



BITS Pilani
Hyderabad Campus

Cloud Computing (CS G527)

A report on

MINI PROJECT

Group 11

Prepared by :

| Name | BITS ID |
|-------------------|---------------|
| Dhruti Dobariya | 2024H1030052H |
| Pranathi Voora | 2022A7PS2012H |
| Maedhika Malhotra | 2021B5A82006H |

Instructor
Prof. Shubhrakanta Panda

MINI PROJECT

BRAINSTORMING

- **healthcare chatbot**, add healthcare data retrieval of old patient data and receive medical advice based on history as well
- **interview chatbot**, get data from people and assess based on how chatbot needs it to be assessed. Data can be added by interviewees and chatbot can be used by interviewer. Interviewee can also query to see where they stand and improvements?

Chatbot for Personalized AWS Cost Management

AWS Services Used: DynamoDB + CloudWatch + Lex

Why DynamoDB?

- Stores user-specific cost data for instant retrieval.
- Tracks historical spending trends per service/account.
- Works with CloudWatch to set alerts for unexpected cost spikes.

Example Chatbot Functions:

- “How much did I spend on EC2 this month?” → Queries cost data.
- “Alert me if my AWS bill exceeds \$100.” → Uses CloudWatch Alarms.
- “What services are costing me the most?” → Fetches top cost contributors.

how about course selection ?upload interests, marksheets, etc

See idea from streaming data

Detect number of phones in an outlet

Exam Attendance Fingerprint Chatbot

The problem is that exam attendance must be monitored very closely and leaving and entering the exam hall is an issue. Times are not noted precisely. This can be monitored easily using this system. Professors can also see how much time the average student is spending in the exam hall and receive analytics on the students. It is a conceptual model that builds on the desire for transparency in the exam hall and professor's knowledge.

Getting Fingerprint Data

Create third party API to receive fingerprint data

Fingerprint data is stored in the BITS system in the form of .tif

But as this data is private, we are using fingerprint data (8 images each of 110 fingerprints) from this dataset <http://bias.csr.unibo.it/fvc2000/download.asp>. (DB1_B.zip)

The dataset structure is:

- 100_8 where 100 is person, 8 is 8th fingerprint

Here's an example of the structure:

| DB1_B | |
|-----------|-------|
| 101_1.tif | 91 KB |
| 101_2.tif | 91 KB |
| 101_3.tif | 91 KB |
| 101_4.tif | 91 KB |
| 101_5.tif | 91 KB |
| 101_6.tif | 91 KB |
| 101_7.tif | 91 KB |
| 101_8.tif | 91 KB |
| 102_1.tif | 91 KB |
| 102_2.tif | 91 KB |
| 102_3.tif | 91 KB |
| 102_4.tif | 91 KB |
| 102_5.tif | 91 KB |
| 102_6.tif | 91 KB |
| 102_7.tif | 91 KB |
| 102_8.tif | 91 KB |

Here's a fingerprint example:



Store Fingerprint Data

Store every user fingerprint in **S3 bucket** and implement **lifecycle policies**. As students pass out, fingerprint data gets removed and switched to another lifecycle policy. Keep the links to each of these fingerprints in order to link to the RDBMS which will store all the user data. why s3 bucket and not direct RDBMS

PLANNING AHEAD: Assessing for the future:

How should I store the items?

1. Dump the items in s3?
2. store items as folders. Multiple fingerprints per person or access each fingerprint

While Amazon S3 doesn't have a true folder structure, you can organize objects within a bucket by using "folders" as a visual representation through object keys with forward slashes (/) to mimic a hierarchical structure, according to AWS.



We will use this method to establish a hierarchy so that each person's folder has multiple fingerprints to be matched inside. Once there is a matched fingerprint the parent folder is returned.

Here's how we implemented this in AWS S3 BUCKET STORAGE:

Handling Incoming Fingerprints

Streaming data: Data is incoming for fingerprints from the fingerprint reader. Every time a user scans their fingerprint on the reader, it must be matched with the database and register the user as valid. We can handle streaming data (live data) with Amazon Kinesis and data warehouse. why

Fortunately, this system is pre implemented by BITS fingerprint recognition. So, once there is a positive recognition, we must update the user entry and exit timing in the database.

For the sake of the project, where we do not have access to BITS fingerprint data, we will implement the handling of incoming streaming fingerprint data.

FIND A METHOD THAT IS EASY



Data

Match Fingerprint Data

Amazon Rekognition + AWS Lambda

Match data and trigger the backend accordingly. Amazon Rekognition allows you to compare fingerprint images.

AWS Lambda: For custom logic, API calls, and dynamic responses.

Search procedure:

User enters a fingerprint. If the fingerprint matches, then continue. Match is defined by searching Using Amazon recognition.

If not match display error.

Search procedure in code for AWS lambda:

Store User Data

Store all USER data in **RDBMS**

why RDBMS over dynamo db

- Can query with SQL
- Supports complex relationships
- Good to find precise extracted data answers to send to LLM
- Changing data

Store S3 object URL for user fingerprint and all user metadata:

```
CREATE TABLE Users (
    StudentID VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(100),
    BatchNumber INT,
    Branch VARCHAR(100),
    FingerprintURL TEXT
);
```



```
CREATE TABLE Professor (
    ProfessorID VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(100)
);
```

Store Attendance Data

Store all attendance data in **RDBMS**

```
CREATE TABLE Course (
    CourseID VARCHAR(20) PRIMARY KEY,
    CourseName VARCHAR(100),
    ProfessorID VARCHAR(20)
);
```

```
CREATE TABLE Exam (
    ExamID VARCHAR(20) PRIMARY KEY,
    StartTime DATETIME,
    EndTime DATETIME,
    CourseID VARCHAR(20),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);
```

```
CREATE TABLE ExamAttendance (
    EntryID VARCHAR(36) PRIMARY KEY,
    CourseID VARCHAR(20),
    StudentID VARCHAR(20),
    StudentInTime DATETIME,
    StudentOutTime DATETIME,
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
    FOREIGN KEY (StudentID) REFERENCES Users(StudentID)
);
```

```
CREATE TABLE CourseRegistrations (
    RegistrationID VARCHAR(36) PRIMARY KEY,
    CourseID VARCHAR(20),
    StudentID VARCHAR(20),
```



FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
 FOREIGN KEY (StudentID) REFERENCES Users(StudentID)
);

Implementation

Using the RDBMS- Amazon RDS + MySQL

Creating the Database with Standard Create and MySQL engine options.

The screenshot shows the 'Create database' wizard in the AWS RDS console. The top navigation bar includes the AWS logo, search bar, and user information (Europe (Stockholm) and Pranathi Voora). The main page title is 'Create database' with an 'Info' link. Below it, the heading 'Choose a database creation method' has two options: 'Standard create' (selected) and 'Easy create'. The 'MySQL' section on the right describes MySQL as the most popular open source database and lists its features. The 'Engine options' section shows various engine types with icons: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), PostgreSQL, MariaDB, and Oracle. At the bottom, there are links for CloudShell, Feedback, and footer links for Privacy, Terms, and Cookie preferences.

Create database [Info](#)

Choose a database creation method

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

| | |
|---|--|
| <input type="radio"/> Aurora (MySQL Compatible) | <input type="radio"/> Aurora (PostgreSQL Compatible) |
| | |
| <input checked="" type="radio"/> MySQL | <input type="radio"/> PostgreSQL |
| | |
| <input type="radio"/> MariaDB | <input type="radio"/> Oracle |
| | |

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Set the engine version to the default version and press Free Tier for the sake of this project.

Engine version [Info](#)
View the engine versions that support the following database features.

▼ Hide filters

Show only versions that support the Multi-AZ DB cluster [Info](#)
Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show only versions that support the Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version

Enable RDS Extended Support [Info](#)
Amazon RDS Extended Support is a [paid offering](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

MySQL >

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Templates

Choose a sample template to meet your use case.

Production

Use defaults for high availability and fast, consistent performance.

Dev/Test

This instance is intended for development use outside of a production environment.

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

Add a name for the database and select a master username. Then create a master password.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Password strength Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ * @

Confirm master password [Info](#)

Choose this as the instance configuration.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class | [Info](#)

▼ Hide filters

Show instance classes that support Amazon RDS Optimized Writes

[Info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Include previous generation classes

Standard classes (includes m classes)

Memory optimized classes (includes r and x classes)

Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

Choose general purpose default storage (20 gb), the minimum amount of storage

Storage

Storage type [Info](#)

Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage [Info](#)

20

GiB

Allocated storage value must be 20 GiB to 6,144 GiB

► Additional storage configuration

Set the public access to “Yes”

Public access [Info](#)

Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group: create a new one to allow sql

Creating a new VPC security group:

In ec2, go to security groups. Click create security group

The screenshot shows the AWS EC2 console with the 'Security Groups' page. On the left, there's a navigation sidebar with sections like 'Elastic Block Store', 'Network & Security' (which is expanded to show 'Security Groups'), 'Load Balancing', and 'Auto Scaling'. The main area displays a table of security groups. One row is selected, showing 'sg-0c51e8e08acb73c16' as the 'Name' and 'Security group ID', 'default' as the 'Security group name', and 'vpc-00af74bda21a17d17' as the 'VPC ID'. A large button at the bottom right says 'Create security group'.

Add a security group name and select the same vpc as the rds.

The screenshot shows the 'Create security group' wizard. In the 'Basic details' step, the 'Security group name' is set to 'miniproject', 'Description' is 'sql functions, db', and the 'VPC' dropdown is set to 'vpc-00af74bda21a17d17'. A note at the top says: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.'

Add and select the following inbound rules:

add the type as mysql and the source as 0.0.0.0/0 so that anyone can access it on the internet for the sake of the project.

The screenshot shows the 'Inbound rules' configuration screen. It has fields for 'Type' (set to 'MySQL/Aurora'), 'Protocol' (set to 'TCP'), 'Port range' (set to '3306'), 'Source' (set to 'Any...'), and 'Description - optional' (left empty). Below these, a table lists a single rule: '0.0.0.0/0' with a delete button. At the bottom, there's a 'Delete' button and a 'Add rule' button.

Also select anywhere for outbound rules.

Outbound rules [Info](#)

| Type | Info | Protocol | Info | Port range | Info | Destination | Info | Description - optional | Info |
|-------------|----------------------|----------|----------------------|--------------------------|------------------------|-----------------------------|------------------------|------------------------|----------------------|
| All traffic | ▼ | All | All | Any... ▼ | Search | 0.0.0.0/0 X | Delete | | |

[Add rule](#)

⚠️ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses. [X](#)

Now go back to the rds and choose the security group just made.

