Project Report

Blog-GPT : AI Powered Blog Generator

Sanskar Sharma (E22CSEU1008)

A report submitted in part fulfilment of the degree of

B.Tech in Computer Science

Supervisor: Dr. Naween Kumar

School of Computer Science Engineering and Technology
Bennett University

1. Introduction

1.1 Overview

Blog GPT is a serverless web application that enables users to generate AI-written blogs on custom topics and translate them into multiple languages. The system is built using AWS services like Amazon API Gateway, AWS Lambda, Amazon Bedrock (for content generation), AWS Translate (for translation), and Amazon S3 (for storage). Users interact with a frontend interface hosted locally to enter their blog topic and select a preferred language.

.

1.2 Motivation

Creating consistent, engaging content for blogs is a challenge faced by content creators, marketing teams, and individuals. Language barriers further complicate content outreach on a global scale. Manual blog writing is time-consuming, requires creativity, and often demands professional translation for multilingual distribution. Blog GPT addresses these challenges by offering a one-click solution that generates AI-based content and translates it, allowing creators to focus on strategy and engagement rather than production. Its cloud-native architecture ensures ease of deployment and maintenance.

.

2. Key Features

- **AI-Powered Blog Generation:** The user enters a prompt or topic, and Amazon Bedrock—leveraging large language models (LLMs) like Claude or Titan—generates a coherent, structured, and grammatically correct blog.
- **Multilingual Translation:** Once the content is generated, AWS Translate is used to convert the blog into various languages with high contextual accuracy and cultural relevance, expanding reach across geographies.
- **Local Web Interface:** The application includes a lightweight frontend that enables user interaction. It provides a clean form to input topics, choose target languages, and display results dynamically.
- **Serverless Architecture:** The use of AWS Lambda and API Gateway ensures the application can scale on-demand, reducing costs when idle and handling sudden traffic spikes without infrastructure concerns.
- **S3 Integration:** Generated and translated blogs are stored in Amazon S3 buckets, organized into folders, and accessible via pre-signed URLs. This ensures secure, durable, and versioned content storage.
- **Extensibility:** The modular design allows easy future expansion, such as adding sentiment control, keyword targeting, or integration with CMS platforms like WordPress or Ghost.

3. System Architecture

3.1 High-Level Architecture Diagram

- **Frontend:** Built using HTML, CSS, and JavaScript (or React), running on localhost for now, with plans for S3 deployment. It sends user input to the backend and displays generated blogs.
- **API Gateway:** Handles RESTful communication between frontend and AWS Lambda, ensuring secure and structured request flow.
- **AWS Lambda:** Acts as the core processing unit, invoking Amazon Bedrock to generate content and AWS Translate for multilingual conversion.
- **Amazon Bedrock:** Uses foundation models like Claude or Titan to generate human-like text based on user-provided prompts.
- **AWS Translate:** Offers real-time, accurate translation of the blog into a wide range of languages.
- **Amazon S3:** Provides reliable, scalable storage for blog files. It categorizes data into 'original' and 'translated' folders and returns URLs for frontend access.

