Cloud Infrastructure Basics

Cloud Foundations-Infrastructure Fundamentals

[Pick the date]

[Type the company name]

Chhaya Nikam

# What is Cloud Migration?

In today’s competitive business world, cloud technology is capable enough to provide cloud services anytime and anywhere.

## **Cloud Migration :**

Cloud Migration is a transformation from old traditional business operations to digital business operations and the process refers to moving the digital business operations to cloud. That means data, applications or other business elements are moved into a cloud computing environment. For example moving data and applications from a local, on-premises data center to the cloud.

This is usually driven by business decisions for cost optimization, finding more agility, or simply because of the update of old systems. Here are some basic examples of general application migration:

* Migrate an application from one on-premises data center to another
* From a public cloud to a private cloud environment or vice versa
* From an on-premises server to a public cloud environment
* Migrating an application from one cloud provider to another

## Cloud Migration Checklist/phases/stages

* **Discovery Phase**
  + Analysis of the existing system landscape and infrastructure
  + Formulating KPIs
  + Understanding the key pain points of the legacy infrastructure
  + Determining success drivers for the future cloud infrastructure
* **Design Phase**
  + Selecting a cloud deployment model
  + Choosing a cloud vendor
  + Documenting the infrastructure architecture
  + Planning the migration project
* [**Migration**](https://sam-solutions.us/cloud-migration-strategy-guide-5-steps/#migration)
  + Infrastructure migration
  + Application migration
  + Data migration
* [**Going Live**](https://sam-solutions.us/cloud-migration-strategy-guide-5-steps/#going-live)
* [**Ongoing Support**](https://sam-solutions.us/cloud-migration-strategy-guide-5-steps/#ongoing-support)

## On-premises to cloud migration process :

Every business starting from small to large organizations follows slightly different process for cloud migrations. Some of the common elements which are considered before cloud migration are

* Evaluation of requirement and performance
* Selection of cloud provider
* Calculation of operational costs

The basic steps which are followed as follows

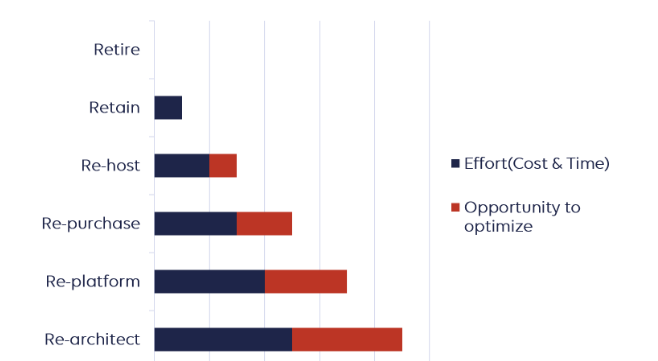
* Establishing migration goals
* Creating a security strategy
* Replicating existing database
* Move business intelligence
* Then switch production from on-premises to cloud

**Cloud Migration Strategy :**

**Ref:** [**https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/**](https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/)



These six strategies differ in their complexity, opportunity to optimize and effort (money and time). The graph below should help you understand which of the strategies is easiest and which is more complex.



### Re-architect

This is the most advanced application migration strategy. Put simply it is a redesign of your application in a more cloud-native manner. In general, you create a [serverless](https://aws.amazon.com/serverless/) version of a legacy application.

#### Example

* Migrate your on-premises Oracle database to [Amazon Aurora PostgreSQL](http://aws.amazon.com/rds/aurora/)
* You can choose to modernize your monolithic application by breaking it down into smaller microservices and leverage services like [Amazon Elastic Kubernetes Service](https://aws.amazon.com/eks/), [AWS Lambda](https://www.stormit.cloud/aws-lambda/), or [AWS Fargate](https://aws.amazon.com/fargate)
* **This type of app migration strategy was used in our case study:** [**InScope Choses StormIT and AWS for Deployment of their AML Solution**](https://www.stormit.cloud/blog/inscope-case-study/)

#### Pros

* Highly increased efficiency, agility and improved cost
* Eliminating dependency on custom hardware or proprietary technology platforms
* No infrastructure/servers to maintain

#### Cons

* Careful evaluation of partners/vendors required
* Rearchitecting usually grows in time and effort
* Requires a very good understanding of all aspects of the application, compliance, code, design, and so on

Example: lets say I have a java Monolithic Application, there is java app running on tomcat server connecting to a Database and also storing some files in a file storage.

