

AWS (The project had to be done in multiple phases.)

Final Attempt

Temperature and top-p control the language model text generation. Temperature decides how random the words will be. A low temperature will make the output predictable and repetitive. A high temperature means a creative and varied output. Top-p, decides the possible words the model will consider. A low top-p means the code chooses only the most likely words. Using this we can configure balance between safe and focused answers.

Image 1

```
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1# terraform init

Initializing the backend...
Initializing modules...
- aurora_serverless in ../../modules/database
  Downloading registry.terraform.io/terraform-aws-modules/s3-bucket/aws 3.15.2 for s3_bucket...
- s3_bucket in .terraform/modules/s3_bucket
  Downloading registry.terraform.io/terraform-aws-modules/vpc/aws 5.21.0 for vpc...
- vpc in .terraform/modules/vpc

Initializing provider plugins...
- Finding latest version of hashicorp/random...
  provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
```

Image 2

```
- aurora_serverless in ../../modules/database
  Downloading registry.terraform.io/terraform-aws-modules/s3-bucket/aws 3.15.2 for s3_bucket...
- s3_bucket in .terraform/modules/s3_bucket
  Downloading registry.terraform.io/terraform-aws-modules/vpc/aws 5.21.0 for vpc...
- vpc in .terraform/modules/vpc

Initializing provider plugins...
- Finding latest version of hashicorp/random...
  provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1#
```

Terraform init (Image 1 and 2)

```
aurora_endpoint = "my-aurora-serverless.cluster-cfpjqajvwgv.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cfpjqajvwgv.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cfpjqajvwgv.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
    "subnet-0b0dd2b5fdb9cbb1a",
    "subnet-042cc3be46a8597ae",
    "subnet-08b24cc8f4f58c6af",
]
public_subnet_ids = [
    "subnet-0fdc16ee5551616bb",
    "subnet-0bd4cd36603b97621",
    "subnet-06edad9cc4bea7f7d",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-rzyen9"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-245403145430"
vpc_id = "vpc-0fd8ca44b1c419219"
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1#
```

Image3: terraform apply

The screenshot shows a code editor interface with two tabs: 'CD13926-BUI...' and 'stack2'. The 'stack2' tab is active and displays the contents of 'main.tf'. The code in 'main.tf' is as follows:

```
1 provider "aws" {
2 }
3
4
5 module "bedrock_kb" {
6   source = "../modules/bedrock_kb"
7
8   knowledge_base_name      = "my-bedrock-kb"
9   knowledge_base_description = "Knowledge base connected to Aurora Serverless"
10
11   aurora_arn                = "arn:aws:rds:us-west-2:2454:cluster:myapp"
12   aurora_db_name              = "myapp"
13   aurora_endpoint             = "my-aurora-serverless.cluster-w2qjwv3gk33c.aurora-serverless.amazonaws.com"
14   aurora_table_name           = "bedrock_integration.bedrock_kb"
15   aurora_primary_key_field    = "id"
16   aurora_metadata_field       = "metadata"
17   aurora_text_field            = "chunks"
18   aurora_verctor_field        = "embedding"
19   aurora_username              = "dbadmin"
20   aurora_secret_arn            = "arn:aws:secretsmanager:us-west-2:2454:secret:myappSecret"
21   s3_bucket_arn                = "arn:aws:s3:::bedrock-kb-2454"
22
23 }
```

Image 4: Updated information

The screenshot shows the AWS S3 console. On the left, a sidebar menu includes sections for Buckets, Access management and security, and Storage management and insights. The main area displays a list of General purpose buckets under the heading "General purpose buckets (1)". A table lists the single bucket, showing its name, AWS Region (US West (Oregon)), and creation date (November 27, 2025). Buttons for Copy ARN, Empty, and Delete are available, along with a "Create bucket" button.

Name	AWS Region	Creation date
bedrock-kb-245403145430	US West (Oregon) us-west-2	November 27, 2025, 15:05:08 (UTC-08:00)

Image 5: S3 created

The screenshot shows the AWS VPC Subnets dashboard. The sidebar menu includes sections for Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, and Edge locations. The main area displays a list of Subnets under the heading "Subnets (2)". A table lists the two subnets, showing their Subnet ID, State (Available), and associated VPC (vpc-070e33645cdc). Buttons for Actions and Create subnet are available.

Subnet ID	State	VPC
subnet-03679d97b7d8f2f88	Available	vpc-070e33645cdc
subnet-0db17e3b1df957c07	Available	vpc-070e33645cdc

