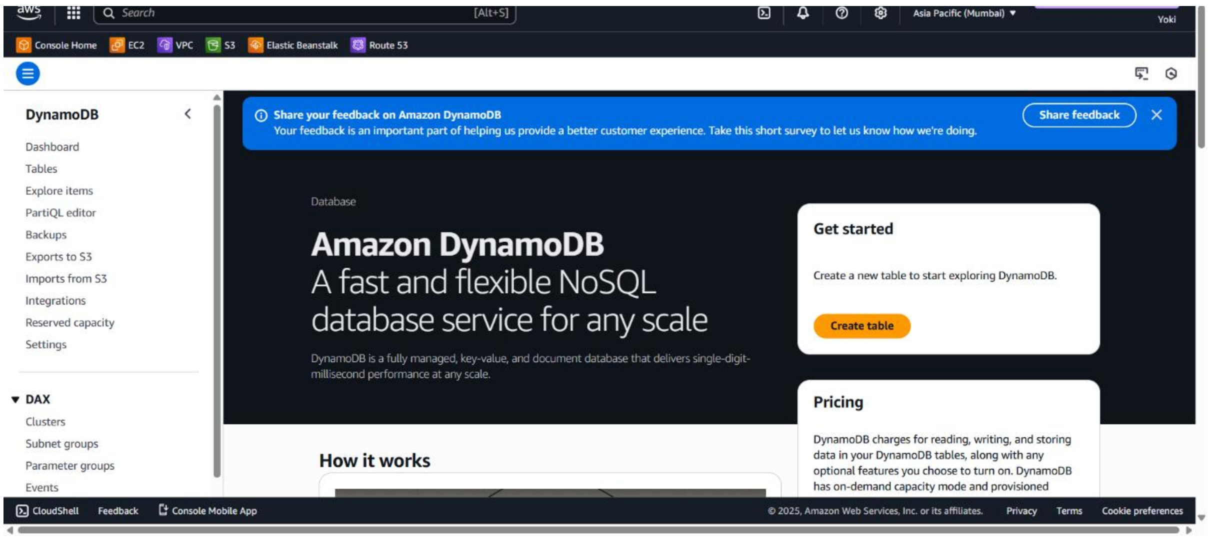
3 - GenAI Version (Make Facts Witty):

~ In this stage, you will make those facts funny and engaging using **Amazon Bedrock.**

~ Amazon Bedrock allows you to use foundation models (like **Anthropic** Claude, AI21, etc.) directly from AWS without managing infrastructure.

~ This means you can enhance your cloud facts dynamically.



Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Database

Amazon DynamoDB
A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

Get started

Create a new table to start exploring DynamoDB.

Pricing

DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to turn on. DynamoDB has on-demand capacity mode and provisioned

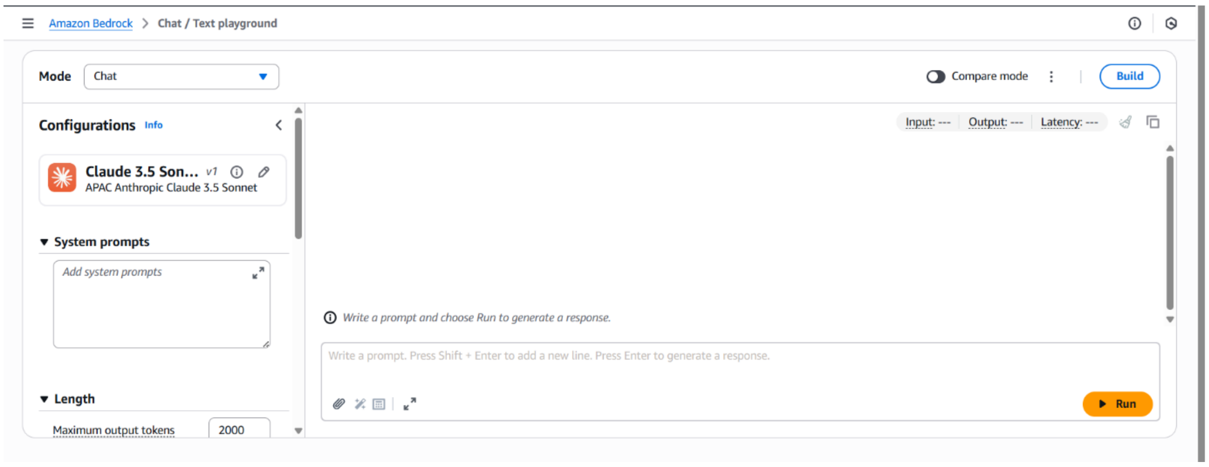
How it works

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

Step - 1: Request Model Access in Bedrock:

1. Go to the Amazon Bedrock Console in your region.
2. In the left menu, choose Model Catalog.
3. Find Anthropic Claude (e.g., Claude 3.5 Sonnet)
4. Click on Modify the model access button.
5. Select the Claude 3.5 Sonnet under Anthropic > Click Next > Submit.

~ After approval, go back to Model access page, verify if the status has changed from "Access required" → "Access granted".



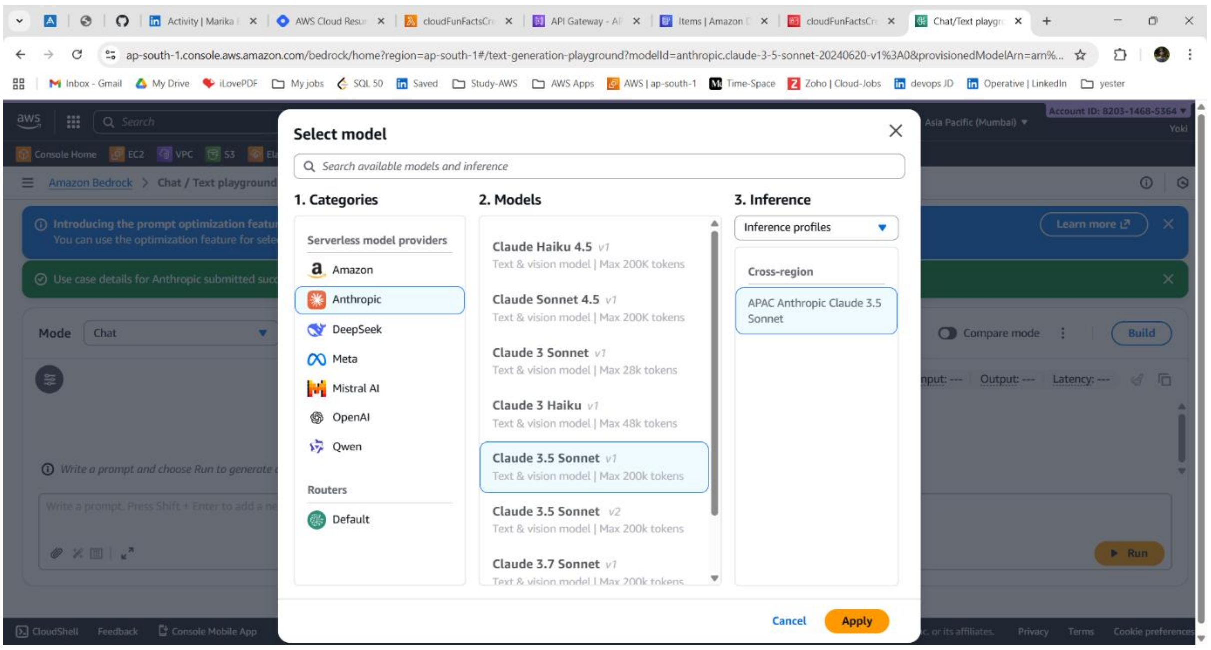
Claude 3.5 Son... v1
APAC Anthropic Claude 3.5 Sonnet