Requirement is to convert this Monolithic to Microservices architecture using the cloud native features



Solution :

For file storage : s3 can be used

For Data archival : s3 glacier can be used

For hosting web services: Amazon elastic container service

For database: dynamodb and RDS with postgres or mysql can be used to save licence cost

### Re-platform

We also have Re-Platform which is sometimes called Lift and Reshape, because you can potentially take advantage of some of the special capabilities of the AWS cloud.

#### An example of this app migration strategy:

* Take MySQL database, then deploy it using [Amazon RDS MySQL](https://aws.amazon.com/rds/mysql/)
* Migrating your application to a fully managed platform like [AWS Elastic Beanstalk](https://aws.amazon.com/elasticbeanstalk/)

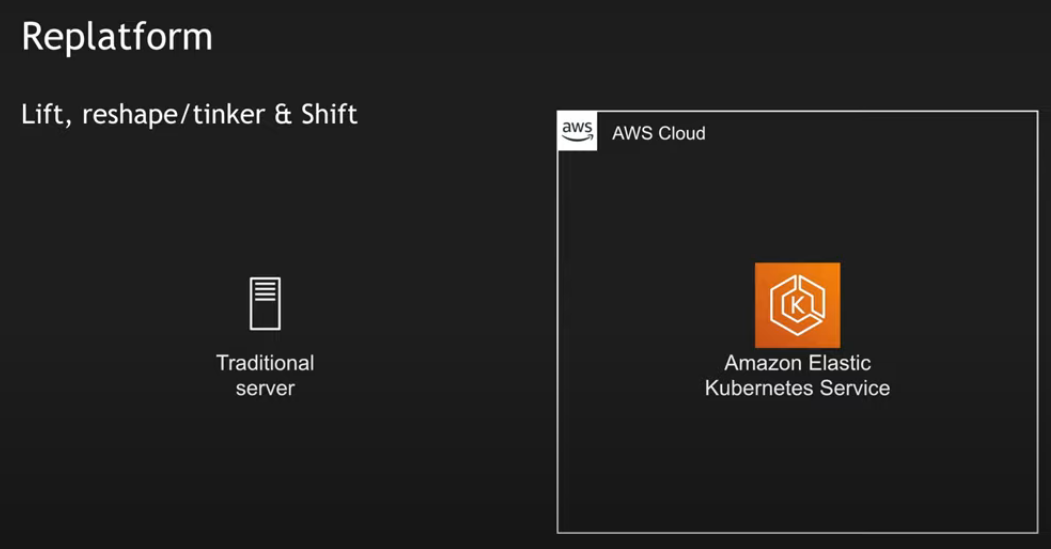
#### Pros

* Option to use services that are close to existing infrastructure without requiring code change
* No dependency on OS, hypervisor or underlying physical hardware platform
* Opportunity to utilize modern OS/DB features/functionality
* In some cases, it is possible to use [automated tools](https://www.stormit.cloud/blog/best-aws-tools-for-migration/)

#### Cons

* Requires additional planning and coordination and these projects may be time-consuming and costlier
* Necessary knowledge of these “new” cloud services

Example: We have a traditional web application running on a traditional server , slightly the application can be changed. Like the application can be containerized and deploy the containers to eks



### Re-host

One of the most basic ways of migrating to the AWS Cloud is simply to Re-Host something, and this is sometimes called, **“Lift and Shift”**. This means to take your applications or systems and move them exactly as is into the AWS Cloud environment.

This approach is great for organizations that need to migrate quickly, for example, data center leases that are about to expire.

For the majority of applications, this type of strategy is also the first step, because it’s easier to optimize/re-architect them once they’re already running in the AWS Cloud**.**

#### Example:

* Move MySQL database to an [Amazon Elastic Compute Cloud (Amazon EC2)](https://aws.amazon.com/ec2) instance on AWS.
* **This type of app migration strategy was used in our case study:** [**Times of Malta Seamlessly Migrates to AWS Cloud with StormIT**](https://www.stormit.cloud/blog/times-of-malta-migrates-to-aws/)

