

# CASE STUDY

## **AWS SERVERLESS APPLICATION MODEL IMPROVES COST EFFICIENCY FOR PRECISION EXAM'S INFRASTRUCTURE**

### **Client Background**

Our client provides a web platform for taking exams. The platform empowers teachers nationwide to enroll students, authorize, conduct exams online, and review results.

### **Business Challenge**

The existing client product is currently using older technologies and architectures that can be replaced by modern services provided by AWS. Although the system modules are decoupled and provide a robust solution, there were improvements that could be made to the architecture and CI/CD environment to enhance availability and scalability approaches based on AWS services.

Due to increases in customer usage, driving new business requirements and in order to improve current infrastructure in the cost-efficient way, it was decided to start development of serverless application

within the Proof of Concept stage for further migration to the production-ready solution. That's why the client requested SoftServe to provide a prototype of a system, built on top of AWS Cloud, using the best-practices approach in architecture and managed AWS Services.

### **Challenges for the new solution:**

- Build a highly-available, scalable and robust system
- Use AWS managed and serverless services, in order to cover all application requirement
- Follow AWS best practices for infrastructure design
- Serverless for local development and testing
- Continuous Integration/Delivery implementation
- Infrastructure as a Code implementation
- Application Monitoring and Log aggregation

## Project Description

Based on the best-practices were chosen AWS Lambda, API Gateway, and DynamoDB as serverless solutions that allow running code, describe RESTful API and store persistent data without any server provisioning and with minimal managing. Amazon Cognito provides user management, sign-up, sign-in, and access control to web apps quickly and easily.

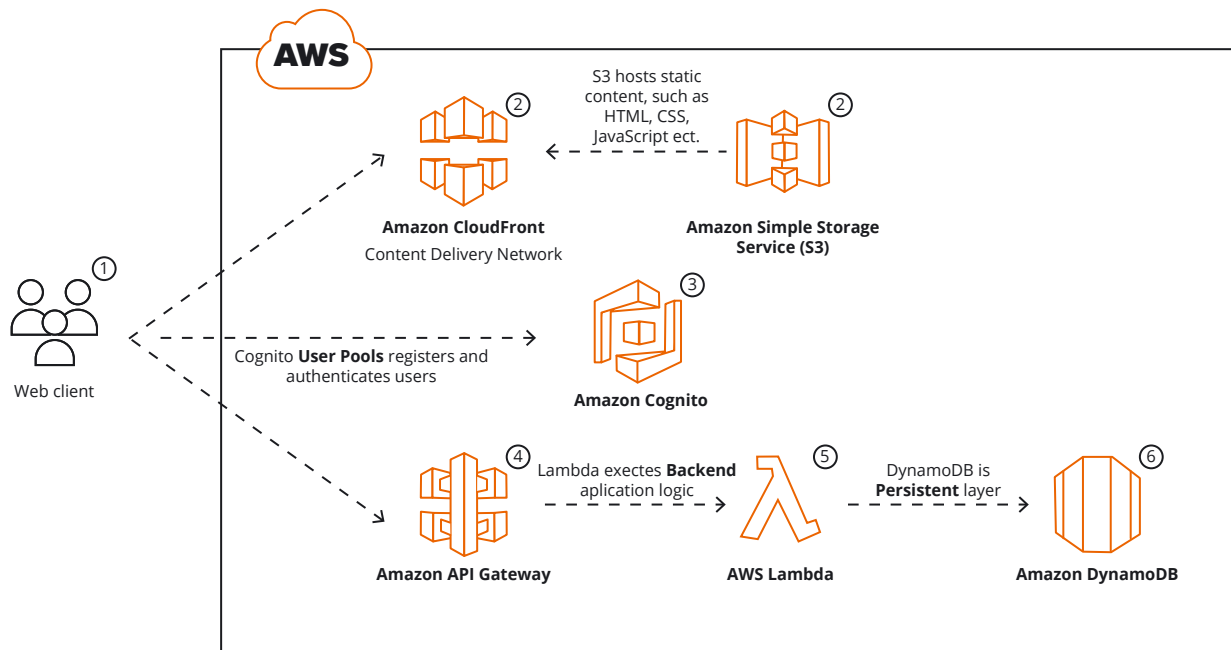


Figure 1: High level architecture diagram

Adhere to the principle of microservices, it was decided to create small lambda functions and avoid huge ones. Lambda functions essentially were designed to have only one class/function inside without various dependencies and repetitive functions. We used AWS Lambda Layers functionality to carry out the repetitive code and shared libraries into the separate essence (layer). In spite of the chosen approach to store a set of lambda code in a git mono-repository, lambdas were designed in the way of independent deployment. Basically, only changed lambdas will be re-deployed without deploying any other unchanged function within the same repository.

