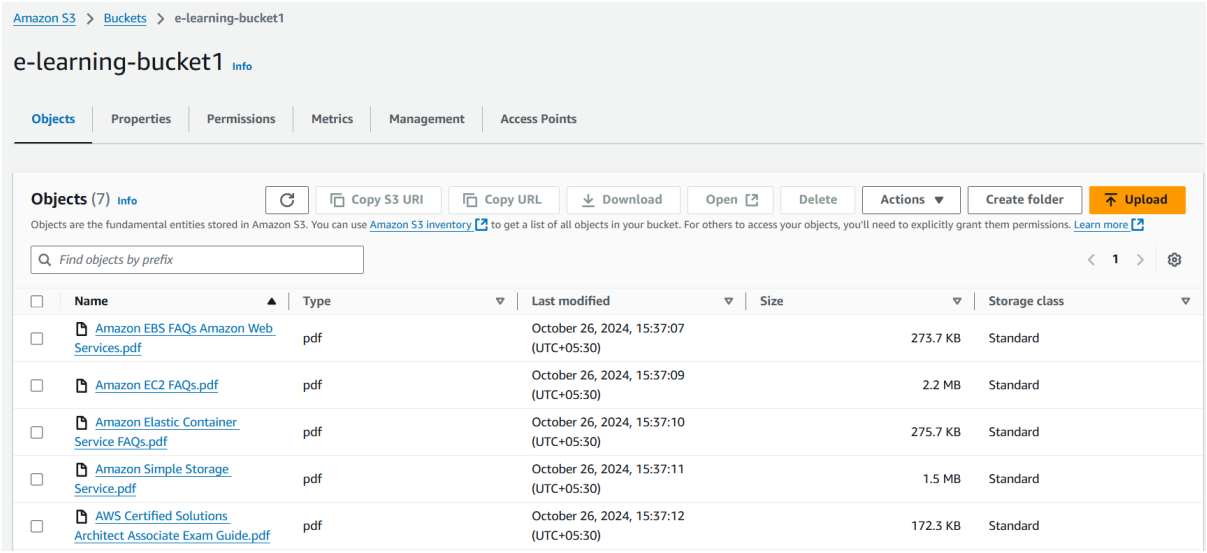
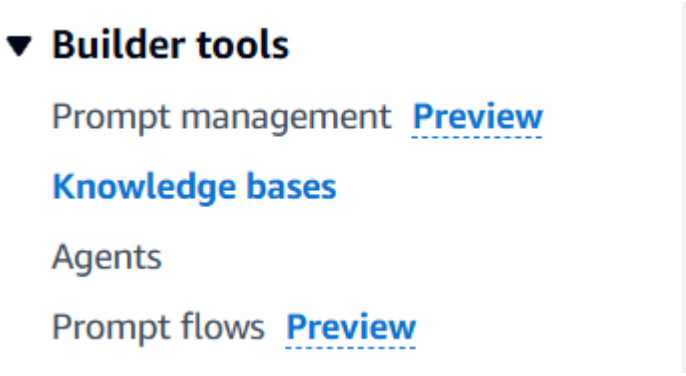
😊 Serverless E-Learning Application

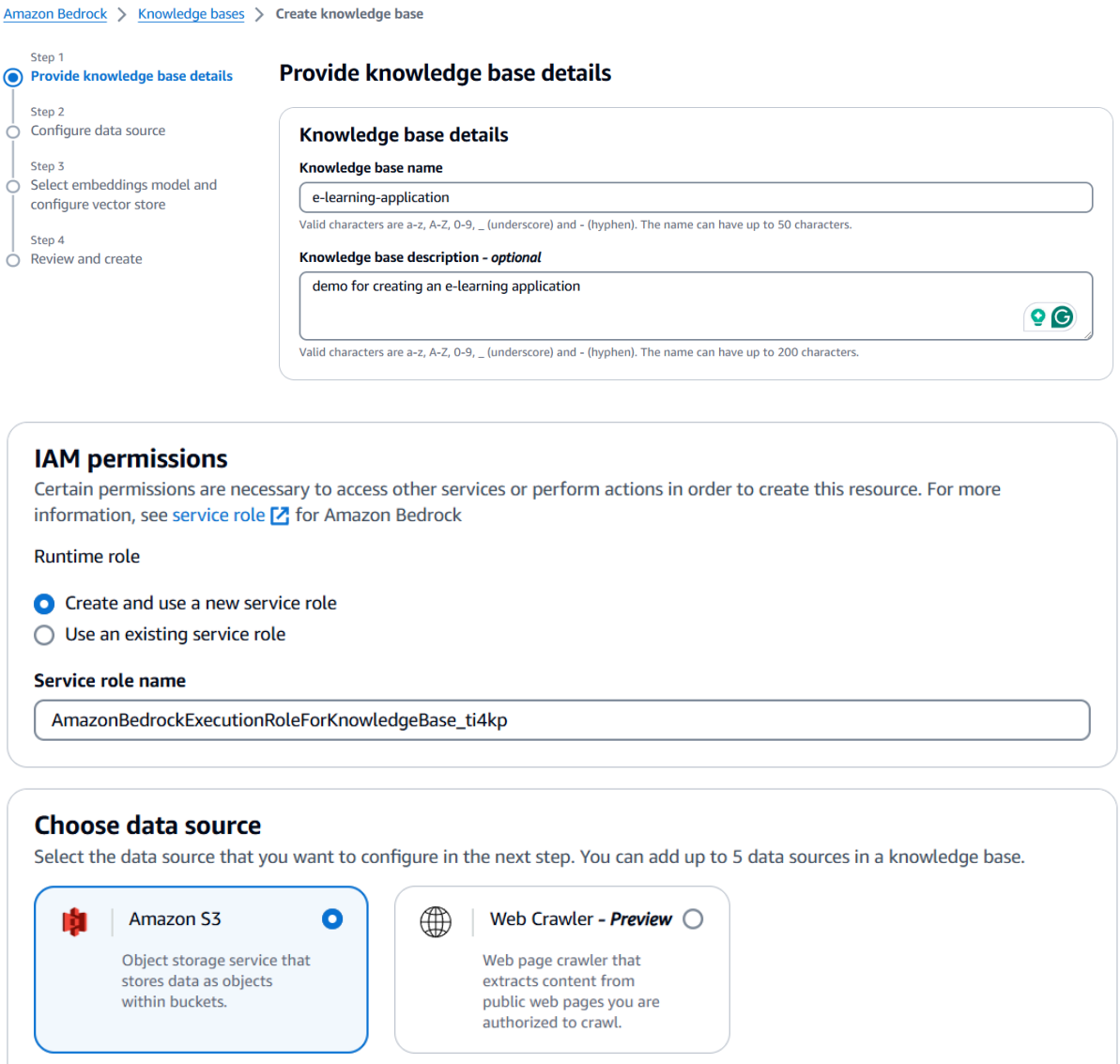
1. In this lab we are going to make a Serverless E-learning application using AWS Lambda, API gateway, and Bedrock foundational models. We will also S3 bucket where we will upload all the necessary files.
2. Below is some important information.
    - **Amazon Bedrock Knowledge Base** supports building **generative AI applications** using **retrieval-augmented generation (RAG)** by automating data ingestion workflows.
    - In an e-learning solution, content like **PDFs** stored in an **S3 bucket** can be broken down into smaller **chunks** (e.g., 300 characters) for vector embeddings.
    - **Embedding models** such as **Cohere** or **Titan** can create embeddings from the PDF chunks, which are stored in a **vector database** like **Amazon OpenSearch** or **Pinecone**.
    - Amazon Bedrock Knowledge Base simplifies the process with **just a few clicks**, automating steps like data loading, chunking, embedding creation, and vector storage.
    - Currently, it supports **S3** as the data source, chunking strategies, Cohere and Titan embedding models, and vector databases like **Amazon OpenSearch**, **Aurora**, **Pinecone**, and **Redis**.
3. So, there is a prerequisite for this lab and that is you need to create an IAM user with administrator privileges and then provide this user the Console access and then you need to log in with this user to work with this lab.
4. Now the first thing we need to do is create an S3 bucket and upload all the relevant files to it.



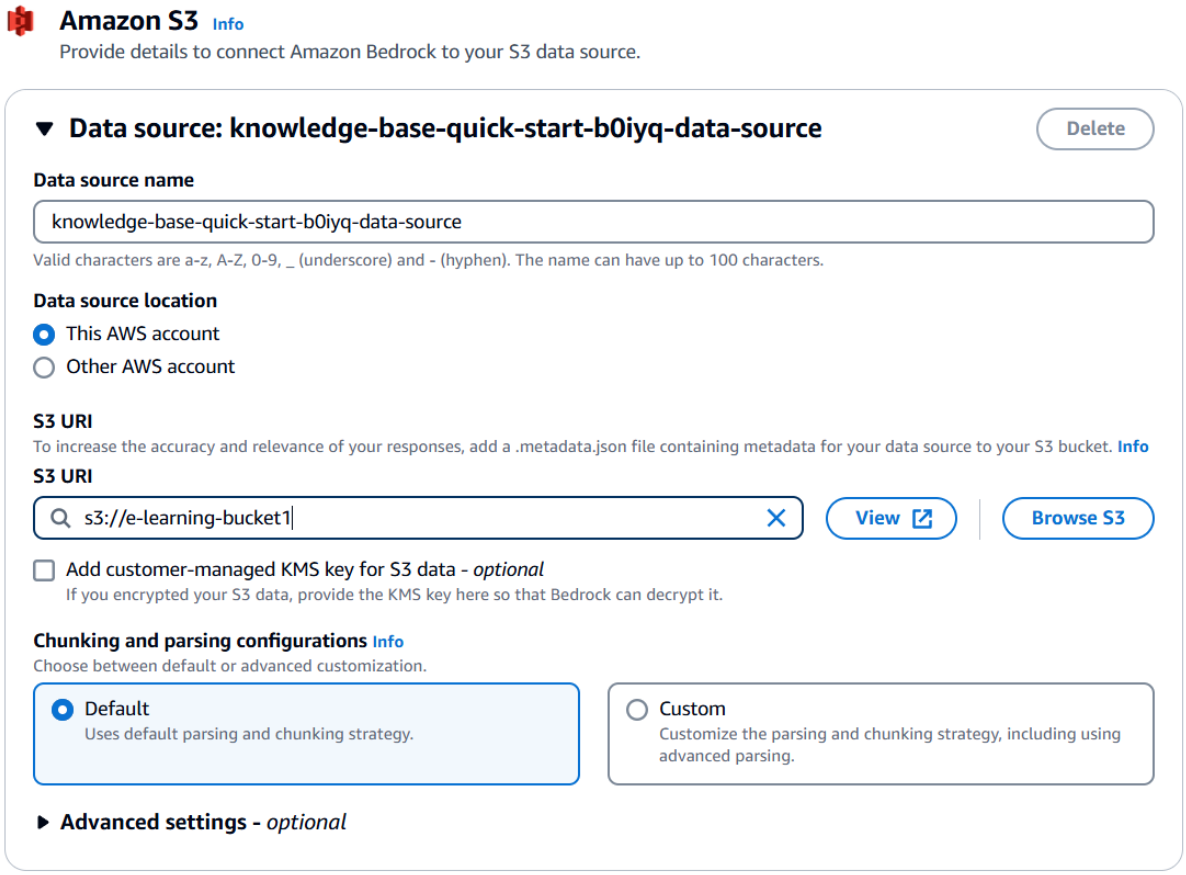
5. Then search for Amazon Bedrock and navigate to it, from the left pane expand builder tools choose knowledge bases, and click on Create knowledge base. **Also keep in mind that using a knowledge base will cost you $0.50 cents per hour**.



