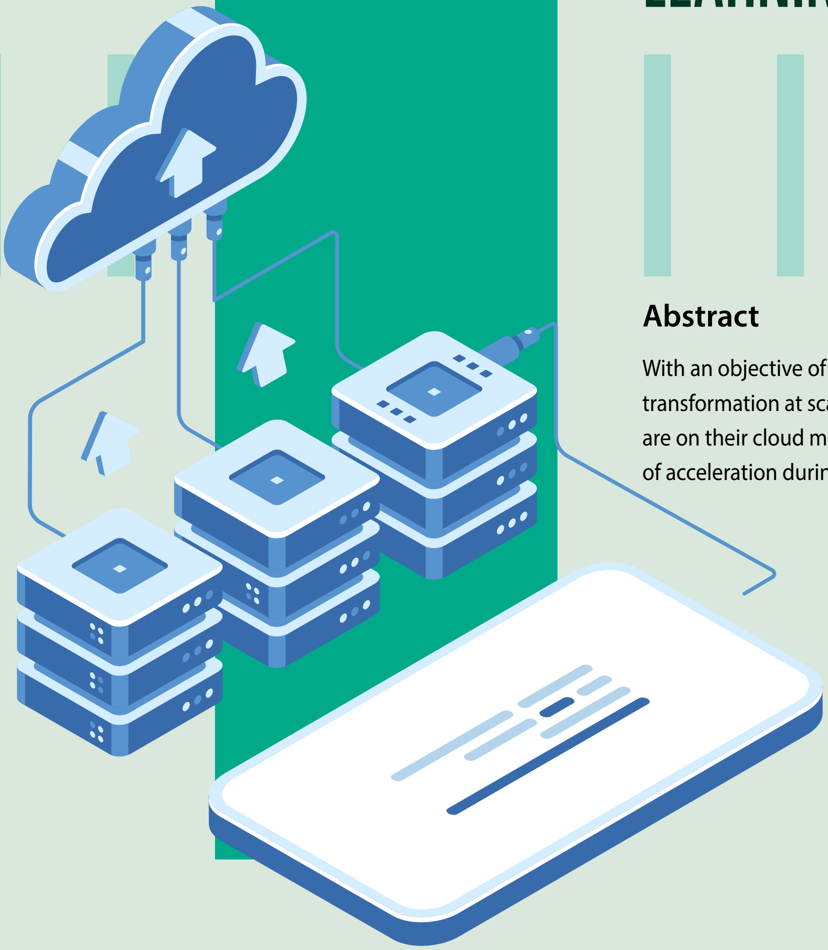
WHITE PAPER

CLOUD MIGRATION AT SCALE LEARNINGS



Abstract

With an objective of achieving digital transformation, innovation, engineering transformation at scale and to optimize infrastructure costs, financial services firms Globally are on their cloud migration journey. The progress has been slower than expected in-spite of acceleration during the pandemic, with only 20-45% workloads on cloud while the objective for most firms was to achieve over 70% by 2023. Besides the slow progress, for many Enterprises the cost advantages haven't also been realized yet. Additionally, there are regulatory questions around cloud dependence and strategies for cloud exit for critical business workloads to take care of operational resilience.

This paper aims to examine the challenges that have hindered the adoption of cloud and some of the strategies and frameworks that will help accelerate the progress while maintaining cost control.

This paper will present a decision matrix for on prem vs. cloud workload migration strategies and delve deeper into successful elements of executing a cloud migration program.

Infosys®
Navigate your next

Contents

Challenges

Cloud migration at scale involves moving many applications, data and other business assets from an on-premises or private cloud environment to a public or different private cloud. This can be a complex process, as it requires careful planning and execution to ensure that the migration is successful and minimizes disruption to business operations. This document brings forward learnings from such projects so that future projects can be planned. Several issues/surprises arise in large scale cloud migration programs. While learnings are specific to migrations of workload from on-prem to Azure, they can be applied to other cloud providers as well.

Below are some of the reasons why cloud migration is delayed.

Lack of clear objectives: It is important to have clear goals and objectives for a cloud migration project to ensure that it stays on track. If these objectives are not well-defined, it can be difficult to measure progress and identify any issues that may arise, which can lead to delays.

