

Stepping-Up Banking Infrastructure with ACC

TABLE OF CONTENTS

1	Our Client	1
2	Overview	2
3	Workflow of the project by ACC	2
4	The need for Containerization in Applications	3
5	Let us see what is AWS Lambda	4
6	A planned 3-tier architecture	4
7	The need for landing zones	5
8	The need for Identity Access Management	6
9	Kubernetes For Migration	7
10	Kubernetes Advantages include	7

Our Client

Our client is one of the most famous cultural examples of invention combined with strong business and trade procedures. They have progressed steadily and dynamically over the years, motivated by a desire to provide our customers with financial services that meet the industry's highest quality standards.

They commenced operations in 1994, catering to the needs of both consumer and corporate customers. It is a universal bank having a large financial footprint across the country, with over 2.5 crore customers, over 5,000 distribution sites, and more than 2,000 branches.

Their mission is that they will continue to offer value to all stakeholders by improving the organization's sustainability and emerging as India's most convenient bank with industry-leading financial metrics. Its technology platform supports multi-channel delivery capabilities. It enjoys clearing bank status for both major stock exchanges - BSE and NSE - and major commodity exchanges in the country, including MCX, NCDEX and NMCE.



Overview

In this case different vendors are managing different applications but the main infrastructure to manage all the applications is designed by ACC team, the aim was to bring the AWS accounts which are hosting these applications as child accounts in the new proposed Landing Zone . The Next Generation Landing Zone will be leveraging the features and functionalities of AWS Organizations to centrally organize and manage the AWS accounts that are created under it with the ability to host both external facing applications and internal facing applications.

Workflow of the project by ACC

The project was started with the main aim to have a secure and proper governance of the initially used system. The earlier Infra in AWS cloud was managed according to different applications and business owners thus governance was difficult. It needed a proper upgrade to an efficient landing zone where governance of AWS infra would be easier and efficient.

The client wanted all of its cloud governance under one umbrella; they basically had six accounts basically for security, shared services, log archives, master account and so on to move from on premise to cloud service and a proper landingzone was implemented to govern the multiple accounts of the client.

The client wanted a centralized management of logs in a single account (Log archive account) similarly shared services like direct connect, transit gateway etc to be in the shared services account. The master account to govern the SSO integration which also includes centralized billing of all the accounts in AWS Organizations. We had assisted the AWS Proserve team in building the architecture for the landing zone. Our team assisted in the configurations and the project was completed as promised.

The need for Containerization in Applications

Containerization enables our development teams to work at a faster pace, distribute software more efficiently, and function at a larger scale than ever before. We've packaged over a decade of expertise launching billions of containers per week into Google Cloud so that developers and organizations of all sizes can quickly use the most cutting-edge container technology.

It is a software package that contains all of the necessary dependencies, such as code, runtime, configuration, and system libraries, to execute on any host system. CaaS enables software teams to launch and scale containerized applications to high-availability cloud infrastructures quickly and easily. CaaS is distinct from platform as a service (PaaS) since it makes use of containers. PaaS is focused with deployments of certain "language stacks" such as Ruby on Rails or Node.js, whereas CaaS can deploy numerous stacks per container. Containers and CaaS make deploying and composing distributed systems and microservice architectures a lot easier. Various responsibilities or code language ecosystems can be managed by a set of containers during development. Container network protocol relationships can be defined and committed for use in other environments. CaaS promises to swiftly deploy these defined and committed container architectures to cloud hosting.

Let us see what is AWS Lambda

AWS Lambda is a serverless compute solution that automatically maintains the underlying compute resources for you while running your code in response to events. Changes in state or updates, such as a user adding an item to a shopping cart on an eCommerce website, are examples of these events. One can use AWS Lambda to add custom logic to other AWS services or build your own backend services that run on AWS scale, performance, and security. AWS Lambda automatically executes code in response to a variety of events, including HTTP requests via Amazon API Gateway, object modifications in Amazon Simple Storage Service (Amazon S3) buckets, Amazon DynamoDB table updates, and AWS Step Function state transitions.

There are no new languages, tools, or frameworks to learn with AWS Lambda. Any third-party library, including native ones, can be used. One may also use Lambda Layers to package any code (frameworks, SDKs, libraries, and more) and manage and share it across many functions. Lambda supports Java, Go, PowerShell, Node.js, C#, Python, and Ruby code natively, as well as a Runtime API that allows you to write your functions in any other programming language.

A planned 3-tier architecture

The infrastructure is divided into three layers in this pattern: one public layer and two private layers. The notion is that the public layer functions as a barrier between the private levels and the outside world. The public layers are open to the public, but the private layers are only accessible from within the network.

