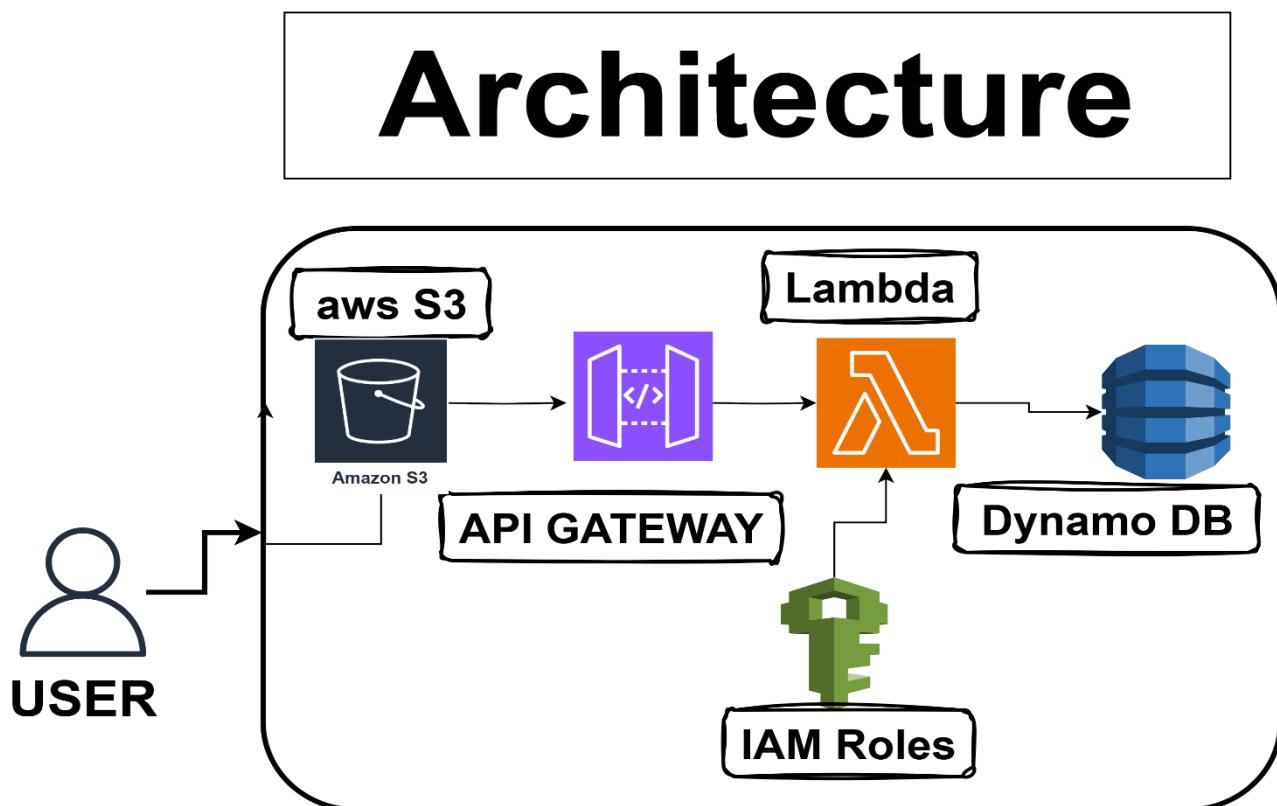


Asset Management System- Serverless Cloud Migration Project

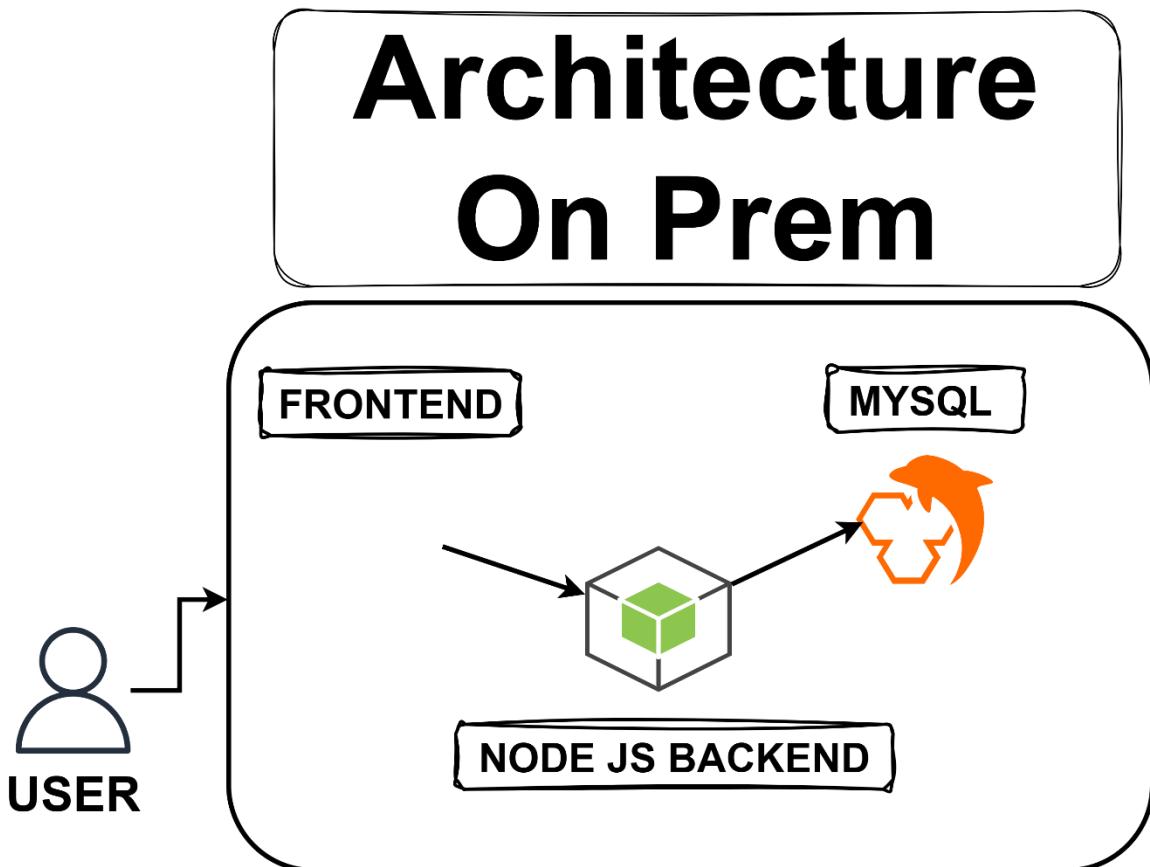
Project Overview

This project showcases the migration of a legacy on-premise Asset Management System to a modern, fully serverless architecture on AWS. It includes frontend deployment, backend re-architecture, and database migration from MySQL to DynamoDB. The application allows for complete CRUD (Create, Read, Update, Delete) operations on asset data.



Video Demo Link: - [Click Here for Video Demo](#)

- On premises



1. Amazon S3 (Frontend Hosting)

The frontend of the Asset Management System is hosted on Amazon S3 using static website hosting. It serves the index.html file to users over HTTP.

The screenshot shows the AWS S3 console interface. The top navigation bar includes tabs for Cloud Migration, API Gateway, Items, CloudWatch, and getAssets-role. The main navigation bar shows the user is in the 'Amazon S3 > Buckets > ams-frontend-static-site' section. The user 'SULAMAN @ sulaman-admin' is logged in. The 'Info' button for the bucket is visible. Below the navigation, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is selected. A sub-header 'Objects (1)' is shown. There are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A note states: 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' A search bar with the placeholder 'Find objects by prefix' is present. A table lists the single object: 'index.html' (Type: html, Last modified: June 30, 2025, 21:49:17 (UTC+05:30), Size: 6.4 KB, Storage class: Standard).

2. API Gateway (Routing Requests)

API Gateway routes HTTP requests to the correct Lambda function (GET, POST, PUT, DELETE). It also handles CORS and connects frontend to backend.

The screenshot shows the AWS API Gateway Routes page. On the left, a sidebar menu includes Develop, Deploy, Monitor, and Protect sections. The main content area displays a list of routes for the AssetGetAPI. A specific route for `PUT /assets/{id}` is selected, showing its details: ARN `arn:aws:apigateway:ap-south-1::apis/wbm4nm5au4/routes/5nofhtu`, Authorization status (No authorizer attached), and Integration details (Integration ID `hyj2ul8`, Configure button). A 'Deploy' button is visible at the top right.

