Serverless Quiz Application with AWS DynamoDB and Lambda

A Course Project Report Submitted in partial fulfillment of the course requirements for the award of grades in the subject of

CLOUD BASED AIML SPECIALITY (22SDCS07A)

by

Chandra Chinmayee

(2210030380)

Under the esteemed guidance of

Ms. P. Sree Lakshmi
Assistant Professor,
Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L Deemed to be UNIVERSITY

Aziznagar, Moinabad, Hyderabad,
Telangana, Pincode: 500075

April 2025

K L Deemed to be UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is Certified that the project entitled **"Serverless Quiz Application with AWS DynamoDB and Lambda"** which is an Experimental & Simulation work carried out by Chandra Chinmayee (2210030380), in partial fulfillment of the course requirements for the award of grades in the subject of **CLOUD BASED AIML SPECIALITY**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

Ms. P.Sree Lakshmi

Course Coordinator

Dr. Arpita Gupta

Head of the Department

Ms. P. Sree Lakshmi

Course Instructor

CONTENTS

1. INTRODUCTION

The Serverless Quiz Application is a modern, lightweight, and scalable web-based platform designed using a fully serverless architecture on Amazon Web Services (AWS). The primary goal of the application is to provide users with an interactive quiz experience without the need to manage traditional backend servers. The frontend is built using HTML, CSS, and JavaScript, and is deployed using an Amazon S3 bucket configured for static website hosting. On the backend, AWS Lambda functions are used to implement core functionalities such as retrieving quiz questions and storing user responses [1]. These Lambda functions operate in a serverless environment, which means they only run when triggered and automatically scale based on demand, resulting in efficient resource utilization and reduced costs.

Amazon DynamoDB, a fully managed NoSQL database service, is used for persistent storage. It holds all the quiz questions, options, correct answers, and user submissions, enabling fast read and write operations. To enable communication between the frontend and backend, Amazon API Gateway is employed [3]. IAM roles are configured to grant specific permissions required by Lambda to interact with DynamoDB and log events to CloudWatch for monitoring and debugging. By combining these AWS services, the application achieves high availability, seamless scalability, and minimal operational overhead and S3 is used to host a static web application. This architecture showcases the power and simplicity of serverless computing on AWS, allowing developers to focus solely on application logic and user experience while AWS handles infrastructure, scaling, and security in the background.

2. AWS SERVICES USED AS PART OF THE PROJECT

1. **Amazon S3(Simple Storage Service)**

   Amazon S3 is a scalable object storage service used for storing and retrieving any amount of data, commonly used for hosting static websites and serving frontend content.

- Hosting the frontend files (HTML, CSS, JS) of the quiz application.
- Serving the application directly from a public URL.
- Providing a static website hosting environment to users with high availability.

2. **Amazon DynamoDB**

   Amazon DynamoDB is a fully managed NoSQL database service that delivers high performance with low latency at any scale. It supports key-value and document data structures and provides built-in security, backup and restore, and in-memory caching [2].

- Recording user quiz submissions and responses.
- Acting as the backend database for all quiz-related data operations.

3. **AWS Lambda**

AWS Lambda is a serverless compute service that runs code in response to events, eliminating the need for server management.

- Executing backend logic to fetch quiz questions and handle submissions.
- Processing quiz evaluations and calculating user scores.
- Connecting the API Gateway to the database using event-driven execution.

4. **Amazon API Gateway**

Amazon API Gateway is a fully managed service that allows developers to create and publish secure APIs at scale. It acts as an interface for accessing backend services like AWS Lambda.

- Provides HTTP endpoints to trigger Lambda functions.
- Acts as a communication layer between the frontend and backend logic.

5. **AWS IAM (Identity and Access Management)**
AWS IAM is a service that helps you securely manage access to AWS services and resources. It lets you create roles, users, and policies to control who can do what in your AWS environment.

- Assigns permissions to Lambda functions so they can read from and write to DynamoDB securely.
- Ensures secure and controlled access to other AWS services such.
- Prevents unauthorized access and enhances the security posture of the serverless application.
- Attaches policies to enable specific actions.

3. STEPS INVOLVED IN SOLVING PROJECT PROBLEM STATEMENT

Understanding the Requirements

- Analyze the problem statement and define the core objectives of building a quiz application.
- Identify the appropriate AWS services such as Lambda, DynamoDB, API Gateway, IAM, and S3 for a serverless approach [6].

Designing the Architecture

- Design a serverless architecture that ensures scalability, flexibility, and cost-efficiency.
- Define data flow between the frontend, API Gateway, Lambda functions, and DynamoDB.
- Define scalability, security, and high availability requirements.

Setting Up AWS Environment

- Create IAM roles with policies like AmazonDynamoDBFullAccess and AWSLambdaBasicExecutionRole for secure access and logging.
- Set up DynamoDB tables to store quiz questions, options, and user responses.
- Set up an S3 bucket to host the frontend code (HTML, CSS, JS).

Developing & Configuring the Application

- Write Lambda functions to handle backend logic for fetching quiz questions and submitting user responses.
- Use Python in Lambda to process API requests and interact with DynamoDB.
- Configure API Gateway to expose endpoints to the frontend for invoking Lambda functions.

Deploying & Monitoring

- Deploy the frontend application to the S3 bucket as a static website.

- Deploy Lambda functions and test API Gateway integrations using the generated invoke URLs [7].

Testing & Optimization

- Test the quiz flow (question rendering, answer selection, submission, and score display).

Maintenance & Scaling

- Configure automatic scaling in DynamoDB (on-demand mode) to handle varying loads.
- Maintain application reliability and performance through monitoring and regular updates.

4. STEPWISE SCREENSHOTS WITH BRIEF DESCRIPTION

Step 1: Accessing AWS Management Console

- Open AWS Management Console.
- Log in with your Free Tier account.

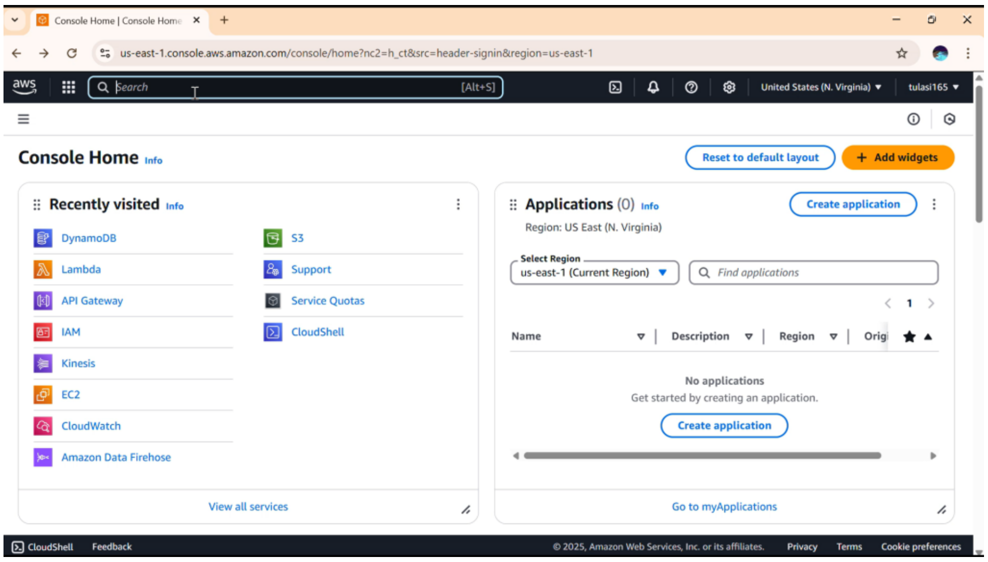


Fig. 4.1: Accessing AWS Management Console

Step 2: Open IAM Service

- Type IAM in search bar .
- Select IAM.