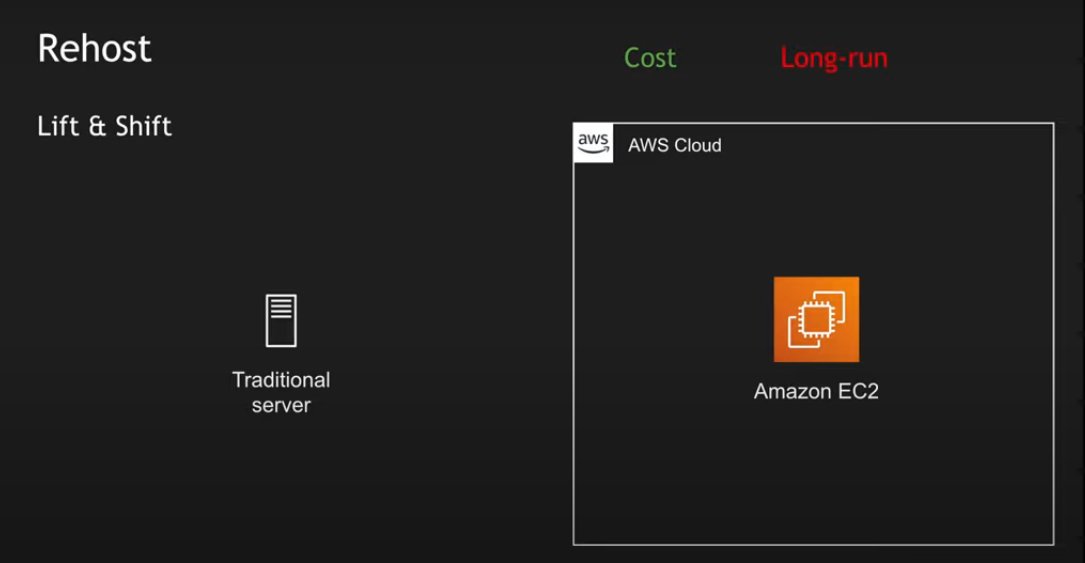
#### Pros

* Speed of migration
* Reduced risk of migration
* Could be automated or assisted by tools.- [AWS + third-party tools](https://www.stormit.cloud/blog/best-aws-tools-for-migration/) natively support this migration strategy
* Migration of whole systems (databases, VMs) with configuration

#### Cons

* You may not use AWS services that give you operational advantages like managed databases, Amazon RDS, DynamoDB, Aurora, Amazon Redshift, and so on
* Potentially inherit performance issues or other problems

Example: get all the runtime for traditional running application and deploy to cloud like in a EC2.



### Re-purchase

This involves abandoning the existing software that you have and migrating to a cloud-first application. This migration can take two forms: transferring your software license from an on-premises server to AWS (e.g. using a [VM Import/Export](https://aws.amazon.com/ec2/vm-import/) tool), or completely replacing your current application with SaaS options. These solutions can usually be found via the [AWS Marketplace](https://aws.amazon.com/marketplace/search/results/?page=1&filters=FulfillmentOptionType&FulfillmentOptionType=Saas) or [AWS Service Catalog](https://aws.amazon.com/servicecatalog).

#### Example:

* Changing your [web application firewall(WAF)](https://www.stormit.cloud/blog/what-is-a-web-application-firewall/) to the one available on AWS(AWS WAF) with managed rules
* Swapping a self-run email system for an online email-as-a-service offering

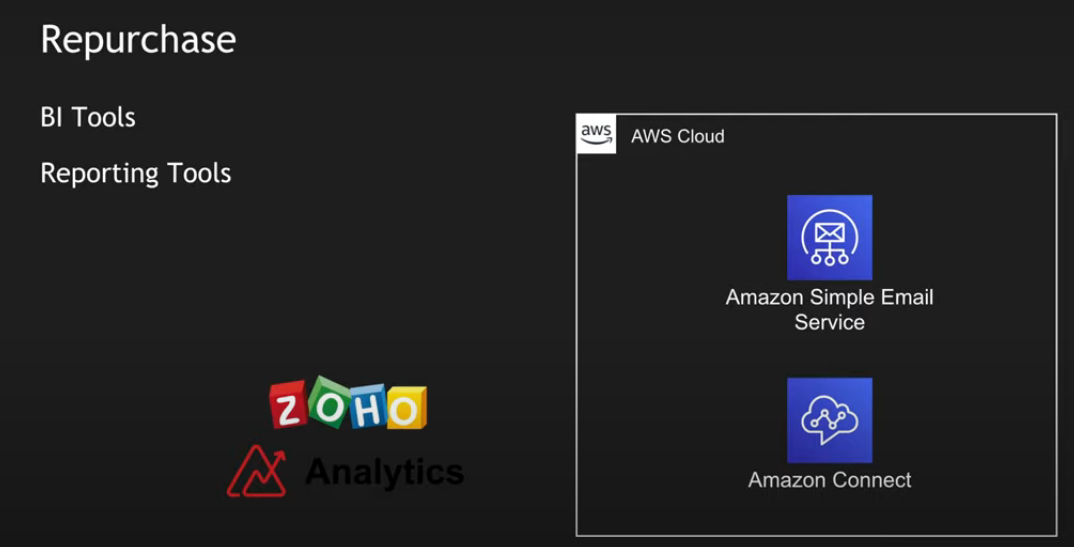
#### Pros

* Reduces effort/increases speed of migration
* Replace legacy systems with modern applications and SaaS

#### Cons

* Difficulty with an understanding of dependencies in your current systems
* Your team needs to learn how to administer a new cloud system

Example: discard the traditional application and buy a new service.

