Case study:

LCloud Ltd. for Soulmates Guardian News & Media Limited

CUSTOMER:
Guardian News & Media Limited

FIELD:
Media, social networks

SERVICE / APPLICATION:
soulmates.theguardian.com

TAGS:
Media, social networks, mobile applications

DESCRIPTION:
Social portal – web version and mobile application

TIME FRAME OF THE PROJECT:
September - December 2019

AWS SERVICES USED FOR IMPLEMENTATION:

- AMAZON VPC
- AMAZON ELB
- AMAZON CLOUDFRONT

SOLUTIONS USED FOR THE IMPLEMENTATION:

- SECURITY AND COMPATIBILITY
- APPLICATION SACLING
- AUTOMATION

Client

Soulmates is the UK's most popular online social networking portal for over 15 years. It is visited by over 500,000 users each month.

The owner of the site is one of the longest traditions - media concern The Guardian.

Soulmates is available both through a browser and in a mobile application.

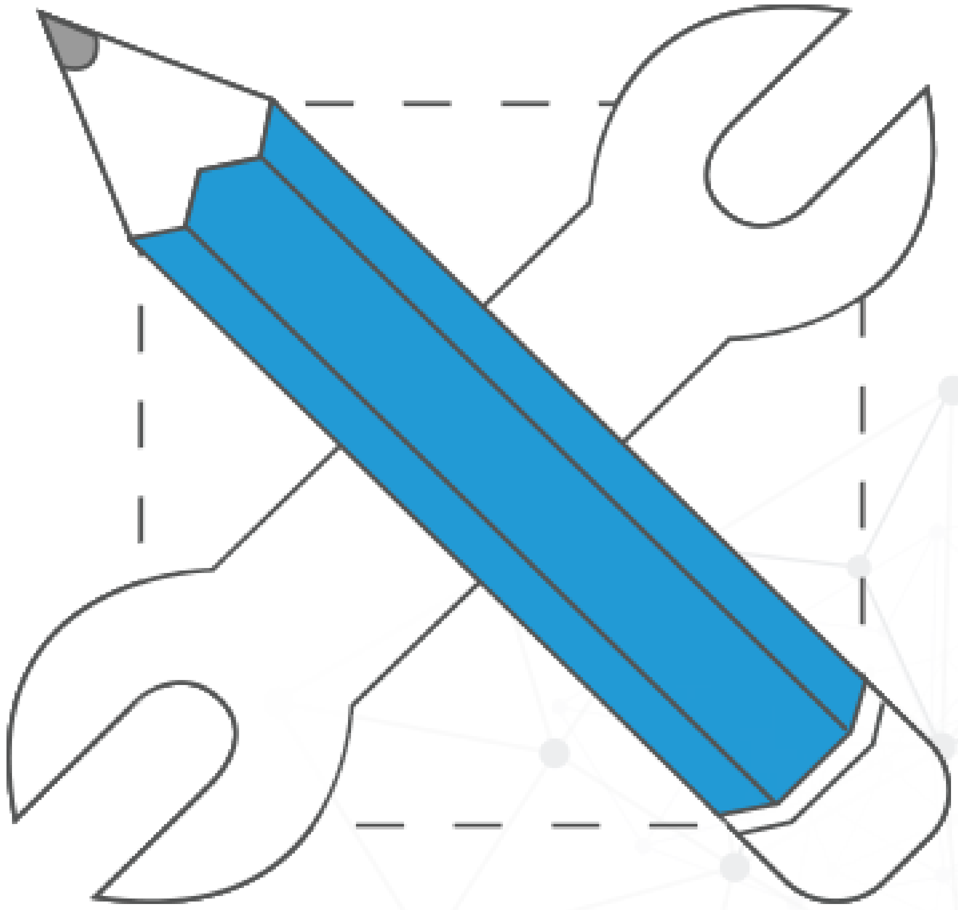
2 Challenges

- The AWS service **architecture** with which the client came to us **was insufficient to ensure the required, very high level of service provision security, in the web portal and mobile application.** The environment required the completion of important AWS functionalities and services.
- The **problem** was **the operation of the infrastructure outside VPC.** Moreover, common groups were used in security, too many roles and IAM user privileges.
- The customer noted **frequent problems with deployment** - the process was not always successful, and it was difficult to diagnose the cause. Additionally, large client's team was involved in the process.

