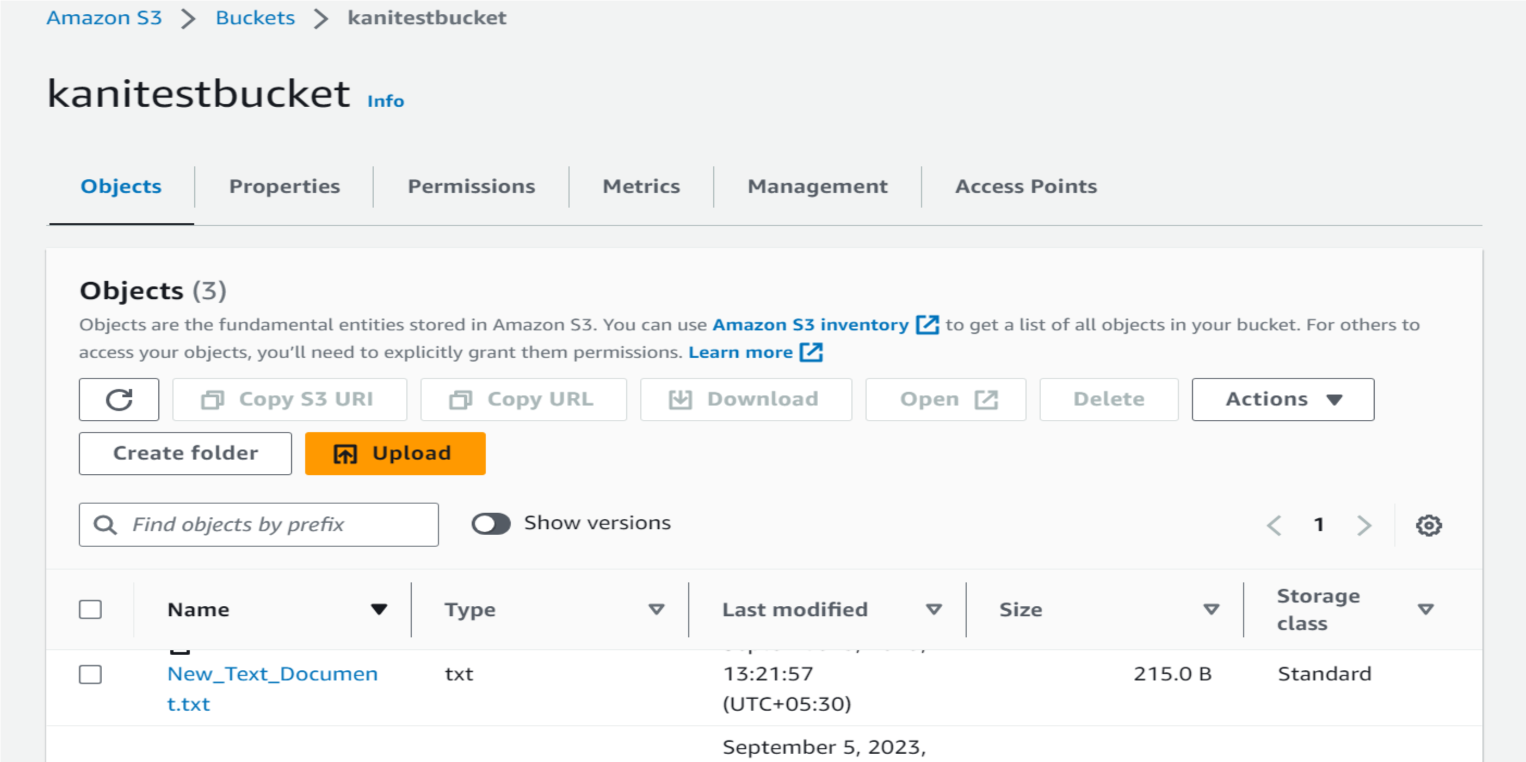
KANISHK MUDGAL PROGRESS REPORT ON S3 , LAMBDA , API GATEWAY , JUPYTER INTEGRATION

