Contents

Implementation flow

Sr. No.	Module	Coverage
1	**Amazon S3 Bucket** Creation for Data Storage	Provisioned and configured Amazon S3 buckets to securely store input datasets and processed output files. Ensured optimal organization and access control to support efficient data workflows and integration with downstream analytics and machine learning pipelines.
2	**AWS Glue** ETL Workflow for ML-Ready Data Preparation	Designed and implemented an AWS Glue ETL pipeline to apply business transformation rules, creating augmented datasets optimized for machine learning consumption. The ETL process output was stored in both Amazon S3 and Amazon RDS Aurora, enabling scalable storage and transactional support. Leveraged Amazon Athena for interactive data analysis and validation of transformed datasets.
3	**Amazon SageMaker** - Feature Engineering and Model Development	Utilized Amazon SageMaker for advanced feature engineering and machine learning model development. Leveraged SageMaker Notebooks to preprocess data, train, and validate machine learning models. Successfully deployed the trained model as a managed endpoint for seamless integration and real-time inference.
4	**AWS Lambda Function** for ML Endpoint Integration	Developed an AWS Lambda function to test, access, and deploy the machine learning endpoint. The function was designed to preprocess input data, invoke the ML endpoint for inference and handle responses efficiently. This streamlined integration with the ML endpoint ensured seamless deployment and scalability for real-time and batch processing use cases.
5	**Amazon API Gateway** for Application Integration	Configured Amazon API Gateway to serve as a scalable endpoint for the end application. The gateway routes incoming requests to the AWS Lambda function, enabling seamless preprocessing and invocation of the ML endpoint. This setup provides secure, low-latency

		access interaction with the deployed machine learning service.
6	**Postman** API Testing for ML Endpoint Validation	Utilized Postman to test the Amazon API Gateway endpoint by sending sample data inputs. Verified the functionality, response accuracy, and latency of the integrated ML processing pipeline, ensuring the endpoint's readiness and reliable interaction with the end application.
7	**IAM Roles and Policies** for Secure Managed Service Access	Defined and configured AWS Identity and Access Management (IAM) roles with policy-level permissions to enable secure access to managed services across the workflow. These roles ensured seamless integration and controlled access to resources such as S3, Glue, Lambda, API Gateway, and SageMaker while adhering to the principle of least privilege for enhanced security and compliance.

Document Overview

Purpose and Objective

This document highlights the capabilities of the AWS cloud environment in addressing healthcare challenges through managed services. The focus is on leveraging AWS's robust, scalable, and secure services to streamline data processing, enhance accessibility and improve healthcare solutions without delving into complex machine learning problems.
By prioritizing simplicity and efficiency, the outlined approach demonstrates how AWS managed services can optimize healthcare operations, enable seamless integration and support actionable insights with minimal technical overhead.

Scope

This document demonstrates AWS Machine Learning capabilities through a sample use case of diabetes prediction based on patient test reports. The use case employs fabricated data to showcase how AWS platform services can address real-world healthcare challenges, enabling predictive analytics and decision support in a scalable and efficient manner.

Assumptions & Constraints

Purpose and Focus: The primary goal of the assignment is to demonstrate the machine learning capabilities of AWS services using a practical use case. The exercise showcases the candidate's proficiency with AWS infrastructure, managed services, and related skill sets.

Cost Management: The AWS setup incurs an approximate cost of $2 per day, emphasizing the need for efficient use of resources. To save costs, the environment will be deleted after execution and the implementation details will be documented through screenshots and a Git repository.

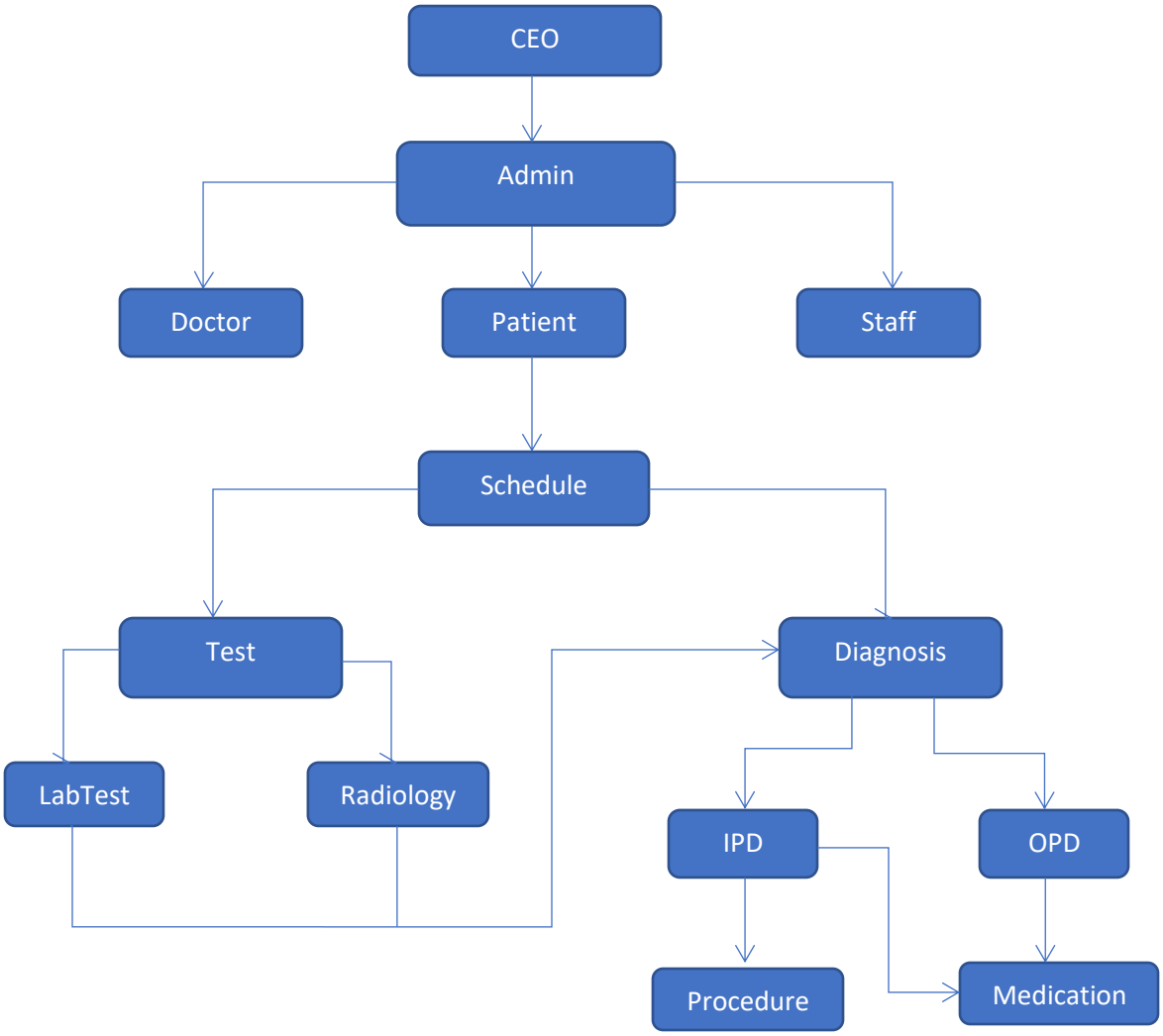
Resource and Execution Scope: The demonstration uses fabricated data for simulation and avoids deploying large-scale or resource-intensive ML solutions.

