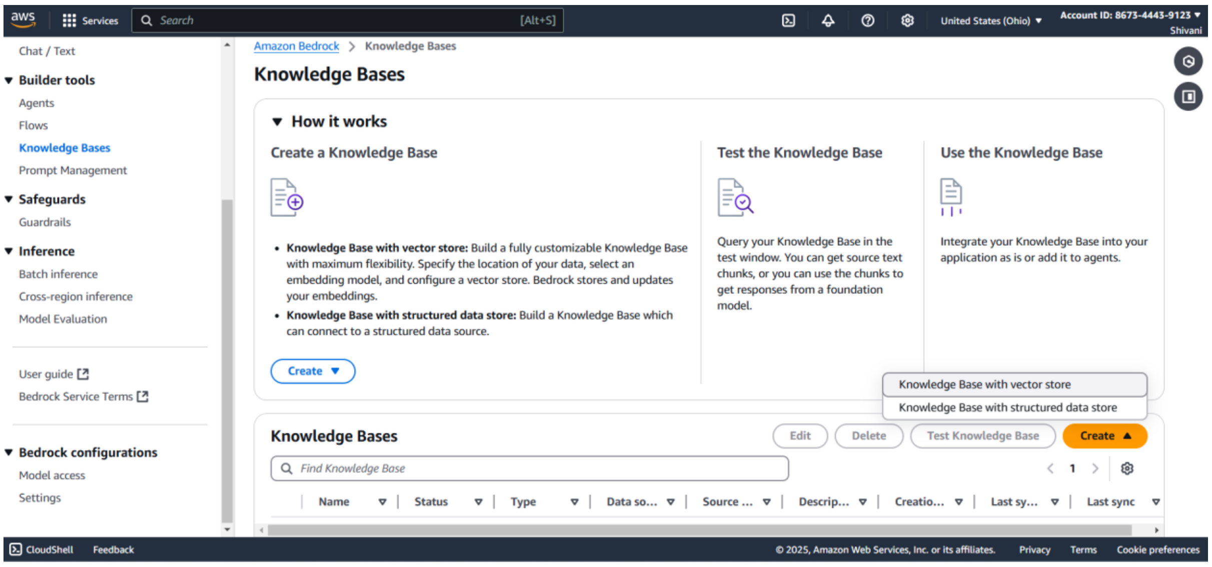
Practical-2

Deploying the RAG Model with knowledge base using AWS bedrock:

Step1: Creating a knowledge base in AWS Bedrock.



Knowledge Bases

How it works

Create a Knowledge Base

- **Knowledge Base with vector store:** Build a fully customizable Knowledge Base with maximum flexibility. Specify the location of your data, select an embedding model, and configure a vector store. Bedrock stores and updates your embeddings.
- **Knowledge Base with structured data store:** Build a Knowledge Base which can connect to a structured data source.

Test the Knowledge Base

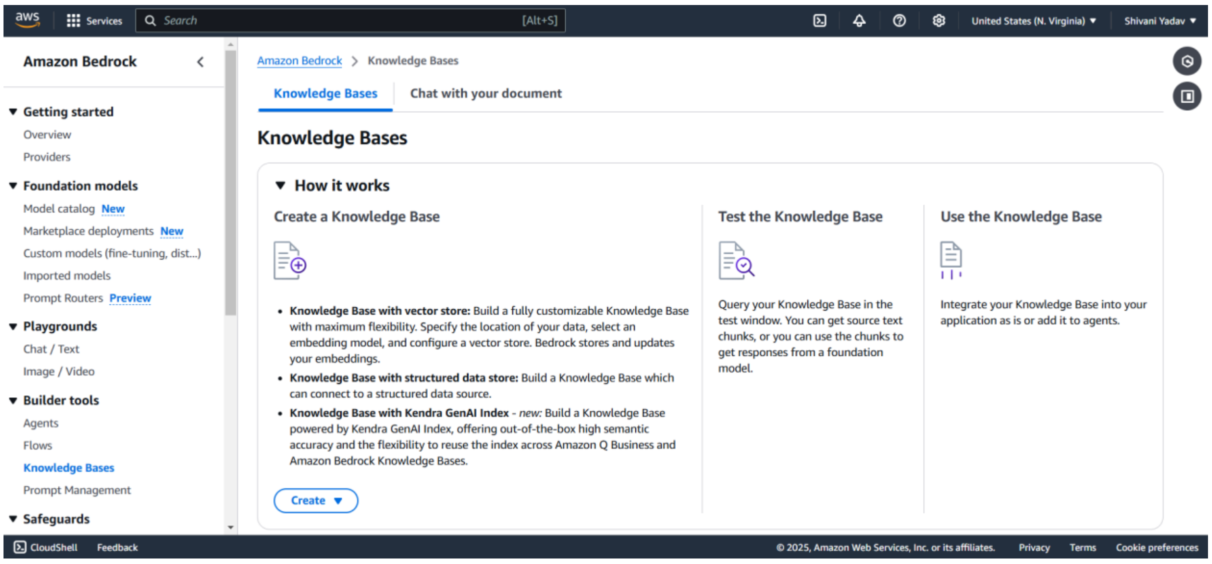
Query your Knowledge Base in the test window. You can get source text chunks, or you can use the chunks to get responses from a foundation model.

Use the Knowledge Base

Integrate your Knowledge Base into your application as is or add it to agents.

Knowledge Base with vector store
Knowledge Base with structured data store
Create

Step2: Select vector store in knowledge base.



Knowledge Bases

How it works

Create a Knowledge Base

- **Knowledge Base with vector store:** Build a fully customizable Knowledge Base with maximum flexibility. Specify the location of your data, select an embedding model, and configure a vector store. Bedrock stores and updates your embeddings.
- **Knowledge Base with structured data store:** Build a Knowledge Base which can connect to a structured data source.
- **Knowledge Base with Kendra GenAI Index** - *new:* Build a Knowledge Base powered by Kendra GenAI Index, offering out-of-the-box high semantic accuracy and the flexibility to reuse the index across Amazon Q Business and Amazon Bedrock Knowledge Bases.