1. S3 Bucket and object creation:

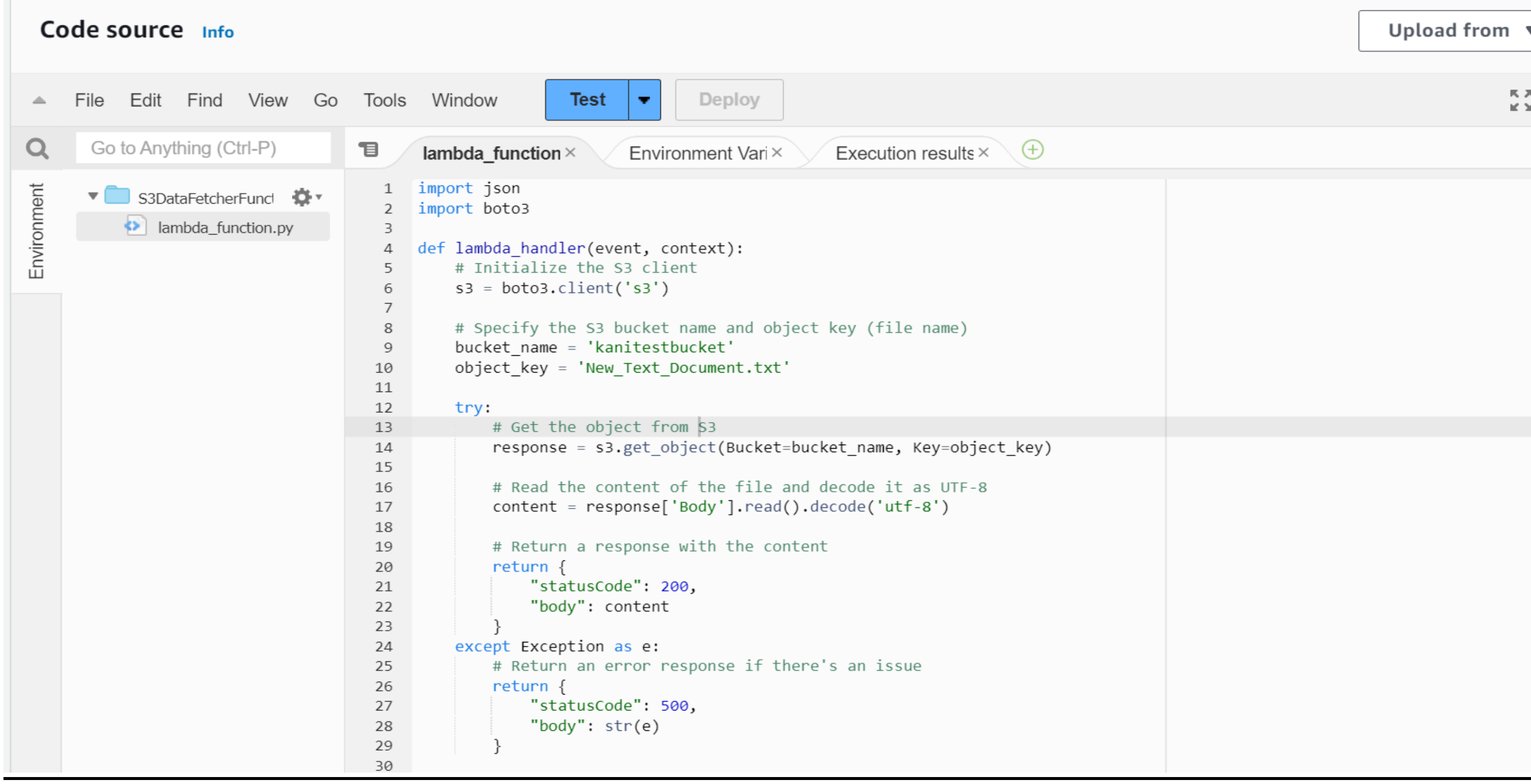


Objects (3)

