Feedback form here). We also have a nonstop stream of ideas about cool stuff we would like to see as well. All of this means that we've become more dedicated to regular updates/upgrades to DevCentral, in both the software we use AND the customization we build on top of it. As a result, you may have seen a few more maintenance pages over the past few months than usual. If you're interested in what we've been doing, read on!

The Updates

Yesterday, we ran our second update in the past couple months. For this update, our major focus was on streamlining performance at the application layer. By all measures, the upgrade went smoothly. Just a few things we focused on included:

- **Removing unused features/functionality**: we're constantly messing around with new features. Often, they don't see the light of day. So, to stay lean and mean, we removed some clutter to make replication smaller and less complicated.
- **Registration process simplification**: we've had some feedback that the registration process and form were just a bit long. So, we've shortened this and hopefully new members will be able to register faster. Less time, faster access to the good stuff.
- **Reducing calls out to external services**: in today's web app world, it's scarily easy to add objects to application skins that call out to the latest widget or script hosted elsewhere on the web. However, do you need to be able to "Add This!" to 3,562 different social sites on the web? No – we doubt it so we removed it.
- **Group enhancements**: Groups have really taken off in the community. And, we keep tweaking and improving them. While nothing immediately viewable, we've done some stuff that lays the groundwork for continued expansion of how you can benefit from Groups on DevCentral. Let's just say that while not apparent now, we think you'll like what you see in some future updates.
- There's more stuff… but this gives you the feel of what we do, as part of a continued roadmap, to continue to evolve and improve DevCentral.

After working through development over the past month or so as well as our usual Staging/QA process, we pushed the button yesterday and rolled out the updates. However, it wasn't until later in the day that things got a little bit interesting…

… and the Hitches

Without going into the gory details (we'll probably do that later as it's probably interesting to some of you), we run our application in redundant datacenters behind a whole host of F5 gear (GTM, LTM, WA, ASM). We use iRules extensively. We're sort of biased but we think F5 gear rocks (and it would be lame if we didn't use it extensively…).

As part of our infrastructure, our IT team manages a pretty extensive monitoring system to help us know what's happening with our application, servers, and the infrastructure. Around 5pm (PST) yesterday, we started getting some funky alerts. Nothing serious but enough to be monitored more closely. Eventually, through F5 health monitors on LTM and GTM, we were able to flop datacenters automagically to keep users connecting to the application. All good.

But, in email to MVP and other active users, we learned that all was not completely ideal…

> *Jeff: "Hey – we're seeing some funky alerts about the application. What are you seeing?"*
>
> *DevCentral Member: "I was getting TCP resets consistently tonight. The IP seemed to respond very consistently to pings. So I was guessing it was an app layer issue."*

Hmmm. Thanks to our ninja IT team and the DC gang, we took some measures that seemed to resolve/stabilize things and we went to sleep. However, this morning, the issues reappeared and we dug deeper.

It turns out that the upgrade flipped a bit in the database that told certain scheduled jobs to run on multiple servers. Combine this with the fact that a couple of these jobs were pretty resource intensive and were running against very large tables, and you end up with some DB deadlocking. Deadlocking is bad, and will drag a server to its knees quickly, even if not under load, let alone serving thousands of pages.

This took a while to find because the only symptom presenting itself was pegged CPUs on the DB systems. Fortunately we've got an ace team of infrastructure & app folks that work together quite well, so we were able to track this down quickly. It can't be stressed enough here how important it is to have a combined team that can crack down on these kind of issues from both angles (infrastructure and application sides).

Once the bits were set back to the intended settings and centralized job scheduling for log and notification management was back in place, the issue went away completely, and it was back to business as usual.

What We Learned

This was a bit of a wild goose chase and it cropped up from a place we never would have expected. Nothing was changed in the app code surrounding the jobs that got flipped on universally. It was just a complication of the upgrade itself. A few thoughts we've come away from this with:

- Logging of various types is vital to any application, but can cause more headaches than most people realize if not carefully monitored.
- Running multiple application servers in front of a single database server creates an interesting situation wherein you can easily have multiple servers attempt to fire up the same thread within the DB … this is bad, generally speaking, and anything you can do to monitor for and/or prevent this is a good thing.
- To control this, it's imperative that you tightly monitor your logging, cleanup and truncation processes. Space them appropriately, run them from a centralized source, etc. Also, after any upgrade this would be an excellent thing to add to your QA checklist, even if changes weren't made to this portion of your app. It's on our list now, that's for sure. ;)
- It cannot be stressed enough how important it is to have a collaborative mindset between both development and operations. If you work in an org where ops and the apps teams work well together (like we do), cherish and protect it. If you don't, you might want to consider trading some of your investment in technology for spending a little more time figuring out how you can work better with your peers.

So, there you have it – a little insight into what we've been up to. We believe this continued focus on enhancements will deliver an even better community resource for you. And – maybe you'll even benefit a little from some of the lessons we learned from this most recent upgrade that help your next upgrade go more smoothly.

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com