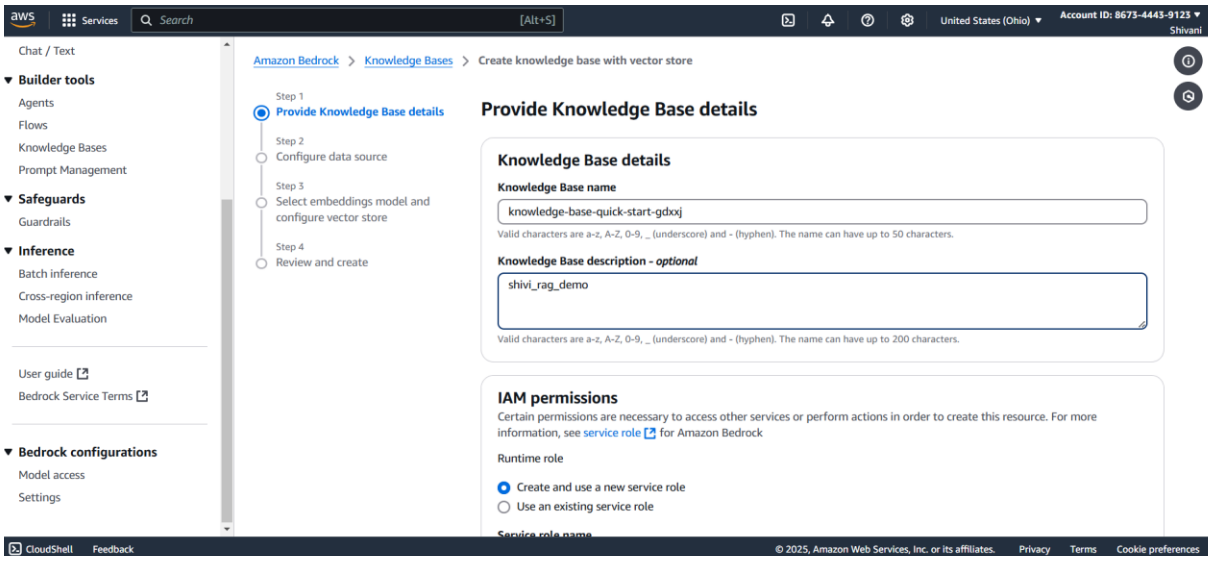
Test the Knowledge Base

Query your Knowledge Base in the test window. You can get source text chunks, or you can use the chunks to get responses from a foundation model.

Use the Knowledge Base

Integrate your Knowledge Base into your application as is or add it to agents.

Step3: Configuring the knowledge base.



Amazon Bedrock > Knowledge Bases > Create knowledge base with vector store

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: knowledge-base-quick-start-gdxxj

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

Knowledge Base description - *optional*: shivi_rag_demo

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

IAM permissions

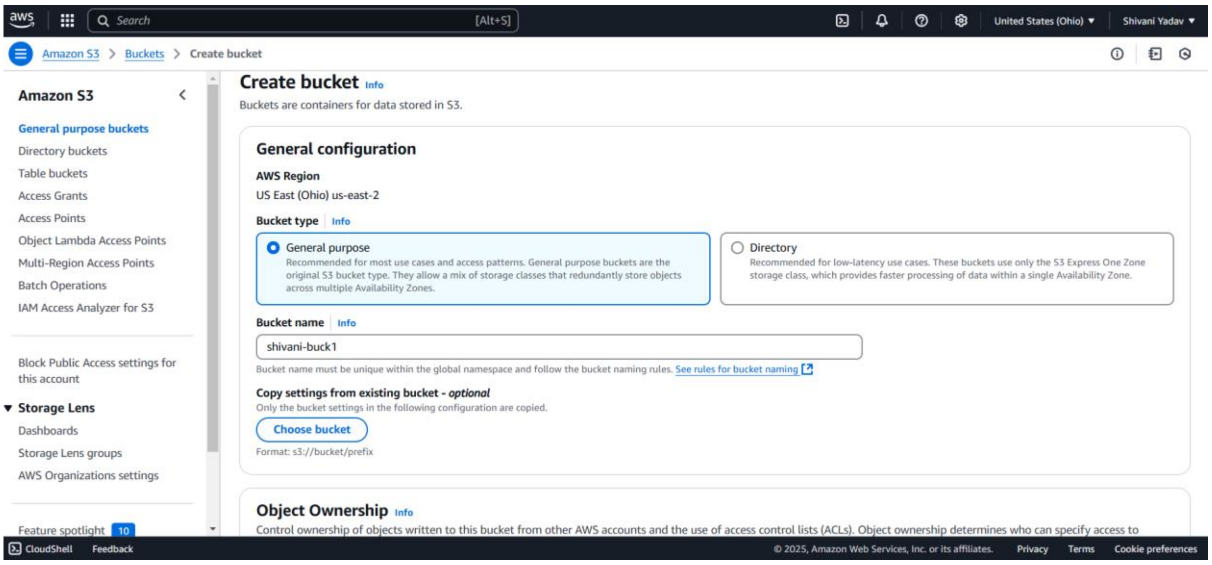
Certain permissions are necessary to access other services or perform actions in order to create this resource. For more information, see service role for Amazon Bedrock

Runtime role

[✓ Create and use a new service role]
Use an existing service role

Service role name

Step4: creating S3 bucket.



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region: US East (Ohio) us-east-2

Bucket type Info

[✓ General purpose] Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory: Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info: shivani-buck1

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

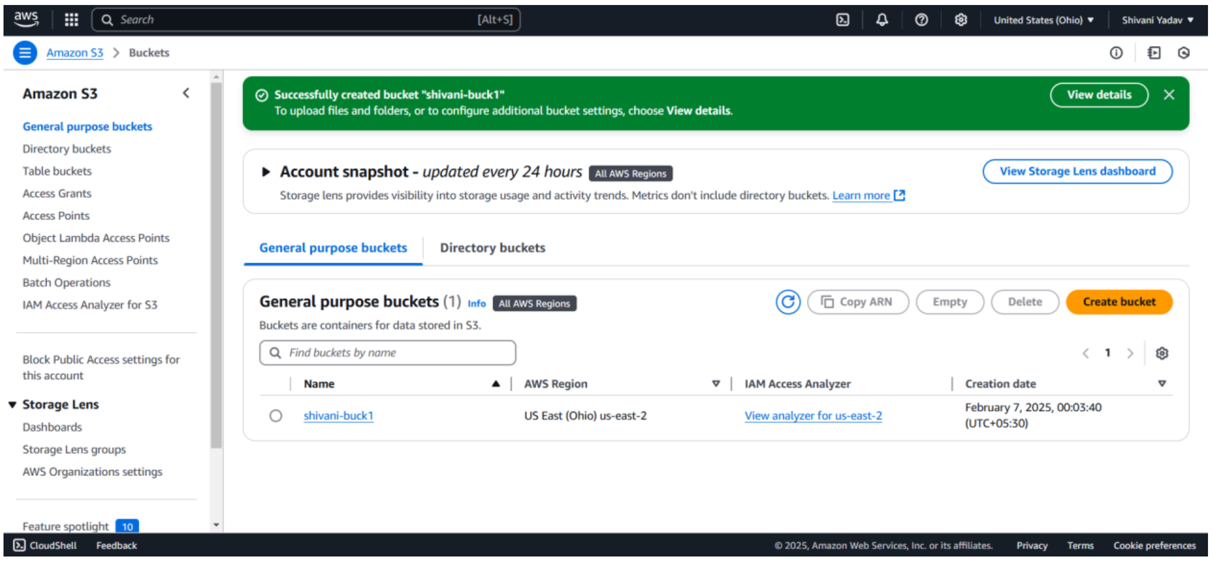
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to



Amazon S3 > Buckets

Successfully created bucket "shivani-buck1"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

