Migrating Applications to Public Cloud Services: Roadmap for Success

December 17, 2013

Contents

Acknowledgements

The major contributors to this whitepaper are: Claude Baudoin (cébé IT & Knowledge Management), Chris Carlson (Retriever Consulting), Giuseppina Cretella (Second University of Naples), Beniamino Di Martino (Second University of Naples), Mike Edwards (IBM), Shamun Mahmud (DLT Solutions), John McDonald (CloudOne), John Meegan (IBM), Sujatha Perepa (IBM), Keith Prabhu (Confidis), Ram Ravishankar (IBM), Michael Salsburg (Unisys), Muralidhar Seelam (IBM) and Joe Talik (AT&T).

Executive Overview

Across all industries, momentum is building to migrate applications to cloud computing. While cost savings, speed of deployment and scalability top the list of business motivations, an increasing number of enterprises view cloud computing as a key enabler of business transformation that can help improve customer engagement, forge new partnerships and drive competitive advantage.

However, the migration of applications to cloud computing must be done in a strategic and methodical manner. Existing enterprise applications must be thoroughly assessed to determine which workloads can benefit most from early migration to the cloud. Key considerations including costs of migration, application redesign, application performance and availability, security and privacy requirements, and regulatory requirements must be taken into account.

An enterprise strategy for cloud computing must identify individual business problems with existing applications that cloud computing can potentially address and provide specific business justification that the cloud is the right strategic alternative. A business case for migrating applications to the cloud must describe the current state and demonstrate the advantages of cloud computing to not only reduce costs but also deliver meaningful business value.

In most cases, starting small and expanding after initial success has been proven as the most prudent approach to application migration to the cloud. It often makes sense to start with the most cloud-ready applications—those with minimal customer data and other sensitive information—or applications that derive the most immediate advantage from the cloud's elasticity. Create a pilot for one or two of these types of applications, test thoroughly, and gather customer feedback so that improvements can be made before going live. Once the application migration processes have been proven and required cloud computing skills have been developed, migration of more business-critical applications can be considered.

The aim of this guide is to provide a practical reference to help enterprise information technology (IT) and business decision makers analyze and consider application migration to cloud computing. *The paper focuses primarily on the migration of applications to public cloud services*. It includes a list of steps, along with guidance and strategies, that takes into consideration both business and technical requirements.

The section titled "Motivation and Considerations" provides an overview of the potential impact that the migration of enterprise applications to cloud computing will have on new and existing business processes. This section provides guidance on the types of applications that are best suited for migration to the cloud.

The section titled "Migration Roadmap" is the heart of the guide and includes the basic steps of a formalized migration process. It details both strategic and tactical activities for decision makers to develop a business plan and detailed migration plan.

Motivation and Considerations

The business community might have the following motivations for the migration of an application to the cloud:

- Broader reach
- Easier mobile access
- Business agility and flexibility
- Improved security
- Improved responsiveness
- Better analytics on application usage
- Improved availability
- Reduced and/or re-allocated costs

Most of these objectives are well matched by the some key cloud computing characteristics:

- *Rapid elasticity*. The ability to rapidly scale the IT infrastructure up or down to match changing requirements, on a pay-per-use basis, is extremely attractive to large and small organizations alike. A cloud computing environment offers increased resources which can lead to performance improvements for certain applications. Applications that are designed to spread their workload across multiple servers will be able to benefit from automated scaling of resources to match the current demand. This is especially appealing for applications with unpredictable or cyclical usage patterns, because a cloud orchestration tool can monitor usage and can dynamically scale resources up or down. This behavior combined with the pay-by-usage characteristic of a cloud can lead to significant financial savings.
- *Pay-as-you-go versus install-and-own*. The shift in up-front capital requirements from the customer to the service provider is equally attractive. In particular, small organizations and start-ups face much lower infrastructure costs than were necessary pre-cloud.
- *Organization streamlining*. Buying capabilities such as "security as a service," "collaboration as a service," "communication as a service," etc., decreases the need for specialized in-house IT skills and can remove some of IT's greatest non-value-adding challenges.

The main concerns when thinking about cloud computing are:

- *Security*. Moving your data and code to a third party provider creates some security risks. Although the technology to make cloud computing safe is available, securing cloud workloads often requires new concepts and skills that may take time to acquire.
- *Loss of control*. For software-as-a-service (SaaS) and some platform-as-a-service (PaaS) solutions, the entire control of hardware, software, security policies, etc, is placed in the hands of a third party provider.
- *Integration*. Most customers will need to integrate internal systems with cloud systems. Having these two systems communicate with each other is always a challenge.
- *Availability and reliability of cloud applications*. Issues may arise from a combination of server performance, configuration errors, network design, and application architecture, possibly in combination, which can initially make them difficult to resolve.

- *Cloud service provider lock-in*. The concern is that once a cloud service of one provider is adopted, it will not be easy to switch to using an equivalent cloud service of a different provider. Emerging standards will increase the portability and interoperability of systems across cloud service providers, and will reduce or eliminate this current barrier to cloud adoption.

To prioritize applications for migration to cloud computing, it is necessary to first identify and understand the business and technical factors for the migration.

Migration Roadmap

As customers transition their applications and data to cloud computing, it is important that the level of service provided in the cloud environment be comparable to the service provided by their traditional IT environment. Failure to properly migrate applications to cloud computing could ultimately result in higher costs and potential loss of business, thus canceling any of the potential benefits of cloud computing.

This section provides a prescriptive series of steps end users should take to ensure successful migration of existing applications to cloud computing:

1. Assess your Applications and Workloads
2. Build the Business Case
3. Develop the Technical Approach
4. Adopt a Flexible Integration Model
5. Address Security and Privacy Requirements
6. Manage the Migration

Requirements and best practices are highlighted for each step in the sections that follow.

Step 1: Assess your Applications and Workloads

Assessing applications and workloads for cloud readiness allows organizations to determine what applications and data can – and cannot – be readily moved to a cloud environment and what delivery models (public, private, or hybrid) can be supported. It often makes sense to start with the lowest-risk applications—those with minimal customer data and other sensitive information—or applications that take advantage of the cloud's elasticity. Alternatively, you might start by determining which applications you do not want to move to the cloud initially. The decision criteria may be refined as the assessment progresses.

Table 1 highlights suitable and less suitable types of applications for migration to cloud computing.

