



AWS Bedrock (Building a Retail Bank Agent)

1. In this lab, we are going to build a **Retail banking agent** using **Bedrock agents** and **Knowledge base**.
2. Navigate to Dynamo DB and there we will create a table. In the table name give the same name and then in the partition key write Account ID and choose number then click to create this table.

Create table

Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
 ▼
1 to 255 characters and case sensitive.

3. Once your table is created, you need to go inside it, click on **Explore Items**, scroll down, and click on **Create Items**.
4. You need to create the same items in the same way as you can see below.

DynamoDB > [Explore items: customerAccountStatus](#) > Create item

Create item [Form](#) [JSON view](#)

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#) (2)

Attributes		Add new attribute ▾
Attribute name	Value	Type
<input checked="" type="checkbox"/> AccountID - Partition key	5555	Number
<input type="text" value="AccountName"/>	<input type="text" value="John"/>	String <input type="button" value="Remove"/>
<input type="text" value="AccountStatus"/>	<input type="text" value="Active"/>	String <input type="button" value="Remove"/>
<input type="text" value="Reason"/>	<input type="text" value="Active"/>	String <input type="button" value="Remove"/>

DynamoDB > Explore items: customerAccountStatus > Create item

Create item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

Attributes		Type	Add new attribute ▾
Attribute name	Value	Type	
AccountID - Partition key	6666	Number	<input type="button" value="Remove"/>
AccountName	Thomas	String	<input type="button" value="Remove"/>
AccountStatus	Pending	String	<input type="button" value="Remove"/>
Reason	InvalidIdentification	String	<input type="button" value="Remove"/>

Attributes		Type	Add new attribute ▾
Attribute name	Value	Type	
AccountID - Partition key	7777	Number	<input type="button" value="Remove"/>
AccountName	Manju	String	<input type="button" value="Remove"/>
AccountStatus	Pending	String	<input type="button" value="Remove"/>
Reason	InvalidAddressProof	String	<input type="button" value="Remove"/>

5. Below you can see that we have created all three items that we needed.

Items returned (3)

<input type="checkbox"/>	AccountID (Number)	AccounNa...	AccountStatus	Reason	▼
<input type="checkbox"/>	7777	Manju	Pending	InvalidAddressProof	<input type="button" value="Actions ▾"/>
<input type="checkbox"/>	6666	Thomas	Pending	InvalidIdentification	<input type="button" value="Create item"/>
<input type="checkbox"/>	5555	John	Active	Active	<input type="button" value="Actions ▾"/> <input type="button" value="Copy"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

6. Now we need to create a lambda function. So, navigate to it and click on Create. Give it a name and choose Python as your runtime environment.

- Author from scratch
Start with a simple Hello World example.
- Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
- Container image
Select a container image to deploy for your function.

Basic information

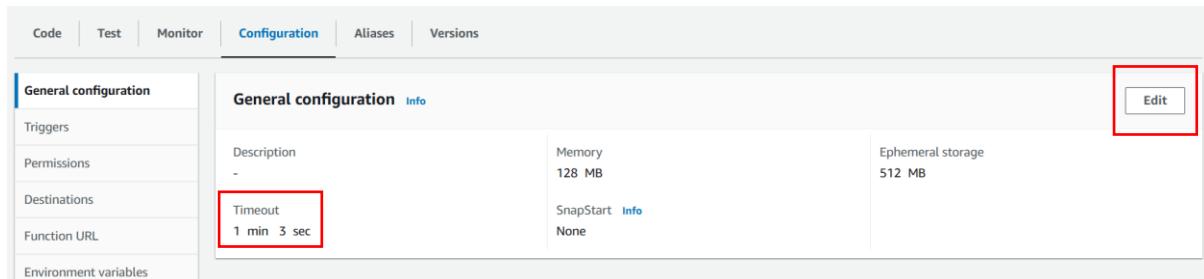
Function name
Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 [▼](#) [C](#)

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

7. Once your function is created you need to go to configuration and click on edit to increase the timeout as you can see below.