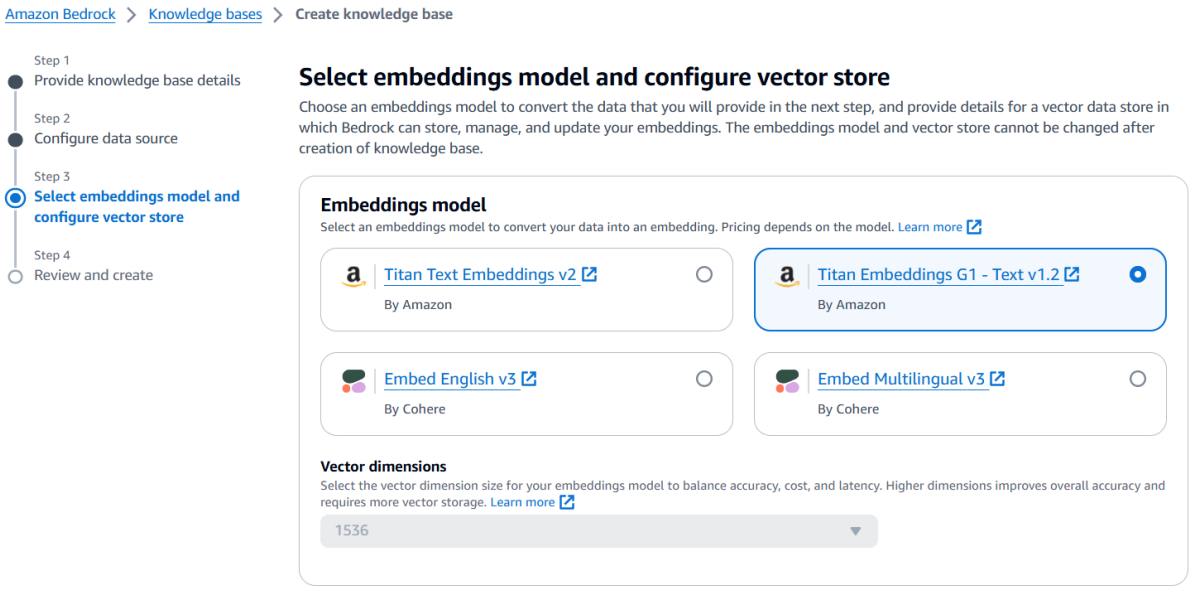
6. In step 1, you need to give it a name then scroll down, then for IAM choose Create New Role and for the data source choose S3. Click on next.

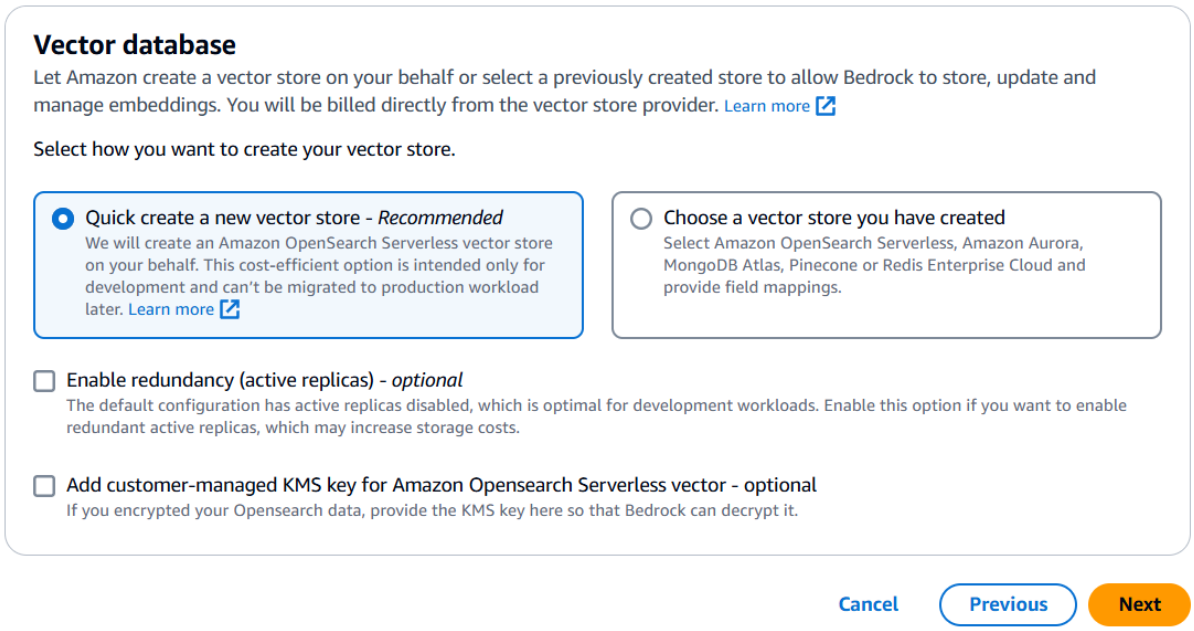


7. After that you need to choose your bucket and click on next.

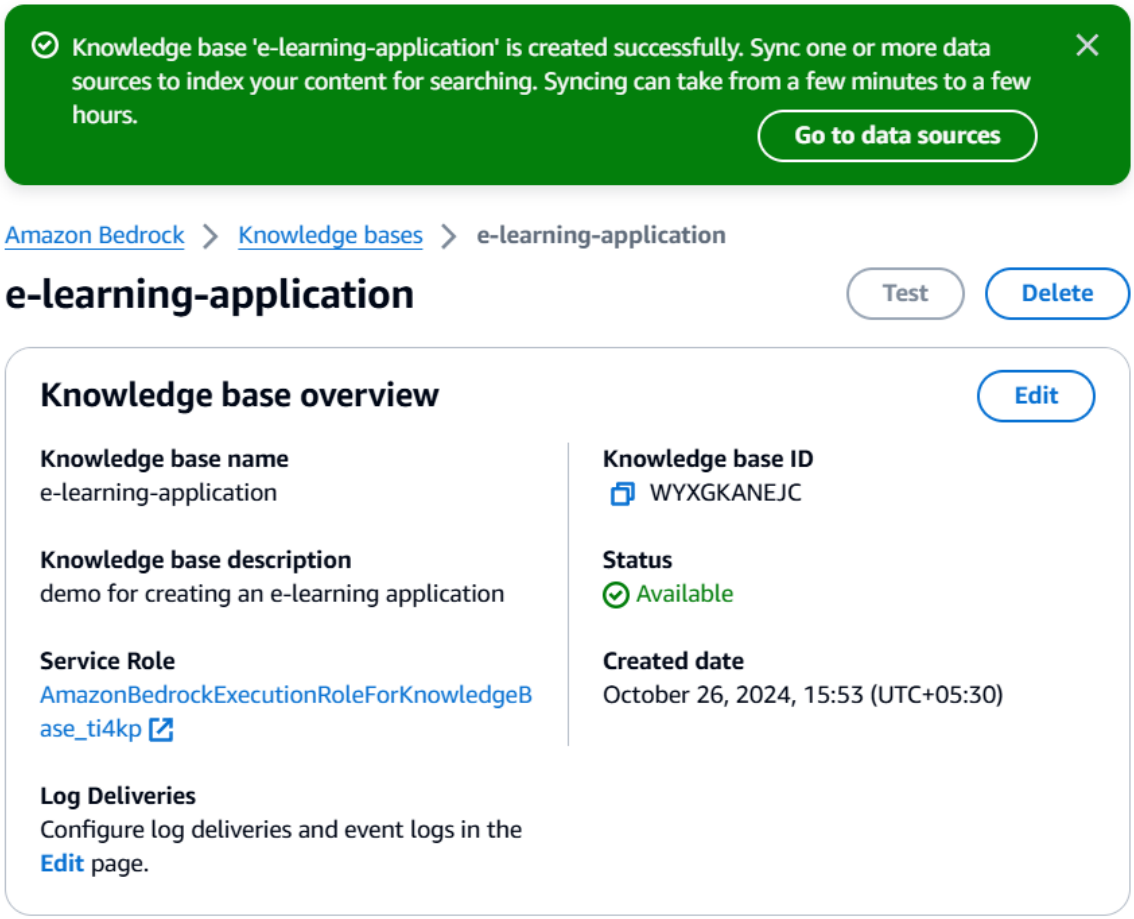


8. In step 3, choose Titan embeddings G1 as your embedding model, and for vector base choose Quick create and click on next.
9. Then from the review page you need to click on create knowledge base.

