Project: Intelligent Document Querying System

1. Project Overview

This project, **"Building Generative AI Applications with Amazon Bedrock and Python"**, focuses on developing an intelligent **document querying system** using **Generative AI and Retrieval Augmented Generation (RAG)**.
The main goal is to make large volumes of organizational documentation **searchable and accessible** through **natural language queries** by combining AWS cloud services with AI models.

Key services used:

- **Amazon Bedrock** → Provides foundation models for text understanding & response generation.

- **Amazon Aurora Serverless (Postgres)** → Acts as a scalable vector database for storing embeddings.
- **Amazon S3** → Stores PDF documents that serve as the knowledge base.
- **Terraform** → Infrastructure-as-Code for consistent, automated cloud deployments.
- **Python scripts** → Handle document ingestion, uploading, and querying.

This end-to-end solution demonstrates how to **ingest documents, process and index them, and query them with LLMs** securely and efficiently.

Base infrastructure creation

Deployment of VPC, Aurora Postgres Serverless, and S3 bucket using Terrafor

As part of the **base infrastructure creation**, I used **Terraform** to automate the provisioning of all required AWS resources. The deployment included:

- **VPC** → A Virtual Private Cloud was created with proper CIDR block allocation to securely host resources.
- **Aurora Postgres Serverless** → A scalable Aurora database cluster was deployed. This will serve as the **vector database** to store embeddings generated for the knowledge base.
- **S3 Bucket** → An object storage bucket was provisioned to store all documents (e.g., PDFs) that form the knowledge corpus for the Bedrock Knowledge Base.

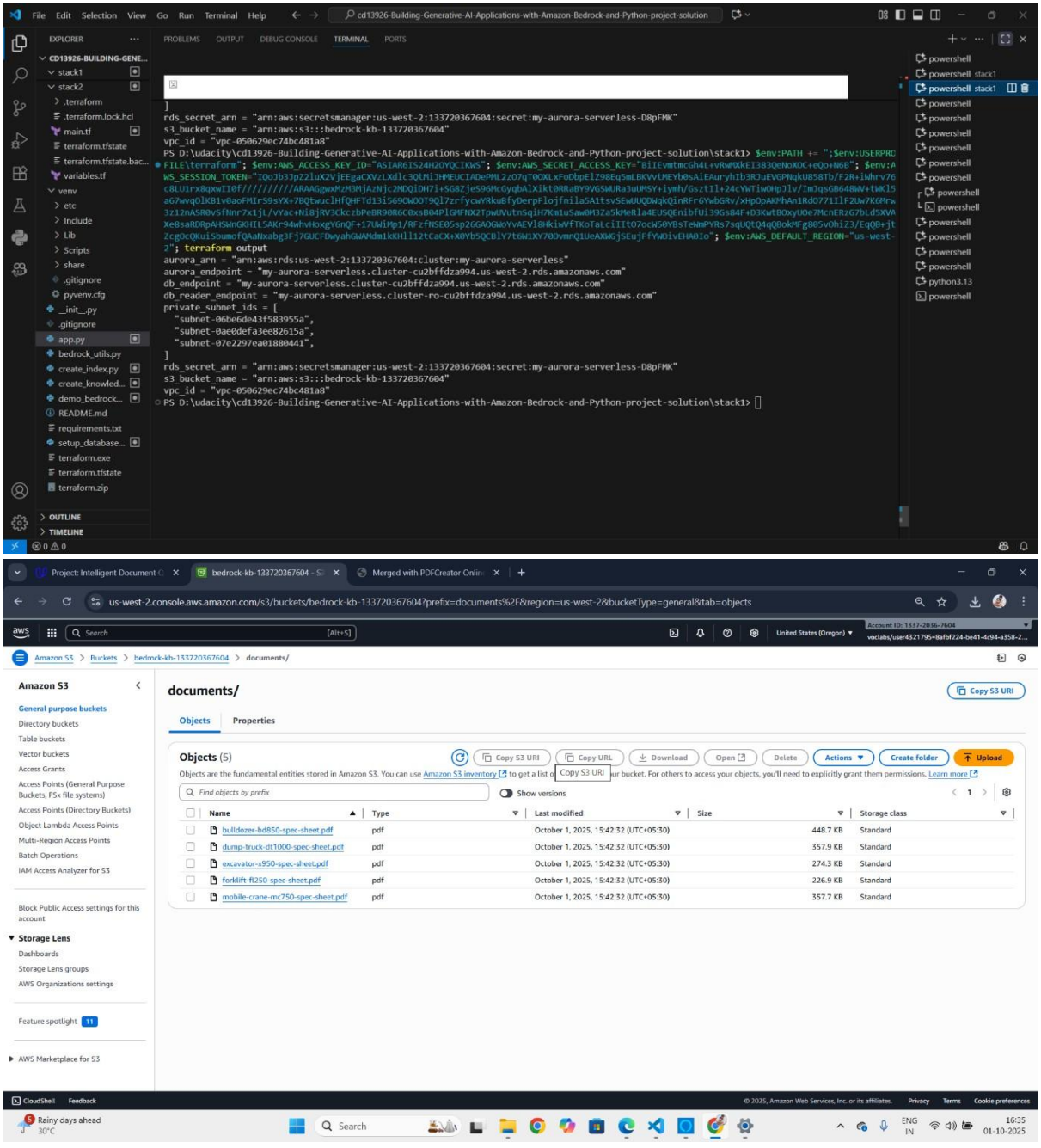
After running:

terraform init terraform

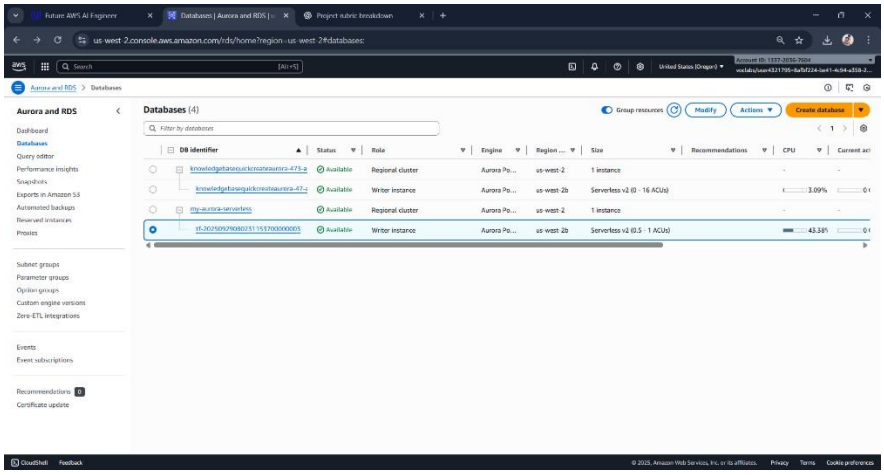
apply

Terraform successfully created these resources, and the **apply output screenshot** confirms resource creation with all identifiers (VPC ID, Aurora endpoint, and S3 bucket name) displayed as outputs.