LCLOUD

Challenges - continuation

- **The application placed on the existing infrastructure did not work steadily.** There were frequent, unforeseen interruptions in service availability for users. There were several reasons for the instability. One of them was the fact that Apache Solr was placed on a single instance, without the possibility of scaling.
- Another problem was the fact that **the application used the MongoDB database was located on a single instance.**



3 Proposed Solution & Architecture

Since the client did not have the technical documentation, **we started the project with an audit** in order to be able to know exactly the nature of the application and the resources used for its functionality. Then **we designed a private cloud computing solution (using Amazon VPC) based on Amazon Web Services cloud, with particular emphasis on high security aspects.**

We also **planned the procedures of self-scaling resources**, which are applied when traffic in the portal dynamically changes.

Then we enriched the infrastructure with compatible complementary services, thanks to which both further development and maintenance of the environment and introduction of subsequent versions of the application have become simplified and more comfortable.

LCLOUD

3 Proposed Solution & Architecture - continuation

The **architecture we designed represents the Infrastructure as Code (IaC) approach**, using AWS CloudFormation for infrastructure management.

As an **important goal in the project was to substantially increase the level of security.** We implemented and provided highly secure and auditable access to application instances for programmers and support engineers by **implementing the Certificate Authority**, based on the Netflix BLESS project.

Using SSH keys for remote connection with servers is one of the best practices. However, managing them and managing users requires additional work on access configuration, on each instance. In order to simplify the exchange of keys, we decided to implement the Certificate Authority system based on the Netflix BLESS project. BLESS is an SSH Certification Office based on the AWS Lambda function.

Proposed Solution & Architecture - continuation

SSH certificates allow the certification authority to sign the user's public key together with a list of restrictions, and then the user submits this certificate to the server during authentication.

The server only needs to trust the CA, and does not need previous knowledge of the user's public key.

The **use of BLESS also allows limiting access to the CA by granting limited privileges to IAM users and roles.**

Only authenticated and authorized users can send requests to BLESS and receive a signed certificate.

The full configuration that has been implemented is based on AWS services running on the main AWS production account:

- *AWS Identity and Access Management,*
- *AWS Key Management Service,*
- *AWS Lambda,*
- *Amazon CloudWatch and Amazon CloudWatch Events,*
- *Amazon Simple Storage Service,*
- *Amazon EC2 Systems Manager.*

Proposed Solution & Architecture - continuation

The **configuration has been implemented in two different AWS regions** to **ensure high availability** for the Certification Authority.

If dependent services, such as AWS Lambda or AWS Key Management Service, have problems with proper operation or even the entire region is not available, we can still use the Certificate Authority from another region.

Two separate BLESS Lambda functions are configured in each region for use in separate test and production environments.

The functions for each environment are using separate Certificate Authority keys.

BLESS was configured on the main production account, which at that time was treated as a login account, where all IAM users are created and from which users can switch to the connected "DEV" account using cross-account IAM roles.

Proposed Solution & Architecture - continuation

IAM users inherit, according to the IAM Group, **limited privileges**, which require prior authorization with an MFA token before they can apply to BLESS to sign their key.

Users are also limited in their ability to choose the username for whom the certificate will be generated. This is done by means of an additional Python library - kmsauth, and the requirement to use a temporary AWS Key Management Service key encrypted token. This allows us to confirm the identity of the user and make a statement on how to authenticate it.

To simplify the process for the end user, we have prepared a Bash script, which automates connections with public and private instances operating in VPC.

Proposed Solution & Architecture - continuation

The next step in implementation was to **automate SSH server and user configuration on instances.**

Most of them work in automatic scaling groups and are often replaced. It was also important to be able to manage users manually, without having to prepare new AMI or run new instances.