Lack of planning: Failing to carefully plan a cloud migration project can lead to delays as well. It is important to carefully assess the needs and requirements of the application and the organization, as well as to develop a comprehensive plan for the migration.

Complexity of the existing infrastructure: If an organization has a complex or outdated IT infrastructure, it can be more challenging to migrate to the cloud, leading to delays.

Complexity of the migration: Migrating an application or data to the cloud can be a complex process, especially if the application or data is large or has complex dependencies. This can lead to delays as the migration team works to understand the application or data and plan the migration. This can make it difficult to accurately estimate the time required for the migration and can lead to delays.

Technical challenges: There may be technical challenges that arise during the migration process, such as compatibility issues or data transfer issues, which can cause delays.

Lack of resources: Migrating to the cloud often requires dedicated resources, including personnel with the necessary skills and expertise. If an organization does not have the necessary resources available, it may struggle to make progress on the migration in a timely manner.

Budget constraints: Migrating to the cloud can be expensive, and budget constraints may limit an organization's ability to allocate the necessary resources for the migration. This can lead to delays as the organization waits for the necessary funding to become available.

Regulatory or compliance issues: Depending on the industry and location of an organization, there may be regulatory or compliance issues that must be considered when migrating to the cloud. This can add complexity to the process and may lead to delays as the organization works to ensure compliance. For example, if an organization needs to ensure that certain data is stored in a specific location, it may need to take additional steps to ensure compliance before migrating to the cloud.

Compatibility issues: If an application or workload is not compatible with the cloud infrastructure or services being used, it may be necessary to make modifications or updates to the application before it can be migrated. This can lead to delays as the necessary changes are made.

Resistance to change: Changing to a new technology or platform, such as the cloud, can be disruptive and can cause resistance within an organization. This resistance can lead to delays as the organization works to overcome it.

Dependency on third-party vendors: If an organization is reliant on third-party vendors for certain aspects of their IT infrastructure, the migration process may be delayed if those vendors are not able to support the migration in a timely manner.

Changes to the business or requirements: Changes to the business or requirements for the application being migrated can cause delays if the migration process needs to be adjusted to accommodate those changes.

Dependencies on other systems or applications: If the application being migrated is dependent on other systems or applications, any delays or issues with those systems or applications can impact the migration.

Integration challenges: If the application being migrated needs to integrate with other systems or applications, there may be challenges in getting everything to work together properly. This can lead to delays as teams work to address integration issues.

Overall, there are many factors that can contribute to delays in cloud migration, and it is important for organizations to carefully plan and manage the process to minimize these delays as much as possible.

Strategies

There are three important key strategies that needs to be considered when planning a cloud migration at scale, including:

- **Planning:** This involves identifying the assets that need to be migrated and determining the most appropriate cloud platform for them. And determine cloud dispositions (re-hosting, re-platforming, re-architecting, etc.) by using application inventory data and assessment questions and evaluating the ease of migration and potential benefits for each application, along with cloud migration patterns and hosting locations (virtual, bare metal, containerized).
- **Way of Execution:** When it comes to implementing the migration to the cloud, there are several factors that a large enterprise with a vast array of applications must consider. The first consideration is the strategy, including who will take responsibility for driving the migration. Should it be the current application teams who manage and maintain the applications, or would it be more appropriate to engage an external team or vendor with the necessary expertise to oversee the migration process.
- **Cost Control:** To ensure that we maintain cost control throughout the migration process, it's important to address cost-related issues at different stages of the migration. These stages include intake, which is when teams first begin the migration process, as well as the design and build phase, testing and acceptance phase, the ready-to-go-live phase, and finally, the business-as-usual (BAU) stage.

Overall, cloud migration at scale requires a thorough understanding of the business needs and the capabilities of the target cloud platform, as well as careful planning and execution to ensure a smooth transition.

Planning

To ensure a successful migration to the cloud, it is necessary to first identify the assets that need to be migrated and then determine the best cloud platform for each of them. This involves analysing application inventory data and utilizing assessment questions to determine the most suitable cloud disposition, which may involve options such as re-hosting, re-platforming, or re-architecting. The ease of migration and potential benefits for each application must also be evaluated, along with considerations such as cloud migration patterns and hosting locations, which may include virtual, bare metal, or containerized environments.