The screenshot shows the AWS API Gateway Integrations page. The sidebar menu is identical to the previous screen. The main content area shows the integration details for the `DELETE /assets/{id}` route. It specifies the Lambda function `deleteAsset (ap-south-1)` and provides options to Detach integration or Manage integration. Other integration details include the Integration ID `1g9a5yq`, Description (empty), Payload format version (2.0), Invoke permissions (Example policy statement), and Timeout (30000 milliseconds).

3. AWS Lambda (Backend Logic)

AWS Lambda hosts the backend logic for CRUD operations. Each function handles one HTTP method and interacts with DynamoDB securely.

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with navigation links like Dashboard, Applications, Functions, Additional resources, and Related AWS resources. The main area is titled "Functions (4)" and lists four functions:

Function name	Description	Package type	Runtime	Last modified
getAssets	-	Zip	Python 3.10	36 minutes ago
deleteAsset	-	Zip	Python 3.10	29 minutes ago
addAsset	-	Zip	Python 3.10	1 hour ago
updateAsset	-	Zip	Python 3.10	18 minutes ago

4. DynamoDB (Database)

DynamoDB stores asset records with id as the primary key. All CRUD operations from Lambda read/write to this table.

The screenshot shows the AWS DynamoDB Items Explorer. On the left, there's a sidebar with navigation links like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main area is titled "Table: Assets - Items returned (11)" and displays the following data:

id (String)	assigned_to	purchase_date	status
3	Reception Desk	2021-11-22	In Use
4	Design Team	2023-09-01	In Use
5	IT Inventory S...	2022-08-03	Available
6	Support Engi...	2022-04-18	End Of Life ...
7	Graphics Dep...	2023-03-27	In Use
8	Conference R...	2022-01-30	In Use
9	HR Department	2023-02-14	Available
ad827aa6-38b1-4a0b...	test123	2025-04-30	End Of Life ...