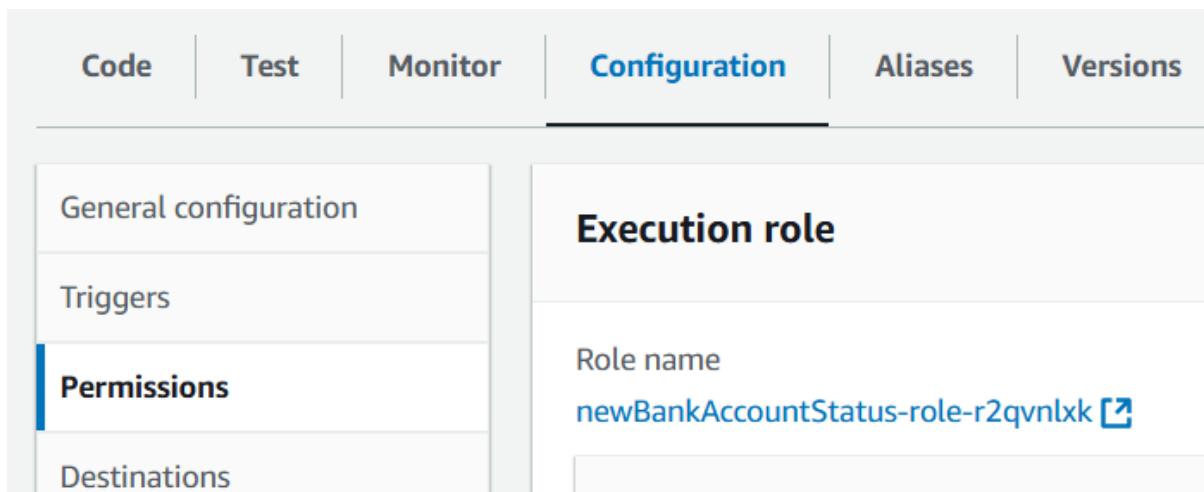
Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration [Info](#)

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart	
1 min 3 sec	Info	
	None	

[Edit](#)

8. Then go to Permission and click on the role to edit it with the appropriate permission. So, for the permission you need to add Dynamo DB full access and Bedrock full access.



Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration

Triggers

Permissions

Destinations

Execution role

Role name
[newBankAccountStatus-role-r2qvnlxk](#) 

Permissions	Trust relationships	Tags	Last Accessed	Revoke sessions
Permissions policies (3) Info				
You can attach up to 10 managed policies.				
Filter by Type				
<input type="text" value="Search"/> All types				
Policy name	Type	Attached entities		
AmazonBedrockFullAccess	AWS managed	3		
AmazonDynamoDBFullAccess	AWS managed	36		
AWSLambdaBasicExecutionRole-827f711...	Customer managed	1		

9. Then use the lambda code file you get with this lab copy the code and then paste it in the lambda function.

```

import json
#1 Import boto3 and create client connection with DynamoDB - Link to
documentation
https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/dynamodb/client/get\_item.html
import boto3
client=boto3.client('dynamodb')

def lambda_handler(event, context):
#2 Print event value and store the event details in a variable
    print(f"This is the input from agent{event}")
    account_id=event['parameters'][0]['value']#3 Create a request syntax to
retrieve data from the DynamoDB Table using GET Item method -
https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/dynamodb/client/get\_item.html
    response = client.get_item(
        TableName='customerAccountStatus',
        Key={'AccountID':{'N': account_id}})
#4 Store and print the response
    print(response)
#5 Format the response as per the requirement of Bedrock Agent Action Group -
https://docs.aws.amazon.com/bedrock/latest/userguide/agents-lambda.html
    response_body = {
        'application/json': {
            'body': json.dumps(response)
        }
    }

    action_response = {
        'actionGroup': event['actionGroup'],
        'apiPath': event['apiPath'],
        'httpMethod': event['httpMethod'],
    }

