



AWS Cloud Adoption: Real Life Reflections

Author
Vikram Naik

Cloud computing has helped transform businesses across the globe, across domains and functions. From basic productivity tools and massive databases to enterprise-grade applications, organizations have successfully managed to move enterprise infrastructure to the cloud. However, as far as their AWS (Amazon Web Services) implementations are concerned, organizations have not achieved what they had set out to, making these implementations look like just an extension of their existing on-premise datacentres, with no value adds or transformation, other than just being on the cloud.

Why do organizations end up there? There could be many reasons - lack of understanding of business drivers, technical knowledge, or even limitations of the landscape that they set up to transform to. Cost overruns, lack of governance, and visibility into the environment can continue to occur on the cloud as well, if you are not careful and choose the right tools and frameworks. Leveraging key AWS features, such as scalability and reliability, in addition to a robust security framework, organizations have achieved their business goals of cost, security, and availability SLAs. Here, I look into key areas which need to be considered while migrating to cloud:

PaaS or IaaS?

One of the key concerns of the top-level executives running data centers, is the cost for upkeep of the infrastructure - software and hardware. This could include maintaining n-1 versions for the O/S, databases, and applying hotfixes for security vulnerabilities. For example, would it be not prudent to move an Oracle database to AWS RDS (Relational Database Service) and forget about upkeep and maintenance of the database and its O/S storage etc? It is not that an easy decision to make. What if your database is a RAC (Real Application Cluster) database? It may have specific load specification requirements and may be architected to work to utilize RAC features?

There are tools that will help you understand how feasible the application is to adopt PaaS or IaaS model on AWS. However, the decision will depend on, among other factors - technical, financial (rework needed to change configuration/code, testing, etc.), and contractual obligations (data center contract expiring) that will determine which model we should use.

Legacy Systems

What do you do when you have to migrate an Application written in Pro Cobol in early 2000, running on a Solaris SPARC machine, and Oracle RDBMS for the backend? What if there are some 40 odd interfaces to this application which are internal and TPAs (Third Party Agents)? To make it more complex, what if this application is the heart of your core applications and whether you touch this application or not, moving the other applications will have an impact? Finally, being such an old system, the people who have built this application may no longer be working with the organization and the documentation may not be top class.

These are real world scenarios that organizations and the partners they employ face during migrations. Your cloud journey will need to take a deep dive into these applications, and tools may not give you the depth of details required to perform a successful migration.

AWS Foundation – Landing Zones

While AWS provides tonnes of documentation, reference architectures, and best practices on how the foundation layer or the landing zones are to be built, there are organizations who have ended up making incoherent and non-flexible architectures. Whether the infrastructure will be self-managed by the organization or managed by AMS or/and MSP, it is extremely important to get this design correct. Choosing the right regions and services; implementing the principle of least privilege, and segregating the roles and accounts, should be envisaged correctly in the design phase.

Selection of additional services – like IDS/IPS/WAF, DNS, Active Directory, Anti-Virus, and how these will be deployed to manage security of the AWS environment. It is also important to know that there are costs associated with these services and tools, and choosing the optimal tool and configuration is important to managing costs. Further, monitoring tools, logs and analytics, and log accounts must also be configured.

Governance Security & Cost

Account segregation is one of the most efficient ways of segregating access rights. For e.g., containing production assets in their own account can help restrict access to key personnel. At the same time, migration planning phase should include tools (AWS Native or AWS partner tools) that can be used to monitor and manage security and cost.

The reports from these tools can be used to fix open issues in security as well as to create cost optimization plan based on resource utilization and actual allocation. This could include conversion to reserved instances for instances that are constantly being used or to reduce resources allocated (Compute, Memory, Storage) by choosing a different VM model or to automate stop/start during non-usage windows.

The migration team may have created lot of resources as part of POC and testing. These must be decommissioned to avoid incurring costs. Account isolation for logically distinct business units will enable at-a-glance billing information for each of these accounts. This will also simplify identification of cost for resource that are ungovernable by a tagging policy.

If the customer is going to manage the environment, ensure that you have clearly defined RACI (a tabular representation of who is Responsible, Accountable, Consulted, and Informed for the entire duration of the project/program) and access controls to the foundation services like IDS/IPS, AV etc. This must be done for traceability when things go wrong. Deployments and configuration changes to the environment must also have traceability. From a small O/S hotfix to a full patch-set, all must be traceable and follow process of Change Management.

Infrastructure as Code (IaaS)

One of the most visible skill set issues with Cloud engineers, who have graduated from the traditional Infrastructure streams, is that they still believe in provisioning resources manually, using the AWS console. The lack of focus or adoption of IaaS, right from inception, causes lot of stress in managing the environment at later stages.

This is more evident, when the migration involves small landscape or footprint and the team considers it futile to build the IaaS templates, scripts, and deployment/configuration process. This leads to slip in standards across the board – different AMIs, different filesystem configurations and sizes, different methods for configuring filesystems, different patch levels or services running on a server, etc.

To alleviate or eliminate this problem, it is most important to ensure your design phase of the migration journey, gets in to nuts and bolts of IaaC and should cover all your resources from landing zones to VPCs, etc.

Using scripting or coding (in their own languages) for infrastructures: installation and start of an http service, configuring app server, creation of directories for log files.

IaaC enables you to have a tightly controlled environment and provides the following benefits

- Ensure a repeatable and reliable process (no human intervention error factor), including the ability to handle rollback mechanisms.
- Productivity - "one-click deployment" rather than a set of manual tasks that will allow you to go faster.
- Traceability to explain, understand, and facilitate analyzes (during post mortem).
- More Declarative scripting than Imperative or Procedure-based scripting to ensure a stable environment state.

In order to meet strict project timelines, the migration team may look for shortcuts to create resources using the AWS Console, as a one-time action. This one-time action becomes repeated and there is no traceability and control. All migrations must emphasize the need for IaaC, for all deployments and must have the necessary skillsets to write code, scripts, and templates to achieve this.

IaaC can be deployed using the AWS service Cloud formation and scripts. Eventually, every interaction with AWS should be done through it, either directly using it or using over-layers such as Sceptre or Troposphere. Other tools like Terraform are also widely used by customers for IaaC.

Conclusion

In conclusion, migration to a cloud providers, like AWS can be a nightmare or a cakewalk. It all depends on how much you follow the documentation, best practices, and reference architectures. Project timelines are dreadful, and so are pressures on the executive to manage costs – by exiting a data center or decommissioning applications. However, the key business goals that are to be achieved should be realistic – either cost wise or time wise and should be reflected in the project plan for migration.

About the Author



Vikram Naik

Sr. Project Manager

Vikram comes with a rich experience of over 13 years in solution designing and implementation of AWS and Azure. He has worked on complex projects involving various cloud technologies, virtualization, storage, and Windows domains. Vikram has been involved in architecting and implementing projects for a range of platforms with high scalability/elasticity, performance, etc. He holds an AWS Certified Solutions Architect – Professional Certification and is a certified VMware Certified Professional 5 – Data Center Virtualization (VCP5-DCV).

LTIMindtree is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 700+ clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by nearly 90,000 talented and entrepreneurial professionals across more than 30 countries, LTIMindtree – a Larsen & Toubro Group company – combines the industry-acclaimed strengths of erstwhile Larsen and Toubro Infotech and Mindtree in solving the most complex business challenges and delivering transformation at scale. For more information, please visit www.ltimindtree.com.