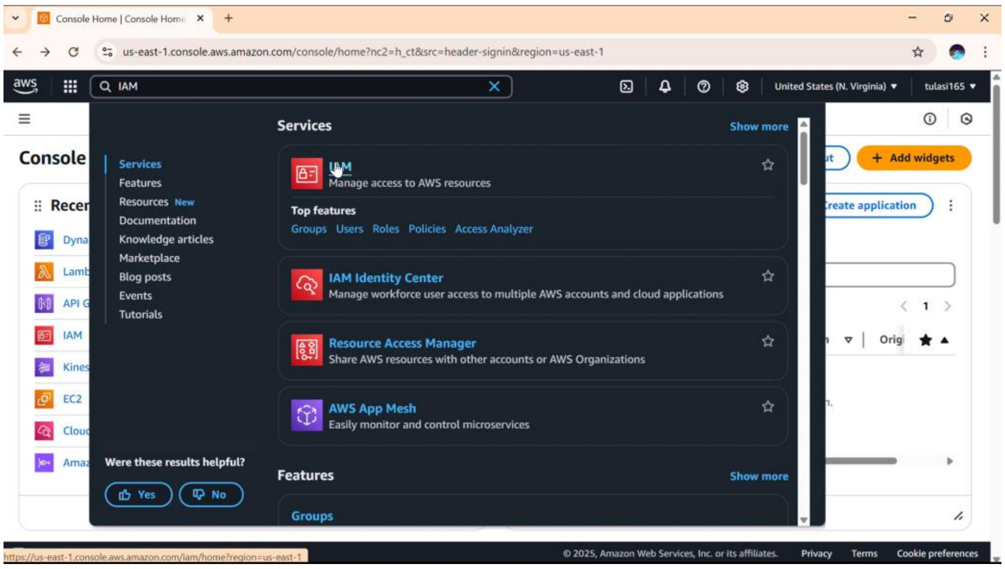
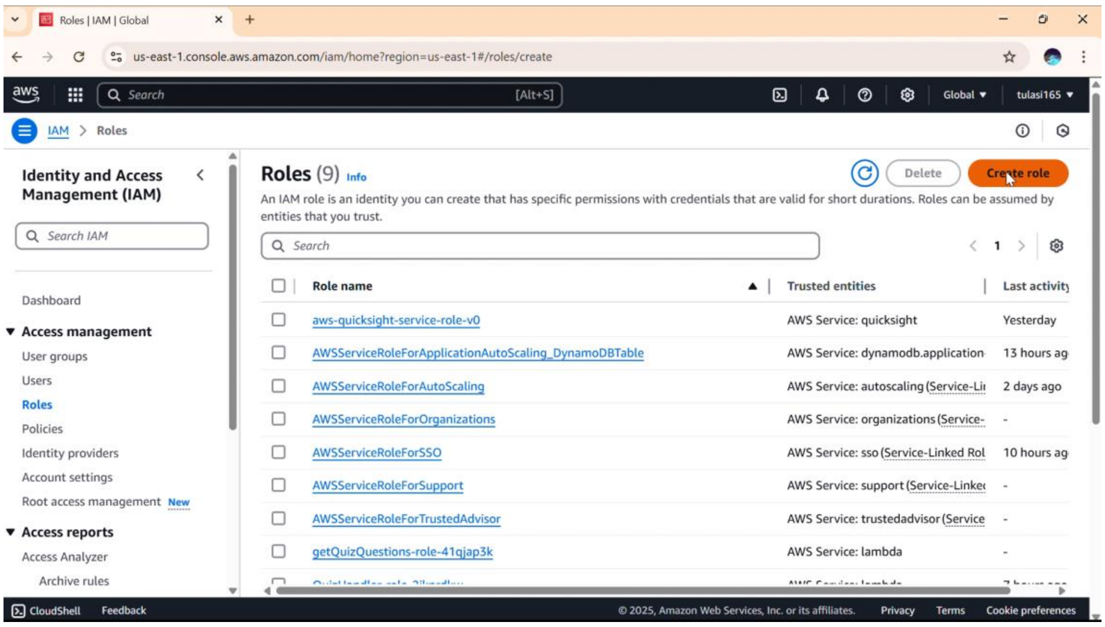

IAM
Services
Show more
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
IAM
Manage access to AWS resources
Top features
Groups
Users
Roles
Policies
Access Analyzer
IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications
Resource Access Manager
Share AWS resources with other accounts or AWS Organizations
AWS App Mesh
Easily monitor and control microservices
Were these results helpful?
Yes
No
Features
Show more
Groups
https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1

Fig. 4.2: Open IAM Service

Step 3: Creating a role.

- Go to Roles.
- Click on Create role


Roles | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create

IAM > Roles

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

Roles (9) Info — Delete — Create role

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.

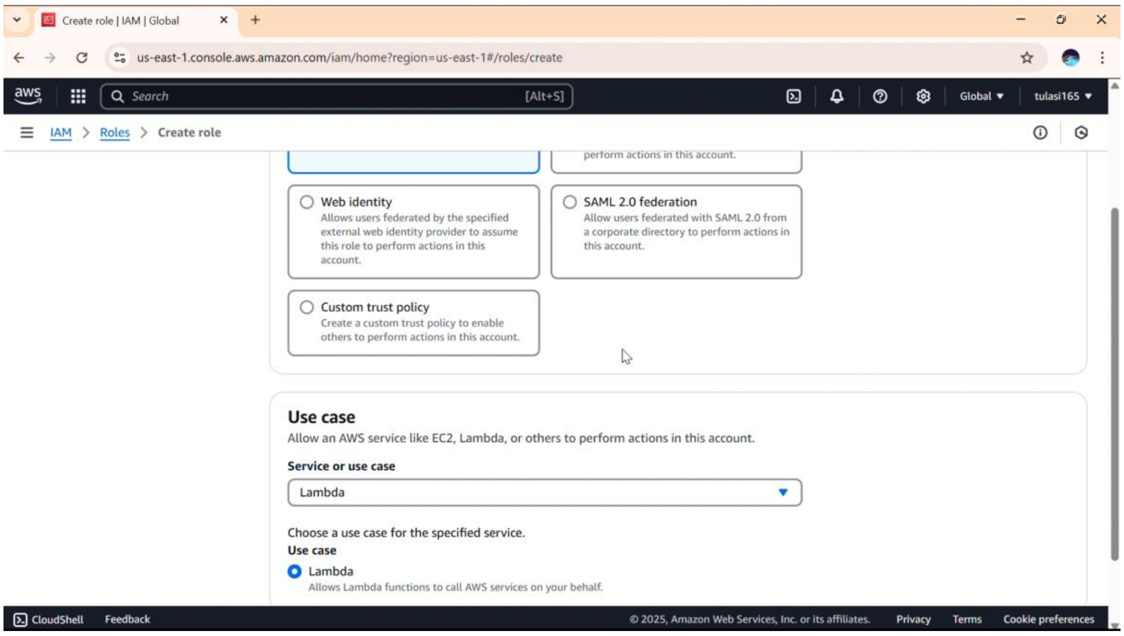
Role name	Trusted entities	Last activity
aws-quicksight-service-role-v0	AWS Service: quicksight	Yesterday
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application	13 hours ag
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	2 days ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-	-
AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Rol	10 hours ag
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
getQuizQuestions-role-41qjap3k	AWS Service: lambda	-

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

Fig. 4.3: Creating Role

Step 4: Selecting entity.

- Choose Trusted entity type as AWS Service.
- Choose Lambda as the Use case and click Next.


Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create

IAM > Roles > Create role

perform actions in this account.

Web identity — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy — Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: Lambda

Choose a use case for the specified service.