Add system prompts

Maximum output tokens 2000

Write a prompt and choose Run to generate a response.

Write a prompt. Press Shift + Enter to add a new line. Press Enter to generate a response.


Select model

Search available models and inference

1. Categories

Serverless model providers

- Amazon
- Anthropic
- DeepSeek
- Meta
- Mistral AI
- OpenAI
- Qwen

Routers

- Default

2. Models

- Claude Haiku 4.5 v1 — Text & vision model | Max 200K tokens
- Claude Sonnet 4.5 v1 — Text & vision model | Max 200K tokens
- Claude 3 Sonnet v1 — Text & vision model | Max 28k tokens
- Claude 3 Haiku v1 — Text & vision model | Max 48k tokens
- Claude 3.5 Sonnet v1 — Text & vision model | Max 200k tokens
- Claude 3.5 Sonnet v2 — Text & vision model | Max 200k tokens
- Claude 3.7 Sonnet v1 — Text & vision model | Max 200k tokens

3. Inference

Inference profiles

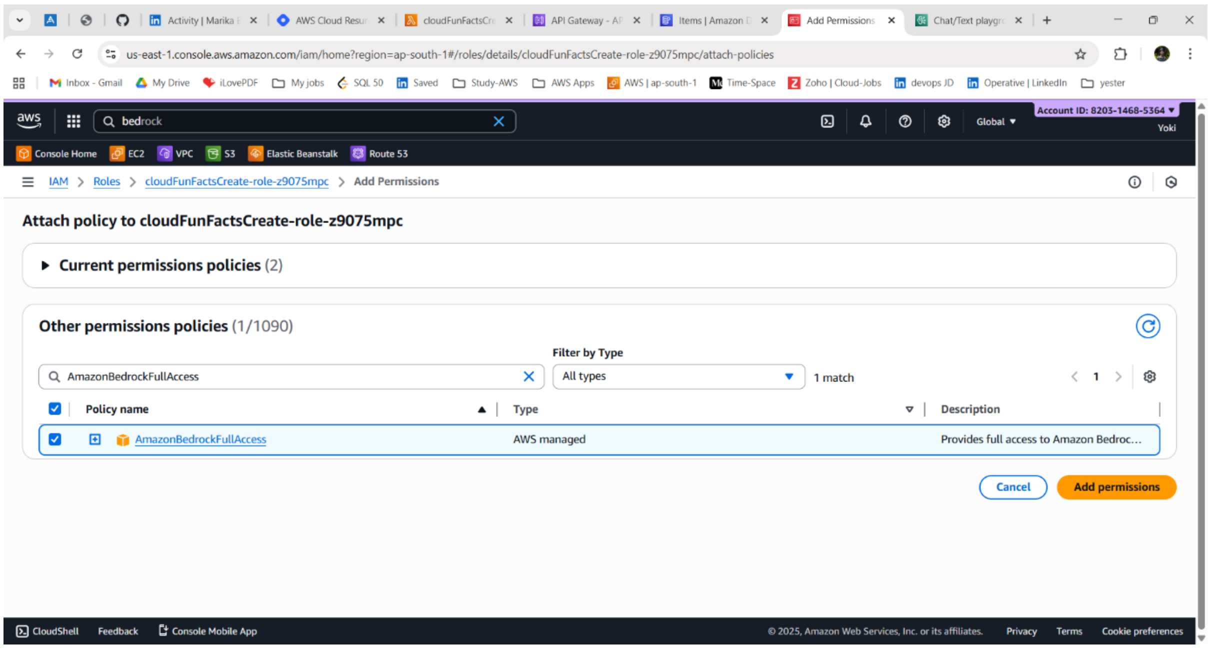
Cross-region

APAC Anthropic Claude 3.5 Sonnet

Cancel Apply

Step - 2: Update IAM Role:

1. Go to IAM Console → **Roles**.
2. Open the role attached to your Lambda function (**CloudFunFactsRole**).
3. Attach the policy: **AmazonBedrockFullAccess** (this allows your Lambda to call Bedrock models).


IAM > Roles > cloudFunFactsCreate-role-z9075mpc > Add Permissions

Attach policy to cloudFunFactsCreate-role-z9075mpc

Current permissions policies (2)

Other permissions policies (1/1090)

Search: AmazonBedrockFullAccess — Filter by Type: All types — 1 match

Policy name	Type	Description
✓ AmazonBedrockFullAccess	AWS managed	Provides full access to Amazon Bedroc…

