

29-02-2024

# Building a Cost-Effective REST Application on AWS cloud

---

A complete Serverless Application on AWS

AWS UG Calabria 2024



# Meet the Speaker

---

Building a Cost-Effective REST Application on  
AWS cloud



Alessandro Pagliaro  
Senior Software Engineer

# Agenda

Building a Cost-Effective REST  
Application on AWS cloud

AWS UG Calabria 2024

---

A brief Introduction to the App

1

---

Why Choosing Lambda

2

---

Introduction to GraalVM and Quarkus

3

---

Exploring AWS services: S3, DynamoDB, API  
Gateway, Cognito, WAF.

4

---

Application Architecture

5

---

Deploy the POC in your environment

6

---

DEMO and Cost Analysis

7



# Introduction

The Movie Store Application represents a proof-of-concept (POC) solution that demonstrates the power of serverless AWS architecture.

## Goal of this sample project:

Create a cost effective and scalable application that has a near zero cost if not used.

## What it does:

- Manages movie data including titles, actors, genres, min age, and cover images.
- Enables users to browse, search, and view detailed movie information.

```
{
  "title": "The pursuit of happiness",
  "minAge": 3,
  "genre": ["comedy", "thriller"],
  "actors": [
    {
      "name": "Will Smith",
      "role": "Chris Gardner",
      "gender": "MALE"
    },
    {
      "name": "Jaden Smith",
      "role": "Christopher Gardner",
      "gender": "MALE"
    },
    {
      "name": "Thandie Newton",
      "role": "Linda",
      "gender": "FEMALE"
    }
  ]
}
```

# Why Choosing Lambda

## **Problem:**

Running a backend without incurring costs during idle periods is challenging.

## **Solution:**

AWS Lambda's FaaS model eliminates idle costs by executing code in response to events, scaling automatically to match demand.

## **Limitations:**

**Cold starts:** especially with Java-based functions, may impact performance.

**Size limitation:** it's essential to use lightweight frameworks and libraries.

**File Storage:** Lambda functions are stateless and lack persistent storage.



# Addressing Cold Start: GraalVM

## GraalVM's native compilation feature:

- ahead of time (AOT) compilation (No JVM needed);
- reduced startup times and memory consumption;
- ideal for modern cloud-native and serverless environments.
- Support polyglot programming



Cold start duration with  
standard JDK compilation  
~ 6 seconds



AWS lambda Cold start duration  
with GraalVM compilation  
~ 1 seconds

# Addressing size limitation: Quarkus

Quarkus is a Kubernetes-native Java framework designed to build lightweight, fast, and efficient Java applications suitable for modern cloud environments.

With its ahead-of-time compilation capabilities and support for GraalVM, Quarkus offers significantly reduced startup times and memory usage compared to traditional Java frameworks.



The Quarkus Universe provides a collection of extensions which facilitates integration with AWS services such as:

- Lambda;
- S3;
- DynamoDB;
- etc.

and offers compatibility with GraalVM compilation

# Addressing File Storage: S3

Amazon Simple Storage Service (S3) is a highly scalable, cost effective and durable object storage service offered by AWS.

With **Signed URLs** feature you can Generate secure, temporary URLs to grant controlled access to objects stored in your S3 bucket.

## **Problem:**

How to know that a file is uploaded in my backend?

## **Solution:**

**S3 Notifications:** Receive notifications when specific events occur in your S3 bucket, such as object creation, deletion, or metadata changes.



S3



# Api Gateway

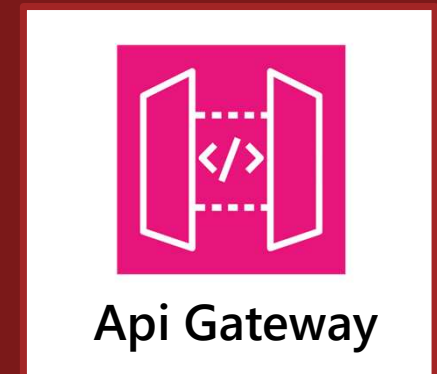
Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

Key features of API Gateway include:

- Cognito Authorizer;
- WAF (Web Application Firewall) integration;
- Custom Domain;
- Lambda Integration;
- ...

## **Problem:**

API Gateway REST API's default integration timeout is 29 seconds.



# Cost Effective Database: DynamoDB

DynamoDB is a fully managed NoSQL database service provided by AWS, offering:

- seamless scalability;
- high performance;
- low latency for applications of any scale.

DynamoDB have limitations when it comes to querying:

- primarily supports querying based on the primary key;
- querying based on attributes other than the primary key often requires filtering data on the client side;

To enhance query:

- Global Secondary Index (GSI);
- Local Secondary Index (LSI):



# AWS Cognito and WAF

## **AWS Cognito:**

AWS Cognito serves as a robust identity and access management solution, streamlining user authentication and authorization for applications. With features like user pools and identity pools, it provides scalable user directories and secure access to AWS resources.

## **AWS WAF (Web Application Firewall):**

AWS WAF (Web Application Firewall) fortifies your applications against common web vulnerabilities and malicious attacks. Offering managed and custom rule sets, along with rate-based rules and seamless integration with other AWS services, AWS WAF ensures the protection and integrity of your web applications.



