

# Road to the CLOUD

## Three Approaches to Cloud Migration



### Lift and Shift to the Cloud



The 'Lift and Shift' model replicates current infrastructure in the cloud.

This model is a strategic choice to achieve greater flexibility and a broader range of services and features. It is often chosen when expensive software license renewals, costly hardware replacement, or security issues associated with the end of support life arise.

- 1 Determine which architecture fits the needs of the existing systems.
- 2 Work with vendors to begin transferring licenses of existing software to the cloud.
- 3 Select a container system such as Docker to replicate existing software configurations in cloud environments.

- 4 Migrate data storage to cloud environments using backup and restore.
- 5 Retire the old systems once approved by stakeholders.
- 6 Your application is now in the cloud. Next steps include optimizing the architecture to get the benefits of the cloud.

#### Best under these circumstances:

- Immediate need to replace on premise servers or avoid costly license renewals.
- Committed to moving all resources to the cloud.
- Systems are well-documented and well-understood.
- Willing to incur extra short-term costs to accelerate cloud deployment.
- Relatively stable systems.
- Software can be deployed to the cloud.

#### Pitfalls to watch out for:

- Scope creep beyond infrastructure design changes.
- Insufficient diligence with planning, requirements, and automated testing.
- Arbitrary deadlines that don't account for technical demands.
- Planning system changes, such as introducing new features, during the migration and cutover.
- Lack of preparation for the high levels of security and monitoring required in the cloud.

### Start Brand New in the Cloud

Creating an application from scratch has never been cheaper, faster, or easier. Even though new systems may be started in the cloud, they still access external systems and data. Here, we outline steps to catalog and migrate data to the new environment.

- 1 Gather and prioritize a backlog of requirements from the existing system.
- 2 Determine a data migration plan, and start cleaning data in source systems where possible.
- 3 Identify best ways to redesign and migrate functionality of the existing system.
- 4 Run test data migrations along with development to better prepare for migration and build quality test data.

- 5 Check in with stakeholders frequently to ensure their vision and needs are being met.
- 6 Determine a cutover plan and prepare for final deployment, data migration, and retirement of the old systems.
- 7 Deploy and enjoy.

#### Best under these circumstances:

- Personnel are experienced in providing balanced, well-designed architectures.
- Commitment to specific infrastructure principles.
- Strategic vision that aligns infrastructure decisions with business goals.
- Systems targeted for migration are relatively small.
- Legacy system is very old and cannot be recreated on modern infrastructure.

#### Pitfalls to watch out for:

- Committing to unorthodox and untested architectures.
- Design choices that compromise future extensibility.
- Not planning and coordinating estimated costs with stakeholders.
- Failing to limit architecture options and falling into "analysis paralysis."
- Increasing risk by trying to update and revise the legacy system while building the new one.

### Hybrid Cloud



### Hybrid Cloud

A third option is to run both the legacy and new application simultaneously while you modernize. For this approach, use the Strangler Pattern and place a layer in front of the entire system to intercept all incoming events. Route them appropriately to either a legacy service or a modernized cloud application (or sometimes both).

As legacy features are fully migrated, they are turned off in favor of the cloud service. Eventually, the legacy system will be shut down and the interception layer either removed or left running to capture events from outside systems.

- 1 Determine the modernization effort and assess the environment to reduce risk of losing data, breaking reports, or wasting effort.
- 2 Analyze the application documents, architecture, business processes, business rules, and associated services.
- 3 Determine the features and data involved in delivering the MVP, and build the first portion of the new application and interception layer.

- 4 The interception layer calls to the legacy system can be implemented via a message platform (e.g. Kafka, Amazon Kinesis, Microsoft Event Hub, etc.) to route data appropriately.
- 5 Once the routing system is running, new features can be deployed to the cloud as microservices and prioritized based on the Product Roadmap.
- 6 As the migration proceeds, continually confirm and adjust the data architecture, system architecture, and interception layer. This leads to faster, more secure delivery of the new system.

#### Best under these circumstances:

- The legacy system is a monolith of medium to large complexity (usually web or mainframe systems).
- Incoming and outgoing events can be intercepted.
- The underlying technologies for the legacy application have accessible interfaces.
- The upfront cost of discovery and setting up the monolith is significantly less than rewriting a legacy application or moving to a commercial product.

#### Pitfalls to watch out for:

- Not deploying new features regularly to confirm the data migration strategy and interception layer work as intended.
- Failing to ensure all data is caused via a common method, which will cause the legacy and modernized applications to get out of sync.
- Selecting large portions of functionality for the initial parts of the migration, slowing down the work and the validation of associated technologies.

Moving to the cloud is inevitable for most organizations, and there are multiple strategies to make the switch. Understanding the current system layout, cloud services available, and the pros and cons of each strategy can save time and money in the long run. Whether you choose to replicate your systems directly in the cloud, rewrite an entirely new cloud native system, or revise the architecture over time, make sure to do the research before choosing a strategy.

Looking for more information on all things Cloud?

Contact Us