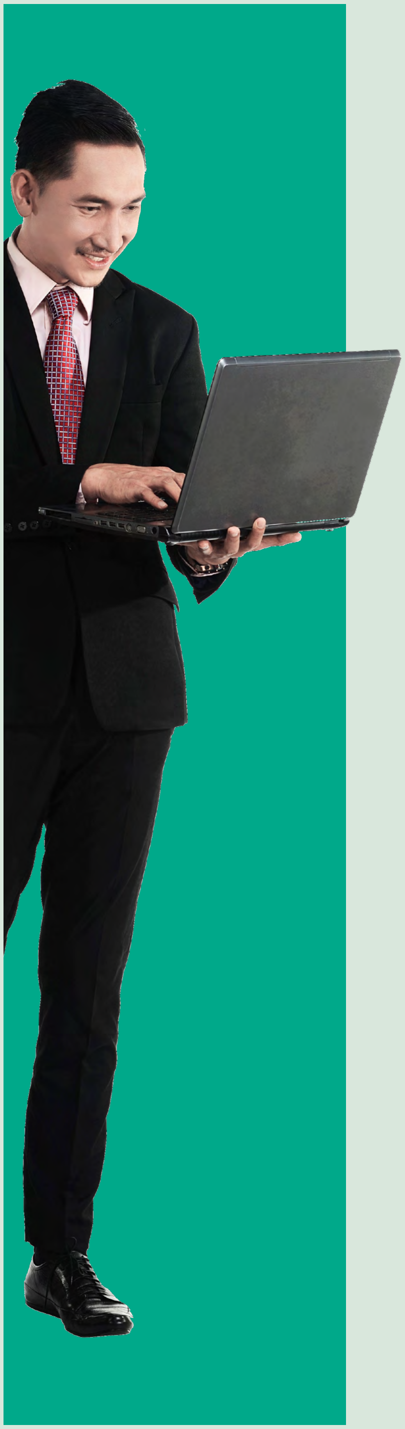
WHITE PAPER



ACCELERATING DIGITAL TRANSFORMATION WITH THE PUBLIC CLOUD

Abstract

This paper focusses on how organizations can accelerate their digital transformation by migrating their workload to public cloud and innovate faster as well as take advantage of the cost saves offered by the public cloud. The aspects described in the paper are curated from our experience of performing cloud migration across our clients leveraging a factory-based model, pointing out on ten key recommendations that can benefit organizations embarking on a similar journey to the public cloud.

Infosys®
Navigate your next

DIGITAL TRANSFORMATION AT SCALE

The pandemic in fact had made it a mandatory requirement for every organization to accelerate their digital transformation journey by multiple times and at scale. The traditional approach to leverage cloud selectively is no longer possible. Organizations that were using the above strategy are incurring huge efforts and cost of running the IT assets in data centers with over-provisioning and underutilization of assets of disparate technology currencies with fragmented monitoring capabilities leading to a multitude of issues failing to support the growth requirements of the business. With the passage time since pandemic there is huge impact across industries to rethink public clouds as an extension to the data center or if possible, replace the datacenters where it makes sense and viable to accelerate the digital transformation with agility and at scale. Many of our clients that we work with had already embraced public cloud as a critical pillar of their IT Strategy. After all, the estimated $3 trillion value potential of cloud is certainly for those who are ready with a plan of action to embrace cloud now.

PUBLIC CLOUD ADOPTION – CLOUD NATIVE DEVELOPMENT VS. WORKLOAD MIGRATION TO THE PUBLIC

While public cloud is a great place for building new capabilities often this takes more time and are better suited for building new capabilities (green-field applications developed with modern software engineering practices – Automated CI/CD etc.) as part of a transformation initiative for a given area, lines of business within an organization. However, to get the real cost saves and scale benefits from cloud, there is a need to have a strategy to migrate the on-premises workload to the public cloud. In our experience we see that around 10-15% of the cloud consumption would be using cloud capabilities to build new apps and platforms using cloud native stacks, platforms the remaining needs to find a place in the migration bucket. From our experience we see that each of our fortune 500, CIOs today takes an OKR of a certain % cloud consumption year on year to show the public cloud adoption strategy in action and the commitment to lower the capex and the operating cost while with an increased agility and time to market benefits to the business stakeholders.

MIGRATING TO THE PUBLIC CLOUD

While public cloud such AWS, Azure, GCP are all quite matured with their offerings, organizations starting their journey with public clouds need a clear approach to determine the target state of their various IT assets it is running on-premises. This is often not an easy exercise and thus a non-starter for many due to the following reasons.

