

😊 Secure your API Gateway with Amazon Cognito User Pools

😊 Step1: Create Lambda Function and API Gateway

1. Now go to Lambda and click on the Create function. Now you have to give your function a name choose runtime as Python 3.11 and create your function.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The first step, 'Basic information', is selected. It includes three options for creating the 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.

Below these options, the 'Function name' field is filled with 'demo-ecommerce-function'. A note says to use only letters, numbers, hyphens, or underscores with no spaces. The 'Runtime' dropdown is set to 'Python 3.11'. The 'Architecture' dropdown is set to 'x86_64'.

2. Then come inside of your function and click on the Add trigger.

The screenshot shows the 'Function overview' tab for the 'demo-ecommerce-function' Lambda function. The function is currently empty, indicated by '(0)' next to the 'Layers' section. At the bottom left, there is a red box highlighting the '+ Add trigger' button, which is used to add triggers to the function.

3. Now in the trigger configuration you have to choose API gateway, then choose to create a new API, choose Rest API and keep security to Open. Then click on Add.

Trigger configuration [Info](#)

API Gateway [aws](#) [api](#) [application-services](#) [backend](#) [HTTP](#) [REST](#) [serverless](#)

Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. [Learn more](#)

Intent
Use an existing api or have us create one for you.

Create a new API
 Use existing API

API type

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

REST API
Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Security
Configure the security mechanism for your API endpoint.

[Open](#)

► Additional settings

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

4. In the configuration tab you can see that our API gateway has been created and added as a trigger to our lambda function. Now if you click on the API endpoint then you will get a hello from lambda.

Code	Test	Monitor	Configuration	Aliases	Versions						
General configuration <ul style="list-style-type: none"> Triggers Permissions Destinations Function URL Environment variables Tags <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Triggers (1) Info <div style="display: flex; justify-content: space-between;"> <input type="button" value="C"/> Fix errors Edit Delete Add trigger </div> <div style="margin-top: 5px;"> <input type="text" value="Find triggers"/> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10px;"></th> <th style="width: 10px;">Trigger</th> <th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input type="checkbox"/></td> <td> API Gateway: demo-ecommerce-function-API arn:aws:execute-api:ap-south-1:878893308172:qhgg71wjff/*/*demo-ecommerce-function API endpoint: https://qhgg71wjff.execute-api.ap-south-1.amazonaws.com/default/demo-ecommerce-function </td> </tr> </tbody> </table> </div>							Trigger		<input type="checkbox"/>	<input type="checkbox"/>	API Gateway: demo-ecommerce-function-API arn:aws:execute-api:ap-south-1:878893308172:qhgg71wjff/*/*demo-ecommerce-function API endpoint: https://qhgg71wjff.execute-api.ap-south-1.amazonaws.com/default/demo-ecommerce-function
	Trigger										
<input type="checkbox"/>	<input type="checkbox"/>	API Gateway: demo-ecommerce-function-API arn:aws:execute-api:ap-south-1:878893308172:qhgg71wjff/*/*demo-ecommerce-function API endpoint: https://qhgg71wjff.execute-api.ap-south-1.amazonaws.com/default/demo-ecommerce-function									

Pretty-print

"Hello from Lambda!"

Step 2: Create Cognito User Pool

1. Now navigate to Cognito and click on Create User Pool. In the step 1 choose Email as your sign-in option.

Authentication providers
Configure the providers that are available to users when they sign in.

Provider types
Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. [Learn more about pricing](#)

Cognito user pool
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

Federated identity providers
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

Cognito user pool sign-in options | [Info](#)
Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

User name
 Email
 Phone number

 Cognito user pool sign-in options can't be changed after the user pool has been created.

2. Then in step 2 you have to choose custom for password policy mode and set the password minimum length to 6 characters. After that disable all password requirements and scroll down.

Password policy [Info](#)

Create a password policy to define the length and complexity of the passwords your users can set.

Password policy mode | [Info](#)

Cognito defaults

Use default password requirements.

Custom

Use password requirements that you define.

Password minimum length

6

character(s)

Must be a number between 6 and 99. We strongly recommend that you require passwords to be at least 8 characters in length.

Password requirements

- Contains at least 1 number
- Contains at least 1 special character
- Contains at least 1 uppercase letter
- Contains at least 1 lowercase letter

Temporary passwords set by administrators expire in

7

day(s)

Must be a number between 0 and 365.