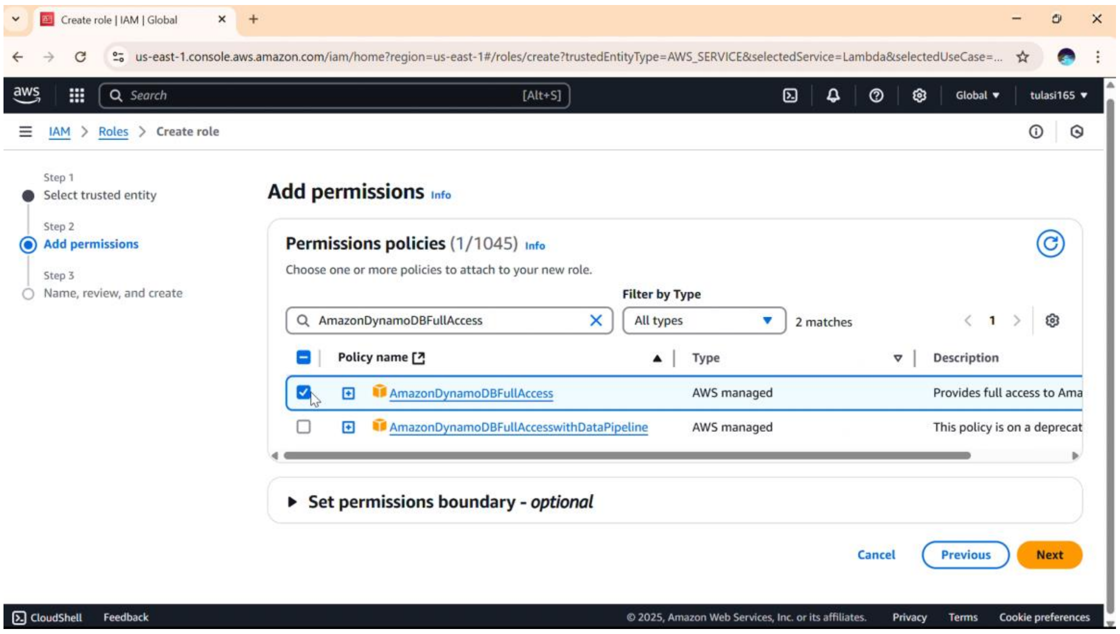
Use case: [✓ Lambda] — Allows Lambda functions to call AWS services on your behalf.

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

Fig. 4.4: Selecting entity

Step 5: Adding permissions to Lambda.

- Attach AmazonDynamoDBFullAccess (for DynamoDB operations)
- Attach AWSLambdaBasicExecutionRole (for logging in CloudWatch)
- Click on Next.



Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=Lambda&selectedUseCase=...

IAM > Roles > Create role

Step 1 Select trusted entity
Step 2 Add permissions
Step 3 Name, review, and create

Add permissions Info

Permissions policies (1/1045) Info

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

Filter by Type: All types — 2 matches

Search: AmazonDynamoDBFullAccess

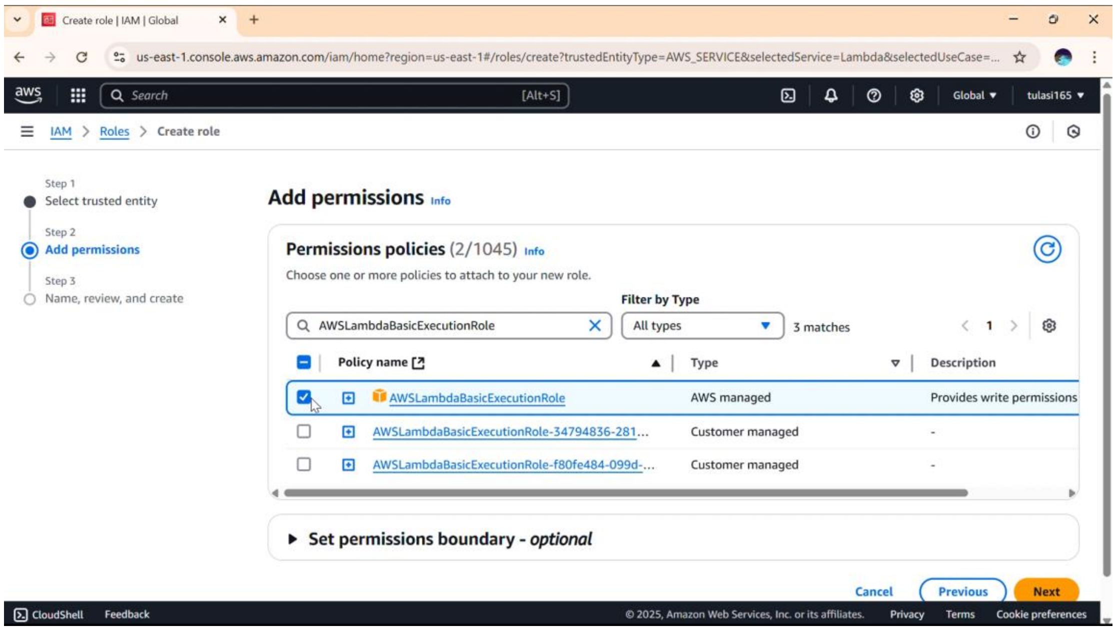
	Policy name	Type	Description
✓	AmazonDynamoDBFullAccess	AWS managed	Provides full access to Ama
	AmazonDynamoDBFullAccesswithDataPipeline	AWS managed	This policy is on a deprecat

▶ Set permissions boundary - *optional*

Cancel | Previous | Next

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

Fig. 4.5.1: Attach AmazonDynamoDBFullAccess



Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=Lambda&selectedUseCase=...

IAM > Roles > Create role

Step 1 Select trusted entity
Step 2 Add permissions
Step 3 Name, review, and create

Add permissions Info

Permissions policies (2/1045) Info

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

Filter by Type: All types — 3 matches

Search: AWSLambdaBasicExecutionRole

	Policy name	Type	Description
✓	AWSLambdaBasicExecutionRole	AWS managed	Provides write permissions
	AWSLambdaBasicExecutionRole-34794836-281...	Customer managed	-
	AWSLambdaBasicExecutionRole-f80fe484-099d-...	Customer managed	-

▶ Set permissions boundary - *optional*

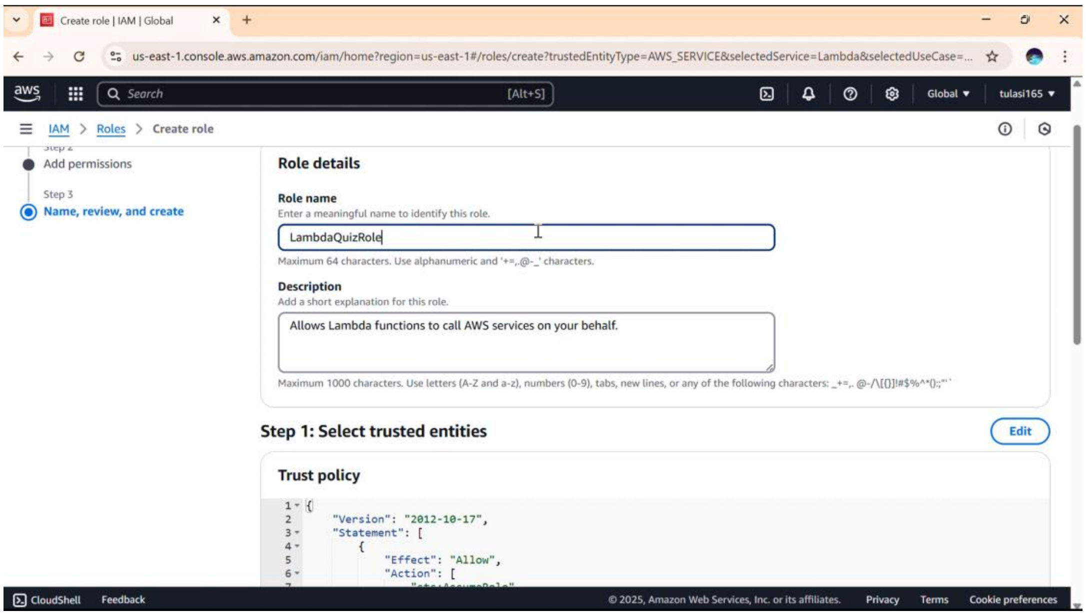
Cancel | Previous | Next

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

Fig. 4.5.2: Attach AWSLambdaBasicExecutionRole

Step 6: Filling Role Details.