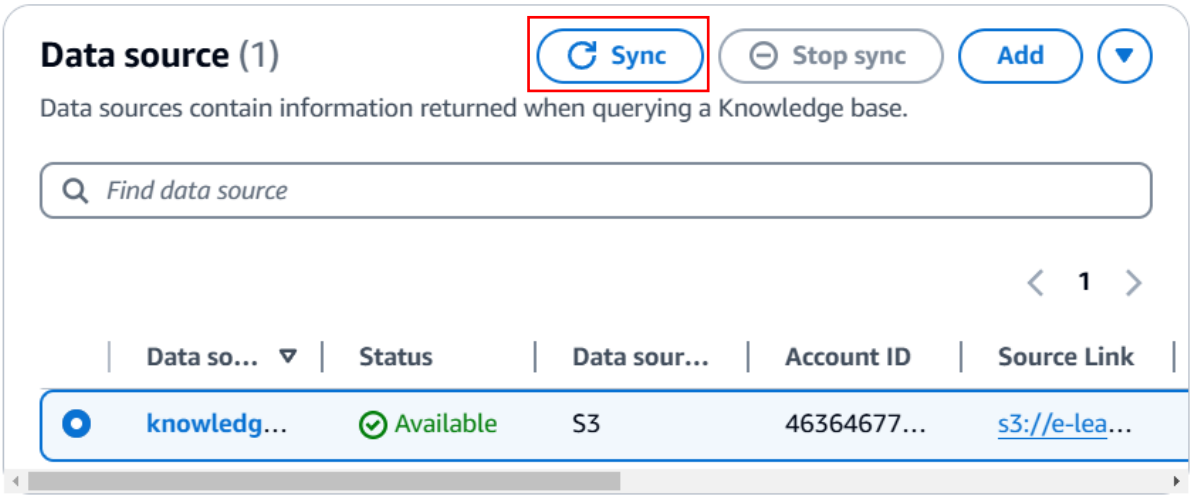




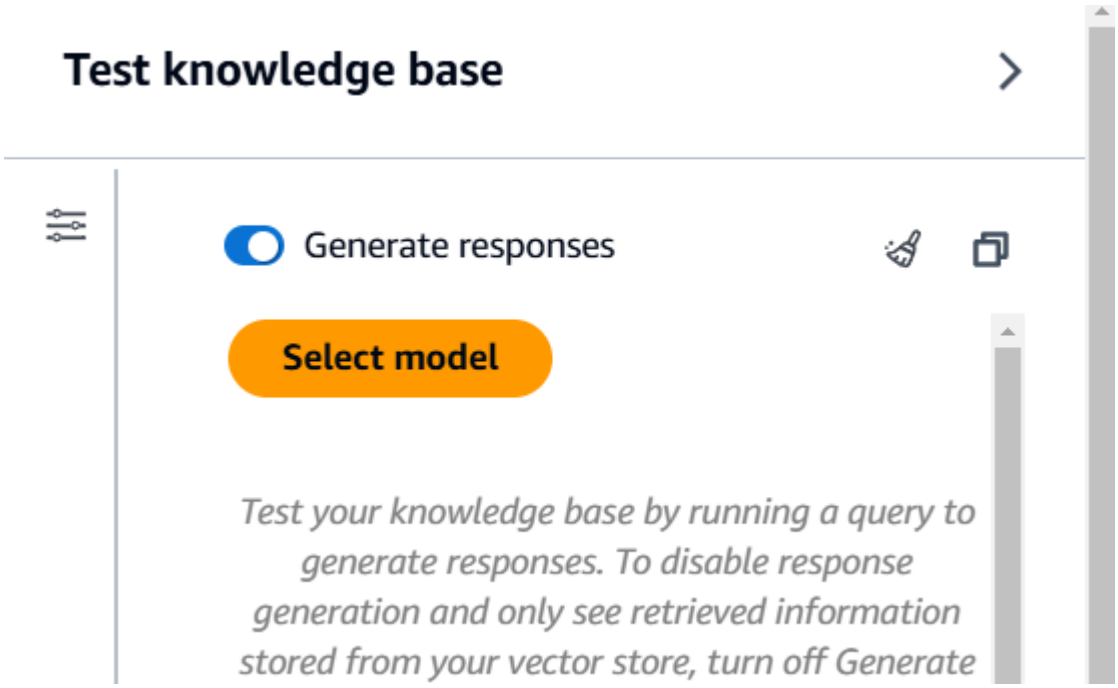
10. The creation of a knowledge base will take 4-5 minutes may be more. Below you can see that our knowledge base has been created.