3. Now say no to multi-factor authentication but enable user account recovery as shown below and click on next.

Multi-factor authentication

Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

MFA enforcement | [Info](#)

Require MFA -

Recommended

Users must provide an additional authentication factor when signing in.

Optional MFA

Users can sign in with a single authentication factor, and can choose to add additional authentication factors.

No MFA

Users can only sign in with a single authentication factor. This is the least secure option.

User account recovery

Configure how users will recover their account when they forget their password. Recipient message and data rates apply.

Self-service account recovery | [Info](#)

Enable self-service account recovery - Recommended

Allow forgot-password operations in your user pool. In the hosted UI sign-in page, a "Forgot your password?" link is displayed. When this feature is not enabled, administrators reset passwords with the Cognito API.

Delivery method for user account recovery messages | [Info](#)

Select how your user pool will deliver messages when users request an account recovery code. SMS messages are charged separately by Amazon SNS. Email messages are charged separately by Amazon SES. [Learn more about pricing](#).

Email only

SMS only

Email if available, otherwise SMS

SMS if available, otherwise email

SMS if available, otherwise email, and allow a user to reset their password via SMS if they are also using it for MFA

[Cancel](#)

[Previous](#)

[Next](#)

4. So, for step 3 you have to keep this step to default settings and move to the next step.

[Amazon Cognito](#) > [User pools](#) > [Create user pool](#)

Step 1

[Configure sign-in experience](#)

Step 2

[Configure security requirements](#)

Step 3

[Configure sign-up experience](#)

Step 4

[Configure message delivery](#)

Step 5

[Integrate your app](#)

Step 6

[Review and create](#)

Configure sign-up experience [Info](#)

Determine how new users will verify their identities when signing up and which attributes should be required or optional during the user sign-up flow.

Self-service sign-up [Info](#)

Choose whether new users of your app can register for an account themselves.

[Self-registration](#) | [Info](#)

Enable self-registration

Display a "Sign up" link on the sign-in page in the hosted UI, and allow the use of public APIs to create new user accounts. When this feature is not enabled, federation and administrative API operations create user profiles.

ⓘ If you activate user sign-up in your user pool, anyone on the internet can sign up for an account and sign in to your apps. Don't enable self-registration in your user pool until you want to open your app to public sign-up.

Attribute verification and user account confirmation

Choose between Cognito-assisted and self-managed user attribute verification and account confirmation. Only verified attributes can be used for sign-in, account recovery, and MFA. A user account must be confirmed either by attribute verification, or user pool administrator confirmation, before a user is allowed to sign in.

5. Now for step 4 choose Send email with Cognito and click on next.

Email

Configure how your user pool sends email messages to users.

Email provider | [Info](#)

Send email with Amazon SES - Recommended
Send emails using an Amazon SES verified identity in your account. We recommend this option for higher email volume and production workloads.

Send email with Cognito
Use Cognito's default email address as a temporary start for development. You can use it to send up to 50 emails a day.

You must have configured a verified sender with [Amazon SES](#) to use the SES feature. [Learn more](#)

SES Region [Info](#)
Asia Pacific (Mumbai)

FROM email address | [Info](#)
By default "no-reply@verificationemail.com" will be used. You can also choose a different email address that you have previously verified with Amazon SES.

▼

REPLY-TO email address - *optional* | [Info](#)
If you set an invalid reply-to address, sending restrictions may be imposed on your account.

Cancel
Next

6. Now on step 5, first you should give a user pool name and enable the Cognito Hosted UI.

User pool name

Create a friendly name for your user pool.

User pool name

User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = . @ -

Your user pool name can't be changed once this user pool is created.

Hosted authentication pages

Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.

Use the Cognito Hosted UI
Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

7. Then for the domain choose use a Cognito Domain and give a unique domain name.

Domain [Info](#)

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. To use the Hosted UI, you must choose a domain where authentication endpoints will be created.

Domain type

Use a Cognito domain

Enter an identifying prefix to use in an Amazon-owned domain. For production apps, we recommend using a custom domain instead.

Use a custom domain

Enter a domain that you own for Cognito-hosted sign-up and sign-in pages. You must provide a DNS record and an AWS Certificate Manager (ACM) certificate to use a custom domain. We recommend using a custom domain for production workloads.

Cognito domain

Enter a domain prefix.

