Cloud Strategy for Financial Service Institution

Sachin Yadav (Matriculation No: 3574915)

University of Stuttgart, Pfaffenwaldring, 70569 Stuttgart, Germany

Abstract.

The financial institution has more than 2000 applications with a variety of architectures, several versions of multiple middleware stacks, and various deployment modes. The core banking application is deployed on mainframes, many other applications are monoliths which are using a traditional multi-tier stack on Unix Systems. Many of them are already virtualized, and also there is some initial experience with containers, and also with some small cloud deployments. Some of the applications also follow a service-oriented architecture approach. The 6R technique will be used in our implementation proposal to decide which applications should be moved to the cloud initially. We will evaluate the readiness, relevance, and risk of each application before deciding whether to reshot, refactor, rewrite, rebuild, replace, or retire it. Applications will be ranked according to business significance, ease of transfer, and cloud advantages. Additionally, we will create a reference design that takes industry best practices and recommendations into account for our cloud implementation

Keywords: Finance, Cloud strategy, Architecture, Microservices, DevOps.

1 Migration to cloud

1.1 Cloud is Unavoidable

The Finance industry, like other industries, has recognized the importance of innovation for progress. Many small and medium-sized FinTech organizations have created advanced applications hosted on the cloud, providing impressive features to customers. Cloud technology is now a present reality for banks rather than just a future possibility, as it offers diverse capabilities for improving performance and scalability. Moving to the cloud is the first step toward modernizing digital and application infrastructure, which involves transferring applications, infrastructure, hardware, workloads, and data to a private or public cloud. This process is particularly challenging for the banking industry, given the nature of their business. Ensuring the secure transfer of critical financial data, minimizing service downtime, and complying with regulatory requirements are all important considerations. A detailed cloud migration plan is necessary to address these and other issues. Such a plan will be customized to the specific needs and

skillset of each organization, providing a reference guide for their cloud migration strategy.

1.2 Early Migrant and Target Architecture

The IBM reference architecture can be used to modernize, scale, and profitably change one's banking operation. (See Fig.2). The reference architecture provides a high-level overview of the main features provided by IBM, such as connection, security, and scalability, some of the most crucial factors to take into account when establishing a financial institution's cloud adoption strategy. AWS additionally offers equivalent reference architectures. (See Fig. 3) provides information on the specific implementation process for the cloud-hosted banking application.

1.2.1 Migrating Monolith Application

The monolithic nature of traditional apps makes them unsuitable for the cloud. Monolithic applications are hard to scale as demand increases and are tightly connected. Due to the massive amount of data that is used in the banking sector, developing a monolithic banking application is not only difficult but also time- and money-consuming. These programs are not designed to handle changes that are inescapable in the current environment for banks.

Furthermore, the deployment procedure for a monolith requires that the complete program be released rather than just a few small modifications

1.2.2 Rehosting Simple Monolith Application into Cloud

Application which are loosely coupled and doesn't involved any business logic and not associated with databases we can simply lift and shift those application into cloud without any changes. AWS provide two deployment model for re-host application

1. for complete control over the configuration of your compute environment, including memory and storage settings, and control over operating system patches: migrate application to Amazon EC2 (See Fig.1).
2. for partial control over the infrastructure: use Elastic Beanstalk. Elastic Beanstalk automatically sets up a managed environment for your application. (See Fig.1).

1.2.3 Refactoring complex, tightly coupled application into cloud

A micro service-based application, in contrast to a monolith, is made up of a number of small, loosely connected services that each carry out a specific function. For instance, when a fund transfer transaction is started, a number of atomic services are invoked and carried out in order to complete the provided request.

Because the application is composed of smaller, loosely connected services, adding new features or updating the application may be done so efficiently without deploying the entire thing. The services are also compact in size, which facilitates the adoption of contemporary DevOps processes, tools, and containerization technologies like Docker to implement automated build, test, and deployment. Adopting an agile architecture

design approach like micro services is essential to help achieve the aim of creating new and modern cloud-based apps. [2]

1.3 Migrating CRM

Utilizing commercial off-the-shelf (COTS) goods from outside vendors allows an organization to significantly reduce migration and cloud setup costs. These could be obtained either directly through the AWS Marketplace or as software as a service (SaaS).[3][4]

Some examples of application which can be repurchased:

- Swapping an internally administered email server for an online email-as-a-service offering.
- Replacing a self-built VPN server with a vendor-built appliance.
- Moving from a home-grown CRM system to Salesforce.com or Hub Spot.

