



# COMMUNITY DAY

RAJASTHAN 2023

## Dive Deep into the World of Generative AI with Amazon Bedrock



**Abhishek Gupta**

Founding Engineer (**Omnistrat**)  
Early-stage SaaS Startup  
Ex-Walmart, Ex-LinkedIn



# ChatGPT



LE

how can I start the app in debug log level?



To start your Streamlit app with the debug log level, you can modify the logging configuration by setting the log level to `logging.DEBUG`. Here's an example:

python

```
import logging
import streamlit as st

# Configure logging
logging.basicConfig(
    level=logging.DEBUG # Set the log level to DEBUG
)
```

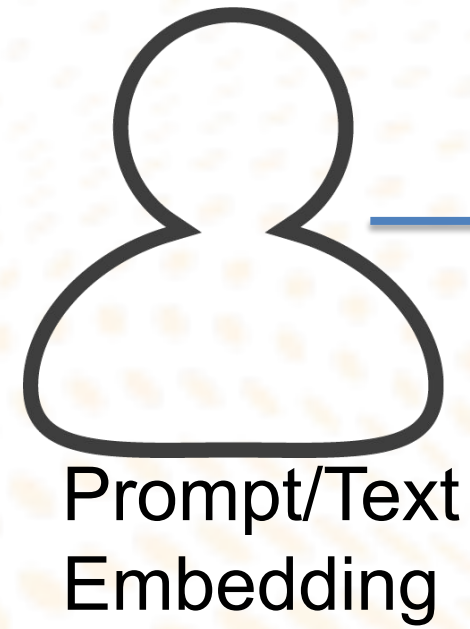
## OpenAI API

 GPT-3.5

 GPT-4

## Application

## LLM



Data

Amazon Bedrock

API Layer



Amazon  
Bedrock Models





**Text generation**



**Chatbot**



**Summarization**



**Question  
answering**



**Image  
generation**



**Code  
generation**

# Code Generation

WITH SIMPLE PROMPT



Prompt Input  
Request

Amazon Bedrock  
Foundation Model

Generated  
Output



# Demo

## Amazon Bedrock



## Getting started

Overview

Examples

## Foundation models

Base models

Custom models

Providers

## Playgrounds

Chat

**Text**

Image

## Deployment

Provisioned Throughput

Model access

Settings

User guide

Bedrock Service Terms

[Amazon Bedrock](#) > Text playground

## Text playground

[Info](#)☒ Streaming

Anthropic

Claude V2  
v2Using on-demand capacity. [Switch Provisioned Throughput](#)

Human: Generate a tweet welcoming audience of AWS User Group Rajasthan in the event.

Assistant:

Here is a draft 127 character tweet welcoming the audience at the AWS User Group Rajasthan event:

A warm welcome to all the attendees at #AWSUGRajasthan! We're thrilled you could join us for this exciting event. Looking forward to connecting with the Rajasthan cloud computing community and learning together. Let's have a great day ahead! #AWS #cloud



## Inference configuration

[Reset](#)▼ Randomness and diversity [Info](#)

Temperature

1



Top P

0.999



Top K

250

▼ Length [Info](#)