.auth.ap-south-1.amazonaws.com

Domain prefixes may only include lowercase, alphanumeric characters, and hyphens. You can't use the text aws, amazon, or cognito in the domain prefix. Your domain prefix must be unique within the current Region.

Available

- Now in the app type choose public client and give an app client name. Then disable client's secret.

Initial app client

Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

App type [Info](#)

Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

Public client

A native, browser or mobile-device app. Cognito API requests are made from user systems that are not trusted with a client secret.

Confidential client

A server-side application that can securely store a client secret. Cognito API requests are made from a central server.

Other

A custom app. Choose your own grant, auth flow, and client-secret settings.

App client name [Info](#)

Enter a friendly name for your app client.

App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Client secret [Info](#)

Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API requests. Using a client secret can prevent a third party from impersonating your client.

Generate a client secret

Don't generate a client secret

- Then in the allowed callback URLs give a URL and then expand the advanced app client settings and check in the authentication flows, these option must be selected.

Allowed callback URLs | [Info](#)

Enter at least one callback URL to redirect the user back to after authentication. This is typically the URL for the app receiving the authorization code issued by Cognito. You may use HTTPS URLs, as well as custom URL schemes.

URL

<https://new-app.com/callback>

[Remove](#)

Length of callback URL must be between 1 and 1024 characters. Valid characters are letters, marks, numbers, symbols, and punctuations. Amazon Cognito requires HTTPS over HTTP except for <http://localhost> for testing purposes only. App callback URLs such as <myapp://example> are also supported. Must not contain a fragment.

[Add another URL](#)

You can add 99 more URLs

▼ Advanced app client settings

We have populated suggested authentication flows, OAuth 2.0 grant types, and OIDC scopes based on the selections you made earlier.

Authentication flows | [Info](#)

Choose authentication flows that your app will support. Refresh token authentication is always enabled. We have populated options based on your app type.

[Select authentication flows](#)

ALLOW_REFRESH_TOKEN_AUTH [X](#)
Refresh token based authentication

ALLOW_USER_SR_P_AUTH [X](#)
SRP (secure remote password) protocol based authentication

ALLOW_CUSTOM_AUTH [X](#)
Lambda trigger based custom authentication

10. After that scroll down to OAuth 2.0 grant types and here you have to add implicit grant.
11. Then just go ahead to the review page and create your user pool.

OAuth 2.0 grant types | [Info](#)

Choose at least one OAuth grant type to configure how Cognito will deliver tokens to this app. We have populated suggested options based on the app type you selected.

[Select OAuth 2.0 grant types](#)

Authorization code grant [X](#)
Provides an authorization code as the response

Implicit grant [X](#)
Specifies that the client should get the access token (and, optionally, ID token, based on scopes) directly

⚠ The implicit grant flow exposes OAuth tokens in the url. We recommend that you use only the authorization code flow with PKCE for public clients.

Step 3: Create Authorizer in API Gateway

1. Now go to API gateway, open your API which was created using lambda and go to authorizer from the left pane.

The screenshot shows the AWS API Gateway interface. On the left, a sidebar lists various API settings like APIs, Custom domain names, VPC links, and stages. Under the 'demo-ecommerce-function-API' stage, the 'Authorizers' link is highlighted with a red box. The main panel displays the 'Resources' section for the root path '/'. It shows a single resource entry for '/demo-ecommerce-function' with the method 'ANY'. The 'Resource details' pane shows the path '/' and resource ID 'pdttagwflg4'. The 'Methods (0)' pane indicates 'No methods defined.'

- Now click on Create Authorizer. Then you have to give it a name and in the authorizer type choose Cognito, choose your user pool and in the token source write Authorization.

The screenshot shows the 'Authorizers' page for the 'demo-ecommerce-function-API'. It displays a message 'No authorizers' and 'This API does not have any authorizers'. Below this is a 'Create an authorizer' button. At the top right, there are 'Edit', 'Delete', and 'Create authorizer' buttons. The 'Create authorizer' button is highlighted with a red box.

Create authorizer Info

Authorizer details

Authorizer name
new-authorizer

Authorizer type Info
Choose to authorize your API calls using one of your Lambda functions or a Cognito User Pool.
 Lambda
 Cognito

Cognito user pool
Select the Cognito user pool that will authenticate requests to your API.
ap-south-1 ▼ X

Token source
Enter the header that contains the authorization token.
Authorization

Token validation - optional
Enter a regular expression to validate tokens.