```

```

    'httpStatusCode': 200,
    'responseBody': response_body
}

session_attributes = event['sessionAttributes']
prompt_session_attributes = event['promptSessionAttributes']

api_response = {
    'messageVersion': '1.0',
    'response': action_response,
    'sessionAttributes': session_attributes,
    'promptSessionAttributes': prompt_session_attributes
}

return api_response

```

#6 Print the final response

TODO implement

10. The Bedrock agent uses an OpenAPI schema to determine which Lambda function to invoke based on user queries. The schema includes details like API title, version, and descriptions to map user inputs, such as account ID. It specifies the request parameters and response formats for the Lambda function. This structured approach allows for clear communication between the Bedrock agent and the Lambda functions. Finally, the schema must be uploaded to an S3 bucket for the Bedrock agent to access it during operation.
11. So, you need to create an S3 bucket and upload the YAML to it. You will get this YAML with the lab. Also, read the YAML file.

The screenshot shows the Amazon S3 console interface. At the top, the navigation bar reads "Amazon S3 > Buckets > bankaccountstatusbucket". Below the navigation, the bucket name "bankaccountstatusbucket" is displayed with a "Info" link. A horizontal menu bar includes "Objects" (which is selected), "Properties", "Permissions", "Metrics", "Management", and "Access Points". Under the "Objects" tab, there is a sub-header "Objects (1) Info". To the right of this are several actions: "Copy S3 URI", "Copy URL", "Download", "Open", "Delete", "Actions", "Create folder", and a yellow "Upload" button. Below these actions is a search bar with the placeholder "Find objects by prefix". Further down is a table listing the single object. The table has columns: "Name", "Type", "Last modified", "Size", and "Storage class". The object listed is "newBankAccountStatus.yaml", which is a "yaml" file last modified on "October 29, 2024, 11:26:35 (UTC+05:30)" with a size of "1.5 KB" and "Standard" storage class. Navigation arrows and a refresh icon are at the bottom of the table.

Name	Type	Last modified	Size	Storage class
newBankAccountStatus.yaml	yaml	October 29, 2024, 11:26:35 (UTC+05:30)	1.5 KB	Standard

12. After that we are going to create the Bedrock agent for that first navigate to Bedrock. Here you need to expand Builder tools and click to open Agents.

▼ Builder tools

Prompt management [Preview](#)

Knowledge bases

Agents

Prompt flows [Preview](#)

13. First, we are going to create our agent. Click on create agent.

The screenshot shows the 'Agents' section of the Amazon Bedrock interface. At the top, there are two tabs: 'Overview' (selected) and 'Deploy'. Below the tabs, there are two sections: 'Prepare' (with a plus icon) and 'Deploy' (with a cube icon). Under 'Prepare', there is a brief description: 'Create your Agent by selecting a Foundation model, and adding Action groups. After creation you can test out the Agent in real-time and create multiple versions.' Under 'Deploy', there is a brief description: 'Create and associate Aliases to deploy an Agent version in your application. Point an Alias to a specific version of your Agent to test it before deploying it to your client application.' At the bottom, there is a table titled 'Agents (0)' with columns for Name, Status, Description, and Last updated. A search bar labeled 'Find Agents' is at the top of the table. There are buttons for Delete, Edit, and Create Agent (highlighted in orange). Below the table, it says 'No agents' and 'No agents to display'.

14. First, give this agent a name and a description and click on create.

Create Agent



Name

bank-agent

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.

Description - optional

This agent will help customers with the status of their opened bank accounts.

The description can have up to 200 characters.

[Cancel](#)

[Create](#)

15. Now you need to choose various properties for your agent so, here you can see that we need to create a new role for our agent.

[Amazon Bedrock](#) > [Agents](#) > [bank-agent](#) > Agent builder: bank-agent

Agent builder [Info](#)

Manual Assistant Test Prepare Save **Save and exit**

Agent details

Agent name
bank-agent
Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.

Agent description - optional
This agent will help customers with the status of their opened bank accounts.
The description can have up to 200 characters.