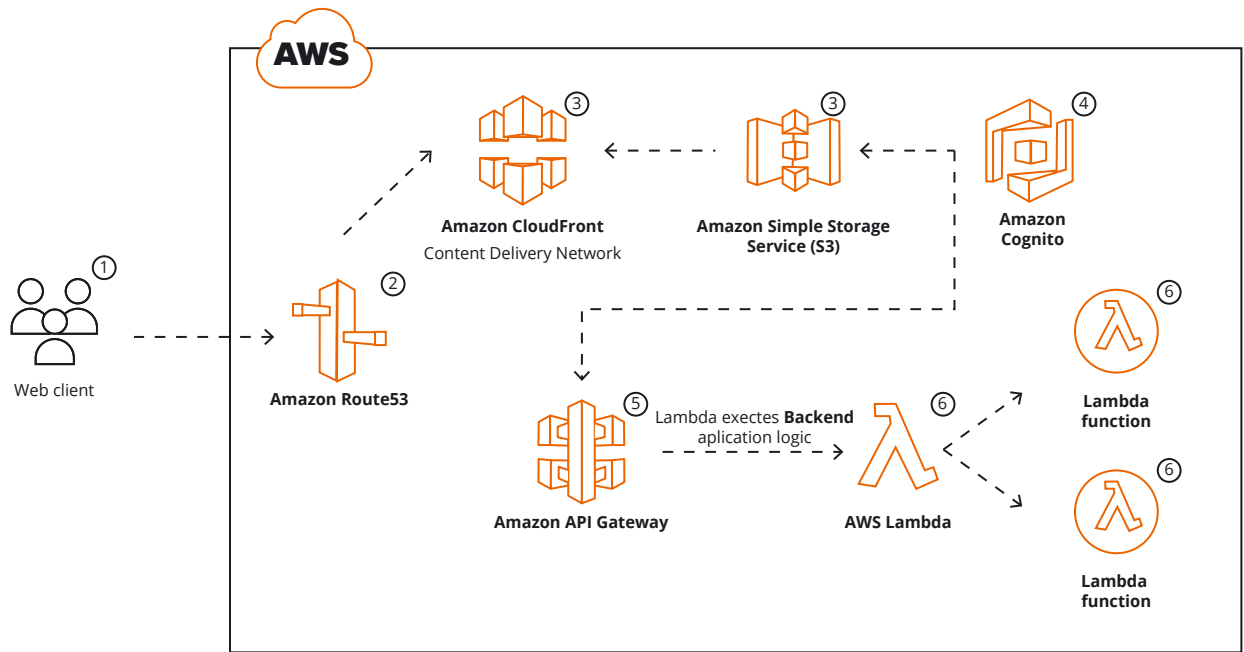


Figure 2: Serverless Backend Service and RESTful API

On Figure 3 (bellow) shown CI/CD flow of serverless backend with using of AWS CodeCommit, CodeBuild and CodePipeline services. Since client requirement was to use BitBucket service as a git repository, we integrated it with AWS CodeCommit for having full-featured CI/CD Pipeline. We have git mono-repository, each lambda function has it is own directory for the codebase. Once changes on CodeCommit trigger AWS CodeBuild, it assembles the application, sends code to S3 bucket and moves CloudFormation template file up the pipeline chain to deploy the changed resources. In order to deploy only that lambda function which code was changed, we set up Maven reproducible builds plugin.