Cancel **Create authorizer**

3. Now from the left pane choose Resources come to this ANY Method and go to method request here you can see that the authorization is set to none. So, now we need to click on edit.

The screenshot shows the AWS Lambda API resource configuration. On the left, under 'Create resource', there's a tree view with a root node 'demo-ecommerce-function' containing an 'ANY' method. The 'Method request' tab is selected in the top navigation bar. The 'Method request settings' panel shows the following configuration:

Setting	Value
Authorization	NONE
Request validator	None
API key required	False
SDK operation name	Generated based on method and path

At the bottom of the settings panel, there is a 'Request paths (0)' section with navigation arrows.

4. Then in the authorization choose your authorizer and in the authorization, scopes write email and click on add. Then just save your settings.

Edit method request

The screenshot shows the 'Edit method request' dialog for the 'ANY' method. The 'Method request settings' panel includes the following fields:

- Authorization:** new-authorizer (selected from a dropdown)
- Authorization scopes:** email (selected in a dropdown, with an 'Add' button next to it)
- Request validator:** None (selected from a dropdown)
- API key required:** An unchecked checkbox.
- Operation name - optional:** GetPets (entered into a text input field)

5. Now you can see that in the method request your authorizer has been added.

The screenshot shows the 'Method request' tab of the API resource configuration. The 'Method request settings' panel displays the following configuration after saving the changes:

Setting	Value
Authorization	new-authorizer
Request validator	None
API key required	False
SDK operation name	Generated based on method and path

6. Now come back to your resource and click on Deploy API.

The screenshot shows the AWS API Gateway 'Resources' page. On the left, there's a tree view with a root node '/' and a child node '/demo-commerce-function'. Under '/demo-commerce-function', there is a single method entry labeled 'ANY'. On the right, the 'Resource details' section shows the path '/demo-commerce-function' and the resource ID '2aiijr'. Below this, the 'Methods (1)' section lists the method type as 'ANY', the integration type as 'Lambda', and the authorization as 'Cognito user pools' (not required). At the top right, there are buttons for 'API actions' (Delete, Update documentation, Enable CORS) and a prominent orange 'Deploy API' button.

7. Here choose default as your deploy stage and click on deploy.

The screenshot shows the 'Deploy API' dialog box. It starts with a heading 'Deploy API' and a close button 'X'. Below this, a note says: 'Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)' with a help icon. A warning message in a yellow box states: '⚠ When you deploy an API to an existing stage, you immediately overwrite the current stage configuration with a new active deployment.' The 'Stage' dropdown is set to 'default'. There is a 'Deployment description' input field which is currently empty. At the bottom, there are 'Cancel' and 'Deploy' buttons, with 'Deploy' being orange.

Step 4:

1. Now go to Cognito and open your user pool then go to App Integration.

Users Groups Sign-in experience Sign-up experience Messaging App integration Advanced security User pool properties

Configuration for all app clients
Domain and resource server settings for the user pool. All app clients that enable the Hosted UI use the user pool domain. All app clients can authorize access to user pool resource servers.

2. Scroll down to app clients and analytics and open your user-app.

App client list
The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

App clients and analytics (1) [Info](#)

Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

< 1 >

App client name	Client ID
<input type="radio"/> user-app	5m5e6afcrv54r87gfpsf4obids

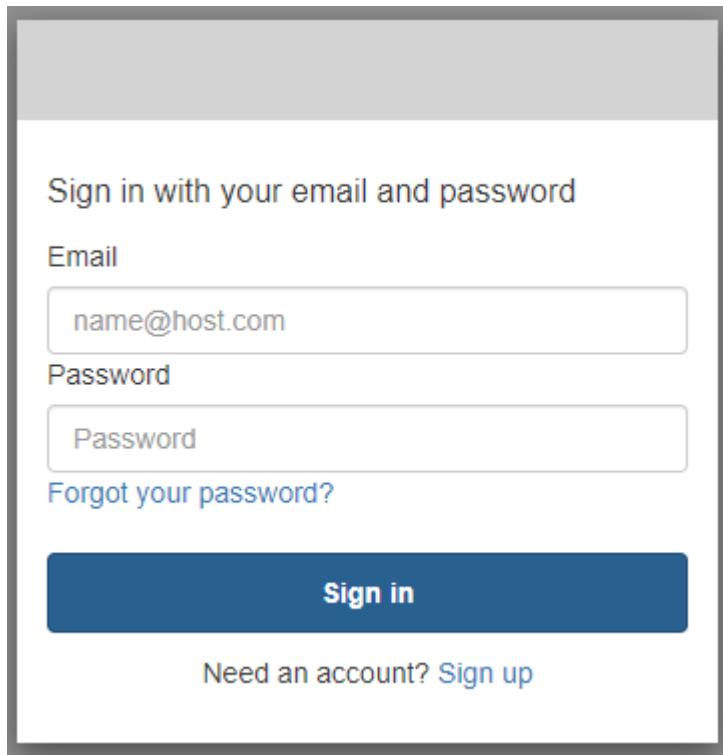
3. Here you will see a hosted UI, click on view hosted UI.