- Name the role as LambdaQuizRole.
- Click Create Role.



Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=Lambda&selectedUseCase=...

IAM > Roles > Create role

Add permissions

Step 3
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
LambdaQuizRole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,.@-/\[{()}]!#$%^*();:"'

Step 1: Select trusted entities Edit

Trust policy

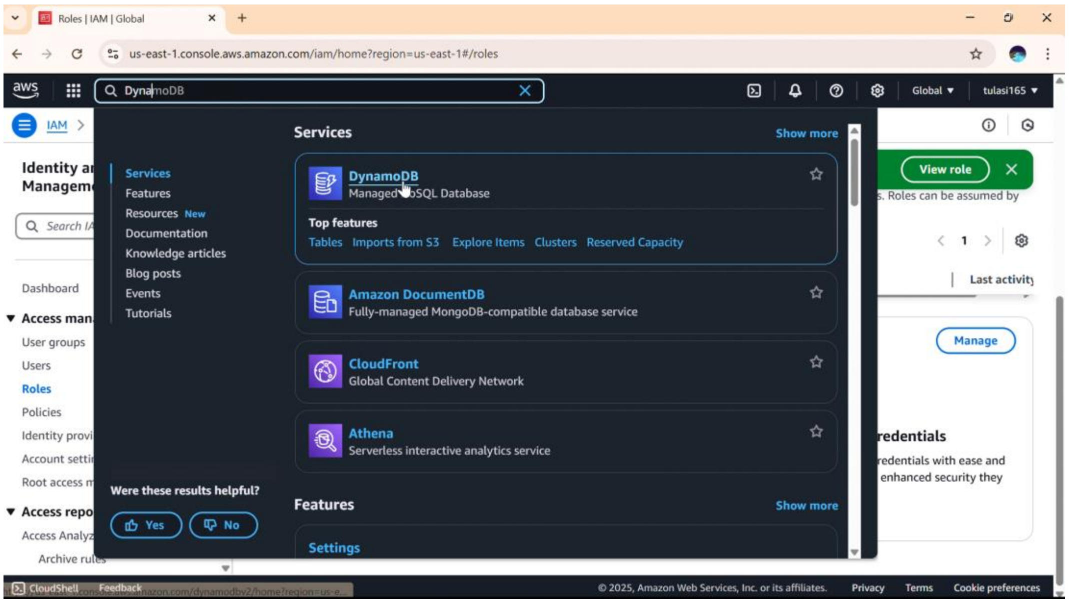
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

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

Fig. 4.6: Filling Role Details

Step 7: Opening DynamoDB.

- Open AWS Console.
- Choose DynamoDB



Roles | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles

DynamoDB

Services Show more

Services
Features
Resources New
Documentation
Knowledge articles
Blog posts
Events
Tutorials

DynamoDB
Managed NoSQL Database
Top features
Tables Imports from S3 Explore Items Clusters Reserved Capacity

Amazon DocumentDB
Fully-managed MongoDB-compatible database service

CloudFront
Global Content Delivery Network

Athena
Serverless interactive analytics service

Were these results helpful?
Yes No

Features Show more

Settings

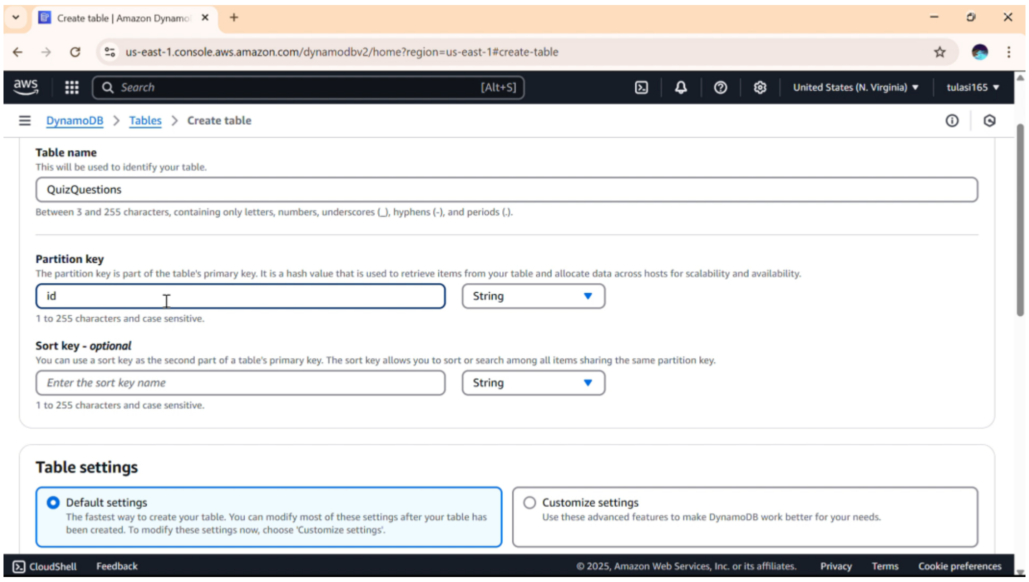
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Fig. 4.7: Opening DynamoDB

Step 8: Creating a table for storing quiz questions.

- Click on Create Table.
- Set: Table Name: QuizQuestions

  Primary Key: id (Type: String)



Create table | Amazon Dynamo

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table

DynamoDB > Tables > Create table

Table name
This will be used to identify your table.
QuizQuestions
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
id | String
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name | String
1 to 255 characters and case sensitive.

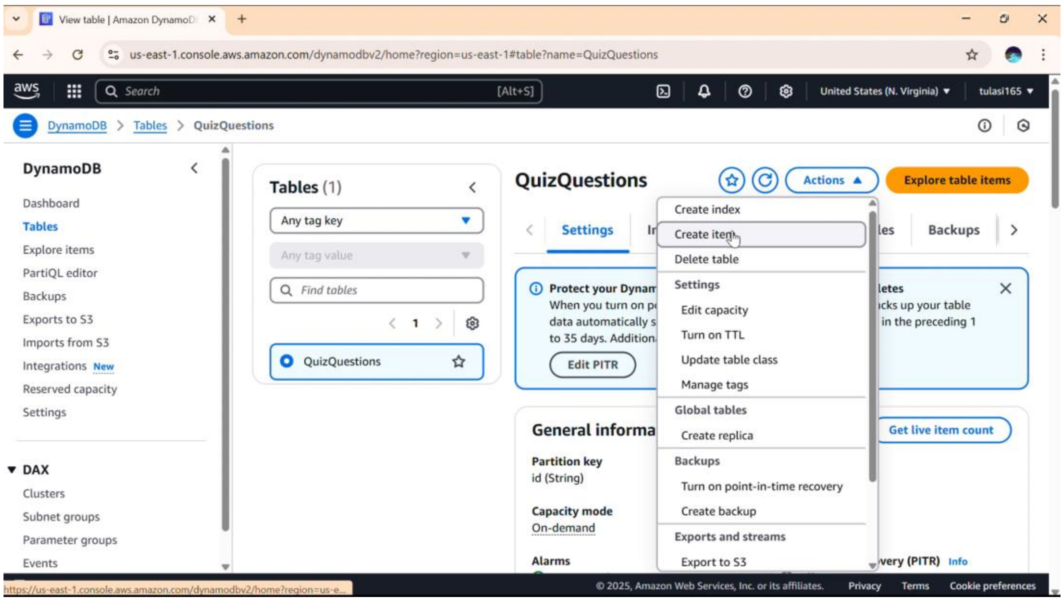
Table settings
[✓ Default settings] The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.
Customize settings — Use these advanced features to make DynamoDB work better for your needs.

