

[AI for Bharat] - Build an Image Editing Serverless App

Deploy a serverless web application to edit images using Amazon Bedrock

▶ Getting Started

▶ Lab

▶ Summary and Resources

▼ AWS account access

[Open AWS console \(us-east-1\)](#)

[Get AWS CLI credentials](#)

Workshop catalog in AWS Builder Center

Exit event

[Event dashboard](#) > Deploy a serverless web application to edit images using Amazon Bedrock

[AI for Bharat] - Build an Image Editing Serverless App

Event information

Start time

12/11/2025 03:59 PM

Duration

48 hours

Accessible Regions

us-east-1

Description

Creating image editing applications traditionally requires complex infrastructure, specialized ML expertise, and ongoing maintenance—making it costly and time-consuming for developers to build and scale. In this hands-on workshop, you'll build a complete AI-powered image editing application without managing a single server or ML model. Users can transform images with simple text prompts like "make this sunset more dramatic" or "remove the background and replace with mountains."

[Show less](#)

Workshop

[Get started](#)

Title

Deploy a serverless web application to edit images using Amazon Bedrock

Complexity level

200

Topics

Front-End Web & Mobile, Serverless, Generative AI

AWS services

Amazon API Gateway, Amazon Bedrock, AWS Lambda

Description

Deploy a serverless web application to edit images using Amazon Bedrock. In this workshop we will leverage Amazon Cognito, AWS Lambda, Amazon API Gateway and AWS Amplify. We will walk through creating all of these components and connecting them together to get a working solution at the end of the workshop.

[Show less](#)

Event outputs (0)



Amazon Cognito > Set up resources for your application

>Your application "ImageEditApp" and user pool "User pool - mzqg6" have been created successfully! Follow the instruction to continue the setup.

Set up resources for your application Info

Check out your sign-in page



Your application login pages are ready. Explore a new sign-in and sign-up experience powered by Cognito.

[View login page](#)

[What's a user pool?](#)

Build authentication components for your application



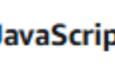
Review a quick setup guide for your application type. Deploy a basic sample app or add code to your existing one.

Quick setup guide

What's the development platform for your single page application?

 **React**

 **Angular**

 **JavaScript**

Add the example code to your application

- 1 Configure your user pool app client with allowed callback URLs, logout URLs, and the scopes that you want to request, for example `openid` and `profile`. [Learn more](#)
- 2 Install the [oidc-client-ts](#) and [react-oidc-context](#) libraries.

```
1 npm install oidc-client-ts react-oidc-context --save
```
- 3 Configure `react-oidc-context` with the OIDC properties of your user pool.

```
1 // index.js
2 import React from "react";
3 import ReactDOM from "react-dom/client";
4 import App from "./App";
5 import { AuthProvider } from "react-oidc-context";
```

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: 156-925-912 | WSParticipantRole/Participant

DynamoDB > Tables

The ImageGenerationTable table was created successfully.

Tables (1) Info Last updated December 12, 2025, 13:42 (UTC+5:30) Actions Delete Create table

Find tables Filter by tag Any tag key Filter by tag value Any tag value

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode	Total size	Table class
ImageGenerationTable	Active	id (S)	-	0	0	Off	☆	On-demand	On-demand	0 bytes	Standard

Dashboard Tables Explore items PartiQL editor Backups Exports to S3 Imports from S3 Integrations Reserved capacity Settings

DAX Clusters Subnet groups Parameter groups Events

Identity and Access Management (IAM) > Policies

Policy DynamoDBImageGenerationPolicy created. [View policy](#) [X](#)

Policies (1441) [Info](#)

A policy is an object in AWS that defines permissions.

Filter by Type: All types

Policy name	Type	Used as	Description
AccessAnalyzerServiceRolePolicy	AWS managed	None	Allow Access Analyzer to analyze resou...
AccountManagementFromVercel	AWS managed	None	For use with accounts created through ...
AdministratorAccess	AWS managed - job function	Permissions policy (1)	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissi...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permissi...
AIOpsAssistantIncidentReportPolicy	AWS managed	None	Provides permissions required by the A...
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions require...
AIOpsConsoleAdminPolicy	AWS managed	None	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	None	Grants access to the Amazon AI Operat...
AIOpsReadOnlyAccess	AWS managed	None	Grants ReadOnly permissions to the A...
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to Al...
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessNetworkProfileServicePolicy	AWS managed	None	This policy enables Alexa for Business t...
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to AlexaForBu...
AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/edit/dele...
AmazonAPIGatewayInvokeFullAccess	AWS managed	None	Provides full access to invoke APIs in A...