View details

Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (1) Info All AWS Regions

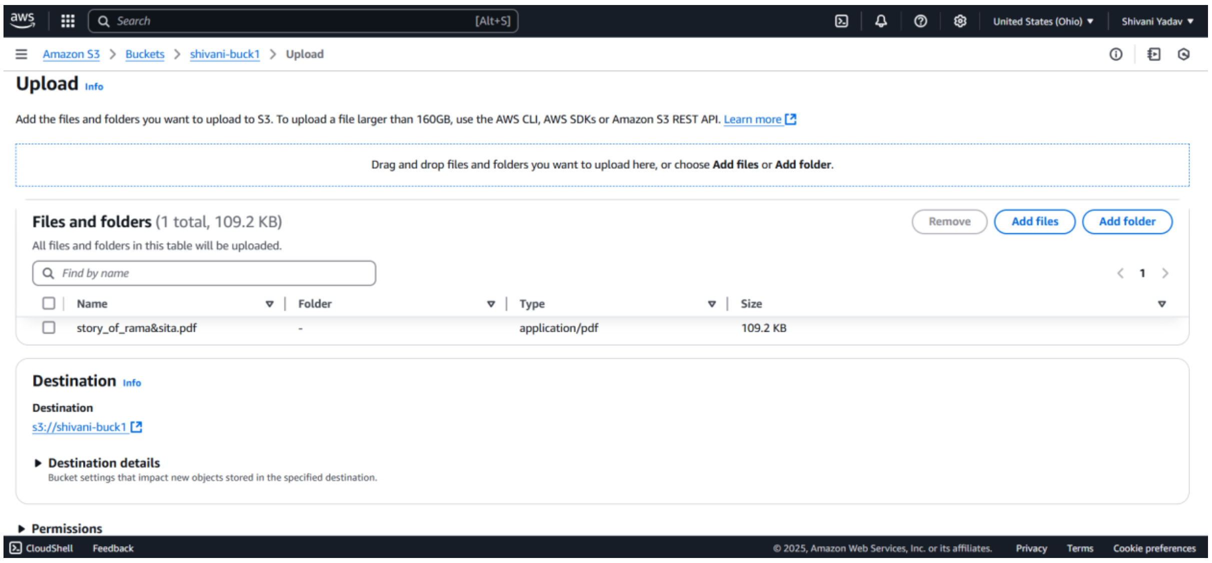
Copy ARN | Empty | Delete | Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
shivani-buck1	US East (Ohio) us-east-2	View analyzer for us-east-2	February 7, 2025, 00:03:40 (UTC+05:30)

Step5: upload the knowledge base pdf to S3 bucket.



Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 total, 109.2 KB)

All files and folders in this table will be uploaded.

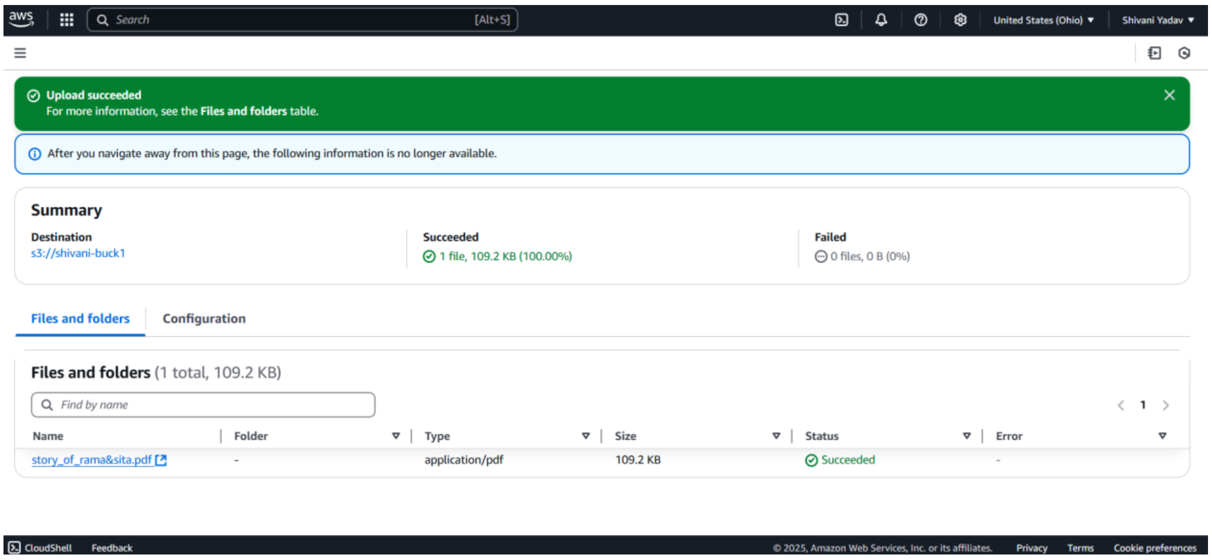
Name	Folder	Type	Size
story_of_rama&sita.pdf	-	application/pdf	109.2 KB

Destination Info

Destination
s3://shivani-buck1

▶ **Destination details**
Bucket settings that impact new objects stored in the specified destination.

▶ **Permissions**



Upload succeeded
For more information, see the **Files and folders** table.

After you navigate away from this page, the following information is no longer available.

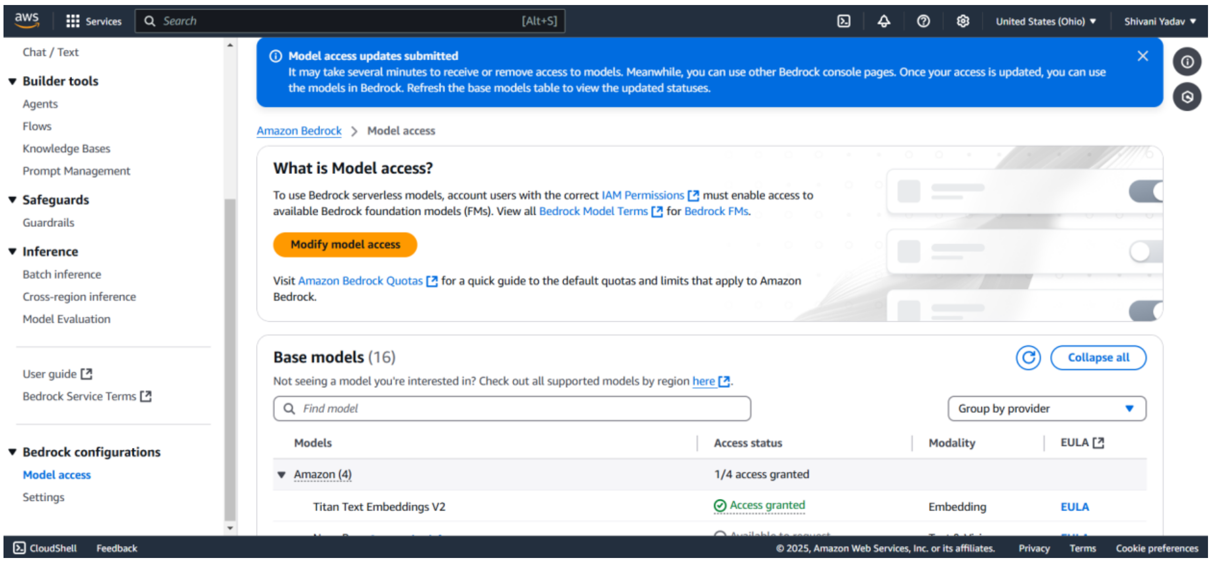
Summary

Destination	Succeeded	Failed
s3://shivani-buck1	1 file, 109.2 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 total, 109.2 KB)

Name	Folder	Type	Size	Status	Error
story_of_rama&sita.pdf	-	application/pdf	109.2 KB	Succeeded	-

Step6: Enabled Titan text Embeddings V2 in bedrock.