Agent resource role
 Create and use a new service role
 Use an existing service role
AmazonBedrockExecutionRoleForAgents_FGW8ET85711

16. Then we are selecting Claude 3 Sonnet as our model and we are proving an instruction for our agent. If this model does not work or it gives errors then you can change it.

“You are a banking assistant at a retail bank. You are friendly and polite. You help resolve customer queries by providing bank customers with the status of their new bank accounts.”

Select model
A Claude 3 Sonnet v1 | On-demand
[Change](#)

Instructions for the Agent
Provide clear and specific instructions for the task the Agent will perform. You can also provide certain style and tone.
You are a banking assistant at a retail bank. You are friendly and polite. You help resolve customer queries by providing bank customers with the status of their new bank accounts.
This instruction must have a minimum of 40 characters.

► **Additional settings**

17. Expand the additional settings and enable the user input.

▼ Additional settings

Code Interpreter [Preview](#)

Code Interpreter enables agents to handle tasks that involve writing, running, testing, and troubleshooting code in a secure environment.

- Enabled
 Disabled

User input

Select whether the agent can prompt additional information from the user when it does not have enough information to respond to an utterance.

- Enabled

Allow agent to ask the user clarifying questions to capture necessary inputs.

- Disabled

The selected foundation model within the Agent will make a best guess at invoking the appropriate action groups.

18. After that scroll up and click on save and click on save and exit button to save your changes. Below you can see that all the changes were saved now we need to click on Prepare to prepare the agent.

The screenshot shows the Amazon Bedrock Agent Builder interface. On the left, the 'Agent overview' section displays the following details for the 'bank-agent':

Name	ID	bank-agent	45LZTGMJ1D
Description	Status	This agent will help customers with the status of their opened bank accounts. NOT_PREPARED	
Creation date	Last prepared	October 29, 2024, 11:54 (UTC+05:30)	
Permissions	Agent ARN	arn:aws:iam::463646775279:role/service-role/AmazonBedrockExecutionRoleForAgents_FWG8ET8571I	
User Input	Memory	DISABLED	
Idle session timeout	KMS key	600 seconds	

On the right, the 'Test' tab is active, showing a message box with the instruction: "Prepare the Agent to test the latest changes." A 'Prepare' button is visible. Below the message box is a text input field labeled "Enter your message here" and a "Run" button.

19. Here you can see that our agent status has been changed to prepared.

Agent overview

Name bank-agent	ID 45LZTGMJ1D
Description This agent will help customers with the status of their opened bank accounts.	Status PREPARED
Creation date October 29, 2024, 11:54 (UTC+05:30)	Last prepared October 29, 2024, 12:01 (UTC+05:30)
Permissions arn:aws:iam::463646775279:role/service-role/AmazonBedrockExecutionRoleForAgents_FGW8ET85711	Agent ARN arn:aws:bedrock:us-east-1:463646775279:agent/45LZTGMJ1D
User Input ENABLED	Memory Disabled
Idle session timeout 600 seconds	KMS key -

Test TestAlias: Working draft Info Using ODT Change

Enter your message here

Run

20. Here you can see that our model is responding us but it does not have the information for the accounts.
21. For that we need to create an action group for our agent. So, you need to click on Edit in the agent builder button.

Test TestAlias: Working draft Info Using ODT Change

Hi

Hi!

What is the status for account 5555?

Sorry, I don't have enough information to answer that.

Enter your message here

Trace Trace (1) Copy all

Request ID: 18478af6-1621-422c-b2c2-c01e730232f3

Pre-Processing Trace Orchestration and knowledge base Post-Proc

Pre-Processing Trace Show config

After running an input in the test window, this pre-processing trace allows you to explore the trace steps to understand if the user input was identified as malicious or outside of the agent's domain.