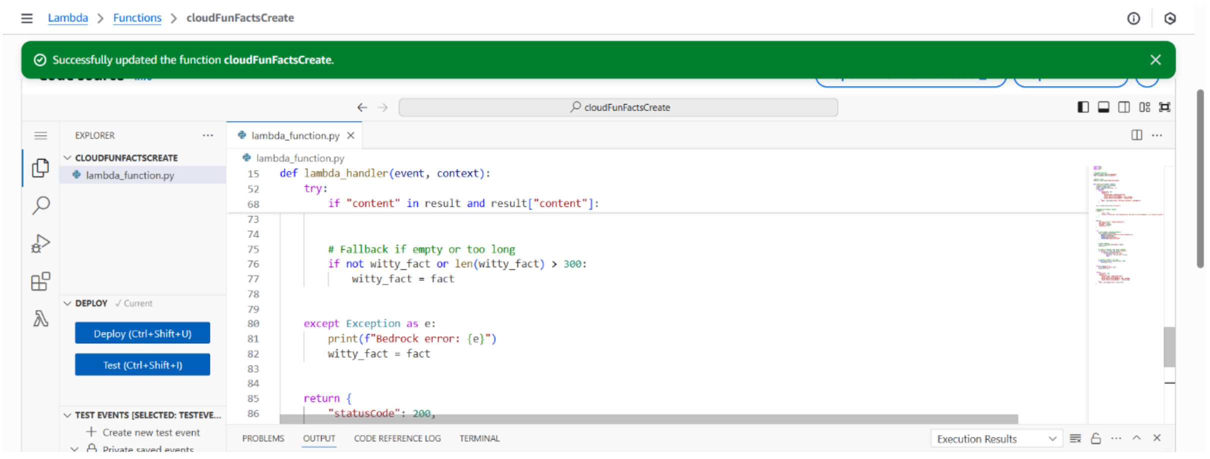
Cancel Add permissions

<u>Step - 3: Update Lambda Code:</u>

1. Replace your **DynamoDB Lambda code** with this:

Lambda updated code:

```
 import boto3
import random

import json

# DynamoDB connection

dynamodb = boto3.resource("dynamodb")

table = dynamodb.Table("CloudFacts")

# Bedrock client

bedrock = boto3.client("bedrock-runtime")

def lambda_handler(event, context):

    # Fetch all facts from DynamoDB

    response = table.scan()

    items = response.get("Items", [])


    if not items:

        return {

            "statusCode": 200,

            "headers": {

                "Content-Type": "application/json",

                "Access-Control-Allow-Origin": "*",

                "Access-Control-Allow-Methods": "GET, OPTIONS",

                "Access-Control-Allow-Headers": "Content-Type"

            },

            "body": json.dumps({"fact": "No facts available in DynamoDB."})

        }
    fact = random.choice(items)["FactText"]
    # Messages for Claude 3.5 Sonnet
    messages = [
        {
            "role": "user",
            "content": f"Take this cloud computing fact and make it fun and engaging in 1-2 sentences
maximum. Keep it short and witty: {fact}"
        }
    ]
    body = {
        "anthropic_version": "bedrock-2023-05-31",
        "max_tokens": 100,
        "messages": messages,
        "temperature": 0.7
    }
    try:
        # Call Claude 3.5 Sonnet on Bedrock
        resp = bedrock.invoke_model(
            modelId="anthropic.claude-3-5-sonnet-20240620-v1:0",
            body=json.dumps(body),
            accept="application/json",
            contentType="application/json"
        )

        # Parse response
        result = json.loads(resp["body"].read())
        witty_fact = ""
        # Claude v3 response: look inside "content"
        if "content" in result and result["content"]:
            for block in result["content"]:
                if block.get("type") == "text":
                    witty_fact = block["text"].strip()
                    break
        # Fallback if empty or too long
        if not witty_fact or len(witty_fact) > 300:
            witty_fact = fact
    except Exception as e:
        print(f"Bedrock error: {e}")
        witty_fact = fact
    return {
        "statusCode": 200,
        "headers": {
            "Content-Type": "application/json",
            "Access-Control-Allow-Origin": "*",
            "Access-Control-Allow-Methods": "GET, OPTIONS",
            "Access-Control-Allow-Headers": "Content-Type"
        },
        "body": json.dumps({"fact": witty_fact})
    }
```



Lambda > Functions > cloudFunFactsCreate

Successfully updated the function **cloudFunFactsCreate**.

```
lambda_function.py
def lambda_handler(event, context):
    try:
        if "content" in result and result["content"]:


        # Fallback if empty or too long
        if not witty_fact or len(witty_fact) > 300:
            witty_fact = fact


    except Exception as e:
        print(f"Bedrock error: {e}")
        witty_fact = fact


    return {
        "statusCode": 200,
```

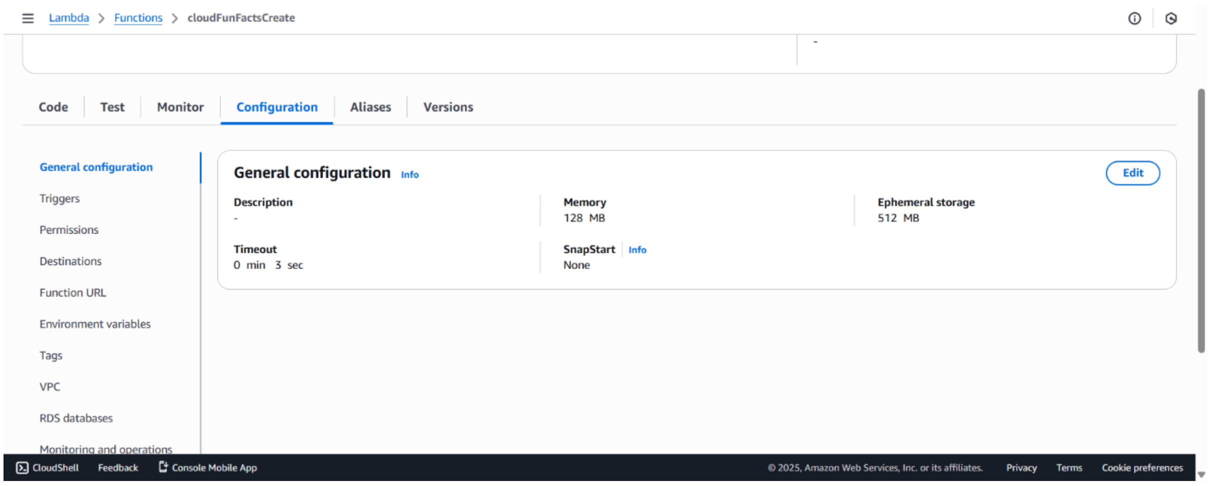
2. Click deploy to save the code.