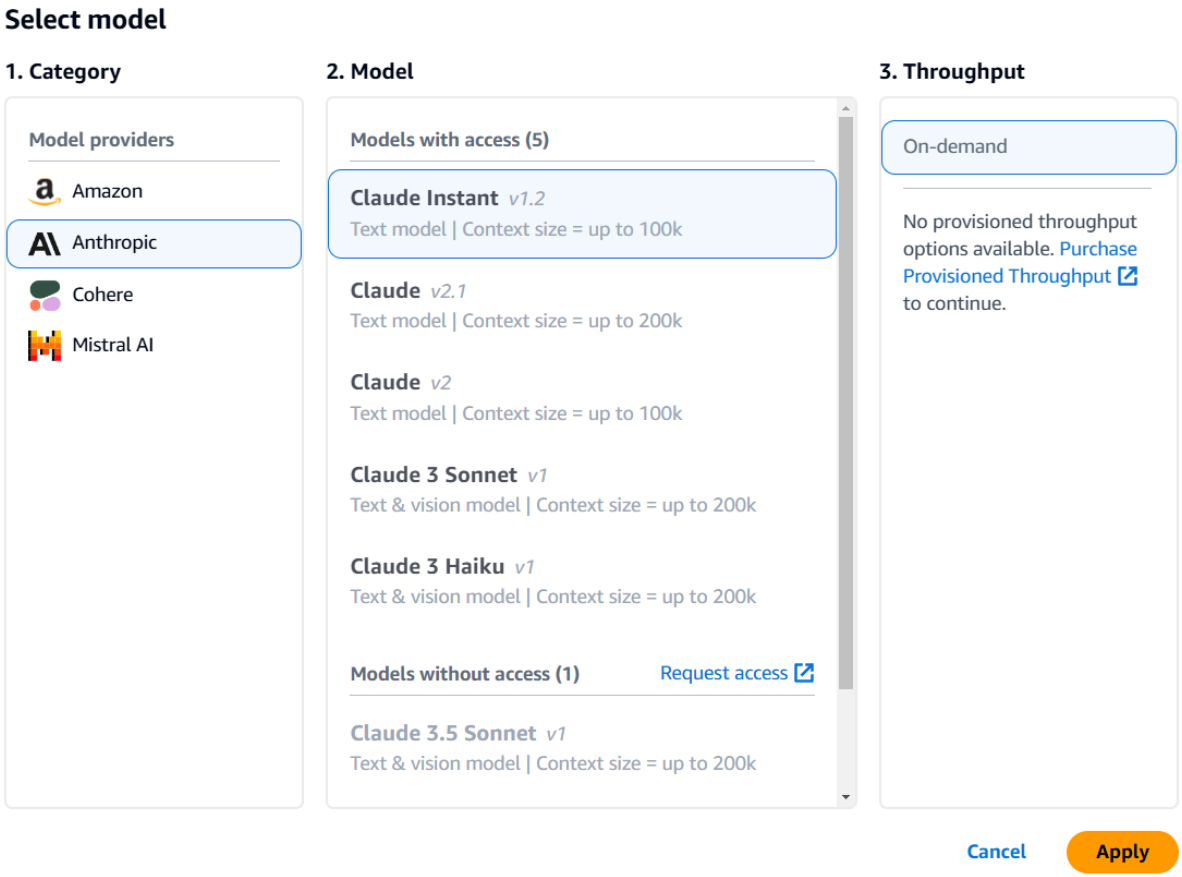


11. Now scroll down to the data source section and select your data source then click on sync.

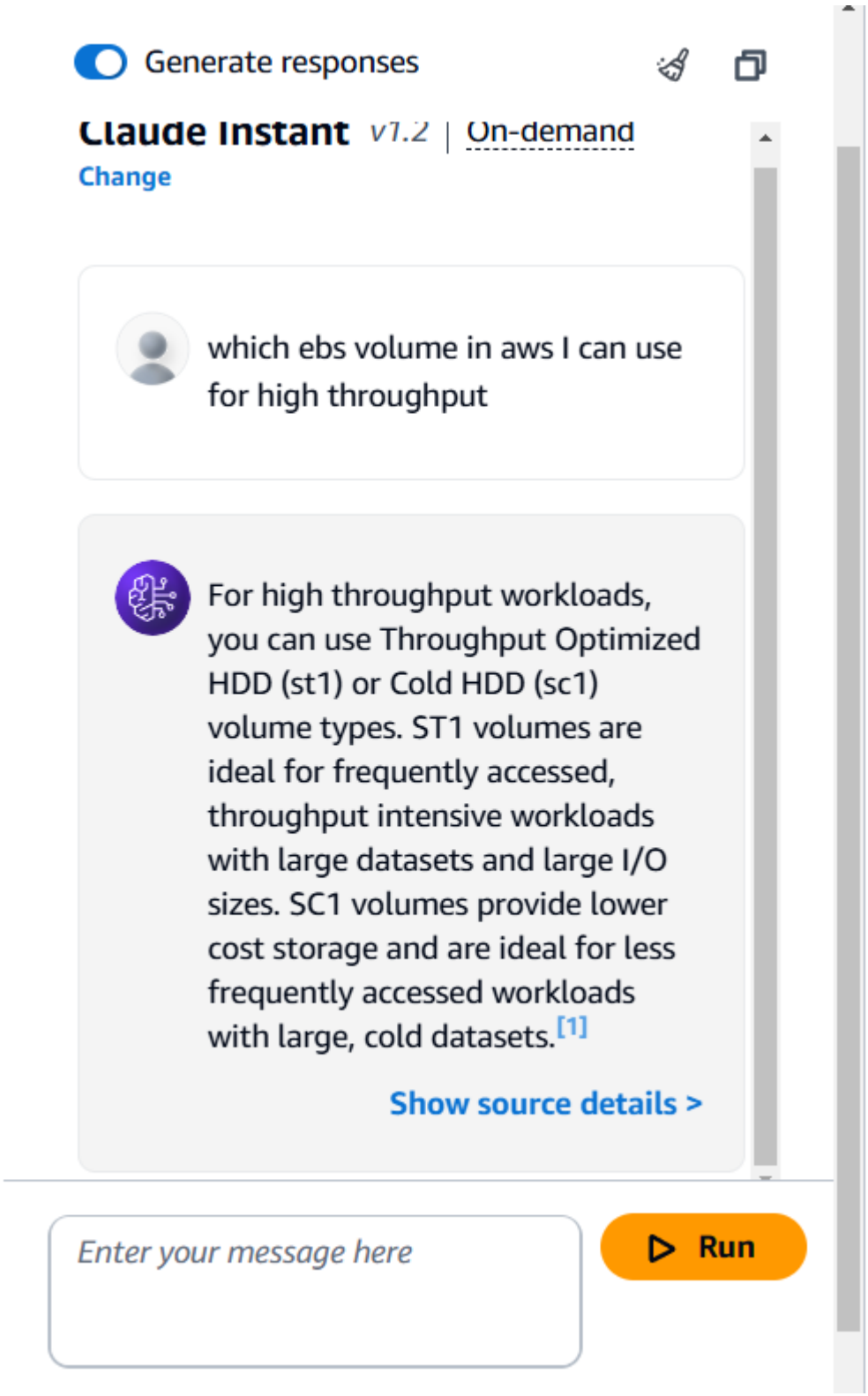


12. Once your data source has been synced then click on select model and choose a model of your choice.

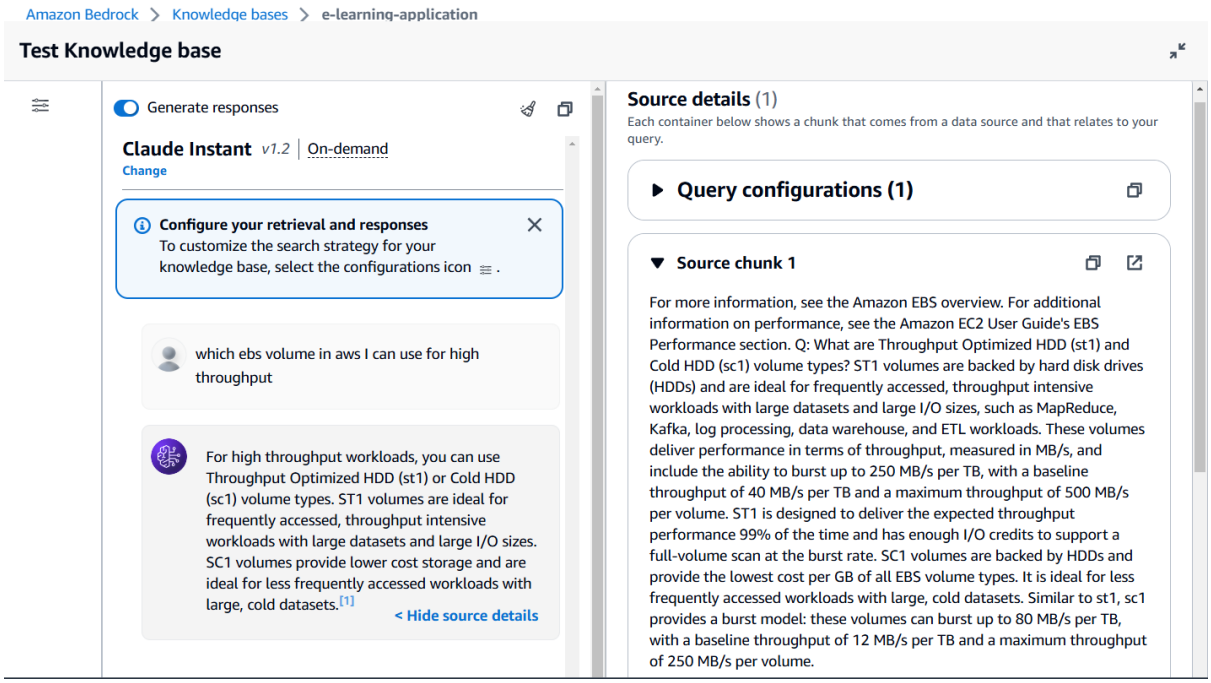




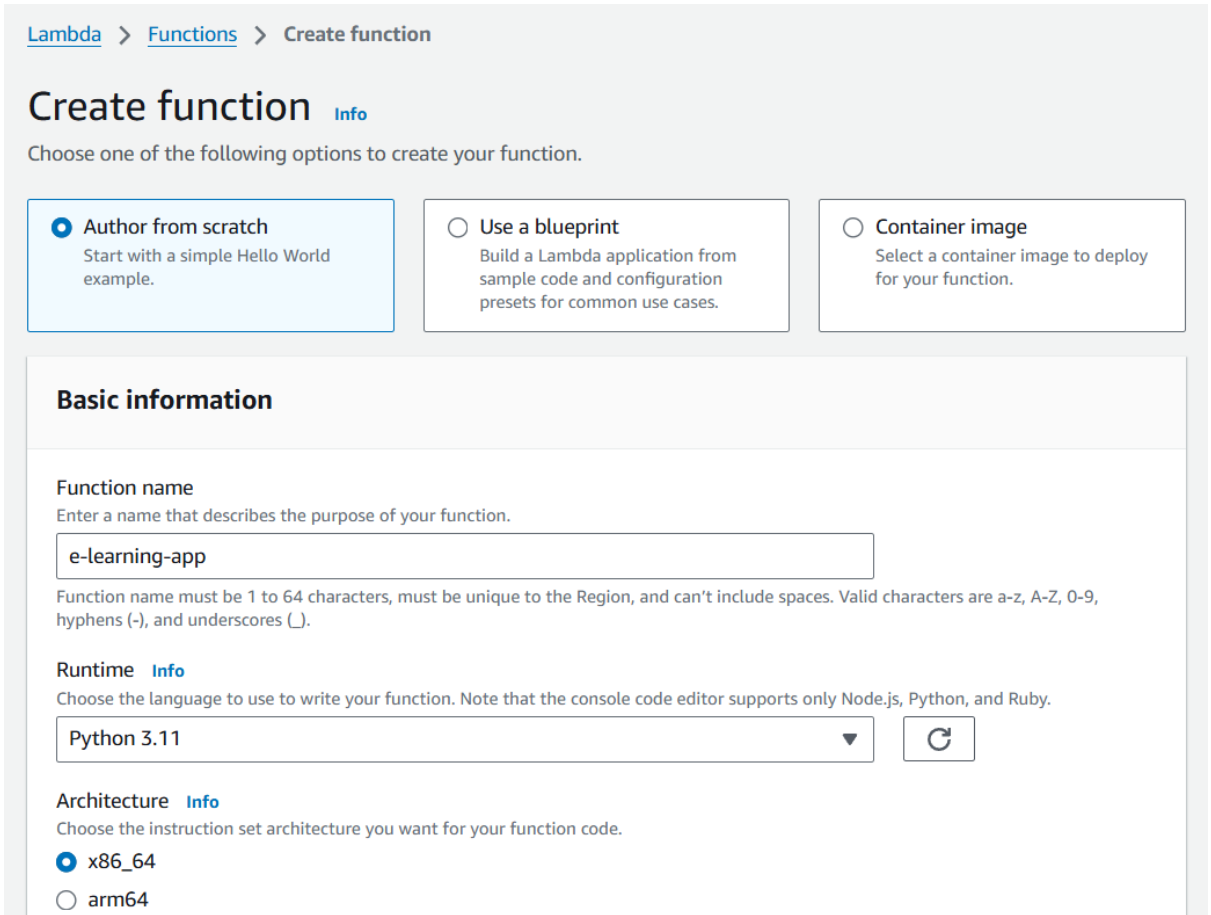
13. Here you can see that we get the answer to our question and we can also see the source of information.



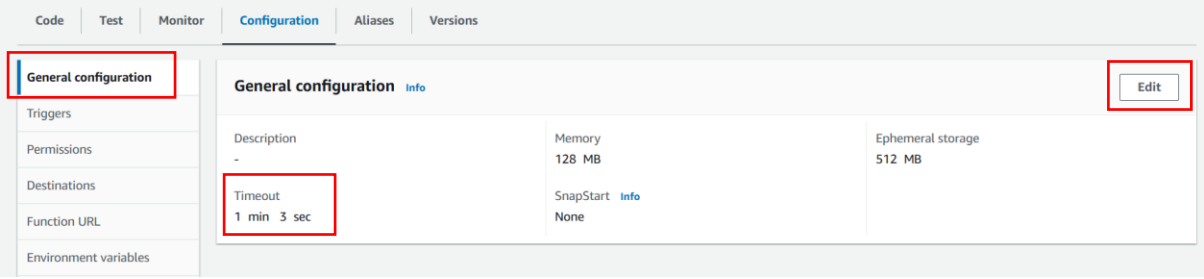
14. If you click on show source details then you can see source chunks too.



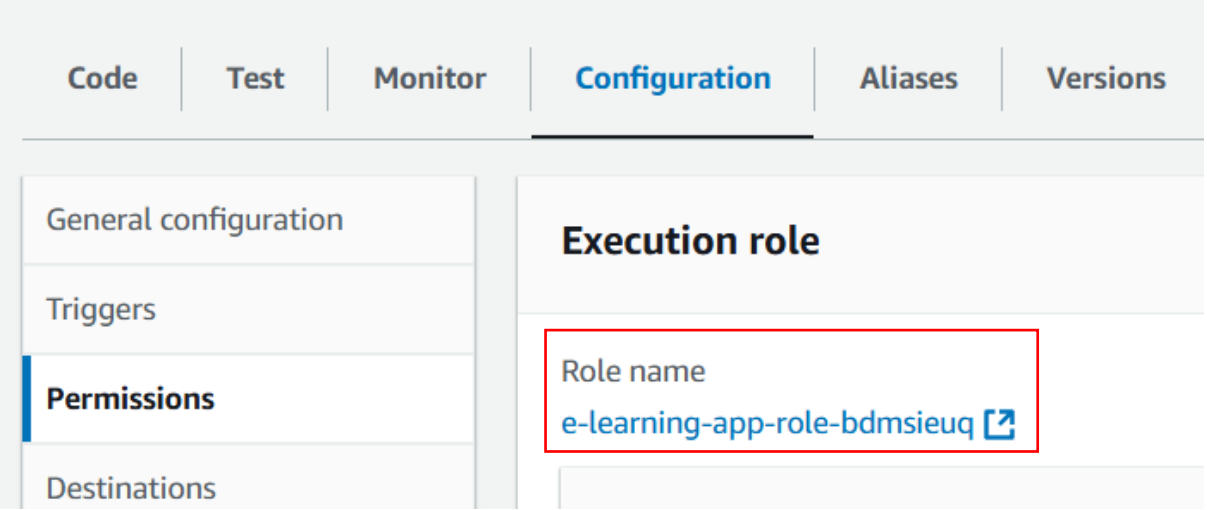
15. Now we need to create an AWS Lambda function to move forward with our lab. Here give your function a name choose runtime environment as Python and click on create.