Suitable Candidates for Cloud	Less Suitable Candidates for Cloud
• Applications that are used by a group of mobile workers to manage their time and activity, and that contribute only limited information to the company's broad management information databases. • Applications that are run infrequently but require significant computing resources when they run. • Applications that are run in a time zone different from that where your company's IT personnel are located. • Development, testing and prototyping of application changes, even if the final applications will be run on your own infrastructure. • Service Oriented Architecture (SOA) applications.	• Applications that involve extremely sensitive data, particularly where there is a regulatory or legal risk involved in any disclosure. These will at minimum require special treatment if they are to be run in a cloud service. • Applications now being run on the company's private network and that are very performance-sensitive. • Applications that require frequent and/or voluminous transactions against an on-premises database that cannot be migrated to cloud computing. • Applications that run on legacy platforms that are typically not supported (or may not be supported in the long run) by cloud providers.

Table 1. Application Candidates for Migration to Cloud Computing

It often makes sense to start with the lowest-risk applications—those with minimal customer data and other sensitive information—or applications that take advantage of the elasticity of cloud computing. Alternatively, you might start by determining which applications you do not want to move to cloud computing initially. Assessing applications and workloads for cloud computing readiness allows organizations to determine what applications and data can – and cannot – be readily moved to a cloud environment and what delivery models (public, private, or hybrid) can be supported. The decision criteria may be refined as the assessment progresses.

The readiness assessment spans the following areas:

- *Business Considerations*. Business considerations include the overall organizational readiness for using cloud computing. Is the application owner willing and comfortable with a cloud platform? How important is the application to the business or the mission? What is the risk tolerance level of the business, and is the culture favorable or resistant to change?
- *Application Lifecycle Considerations*. Is the application still being defined? Is it up for a refresh? Is the application approaching retirement? Can the application be redesigned or undergo a technology refresh for cloud computing? Will there be an efficiency gain in using cloud computing? Instead of migrating the existing application to cloud computing, using an IaaS or PaaS approach, would it be better to replace it with a new SaaS solution?
- *Application Architecture Considerations*. Is the application web-based, or built with a service-oriented architecture (SOA)? If not, is the application such that it can be split into modular services? Is it monolithic, two-tier, three-tier, or n-tier? What is the level of effort required to modularize it or separate the tiers? Does the application scale out? Does it scale up? What are the demand fluctuations in the application? What impact will moving to cloud computing have on demand?

- *Data Considerations*. Data governance, confidentiality, integrity and quality need to be preserved by the migration. Is the data bound by statutory compliance? Are there data sensitivity and privacy or confidentiality concerns? What data integrity concerns are there? How does the application manage data requests from a safety and security perspective? How much data exchange will occur between the components of the application and between the application and the user? Frequent data transfers may impose a higher cost as well as a performance lag.
- *Technology Considerations*. These include the performance and resiliency of the network infrastructure. The migration design must account for multiple components communicating across network boundaries. Techniques such as network isolation, virtual private networks, elastic addressing and network segmentation can provide for a very robust and secure cloud environment. Ensure the application is designed (or can be modified) for resiliency—immunity to the interruption of transactions in midstream, as well as local fault tolerance. Is the application designed for high availability and disaster recovery? Finally, standard and open protocols are more readily supported across firewalls and on a public infrastructure than proprietary ones.
- *Security Considerations*. The different parties – application owner, cloud service provider(s) and the customer's IT department – must understand that security is their joint responsibility. Authentication and authorization remain the responsibility of the customer at the application level. The cloud service provider is responsible for security controls, identification and correction of system vulnerabilities, and defense against specific cloud-oriented attacks (e.g., at the virtual machine level), consistent with the level of service selected. Continuous monitoring is now common among cloud service providers and should be expected.
- *Integration Considerations*. What are the dependencies between the application being migrated and other systems? Applications may depend on each other through control integration (they invoke each other), data integration (they read or write the same databases or files), or presentation integration (they are mashed up on the same window or Web page). The migrated application may even be the "system of record" for some key data in a Master Data Management (MDM) scheme. Finally, the migrated application may rely on common facilities such as a user directory for single sign-on and access control. The assessment must discover how extensive these integrations are, what protocols they use, what additional utilities or runtime libraries they rely on, and what their performance requirements are, including the frequency of connections and the amount of data involved.

Appendix A provides examples of some of the most cloud-ready workloads and their benefits.

Step 2: Build a Business Case

Developing a business case for migrating applications to cloud computing requires an overall cloud computing strategy and specific information that describes the current state and demonstrates the advantages of cloud computing to not only reduce costs but to deliver meaningful business value. High level value propositions for cloud computing, including the shift of capital expenditures (CAPEX) to operational expenses (OPEX), cost savings, faster speed of deployment, elasticity, etc., are necessary but

insufficient unless quantified. Within the context of an enterprise strategy for cloud computing, individual business problems with existing applications that cloud computing can potentially address need to be identified, and specific business justification must prove that cloud computing is the right strategic alternative. Refer to the CSCC *Practical Guide to Cloud Computing* for specific considerations that need to be taken into account when developing an enterprise strategy for cloud computing. [1]

Cost Analysis

Once an application is identified as a potential candidate for migration to cloud computing, a thorough cost analysis must be performed. In order for meaningful comparisons to be made, one must have specific baseline costs for the current environment. The overall cost of application migration to cloud computing must include the following elements:

- *On-going cloud service costs*. The cloud service provider fees must be taken into account, including the effects of variable demand, such as extra fees to handle peak loads.
- *Service management*. Managing services and service providers is a skill that is often not well developed in firms, and yet it is of critical importance to the success of cloud computing.
- *License management*. It is important to understand third-party software dependencies and impact to licensing contracts (and ongoing management of these licenses) when migrating an application to cloud computing.
- *Application re-designs*. The application may require design changes in order to be compatible with or to take full advantage of cloud deployment.
- *Application deployment and testing*. Once the application has gone through design changes it must be configured, deployed, and tested in the cloud environment.
- *Application maintenance and administration*. Ongoing maintenance and administration of the cloud-based application will remain the client's responsibility.
- *Application integration*. In many cases, there will be a need for connections between the migrated application and applications and data that remain in-house, potentially requiring new integration software.
- *Cost of developing cloud skills*. Internal personnel may need to be trained to support the migration to cloud computing.
- *Human resources and talent management implications*. New skills and abilities such as preparation and deployment of virtual machine images may require changes to supervisory, control and compensation systems. Job descriptions, bonus plans, etc. may change.

