coditas

Basix.ai

Accelerating Sales with Machine Learning

coditas

The Client

Basix.ai is a **FinTech sales product** company that helps build team performance, increasing sales by providing an in-depth financial market knowledge pipeline. With Basix, the AI engine learns which activities and prospects to focus on so a **brand's sales team can get better results**.

Project Duration

5+ Years

Problem Definition

Basix.ai wanted to create a system to understand their clients' purchase behavior and cluster them according to their demographics. They wanted to find similarities among the prospects, predict sales, learn the time taken, and convert them to sales — **prioritizing leads to achieve better results**.

From the development perspective, major constraints included:

- Logging solution for **ECS, MongoDB,** and **Lambdas**.
- **Monitoring ECS, MongoDB, lambdas**, and other AWS resources.
- Creating a **deployment strategy** while optimizing cost.

coditas

Why Amazon Web Services?

AWS is the most suitable option due to its global presence that brings along the best infrastructure, simplicity, storage options, backups, scalability, security, and performance. AWS cloud services proved to be easy to use via the Management Console, APIs, and tools. Using AWS tools including **Auto Scaling** and **Elastic Load Balancing** helps applications scale up or down based on demand.

ISV Tools & Technologies Used

- **Jenkins CI/CD** to set up and **automate** the code deployment.
- **Prometheus, Alertmanager,** and **Grafana** for monitoring, alerts, alarms, and metrics.
- **OpenVPN** for secure access controls.
- **ELK** for centralized logging to identify problems with the application.
- **Twilio** for sending and receiving text messages to and from prospects.