- No single version of truth for the IT Assets (CMDB) leading to no clear view of systems and tools deployed across the data centers owned by the enterprise.
- The organization design across different lines-of-businesses having their own focus not aligned to migrating their workload to the cloud rather focus on building new capabilities.
- Not having contracts for commercial off-the-shelf applications & tools running on-premises to move to the public cloud.
- Need for remediation of the code for the custom-built apps built on legacy technologies, technology currency challenges for OS, Application Frameworks (.net, java)
- Code vulnerabilities, credential management in the application code base
- Multiple source-code control systems (legacy) or absence of a source-code control systems for applications
- Not having an automated code integration and deployment pipelines (CI/CD)
- Non availability of system appreciation documents due to multiple individuals changing hands in some of the areas and not having a reliable SME
- No mapping of the IT systems to the business capability model – blueprint
- Need for clear vision on the point of arrival technologies on the cloud vis-à-vis the on-premises legacy technologies that are expensive and locked-in with complex license models by the vendors.
- Establishing a business case for funding the migration of the workloads to the public cloud – this often might sound easy but the effort/cost that is required one-time to migrate to the cloud is not a small one as it would run in millions of dollars if the apps to be migrated are in hundreds if not in thousands.

1. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/projecting-the-global-value-of-cloud-3-trillion-is-up-for-grabs-for-companies-that-go-beyond-adoption

THE 10 KEY RECOMMENDATIONS FOR A SUCCESSFUL MIGRATION TO PUBLIC CLOUD

The challenges mentioned above need a step-by-step approach for a successful migration journey to the cloud. The following are 10 key recommendations that are curated from our experience with multiple clients to provide a guided approach for an organization planning to embrace public cloud to optimize its cost and increase business agility. The following sections attempt to deep-dive into each of the key recommendations in greater level of detail.

[2] https://cloud.google.com/learn/what-is-finops

[3] https://www.comptia.org/content/articles/what-is-paas

1. Effective IT Strategy and an Operating Model for Cloud Migration

It’s often critical for the organization embarking cloud adoption by migrating its on-premises workload to clearly define what cloud means to its overall IT Strategy and to help support its overall business objectives and growth prospects. Successful companies are approaching this not only from capex reduction and cost-saves perspective with clear goals established year or year for a 5-year horizon but also how this would be tied to improve agility to support the ever-evolving business dynamics requiring always-on, resilient infrastructure and application platforms.

The real challenge is to have an operating model that can help make this realization possible due to the sheer amount of change management required for the organization to go through this to come to terms with. We have seen this work well when the drive for changing to the cloud based operating model is pushed top-down from the CIO to both the lines-of-business heads as well as infrastructure head (mostly in the on-premises world the divide is very clear) and a single leadership is clearly defined to drive the change to cloud adoption, setting up of a centralized cloud platform teams, architecture teams, Center of Excellence to evangelize cloud adoption across the organization. It is also critical to establish a central governance for managing cost of cloud with a tool-based approach – referred to FinOps - to ensure the cost projected for the business-case for moving workloads to cloud is in alignment to realize the projected savings while accelerating business transformation.

Once the above is established and matured which takes several months or a year depending on the cloud adoption strategy, it is time now to define the roadmap to cloud which often done by the lines-of-business leaders working with the business and the enterprise architects and the application teams under them to determine a vision for their area of business (retail banking for example).

While new capabilities can be built by the application teams directly on the cloud leveraging cloud native capabilities offered by the cloud service providers (CSP) like AWS, Azure, GCP, we see our clients build them in a cloud neutral manner using Platform as a Service model (ex: RedHat OpenShift Platform) on any of the cloud provider to start with providing portability across public clouds. In some of our clients, the journey to cloud started with having the PaaS environment established on a private cloud before venturing into the public cloud.

The focus of this paper is migrating the workloads to public cloud and not about building new applications ground-up.

4. https://www.gartner.com/en/documents/3905663

2. Application Portfolio Rationalization to determine the Application disposition.

The need for application portfolio rationalization is often a precursor to beginning a transformation journey for organizations and this is a well-established process with a methodology and toolsets. The most-common methodology we advocate is Gartner's TIME model to arrive at a disposition of applications as Tolerate, Invest, Migrate and Eliminate plotted as a 2x2 matrix against application quality and business value.