3. Increase Lambda Timeout

Go to Configuration → General Configuration → Edit

4. Set Timeout to 15–30 seconds (or more if your model takes longer to respond)

5. Save changes

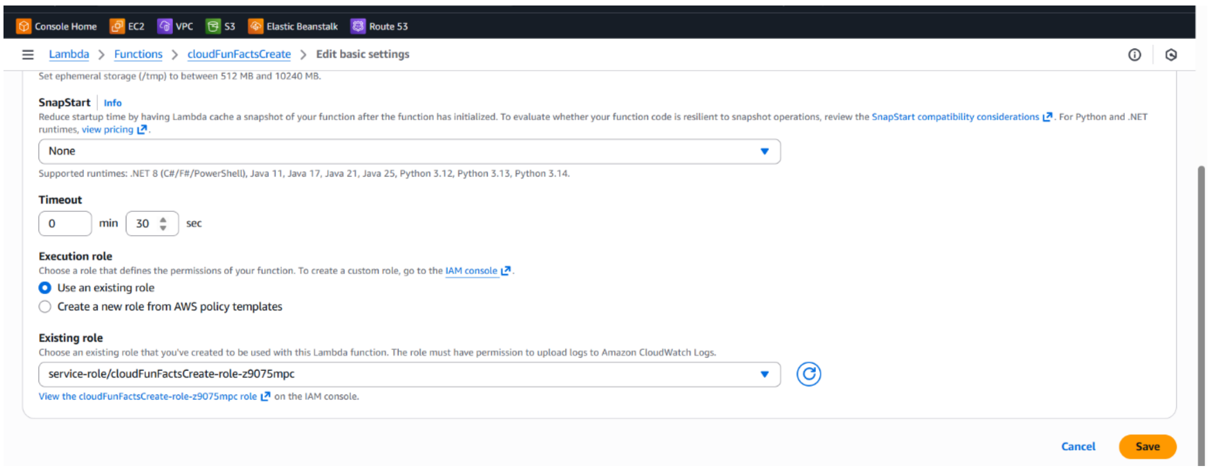


Lambda > Functions > cloudFunFactsCreate

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration

Description: -
Memory: 128 MB
Ephemeral storage: 512 MB
Timeout: 0 min 3 sec
SnapStart: None



Lambda > Functions > cloudFunFactsCreate > Edit basic settings

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations. For Python and .NET runtimes, view pricing.
None
Supported runtimes: .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Java 25, Python 3.12, Python 3.13, Python 3.14.

Timeout
0 min 30 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
[✓ Use an existing role]
Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/cloudFunFactsCreate-role-z9075mpc
View the cloudFunFactsCreate-role-z9075mpc role on the IAM console.

<u>Step - 4: Test the GenAI-Powered Lambda:</u>

1. Go to Lambda → Select your function → Test.

2. Run the function a few times.

3. Instead of plain facts, you'll see playful AI-generated versions of the same facts.

Example:

Input fact: "AWS S3 was launched in 2006."

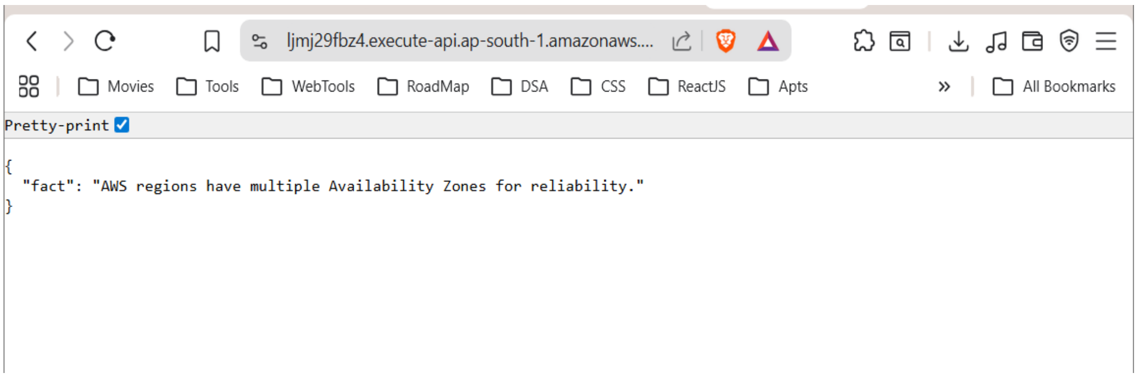
Output witty fact: "AWS S3 has been around since 2006, older than the first iPhone, and still storing your selfies!"

Step - 5: Test the API:

1. Go back to **API Gateway.**

2. Hit your endpoint:

Copy the URL (e.g., https://abcdedf.execute-api.ap-south -1.amazonaws.com).



```
{
  "fact": "AWS regions have multiple Availability Zones for reliability."
}
```

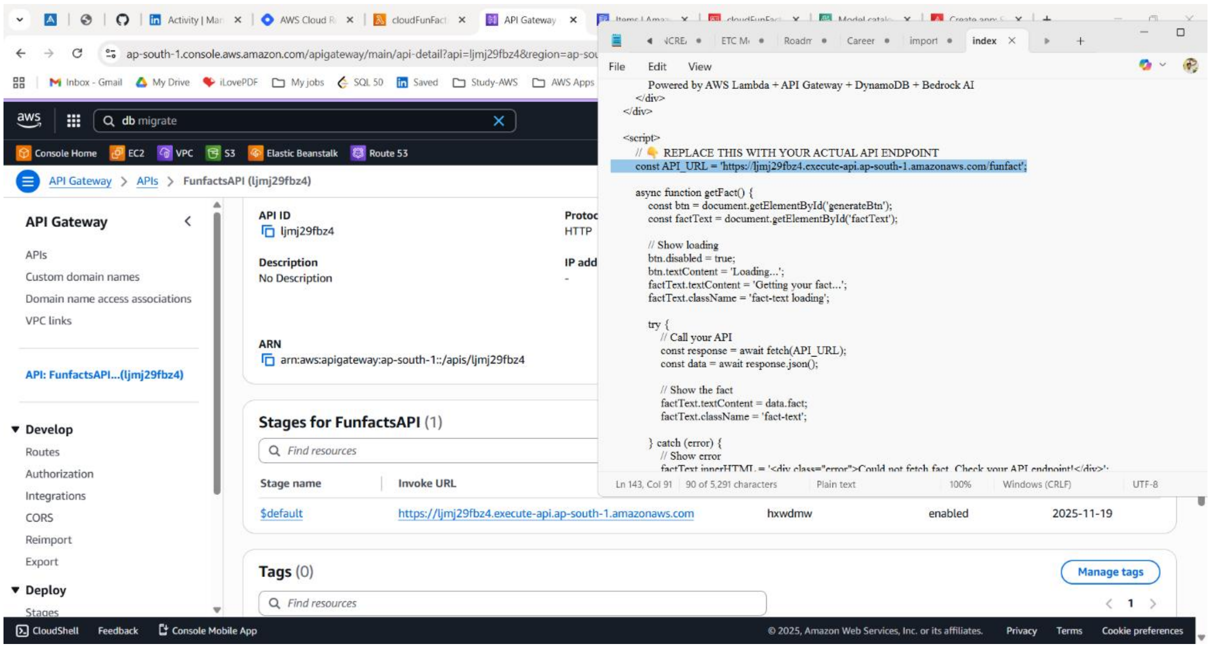
This time, the response should come as a witty fact from Bedrock.

