AWS MIGRATION WELL ARCHITECTED FRAMEWORK

Phases, strategies and best practices in migration with the AWS Well Architected Framework



AWS MIGRATION WELL ARCHITECTED FRAMEWORK

AWS migrations involve the process of moving any workload from an on-premises environment, hosting facility, or other public cloud, with the goal of generating IT cost savings and improvements in productivity, business agility, and operational resilience.

This migration process consists of 5 phases: 1) Evaluation of opportunities, 2) Discovery of the portfolio of workloads and planning, 3) Design of the applications, 4) Migration and validation, and finally 5) Operation.



Once the opportunities have been evaluated and the application portfolio identified, there are a series of strategies, as well as a set of best practices and considerations that can be helpful to carry out a successful migration to the AWS cloud.



MIGRATION STRATEGIES TO AWS



When it comes to migration strategies, AWS follows a series of recommendations based on Gartner guidelines, in particular the 5 R's, Rehosting, Refactor, Revise, Rebuild and Replace. They were later renamed as: Rehosting (lift & shift), Replatforming, Repurchasing, Refactoring / Re-architecting and Remove.



1) Rehosting (lift & shift)

This strategy seeks to bring legacy applications to the cloud that cannot be migrated or take advantage of the functionalities of the cloud. For example, workloads that run as they are and that have not been touched in years and that it may even be the case that the organization does not have updated documentation or that the team or supplier that developed it is no longer linked to the business. For this type of workload, it is still possible to take advantage of the fact of migrating to the AWS cloud, since in terms of costs it can mean savings of up to 30%.

One of the advantages of this strategy is that once the workloads are in the cloud, many of the AWS services can be taken advantage of, such as volume backups. In addition, a strategy of refactoring to modernize them can be later used.

2) Replatforming

Another strategy to consider is to bring the workloads with the current architecture onpremise but replacing the underlying platform with those provided by AWS. For example, migrate a database to **AWS RDS**.

This has the advantage that some of the operations that had to be performed to keep the platform on-premise would be relegated to AWS, such as updating the database engine software or implementing infrastructure security.

It is advisable to identify these services. Some of them are **AWS RDS, AWS ElasticSearch, AWS AWS DynamoDB - Document DB**, among others.



3) Repurchasing

It is also viable to consider the possibility of switching to other products, for example bringing functionalities to **SaaS** platforms. It is worth evaluating the costs of maintaining systems (engineers, licenses, etc.) and the need to focus resources on the business.

For example, does it make sense to keep a suite like Atlassian's (Bitbucket, Confluence, Jira, etc), if there aren't any special requirements from the business side? No. It is preferable to focus on business logic and consume tools such as **SaaS**.

4) Refactoring / Re-architecting

This strategy is oriented to take advantage of the functionalities that AWS offers, since it seeks to redesign the workloads taking into account the native technologies of the cloud.

This is perhaps the one that can involve the most changes but is the most beneficial. For example, decoupling a monolithic application into microservices allows each service to scale independently, have its own development cycle, and independent deployment technology.

It also allows to include improvements on workloads to, for example, take advantage of other resources provided by AWS, balancers (AWS Application Load Balancers), web application firewalls (AWS WAF), Content Delivery Networks (AWS CloudFront), among others of the wide AWS product range.



5) Retire

Some workloads will emerge in the discovery process that may not have been identified until the moment it is decided to migrate. Many of these workloads may no longer be useful for the business, for example, applications that are not inventoried and that are not being used.

These workloads can simply be shut down on-premises without the need to migrate to the cloud.

These decommissioning workloads can represent savings from 10% to 20%, since it would stop investing in on-premises infrastructure, licenses, and their maintenance.

6) Retain

In the discovery stage, workloads will emerge that do not fit into any of the above strategies. These applications can be "paused" for migration, and analyzed again in a next iteration.

Hence, the migration of those applications that make the most sense for the business can be prioritized.



WHAT IS THE AWS WELL-ARCHITECTED FRAMEWORK?

The AWS Well-Architected Framework is a set of recommendations and considerations that should be taken into account when architect solutions on AWS. With this framework, it is possible to constantly measure application architectures against best practices and identify areas for improvement.

The good architecture framework was created from the experiences learned by AWS engineers and architects in thousands of deployments around the world. It consists of five pillars (by their names in English): Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization.





WHAT IS THE AWS WELL-ARCHITECTED FRAMEWORK?

PILLAR	DESCRIPTION
Operational Excellence	Ability to run and monitor systems to deliver business value and continually improve support processes and procedures.
Security	Capacity to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
Reliability	Faculty of a system to recover from infrastructure or service interruptions, dynamically acquire computing resources to meet demand, and mitigate interruptions such as misconfigurations or transient network problems.
Performance Efficiency	Aptitude to use computing resources efficiently to meet system requirements and maintain that efficiency as demand changes and technologies evolve.
Cost Optimization	Capability to run systems to deliver business value at the lowest price.



BEST MIGRATION PRACTICES WITH THE AWS WELL-ARCHITECTED FRAMEWORK

The importance of the AWS Well-Architected Framework in the migration process is vital, even with each of the 6 R's migration strategies, as each of the pillars have some degree of participation in decision-making. For example, for **Rehosting and Retirement,** the most important pillar is **Cost Optimization**, since with these strategies significant savings can be obtained compared to on-premise.

For the **Replatforming** strategy, there will be some pillars to consider over others, surely giving more weight to Operational Excellence, Reliability, and Performance Efficiency.

With the buyback strategy, it would surely be targeting other groups of pillars such as **Security, Reliability and Efficiency in Performance**. It is worth mentioning that with this strategy, the **Cost Optimization** pillar may seem the least adequate to shore up, but the costs must not only be considered in terms of licenses, but also in terms of benefits to obtain when focusing business resources.

In the **Refactoring/Re-architecting** strategy, all the pillars must be thoroughly analyzed, since they will allow evaluating the current architecture and asking the right questions when re-architecting a solution.



For example, if it's needed to decouple a monolith with a suitable implementation with microservices, some of the questions that must be ask are already covered in the AWS Well-Architected Framework:

How to mitigate implementation risks? - Operational Excellence How to monitor usage and cost? - Cost Optimization Are we prepared to endure high demands? Can we scale down when the demand is low?- Reliability Is the architecture designed the one that has the best performance? -Performance Efficiency How to protect networks? - Security

AWS solutions are designed to help solve common problems efficiently, safely, and operationally cost-effective. All of them incorporate a detailed architecture, an implementation guide, and instructions to carry out the processes manually and automatically.





References

- [1] Migrating Applications to the Cloud: Rehost, Refactor, Revise, Rebuild, or Replace?
- [2] The 5 R's of Cloud Migration Strategy
- [3] 6 Strategies for Migrating Applications to the Cloud
- [4] Considering a Mass Migration to the Cloud?
- [5] A Process for Mass Migrations to the Cloud
- [6] The 5 Pillars of the AWS Well-Architected Framework
- [7] The AWS Well-Architected Framework whitepaper



partner network

Advanced Consulting Partner

Public Sector Partner Immersion Day Partner Oracle Competency

Amazon EC2 for Microsoft Windows Server



distribution

aws partner network

service delivery

nubiral

We are a global technology company, specialized in business digital transformation and innovation, with a focus on personalized solutions and professional services.



www.nubiral.com