Service Levels

In addition to assessing the costs of application migration, it is equally important to ensure that the level of service provided by the cloud-based application will be comparable to current service levels. The required service levels should be agreed with the cloud service provider and explicitly documented in the cloud service agreement. In fact, the service levels provided by an internal IT department to its business customers are often not well specified, or not specified at all. Migrating an application to cloud computing places a spotlight on those essential commitments. Refer to the CSCC *Practical Guide to Cloud SLAs* and the CSCC *Public Cloud Service Agreements: What to Expect and What to Negotiate* for

specific considerations that need to be taken into account when developing an enterprise strategy for cloud computing. [2,3]

For each application being migrated to cloud computing, consider the following application characteristics:

- *Application availability*. The criticality of the application to business operations will determine the availability requirements that must be clearly specified in the cloud SLA.
- *Application performance*. Depending on the performance requirements of the application, specific performance targets may need to be achievable with the cloud service.
- *Application security*. Moving an application to the cloud will require due diligence on the part of the cloud service customer to ensure proper security controls are in place and operating effectively.
- *Privacy*. Personally Identifiable Information (PII) handled by a cloud-based application must be properly stored and maintained. Access to PII stored in a cloud service must be restricted as required, including from cloud service provider personnel.
- *Regulatory compliance*. Government and industry regulations may require additional measures, such as restricting the migrated applications and data to reside in a specific geographic region.

Business Impact

Assuming the cost and service analyses described above are favorable, then additional business factors must be weighed in order to develop a complete business analysis, and should be *monitored* on an ongoing basis:

- *Revenue impact*. If the application is used to generate revenue, is the move to cloud computing expected to increase that revenue?
- *Customer acquisition or engagement impact*. For a customer-facing application, is the move to cloud computing expected to increase the number of customers accessing it?
- *User satisfaction*. Does one expect an improvement in availability or response times that will result in increased user satisfaction?
- *Time to market improvements.* Will the move to cloud computing shorten the time it takes to deliver functional enhancements to end users?
- *Cost of handling peak loads*. The cost of scaling server capacity up and down to match spikes in demand for the cloud-based application should be compared with similar costs before migration.

Obtaining executive support for the initiative is critical. Executives from IT, Lines of Business (LOBs), procurement and executive management must review and approve the business plan before proceeding. Getting key executives on-board early in the process will help alleviate potential issues down the line.

Step 3: Develop a Technical Approach

Broadly speaking, there are two potential target service models for the migration of an existing application – Infrastructure as a Service (IaaS) and Platform as a Service (PaaS).

PaaS Migration

To use a PaaS cloud service as the target for migration, the application itself must be designed for one or more runtime environments available in the target PaaS service. An example of such an application is one where the business logic is implemented as a set of components which run on an application server (such as IBM's WebSphere, Oracle's WebLogic, or the JBoss server) in combination with a database (such as the Oracle database or IBM's DB2) containing the application's data and also possibly associated code in the form of stored procedures.

In general, a PaaS solution must provide the elements of the particular software stack required by applications such as the operating system, an application server and a database, so that the customer only has to be concerned with the specific application components and data. One must also ensure that the PaaS environment offers the configuration(s) required by the application. This may include software levels, the ability to run scripts, and the presence of certain tools for setup, reporting, monitoring, etc., identical or similar to those present before migration.

IaaS Migration

To migrate an application to an IaaS service, the requirements on the cloud service itself tend to be lower. The entire software stack is migrated: the application code itself, plus any supporting code it requires – including the underlying operating system. To achieve this, it must be possible to package the complete software stack as one or more virtual machine (VM) images, which can then be copied into the cloud service and executed there.

Whether the software stack involved will work in a virtual machine environment may depend on whether there is use of specialized device drivers or hardware devices that are unlikely to be supported by an IaaS provider; an application depending on these capabilities is not a good candidate for migration. This can be tested by preparing the virtual machine image of the application and of the supporting code and attempting to execute it on a trial VM environment (either in-house or a test system offered by the cloud service provider). Hidden dependencies can thus be found, corrected, and the process repeated until successful or until it is determined that there is no affordable solution.

Common Technical Considerations

In both the PaaS and IaaS cases, the following technical considerations must be taken into account:

- *Skills*. The organization needs to possess, or to be able to acquire, the skills necessary to prepare and migrate the application components. The preparation of virtual machine images and their deployment to a cloud service may involve skills new to the organization, and this must be considered in the project plan.
- *Security*. Ensuring appropriate security of the application and its data is vital. The cloud service's security features may be very different from those of the in-house environment, and the

security risks and the measures applied to counter them must be assessed carefully. For example, if the application is only used by staff working for the organization, then it may be wise to place the application within a Virtual Private Network (VPN) in the cloud service, providing secured access for staff while preventing access from outsiders over the internet. Data that is migrated to a cloud service needs similar technical decisions – can the data be stored in clear text, or does it need to be encrypted, even at rest, to reduce the risk of exposure or theft?

Other technical choices depend on the security measures applied by the cloud provider to the cloud service. Does the provider implement strong user authentication techniques for the service? Does the provider offer security tools or services, for example to implement encryption of data in transit or at rest?

- *Integration*. Integration with other applications and services within the customer organization, which may be bidirectional, may involve configuration changes (e.g., to reflect new addresses), new authentication methods, and other technical changes to avoid network latency and throughput issues that may impact the performance of the application or its dependent systems. This is addressed in greater depth in Step 4. Note that the current absence of integration or interoperability between two applications does not mean that such a need will not arise later as the result of a change in business requirements.
- *Monitoring and Management*. The monitoring and management of the application running in the cloud service must be considered. Can in-house tools still be used, or is it necessary to adapt to new monitoring and management facilities supplied by the cloud service? Monitoring resource usage by the application is important, since undetected high usage is likely to inflate the cloud service usage fees.
- *Scalability*. While scalability is a common advantage of cloud services, applications have to be structured appropriately to take advantage of scalable cloud resources, and this may require changes to the application code. In particular, the challenge of reprogramming an application to use multiple processors or multiple machines in parallel can be significant.
- *Availability and Backup*. In-house designs to support the availability of the application may need significant adaptation to deal with the cloud service environment, especially for PaaS services. Backup processes for the application may need to be adapted to the environment of the cloud service.

These technical considerations must feed back into the business case for migration and possibly may call into question its very feasibility. The technical issues also feed into the migration plan (refer to Step 6 for details).

Patterns

One approach that may help with the migration of applications to cloud computing is the use of *patterns*. Patterns describe common aspects of cloud computing environments and of application design for cloud computing. Some patterns can be useful in understanding the appropriate organization of the software stacks on which applications depend. Patterns can also be useful in understanding what

changes may be necessary to the application code for successful migration to the cloud computing environment.