Example:

{

  "fact": "NASA uses AWS to store and share Mars mission data with the public."

}



API ID: ljmj29fbz4

Description: No Description

Protocol: HTTP

ARN: arn:aws:apigateway:ap-south-1::/apis/ljmj29fbz4

Stages for FunfactsAPI (1)

Stage name	Invoke URL			
$default	https://ljmj29fbz4.execute-api.ap-south-1.amazonaws.com	hxwdmw	enabled	2025-11-19

Tags (0)

```
      Powered by AWS Lambda + API Gateway + DynamoDB + Bedrock AI
    </div>
  </div>

  <script>
    // 👇 REPLACE THIS WITH YOUR ACTUAL API ENDPOINT
    const API_URL = 'https://ljmj29fbz4.execute-api.ap-south-1.amazonaws.com/funfact';

    async function getFact() {
      const btn = document.getElementById('generateBtn');
      const factText = document.getElementById('factText');

      // Show loading
      btn.disabled = true;
      btn.textContent = 'Loading...';
      factText.textContent = 'Getting your fact...';
      factText.className = 'fact-text loading';

      try {
        // Call your API
        const response = await fetch(API_URL);
        const data = await response.json();

        // Show the fact
        factText.textContent = data.fact;
        factText.className = 'fact-text';

      } catch (error) {
        // Show error
        factText.innerHTML = '<div class="error">Could not fetch fact. Check your API endpoint!</div>';
```

Ln 143, Col 91 | 90 of 5,291 characters | Plain text | 100% | Windows (CRLF) | UTF-8

Outcome of Stage - 3:

→ At this point, you've just turned your Fun Facts API into a Generative AI-powered service

~ It creates an **AI-generated version** of this content where the facts stay accurate, but the delivery is **clever, entertaining, or humorous.**

DEPLOYING APPLICATION:

~ In the previous stages, we built a complete **serverless API** that evolved from hardcoded facts → database-driven → AI-enhanced. Now you'll create a beautiful **web frontend** so users can interact with your Cloud Fun Facts API through a browser.

1. Create frontend file:

Make `index.html` → paste the provided HTML/JS code → replace `API_URL` with your real API endpoint.

2. Get your API URL:

In API Gateway → Stages → default → copy the Invoke URL → ensure it ends with `/funfact`.

3. Enable CORS:

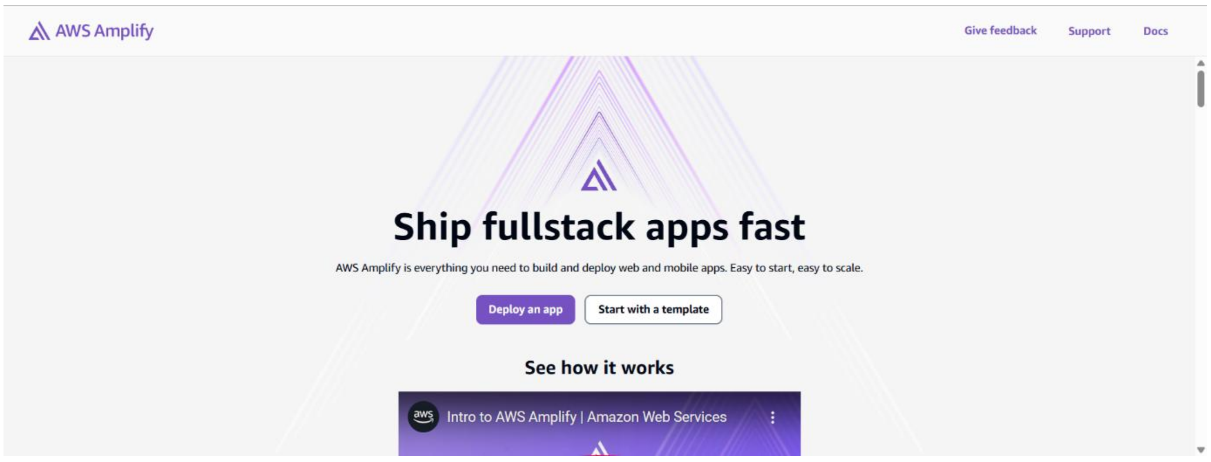
In API Gateway → CORS → allow your Amplify domain (or `*` for testing) → allow GET/OPTIONS → save.

4. Deploy to AWS Amplify:

Go to AWS Amplify → Host web app → "Deploy without Git" → upload a ZIP containing `index.html`.

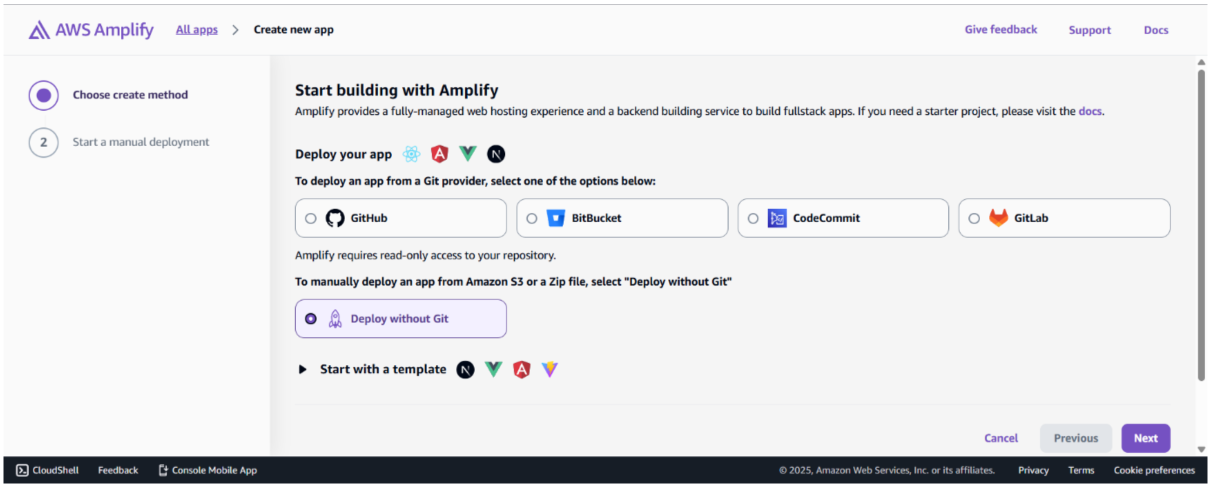
5. Test the app:

Open the Amplify URL → click "Generate Fun Fact" → confirm it loads facts from your API.