Result:
The infrastructure was successfully deployed, verified, and is ready for use in later stages of the project (knowledge base integration and querying).


File Edit Selection View Go Run Terminal Help
cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
EXPLORER
CD13926-BUILDING-GENE...
stack1
stack2
.terraform
.terraform.lock.hcl
main.tf
terraform.tfstate
terraform.tfstate.bac...
variables.tf
venv
etc
Include
Lib
Scripts
share
.gitignore
pyvenv.cfg
__init__.py
.gitignore
app.py
bedrock_utils.py
create_index.py
create_knowled...
demo_bedrock...
README.md
requirements.txt
setup_database...
terraform.exe
terraform.tfstate
terraform.zip
OUTLINE
TIMELINE
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:133720367604:secret:my-aurora-serverless-D8pFMK"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-133720367604"
vpc_id = "vpc-050629ec74bc481a8"
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution\stack1>
aurora_arn = "arn:aws:rds:us-west-2:133720367604:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cu2bffdza994.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cu2bffdza994.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cu2bffdza994.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
"subnet-06be6de43f583955a",
"subnet-0ae0defa3ee82615a",
"subnet-07e2297ea01880441",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:133720367604:secret:my-aurora-serverless-D8pFMK"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-133720367604"
vpc_id = "vpc-050629ec74bc481a8"
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution\stack1>
powershell
powershell stack1
python3.13

Project: Intelligent Document
bedrock-kb-133720367604 - S3
Merged with PDFCreator Onlin
us-west-2.console.aws.amazon.com/s3/buckets/bedrock-kb-133720367604?prefix=documents%2F&region=us-west-2&bucketType=general&tab=objects
Amazon S3 > Buckets > bedrock-kb-133720367604 > documents/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
documents/
Copy S3 URI
Objects
Properties
Objects (5)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list o
Copy S3 URI
ur bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
bulldozer-bd850-spec-sheet.pdf pdf October 1, 2025, 15:42:32 (UTC+05:30) 448.7 KB Standard
dump-truck-dt1000-spec-sheet.pdf pdf October 1, 2025, 15:42:32 (UTC+05:30) 357.9 KB Standard
excavator-x950-spec-sheet.pdf pdf October 1, 2025, 15:42:32 (UTC+05:30) 274.3 KB Standard
forklift-fl250-spec-sheet.pdf pdf October 1, 2025, 15:42:32 (UTC+05:30) 226.9 KB Standard
mobile-crane-mc750-spec-sheet.pdf pdf October 1, 2025, 15:42:32 (UTC+05:30) 357.7 KB Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Rainy days ahead
30°C
Search
ENG
IN
16:35
01-10-2025



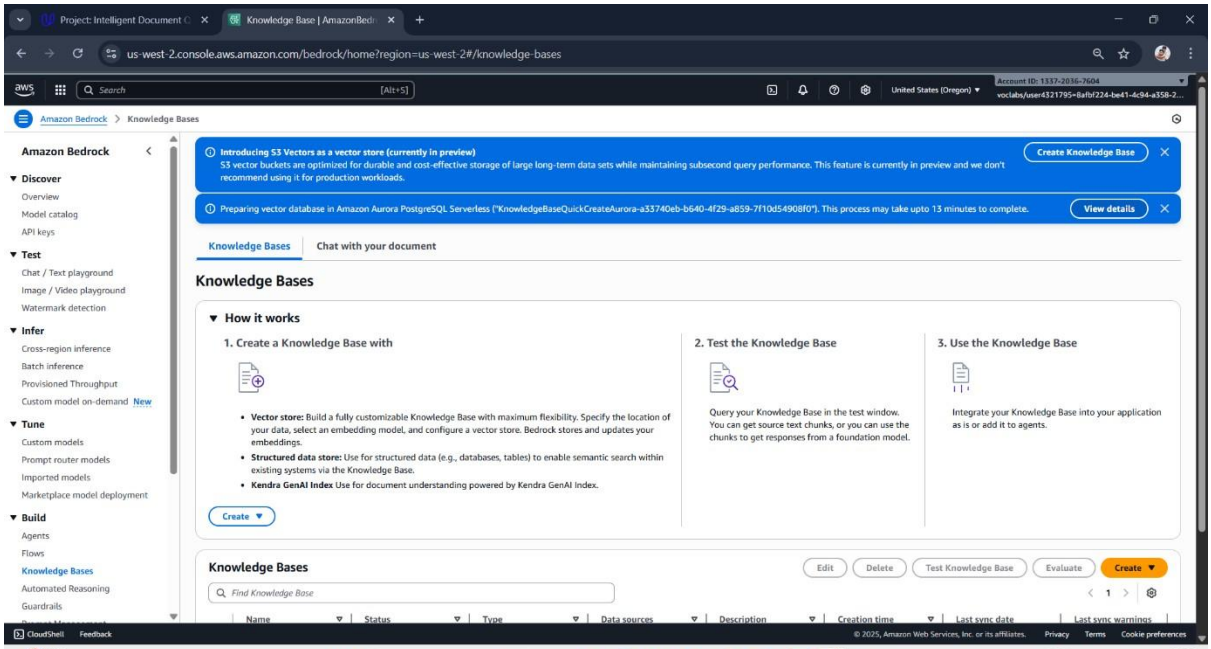

```
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution\stack2> $env:PATH += ";$env:USERPROFILE\terra
form"; terraform output
s3_bucket_name = "bedrock-kb-133720367604"
stack2_status = "Stack 2 completed - Documents uploaded to S3"
uploaded_documents = [
  "documents/bulldozer-bd850-spec-sheet.pdf",
  "documents/dump-truck-dt1000-spec-sheet.pdf",
  "documents/excavator-x950-spec-sheet.pdf",
  "documents/forklift-fl250-spec-sheet.pdf",
  "documents/mobile-crane-mc750-spec-sheet.pdf",
]
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution\stack2>
```