1.4 Application that remain on premise

Applications that have not yet been decided to be moved to the cloud immediately due to things like the cost of doing so is not justified by the benefits it would yield or compliance and regulatory difficulties. Additionally, these apps are able to stay in the data center longer than others. To handle these situations, the cloud migration strategy should have a process for reviewing these applications and perhaps moving them to the cloud in the future. For instance, the present migration strategy should be updated if regulations have changed six months into the cloud migration endeavor and the application appears to be a viable candidate to be migrated to the cloud. Additionally, application in scope has real-time latency requirements as it is essentially running your production line. Also, it involves secret information to be stored and hence the use of filesystem encryption, demanding the sole control of yours. It is unlikely (still not impossible) that a public cloud data center is the right candidate for your application [4]

1.5 API Managment

Application programming interfaces (APIs) offer a way for applications or services located on various servers to communicate with one another. The cloud strategy must include API management in order to create microservices-based cloud native applications. Building, implementing, scaling, and managing APIs shouldn't be difficult as part of effective API administration. These APIs will also probably be used by other developers. The security of the APIs should therefore be of the utmost significance. Advanced API management platforms should include the following six essential features, according to AWS: "API access control, API protection, creation and design, support for hybrid models, high performance, and customizable developer portal." The API administration tools provided by AWS are shown in Fig. 4. [9]

1.6 Challenge Moving Into Cloud

The process of moving apps to the cloud comes with challenges. In a recent survey, Accenture identified four major challenges that banks must overcome in order to successfully migrate to the cloud. Risk related to compliance and security is the most urgent problem. Regulators do not have the same faith in the cloud as the banking sector has for highly regulated industries like financial institutions. This is a worry, but it can be reduced if industry leaders and cloud service providers collaborate closely with regulators to assist them better understand the security and dependability of the cloud. [6]

When moving to the cloud, it may become problematic if the bank has different IT and business goals. Determining which applications and services should be migrated to and expanded in the cloud first is crucial. This is a key tactic that, if implemented properly, gives banks a competitive edge. Globally utilized core banking systems that run on legacy codebases are effective, secure, and stable. Such programs can be expensive to migrate to the cloud, which makes the financial case for doing so less compelling. Hosting these programs in the cloud as opposed to on-premises would be advantageous if they undergo frequent or significant changes. In addition to them, having little cloud proficiency can be challenging.[6]

1.6.1 Regulation and Compliance

Your business partners and jurisdictions as a bank demand that you are committed to very high requirements. Your options for keeping and managing your information are frequently governed by data protection rules. Executives are worried that by putting everything in the cloud, important details could be overlooked, leading to costly fines and bad press [7]

1.7 Pros and Cons of VMs, Containers and Event Based Architecture

When several machines with its own operating system (OS), memory and hardware resources are hosted on a single physical hardware, they are known as Virtual machines (VMs). Unlike VMs, a container only emulates the OS and typically runs a single application. Although both VMs and containers provide isolation, containers are smaller in size making them faster and easily portable which makes them a suitable choice for microservices based applications hosted in the cloud. Containerization is the process of packaging application and all its dependencies ready to be hosted on a VM or cloud. Hence in a certain sense, they complement each other. VMs can be used to isolate entire systems while containers are used to isolate applications. However, containerized applications can only run on one OS so they do not provide the capability of a VM where any application can run on any host. [6][7] A serverless application is composed of two basic components – functions and events that trigger those functions. AWS serverless compute service, AWS Lambda was introduced in 2014 that offers these capabilities. These applications are completely event driven, follow the pay-per-execution model and zero cost for idle time. This is fantastic for variable traffic management because if

there is no traffic, the cost is zero. Unlike serverless, it is impossible to shut down containers completely when there is no traffic and there are always runtime costs. However, in serverless there are more chances of ecosystem lock-in and one can only utilize the programming language and environments that the cloud service provider offers. Additionally, serverless is comparatively newer and has the potential to include many new and sophisticated tools. There 6 also exist limitations related to the size of the functions that can be built in serverless whereas with containers, one can build applications as large as necessary. [10]

