



Strategies for Large-Scale Cloud Migration A 5-Part Series

Section 1: A Process Framework to Help Plan Migrations

In its most recent annual CEO Survey, Gartner found that the number two priority among CEOs across industries is technology-related business change, following only profit growth. Driving interest in technology is a mandate to modernize in order to better support the business as IT is having trouble delivering solutions at the rate business is desiring them, thereby slowing down business.

A cloud migration -- and the ability to take advantage of cloud benefits such as greater agility, scalability and enhanced security -- is often seen as a primary way that organizations can positively affect change and create greater productivity at optimized cost. Although momentum is pushing enterprises toward these benefits -- with 46% of I&O leaders citing agility a top priority, and lowering the costs of service delivery their collective second top priority -- this approach is not without its challenges. For, there are many approaches to large-scale cloud migration, which can be overwhelming and confusing, especially for large enterprises with a myriad of business-critical applications. As a result, this paper will discuss approaches for large-scale cloud migrations in support of strategic business change, as supported by a best of breed team and set of vendors.

Cloud Migration Options

As an organization begins looking at its larger technology landscape through the lens of cloud computing and strategic change, several options emerge. For companies that want to centralize and standardize their infrastructure in order to reduce overall system maintenance, grow GRC and reduce costs, there are four distinct migration approaches and three non-migration choices. These approaches also offer the opportunity to renovate hardware, software, and processes, and to simplify systems in the process.

- **Rehost** -- Sometimes referred to as lift and shift, rehosting takes a forklift approach to moving business applications to the cloud, moving them without any code modification. In this case, a VM is treated like a blackbox and simply copied bit by bit.
- **Reinstall** -- Similar to rehosting, a reinstallation entails a wholesale move of applications to the cloud. While new VMs are created in the cloud, the same software is installed from scratch with no configuration changes. Although it's more work than a re-host, this approach has the benefit of allowing some cleanup to happen during the migration, allowing companies to avoid copying unnecessary software and configurations that may have accumulated over the years and are no longer in use. (E.g., an agent that was installed for monitoring software that is no longer in use at the company.) It also allows for the addition of new practices, e.g., installation of a new vulnerability detection agent on all VMs, or the use of standardized VM images for all apps.

1 According to Gartner Leadership Vision for 2018: Infrastructure & Operations Leaders Published: 08 August 2017 ID: G00337290

- **Replatform** -- Called "lift, tinker and shift" by Stephen Orban, Head of Enterprise Strategy at AWS, replatforming is the process of moving applications to the cloud with a small amount of up-versioning -- perhaps of an OS or database -- to benefit from cloud infrastructure. Replatforming takes advantage of containers and VMs, only changing application code when needed to use base platform services like Amazon RDS and Amazon ElastiCache, and advanced Amazon EC2 services like autoscaling and ELB.
- **Refactor** -- The process of re-architecting some portion of an existing application to take advantage of cloud-native frameworks and functionality. Most often, refactoring entails changing middleware, and application components to take advantage of cloud-native features and advanced concepts like microservices and serverless. Application code itself is not refactored, but rather the services composing it. While the application business logic remains the same, the application itself is factored into different tiers and pieces with services, like databases, swapped out for the cloud service equivalent.
- **Repurchase** -- The move from software that previously ran on data center hardware to a new SaaS (or similar) version of the same application. For example, moving from an on-premise CRM to Salesforce.com.
- **Retain** -- Leave the application on-premise, running just as it currently does.
- **Retire** -- Determine that an application has met end-of-life criteria.

Naturally, each of these options has its own pros and cons which should be considered on an application-by-application basis as not every app will pass a cost-benefit or feasibility analysis of a refactoring project.