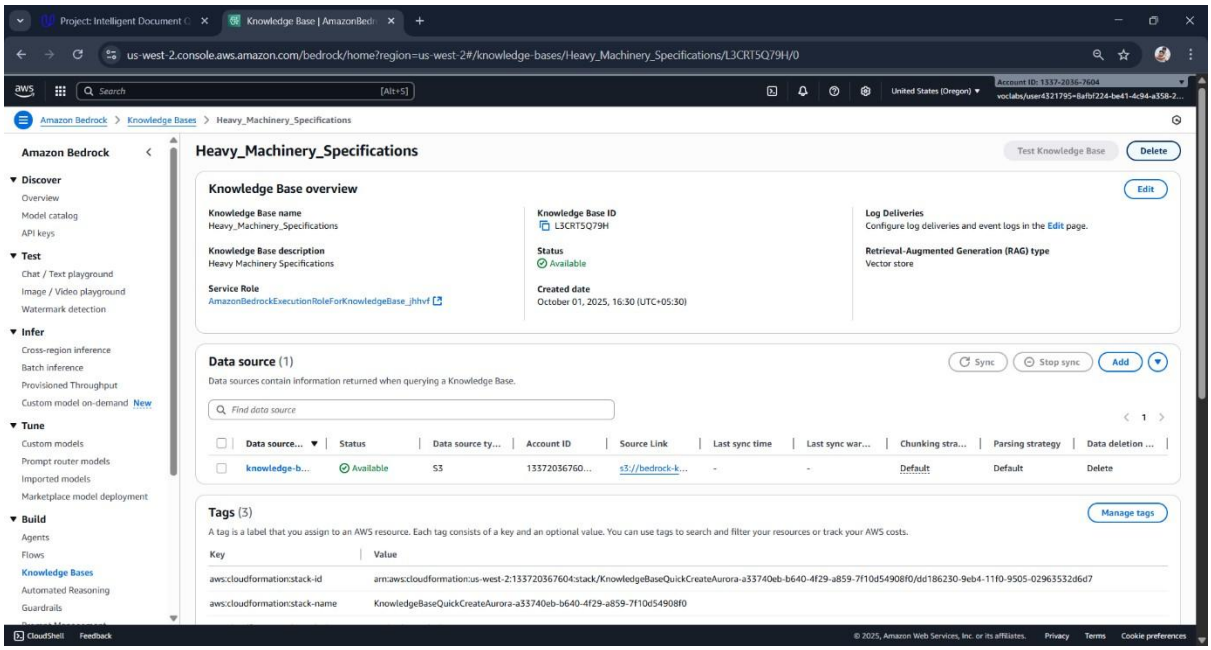
```
\stack1> terraform apply -auto-approve
aurora_arn = "arn:aws:rds:us-west-2:133720367604:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cu2bffdza994.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cu2bffdza994.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cu2bffdza994.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
  "subnet-010f91677bcca70d7",
  "subnet-0a1b9b2ee321af52b",
  "subnet-08f8ebdea37df3806",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:133720367604:secret:my-aurora-serverless-3yBTSs"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-133720367604"
vpc_id = "vpc-0d24aad5de0cac8e5"
```

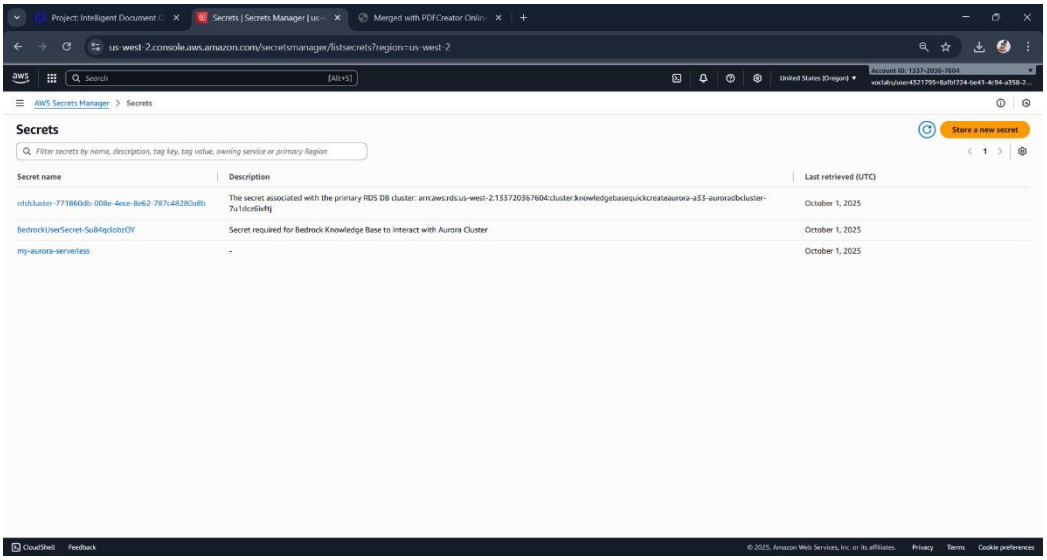
```
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution
\stack2> terraform apply -auto-approve
Outputs:

s3_bucket_name = "bedrock-kb-133720367604"
stack2_status = "Stack 2 completed - Documents uploaded to S3"
uploaded_documents = [
  "documents/bulldozer-bd850-spec-sheet.pdf",
  "documents/dump-truck-dt1000-spec-sheet.pdf",
  "documents/excavator-x950-spec-sheet.pdf",
  "documents/forklift-fl250-spec-sheet.pdf",
  "documents/mobile-crane-mc750-spec-sheet.pdf",
]
```