Some of the general patterns which apply to cloud computing are described in the paper "A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures." [4] Patterns can also be specific to a cloud computing platform like those provided by Amazon, Microsoft, IBM, and others. [5] [6] [7]

Step 4: Adopt a Flexible Integration Model

It is common for an application being migrated to a cloud service to have connections of various kinds with other applications and systems; therefore, the application owners need to understand the impact of these connections and address it.

Integration between applications is typically classified into three types:

- Process (or control) integration, where an application invokes another in order to execute a certain workflow
- Data integration, where applications share common data, or one application's output becomes another application's input
- Presentation integration, where multiple applications present their results simultaneously to a user through a dashboard or mashup.

The purpose of these integrations may be to perform an end-to-end workflow that crosses the boundaries between multiple business capabilities or systems (for example, entering a transaction in an accounts receivable system when a customer places an order in an e-commerce application). Another form of integration is when the migrated application must continue to be monitored and managed by an existing suite of on-premises IT tools.

In many cases, the challenge is not so much "integration" as it is "re-integration" or "maintaining the integration" between pieces of the entire system that are coupled in a certain way.

The first task in addressing integration is to inventory the connections or "integration points" in question. Below are two examples. Note that a variety of approaches may be used to address the challenges, and there is usually not a single one that works in all cases. Therefore, we recommend that integration approaches:

- be flexible, potentially including several different techniques according to specific situations;
- be based on standards, in order to be more maintainable and less fragile with respect to changes the cloud provider might make;
- consider the possibility that more migrations may occur in the future – therefore cloud migration is an opportunity to modernize the architecture and render it more resilient to such changes.

Example 1. One application may invoke another in order to execute a certain business process. For example, at a company that has automated the processing of vendor invoices, an application used to receive goods on the delivery dock may communicate with an ERP application to check that the received goods were ordered and, if so, to record the delivery, which then schedules the payment of the amount specified in the invoice received electronically from the vendor.

Consider what happens if the receiving application is migrated to a cloud service, and the ERP application remains in-house. What protocols do these applications use to talk to each other? Can the same connection continue working, that is, is the protocol supported by the cloud provider, and if it is, does the response time remain adequate, given the frequency of these transactions, the size of the payload, and the bandwidth of the connection to the cloud service?

In this situation, the architects and developers have several options:

- The existing communication between applications may work perfectly well, both in terms of the communication protocol being supported across the Internet, but also in terms of performance.
- The connection may need to be slightly redesigned to use a standard protocol that the cloud-based system supports, or to streamline the "payload" of the messages in order to avoid performance issues.
- An on-premises mirror of certain key information (in the above example, a list of purchase order numbers and line items, including vendor part numbers and quantities) may need to serve as a "proxy cache" for the application now located in the cloud, so that the real-time responsiveness of the on-premises application is not affected (it might even improve!). This solution has significant implications in turn: the mirror needs to be kept up to date at a suitable frequency, the application may still need to be modified to access the mirrored data, etc.
- A more substantial architectural change may be required. This may in turn take several forms:
    1. Using SOA to replace a legacy architecture
    2. Using an established message-based integration approach, such as an Enterprise Service Bus (ESB), that is compatible with Internet-based communications. In fact, extending the concept of service bus so that it can be used between on-premises and cloud-based systems is a recent trend followed by multiple providers.
    3. Using special cloud integration solutions, typically based on customizable templates, specifically designed to connect cloud solutions to internal enterprise applications. Refer to step 7 of the CSCC whitepaper *Convergence of Social, Mobile and Cloud: 7 Steps to Ensure Success* for more details. [8]

If we generalize this example, the customer's systems, including the ones targeted for migration to cloud computing as well as those that will remain on-premises, often have many connections. In the worst case, N applications each connect to the other N-1, creating N x (N-1) associations. In such a case, migrating one or more applications to use cloud computing presents a challenge whose complexity depends on the way in which the applications connect:

- If a proprietary message queuing protocol is used, that protocol may not be supported (or may run into performance issues) across the connection between the enterprise's network and the cloud provider's systems.
- If an Enterprise Service Bus (ESB) is used, the issue may be less acute, as an ESB is designed to hide the protocol differences across different links. It is therefore possible to replace the protocol used between the migrated application and the ESB with another one, without requiring a change in any of the other applications. Only the migrated application may need to be tweaked.

A Service Oriented Architecture (SOA) does not remove this problem. With SOA, each application exposes some services, the connections use Web protocols, but the N x (N-1) connection issue remains, therefore each application that connects to the migrated one may be impacted by the migration.

In this situation, short of concluding that the challenge is too complex or costly to solve, there are again several options:

- One is to "bite the bullet" and modify each integration point individually, as done in the first example. This will probably be costly, and may not yield a solution that will be significantly better next time there is another application migration to undertake.
- One is to actually migrate *more* systems than initially planned – that is, move the entire "spaghetti" of integrated applications to cloud computing. The connections between the applications, which used to be local to the customer's data center, are now local to the cloud service, and as a result they are less likely to be impeded by firewall rules or performance issues. If the configuration of the cloud provider's systems is different enough from the legacy environment, some changes may still be needed for the migrated collection of applications to work.
- Finally, a smart but potentially difficult solution is to eliminate the "technical debt" accumulated with the old architecture and move to an ESB or equivalent. The new architecture incorporates from the start the requirement that some of the systems connected to the service bus, or some of the applications providing the Web services, may be cloud-based while others are on-premises. This would in general be the recommended approach, unless it brings performance problems.

Example 2. Multiple applications, some located in cloud services and some on-premises, may depend on shared master data. For example, an Accounts Receivables application, a CRM application, and a Call Center application all require an authoritative list of clients. Here, the ease of migrating to use cloud computing depends on whether there is a Master Data Management (MDM) initiative in place:

- Without an MDM strategy, each application has its own separate client database, updated manually. This is not a desirable situation in terms of data quality and end-to-end business process integration, but on the other hand, migrating one or more of these applications (with their associated client databases) to a cloud service will not be a problem.
- Organizations that have implemented MDM have either chosen one of these applications to hold the master customer data, or implemented a separate "system of record" for customer data. Each time a transaction (human or automated) submits a customer name or number, the application invokes the system or record to verify the information and to obtain the other customer data attributes it needs to do its work. In that case, moving some of the applications to the cloud may be problematic: the frequency of invocations may be high, the protocol used (it might be direct reads from an SQL database, or a file transfer protocol for replication) may not be supported between the on-premises system and the cloud environment, or it may be too slow across the Internet.