Image 6: Subnets created

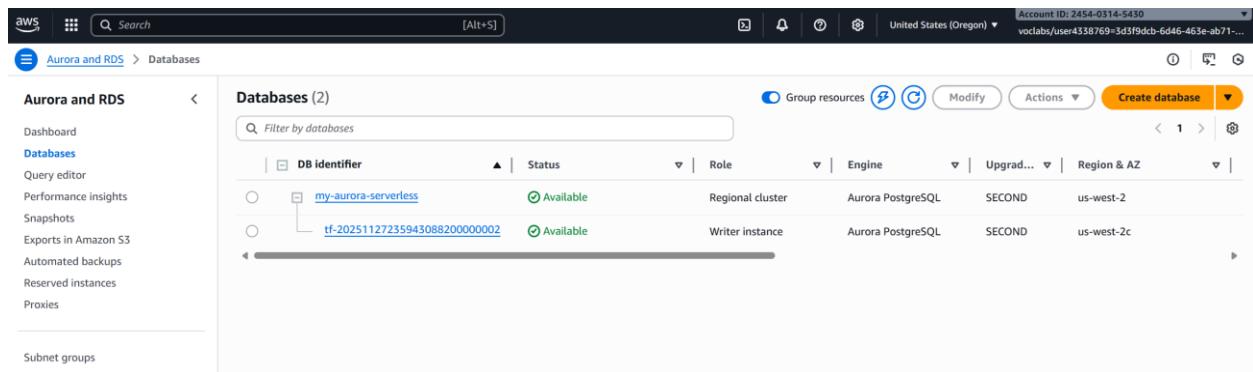


Image 7: Cluster created

```
aws rds describe-db-instances \
  --region us-west-2 \
  --query "DBInstances[].[ID:DBInstanceIdentifier,Status:DBInstanceStateStatus]"
[
  {
    "ID": "my-aurora-serverless",
    "Status": "available"
  }
]
[
  {
    "ID": "tf-20251127235943088200000002",
    "Status": "available"
  }
]
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applica
```

Image: checked through command line

```
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack2# aws rds-data execute-statement \
--resource-arn arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless \
--secret-arn arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-rzyen9 \
--database myapp \
--sql "SELECT * FROM pg_extension;" \
--region us-west-2
{
    "records": [
        [
            {
                "stringValue": "14506"
            },
            {
                "stringValue": "plpgsql"
            },
            {
                "stringValue": "10"
            }
        ]
    ]
}
```

Apply complete! Resources: 35 added, 0 changed, 0 destroyed.

Outputs:

```
aurora_arn = "arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cfpcpqajvwgv.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cfpcpqajvwgv.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cfpcpqajvwgv.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
    "subnet-04db308d423cf34e7",
    "subnet-07cefaeb44aba4468",
    "subnet-0b1e02e06ff07449e",
]
public_subnet_ids = [
    "subnet-02b257e1dac3c7015",
]
```

Redid due to connection issue

Secret details

Encryption key
aws/secretsmanager

Secret name
my-aurora-serverless

Secret ARN
arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-S9RsET

Secret description
-

Secret type
-

Overview | **Rotation** | **Versions** | **Replication** | **Tags**

Secret value Info
Retrieve and view the secret value.

Resource permissions - optional Info
Add or edit a resource policy to access secrets across AWS accounts.

Secret: Screenshot of AWS Secrets Manager showing the RDS secret created successfully

```
(.venv) root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution# python scripts/upload_s3.py
Successfully uploaded bulldozer-bd850-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/bulldozer-bd850-spec-sheet.pdf
Successfully uploaded dump-truck-dt1000-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/dump-truck-dt1000-spec-sheet.pdf
Successfully uploaded excavator-x950-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/excavator-x950-spec-sheet.pdf
Successfully uploaded forklift-f1250-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/forklift-f1250-spec-sheet.pdf
Successfully uploaded mobile-crane-mc750-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/mobile-crane-mc750-spec-sheet.pdf
(.venv) root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution#
```

Script executed and files uploaded

```
Changes to Outputs:
+ bedrock_knowledge_base_arn = (known after apply)
+ bedrock_knowledge_base_id = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

module.bedrock_kb.aws_bedrockagent_knowledge_base.main: Creating...
module.bedrock_kb.aws_bedrockagent_knowledge_base.main: Creation complete after 3s [id=QHUGGZNJNV]
module.bedrock_kb.aws_bedrockagent_data_source.s3_bedrock_bucket: Creating...
module.bedrock_kb.aws_bedrockagent_data_source.s3_bedrock_bucket: Creation complete after 1s [id=65YR4NEEXE,QHUGGZNJNV]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

bedrock_knowledge_base_arn = "arn:aws:bedrock:us-west-2:245403145430:knowledge-base/QHUGGZNJNV"
bedrock_knowledge_base_id = "QHUGGZNJNV"
root@Ishwin:/mnt/c/Users/sukhm/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack2#
```

KB created or stack 2 apply: Screenshot of terraform apply output showing successful resource creation

https://us-west-2.console.aws.amazon.com/bedrock/home?region=us-west-2#/knowledge-bases

Amazon Bedrock

- Discover
 - Overview
 - Model catalog
 - API keys
- Test
 - Chat / Text playground
 - Image / Video playground
 - Watermark detection
 - Tokenizer [New](#)
- Infer
 - Cross-region inference
 - Batch inference
 - Provisioned Throughput
 - Custom model on-demand
- Tune
 - Custom models
 - Prompt router models
 - Imported models
 - Marketplace model deployments
- Build
 - Agents
 - Flows