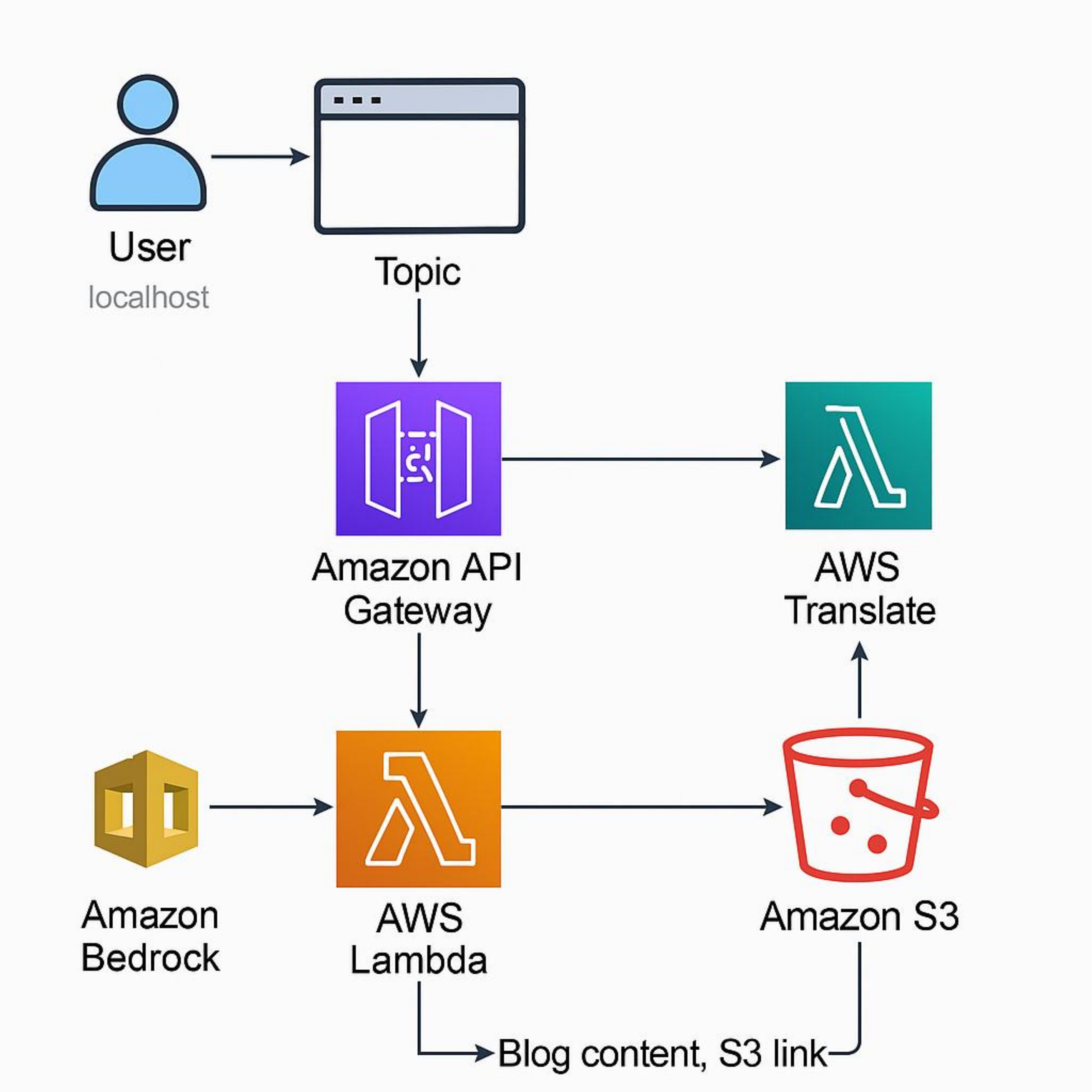
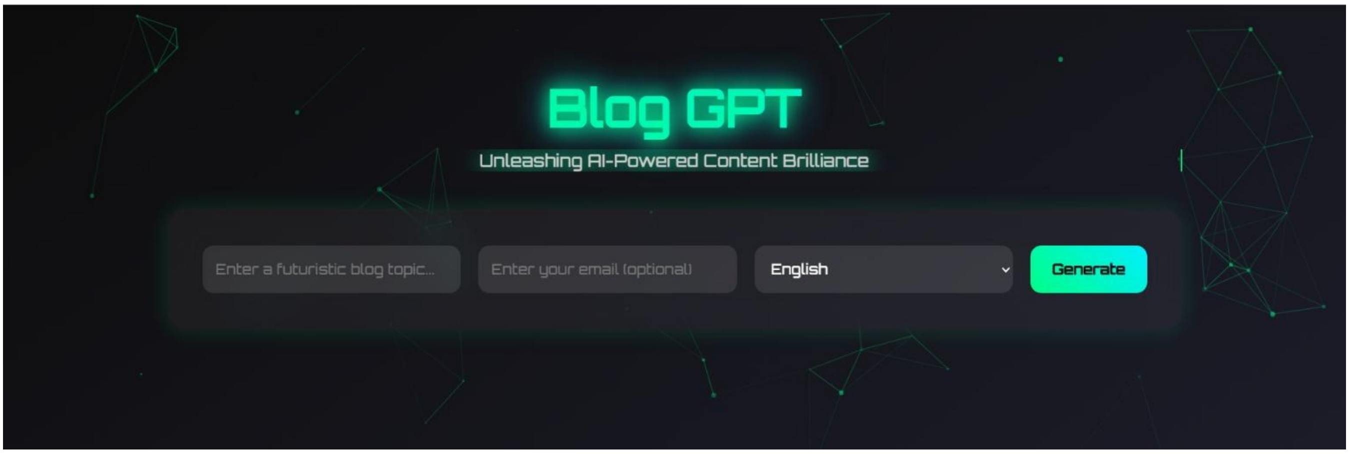


Figure 1: Project Architecture

3.2 Workflow

1. User inputs a topic and chooses a target language (optional).



Blog GPT

Unleashing AI-Powered Content Brilliance

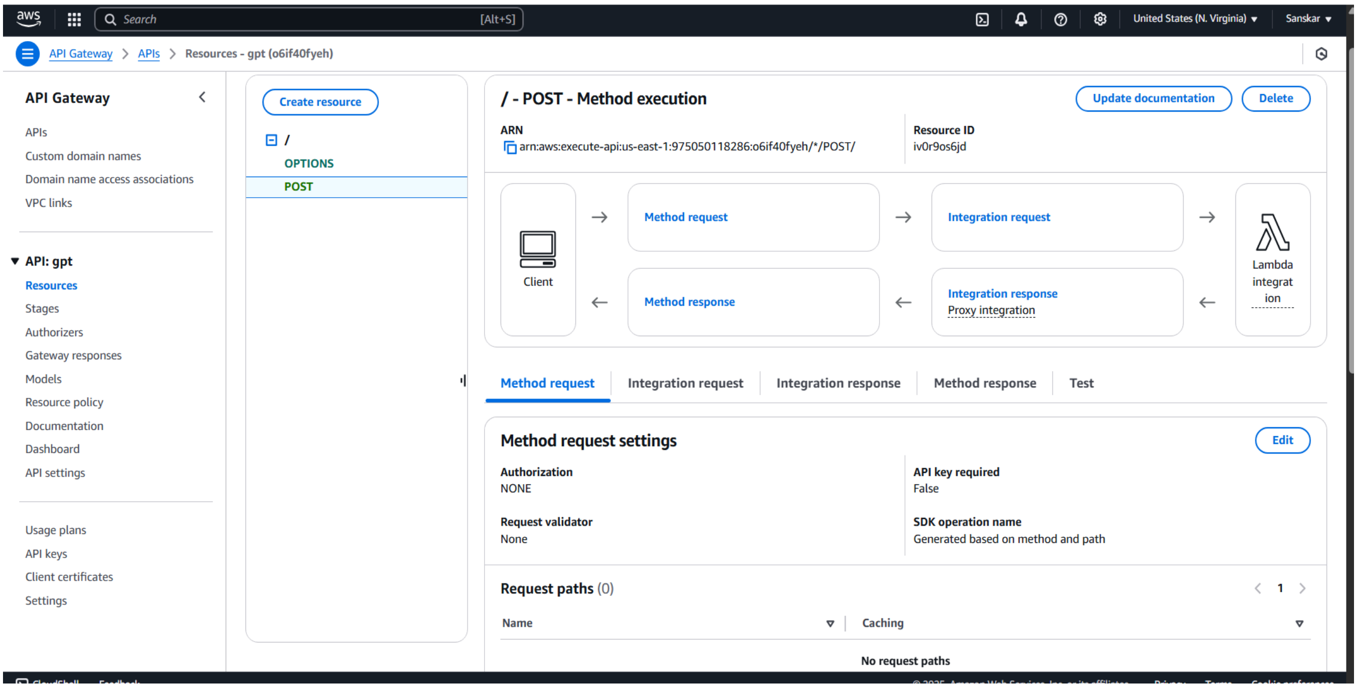
Enter a futuristic blog topic...