Model access updates submitted
It may take several minutes to receive or remove access to models. Meanwhile, you can use other Bedrock console pages. Once your access is updated, you can use the models in Bedrock. Refresh the base models table to view the updated statuses.

What is Model access?

To use Bedrock serverless models, account users with the correct IAM Permissions must enable access to available Bedrock foundation models (FMs). View all Bedrock Model Terms for Bedrock FMs.

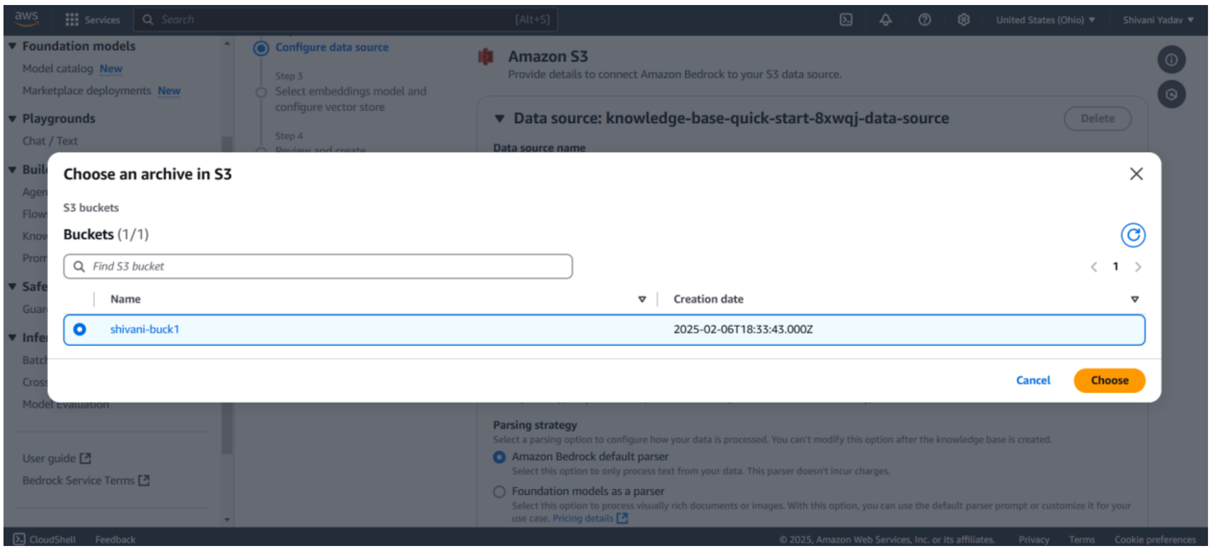
Visit Amazon Bedrock Quotas for a quick guide to the default quotas and limits that apply to Amazon Bedrock.

Base models (16)

Not seeing a model you're interested in? Check out all supported models by region here.

Models	Access status	Modality	EULA
▼ Amazon (4)	1/4 access granted		
Titan Text Embeddings V2	Access granted	Embedding	EULA

Step7: Choose the required S3 Bucket.



Choose an archive in S3

S3 buckets

Buckets (1/1)

Find S3 bucket

	Name	Creation date
●	shivani-buck1	2025-02-06T18:33:43.000Z

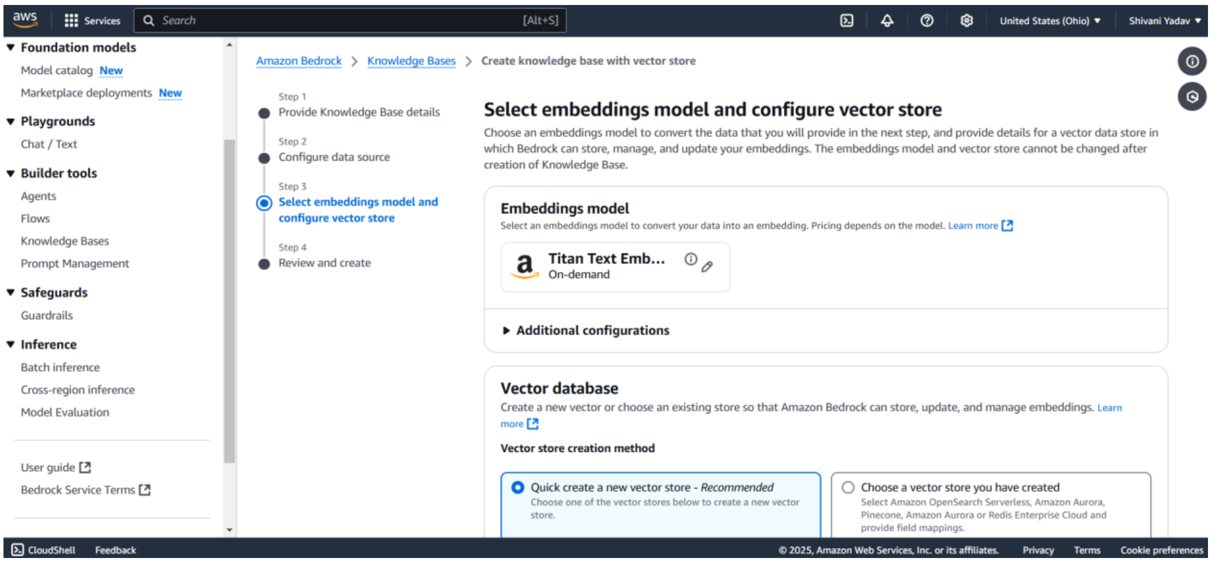
Cancel Choose

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

Data source: knowledge-base-quick-start-8xwqj-data-source

Parsing strategy
Select a parsing option to configure how your data is processed. You can't modify this option after the knowledge base is created.
Amazon Bedrock default parser
Select this option to only process text from your data. This parser doesn't incur charges.
Foundation models as a parser
Select this option to process visually rich documents or images. With this option, you can use the default parser prompt or customize it for your use case. Pricing details

Step8: Create a Vector database out of knowledge base.



