



# INCISION GROUP B.V. NETHERLANDS CASE STUDY

**Name of the customer:** INCISION GROUP B.V.

**Start date of project:** July 15th, 2019

**End date of project:** Active development is still in progress

## About the Customer

Incision is an education and workflow platform for surgeons worldwide. Incision produces content in-house with a team of experienced surgeons, 3d modelers and video editors. The main purpose of the platform is to give surgeons access to shared experiences, skills and expertise by providing tools to get a quicker and more thorough preparation for the OR. The HQ office is located in Amsterdam, the Netherlands. However, The Incision sales team is located world-wide. The platform currently counts over 100.000 registered users.

## Customer Challenge

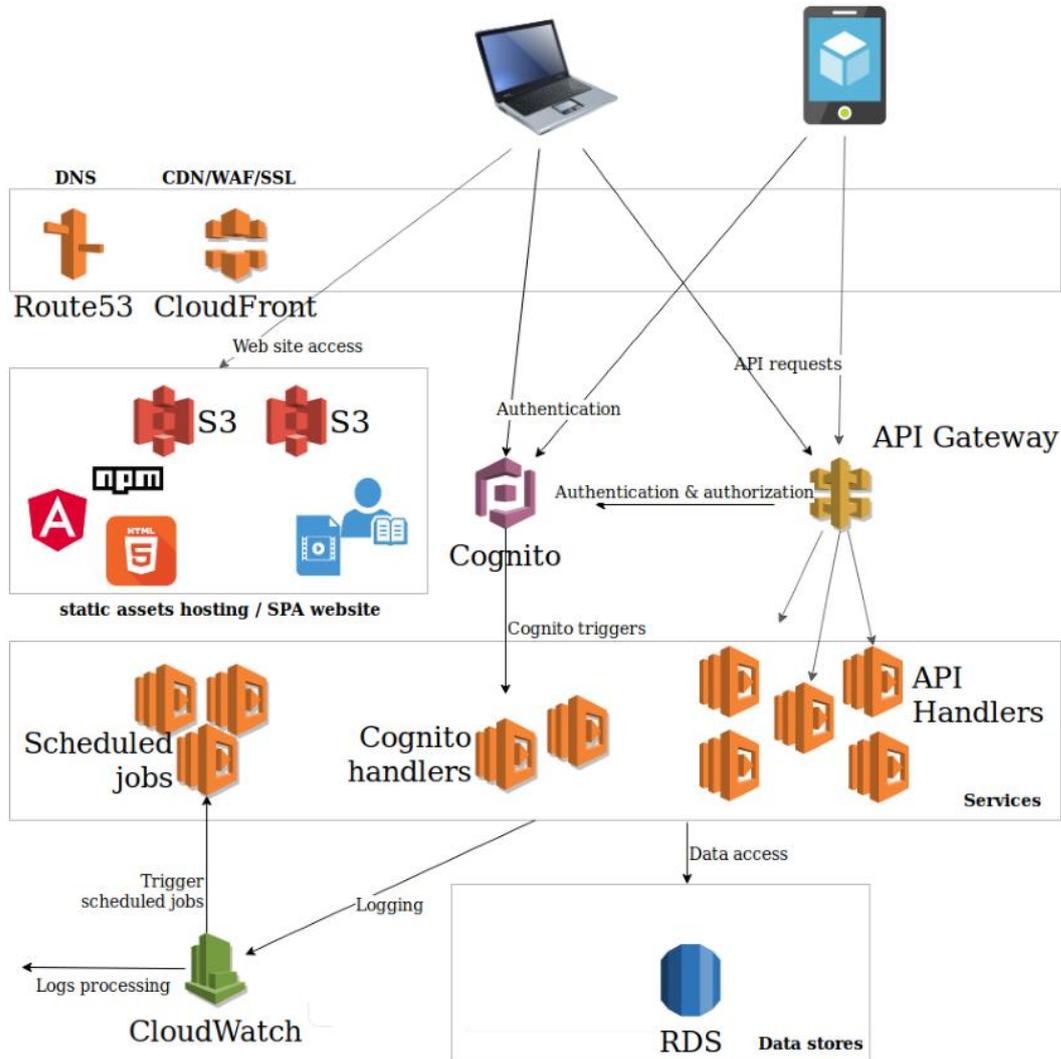
The existing solution (<https://www.incision.care/>) was built using WordPress (WP). To support various features, Incision developed a set of custom WP plugins. Maintaining WP plugins is a challenge. Due to various custom plugins WordPress could not be updated to the latest versions, making the platform potentially vulnerable to security threats. Additionally, the data structure that WordPress imposes started slowing down the application, creating a negative user experience. Some of Incision's customers indicated that they might stop using the platform due to poor performance.