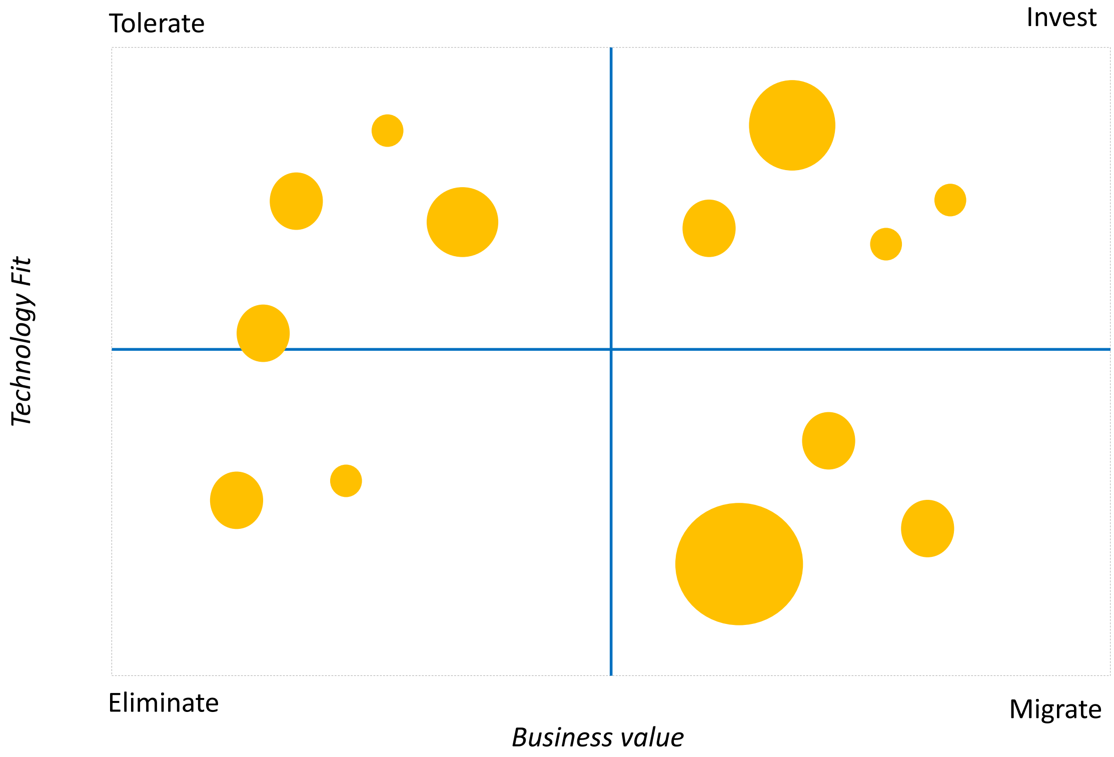
This is often done by the enterprise architects aligned to each of the lines of business within the enterprise after defining a target state business capability model for each of the portfolio with a mapping of existing applications to the model. According to Gartner, it is important to apply top-down analysis and TIME categorization to prioritize opportunities for portfolio improvement using business and technology fitness, risk and cost. In the diagram above, each of the bubble refer to an application falling under different categories of the TIME model.

The candidates for migrating to the public cloud often are the ones that do not have investment from the lines of business teams but are required to be maintained to support the business like those fall under the Tolerate & Migrate categories. These applications might require remediations like operating system and framework upgrades, at times some modernization of the application with a better design. We have also seen sometimes even a product upgrade (if the application is a commercial off-the-shelf one) might be done under this category to keep the lights on for the business.

Eliminate category of applications would require effort to decommission them from the data center. Invest category applications mostly would be under the "Transformation" roadmap of the enterprise where it requires effort to reimagine the business capability and build something cloud native and might not be a good candidate to just move to the cloud though that is still a possibility in some cases.

At this point it is also important to establish the business-case of moving to cloud and there is a need establish a FinOps function that tracks the overall saves by moving these workloads to the public cloud and start tracking the same as the application migration to cloud starts as described later in the factory model for execution. FinOps is an evolving space emphasizing the need for communications and collaboration between business and engineering teams.

Application Portfolio Mapping using Gartner's TIME Model



[5] https://www.finops.org/introduction/what-is-finops/

[6] https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-migration/aws-landing-zone.html

3. Landing-zone creation and Cloud foundation

As part pf the strategy, clearly laying out cloud foundation is extremely critical before the migration journey starts and in fact this must be started several months ahead or in parallel to the application portfolio rationalization and disposition. As for moving the workload from on-prem to the public cloud basically needs identification of the public cloud provider. Once the necessary contracts are established with the cloud provider, a landing zone needs to be designed.

According to AWS, a landing zone is a well-architected, multi-account AWS environment that is a starting point from which the customer can deploy workloads and applications. It provides a baseline to get started with multi-account architecture, identity and access management, data security, network design, and logging, governance, cost management with a billing framework.

A strong Identity and Access Management (IAM) strategy is essential for an effective cloud migration as it provides a cost-effective, agile, and highly scalable integrated access solution that helps enable new authentication methods.

The landing zone design needs to take care of the hybrid state of on-premises services co-existing with the cloud services like – Active Directory, DNS etc. and it needs to make the required network connectivity (AWS DirectConnect, Azure ExpressRoute) to the public cloud from its data center to make it all look like a single homogenous network for the enterprise users.

The landing zone typically accounts for multiple environments like Development, Testing, Pre-Production, Production along with a disaster recovery (DR) environment. The landing zone design also with network level segregations based on the workload (applications) basis the regulatory retirements (ex: Payment apps).

It is also a good idea to establish a central platform team to define the infrastructure as code (IaC) pipelines by building the automation modules to provision the cloud resources in a controlled fashion using tools like Terraform or AWS CloudFormation template.

[7] https://www.stealthlabs.com/blog/top-8-best-practices-to-accelerate-cloud-iam-adoption/

4. Automation to embed security controls and observability aspects into the pipeline (Infrastructure as a Code & DecSecOps).

Cloud automation is fundamental to achieve a good throughput in application migration to cloud. Basically, this includes automation of both infrastructure provisioning through code (IaC) as well as build, test and deploy of the application code on the provisioned cloud infrastructure. The IaC for the most part covers all the organization controls to ensure usage of designated versions of the operation systems on the cloud (Ex: in AWS there are standard AMIs available for Windows and Linux) by customizing the machine image from the cloud provider with additional security controls. It is often important to ensure this infrastructure-as-code is scanned for security exposures due to any misconfigurations in the Terraform/CloudFormation templates using tools like PrismaCloud . Also, for commercial software (COTS), it is critical to scan the executable package for any security vulnerabilities in its executables part of the automaton process.