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

Fig. 4.8: Creating a table for storing quiz questions

Step 9: Creating an Item in Table.

- Click Actions.
- Select Create Item



View table | Amazon DynamoDB

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#table?name=QuizQuestions

DynamoDB > Tables > QuizQuestions

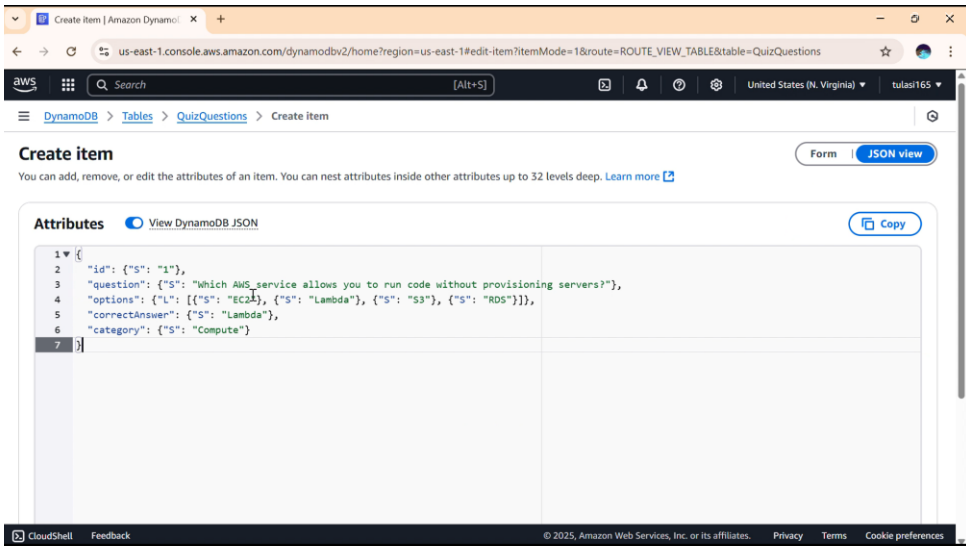
General informa...
Partition key: id (String)
Capacity mode: On-demand
Alarms

Fig. 4.9: Creating an Item in Table

Step 10: Inserting 10 questions.

- Choose JSON view and add any number of questions in the following format:

  {"id": "1",
   "question": "What is the default region for AWS CLI?",
   "options": ["us-east-1", "us-west-2", "ap-south-1", "eu-west-1"],
   "answer": "us-east-1"}



Create item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes — View DynamoDB JSON

```
{
  "id": {"S": "1"},
  "question": {"S": "Which AWS service allows you to run code without provisioning servers?"},
  "options": {"L": [{"S": "EC2"}, {"S": "Lambda"}, {"S": "S3"}, {"S": "RDS"}]},
  "correctAnswer": {"S": "Lambda"},
  "category": {"S": "Compute"}
}
```

Fig. 4.10: Inserting 10 questions

Step 11: In Search bar search and select Lambda.

- Select Lambda .
- Click on Create Function.

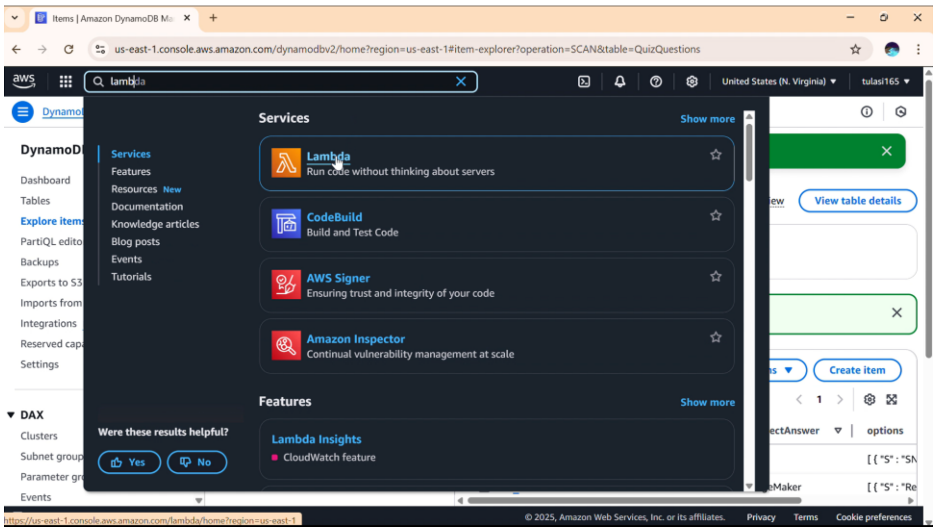
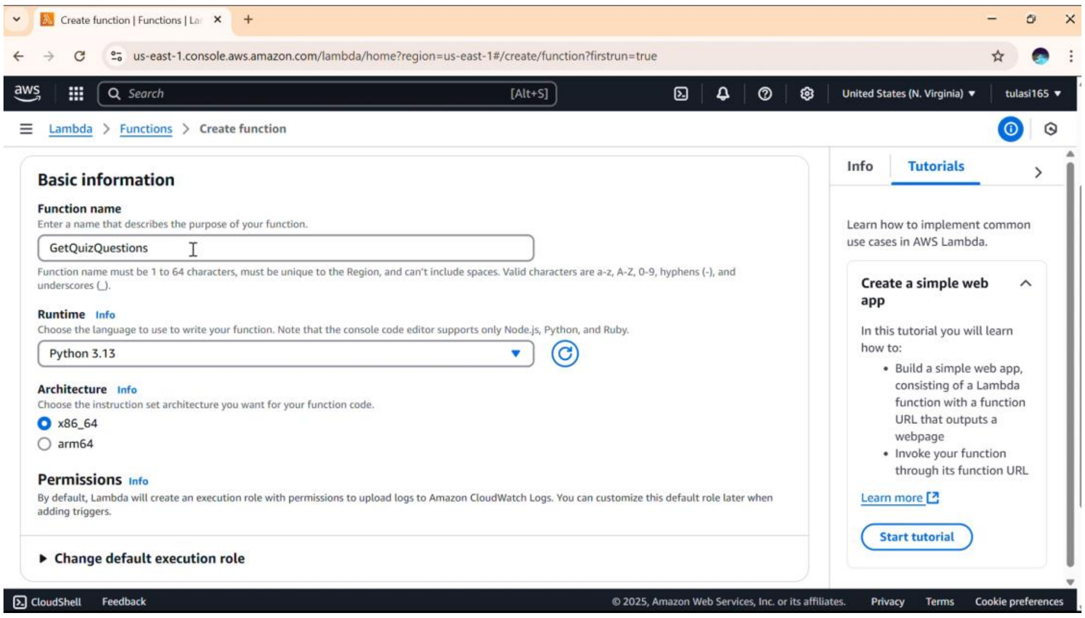


Fig. 4.11: In Search bar searching and selecting Lambda

Step 12: Filling Basic Information about the fucntion.

