|  |  |
| --- | --- |
| Oshawa Public Library  AWS ARCHITECTURE | Abstract  This document explains decision choices behind an AWS cloud architecture solution for the Oshawa Public Library with regard to creating and deploying a new microservice-based web application. The objective was to ensure the architecture was secure, highly available, and scalable to meet peak traffic demand for library users along with consideration for cost-effectiveness and maintainability. |

**Table of Contents**

[**Introduction** 2](#_Toc183385709)

[**Background** 2](#_Toc183385710)

[**Design Methodologies and Decisions** 2](#_Toc183385711)

[**Cloud Provider Selection** 3](#_Toc183385712)

[**Service Architecture** 3](#_Toc183385713)

[**Security** 3](#_Toc183385714)

[**Scalability & High Availability** 3](#_Toc183385715)

[**Backup & Monitoring** 3](#_Toc183385716)

[**Cost optimization** 3](#_Toc183385717)

[**Summary** 4](#_Toc183385718)

[**AWS Architecture** 4](#_Toc183385719)

[**Diagram** 4](#_Toc183385720)

[**Detailed Analysis of AWS Services** 5](#_Toc183385721)

[**References** 8](#_Toc183385722)

# **Introduction**

The project's objective was to present an appropriate cloud architecture solution for the Oshawa Public Library, which would support both the development and deployment of a newly developed microservice-based web application. This web application will be crucial in enhancing public access to library services through things such as user account management and processing of library data. This was to ensure that the architecture will be secure, of high availability, and scalable to meet the peak traffics of the demands from library users, also considering cost-effectiveness and maintainability.

The OPL, which currently is not on cloud infrastructure, has to migrate onto a cloud environment. The project involves the selection of the best cloud provider-AWS-for the architecture and incorporating various cloud services that will ensure high availability, security, and cost optimization.

# **Background**

The OPL's existing IT infrastructure is totally on-premise, and the customer wants to take advantage of cloud technology in order to increase flexibility, scalability, and maintainability. The new web application consists of two services:

* **Public Front-End Service**: Through this service, library users can access their accounts and interact with library resources.
* **Private Back-End API Service**: It will be used for data processing, maintenance of connections to the library database, and will implement business logic to support user requests from the front end.

The team was tasked with a cloud architecture solution to meet the following requirements:

* **Scalability**: The system should be able to handle all variable traffic loads, especially during peak usage times.
* **Security**: Sensitive data, such as user credentials and library data, should be appropriately safeguarded.
* **High Availability**: Ensure minimal downtime and uniform performance of the service.
* **Cost Efficiency**: Design the architecture in such a way that cloud resources are utilized effectively without overprovisioning or excessive costs.

***(The 6 Pillars of the AWS Well-Architected Framework | Amazon Web Services, 2022)***

# **Design Methodologies and Decisions**

The rationale behind the architecture decisions is centered on meeting the Oshawa Public Library's (OPL) requirements for scalability, security, high availability, and cost-effectiveness.

## **Cloud Provider Selection**

AWS was chosen because the services it offers are mature, its documentation is a good resource, and it has great support for services such as EKS, RDS, and S3. AWS also boasts data centers across the globe, with availability zones located in Toronto, Canada, to meet the library's geographic requirements.

## **Service Architecture**

The architecture was designed to be scalable and modular, with a microservices model. The public front-end and the private back-end were encapsulated in Docker containers, which were then deployed on Amazon EKS for efficient container management. ***(Mark Maglana, 2021)***

## **Security**

To ensure application is secure:

* AWS WAF, or Web Application Firewall, was set up to protect the application against common web exploits.
* IAM roles and AWS KMS were set up to manage access and encrypt sensitive data.
* VPC with private subnets was used to isolate sensitive services such as the backend API and RDS database from the public internet.

## **Scalability & High Availability**

* Amazon ALB, or Application Load Balancer, was used to distribute the incoming traffic across several EKS worker nodes deployed in two different AZs in Toronto. The cluster is designed for high availability and horizontal scalability.
* Amazon RDS was deployed in a multi-AZ setup to ensure that the database is resilient, fault-tolerant.