Trace Step 1

22. Then scroll down to action groups and click on Add.

Action groups (0) [Info](#)

Delete Add

Find action groups

< 1 > |

Name	Description	State	Last updated
------	-------------	-------	--------------

No action groups

No actions group to display

Add

23. First, you need to give it a name then in the action group type choose Define with API schemas.

[Amazon Bedrock](#) > [Agents](#) > [bank-agent](#) > [Agent builder: bank-agent](#) > Add action

Create Action group

Action group details

Enter Action group name

Description - optional

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). This description can have up to 200 characters.

Action group type

Select what type of action group to create

Define with function details
Specify functions and define parameters as JSON objects that will be associated to the action group invocation.

Define with API schemas
Specify a Lambda or API Gateway and define a schema to specify the APIs that the agent can invoke to carry out its tasks.

24. Here you need to click on Select an existing Lambda function and then choose your lambda function.

Action group invocation

Specify a Lambda function that will be invoked based on the action group identified by the Foundation model during orchestration.

Select how to define the Lambda function

Quick create a new Lambda function - *recommended*

An Amazon Lambda function will be created in your account on your behalf. No further configurations are necessary.

Select an existing Lambda function

Use an existing Lambda function for this action group.

Return control

Agent responses in the test window will prompt the user for function details to generate a response. No further configurations are necessary.

Select Lambda function

Select a previously created Lambda function or visit [AWS Lambda](#) ([Label= '\\$AWS Lambda \(\\$opens in new tab\)'](#))  to create a new function.

newBankAccountStatus 

Function version

\$LATEST 

[View](#) 



25. After that in the action group schema you need click on select an existing API schema and give the URL of your YAML file stored in the S3 bucket. Then click on Create.

Action group schema

Select an existing schema or create a new one via the in-line editor to define the APIs that the agent can invoke to carry out its tasks.

Select an existing API schema

Select from an existing S3 or define a schema

Define via in-line schema editor

Use provided sample code or import and edit from S3

S3 Url

 s3://bankaccountstatusbucket/newBankAccountSt 

[View](#) 

[Browse S3](#)

[Cancel](#)

[Create](#)

26. Once your action group has been created then you need to click on save and exit. Also click on prepare to prepare your data for the new settings.
27. Now if you ask this agent the questions about the account status then you will see that it is still giving you the errors.

Test **TestAlias: Working draft** ▾ **Info**
Using ODT **Change**



What is the status of Account 5555?



Access denied while invoking Lambda function
arn:aws:lambda:us-east-
1:463646775279:function:newBankAccountStat
us. Check the permissions on Lambda function
and retry the request.



Show trace >



28. So, to resolve it you need to come back to your lambda function. Go to configuration, then to permission and scroll down to resource-based policy statements. Click on add permission.

Resource-based policy statements **Info**

Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.

Statement ID	Principal	PrincipalOrgID	Conditions	Action
No policy statements				
Add permissions				

29. Here you need to choose AWS Service and, in the service, choose others. Then int eh Statement ID, Principal write the same as you can see below.
30. For the source ARN you need to copy the ARN of your Bedrock agent and paste here. In the action you need to choose Lambda invoke function, same as shown below. Then click on save.

Edit policy statement

AWS account

Grant permissions to another AWS account, user, or role.

AWS service

Grant permissions to another AWS service.

Function URL

Grant permissions to invoke your function through the function URL.

Service

The AWS service to grant permissions to.

Other



Statement ID

Enter a unique statement ID to differentiate this statement within the policy.

bedrock-agent

Principal

The service principal for this AWS service. [Learn more](#)

bedrock.amazonaws.com

Source ARN

The ARN for a resource. Find the ARN in the related service console.

arn:aws:bedrock:us-east-1:463646775279:agent/45LZTGMJ1D

Action

Choose an action to allow.

lambda:InvokeFunction



31. Now you can see that it has answered us the question for status of bank accounts.

Test TestAlias: Working draft ▾ [Info](#)

Using ODT [Change](#)



What is status of account 5555?



The status of account 5555 is Active.



[Show trace >](#)



What is the status of account 6666?



The status of account 6666 is Pending.



[Show trace >](#)

32. It will also answer you if you ask it the proper questions too.



what is the status of AccountID=7777?