Knowledge Bases

How it works

- Create a 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.
 - Structured data store:** Use for structured data (e.g., databases, tables) to enable semantic search within existing systems via the Knowledge Base.
 - Kendra GenAI Index:** Use for document understanding powered by Kendra GenAI Index.
- Test**
- Integrate**

Knowledge Bases (1)

Name	Status	Type	Description	Creation time	Last sync date	Last sync warn...
my-bedrock-kb	Available	Vector store	-	November 27, ...	-	-

https://us-west-2.console.aws.amazon.com/bedrock/home?region=us-west-2#/knowledge-bases/my-bedrock-kb

Amazon Bedrock

- Discover
 - Overview
 - Model catalog
 - API keys
- Test
 - Chat / Text playground
 - Image / Video playground
 - Watermark detection
 - Tokenizer [New](#)
- Infer
 - Cross-region inference
 - Batch inference
 - Provisioned Throughput
 - Custom model on-demand
- Tune
 - Custom models
 - Prompt router models
 - Imported models
 - Marketplace model deployments
- Build
 - Agents
 - Flows

my-bedrock-kb

Knowledge Base overview

Knowledge Base name my-bedrock-kb	Knowledge Base ID QHUGGZNJVNV	Log Deliveries Configure log deliveries and event logs in the Edit page.
Knowledge Base description —	Status Available	Retrieval-Augmented Generation (RAG) type Vector store
Service Role my-bedrock-kb-role	Created date November 27, 2025, 22:00 (UTC-08:00)	

Data source (1)

Data so... Status Data sour... Account ID Source Link Last sync t... Last sync ... Text chun... Parsing st... Data dele...
s3_bedroc... Available S3 24540314... s3://bedro...

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
No tags	

[Manage tags](#)

localhost:8501

Configuration

Select LLM Model: anthropic.claude-3-haiku-20240...

Knowledge Base ID: your-knowledge-base-id

Temperature: 1.0

Top_P: 1.0

Bedrock Chat Application

What would you like to know? ➤

localhost:8501

Select LLM Model: anthropic.claude-3-haiku-20240...

Knowledge Base ID: QHUGZJNV

Temperature: 0.70

Top_P: 0.99

Bedrock Chat Application

What would you like to know? ➤

localhost:8501

Bedrock Chat Application

Select LLM Model
anthropic.claude-3-haiku-20240...

Knowledge Base ID
QHUGGZNJVN

Temperature
0.00 0.70 1.00

Top_P
0.00 0.90 1.00

What are the safety protocols for operating a backhoe loader?
Sorry, I can only answer questions related to heavy machinery.

Connected to database my-aurora-serverless successfully.

Editor **Recent** **Saved queries**

Query editor: my-aurora-serverless

```
1 SELECT * FROM pg_extension;
```

Run Save Clear

Output **Result set 1 (2)**

Rows returned (2)

oid	extname	extowner	extnamespace	extrelocatable	extversion	extconfig	extc
14506	plpgsql	10	11	false	1.0	NULL	NULL
16464	vector	10	2200	true	0.8.0	NULL	NULL

Export to csv

Screenshot of the AWS Aurora and RDS Query editor interface:

Left Sidebar:

- Aura and RDS
- Dashboard
- Databases
- Query editor** (selected)
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations
- Events
- Event subscriptions
- Recommendations (0)
- Certificate update

Query Editor Area:

Message Bar: Connected to database my-aurora-serverless successfully.

Editor Tab: Editor | Recent | Saved queries

Query Editor: my-aurora-serverless

```

1 SELECT
2   table_schema || '.' || table_name AS show_tables
3 FROM
4   information_schema.tables
5 WHERE
6   table_type = 'BASE TABLE'
7   AND table_schema = 'bedrock_integration';
8
9

```

Buttons: Run | Save | Clear

Output Tab: Output | Result set 1 (1)

Result Set: Rows returned (1) | Export to csv

show_tables
bedrock_integration.bedrock_kb

Right Sidebar:

- Info | Tutorials
- Getting started with Aurora Serverless v2
- Introduction to Amazon Aurora
- Amazon Aurora Serverless

Integration with KB

Screenshot of the AWS S3 console showing the contents of a bucket named bedrock-kb-245403145430:

Left Sidebar:

- Amazon S3
- Buckets
 - General purpose buckets
 - Directory buckets
 - Table buckets
 - Vector buckets
- Access management and security
 - Access Points
 - Access Points for F5x
 - Access Grants
 - IAM Access Analyzer
- Storage management and insights
 - Storage Lens
 - Batch Operations
- Account and organization settings
- AWS Marketplace for S3

Bucket Details: spec-sheets/

Objects (5):

Name	Type	Last modified	Size	Storage class
bulldozer-bd850-spec-sheet.pdf	pdf	November 27, 2025, 20:47:43 (UTC-08:00)	448.7 KB	Standard
dump-truck-dt1000-spec-sheet.pdf	pdf	November 27, 2025, 20:47:43 (UTC-08:00)	357.9 KB	Standard
excavator-x950-spec-sheet.pdf	pdf	November 27, 2025, 20:47:44 (UTC-08:00)	274.3 KB	Standard
forklift-fl250-spec-sheet.pdf	pdf	November 27, 2025, 20:47:44 (UTC-08:00)	226.9 KB	Standard
mobile-crane-mc750-spec-sheet.pdf	pdf	November 27, 2025, 20:47:45 (UTC-08:00)	357.7 KB	Standard

Data from S3 bucket correctly synchronized

Connected to database **my-aurora-serverless** successfully.

