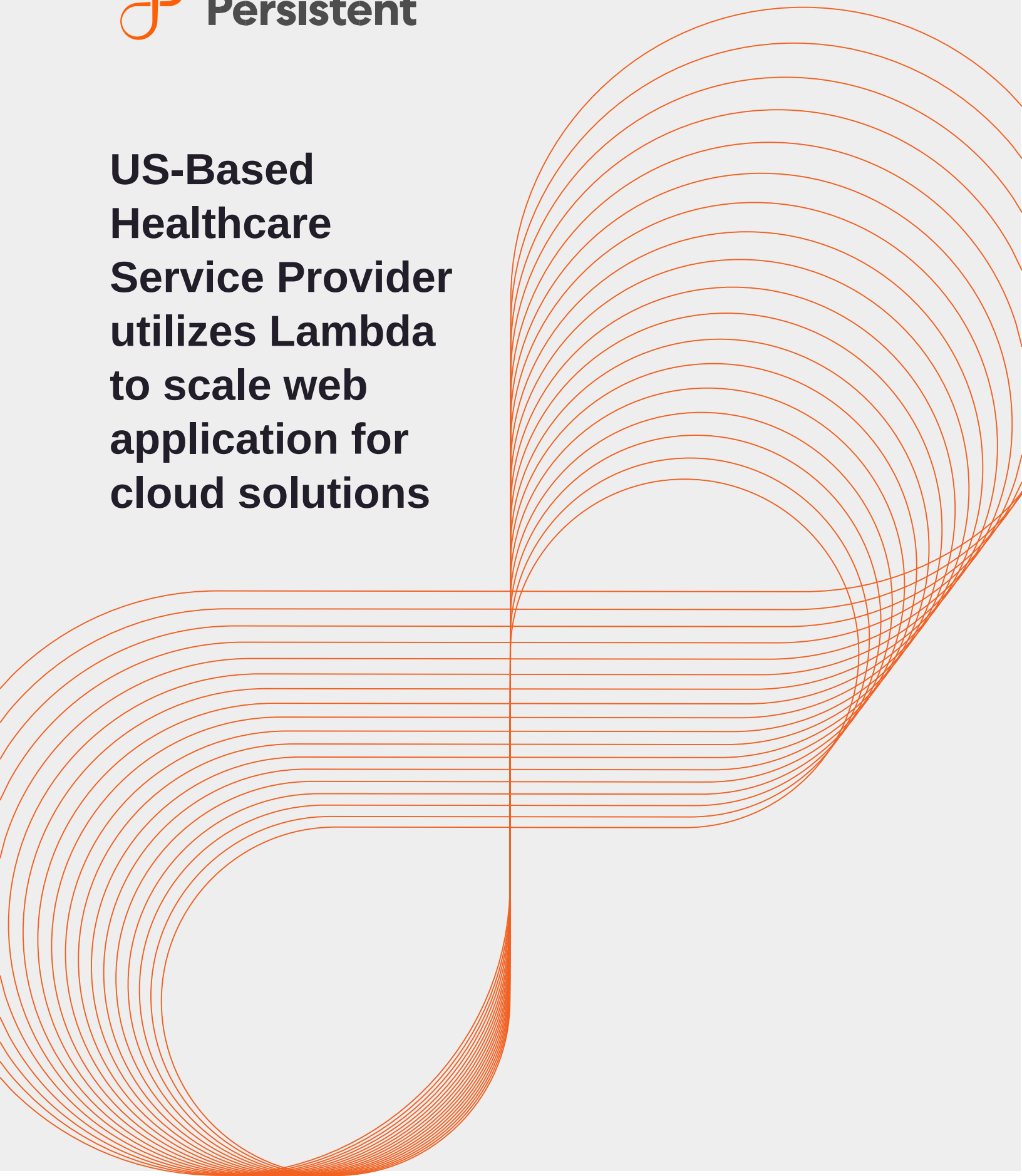
Persistent

US-Based Healthcare Service Provider utilizes Lambda to scale web application for cloud solutions



Persistent

Contents

1 About the Client 3
2 Problem Statement 3
3 What We Implemented 3
4 Outcomes and Benefits 5

1 About the Client

The customer is the market leader for healthcare cloud computing and information security services for providers, life sciences, payers, and healthcare technology organizations Their solutions enable customers to fully automate, protect, and securely manage healthcare applications, data, and IT infrastructure in the cloud.