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups

Choose one or more options [▼](#)

miniproject [X](#)

Certificate authority - optional [Info](#)

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 25, 2061 [▼](#)

If you don't select a certificate authority, RDS chooses one for you.

Now press create database

Aurora and RDS > Databases

Aurora and 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

Creating database attendance-db
Your database might take a few minutes to launch. You can use settings from attendance-db to simplify configuration of suggested database add-ons while we finish creating your DB for you.

[View credential details](#) [X](#)

Databases (1)

Notifications [0 0](#) [Δ 0](#) [0 0](#) [0 4](#) [0 1](#) [▼](#)

Group resources [C](#) [Modify](#) [Actions](#) [▼](#) [Create database](#) [▼](#)

| DB identifier | Status | Role | Engine | Region ... | Size | Recommen |
|---------------|--------------------------|----------|-------------|------------|-------------|----------|
| attendance-db | Creating | Instance | MySQL Co... | - | db.t3.micro | |

Wait for the database to be created and has the tag "Available"

The screenshot shows the AWS RDS console under the Aurora and RDS section. A modal window titled 'Creating database attendance-proj' is open, stating 'Your database might take a few minutes to launch. You can use settings from attendance-proj to simplify configuration of suggested database add-ons while we finish creating your DB for you.' Below the modal, the 'Databases (2)' list is visible, showing one entry: 'attendance-db' which is 'Available', an 'Instance', running 'MySQL Co...', in 'eu-north-1b' region, and 'db.t3.micro' class. There are buttons for 'View credential details', 'Notifications', 'Group resources', 'Modify', 'Actions', and 'Create database'.

Populating the database

Using MySQL workbench to populate the database

```
• USE attendance;

• CREATE TABLE Users (
    StudentID VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(100),
    BatchNumber INT,
    Branch VARCHAR(100),
    FingerprintURL TEXT
);

• CREATE TABLE Professor (
    ProfessorID VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(100)
);
```

```
16
17 • ⊖ CREATE TABLE Course (
18     CourseID VARCHAR(20) PRIMARY KEY,
19     CourseName VARCHAR(100),
20     ProfessorID VARCHAR(20)
21 );
22
23 • ⊖ CREATE TABLE Exam (
24     ExamID VARCHAR(20) PRIMARY KEY,
25     StartTime DATETIME,
26     EndTime DATETIME,
27     CourseID VARCHAR(20),
28     FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
29 );
30
31
32 • ⊖ CREATE TABLE ExamAttendance (
33     EntryID VARCHAR(36) PRIMARY KEY,
34     CourseID VARCHAR(20),
35     StudentID VARCHAR(20),
36     StudentInTime DATETIME,
37     StudentOutTime DATETIME,
38     FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
39     FOREIGN KEY (StudentID) REFERENCES Users(StudentID)
40 );
41
```

```
• ⊖ CREATE TABLE CourseRegistrations (
    RegistrationID VARCHAR(36) PRIMARY KEY,
    CourseID VARCHAR(20),
    StudentID VARCHAR(20),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
    FOREIGN KEY (StudentID) REFERENCES Users(StudentID)
);
```

Populating the DB for usage:

```

1
2 • USE attendance;
3
4 -- Insert 5 Professors
5 • INSERT INTO Professor (ProfessorID, Name) VALUES
6   ('P001', 'Dr. Smith'),
7   ('P002', 'Dr. Johnson'),
8   ('P003', 'Dr. Williams'),
9   ('P004', 'Dr. Brown'),
10  ('P005', 'Dr. Davis');
11
12 -- Insert 4 Courses
13 • INSERT INTO Course (CourseID, CourseName, ProfessorID) VALUES
14   ('C001', 'Database Systems', 'P001'),
15   ('C002', 'Algorithms', 'P002'),
16   ('C003', 'Operating Systems', 'P003'),
17   ('C004', 'Computer Networks', 'P004');
18
19 -- Insert 20 Students
20 • INSERT INTO Users (StudentID, Name, BatchNumber, Branch, FingerprintURL) VALUES
21   ('S001', 'John Doe', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
22   ('S002', 'Jane Smith', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
23   ('S003', 'Robert Johnson', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
24   ('S004', 'Emily Williams', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
25   ('S005', 'Michael Brown', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
26   ('S006', 'Sarah Davis', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
27   ('S007', 'David Miller', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
28   ('S008', 'Jessica Wilson', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
29   ('S009', 'Thomas Moore', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
30   ('S010', 'Lisa Taylor', 2022, 'Computer Science', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
31   ('S011', 'Daniel Anderson', 2023, 'Information Technology', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
32   ('S012', 'Nancy Thomas', 2023, 'Information Technology', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
33   ('S013', 'Kevin Jackson', 2023, 'Information Technology', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
34   ('S014', 'Amy White', 2023, 'Information Technology', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001'),
35   ('S015', 'Paul Harris', 2023, 'Information Technology', 'https://student-fingerprints.s3.eu-north-1.amazonaws.com/S001');
36
37 -- Insert 4 Exams
38 • INSERT INTO Exam (ExamID, StartTime, EndTime, CourseID) VALUES
39   ('E001', '2023-06-10 09:00:00', '2023-06-10 12:00:00', 'C001'),
40   ('E002', '2023-06-12 14:00:00', '2023-06-12 17:00:00', 'C002'),
41   ('E003', '2023-06-15 10:00:00', '2023-06-15 13:00:00', 'C003'),
42   ('E004', '2023-06-17 09:00:00', '2023-06-17 11:30:00', 'C004');
43
44 -- Course Registrations (all students registered for all courses)
45 • INSERT INTO CourseRegistrations (RegistrationID, CourseID, StudentID) VALUES
46   -- Course C001 (Database Systems) - 18 students registered
47   ('R001', 'C001', 'S001'),
48   ('R002', 'C001', 'S002'),
49   ('R003', 'C001', 'S003'),
50   ('R004', 'C001', 'S004'),
51   ('R005', 'C001', 'S005'),
52   ('R006', 'C001', 'S006'),
53   ('R007', 'C001', 'S007'),
54   ('R008', 'C001', 'S008'),
55   ('R009', 'C001', 'S009'),
56   ('R010', 'C001', 'S010'),
57   ('R011', 'C001', 'S011'),
58   ('R012', 'C001', 'S012'),
59   ('R013', 'C001', 'S013'),
60   ('R014', 'C001', 'S014'),
61   ('R015', 'C001', 'S015'),
62   ('R016', 'C001', 'S016');
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Auto
con
hel
disa
Use
toolb
man
get
for
cur
ca
posit
to t
auto
he

```

-- Course C002 (Algorithms) - 19 students registered
('R019', 'C002', 'S001'),
('R020', 'C002', 'S002'),
('R021', 'C002', 'S003'),
('R022', 'C002', 'S004'),
('R023', 'C002', 'S005'),
('R024', 'C002', 'S006'),
('R025', 'C002', 'S007'),
('R026', 'C002', 'S008'),
('R027', 'C002', 'S009'),
('R028', 'C002', 'S010'),
('R029', 'C002', 'S011'),
('R030', 'C002', 'S012'),
('R031', 'C002', 'S013'),
('R032', 'C002', 'S014'),
('R033', 'C002', 'S015'),
('R034', 'C002', 'S016'),
('R035', 'C002', 'S017'),
('R036', 'C002', 'S018'),
('R037', 'C002', 'S019'),
-- S020 not registered for C002

-- Course C003 (Operating Systems) - 17 students registered
('R038', 'C003', 'S001'),
('R039', 'C003', 'S002'),
('R040', 'C003', 'S003'),
('R041', 'C003', 'S004'),
('R042', 'C003', 'S005'),
('R043', 'C003', 'S006'),
('R044', 'C003', 'S007'),
('R045', 'C003', 'S008'),
('R046', 'C003', 'S009'),
('R047', 'C003', 'S010'),
-- R048, R049, R050 not registered for C003

```

```

113
114 -- Course C004 (Computer Networks) - 16 students registered
115 ('R055', 'C004', 'S001'),
116 ('R056', 'C004', 'S002'),
117 ('R057', 'C004', 'S003'),
118 ('R058', 'C004', 'S004'),
119 ('R059', 'C004', 'S005'),
120 ('R060', 'C004', 'S006'),
121 ('R061', 'C004', 'S007'),
122 ('R062', 'C004', 'S008'),
123 ('R063', 'C004', 'S009'),
124 ('R064', 'C004', 'S010'),
125 ('R065', 'C004', 'S011'),
126 ('R066', 'C004', 'S012'),
127 ('R067', 'C004', 'S013'),
128 ('R068', 'C004', 'S014'),
129 ('R069', 'C004', 'S015'),
130 ('R070', 'C004', 'S016');
131 -- S017, S018, S019, S020 not registered for C004
132
133 -- Exam Attendance (minimum 15 students per exam, some registered students don't attend)
134 • INSERT INTO ExamAttendance (EntryID, CourseID, StudentID, StudentInTime, StudentOutTime) VALUES
135 -- Exam E001 (Database Systems) - 15 students attended (3 registered didn't attend)
136 ('EA001', 'C001', 'S001', '2023-06-10 09:05:12', '2023-06-10 11:45:23'),
137 ('EA002', 'C001', 'S002', '2023-06-10 09:07:34', '2023-06-10 11:50:12'),
138 ('EA003', 'C001', 'S003', '2023-06-10 09:10:45', '2023-06-10 11:55:34'),
139 ('EA004', 'C001', 'S004', '2023-06-10 09:12:56', '2023-06-10 11:40:45'),
140 ('EA005', 'C001', 'S005', '2023-06-10 09:15:23', '2023-06-10 11:30:56'),
141 ('EA006', 'C001', 'S006', '2023-06-10 09:20:34', '2023-06-10 11:35:23'),
142 ('EA007', 'C001', 'S007', '2023-06-10 09:25:45', '2023-06-10 11:25:34'),
143 ('EA008', 'C001', 'S008', '2023-06-10 09:30:56', '2023-06-10 11:20:45'),
144 ('EA009', 'C001', 'S009', '2023-06-10 09:35:12', '2023-06-10 11:15:56'),
145 ('EA010', 'C001', 'S010', '2023-06-10 09:40:23', '2023-06-10 11:10:12'),
146 ('EA011', 'C001', 'S011', '2023-06-10 09:45:34', '2023-06-10 11:05:23'),
147 ('EA012', 'C001', 'S012', '2023-06-10 09:50:45', '2023-06-10 11:00:34')

```