Query editor: my-aurora-serverless

```

1 SELECT
2   column_name,
3   data_type
4 FROM
5   information_schema.columns
6 WHERE
7   table_schema = 'bedrock_integration'
8   AND table_name = 'bedrock_kb';
9
10
11

```

Output **Result set 1 (4)**

column_name	data_type
id	uuid
chunks	text
embedding	USER-DEFINED
metadata	jsonb

Metadata database

spec-sheets/

Folder overview

AWS Region
US West (Oregon) us-west-2

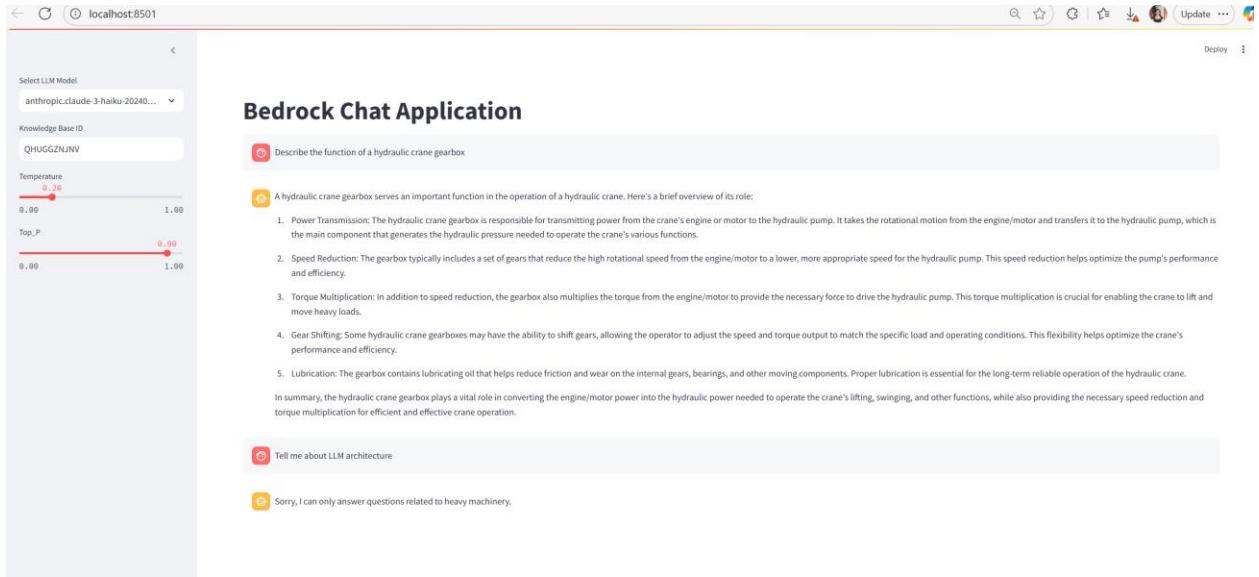
S3 URI
s3://bedrock-kb-245403145430/spec-sheets/

Amazon Resource Name (ARN)
arn:aws:s3:::bedrock-kb-245403145430/spec-sheets/

Copy S3 URI

S3

Python function implemented to query the knowledge base



```
def query_knowledge_base(query: str, kb_id: str, num_results: int = 3):
    """
    Calls Bedrock and retrieve info.
    """
    try:
        resp = bedrock_kb.retrieve(
            knowledgeBaseId=kb_id,
            retrievalQuery={"text": query},
            retrievalConfiguration={"vectorSearchConfiguration": {"numberOfResults": num_results}}
        )
        return resp.get("retrievalResults", [])
    except (NoCredentialsError, PartialCredentialsError):
        print("AWS credentials not found or incomplete. Set AWS env/profile.")
        return []
    except ClientError as e:
        print(f"Client Error query KB:{e}")
        return []
    except Exception as e:
        print(f"Unexpected error query run for KB:{e}")
        return []
```

Python function implemented to query the knowledge base

```

def generate_response(prompt: str, model_id: str, temperature: float = 0.7, top_p: float = 0.9):
    """
    Call for Bedrock .
    """

    try:
        messages = [{"role": "user", "content": [{"type": "text", "text": prompt}]}

        resp = bedrock.invoke_model(
            modelId=model_id,
            contentType="application/json",
            accept="application/json",
            body=json.dumps({
                "anthropic_version": "bedrock-2023-05-31",    # REQUIRED
                "messages": messages,
                "max_tokens": 400,
                "temperature": temperature,
                "top_p": top_p
            })
        )

        body = resp["body"].read()
        parsed = json.loads(body)
        return parsed.get("content", [{}])[0].get("text", "").strip()

    except (NoCredentialsError, PartialCredentialsError):
        return "Model call skipped: AWS credentials missing or incomplete."