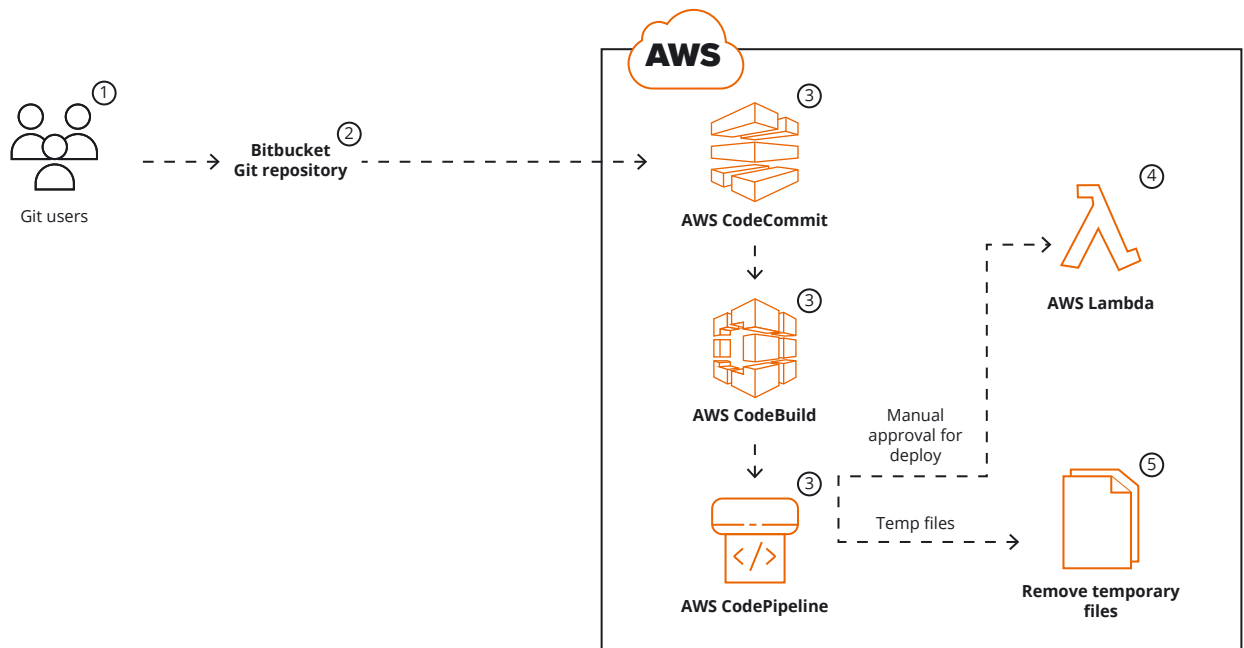


Figure 3: CI/CD for backend Lambda functions

Accordingly, to the decision of using serverless services as much as possible, we migrated static web content from reserved EC2 instances to AWS S3 buckets with enabled Static Web Hosting option, CloudFront, Route53. All of the static web content, including HTML, CSS, JavaScript, images and other files are securely served via Amazon CloudFront. End users access the website using the public website URL exposed by Amazon Route 53, which points to Amazon CloudFront distribution. Amazon CloudFront distribution works with static content on Amazon S3.