- Set Function name, Runtime and Execution Role.
- Click on Create Function.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true

Lambda > Functions > Create function

Basic information

Function name
Enter a name that describes the purpose of your function.
GetQuizQuestions
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.
Python 3.13

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

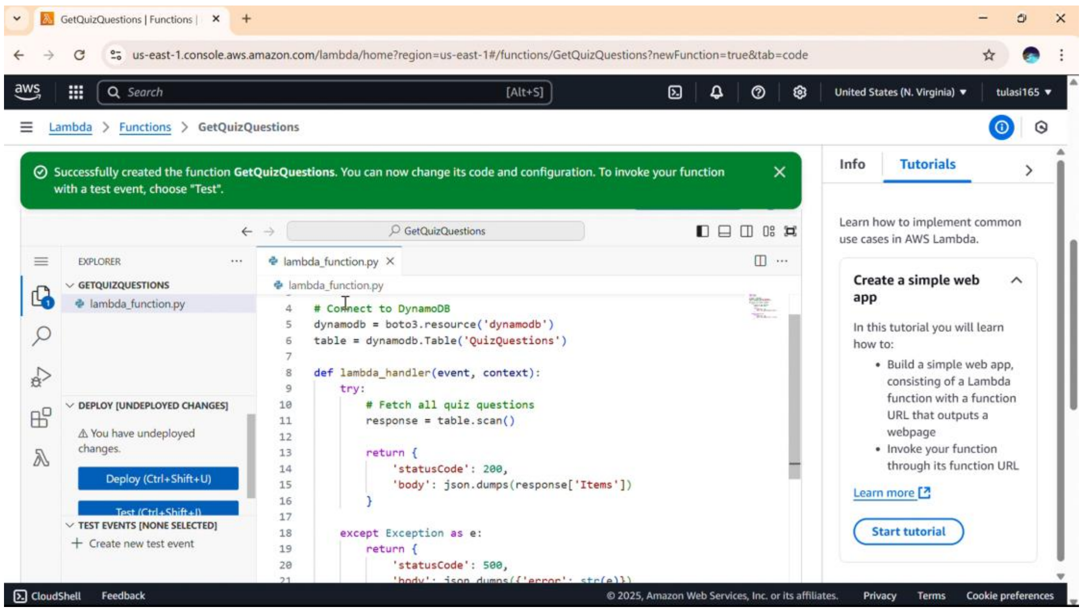
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▸ Change default execution role

Fig. 4.12: Filling Basic Information about the fucntion.

Step 13: Writing Lambda code to Fetch Questions.

- Write the code in Lambda Function.
- Click on Deploy.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/GetQuizQuestions?newFunction=true&tab=code

Lambda > Functions > GetQuizQuestions

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

lambda_function.py

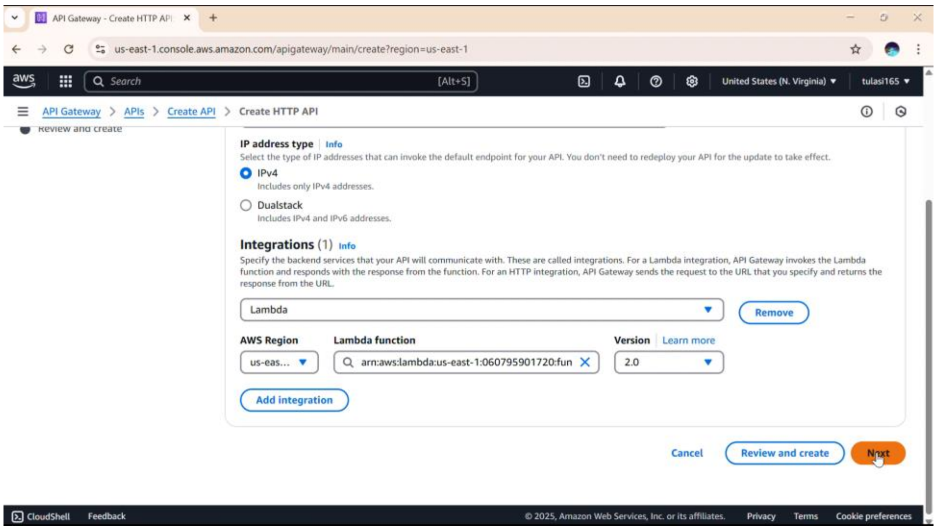
```
# Connect to DynamoDB
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('QuizQuestions')

def lambda_handler(event, context):
    try:
        # Fetch all quiz questions
        response = table.scan()

        return {
            'statusCode': 200,
            'body': json.dumps(response['Items'])
        }

    except Exception as e:
        return {
            'statusCode': 500,
            'body': json.dumps({'error': str(e)})
```

DEPLOY [UNDEPLOYED CHANGES]
You have undeployed changes.
Deploy (Ctrl+Shift+U)

Fig. 4.13: Writing Lambda code to Fetch Questions

Step 14: Create an API for GetQuizQuestion Lambda function.

- Click Create API then HTTP API. Next click Build.
- Set Name as Quiz API, in Integration add Lambda Function, then choose GetQuizQuestions function and click Next. Click Create.
- Configure routes.



API Gateway - Create HTTP API

us-east-1.console.aws.amazon.com/apigateway/main/create?region=us-east-1

API Gateway > APIs > Create API > Create HTTP API

Review and create

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.

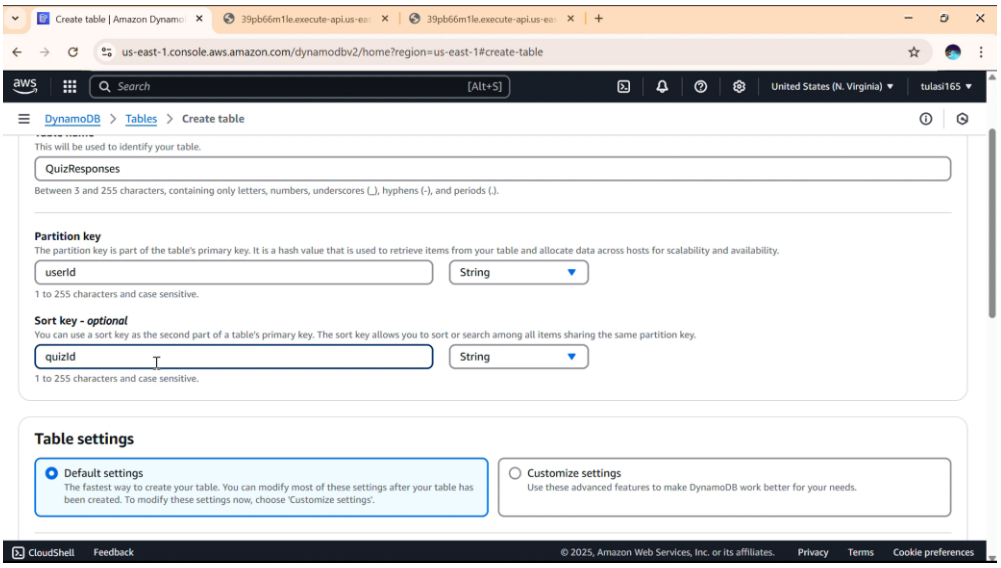
Lambda

AWS Region	Lambda function	Version
us-eas...	arn:aws:lambda:us-east-1:060795901720:fun	2.0

Fig. 4.14: Creating an API for GetQuizQuestion Lambda function

Step 15: Create a new table in DynamoDB for storing user responses.

- Create a new table in DynamoDB with name QuizResponses with userId as Partition Key and quizId as the Sort key and click Create Table.



Create table | Amazon DynamoDB

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table

DynamoDB > Tables > Create table

This will be used to identify your table.

QuizResponses

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

userId — String

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

quizId — String

1 to 255 characters and case sensitive.

