Insurance Tech

Book a Consultation

SERVICES

Insurance Tech, Financial services products

TECH STACK

Gitlab CI/CD, k8s, AWS, EKS, RDS, S3, ECR, Route53, ruby, rails, sidekiq, PostgreSQL, Redis, Rspec, javascript, nodejs, next.js, react, redux

CASE STUDY CATEGORIES

Cloud, Insurance Tech

zing Insurance Infrastructure for Scale, Security, and Growth. Modernizing Insurance Infrastructure for Scale, Security, and Growth. Modernizi

01. Challenge

– *“10 Years On The Market, Time To Refresh.”*

A client came to us with the idea of rebuilding an existing platform on the market for 10 years. Even though his customers were satisfied with the platform it was hard to maintain and expand. We have partnered with the client as a technology provider and advisor.

Given Client’s main product is a CRM system, the Client delivers the product as a set of white-label solutions to its own customers. For customers, it is crucial to be sure that their environment is isolated from others and that their data is protected. For our Client, it is essential to scale its infrastructure in an easily predictable way.

After The Discovery Phase, We Decided To:

> Create a new generation of product UI/UX based on years of experience

> Develop a health insurance platform for companies and individual agents

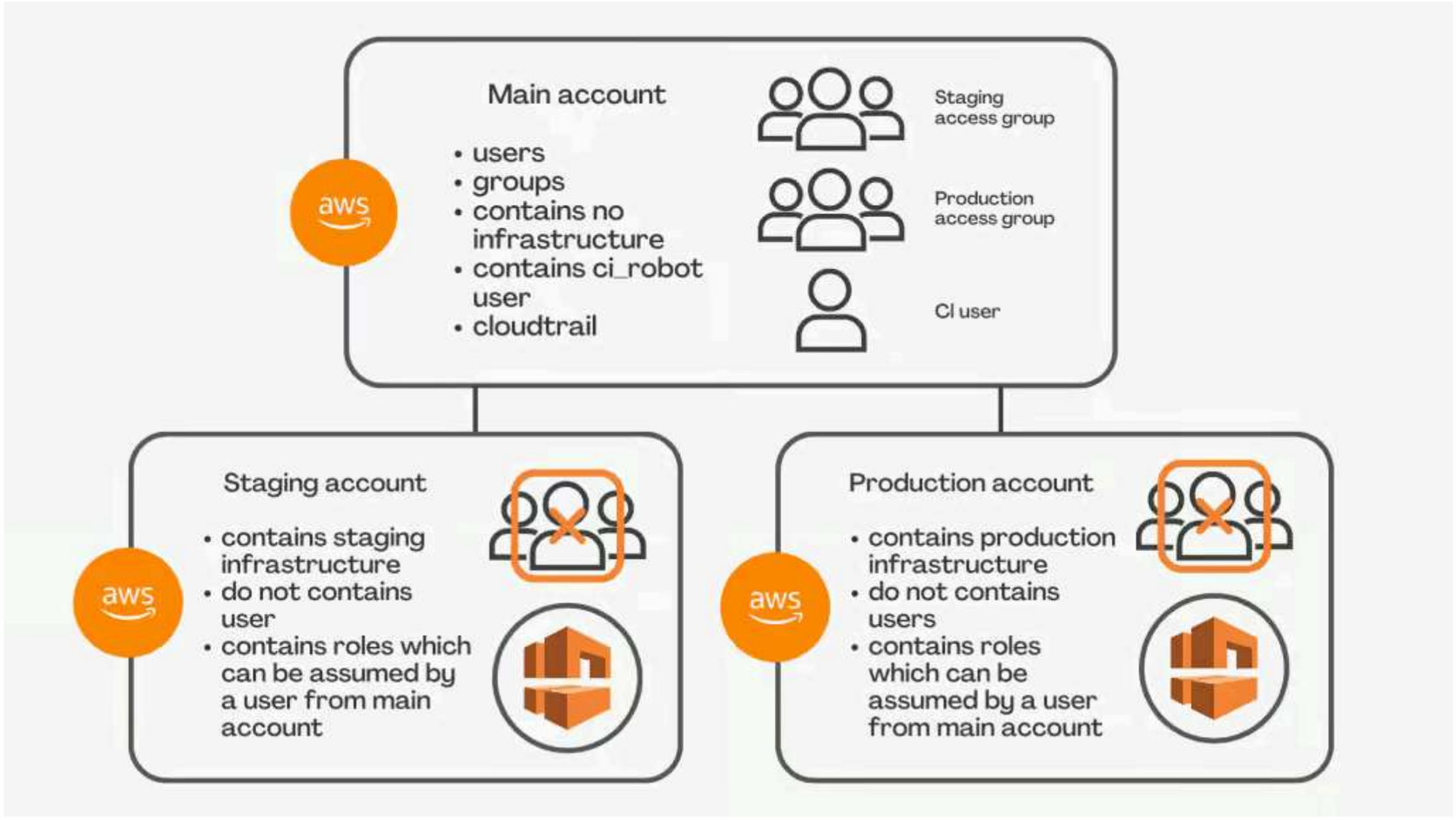
> The key point was to safely migrate data & clients from the legacy system to the new one