```

Successful invocation of the model

```

def valid_prompt(prompt: str) -> bool:
    """
    to chk if a prompt is about heavy machinery.

    """
    if not prompt or not prompt.strip():
        return False

    p = prompt.lower()

    keywords = [
        "machine", "machinery", "heavy machinery", "excavator", "bulldozer",
        "hydraulic", "engine", "crane", "loader", "gearbox", "transmission",
        "lift", "drill", "compressor", "industrial"
    ]
    return any(k in p for k in keywords)

```

Correct implementation of valid_prompt function in bedrock_utils.py

```

>>> from bedrock_utils import valid_prompt
>>> print(valid_prompt("Describe the function of a hydraulic crane gearbox"))
True

>>> print(valid_prompt("Tell me about Sukhmani"))
False

```

Sample output filtering undesired prompts

Previous attempts

Fourth Attempt using AWS console

```
[7m0s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating...
[7m10s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Creation complete
after 7m20s [id=tf-20251122033238412900000002]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aurora_arn = "arn:aws:rds:us-west-2:448049796647:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-c5m6c2ik29sc.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-c5m6c2ik29sc.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-c5m6c2ik29sc.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
    "subnet-02f145da6207dc354",
    "subnet-0aa929befcc0dd09b",
    "subnet-09d80081a9a72ab24",
]
public_subnet_ids = [
    "subnet-0d77b46d5da67fe95",
    "subnet-02c2d860462b9df40",
    "subnet-0912038ad03a9f6da",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:448049796647:secret:my-aurora-serverless-y8crm8"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-448049796647"
vpc_id = "vpc-09b027c695e6b0f49"
stack1 $ ||
```

Databases (2)			
<input type="checkbox"/> Group resources		Actions ▾	Create database
<input type="text"/> Filter by databases			
		< 1 >	
DB identifier	Status	Role	
my-aurora-serverless	Available	Regional c...	
tf-20251122033238412900000002	Available	Writer ins...	

Aurora and RDS > Query editor: my-aurora-serverless

Connected to database **my-aurora-serverless** successfully.

Aurora and RDS
Editor Recent S >

Query editor: my-aurora-serverless

```
1 select * from information_schema.tables
2 # Press run and see the current data
```

Run
Save
Clear

Info
Tutorials

Getting started with Aurora Serverless v2

Introduction to Amazon Aurora



The screenshot shows the AWS CloudShell interface in the us-east-1 region. The top navigation bar includes the AWS logo, search, refresh, notifications, help, account ID (4480-4979-6647), location (United States), and user (Sukhmani). The main area is titled "CloudShell" and shows the following Terraform output:

```
+ field_mapping {
+   metadata_field    = "metadata"
+   primary_key_field = "id"
+   text_field        = "chunks"
+   vector_field      = "embedding"
}
}
}
}

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ bedrock_knowledge_base_arn = (known after apply)
+ bedrock_knowledge_base_id  = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

module.bedrock_kb.aws_bedrockagent_knowledge_base.main: Creating...
module.bedrock_kb.aws_bedrockagent_knowledge_base.main: Creation complete after 2s [id=AKWPX55T82]
module.bedrock_kb.aws_bedrockagent_data_source.s3_bedrock_bucket: Creating...
module.bedrock_kb.aws_bedrockagent_data_source.s3_bedrock_bucket: Creation complete after 0s [id=NDTKBVJSCH,AKWPX55T82]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

bedrock_knowledge_base_arn = "arn:aws:bedrock:us-west-2:448049796647:knowledge-base/AKWPX55T82"
bedrock_knowledge_base_id = "AKWPX55T82"
stack2 $
```

Below this, a terminal window shows the output of a Python script:

```
scripts $ python upload_s3.py
Successfully uploaded forklift-f1250-spec-sheet.pdf to bedrock-kb-448049796647/spec-sheets/forklift-f1250-spec-sheet.pdf
Successfully uploaded excavator-x950-spec-sheet.pdf to bedrock-kb-448049796647/spec-sheets/excavator-x950-spec-sheet.pdf
Successfully uploaded mobile-crane-mc750-spec-sheet.pdf to bedrock-kb-448049796647/spec-sheets/mobile-crane-mc750-spec-sheet.pdf
Successfully uploaded dump-truck-dt1000-spec-sheet.pdf to bedrock-kb-448049796647/spec-sheets/dump-truck-dt1000-spec-sheet.pdf
Successfully uploaded bulldozer-bd850-spec-sheet.pdf to bedrock-kb-448049796647/spec-sheets/bulldozer-bd850-spec-sheet.pdf
scripts $
```