```

52 -- Exam E002 (Algorithms) - 16 students attended (3 registered didn't attend)
53 ('EA016', 'C002', 'S001', '2023-06-12 14:02:34', '2023-06-12 16:45:23'),
54 ('EA017', 'C002', 'S002', '2023-06-12 14:05:45', '2023-06-12 16:50:34'),
55 ('EA018', 'C002', 'S003', '2023-06-12 14:08:56', '2023-06-12 16:55:45'),
56 ('EA019', 'C002', 'S004', '2023-06-12 14:12:12', '2023-06-12 16:40:56'),
57 ('EA020', 'C002', 'S005', '2023-06-12 14:15:23', '2023-06-12 16:35:12'),
58 ('EA021', 'C002', 'S006', '2023-06-12 14:18:34', '2023-06-12 16:30:23'),
59 ('EA022', 'C002', 'S007', '2023-06-12 14:21:45', '2023-06-12 16:25:34'),
60 ('EA023', 'C002', 'S008', '2023-06-12 14:25:56', '2023-06-12 16:20:45'),
61 ('EA024', 'C002', 'S009', '2023-06-12 14:30:12', '2023-06-12 16:15:56'),
62 ('EA025', 'C002', 'S010', '2023-06-12 14:35:23', '2023-06-12 16:10:12'),
63 ('EA026', 'C002', 'S011', '2023-06-12 14:40:34', '2023-06-12 16:05:23'),
64 ('EA027', 'C002', 'S012', '2023-06-12 14:45:45', '2023-06-12 16:00:34'),
65 ('EA028', 'C002', 'S013', '2023-06-12 14:50:56', '2023-06-12 15:55:45'),
66 ('EA029', 'C002', 'S014', '2023-06-12 14:55:12', '2023-06-12 15:50:56'),
67 ('EA030', 'C002', 'S015', '2023-06-12 15:00:23', '2023-06-12 15:45:12'),
68 ('EA031', 'C002', 'S016', '2023-06-12 15:05:34', '2023-06-12 15:40:23'),
69 -- S017, S018, S019 registered but didn't attend
70

72 -- Exam E003 (Operating Systems) - 15 students attended (2 registered didn't attend)
73 ('EA032', 'C003', 'S001', '2023-06-15 10:03:45', '2023-06-15 12:55:23'),
74 ('EA033', 'C003', 'S002', '2023-06-15 10:07:56', '2023-06-15 12:50:34'),
75 ('EA034', 'C003', 'S003', '2023-06-15 10:12:12', '2023-06-15 12:45:45'),
76 ('EA035', 'C003', 'S004', '2023-06-15 10:15:23', '2023-06-15 12:40:56'),
77 ('EA036', 'C003', 'S005', '2023-06-15 10:18:34', '2023-06-15 12:35:12'),
78 ('EA037', 'C003', 'S006', '2023-06-15 10:21:45', '2023-06-15 12:30:23'),
79 ('EA038', 'C003', 'S007', '2023-06-15 10:25:56', '2023-06-15 12:25:34'),
80 ('EA039', 'C003', 'S008', '2023-06-15 10:30:12', '2023-06-15 12:20:45'),
81 ('EA040', 'C003', 'S009', '2023-06-15 10:35:23', '2023-06-15 12:15:56'),
82 ('EA041', 'C003', 'S010', '2023-06-15 10:40:34', '2023-06-15 12:10:12'),
83 ('EA042', 'C003', 'S011', '2023-06-15 10:45:45', '2023-06-15 12:05:23'),
84 ('EA043', 'C003', 'S012', '2023-06-15 10:50:56', '2023-06-15 12:00:34'),
85 ('EA044', 'C003', 'S013', '2023-06-15 10:55:12', '2023-06-15 11:55:45'),
86 ('EA045', 'C003', 'S014', '2023-06-15 11:00:22', '2023-06-15 11:50:56')
87

89 ('EA046', 'C003', 'S013', '2023-06-15 11:05:34', '2023-06-15 11:45:12'),
90 -- S016, S017 registered but didn't attend