To address this MDM scenario, there are some possibilities. One approach is to place a partial cache of the master data on the same system (or the same local network) where the migrated application will reside, and synchronize the cache with the master as the network bandwidth permits, or even through a daily batch refresh (taking into account how current the cache needs to be in order for business to be transacted as required). Another solution is to use stored procedures, which reduce the interactions with a central relational database, as well as the amount of data sent across the network. Stored

procedures may also address some security issues by keeping sensitive data within an on-premises database server.

Step 5: Address Security & Privacy Requirements

Security and privacy are two of the issues that concern cloud service customers the most. Depending on the sector, these may be just above or below concerns about availability and performance as highest priority. At the same time, cloud service customers should remember that many of the security and privacy concerns raised by cloud computing have existed since the first forms of IT outsourcing were introduced.

Security involves multiple concerns. It includes such aspects as:

- How hard is it for an intruder to steal confidential data from the cloud provider's systems (external threat)?
- If this happens, will you even know it?
- Can you trust the provider's personnel, especially system administrators who have many privileges over the systems you use (internal threat)?
- What does the SLA promise you in terms of security measures?
- What is the impact on your business of a denial-of-service attack, which may not endanger your data but prevents users from accessing the application?
- How do you authorize an employee to access a system or application in the cloud? How do you *unauthorize* them, and how quickly can you do that in a serious situation (termination for cause, etc.)? What levels of trust do you grant different users, and how do you identify and authenticate trusted users?
- How do you prove to a client or an auditor that adequate security measures are in place, now that this is not only your problem, but a shared responsibility between you and a cloud provider?
- How can you verify that the virtualization platform or cloud management software running on the systems you use, which you did not install and do not control, does not contain malware?
- How can you protect yourself from malware that could be introduced by another customer in a multi-tenant environment?
- What is the risk that your data will be delivered to a domestic or foreign law enforcement agency by the cloud service provider in response to a legally binding request?

The reader will find more extensive treatment of these issues in the CSCC's white paper entitled *Security for Cloud Computing: 10 Steps to Ensure Success.* [9]

Privacy is closely related to security, but it carries with it the additional burden that a violation of privacy, for example the disclosure of PII about your own users or customers to people who do not have a right to access it, will cause major damage to your company, including:

- Loss of business
- Legal action by the people whose information has been disclosed
- Non-compliance with government regulations

In addition, data subjects may have rights to inspect and correct PII that relates to them, which will need to be supported by the application even when it runs in a cloud service.

Now that we have examined all the risks and threats that arise when migrating an application to the cloud, it turns out that doing so may, in fact, *increase* its security. This statement is based on two facts:

- Cloud service providers have, in all likelihood, more expert resources at their disposal than many of their customers. They need to make that investment because a successful attack could damage their entire business. Therefore, customer data may be safer in the cloud provider's custody than in customer's in-house systems. This is the same principle that leads people to rent a safe deposit box at their bank rather than keeping their valuables at home.
- Once your data is held in a cloud service, an attacker who specifically wants to gain access to your information no longer knows exactly where to attack. Even if they successfully penetrate the network of the cloud provider, there may be thousands of virtual servers whose names do not reveal whose data they contain.

Knowing all the above, here are some logical steps to follow (again, see the "Security in the Cloud" white paper for more information). Note that as a result of performing these tasks, you will never be 100% protected, and after a risk analysis you may even end up deciding that you cannot in fact migrate certain data or applications. But they will certainly increase the chances of success.

1. Understand exactly what data (including what code, since code may be the confidential asset to protect) will be migrated to the cloud service.
2. Map this data to your security classification. If a security classification does not exist, or if it does not specify where and in which format (cleartext vs. encrypted) data may be held on the basis of its classification, this is an issue that must be resolved.
3. Identify which information raises privacy concerns – for example, account numbers, dates of birth, addresses, etc.
4. Examine applicable regulations (especially in the finance and health domains) and determine what needs to be done to meet these regulations, and whether it is possible to meet these demands while migrating to cloud computing.
5. Perform the normal risk management tasks of assessing the risk of security or privacy violations, and the impact on the business.
6. Review the cloud providers' security/privacy measures (including physical security, personnel screening, incident notifications, etc., not just the technical security protection measures), and make sure that they are documented in the cloud SLA.
7. Determine whether the results of these steps actually allow the project to continue.
8. Consider and implement ways in which the information can be protected in four different situations:
    a. During the bulk migration of data from the on-premises system to the cloud service, when the cloud service is provisioned. This can be a weak point of the whole process, as an entire database backup may be carried physically, or shipped via courier, to the cloud service provider's site.
    b. "Data at rest," while stored in the cloud. An obvious solution for sensitive data is to encrypt the data, and the practical question is whether the provider can perform this service, or whether the client needs to research and implement a solution.
    c. "Data in motion," during the routine exchange of data that occurs while using the cloud based application. Encrypting data in transit is advisable, but runs into some issues: the cloud provider must support the encryption chain, cryptographic keys may need to be installed at both ends (requiring a key management solution), and on-the-fly encryption may affect transfer speeds.

d. "Data in use," that is when the data is actually read and processed by an application. For sensitive data, it may be advisable for the application to encrypt the data. This may not be possible if the migrated application is a commercial one that can only read the data in clear text from a database. A customer written application, on the other hand, can be modified to read/write encrypted data, so that only some temporary memory buffers will contain clear text data. The handling of encryption keys is a concern.

9. Design how to authenticate and authorize users. For systems that have their own sign-on facility, there may be no impact (as long as passwords are not sent in clear text from the user's workstation to the cloud-based system, which should not be the case even for an on-premises system). But if there is any form of Enterprise or Single Sign-On (SSO) facility, making this work from an application running in a cloud service may require integration work. An enterprise identity and access management system (IdAM) needs to be accessible from the application migrated to the cloud service. You will need to understand which protocols are supported by the IdAM and by the cloud service – additional integration components may be required to enable them to interoperate.[1] The silver lining is that once that effort has been made for the first migration, it should make future migrations easier.
10. Regardless of the solution chosen for authentication and authorization, you need to make sure that your user *de-provisioning* process can be executed quickly. Disabling a user's credentials for access to cloud systems may be even more critical than disabling their access to an on-premises system. The reason is that access to an internal system may be made immediately impossible or more difficult if someone has been escorted out the door; but might still be able to access the login page of a cloud application from the browser on their smartphone.[2]

Step 6: Manage the Migration

Finally, having thoroughly defined "on paper" the "why, what, and how" of the application migration project, the IT department can plan, execute and manage the actual application migration. Executing a migration is a complex and delicate project, and as such it should have a formal project plan and a skilled project manager. The migration plan, like all project plans, should track tasks, durations, resources, costs, and risks. Table 2 highlights the key components and considerations of the application migration procedure.