AWS Amplify | Give feedback | Support | Docs

Ship fullstack apps fast

AWS Amplify is everything you need to build and deploy web and mobile apps. Easy to start, easy to scale.


AWS Amplify | All apps > Create new app | Give feedback | Support | Docs

Start building with Amplify

Amplify provides a fully-managed web hosting experience and a backend building service to build fullstack apps. If you need a starter project, please visit the docs.

Deploy your app

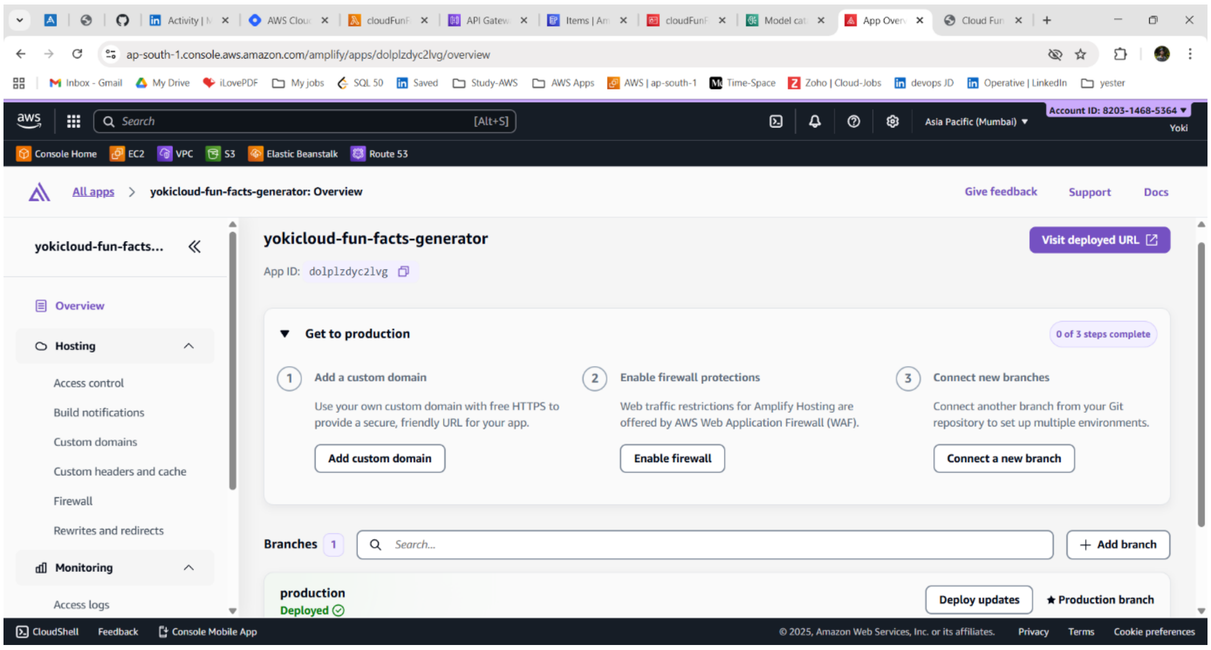
To deploy an app from a Git provider, select one of the options below:

Amplify requires read-only access to your repository.

To manually deploy an app from Amazon S3 or a Zip file, select "Deploy without Git"

[✓ Deploy without Git]