Identifying the right applications to migrate to cloud.

Not all applications are suitable for the cloud. Some applications may be better suited to on-premises deployment, depending on a variety of factors. When deciding whether to move an application to the cloud, it is important to consider the following factors:

Cost: Cloud services can be more cost-effective than on-premises solutions in some cases, but this is not always the case. Be sure to carefully compare the costs of cloud and on-premises options to determine which is more cost-effective for your organization

Deployment model: Some applications are better suited to the cloud than others, depending on their deployment model. For example, applications that are designed to be highly scalable and able to handle large amounts of traffic may be better suited for the cloud than applications that are designed to be run on a single server.

Compatibility: Some applications may not be compatible with cloud environments or may require significant modification to work properly in the cloud.

Dependencies: If your application relies on other systems or resources that are not available in the cloud, it may not be suitable for migration.

Compliance requirements: If an application must meet certain compliance requirements, such as HIPAA (Health Insurance Portability and Accountability) or PCI DSS (Payment Card Industry Data Security Standard), Schrems-II, it may be more appropriate to keep it on premises or in a private cloud rather than in a public cloud.

Data requirements: Applications with large amounts of data or that require real-time data processing may be better suited for the cloud, as the cloud can provide the storage and processing power needed to handle these requirements.

Performance: Some applications may require the high-performance computing and storage capabilities that can only be provided by on-premises solutions for example On-premises GPUs are better suited for machine learning applications that require high memory bandwidth and low latency. These can include deep learning applications that require high-performance graphics processing units (GPUs) with extensive memory bandwidth requirements.

Integration with existing systems: If your application needs to be integrated with other on-premises systems or applications, it may be more practical to keep it on-premises. The organization can also look at hybrid strategy, but additional attention will be needed in terms of setting up the required security.

Need for Scalability: The cloud allows organizations to easily scale their applications up or down as needed, without the need to purchase additional hardware or make other significant investments.

Need for Flexibility: Cloud hosting provides organizations with the flexibility to access and use their applications from anywhere with an internet connection.

Need for Reliability: Cloud providers typically offer high levels of uptime and reliability, which can be important for mission-critical applications.

Performance and latency: The performance and latency of an application can be affected by the location of the infrastructure it is hosted on. If low latency or high performance are critical requirements for an application, hosting it on-premises or in a nearby data centre may be a better option.

Data sovereignty and compliance: In some cases, data sovereignty and compliance regulations may require that certain types of data or applications be hosted on-premises or in specific locations.

The decision to move an application to the cloud will depend on the specific needs and requirements of your organization. It is important to carefully evaluate the pros and cons of cloud and on-premises deployment to determine the best solution for your needs.

Operating Model: DevOps vs Managed services

Based on our experience working with large-scale migration programs, we have developed the following model: a score of 1 to 5 is given for each parameter and weighted average. This will give a general indication whether to migrate and what model to select on migration.

Rapid Application Rationalization Model

Based on our experience working with one of the Largest European banks that has started its migration journey to Azure in 2019 with more that 600+ application in scope, we have developed model. This model when applied to a specific application will result in score, based on score (1-5) for each parameter and weighted average. This will give an indication whether to migrate or retain on prem.

There are various models that are available on market, but the idea here is to have a quick assessment with limited setoff parameters to arrive at guidance. Both Technological and no technological aspect of the application is considered to arrive at a decision.

1. *Application Maturity: Maturity of the application, that is how close it is to To-Be state*
2. *Infra Utilization: Utilization of cloud native services*
3. *Elasticity Requirements: Variation of infra requirement, frequent changes to infra sizing*
4. *Integration Complexity: Integration complexity with dependent systems*
5. *Migration Complexity: Complexity of the application with total time for migration*

Parameters	Score	Importance	Weightage
Application Maturi-ty	1 (Mature) to 5 (Evolving)	Medium	1.25
Infra Utilization	1 (Limited) to 5 (Maximum)	High	1.35
Elasticity Require-ments	1 (Limited) to 5 (Maximum)	Medium	1.15
Integration Com-plexity	1 (Tight Coupling) to 5 (loose Cou-pling)	Low	1.1
Migration Complex-ity	1 (High) to 5 (Low)	High	1.35
Operating Model	1 (Managed) to 5 (DevOps)	Low	1.1