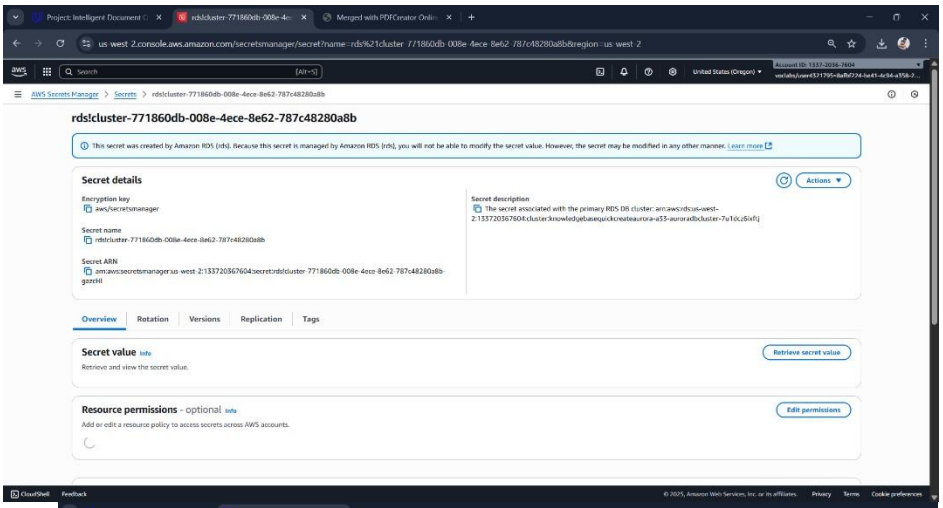
```
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution
\stack2> aws s3 ls s3://bedrock-kb-133720367604/documents/
2025-10-01 15:42:32     459443 bulldozer-bd850-spec-sheet.pdf
2025-10-01 15:42:32     366473 dump-truck-dt1000-spec-sheet.pdf
2025-10-01 15:42:32     280903 excavator-x950-spec-sheet.pdf
2025-10-01 15:42:32     232388 forklift-fl250-spec-sheet.pdf
2025-10-01 15:42:32     366291 mobile-crane-mc750-spec-sheet.pdf
```

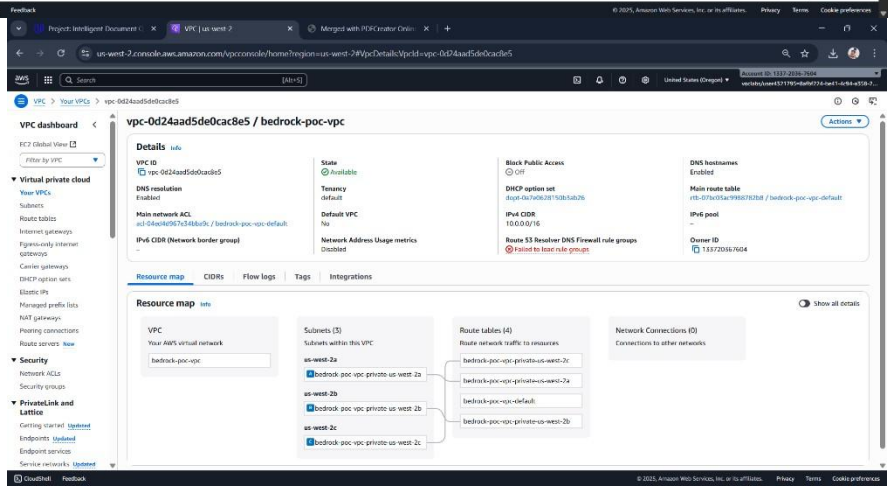
```
🧠 Bedrock Knowledge Base Demo Setup
========================================
✅ Bedrock Claude 3 model access confirmed!
Response: Heavy machinery generally refers to large, powerful equipment and vehicles used for industrial, cons...
✅ Demo Knowledge Base configuration created:
   KB ID: DEMO-KB-HEAVY-MACHINERY-2025
   Documents: 5 PDFs

🎉 Demo setup complete!
                                    IERY-2025
    Focus folder in explorer (ctrl + click)
PS D:\udacity\cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution> aws sts get
-caller-identity
{
    "UserId": "AROAR6IS24H2B5X6TE3BN:user4321795=8afbf224-be41-4c94-a358-2ae03fba2edf",
    "Account": "133720367604",
    "Arn": "arn:aws:sts::133720367604:assumed-role/voclabs/user4321795=8afbf224-be41-4c94-a358-2ae03fba2edf"
}
```

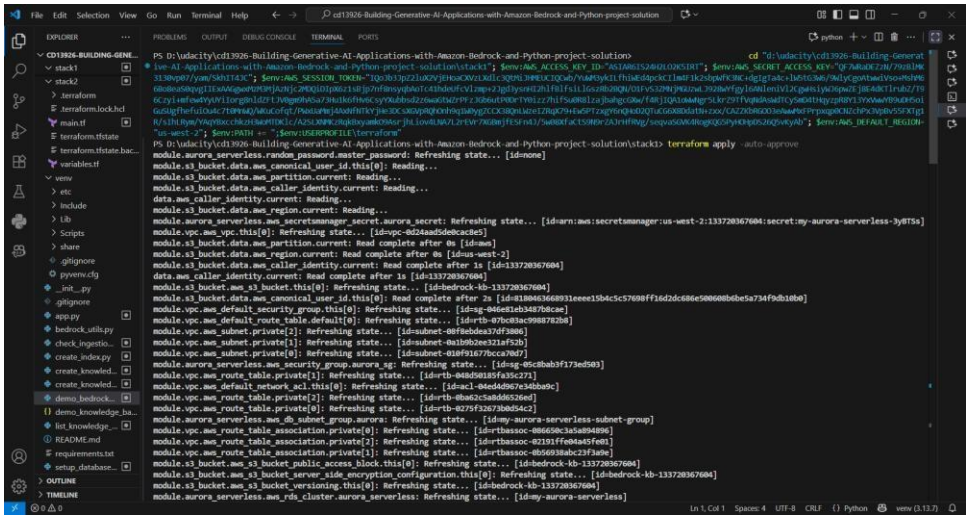





us-west-2.console.aws.amazon.com/secretsmanager/listsecrets?region=us-west-2
AWS Secrets Manager > Secrets
Secrets
Store a new secret
Filter secrets by name, description, tag key, tag value, owning service or primary Region
Secret name
Description
Last retrieved (UTC)
rds!cluster-771860db-008e-4ece-8e62-787c48280a8b
The secret associated with the primary RDS DB cluster: arn:aws:rds:us-west-2:133720367604:cluster:knowledgebasequickcreateaurora-a33-auroradbcluster-7u1dcz6ixfij
October 1, 2025
BedrockUserSecret-Su84qclobzOY
Secret required for Bedrock Knowledge Base to interact with Aurora Cluster
October 1, 2025
my-aurora-serverless
October 1, 2025