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

X

ImageEditBackend

[Throttle](#)[Copy ARN](#)[Actions ▾](#)

Function overview Info

[Diagram](#) | [Template](#)[Export to Infrastructure Composer](#)[Download ▾](#)[+ Add trigger](#)[+ Add destination](#)**Description**

-

Last modified

57 seconds ago

Function ARN[arn:aws:lambda:us-east-1:156592059312:function:ImageEditBackend](#)[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)[Open in Visual Studio Code](#)[Upload from](#) ▾

Code source Info

```
lambda_function.py
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

EXPLORER

IMAGEEDITBACKEND

lambda_function.py

DEPLOY ✓ Current

aws Search [Alt+S] United States (N. Virginia) ▾ Account ID: 1565-9205-9312 ▾ WSParticipantRole/Participant

Lambda > Functions > ImageEditBackend

Successfully updated the function ImageEditBackend. X

Code source Info Open in Visual Studio Code ↗ Upload from ▾ ▼

EXPLORER λ ... λ λ ...

IMAGEEDITBACKEND ...

lambda_function.py

lambda_function.py

```
1 import os
2 import json
3 import base64
4 import time
5 import uuid
6 import os
7 from io import BytesIO
8 from random import randint
9 from datetime import datetime
10 def get_cors_headers():
11     return {
12         'Access-Control-Allow-Origin': '*',
13         'Access-Control-Allow-Headers': 'Access-Control-Allow-Headers,Access-Control-Allow-Origin,Content-Type,Authorization',
14         'Access-Control-Allow-Methods': 'OPTIONS,POST,GET'
15     }
16 def calculate_base64_size(base64_string):
17     """Calculate the size of a base64 encoded string in bytes"""
18     try:
19         # Remove data URL prefix if present
20         if ',' in base64_string:
21             base64_data = base64_string.split(',')[1]
22         else:
23             base64_data = base64_string
24         # Calculate size: (length * 3/4) - padding
25         padding = base64_data.count('=')
26         size = (len(base64_data) * 3 // 4) - padding
27         return size
28     except Exception:
29         return 0
30 def calculate_output_images_size(images):
31     """Calculate total size of output images in bytes"""

```

DEPLOY ✓ Current

Deploy (Ctrl+Shift+U) Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED] + Create new test event

ENVIRONMENT VARIABLES

✓ Lambda Deployed 0 0 Amazon Q

Successfully updated the function ImageEditBackend. Ln 309, Col 10 Spaces: 4 UTF-8 LF Python λ Lambda Layout: US ...

Code properties Info

Package size 2.8 kB

SHA256 hash 9+LsBhvAqpFw+N6y6gVuO0uWRssosvWq/VrX/8HR7Ac=

Last modified 9 seconds ago

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

aws | Search [Alt+S] | United States (N. Virginia) | WSParticipantRole/Participant | Account ID: 1565-9205-9312

Lambda > Functions > ImageEditBackend

Successfully updated the function ImageEditBackend.

ImageEditBackend

Throttle | Copy ARN | Actions ▾ | Export to Infrastructure Composer | Download ▾

Function overview | Info | Diagram | Template | + Add trigger | + Add destination

Description: -

Last modified: 22 seconds ago

Function ARN: arn:aws:lambda:us-east-1:156592059312:function:ImageEditBackend

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration | Info | Edit

Description: -

Memory: 128 MB | Ephemeral storage: 512 MB

Timeout: 1 min 0 sec | SnapStart: None

General configuration | Info | Edit

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

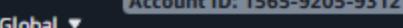
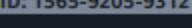
VPC

RDS databases

Monitoring and operations tools

Concurrency and recursion detection

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

aws  Search [Alt+S]  Account ID: 1565-9205-9312    Global 

 IAM > Roles > lambda-iam-role-cfn-ImageEditBackendRole-sxYp9fhfBFEI 

 Policy was successfully attached to role.

lambda-iam-role-cfn-ImageEditBackendRole-sxYp9fhfBFEI

Summary

Creation date: December 10, 2025, 18:00 (UTC+05:30)  ARN: arn:aws:iam::156592059312:role/lambda-iam-role-cfn-ImageEditBackendRole-sxYp9fhfBFEI

Last activity: - Maximum session duration: 1 hour

Permissions policies (2)

You can attach up to 10 managed policies.    

Policy name	Type	Attached entities
 AmazonBedrockAccess	Customer inline	0