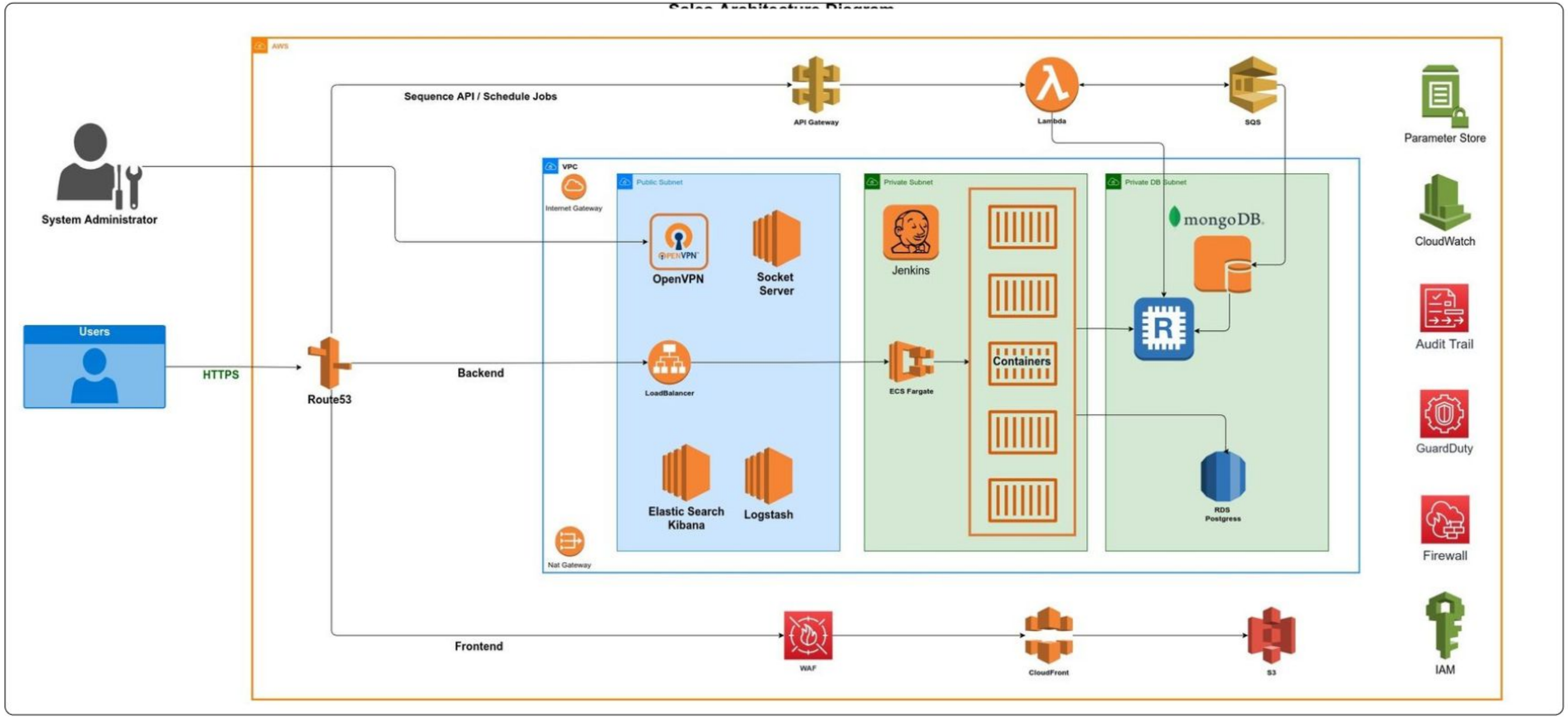
coditas

Proposed Solution

- We **devised a new architecture to improvise** the system. We built structured tables that could be used as input in ML algorithms to replace the current unstructured data. Different ML algorithms were used to cluster prospects based on various parameters.
- For ranking and prioritization of leads, we used a **Multi-armed Bandit**.
- Based on **historical data**, we implemented **distance-based methods** to identify lookalike prospects and predict their turnaround time.
- We utilized **Amazon Simple Storage Service** (**Amazon S3**) for storing static content and hosting frontends; **Amazon Elastic Compute Cloud** (**Amazon EC2**) for computing; **Amazon Elastic Container Service** (**Amazon ECS**) for managing microservices.
- As Fargate removes the operational overhead of scaling, patching, securing, and managing servers, we **created containers using the ECS feature**.
- **Amazon CloudWatch** for setting alerts and overall monitoring of applications.
- **Amazon CloudFront** for global distribution of applications.
- **Amazon SNS** Service for setting up various notifications on email & Slack.
- **Amazon API Gateway** service to handle backend application APIs.
- **Amazon Lambda Service** to handle scheduled jobs and Parameter Store for storing secrets and configurations for the application.
- **Simple Email Service** for sending out emails.
- **Amazon Certificate Manager** for managing SSL certificates.
- **Amazon WAF** for security.

coditas

Solution Architecture



coditas

Outcome and Success Metrics

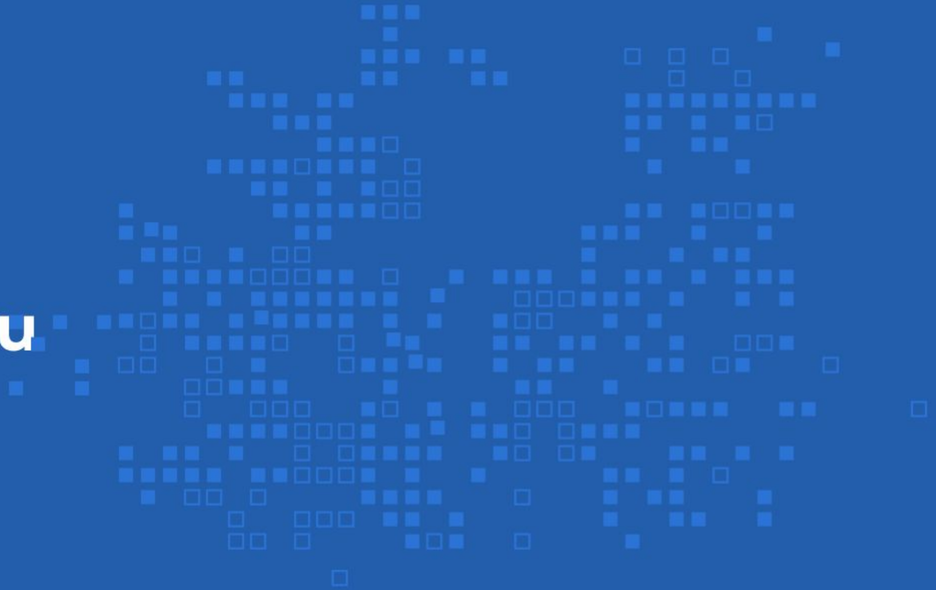
- ✓ **The speed of building servers** has taken a considerable hike.
- ✓ AWS autoscaling service helped **us achieve high availability for the application**.
- ✓ Implementation of all kinds of services is **now easier and faster**.
- ✓ AWS Lambda helped **enhance app performance and is also cost-efficient**.
- ✓ Using CloudWatch helped us detect anomalous behavior in our environments, set alarms, visualize logs and metrics side by side, automate actions, troubleshoot issues, and uncover insights to ensure that **our applications run smoothly**.
- ✓ With Cloudwatch synthetics, we get the **API alerts instantly on different communication mediums** which helps in taking instant action.
- ✓ We could even get the custom metrics to Cloudwatch with the help of elastic agent **to view the metrics on the console and set** alerts accordingly.

As a result, sales representatives can now convert **more leads in less time and effort**. Finding similarities between **prospects and predicting clients' sales is easier** and **less time-consuming**. **Leads would be prioritized** in order to achieve better results.

TCO Analysis

Migration to **ECS** from **EC2** has helped to reduce the overhead cost. We are planning to migrate **MongoDB EC2** to **MongoDB Atlas** because of scalability issues and management difficulties.

coditas



We’d love to hear from you

Let’s start talking at

business@coditas.com

coditas.com