See the next page for our [Migration Options Comparison Guide](#)

Migration Options Comparison Guide

Approach	Maturity	Pros	Cons
Rehosting	Uses only pure IaaS services (VMs and networking)	<ul style="list-style-type: none"> Faster migration with fewer resources thanks to automation tools Migration operator can treat VMs like a black box, as it requires minimal tribal knowledge Once a migration factory has been built, migration becomes a push-button activity 	<ul style="list-style-type: none"> Can be more costly to run apps in the cloud this way as compared to a replatformed or refactored app Apps don't benefit from all cloud features like managed services
Reinstall	Use fresh VMs in the cloud with some new tools like a vulnerability detector. Automates provisioning of infrastructure but not full-stack.	<ul style="list-style-type: none"> While not the fastest, migration is still fairly speedy Takes advantage of some new tools Allows retirement of technical debt that may have accumulated on-premise servers 	<ul style="list-style-type: none"> Can be more costly to run apps in the cloud this way as compared to a replatformed or refactored app. Apps don't benefit from many cloud features Migration operator requires deeper knowledge of the application install process Effort required for manual installs
Replatform	Use advanced IaaS features like auto-scaling, native load balancers, and container services. Also, use PaaS services e.g., AWS RDS and SQS. Lastly, automate installation of the entire stack	<ul style="list-style-type: none"> Middle ground between rehost and refactor, allowing app to take advantage of cloud functionality Automation of operations like auto-scaling saves effort Automation of software provisioning provides agility and CI/CD capabilities Increased agility opens door for more innovation Once common app architecture patterns have been identified and automated, migration happens at a rapid pace in a self-service manner 	<ul style="list-style-type: none"> Slows migration time Requires more knowledge and resources
Refactor	Use advanced platform services that lead to paradigm changes, e.g., moving a monolith to microservices using containers and serverless technologies.	<ul style="list-style-type: none"> Apps can be modified to take full advantage of cloud-native features Apps can be modified to maximize cost efficiency in cloud Best cloud ROI 	<ul style="list-style-type: none"> Takes longer to develop Requires more resources upfront
Repurchase (as SaaS)	Use a brand new application or the SaaS version of the app, e.g., switching to Salesforce rather than an on-premise CRM	<ul style="list-style-type: none"> Faster than refactoring Generally cost-efficient IT is permanently freed from upgrades, patching, etc. 	<ul style="list-style-type: none"> Scalability, availability, security, compliance and more are dependent on SaaS provider Integration can be challenging

Migration Option Self-Assessment

Assessing the migration path is an exercise in feasibility and cost-benefit analysis. For example, although an application may benefit from cloud features, if it is only used once a year by two people in the organization, the cost is most likely not worth the benefit. To help assess the cost and benefit of each strategy, here are sample questions that Flux7 recommends organizations answer for each application they consider migrating to the cloud:

1. How strategically imperative is this application to the business? Is it an application that contributes to revenue and should be invested in, or an application necessary to just run the business and hence shall be sustained at the lowest possible TCO? This is important because migrations are often bound by time and budget, and the energy required to replatform/refactor is better invested in invest applications, rather than sustain applications.

For example, an e-commerce website for a retailer is an app that should be invested in, while the employee vacation reporting system for HR is something to sustain. Given typical constraints, it's best to consider a refactor/replatform for the e-commerce website and a simple rehost for the vacation reporting app.

2. For sustain applications, is it even possible to rehost the application, i.e., are all technologies even runnable in the cloud? If yes, rehosting is the best option. If not, you should look for a SaaS alternative. If found, retire the app. If not, retain it.

For example, if the vacation reporting app uses an IBM AS400 machine, it may be best to either replace the app entirely with a different tool or retain the application on-premise rather than investing developer resources to refactor (rewrite) it for the cloud.