For these reasons we **decided to use AWS Systems Manager Run Command to run Ansible playbooks on managed instances, in combination with Amazon CloudWatch Events and AWS Lambda services.**

The project was carried out by a team consisting of dedicated **Project Manager, Solution Architects, DevOps, DevSecOps, FinOps and SysOps.**

What AWS services were used as part of the solution

Amazon VPC

AWS Lambda

Amazon EC2

Amazon Simple Storage Service

AWS KMS

AWS IAM

AWS CodeDeploy

Amazon CloudWatch

AWS Systems Manager

Third party applications and solutions used in the project

GitHub

PagerDuty

REDMINE flexible project management

slack

The project used several solutions and applications that enriched and improved it. Some of them were determined by the client's previous experience.

GitHub was used as a code repository. Redmine was used to manage the milestones. Current communication in the project was carried out using Slack.

LCLOUD

Diagram of a selected part of AWS infrastructure and services used in the project soulmates.theguardian.com

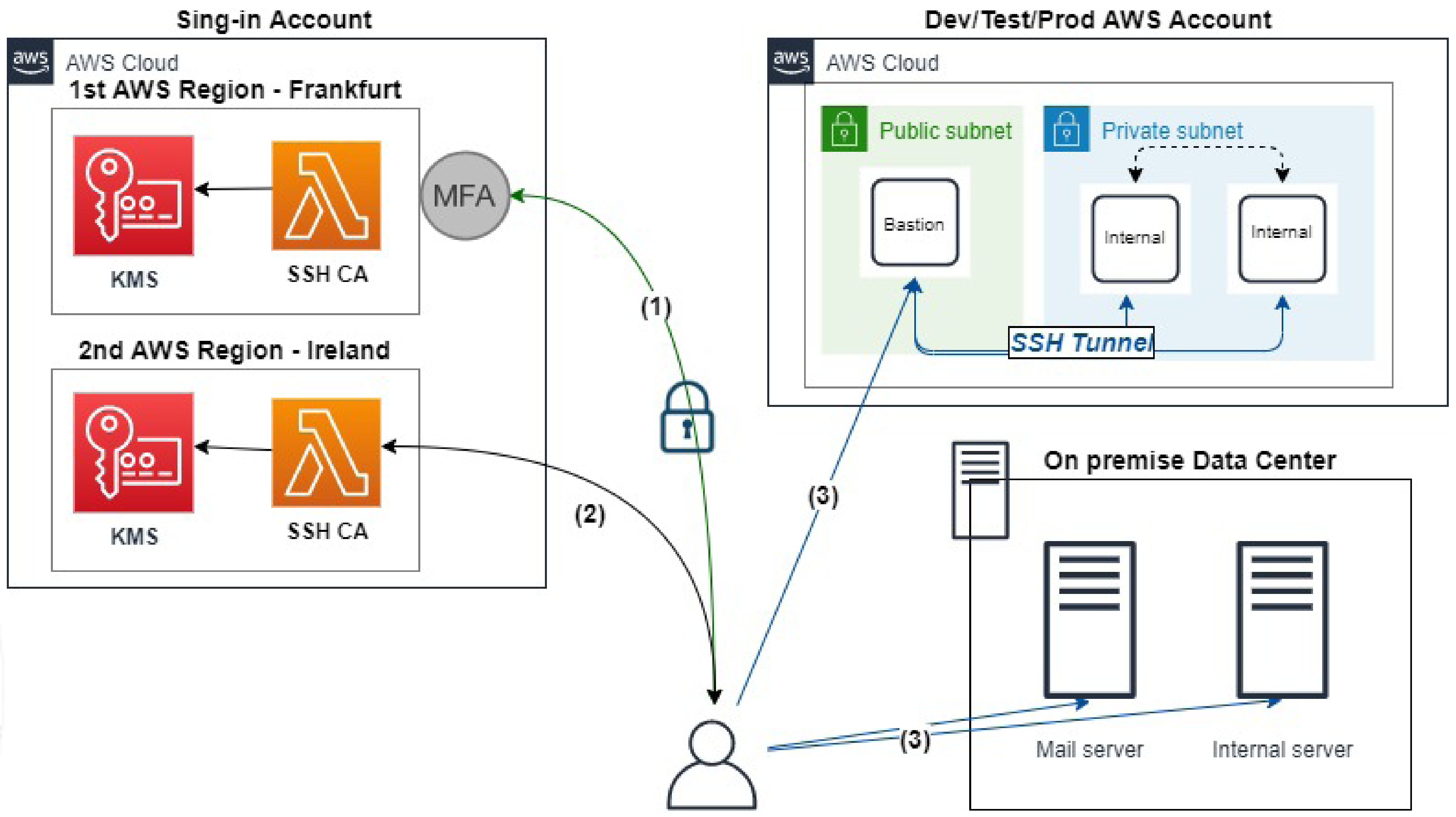


Diagram of the AWS infrastructure and services structure used in the soulmates.theguardian.com project

