

1. DynamoDB

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity
Orders	Active	id (S)	-	0	0	Off	On-demand	1
Products	Active	id (S)	-	0	0	Off	On-demand	1
Users	Active	id (S)	-	0	0	Off	On-demand	1

Product table:

id (String)	category	description	image	name	price
2	Electronics	Track your fit...	electronics/...	Smart Watch	24
8	Electronics	Portable Blue...	electronics/...	Bluetooth Sp...	8
1	Electronics	Premium nois...	electronics/...	Wireless He...	19
6	Electronics	Fast wireless c...	electronics/...	Wireless Ch...	39
5	Home	Double-walle...	home/stain...	Stainless St...	34
4	Accessories	Handcrafted ...	accessories...	Leather Wa...	59
7	Home	Handmade ce...	home/cera...	Ceramic Co...	19
3	Clothing	Comfortable ...	clothing/or...	Organic Cot...	29

2. Lambda Function

GetProductsFunction:

Screenshot of the AWS Lambda Function Overview page for 'getProductsFunction'.

Function Overview:

- Description:** -
- Last modified:** 2 hours ago
- Function ARN:** arn:aws:lambda:us-east-1:183295414469:function:getProductsFunction
- Function URL:** Info

Triggers:

- API Gateway
- + Add trigger

Code: Test | Monitor | Configuration | Aliases | Versions

Test event: Info

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save.

Actions: Delete | CloudWatch Logs Live Tail | Save | Test

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

Screenshot of the AWS Lambda Function Editor page for 'getProductsFunction'.

EXPLORER:

- GETPRODUCTSFUNCTION
 - node_modules
 - src
 - functions
 - orders
 - products
 - getProduct.mjs
 - getProducts.mjs
 - users
- TEST EVENTS (SELECTED: GETPRODUCTSTEST)
 - + Create new test event
 - Private saved events
 - getProductsTest
- ENVIRONMENT VARIABLES

Code Editor:

```
getProducts.mjs
src > functions > products > JS getProducts.mjs > ...
3  export const handler = async (event) => {
4    try {
5      const { queryStringParameters } = event;
6      const query = queryStringParameters?.query || "";
7
8      // Get products from Dynamodb
9      const products = await db.scan({ TableName: env.PRODUCT_TABLE });
10
11      // If query parameter is provided, filter products
12      let filteredProducts = products;
13      if (query) {
14        const lowercaseQuery = query.toLowerCase();
15        filteredProducts = products.filter(
16          (product) =>
17            product.name.toLowerCase().includes(lowercaseQuery) ||
18            product.description.toLowerCase().includes(lowercaseQuery)
19        );
19
20
21      return {
22        statusCode: 200,
23        headers: {
24          "Content-Type": "application/json",
25          "Access-Control-Allow-Origin": "*",
26        },
27        body: JSON.stringify(filteredProducts)
28      };
29    }
30  }
```

Actions: Deploy (Ctrl+Shift+U) | Test (Ctrl+Shift+I)

Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | JavaScript | Lambda | Layout: US

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

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/getProductsFunction?subtab=enVars&tab=code>. The page displays the function's code properties, runtime settings, and layers. On the right side, there is a 'Tutorials' sidebar titled 'Create a simple web app'.

Code properties

- Package size: 3 MB
- SHA256 hash: nQW55kFgs0nnz0n8Yx7inCrgqdc7Fr2wOYSSGeK8uBE=
- Last modified: 2 hours ago

Runtime settings

- Runtime: Node.js 22.x
- Handler: src/functions/products/getProducts.handler
- Architecture: x86_64

Layers

There is no data to display.

Tutorials

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](#)

PlaceOrderFunction

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/placeOrderFunction?subtab=enVars&tab=code>. The page displays the function's overview, code source, and deployment details. On the right side, there is a 'Tutorials' sidebar titled 'Create a simple web app'.

placeOrderFunction

Function overview

- Description: -
- Last modified: 2 hours ago
- Function ARN: arn:aws:lambda:us-east-1:183295414469:function:placeOrderFunction
- Function URL: -

Code source

[Upload from](#)

Tutorials

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](#)

```

src/functions/orders/createOrder.mjs
1 import { v4 as uuidv4 } from "uuid";
2 import db from ".././utils/db.mjs";
3
4 export const handler = async (event) => {
5   try {
6     const orderData = event['body-json'].body;
7
8     // Get user information from the event context (Auth from cognito)
9     const userId = event.context.sub;
10    const userEmail = event.context.email;
11
12    // Create new order with generated ID
13    const order = {
14      id: uuidv4(),
15      userId,
16      userEmail,
17      items: orderData.items,
18      totalPrice: orderData.totalPrice,
19      status: "pending",
20      shippingAddress: orderData.shippingAddress,
21      paymentMethod: orderData.paymentMethod,
22      createdAt: new Date().toISOString(),
23    };
24
25    // Save order to DynamoDB

