GUIDE TO PUBLIC CLOUD ADOPTION IN THE ENTERPRISE

CLOUD-A

GEOFF SULLIVAN

LAST UPDATED 06.15.2015

TABLE OF CONTENTS

CLOUD-A'S ENCLOUD-A'S ENTERPRISE STORYTERPRISE STORY

In the beginning, most of Cloud-A's client base were young, small, innovative and nimble companies who were looking for a production ready public cloud environment to commercialize their SaaS products. As time goes on we are seeing more and more enterprises coming on board, usually starting out with a small project to prove the concept of building scalable applications on the public cloud. Not by design, we were seeing our products being used in "shadow IT," where a particular developer wanted to build an application on Cloud-A and knew that the process to get the project approved would be too tough a task to take on.

In an effort to reduce the use of Cloud-A in "shadow IT," and promote the use of public cloud in the enterprise we've prepared this white paper to help enterprises adopt true cloud technology, understand the considerations to make, how to maximize business agility and make recommendations for successful adoption of the new technology .

DEVELOPER ADOPTION IS KEY

There may be cases to be made for moving all or some of your existing applications to a public cloud like Cloud-A, but the true value is in the "greenfield" where your developers can create modern, scalable software and bring it to market faster on the new, agile cloud environment.

Achieving buy-in from your development team is the key to a successful voyage into modern cloud technology. It is important to highlight the benefits of this technology in terms that make sense to developers. Technology like OpenStack empowers developers with the right toolsets like a self-serve model, where they are enabled to rapidly spin up their own environments on demand.

ACCELERATE NON-PRODUCTION ACTIVITY

The agile self-serve, rapid deployment model that cloud technology like Cloud-A provides allows developers to spin up non-production environments like test, dev and QA on demand to accelerate these activities. Many fortune 500 companies like Ebay and Paypal have moved to a 100% agile cloud model for all of their test, development and QA activities. Learn how development teams are using Cloud-A for continuous integration here.

CONSISTENT PLATFORM FOR TEST, DEV AND QA

Rather than performing development and testing on internal, physical systems and launching to production on different but more powerful and resilient platform, modern cloud environments like Cloud-A empower developers to perform development, testing and QA on the same technology as production for a more consistent product. The utility billing model of public clouds also allows developers to build up and tear down these environments on demand and only pay for them when they are in use. Learn more about the cost benefits of doing test/dev/QA on Cloud-A here.

APPLICATION CONSIDERATIONS

Anytime an organization is considering moving away from their legacy infrastructure to modern cloud technology, analyzing the catalogue of existing applications is a crucial initial step. Application architecture, age, function and data type are just some of the things that need to be taken into consideration for a successful migration.

SELECTING THE "LOW HANGING FRUIT" FOR MIGRATION

It is essential to identify the "low hanging fruit" of existing applications for migrating to a modern cloud platform like Cloud-A. Dave Pitzely, Director of Architecture of Comcast's Private OpenStack Cloud noted recently at the OpenStack Summit that while vertically scaled applications like ERP, or financial systems can be moved to a modern cloud platform, they are likely not your best bet to start. Pitzely stated that these applications tend to take time to migrate, may not perform as well as they did on their legacy environment, may have reliability issues, and the value yield will be low.

It is important to focus on existing applications that are highly transactional such as data analytics, event mediation or applications with advertisement and click through data. These applications are highly variable and highly scalable and lend themselves to the API driven infrastructure of a modern cloud platform, thus producing a high yield of value. Learn more about how the application "Lift and Shift" to the cloud technique fails and some best practises for refactoring your apps for the cloud here.

Another consideration to make when selecting applications to migrate to a modern cloud platform is application lifecycle. Applications that are approaching end of life are not likely candidates for migration, and are probably better suited to stay on the legacy infrastructure until their end of life. Conversely, newer applications that have been built on legacy infrastructure, but have the qualities of modern architecture (web front ends, highly variable and scalable) are not only better candidates for

migration, but they will likely perform better, become more reliable and available on a modern cloud infrastructure.

DIFFERENCES IN APPLICATION ARCHITECTURE

Traditionally architected applications and "cloud-aware" applications are architected differently. Traditional applications tend to be monolithic, centralized, have tightly coupled components, are synchronous and typically single tenancy, whereas "cloud-aware" applications have distributed microservices, are asynchronous, multi-tenant, have built-in failure-resistance, have decoupled components and are eventually consistent.