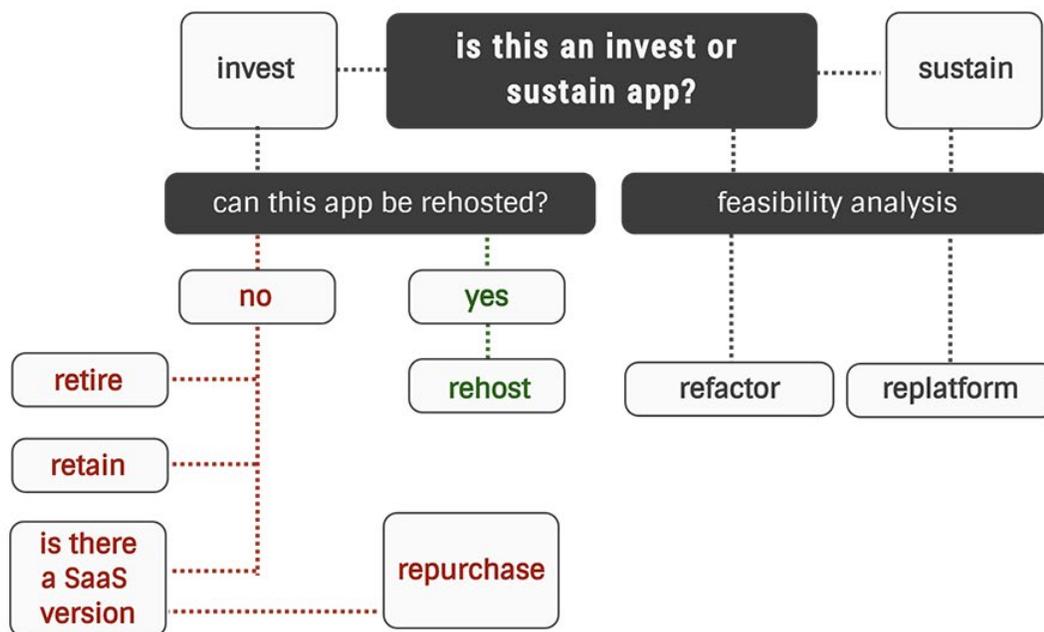
3. Invest in applications that require a cost-benefit analysis. Analyze cost in terms of development resources and any business interruptions that may be required from a significant rewrite. Note that sometimes it's often impossible to refactor applications because the original development teams are currently inaccessible as they have other priorities. Also to be considered are the benefits of refactoring which must be computed through careful analysis. If the benefits outweigh the cost, and the direction is achievable within the constraints, refactoring is the right choice. Due to these constraints, most applications are not easy to re-factor at the onset and companies often re-factor a very small fraction (<10%) of their portfolio.

For example, if an application suits the serverless computing model (e.g., using AWS Lambda) and the development team has the resources to refactor to use Lambda, refactoring is possible. Next, we evaluate the benefit of moving to Lambda, i.e., will this app really benefit from effortless-scaling and minimal ongoing maintenance. If these benefits are important for the application, refactoring it is feasible and the right choice.

4. Lastly, for invest applications, if refactoring is not feasible, replatforming is the right choice. Given tight constraints we see companies opting to replatform 25%-30% of their portfolio. In these cases, the burden is on the DevOps team to build a harness for the application, which without requiring major code changes, pr the benefits of cloud features such as auto-scaling, self-healing, containers, etc.

For example, an e-commerce website written in a framework which is not suitable for serverless may be replatformed.

Rationalization in this way can be an effective first step before making a decision about which path to take for any given application. By looking at the totality of an IT system, it may be possible to cut through the noise and identify which infrastructure and workloads are low complexity and low business value. These are ideal candidates for retirement. It can also identify which are low complexity, but high business value, as these can be more easily relocated to the cloud.



Yet, modernizing a highly complex system often meets political resistance from the various interested members within the IT and management teams. A rationalization process should help to combat political push-back against a comprehensive modernization project by making clear exactly the systems which need to be relocated or retired and why.