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**

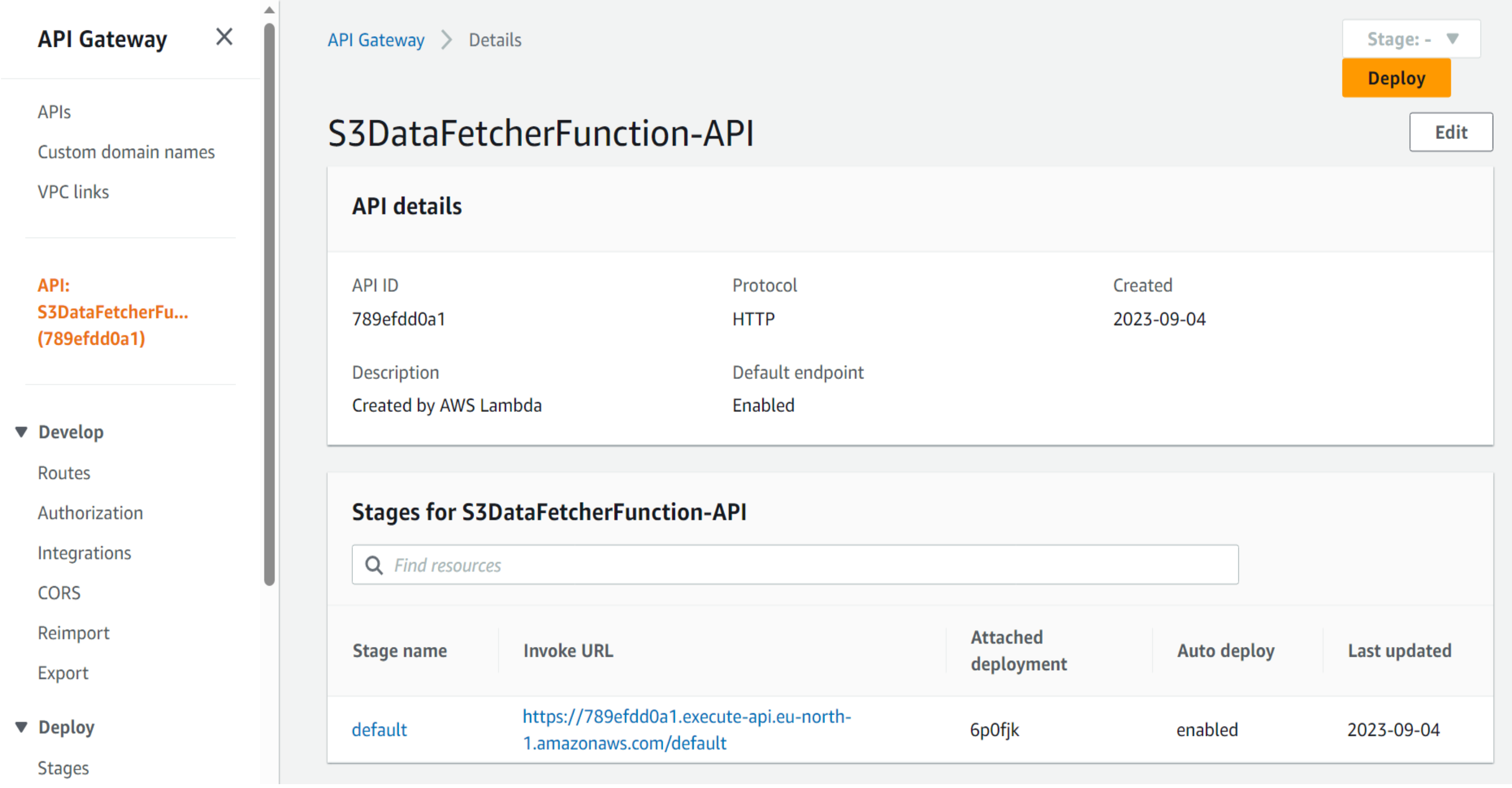
□	Name	Type	Last modified	Size	Storage class
□	New_Text_Document.txt	txt	13:21:57 (UTC+05:30)	215.0 B	Standard
			September 5, 2023,		