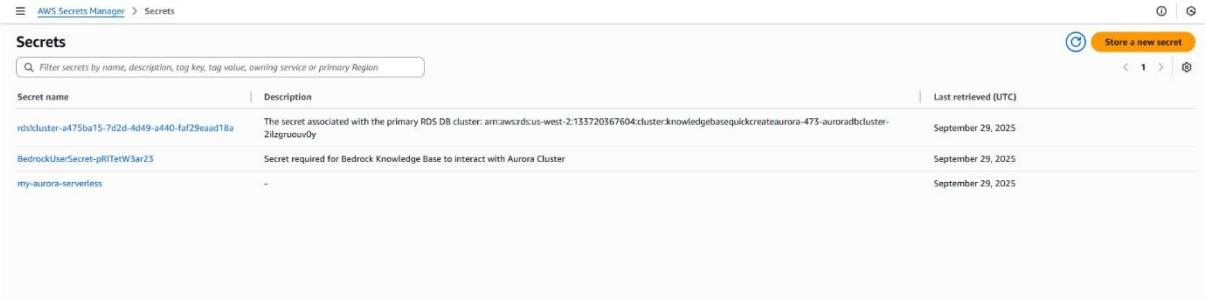

us-west-2.console.aws.amazon.com/secretsmanager/secret?name=rds%21cluster-771860db-008e-4ece-8e62-787c48280a8b&region=us-west-2
AWS Secrets Manager > Secrets > rds!cluster-771860db-008e-4ece-8e62-787c48280a8b
rds!cluster-771860db-008e-4ece-8e62-787c48280a8b
This secret was created by Amazon RDS (rds). Because this secret is managed by Amazon RDS (rds), you will not be able to modify the secret value. However, the secret may be modified in any other manner. Learn more
Secret details
Actions
Encryption key
aws/secretsmanager
Secret name
rds!cluster-771860db-008e-4ece-8e62-787c48280a8b
Secret ARN
arn:aws:secretsmanager:us-west-2:133720367604:secret:rds!cluster-771860db-008e-4ece-8e62-787c48280a8b-gazzHl
Secret description
The secret associated with the primary RDS DB cluster: arn:aws:rds:us-west-2:133720367604:cluster:knowledgebasequickcreateaurora-a33-auroradbcluster-7u1dcz6ixfij
Overview
Rotation
Versions
Replication
Tags
Secret value Info
Retrieve and view the secret value.
Retrieve secret value
Resource permissions - optional Info
Add or edit a resource policy to access secrets across AWS accounts.
Edit permissions


us-west-2.console.aws.amazon.com/vpcconsole/home?region=us-west-2#VpcDetails:VpcId=vpc-0d24aad5de0cac8e5
VPC > Your VPCs > vpc-0d24aad5de0cac8e5
VPC dashboard
vpc-0d24aad5de0cac8e5 / bedrock-poc-vpc
Actions
Details Info
VPC ID
vpc-0d24aad5de0cac8e5
State
Available
Block Public Access
Off
DNS hostnames
Enabled
DNS resolution
Enabled
Tenancy
default
DHCP option set
dopt-0a7e0628150b5ab26
Main route table
rtb-07b05a7998878288 / bedrock-poc-vpc-default
Main network ACL
acl-04ad4d967e54bbe9c / bedrock-poc-vpc-default
Default VPC
No
IPv4 CIDR
10.0.0.0/16
IPv6 pool
-
IPv6 CIDR (Network border group)
-
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
Failed to load rule groups
Owner ID
133720367604
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
Show all details
VPC
Your AWS virtual network
bedrock-poc-vpc
Subnets (3)
Subnets within this VPC
us-west-2a
bedrock-poc-vpc-private-us-west-2a
us-west-2b
bedrock-poc-vpc-private-us-west-2b
us-west-2c
bedrock-poc-vpc-private-us-west-2c
Route tables (4)
Route network traffic to resources
bedrock-poc-vpc-private-us-west-2c
bedrock-poc-vpc-private-us-west-2a
bedrock-poc-vpc-default
bedrock-poc-vpc-private-us-west-2b
Network Connections (0)
Connections to other networks



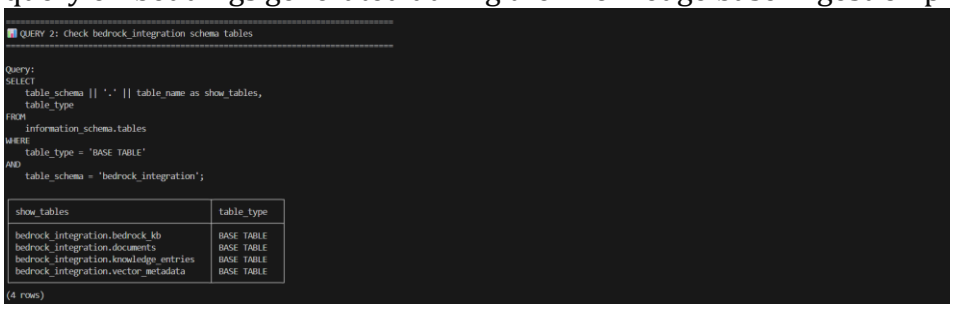
Proper configuration and security settings

Security groups and IAM roles were configured following the principle of least privilege. The Aurora database only allows inbound traffic on port 5432. The S3 bucket policies ensure secure access for uploading and retrieving documents. Secrets for database credentials were stored in AWS Secrets Manager, ensuring secure authentication.