Hosted UI [Info](#)

Configure the Hosted UI for this app client.

Hosted UI status Available	Identity providers Cognito user pool directory
Allowed callback URLs https://new-app.com/callback	OAuth grant types Authorization code grant Implicit grant
Allowed sign-out URLs -	OpenID Connect scopes email openid phone
	Custom scopes -

5. Then you will see that it is asking you to sign in.



6. But you need to copy the URL of this sign-in page you will see that in the type you will be having code, so you need to change it to the token and then again the sign-in page will be opened.
7. So, now you need to choose Sign up and create a new user.

```
https://new-user-app.auth.ap-south-1.amazoncognito.com/login?client_id=5m5e6afcrv54r87gfpsf4obids&response_type=token&scope=email+openid+phone&redirect_uri=https%3A%2F%2Fnew-app.com%2Fcallback
```

Sign up with a new account

Email

Password

- ✓ Password must contain at least 6 characters
- ✓ Password must not contain a leading or trailing space

Already have an account? [Sign in](#)

8. Then you need to give the verification code to it.

Confirm your account

We have sent a code by email to m***@h***. Enter it below to confirm your account.

Verification code

Didn't receive a code? [Send a new code](#)

9. Below you can see that the error has occurred, but you need to copy the URL and paste it in a notepad.

Looks like this domain isn't connected to a website yet

Is this your domain?

Connect it to your Wix website in just a few easy steps:

1. Go to [Wix.com](#) > Subscriptions > Domains
2. Click Use a Domain You Already Own
3. Follow the steps to connect your domain to your website

Need more help?