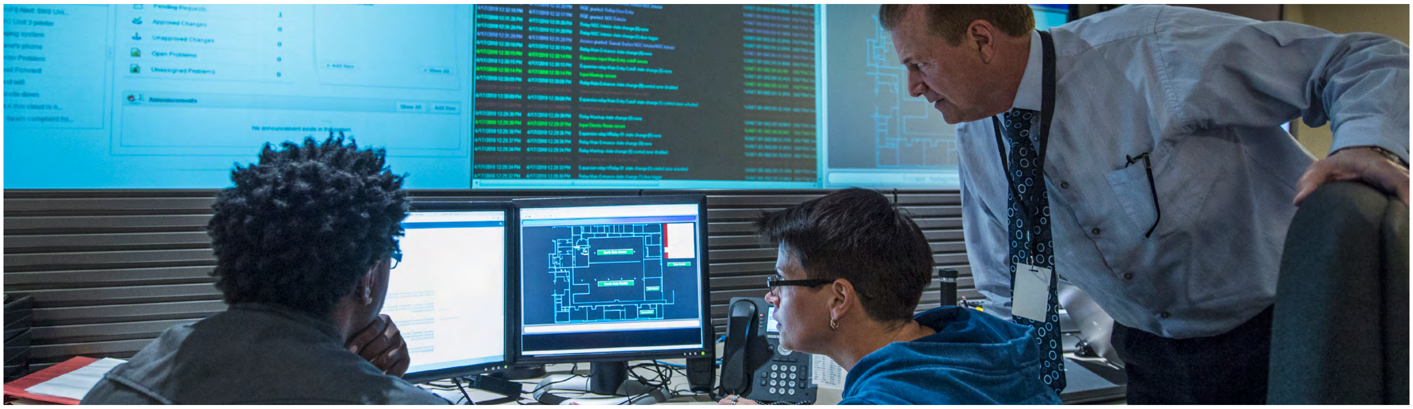
Similarly, the DecSecOps pipelines usually ensures the security aspects like Static and Dynamic Application Security Testing (DAST & SAST) to ensure any code vulnerabilities are caught ahead during the life cycle of migration. The OWASP Top10 security vulnerabilities are constantly updated as they are evolving and hence this requires a robust tooling to be integrated part of the overall CI/CD process. Tools like Qualys can provide the list of code vulnerabilities by scanning the code at runtime while tools like Fortify integrated through the CI/CD pipeline can spit out the exploitable vulnerabilities in the code using a static code analyzer that uses multiple algorithms and multiple secure coding rules built into it.

As organizations need a consistent and standard way to support the operations of application workloads once migrated to the public cloud, the observability aspect becomes very critical and often installing an agent like CloudWatch in AWS or DataDog (for all major cloud providers) in the provisioned AWS EC2 or an Azure VM needs to be part of the automation process. This is nothing but observability as code with Terraform that ensures a standard way by teams to bring in observability part of the provisioning stage.

8. https://www.paloaltonetworks.com/prisma/cloud/infrastructure-as-code-security

9. https://owasp.org/www-project-top-ten/

5. Self-service of cloud resources provisioning to empower teams to drive cloud adoption



One of the key enablers to speed-up the cloud adoption by the teams is to provide empowerment and make the cloud resource provisioning and creating the required build pipelines an automated task with a single-click. There is a cost to implement this end-to-end automation, but the benefit outweighs the cost as it helps the democratize the otherwise centralized process creating more bureaucracy, unnecessary paperwork leading into migration delays and cost over-run.

The following schematic provides a view of how the end-to-end automation as a self-service works in the context of a cloud migration program empowering the application teams to provision the required infrastructure components on the cloud along with the required software installation and configuration through a pipeline-based approach.

The developer can now choose from an intranet self-service portal that has the required levels of authorization, the application (which typically comes from a configuration management system aka CMDB like ServiceNow for example) that requires the infrastructure provisioning in a development environment on the cloud.

All the approvals are in-built as policy-as-code to check whether the requested application has the necessary funding and budgets to provision the cloud resources. Then the developer can select from a list of possible infrastructure patterns as to what best suits the application – like whether it is a 3- tier architecture and 2 tier architecture, multi-AZ or a single AZ and what is the high availability pattern like active-active or active-standby etc.