Amazon Bedrock > Knowledge Bases > Create knowledge base with vector store

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

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 Emb...
On-demand

Additional configurations

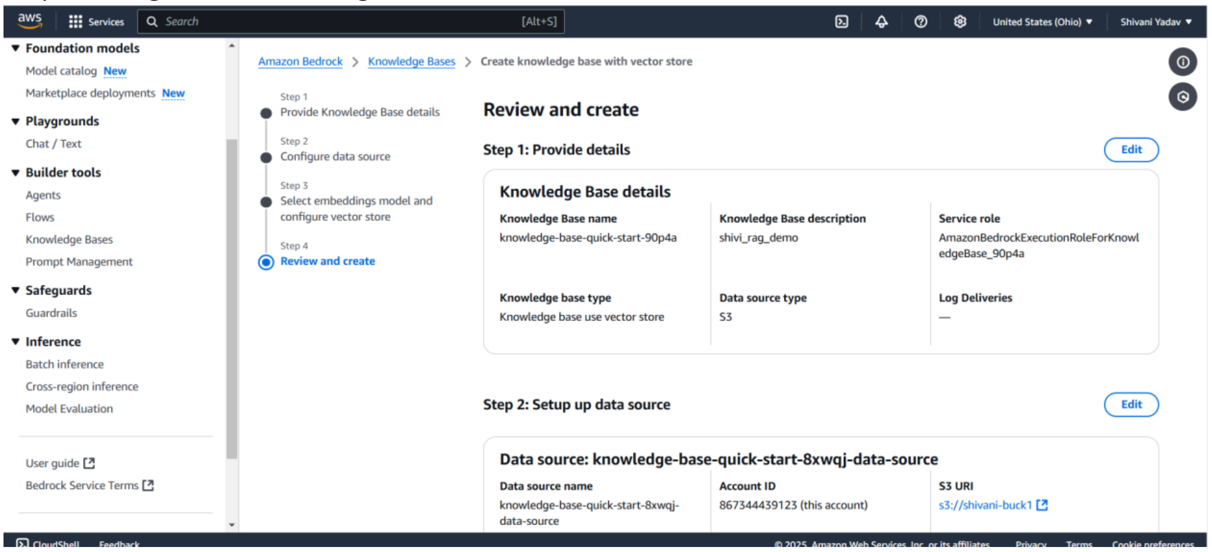
Vector database
Create a new vector or choose an existing store so that Amazon Bedrock can store, update, and manage embeddings. Learn more

Vector store creation method

● Quick create a new vector store - *Recommended*
Choose one of the vector stores below to create a new vector store.

○ Choose a vector store you have created
Select Amazon OpenSearch Serverless, Amazon Aurora, Pinecone, Amazon Aurora or Redis Enterprise Cloud and provide field mappings.

Step8: Configure the knowledge base.



Amazon Bedrock > Knowledge Bases > Create knowledge base with vector store

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

Review and create

Step 1: Provide details Edit

Knowledge Base details

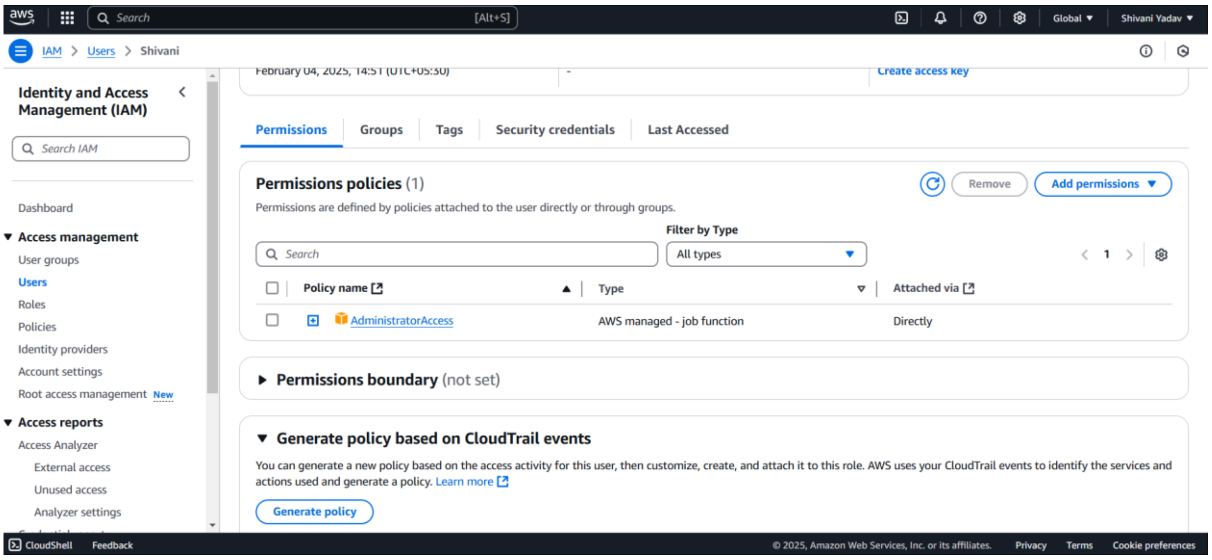
Knowledge Base name	Knowledge Base description	Service role
knowledge-base-quick-start-90p4a	shivi_rag_demo	AmazonBedrockExecutionRoleForKnowledgeBase_90p4a
Knowledge base type	**Data source type**	**Log Deliveries**
Knowledge base use vector store	S3	—

Step 2: Setup up data source Edit

Data source: knowledge-base-quick-start-8xwqj-data-source

Data source name	Account ID	S3 URI
knowledge-base-quick-start-8xwqj-data-source	867344439123 (this account)	s3://shivani-buck1

Step9: Add the required authorizations using IAM service.



IAM > Users > Shivani

February 04, 2025, 14:51 (UTC+05:30)