## **Backup & Monitoring**

* AWS Backup was implemented to provide a daily backup for EKS Cluster and RDS.
* Amazon CloudWatch was set up to monitor the health of the infrastructure and log relevant system metrics.

## **Cost optimization**

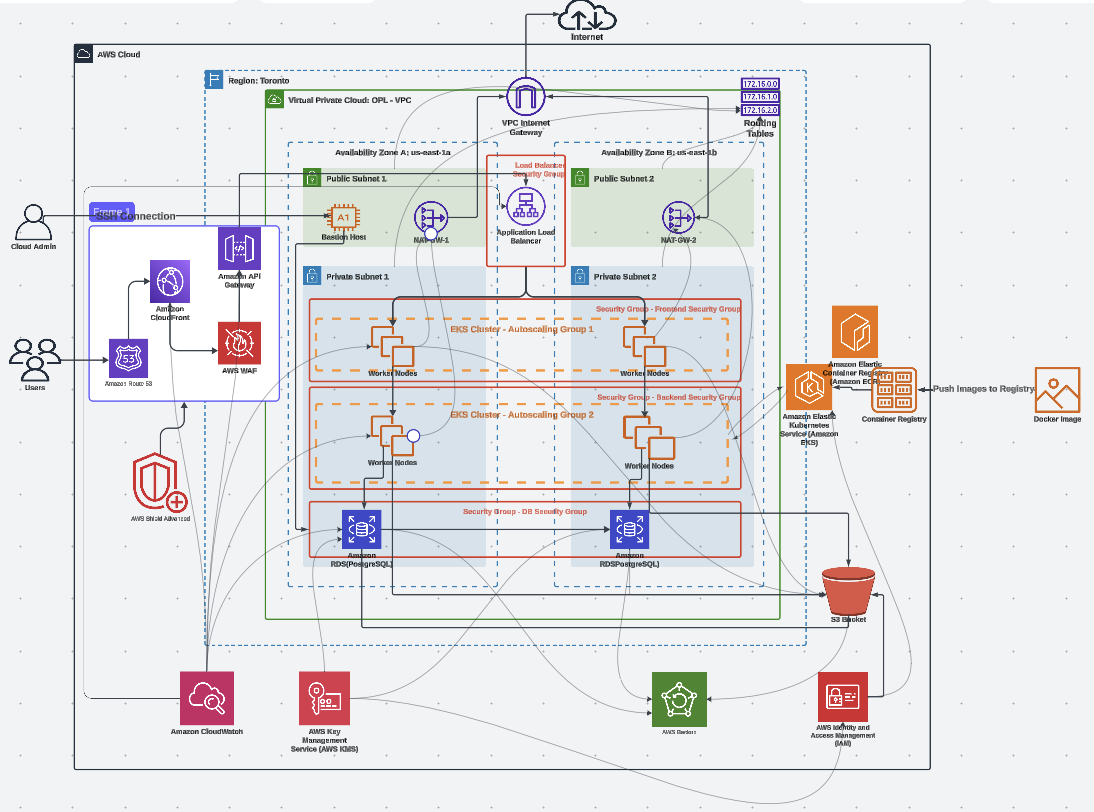
Using on-demand and auto-scaling features optimizes costs while meeting demand.

## **Summary**

Oshawa Public Library's proposed architecture has been designed to utilize AWS services to provide a scalable, secure, and highly available solution for their web application, which is microservice-based. Amazon EKS orchestrates containerized services for efficient workload management, while RDS ensures database operations are reliable and encrypted. Storage needs are addressed by S3 for library assets and EBS for high-performance workloads. Networking components such as VPC, subnets, NAT Gateway, and ALB allow for secure and effective management of traffic. Security measures such as AWS WAF, Shield, KMS, IAM provide protection against various types of threats to ensure data is kept safe. CloudFront enhances content delivery at low latency, while CloudWatch, AWS Backup, and X-Ray ensure monitoring, traceability, and data recovery. This design gives a solid, cost-effective basis for enhancing public access to library services. ***(The 6 Pillars of the AWS Well-Architected Framework | Amazon Web Services, 2022)***