**Cognito**



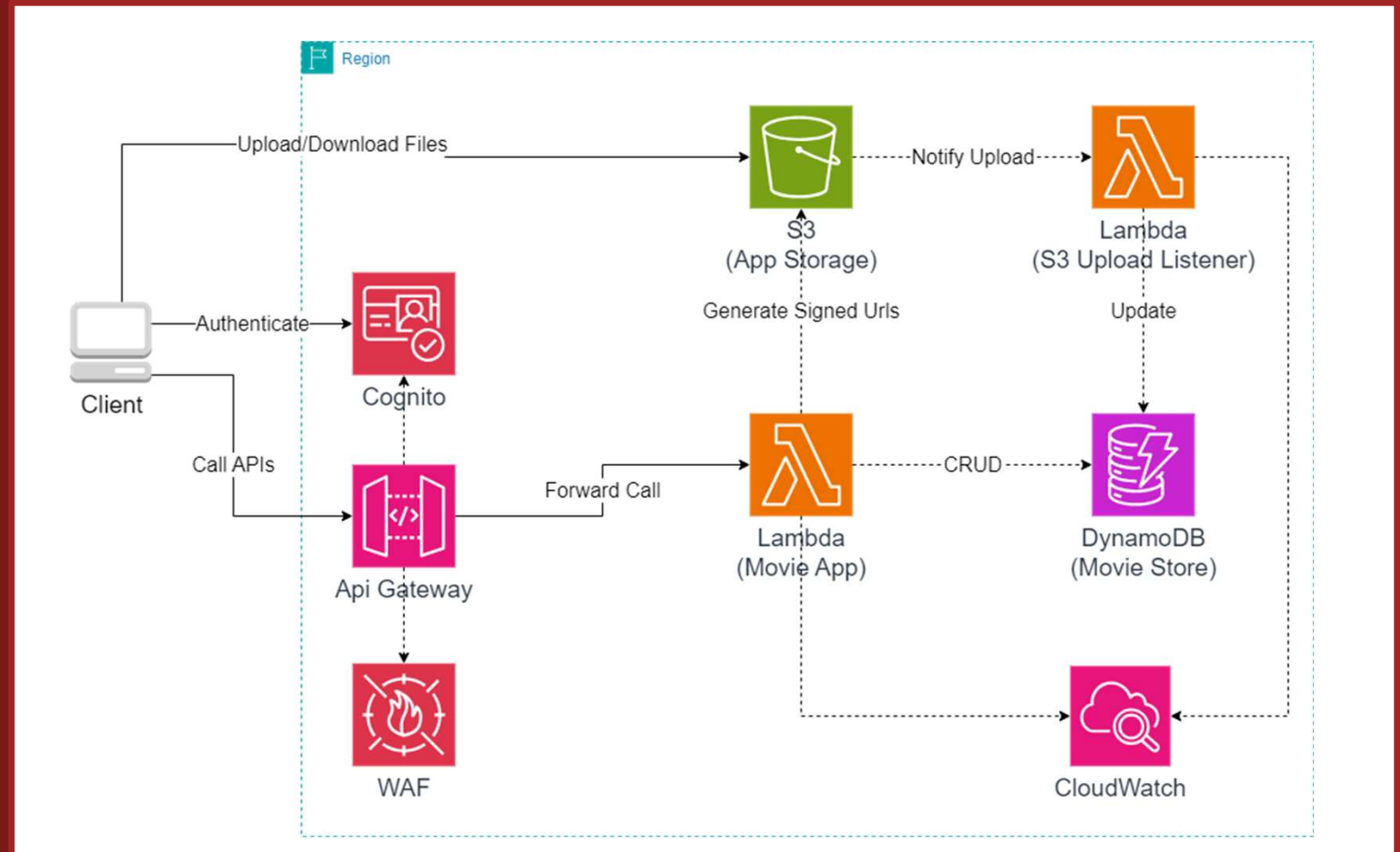
**WAF**



# Application Architecture

## Services Used:

- AWS Lambda;
- AWS S3;
- DynamoDB;
- AWS API Gateway;
- AWS Cognito;
- AWS WAF
- CloudWatch.



# Deploy the POC in your environment

To deploy the sample application in your environment, you can follow the instructions provided in the open-source repository. The repository contains all the necessary files and documentation to simplify the deployment process.

## Repository Link:

<https://github.com/paaxel/graalvm-quarkus-lambda>

This sample application is open-source and distributed under the GNU General Public License version 3.0 (GPLv3).

## Suggested deployment process:

- Clone the Repository;
- Read the README instructions;
- Use CloudFormation file and run the PowerShell Script.







# DEMO TIME

Exploring the Movie Store Application



# Cost Analysis

**Total Daily Cost:** The solution operates at a low total daily cost of \$0.35, with the majority attributed to WAF expenses amounting to \$0.28.

**Total Monthly Cost:** ~ \$11

Is always necessary WAF? Not always:

- Rate limiting by IP is prebuilt in ApiGateway;
- IP filtering can be done using Api Gateway resource policy;

Without WAF:

- Daily Cost: ~ \$0.07
- Monthly Cost: ~ \$2

# Api Gateway Allow IP Resource Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:eu-central-1:{ACCOUNT_ID}:{REST_API_ID}/*//*/.*"
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:eu-central-1:{ACCOUNT_ID}:{REST_API_ID}/*//*/.*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            {SOURCE_IPS_TO_ALLOW}
          ]
        }
      }
    }
  ]
}
```

# Conclusions



The solution developed is:

- **High available:** Our system ensures uninterrupted access to services, even during peak demand or unexpected failures;
- **Scalable:** Designed to adapt to changing workloads;
- **Secure:** AWS WAF and Amazon Cognito increase protection against potential threats and vulnerabilities; Serverless services are updated by AWS.
- **Cost Effective:** Low operational cost, low cost on maintenance directly managed by AWS.





# Thanks for Attention

A complete Serverless Application on AWS

AWS UG Calabria 2024