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



ap-south-1.console.aws.amazon.com/amplify/apps/dolplzdyc2lvg/overview

All apps > yokicloud-fun-facts-generator: Overview

yokicloud-fun-facts-generator

App ID: dolplzdyc2lvg

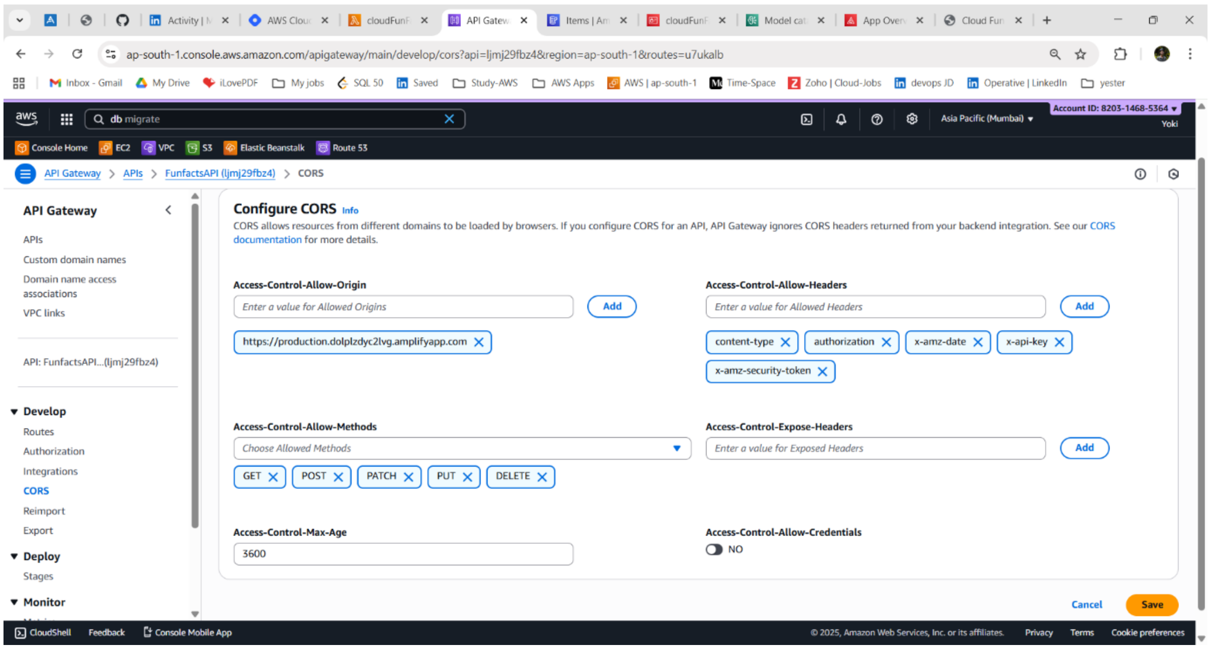
Visit deployed URL

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

production — Deployed — Production branch



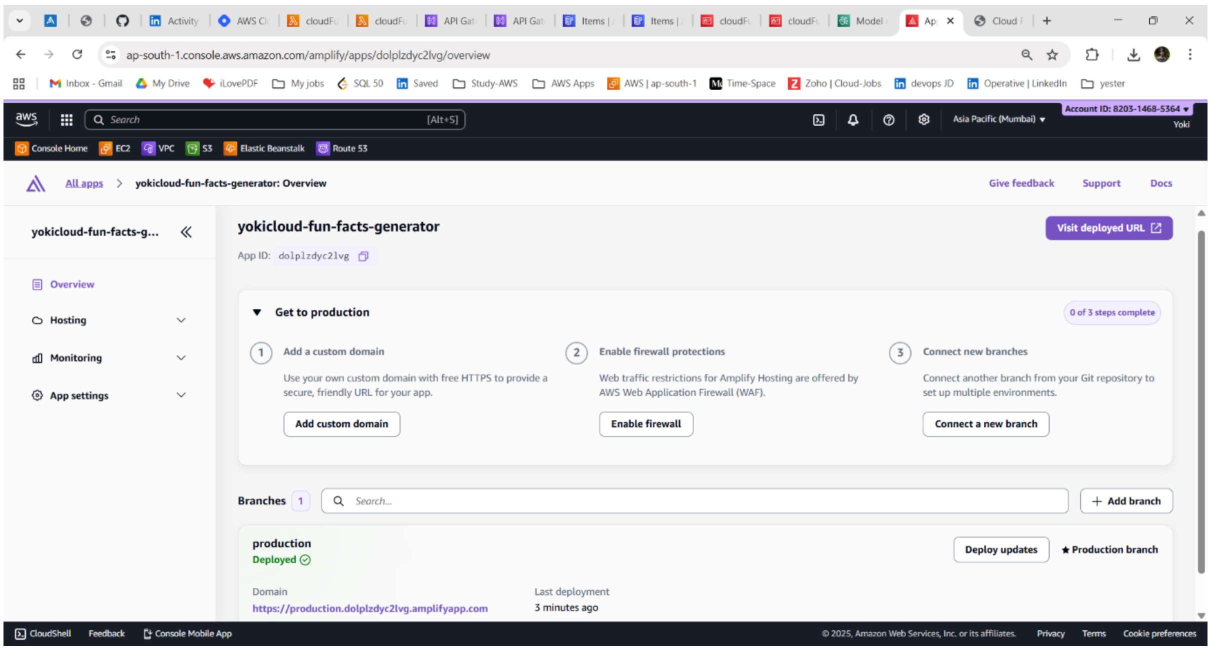
ap-south-1.console.aws.amazon.com/apigateway/main/develop/cors?api=ljmj29fbz4&region=ap-south-1&routes=u7ukalb

API Gateway > APIs > FunfactsAPI (ljmj29fbz4) > CORS

Configure CORS

CORS allows resources from different domains to be loaded by browsers. If you configure CORS for an API, API Gateway ignores CORS headers returned from your backend integration. See our CORS documentation for more details.

Access-Control-Allow-Origin	https://production.dolplzdyc2lvg.amplifyapp.com
Access-Control-Allow-Headers	content-type, authorization, x-amz-date, x-api-key, x-amz-security-token
Access-Control-Allow-Methods	GET, POST, PATCH, PUT, DELETE
Access-Control-Expose-Headers	
Access-Control-Max-Age	3600
Access-Control-Allow-Credentials	NO


ap-south-1.console.aws.amazon.com/amplify/apps/dolplzdyc2lvg/overview

All apps > yokicloud-fun-facts-generator: Overview

yokicloud-fun-facts-generator

App ID: dolplzdyc2lvg

Get to production — 0 of 3 steps complete

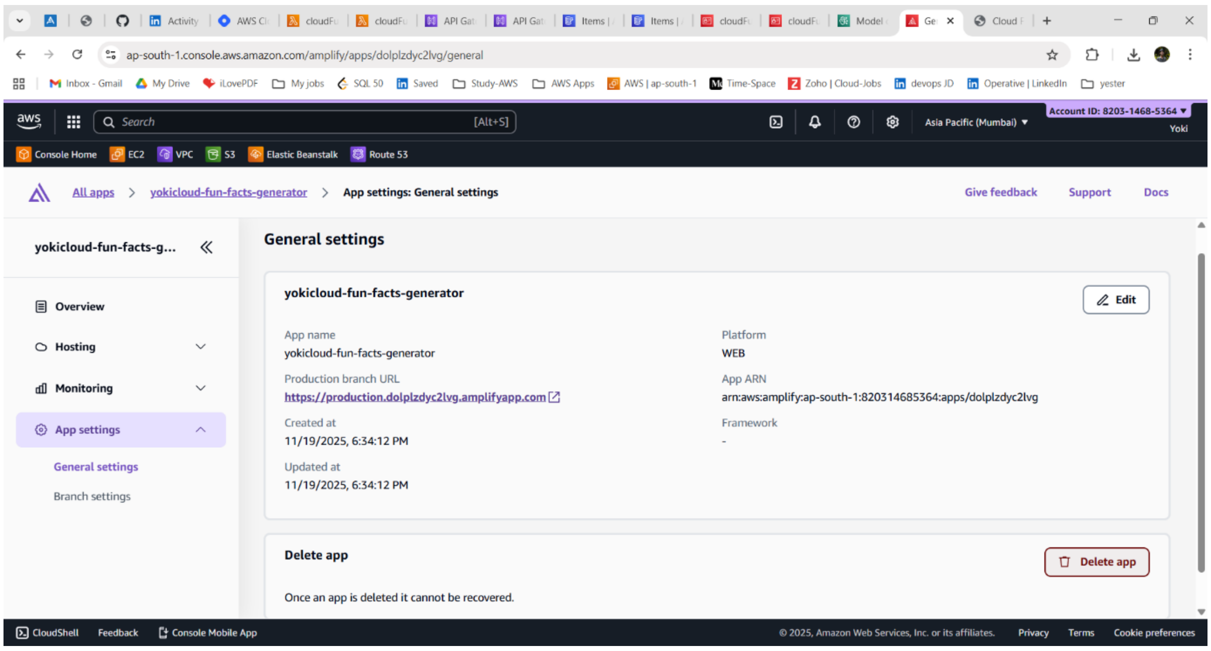
1. Add a custom domain
   Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app.
   Add custom domain
2. Enable firewall protections
   Web traffic restrictions for Amplify Hosting are offered by AWS Web Application Firewall (WAF).
   Enable firewall
3. Connect new branches
   Connect another branch from your Git repository to set up multiple environments.
   Connect a new branch

Branches 1

production
Deployed

Domain: https://production.dolplzdyc2lvg.amplifyapp.com
Last deployment: 3 minutes ago