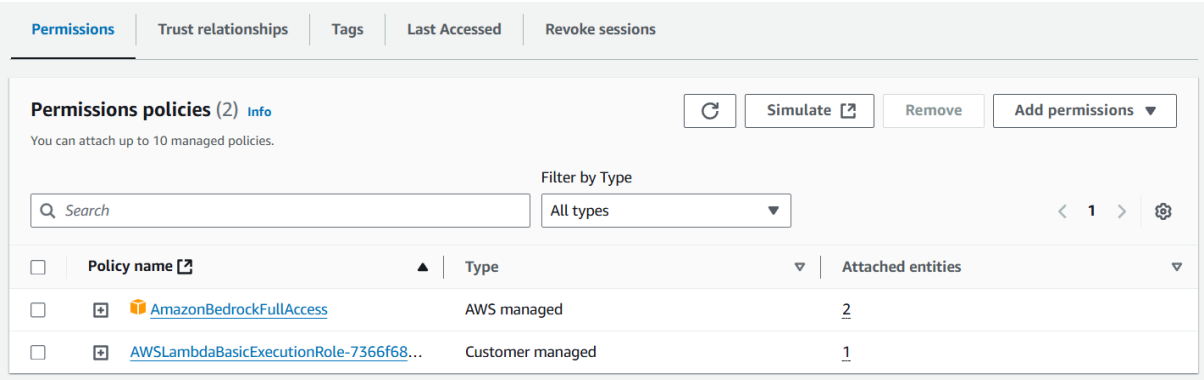


16. Once your function is created you need to increase the time out time from the general configuration.



17. Then we need to provide the required permission to our lambda function. So, to your role attach the bedrock full access permission.





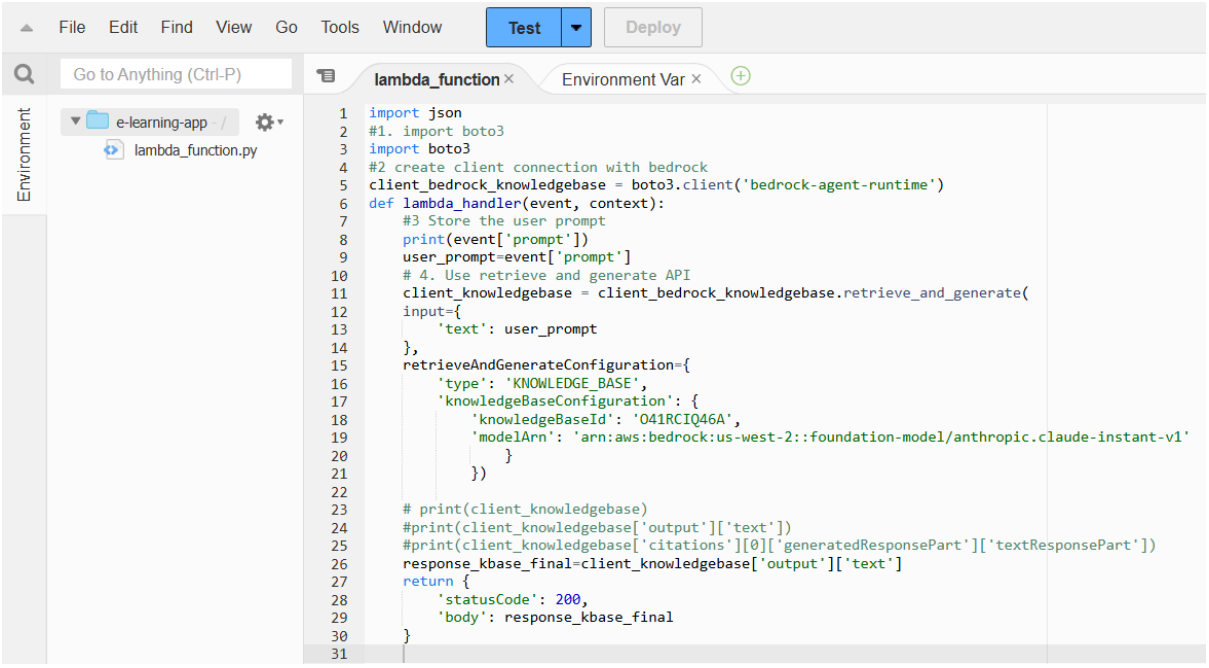
18. Then in your lambda function come to the code section use the code given below and deploy it.

```
import json
#1. import boto3
import boto3
#2 create client connection with bedrock
client_bedrock_knowledgebase = boto3.client('bedrock-agent-runtime')
def lambda_handler(event, context):
    #3 Store the user prompt
    print(event['prompt'])
    user_prompt=event['prompt']
    # 4. Use retrieve and generate API
```

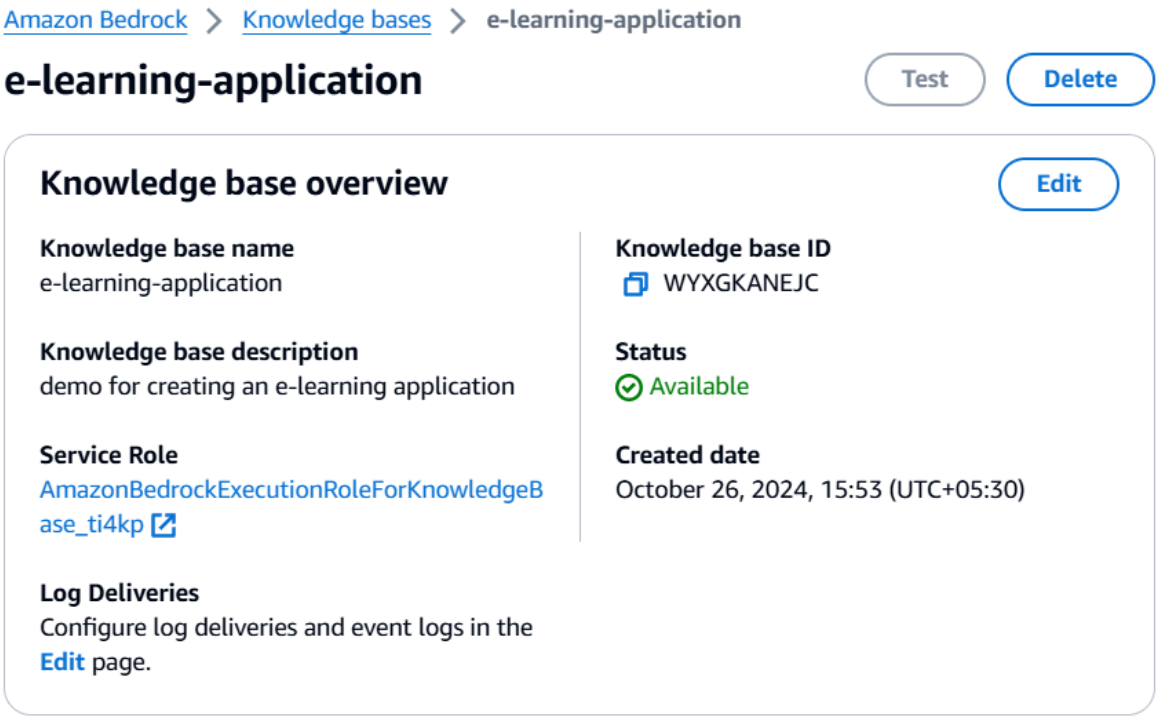
```
    client_knowledgebase =
client_bedrock_knowledgebase.retrieve_and_generate(
    input={
        'text': user_prompt
    },
    retrieveAndGenerateConfiguration={
        'type': 'KNOWLEDGE_BASE',
        'knowledgeBaseConfiguration': {
            'knowledgeBaseId': 'WYXGKANEJC',
            'modelArn': 'arn:aws:bedrock:us-east-1::foundation-
model/anthropic.claude-instant-v1 '
            }
        })

    # print(client_knowledgebase)
    #print(client_knowledgebase['output']['text'])

#print(client_knowledgebase['citations'][0]['generatedResponsePart']['textRespo
nsePart'])
    response_kbase_final=client_knowledgebase['output']['text']
    return {
        'statusCode': 200,
        'body': response_kbase_final
    }
```

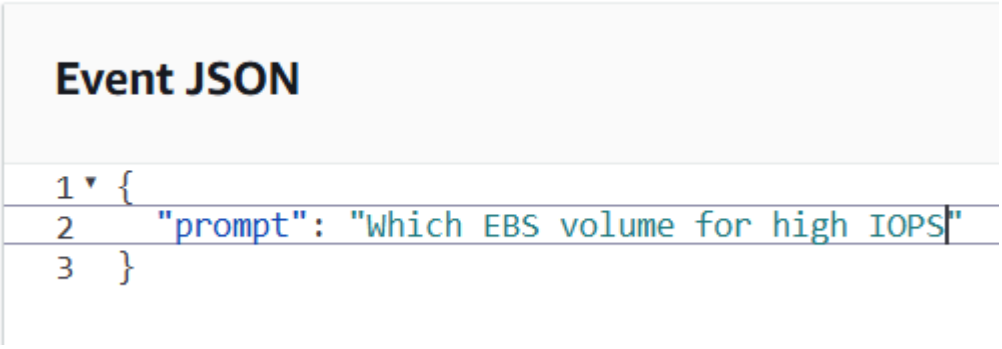


19. In the above code in line number 18 you need to provide the knowledge base ID and you can get this ID from your knowledge base, so go to your knowledge base, and there in the overview you can get this ID.

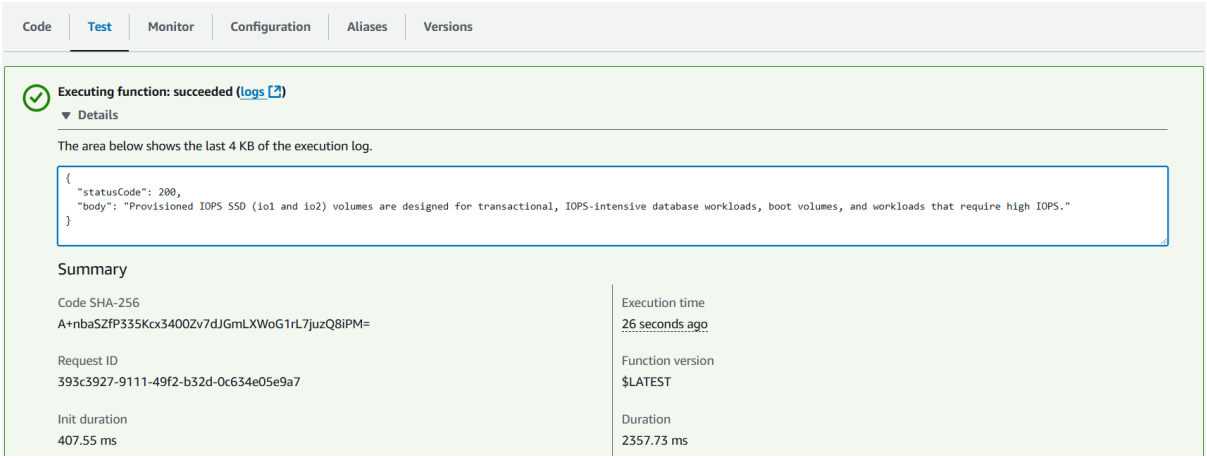


20. Now after deploying your code, you need to create a test event. In the event JSON area, you need to write your prompt like this to get the proper output. Click on save and then on test.