Enter your email (optional)

English

Generate

2. Frontend makes a POST request to API Gateway.



API Gateway > APIs > Resources - gpt (o6if40fyeh)

API Gateway

- APIs
- Custom domain names
- Domain name access associations
- VPC links

API: gpt

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

- Usage plans
- API keys
- Client certificates
- Settings

Create resource

/
OPTIONS
POST

/ - POST - Method execution

Update documentation · Delete

ARN
arn:aws:execute-api:us-east-1:975050118286:o6if40fyeh/*/POST/

Resource ID
iv0r9os6jd

Client → Method request → Integration request → Lambda integration

Client ← Method response ← Integration response (Proxy integration) ← Lambda integration

Method request · Integration request · Integration response · Method response · Test

Method request settings

Edit

Authorization
NONE

API key required
False

Request validator
None

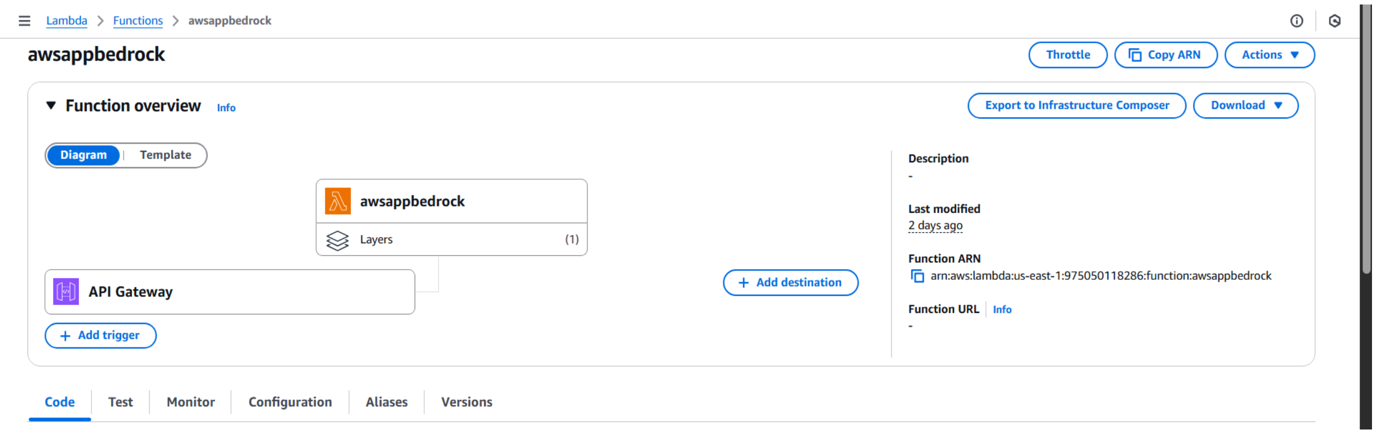
SDK operation name
Generated based on method and path

Request paths (0)

Name	Caching

No request paths

3. API Gateway routes the request to an AWS Lambda function.



Lambda > Functions > awsappbedrock

awsappbedrock

Throttle · Copy ARN · Actions

Function overview Info

Export to Infrastructure Composer · Download

Diagram · Template

awsappbedrock
Layers (1)