92

94 -- Exam E004 (Computer Networks) - 15 students attended (1 registered didn't attend)
95 ('EA047', 'C004', 'S001', '2023-06-17 09:04:56', '2023-06-17 11:25:23'),
96 ('EA048', 'C004', 'S002', '2023-06-17 09:08:12', '2023-06-17 11:20:34'),
97 ('EA049', 'C004', 'S003', '2023-06-17 09:12:23', '2023-06-17 11:15:45'),
98 ('EA050', 'C004', 'S004', '2023-06-17 09:16:34', '2023-06-17 11:10:56'),
99 ('EA051', 'C004', 'S005', '2023-06-17 09:20:45', '2023-06-17 11:05:12'),
100 ('EA052', 'C004', 'S006', '2023-06-17 09:25:56', '2023-06-17 11:00:23'),
101 ('EA053', 'C004', 'S007', '2023-06-17 09:30:12', '2023-06-17 10:55:34'),
102 ('EA054', 'C004', 'S008', '2023-06-17 09:35:23', '2023-06-17 10:50:45'),
103 ('EA055', 'C004', 'S009', '2023-06-17 09:40:34', '2023-06-17 10:45:56'),
104 ('EA056', 'C004', 'S010', '2023-06-17 09:45:45', '2023-06-17 10:40:12'),
105 ('EA057', 'C004', 'S011', '2023-06-17 09:50:56', '2023-06-17 10:35:23'),
106 ('EA058', 'C004', 'S012', '2023-06-17 09:55:12', '2023-06-17 10:30:34'),
107 ('EA059', 'C004', 'S013', '2023-06-17 10:00:23', '2023-06-17 10:25:45'),
108 ('EA060', 'C004', 'S014', '2023-06-17 10:05:34', '2023-06-17 10:20:56'),
109 ('EA061', 'C004', 'S015', '2023-06-17 10:10:45', '2023-06-17 10:15:12'),
110 -- S016 registered but didn't attend

```

Now, to ensure the database was actually populated:

Click on the database within RDS and check the logs and events tab for the recent activity.

Interesting observation: time is given according to the timezone.

Connectivity & security | Monitoring | **Logs & events** | Configuration | Zero-ETL integrations | Maintenance & backup | >

CloudWatch alarms (0)

Edit alarm Create alarm

| Name | State | More options |
|-----------------|-------|--------------|
| No Alarms found | | |

Recent events (2)

Last 1 day C

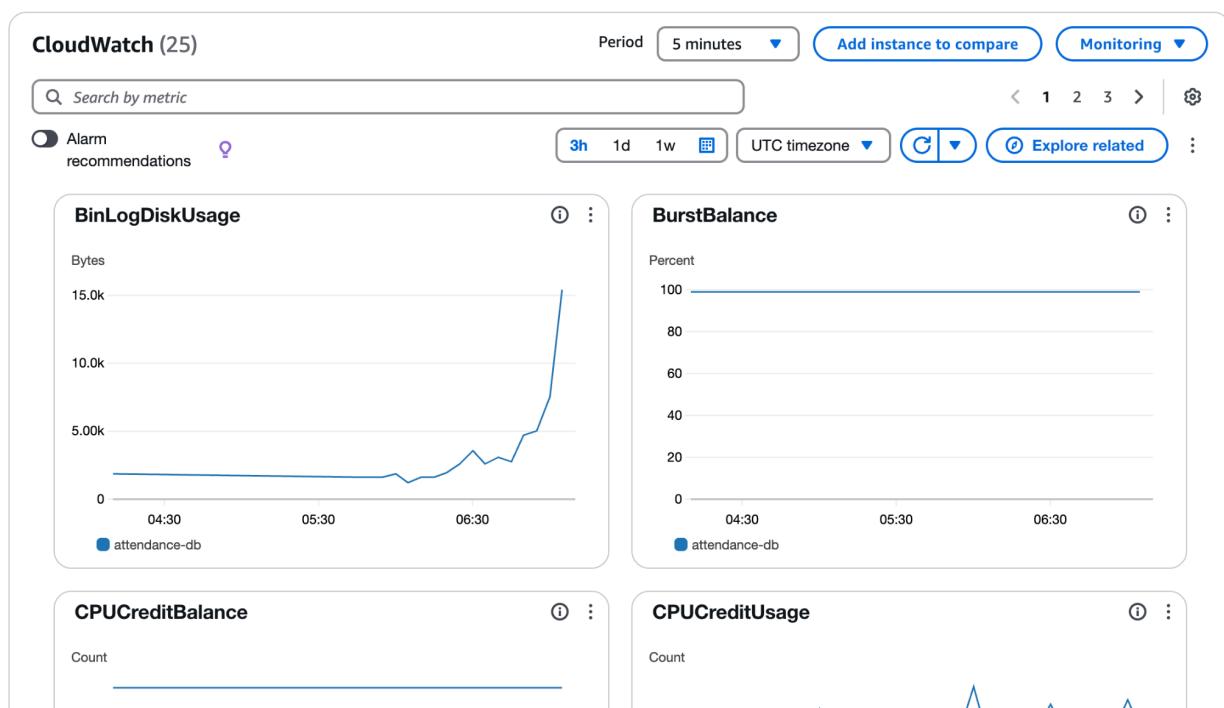
| Time | System notes |
|-----------------------------------|-----------------------------|
| April 22, 2025, 07:34 (UTC+05:30) | Finished DB Instance backup |
| April 22, 2025, 07:33 (UTC+05:30) | Backing up DB instance |

Logs (48)

View Watch Download

| Name | Last written | Size |
|---|-----------------------------------|---------|
| error/mysql-error-running.log | April 22, 2025, 12:30 (UTC+05:30) | 0 bytes |
| error/mysql-error-running.log.2025-04-15.10 | April 15, 2025, 15:15 (UTC+05:30) | 136 B |
| error/mysql-error-running.log.2025-04-15.11 | April 15, 2025, 15:55 (UTC+05:30) | 2.7 kB |
| error/mysql-error-running.log.2025-04-15.14 | April 15, 2025, 18:35 (UTC+05:30) | 138 B |
| error/mysql-error-running.log.2025-04-15.15 | April 15, 2025, 20:05 (UTC+05:30) | 140 B |
| error/mysql-error-running.log.2025-04-15.23 | April 16, 2025, 03:35 (UTC+05:30) | 141 B |
| error/mysql-error-running.log.2025-04-15.8 | April 15, 2025, 13:10 (UTC+05:30) | 3.2 kB |
| error/mysql-error-running.log.2025-04-16.0 | April 16, 2025, 05:20 (UTC+05:30) | 139 B |
| error/mysql-error-running.log.2025-04-16.11 | April 16, 2025, 15:35 (UTC+05:30) | 141 B |
| error/mysql-error-running.log.2025-04-16.12 | April 16, 2025, 16:35 (UTC+05:30) | 139 B |
| error/mysql-error-running.log.2025-04-16.15 | April 16, 2025, 20:10 (UTC+05:30) | 140 B |
| error/mysql-error-running.log.2025-04-16.17 | April 16, 2025, 21:55 (UTC+05:30) | 140 B |
| error/mysql-error-running.log.2025-04-16.18 | April 16, 2025, 23:15 (UTC+05:30) | 150 B |
| error/mysql-error-running.log.2025-04-16.5 | April 16, 2025, 09:35 (UTC+05:30) | 139 B |

Also note RDS ALREADY has Cloudwatch implemented to see the load:
Check this in the monitoring tab:



Using S3 bucket

Lifecycle policies

Go to s3 and click create bucket. Add a name for the bucket and click general purpose. Ensure region selected is the same as your account's current region.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)

Format: s3://bucket/prefix

Remove “Block all public access” settings for the sake of the project.

Screenshot of the AWS S3 'Create bucket' page.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Press create bucket

Screenshot of the AWS S3 'Buckets' page showing a successful bucket creation.

Successfully created bucket "student-fingerprints"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Account snapshot - updated every 24 hours All AWS Regions [View Storage Lens dashboard](#)

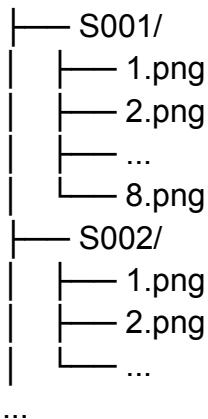
[General purpose buckets](#) [Directory buckets](#)

| General purpose buckets (1) Info All AWS Regions | | Copy ARN | Empty | Delete | Create bucket |
|--|-------------------------------|--|--------------------------------------|------------------------|-------------------------------|
| Buckets are containers for data stored in S3. | | | | | |
| <input type="text"/> Find buckets by name | | < 1 > ⚙️ | | | |
| Name | AWS Region | IAM Access Analyzer | Creation date | | |
| student-fingerprints | Europe (Stockholm) eu-north-1 | View analyzer for eu-north-1 | April 16, 2025, 00:07:10 (UTC+05:30) | | |

Populating the S3 bucket

Go to the newly created s3 bucket. We will be following the below folder structure so we will be populating the data accordingly.

student-fingerprints-2025/



Now click create folder and name the folder the serial number of the first user. Then click create folder.

Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. [Learn more](#)

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the [upload configuration](#) to upload an empty folder and specify the appropriate settings.

Folder

Folder name /

Folder names can't contain "/". [See rules for naming](#)

Server-side encryption Info
Server-side encryption protects data at rest.

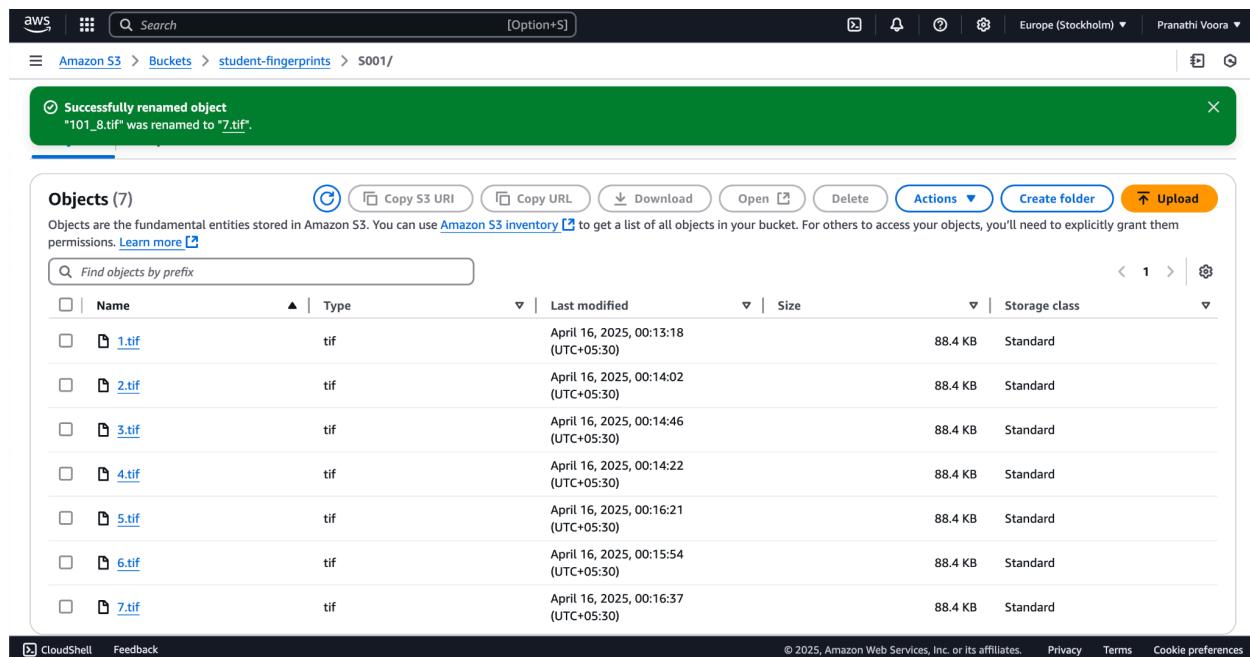
The following encryption settings apply only to the folder object and not to sub-folder objects.

Server-side encryption

Don't specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

Now, populate the folder by uploading the fingerprint images in that folder. Click on the folder and then press upload. Select all the files and rename them to 1.tif, 2.tif.. Etc.



The screenshot shows the AWS S3 console interface. At the top, there's a success message: "Successfully renamed object *101_8.tif* was renamed to *7.tif*". Below this, the "Objects (7)" section lists seven files named 1.tif through 7.tif, all of which are tif files. The table includes columns for Name, Type, Last modified, Size, and Storage class. All files are 88.4 KB and Standard storage class. The "Actions" dropdown menu is visible above the table.

| Name | Type | Last modified | Size | Storage class |
|-------|------|--------------------------------------|---------|---------------|
| 1.tif | tif | April 16, 2025, 00:13:18 (UTC+05:30) | 88.4 KB | Standard |
| 2.tif | tif | April 16, 2025, 00:14:02 (UTC+05:30) | 88.4 KB | Standard |
| 3.tif | tif | April 16, 2025, 00:14:46 (UTC+05:30) | 88.4 KB | Standard |
| 4.tif | tif | April 16, 2025, 00:14:22 (UTC+05:30) | 88.4 KB | Standard |
| 5.tif | tif | April 16, 2025, 00:16:21 (UTC+05:30) | 88.4 KB | Standard |
| 6.tif | tif | April 16, 2025, 00:15:54 (UTC+05:30) | 88.4 KB | Standard |
| 7.tif | tif | April 16, 2025, 00:16:37 (UTC+05:30) | 88.4 KB | Standard |

Lambda Function

We will first set up the lambda function and then go over the logic of the function.

The lambda function will be triggered upon entry of a fingerprint from the site. The attendance entry requires the following fields:

1. EntryID VARCHAR(36) PRIMARY KEY : autogenerated
2. CourseID VARCHAR(20) : manually entered/set at user interface
3. StudentID VARCHAR(20): retrieved from the USERS table when fingerprint is matched
4. StudentAttendance BOOLEAN : true
5. StudentInTime DATETIME : none
6. StudentOutTime DATETIME: none
Updated based on what the last entry was. If last entry was studentintime, then this entry is studentouttime
If the last entry was studentouttime or no entry, then this entry is studentintime
Updated based on current time
7. FOREIGN KEY (CourseID) REFERENCES Courses(CourseID),
8. FOREIGN KEY (StudentID) REFERENCES Users(StudentID)

Here, we create a new lambda function and fill the following fields. Select the author from scratch, and then put down a function name. Select python runtime as we will be writing the lambda function in python. Then click create function.

Create function Info
AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Search [Option+S] Europe (Stockholm) Pranathi Voora

Lambda > Functions > LogFingerprintAttendance

Successfully created the function LogFingerprintAttendance. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

LogFingerprintAttendance

Function overview Info

LogFingerprintAttendance
 Layers (0)

Description
-

Last modified
2 seconds ago

Function ARN
 arn:aws:lambda:eu-north-1:586794464928:function:LogFingerprintAttendance

Function URL Info
-

Code Test Monitor Configuration Aliases Versions

Code source Info

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now we must create a react frontend that does the following:

- Triggered from your React frontend.
- Takes a fingerprint image and course ID.

- Matches fingerprint with S3.
- Gets student info from RDS.
- Check the last attendance entry for the student.
- Updates the current record with **StudentInTime** or **StudentOutTime**.

Create IAM role

Why create an IAM role?

Go to IAM and click Roles.

Identity and Access Management (IAM)

Roles (6) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

| Role name | Trusted entities | Last activity |
|--|--|----------------|
| AWSServiceRoleForRDS | AWS Service: rds (Service-Linked Role) | 12 minutes ago |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linked Role) | - |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service) | - |
| bits-email-check-role-aq8ydu1n | AWS Service: lambda | 52 days ago |
| LogFingerprintAttendance-role-ks5a8djg | AWS Service: lambda | - |
| rds-monitoring-role | AWS Service: monitoring.rds | - |

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you

X.509 Standard

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

Find the role corresponding to the lambda function created. Click attach policies.

Screenshot of the AWS IAM Roles page for the role "attendanceLambda-role-tfkkqvg8".

Identity and Access Management (IAM)

Last activity: tfkkqvg8

Maximum session duration: 1 hour

Permissions (selected) | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (2)

You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|--------------------------------------|------------------|-------------------|
| AWSLambdaBasicExecutionRole-1961f... | Customer managed | 1 |
| LambdaPolicyAttendance | Customer inline | 0 |

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

https://us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/roles/details/attendanceLambda-role-tfkkqvg8/attach-policies

Add the policies **AWSLambdaBasicExecutionRole** and **AmazonS3ReadOnlyAccess** (write why)

Screenshot of the AWS IAM Create Role page, Step 1: Add permissions.

Step 1: Select trusted entity

Step 2: Add permissions (selected)

Step 3: Name, review, and create

Add permissions

Permissions policies (2/1040)

Choose one or more policies to attach to your new role.

| Policy name | Type | Description |
|------------------------|-------------|--|
| AmazonS3ReadOnlyAccess | AWS managed | Provides read only access to all bucket... |

Set permissions boundary - optional

Cancel | Previous | Next

Press create inline policy once role is created

Screenshot of the AWS IAM Roles page showing the LambdaRDSFingerprint role.

Identity and Access Management (IAM)

- Dashboard
- Access management**
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
 - Root access management [New](#)
- Access reports**
 - Access Analyzer
 - Archive rules
 - Analyzers
 - Settings
 - Credential report
 - Organization activity

LambdaRDSFingerprint [Info](#)

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
April 16, 2025, 13:46 (UTC+05:30)

Last activity
-

ARN
[arn:aws:iam::586794464928:role/LambdaRDSFingerprint](#)

Maximum session duration
1 hour

Permissions [Edit](#) [Trust relationships](#) [Tags](#) [Last Accessed](#) [Revoke sessions](#)

Permissions policies (2) [Info](#)

You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|---|-------------|-------------------|
| AmazonS3ReadOnlyAccess | AWS managed | 2 |
| AWSLambdaBasicExecutionRole | AWS managed | 2 |

Add permissions ▾

[Attach policies](#) [Create inline policy](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

In json, add the following data

Screenshot of the AWS IAM Create Policy page.

Step 1

Specify permissions [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

JSON