[1] Example IdAM protocols include LDAP, OpenID, SAML, WS-Federation and Active Directory.

[2] Note that if a single federated IdAM is used this risk is reduced since the user is removed once centrally for all applications.

Migration Procedure	Migration Details
1. *Deploy the Cloud Environment.* Provision, install and test the necessary storage, compute, network and security resources that constitute the cloud environment in which the migrated application will run.	• The first part of the cloud environment to be laid down is the structure of the virtual network. In private cloud architecture, this would typically be done according to the organization's pre-established standards for network addressing. For public cloud services, however, the network structure is often prescribed in advance by the cloud service provider. For virtual private cloud implementations, connecting the VPN to existing internal networks may require significant work to match network addressing spaces, namespaces and other network aspects. • Create individual virtual machines and attach them to their respective storage units. Reconfigure the domain name service (DNS) by updating the name servers to resolve the newly created VMs through the network gateways. • Provision security devices including firewalls and VPN routers. Configure directory services access by implementing and testing the connections between the cloud service and the organization's directory server (LDAP, Active Directory, etc.) or, if specified by the architecture, the federation between the cloud service provider's authentication system and the customer's.
2. *Install and Configure the Applications*. The applications and supporting middleware should now be installed and configured on the cloud servers. Cloud service providers frequently do this through automated deployment of templates.	• Implement all integrations between cloud applications and other applications or resources, including directory services. • All monitoring solutions should be implemented and tested, including any add-on monitoring tools. • If the cloud application servers are to manage and monitor licenses, apply the activation kits and keys. If the existing monitoring and key services are to be reused, make and test the connections from the application servers to these resources.
3. *Harden the Production Environment*. Install additional utilities for business continuity and security. Note that some of these services may be provided by the cloud service provider, in which case they do not need to be installed, but they should still be tested.	• Put in place and test automated backup capabilities. • Install and configure anti-virus software or malware protection. • Issue to all project team members their initial credentials for cloud service access, per their role in the project or ongoing operation.
4. *Execute a Mock Migration*. Undergo a trial run of the migration project plan to uncover unintended results or unnoticed issues during the planning phase. The mock migration date should be	• Ensure that all contractual aspects are in place with the cloud service provider, since the subsequent tasks will start consuming cloud services. • Since it is important to simulate all aspects of the final migration, schedule downtime for the existing systems during the time required to make the migration, and notify users in advance. • Import application data and configuration settings into the cloud

	sufficiently distant from the desired final cutover date to have time to rectify problems. Involve the cloud service provider in the migration date selection.	environment. • Run test scripts to validate application and data migration, connectivity from all endpoints, and proper access and authority. • Start the cloud environment and the applications. • Ask a preselected group of test users to validate that their work environments and systems are functional on the cloud-based system. These test users should follow formal test plans, designed in advance to exercise as many possible features of the applications within the allotted time. • Restart the on-premises production environment. • Document migration duration and metrics.
5.	*Cutover to Production Cloud*. Assuming a successful mock migration, or one that only encountered minor issues with a clear fix, establish a formal cutover schedule. If the mock migration ran into serious issues, then it needs to be repeated after correcting the causes.	• Update the migration plan, taking into account the lessons learned during the mock migration about the tasks to be added or removed, the actual durations measured, the change in resources required vs. initially expected, etc. • Line up the necessary resources from the cloud service provider, which may be different from the mock migration since this will now be the real, definitive migration. • Communicate the migration steps, timeline and impact to all users (and a summary to their management), including instructions for day-one steps that individual users must perform to access cloud services. • Re-execute the "Mock Migration" procedures but at the end, instead of restarting the old production environment, inform all users to apply the instructions they have received to restart their work using the migrated application. • Begin license, application and database monitoring for the production cloud environment. This monitoring continues indefinitely. • For some time after the cutover, a special "hotline" should be established for triaging and solving issues during initial usage time. • Hold one or more formal checkpoint meetings after migration to track any issues, until resolution, that need additional project tasks and resources.

Table 2. Application Migration Procedure

A more detailed description of migration tasks is provided in Appendix B.

References

Works Cited

[1] Cloud Standards Customer Council (2011). *Practical Guide to Cloud Computing*. www.cloud-council.org/2011_Practical_Guide_to_Cloud%20Computing.pdf

[2] Cloud Standards Customer Council (2012). *Practical Guide to Cloud SLAs*. www.cloud-council.org/2012_Practical_Guide_to_Cloud_SLAs.pdf

[3] Cloud Standards Customer Council (2013). *Public Cloud Service Agreements: What to Expect and What to Negotiate*. www.iaas.uni-stuttgart.de/institut/mitarbeiter/fehling/TR-2011-05%20Patterns_for_Cloud_Computing.pdf

[4] University of Stuttgart. *A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures*. www.cloud-council.org/Convergence_of_Cloud_Social%20_Mobile_Final.pdf

[5] CloudDesignPattern.org. *AWS cloud design patterns* http://en.clouddesignpattern.org

[6] Microsoft. *Windows Azure design patterns* www.windowsazure.com/en-us/develop/net/architecture/

[7] IBM. *IBM Workload Deployer Pattern-based Application and Middleware Deployments in a Private Cloud* www.redbooks.ibm.com/redbooks/pdfs/sg248011.pdf

[8] Cloud Standards Customer Council (2013). *Convergence of Social, Mobile and Cloud: 7 Steps to Ensure success*. www.cloud-council.org/Convergence_of_Cloud_Social%20_Mobile_Final.pdf

[9] Cloud Standards Customer Council (2012). *Security for Cloud Computing: 10 Steps to Ensure Success.* www.cloud-council.org/Security_for_Cloud_Computing-Final_080912.pdf

Additional References

Bridgewater, Adrian: *"Cloud Migration: the Problem with Legacy Software." CloudPro, www.cloudpro.co.uk/saas/4249/cloud-migration-problems-legacy-software*

Ul Haq, Salman: "*Issues in Migrating Legacy Systems to the Cloud."* Cloud Tweaks, July 2013. www.cloudtweaks.com/2013/07/issues-in-migrating-legacy-systems-to-the-cloud/

Vellante, David: *"IT's Online Enterprise Integration Crisis."* Internet Evolution, March 2010. www.internetevolution.com/author.asp?section_id=654&doc_id=188706

Varia, Jinesh: *"Migrating Existing Applications to the AWS Cloud."* Amazon Web Services, October 2010. http://media.amazonwebservices.com/CloudMigration-main.pdf

Using Patterns to Move the Application Data Layer to the Cloud www.iaas.uni-stuttgart.de/RUS-data/INPROC-2013-17%20-%20Using%20Patterns%20to%20Move%20the%20Application%20Data%20Layer.pdf

Appendix A: Examples of Cloud-ready Workloads

This appendix highlights some of the most cloud-ready workloads and their benefits.