API Gateway

+ Add trigger

+ Add destination

Description
-

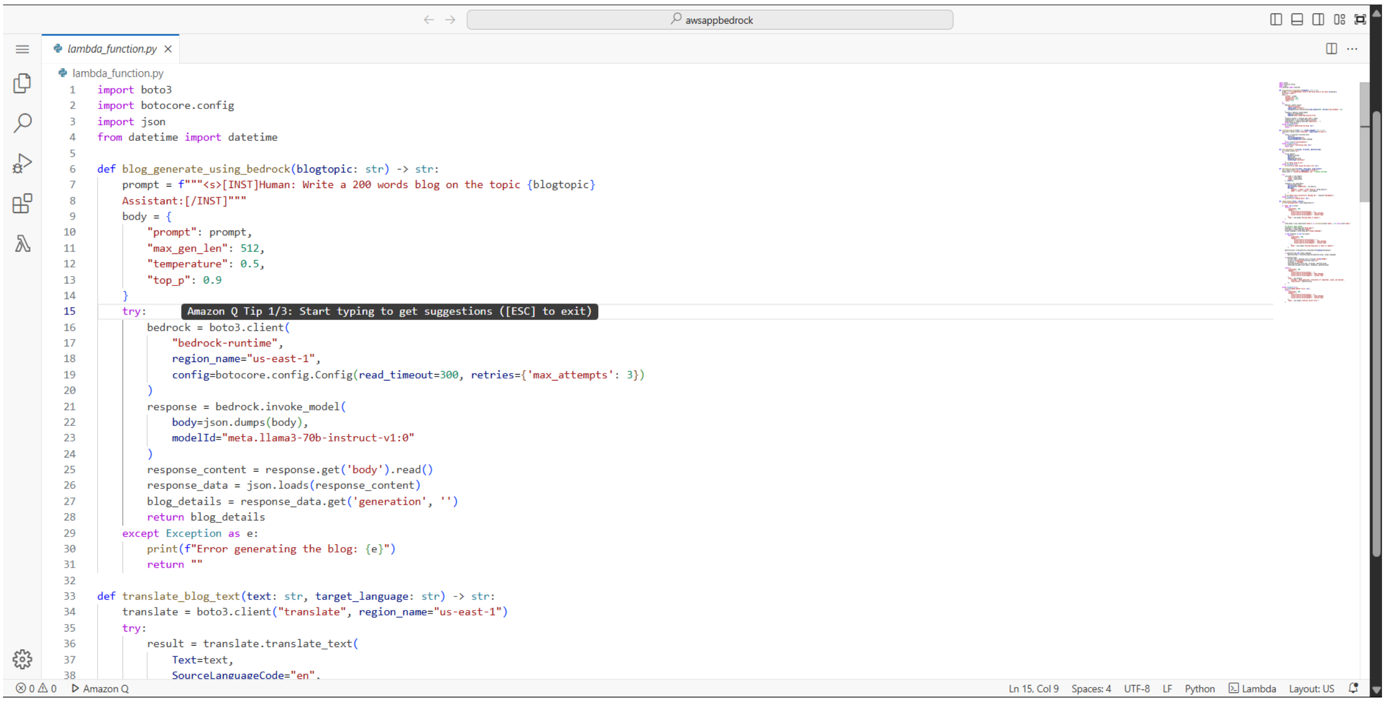
Last modified
2 days ago

Function ARN
arn:aws:lambda:us-east-1:975050118286:function:awsappbedrock

Function URL Info
-

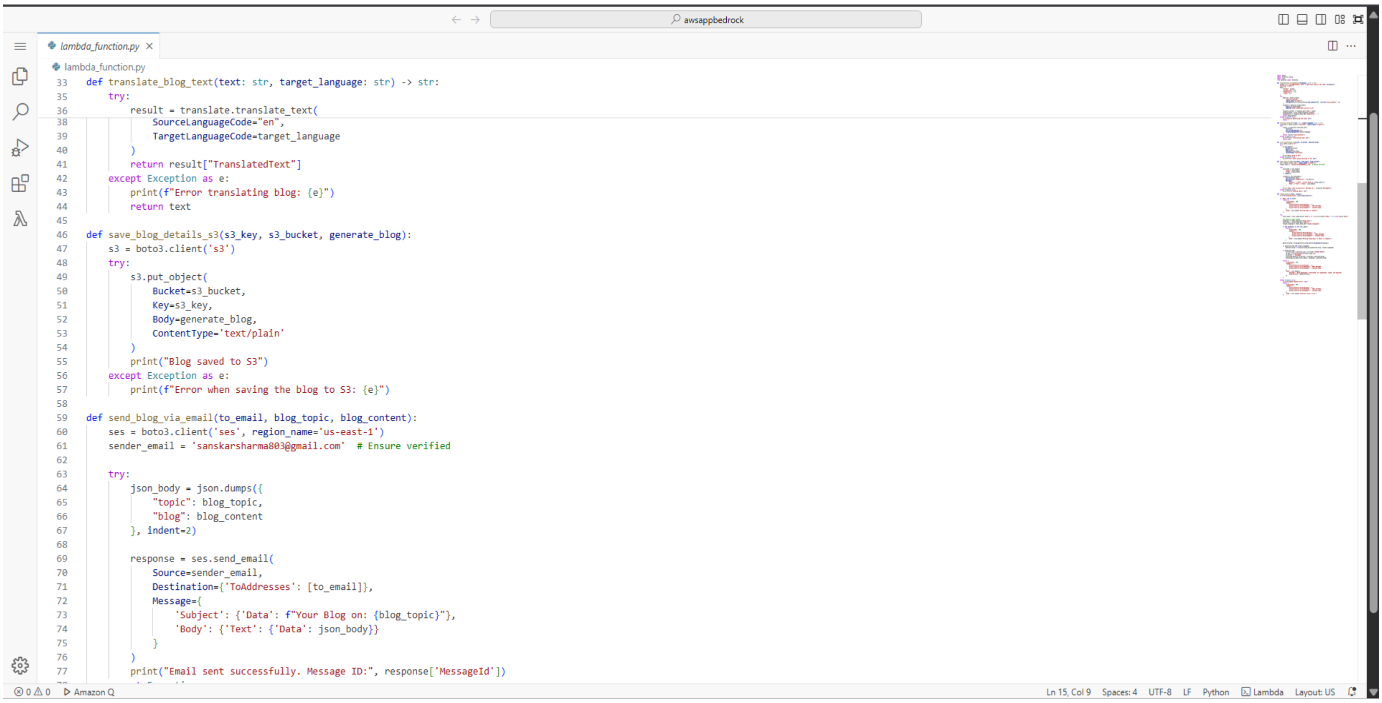
Code · Test · Monitor · Configuration · Aliases · Versions