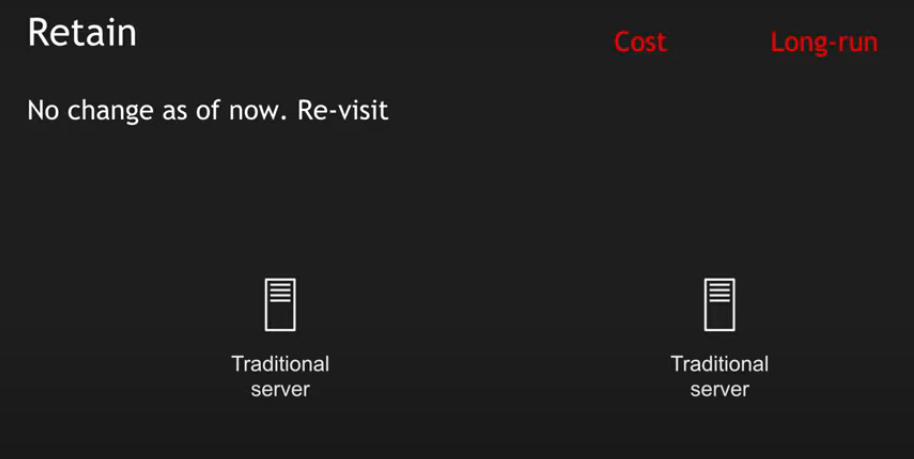


## REATAIN

You can decide to keep applications exactly as is, before a re-evaluation sometime in the future.

#### An example of this app migration strategy:

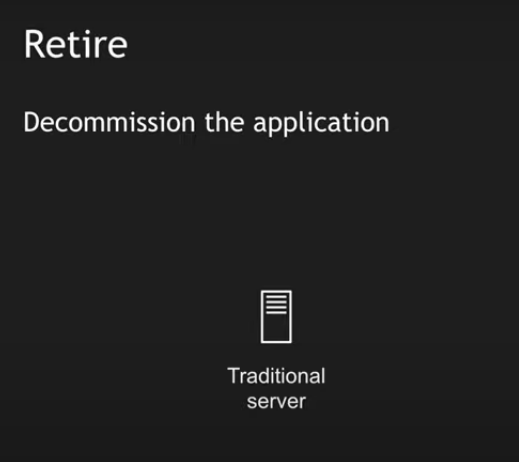
* Unsupported OS and applications
* Legacy applications that do not have a business justification for migrating to the cloud



## Retire

After application discovery, you can also decide that you don't need these new applications in the future, so you’re just going to retire them. This is **just getting rid of them** as you’re not going to use them anymore.

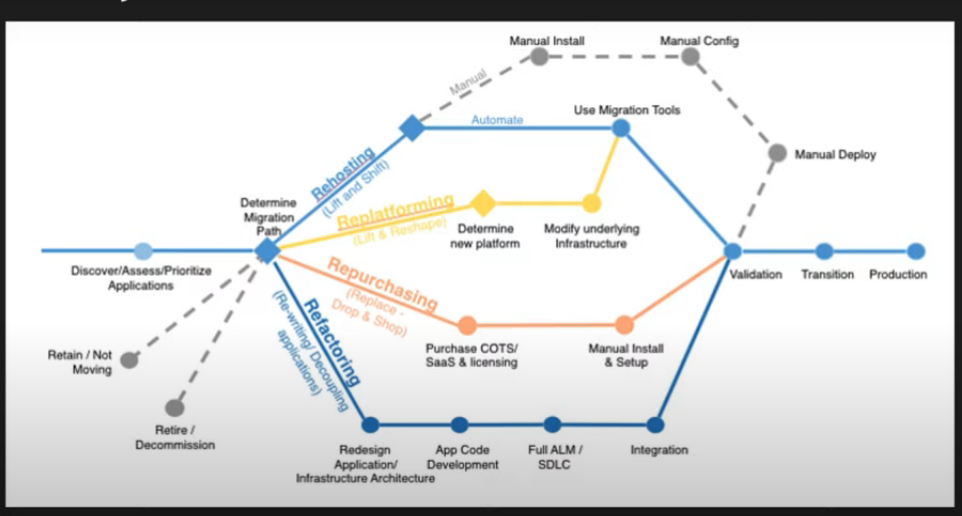
Example: we moved the application to cloud platform running on a traditional server, Now you no longer require the traditional server. So you can remove it and do no maintenance of the application on prem