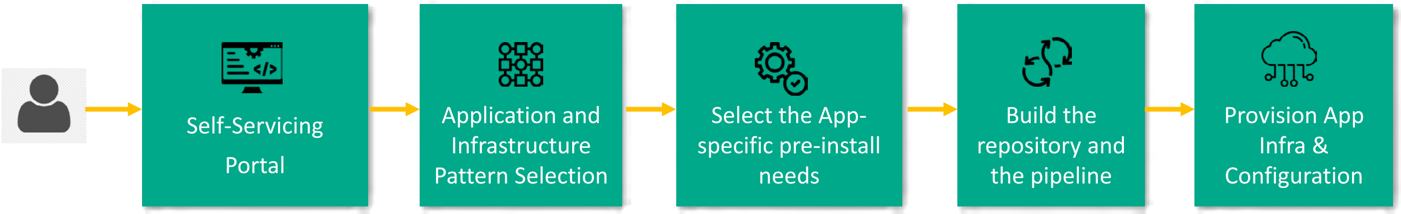
Then the next logical step is to specify what all application software that are required to be installed on the infrastructure once provisioned like Java v11, .Net framework 4.8, npm 8.x, Tomcat 9.0.x etc. The workflow then proceeds to create the repository (git/bitbucket) and creates a pipeline for the team to build the code.

The pipeline integrates the code vulnerability scans and code quality scan tools as well as builds the required logic for code promotion and PR approval process.

The last step of the workflow is to provision the infrastructure and the application specific installation and configuration, perform runtime security scan as well as execute any of the policy-as-code to automate provisioning approvals and also perform the build validation.

This end-to-end orchestration reduces the overall provisioning time from several weeks or days to matter of an hour or two for the development teams to start working with their application on the cloud and the automation efforts are well justified as this increases the adoption across the organization while setting the standards, policies, security aspects as code that has been built in the end-to-end automation tool.

End To End Automation from a developer experience perspective


Self-Servicing Portal
Application and Infrastructure Pattern Selection
Select the App-specific pre-install needs
Build the repository and the pipeline
Provision App Infra & Configuration

[10] https://learn.microsoft.com/en-us/azure/azure-sql/managed-instance/timezones-overview?view=azuresql

6. Cloud-readiness assessment of the on-prem workloads to determine the target cloud patterns along with dependencies in the cloud & on-premises

The most challenging part of a public cloud migration journey from on-premises data centers is to be able to plan the backlog of applications that with some remediations can make it to the public cloud. Assessing the readiness is paramount to be able to come up with the overall program plan from an execution standpoint.

While for some applications it might be straight forward if they don't have many dependencies with other applications, data warehouses, middleware that connects to mainframe systems etc., many of the workloads that we have seen this is often not straight forward. Also, we need to keep in mind that the migration to the public cloud is not a stand-alone effort in any enterprise as there would be multitude of in-flight transformation initiatives that would have an impact on the workloads considered for a cloud move.

The recommended approach to assess the readiness of workload is to take the list of applications that are qualified from the application rationalization exercise that we discussed in section 2 which would have most of the application profile and perform a quick assessment by capturing the response for the following key dimensions from the application owner/team.

- Application architecture to understand whether it is 2-tier, 3-tier, client-server etc.
- Presence of any obsolete technology components used by the application that would not be supported in the cloud.
- Upstream and downstream dependencies (middleware – MQ, External connectivity through firewalls)
- Jobs that the application has part of its configuration.
- Low-latency response time and security requirements that require special handling on the public cloud due to the criticality and the classification of the application.
- Dependency on file shares / networks storage on-premises
- Hard-coded to work on a particular time zone (this would have a challenge in the cloud provider recommends the UTC time zone)
- If it is COTS, checking for a commercial support contract with the vendor and if yes, whether the vendor has a working version of the software on the public cloud or a reference architecture.

Some of the responses to the above dimensions would provide a disposition strategy as to whether the workload is ready to move cloud or require more time to prepare for the same. Also, the other key outcome from the assessment is to determine the type of cloud migration – the four "R" s – Rehost, Replatform, Refactor & Repurchase.

Rehost – Also known as lift-n-shift to the cloud infrastructure from on-prem without any changes to the code.

Replatform – Here the workload needs to be tweaked to leverage the cloud capabilities for optimization but not changing the core architecture. Example could be to leverage a managed database like RDS in AWS vs database installed in a VM on-premises.

Refactor – Here the application is redesigned to be cloud-native and leverages cloud services to provide an optimal solution that takes care of business scale, performance, high-availability, and low-latency requirements. Some of the application batch jobs can be coded using a serverless architecture model as an example (Lambda in AWS) or a changing the database to use an open-source database like PostgreSQL. Sometimes introduce a caching layer in the design using the cloud provided cache service etc.

Repurchase – this is more of commercial off-the-shelf applications that are not supported on the cloud requiring a new product that is built for cloud or moving to a SaaS based offering given the same business capability.

Once the above assessment is performed, based on the logical grouping of applications by the lines-of-business and the owners, multiple waves can be formed for executing the migration. These waves can be planned in such a way so that the amount of support required from the application owners, technical SMEs would be optimal to come up with the target state architecture for each of the workload as well as the required work by the client's cross-functional teams (Security, Network, Identity & Access Management, Risk) to provide their approval and enable opening the required ports to communicate to the on-premises resources as well as provision the new security certificates, private keys, IAM roles in time. But the devil is in the details and often a manual-effort of capturing all the details required for a given application is time-consuming and may not be comprehensive and a tool-based discovery is mandatory.