4. If translation is needed, Lambda passes the blog to AWS Translate



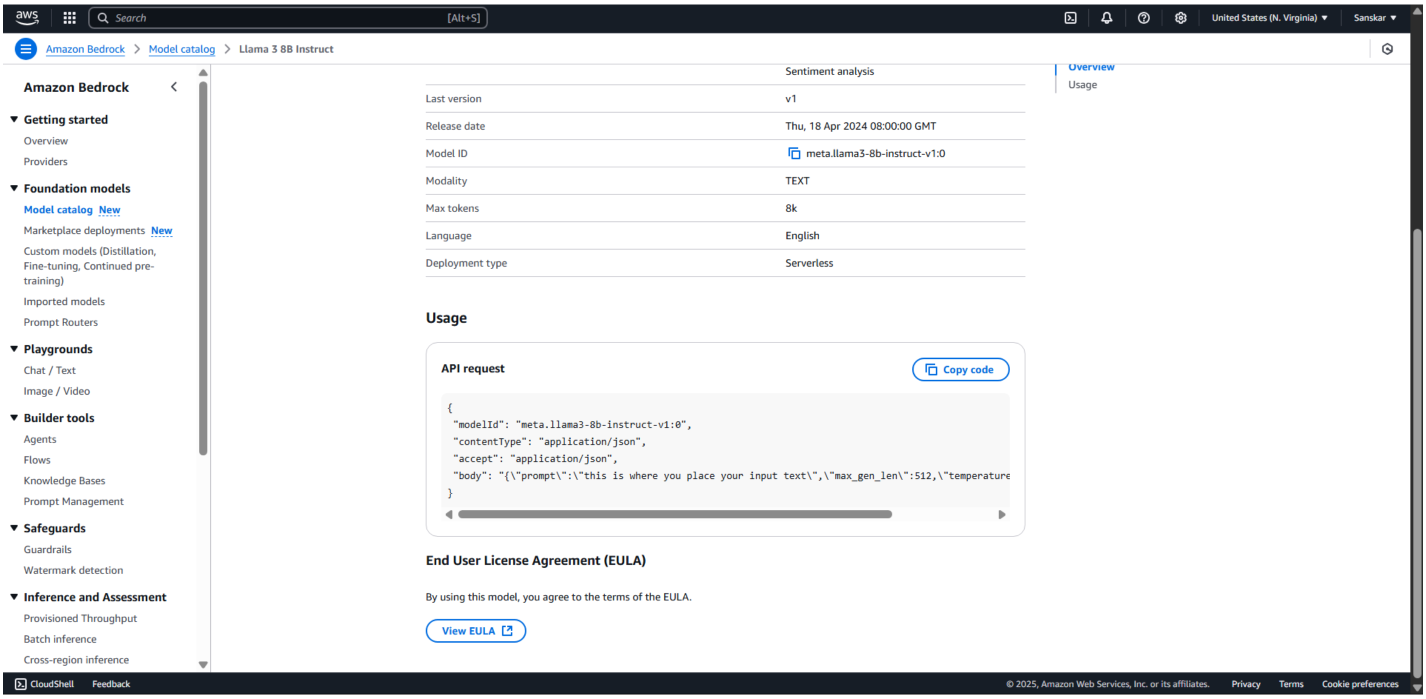
```
import boto3
import botocore.config
import json
from datetime import datetime

def blog_generate_using_bedrock(blogtopic: str) -> str:
    prompt = f"""<s>[INST]Human: Write a 200 words blog on the topic {blogtopic}
    Assistant:[/INST]"""
    body = {
        "prompt": prompt,
        "max_gen_len": 512,
        "temperature": 0.5,
        "top_p": 0.9
    }
    try:
        bedrock = boto3.client(
            "bedrock-runtime",
            region_name="us-east-1",
            config=botocore.config.Config(read_timeout=300, retries={'max_attempts': 3})
        )
        response = bedrock.invoke_model(
            body=json.dumps(body),
            modelId="meta.llama3-70b-instruct-v1:0"
        )
        response_content = response.get('body').read()
        response_data = json.loads(response_content)
        blog_details = response_data.get('generation', '')
        return blog_details
    except Exception as e:
        print(f"Error generating the blog: {e}")
        return ""

def translate_blog_text(text: str, target_language: str) -> str:
    translate = boto3.client("translate", region_name="us-east-1")
    try:
        result = translate.translate_text(
            Text=text,
            SourceLanguageCode="en",
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)



```
def translate_blog_text(text: str, target_language: str) -> str:
    try:
        result = translate.translate_text(
            SourceLanguageCode="en",
            TargetLanguageCode=target_language
        )
        return result["TranslatedText"]
    except Exception as e:
        print(f"Error translating blog: {e}")
        return text

def save_blog_details_s3(s3_key, s3_bucket, generate_blog):
    s3 = boto3.client('s3')
    try:
        s3.put_object(
            Bucket=s3_bucket,
            Key=s3_key,
            Body=generate_blog,
            ContentType='text/plain'
        )
        print("Blog saved to S3")
    except Exception as e:
        print(f"Error when saving the blog to S3: {e}")

def send_blog_via_email(to_email, blog_topic, blog_content):
    ses = boto3.client('ses', region_name='us-east-1')
    sender_email = 'sanskarsharma803@gmail.com'  # Ensure verified

    try:
        json_body = json.dumps({
            "topic": blog_topic,
            "blog": blog_content
        }, indent=2)

        response = ses.send_email(
            Source=sender_email,
            Destination={'ToAddresses': [to_email]},
            Message={
                'Subject': {'Data': f"Your Blog on: {blog_topic}"},
                'Body': {'Text': {'Data': json_body}}
            }
        )
        print("Email sent successfully. Message ID:", response['MessageId'])
```

5. Lambda triggers Amazon Bedrock to generate the blog.



	Sentiment analysis
Last version	v1
Release date	Thu, 18 Apr 2024 08:00:00 GMT
Model ID	meta.llama3-8b-instruct-v1:0
Modality	TEXT
Max tokens	8k
Language	English
Deployment type	Serverless

Usage

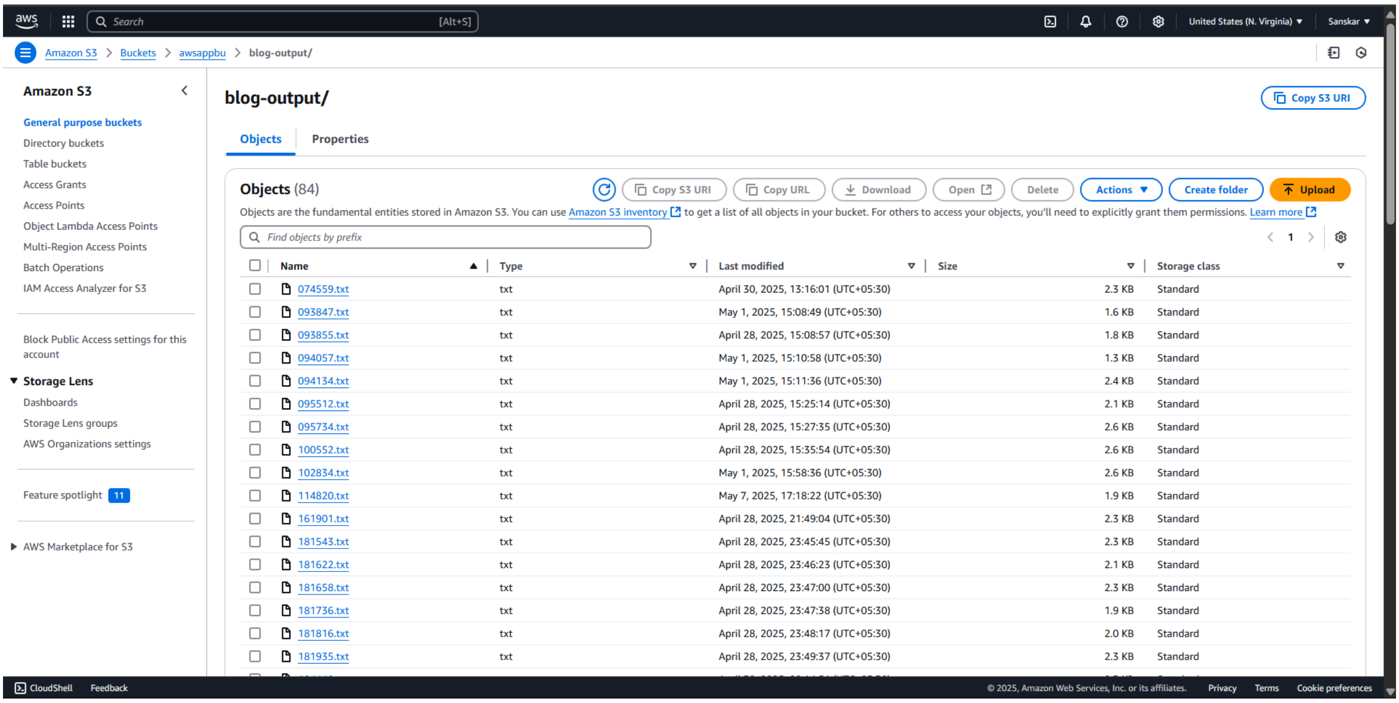
API request