 DynamoDBImageGenerationPolicy	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](#)



No requests to generate a policy in the past 7 days.

 CloudShell  Feedback 

API Gateway > APIs > Resources - ImageEditingAppBackendAPI (e34nt8nw07)

API Gateway ×

APIs

Custom domain names

Domain name access associations

VPC links

AgentCore targets [New](#)

API:
ImageEditingAppBackendAPI

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

API keys

Client certificates

Settings

Developer portals [New](#)

Portals

Portal products

Resources

Successfully enabled CORS Details ×

Resource details

Path: /

Resource ID: 3cru9qr4ie

Methods (2)

Method type	Integration type	Authorization	API key
<input type="radio"/> OPTIONS	Mock	None	Not required
<input type="radio"/> POST	Lambda	None	Not required

[Create resource](#) [API actions ▾](#) [Deploy API](#) [Update documentation](#) [Enable CORS](#) [Delete](#) [Create method](#)

aws  Search [Alt+S]    United States (N. Virginia)  Account ID: 1565-9205-9312  WSPParticipantRole/Participant  

API Gateway > APIs > ImageEditingAppBackendAPI (e34nt8nw07) > Authorizers

API Gateway  

APIs

Custom domain names

Domain name access associations

VPC links

AgentCore targets 

▼ API: ImageEditingAppBackendAPI

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

API keys

Client certificates

Settings

Developer portals 

Portals

Portal products

Authorizers (1)    

CognitoAuthorizer 

Authorizer ID
cuoja6

Cognito pool
User pool - mzqg6 - cpJvW5TA6 (us-east-1)

Token source
Authorization

Token validation - optional
none

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

aws [Alt+S] Q Search United States (N. Virginia) Account ID: 1565-9205-9312 WSParticipantRole/Participant

API Gateway > APIs > Resources - ImageEditingAppBackendAPI (e34nt8nw07)

API Gateway ×

APIs

- Custom domain names
- Domain name access associations
- VPC links
- AgentCore targets New

API: ImageEditingAppBackendAPI

Resources

- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

Usage plans

API keys

Client certificates

Settings

Developer portals New

Portals

Portal products

Resources

Successfully edited method request for 'POST'. Redeploy your API for the update to take effect.

/ - POST - Method execution

ARN: [arn:aws:execute-api:us-east-1:156592059312:e34nt8nw07/*/POST/](#)

Resource ID: 3cru9qr4ie

Method request **Integration request** **Integration response** **Method response** **Test**

Method request settings Edit

Authorization: [CognitoAuthorizer](#)

Request validator: None

API key required: False

SDK operation name: Generated based on method and path

Request paths (0)

Name: Caching

No request paths

No request paths defined

URL query string parameters (0)

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

AWS Account ID: 1565-9205-9312 | United States (N. Virginia) | WSParticipantRole/Participant

Search [Alt+S] API Gateway > APIs > ImageEditingAppBackendAPI (e34nt8nw07) > Stages

API Gateway APIs Custom domain names Domain name access associations VPC links AgentCore targets New

▼ API: ImageEditingAppBackendAPI Resources Stages Authorizers Gateway responses Models Resource policy Documentation Dashboard API settings

Usage plans API keys Client certificates Settings

Developer portals New Portals Portal products

Successfully created deployment for ImageEditingAppBackendAPI. This deployment is active for dev.

Stages

Stage details [Info](#) Edit

Stage name dev

Rate [Info](#) 10000

Cache cluster [Info](#) Inactive

Burst [Info](#) 5000

Default method-level caching Inactive

Invoke URL <https://e34nt8nw07.execute-api.us-east-1.amazonaws.com/dev>

Active deployment bkpdk on December 12, 2025, 14:46 (UTC+05:30)

Logs and tracing [Info](#) Edit

CloudWatch logs Inactive

Detailed metrics Inactive

X-Ray tracing Inactive

Data tracing Inactive

Custom access logging Inactive

Stage variables Deployment history Documentation history Canary Tags

Stage variables (0/0) [Edit](#)

Find resources

CloudShell Feedback 1 / 1 © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

File Edit Selection View Go Run Terminal Help ← → Search

Js config.js X

C: > Users > dines > Downloads > AWS-Amplify-Code > config > Js config.js > api > invokeUrl