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


ap-south-1.console.aws.amazon.com/amplify/apps/dolplzdyc2lvg/general

All apps > yokicloud-fun-facts-generator > App settings: General settings

General settings

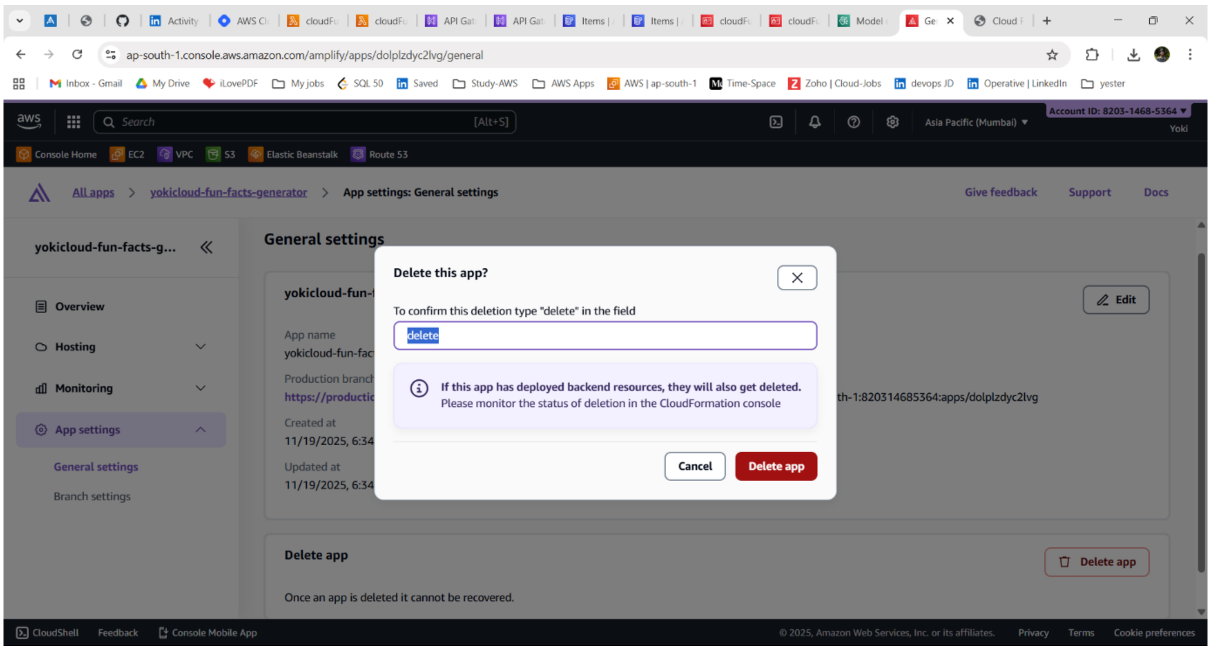
yokicloud-fun-facts-generator

App name	Platform
yokicloud-fun-facts-generator	WEB
Production branch URL	App ARN
https://production.dolplzdyc2lvg.amplifyapp.com	arn:aws:amplify:ap-south-1:820314685364:apps/dolplzdyc2lvg
Created at	Framework
11/19/2025, 6:34:12 PM	-
Updated at	
11/19/2025, 6:34:12 PM	

Delete app

Once an app is deleted it cannot be recovered.

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



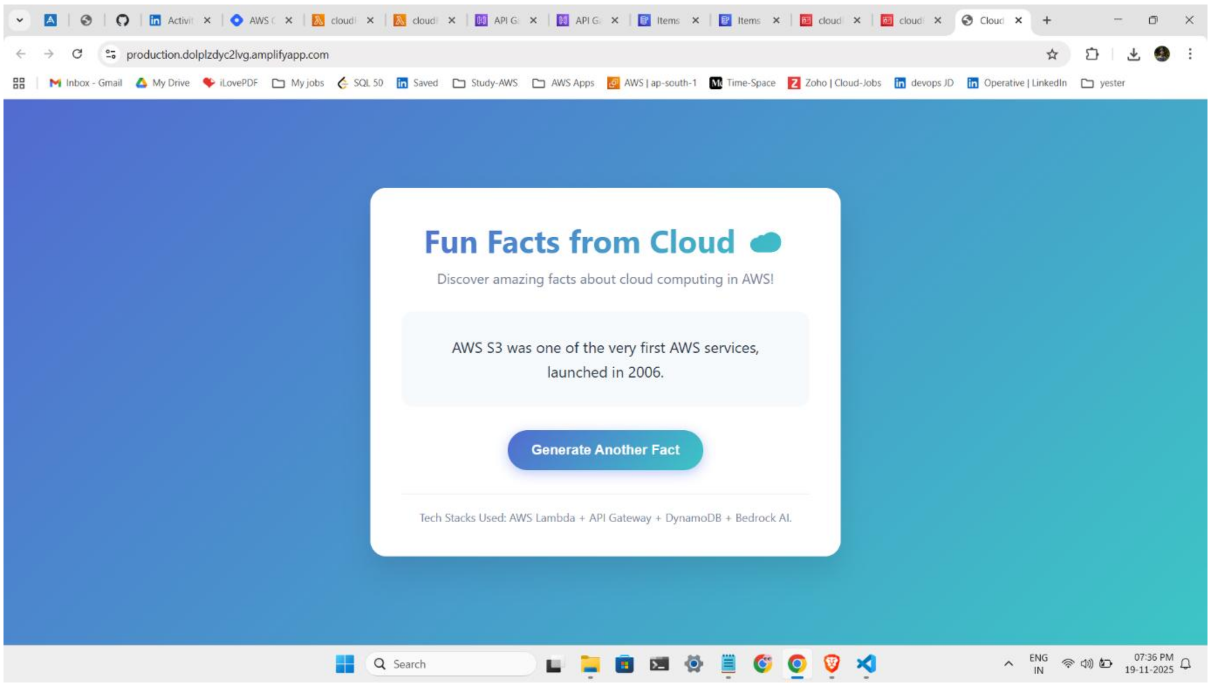
Delete this app?

To confirm this deletion type "delete" in the field

delete

If this app has deployed backend resources, they will also get deleted.
Please monitor the status of deletion in the CloudFormation console

All apps > yokicloud-fun-facts-generator > App settings: General settings
General settings
Overview
Hosting
Monitoring
App settings
General settings
Branch settings
Cancel
Delete app
Delete app
Once an app is deleted it cannot be recovered.



Fun Facts from Cloud

Discover amazing facts about cloud computing in AWS!

AWS S3 was one of the very first AWS services, launched in 2006.

Generate Another Fact

Tech Stacks Used: AWS Lambda + API Gateway + DynamoDB + Bedrock AI.