11. https://developer.hashicorp.com/terraform/cloud-docs/policy-enforcement

12. https://www.openpolicyagent.org/docs/latest/

7. Tool based application data discovery with dependencies (with on-prem systems)

Each of the public cloud service providers do provide a discovery tool, it is also not uncommon for customers to deploy tools to understand a given applications' dependencies by scanning the applications communication in the network and bring up those IP addresses in a report often helps the migration team not to miss any critical dependencies. Tools in this category are capable of doing this with both agent based as well as agent-less while scanning the traffic to and from the application with other dependent systems.

Tools like Cloudamize, Matilda, Cloudscape can provide agent-based or agent-less discovery and assessment of application dependencies that can aid the migration teams to define the target state architecture patterns on the cloud with interfaces to systems running on-premises.

8. Establishing a Migration Factory Model for execution

Given the repetitive nature of many of the activities in the migration life cycle, organization that are looking at migrating hundreds of apps require a factory model that would allow for clear separation of concerns with a well-defined factory model having multiple well-defined steps supported by automation toolsets to enable a steady throughput.

The below infographics explain how a migration factory model would look like for carrying out migration of Refactor/Replatform category as discussed earlier requiring a change in application architecture for cloud-native design, enhanced security, application resiliency and performance. Intentionally, the Rehost option is not discussed in this paper as it is not the most desired strategy as it increases risk and the overall TCO taking the on-prem system as-is to the cloud.

One common thing that would delay the execution is the amount of approval required from various cross-functional teams in the enterprise and mostly this is ticket-based, and those team would mostly thin in capacity supporting the entire enterprise and not just dedicated for the cloud migration effort alone. Given this it is very important to automate the approvals through policy-as-code and security-as-code leveraging tools like Open Policy Agent (OPA), Hashicorp Sentinel that helps expedite the approvals real-time rather than having to wait for days (SLAs).

Often during the target state solution design for a given on-premises application workload, teams would determine the need for a new service from the cloud-provider that requires white-listing as well as some technical proof-of-concepts to prove the need. This process though not frequent it is better to define this part of the factory model as automating the provisioning of such new service modules using IaC would require time and planning upfront for the factory to continue execution without having to wait in the middle of execution.

For every application once the target cloud pattern is determined to asset the overall cost of cloud resources using the calculator provided by the cloud providers (Ex: AWS Pricing Calculator) and have this tracked using the overall Fin-Ops process established for the migration initiative.

Data migration is another critical activity that requires toolsets like AWS DMS and in case of a change to the database, ex: Oracle to PostgreSQL, AWS Schema Conversion Tool (SCT) is required to assess the impact in the code base due to the database platform change.

Testing the end-to-end flow is often done by the testing teams with various automation toolsets. The one that is unique to the cloud migration program is to test the resiliency of the migrated application on the cloud – across availability zones for example. Gremlin is a chaos engineering tool that can be used to do the resiliency testing. When the client stakeholders have provided a sign-off that the application is good the production readiness process kicks in. Production readiness involves multiple teams application team, support team (cloud, security, DR, DNS and network) and leadership teams' approval for a go-no-go decision with a clear roll-back strategy should there be any issue in the new deployment in production post cut-over of the traffic to the on-premises application. Also, it is critical to ensure that the application and support teams have the most updated runbooks/documentation as well as real-time monitoring dashboards setup to help understand the runtime behavior as production traffic hits the application on the cloud.