Section 2: The Case Studies

Reinstall Case Study: TechNip

TechNip is a renewable energy leader that had two parallel goals. First, it wanted to use cloud migration as an opportunity to overhaul its business processes. Secondly, in the migration process, this organization looked to build standardization. In the process, TechNip aimed to increase developer agility, grow global access for its workers and decrease capital expenses. While time was of the essence, and TechNip had a desire to move the enterprise wholesale to AWS, a reinstall approach was pursued. Following the migration, IT transformation was championed, especially once standardization had been achieved.

The project began with Flux7's AWS migration experts reviewing critical data about the applications to be moved, and designing push-button deployment frameworks which were then used to quickly select and stand up new environments, allowing the client's DevOps team to focus their efforts on specific applications and not the foundational policies that are part of the AWS automation. The teams moved one application at a time into AWS, starting with a residential solar monitoring Web app.

In order to introduce standardization and maintain separation of duties, the Flux7 team introduced AWS Service Catalog, where 80% of applications are able to be defined by a small number of templates. In this way, the energy company was able to standardize builds with security controls built-in, encouraging best practices to be followed by default. And, in order to streamline DevOps, AWS was used to create multiple environments for each team, increasing speed to market and operational efficiencies.

Single App Replatform Case Study: Rent-A-Center

Rent-A-Center (RAC) wanted to move its eCommerce platform to the cloud in time for the busy holiday season. The aim was to roll out an eCommerce platform that would support the entire online shopping workflow for RAC using SAP's Hybris platform which required a cluster of Hybris servers called nodes that would cater to online web-based demand. Using advanced Amazon features like autoscaling, ELB, and container services, together we created an architecture that would scale up to meet growing demand and scale down once demand subsided.

With Amazon ECS as a backbone technology, RAC has been able to deploy a Hybris setup with auto-scaling, self-healing, one-click deployment, CI/CD, and PCI compliance consistent with the company's latest technology guidelines and meeting the requirements of their newly formed culture of DevOps and extreme agility. In addition to taking advantage of advanced cloud functionality--like extreme scalability--by refactoring its eCommerce platform, RAC has set the stage for refactoring in the future, allowing it to plan for even greater agility and experimentation.

Mass Replatforming Case Study: Enterprise Media

Flux7 had the opportunity to work with an enterprise media group on its modernization project. Working with the specialists at Flux7, this firm walked through the migration options for its on-premise applications and built a secure sandbox where business units could experiment and learn. From these exercises, the two teams created a 12-month roadmap, broken down by quarter, mapping each application to its migration path and a specific timeline.

Using the Flux7 approach and its Enterprise DevOps Framework (EDF), systems of Innovation and Differentiation were identified and 200+ apps were separated from the strategy for systems of record. Flux7 designed a platform for innovation and helped the customer gain knowledge about the strategies and underlying technologies in order to gain greater business agility.

Of the firm's 400 apps that needed to be replatformed or refactored, application technologies included 80 .Net IIS apps on Windows, 40 Java/Tomcat apps on Kubernetes, 80 Java/Tomcat apps in Linux, and Adobe Experience Manager (AEM), Atlassian tools, Wordpress, and SugarCRM. Approximately 40% of its apps, which were maintenance-level, yet business-critical, were flagged for replatforming/refactoring. In this way, the firm was able to strategically migrate its assets, maximizing cloud benefits for those applications that would gain the most while not spending resources on applications that would benefit minimally.

Refactor Case Study: Healthcare Provider

This healthcare organization was looking to move its monolithic application to the cloud, in the process re-architecting it for greater agility, visibility and business responsiveness. Using a containerized approach, the Flux7 team helped this firm refactor its application, moving it to an AWS-based microservices architecture.

A blueprint for the new microservices environment was created, with AWS ECS and Docker as the backbone. One of the initial concerns the team had was to create a new architecture and process flow that incorporated AWS automation, security controls, greater visibility, and repeatability of builds. As such, Flux7 recommended AWS Service Catalog to fill this role. With the AWS Service Catalog in place, the development team was able to easily achieve AWS provisioning of the assets they needed within minutes.