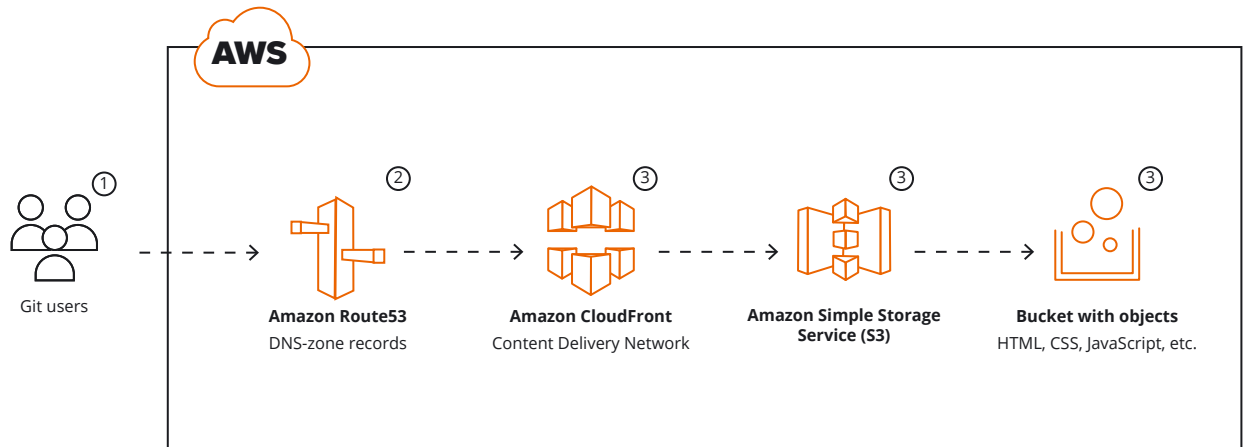


Figure 4: Static web hosting diagram

AWS Serverless Application Model (AWS SAM) is a framework that we use for emulating lambdas and API resources locally. It provides the ability to test the code internally and does not upload the code into AWS each time in order to test the changes. Mostly, we use SAM CLI for API Gateway emulation and local function invocation. In its essence, SAM is a CloudFormation library, therefore we have serverless resources (Lambdas, API gateways) described as a code. Moreover, SAM allows us to build serverless applications faster and smoother.

```

PROBLEMS (8)  OUTPUT  DEBUG CONSOLE  TERMINAL  1: Python

$ sam local start-api
2019-01-09 13:56:42 Found credentials in shared credential file: ~/.aws/credentials
2019-01-09 13:56:42 Mounting TempConversionFunction at http://127.0.0.1:3000/{conversion}/{value} [GET]
2019-01-09 13:56:42 You can now browse to the above endpoints to invoke your functions. You do not need to restart/reload SAM CLI while working
changes will be reflected instantly/automatically. You only need to restart SAM CLI if you update your AWS SAM template
2019-01-09 13:56:42 * Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)

```

Figure 5: AWS API Gateway emulation example

```

2019-01-04 14:20:50 Reading invoke payload from stdin (you can also pass it from file with --event)
2019-01-04 14:20:50 Found credentials in shared credentials file: ~/.aws/credentials
2019-01-04 14:20:50 Inworking app. lambdaHandler (nodejs8.10)

Fetcing lambci/lambda:nodejs8.10 Docker container image . . . . .
2019-01-04 14:20:52 Mounting /Users/ /Desktop/sam-test/sam-app/hello-world as /var/task:ro inside runtime
START RequestId:  Version: SLATEST
END RequestId:
REPORT RequestId:  Duration: 292.35 ms  Billed Duration: 300 ms Memory Size: 128 MB

{"statusCode":200,"body":{"message":"hello world","location":" "}}

```

Figure 6: AWS Lambda local invocation example

For serverless service monitoring and troubleshooting, we configured AWS CloudWatch and AWS X-Ray which recognize problems and mitigate any risks. With CloudWatch, it easy to monitor key metrics and logs, visualize application and infrastructure stack, create alarms, and correlate metrics and logs to understand and resolve the root cause of performance issues. The AWS X-Ray also provides two powerful features that can improve the efficiency in Active tracing, provides distributed tracing capabilities as well as visual service maps for faster troubleshooting. X-Ray helps to identify performance degradation and quickly understand anomalies, including latency distributions.