Table 1

Model Interpretation

1. Application Maturity

When we say maturity, we don't mean the stability or quality of the application but more to-wards understanding quantitatively current As-Is state of the application and based on the business vision/ goals the To-Be state. Application Change Rate is an indicator for application Maturity. It is a change rate of the business processes that the applications support, and how frequently those business processes will change in the future. Higher change rate means high-er expected business value. Public Cloud is expected to provide higher business values for ap-plication with long life span.

Importance of this parameter: Medium

Weightage: 1.25

Questions for data collection:

- Are there frequent functional requirements coming in/ pending for the application?
- Are there frequent changes to the application?

Source for data collection:

- Data retrieved from Change Management Tool like ServiceNow
- Application / Product Owner

Anticipated values from user:

- **Continuous Change:** Highly active application with daily or weekly functional changes
- **Regular Change:** application with moderate change rate, on monthly/quarterly basis
- **Rare Change:** Functionality changes rarely, every half year or yearly

2. Infra Utilization

Infra utilization in terms of IaaS, PaaS components including CICD and monitoring require-ments. Expectation is to capture current utilization trend as well as future keeping in mind To-Be state. Public Cloud is expected to provide higher business values for application with high infra utilization requirements with minimal turnaround time.

Importance of this parameter: High

Weightage: 1.35

Questions for data collection:

- What is the current Infra utilization?
- What would be the utilization trend for next 2 years? Will your application need Public Cloud native services, and will it benefit from the same?

Source for data collection:

- Data retrieved from CMDB Tool like ServiceNow or Discovery tool
- Inputs from Application Owner on future trend

Anticipated values from user:

- **High:** High utilization and application will benefit greatly if running on Cloud
- **Medium:** Medium to Low utilization but looking at future state application would benefit if running on cloud • **Low:** Low utilization and not much benefit expected if running on cloud

3. Elasticity Requirements

It focuses on application need for variation of infra requirements like frequent changes to infra sizing due to periodic or seasonal demands. Most of the On-prem platforms fail to serve elas-ticity requirements with lower turnaround time and cost.

Importance of this parameter: Medium

Weightage: 1.15

Questions for data collection:

- Does application infrastructure need change often?

Source for data collection:

- Data retrieved from CMDB Tool like ServiceNow or Discovery tool
- Inputs from Application Owner on future trend

Anticipated values from user:

- **Continuous Change:** Change frequently (daily/weekly)
- **Regular Change:** Regular/periodic change (monthly/quarterly)
- Rare Change: Changes rarely (change is every half year or more)

4. Integration Complexity

Cloud integration is not straightforward. It is about integrating various IT environments like on-prem, mainframe, public cloud, or private cloud etc. For this parameter we focus on under-standing enterprise ecosystem and identify the type of coupling that application has within this ecosystem. That is, interconnectedness with/from other related applications. With ever changing requirements around connectivity protocols and security requirements, most of the On-prem platforms and operation teams fail to keep up with integration and networking relat-ed requirements with lower turnaround time and cost. On the other hand, if your application is tightly coupled with on-prem specific interfaces like Mainframe then it's wise to stay on-prem.

Importance of this parameter: Low

Weightage: 1.10

Questions for data collection:

- Where are the dependent systems running (external / internal facing)?
- How complex is the integration considering coupling type, protocol, security?

Source for data collection:

- Data retrieved from CMDB Tool like ServiceNow or Discovery tool
- Inputs from Application Architects

Anticipated values from user:

- **Highly complex:** Most of interfaces are internal facing with tight coupling like integration with Mainframe or other non-standard solutions. In such cases application is suitable to stay on On-prem platform
- **Medium complex:** Mix of Internal and External interfaces with lose coupling and standard interfacing technical requirements
- **Low complex:** Changes rarely (change is every half year or more)

5. Migration Complexity

It focuses on determining the complexity of migration based on the standard technology stack defined within organization and how well the application supports the accepted architecture standards (development and deployment principles) on Cloud platform. In short to answer this parameter, team need to understand current technical debt as well as technical changes that they need to perform to align with Cloud Standards.