2. S3 data fetch Lambda function:



```python
import json
import boto3

def lambda_handler(event, context):
    # Initialize the S3 client
    s3 = boto3.client('s3')

    # Specify the S3 bucket name and object key (file name)
    bucket_name = 'kanitestbucket'
    object_key = 'New_Text_Document.txt'

    try:
        # Get the object from S3
        response = s3.get_object(Bucket=bucket_name, Key=object_key)

        # Read the content of the file and decode it as UTF-8
        content = response['Body'].read().decode('utf-8')

        # Return a response with the content
        return {
            "statusCode": 200,
            "body": content
        }
    except Exception as e:
        # Return an error response if there's an issue
        return {
            "statusCode": 500,
            "body": str(e)
        }

```

3. Added Trigger to lambda function – API Gateway



S3DataFetcherFunction-API

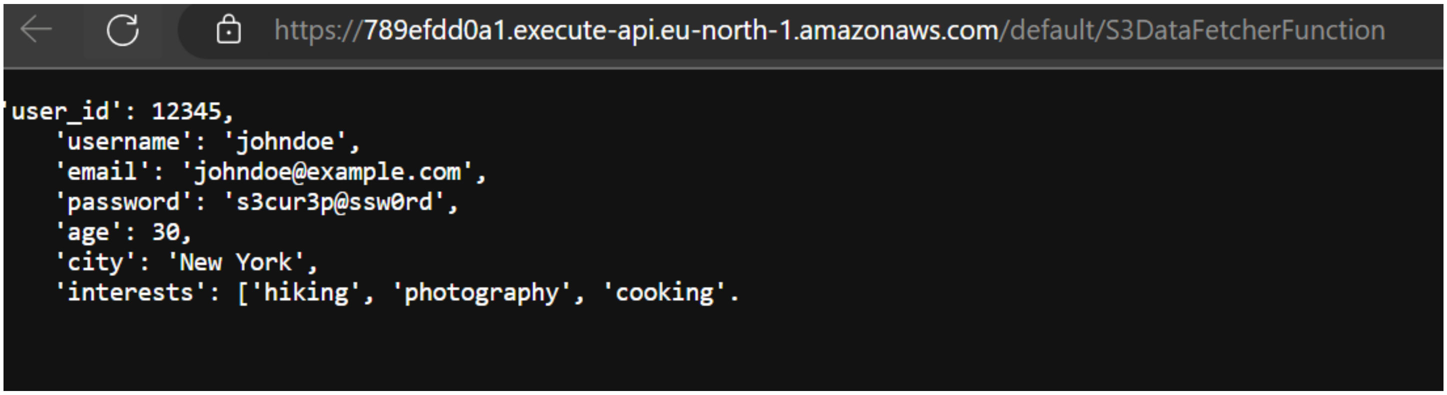
API details

API ID: 789efdd0a1
Protocol: HTTP
Created: 2023-09-04
Description: Created by AWS Lambda
Default endpoint: Enabled

Stages for S3DataFetcherFunction-API

Stage name	Invoke URL	Attached deployment	Auto deploy	Last updated
default	https://789efdd0a1.execute-api.eu-north-1.amazonaws.com/default	6p0fjk	enabled	2023-09-04

4. API direct link response when data feeded through Jupyter in S3 bucket object.


https://789efdd0a1.execute-api.eu-north-1.amazonaws.com/default/S3DataFetcherFunction

```
'user_id': 12345,
    'username': 'johndoe',
    'email': 'johndoe@example.com',
    'password': 's3cur3p@ssw0rd',
    'age': 30,
    'city': 'New York',
    'interests': ['hiking', 'photography', 'cooking'.
```

