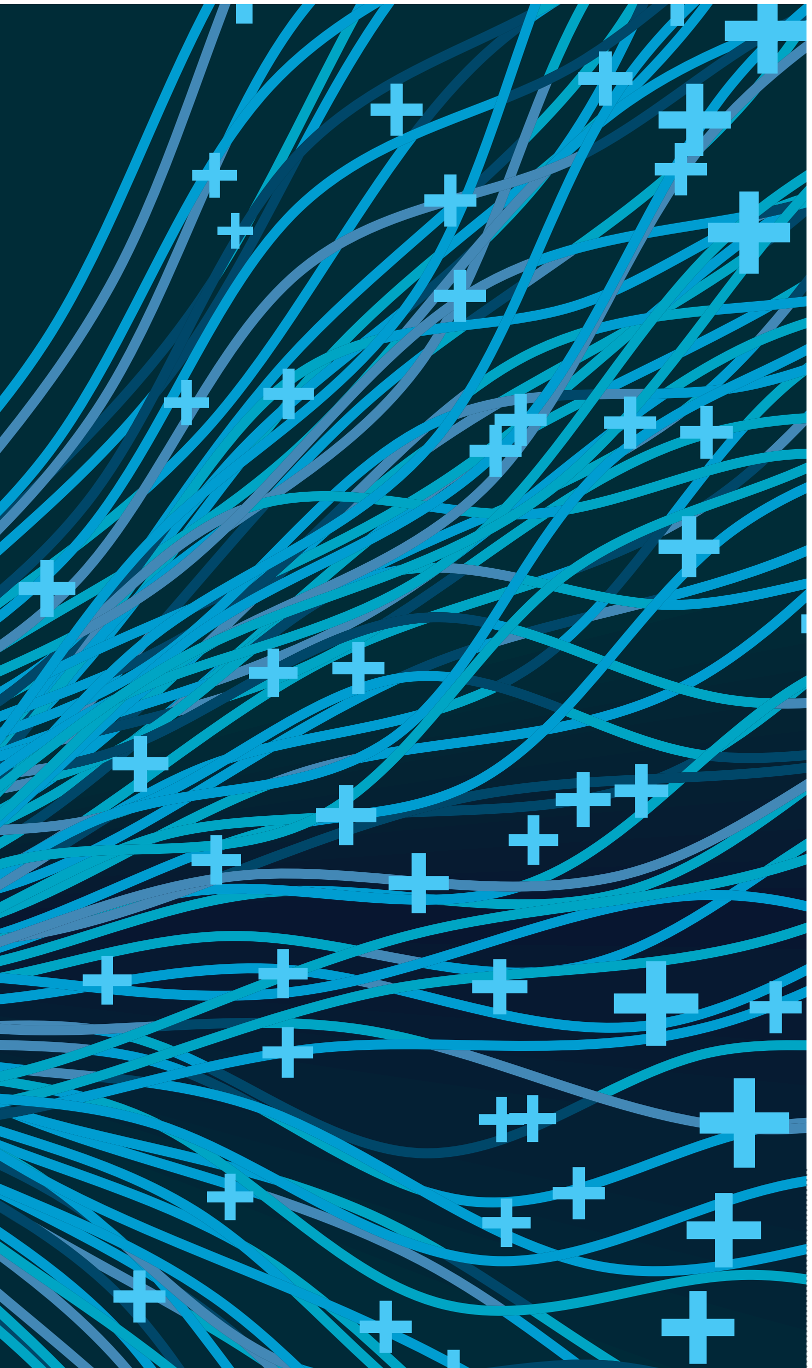

E-Guide

CLOUD APPLICATION INTEGRATION AND DEPLOYMENT MADE SIMPLE

TechTarget | SearchCloud Applications

C**LOUD APPLICATION INTEGRATION** and deployment continue to be some of the top challenges for software developers. In this e-guide, learn how cloud integration tools can help relieve some of the latest deployment challenges. Additionally, discover the most common mistakes in cloud application deployment that you should avoid.

CLOUD INTEGRATION TOOLS RELIEVE THE LATEST DEPLOYMENT CHALLENGES

Tom Nolle

Integration is a required element in nearly all modern application development planning. Over the years, experience with SOA and with front-end web development has taught planners and architects a lot about integration. Early experience in virtualization has expanded this all the more, but the cloud breaks down many modern integration practices, so planners and architects need to start their cloud integration projects by asking why the cloud is different. They then need to assess cloud plans with integration in mind. For most, the key point will be how to accommodate application integration tools within cloud deployment.