More the technical debt or more the changes required during migration then the Migration Complexity is high and thus less short-term business value.

Importance of this parameter: High

Weightage: 1.35

Questions for data collection:

- Level of current technical Debt?
- Level of changes required to adopt Cloud Standards.

Source for data collection:

- Count of current Technical Debts items from CMDB or other tools
- Existing Data in terms of T-shirt sizing
- Inputs from Discovery tool that give complexity rating
- Inputs from Application Architects

Anticipated values from user:

- **Highly complex:** Complexity translated into migration duration greater than 10 months. In such cases application is suitable to stay on On-prem platform or organization can think of refactoring such application with Cloud Native development.
- **Medium complex:** Complexity translated into migration duration between 6 to 10 months. Such cases can be considered for Public Cloud platform but with organizations need to track such migrations closely to avoid losing business value.
- **Low complex:** Complexity translated into migration duration between 1 to 6 months. Such applications are the best candidates for Public Cloud platform (Re-Host / Re-Platform sce-narios).

6. Operating Model

Many organizations link Public Cloud migrations with DevOps as Way of Working (WOW). Many a times DevOps related change is done during migration phase or sometimes immediately af-ter Public Cloud migration phase. DevOps way of working on Public Cloud Platform demands application teams to acquire additional skills which may or may not be always possible consid-ering team structure and business mandate. Hence due to any constraint if team wants to con-tinue in old way of working (which is Managed Services) then it's wise to retain such applica-tion on On-prem.

Importance of this parameter: Low

Weightage: 1.10

Questions for data collection:

- Current WOW?
- Future WOW?

Source for data collection:

- Inputs from Application Product Owner

Anticipated values from user:

- Managed service: Retain application on On-prem that will work fine with Managed service model
- **DevOps:** migrate application to Public Cloud and it will benefit greatly with DevOps WOW

Model: Above parameters are translated into an excel based model. After data collection and population, the model shows platform recommendation. For example, in below snapshot you would see application is suitable for Public Cloud

Sample Applications

Example Application 1:

An End user Developed Application that allows end-users to create, modify, and manage data in a database. The application is built using Microsoft Access as the front-end interface and Microsoft SQL Server as the backend database. The application is designed to provide an easy-to-use interface for end-users, who may not have programming or database administration experience. The Access forms are used to display, edit, and add data to the database. The forms can be customized to suit the specific needs of the end-users, including the layout, formatting, and functionality. The application must be refactored before moving to azure. The application was with limited user and run to retire state.

Parameters	Score	Importance	Weightage
Application Maturity	1	Medium	1.25
Infra Utilization	1	High	1.35
Elasticity Requirements	1	Medium	1.15
Integration Complexity	1	Low	1.1
Migration Complexity	5	High	1.35
Operating Model	1	Low	1.1

Table 2

Score = 11.6 – on-prem Retain buckets

Example Application 2:

Front end application part of Documents and Record management system. Interfacing with user to retrieve and store documents. The application is designed to handle a large volume of documents and users. It includes security features to protect documents and ensure data privacy. The front-end application interfaces with the back-end system to retrieve and store documents. The application needed to be refactored before moving to azure.

Parameters	Score	Importance	Weightage
Application Maturity	2	Medium	1.25
Infra Utilization	4	High	1.35
Elasticity Requirements	3	Medium	1.15
Integration Complexity	2	Low	1.1
Migration Complexity	3	High	1.35
Operating Model	5	Low	1.1

Table 3

Score = 23.1 – Migrate to Public cloud buckets

Example Application 3:

The COTS application has a client-server architecture, where the client is a web-based interface that communicates with the Oracle backend through APIs. The application uses a standard three-tier archi-tecture, with the client, application server, and database server separated into three layers. There was no significant Platform as a Service (PaaS) utilization or scaling requirement. The application uses standard web security mechanisms, such as SSL/TLS encryption and authentication, to protect user data and prevent unauthorized access.