13. https://calculator.aws

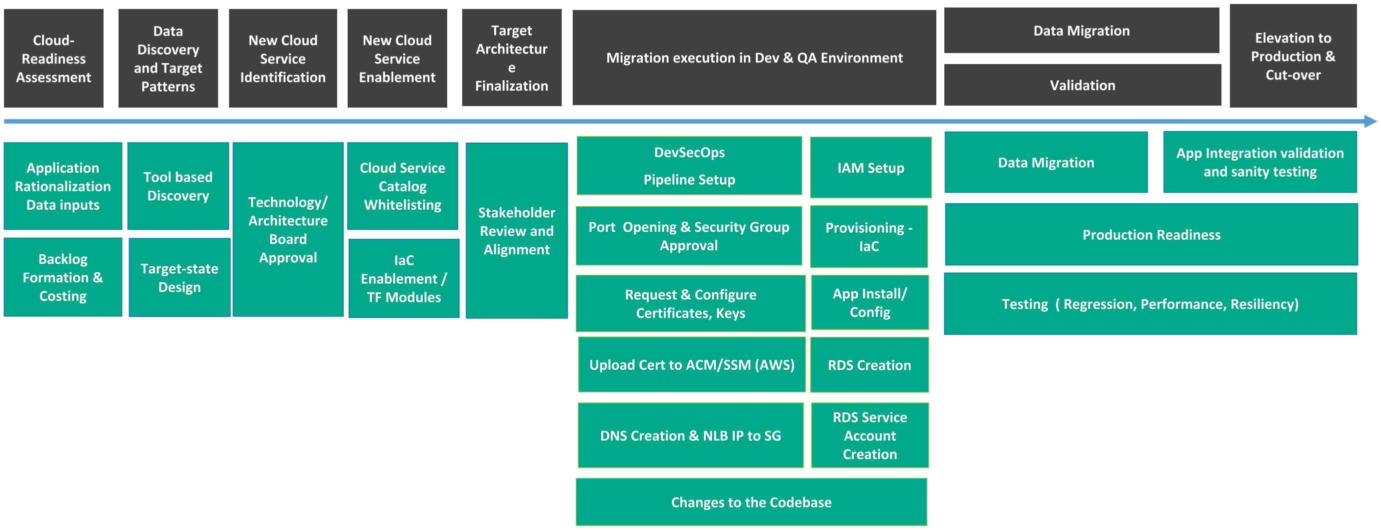
14. https://aws.amazon.com/dms/

15. https://aws.amazon.com/dms/schema-conversion-tool/

16. https://www.gremlin.com/chaos-engineering/

17. https://dzone.com/articles/5-principles-of-production-readiness

Factory Model to Execute Migration to the Public Cloud (Illustration for Refactor / Replatform Strategy)



Cloud-Readiness Assessment

Data Discovery and Target Patterns

New Cloud Service Identification

New Cloud Service Enablement

Target Architecture Finalization

Migration execution in Dev & QA Environment

Data Migration

Validation

Elevation to Production & Cut-over

Application Rationalization Data inputs

Backlog Formation & Costing

Tool based Discovery

Target-state Design

Technology/ Architecture Board Approval

Cloud Service Catalog Whitelisting

IaC Enablement / TF Modules

Stakeholder Review and Alignment

DevSecOps Pipeline Setup

IAM Setup

Port Opening & Security Group Approval

Provisioning - IaC

Request & Configure Certificates, Keys

App Install/ Config

Upload Cert to ACM/SSM (AWS)

RDS Creation

DNS Creation & NLB IP to SG

RDS Service Account Creation

Changes to the Codebase

Data Migration

App Integration validation and sanity testing

Production Readiness

Testing (Regression, Performance, Resiliency)

Note: Here it is assumed that an on-premises application is refactored to use cloud services with better resiliency (multi-AZ) as well as the database is changed to RDS PostgreSQL

18. https://docs.aws.amazon.com/vpc/latest/privatelink/what-is-privatelink.html

19. https://aws.amazon.com/ec2/instance-types/

9. Considerations for migrating the business-critical Tier-1 and Tier-2 applications

While cloud migration of on-premises workloads with a factory-based approach works most of the time, we have seen that there are nuances when it comes to business-critical Tier-1 and Tier-2 applications as they would come with very specific QoS requirements unlike that of the Tier-3, non-critical workloads. Also, as a bare minimum most of them required a multi-availability zone requirement and a multi-region deployment to take care of the DR requirement. Let us look at some specific examples below!

- Creating a separate VPC or VNet to ring-fence a confidential application – ex: Treasury data with special access privileges, data encryption methods etc.
- In case of AWS, creating a private link to third-party cloud-hosted solution in a secured manner – in case of Financial services industry, trading applications having to connect to the market in a secured manner is a critical requirement as with a private link the VPC hosting the mission critical application is not exposed to the public internet (while on-premises it might have leveraged an IPSec/VPN connectivity over internet).
- Again, in the banking industry, traders use very sophisticated applications connecting with exchanges (ex: Bloomberg) with huge computing power requiring the need for more sophisticated compute instance types on the public cloud. Ex: Amazon EC2 C7gn instances are powered by Arm-based AWS Graviton3E processors. They offer up to 200 Gbps of network bandwidth and up to 50% higher packet-processing performance than previous generation C6gn instances. Hence the target design on the cloud need to account for the right selection of instance type and the cost aspects.
- In the retail industry, the online ecommerce web sites and the services that enable them running on the cloud 24x7 for its global users requiring a very high availability with multi-region deployment design for all its components including the database in an active-active design. A common reason for active/active database deployment is to deploy a single application across many regions for high availability and query response times but still have all the data available to all users. In this scenario, user requests are routed to the closest region and any data changes are asynchronously replicated to the other regions by the database layer, ensuring both visibility of these changes globally as well as acting as disaster recovery copies in case of outages.
- While most of the cloud providers do provide their own always-on managed database offering, some of the commercial off-the-shelf products used by clients for sophisticated business domains not yet leverage cloud databases and expect a lot of heavy-lifting to make it work in the public cloud by installing on a VM/EC2 instance and manage replication etc. A good example again could be the QRM (Quantitative Risk Management) application in the Mortgage Services area that helps the client to forecast the future capital positions and is packed with a lot of regulatory requirements. Being a commercial off-the-shelf product and the technical architecture is very complex due to a monolithic design with multiple components wired to each other and sharing a common storage. Realizing this architecture on an EC2 instance on the AWS cloud is not straight-forward as a common file service was required and FSx service is a default choice and to be made available across all availability zones for this purpose. Similarly, there are challenges like auto-scaling aspects – that the product is not built to leverage the native cloud provider's autoscaling features but its own scrips and scheduling jobs.
- Also, it is important to note that most of the above nuances require a lot of back-n-forth with the cloud provider as well as with the vendor if the workload is a COTS product. Due to this it is critical to have all the components of the design on the cloud to be detailed much ahead of the start of the migration to avoid schedule slippage.