Permissions | Groups | Tags | Security credentials | Last Accessed

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

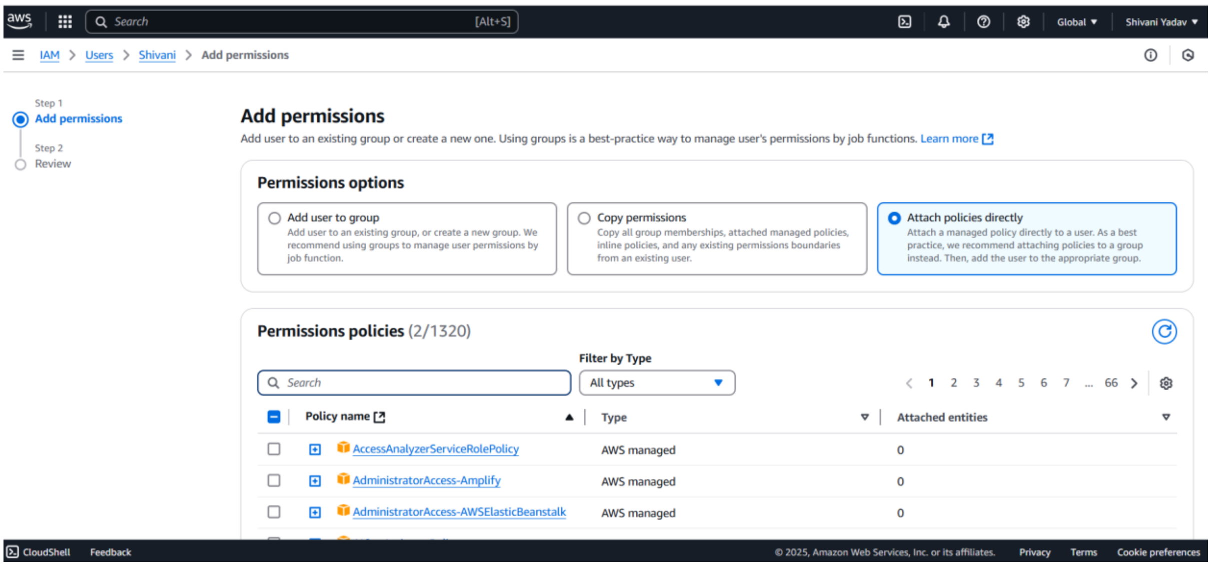
Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

Step10: Add the required permissions and policies.



IAM > Users > Shivani > Add permissions

Step 1 Add permissions

Step 2 Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job function. Learn more

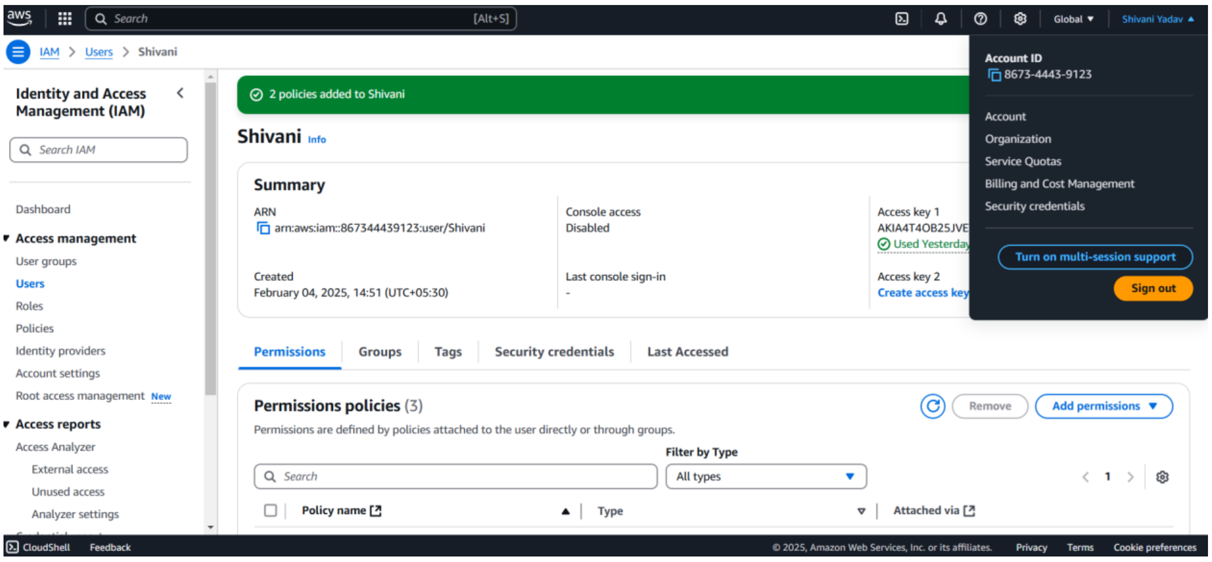
Permissions options

- Add user to group: Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions: Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- [✓ Attach policies directly]: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (2/1320)

Filter by Type: All types

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess-Amplify	AWS managed	0
AdministratorAccess-AWSElasticBeanstalk	AWS managed	0



IAM > Users > Shivani

2 policies added to Shivani

Shivani Info

Summary

ARN arn:aws:iam::867344439123:user/Shivani	Console access Disabled	Access key 1 AKIA4T4OB25JVE Used Yesterday
Created February 04, 2025, 14:51 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key

Account ID
8673-4443-9123

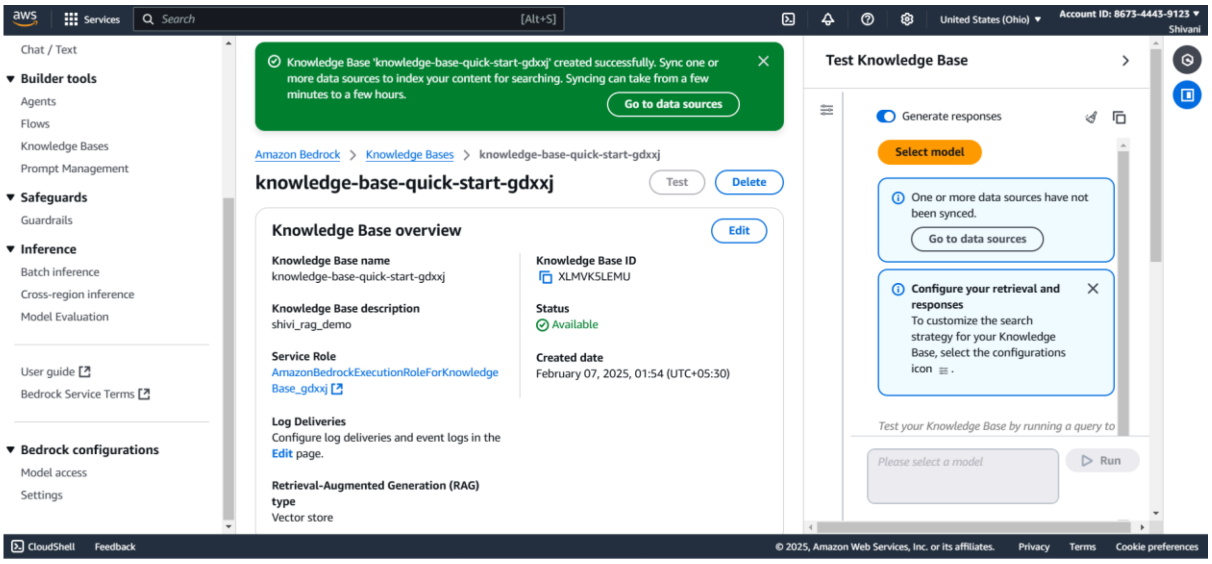
Permissions | Groups | Tags | Security credentials | Last Accessed

Permissions policies (3)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via



Knowledge Base 'knowledge-base-quick-start-gdxxj' created successfully. Sync one or more data sources to index your content for searching. Syncing can take from a few minutes to a few hours.