## How to decide which application migration strategy to choose?

This decision always depends on the specifics of your current application. If you have two or more applications, you could, for example, **start with the simpler ones and migrate them**, and when you gain more knowledge and experience, migrate the more advanced applications.

Making the right decision also involves **understanding how your new environment will operate once you get your data to the cloud** and what the specific benefits of using an AWS Cloud environment are for your business.



## Benefits of cloud migration :

1. **Scalability:** Scalable enough to support various workloads and users. So it offers to expand without impacting performance.
2. **Performance:** Moving into cloud provides higher performance and customer satisfaction as compared to traditional business processes.
3. **Productivity:**As it manages the complexity of infrastructure, so improved productivity is more focused with a continuous process of growing business.
4. **Flexibility:** It allows to use the services flexibly as well as from any where and any time cloud services can be accessed as per demand/need.
5. **Cost:** Moving into cloud technology offers reduced cost in managing, operating, upgrading and maintaining IT operations or infrastructure.
6. **Security:** Security is a major concern which is taken care by cloud service providers.
7. **Profitability:** As it follows pay per use model so it delivers a greater profitability to the customers.
8. **Agility:** It is flexible enough to go with rapid changes in technology and it provides producing newer and advanced setup quickly as per requirement.
9. **Recovery:** It provides backup and recovery solutions to businesses with less time and upfront investment.
10. Deliver immediate business results

## Cloud migration Challenges :

1. Moving a database is a difficult task as there are large amounts of data involved and mostly transferred over internet.
2. After data is transferred into cloud database, another problem is to check the transferred data is intact and secure as well as there is no data loss has been occurred during this process.
3. During migration a problem arises as some of operations or data are already moved into cloud and some are still available on-premises. So ensuring current system is operational and ensuring on going cloud migration process is taking place correctly needs a careful attention.
4. Interoperability becomes a problem as it is not easy to establish a perfect communication in between existing applications and newer cloud environments.
5. Using cloud services, getting good with newer cloud procedures, managing resources and cloud activities requires trained IT professionals who can work in the cloud eco system.

[**https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html**](https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html)

**The pillars of the AWS Well-Architected Framework**

| **Name** | **Description** |
| --- | --- |
| **Operational Excellence** | The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value. |
| **Security** | The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture. |
| **Reliability** | The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it’s expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth, best practice guidance for implementing reliable workloads on AWS. |
| **Performance Efficiency** | The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve. |
| **Cost Optimization** | The ability to run systems to deliver business value at the lowest price point. |
| **Sustainability** | The ability to continually improve sustainability impacts by reducing energy consumption and increasing efficiency across all components of a workload by maximizing the benefits from the provisioned resources and minimizing the total resources required. |

In the AWS Well-Architected Framework, we use these terms:

* A **component** is the code, configuration, and AWS Resources that together deliver against a requirement. A component is often the unit of technical ownership, and is decoupled from other components.
* The term **workload** is used to identify a set of components that together deliver business value. A workload is usually the level of detail that business and technology leaders communicate about.
* We think about **architecture** as being how components work together in a workload. How components communicate and interact is often the focus of architecture diagrams.
* **Milestones** mark key changes in your architecture as it evolves throughout the product lifecycle (design, implementation, testing, go live, and in production).
* Within an organization the **technology portfolio** is the collection of workloads that are required for the business to operate.

When architecting workloads, you make trade-offs between pillars based on your business context. These business decisions can drive your engineering priorities. You might optimize to improve sustainability impact and reduce cost at the expense of reliability in development environments, or, for mission-critical solutions, you might optimize reliability with increased costs and sustainability impact. In ecommerce solutions, performance can affect revenue and customer propensity to buy. Security and operational excellence are generally not traded-off against the other pillars.

Ref: <https://youtu.be/1O4f2jOmbPg>

<https://www.stormit.cloud/blog/6-rs-aws-migration-strategies/>

knowledge check

Accidental deletion 🡺 Soft delete

Backup for on prem Virtual machine 🡺 Azure backup