```
{
 "modelId": "meta.llama3-8b-instruct-v1:0",
 "contentType": "application/json",
 "accept": "application/json",
 "body": "{\"prompt\":\"this is where you place your input text\",\"max_gen_len\":512,\"temperature
}
```

End User License Agreement (EULA)

By using this model, you agree to the terms of the EULA.

6. Lambda saves the final content in S3 and retrieves a pre-signed URL.



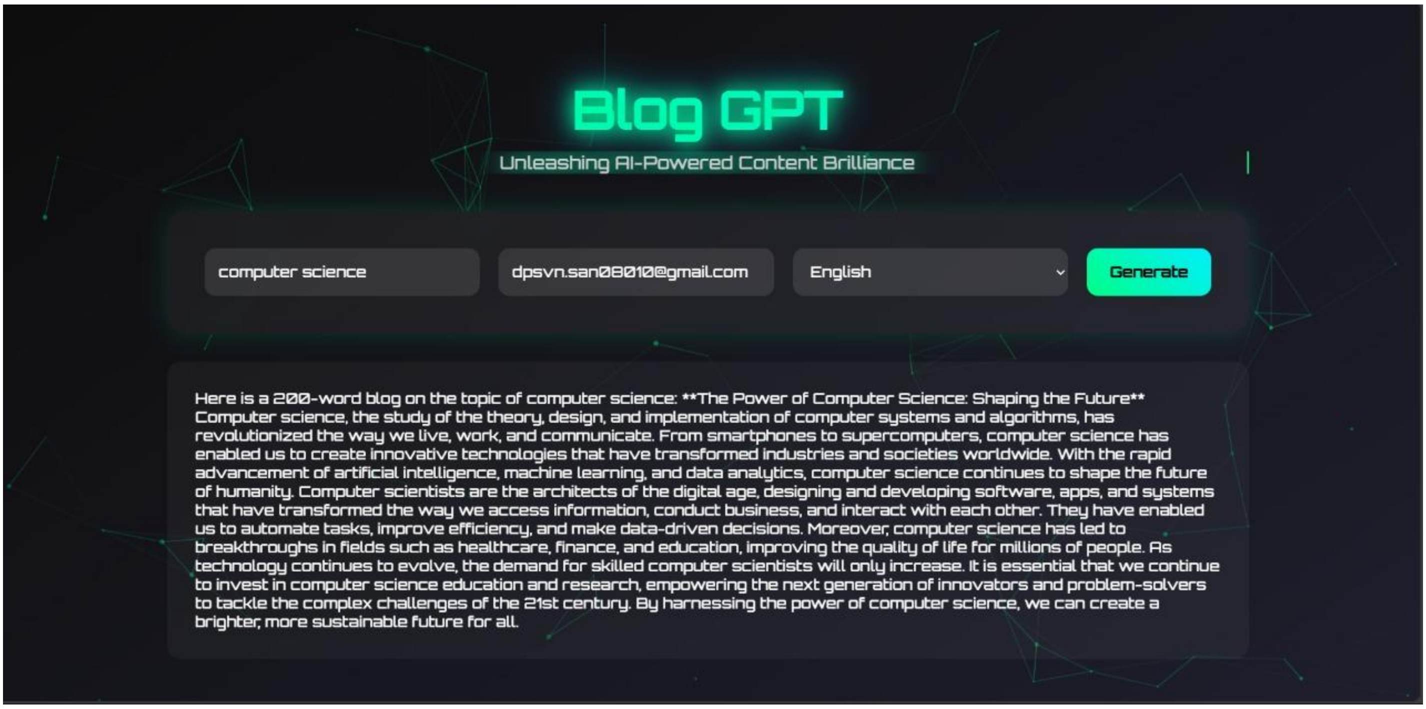
blog-output/

Objects (84)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
□	074559.txt	txt	April 30, 2025, 13:16:01 (UTC+05:30)	2.3 KB	Standard
□	093847.txt	txt	May 1, 2025, 15:08:49 (UTC+05:30)	1.6 KB	Standard
□	093855.txt	txt	April 28, 2025, 15:08:57 (UTC+05:30)	1.8 KB	Standard
□	094057.txt	txt	May 1, 2025, 15:10:58 (UTC+05:30)	1.3 KB	Standard
□	094134.txt	txt	May 1, 2025, 15:11:36 (UTC+05:30)	2.4 KB	Standard
□	095512.txt	txt	April 28, 2025, 15:25:14 (UTC+05:30)	2.1 KB	Standard
□	095734.txt	txt	April 28, 2025, 15:27:35 (UTC+05:30)	2.6 KB	Standard
□	100552.txt	txt	April 28, 2025, 15:35:54 (UTC+05:30)	2.6 KB	Standard
□	102834.txt	txt	May 1, 2025, 15:58:36 (UTC+05:30)	2.6 KB	Standard
□	114820.txt	txt	May 7, 2025, 17:18:22 (UTC+05:30)	1.9 KB	Standard
□	161901.txt	txt	April 28, 2025, 21:49:04 (UTC+05:30)	2.3 KB	Standard
□	181543.txt	txt	April 28, 2025, 23:45:45 (UTC+05:30)	2.3 KB	Standard
□	181622.txt	txt	April 28, 2025, 23:46:23 (UTC+05:30)	2.1 KB	Standard
□	181658.txt	txt	April 28, 2025, 23:47:00 (UTC+05:30)	2.3 KB	Standard
□	181736.txt	txt	April 28, 2025, 23:47:38 (UTC+05:30)	1.9 KB	Standard
□	181816.txt	txt	April 28, 2025, 23:48:17 (UTC+05:30)	2.0 KB	Standard
□	181935.txt	txt	April 28, 2025, 23:49:37 (UTC+05:30)	2.3 KB	Standard

7. The URL and content are returned to the frontend and displayed to the user.

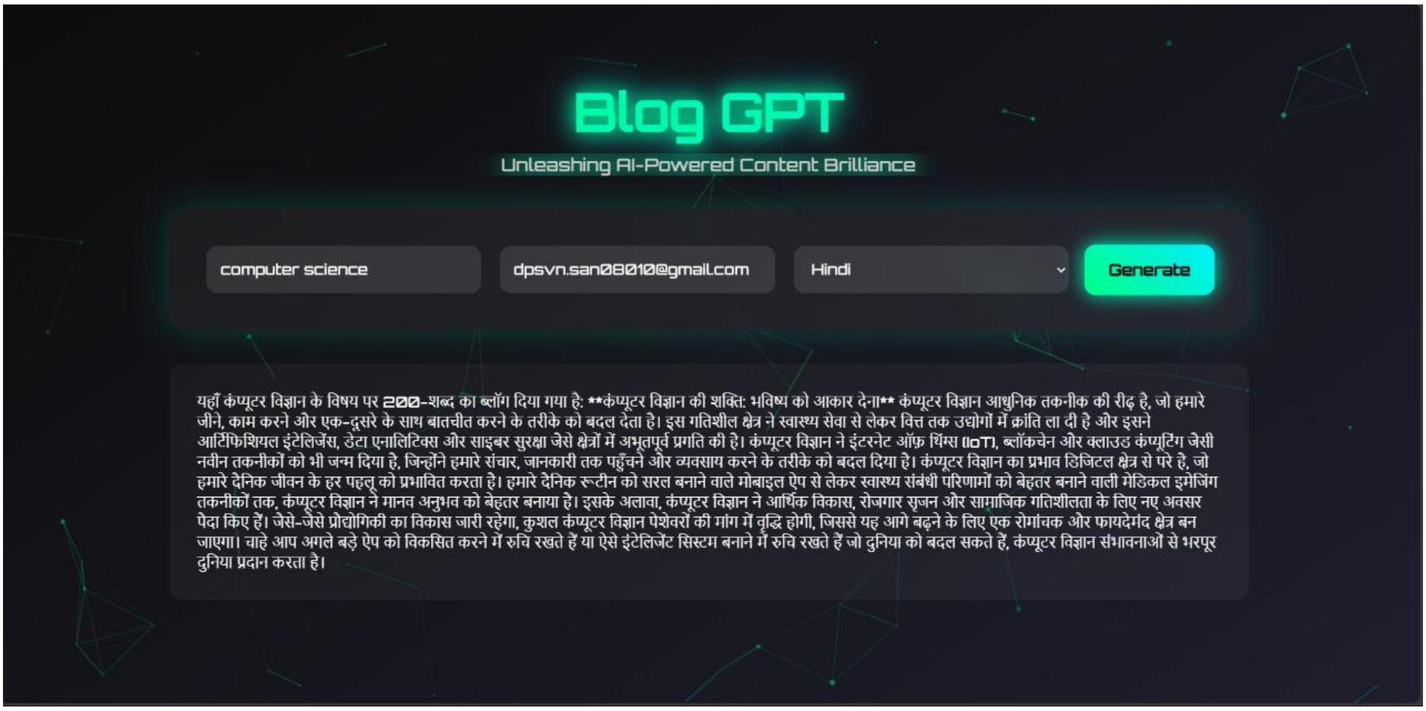


computer science

dpsvn.san08010@gmail.com

English

Here is a 200-word blog on the topic of computer science: **The Power of Computer Science: Shaping the Future** Computer science, the study of the theory, design, and implementation of computer systems and algorithms, has revolutionized the way we live, work, and communicate. From smartphones to supercomputers, computer science has enabled us to create innovative technologies that have transformed industries and societies worldwide. With the rapid advancement of artificial intelligence, machine learning, and data analytics, computer science continues to shape the future of humanity. Computer scientists are the architects of the digital age, designing and developing software, apps, and systems that have transformed the way we access information, conduct business, and interact with each other. They have enabled us to automate tasks, improve efficiency, and make data-driven decisions. Moreover, computer