In early applications, developers either wrote monolithic applications or tightly coupled separate components into a common load image. A large number of applications are still written this way, but SOA and agile development have encouraged architects to build independent functional components that can be assembled into applications. As business applications became more integrated with business processes and with one another, they required a looser coupling among the pieces simply to eliminate a single enormous load image, an all-or-nothing IT model.

Directory-based integration of these components has become the rule. With directory-based integration, a component registers itself somewhere, and through that registration it can be located and sent work. The directories can be fairly simple, like DNS, or can be a repository for functionality-based browsing, as SOA UDDI would ideally be. In all cases though, the directory expects to create a link to a loaded component or to trigger component loading for initial use.

The cloud challenges this in two ways. First, the cloud presumes a high level of dynamic resource allocation. A component could be put anywhere in the cloud, and so the issue of how to reach it has few boundaries, while in the past you could assume everything resided in at least a fixed data center. Second, one of the principle goals of the cloud is management of availability and performance through scaling of component instances. This means that many components will have to share work or failover in real time. Often the process of establishing integration links to components is not instantaneous, and in the delay period, transaction processing may be compromised.

Meeting these challenges is a matter of assessing cloud plans to identify integration pain points. To start, look for any places where a component could be cloudsourced under load or in failure conditions. Also, look for situations where a cloud component could be relocated by the cloud provider to respond to problems. Any of these scenarios will require some special handling in integration with other components and workflows.

Cloud users report a preference for DNS-based load balancing as a means of steering work-to-cloud components that could be failed over or horizontally scaled. In scaling situations, DNS-based load balancing will allow work to continue with current component links while a new one is added, so the only risk in QoE comes with component failure, which most companies will accept. If any downtime is intolerable, that can be addressed by having at least two copies of the components available at any point and integrated via the DNS.

The issue with DNS-based load balancing is that it doesn't support component browsing (there's no WSDL) and it may create problems for workflows to components that are stateful. If it's not possible to use DNS-based load balancing for either reason, the next-best strategy is to rely on the UDDI and WSDL or BPEL to select among components. That is a potential problem if the application control processes that manage the component links aren't responsible for moving components in the cloud. A component could be un-linked if moving it changes the address. Amazon's solution to this is the elastic IP addresses, which let a static URL reference a movable component. This approach of address translation can be used within private clouds as well.

The Amazon elastic IP address model demonstrates a basic truth about cloud integration. There are two forms of "component mobility": one that must recognize separate components as discrete elements to be linked into workflows, and one that recognizes successor component copies created by cloud processes, rather than by application processes. Accommodating this combination with standard integration tools (including DevOps or CAMP) is easier if you adopt the principle that the URL is the boundary between logical component movement and physical component location.

Integration tools should be used to bring components together whenever the components are explicitly separate because workflows have to be directed to them individually. The goal of these tools is to direct work to URLs, with the expectation that the URLs will then be matched to a resource location by a separate back-end integration process.

It's possible to manage the address represented by a URL in an integration tool, providing that the tool can be invoked by resource managers when the address of a component changes. The critical issue is not managing the change but managing the impact of the change on in-process transactions. It is very dangerous to allow any stateful flow to change elements mid-stream. This could cause what was once called "tail-ending," wherein someone can ride in on the end of an authentic transaction and inherit the rights of the initiator. Thus it's probably best for stateful workflows to report failures on in-process transactions before changing URL target addresses.

Security and compliance should always be the final item on any integration checklist. Workflows may present stateful-related, tail-ending security and compliance issues, but even the component links can create problems an application security audit might blanch at discovering. Elasticity in component loading multiplies the opportunity to introduce a non-authentic version of a component. As a result, the more integration work is needed in the cloud to insure workflows are sustained through elastic resource use, the more you'll need to examine your component on-boarding processes to insure that you introduce only suitable and authentic elements into your workflows.

COMMON MISTAKES IN CLOUD APPLICATION DEPLOYMENT

Bill Claybrook, New River Marketing Research

There are common mistakes many organizations make when they deploy apps in clouds. The obvious ones usually center on app performance, app security and tools to monitor virtual environments. But there are other common mistakes that occur as well.

The mistakes associated with application deployment in private clouds are more pressing than the challenges of public clouds. IT organizations are responsible for managing the implementation of the private cloud, and today organizations are focusing more on private clouds than public ones.

ORGANIZATIONS FAIL TO DO THE UPFRONT PLANNING REQUIRED TO DETERMINE WHICH APPS ARE GOOD CANDIDATES FOR CLOUD DEPLOYMENT.

Apps requiring big iron, apps running on UNIX clusters, and many legacy apps running on mainframes aren't meant to be moved into the cloud. Some of these apps are complex, and deployment in x86-based hardware and software virtual environments may require re-architecting or rewriting the apps. If the deployment cloud happens to be a public cloud, then apps with high security requirements may not be good candidates.