Table settings

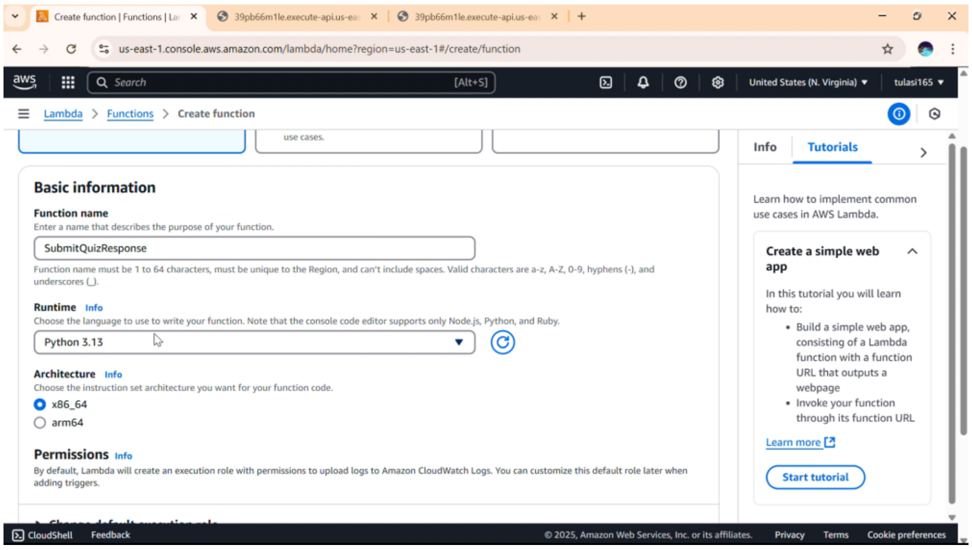
[✓ Default settings] The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

Customize settings — Use these advanced features to make DynamoDB work better for your needs.

Fig. 4.15: Creating a new table in DynamoDB for storing user responses.

Step 16: Create a new lambda function to save quiz responses.

- Open AWS Lambda and create a new function. Name it SubmitQuestions.
- Update the Lambda function to save quiz responses. Click Deploy.
- Set Permissions.
- Go to API Gateway and select API QuizAPI and create a route with POST method.


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function

Lambda > Functions > Create function

Basic information

Function name
Enter a name that describes the purpose of your function.
SubmitQuizResponse
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.
Python 3.13

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.

Info | Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

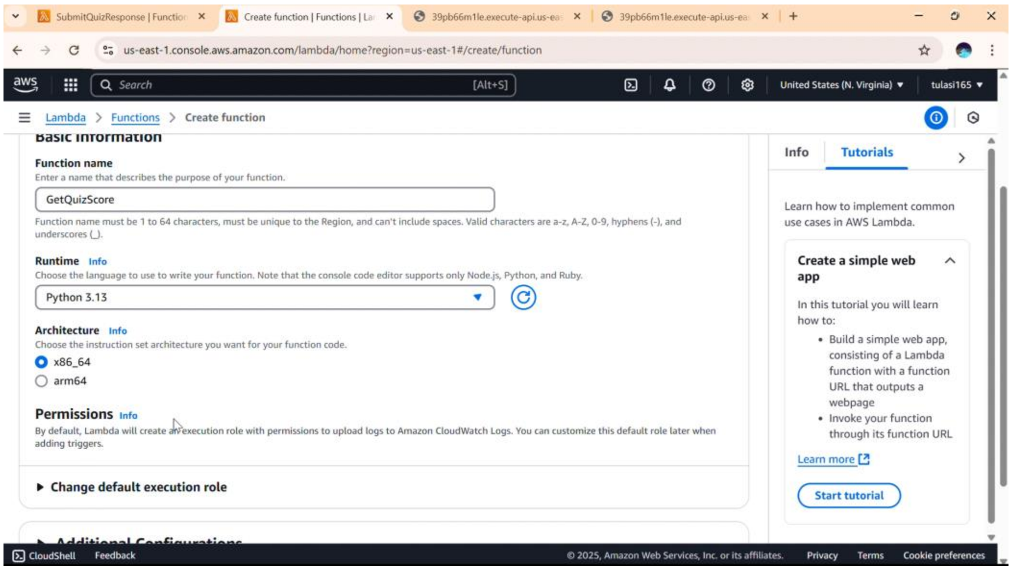
Start tutorial

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

Fig. 4.16: Creating a new lambda function to save quiz responses.

Step 17: Create a new lambda function to return score from DynamoDB.

- Change the lambda function which returns the score from DynamoDB.
- Give IAM Permissions, go to API Gateway and create an API with Get method.


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function

Lambda > Functions > Create function

Basic information

Function name
Enter a name that describes the purpose of your function.
GetQuizScore
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.
Python 3.13

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

Info | Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

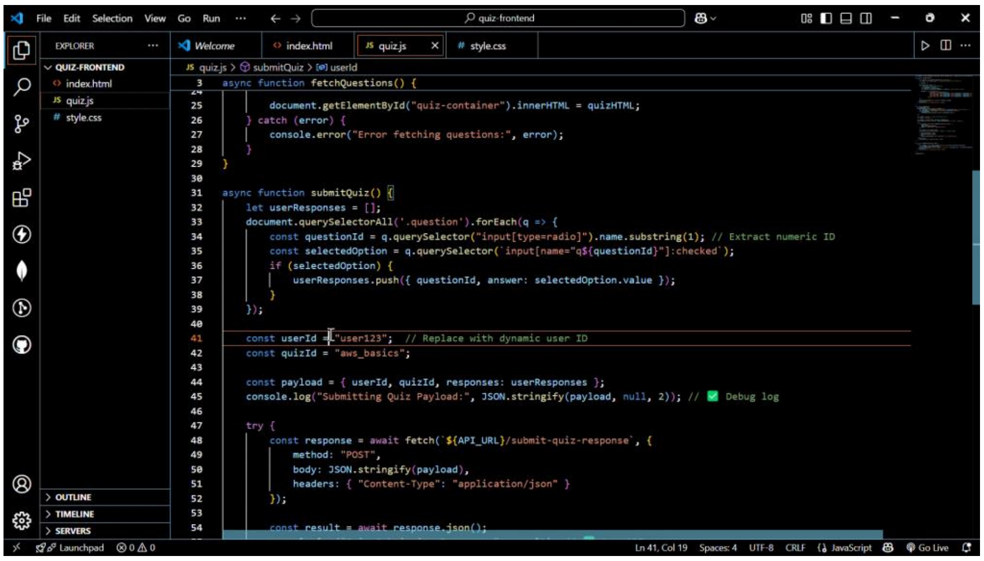
Start tutorial

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

Fig. 4.17: Creating a new lambda function to return score from DynamoDB

Step 18: Develop frontend for the serverless quiz application.

- Write code for frontend.
- Create index.html, style.css and quiz.js.
- Upload these codes in S3 and copy the public URL.
- Paste the URL in a new tab, the quiz application is available.

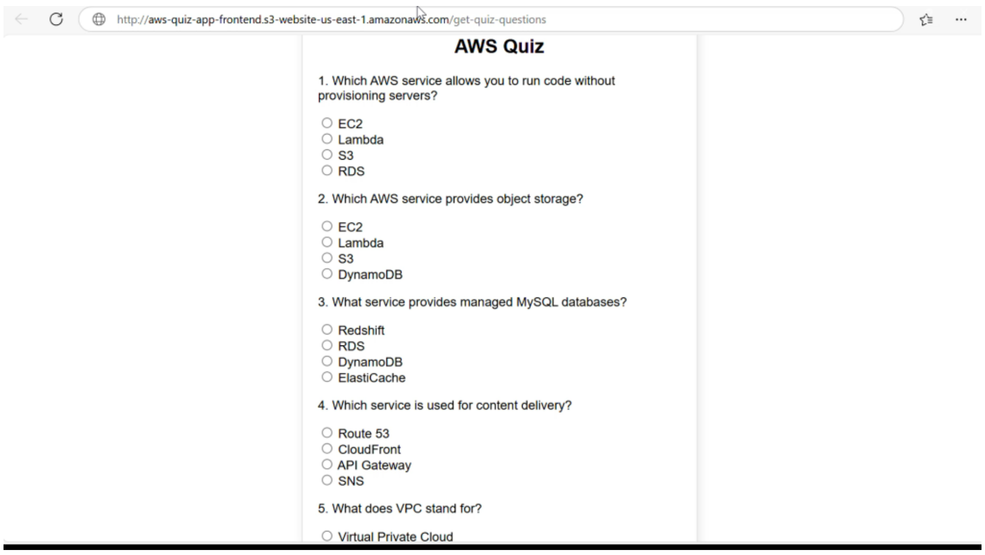


```
async function fetchQuestions() {
        document.getElementById("quiz-container").innerHTML = quizHTML;
    } catch (error) {
        console.error("Error fetching questions:", error);
    }
}

async function submitQuiz() {
    let userResponses = [];
    document.querySelectorAll('.question').forEach(q => {
        const questionId = q.querySelector("input[type=radio]").name.substring(1); // Extract numeric ID
        const selectedOption = q.querySelector(`input[name="q${questionId}"]:checked`);
        if (selectedOption) {
            userResponses.push({ questionId, answer: selectedOption.value });
        }
    });

    const userId = "user123";  // Replace with dynamic user ID
    const quizId = "aws_basics";

    const payload = { userId, quizId, responses: userResponses };
    console.log("Submitting Quiz Payload:", JSON.stringify(payload, null, 2)); // ✅ Debug log

    try {
        const response = await fetch(`${API_URL}/submit-quiz-response`, {
            method: "POST",
            body: JSON.stringify(payload),
            headers: { "Content-Type": "application/json" }
        });

        const result = await response.json();
```