AWS Roles and Policies

Name	**Permission**
Amazon SageMaker role	Amazon RDS Access, Amazon S3 Access, Sage Maker Access, SageMaker Execution Access, Amazon CloudWatch Logs Access
AWS Glue ETL Role	Amazon Athena Access, Amazon RDS Access, Amazon S3 Access, AWS Glue Service Access, Amazon CloudWatch Logs Access, AWS Glue Catalog Permissions
Lambda Function Role	Amazon S3 Access, Amazon Sage Maker Access,AWS Lambda Execution Access, Amazon CloudWatch Logs Access
Anthena Role	AWS Athena Spark Execution Access, Amazon S3 and GetObject Access, AWS Glue Catalog Permissions

Understanding the requirement

General Process Model



Understanding the source data

Table: Patient

Column	Data Type	Reference Key
patient_id	Int	Primary Key
fname	String	
mname	String	
lname	String	
gender	String	
marital_status	String	
date_of_birth	Date	
aadhar_number	String	

Table: patient_labtest

Column	Data Type	Reference Key
patient_labtest_id	Int	Primary Key
patient_id	Int	Foreign Key to Patient.patient_id
heart_test_id	Int	Foreign Key to heart_test.heart_test_id
blood_test_id	Int	Foreign Key to blood_test.blood_test_id
general_body_test_id	Int	Foreign Key to general_body_test.general_body_test_id

Table: heart_test

Column	Data Type	Reference Key
heart_test_id	Int	Primary Key
patient_id	Int	Foreign Key to Patient.patient_id
heart_rate	Float	
heart_disease	Int	
blood_pressure	Float	

Table: blood_test

Column	Data Type	Reference Key
blood_test_id	Int	Primary Key
patient_id	Int	Foreign Key to Patient.patient_id
glucose	Float	
insulin	Float	
hbA1c_level	Float	

Table: general_body_test

Column	Data Type	Reference Key
general_body_test_id	Int	Primary Key
patient_id	Int	Foreign Key to Patient.patient_id
skin_thickness	Float	
bmi	Float	
diabetes_pedigree_function	Float	
weight	Float	
height	Float	
smoking_history	String	
pregnancies	Int	
hypertension	Int	

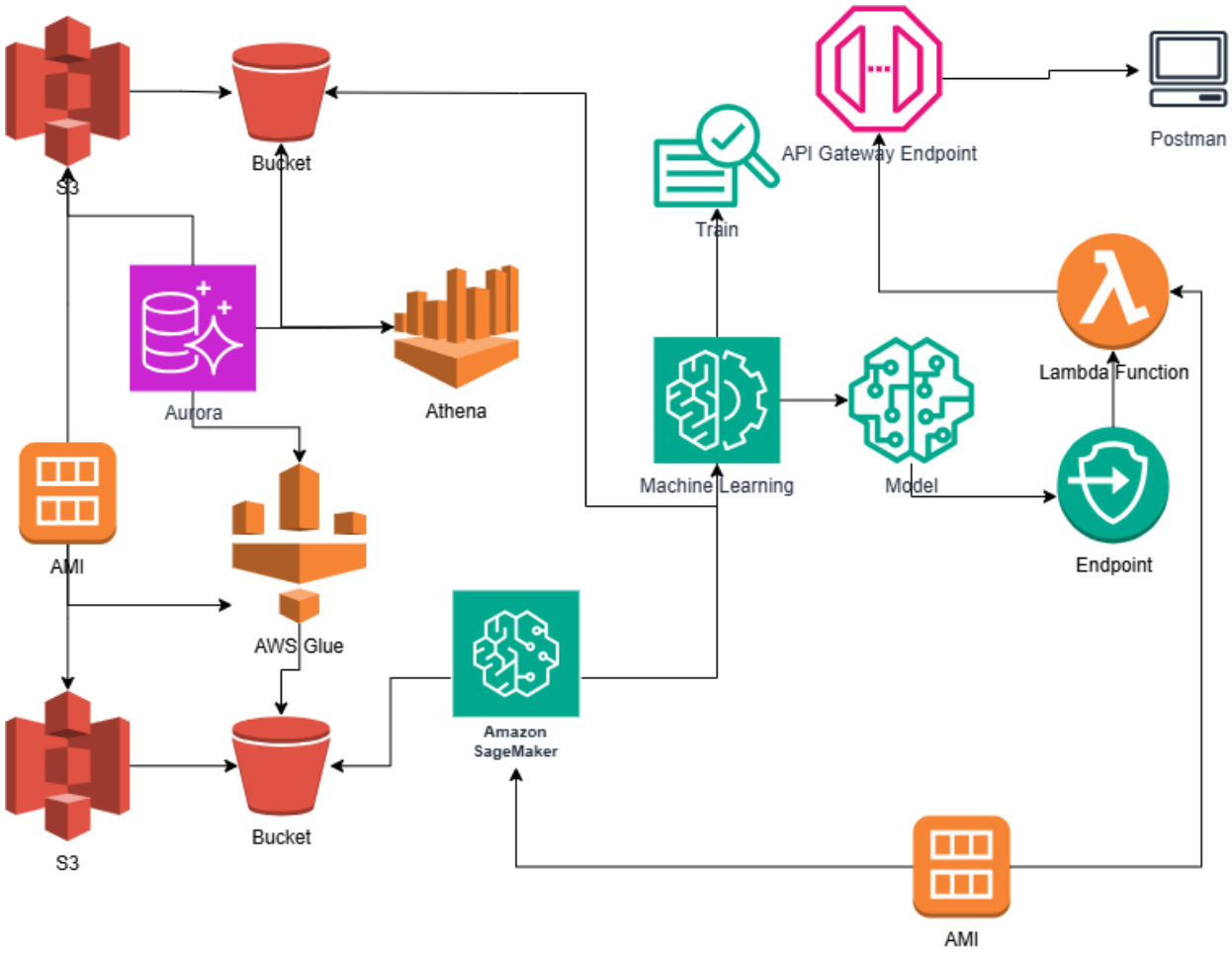
Table: diagnosis

Column	Data Type	Reference Key
patient_id	Int	Foreign Key to Patient.patient_id
patient_labtest_id	Int	Foreign Key to patient_labtest.patient_labtest_id
is_diabetic	Int	

