So You Put an Application in the Cloud. Now what?

Lori MacVittie, 2010-23-08

We need to stop thinking of cloud as an autonomous system and start treating it as part of a global application delivery architecture.

When you decided you needed another garage to house that third car (the one your teenager is so proud of) you probably had a couple choices in architecture. You could build a detached garage that, while connected to your driveway, was not connected to any existing structures or you could ensure that the garage was in some way connected to either the house or the garage. In both cases the new garage is a part of your location in that both are accessed (most likely) from the same driveway. The only real question is whether you want to extend your existing dwellings or not.

When you decide to deploy an application to the cloud you have a very similar decision: do you extend your existing dwelling, essentially integrating the environment with your own or do you maintain a separate building that is still "on the premises" but not connected in any way except that it's accessible via your shared driveway.

In both cases the cloud-deployed application is still located at your "address" – or should be – and you'll need to ensure that it looks to consumers of that application like it's just another component of your data center.

THE OFF-SITE GARAGE

Global application delivery (a.k.a. Global Server Load Balancing) has been an integral part of a multi-datacenter deployment model for many years. Whether a secondary or tertiary data center is leveraged for business continuity, a.k.a. "OMG our main site is down", or as a means to improve performance of applications for a more global user base is irrelevant. In both cases all "sites" have been integrated to appear as a single, seamless data center through the use of global application delivery infrastructure. So why, when we start talking about "cloud" do we treat it as some external, disconnected entity rather than as the extension of your data center that it is?

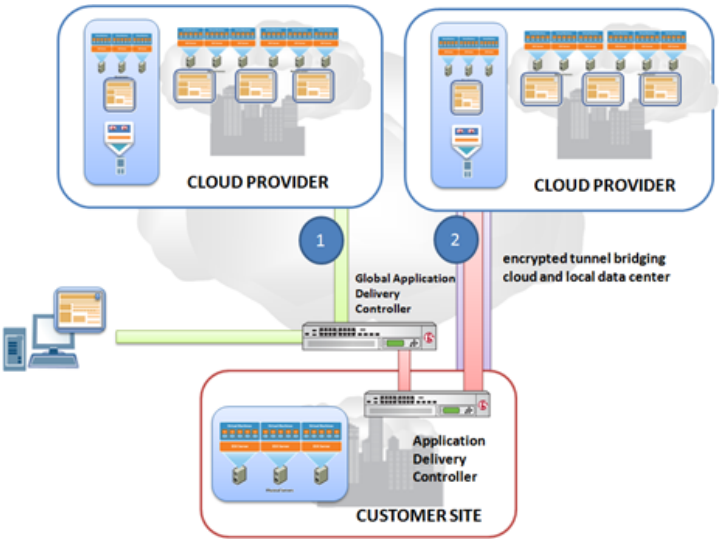
Like building a new garage you have a couple choices in architecture. There is, of course, the continued treatment of a cloud-deployed application as some external entity that is not under the management or control of the organization. That's like using an off-site garage. That doesn't make a lot of sense (unless your homeowners association has judged the teenager's pride and joy an eyesore and forbids it be parked on premise) and neither does it make a lot of sense to do the same with a cloud-deployed application. You need at a minimum the ability to direct customers/users to the application in whatever situation you find yourself using it – backup, failover, performance, geo-location, on-demand bursting. Even if you're only using off-premise cloud environments ***today*** for development or testing, it may be that in the future you'll want to leverage the on-demand nature of off-premise cloud computing for more critical business cases such as failover or bursting. In those cases a completely separate, unmanaged (in that you have no real operational control) off-premise cloud is not going to provide the control necessary for you to execute successfully on such an initiative. You need something more, something more integrated, something more strategic rather than tactical.

Instead, you want to include cloud as a part of your greater, global (multi-site) application delivery strategy. It's either detached or attached, but in both cases it is just an extension of your existing property.

ATTACHED CLOUD

In the scenario in which the cloud is “attached” to your data center it actually becomes an extension of your existing architecture. This is the “private virtual cloud” scenario in which the resources provisioned in a public cloud computing environment are not accessible to the general Internet public directly. In fact, customers/users should have no idea that you are leveraging public cloud computing as the resources are obfuscated by leveraging the many-to-one virtualization offered by an application delivery controller (load balancer).