By breaking its single code base into small, individual services, the healthcare provider achieved greater human resource utilization as bottlenecks were removed and developers could work autonomously on their individual services, much more flexibly and nimbly incorporating changes and/or introducing new services to the business.

² A High-Level Framework for Planning Your Migration to Public Cloud Services", June 20, 2017, Craig Lowery

Conclusion: A mass migration from an on-premise data center to the cloud can seem formidable, especially for enterprises with hundreds of business-critical applications. Using these approaches and the EDF, a best-of-breed team with specialists who have deep cloud computing, migration, and enterprise experience can create a strategic migration roadmap that meets a company's digital transformation objectives. With the ability to accelerate productivity, grow agility and optimize costs, advanced AWS services provide the technology backbone to help businesses fully achieve digital transformation.

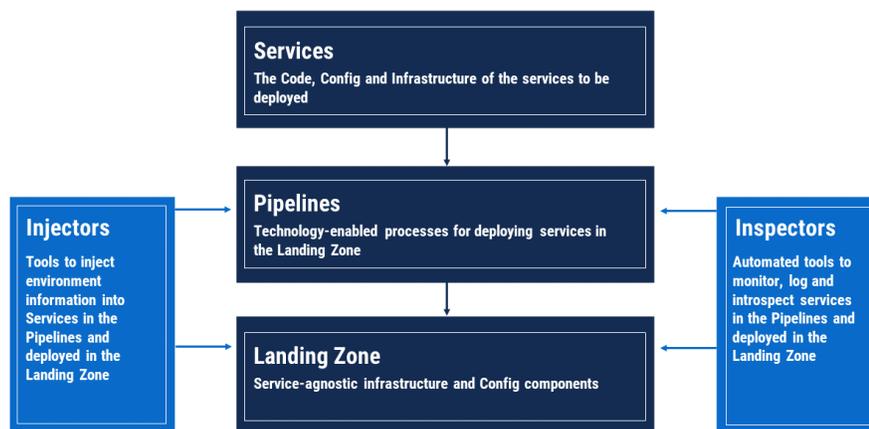
Section 3: The Enterprise DevOps Framework

Once a migration strategy is established, Flux7 consultants use the Enterprise DevOps Framework (EDF) as a foundation for migrating applications and creating the needed infrastructure to effectively support them in the cloud. Regardless of the migration approach, the EDF is a helpful framework for this work. Using the EDF as a guide for planning and implementation, operators can build a platform for change that incorporates advanced cloud technologies alongside services, landing zones, pipelines, inspectors and injectors.

Organizations benefit from the EDF as it provides:

- A clear structure and process for adoption, with technology mapping to DevOps processes
- Accelerated time frames for an initial POC
- Reduced risk by starting with best practices
- Hardened security through increased adherence to security and governance
- The agility of independent management

The Flux7 Enterprise DevOps Framework



Mapping the EDF to cloud migration approaches, illustrating how each approach brings assets to the cloud.

Section 4: Establish a Beachhead: The Landing Zone

In the EDF, the traditional IT operations team is converted into a concept called the landing zone. The landing zone is where services deploy and as a result is focused on catching service agnostic components as they are delivered via pipelines. In the EDF, the concept of a service-agnostic landing zone is very critical as is the idea of service teams owning more of their dependencies. A landing zone is a requirement regardless of the migration strategy taken.

This approach provides a beachhead and it allows teams to decouple field planning from creating a landing zone in AWS for services. Teams can quickly start sending applications to it as the work can be done in parallel, starting with the applications teams prepping their applications to be sent to the landing zone. Because the landing zone is agnostic, application teams don't need to worry what the landing zone looks like. They need only to worry about getting their applications set for migration. This approach can breed quick success as the development and DevOps teams work in tandem.