# **AWS Architecture**

## **Diagram**



***(Yahoo Forma Parte De La Familia De Marcas De Yahoo, n.d.)***

## **Detailed Analysis of AWS Services**

Below is the list of all AWS services discussed, categorized by their function, with their roles in solving the OPL problem:

**Compute**

**Amazon Elastic Kubernetes Service (EKS)**

* Runs the containerized public front-end and private back-end services.
* Provides scalability and orchestration for the microservices with worker nodes.
* Manages Docker containers and is highly available.

***(What Is Amazon EKS? - Amazon EKS, n.d.)***

**Amazon Elastic Container Registry (ECR)**

* Securely stores the Docker images of the front-end and back-end services.
* Ensures easy integration with EKS for image deployment.

**Storage**

**Amazon S3**

* Provides object storage for library assets, like images, in a durable and cost-effective manner.
* Integrated with KMS for encryption of stored data for security.

**Database**

**Amazon RDS (PostgreSQL)**

* Provides a scalable, multi-AZ relational database that will be used for managing library data.
* Configured for High Availability and fault tolerance to provide uninterrupted access to library records.
* It supports encryption at rest using AWS KMS

**Networking**

**Virtual Private Cloud (VPC)**

* Provides network isolation and segmentation for the library resources.
* Isolated to public and private subnets to securely deploy internet-facing services and backend services.

**Subnets (Public and Private)**

* Public Subnets: This contains bastion host and NAT Gateway.
* Private Subnet: Contains EKS worker nodes, RDS and sensitive backend components.

**Internet Gateway (IGW)**

Provides internet access for public-facing resources such as the ALB and CloudFront.

**NAT Gateway**

Allows private subnet resources-for example, EKS worker nodes and RDS-to safely reach out to the internet for updates without being directly exposed to it.

**Application Load Balancer (ALB)**

* Distributes incoming traffic to EKS worker nodes for efficient handling of requests.
* Ensure scalability and high availability.

**Route Tables**

Routing paths definition for traffic within the VPC, which connects subnets to IGW, NAT Gateway, and other components.

***(AWS Global Infrastructure Explainer Video | Amazon Web Services, n.d.)***

**Security**

**AWS IAM**

* Manages permissions and access policies for users and services interacting with the architecture.
* Ensures least-privilege access to resources.

**AWS Key Management Service (KMS)**

* Encrypts sensitive data in RDS, S3, CloudWatch logs, and other critical storage systems.
* Provides key management for data security and compliance.

**Bastion Host**

Provides secure SSH access to administrative users to private resources in the VPC. A jump server onto which sensitive components should not be directly exposed to the internet.

**AWS Web Application Firewall (WAF)**

* Protects the application from common web exploits like SQL injection and cross-site scripting.
* Integrated with CloudFront for enhanced security at the edge.

**API Gateway**

API Gateway securely exposes API endpoints to the public front-end service or internal consumers. API Gateway integrates with backend services like EKS to process API requests.

**AWS Shield Advanced**

Prevents DDoS attacks, leveraging the architecture to ensure services, such as the ALB and CloudFront, are available even under attack conditions.

***(Cloud Security, Identity, and Compliance Products – Amazon Web Services (AWS), n.d.)***

**Monitoring and Logging**

**Amazon CloudWatch**

* Monitor metrics, logs, and alarms on all EKS, RDS, and ALB resources.
* Provides real-time insights into system performance and troubleshooting.

**Backup and Recovery**

**AWS Backup**

* Automates backups of EKS, RDS, and S3 to ensure data is not lost.
* Provides point-in-time recovery for critical resources in case of failure or disaster.

**Content Delivery**

**Amazon CloudFront**

* Delivers static and dynamic content with low latency.
* Provides caching and integrates with WAF for enhanced security at the edge.