Target Table: diabetes_prediction

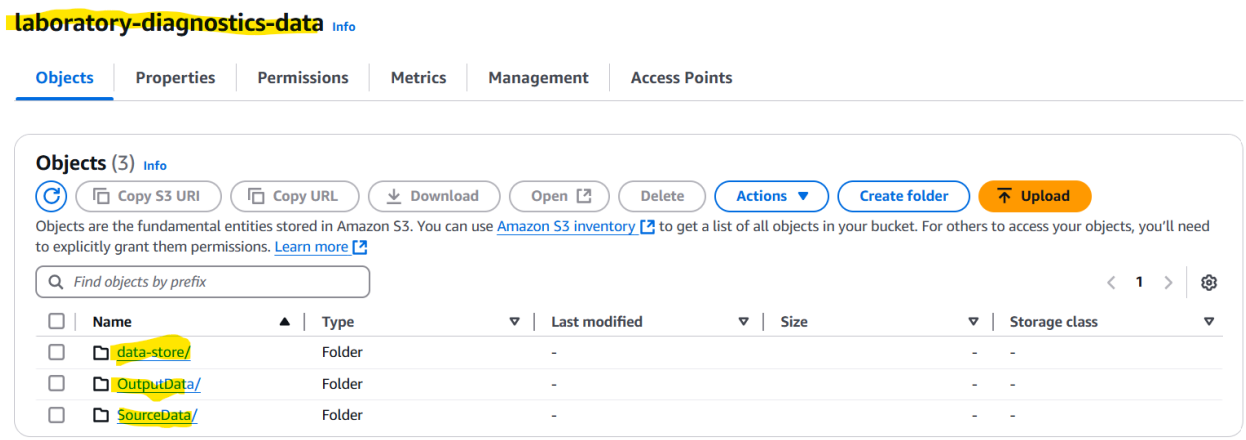
Column	**Data Type**	**Reference Key**
gender	String	
hypertension	Int	
heart_disease	Int	
smoking_history	String	
hbA1c_level	Float	
pregnancies	Int	
glucose	Float	
blood_pressure	Float	
skin_thickness	Float	
insulin	Float	
bmi	Float	
diabetes_pedigree_function	Float	
age	Int	
is_diabetic	**Int**	

AWS Components Architecture



Technical Implementation of AWS services

AWS S3 Bucket



SourceData/

Objects | Properties

Objects (7) Info

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

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, yo
to explicitly grant them permissions. Learn more

Find objects by prefix

Name	Type	Last modified	Size	Storage class
bloodtest/	Folder	-	-	-
diabeticprediction/	Folder	-	-	-
diagnosis/	Folder	-	-	-
generalbodytest/	Folder	-	-	-
hearttest/	Folder	-	-	-
patient/	Folder	-	-	-
patientlabtest/	Folder	-	-	-

bloodtest/

Copy S3 URI

Objects | Properties

Objects (1) Info

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

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. Learn more

Find objects by prefix

Name	Type	Last modified	Size	Storage class
blood_test.csv	csv	November 27, 2024, 13:24:40 (UTC+05:30)	436.0 B	Standard

AWS RDS

RDS > Databases

Amazon RDS

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Easy path homogeneous data migrations from EC2 database to RDS
With integrated homogenous data migration powered by AWS DMS, the Amazon RDS console leverages simple and performant data migration from EC2 database to equivalent RDS database. To get started, select an existing RDS database and choose the **Migrate data from EC2 database** in the **Actions** menu. Make sure you check the supported engine types and feature limitations. Learn more

Databases (2)

Group resources | Modify | Actions | Restore from S3 | Cre

Filter by databases

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Curren
database-1	Available	Regional cluster	Aurora MyS...	us-east-2	1 instance	1 Informational	-	-
database-1-instance-1	Available	Writer instance	Aurora MyS...	us-east-2b	db.r7g.large		4.93%	

RDS > Databases > database-1

Amazon RDS

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New

- Events
- Event subscriptions

Filter by databases

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU
database-1	Available	Regional cl...	Aurora My...	us-east-2	1 instance	1 Informational	-
database-1-instance-1	Available	Writer inst...	Aurora My...	us-east-2b	db.r7g.large		5.56%

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Data migrations - *new* | Tags

Endpoints (2)

Actions

Find resources

Endpoint name	Status	Type
database-1.cluster-c1musyekgsng.us-east-2.rds.amazonaws.com	Available	Writer
database-1.cluster-ro-c1musyekgsng.us-east-2.rds.amazonaws.com	Available	Reader

Manage IAM roles

Select IAM roles to add to this cluster
Add an existing IAM role to this cluster.

RDS > Databases > database-1 > database-1-instance-1

Amazon RDS

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New

- Events
- Event subscriptions

- Recommendations 1 1
- Certificate update

Filter by databases

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	C
database-1	Available	Regional cl...	Aurora My...	us-east-2	1 instance	1 Informational	-
database-1-instance-1	Available	Writer inst...	Aurora My...	us-east-2b	db.r7g.large		

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags | Recommendations

Connectivity & security

Endpoint & port

Endpoint
database-1-instance-1.c1musyekgsng.us-east-2.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-2b

VPC
vpc-096106e301d450aff

Subnet group
default-vpc-096106e301d450aff

Subnets
subnet-02ff71a055d318550
subnet-04963906eb6e9e3c6
subnet-0d328ec196ea61a33