```

1▼ {
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Action": [
6        "rds-data:*",
7        "secretsmanager:GetSecretValue"
8      ],
9      "Effect": "Allow",
10     "Resource": "*"
11   }
12 ]
13 }
14

```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then click create policy

Screenshot of the AWS IAM Roles page showing the creation of a new policy.

Identity and Access Management (IAM)

Policy AllowRDSSecretsPolicy created.

Creation date: April 16, 2025, 15:46 (UTC+05:30) **ARN:** arn:aws:iam::586794464928:role/LambdaRDSFingerprint

Last activity: - **Maximum session duration:** 1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (3) Info
You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|-----------------------------|-----------------|-------------------|
| AllowRDSSecretsPolicy | Customer inline | 0 |
| AmazonS3ReadOnlyAccess | AWS managed | 2 |
| AWSLambdaBasicExecutionRole | AWS managed | 2 |

Permissions boundary (not set)

Now back at the lambda function, click add environment variables

Screenshot of the AWS Lambda Functions page showing the configuration of an attendanceLambda function.

TEST EVENTS [NONE SELECTED]
+ Create new test event

ENVIRONMENT VARIABLES
You haven't configured any environment variables.
Add environment variables

Code properties [Info](#)

- Package size: 295 byte
- SHA256 hash: q8E7Nexf5xxhKT9/d4bGpAYOXJYFAUj0UDj8OivK8E=
- Last modified: 6 minutes ago

Encryption with AWS KMS customer managed KMS key [Info](#)

Runtime settings [Info](#)

- Runtime: Node.js 22.x
- Handler: [Info](#) index.handler
- Architecture: [Info](#) x86_64

Edit | **Edit runtime management configuration**

Runtime management configuration

Info | **Tutorials**

Function configuration

Use the function overview to see your function's configuration. You can choose to view this as a diagram or as an AWS Serverless Application Model (AWS SAM) template. AWS SAM is an open source framework for building serverless applications using infrastructure as code (IaC). You can see the following types of resources in the diagram or template:

- Triggers** are AWS services or resources that invoke the function.
- Destinations** are AWS resources that receive a record of an invocation after success or failure. You can configure Lambda to send invocation records when your function is invoked asynchronously, or if your function processes records from a stream. The contents of the invocation record and supported destination services vary by source.
- Layers** are resources that contain libraries, a custom runtime, or other dependencies.

Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

There are no environment variables on this function.

[Add environment variable](#)

Encryption configuration

[Cancel](#) [Save](#)

Environment variables

An environment variable is a key-pair of strings that are stored in a function's version-specific configuration. Environment variables are available through the Lambda runtime and additional environment variables are set that contain information about the function and invocation request.

Node.js – let region = process.env.AWS_REGION

Python – region = os.environ['AWS_REGION']

Ruby – region = ENV["AWS_REGION"]

Java – String region =

Add the following environment variables

Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

| Key | Value | Action |
|----------------|----------------------|------------------------|
| DB_SECRET_ARN | | Remove |
| DB_NAME | attendance-db | Remove |
| DB_CLUSTER_ARN | | Remove |
| BUCKET_NAME | student-fingerprints | Remove |

[Add environment variable](#)

Encryption configuration

[Cancel](#) [Save](#)

Environment variables

An environment variable is a key-pair of strings that are stored in a function's version-specific configuration. Environment variables are available through the Lambda runtime and additional environment variables are set that contain information about the function and invocation request.

Node.js – let region = process.env.AWS_REGION

Python – region = os.environ['AWS_REGION']

Ruby – region = ENV["AWS_REGION"]

Java – String region =

System.getenv("AWS_REGION");

Go – var region = os.Getenv("AWS_REGION")

C# – string region =

Create a secret for aws rds db created. Go to aws secret manager

The screenshot shows the AWS Secrets Manager interface. At the top, there's a search bar and navigation links for 'Europe (Stockholm)' and 'Pranathi Voora'. Below the header, a banner says 'No secrets' with a 'Store a new secret' button. The main area has columns for 'Secret name', 'Description', and 'Last retrieved (UTC)'. A small robot icon is displayed above the table.

Add the secret credentials as according to the rds db.

This screenshot shows the 'Store a new secret' wizard at Step 2: 'Configure secret'. It asks for the 'Secret type': 'Credentials for Amazon RDS database' is selected. Other options include 'Credentials for other database' and 'Other type of secret'. The 'User name' field contains 'admin' and the 'Password' field contains '*****'. The 'Encryption key' dropdown is set to 'aws/secretsmanager'.

Select the correct db. Add name for secret.

This screenshot shows the 'Configure secret' step at Step 3: 'Configure secret'. The 'Secret name' is 'rds-attendance-db' and the 'Description - optional' is 'Access to MySQL prod database for my AppBeta'.

Now copy its arn

aws Search [Option+S] Europe (Stockholm) Pranathi Voora

AWS Secrets Manager > Secrets > rds-attendance-db

You successfully stored the secret rds-attendance-db. To show it in the list, choose Refresh. Use the sample code to update your applications to retrieve this secret.

[View details](#) [See sample code](#)

rds-attendance-db

Secret details

| | |
|--|-------------------------|
| Encryption key aws/secretsmanager | Secret description - |
| Secret name Secret ARN copied arn:aws:secretsmanager:eu-north-1:586794464928:secret:rds-attendance-db-nlBOAh | Actions |

[Overview](#) [Rotation](#) [Versions](#) [Replication](#) [Tags](#)

Secret value [Info](#) [Retrieve secret value](#)

Retrieve and view the secret value.

Find the db cluster arn too

aws management console Europe (Stockholm) Pranathi Voora

Aurora and RDS > Databases > attendance-db

[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](#) [Events](#) [Event subscriptions](#) [Recommendations](#) [Certificate update](#)

[CloudShell](#) [Feedback](#)

| Instance | | | |
|---|--------------------------------------|---|---|
| Configuration | Instance class | Storage | Monitoring |
| DB instance ID attendance-db | Instance class db.t3.micro | Encryption Enabled | Monitoring type Database Insights - Standard |
| Engine version 8.0.40 | vCPU 2 | AWS KMS key aws/rds | Performance Insights Disabled |
| RDS Extended Support Disabled | RAM 1 GB | Storage type General Purpose SSD (gp2) | Enhanced Monitoring Disabled |
| DB name - | Availability | Storage 20 GiB | DevOps Guru Disabled |
| License model General Public License | Master username admin | Provisioned IOPS - | |
| Option groups ARН copied | Master password ***** | Storage throughput - | |
| Promote resource Name (ARN) arn:aws:rds:eu-north-1:586794464928:db:attendance-db | IAM DB authentication Not enabled | Storage autoscaling Enabled | |
| Resource ID db- | Multi-AZ No | Maximum storage threshold 1000 GiB | |

Now copy paste the lambda code into the lambda function code section with the file name index.py. Then click deploy.

```

EXPLORER
lambda_function.py

lambda_function.py
45  def get_last_attendance(student_id, course_id):
46      if result['records']:
47          return {
48              ...
49      }
50
51  return None

52
53  def insert_attendance(student_id, course_id, student_in):
54      sql = """
55      INSERT INTO Attendance (EntryID, CourseID, StudentID, StudentAttendance, StudentInTime)
56      VALUES (:eid, :cid, :sid, :att, :intime)
57      """
58
59      rds.execute_statement(
60          secretArn=DB_SECRET_ARN,
61          resourceArn=DB_CLUSTER_ARN,
62          database=DB_NAME,
63          sql=sql,
64          parameters=[
65              {'name': 'eid', 'value': {'stringValue': str(uuid.uuid4())}},
66              {'name': 'cid', 'value': {'stringValue': course_id}},
67              {'name': 'sid', 'value': {'stringValue': student_id}},
68              {'name': 'att', 'value': {'booleanValue': True}},
69              {'name': 'intime', 'value': {'stringValue': student_in}}
70          ]
71      )

72
73  def update_out_time(entry_id, student_out):
74      sql = """
75      UPDATE Attendance SET StudentOutTime = :outtime
76      WHERE EntryID = :eid
77      """

78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

Deploy (D&U)

Test (T&E)

ENVIRONMENT VARIABLES

- BUCKET_NAME = student-fingerprints
- DB_CLUSTER_ARN = arn:aws:rds:eu-nort...
- DB_NAME = attendance-db
- DB_SECRET_ARN = arn:aws:secretsmana...

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then check the lambda code to ensure it can do what i want it to do and how to handle the data correctly

Setting up API gateway

Click http api in the api gateway

Choose an API type [Info](#)

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

[Import](#) [Build](#)

WebSocket API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

[Build](#)

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Configure the API. add an API name and select integration as lambda. Then choose the aws region as eu-north-1 and select the lambda function created.

Step 1

Configure API

API details

API name
An HTTP API must have a name. The name is a non-unique value you use to identify and organize your APIs. To programmatically refer to this API, use the API ID that API Gateway generates for you.

IP address type [Info](#)
Select the type of IP addresses that can invoke the default endpoint for your API. You don't need to redeploy your API for the update to take effect.
 IPv4 Includes only IPv4 addresses.
 Dualstack Includes IPv4 and IPv6 addresses.