```
1 // AWS Workshop Configuration File
2 // Replace the placeholder values below with your actual AWS resource identifiers
3 //
4 // Instructions:
5 // 1. Replace 'REPLACE_WITH_YOUR_USER_POOL_ID' with your Cognito User Pool ID
6 // Example: us-west-2_uXboG5pAb
7 // Find this in: AWS Console > Cognito > User Pools > [Your Pool] > General Settings
8 //
9 // 2. Replace 'REPLACE_WITH_YOUR_CLIENT_ID' with your Cognito App Client ID
10 // Example: 25ddkmj4v6hfsfvruhpfi7n4hv
11 // Find this in: AWS Console > Cognito > User Pools > [Your Pool] > App Clients
12 //
13 // 3. Replace 'REPLACE_WITH_YOUR_REGION' with your AWS region
14 // Example: us-west-2, us-east-1, eu-west-1
15 //
16 // 4. Replace 'REPLACE_WITH_YOUR_API_URL' with your API Gateway endpoint
17 // Example: https://abc123def.execute-api.us-west-2.amazonaws.com/prod
18 // Find this in: AWS Console > API Gateway > [Your API] > Stages > [Stage Name]
19
20 window._workshopConfig = {
21   cognito: {
22     userPoolId: 'us-east-1_cpJvW5TA6', // e.g. us-west-2_uXboG5pAb
23     userPoolClientId: '7q2ub4p6218ph6ioqlt5o166nk', // e.g. 25ddkmj4v6hfsfvruhpfi7n4hv
24     region: 'us-east-1' // e.g. us-west-2
25   },
26   api: [
27     { invokeUrl: 'https://e34nt8nw07.execute-api.us-east-1.amazonaws.com/dev' } // e.g. https://abc123def.execute-api.us-west-2.amazonaws.com/prod
28   ]
29 };
30
31 // Configuration validation flag
32 window._configLoaded = true;
33
34 // Development note: This file is automatically generated during the build process.
35 // Do not modify this file structure, only update the placeholder values above.
36
```

Launchpad AWS: profile:default Amazon Q

Ln 27, Col 75 Spaces: 2 UTF-8 LF { } JavaScript Go Live

CloudShell | Feedback | Search | [Alt+S] | All apps > app6283: Overview | Give feedback | Support | Docs

app6283 | Overview | Hosting | Monitoring | App settings | app6283 | App ID: d2d56g42as01xk | Visit deployed URL

app6283

App ID: d2d56g42as01xk

Get to production

0 of 3 steps complete

1 Add a custom domain

Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app.

Add custom domain

2 Enable firewall protections

Web traffic restrictions for Amplify Hosting are offered by AWS Web Application Firewall (WAF).

Enable firewall

3 Connect new branches

Connect another branch from your Git repository to set up multiple environments.

Connect a new branch

Branches 1 | Search... | + Add branch

staging | Deployed (✓)

Domain: <https://staging.d2d56g42as01xk.amplifyapp.com> | Last deployment: 0 minutes ago

Deploy updates | ★ Production branch

Authentication Required

Job Zero: Use your Amazon Cognito credentials

\$ Please authenticate to access the workshop tools

Username:

Enter your username

Password:

Enter your password

Login

Step 1 - Create a mask on your image



Stroke size - **50** +



Application Health



Status

\$ Region: **us-east-1**
\$ User Pool: **READY**
\$ API Gateway: **READY**
\$ Auth Token: **VALID**

JWT Token