The characteristics of "cloud-aware" applications gives them functionality that traditional applications simply cannot compete with. One example of this is "Cloud-aware" applications can be architected with redundancies and resilience built into them and leverage the APIs provided by the modern cloud infrastructure, whereas traditional, monolithic applications tend to rely on extremely expensive hardware for all of their reliability and uptime. learn about existing DevOps tools that help automate deployments and management of modern cloud applications here.

APPLICATION MIGRATION STRATEGIES

LIFT AND SHIFT

As previously mentioned, the Lift and Shift approach includes moving an application to the public cloud without any significant changes in the architecture of the application itself. Typically an organization will select a new cloud host virtual machine(s) based on the workload requirements of the application. The main benefit of the lift and shift strategy is cost savings. Because this strategy avoids large amounts of development labour, your costs are limited to migration and testing on the new platform. The downside of the Lift and Shift technique is that the application is not likely optimized for the public cloud and therefore you miss out on the ability to scale the application and associated costs up or down on-demand, based on load requirements. In addition to this, traditionally architected applications are often tightly coupled which can lead to performance, reliability and security complications in the public cloud.

HYBRID PARTITIONING

Just as the components of an application can be partitioned and span multiple internal cloud platforms, components of applications can span a combinations private and public cloud platforms. This can usually be achieved with minimal modification to the application if it was architected with decoupled components, thus reducing risk and cost. If one component of your application is highly scalable and can benefit from the on-demand elasticity and utility billing of the public cloud, but another component is better suited in your private cloud environment, the

application can be partitioned to benefit from the best of both, public and private cloud.

In order to successfully partition an application and deploy components in both public and private cloud, a deep understanding of the architecture and behavior of the application is required. While this method will require more modification and testing than lift and shift, the costs benefits of deploying the right workload on the right cloud platform may very well outweigh the cost of modification.

REFACTOR

Refactoring an application includes rewriting all or most of an application to take advantage of the features of public cloud like API driven infrastructure, elastic scaling and utility billing. You will usually want to take a look at all of the underlying layers of cloud infrastructure to identify how best to utilize them. This includes the hypervisor, the underlying resources like storage and data and provisioning and tenant management.

Refactoring an application provides the opportunity to get more performance out of your application. Refactored applications that are optimized for cloud environments can have the ability to access the infrastructure APIs of a public cloud to make more efficient use of the underlying cloud resources, which can result in an overall infrastructure cost reduction.

The obvious downside to refactoring an application for the public cloud is the cost associated with development work. This cost, however, can be dramatically less than if you lift and shift a legacy application to the public cloud and do not use your public cloud resources effectively. The accumulated cost of moving and managing an inefficient, legacy application in the public cloud is usually much higher than the cost of refactoring an application in the first place.

HANDLING INTERNAL POLITICS

As with any department in a large organization, IT will have hurdles to overcome when making a change like moving to a modern cloud platform like a private OpenStack deployment. While this discussion was focused on private cloud deployments, organizations will likely also come up against internal resistance to adopting public cloud technology like Cloud-A

BUDGETARY CONSIDERATIONS

The way that budgets are determined and allocated for funding a project in the public cloud is vastly different than budgeting for internal infrastructure. Because of this, organizations may come up against resistance from their purchasing teams. Utility billing, while very convenient, in most cases less expensive, is difficult to

predict, and there is less ability for purchasing teams to negotiate pricing with a public cloud vendor as they would with a hardware reseller. In addition to all of this, purchasing departments are also accustomed to up front fees hardware and licensing fees, and long, drawn out RFP cycles. More about writing an RFQ/RFP to procure public cloud infrastructure here.

The other problem that modern cloud technology can create with the financial decision makers in an organization is the over provisioning of infrastructure. Because infrastructure can be rapidly deployed on demand on a cloud platform like Cloud-A, sometimes developers feel empowered to deploy an unnecessary amount of resources, which can be expensive. The panel recommended either a bill back model, where an internal department is billed for their infrastructure, or a show back model, where an internal department has regular visibility of their usage, both which can make individual departments in an organization accountable for the infrastructure that they deploy.

EXECUTIVE BUY-IN

Receiving buy-in from the C-level prior to taking on a new, disruptive project by being transparent and highlighting and measuring the business value of modern cloud technology like business agility, innovation and faster time to market, which often times, at scale, can far outweigh the costs associated with cloud infrastructure.

MOVING FORWARD WITH CLOUD-A

Over the next several months Cloud-A is focusing on enhancing our user experience to enable development team adoption of our modern cloud infrastructure. We have been working with existing users, putting on focus groups and releasing surveys to help understand the the path of least resistance for enterprises adopting our technology. Stay tuned for some exciting announcements.

FOR MORE INFORMATION CONTACT: GEOFF SULLIVAN
GEOFF@CLOUDA.CA