Parameters	Score	Importance	Weightage
Application Maturity	2	Medium	1.25
Infra Utilization	2	High	1.35
Elasticity Requirements	2	Medium	1.15
Integration Complexity	2	Low	1.1
Migration Complexity	1	High	1.35
Operating Model	1	Low	1.1

Table 4

Score = 12.15 – on-prem Retain buckets

Example Application 4:

Part of Enterprise Content Management stream mainly delivery Output management capability across organization. The application was expected to use PaaS offerings on cloud and had scalability requirement. The system is designed to handle large volumes of data and users.

Parameters	Score	Importance	Weightage
Application Maturity	3	Medium	1.25
Infra Utilization	4	High	1.35
Elasticity Requirements	4	Medium	1.15
Integration Complexity	3	Low	1.1
Migration Complexity	2	High	1.35
Operating Model	5	Low	1.1

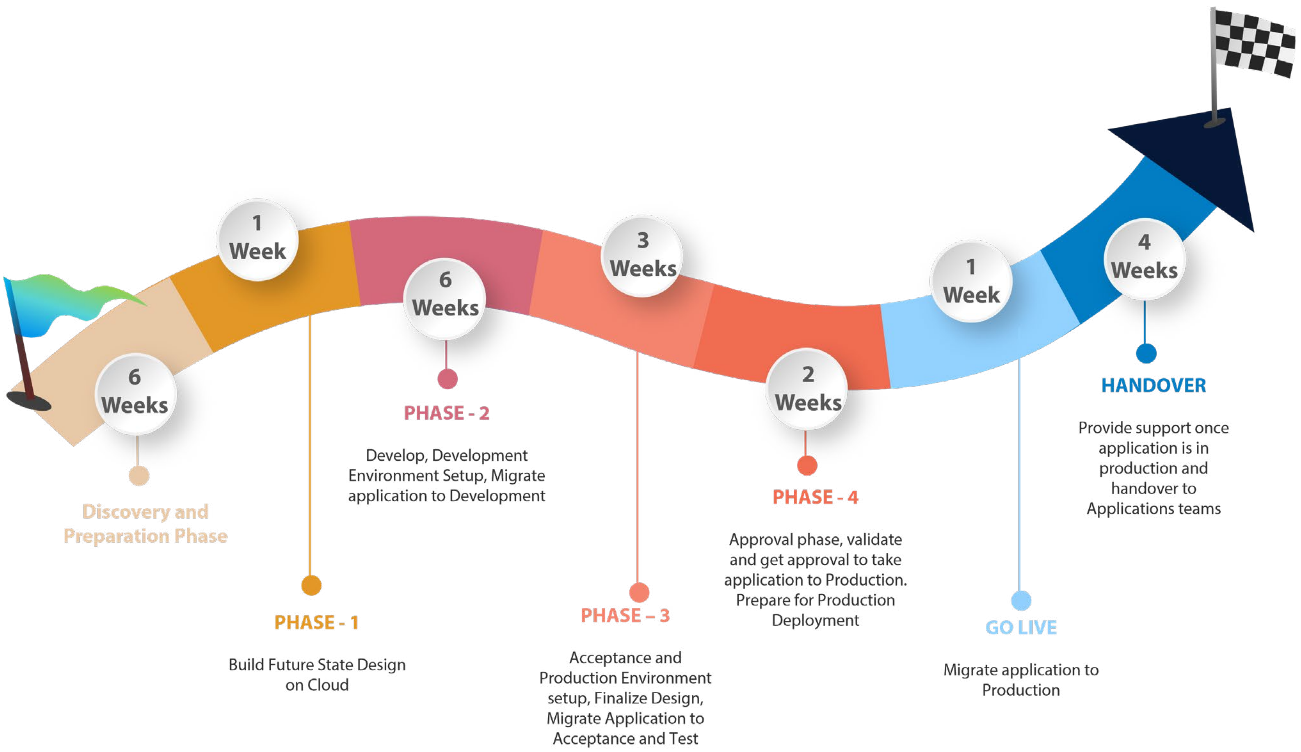
Table 5

Score = 25.24 – Migrate to Public cloud buckets

As you see using the 6 parameters, we can derive and bucket the application into public cloud bucket and on-prem bucket. But do note these are just suggestions and there could be lot of other factors that can influence like type of services organisation is looking for, Licensing implications, Cost, Market value impact, organisation platform strategy etc the decision of moving an application towards cloud..