Go to data sources

Amazon Bedrock > Knowledge Bases > knowledge-base-quick-start-gdxxj

knowledge-base-quick-start-gdxxj

Test | Delete

Knowledge Base overview | Edit

Knowledge Base name
knowledge-base-quick-start-gdxxj

Knowledge Base description
shivi_rag_demo

Service Role
AmazonBedrockExecutionRoleForKnowledgeBase_gdxxj

Log Deliveries
Configure log deliveries and event logs in the Edit page.

Retrieval-Augmented Generation (RAG) type
Vector store

Knowledge Base ID
XLMVK5LEMU

Status
Available

Created date
February 07, 2025, 01:54 (UTC+05:30)

Test Knowledge Base

Generate responses

Select model

One or more data sources have not been synced.

Go to data sources

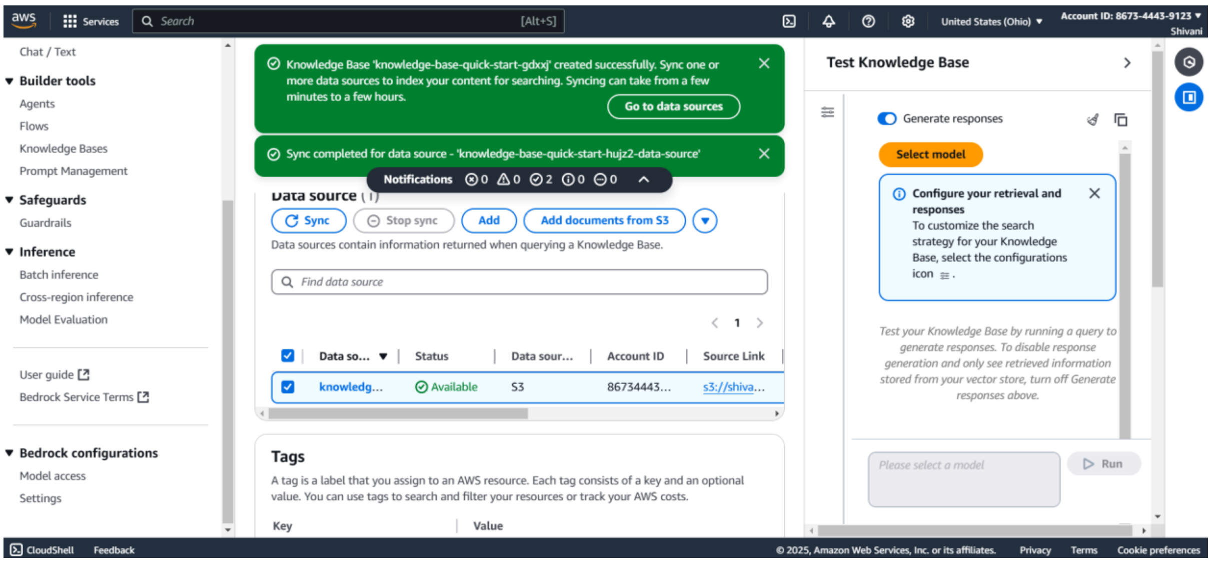
Configure your retrieval and responses
To customize the search strategy for your Knowledge Base, select the configurations icon ≡.

Test your Knowledge Base by running a query to

Please select a model

Run

Step11: Sync that data source to the index of content for searching.



Knowledge Base 'knowledge-base-quick-start-gdxxj' created successfully. Sync one or more data sources to index your content for searching. Syncing can take from a few minutes to a few hours.

Go to data sources

Sync completed for data source - 'knowledge-base-quick-start-hujz2-data-source'

Notifications ⊗0 ⚠0 ✓2 ⓘ0 ⊖0

Data source (1)

Sync | Stop sync | Add | Add documents from S3

Data sources contain information returned when querying a Knowledge Base.

Find data source

	Data so...	Status	Data sour...	Account ID	Source Link
✓	knowledg...	Available	S3	86734443...	s3://shiva...

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value

Test Knowledge Base

Generate responses

Select model

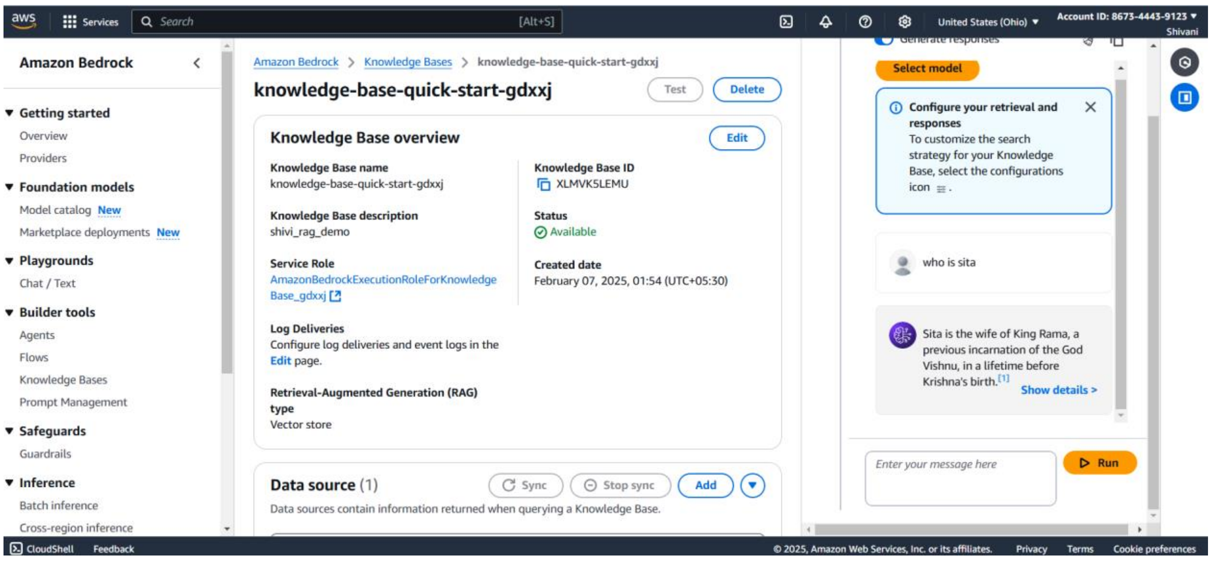
Configure your retrieval and responses
To customize the search strategy for your Knowledge Base, select the configurations icon ≡.

Test your Knowledge Base by running a query to generate responses. To disable response generation and only see retrieved information stored from your vector store, turn off Generate responses above.

Please select a model

Run

Step12: Quering the bedrock instance using web interface.



Amazon Bedrock > Knowledge Bases > knowledge-base-quick-start-gdxxj

knowledge-base-quick-start-gdxxj

Test | Delete

Knowledge Base overview | Edit

Knowledge Base name
knowledge-base-quick-start-gdxxj

Knowledge Base description
shivi_rag_demo

Service Role
AmazonBedrockExecutionRoleForKnowledgeBase_gdxxj

Log Deliveries
Configure log deliveries and event logs in the Edit page.