## Proposed Solution and Architecture

Together with the CTO of Incision, the direction was chosen to rebuild the platform from the beginning in order to overcome existing issues and support growth in the future. A move of the platform to AWS and the use of AWS managed services and serverless architecture would enable our team to quickly create the new platform and prevent existing users leaving the platform by offering a highly scalable system.

## How AWS services were used as part of the solution

All static content and a single page website file are hosted on S3, sitting behind CloudFront and Route53. The backend consists of a REST API (API Gateway) and a RDS MySQL database. Cognito is used for user authentication and authorization. Welcome emails, as well as transactional emails, are sent to users using Simple Email Service (SES). The REST API consists of several CloudFormation stacks with Node.js Lambda functions deployed using the Serverless framework. The application uses Lambda functions for Cognito triggers and a number of scheduled tasks. Deployments are automated and executed via CodeBuild. The infrastructure is described in CloudFormation templates as a part of the code base. This architecture is illustrated in the image below:



### Third party applications or solutions used

- LTI integration with Learning Management Systems (LMS) for users to take tests and have their grades from Incision submitted to their organization's LMS
- Stripe integration for individual user payments for monthly/yearly subscriptions
- SAML integration with 3rd party identity provider via Keycloak open-source solution

### Results and Benefits

#### Faster Software Delivery

The previous WordPress website was released with a low frequency. Having moved to AWS, the DevOps team is able to release features granularly – stories are published to production when they are ready, so the deployment frequency is now multiple times per sprint. The team is using AWS CodeBuild, which is integrated with BitBucket. Whenever a code merge is detected, the build process is triggered with CodeBuild. This software delivery pipeline contributes to the speed of software delivery. Additionally, using a serverless approach and by using AWS's managed services means that the team could use proven, existing services for the project, further increasing the speed to set up the software delivery.

#### Shift left

In order to detect errors as soon as possible in the process, unit tests are run on each build. Additionally, every night, a set of end-to-end tests is run against the latest version on the development environment, thus providing a reliable regression test set. End-to-end tests are defined using Selenium and Cucumber. The team is using Serenity to obtain reports on any eventual errors or problems.

### **Automation**

In order to achieve a repeatable, error-free infrastructure setup, the team uses Infrastructure as a Code (IaC) for most resources on AWS. The network setup (VPC), the RDS database and the S3 buckets for the website hosting are all defined in CloudFormation templates. The Serverless framework is used to deploy the API, where the API Gateway and Lambda services offer automatic scaling.

### **Monitoring**

In order to keep track of the health of the platform, the team has set up monitoring dashboards and alarms using AWS CloudWatch. The team uses API monitoring to track any API errors, and team members receive an email when an API error is detected. For the serverless part of the system, a Lambda error count dashboard is configured using CloudWatch.

### **Lessons learnt**

The new platform went live on February 4th, 2020 after approx. 6 months of development. One of the many clearly visible benefits is that using managed services really speeds up feature development and delivery, while improving maintainability and cost control.

AWS provides the tools that are needed for a DevOps approach, such as Infrastructure as a code, a Continuous Delivery Pipeline, and system health monitoring.

We conducted a Well Architected Review on the AWS setup and one of the outcomes was that at that time, monitoring was a weak spot and security needed tightening up. The team then configured AWS CloudWatch dashboards and alarms and improved security both on the user level and on the network level as a remediation measure.