```

Code properties

- Package size: 3 MB
- SHA256 hash: nQW5SkFgs0nnz0n8Yx7inCrgqdc7Fr2wOYSSGeK8
uBE=
- Last modified: 2 hours ago

Runtime settings

- Runtime: Node.js 22.x
- Handler: src/functions/orders/createOrder.handler
- Architecture: x86_64

Layers

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

3. API gateway

The screenshot shows the AWS API Gateway Resources page. On the left, the navigation sidebar includes sections for APIs, Custom domain names, Domain name access associations, VPC links, and the selected API: serverless-backend-gw. Under Resources, there are options for Stages, Authorizers, Gateway responses, Models, Resource policy, Documentation, Dashboard, and API settings. Below these are Usage plans and API keys. The main content area is titled "Resources" and shows a tree structure of resources under the root path "/". The "/orders" resource has two methods listed: OPTIONS and POST. The "/products" resource has two methods listed: GET and OPTIONS. On the right, the "Resource details" panel shows the Path as "/" and the Resource ID as "geuohmh2d1". It also contains tabs for "Methods (0)" and "No methods defined." There are buttons for "Update documentation" and "Enable CORS". At the top right, there are "API actions" and "Deploy API" buttons.

The screenshot shows the AWS API Gateway Method execution page for the POST method of the /orders resource. The left sidebar is identical to the previous screenshot. The main content area is titled "/orders - POST - Method execution". It shows the ARN as "arn:aws:execute-api:us-east-1:183295414469:u7eiihabee/*:POST/orders" and the Resource ID as "0zxdbp". A flow diagram illustrates the request and response paths: Client → Method request → Integration request → Lambda integration → Integration response → Method response. Below this, the "Method request settings" section includes fields for Authorization (set to "cognito-token-authorizer"), Request validator (set to "None"), API key required (set to "False"), and SDK operation name (set to "Generated based on method and path"). There are tabs for Method request, Integration request, Integration response, Method response, and Test.

4. Cognito User pool

The screenshot shows the 'Users' section of the Amazon Cognito User Pools console. On the left, a navigation sidebar lists various sections like Applications, User management, Authentication, Security, and more. The 'User pool - ifshs9' is selected. The main area displays a table of users with one entry:

User name	Email address	Email verified	Confirmation status	Status
a4f8b448-1071-7022-0...	nle@gmail.com	No	Confirmed	Enabled

Below the user table, there is a section for 'Import users (0)' which allows creating CSV import jobs.

Resource server (Scopes)

The screenshot shows the 'Domain' section of the Amazon Cognito User Pools console. The left sidebar is identical to the previous screenshot. The main area shows a 'Cognito domain' configuration with a URL listed: <https://us-east-15hz4csevq.auth.us-east-1.amazoncognito.com>. To the right, there is a 'Custom domain' section which is currently empty. Below this, the 'Resource servers (1)' section is shown, listing a single resource server:

Resource server name	Resource server identifier	Custom scopes
e-commerce-api-resourceserver	https://u7eihabee.execute-api.us-east-1.amazonaws.com/prod	2 custom scopes

Screenshot of the AWS Cognito Resource Server configuration page.

Resource server
 Configure resource servers. A resource server is an integration between your user pool and an API. Each resource server has custom scopes that you must activate in your app client. When you configure a resource server, your app can generate access tokens with OAuth scopes that authorize read and write operations to an API server.

Resource server name
 Enter a friendly name for the resource server.

 Must contain between 1 and 256 characters. The resource server name can only contain: letters, numbers, spaces, underscores, and the symbols +_=,@_-

Resource server identifier

Custom scopes - optional
 Add custom scopes to this resource server. Custom scopes are added in the scope claim in the access token.

Scope name	Description	Action
read	allow read api	<button>Remove</button>
write	allow write api	<button>Remove</button>

Must contain between 1 to 256 characters. Custom scope cannot contain spaces, double quotes and slashes.

[Add another](#)
 You can add up to 98 more custom scopes.

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

App client

Screenshot of the AWS Cognito App Client configuration page.

App client: My Frontend

App client information

App client name My Frontend	Authentication flow session duration 3 minutes	Created time April 21, 2025 at 08:26 CDT
Client ID <input type="text" value="48foonrut8a0pfog35678mro47"/>	Refresh token expiration 5 day(s)	Last updated time April 21, 2025 at 18:51 CDT
Client secret -	Access token expiration 60 minutes	
Authentication flows Choice-based sign-in Secure remote password (SRP) Get user tokens from existing authenticated sessions	ID token expiration 60 minutes	Advanced authentication settings Enable token revocation Enable prevent user existence errors

[Edit](#) [View login page](#)

Managed login pages configuration [Info](#)
 Configure the managed login pages for this app client.