7

Results from the solution

We have **increased infrastructure security by implementing secure VPN connections and Ossec solution**. In addition, we **migrated the environment to a multi-level VPC**, where we separated the public and private parts.

We **swapped CLI users for IAM users**, with the appropriate permission limits.

We **implemented a fully automatic authentication process for users connecting to the infrastructure**.

In order to facilitate configuration management and periodic updates, each of the related parts of the configuration of this automatic process has been implemented as a Serverless service.

The configuration files, including Certification Authorities' public keys and the list of system users, are dynamically downloaded from the central configuration bucket.

LCLOUD

7 Results from the solution - continuation

We **solved the problem of frequent unavailability of the portal and mobile application**, which was caused by overload of servers, not prepared for autoscaling.

We **replaced the types of instances that did not meet the requirements of individual application layers** (frontend, backend, cache, database).

To fix the problem with an unstable application, we created a cluster for Solr and launched its autoscaling. To the same end, we designed and launched a MongoDB cluster.

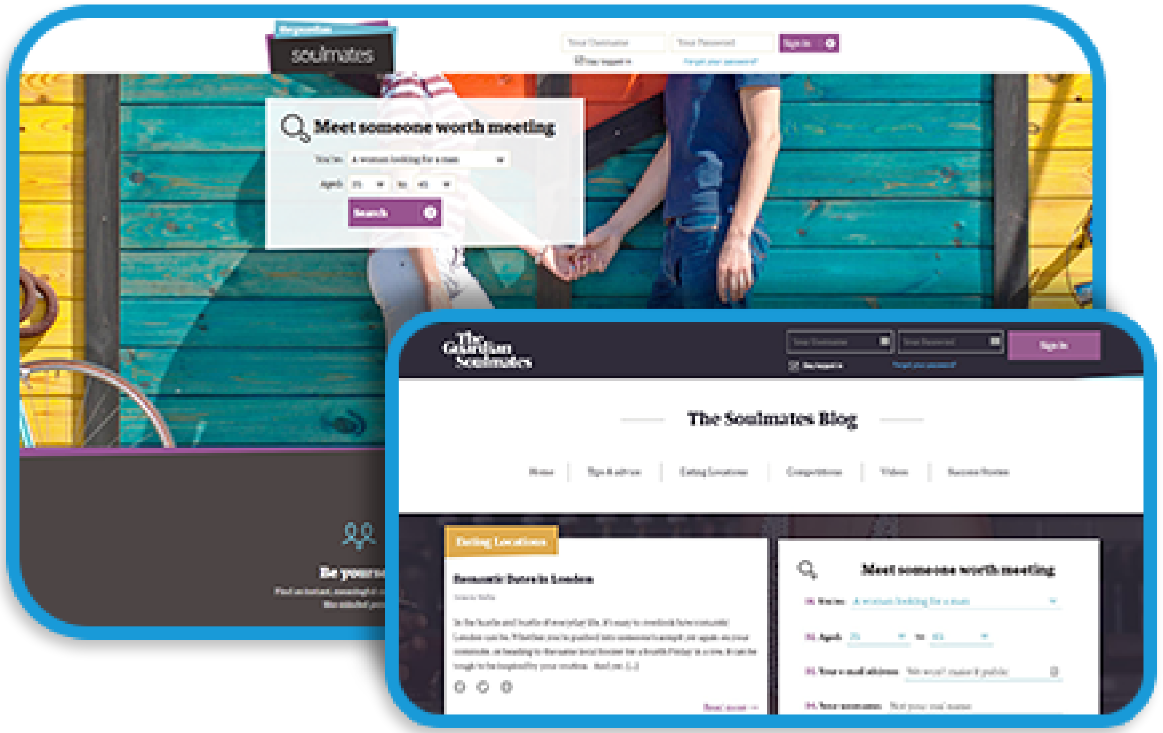
We **automated the deployment process using AWS CodeDeploy and Teamcity** and transferred the process to the environment in one VPC.