We want to create some form of high availability in addition to separating the network into three layers. By distributing the application over various Availability Zones, AWS allows you to achieve high availability. Each Availability Zone is a physical data center that is located in a separate part of the world.

The display layer, the business logic layer, and the data storage layer are the three logical levels of a three-tier architecture. The frontend, backend, and database are all part of this architecture in a client-server application like a web application. Each of these layers or tiers performs a distinct function and can be handled separately. This is a departure from the traditional monolithic approach to app development, in which the frontend, backend, and database are all housed in the same location.

The need for landing zones

A landing zone is a set of configurable cloud infrastructure, policies, best practices, guidelines, and centrally managed services. It's the initial step in a factory model application's cloud migration journey. The core parts of cloud deployments are covered by a landing zone, which is made up of building blocks. Governance, data security, network design and logging, as well as multi-account architecture and identity/access management, are examples of these building pieces. They're set up in accordance with the organization's overall strategy and industry norms, ensuring that any workload deployed to the cloud follows best practices.

Critical hygiene elements such as security and compliance are baked in, allowing developers to devote more time and effort to tasks that add value. When these foundations are in place, they can be allowed more autonomy to explore the full potential of the cloud. Meanwhile, because landing zones provide guardrails to prevent the environment from becoming an ungainly, invisible, and costly entity, the overall risk is decreased.

Let's go over the four considerations that a well-designed cloud landing zone should have:

Security and Compliance: Take a centralized approach to security, monitoring, and logging. Landing zones can be used to execute corporate compliance and data residency regulations.

The need for Identity Access Management

The main purpose of Identity and Access Management (IAM) services is to provide the appropriate access to the appropriate user at the appropriate time in the appropriate context of the company. IAM services provide a dedicated digital entity for each individual user, whether it's for external parties such as end-users or business partners, or internal parties such as internal personnel or contractors.

IAM helps administrators protect against security problems by automating a variety of user account-related operations. This includes the ability to create an automated onboarding workflow for employees, allowing access to systems and applications to which they are permitted based on their function. It also features a "one-button" control to deactivate employee access to all systems to which they have been granted access via the IAM platform.

IAM solutions assist firms in meeting industry regulatory requirements while also reducing expenses by reducing the amount of time spent dealing with user account-related issues. Identity and access management standardized and even automates important components of identity, authentication, and authorization management, saving time and money while lowering corporate risk.



Kubernetes For Migration

Kubernetes is a flexible platform for automating and simplifying your container process while also providing instant scalability. companies should obviously use Amazon EKS since it can avoid single points of failure by running across various AWS availability zones. Any apps that are now running on Kubernetes will be compatible with Amazon EKS and will be simple to migrate.

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that lets you run Kubernetes on AWS without having to set up, administer, or maintain your own control plane or nodes. Kubernetes is an open-source framework for automating containerized application deployment, scaling, and administration. EKS on Amazon:

To achieve high availability, it runs and grows the Kubernetes control plane across different AWS Availability Zones. Control plane instances are automatically scaled based on load, and unhealthy control plane instances are detected and replaced, as well as automated version upgrades and patching.

There's no need to set up, run, or maintain your own Kubernetes control plane when you use Amazon EKS. Rather, you can use the Kubernetes Dashboard to control your cluster's applications and individual resources. .

Kubernetes Advantages include

- Control and automate upgrades and deployments.
- Save money by maximizing infrastructure resources through better hardware utilization.
- Container orchestration across many hosts
- Many problems caused by the growth of containers can be solved by grouping them into "pods".
- Real-time scaling of resources and applications
- Application testing and autocorrection

ABOUT ACC

ACC is an AWS Advance Partner with AWS Mobility Competency. Awarded The Best BFSI industry Consulting Partner for the year 2019, ACC has had several successful cloud migration and application development projects to its credit.

Our business offerings include Digitalisation, Cloud Services, Product Engineering, Big Data & Analytics and Cloud Security. ACC has developed several products to its credit. These include Ottohm – Enterprise Video and OTT Platform, Atlas API – API Management and Development Platform, Atlas CLM – Cloud Life Cycle Management, Atlas HCM – HR Digital Onboarding and Employee Management, Atlas ITSM – Vendor Onboarding and Service Management and Smart Contracts – Contract Automation and Management.



www.appliedcloudcomputing.com

Shubho Pramanik

(+91) 90297 20294

shubho@acc.ltd

Rogin Rappai

(+91) 8828478321

rogin.rappai@acc.ltd

also reach us at