The screenshot shows the AWS CloudShell interface in a browser window. The top navigation bar includes the AWS logo, a grid icon, a search icon, a refresh icon, a bell icon, a question mark icon, a gear icon, the text "United States", and dropdown menus for "Account ID: 4480-4979-6647" and "Sukhmani". Below the header is a dark-themed terminal window titled "CloudShell". The title bar has a "Actions" button with a dropdown arrow, a maximize/minimize button, and a settings gear icon. The main area displays a command-line session:

```
> --retrieval-query "What are the specs of the bulldozer BD850?"  
aws: [ERROR]: argument operation: Found invalid choice 'retrieve'  
  
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]  
To see help text, you can run:  
  
aws help  
aws <command> help  
aws <command> <subcommand> help  
  
stack2 $ aws bedrock-agent start-ingestion-job \  
> --region us-west-2 \  
> --knowledge-base-id AKWPX55T82 \  
> --data-source-id NDTKBVJSCH  
{  
  "ingestionJob": {  
    "knowledgeBaseId": "AKWPX55T82",  
    "dataSourceId": "NDTKBVJSCH",  
    "ingestionJobId": "EUGIK2N6SZ",  
    "status": "STARTING",  
    "statistics": {  
      "numberOfDocumentsScanned": 0,  
      "numberOfMetadataDocumentsScanned": 0,  
      "numberOfNewDocumentsIndexed": 0,  
      "numberOfModifiedDocumentsIndexed": 0,  
      "numberOfMetadataDocumentsModified": 0,  
      "numberOfDocumentsDeleted": 0,  
      "numberOfDocumentsFailed": 0  
    },  
    "startedAt": "2025-11-22T04:20:47.266765+00:00",  
    "updatedAt": "2025-11-22T04:20:47.266765+00:00"  
  }  
}  
stack2 $
```

Secrets Manager (RDS credentials)

The screenshot shows the AWS Secrets Manager interface. At the top, there's a navigation bar with the AWS logo, search, notifications, and account information (Account ID: 4480-4979-6647, Sukhmani). Below the navigation is a breadcrumb trail: AWS Secrets Manager > Secrets > my-aurora-serverless. To the right of the breadcrumb are two icons: a help icon and a refresh icon.

The main content area is titled "my-aurora-serverless". Under "Secret details", it shows:

- Encryption key: aws/secretsmanager
- Secret name: my-aurora-serverless
- Secret ARN: arn:aws:secretsmanager:us-west-2:448049796647:secret:my-aurora-serverless-y8crm8

To the right of these details are "Secret description" and "Secret type", both currently listed as "-". Above the "Actions" button is a "C" icon.

Below the "Secret details" section are tabs for Overview (which is selected), Rotation, Versions, Replication, and Tags.

Under the "Overview" tab, there's a section for "Secret value" with an "Info" link and a "Retrieve secret value" button.

Open RDS Query Editor (Aurora PostgreSQL)

< Editor Recent >

Query editor: my-aurora-serverless

```
1 SELECT
2   table_schema || '.' || table_name
3 FROM information_schema.tables
4 WHERE table_type = 'BASE TABLE'
5   AND table_schema = 'bedrock_inte
6
7
```

Run Save Clear

Output Result set 1 (2)

Rows returned (2)

Search rows

oid	extname	extowner
14498	plpgsql	10
16450	vector	10

Query editor: my-aurora-serverless

```
1 CREATE INDEX IF NOT EXISTS bedrock_embeddings_idx
2 ON bedrock_integration.bedrock_kb
3 USING hnsw (embedding vector_cosine
4
```

Run

Save

Clear

Output

Statements (1)

Search rows

Id	Start	Statement
1	20:26:43	CREATE INDEX IF NOT EXISTS bedrock_embeddings_idx ON bedrock_integration.bedrock_kb USING hnsw (embedding vector_cosine

Knowledge Base | AmazonBedrock | us-west-2 - [InPrivate]

https://us-west-2.console.aws.amazon.com/bedrock/home?region=us-west-2#knowledge-bases/my-bedrock-kb/AKWPX5STB2/0

InPrivate Update ... Account ID: 4480-4979-6647 United States (Oregon) Sukhmani

Amazon Bedrock > Knowledge Bases > my-bedrock-kb

my-bedrock-kb

Test Knowledge Base Delete Edit

Knowledge Base overview

Knowledge Base name: my-bedrock-kb

Knowledge Base ID: AKWPX5STB2

Knowledge Base description: —

Status: Available

Service Role: my-bedrock-kb-role

Created date: November 21, 2025, 20:04 (UTC-08:00)

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

Retrieval-Augmented Generation (RAG) type: Vector store

Data source (1)

Sync Stop sync Add

Find data source

Data source details:

Data so...	Status	Data sour...	Account ID	Source Link	Last sync ...	Last sync ...	Text chun...	Parsing st...	Data dele...
s3_bedro...	Available	S3	44804979...	s3://bedr...	November...	-	Default	Default	Delete