Retrieval-Augmented Generation (RAG) type
Vector store

Knowledge Base ID
XLMVK5LEMU

Status
Available

Created date
February 07, 2025, 01:54 (UTC+05:30)

Data source (1)

Sync | Stop sync | Add

Data sources contain information returned when querying a Knowledge Base.

Select model

Configure your retrieval and responses
To customize the search strategy for your Knowledge Base, select the configurations icon ≡.

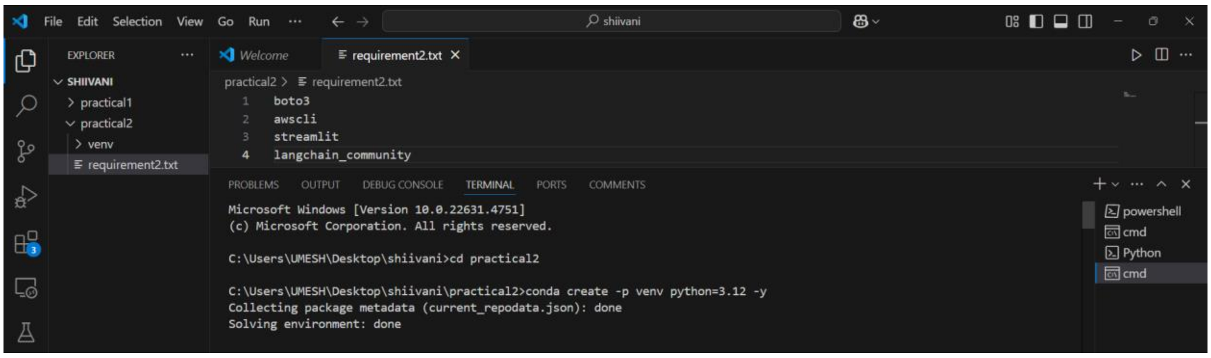
who is sita

Sita is the wife of King Rama, a previous incarnation of the God Vishnu, in a lifetime before Krishna's birth.[1] Show details >

Enter your message here

Run

Step13: Set the environment using conda and installing all the dependencies.

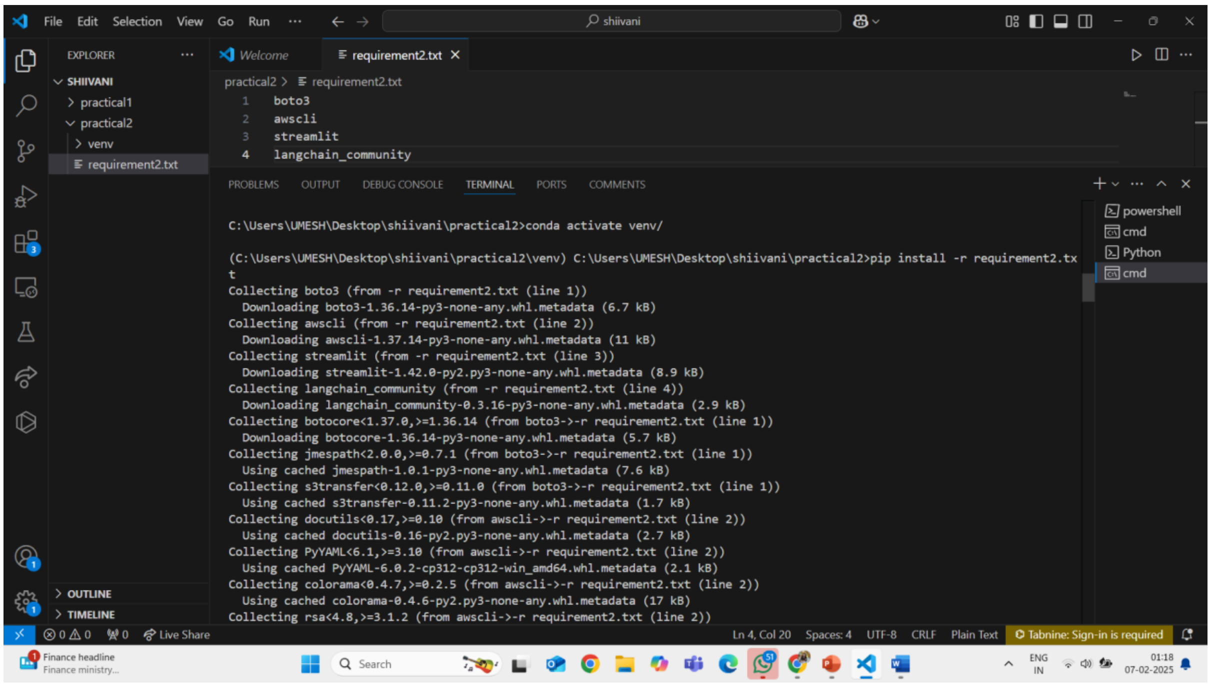


```
boto3
awscli
streamlit
langchain_community
```

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\UMESH\Desktop\shiivani>cd practical2

C:\Users\UMESH\Desktop\shiivani\practical2>conda create -p venv python=3.12 -y
Collecting package metadata (current_repodata.json): done
Solving environment: done
```



```
boto3
awscli
streamlit
langchain_community
```

```
C:\Users\UMESH\Desktop\shiivani\practical2>conda activate venv/

(C:\Users\UMESH\Desktop\shiivani\practical2\venv) C:\Users\UMESH\Desktop\shiivani\practical2>pip install -r requirement2.tx
t
Collecting boto3 (from -r requirement2.txt (line 1))
  Downloading boto3-1.36.14-py3-none-any.whl.metadata (6.7 kB)
Collecting awscli (from -r requirement2.txt (line 2))
  Downloading awscli-1.37.14-py3-none-any.whl.metadata (11 kB)
Collecting streamlit (from -r requirement2.txt (line 3))
  Downloading streamlit-1.42.0-py2.py3-none-any.whl.metadata (8.9 kB)
Collecting langchain_community (from -r requirement2.txt (line 4))
  Downloading langchain_community-0.3.16-py3-none-any.whl.metadata (2.9 kB)
Collecting botocore<1.37.0,>=1.36.14 (from boto3->-r requirement2.txt (line 1))
  Downloading botocore-1.36.14-py3-none-any.whl.metadata (5.7 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3->-r requirement2.txt (line 1))
  Using cached jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.12.0,>=0.11.0 (from boto3->-r requirement2.txt (line 1))
  Using cached s3transfer-0.11.2-py3-none-any.whl.metadata (1.7 kB)
Collecting docutils<0.17,>=0.10 (from awscli->-r requirement2.txt (line 2))
  Using cached docutils-0.16-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting PyYAML<6.1,>=3.10 (from awscli->-r requirement2.txt (line 2))
  Using cached PyYAML-6.0.2-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting colorama<0.4.7,>=0.2.5 (from awscli->-r requirement2.txt (line 2))
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting rsa<4.8,>=3.1.2 (from awscli->-r requirement2.txt (line 2))
```