

Send in the Clouds

maddog lets his need for control, stability, availability, speed, and resources guide his cloud service decisions. *By Jon "maddog" Hall*

One of the main reasons a person or company would use Free and Open Source Software (FOSS) is control. In today's world, when you do not have control of your software, it is hard to have control of your business or even your life. When someone else tells you where to put your software, how to use it, when to update it, and when to retire it, they inject indecision and uncertainty into your business, your life, or both.

Such lack of control is one of the issues I have with some of the "cloud" models. In the three main models – Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS) – you must balance ease of use and cost savings with control, and that balance can only be determined by you, the user.

Recently, a large commercial provider of cloud computing made several attempts to introduce new functionality to end users. After huge amounts of advertising hoopla, thousands of organizations and individuals tried this new functionality, some of which was not out of alpha stage. Several months later, the company dropped development on the software and instead advocated other paths for a solution, which required migration of data, retraining of people, and other costs on the part of users.

The loss of this functionality would not have been too bad if the people using it had been able to access the binaries or source code. Then the users could have made a business decision either to keep using the code and maintain it themselves or to go in the new direction. However, because this software was "in the cloud," the users had no alternative but to follow the software creators and go in the new direction – and do so in the time frame that they dictated.

I am not slighting the company that created this software. They were very up-front in telling the users that the software was "alpha" in one case and "beta" in another. However, the companies that jumped on the bandwagon to use this alpha software (and particularly the beta software) were hurt when the availability was removed.

My own solution for this is to break my business and personal cloud computing needs into various categories. For categories in which I need extreme control, stability, and data availability, I aim toward the PaaS model, so I

can buy additional computational and storage needs as required. Because I have these extreme needs for stability, I will plan far ahead for the maximum needs in each space, but I can still treat these needs as expense-based "services" instead of having to buy capital-oriented "products."

For categories in which I need faster deployment or anticipate future shrinking of computer resources, I will probably go with an IaaS model. I will be careful to use virtualization and standards fully to hide the vagaries that could cause issues in movement of the virtualization containers from my desktop to my local servers and (with even more demands of computation) to the cloud. Inside these virtualized containers, I will build a solid and stable solution that lends itself to my control, which I can easily update, patch, and test.

For the most part I recognize that SaaS models for areas like email or "office work" have huge appeal, but you also need a plan to recover your email and documents and a plan to use them without the SaaS – just in case that service were to disappear. For example, I use Google's Gmail, but I also have that email forwarded to another site where it is archived for me so that I can make instant use of it if Google were to disappear tomorrow. I am often asked to use Google Docs because some of my clients use that tool, and they like to share these documents through Google Docs. Yet, I also make sure these documents are transferred to another storage area for redundancy, and in a format that is open, so I can instantly recover them in case of emergency.

Some companies might not have the resources or expertise to set up such detailed procedures and policies – or the discipline to make sure they are carried through. I prefer making these procedures as transparent to users as possible, just to help facilitate their use. Not following through with these types of policies, however, is an invitation for disaster, particularly as we struggle to define clouds and exactly how to use them. ■■■