2 Problem Statement

The customer needed a centralized web application solution for making cloud accounts HIPPA compliant. Any anomalies or non-compliance cloud resources needed to be addressed and remediated. As part of the application workflow requirements, processing logic needed to be incorporated at various levels. Due to the integration of a vast number of functionalities handled by different teams, having the processing logic distinct and independent is very important. Also, giving the choice of development platforms to their developers added to the requirement.

3 What We Implemented

Services Overview

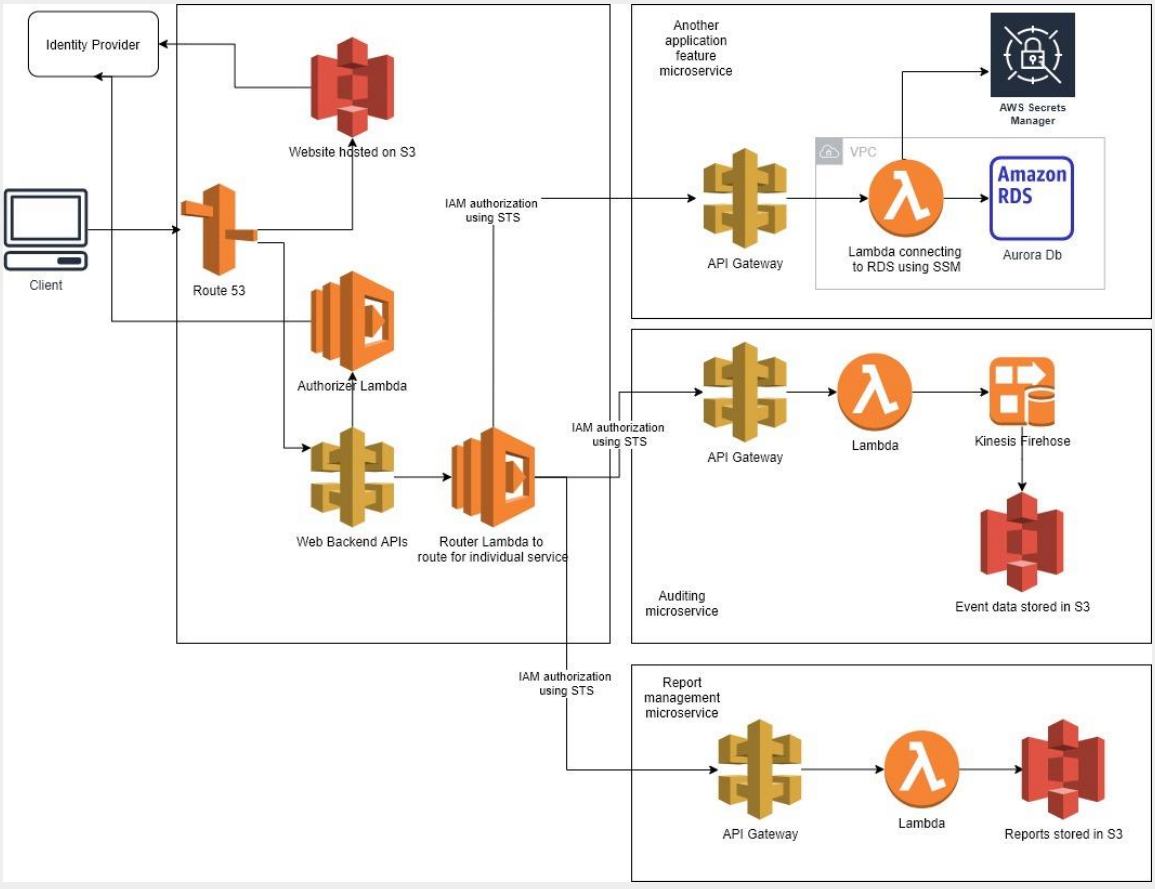
Lambda functions are used as a processing engine behind API Gateway which exposes functionalities either directly to other clients or to the web application. Automated patching on different customer environments is done using AWS scheduled automation with SSM, Lambda, and CloudWatch. HIPPA compliant AWS infrastructure is provisioned using Terraform, Lambda and config automation. Lambda is also used to handle and process the core data flows for the system. In total, 10 Lambda functions were developed for the project.

