



CASE STUDY

Application Modernization Journey from GCP to AWS

Access.io successfully migrated PyjamaHR's infrastructure from Google Cloud to the cost-effective AWS Cloud and resolved ML server crashes with ECS containerization



EXECUTIVE SUMMARY

PyjamaHR's SaaS infrastructure was originally hosted on the Google Cloud Platform (GCP). Due to ballooning GCP costs and lack of scalability, PyjamaHR decided to migrate to the more flexible, cost-effective, and scalable AWS Cloud. They also needed to resolve ongoing problems related to Machine Learning (ML) server crashes. Again, they chose AWS for its containerization capabilities and blazing fast in-memory data store. For this two-phase project, PyjamaHR's technology/consulting partner of choice was Access.io. Access.io migrated their existing infrastructure from GCP to AWS, set up a fault-tolerant AWS prod environment and fast in-memory data store, and containerize the ML server using Amazon Elastic Container Service (ECS).

ABOUT ACCESS.IO

Access.io is a cloud services company specializing in cloud transformation, DevOps automation, managed services, and cloud-native application development. We enable organizations in many geographies and industries to transform their IT infrastructure and secure business-critical data with the cloud.

ABOUT PYJAMA HR

PyjamaHR is a SaaS-based HR technology platform that enables companies & recruiters to streamline their recruitment processes, including candidate sourcing and screening, resume parsing, interview scheduling, and offer rollouts. This applicant tracking system (ATS) offers many features and artificial intelligence-powered capabilities that allow HR professionals to save time and effort during the hiring process.