ORGANIZATIONS FAIL TO PICK THE CORRECT CLOUD MODEL, PRIVATE OR PUBLIC, FOR APP DEPLOYMENT.

Apps can be deployed in private clouds or public clouds. Private clouds are on-premises clouds under the control of the IT organization that created them. They have more similarities with the traditional data center (they are on-premises, under the control of IT, do not have some of the security issues of public clouds, etc.) than do public clouds.

Public clouds are off-premises. The infrastructure of a public cloud is determined by the cloud provider and may present a much different look and feel than the traditional data center, or even a private, on-premises cloud. The mistake that organizations make is failing to determine which apps are good fits for public clouds and which are better suited to private clouds. Another common mistake is failing to determine the costs, long term and short term, in deploying apps in each cloud model.

Organizations tend to focus on "migrating" servers to the cloud versus deploying apps in the cloud.

When organizations decide to move from the traditional data center to a private cloud, the motivation is frequently server consolidation, which leads to improved server utilization and reduction in capital and operating expenses. This should not be the focus. The focus should be on deploying apps in the cloud. By focusing on app deployment, enterprises will gain insight into the makeup of an app and the management tools needed by the app in a cloud environment. This mistake fosters a number of other common mistakes.

FAILING TO PLAN FOR CHANGES IN APP PERFORMANCE IN THE CLOUD

Deploying an app in a cloud may result in a performance level that is lower than in the traditional data center because of the differences between the two. Organization administrators usually focus on CPU power, memory, disk storage, etc. when they think of app performance. In the traditional data center, the app is probably the only app running on a server. The app is tuned on that server to reach an acceptable performance level using physical server monitoring tools.

When an app is deployed in a cloud, it shares physical CPUs, physical memory, etc., on a single virtual host server with several other apps in a virtual environment created by hypervisor software such as VMware ESXi or Xen.

These apps are simultaneously contending for the physical resources of the virtual host server. Performance tuning for the app in the cloud begins in this new ecosystem.

Before an app is deployed in a cloud, you should create a baseline performance for the app that is satisfactory for fulfilling business needs. When the app is deployed in the cloud, you should examine its performance and compare it against the baseline performance, making adjustments until an acceptable performance level in the cloud is reached. To do this type of performance analysis, you will need performance monitoring tools that work in virtual environments.

FAILURE TO UNDERSTAND THAT NEW TOOLS ARE NEEDED TO MONITOR APP PERFORMANCE, SECURITY AND NETWORK TRAFFIC

Some organizations fail to understand that tools that worked in the traditional physical environment are not sufficient for the virtual environment of a cloud. Monitoring tools help answer questions such as: What is the performance of an app? Is an app gaining access to compute and storage bandwidth when needed? What is the response time from storage devices that an app accesses? Is my app being protected from intruders?

Virtualization has added a layer of abstraction to traditional monitoring. You can no longer monitor performance just by looking at physical devices. Network operations teams have struggled to look through this abstraction to determine what is actually happening at both the virtual and physical levels.

Because lots of traffic occurs within a hypervisor without making it to the physical network, you need tools designed to work in virtual environments. Physical-based monitoring tools do not see the traffic flowing among virtual elements such as virtual servers, virtual routers, virtual switches, etc. Monitoring app performance and the performance of resources that surround and interact with an app in a cloud environment requires new tools designed for virtual environments. The same can be said for app security. Tools such as Catbird Network's vSecurity are available for addressing security concerns by monitoring the traffic on virtual networks.

FAILURE TO UNDERSTAND HOW AN APP FITS INTO THE BIG PICTURE OF CLOUD COMPUTING

When an app is deployed in a cloud, just about everything associated with that app is different. The performance is different, the monitoring tools are different, security is different, system management tools used to manage virtual servers are different, and the act of deploying an app is different. These differences place a strain on the organization whose job is to manage the cloud, requiring changes to traditional processes for deploying and managing apps in a cloud environment.

Cloud vendor selection usually implies an infrastructure and ecosystem that can have a large effect on the deployment of apps in a cloud. Proper selection of vendor(s) and virtualization software, such as hypervisors, involves understanding how an app fits into the big picture of cloud computing, and determines, to a great extent, whether or not you can move apps between private and public clouds to take advantage of the hybrid cloud model.

FREE RESOURCES FOR TECHNOLOGY PROFESSIONALS

TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific Web sites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more —drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily. Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

WHAT MAKES TECHTARGET UNIQUE?

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise IT professionals across all industries and markets.