Lambda functions were utilized and developed for the following use cases:

- Consumer of API Gateway to processes requests sent to it
- Backend for web application via API Gateway so that each functionality can be independent
- Backend for web service via API Gateway for catering to requirements of independent functionalities
- Invoke functionalities from different accounts by invoking Lambda and API-Gateway
- Interface with and operate RDS instance in a VPC
- Lambda behind API Gateway used to consume events and log them to Kinesis Firehose

- Lambda behind API Gateway which in-turn invokes APIs from other environments for its business functionality
- Lambda behind API Gateway which in-turn invokes Lambdas from other environments for its business functionality

Lambda connects with S3, Kinesis, API Gateway, RDS (Aurora DB), SSM, VPC, CloudWatch, and IAM. Microservices serve as a backend for the web application, as auditing functionality used by multiple applications, and to interact with an RDS instance in a VPC. API Gateway is an interface for all the microservices written in Lambda. This ensures complete independence on changing the implementation as per requirement. Kinesis Firehose is used to stream continuous data while utilizing its features of buffering and direct delivery to S3. S3 is used to store certain data of the application such as event data and VPC ensures the database is secure and not accessible from outside. RDS (Aurora DB) is used a data store for one of the functionalities for which a relational data store is needed. SSM is used for the use case of storing the secrets such as passwords, usernames for the database and IAM ensures security of all the components as they interact with each other with minimum required privileges. CloudWatch is used to debug application functionality by emitting the traces of the Lambda functions and Go-Swagger is used to have an OpenAPI definition-based implementation for API Gateway to Lambda interfaces.



Solution Characteristics

For IAM policy definition, roles and policies are created for permissions to be granted to Lambda functions. Only the minimum required permissions are granted. Lambda Functions are designed to return an appropriate success/failure code, as it is behind API Gateway, and failures are logged in CloudWatch. Each microservice function is implemented as a different code module, defined independently in OpenAPI Specification via API gateway, and deployed as a module using Terraform. In most use cases, Lambda functions are executed outside VPC. For use cases where the Lambda functions need to interact with Amazon RDS, which is deployed in VPC, Lambda functions are deployed in VPC. The metrics of Lambda functions are used to manually monitor the executions over a period of time.

Lessons Learned

The Persistent team learned to keep independent functionalities in different Lambda functions to achieve total independence. Also, if Lambda is to be used as a backend for API Gateway which is based on a specification of Open API, it is best to use packages such as Go-Swagger to keep in sync with the structures.

4 Outcomes and Benefits

The result of the project utilizing Lambda provided better scalability, independent code structure, and freedom of choice of application language for the web application solution. Migration of functionalities from the legacy web application to the new web application was enabled with minimal impact. Due to the modular nature of the entire software that is designed using Lambda functions and API Gateway, onboarding of new members to the team and ensuring that they are productive has drastically reduced. Deployment can now be done independently, which is a major benefit as it reduces impact on the live production environment.

About Persistent Systems

Persistent Systems is a global company specializing in solutions, software products, and technology development services. Persistent Systems builds innovation that drives the business of our customer's enterprises with software at the core of their digital transformation. To learn more, visit http://www.persistent.com

India

Persistent Systems Limited

Bhageerath, 402,
Senapati Bapat Road, Pune 411 016
Tel: +91 (20) 6703 0000
Fax: +91 (20) 6703 009
Email: info@persistent.com

USA

Persistent Systems Inc

2055 Laurelwood Road, Suite 210,
Santa Clara, CA 95054
Tel: +1 (408) 216 7010
Fax: +1 (408) 451 9177
Email: info@persistent.com