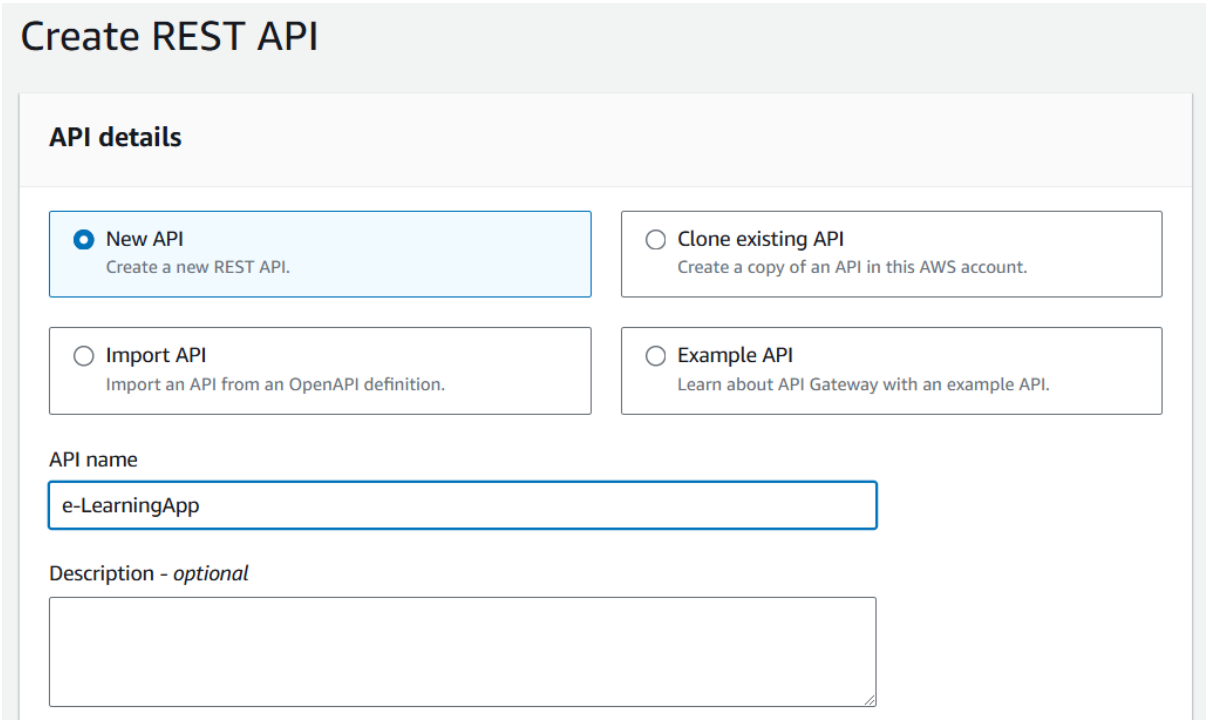
```
{
  "prompt": "Which EBS volume for high IOPS"
}
```



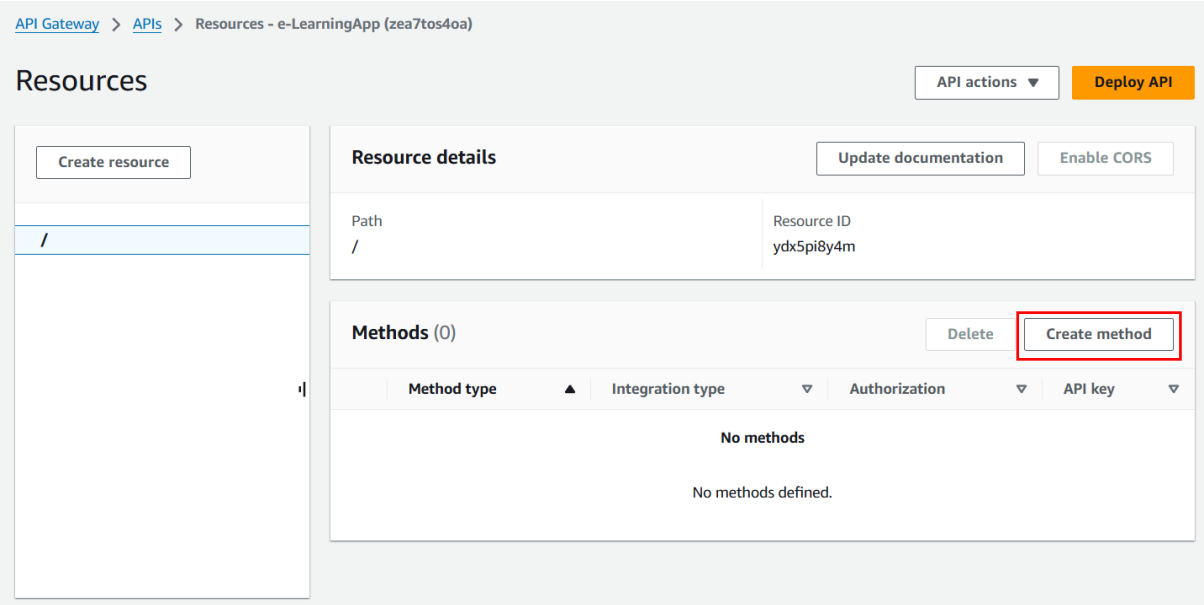
21. Below you can see that we got the response as expected.



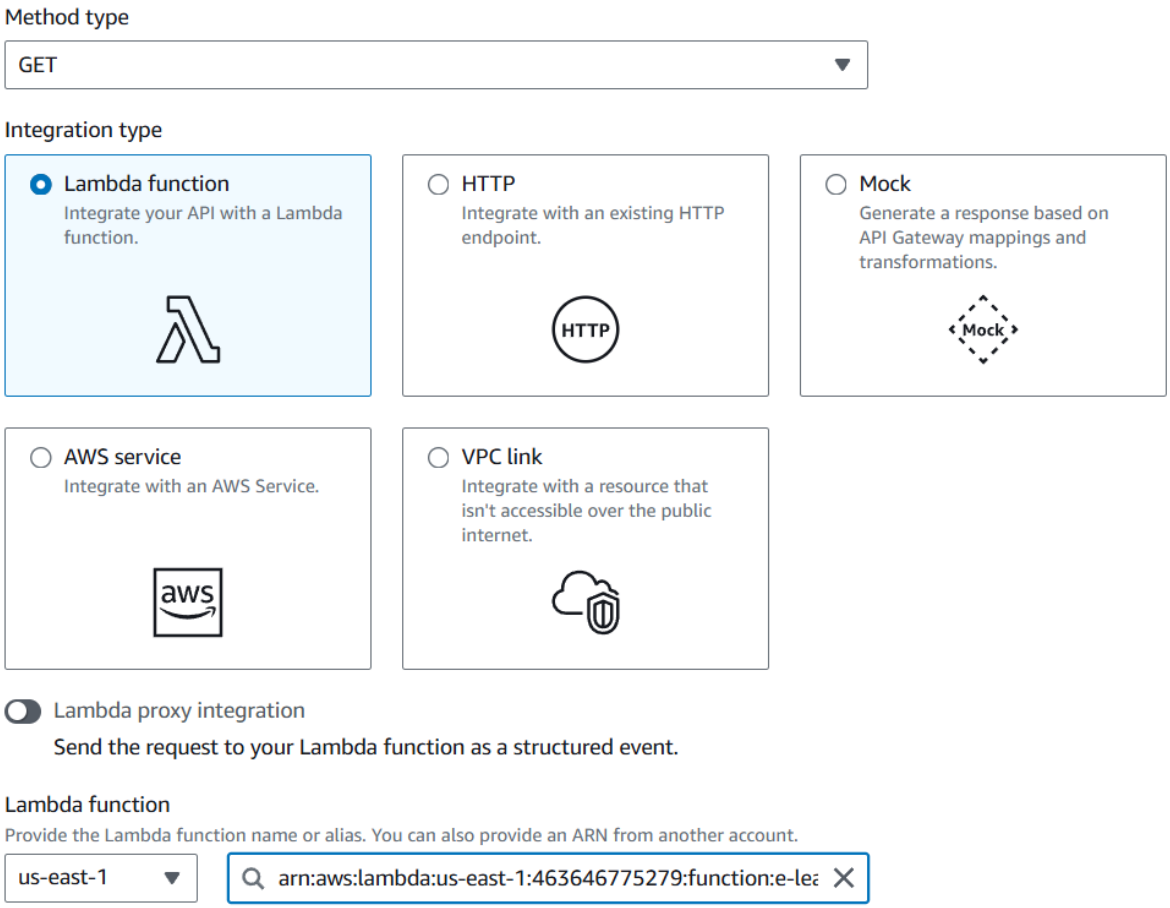
22. So, we have completed our work on the lambda function now we are going to integrate our lambda function with API Gateway.
23. Search and navigate to the API gateway and click on build REST API. Then give it a name and click on build.