Network type
IPv4

Security

VPC security groups
default (sg-074668185e7c686bd)
Active

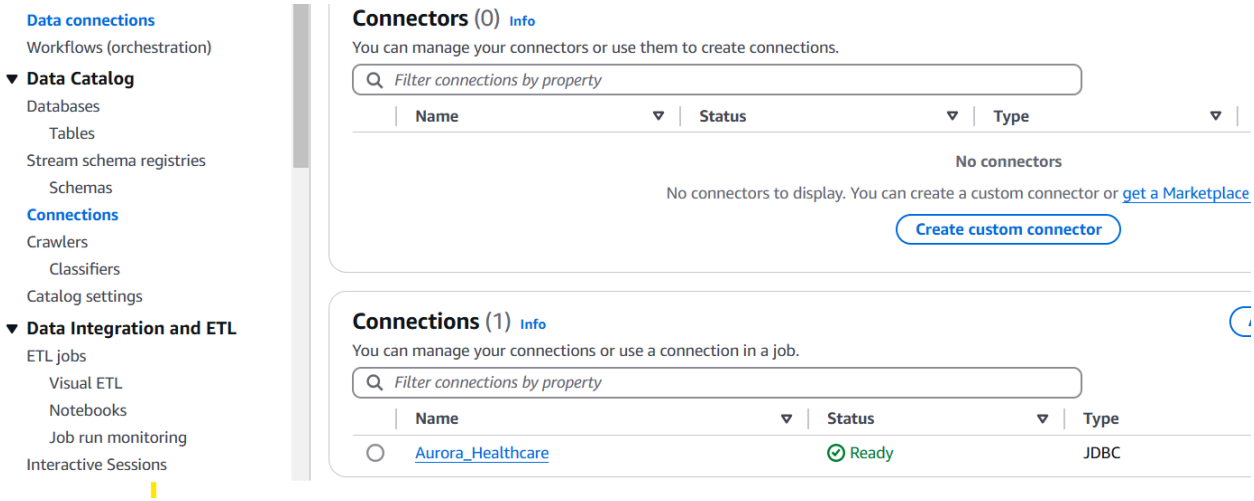
Publicly accessible
Yes

Certificate authority Info
rds-ca-rsa2048-g1

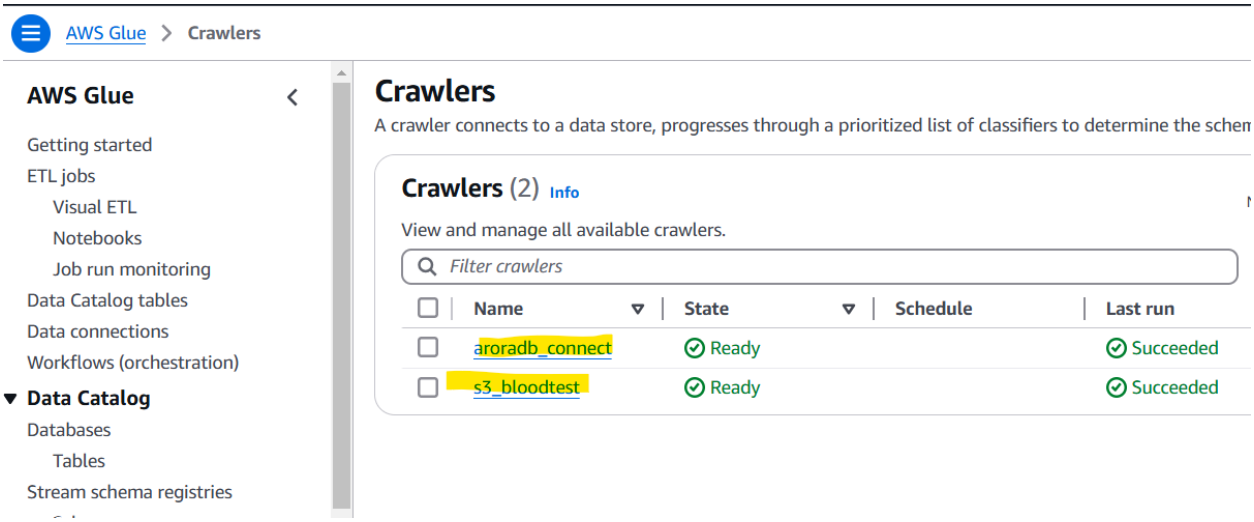
Certificate authority date
May 22, 2061, 05:34 (UTC+05:30)

DB instance certificate expiration date
November 27, 2025, 13:23 (UTC+05:30)

AWS Glue – Data Connections (Catalog)



AWS Glue - Crawlers



AWS Glue > Crawlers > s3_bloodtest

AWS Glue

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- ▼ Data Catalog
- Databases
  - Tables
- Stream schema registries
  - Schemas
- Connections
- Crawlers
  - Classifiers
- Catalog settings
- ▼ Data Integration and ETL
- ETL jobs

s3_bloodtest

Last updated (UTC)
November 29, 2024 at 10:20:45

Crawler properties

Name	IAM role	Database
s3_bloodtest	GlueRole	healthcare
Description	**Security configuration**	**Lake Formation configuration**
-	-	-
Maximum table threshold		
-		

▶ Advanced settings

Crawler runs | Schedule | Data sources | Classifiers | Tags

Data sources (1) Info

The list of data sources to be scanned by the crawler.

Edit

Type	Data source	Parameters
S3	s3://laboratory-diagnostics-data/SourceData/	Recrawl all

AWS Glue > Crawlers > aroradb_connect

AWS Glue

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- ▼ Data Catalog
- Databases
  - Tables
- Stream schema registries
  - Schemas
- Connections
- Crawlers
  - Classifiers
- Catalog settings

aroradb_connect

Last updated (UTC)
November 29, 2024 at 10:24:13

Crawler properties

Name	IAM role	Database	S
aroradb_connect	GlueRole	aroradb	R
Description	**Security configuration**	**Table prefix**	
-	-	-	

▶ Advanced settings

Crawler runs | Schedule | Data sources | Classifiers | Tags

Data sources (1) Info

The list of data sources to be scanned by the crawler.

Edit

Type	Data source	Parameters
JDBC	healthcare/%	-

AWS Glue – Databases

AWS Glue > Databases

AWS Glue

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- ▼ Data Catalog
- Databases
  - Tables
- Stream schema registries

Databases (3)

Last updated
November 29, 2024 at 10:

A database is a set of associated table definitions, organized into a logical group.

Filter databases

Name	Description	Location URI
aroradb	-	-
default	Default Hive database	file:/tmp/spark-warehouse
healthcare	This database is for healthcare analysis	-

AWS Glue

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- **Data Catalog**
  - Databases
  - Tables
- Stream schema registries
  - Schemas
- Connections
- Crawlers
  - Classifiers
- Catalog settings
- **Data Integration and ETL**
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Interactive Sessions

Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more

healthcare

Novembe

Database properties

Name	Description	Location
healthcare	This database is for healthcare analysis	-

Tables (7)

Last updated (UTC)
November 29, 2024 at 10:29:03

Delete

View and manage all available tables.