02. Solution

To develop a new solution, we built it in parallel with the existing legacy system, which continued to be used by our existing clients. Our success hinged on successfully migrating sensitive data to the new system, which took approximately three quarters to accomplish. Our migration strategy enabled us to successfully transfer 80% of our active clients to the new system quickly. Would you like more details on how we accomplished this?

The Solution Has Three Layers (Aspects):

>Separation of access
>Infrastructure implementation
>Tenant separation



Separation Of Access Is Done With AWS Accounts. As A Result, There Are Two Environments:

> Development. This environment doesn't contain customer data. All the developers have full(root) access to this environment.
> Production. This environment contains customers` data. A limited number of developers have access to this environment. All the operations with this environment should be done through CI/CD.

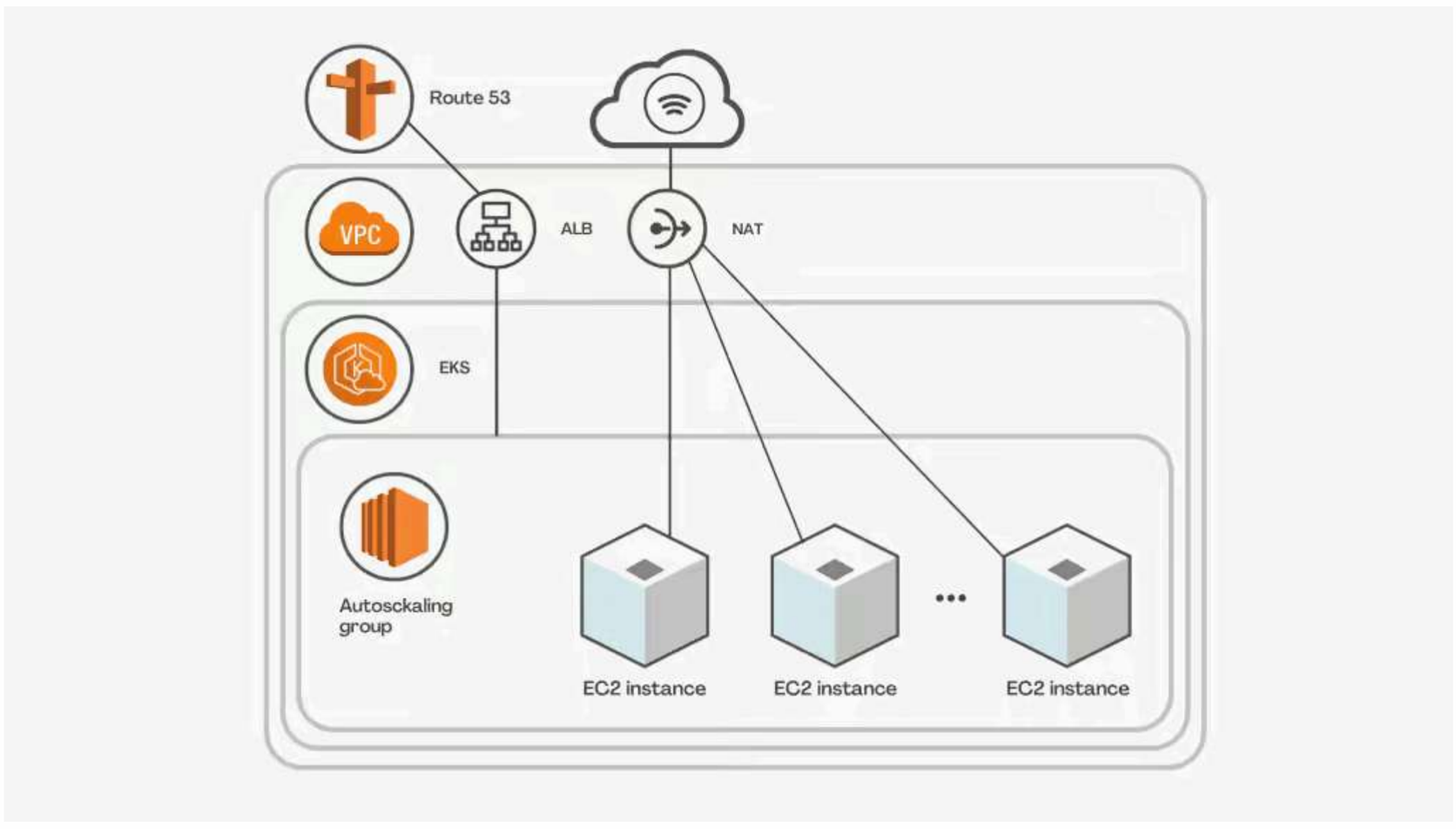
Benefits We Are Getting:

> Environments are fully isolated

> We control a list of users and their permissions in one place

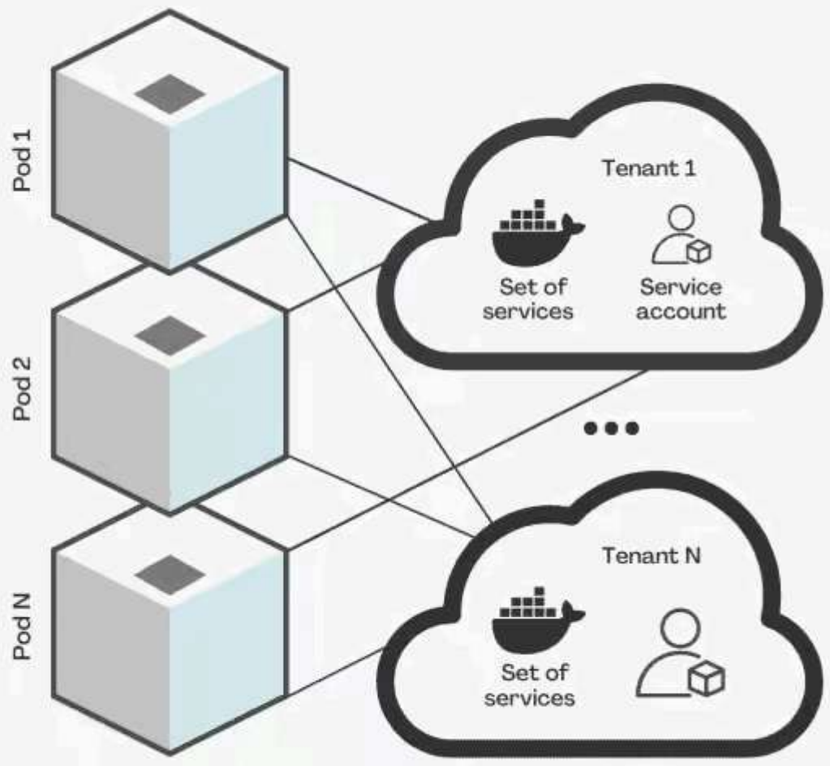
> We are logging access with cloud trail

Developers can't directly access production they are forced to implement Infrastructure as code (IaC) and automate all operations.



Infrastructure Perspective

Each environment is isolated inside its own VPC. Each environment has an EKS and an autoscaling group attached to the environment. The instances do not belong to the public subnets, so we also need to have NAT.



Set Of Services Specific For A Tenant

- Redis
- Amazon S3
- Amazon RDS

03. Impact

Some Numbers:

Prospects Records – 34 216 928
Users – 10 907
Tenants – 12

> Successful launch of the brand-new product

> Help to reach product-market fit and evolve technical part accordingly (move to tenant model with migration into Kubernetes)

> Developed optimised search for prospects that accounts for the lifecycle of the lead

> Implemented a custom platform for voice communication via phone from the system

Created a solution for paper and digital mail delivery that depends on the prospects

04. Want To Know More?

Use the power of robust technologies to drive better business results, with our high-quality team

Book a Call