Tags: 0

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
No tags	

Knowledgebase

The screenshot shows the AWS CloudShell interface. At the top, there are navigation icons (aws, grid, search, refresh, help, settings), account information (Account ID: 4480-4979-6647, Sukhmani), and a dropdown for United States. Below the header is a toolbar with actions (Actions ▾, copy, settings). The main area is a terminal window titled "CloudShell" with the region set to "us-east-1". The terminal output shows a command being run:

```
-solution $ aws dearock-agent list-ingestion-jobs \
> --region us-west-2 \
> --knowledge-base-id AKWPX55T82 \
> --data-source-id NDTKBVJSCH
{
  "ingestionJobSummaries": [
    {
      "knowledgeBaseId": "AKWPX55T82",
      "dataSourceId": "NDTKBVJSCH",
      "ingestionJobId": "Y56LGZYKRE",
      "status": "COMPLETE",
      "startedAt": "2025-11-22T04:08:55.593665+00:00",
      "updatedAt": "2025-11-22T04:08:59.382467+00:00",
      "statistics": {
        "numberOfDocumentsScanned": 2,
        "numberOfMetadataDocumentsScanned": 0,
        "numberOfNewDocumentsIndexed": 2,
        "numberOfModifiedDocumentsIndexed": 0,
        "numberOfMetadataDocumentsModified": 0,
        "numberOfDocumentsDeleted": 0,
        "numberOfDocumentsFailed": 0
      }
    },
    {
      "knowledgeBaseId": "AKWPX55T82",
      "dataSourceId": "NDTKBVJSCH",
      "ingestionJobId": "EUGIK2N6SZ",
      "status": "COMPLETE",
      "startedAt": "2025-11-22T04:20:47.266765+00:00",
      "updatedAt": "2025-11-22T04:21:15.310033+00:00",
      "statistics": {
        "numberOfDocumentsScanned": 7,
        "numberOfMetadataDocumentsScanned": 0,
        "numberOfNewDocumentsIndexed": 5,
        "numberOfModifiedDocumentsIndexed": 0,
        "numberOfMetadataDocumentsModified": 0,
        "numberOfDocumentsDeleted": 0,
        "numberOfDocumentsFailed": 0
      }
    }
  ]
}
```

Ingestion

Third attempt

```

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

aurora_arn = "arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
    "subnet-07bf374a571a4c2b5",
    "subnet-008e34d58643374c8",
    "subnet-0273daa2e3a343efb",
]
public_subnet_ids = [
    "subnet-00534bc4c16234d26",
    "subnet-0b1d970a2480b7181",
    "subnet-094f3ab00526d9046",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-gSXYiq"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-245403145430"
vpc_id = "vpc-090536abbd4dd0b52"
@sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1 (main) $
```

DB identifier	Status	Role	Engine	Upgrad...	Region & AZ
my-aurora-serverless	Available	Regional cluster	Aurora PostgreSQL	SECOND	us-west-2
tf-2025112201280042320000002	Available	Writer instance	Aurora PostgreSQL	SECOND	us-west-2

Images from second Attempt

```

@sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1 (main) $ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/aws v6.20.0
- Using previously-installed hashicorp/random v3.7.2

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```

modules/
└── bedrock_kb
    ├── main.tf
    └── outputs.tf
  └── database
      ├── main.tf
      └── outputs.tf
  └── variables.tf
scripts/
stack1
stack2
__init__.py
.gitignore
app.py
bedrock_utils.py
README.md
requirements.txt
variables.tf

bash - stack1 + 1 2 3 ... | ⌂
@sukhman: ~/workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1 (main) $ terraform apply
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating... [96m50s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating... [97m00s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating... [97m10s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating... [97m20s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Still creating... [97m30s elapsed]
module.aurora_serverless.aws_rds_cluster_instance.aurora_instance: Creation complete after 9m40s [id=tf-202511105242893070000001]

Warning: Deprecated attribute
  on .terraform/modules/vpc/vpc-flow-logs.tf line 28, in locals:
  28:   "arn:${data.aws_partition.current[0].partition}:logs:${data.aws_region.current[0].name}:${data.aws_caller_identity.current[0].account_id}:log-group:${log_group.name}":"
The attribute "name" is deprecated. Refer to the provider documentation for details.

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

Outputs:
aurora_arn = "arn:aws:rds:us-west-2:621074666262:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cs20rasuw1hx.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cs20rasuw1hx.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cs20rasuw1hx.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
  "subnet-0baaa6242cefbf9dd",
  "subnet-03dbe6977ad571d31",
  "subnet-03e4d12be5dd4a0fa",
]
public_subnet_ids = [
  "subnet-0876089fae55c8714",
  "subnet-0d789b94de98673",
  "subnet-02fa219f4ce75f429",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:621074666262:secret:my-aurora-serverless-rovkok"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-621074666262"
vpc_id = "vpc-09be14d9023047cf6"

```

terraform_apply_output.png

stack1_apply_success.png

DB identifier	Status	Role	Engine	Region	Size	Recommendation
my-aurora-serverless	Available	Regional cluster	Aurora Po...	us-west-2	1 instance	
tf-202511105242893070000001	Available	Writer instance	Aurora Po...	us-west-2c	Serverless v2 (0.5 - 1 ACUs)	

aws_console_rds_cluster.png

The screenshot shows the AWS Aurora and RDS console. The left sidebar has sections for Aurora and RDS, Databases, Query editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main area is titled 'my-aurora-serverless'. It shows a table of databases with columns: DB identifier, Status, Role, Engine, Region ..., Size, Recom..., CPU, and GPU. One row is selected: 'my-aurora-serverless' (Available, Regional cluster, Aurora PostgreSQL, us-west-2, 1 instance). Another row is 'tf-20251111052428930700000001' (Available, Writer instance, Aurora PostgreSQL, us-west-2c, Serverless..., 51.49%, 0%). Below this is a tab bar with Connectivity & security, Monitoring, Logs & events, Configuration, Zero-ETL integrations, Maintenance & backups, and Data migrations. Under 'Connectivity & security', there is a section for Endpoints (2) with two entries: 'my-aurora-serverless.cluster-cs20rasuw1hx.us-west-2.rds.amazonaws.com' (Available, Writer, 5432) and 'my-aurora-serverless.cluster-ro-cs20rasuw1hx.us-west-2.rds.amazonaws.com' (Available, Reader, 5432). There is also a 'Manage IAM roles' section with tabs for Add IAM roles to this cluster and Feature.

