INTEGRITYLOG

Information Processing and Technical Operations

General description

IntegrityLog is a SaaS (Software as a Service) solution that is entirely hosted in the cloud.

Leveraging Amazon Web Services (AWS) - the leading cloud provider - we have built a solution conforming to the highest security standards.

This product is built in a multi-tenant architecture, which means the underlying software has been designed to provide every tenant a dedicated share of the running instances. This design allows our solution to be scalable and flexible in terms of demand in a seamless way to the end user. Despite the shared nature of this architecture, the system is heavily compartmentalised to ensure segregation of customer data at rest and in transit.

The product is structured around internal services that are self-contained in different containers. Leveraging modern container management technology (Kubernetes) it is possible to achieve by design a fail-sail mechanism for core components. Our product is able to balance its load across multiple availability zones (AWS redundancy mechanism) delivering high availability with near zero recovery time in case of failure.

INTEGRITYLOG

Technical overview

Front-end

IntegrityLog is composed of several front-end applications:

- Claim application - used by whistleblowers to submit and follow cases
- Compliance application - used by the customer compliance officers to manage claims, access reports, manage work quests, etc.
- Administration application - used by the IntegrityLog team to monitor and manage customer accounts

The front-end was designed with special focus on accessibility (multi-device, access for persons with disability) and user experience (simple and comprehensive style). It was built with modern web technologies (react framework), that interfaces with our backend service layer according to granular data segregation rules.

By design each front-end application can only access the necessary services of our API.

Back-end

The back-end layer of our product is built around a set of APIs that allow access to several atomic microservices developed in Java Spring boot.

These microservices are built and packaged in individual containers, facilitating an elastic scalability according to demand.

These service containers are orchestrated by Kubernetes, that delivers mechanisms to monitor the health status of cluster, and ensure that the correct capacity to the incoming demand is provisioned.

INTEGRITYLOG

Data layer

Our product handles 2 types of data: structured information and file-based information.

In order to store this information, we rely on AWS native services like Dynamo DB to store all application data, and S3 buckets to store all files. All data is encrypted when in transit (end-to-end encryption) and when at rest (in Dynamo and S3), to ensure full confidentiality.

Infrastructure

IntegrityLog is entirely hosted in AWS, and is spread across several availability zones to comply with high availability non-functional requirements.
Following market best practices, all our infrastructure is provisioned using Terraform Infrastructure as Code (IaC), and subject to version control as any other price of software code.

The usage of IaC approach enables standardisation, reduces the risk of misconfiguration and improves overall security. The infrastructure was designed to react to peak demand and automatically provision new processing power via horizontal scaling without any impact on product operation.

Product operation

Monitoring

Our product is monitored 24x7 by the Euronext specialised operations centre, which closely tracks multiple near real-time metrics.

These metrics cover all relevant aspects of infrastructure such as capacity provisioning, health status of database and computing clusters, but also application aspects as services health status and front-end error handling. The centralisation of all application logs allows for any forensic investigation and ensures the auditability of what is happening in the product in any given timeframe.

Data backup

All data that is stored within our product has well defined data policies to ensure compliance with data retention regulation.
Data backups are performed incrementally via internal AWS mechanisms, and ensure a near zero Recovery Time Objective (last committed information).

Software upgrades

The usage of containers to host our product, ensures granular control over different services.
We are able to perform a phased rollout of a new product version by incrementally upgrading the different services, or in a big bang approach if that is what is required.

Any software upgrade is performed under the defined maintenance windows, and will have no impact on service availability.

INTEGRITYLOG

Information Security

Identity and access management

The product has been designed to leverage several AWS services natively, and the ones used to handle authentication and authorisation are Cognito and Key Management Service (KMS).

Amazon Cognito facilitates the authentication and authorisation of application users and its tightly integrated with the architecture of the product. Cognito is configured to use 2factor authentication (SMS token) which increases security when accessing the application. Amazon KMS provides the capability to manage cryptographic keys, and its where each Customer Master Key is stored. Access to KMS is managed within the security context of each tenant, meaning that it's impossible for any type of user to access keys that access was not specifically granted to.

Development process

Our products are secure by design, and this starts in the development phase. It is present from inception in the architectural phase, and throughout development and testing activities and extends to day to day production monitoring. As part of our development process, every code change that goes into the build pipeline is automatically validated by static code scan (SCA) tools and validated against code quality metrics.

The introduction of a possible security flaw is validated against known CWEs (community-developed list of software weakness types) and identified early in the product development phase. Every year an in-depth manual penetration test is performed using a grey box strategy, to ensure no new exploit vectors have been created.

Operational monitoring

From the moment our product is brought into a production environment, our Security Operation Center (SOC) actively monitors security perimeters of the product. This involves an outside-in approach by continuously validating public endpoints, and via Security Event Monitoring of the product.

Mechanisms are in place to monitor and respond to any DDOS attack vector, and strong controls are implemented through AWS Shield, to ensure minimal impact on any aspects of the Product.

CONTACT US