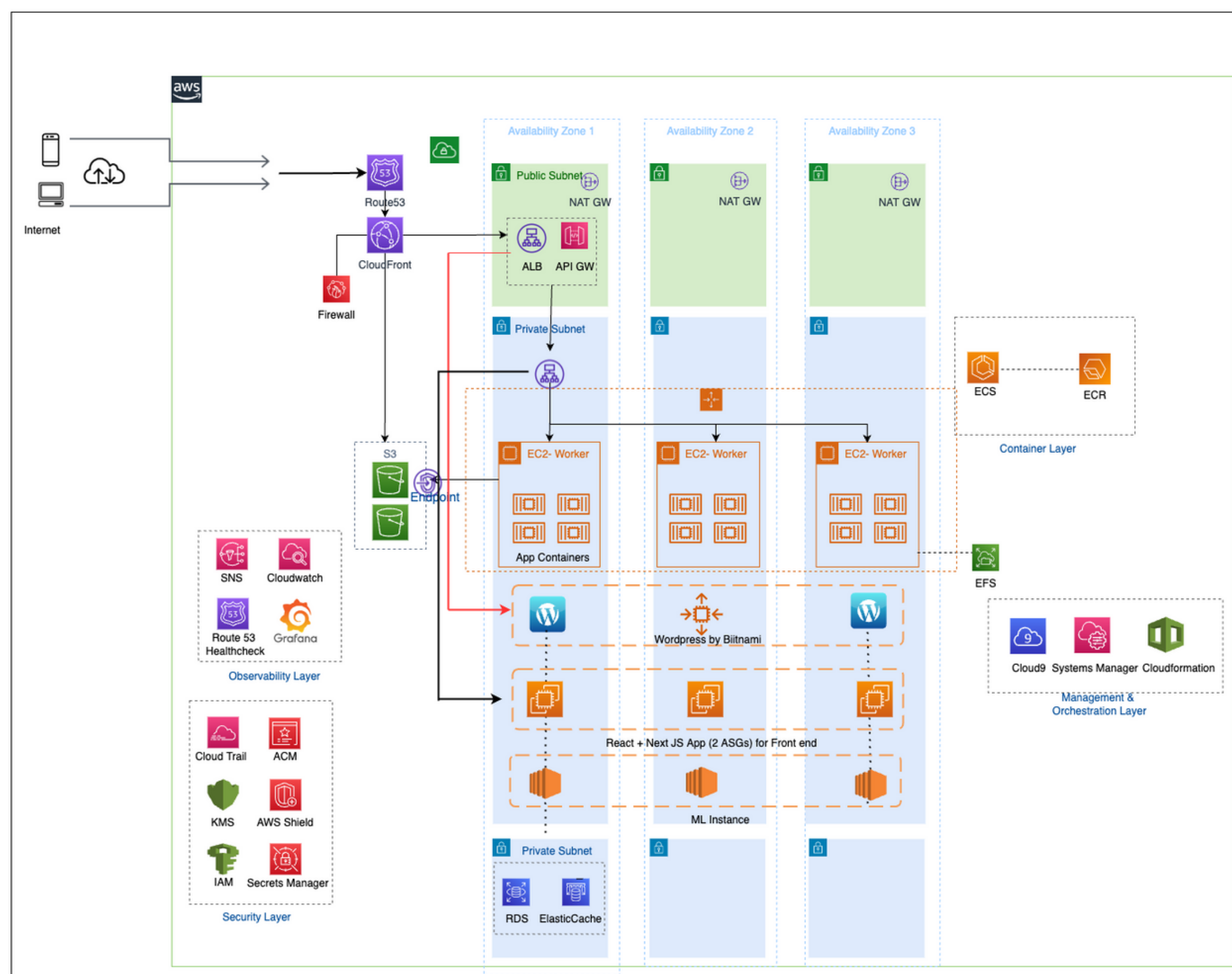
THE CHALLENGE

- Struggling to manage, update, and scale their existing GCP infrastructure.
- Unsustainably high costs for features such as compute time and storage.
- Additional costs for third-party services, such as automation and data analytics tools.
- Their ML server's CPU was spiking continuously due to which the server had crashed multiple times.
- They needed a fault-tolerant prod environment and a scalable infrastructure with predictable and controllable costs.

OUR SOLUTION

In phase 1 of the project, Axxess.io deployed PyjamaHR's servers in multiple AWS AZs to ensure high availability for all workloads and applications. We also placed all compute servers under AWS auto scaling groups (ASG). As collections of EC2 (Amazon Elastic Compute Cloud) instances, ASGs enable PyjamaHR to take advantage of automatic scaling and management based on traffic demand. PyjamaHR can also use scaling policies to increase or decrease the number of instances in their AWS ASG to dynamically meet changing business conditions.