This approach also helps speed the success of POCs as it can support small, impactful first steps. While many teams might want to start by building a best-practices network architecture, for example, it is not impactful by itself as it does not add business value. It is analogous to building an airport without having any planes to land. You need the planes to demonstrate the value. In this way, the landing zone provides the landing strip to easily showcase early migration wins, in turn, helping to build internal momentum.

Section 5: Sandboxes as a Strategic Enabler For Cloud Migration

For the list of applications to be moved -- and as operators discern whether to rehost, replatform or refactor them -- secure sandboxes can be a helpful device. Secure sandboxes are places where teams can experiment as a way to help them determine where to start with their migration. Specifically, this approach allows business units to cultivate ideas how a cloud migration can help their efforts by giving them a secure space to explore the platform, without creating barriers to entry.

Flux7 consultants recommend creating secure sandbox environments as organizations start down the path to a cloud migration, as they can help prioritize migration targets and can generally shed light on the benefits cloud computing will bring to different applications.

"Develop cloud-native architecture and infrastructure experience. If you are just starting out, ensure you enable a good architecture practice and set up a center of excellence (COE) to experiment with cloud-native concepts and formulate best practices. Start with a few greenfield pilots to enable architects and developers to familiarize themselves with MASA, API design and event-driven architectures. After each pilot, debrief, capture lessons and assess the pilot on the basis of your business outcomes and metrics."

-- Gartner, "CTO Alert: Master Event-Driven IT to Master Digital Business"

Best in Breed Support

There are many professional services organizations -- from managed services providers to traditional IT service houses -- that portend to have the expertise to help with large scale cloud migrations. To find the best fit partner for a migration project, there are several best practice questions to ask:

Specialist or Jack of all Trades?

Service organizations tend to take one of two routes, choosing to grow deep or wide, expanding across a breadth of services or diving deeply into a few specific areas. While a jack of all trades is a master of none, they may not be the best choice, especially for organizations that need advice along the way or need someone to design a migration blueprint for them. To that end, specialists are more likely to have implemented a similar project and as a result, are more educated about potential pitfalls to avoid and what steps to take to ensure success.

In addition, specialists are more likely to have established partnerships with complementary vendors that help usher in success. For example, partnerships with tool and ecosystem vendors from AWS to Docker, Ansible, Hashicorp and more are extremely helpful when designing and executing a mass cloud migration. This is especially true for migrations where advanced services are needed to replatform and refactor applications.

Lock-In or Master of Own Destiny?

Many organizations work to avoid long-term contracts and other lock-in devices. Yet, depending on a company's particular needs, a managed services provider (MSP) might be a good fit. MSPs offer the ability to run your service for you on an ongoing basis. The MSP benefits from a recurring revenue stream and the client benefits from a negligible liability setup. Conversely, firms like Flux7 prioritize knowledge transfer as their business is dependent upon a smooth transition and they want their clients to seamlessly takeover system management once setup. Just as importantly, they want to ensure the company's ability to extend the architecture moving forward, ensuring ongoing success.

Flux7: Building a Platform for Change

Enterprises ready to migrate to AWS infrastructure face hundreds of critical questions. To help increase the success and speed of AWS migrations, Flux7 has created AWS Migration Services for enterprises who want to learn “how to fish” in order to take control of their infrastructure and improve their environments along the way.

Using the EDF as a framework, Flux7 empowers enterprises to migrate by providing infrastructure, coaching, and automation through all phases of complex migration projects. Specializing in replatforming and refactoring migrations, Flux7’s award-winning services deliver best practices based templates, and automation to migrate, manage and extend infrastructure for ongoing success.

About Flux7

Flux7, an NTT DATA Company, is an IT services firm that helps enterprises reduce the complexities of a new or evolving cloud automation strategy. Agile and DevOps-native, Flux7’s robust services portfolio prioritizes a fast path to ROI that meets the immediate needs of technical and innovation teams focused on transformation while forging a secure and stable pathway for security and operational excellence. Learn how Flux7 helps businesses bring solutions to market faster at <https://www.flux7.com>