Execution:

For the actual execution, there are various aspects of an enterprise with a large set of application that needs to think about.



Discovery and Preparation Phase

PHASE - 1

Build Future State Design on Cloud

PHASE - 2

Develop, Development Environment Setup, Migrate application to Development

PHASE – 3

Acceptance and Production Environment setup, Finalize Design, Migrate Application to Acceptance and Test

PHASE - 4

Approval phase, validate and get approval to take application to Production. Prepare for Production Deployment

GO LIVE

Migrate application to Production

HANDOVER

Provide support once application is in production and handover to Applications teams

First, the strategy, i.e., who should drive the migration. Is it the application teams that are currently managing and maintaining the applications or should it be an external team or vendor with the right expertise to drive migration?

Second, how the organization looks at the end state is it thieving to become a DevOps organization with cross-skilled talents, or is it like to look at it as application teams concentrated on App development only and Infra management on cloud by another team

Here are some popular frameworks that can be considered especially for organization who are looking to operate as DevOps before setting up a team for cloud migration:

- Cloud Migration via Factory Model
- Cloud Migration via Enablement Support

Cloud Migration via Factory Model:

A cloud migration factory is a framework used to migrate infrastructure and applications efficiently and effectively to the cloud at scale. The goal of a cloud migration factory is to streamline and automate the migration process, allowing organizations to move large numbers of assets to the cloud quickly and consistently.

Here the Application team's hands over the application to the Factory and the Factory offers the end-to end services including Strategy, planning, deployment, migration etc. i.e., factory takes the end-to-end responsibility of the migration to cloud platform.

Once the migration is completed there is the handover period determined based on complexity of the application. The factory then hands over the application on cloud to the application teams and moves on with the next application.

Problem?	How to solve it?	Learning?
Intake application is not done in detail as factory onboarded the application due to time pressure/unavailability of the SME (Subject Matter Expert)	Detailed intake needed for future applications.	• Intake should be in detail like from scratch for source code and architecture prospective • The Factory Team should check if any risk and blocker present in current as is application while doing detail intake. • Analysis of source code should be done in detail, and it is good to identify blockers as early as possible.
Especially with application in the pool of migration type Refactor, Scope was not articulated clearly what work will done by factory and what work will be done by application teams.	Defining the scope very clearly and have the right agreement on the timelines.	Scope should be clearly articulate what is in factory scope and what is not in factory scope Examples: • Re-writing of source code - Out of scope. • Upgrade of Java code - In Scope • Database Object Code Remediation - Out of Scope. • Database migration - In Scope. • Functional Testing - Out of Scope.
Not getting adequate support from application team as team is focusing on BAU (Business as Usual) issue	Having more connection and having an impediment process in places with KPIs (Key Performance Indicators) and right stake holders onboarded.	There should be agreed and adequate support available from application team with relevant knowledge to address issues and challenges or doing work from block side to keep migration timeline intact. There should be regular handshake between factory and support team to avoid rework.
Application team is not aware of the current infrastructure setup and limited knowledge on the legacy applications	Deep dive sessions with the SMEs (Subject Matter Expert) along with the factory teams architects	Before starting with migration make sure there is enough information available and if possible, make use of discovery tools like Matilda is present to validate the information provided.

Table 6

Cloud Migration via Enablement Support:

Dedicated cloud migration with enablement support refers to a type of cloud migration service where a team or vendor provides dedicated resources with right azure skills and support to help respective applications teams to migrate its infrastructure and applications to the cloud. This type of service can be particularly helpful for organizations that don't have the in-house expertise or resources to manage the migration process themselves.