Quick setup guide | Attribute permissions | [Login pages](#) (selected) | Threat protection | Analytics

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

The screenshot shows the Amazon Cognito console with the following details:

- Current user pool:** User pool - ifshs9
- Applications:** App clients New
- Managed login pages configuration:**
 - Status:** Available
 - Allowed callback URLs:** http://localhost:3000, https://d1r0w5dxcka06.cloudfront.net, https://d1r0w5dxcka06.cloudfront.net/checkout
 - Allowed sign-out URLs:** https://d1r0w5dxcka06.cloudfront.net
 - Identity providers:** Cognito user pool directory
 - OAuth grant types:** Authorization code grant
 - OpenID Connect scopes:** email, openid, phone
 - Custom scopes:** https://u7elihabee.execute-api.us-east-1.amazonaws.com/prod/read, https://u7elihabee.execute-api.us-east-1.amazonaws.com/prod/write
- Managed login style - new:** Style assigned to this app client: 7e942fce-16a8-4e23-965e-0b1efe4cf74b

5. AWS codepipeline

The screenshot shows the AWS CodePipeline console with the following details:

- Pipeline:** serverless-backend-pipeline
- Stages:**
 - Source:** GitHub (via GitHub App) - 2.hours.ago
 - Build:** AWS CodeBuild - 2.hours.ago
- Actions:**
 - Source: Fix error
 - Build: Fix error

Source stage

The screenshot shows the AWS CodePipeline execution summary for a pipeline named "serverless-backend-pipeline". The execution ID is c3580ed8-5952-4018-a6ee-0826c905f06. The status is Succeeded, and it completed 2 hours ago. The duration was 2 seconds. The action provider is GitHub (via GitHub App). The region is us-east-1. The stage name is Source, and the action name is Source. The artifact name is SourceArtifact.

Stage name	Action name	Action provider	Region
Source	Source	GitHub (via GitHub App)	us-east-1

Message: {"ProviderType": "GitHub", "CommitMessage": "Fix error placeOrder API, and integrate with cognito"}

The screenshot shows the AWS CodePipeline action configuration for the GitHub source action. The branch name is master. The connectionArn is arn:aws:codeconnections:us-east-1:183295414469:connection/24899481-a91b-407b-91f1-44d3179fed1. The detectChanges setting is true. The full repository ID is vnhatle/serverless-backend, and the output artifact format is CODE_ZIP.

BranchName	ConnectionArn	DetectChanges	FullRepositoryId	OutputArtifactFormat
master	arn:aws:codeconnections:us-east-1:183295414469:connection/24899481-a91b-407b-91f1-44d3179fed1	true	vnhatle/serverless-backend	CODE_ZIP

Artifacts

Artifact name	Artifact type	Artifact provider
SourceArtifact	Output	Amazon S3

Output variables

Key	Value
AuthorDate	2025-04-22T00:04:42Z
AuthorDisplayName	vnhatle
AuthorEmail	lenhat.bkit@gmail.com
AuthorId	vnhatle

Build stage:

The screenshot shows the AWS CodeBuild execution summary for a build named 'Build: 4c313a0b'. The status is 'Succeeded' with a green checkmark. The build started 2 hours ago and completed 2 hours ago, taking 1 minute 24 seconds. The pipeline execution ID is 'c3580edb-5952-4018-a6ee-0826c9052f06'. The message field is empty.

Execution summary

Status	Started	Completed	Duration
SUCCEEDED	2 hours ago	2 hours ago	1 minute 24 seconds

Pipeline execution ID: c3580edb-5952-4018-a6ee-0826c9052f06

Message: -

Action details

Stage name	Action name	Action provider	Region
Build	Build	AWS CodeBuild	us-east-1

Namespace: BuildVariables

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

The screenshot shows the AWS CodeBuild action details, configuration, artifacts, and output variables sections. The action details show a build stage named 'Build' using the 'Build' action provider 'AWS CodeBuild' in the 'us-east-1' region. The configuration section shows the project name 'serverless-backend-codebuild'. The artifacts section lists two artifacts: 'SourceArtifact' (Input, Amazon S3) and 'BuildArtifact' (Output, Amazon S3). The output variables section is currently empty.

Action details

Stage name	Action name	Action provider	Region
Build	Build	AWS CodeBuild	us-east-1

Namespace: BuildVariables

Action configuration

ProjectName
serverless-backend-codebuild

Artifacts

Artifact name	Artifact type	Artifact provider
SourceArtifact	Input	Amazon S3
BuildArtifact	Output	Amazon S3

Output variables

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

6. S3 (store product images)

e-commerce-images-s3

Objects

Name	Type	Last modified	Size	Storage class
accessories/	Folder	-	-	-
clothing/	Folder	-	-	-
electronics/	Folder	-	-	-
home/	Folder	-	-	-

Objects

Name	Type	Last modified	Size	Storage class
bluetooth-speaker.png	png	April 21, 2025, 20:31:34 (UTC-05:00)	95.7 KB	Standard
smart-watch.png	png	April 21, 2025, 20:31:36 (UTC-05:00)	382.5 KB	Standard
wireless-charging-pad.png	png	April 21, 2025, 20:31:37 (UTC-05:00)	106.4 KB	Standard
wireless-headphones.jpg	jpg	April 21, 2025, 20:31:37 (UTC-05:00)	7.7 KB	Standard