Filter tables

Name	Database	Location	Classification	Deprecated	View data
bloodtest	healthcare	s3://laboratory-diag	CSV	-	Table data
diabeticprediction	healthcare	s3://laboratory-diag	CSV	-	Table data
diagnosis	healthcare	s3://laboratory-diag	CSV	-	Table data
generalbodytest	healthcare	s3://laboratory-diag	CSV	-	Table data
hearttest	healthcare	s3://laboratory-diag	CSV	-	Table data
patient	healthcare	s3://laboratory-diag	CSV	-	Table data
patientlabtest	healthcare	s3://laboratory-diag	CSV	-	Table data

AWS Glue

- Getting started
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring
- Data Catalog tables
- Data connections
- Workflows (orchestration)
- **Data Catalog**
  - Databases
  - Tables
- Stream schema registries
  - Schemas
- Connections
- Crawlers
  - Classifiers
- Catalog settings
- **Data Integration and ETL**
- ETL jobs
  - Visual ETL
  - Notebooks
  - Job run monitoring

Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Le

aroradb

Database properties

Name	Description	Location
aroradb	-	-

Tables (7)

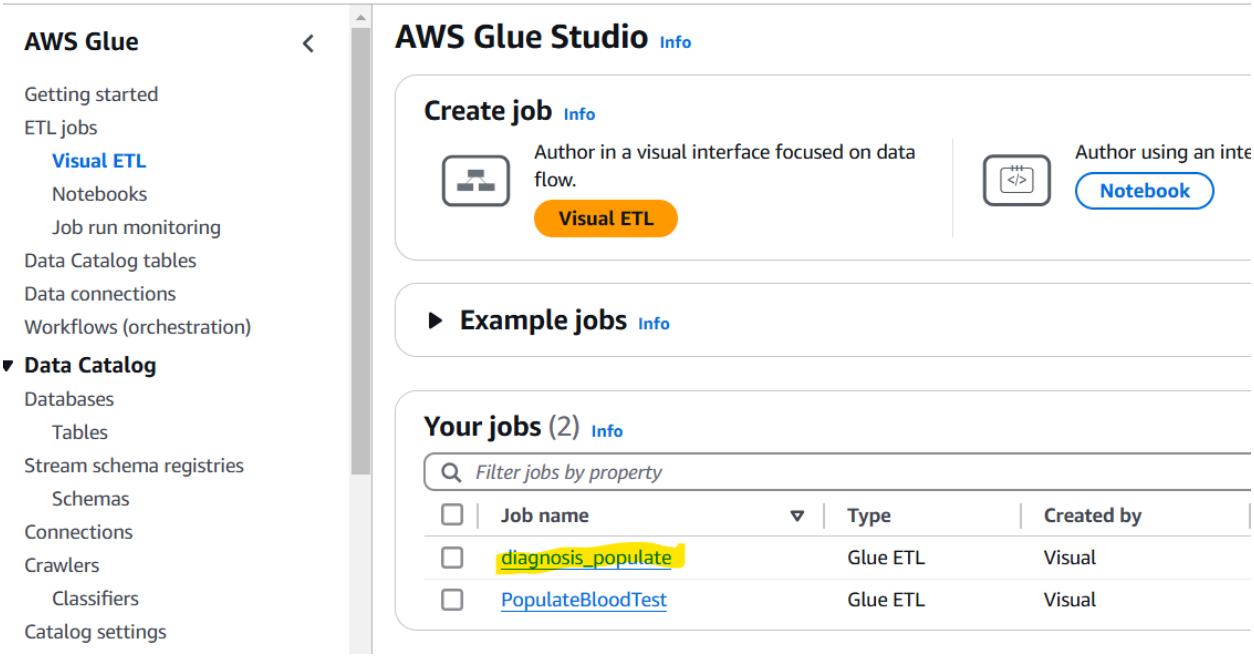
Last updat
November 29, 2024 at

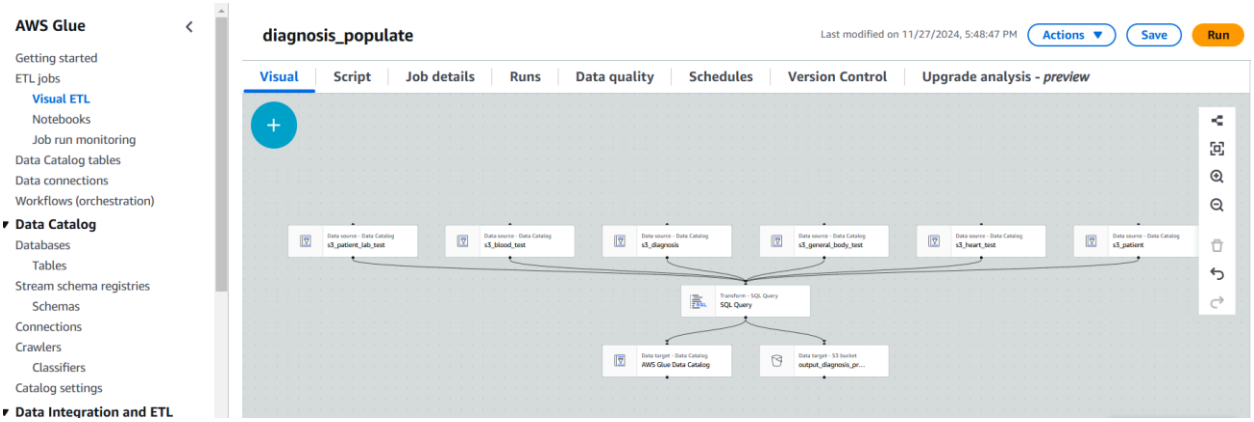
View and manage all available tables.

Filter tables

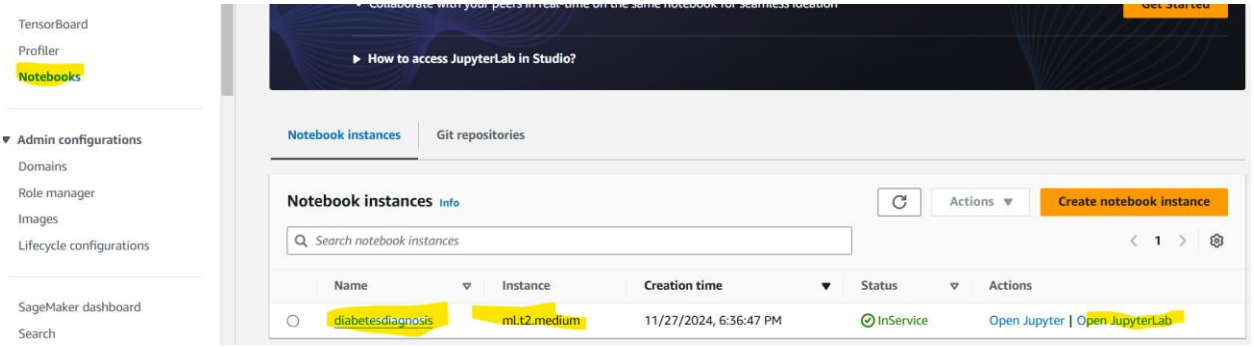
Name	Database	Location	Classification	Deprecate
healthcare_blood_te	aroradb	healthcare.blood_te	mysql	-
healthcare_diabetes	aroradb	healthcare.diabetes_	mysql	-
healthcare_diagnosi	aroradb	healthcare.diagnosis	mysql	-
healthcare_general_	aroradb	healthcare.general_	mysql	-
healthcare_heart_te	aroradb	healthcare.heart_te	mysql	-
healthcare_patient	aroradb	healthcare.Patient	mysql	-
healthcare_patient_	aroradb	healthcare.patient_l	mysql	-

