SearchVirtualDesktop.com

Why delivering applications from the cloud is technically DaaS

By Gabe Knuth

Recently, I gave a presentation where I called Azure RemoteApp a DaaS product, and someone asked how that could be if it was just for delivering applications.

It's a fair question to ask, and one that Brian Madden and I wrestled with while we wrote our Desktops as a Service (DaaS) book. Ultimately, we decided that the same rules apply to DaaS as they do to more traditional forms of desktop virtualization, and that even if we were only getting Windows apps from the cloud, it's still DaaS.

You might be thinking, "Wait, if there's no desktop, how is it DaaS?" You may not know it, but tools like Citrix XenApp and Microsoft Remote Desktop Session Host (RDSH), which allow you to publish just an application to end users, are still dealing with the desktop. As an admin, you are still managing the backend as if it was a desktop too. The only difference is that when we publish an application to the end users, we are actually hiding the desktop from them altogether.

Way back in 1990s, you could sometimes see this process happening, and with the right exploits, you could even make certain elements of the desktop appear. The platforms have gotten better about that, but the desktop has never gone away completely. Instead, they just got better at hiding it.

Now, let's bring that "hidden desktop" principle to the cloud. Azure RemoteApp, Microsoft's first foray into DaaS, allows you to use RDSH (of which, RemoteApp is a feature) instances in the cloud to deliver Windows applications. Even though the users can't see the desktop, it's still there. It has to be to support the applications because Windows is more of a monolithic brick of an environment than it is a compartmentalized wonderland of manageability.

The same holds true for Amazon Web Services (AWS) AppStream. If you're not familiar with it, AppStream is a platform that AWS created so that users can deploy custom, usually high-end Windows applications from Server 2008 R2 instances running on EC2. These applications have to include an SDK that AWS provides, which they use for scalability and connectivity.

Because it uses Windows, AppStream is still a DaaS tool. Unlike other available products, it just happens to address a different problem. If you look hard enough, you'll see similar approaches from other companies. MainFrame2 made a splash with their pre-launch demos by delivering applications that use Nvidia GRID technology, for instance. Companies like nGenx have created a complex, multi-tenant backend that software vendors can use to offer their customers a quick road to deployment and support. Those customers don't know where the app is coming from, and they don't care as long as it works.

The reality is that all of these things are DaaS, despite the fact that we don't see the desktop. If it's a Windows app that we're paying someone else to host for us, it's DaaS, and the desktop is there even if you don't see it.

03 Jun 2014