2 Industry best practices

2.1 Avoid Hyperscaler lock-in

Ensuring your freedom to switch vendors gives organizations a workable exit strategy and the negotiating leverage to reach an agreement with vendors. Businesses that are trapped into contracts will suffer when business requirements, regulations, terms and conditions, vendor products, or tariffs change, as they frequently do. . In order to prevent failures of the cloud from having a long-lasting effect on the financial industry, the idea intends to create a secure cloud environment for banks. Are there plans in place, for instance, to ensure that the workload may be transferred to another provider in the event that a cloud provider hosting apps for a big bank were to fail? Many financial organizations are choosing platforms built and maintained in their own 7 regions to host their ecosystem rather than hosting them on other continents as a result of this examination as well as the compliance and regulations in the banking industry. These efforts led to programs like GAIA-X.[11][12]

To successfully follow the strict guidelines and satisfy compliance standards, banks must design systems that enable a smooth shift of workload from one cloud provider to another, often known as workload portability, as well as techniques to minimize vendor lock-in. Not only is it a good idea to have a clear exit strategy in place when shifting to the cloud, but European banks are required to have one. Banks will be able to use the capabilities of many suppliers by utilizing multicloud architecture. They will also have the freedom to simply move their workloads between vendors with the aid of solutions like Kubernetes. Banks will be able to host some of their apps in one hyperscaler and utilize another for other applications thanks to an open hybrid cloud.[12][13]

2.2 Observability and Management of Workloads

Along with open-source services like AWS Distro for OpenTelemetry, Amazon Managed Service for Prometheus, Amazon Managed Grafana, and Amazon OpenSearch Service, AWS offers a wide range of observability services like Amazon CloudWatch, AWS X-Ray, Amazon CodeGuru Profiler, and Amazon DevOps Guru. [14]
An observability and management platform, such as Oracle Cloud Observability and Management Platform or IBM Cloud Satellite, can assist enterprises in managing work-

loads in multicloud by helping them run their workloads effectively across several platforms. These platforms provide machine learning-based management insights. This makes it easier for businesses to manage their IT infrastructure across all platforms, whether local or cloud-based. [15][16]

3 Impact on sustainability of IT

All industries, including the banking sector, are significantly impacted by sustainability. The financial sector and its long-term strategy are changing as banks now adopt sustainability standards.

We can mitigate co2 footprint by taking following concept

1. Circular economy (procurement comes with the appropriate certifications that is production happens under environment considerations. Example: ISO 14001),
2. The 3 Res (RE-cycle, RE-use, RE-furbish) to avoid electronic waste
3. Use of renewable energy in datacenter, consideration of idle resources [17]

4 Organizational Changes

It is impossible for one group to entirely understand all the complexities of every application and platform in a complex financial banking system. Due to their distinct responsibilities in application and platform availability, AREs and PREs essentially fulfill the same function as SREs. These role can help in managing the rapidly evolving and expanding nature of the digital banking sector by ensuring 100% availability and fostering a climate that allows for quantifiable and tolerated risk.

Along with these obligations, Jamil Mina of RedHat says that implementing continuous feedback with teams through the use of "metrics, weekly feedback sessions, joint problem solving, testing, common automation frameworks, etc.), would help mitigate the risk associated with diverging from their essential function—securing the reliability of application services.[18] Therefore, success in the cloud is not just a chieved by technical changes but also requires organizational culture and role modifications. Adopting industry best practices such as DevOps, SRE and well architected framework is an essential part of cloud migration process

5 Conclusion

The cloud provides financial institutions with cutting-edge tools and software solutions that provide them the ability to expand successfully in this fast changing digital banking environment.

A clear cloud strategy, highlighting the end objectives and goals, adoption of modern and agile architectural patterns, knowledge of vendor and tool options, stakeholder involvement, organizational cultural changes, implementation of best practices, and a comprehensive list of risks and challenges that might be encountered are all necessary

for realizing the benefits of the cloud. This paper provides detailed information about these subjects for the finance business and leads the reader to the appropriate source for more details.