AWS Glue - ETL

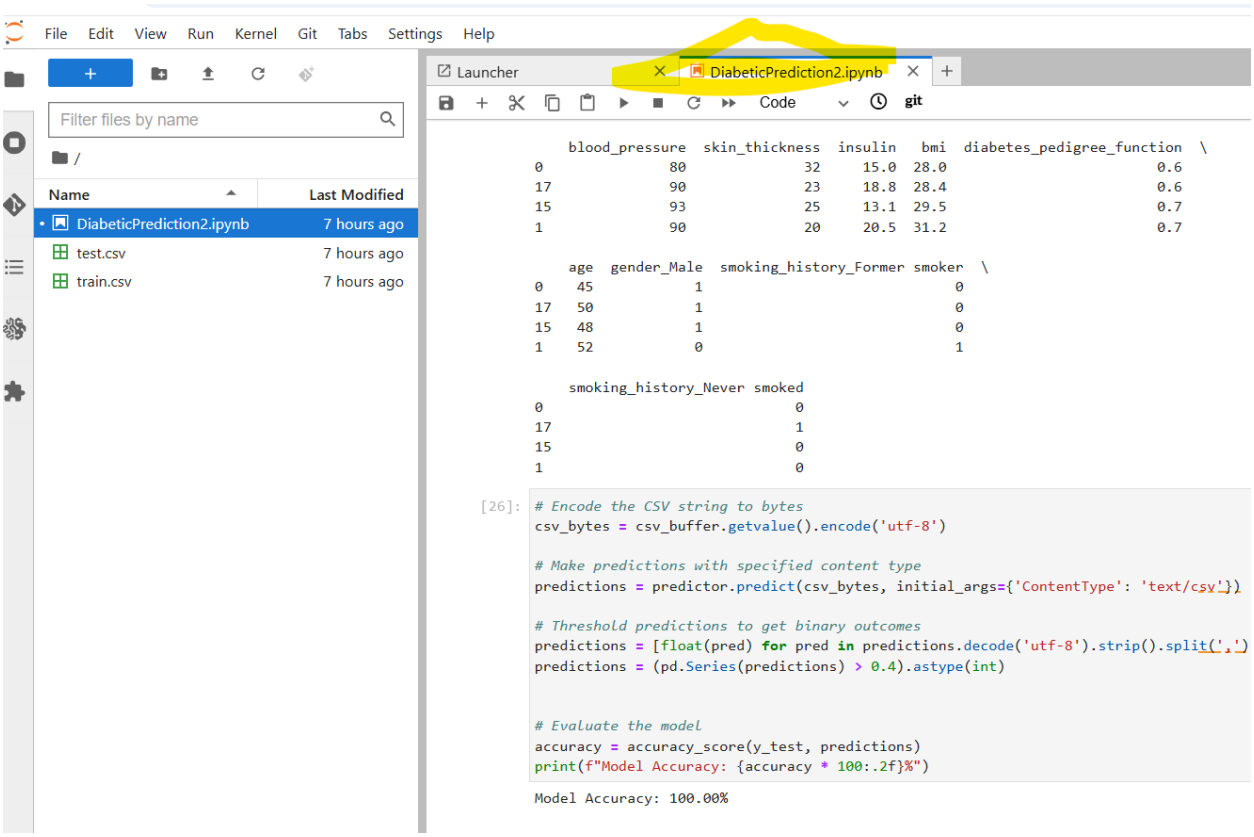




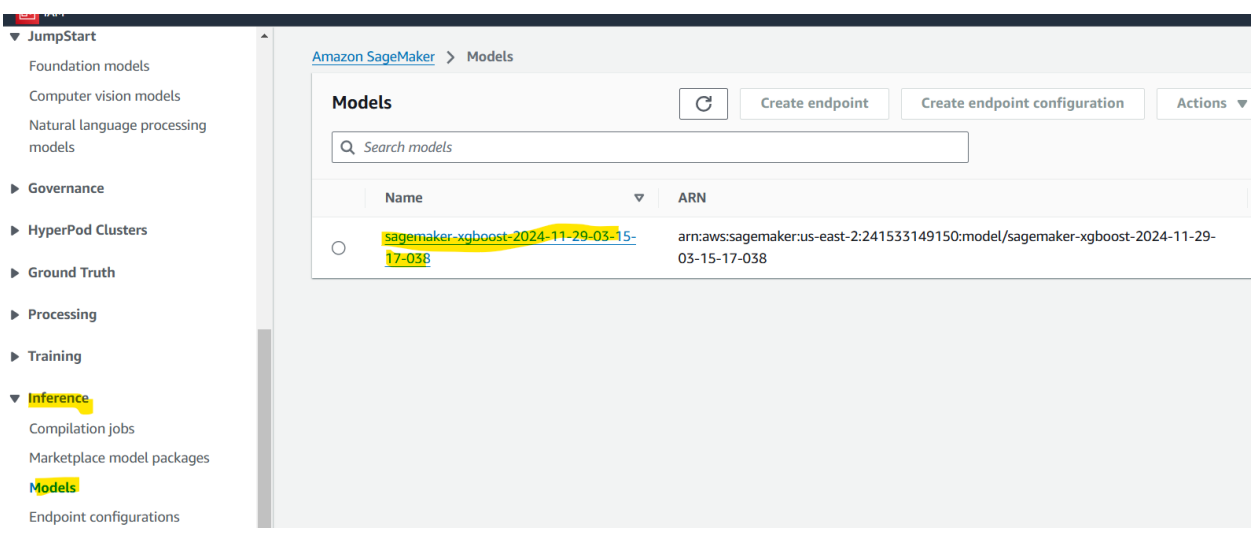
Amazon Sagemaker - Notebook



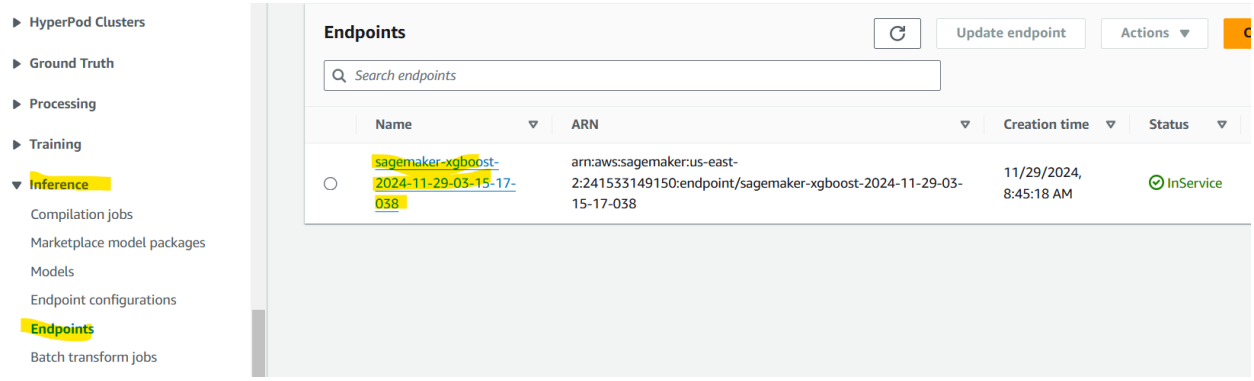
Amazon Sagemaker – DiabeticPrediction notebook



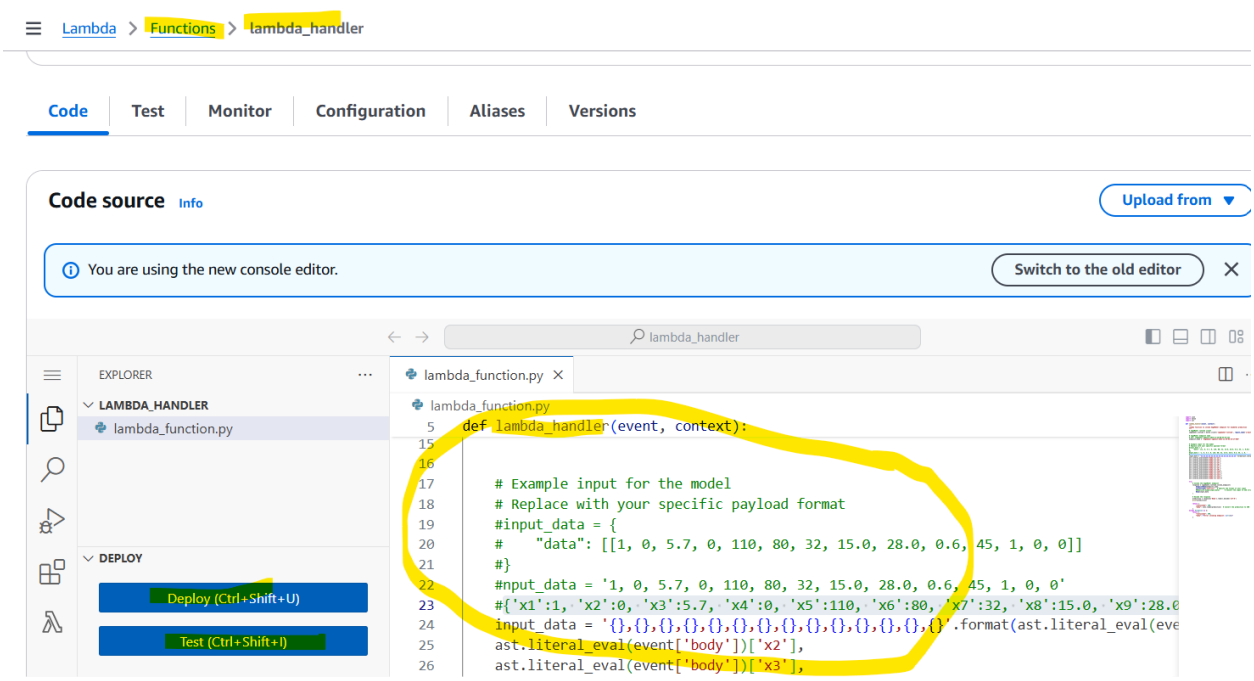
Amazon Sagemaker – Models Deployment



Amazon Sagemaker – Models Endpoints



AWS Lambda



AWS - API Gateway

API Gateway > APIs

API Gateway

APIs
Custom domain names Updated
Domain name access associations New
VPC links

Usage plans
API keys
Client certificates

APIs (1/1)

Find APIs

Name	Description	ID	Protocol	API endpoint type
diabetic-prediction-api		wqtvfbwi19	REST	Regional

AWS - API Gateway (Stages)

API Gateway > APIs > diabetic-prediction-api (wqtvfbwi19) > Stages

API Gateway

APIs
Custom domain names Updated
Domain name access associations New
VPC links

API: diabetic-prediction-api
Resources
Stages
Authorizers
Gateway responses
Models
Resource policy
Documentation
Dashboard
API settings