science has led to breakthroughs in fields such as healthcare, finance, and education, improving the quality of life for millions of people. As technology continues to evolve, the demand for skilled computer scientists will only increase. It is essential that we continue to invest in computer science education and research, empowering the next generation of innovators and problem-solvers to tackle the complex challenges of the 21st century. By harnessing the power of computer science, we can create a brighter, more sustainable future for all.



computer science

dpsvn.san08010@gmail.com

Hindi

यहाँ कंप्यूटर विज्ञान के विषय पर 200-शब्द का ब्लॉग दिया गया है: **कंप्यूटर विज्ञान की शक्ति: भविष्य को आकार देना** कंप्यूटर विज्ञान आधुनिक तकनीक की रीढ़ है, जो हमारे जीने, काम करने और एक-दूसरे के साथ बातचीत करने के तरीके को बदल देता है। इस गतिशील क्षेत्र ने स्वास्थ्य सेवा से लेकर वित्त तक उद्योगों में क्रांति ला दी है और इसने आर्टिफिशियल इंटेलिजेंस, डेटा एनालिटिक्स और साइबर सुरक्षा जैसे क्षेत्रों में अभूतपूर्व प्रगति की है। कंप्यूटर विज्ञान ने इंटरनेट ऑफ़ थिंग्स (IoT), ब्लॉकचेन और क्लाउड कंप्यूटिंग जैसी नवीन तकनीकों को भी जन्म दिया है, जिन्होंने हमारे संचार, जानकारी तक पहुँचने और व्यवसाय करने के तरीके को बदल दिया है। कंप्यूटर विज्ञान का प्रभाव डिजिटल क्षेत्र से परे है, जो हमारे दैनिक जीवन के हर पहलू को प्रभावित करता है। हमारे दैनिक रूटीन को सरल बनाने वाले मोबाइल ऐप से लेकर स्वास्थ्य संबंधी परिणामों को बेहतर बनाने वाली मेडिकल इमेजिंग तकनीकों तक, कंप्यूटर विज्ञान ने मानव अनुभव को बेहतर बनाया है। इसके अलावा, कंप्यूटर विज्ञान ने आर्थिक विकास, रोजगार सृजन और सामाजिक गतिशीलता के लिए नए अवसर पैदा किए हैं। जैसे-जैसे प्रौद्योगिकी का विकास जारी रहेगा, कुशल कंप्यूटर विज्ञान पेशेवरों की मांग में वृद्धि होगी, जिससे यह आगे बढ़ने के लिए एक रोमांचक और फायदेमंद क्षेत्र बन जाएगा। चाहे आप अगले बड़े ऐप को विकसित करने में रुचि रखते हैं या ऐसे इंटेलिजेंट सिस्टम बनाने में रुचि रखते हैं जो दुनिया को बदल सकते हैं, कंप्यूटर विज्ञान संभावनाओं से भरपूर दुनिया प्रदान करता है।

4. Technology Stack

4.1 Frontend

- **HTML/CSS/JavaScript or React:** Provides an interactive user interface for entering prompts and viewing blog results.
- **Tailwind CSS (optional):** Can be used to improve styling efficiency and responsiveness.

4.2 Backend and APIs

- **AWS Lambda (Python/Node.js):** Stateless compute service for processing blog generation and translation.
- **Amazon API Gateway:** Secures and manages API calls from the frontend to backend.

4.3 AI and Translation Services

- **Amazon Bedrock:** Provides access to leading foundational models for text generation.
- **AWS Translate:** Contextually translates generated blogs into selected languages with regional relevance.

4.4 Storage and Delivery