References

1. 1 AWS Migration Whitepaper. https://d1.awsstatic.com/whitepapers/Migration/aws-migration-whitepaper.pdf

2. Megargel, A., Shankararaman, V., Walker, D.: Migrating from Monoliths to Cloud-Based Microservices: A Banking Industry Example. In Book Software Engineering in the Era of Cloud Computing, pp.85-108. Research Collection School Of Information Systems, https://ink.library.smu.edu.sg/sis_research/4725

3. 3 https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-guide/migration-strategies.html

4. 4. https://cloudsoft.io/blog/cloud-migration-strategies-5-repurchase
5. Jean-François Lagassé et.al., “Getting Cloud right: How can banks stay ahead of the curve?”, Deloitte, https://www2.deloitte.com/global/en/pages/financial-services/articles/gettingcloud-right-how-can-banks-stay-ahead-of-curve.ht
6. “API Management”, AWS documentation, https://aws.amazon.com/api-gateway/api-management/?nc1=h_ls.
7. Michael Abbott, “Challenges and opportunities in banks’ cloud migration”, Accenture Banking blog, January 27, 2021, https://bankingblog.accenture.com/challenges-opportunities-banks-cloud-migration
8. https://info.cloudcarib.com/blog/3-main-cloud-computing-challenges-for-banks
9. Jordan Shamir, “5 Benefits of Virtualization”, IBM Cloud, April 8, 2021, https://www.ibm.com/cloud/blog/5-benefits-of-virtualization.
10. Hartmut Schlosser, “Docker Vs. Virtual Machine: Where are the Differences?”, DevOpsCon, November 23, 2021, https://devopscon.io/blog/docker/docker-vs-virtual-machinewhere-are-the-differences/.
11. Philipp Müns, “Serverless (FaaS) vs. Containers - when to pick which?”, Serverless.com blog, October 10, https://www.serverless.com/blog/serverless-faas-vs-containers.
12. Joanna Wright, “EU Proposal Takes Aim at Major Cloud Providers ”, WatersTechnology, September 29, 2020, https://www.waterstechnology.com/regulation/7689646/eu-proposaltakes-aim-at-major-cloud-providers.

13. Meghan Rimol, “Gartner Says Four Trends Are Shaping the Future of Public Cloud”, Gartner, August 2, 2021, https://www.gartner.com/en/newsroom/press-releases/2021-08-02- gartner-says-four-trends-are-shaping-the-future-of-public-cloud.
14. Wulf Schiemann, “Cloud Exit Strategy: Ensure Compliance and Prevent Vendor Lock-in”, meshBlog, June 18, 2020, https://www.meshcloud.io/2020/06/18/cloud-exit-strategy-ensure-compliance-and-prevent-vendor-lock-in/.
15. Observability, AWS documentation, https://aws.amazon.com/products/management-andgovernance/use-cases/monitoring-and-observability/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&blog-posts-cards.sortby=item.additionalFields.createdDate&blog-posts-cards.sort-order=desc.
16. Oracle Cloud Observability and Management Platform, Oracle, https://www.oracle.com/manageability/
17. Laura Noonan, “Get Flexibility and Control to Run Workloads on the Cloud of Your Choice”, IBM Cloud, February 26, 2021, https://www.ibm.com/cloud/blog/control-whereworkloads-run
18. Lecture on Green IT by Kristof Werling https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_file_3185389_download.html
19. Jamil Mina, “Bringing reliability to banking services: a new twist on Site Reliability Engineering”, RedHat Blog, June 4, 2021, https://www.redhat.com/en/blog/bringing-reliabilitybanking-services-new-twist-site-reliability-engineerin

Appendix

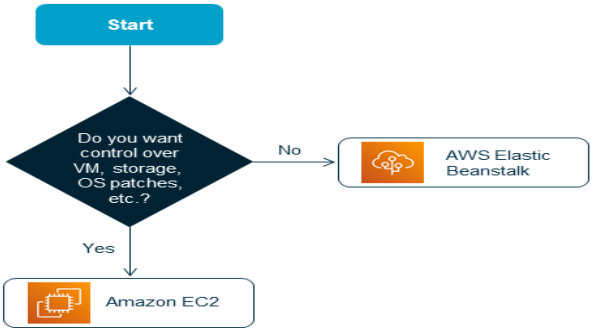

Fig.1https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-net-applications/rehost.html