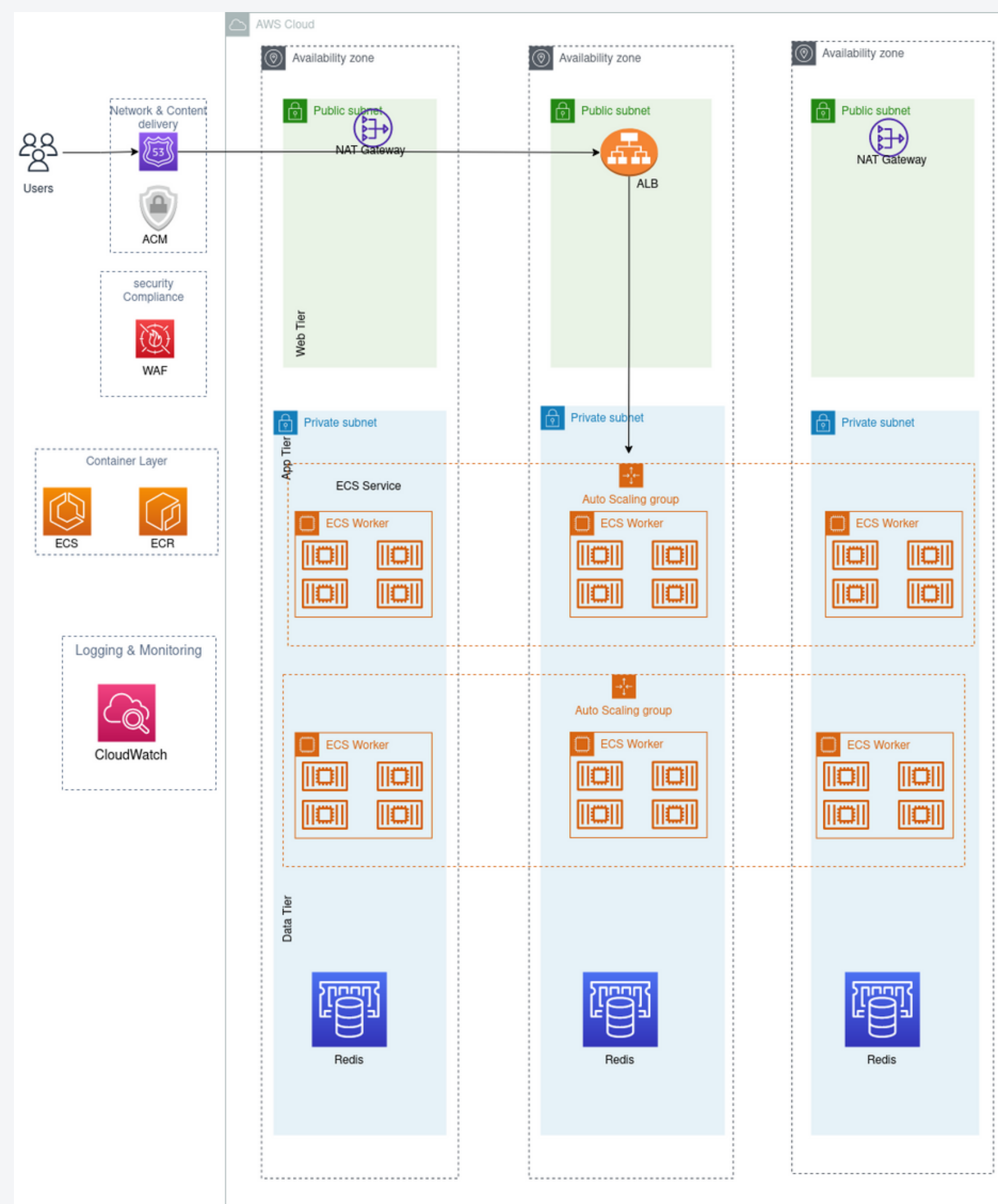
This architecture shows how the PyjamaHR Application will be hosted on AWS.



AWS services used in phase 1:

- Amazon VPC: To launch AWS resources in a logically isolated virtual network
- IAM roles and policies: To securely manage identities, access, and permissions for authorized users to AWS services
- AWS ALB: To automatically distribute incoming application traffic across multiple targets
- CloudTrail: To monitor account activity and API usage across the AWS infrastructure and get control over storage, analysis, and remediation activities
- AWS RDS: To easily set up, operate, and scale databases in the AWS cloud
- Amazon S3 buckets: To store, organize, and protect data for many different use cases in a cost-effective way
- Amazon Route 53: To route end users to web applications with globally-dispersed DNS servers and automatic scaling
- CDN: To deliver website content to end users with low latency and high transfer speeds
- AWS Web Application Firewall (WAF): To protect PyjamaHR's web applications against common web exploits and bot traffic
- Amazon CloudWatch: To simplify infrastructure and application maintenance for the PyjamaHR team

In phase 2 of the project, we containerized PyjamaHR’s ML server using ECS. Eight modules were running on the server and we used ECS to containerize and host all of them. With AWS ECS, PyjamaHR can now run secure, reliable, and scalable containers in AWS using their preferred continuous integration and delivery (CI/CD) and automation tools.



AWS services used in phase 2:

- AWS ECS cluster: To place containers based on PyjamaHR’s resource needs and availability requirements
- ECS Task Definition: To make it easy to launch multiple tasks and enable reliable version control over application specification
- AWS Elastic Container Registry (ECR): To provide PyjamaHR with high-performance hosting to reliably deploy container software anywhere
- Redis Cluster: To partition write traffic across multiple primaries
- ECS Services:

THE BENEFITS



Accelerated
deployment speed by
80%



Cut IT costs
by 35%



Reduced Security
vulnerabilities by 70%



Reduced Server
crashes by 50%