5. Feeding Data & calling Lamba API model in Jupyter notebook

KANISHK MUDGAL PROGRESS REPORT ON S3 , LAMBDA , API GATEWAY , JUPYTER INTEGRATION

In [58]:
```
import boto3
import requests
import json
import s3fs
service_name='s3',
```

In [ ]:
```
#-----------------------------------DEALING WITH S3 BUCKET-----------------------------------------------#
```

In [136]:
```
# Configure AWS credentials
boto3.setup_default_session(
    aws_access_key_id='AKIA4NYFOBBAQ6EVBDV5',
    aws_secret_access_key='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',
    region_name='eu-north-1'
)
```

In [141]:
```
#----Giving input in bucket----#

# Specifying the S3 bucket name and object key for the text file
s3_bucket_name = 'kanitestbucket'
s3_key = 'New_Text_Document.txt'  # Choose a key name for your text file

# Define input data as a string
input_data = """'user_id': 12345,
    'username': 'johndoe',
    'email': 'johndoe@example.com',
    'password': 's3cur3p@ssw0rd',
    'age': 30,
    'city': 'New York',
    'interests': ['hiking', 'photography', 'cooking'.
"""

# Save the input data to a local text file
with open('input_data.txt', 'w') as file:
    file.write(input_data)

# Upload the local text file to S3
s3.upload_file('input_data.txt', s3_bucket_name, s3_key)

print(f"Input data has been saved as {s3_key} in the S3 bucket: {s3_bucket_name}")
```

```
Input data has been saved as New_Text_Document.txt in the S3 bucket: kanitestbucket
```

In [142]:
```
#----Print contents of txt file------#

# Specify the S3 bucket name and file key
s3_bucket_name = 'your-s3-bucket-name'
file_key = 'example.txt'  # Replace with the actual key of your text file

try:
    # Get the text file from S3
    response = s3.get_object(Bucket='kanitestbucket', Key='New_Text_Document.txt')

    # Read the content of the text file
    text_content = response['Body'].read().decode('utf-8')

    # Print the content
    print("Content of the text file:")
    print(text_content)

except Exception as e:
    print("Error:", str(e))
```

```
Content of the text file:
'user_id': 12345,
    'username': 'johndoe',
    'email': 'johndoe@example.com',
    'password': 's3cur3p@ssw0rd',
    'age': 30,
    'city': 'New York',
    'interests': ['hiking', 'photography', 'cooking'.
```

KANISHK MUDGAL PROGRESS REPORT ON S3 , LAMBDA , API GATEWAY , JUPYTER INTEGRATION

In [127]:
```
#----------------------------------------------------------------------------------------------------------------------------
```

In [128]:
```
#------------------------------DEALING WITH LAMBDA function through API HERE---------------------#
```

In [149]:
```
# Set up the S3 and API Gateway clients
s3 = boto3.client('s3')
api_url = 'https://789efdd0a1.execute-api.eu-north-1.amazonaws.com/default/S3DataFetcherFunction'
```

In [151]:
```
import requests

# API Gateway URL
api_url = 'https://789efdd0a1.execute-api.eu-north-1.amazonaws.com/default/S3DataFetcherFunction'

try:
    response = requests.get(api_url)

    if response.status_code == 200:
        # Parse and print the response from the API Gateway
        api_response = response.text
        print("API Response:")
        print(api_response)
    else:
        print("Error:", response.status_code)
        print("Error Message:", response.text)

except Exception as e:
    print("Error:", str(e))
```

```
API Response:
'user_id': 12345,
    'username': 'johndoe',
    'email': 'johndoe@example.com',
    'password': 's3cur3p@ssw0rd',
    'age': 30,
    'city': 'New York',
    'interests': ['hiking', 'photography', 'cooking'.
```

In [ ]:
```
#--------------------------------------------------END-----------------------------------------------------------------------
```

----------------------------END--