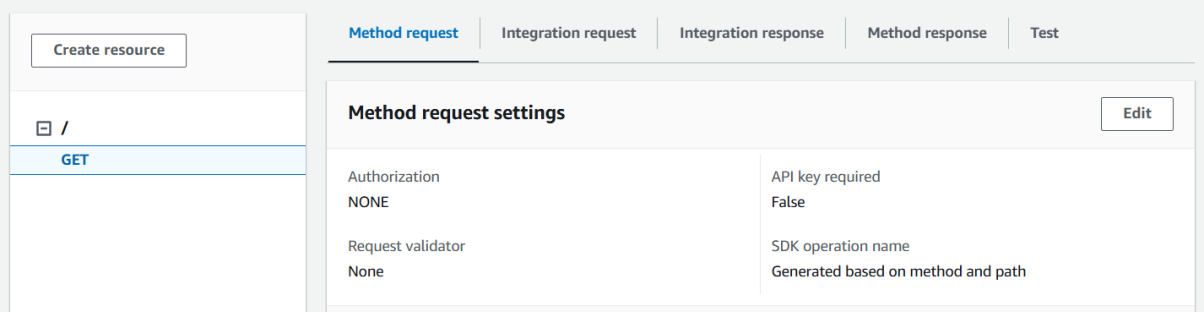
24. So, on our default resource we are going to create a GET method. Click on create method.



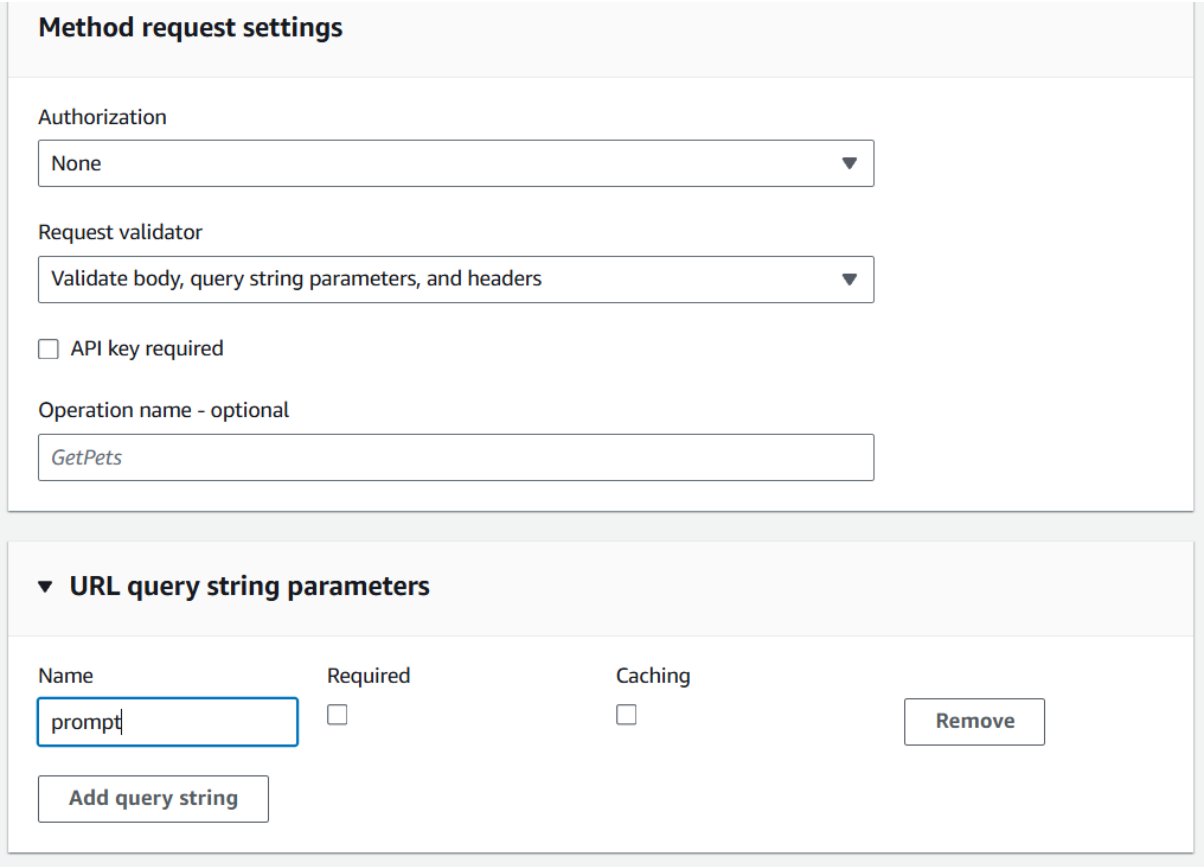
25. First choose method type as GET and then choose your lambda function.



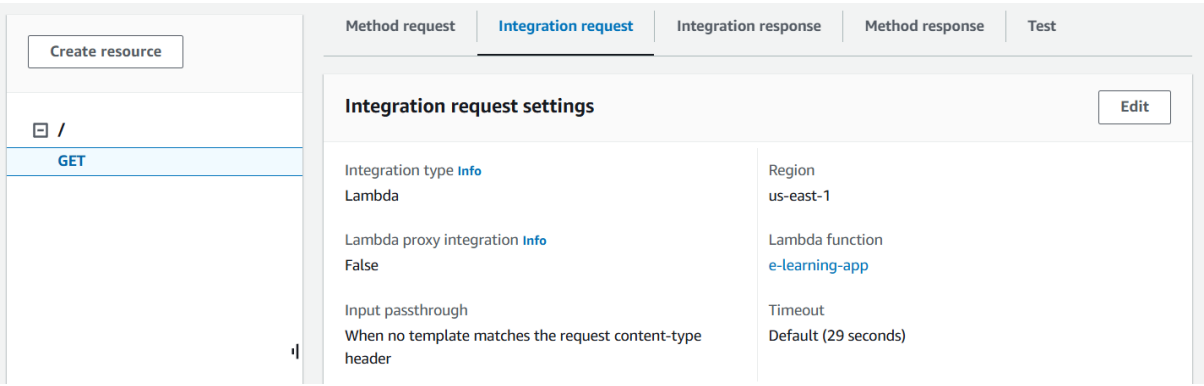
26. In your GET method choose method request and click on edit.



27. Here you need to choose the **same validator in the request validator** expand **URL query string parameters** and click on **add query string** then in the name **write prompt** and click on save.



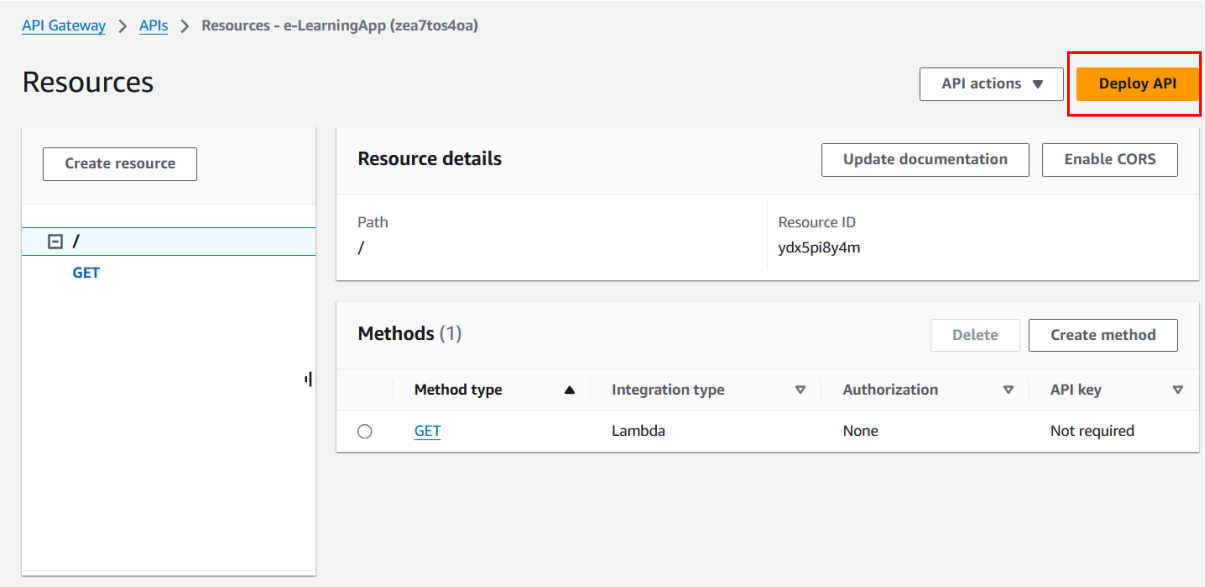
28. Go to integration request in GET method and click on edit.



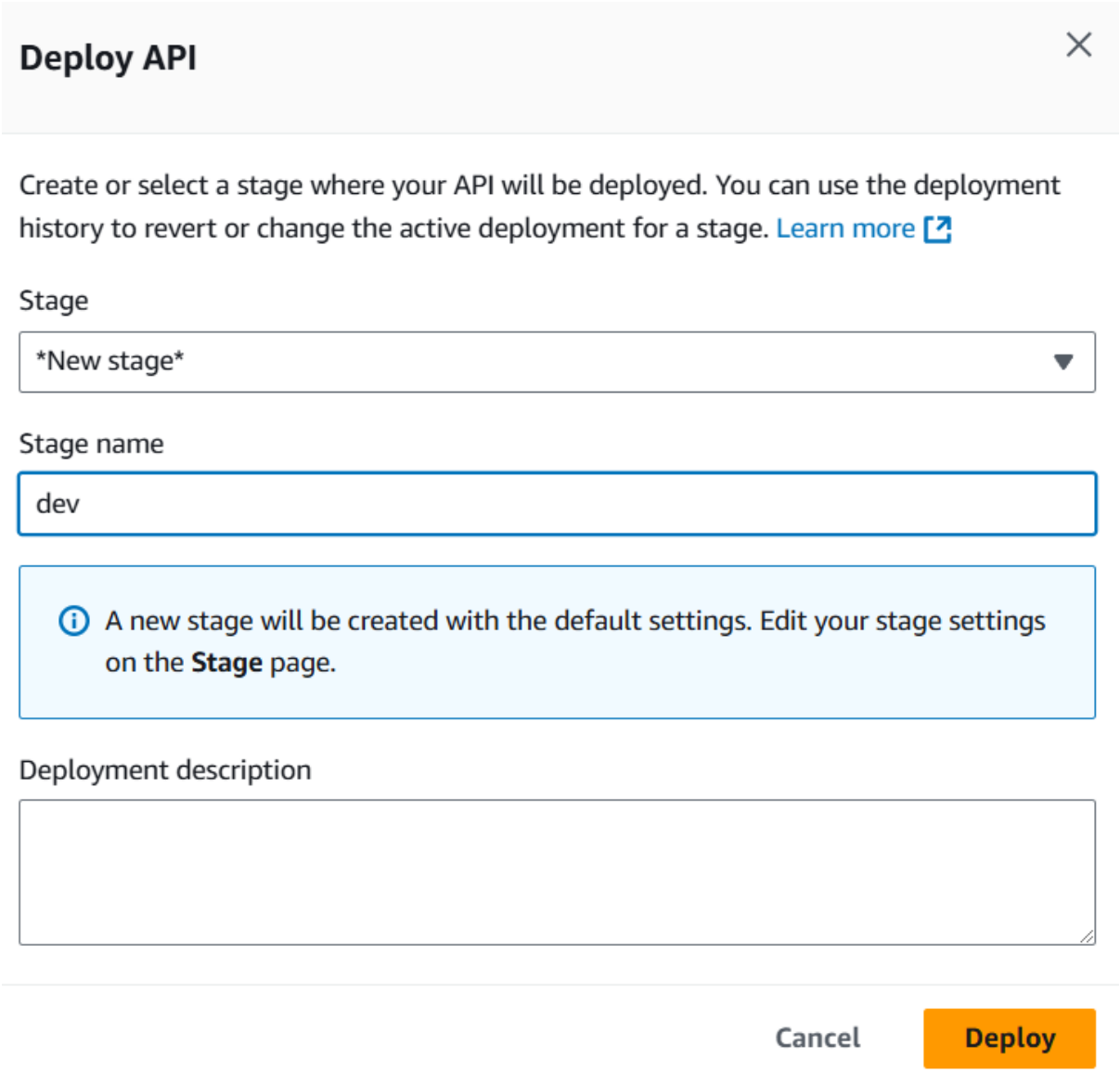
29. Scroll down to the bottom expand mapping templates and click on add mapping templates then in the content type and template body write the same thing as you see below.