The data center is extended and connected to this pool of resources in the cloud via a secured (encrypted) and accelerated tunnel that bridges the network layer and provides whatever routing may be necessary to treat the remote application instances as local resources. This is simply a resource-focused use of VPN (virtual private network), one that was often used to integrate remote offices with the corporate data center as opposed to individual use of VPNs to access a corporate network. Amazon, for example, uses IPSEC as a means to integrate resources allocated in its environments with your data center, but other cloud computing providers may provide SSL or a choice of either. In the case that the provider offers no option, it may be necessary to deploy a virtual VPN endpoint in the cloud in order to achieve this level of seamless connectivity.



Once the cloud resources are “attached” they can be treated like any other pool of resources by the application delivery controller (load balancer).

[This is depicted by connection (2) in the diagram]

DETACHED CLOUD

A potentially simpler exercise (in both the house and cloud scenarios) is to treat the cloud-deployed resources as “detached” from the core networking and data center infrastructure and integrating the applications served by those resources at the global application delivery layer.

[This is depicted by connection (1) in the diagram]

In this scenario the application delivery network and resources it is managing are all deployed within an external cloud environment and can be accessed publicly (if one were to determine which public IP address was fronting them, of course). You don’t want users/customers accessing those resources by some other name (you’d prefer www.example.com/remoteapp over 34.35.164.4-cloud1.provider.com of course) and further more you want to be able to make the decision ***when*** a customer will be using the detached cloud and when they will be using local data center resources. Even if the application deployed is new and no copy exists in the local data center you still want to provide a consistent corporate naming scheme to ensure brand identity and trust that the application is ***yours***.

Regardless, in this case the detached cloud resources require the means by which customers/users can be routed to them; hence the use of global application delivery infrastructure. In this case users attempt to access www.example.com/remoteapp and are provided with an IP address that is either local (in your data center) or remote (in a detached cloud environment). This resolution may be static in that it does not change based on user location, capacity of applications, or performance or it may take into consideration such variables as are available to it: location, performance, security, device, etc… (context).

Yes, you could just slap a record in your DNS of choice and resolve the issue. This does not, however, lay a foundation for more dynamic and flexible integration of off-premise cloud-deployed applications in the future.

FOUR REASONS to LEVERAGE GLOBAL APPLICATION DELIVERY

There are many reasons to include in a global application delivery strategy a global load balancing architecture, but these four stand out as the ones that provide the most benefit to both the organization and the user:

1. **Avoid unnecessary application changes due to changes in providers** If all your images or a certain type of content are served by applications deployed in an external cloud computing environment, normalizing your global namespace eliminates the need to change the application references to that namespace in the case of a change of providers. The change is made at the global application delivery layer and is propagated quickly. This eliminates a form of vendor lock-in that is rarely remembered until a change in providers is desired. Developers should never be codifying domain names in applications, but legacy and third-party applications still need support and these often draw their name and information from configuration files that effectively codify the operational location of the server and application. These configurations are less important when the platforms are deployed behind a global application delivery controller and virtualized.
2. **Normalization of global name spaces preserves corporate identity and enables trust** Applications served by a trusted domain are desirable in an age when phishing and malicious code often re/directs users to oddly named domains for the purposes of delivering a malware payload. Global application delivery normalizes global domain namespaces and provides a consistent naming scheme for applications regardless of physical deployment location.
3. **Enhances decision making processes** Leveraging global application delivery enables more control over the use of resources at a user level as well as a business and technical layer. Decisions regarding which resources will be used by whom and when are the purview of global application delivery controllers (GSLB) and provide the organization with flexibility to determine which application instance is best suited to serve any given request based on the context of the request.
4. **Foundational component.** Like load balancing, global load balancing (application delivery) is a foundational component of a well-rounded cloud computing architecture. It provides the means by which the first integrations of off-site cloud computing will be accomplished, e.g. cloud bursting, and lays the foundation upon which more advanced location-selection algorithms will be applied, e.g. cloud balancing. Without an intelligent, integrated global application delivery strategy it will be difficult to implement and execute on strategies which leverage external and internal cloud computing deployments that are more application and business focused.

External (detached) cloud computing environments need not be isolated (silo'd) from the rest of your architecture. A much better way to realize the benefits associated with public cloud computing is to incorporate them into a flexible, global application delivery strategy that leverages existing architectural principles and best practices to architect an integrated, collaborative and seamless application delivery architecture.

del.icio.us Tags: MacVittie,F5,cloud computing,global application delivery,GSLB,load balancing,cloud balancing,DNS,strategy

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