Workload	Description	Benefits
Analytics	The analysis of massive data sets in near real-time or batch mode allows the generation of new information and intelligence about the business. The iterative exploration and investigation of past business performance provides insight and drives business planning. This is usually done on copies of operational data, not on the original, and therefore can run elsewhere.	• Reduce the capital and operating expenses needed to introduce or support enterprise wide BI services • Scale up rapidly when a massive Business Intelligence project kicks off (and scale down rapidly when it ends)
Collaboration	Collaboration tools provided to participants include e-mail, collaboration, presence and instant messaging, web conferences, file sharing, and enterprise social networking.	• Work beyond the boundaries of a single company and outside firewalls - share information more easily with customers, suppliers and business partners • Affordable and accessible - lower upfront investment and extremely easy to acquire • Eliminate individual departments' motivation to set up their own ad-hoc collaboration environments, which creates a confusing mix of often insecure solutions
Development & Test	Project environments that are used for all phases of the Software Development Life Cycle (SDLC) —and even production rollout, if the organization follows the DevOps approach. Development environments are used to design and build applications, test environments are used for various testing levels, including system integration, security, high availability, and user acceptance.	• Reduce the costs (capital, licenses) and labor to procure, configure, operate, manage, and monitor the environments, which may be different from project to project. • Reduce provisioning cycle times from weeks to minutes
Desktops & Devices	The system and application software that run in a desktop or laptop computer or pervasive devices such as smartphones and tablets. Examples include word processors, spreadsheets, media players, database applications, as well as industry- and/or organization-specific applications designed to be executed on the desktop or laptop (thick client applications).	• Predictability of the total cost of ownership (TCO) per user. • Remove the need for massive rollouts of new environments ("standard images") every couple of years. • Improve end-user productivity and satisfaction • Reduce end-user support complexity and costs • Little to no capital or one-time expense • Fast provisioning

Sporadic Compute-Intensive Applications	Simulation and other CPU-intensive tasks are often needed for only a limited period of time, for example during the validation phase of a new electronic circuit design. During the next burst of CPU demand, weeks or months later, a different application may be required. Moving these applications to a rapidly scalable computing environment, such as a large cluster provided as a cloud service, is much preferable to buying those expensive servers.	• Resources provisioned in minutes rather than weeks • Dynamic response to resource demand with elastic scalability • Consumption-based usage charges • No infrastructure to manage between bursts in demand
Storage	This category includes on-demand access to sporadic, overflow, or specialized storage resources, including high-performance and highly available consolidated storage for demanding or critical applications. In addition, the demand for cloud-based backup is growing, and it allows mobile devices to be backed up from anywhere there is an Internet connection. Many storage management solutions now support the use of cloud services as part of the storage pool. Longer-term archival solutions are also now supported as cloud services.	• High degrees of scalability: petabytes of data and billions of files • Professionally managed security that integrates into existing authentication systems • Built-in data placement and Information Lifecycle Management (ILM), including backup, archiving and retention management, via a global policy engine • Support for multiple tiers of storage including low-cost tape technology • High performance and availability • Access to online backup from anywhere without requiring a VPN connection into the enterprise network.

Appendix B: Application Migration Tasks - Example

Phase	Description
Initiation	1. **Kick-off Meeting** – This event sets the stage for the migration, and should include all members of the project team. Business stakeholders must be included in order to understand the scope, duration, resources, and to discuss the success factors and the success criteria that will be used at the end of the entire process. 2. **Gather Technical Discovery Data** – The objective of this step, which can be started at the kick-off meeting but will usually be completed through interviews and e-mail, is to understand every possible technical objective and constraints. For instance, the amounts of data to migrate, the required availability of the systems during migration, and network performance targets might all be discussed 3. **Identify User Load Parameters** – Discover and document groups of users, and the pattern of their use. For example, are there teams of contractors in foreign countries? When do they need access to the system? Where are the major groups of internal users? Are there any transient users, such as management, that should be counted? Is there a particular time of the day or of the month when usage is especially heavy? 4. **Open Trouble Ticket System** – Put in place a trouble ticketing and issue resolution processes. While this might be considered early in the process based on traditional projects, this establishes a clear discipline and method of communication and tracking, and logging troubles through a traceable system is important from the start.
Design the Production Architecture	1. **Determine integrations** – Decide how each interdependency between the migrated application and the on-premises systems that are not moving to the cloud will be supported. This frequently involves the establishment of sub-teams of subject matter experts to discuss these integrations. Refer to Step 4 of this guide's body. 2. **Specify Database Product and Version** – A critical part of nearly any IT system stack is the database engine, and this step is to specify the version level and operational information for the database core. This is a critical part of the cloud architecture, as data currency, replication and security control are paramount. 3. **Specify the Network Topology** – Look carefully at the network aspects of the architecture. They should consider wide-area network and local-area network performance characteristics relative to the current and future positioning of server and storage assets. Hop counts, latency and reliability of network links should be measured, and targets for post-migration performance established. 4. **Specify a Directory Architecture** – An aspect nearly as difficult as database integration is integration with corporate/organizational directory resources. Many organizations lack a single directory to serve as the "authoritative source" for access control and role-based permissions, and adding cloud computing to the architecture can often complicate the issue. Examining the directory status and desired target architecture is critical. 5. **Document Architecture Details** – Produce a complete architecture document that will serve in later stages as a guidebook for the implementation.