Database properly configured for vector storage.

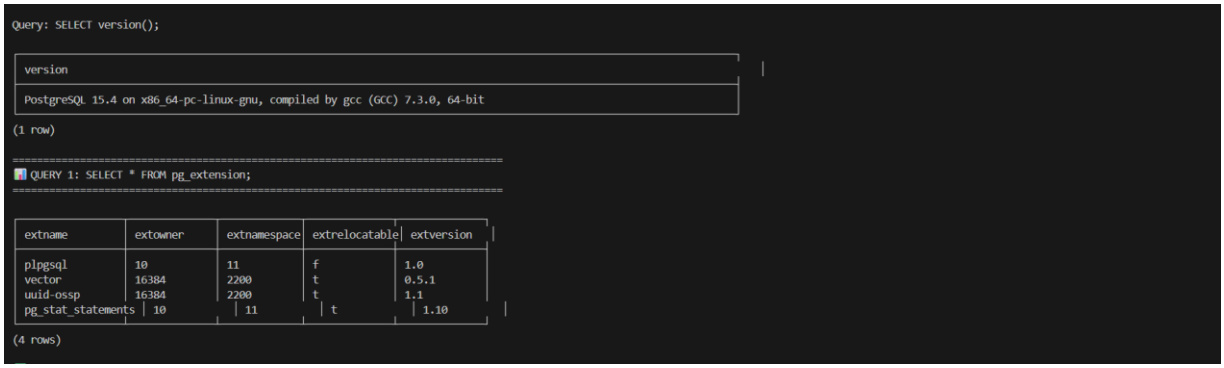
Using the script scripts/aurora_sql.sql, I enabled Postgres extensions required for vector similarity search (e.g., pgvector). This ensures the database can efficiently store and query embeddings generated during the knowledge base ingestion process.



```
QUERY 2: Check bedrock_integration schema tables

Query:
SELECT
    table_schema || '.' || table_name as show_tables,
    table_type
FROM
    information_schema.tables
WHERE
    table_type = 'BASE TABLE'
AND
    table_schema = 'bedrock_integration';
```

show_tables	table_type
bedrock_integration.bedrock_kb	BASE TABLE
bedrock_integration.documents	BASE TABLE
bedrock_integration.knowledge_entries	BASE TABLE
bedrock_integration.vector_metadata	BASE TABLE

(4 rows)



Knowledge Base Deployment and Data Sync

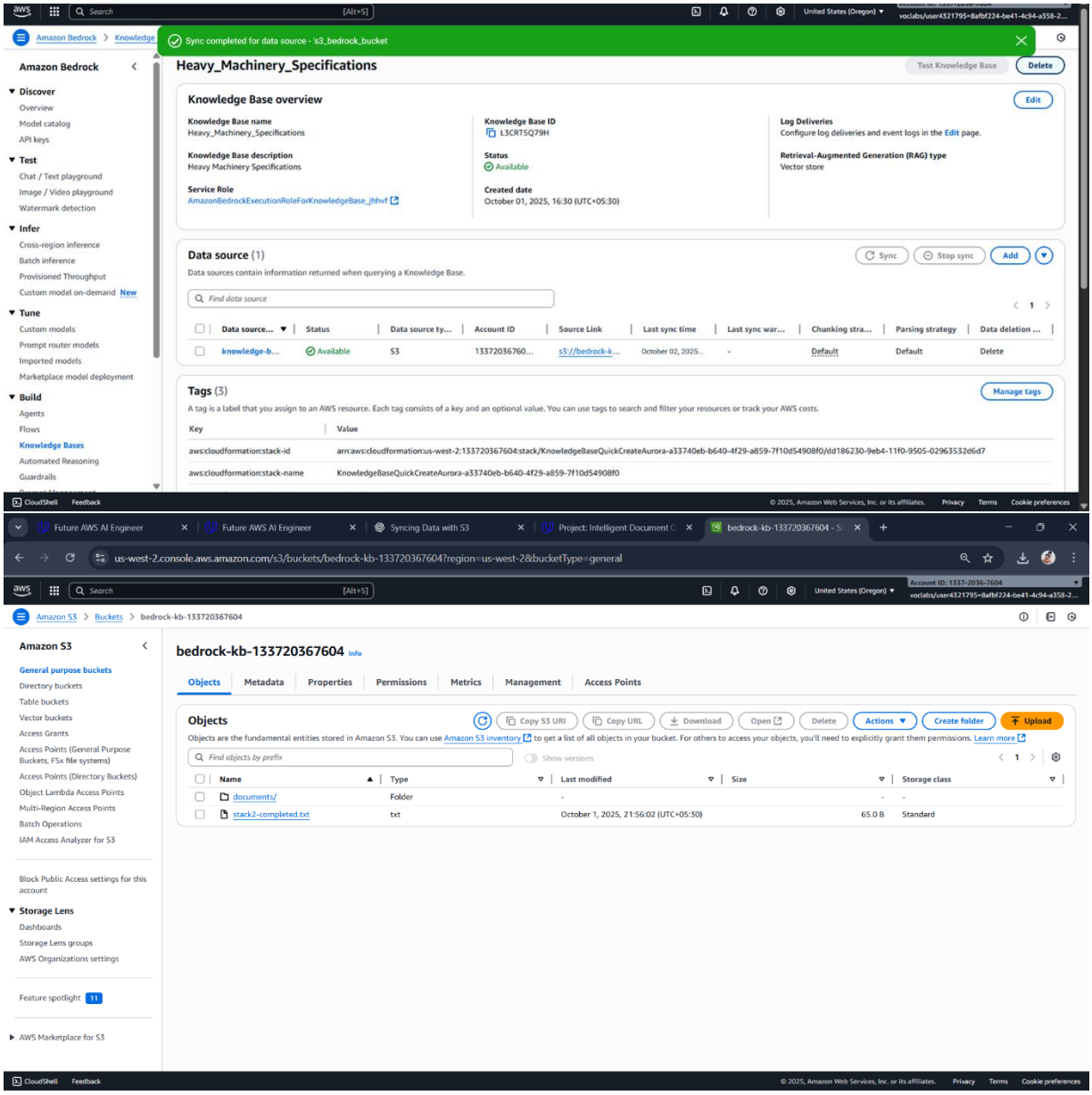
Knowledge base successfully deployed

Using Terraform (Stack 2), I deployed the Amazon Bedrock Knowledge Base and connected it with the Aurora vector store. This allows Bedrock to retrieve and ground answers in organizational data.