20. https://aws.amazon.com/blogs/database/implement-active-active-replication-between-amazon-aurora-clusters-using-oracle-goldengate/

21. https://medium.com/@deep.bbd/qrm-solution-migration-to-aws-d557c77e8904

10. Planning the migration of the shared platforms

While a factory model works best for many of the applications across client portfolio, it needs to be kept in mind that most of business applications would have a dependency on enterprise technology platforms that require special care to determine their cloud strategy.

For example, managed file transfer platforms (MFT) to communicate with external partners, API gateways, Enterprise Schedulers (ex: Autosys, Control-M), ETL (Extract, Transform & Load) and reporting, data warehouse and data lake platforms. It is always beneficial to spend the effort initially part of the cloud foundation efforts to establish a target state technology platform for the above set of enterprise technologies.

It would be prudent to start the migration of these shared enterprise platforms ahead of migrating the on-premises workload as in most cases the change could be very minimal (end-point changes). However, it is also not very uncommon for enterprises to make better choices for enterprise solutions that are more cloud-native than those that were built-for data centers. In such cases, it is a transformation journey that would have an inflight dependency to the public cloud migration effort, and both need to be planned accordingly to avoid any business disruptions.

THE CONCLUSION

The key objective of this paper is to provide the reader a curated approach to look at migrating the on-premises workloads to the public cloud. The paper attempts to cover all the critical aspects from strategy to execution of a successful cloud migration effort to benefit from the cost savings and improved business agility of moving to the public cloud.

The following is a quick recap of the key recommendations discussed in the paper in the previous sections.

The first one was around the need for an effective IT strategy and an operating model to be established before starting the cloud migration by organizations. As this provides a clear vision for the teams to see where the organization is going, it helps to align the Individual teams' priorities with the broader strategy of the organization. It discussed the need for a top-down push with a clear leadership definition to drive the cloud adoption. It then discussed the need to build capability teams to build cloud foundation based on the cloud strategy (single-cloud, multi-cloud) with automation, to evangelize the toolsets and approach to the teams as this is nothing less of an organizational change management in disguise.

The second one focused around getting a closer view of the IT estate and understand the disposition of the applications across different portfolios to determine the opportunities for rationalization as well as define a target business capability model mapping to these systems. Though there are many approaches and toolsets in this space, the paper discussed the Gartner's TIME model to arrive at the dispositions and use that to help build a business case for the potential candidates that might get the benefits of moving to a public cloud.

The third recommendation was about building the cloud foundation that includes landing-zone creation on the target cloud provider and set-up the foundational cloud services integrating to the on-premises eco-system with all the required security and cloud governance aspects.

The fourth one emphasized the need for automating the provisioning of infrastructure through code (IaC) to embed security scans and observability aspects built into the orchestration. Also, it talked about the need for an automated DevSecOps pipeline encompassing code vulnerability and quality scan.

The fifth recommendation was on the need for a self-service portal that the development teams can leverage to provision the cloud resources based on the requirements of their specific application patterns. This is a crucial part of the change management that empowers the teams by reducing the provisioning time to a few hours if not days and weeks and increases the cloud adoption multifold while keeping all-the guardrails codified as policies part of the provisioning orchestration.

The next recommendation pointed out the need to perform the cloud-readiness assessment of the on-premises applications to arrive at a disposition strategy as to whether the workload is ready to move cloud or require more time to prepare for the same. Also, the other key outcome from the assessment is to determine the type of cloud migration – the four "R" s – Rehost, Replatform, Refactor & Repurchase.

The seventh recommendation touched on the need for a tool-based discovery process to understand the application dependencies with upstream and downstream systems, the ports they use to communicate with each other etc. This also helps what ports must be opened on the on-premises firewall for it to communicate with the cloud services.

The eighth recommendation detailed out the process of establishing a Migration Factory Model to execute the cloud migration and the focus was more for Refactor/Replatform application candidates. Here an attempt was made to detail out the end-to-end activities starting from CRA to production go-live and cut-over from the on-premises application with a lot of practical aspects to consider.

The ninth recommendation focused on some of the key considerations for migrating the business-critical Tier-1 & Tier-2 applications as one size does not fit all and special care must be taken to understand the specific set of QoS requirements to be taken care in the cloud by leveraging multi-AZ, multi-region services on the cloud.

The tenth recommendation called out the need for planning the migration of the enterprise shared platforms which are mostly a dependency to the applications moving to the cloud and many of these enterprise shared platforms might require a better alternative on the cloud than their legacy on-premises counterpart. It also stressed the need to have an overall strategy with a forward-looking view for all these platforms before starting the workload migration to the public cloud.

Author

Subramanian Radhakrishnan
Senior Principal Technology Architect

Mentors

Mohammed Rafee Tarafdar
EVP - Chief Technology Officer

Vishwanath Taware
VP - Unit Technology Officer

For more information, contact askus@infosys.com

Infosys®
Navigate your next