Phase	Description
Deploy the Cloud Environment	1. **Provision Virtual Infrastructure** – The first part of the cloud environment to be laid down is the structure of the virtual network. For public clouds, the network structure is often prescribed in advance by the cloud service provider, for the purposes of maintainability and automation. For virtual private cloud implementations, connecting the VPN Virtual LAN to existing internal networks may require significant work to match network addressing spaces, namespaces and other network aspects. 2. **Provision the Edge Firewall** – This is the gateway to the cloud service, and usually involves creating some sort of virtual private network Uniform Resource Identifier (URI) on the edge of the internal network and tying it to on-premises network resources. 3. **Provision Storage** – Create the logical units on the cloud's Storage Area Network (SAN), according to the architecture documents. 4. **Deploy Virtual Machines** – Create individual virtual machines and attach them to their respective storage units. 5. **Reconfigure the Domain Name Service (DNS)** – Update the name servers to resolve the newly created VMs through the network gateways. 6. **Test Network and Server Connectivity** – Fully test the network connectivity, noting performance characteristics and measuring them against the desired targets from the architecture. 7. **Update Documentation** – Update the documentation with the test results and any modifications made from the initial architecture. 8. **Configure Site-to-Site VPN** – If the architecture requires a static site-to-site VPN connection, implement and test this. 9. **Configure Directory Service Connectivity** – Implement and test the connections between the cloud service and the organization's directory service (LDAP, Active Directory, etc.) or, if specified in the architecture, the federation between the cloud service provider's authentication system and the customer's.
Install and Configure Applications	1. **Install Server Software** – Install and configure the application server software on the cloud servers. Cloud providers frequently do this through automated deployment of templates. 2. **Implement the Database** – Implement the database per the architecture. Again, this is often done through automated, template deployment. 3. **Configure the Application** – Configure the application servers and tools as specified, including the application of any customizations or templates. 4. **Enable License Tracking** – If the cloud service is to manage and monitor licenses, apply the activation kits and keys. If the existing monitoring and key services are to be reused, make and test the connections between the application servers to these resources. 5. **Implement Integrations** – Implement all integrations between the migrated application and on-premises resources. 6. **Configure Monitoring** – Implement and test all monitoring solutions, including SNMP services and other add-on monitoring tools.

Phase	Description
Harden Production Environment	1. **Configure Anti-Virus** – Ensure that the cloud service provider installs and configures sufficient anti-virus software or malware protection. 2. **Configure Database Backups** – Implement any specific procedures or servers used to back up the application data (a database backup often requires specific techniques other than ordinary disk backup). 3. **Establish Password Change Mechanism** – Depending on the cloud service provider, there may or may not be automated mechanisms for password and ID changes. Implement and test the process for these frequently used and security-sensitive operational systems, including notifications of changes to appropriate managers. 4. **Obtain and Install SSL Certificates** – For any access secured through SSL (secure browsing or SSL VPN), install the signed certificates. 5. **Establish Management IDs** – Issue to all project team members their initial credentials for cloud service access, per their role in the project or ongoing operation. 6. **Establish User IDs** – Load user IDs and initial passwords into the directory, unless existing directory resources are serving as the credentials source.

Phase	Description
Mock Migration	1. **Set Migration Date & Schedule** – The purpose of this set of steps is to undergo a "mock migration", which is a trial run at the project plan to uncover unintended results or unnoticed issues during the planning phase. The date should be sufficiently distant from the desired final implementation/cutover date to have time to rectify problems. Involve the business users and the cloud service provider in the migration date selection. 2. **Activate Cloud Services Agreement.** Ensure that all contractual aspects are in place with the cloud service provider, since the subsequent tasks will start consuming cloud services. 3. **Notify Users of Outage** – Since it is important to simulate all aspects of the final migration, schedule downtime for the existing systems during the time required to make the move to the cloud service, and notify users in advance. 4. **Stop Applications** – At the appointed day and time, stop the current application servers and shut down on the current production environment. 5. **Capture Database Backups** – Make complete backups of the on-premises databases that will be migrated. Execute validation scripts to ensure the integrity of the backed-up databases. 6. **Export Application Configurations** – Export configurations and customizations from the servers that will be migrated. 7. **Import Application Configurations** – Apply the exported configurations and customizations to the target application servers in the cloud. 8. **Configure Manual Settings** – Apply any additional settings that did not migrate in the earlier configuration export/import. 9. **Restore Database** – Create the cloud-based databases by restoring the validated backups of production data created earlier. 10. **Start Cloud Application Servers** – Restart the cloud application servers and test for integrity and access to data. 11. **Run Database Validation Jobs** – Run a validation of the database again to ensure integrity. 12. **Compare Source and Destination Data** – An extra desirable validation step is to compare sample data from the source and target systems for extra integrity validation. 13. **Customer Validation and User Acceptance Test (UAT)** – Grant a pre-selected group of test users access to the cloud-based system to validate that their work environments and systems are functional. Make sure users understand that they will enter mock transactions, and that if they take them from their real workload, they will have to re-apply them to the current system after the test. Test all user access methods (Web, mobile, etc.) and locations for connectivity and performance. 14. **Test Authentications** – Test samples of all roles and authentication mechanisms for accessibility. **15. Document Migration Duration and Metrics** – Document all migration steps and performance characteristics in the project plan. **16. Restart the on-premises application.** Perform any steps necessary to terminate the mock migration, restoring access to the current environment and application.

Phase	Description
Production Cloud Migration	1. **Formalize Cutover Schedule** – If the mock migration was unsuccessful or exhibited major issues, go back to the appropriate phase in order to correct the problems and run a new test. Assuming a successful mock migration, establish and communicate a formal cutover schedule that takes into account the lessons learned from the mock migration on how much time each task really took. Include the scheduling of all necessary resources from the cloud service vendor. 2. **Communicate Changes to Users** – Communicate the migration steps, timeline and impact to users, including instructions for day-one steps that individual users must perform to access cloud services. Inform users of the procedure to report issues, and train the Help Desk on the new trouble ticket types and escalation rules for migration-related issues. At this point, all the steps of the mock migration should be repeated, followed by these additional steps: **3.** **Open Cloud Migration Hotline** – For some time after the cutover, a special "hotline" should be operated to triage and solve issues. 4. **Flip DNS** – The relevant domain records should be changed to point to the cloud services. 5. **Migration Go/No Go Meeting** – Hold a formal rollback/proceed decision meeting between all stakeholders, including the users affected by the migration and the cloud service provider. 6. **Enable License Monitoring** – Begin the license monitoring for the production cloud service. This process will continue for the life of the application. 7. **Configure Application Monitoring** – Begin application and database monitoring for the production cloud service. This process will also continue indefinitely. **8.** **Post Migration Checkpoint** – Hold a first formal checkpoint meeting shortly after migration to assess any large-scale issues that need additional project plans and resources. This meeting ends with a decision: has the system reached sufficient stability and productivity that this is now "business as usual?" If so, decide to close the project. If not, assign corrective actions, communicate the actual plan to management and to users, and schedule the next checkpoint meeting. **9.** **Project Closure.** Archive all relevant documents, release any temporary resources assigned to the migration, document lessons learned, celebrate success and reward key contributors.