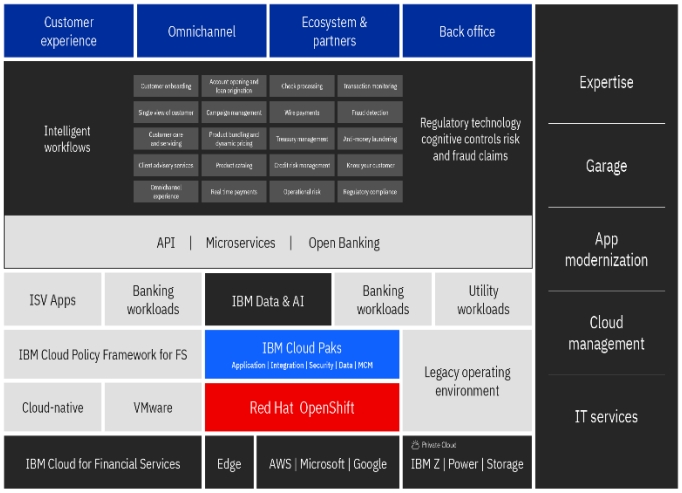

Fig. 2. The IBM Banking Industry Architecture. The Banking Industry Architecture. https://www.ibm.com/cloud/architecture/architectures/banking

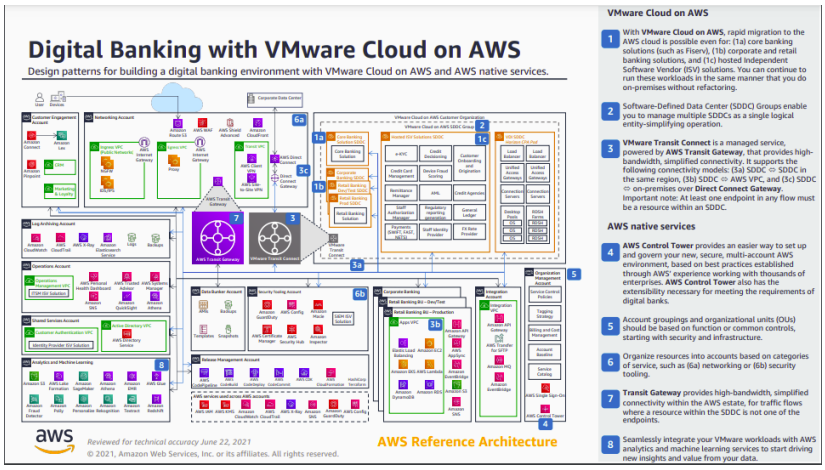

Fig. 3. Reference architecture diagram, Digital Banking with VMware Cloud on AWS. https://d1.awsstatic.com/architecture-diagrams/ArchitectureDiagrams/digital-banking-withvmware-cloud-on-aws-ra.pdf?did=wp_card&trk=wp_card.

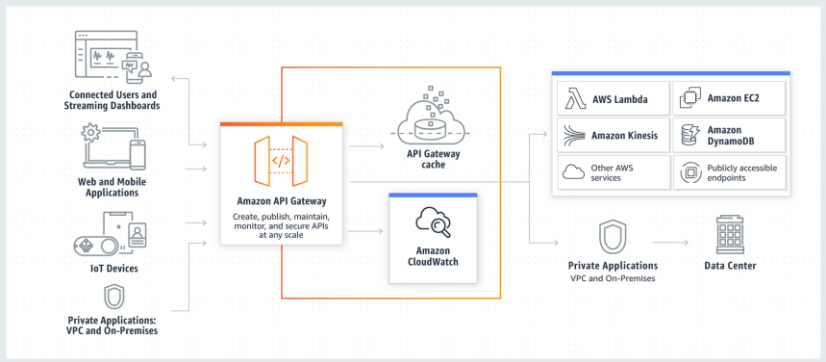

Fig. 4. API Management tools by AWS. https://aws.amazon.com/api-gateway/api-management/?nc1=h_l