**Internship Task 3 Documentation: Multi-Cloud Architecture**

**Objective**

The goal of this task was to design and demonstrate a simple **multi-cloud architecture**, where services are split across two different cloud providers and communicate with each other. The main requirement was to showcase **interoperability** between platforms — in this case, **Google Cloud** and **AWS**.

**Cloud Services Used**

**1. Google Cloud (Frontend)**

* **Service:** Google Cloud Storage
* **Purpose:** Host a static website (HTML file)
* **What it does:** Displays a simple webpage with a button. When clicked, it sends a request to AWS to fetch data.

**2. AWS (Backend)**

* **Service:** AWS Lambda + API Gateway
* **Purpose:** Host a serverless backend function
* **What it does:** Receives requests from the frontend and sends back a JSON message.

**How It Works**

1. A user visits the webpage hosted on **Google Cloud Storage**.
2. The page displays a heading and a button labeled “Get AWS Response”.
3. When the button is clicked, a **JavaScript fetch() request** is sent to an **AWS API Gateway** URL.
4. API Gateway routes the request to an **AWS Lambda function**.
5. The Lambda function returns a response like:
6. { "message": "Hello from AWS Lambda!" }
7. The frontend displays this response on the webpage.

**Steps Followed**

**Google Cloud Setup:**

* Created a **Cloud Storage bucket**
* Enabled public access and made it a **static website**
* Uploaded an index.html file containing HTML + JavaScript

**AWS Setup:**

* Created a **Lambda function** using Python:
* def lambda\_handler(event, context):
* return {
* 'statusCode': 200,
* 'headers': { 'Content-Type': 'application/json' },
* 'body': '{"message": "Hello from AWS Lambda!"}'
* }
* Set up **API Gateway** to make the function publicly accessible
* Enabled **CORS** to allow Google Cloud to make requests to this URL

**What I Learned**

* Basics of hosting a website on **Google Cloud Storage**
* How to create and deploy **AWS Lambda** functions
* How to expose a function via **API Gateway**
* Understanding of **CORS (Cross-Origin Resource Sharing)**
* Real-world application of **multi-cloud interoperability**

**Conclusion**

This project was a beginner-friendly example of how different cloud platforms can work together in a real-world scenario. I built a very simple but functional multi-cloud architecture using free-tier services and learned core cloud computing concepts in the process.