```
eyJraWQiOiI4Yng3eXNLazczTULRT0N0cVgyVFwvNDN4MGV1R  
G...
```

[Copy Token](#)

Stroke size - **50** +

[Reset Mask](#)

[Download Mask & Image](#)



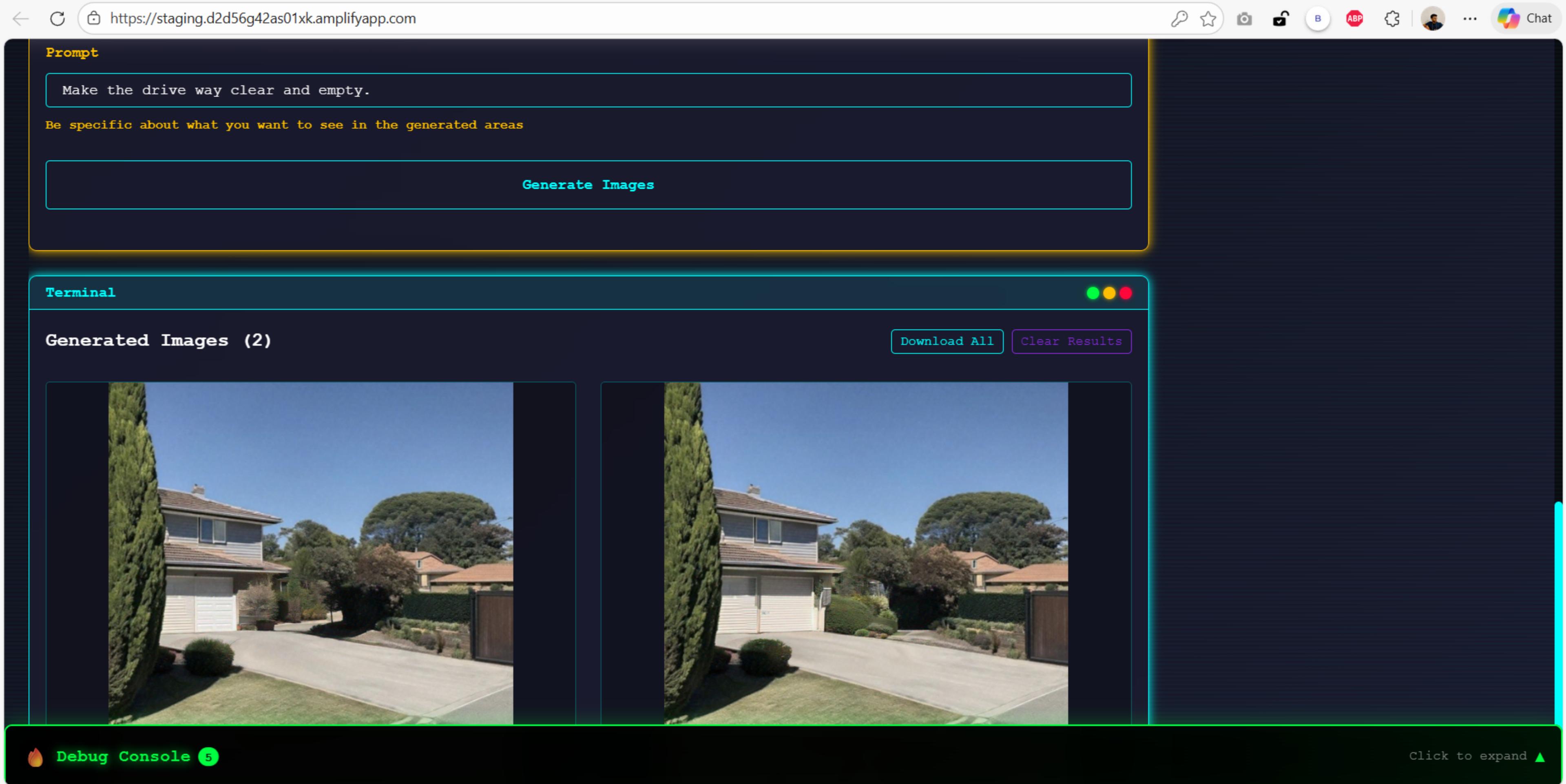
Status

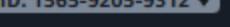
\$ Region: **us-east-1**
\$ User Pool: **READY**
\$ API Gateway: **READY**
\$ Auth Token: **VALID**

JWT Token

eyJraWQiOiI4Yng3eXNLazczTUlRT0N0cVgyVFwvNDN4MGV1R
G...

[Copy Token](#)



aws  Search [Alt+S]    United States (N. Virginia)  Account ID: 1565-9205-932

DynamoDB > Explore items > ImageGenerationTable   

DynamoDB 

- Dashboard
- Tables
- Explore items**
- PartiQL editor
- Backups
- Exports to S3
- Imports from S3
- Integrations
- Reserved capacity
- Settings

DAX 

- Clusters
- Subnet groups
- Parameter groups
- Events

Tables (1) 

Filter by tag - Any tag key 

Filter by tag value - Any tag value 

Find tables 

◀ 1 ▶ 

ImageGenerationTable 

ImageGenerationTable

▼ Scan or query items  [View table details](#)

Scan Query

Select a table or index **Table - ImageGenerationTable** 

Select attribute projection **All attributes** 

▶ Filters - optional

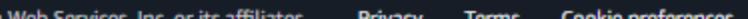
Run **Reset**

 **Completed** · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCU consumed: 2 

Table: ImageGenerationTable - Items returned (1)  

Scan started on December 12, 2025, 15:22:19   

<input type="checkbox"/>	id (String) 	genera... 	image_... 	mask_base64_... 	mode 	model_id 	output_images_... 	prompt 	success 	timestamp 
<input type="checkbox"/>	560d6745-e061-...	19158	121741	10224	INPAINTING	amazon.titan-image-ge...	1067095	Make the drive way clear...	true	2025-12-12...

 CloudShell  Feedback 

Upon completion of this workshop, you will:

1. Gain knowledge of how to leverage various AWS services to build a web application
2. Learn how to deploy Amazon Cognito user pool for authentication
3. Learn how to create a DynamoDB table for persistent data storage
4. Learn how to deploy AWS Lambda functions with secure IAM policies
5. Learn how to create an API Gateway with Cognito integration
6. Deploy an AWS Amplify application for frontend hosting
7. Use the deployed web application to edit images with AI-powered prompts

Getting Started