The status of account 7777 is Pending.



[Hide trace](#)



why the status is pending ?



The status of the account is pending because the address proof provided was invalid, as per the information retrieved from checking the status of account 7777 in previous conversations.

33. The Bedrock agent can handle user queries about error codes like "invalid identification" by integrating with a Retrieval-Augmented Generation (RAG) approach. Users can ask for explanations of these codes, which aren't fully covered in DynamoDB. A PDF document detailing these codes is created and uploaded to an S3 bucket. The RAG solution breaks this PDF into chunks and generates vector embeddings for storage. When a user poses a question, it retrieves relevant information from the vector store, providing comprehensive answers.
34. First, we are going to create knowledge base in Bedrock. Remember you cannot create it using your root account you need to create an IAM account for it.
35. Now in your bedrock service go to Knowledge base and click on create. In step 1, you need to give it a name and description.

[Amazon Bedrock](#) > [Knowledge bases](#) > Create knowledge base

Step 1 **Provide knowledge base details**

Step 2 Configure data source

Step 3 Select embeddings model and configure vector store

Step 4 Review and create

Provide knowledge base details

Knowledge base details

Knowledge base name

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 50 characters.

Knowledge base description - optional

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 200 characters.

36. Then for the IAM permissions choose to create new role and for the data source choose S3 bucket. Click on next.

IAM permissions

Certain permissions are necessary to access other services or perform actions in order to create this resource. For more information, see [service role](#) for Amazon Bedrock

Runtime role

- Create and use a new service role
- Use an existing service role

Service role name

AmazonBedrockExecutionRoleForKnowledgeBase_dl8hp

Choose data source

Select the data source that you want to configure in the next step. You can add up to 5 data sources in a knowledge base.



Amazon S3

Object storage service that stores data as objects within buckets.



Web Crawler - *Preview*

Web page crawler that extracts content from public web pages you are authorized to crawl.

37. In the step 2, you need to configure your S3 bucket. But first you need to create a new S3 bucket and upload the PDF document to it. Then come back to bedrock and choose your S3 bucket.

Amazon S3 > Buckets > retailbankings3bucket12

retailbankings3bucket12 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (1) [Info](#)

[Delete](#) [Actions ▾](#) [Create folder](#)

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](#)

[Find objects by prefix](#) < 1 > ⌂

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Description+of+the+various+ErrorCode+in+New+Account+Opening.pdf	pdf	October 29, 2024, 12:45:13 (UTC+05:30)	79.6 KB	Standard

- Step 1
Provide knowledge base details
- Step 2
- Configure data source**
- Step 3
Select embeddings model and configure vector store
- Step 4
Review and create

Configure data source

Configure for the chosen data source

Amazon S3 Info

Provide details to connect Amazon Bedrock to your S3 data source.

▼ Data source: knowledge-base-quick-start-neqip-data-source Delete

Data source name

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.

Data source location

- This AWS account
- Other AWS account

S3 URI

To increase the accuracy and relevance of your responses, add a .metadata.json file containing metadata for your data source to your S3 bucket. Info

S3 URI


View
Browse S3

- Add customer-managed KMS key for S3 data - *optional*

If you encrypted your S3 data, provide the KMS key here so that Bedrock can decrypt it.

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

Select embeddings model and configure vector store

Choose an embeddings model to convert the data that you will provide in the next step, and provide details for a vector data store in which Bedrock can store, manage, and update your embeddings. The embeddings model and vector store cannot be changed after creation of knowledge base.

Embeddings model

Select an embeddings model to convert your data into an embedding. Pricing depends on the model. [Learn more](#)


[Titan Text Embeddings v2](#)


By Amazon


[Titan Embeddings G1 - Text v1.2](#)


By Amazon


[Embed English v3](#)


By Cohere


[Embed Multilingual v3](#)


By Cohere

Vector dimensions

Select the vector dimension size for your embeddings model to balance accuracy, cost, and latency. Higher dimensions improves overall accuracy and requires more vector storage. [Learn more](#)

1536



Vector database

Let Amazon create a vector store on your behalf or select a previously created store to allow Bedrock to store, update and manage embeddings. You will be billed directly from the vector store provider. [Learn more](#)

Select how you want to create your vector store.

