CASE STUDY : Neogrowth

Neogrowth Witnessed Significant Increase in Performance & Cost Savings With AWS

rapyder
Making CLOUD work for you

NEOGROWTH
Lending simplified. Growth amplified.

Client

Neogrowth

Introduction

Founded in 2012 and an ISO 9001:2015 certified player in the SME digital lending space, NeoGrowth is a Mumbai-based platform with a PAN India presence. Registered with the RBI, the organization's approach includes innovative technology, a digital payment ecosystem, and flexible payment options. NeoGrowth's mission is to have a positive social impact on the financial lives of SMEs by leveraging the digital payments ecosystem.

With over 1,000 employees across 29 branches along with 2000+ channel partners, and over 17,000 customers, NeoGrowth serves industries such as F & B, hospitality, healthcare and pharma, automobile, and apparel.

Business Needs

As customers increased, efficiency, scalability, and continuous delivery became priorities driving NeoGrowth to adopt the calability of microservices architecture along with container orchestration. This would support its developers to bring their applications from the development to the production stage at record speed and provide scalability and flexibility of their applications.

As an existing customer of Rapyder Cloud Solutions for managing its applications on the cloud AWS cloud platform, NeoGrowth, however, was still running microservices on a couple of static servers where-in each task demanded varying amounts of memory and CPU, leading to performance inefficiencies. In the absence of accurate resource allocation, the limitation in the memory would negatively impact the entire application, leading to service delays. To address this challenge, NeoGrowth consulted Rapyder's engineers to move to microservices-based architecture and container orchestration with AWS ECS.

Rapyder Cloud Solutions, an Advanced AWS Consulting partner with expertise and experience in successful cloud migration and managed services, completed the transition and deployment of container orchestration.

Implementation

Covid-19 has undoubtedly accelerated cloud adoption by several years. Companies are increasingly leveraging container technology to fast-track application development, deployment, and portability of apps across different platform infrastructures.

Rapyder's expert team of Cloud Architects conceptualized and stitched a powerful and customized solution around Amazon Web Services for the task. Rapyder's team supported the smooth transition to the successful deployment of microservices-based architecture and AWS ECS enablement.

- The team first made the shift to containerize the microservices architecture at NeoGrowth. Unlike the traditional static server deployment, with the application's components and functions in a single instance, microservices architecture splits the application into multiple services that perform different functions, each with a different logical function. These microservices were then placed in containers. Container orchestration leverages tools and platforms to automate, manage and schedule workloads that are defined by individual containers.

- Containerization of the application was then done, followed by container orchestration with the Amazon ECS platform to manage the entire life cycle. This container orchestration is supported in automating the scheduling, deployment, scaling, and overall management of containers across a fleet of servers. Amazon ECS, being highly scalable with high performance, allowed developers to run effortlessly, scale containerized applications, and automatically reschedule containers if required.

- Rapyder's team also supported the client in leveraging spot instances with AWS services, in an economically priced model with steep discount offers, for test and development.

Reaping Rewards

Building and delivering software reliably are key to an organization's success and positive business outcomes in a digital economy. Equally important is the visibility across and within the software coding and delivery processes to ensure high speed and scalability with demand. The transition to microservices-based architecture did not impact the running of the business.

- Developers could now work faster due to smaller code bases in microservices with lesser complexities and dependencies of functions, bringing about better operational efficiency.

- Code reviews and updates were done faster, and failures could be isolated without bringing the entire system. Moreover, developers could leverage different programming languages effortlessly. All these led to continuous delivery, thereby establishing higher productivity.

- Scalability was independent of each microservice where only the required component of an app could be This ensured optimization of resources making the whole process cost-efficient.

- Leveraging spot instances that came with AWS economically priced model also helped in cost- optimization to a large

- Above all, the ease of use of Amazon ECS in itself was beneficial.