Maximum length

300



Stop sequences

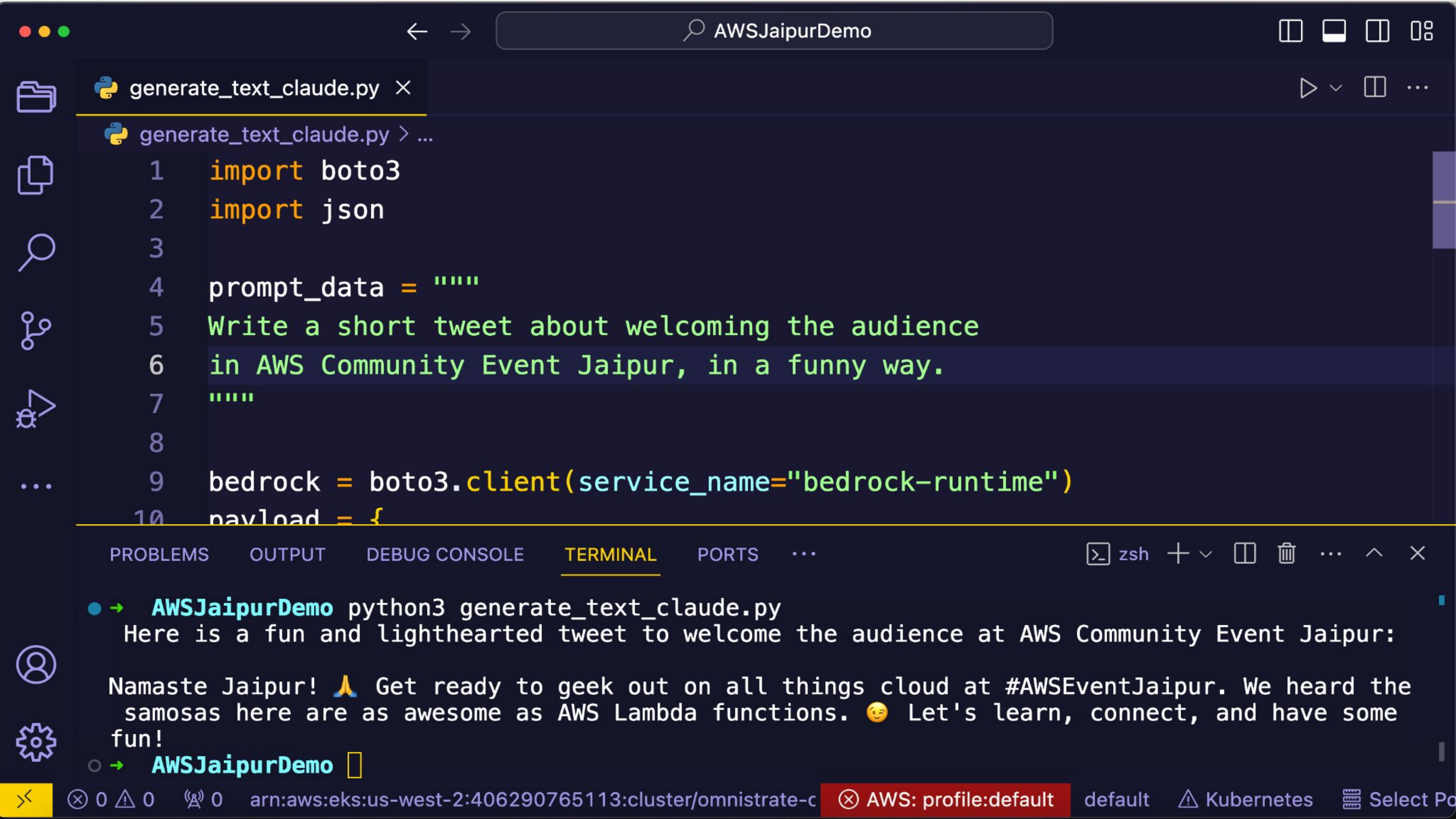
Add

Run

[View API request](#)

Save









▶ ✓ ↺ ↻ □ ...

 generate text i2.py > ...

[M↓ README.md](#)

```
17 body = json.dumps(payload)
```

zsh +  ... ^ x

➔ AWSUGJaipur git:(main) x

main\* 0 0 0 0 arn:aws:eks:us-west-2:406290765113:cluster/omnistrade-cp-eks-cluster-mxw2wpgtfr ✓ AWS: profile:default default ⚠ Kubernetes Select Postgres Server



← → AWSJaipurDemo

generate\_text\_claude.py

generate\_image.py ×


generated-1.png

generated-4.png ×

generate\_image.py > main

```
1 import base64
2 import os
3 import random
4 import boto3
5 import json
6
7
8 prompt_data = """
9 A high-red 4k HDR photo of audience sitting in AWS Community Day Event.
10 Action shot, big auditorium, 200 people smiling. Cinematic film quality.
11 """
12
13
14 def main():
15     for i in range(0, 5):
16         seed = random.randint(0, 100000)
17         generate_image(prompt=prompt_data, seed=seed, index=i)
18
19
20 def generate_image(prompt: str, seed: int, index: int):
21     payload = {
```

output > generated-4.png



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODEWHISPERER REFERENCE LOG

zsh + ×

→ AWSJaipurDemo python3 generate\_image.py

○ → AWSJaipurDemo

○ → AWSJaipurDemo

arn:aws:eks:us-west-2:406290765113:cluster/omnistrade-cp-eks-cluster-mxw2wpgtfi AWS: profile:default default Kubernetes Select Postgres Server ✓ Spell



## Name



generate\_image.py



generate\_text\_claude.py



output



generated-0.png



generated-1.png



generated-2.png



generated-3.png



generated-4.png







Thank You

[abhishek@AceTheCloud.com](mailto:abhishek@AceTheCloud.com)

<https://Blog.AceTheCloud.com>

[@AbhiForTweeting](https://twitter.com/AbhiForTweeting)

<https://linkedin.com/in/enterprise-cloud-architect/>

Code Repo : <https://github.com/AceTheCloud/AWSUGJaipur>