{
"prompt":"$input.params('prompt')"
}

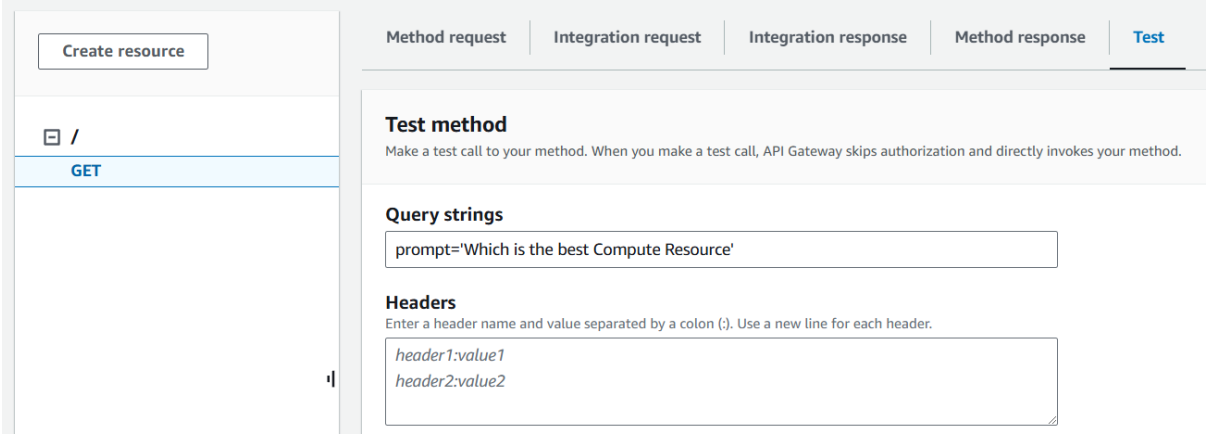
30. After that choose your resource and click on deploy API.



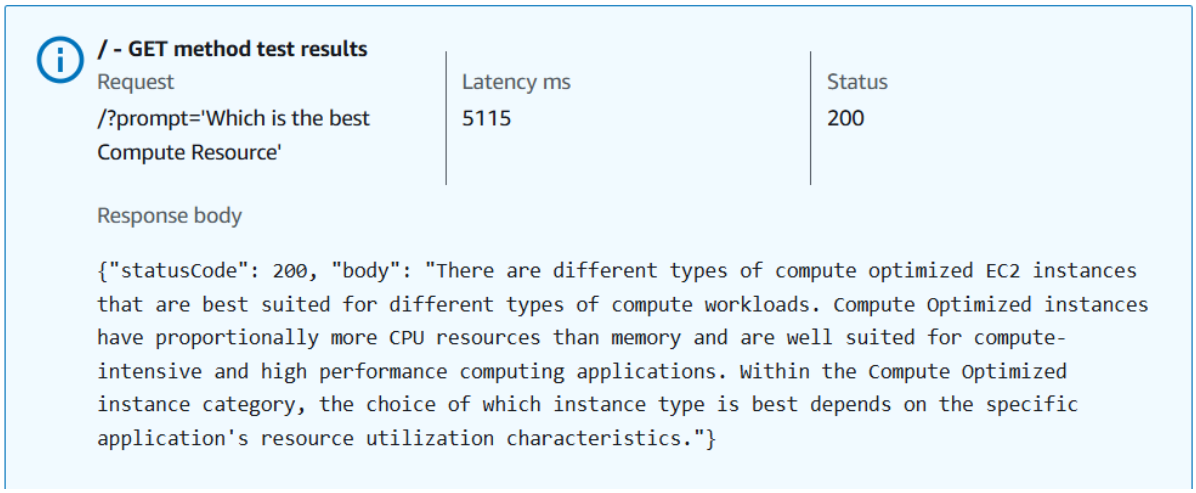
31. Choose new stage and give it a name then click on deploy.



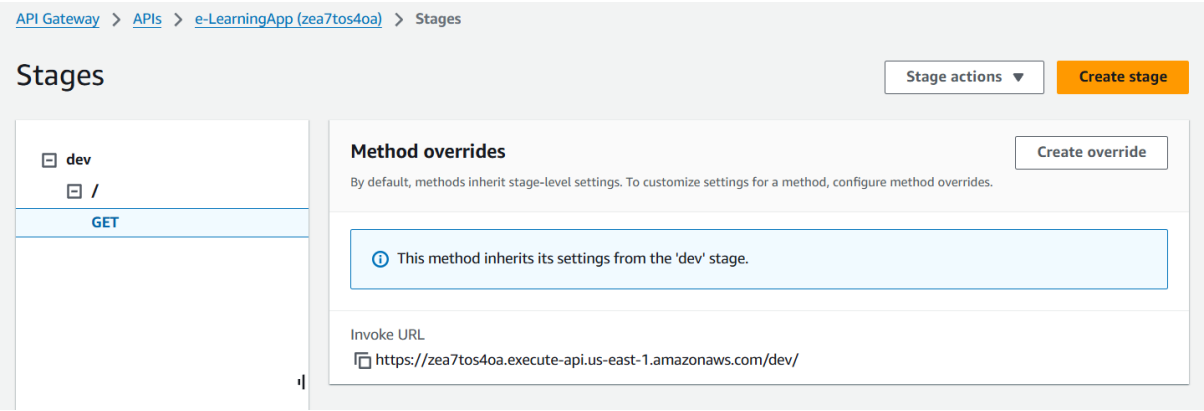
32. Now come to GET method and choose Test then in the query strings write your prompt in the same way and click on test.



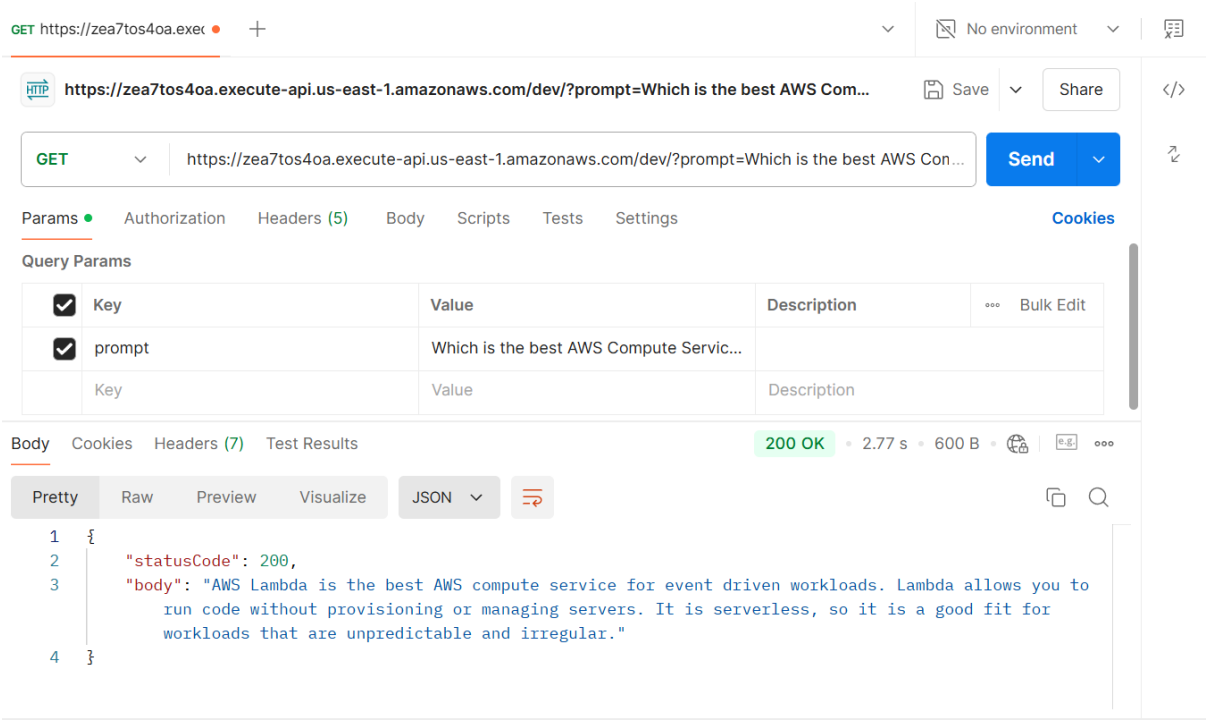
33. Below you can see that you get the response as expected.



34. Then in your API gateway choose stages and then come to the GET stage and from here copy the invoke URL.



35. After that you need to open Post Man and create a new workspace here then choose the GET method and paste the invoke URL here.
36. In the key you need to write the prompt and, in the value, you need to ask it a question then you will see that it was able to give the response.



37. Once you are done with the lab you need to delete your knowledge base from AWS Bedrock. So, go to bedrock then to knowledge choose it and click on delete after that delete all the other resources.