8 Metrics for Success

Thanks to the implementation of several security solutions, **the architecture meets the highest corporate standards of the client**.

The implementation of the BLESS-based solution not only streamlined the access management process, thanks to advanced automation, but also really reduced the costs of security management - reducing the time consumption of **the authorization process by 3% per year**.

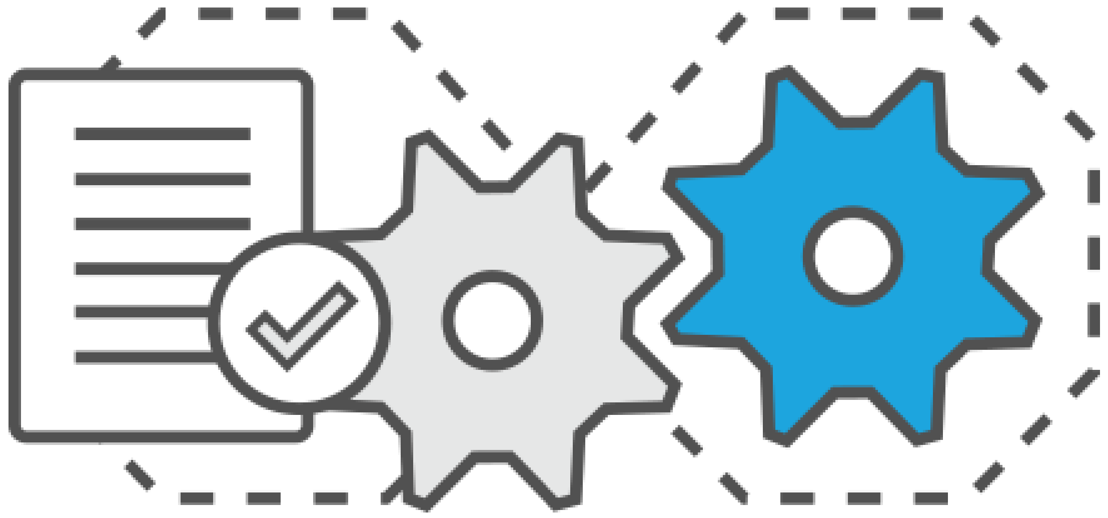
Automation of the deployment process gave the possibility to move programmers to projects related to the development of the portal functionality, so that **the service is dynamically developed**, and **new versions are quickly made available to users.**



8 Metrics for Success

The implemented **solutions made the infrastructure highly accessible, auto scalable and flexible** - **without any downtime**. Such as should be the infrastructure adapted to the needs of portals with high traffic trends. **The well-thought-out project has resulted in a significant reduction of environmental costs by 20% per year.**

The implemented **complementary services increased the level of monitoring and automation**, thus simplifying the environmental management process and increasing control over its use.



Lessons learned

The project has provided clear evidence of how quickly and dynamically cloud services are developing. Once designed, the public cloud architecture should be reviewed and analyzed on an ongoing basis to include new services that automate and streamline the process so you can focus on business growth.

A high level of security is a goal that all companies strive for. It is important that this goal is achieved in a balanced way, so that the security levels set in sequence do not constitute a huge organizational burden. Therefore, it is very important to automate security processes and thus reduce time consuming tasks.

Lessons learned

It is important to have a conscious approach to the analysis of IT processes taking place in the company. Detailed monitoring of these processes, also using cloud services, provides information about bottle necks. It often happens that these limitations are human errors. Therefore, it is worthwhile to automate processes where possible and justified.

By reducing their time consuming, we gain the ability to focus IT teams on what is the real added value, i.e. the development of a portal or application rather than supporting processes such as deployment. By automating these processes we also reduce the risk of human error.

Contact us

LCloud Sp. z o.o.
ul. Złota 59
00-120 Warszawa

+48 22 355 23 55

kontakt@lcloud.pl

Quick contact:

biuro@lcloud.pl

Business:

sale@lcloud.pl
+48 22 355 23 57