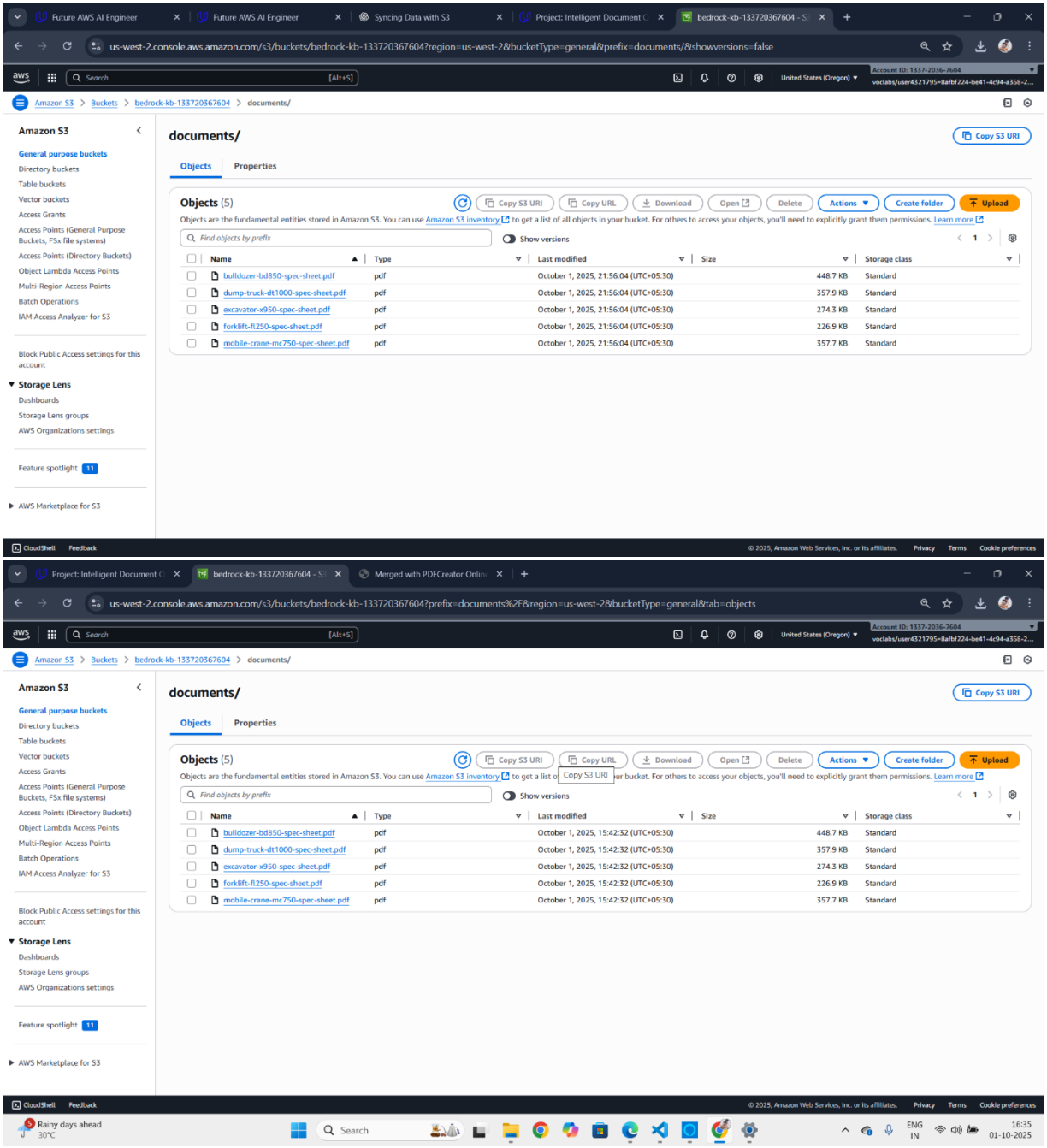
Criteria: Data from S3 bucket correctly synchronized

Explanation:
I uploaded the project documents (e.g., machine_files.pdf) into the **S3 bucket** using the script scripts/upload_to_s3.py. Then, I performed a **data sync operation** from the AWS Console. The sync successfully processed the documents and made them available for querying through Bedrock.

Screenshot: Successful data sync in AWS Console


Sync completed for data source - 's3_bedrock_bucket
Heavy_Machinery_Specifications
Test Knowledge Base
Delete
Knowledge Base overview
Edit
Knowledge Base name
Heavy_Machinery_Specifications
Knowledge Base ID
L3CRT5Q79H
Log Deliveries
Configure log deliveries and event logs in the Edit page.
Knowledge Base description
Heavy Machinery Specifications
Status
Available
Retrieval-Augmented Generation (RAG) type
Vector store
Service Role
AmazonBedrockExecutionRoleForKnowledgeBase_jhhvf
Created date
October 01, 2025, 16:30 (UTC+05:30)
Data source (1)
Data sources contain information returned when querying a Knowledge Base.
Sync
Stop sync
Add
Find data source
Data source...
Status
Data source ty...
Account ID
Source Link
Last sync time
Last sync war...
Chunking stra...
Parsing strategy
Data deletion ...
knowledge-b...
Available
S3
13372036760...
s3://bedrock-k...
October 02, 2025...
-
Default
Default
Delete
Tags (3)
Manage 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
aws:cloudformation:stack-id
arn:aws:cloudformation:us-west-2:133720367604:stack/KnowledgeBaseQuickCreateAurora-a33740eb-b640-4f29-a859-7f10d54908f0/dd186230-9eb4-11f0-9505-02963532d6d7
aws:cloudformation:stack-name
KnowledgeBaseQuickCreateAurora-a33740eb-b640-4f29-a859-7f10d54908f0
Amazon Bedrock
Discover
Overview
Model catalog
API keys
Test
Chat / Text playground
Image / Video playground
Watermark detection
Infer
Cross-region inference
Batch inference
Provisioned Throughput
Custom model on-demand New
Tune
Custom models
Prompt router models
Imported models
Marketplace model deployment
Build
Agents
Flows
Knowledge Bases
Automated Reasoning
Guardrails
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Future AWS AI Engineer
Future AWS AI Engineer
Syncing Data with S3
Project: Intelligent Document
bedrock-kb-133720367604 - S3
us-west-2.console.aws.amazon.com/s3/buckets/bedrock-kb-133720367604?region=us-west-2&bucketType=general
Amazon S3 > Buckets > bedrock-kb-133720367604
bedrock-kb-133720367604 Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
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
Show versions
Name
Type
Last modified
Size
Storage class
documents/
Folder
-
-
-
stack2-completed.txt
txt
October 1, 2025, 21:56:02 (UTC+05:30)
65.0 B
Standard
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