**Amazon Route 53**

* Provides DNS for the application to make sure users can reach the library's public-facing front end with a friendly, easy-to-remember domain name.
* It also supports routing traffic to the Application Load Balancer based on the routing policies.

# **References**

1. *Mark Maglana. (2021, July 3). System Design (Part 1): 3-tier app on AWS EKS [Video]. YouTube.* [*https://www.youtube.com/watch?v=8-M2rK4NRyI*](https://www.youtube.com/watch?v=8-M2rK4NRyI)
2. *The 6 Pillars of the AWS Well-Architected Framework | Amazon Web Services. (2022, March 1). Amazon Web Services.* [*https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/*](https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/)
3. *Yahoo forma parte de la familia de marcas de Yahoo. (n.d.).* [*https://images.search.yahoo.com/search/images;\_ylt=AwrNPNaZTUJnjvEfcryJzbkF;\_ylu=c2VjA3NlYXJjaARzbGsDYnV0dG9u;\_ylc=X1MDOTYwNjI4NTcEX3IDMgRmcgNtY2FmZWUEZnIyA3A6cyx2OmksbTpzYi10b3AEZ3ByaWQDTF9EQ2p3ZGZRbUdWTFpSR0VDcWZRQQRuX3JzbHQDMARuX3N1Z2cDMwRvcmlnaW4DaW1hZ2VzLnNlYXJjaC55YWhvby5jb20EcG9zAzAEcHFzdHIDBHBxc3RybAMwBHFzdHJsAzE3BHF1ZXJ5A2VrcyUyMGFyY2hpdGVjdHVyZSUyMAR0X3N0bXADMTczMjUwNzIzNg--?p=eks+architecture+&fr=mcafee&fr2=p%3As%2Cv%3Ai%2Cm%3Asb-top&ei=UTF-8&x=wrt&type=E210US885G0#id=83&iurl=https%3A%2F%2Fd1.awsstatic.com%2Fpartner-network%2FQuickStart%2Fdatasheets%2Feks-rafay-systems-architecture-diagram.5de8b6d2b5352a48cb909137634810881c09ec2e.png&action=click*](https://images.search.yahoo.com/search/images;_ylt=AwrNPNaZTUJnjvEfcryJzbkF;_ylu=c2VjA3NlYXJjaARzbGsDYnV0dG9u;_ylc=X1MDOTYwNjI4NTcEX3IDMgRmcgNtY2FmZWUEZnIyA3A6cyx2OmksbTpzYi10b3AEZ3ByaWQDTF9EQ2p3ZGZRbUdWTFpSR0VDcWZRQQRuX3JzbHQDMARuX3N1Z2cDMwRvcmlnaW4DaW1hZ2VzLnNlYXJjaC55YWhvby5jb20EcG9zAzAEcHFzdHIDBHBxc3RybAMwBHFzdHJsAzE3BHF1ZXJ5A2VrcyUyMGFyY2hpdGVjdHVyZSUyMAR0X3N0bXADMTczMjUwNzIzNg--?p=eks+architecture+&fr=mcafee&fr2=p%3As%2Cv%3Ai%2Cm%3Asb-top&ei=UTF-8&x=wrt&type=E210US885G0#id=83&iurl=https%3A%2F%2Fd1.awsstatic.com%2Fpartner-network%2FQuickStart%2Fdatasheets%2Feks-rafay-systems-architecture-diagram.5de8b6d2b5352a48cb909137634810881c09ec2e.png&action=click)
4. *AWS Global Infrastructure Explainer Video | Amazon Web Services. (n.d.). [Video]. Amazon Web Services, Inc.* [*https://aws.amazon.com/getting-started/aws-networking-essentials/*](https://aws.amazon.com/getting-started/aws-networking-essentials/)
5. *What is Amazon EKS? - Amazon EKS. (n.d.).* [*https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html*](https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html)
6. *Cloud Security, Identity, and Compliance Products – Amazon Web Services (AWS). (n.d.). Amazon Web Services, Inc.* [*https://aws.amazon.com/products/security/*](https://aws.amazon.com/products/security/)