Quick create a new vector store - Recommended

We will create an Amazon OpenSearch Serverless vector store on your behalf. This cost-efficient option is intended only for development and can't be migrated to production workload later. [Learn more](#)

Choose a vector store you have created

Select Amazon OpenSearch Serverless, Amazon Aurora, MongoDB Atlas, Pinecone or Redis Enterprise Cloud and provide field mappings.

Enable redundancy (active replicas) - optional

The default configuration has active replicas disabled, which is optimal for development workloads. Enable this option if you want to enable redundant active replicas, which may increase storage costs.

Add customer-managed KMS key for Amazon Opensearch Serverless vector - optional

If you encrypted your Opensearch data, provide the KMS key here so that Bedrock can decrypt it.

[Cancel](#)

[Previous](#)

[Next](#)

40. 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.

[Amazon Bedrock](#) > [Knowledge bases](#) > [knowledge-base-retailbanking](#)

knowledge-base-retailbanking

[Test](#)

[Delete](#)

Knowledge base overview

[Edit](#)

Knowledge base name

knowledge-base-retailbanking

Knowledge base description

Bedrock agent will utilize the Knowledge Base for any queries from the customer related to the reason for his AccountID status as pending.

Service Role

[AmazonBedrockExecutionRoleForKnowledgeBase_dl8hp](#)

Knowledge base ID

[NNSB8TYMFF](#)

Status

✓ Available

Created date

October 29, 2024, 12:50 (UTC+05:30)

Log Deliveries

Configure log deliveries and event logs in the [Edit](#) page.

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

Data source (1)				
Sync Stop sync Add ▼				
Data sources contain information returned when querying a Knowledge base.				
<input style="width: 100%; height: 30px; border: 1px solid #ccc; border-radius: 15px; padding: 5px; margin-bottom: 10px;" type="text"/>				
< 1 >				
Data so... ▾ Status Data sour... Account ID Source Link				
● knowledge...	✓ Available	S3	46364677...	s3://retail...

42. Once your knowledge is ready go back to your agent and click on edit in agent builder. After that scroll down to knowledge bases and click on Add.

Knowledge bases (0) Info		
Delete Add		
<input style="width: 100%; height: 30px; border: 1px solid #ccc; border-radius: 15px; padding: 5px; margin-bottom: 10px;" type="text"/>		
< 1 > ⚙️		
Name ▾ State ✎ Instruction		
No knowledge base	No knowledge base to display	Add

43. Then you need to choose your knowledge base and give it an instruction. Click on save exit.

“Bedrock agent will utilize the Knowledge Base for any queries from the customer related to reason for his AccountID status as pending.”

Add knowledge base

Knowledge base: knowledge-base-retailbanking Clear

Select knowledge base
Select a previously created knowledge base or create a new one by visiting Knowledge base. [Learn more about Knowledge bases.](#) 

knowledge-base-retailbanking C

Knowledge base instructions for Agent
Specify instructions based on the design and type of information of the knowledge base. This will impact how the knowledge base interacts with the agent.

Bedrock agent will utilize the Knowledge Base for any queries from the customer related to reason for his AccountID status as pending.

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The instruction can have up to 150 characters.

[Cancel](#)

[Add](#)

44. After that prepare your data. Here you can see that we just greeted our agent and it responded back properly this time.

Test TestAlias: Working draft ▼ Info

Using ODT [Change](#)

Hi

 Hello! How can I help you today? I'm here to answer any questions about your bank account.

  [Hide trace](#)

45. After that we asked it another question and we got the answer for that too.



What is the status of AccountID=7777?



The status of AccountID 7777 is Pending. The reason for the pending status is InvalidAddressProof.



</>

Show trace >

46. Now delete all the resources, start by deleting the Knowledge base then the agent and rest of the things.