Integrations (1) [Info](#)
Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For an HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.
Lambda [Remove](#)
AWS Region **Lambda function** **Version** [Learn more](#)
 eu-north-1 2.0
[Add integration](#)

Choose POST method

Step 1

Configure API

Step 2 - optional

Configure routes

Step 3 - optional

Define stages

Step 4

Review and create

Configure routes - optional

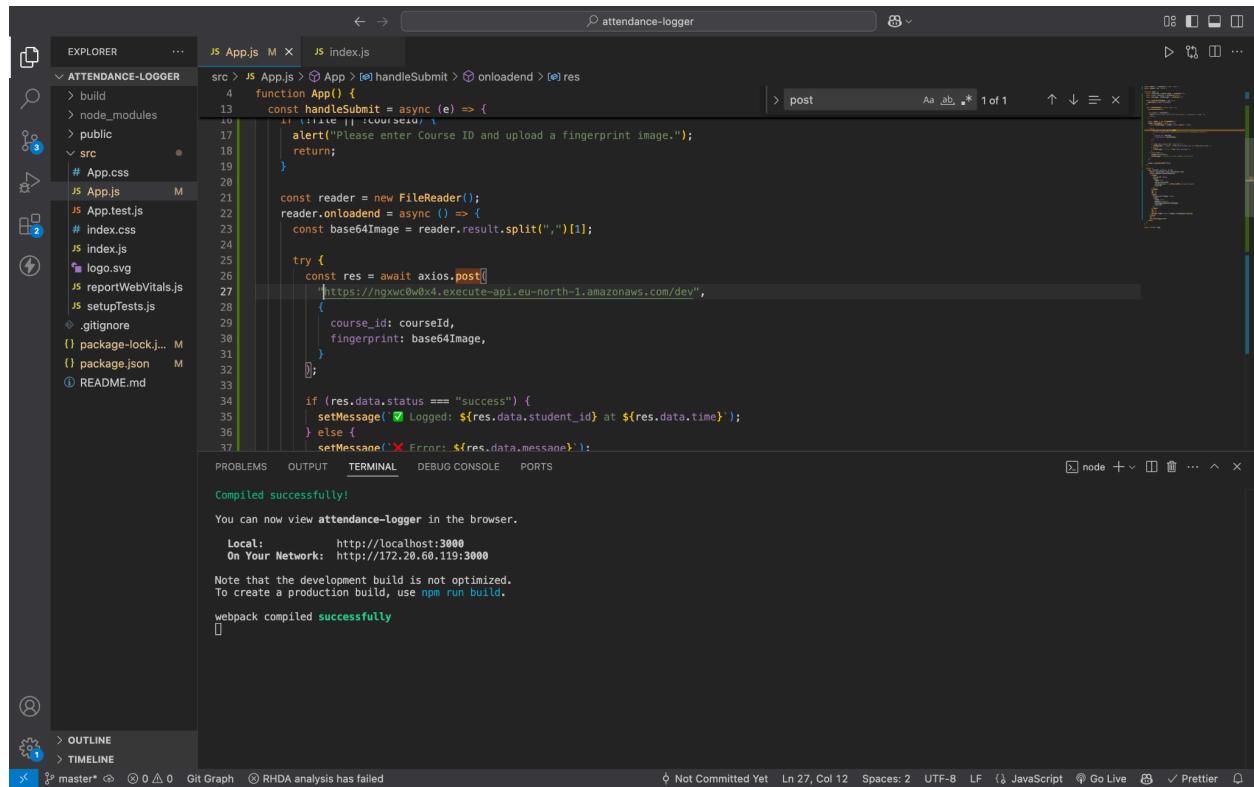
Configure routes [Info](#)
API Gateway uses routes to expose integrations to consumers of your API. Routes for HTTP APIs consist of two parts: an HTTP method and a resource path (e.g., GET /pets). You can define specific HTTP methods for your integration (GET, POST, PUT, PATCH, HEAD, OPTIONS, and DELETE) or use the ANY method to match all methods that you haven't defined on a given resource.

| Method | Resource path | Integration target |
|--------|-------------------|--------------------|
| POST | /attendanceLambda | attendanceLambda |

[Add route](#) [Cancel](#) [Review and create](#) [Previous](#) [Next](#)

Then create the api. You will see the route now.

Now attach frontend in react and use the POST method with axios to send api request



```

    4 function App() {
13     const handleSubmit = async (e) => {
14         e.preventDefault();
15         const course_id = e.target.course_id.value;
16         alert("Please enter Course ID and upload a fingerprint image.");
17         return;
18     }
19
20     const reader = new FileReader();
21     reader.onloadend = async () => {
22         const base64Image = reader.result.split(",")[1];
23
24         try {
25             const res = await axios.post(
26                 "https://ngxwco00x4.execute-api.eu-north-1.amazonaws.com/dev",
27                 {
28                     course_id: courseId,
29                     fingerprint: base64Image,
30                 }
31             );
32
33             if (res.data.status === "success") {
34                 setMessage(`Logged: ${res.data.student_id} at ${res.data.time}`);
35             } else {
36                 setMessage(`Error: ${res.data.message}`);
37             }
38         }
39     }
40
41     if (res.data.status === "success") {
42         setMessage(`Logged: ${res.data.student_id} at ${res.data.time}`);
43     } else {
44         setMessage(`Error: ${res.data.message}`);
45     }
46 }
47
48 
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

Compiled successfully!

You can now view attendance-logger in the browser.

Local: http://localhost:3000
On Your Network: http://172.20.68.119:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Not Committed Yet Ln 27, Col 12 Spaces: 2 UTF-8 LF ↴ JavaScript Go Live Prettier

Now when i put the first entry for student 1 (based on fingerprint 4) I receive:

Log Student Exam Attendance!

Course ID:

Fingerprint Image: 101_4.tif

Student Entry record updated

And when i put second entry for student 1 (based on fingerprint 5), I receive:

Log Student Exam Attendance!

Course ID:

Fingerprint Image: 101_5.tif

Student Exit record updated





Chatbot

LLM

AWS Lex + GPT?

Frontend sends the input to AWS Lex via API calls and displays the bot's responses.

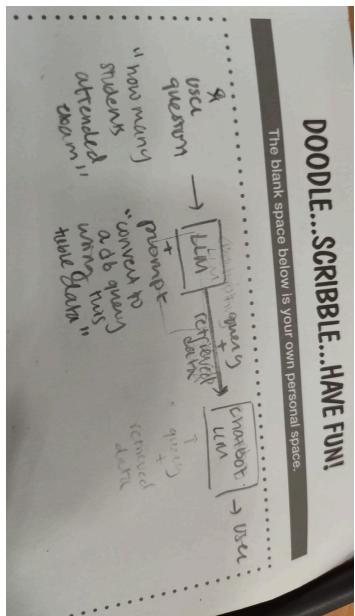
AWS Lex: For recognising user intents, extract slot values

Keep a high x field for the LLM so that answers are precise to the data given.

Pass the exact data in the form of a query based on the type of question asked by the user. If type matches the following, pass the corresponding query results. We can do this by extracting the keywords.

1. Number of students enrolled in course
2. Number of students attending exam in course
3. Number of students that did not attend exam in course
4. Number of students that are late to exam
5. Number of students on time/ early to exam
6. Who is professor for this course
7. Names of students that did not attend exam
8. Names of students late to exam
9. ID of students that did not attend exam
10. ID of students late to exam





little snapshot of architecture brainstorming

Our idea:

https://www.reddit.com/r/educationalgifs/comments/109c44s/how_to_format_the_openai_prompt_to_have_it/

Implementing such a prompting LLM to create SQL queries

Lazy sql query

ISSUE FACED:

Region Unsupported

Amazon Lex is not available in Europe (Stockholm). Please select another region.

Supported Regions

United States (N. Virginia)

United States (Oregon)

Asia Pacific (Seoul)

Asia Pacific (Singapore)

Asia Pacific (Sydney)

Asia Pacific (Tokyo)

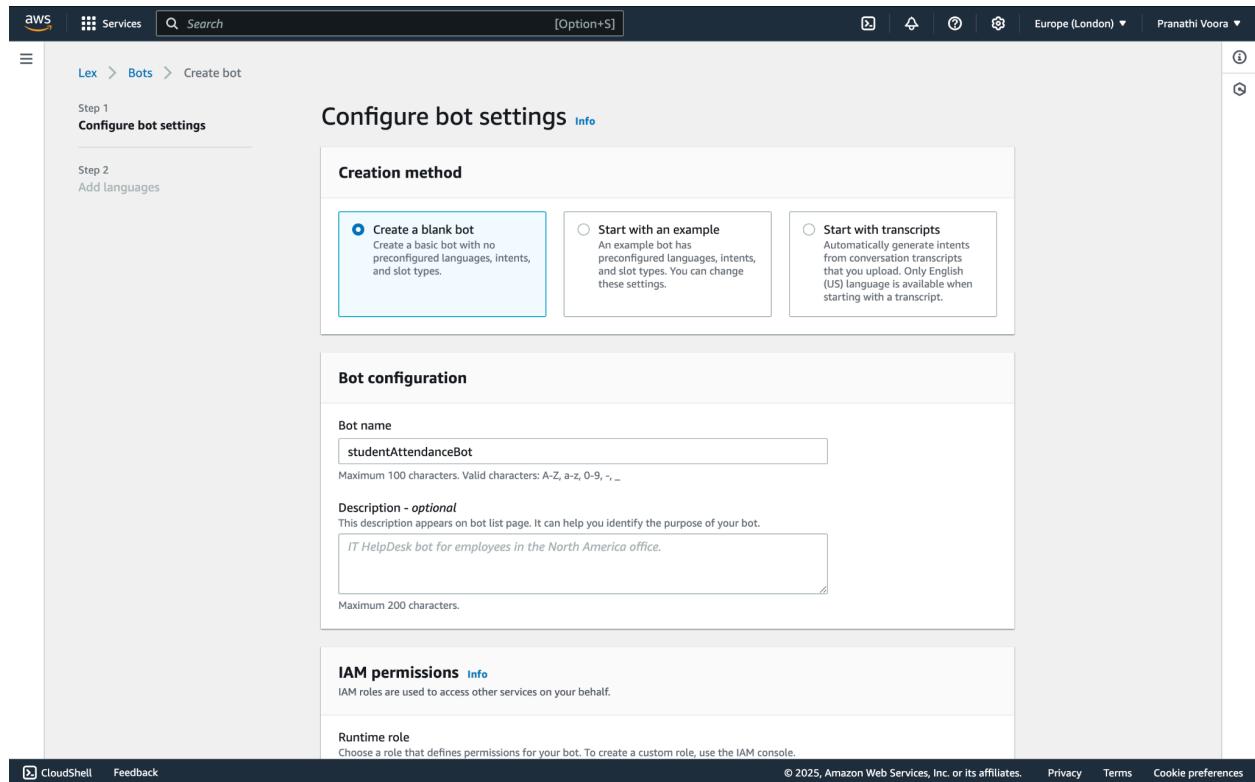
Canada (Central)

Europe (Frankfurt)

Europe (Ireland)

Europe (London)

We will proceed with europe (london)



Configure bot settings Info

Creation method

- Create a blank bot**
Create a basic bot with no preconfigured languages, intents, and slot types.
- Start with an example**
An example bot has preconfigured languages, intents, and slot types. You can change these settings.
- Start with transcripts**
Automatically generate intents from conversation transcripts that you upload. Only English (US) language is available when starting with a transcript.

Bot configuration

Bot name

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - optional
This description appears on bot list page. It can help you identify the purpose of your bot.

Maximum 200 characters.

IAM permissions Info
IAM roles are used to access other services on your behalf.

Runtime role
Choose a role that defines permissions for your bot. To create a custom role, use the IAM console.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Description: operator

This description appears on bot list page. It can help you identify the purpose of your bot.

IT HelpDesk bot for employees in the North America office.

Maximum 200 characters.

IAM permissions [Info](#)

IAM roles are used to access other services on your behalf.

Runtime role

Choose a role that defines permissions for your bot. To create a custom role, use the IAM console.

- Create a role with basic Amazon Lex permissions.
- Use an existing role.

Creating a role takes a few minutes. Don't delete the role or edit the trust or permissions policies in this role until we've finished creating it.

New role

Amazon Lex creates a runtime role with permission to upload to Amazon CloudWatch Logs.

AWSServiceRoleForLexV2Bots_TMGJ8B34NPD

Children's Online Privacy Protection Act (COPPA) [Info](#)

Is use of your bot subject to the [Children's Online Privacy Protection Act \(COPPA\)](#)?

- Yes
- No

Idle session timeout

You can configure how long a session is maintained when the user does not provide any input and the session is idle. Amazon Lex retains context information until a session ends.

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates.

[Privacy](#) [Terms](#) [Cookie preferences](#)

Choose a larger idle session timeout for this usecase:

Idle session timeout

You can configure how long a session is maintained when the user does not provide any input and the session is idle. Amazon Lex retains context information until a session ends.

Session timeout

20

minute(s)



By default, session duration is 5 minutes, but you can specify any duration between 1 and 1440 minutes (24 hours).

Press next

Choose language settings and your preferred voice interaction. Click next.

aws Services Search [Option+S] Europe (London) ▾ Pranathi Voora ▾

Lex > Bots > Create bot Step 1 Configure bot settings Step 2 Add languages

Add language to bot Info

▼ Language: English (US)

Select language

Description - *optional*

Maximum 200 characters.

Voice interaction The text-to-speech voice that your bot uses to interact with users.

Voice sample

Intent classification confidence score threshold Min: 0.00, max: 1.00.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now we have to set intent.

▼ Intent details Info

Intent name Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional* Maximum 200 characters.

ID: EGJ039CKRB

Add utterances for the use case: We have kept table names in brackets for ease of understanding.

Sample utterances (9) [Info](#)

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

Filter Sort by added (ascending) ▾

[Preview](#) [Plain text](#)

I want to know how many {Users} enrolled in {Course}

i want to know how many {Users} attended the {Exam}

how many {Users} that did not attend the {ExamAttendance} are enrolled in the {Course}?

⋮ Number of {Users} early for {Exam} X

Who is {Professor} for this {Course}

⋮ Names of {Users} late to {ExamAttendance} X

ID of {Users} that did not attend {ExamAttendance}

which {Users} missed the {ExamAttendance} for this {Course}

which {Exam} does this {Users} have

Now for the slots, to improve the flexibility of the chatbot and our ‘lazy querying’ feature, we are not looking for specific pieces of information per answer, but rather send the query to the LLM to find the respective SQL query.

Hence, we will be using a general slot.

Initial response [Info](#)

You can provide responses based on user conditions.

► **Responses** [Info](#)

Message:

Add slot

A slot is used to capture information from the user to fulfill the intent.

Required for this intent
The bot will prompt for this slot during the conversation if a value is not provided by the user.

| | |
|--------------|----------------------|
| Name | Slot type |
| UserQuestion | AMAZON.FreeFormInput |

Prompts

What is your question?

[Cancel](#) [Add](#)

You haven't added any slots yet.

Adding intents:

Fulfillment [Info](#) Active

Run a lambda function to fulfill the intent and inform users of the status when it's complete.

▼ On successful fulfillment **In case of failure**

Message: Please head over to this github repository an...

On successful fulfillment

Please head over to this github repository and write your queries in the original value section of the lambda_functi

In case of failure

Something went wrong

Advanced options

Configure success, failure, and timeout responses.

Intent: ProfessorName [Info](#)

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

► Conversation flow [Info](#)

▼ Intent details [Info](#)

Intent name

ProfessorName

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - optional

find out the name of the professor

Maximum 200 characters.

ID: P7PYJ5TMBB

▼ Contexts - optional [Info](#)

[Editor](#)

[Visual builder](#)

[New](#)

[Save intent](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Sample utterances (3) [Info](#)

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

[Filter](#)

Sort by added (ascending) ▾

[Preview](#)

[Plain text](#)

hi have questions regarding student exam attendance

i need to know regarding exams

i need to know about students and exams

Fulfillment Info

Active

Run a lambda function to fulfill the intent and inform users of the status when it's complete.

▼ On successful fulfillment

Message: Which professor are you?

In case of failure

Message: -

On successful fulfillment

Which professor are you?

In case of failure

Something went wrong

Advanced options

Configure success, failure, and timeout responses.

Intent: CourseName Info

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

► Conversation flow Info

▼ Intent details Info

Intent name

CourseName

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - optional

to find out which course the professor is taking

Maximum 200 characters.

ID: 5CPEFLCG2O

▼ Contexts - optional Info

Editor

Visual builder

New

Save intent

Screenshot of the AWS Lambda console showing the configuration of an Amazon Lex intent.

Intent: EXAMATTENDANCEQUERY

Intent details:

- Intent name:** ExamAttendanceQuery
- Description - optional:** Ask questions regarding student exam attendance for courses
- Input contexts:** None

Conversation flow:

```

graph LR
    Start(( )) --> Q1[User: hi there!]
    Q1 --> A1[Bot: i want to know about student exam attendance]
    A1 --> Q2[User: Which professor are you?]
    Q2 --> A2[Bot: My name is Ria]
    A2 --> Q3[User: Great! What do you want to know about exam attendance?]
    Q3 --> A3[Bot: how many students attended my course exam]
    A3 --> Q4[User: Please head over to this github repository and write your queries in the original value section of the lambda_function.py. Enjoy!]
  
```

Test Draft version: Last build submitted: 1 minute ago

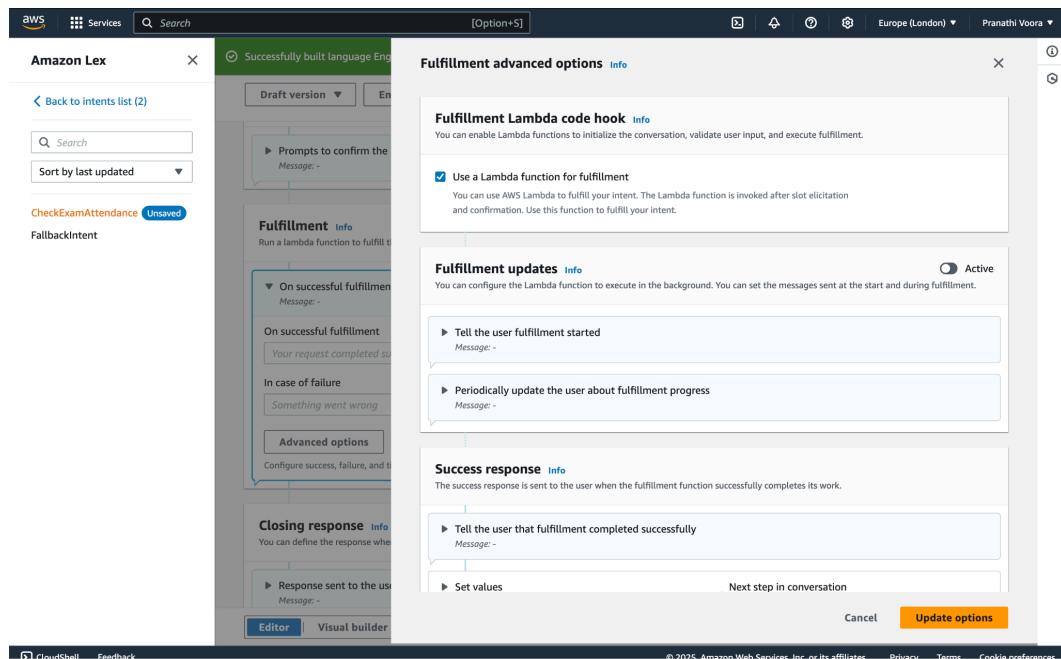
Ready for complete testing

Save intent

Lambda Function

Now we have to connect to a lambda function.

Check the box that says to use lambda function for fulfillment.



Creating the lambda function:

```
lex-aws > λ lambda_function.py > ⌂ lambda_handler
 1  from llm_handler import generate_sql_query, generate_response
 2  from sql_executor import run_query
 3  import os
 4
 5  def lambda_handler(event, context):
 6      user_question = event['sessionState']['intent']['slots']['UserQuestion']['value']['originalValue']
 7
 8      schema = open('schema.md', 'r').read()
 9
10      prompt = f"""
11 Database schema:
12 {schema}
13
14 User asked:
15 "{user_question}"
16
17 Write an SQL query to answer it.
18 Just return the query.
19 """
20
21      sql_query = generate_sql_query(prompt)
22
23      try:
24          data = run_query(sql_query)
25      except Exception as e:
26          return format_response(f"Error while querying: {str(e)}")
27
28      response_prompt = f"User asked: {user_question}\nData: {data}\nRespond in clear English."
29      final_response = generate_response(response_prompt)
30
31      return format_response(final_response)
32
33  def format_response(message):
34      return {
35          "sessionState": {
36              "dialogAction": {
37                  "type": "Close"
38              },
39              "intent": {
40                  "name": "ExamAttendanceQuery",
41                  "state": "Fulfilled"
42              }
43          },
44          "messages": [
45              {
46                  "contentType": "PlainText",
47                  "content": message
48              }
49          ]
50      }
```

```
lex-aws > llm_handler.py > ...
1 import openai
2 import os
3
4 openai.api_key = 'sk-proj-5c7vzYJ2kKFilw1HN4hG1KIAP-zB_baZJN6BDYkvrl2UdRwl0uchIblDQ1ttqK-zYohHLR8jYiT3BlbkFJdVzn2Eu7ZVBH8eHTCr6fGMGYaftB0e637c
5
6 def generate_sql_query(prompt):
7     return call_llm(prompt)
8
9 def generate_response(prompt):
10    return call_llm(prompt)
11
12 def call_llm(prompt):
13     res = openai.ChatCompletion.create(
14         model="gpt-3.5-turbo",
15         messages=[{"role": "user", "content": prompt}]
16     )
17     return res['choices'][0]['message']['content'].strip()
18
```