Ready to build your serverless image editing application?

Proceed through the workshop modules in order:

[Lab - Deploy a serverless web application to edit images using Amazon Bedrock](#)

Cleanup

Don't forget to [Cleanup](#) the environment after completing the labs.

 **Note:** If you have run this workshop in your own account we recommend you cleanup all the resources to avoid any further charges. If you participated in an AWS run workshop you do not have to complete this step.

License Summary

The documentation is made available under the Creative Commons Attribution-ShareAlike 4.0 International License. Download the [LICENSE](#) .

The sample code within this documentation is made available under the MIT-0 license. Download the [LICENSE-SAMPLECODE](#) .

Next

[AI for Bharat] - Build an Image Editing Serverless App <

Deploy a serverless web application to edit images using Amazon Bedrock

▼ Getting Started

Option 1 - Getting Started at an AWS Event

Option 2 - Getting Started with your own AWS Account

► Lab

▼ Summary and Resources

Clean Up Environment

Workshop contributors

▼ AWS account access

[Open AWS console \(us-east-1\)](#)

[Get AWS CLI credentials](#)

Workshop catalog in AWS Builder Center

Exit event

Summary and Resources

Summary

Congratulations on completing this workshop!

You now have hands-on experience with deploying a serverless web application to edit images using Amazon Bedrock.

In this workshop, you:

1. Learned how to create an Amazon Cognito user pool and a test user.
2. Learned how to create a DynamoDB table to store image generation data.
3. Learned how to create a Lambda function and securing its access to Amazon Bedrock and DynamoDB.
4. Learned how to create an Amazon API Gateway and an authorizer to control access. Then created methods of "POST" and "OPTIONS" for your request.
5. Learned how to deploy an application using AWS Amplify
6. Working with the foundation model to edit images using prompts.

Resources

Visit this page to find the latest documentation

- [AWS Serverless](#)
- [Amazon Cognito](#)
- [Amazon DynamoDB](#)
- [AWS Lambda](#)
- [Amazon API Gateway](#)
- [AWS Amplify](#)
- [Amazon Bedrock](#)
- [Amazon Titan Image Generator Models](#)

[Previous](#)[Next](#)