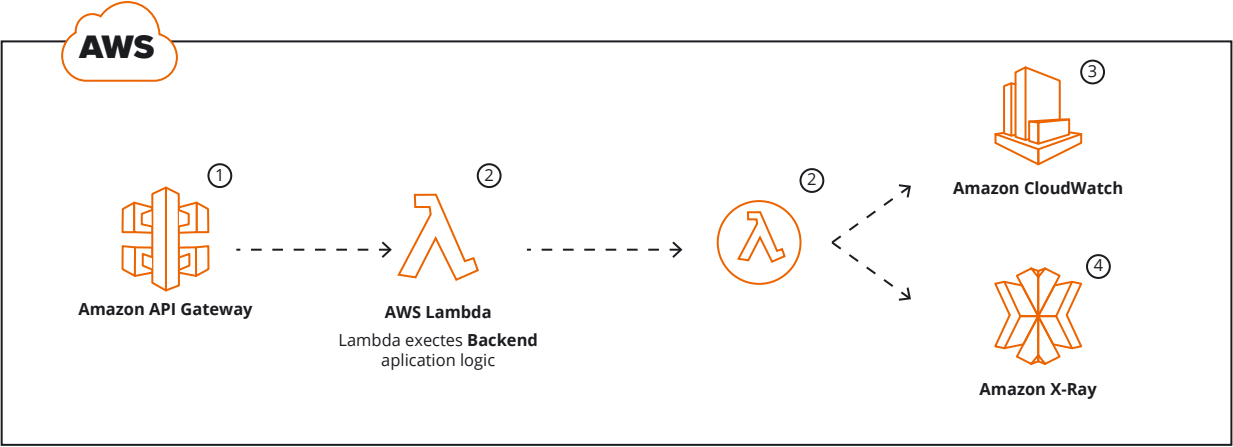


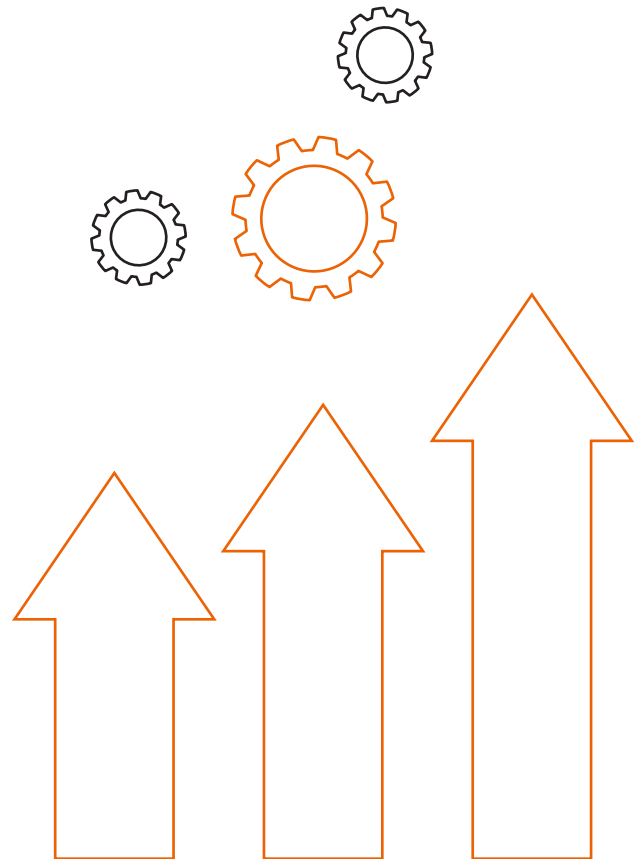
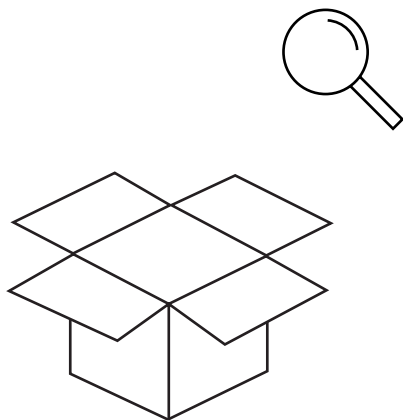
Figure 7: Monitoring and logging general overview

## Value Delivered

SoftServe delivered on time the prototype of the highly-available, robust and scalable serverless system. In tight collaboration with developers, we updated java code to the newer version and wrapped it up to fulfill the necessary serverless criteria. The prototype is ready for further development with no visible red flags for production in perspective.

### Within the PoC SoftServe delivered:

- Designing and implementation of such serverless services as AWS Lambda, API Gateway, DynamoDB, CloudFront, S3.
- Log-In and Registration flow with Amazon Cognito.
- Infrastructure as a Code by means of CloudFormation and AWS SAM.
- Serverless local development and testing using AWS SAM.
- Implemented Continues Integration/ Delivery for application's back-end parts.
- Implemented Continues Integration/ Delivery for application's front-end parts.
- Application Monitoring and Log aggregation with CloudWatch and X-RAY.



## ABOUT US

SoftServe is a digital authority that advises and provides at the cutting-edge of technology. We reveal, transform, accelerate, and optimize the way enterprises and software companies do business. With expertise across healthcare, retail, media, financial services, software, and more, we implement end-to-end solutions to deliver the innovation, quality, and speed that our clients' users expect.

SoftServe delivers open innovation—from generating compelling new ideas, to developing and implementing transformational products and services.

Our work and client experience is are built on a foundation of empathetic, human-focused experience design that ensures continuity from concept to release.

We empower enterprises and software companies to (re)identify differentiation, accelerate solution development, and vigorously compete in today's digital economy—No matter where you are in your journey.

Visit our [website](#), [blog](#), [Facebook](#), [Twitter](#), and [LinkedIn](#) pages.

### NORTH AMERICAN HQ

201 W 5th Street, Suite 1550  
Austin, TX 75703  
+1 866 687 3588

1 University Avenue Suite 11-112  
Toronto, ON M5J 2P1  
+1 647 948 7638

### EUROPEAN HQ

14 New Street  
London EC2M 4HE, UK  
+44 (0) 800 302 9436

[info@softserveinc.com](mailto:info@softserveinc.com)  
[www.softserveinc.com](http://www.softserveinc.com)

**softserve**