- **Amazon S3:** Durable, scalable storage solution. Supports versioning, organized folder structure, and secure access via pre-signed URLs.

4.5 Security and Access Management

- **IAM Roles and Policies:** Enforce least-privilege access for Lambda and S3.
- **S3 Bucket Policies:** Prevent unauthorized access.
- **API Gateway Usage Plans:** Enforce quotas and throttling to prevent abuse.

5. Core Functionalities

5.1 Prompt Submission and Language Selection

Users input a blog topic and optionally choose a language from a list of supported languages. This flexibility allows both monolingual and multilingual content generation.

5.2 AI-Powered Blog Generation

Amazon Bedrock is invoked with a carefully designed prompt structure to ensure the generated blog has an introduction, body, and conclusion. The LLM ensures content is natural, engaging, and coherent.

5.3 Real-Time Translation

If a language is selected, AWS Translate converts the blog to that language. It uses neural machine translation for high fidelity and understands sentence-level semantics.

5.4 Cloud Storage and Access

Both original and translated versions are stored in S3. The Lambda function returns a secure, pre-signed URL to the frontend so users can view or download their blogs.

6. Security and Authentication

6.1 IAM and Role-Based Access

Lambda uses IAM roles with minimal permissions (access only to S3 and Bedrock/Translate APIs). Policies are crafted to avoid over-permissioning.

6.2 API Gateway Protection

API Gateway enforces HTTPS communication, validates request schemas, and restricts access through throttling and quotas.

6.3 S3 Secure Storage

S3 buckets are configured to be private. Pre-signed URLs ensure secure, time-bound access to blog content without exposing underlying storage.

7. Future Enhancements

1. **Frontend Hosting on S3 + CloudFront:** Deploy frontend for global access.
2. **Authentication via Cognito:** Add user accounts and blog history.
3. **Advanced Prompt Customization:** Let users set tone, target word count, and structure.
4. **Editor and Preview Panel:** Allow users to edit content before final download.
5. **Multiformat Export:** Provide downloadable options like PDF, DOCX, or HTML.
6. **Mobile Support:** Responsive UI for mobile and tablet users.

8. Conclusion

1. **Final Thoughts**
   Blog GPT streamlines the blog creation process, bringing together AI and cloud services to offer a powerful yet user-friendly tool. Its scalability and multilingual support make it valuable for individuals and organizations alike.
2. **Learnings and Outcomes**
   Hands-on experience with serverless AWS architecture
   Deepened understanding of Amazon Bedrock and AWS Translate
   Designed and implemented a scalable, modular AI application
   Prepared the system for production deployment and future growth