Problem?	How to solve it?	Learning?
Application teams require a long time to build the required cloud capabilities within the teams to migrate applications to Azure.	Supplement the team with Enablement support team who have deep understanding of design, architecture, and implementation of azure services	Building cloud capabilities within the Application teams is important for migration and this is something the application teams need to focus on even before the start of leaps if possible
Design Patterns not available for on-prem to cloud migrations	Created various re-usable patterns and snippets that the teams could easily refer to and avoid re-inventing of wheels and automatically enforce standardization	Build a repository of the common patterns that will help the teams to refer and re-use.
Delays for dependency to dependency with other application changes.	Be in constant touch with the teams on which you are dependent on, and track closure of issues.	List down the dependencies at the beginning and make sure to have a transparent plan,
Environment readiness for networks delays overall implementation due to incremental updates on the solutions	Having multiple rounds of reviews and using discovery tools like Matilda that can help application teams to get a clear understanding of the current setup.	Pay attention to complete network design and avoid incremental updates as much as possible.
Application teams becoming too much depend on enablement team for pipelines and monitoring setups impacting their cloud knowledge scale up speed	Stop enablement teams' participation towards the application team during Test env itself, re-evaluate migration timelines based on application team members clouds skill maturity.	Stricter implementation of Prepare phase for application teams to scale up on cloud skills for ensuring a faster migration
Data migration from on premise to cloud requires specific skills and they are one time only. Application teams spend too much time in understanding and learning these skills.	Setup a central Data migration Factory that will take care of the migration aspect and required tooling, including setup.	There are quite a lot of activities that are one time only, need to make sure such activities are done via the central or factory team. To increase the speed of migration.
Application teams are not following requested steps to be done prepare phase for example training, application cleanup, solving quality issues etc. causing delay in migration	Push application teams for quicker completion of mandatory courses and training during the preparatory phase and have a clear entry criterion defined for the application teams to start with cloud migration.	Application Teams entry criteria should be evaluated before starting with migration to avoid showstoppers at the later stage.

Table 7

Cost:

To make sure we are keeping the overall cost in check, address the issue at various stages of migra-tion, Intake i.e., when teams start with the Migration, Design, build, Test, acceptance, Ready to go-live and BAU stage.

Training/Learning: Before starting with cloud migration, have a preparatory phase to provide teams with tips, insights, and sessions to increase their awareness of cloud costs. Document and create wikis on various means of optimally using cloud resources. This will not only help teams to come up with a better design, to begin with, but also take the right decision on the sizing of various environments. Usually, cost optimization is becoming an afterthought, increasing migration costs, and making some solutions more expensive to run on the cloud compared to on-prem.

Sleep and Awake: Create a framework to Freeze and unfreeze cloud services when not in use in Dev/Test and Production. Make sure that we need DTAP (Dev, Test, Acceptance, Production) envi-ronments for applications or environments can be set up on a need basis. Plan the migration better to avoid dual costs both on the private cloud and on-prem.

Cost Optimize Deep Dive: Put in a framework and process to optimize cloud resources of applications that are in Production. Provide special attention to high-cost consuming Applications in Develop-ment/Test Phases.

Utilities/Tools provided by the cloud providers or third-party tools can be made use of. Make sure a dashboard, is available for the teams to visualize the cost incurred and provide the right sizing sugges-tions. Try to gamify the entire cost optimization drive which will make the applications teams be proac-tive and push for the changes

Optimize On-prem: During migration or after migration if there is no transparent process put forth that will make sure the on-prem resources are cleaned and tracked, the organization might end up incurring the double cost.

There should be a push towards and clean factory setup to Analyse and follow up with on-prem ven-dors for migration-bound Apps to optimize/ clean up on-premises resources.

Conclusion

There are many potential benefits to cloud migration, including cost savings, improved scalability and flexibility, and increased access to advanced technologies. However, there are also potential challenges and risks to consider, including security concerns, vendor lock-in, Delay in Migration, Doble cost and the need for ongoing maintenance and management.

In general, a successful cloud migration will involve careful planning, including identifying the specific business needs and goals that the migration is intended to address, evaluating potential cloud providers and technologies, and developing a detailed plan for the migration itself. The conclusion of a cloud migration project will depend on the extent to which the project has met its goals and delivered the intended benefit.

About the Author

Shashi Kiran Masthar
Technology Manager
(Cloud)

About the Mentors

Manish Pande
AVP-Senior Principle
Technology Architect

Viral_Thakkar
AVP - Senior Principal
Technology Architect

For more information, contact askus@infosys.com

Infosys®
Navigate your next