```
lex-aws > sql_executor.py > run_query
1 import pymysql
2
3 def run_query(sql):
4     DB_HOST = 'attendance-db.c7oe4k42eh8z.eu-north-1.rds.amazonaws.com'
5     DB_USER = 'admin'
6     DB_PASS = 'ccgroup11'
7     DB_NAME = 'attendance-db'
8
9     conn = pymysql.connect(
10         host=DB_HOST,
11         user=DB_USER,
12         password=DB_PASS,
13         database=DB_NAME,
14         cursorclass=pymysql.cursors.DictCursor
15     )
16
17     with conn:
18         with conn.cursor() as cursor:
19             cursor.execute(sql)
20             result = cursor.fetchall()
21
22     return result
23
```

Making the schema:

```
lex-aws > ↓ schema.md
1  Tables:
2
3  Users(StudentID, Name, BatchNumber, Branch, FingerprintURL)
4  Professor(ProfessorID, Name)
5  Course(CourseID, CourseName, ProfessorID)
6  Exam(ExamID, StartTime, EndTime, CourseID)
7  ExamAttendance(EntryID, CourseID, StudentID, StudentInTime, StudentOutTime)
8  CourseRegistrations(RegistrationID, CourseID, StudentID)
9
10 Relationships:
11
12 - Exam.CourseID -> Course.CourseID
13 - ExamAttendance.CourseID -> Course.CourseID
14 - ExamAttendance.StudentID -> Users.StudentID
15 - CourseRegistrations.CourseID -> Course.CourseID
16 - CourseRegistrations.StudentID -> Users.StudentID
17 - Course.ProfessorID -> Professor.ProfessorID
18
```

Create a new lambda function

AWS | Search [Option+S] | Europe (Stockholm) | Pranathi Voora | ⓘ | ⓘ

Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 (C)

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Change default execution role

Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

Create function

Since the rds has public access, we do not need to use permissions.

Upload all the code

AWS Lambda > Functions > ConnectLex

Code source Info

Upload from ...

```

EXPLORER
CONNECTLEX
__MACOSX
__lex-aws
lex-aws
lambda_function.py
llm_handler.py
schema.md
sql_executor.py

lambda_function.py
llm_handler.py
schema.md
sql_executor.py

sql_executor.py

import pymysql
def run_query(sql):
    DB_HOST = 'attendance-db.c7oe4k42eh8z.eu-north-1.rds.amazonaws.com'
    DB_USER = 'admin'
    DB_PASS = 'cgroup11'
    DB_NAME = 'attendance-db'

    conn = pymysql.connect(
        host=DB_HOST,
        user=DB_USER,
        password=DB_PASS,
        database=DB_NAME,
        cursorclass=pymysql.cursors.DictCursor
    )

    with conn:
        with conn.cursor() as cursor:
            cursor.execute(sql)
            result = cursor.fetchall()

    return result

```

Deploy [UNDEPLOYED CHANGES]

You have undeployed changes.

Deploy (⌘U) Test (⌘K)

TEST EVENTS [NONE SELECTED]

+ Create new test event

ENVIRONMENT VARIABLES

Deploying code

Log 10 Col 22 Spaces: 4 LFE: 8 LE Python Lambda Layout: US C © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Add these permissions:

IAM > Roles > ConnectLexRole

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management New

Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies New

IAM Identity Center

AWS Organizations

Policy was successfully attached to role.

Creation date
April 22, 2025, 14:23 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::586794464928:role/service-role/ConnectLexRole

Maximum session duration
1 hour

Permissions Trust relationships Tags Last Accessed Revoke sessions

Permissions policies (2) Info

You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|--------------------------------------|------------------|-------------------|
| AmazonLexFullAccess | AWS managed | 1 |
| AWSLambdaBasicExecutionRole-48be8... | Customer managed | 1 |

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

Generate policy

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon lex full access, default -> amazon accessbasic execution role

Screenshot of the AWS IAM "Add permissions" page for a role named "ConnectLexRole". The search bar shows "Search [Option+S]" and the user "Pranathi Voora".

The "Attach policy to ConnectLexRole" section shows "Current permissions policies (1)".

The "Other permissions policies (1049)" section is filtered by "Policy name" containing "amazonlex". It lists three AWS managed policies:

- AmazonLexFullAccess**: Select data for AmazonLexFullAccess, AWS managed
- AmazonLexReadOnly**: AWS managed
- AmazonLexRunBotsOnly**: AWS managed

Buttons at the bottom right include "Cancel" and "Add permissions".

ERROR! BOTH LEX AND LAMBDA FUNCTION NEED TO BE IN SAME REGION SO HAD TO REMAKE THE LAMBDA FUNCTION IN EUROPE LONDON.

Now connect the lambda function by navigating to alias for the bot as shown by the header navigation

AWS Services Search [Option+S] Europe (London) Pranathi Voora

Amazon Lex

- Bots
- Bot templates New
- Networks of bots
- studentAttendanceBot**
 - Bot versions
 - Draft version
 - All languages
 - English (US)
 - Intents
 - Slot types
- Deployment
 - Aliases**
 - Channel integrations
- Analytics New
 - Overview
 - Conversation dashboard
 - Conversation flows
 - Conversations
- Performance dashboard
 - Intent performance
 - Utterance recognition
- CloudWatch metrics

Alias language support: English (US)

Lambda function - optional
This Lambda function is invoked for initialization, validation, and fulfillment.

Source

Lambda function version or alias

Learn more about Lambda

Cancel Save

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Once it is added, we can build the bot again

AWS Services Search [Option+S] Europe (London) Pranathi Voora

Amazon Lex

Back to intents list (2)

Search Sort by last updated

ExamAttendanceQuery FallbackIntent

Building language English (US) in bot: studentAttendanceBot. If your language contains external source slot types, the build might take longer to complete.

Draft version English (US) Building

Build Test

Preview Plain text

I want to know how many students enrolled in course

i want to know how many students attended the course exam?

how many students that did not attend the exam are enrolled in the course?

Success! Successfully built language English (US) in bot: studentAttendanceBot

The screenshot shows the AWS Lambda function configuration interface. The left sidebar lists functions: 'ExamAttendanceQuery' (Unsaved) and 'FallbackIntent'. The main area shows the 'Code' tab for 'ExamAttendanceQuery'. The code editor contains the following Python code:

```
def lambda_handler(event, context):  
    # Add your code here  
    return {  
        "statusCode": 200,  
        "body": "Hello from Lambda!"  
    }
```

The 'Handler' dropdown is set to 'lambda.lambda_handler'. The 'Runtime' dropdown is set to 'File-based'. The 'Memory size' is 128 MB and 'Timeout' is 3 seconds.

ERROR IMPORT ERROR

As the chatbot recognised the question and sent to lambda function, we checked the cloudwatch logs. Then we saw the errors with the lambda function.

So we have to create a lambda layer for imports and dependencies.

Screenshot of the AWS Lambda 'Create layer' configuration page.

Layer configuration

Name: pipLayer

Description - optional: new pip installation as per stackoverflow

Upload a .zip file
 Upload a file from Amazon S3

Choose file: pydantic_layer.zip (7.44 MB)

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - optional | Info
 Choose the compatible instruction set architectures for your layer.

x86_64

Compatible runtimes - optional | Info
 Choose up to 15 runtimes.

Python 3.13

License - optional | Info

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then, connect the lambda layer to the lambda function.

| Layers Info | | | | | |
|-----------------------------|----------|---------------|---------------------|--------------------------|--|
| Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN |
| 1 | pipLayer | 1 | python3.13 | x86_64 | arn:aws:lambda:eu-west-2:586794464928:layer:pipLayer:1 |

STILL DEPENDENCIES ARE NOT WORKING
 SOLUTION: CREATE DOCKER CONTAINER

Creating the container:

Testing the docker locally:

```
[ERROR 2026 (HY000): TLS/SSL error: Certificate verification failure: The certificate is NOT trusted.  
(base) vooora@Pranathis-MacBook-Air ~ % curl -XPOST "http://localhost:9000/2015-03-31/functions/function/invocations" -d '{}'  
[  
{"errorMessage": "'sessionState'", "errorType": "KeyError", "requestId": "a57efea6a-92e2-4a0b-95bc-14f7b1def797", "stackTrace": [" File \\"var/ta  
sk/lambda_function.py\\", line 6, in lambda_handler\n    user_question = event['sessionState']['intent']['slots'][UserQuestion]['value'][\"origin  
alValue\"]\n"]}  
    (base) vo/ba(base)(ba/ba/ba/base)(ba/base)(base) vo/base) vo/base) vo/ba/ba/ba/ba/base  
(base) vooora@Pranathis-MacBook-Air ~ % ]
```

Ecr:

AWS Lambda CloudWatch Aurora and RDS

Search [Option+S]

Pranathi Voora Europe (London)

[Amazon ECR](#) > [Private registry](#) > [Repositories](#)

Amazon Elastic Container Registry

- Private registry**
 - [Repositories](#)
 - [Features & Settings](#)
- Public registry**
 - [Repositories](#)
 - [Settings](#)

[ECR public gallery](#) [Amazon ECS](#) [Amazon EKS](#)

[Getting started](#) [Documentation](#)

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

| Private repositories (1) | | View push commands | Delete | Actions | Create repository | |
|---|------------------|--|------------------------|-------------------------------------|-----------------------------------|-----------------|
| <input type="text"/> Search by repository substring | | Repository name | URI | Created at | Tag immutability | Encryption type |
| <input type="radio"/> | lambda-container | 586794464928.dkr.ecr.eu-west-2.amazonaws.com/lambdacontainer | | April 24, 2025, 12:08:19 (UTC+05:5) | Mutable | AES-256 |

Create new lambda function as container image:

AWS Lambda CloudWatch Aurora and RDS

Search [Option+S]

Pranathi Voora Europe (London)

[Lambda](#) > [Functions](#) > [Create function](#)

Successfully created layer pydanticLayer version 1.

Create function Info

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Container image URI Info
The location of the container image to use for your function.

Browse images

Container image overrides

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

FINAL RESPONSE:

Test Draft version   

Last build submitted: 10 hours ago

Inspect

hi there!

i want to know about exam attendance

Which professor are you?

my name is Pranathi

Great! What do you want to know about exam attendance?

how many students attended the cloud computing exam

Please head over to this github repository and write your queries in the original value section of the lambda_function.py. Enjoy!

 Ready for complete testing

 Type a message

Session History

DynamoDB for user data and user session history

##ALREADY DONE

Monitor and Log

AWS CloudWatch to monitor and log data

- Captures logs from AWS Lex and AWS Lambda.
- Monitors system performance, error rates, and usage patterns.