Fig. 4.18: Developing frontend

Step 19: Output

- Choose the answers for the questions.
- Click on Submit.


http://aws-quiz-app-frontend.s3-website-us-east-1.amazonaws.com/get-quiz-questions

AWS Quiz

1. Which AWS service allows you to run code without provisioning servers?

○ EC2
○ Lambda
○ S3
○ RDS

2. Which AWS service provides object storage?

○ EC2
○ Lambda
○ S3
○ DynamoDB

3. What service provides managed MySQL databases?

○ Redshift
○ RDS
○ DynamoDB
○ ElastiCache

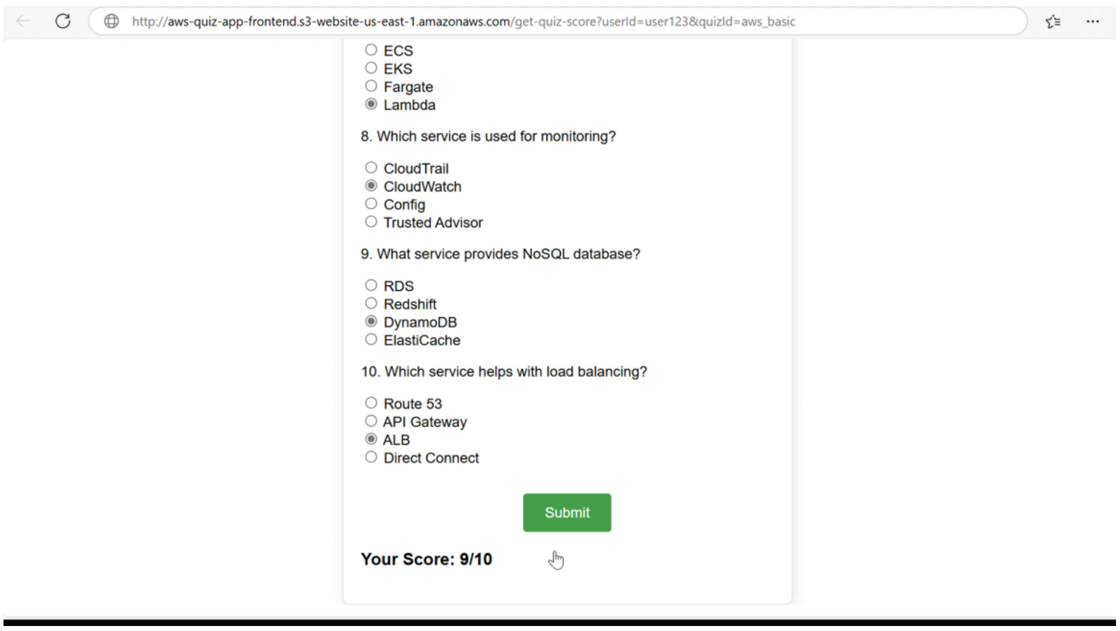
4. Which service is used for content delivery?

○ Route 53
○ CloudFront
○ API Gateway
○ SNS

5. What does VPC stand for?

○ Virtual Private Cloud

Fig. 4.19.1: Serverless Quiz Application


http://aws-quiz-app-frontend.s3-website-us-east-1.amazonaws.com/get-quiz-score?userId=user123&quizId=aws_basic

- ECS
- EKS
- Fargate
- [✓ Lambda]

8. Which service is used for monitoring?

- CloudTrail
- [✓ CloudWatch]
- Config
- Trusted Advisor

9. What service provides NoSQL database?

- RDS
- Redshift
- [✓ DynamoDB]
- ElastiCache

10. Which service helps with load balancing?

- Route 53
- API Gateway
- [✓ ALB]
- Direct Connect

Your Score: 9/10

Fig. 4.19.2: Viewing Score

5. LEARNING OUTCOMES

Understanding of Serverless Architecture

Gained hands-on experience with building and deploying a serverless application, eliminating the need to manage underlying infrastructure.

Proficiency with AWS Services

Learned to effectively use core AWS services such as Lambda, DynamoDB, API Gateway, IAM, and S3 for building scalable and secure applications [5].

Database Management with DynamoDB

Understood how to design and manage NoSQL databases, store structured data like quiz questions and user responses, and perform efficient read/write operations.

Function-as-a-Service Development with AWS Lambda

Gained knowledge of writing backend logic using AWS Lambda, handling stateless request processing, and integrating with other AWS components.

API Development and Integration

Learned to expose backend services securely using Amazon API Gateway and integrate APIs with frontend applications.

Frontend Deployment on AWS S3

Experienced deploying a static website on Amazon S3 and connecting it with backend APIs for a full-stack solution [4].

Security and Role Management using IAM

Developed an understanding of configuring IAM roles and policies to securely control access to AWS services.

6. CONCLUSION

The Serverless Quiz Application project demonstrates the power and efficiency of cloud-native architecture by leveraging AWS services like Lambda, DynamoDB, S3 and API Gateway to build a fully functional, scalable, and cost-effective quiz platform. This architecture eliminates the need for server management, ensuring high availability and seamless performance under varying loads. The application effectively stores and retrieves quiz data from DynamoDB, processes user inputs through Lambda functions, and offers a smooth, responsive interface. Overall, the project showcases how serverless computing can simplify backend logic while providing a reliable and maintainable solution for dynamic web applications.

7. FUTURE ENHANCEMENTS

Future enhancements for the Serverless Quiz Application can significantly improve user experience, engagement, and scalability. Converting the app into a Progressive Web App (PWA) would enable offline access, while incorporating multilingual support and gamification elements like badges and rewards would make the application more inclusive and engaging. Advanced analytics dashboards and voice-enabled quiz features could further elevate the platform, offering insightful user metrics and improved accessibility.

8. REFERENCES

[1] Amazon Web Services (AWS). "Getting started with AWS Lambda."
https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html

[2] Amazon Web Services (AWS). "What is Amazon DynamoDB?"
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

[3] Amazon Web Services (AWS). "Using API Gateway and AWS Lambda for Serverless Applications."
https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

[4] Amazon Web Services (AWS). "Hosting a Static Website on Amazon S3."
https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html

[5] Amazon Web Services (AWS). "Getting Started with AWS Identity and Access Management (IAM)."
https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started.html

[6] Amazon Web Services (AWS). "Tutorial: Build a Serverless Web Application."
https://docs.aws.amazon.com/lambda/latest/dg/building-serverless-apps.html

[7] Amazon Web Services (AWS). "Tutorial: Creating API as an Amazon S3 Proxy in API Gateway."
https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started.html