Please contact our [Support Team](#) >



10. After pasting in the notepad, you will have two things Token ID and Access Token. So, you need to copy the Access Token.

```
https://new-app.com/callback#/id_token=eyJraiwQioiJyMG14VU1wbHp5bux2U0tvQUxSNnhXTmZleGxkcW3awM4cjZQaEhWVG5FPSiImFsZyI6I1JTMjU2In0.eyJhdF9oYXNoIjoiuNUJVx3Z3S2ZIZvQ2cmhWcnprZ31mUiSi1n1Yi16IjQxOTNKZGNHwUNwZDEtNzASz01Yi4LTMY4jhM2rj0WfmosIsImVtYw1sX3Z1cm1maWVkjIp2cnV1LcJpc3Mioi01odHrwzcpl1wv29bm10by1p2AuYXAtc291dGtMS5hbWF625hd3MuV29tXC9hNc1Cb29v0ac8x0tzauFOt3v5aCisImwvZ25pdGvdxN1cm5hblu01i01TkzZGrjys1lMQxLTcv0WutNWlyOC02OGI4YTNY21h2jki1C3hdw1q1011btV1NmFm332NTRyOdnZnBzZjRvYr1kcy1s1m22W50gX21krj0iN21z5h012001001YU41g2ZwYyNDYn1zj2zTdkMD11i1i1w1d97zW5fdXN11joiawQ1LcJhdxRox3Rpbw1j0je3MjQ80DU3MDe1m4c1t6MtcyNDQ40TMwSmiaWFO1joxNz10Rq1NzAxLCJqdGk0i1i1yTdh0W1mzs1jMjQxLTQ3y20t0DjhjZ111TN1YmE3N3Dq02Dui1C11bWfpb16Im1p1cm1wvN4MT1aa9FwawVklmNbVs39.B1R221ACjU_pztGnnfrQsrdIC1-KH7TQjuk99ejgsNPNUnd-e02krlj65jn0h11Cs1a1euRQ80C3bT3anS943cfc8G7jifsvBVDGeNbTuHIP_e070yb56nw8cdjAx10vKwsBPuM3yApuKgjsIm1Bx0DugFgbPsPV-mx-XgchLvxLKnCg_qBTdc8d16GBCyVkgnjzlayDg802kvT722g0xzCyxhn09MdMcK4qsd8LrnrtKT2DQ2BF07pgUCPn-OdF57U9KBgfFn5DnN3fTk15Ma71xMPNgslc90hXbfLjbtY1Mjants5fEEpfvXBHfouBu5ruWpB1LnqvQ8-access_token=eyJraiwQioi3zaInnbkNjZG5PwDNEOFFUv1Fe01c50vhNTzZnVTRErz0wswvUN0vRjPSiImFsZyI6I1JTMjU2In0.eyJzdwi1o1i01TkzZGrjys1lMQxLTcv0hUtnWwMyOCzOGI4YTNY21h2j1lCjc5h0i1odhRwcpcL1wv29bm10by1p2AuYXAtc291dGtMS5hbWF625hdMuY29tC9hcC1z0v0aC0x0tzauFOt3v5aCisIm1z1cnNpb2410j1s1mNsawuVdfpZC16IjYkwNu2Wzjcn1H14N2dmchNNG91awRz1wiz2z1bnRfaQ101i3Yjy2WE5MS00Nz11LTrhntg7ODN1z1i0n1j3hM11N2QwNzU1Lc302t1b191c2u101jYzN1c3M1c2jy29wZS161nBob251Ig9w2WspZC81bwFpb1sIm1fdghF61t2S16M6TcyNDQ4NTcwMSw1zXhw1j0xhNz10Ndg5M2AxLcJpxQx0jE3Mj0Q00D3MDe1m4c1t6jnjhjz2m0WE014NTY4NTGY1my42zWnkLTBhMzEy0D1jMD8iy1yisInvzXjUy11jio1NDE52Rky2Et2T8kMS03MD11LTvMjgtMzhioGEzG0M5Ywv5In0.eETevYmrqRzeefjfvM0r3Ymj5fBnfjb8a4pMrTxahLj21jizFg_juee3Nsc5453R8NzvFdd80er8B55FpyPw1QXmgC40a9njBn0i0vZ77twjtKpkS1_E1952cg1GjAwTPjaSwhQtbPLQF7tBcCGEdA_XCDrop1x1IPFtyRbrbrbzLaZg5cyjJAcdaudXcdq9pcXPUCStbdMlxztafDwm06_uakFBZ7YEIecXduQXN9a37u35y8p33ofPyh2G_w-2M9nfFeiUvotsz0P3Md2Fz2M0o_HsUDfdfdrUg720CGMGKkeTf88852h22tFp4dv7b5zG4zYwqP9j9w&expires_in=3600&token_type=Bearer
```

11. After that open Postman and choose GET, but you will need an invoke URL. So, come back to API gateway, go to stages expand default then come to GET Method, here you will find the invoke URL. Then copy it and paste it in the Postman GET method.

12. After that choose headers and, in the key, write authorization in the value paste the Access Token then click on Send. As you can see below you will get a hello from lambda.

13. This means that we have successfully secured our API gateway by choosing the Amazon Incognito User Pool.

The screenshot shows a POST request to the URL `https://qhg71wjjff.execute-api.ap-south-1.amazonaws.com/default/demo-ecommerce-function`. The Headers tab is selected, showing an Authorization header with the value `eyJraWQiOiJ3azNnbkNJZG5PWDNEOFFUUVVFeDlCS0VhNTZSznVTREZrOW9wV...`. The Body tab shows the response body: `1 "Hello from Lambda!"`. The status bar at the bottom indicates a 200 OK status with a time of 281 ms and a size of 354 B.

14. Now if you go back to Cognito and go to users then you will see that we also have a new user.

The screenshot shows the AWS Cognito User Pools console under the 'Users' tab. It displays one user entry:

User name	Email address	Email verified	Confirmation status	Status
4193ddca-e0d1-709e-5c...	miripac819@hapied.com	Yes	Confirmed	Enabled

15. Now clean up all the resources, so first delete your lambda function, API gateway then go to Cognito and delete your user pools.