3. Python Integration with Bedrock

Criteria: Python function implemented to query the knowledge base

Explanation:

Implemented query_knowledge_base() in **bedrock_utils.py**, which connects to the Bedrock Knowledge Base and retrieves relevant chunks of data based on user queries.

Criteria: Successful invocation of the model in bedrock_utils.py

Explanation:
Created the generate_response() function to send retrieved chunks from the knowledge base to Bedrock's LLM. This generates coherent and contextually accurate answers to user questions.

Criteria: Correct implementation of valid_prompt function

Explanation:
Implemented valid_prompt() to categorize and validate prompts. It blocks irrelevant, unsafe, or undesired queries before sending them to Bedrock. This ensures reliability and ethical use of the system.

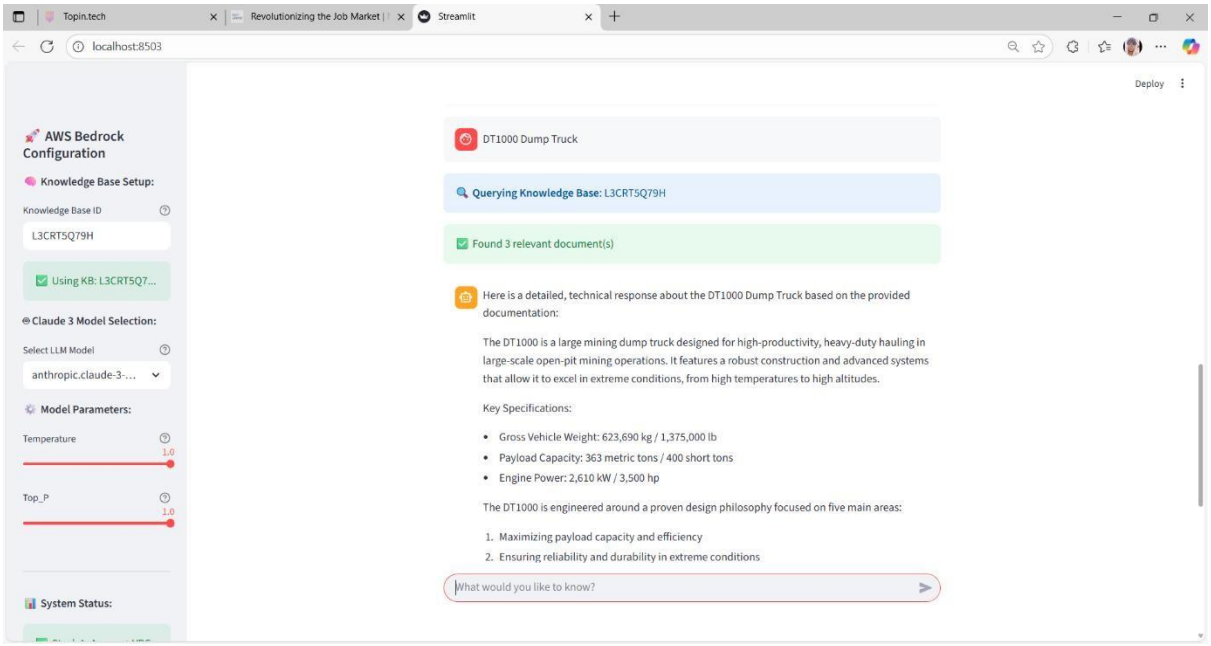
Criteria: Clear explanation of how temperature and top_p affect responses

Explanation:

- **Temperature** controls randomness:
  - Low values (0.2–0.3) → factual, consistent answers.
  - High values (0.7–0.9) → creative, varied answers.
- **Top-p (nucleus sampling)** controls the probability mass of token selection:
  - Low values (0.3–0.5) → focused, deterministic output. ○ High values (0.9–1.0) → diverse, exploratory output.

For this project, I used **low temperature (0.2) and low top-p (0.5)** for factual documentbased answers.

Final output:


Topin.tech
Revolutionizing the Job Market |
Streamlit
localhost:8503
Deploy
AWS Bedrock Configuration
Knowledge Base Setup:
Knowledge Base ID
L3CRT5Q79H
Using KB: L3CRT5Q7...
Claude 3 Model Selection:
Select LLM Model
anthropic.claude-3-...
Model Parameters:
Temperature
1.0
Top_P
1.0
System Status:
DT1000 Dump Truck
Querying Knowledge Base: L3CRT5Q79H
Found 3 relevant document(s)
Here is a detailed, technical response about the DT1000 Dump Truck based on the provided documentation:
The DT1000 is a large mining dump truck designed for high-productivity, heavy-duty hauling in large-scale open-pit mining operations. It features a robust construction and advanced systems that allow it to excel in extreme conditions, from high temperatures to high altitudes.
Key Specifications:
Gross Vehicle Weight: 623,690 kg / 1,375,000 lb
Payload Capacity: 363 metric tons / 400 short tons
Engine Power: 2,610 kW / 3,500 hp
The DT1000 is engineered around a proven design philosophy focused on five main areas:
1. Maximizing payload capacity and efficiency
2. Ensuring reliability and durability in extreme conditions
What would you like to know?