Stages

Stage actions

Dev

Stage details Info

Stage name
Dev

Rate Info
-

Web ACL
-

Cache cluster Info
Inactive

Burst Info
-

Client certificate
-

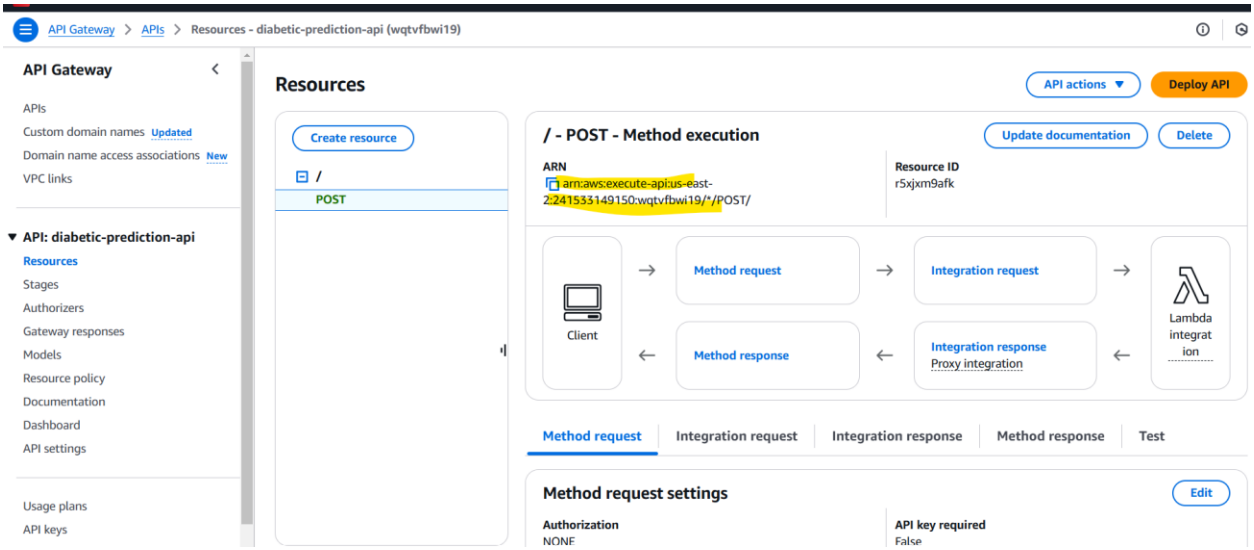
Default method-level caching
Inactive

Invoke URL
https://wqtvfbwi19.execute-api.us-east-2.amazonaws.com/Dev

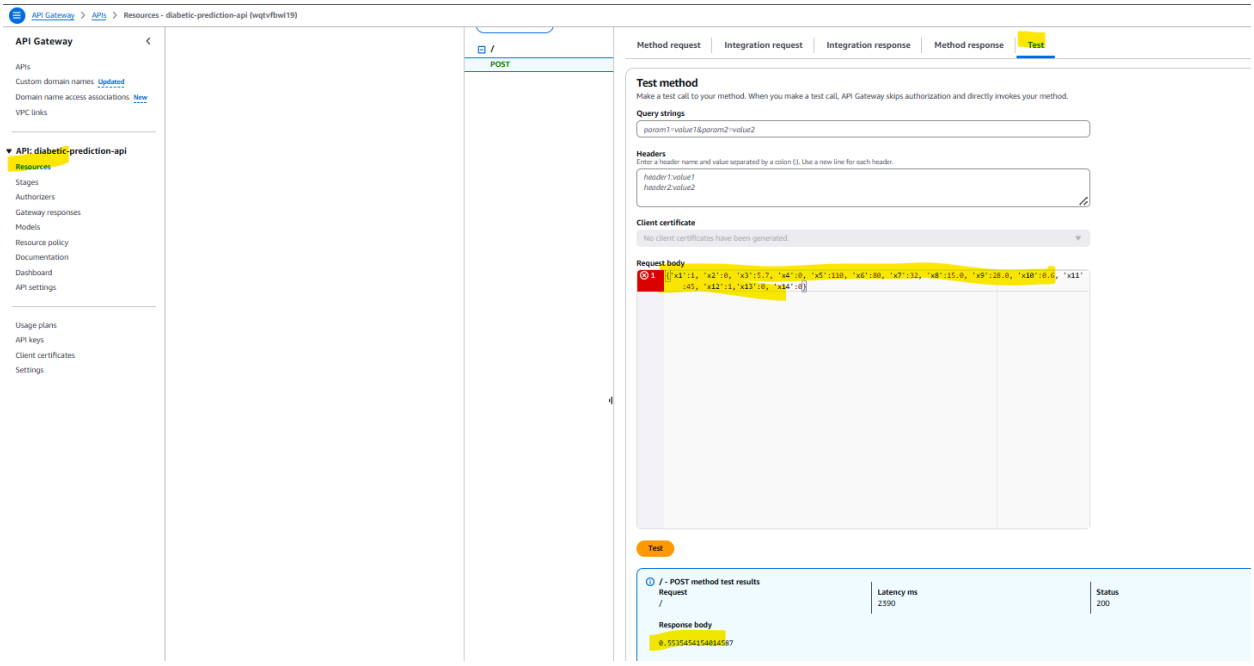
Active deployment
siccou on November 28, 2024, 17:49 (UTC+05:30)

Logs and tracing Info

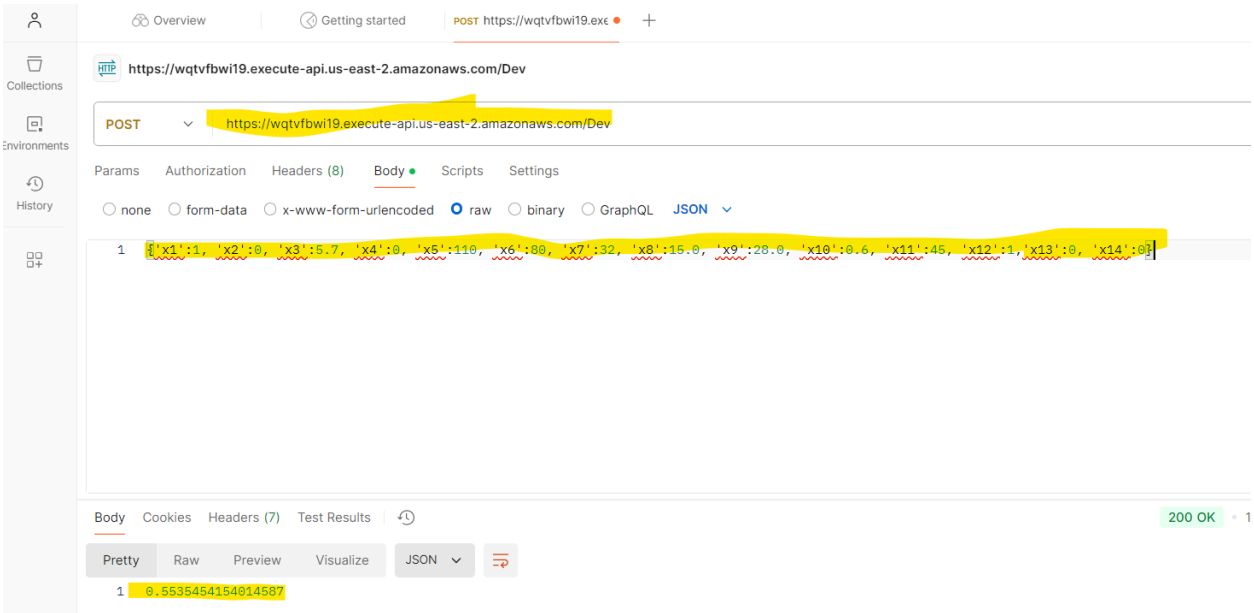
AWS - API Gateway (Resources)



AWS - API Gateway (API Testing)



Postman – API Testing



Additional Features Integration

- **AWS Encryption:** Implement server-side encryption for Amazon S3 buckets to ensure data security at rest. Extend encryption practices to other AWS services, including RDS Aurora and Glue to maintain end-to-end data protection.
- **API Gateway Endpoint Security:** Enforce security measures such as API keys, IAM security against unauthorize access and ensure robust API performance.
- **Minimal Policy Access Rights:** Design IAM roles and policies adhering to the principle of least privilege, granting only the required permissions for each service to ensure enhanced security and compliance.
- **Git Integration:** Integrate with Git for version control, enabling efficient collaboration and tracking of code changes.

Implementation Resources

Git Repository	https://github.com/kapsenitin1/ML-Capabilites_AWS.git
Github	https://github.com/kapsenitin1/ML-Capabilites_AWS