In the Cloud, It's the Little Things That Get You. Here are nine of them.

Don MacVittie, 2012-17-09

#F5 Eight things you need to consider very carefully when moving apps to the cloud.

Moving to a model that utilizes the cloud is a huge proposition. You can throw some applications out there without looking back – if they have no ties to the corporate datacenter and light security requirements, for example – but most applications require quite a bit of work to make them both mobile and stable. Just connections to the database raise all sorts of questions, and most enterprise level applications require connections to DC databases.

But these are all problems people are talking about. There are ways to resolve them, ugly though some may be. The problems that will get you are the ones no one is talking about. So of course, I'm happy to dive into the conversation with some things that would be keeping me awake were I still running a datacenter with a lot of interconnections and getting beat up with demands for cloudy applications.

1. The last year has proven that cloud services WILL go down, you can't plan like it won't, regardless of the hype.
2. When they do, your databases must be 100% in synch, or business will be lost. 100%.
3. Your DNS infrastructure will need attention, possibly for the first time since you installed it. Serving up addresses from both local and cloud providers isn't so simple. Particularly during downtimes.
4. Security – both network and app - will have to be centralized. You can implement separate security procedures for each deployment environment, but you are only as strong as your weakest link, and your staff will have to remember which policies apply where if you go that route.
5. Failure plans will have to be flexible. What if part of your app goes down? What if the database is down, but the web pages are fine – except for that "failed to connect to database" error? No matter what the hype says, the more places you deploy, the more likelihood that you'll have an outage. The IT Managers' role is to minimize that increase.
6. After a failure, recovery plans will also need to be flexible. What if part of your app comes up before the rest? What if the database spins up, but is now out of synch with your backup or alternate database?
7. When (not if) a security breech occurs on a cloud hosted server, how much responsibility does the cloud provider have to help you clean up? Sometimes it takes more than spinning down your server to clean up a mess, after all.
8. If you move mission-critical data to the cloud, how are you protecting it? Contrary to the wild claims of the clouderati, your data is in a location you do not have 100% visibility into, you're going to have to take extra steps to protect it.
9. If you're opening connections back to the datacenter from the cloud, how are you protecting those connections? They're trusted server to trusted server, but "trusted" is now relative.

Of course there are solutions brewing for most of these problems. Here are the ones I am aware of, I guarantee that, since I do not "read all of the Internets" each day (Lori does), I'm missing some, but it can get you started.

1. Just include cloud in your DR plans, what will you do if service X disappears? Is the information on X available somewhere else? Can you move the app elsewhere and update DNS quickly enough? Global Server Load Balancing (GSLB) will help with this problem and others on the list – it will eliminate the DNS propagation lag at least. But beware, for many cloud vendors it is harder to do DR. Check what capabilities your provider supports.
2. There are tools available that just don't get their fair share of thunder, IMO – like Oracle GoldenGate – that replicate each SQL command to a remote database. These systems create a backup that exactly mirrors the original. As long as you don't get a database modifying attack that looks valid to your security systems, these architectures and products are amazing.
3. People generally don't care where you host apps, as long as when they type in the URL or click on the URL, it takes them to the correct location. Global DNS and GSLB will take care of this problem for

correct location. Global DNS and GSLB will take care of this problem for you.

4. Get policy-based security that can be deployed anywhere, including the cloud, or less attractively (and sometimes impractically), code security into the app so the security moves with it.
5. Application availability will have to go through another round like it did when we went distributed and then SOA. Apps will have to be developed with an eye to “is critical service X up?” where service X might well be in a completely different location from the app. If not, remedial steps will have to occur before the App can claim to be up. Or local Load Balancing can buffer you by making service X several different servers/virtuals.
6. What goes down (hopefully) must come back up. But the same safety steps implemented in #5 will cover #6 nicely, for the most part. Database consistency checks are the big exception, do those on recovery.
7. Negotiate this point if you can. Lots of cloud providers don’t feel the need to negotiate anything, but asking the questions will give you more information. Perhaps take your business to someone who will guarantee full cooperation in fixing your problems.
8. If you actually move critical databases to the cloud, encrypt them. Yeah, I do know it’s expensive in processing power, but they’re outside the area you can 100% protect. So take the necessary step.
9. Secure tunnels are your friend. Really. Don’t just open a hole in your firewall and let “trusted” servers in, because it is possible to masquerade as a trusted server. Create secure tunnels, and protect the keys.

That’s it for now. The cloud has a lot of promise, but like everything else in mid hype cycle, you need to approach the soaring commentary with realistic expectations. Protect your data as if it is your personal charge, because it is. The cloud provider is not the one (or not the only one) who will be held accountable when things go awry.

So use it to keep doing what you do – making your organization hum with daily business – and avoid the pitfalls where ever possible.

In my next installment I’ll be trying out the new footer Lori is using, looking forward to your feedback.

And yes, I did put nine in the title to test the “put an odd number list in, people love that” theory. I think y’all read my stuff because I’m hitting relatively close to the mark, but we’ll see now, won’t we?

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

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