

Cloud is an Exercise in Infrastructure Integration



Lori MacVittie, 2011-29-08

When you get down to the architectures involving cloud – whether on or off-premise or hybrid – it's really all about integrating infrastructure.



It remains to be seen if network and operations are better off never using the word “integration” given the nearly violent negative reasons one sees in the development and architecture sides of IT to the word. Integration, even after the introduction of SOA and the nearly messianic view of the role of the enterprise service bus (ESB) in saving us from the horrors of traditional enterprise application integration (EAI), remains problematic for IT. Standards weren't, interoperability didn't and reuse was a concept that was thrown under that bus by developers reluctant to trust or simply unaware of existing services.

But integration also remains necessary. The dominant web 2.0 model leverages APIs instead of endpoints, but ultimately it attempts to do the same thing that every integration model in the history of IT has done: share data and enable business processes to span applications.

What cloud computing is doing is forcing infrastructure –network, storage and application delivery – models to adopt many facets of development. A services-based approach to provisioning as a means to enable IT responsiveness through the application deployment process. Multi-tenant models of management to support fault-isolation and self-service. And integration to support sharing of data and enable operational processes to span components. The data is different, the processes have a different focus, but the concept is exactly the same. Applying the web 2.0 model of API integration to infrastructure enables the sharing of data (context) as well as automation and the ability of infrastructure systems to instruct and be instructed by the management frameworks that drive automation and orchestration.

Today there are very few examples of “public only” cloud computing deployments. Even those that might at first appear to be “public only” such as a SaaS are not; those applications are ultimately integrated with systems and applications and, in many cases, infrastructure that still resides within the physical data center because the data housed in SaaS is only part of the big business picture. It needs to be correlated and integrated and analyzed and warehoused somewhere, and all that requires integration of some sort via APIs to pull the data out of the cloud and put it into systems internal to the data center where they can be used.

Cloud bursting or cloud extension or cloud-what-have-you models that leverage cheaper compute and storage resources from public cloud providers require integration at the infrastructure layers. Using storage resources from the cloud as part of a larger tiering strategy mean that some piece of infrastructure – storage virtualization likely – is integrating those resources via an API. Similarly, compute resources must be integrated – included – in architectures in the data center if they are used as part of a dynamic capacity extension strategy. That requires some integration via an API or infrastructure capable of natively managing those resources in public cloud environments (which, if we peer close enough, we'll see is enabled via .. an API).

Integration means including, to make part of the whole and to do so – hopefully – seamlessly and automatically. Even if an application is completely deployed in a public cloud computing environment it is almost a Heisenberg certainty that its data will be integrated back into some warehouse or application in the data center or that it will ultimately need to participate in some larger process that requires spanning both public cloud and private data center. Or the cloud-deployed application will become so critical to business that it must be managed in the same way as other business-critical applications – it must use corporate identity stores, it must be monitored and managed via existing enterprise application performance management systems, it must be integrated with the rest of the business and operations.

Which means infrastructure integration. Let's hope we've learned enough from the trials and travails of enterprise integration that we get it (mostly) right the first time. I'd also suggest investing heavily in turkeys. Because if enterprise application integration required sacrificial chickens, we're probably going to need something a bit bigger to meet the challenge of integration efforts that will span environments, models, and architectures.

- [📄 The Infrastructure 2.0–Security Connection](#)
- [📄 Strategic Trifecta: Access Management](#)
- [📄 IT as a Service: A Stateless Infrastructure Architecture Model](#)
- [📄 Cloud is the How not the What](#)
- [📄 Challenging the Firewall Data Center Dogma](#)
- [📄 Cloud-Tiered Architectural Models are Bad Except When They Aren't](#)
- [📄 Cloud Chemistry 101](#)
- [📄 You Can't Have IT as a Service Until IT Has Infrastructure as a Service](#)
- [📄 The New Distribution of The 3-Tiered Architecture Changes Everything](#)

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