my-aurora-serverless

The screenshot shows the AWS Amazon S3 console. The left sidebar has sections for General purpose buckets, Directory buckets, Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, and Batch Operations. The main area is titled 'General purpose buckets' with a 'All AWS Regions' button. It shows a table of buckets with columns: Name, AWS Region, and Creation date. One bucket is listed: 'bedrock-kb-621074666262' (US West (Oregon) us-west-2, November 10, 2025, 20:50:00 (UTC-08:00)). There are buttons for Copy ARN, Empty, Delete, and Create bucket. A 'Find buckets by name' search bar is at the top of the table.

aws_console_s3_bucket.png

```
@sukhmani ➔ /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack2 (main) $ terraform init
Initializing the backend...
Initializing modules...
- bedrock_kb in ../modules/bedrock_kb
Initializing provider plugins...
- Finding latest version of hashicorp/time...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/time v0.13.1...
- Installed hashicorp/time v0.13.1 (signed by HashiCorp)
- Installing hashicorp/aws v6.20.0...
- Installed hashicorp/aws v6.20.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other
```

stack 2

```
@sukhmani ➔ /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack2 (main) $ terraform apply
module.bedrock_kb.aws_iam_policy.bedrock_kb_rds_access: Refreshing state... [id=arn:aws:iam::621074666262:policy/bedrock_kb_rds_access]
module.bedrock_kb.aws_iam_role.bedrock_kb_role: Refreshing state... [id=my-bedrock-kb-role]
module.bedrock_kb.data.aws_caller_identity.current: Reading...
module.bedrock_kb.aws_iam_policy.rds_data_api_policy: Refreshing state... [id=arn:aws:iam::621074666262:policy/my-bedrock-kb-rds-data-api-policy]
module.bedrock_kb.data.aws_caller_identity.current: Read complete after 0s [id=621074666262]
module.bedrock_kb.aws_iam_role_policy_attachment.bedrock_kb_policy: Refreshing state... [id=my-bedrock-kb-role/arn:aws:iam::aws:policy/AmazonBedrockFullAccess]
module.bedrock_kb.aws_iam_role_policy_attachment.rds_policy_attachment: Refreshing state... [id=my-bedrock-kb-role/arn:aws:iam::621074666262:policy/bedrock_kb_rds_access]
module.bedrock_kb.aws_iam_role_policy_attachment.rds_data_api_policy_attachment: Refreshing state... [id=my-bedrock-kb-role/arn:aws:iam::621074666262:policy/my-bedrock-kb-rds-data-api-policy]
module.bedrock_kb.time_sleep.wait_10_seconds: Refreshing state... [id=2025-11-11T05:47:12Z]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# module.bedrock_kb.aws_bedrockagent_data_source.s3_bedrock_bucket will be created
+ resource "aws_bedrockagent_data_source" "s3_bedrock_bucket" {
    + data_deletion_policy = (known after apply)
    + data_source_id = (known after apply)
    + id = (known after apply)
    + knowledge_base_id = (known after apply)
    + name = "s3_bedrock_bucket"
    + region = "us-west-2"

    + data_source_configuration {
        + type = "S3"

        + s3_configuration {
            + bucket_arn = "arn:aws:s3:::bedrock-kb-621074666262"
        }
    }
}

# module.bedrock_kb.aws_bedrockagent_knowledge_base.main will be created
+ resource "aws_bedrockagent_knowledge_base" "main" {
```

```
@sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack2 (main) $ terraform apply
# module.bedrock_kb.aws_bedrockagent_knowledge_base.main will be created
+ resource "aws_bedrockagent_knowledge_base" "main" {
  + arn          = (known after apply)
  + created_at   = (known after apply)
  + failure_reasons = (known after apply)
  + id           = (known after apply)
  + name         = "my-bedrock-kb"
  + region        = "us-west-2"
  + role_arn      = "arn:aws:iam::621074666262:role/my-bedrock-kb-role"
  + tags_all      = {}
  + updated_at    = (known after apply)

  + knowledge_base_configuration {
    + type = "VECTOR"

    + vector_knowledge_base_configuration {
      + embedding_model_arn = "arn:aws:bedrock:us-west-2:foundation-model/amazon.titan-embed-text-v1"
    }
  }

  + storage_configuration {
    + type = "RDS"

    + rds_configuration {
      + credentials_secret_arn = "arn:aws:secretsmanager:us-west-2:621074666262:secret:my-aurora-serverless-rovkox"
      + database_name          = "myapp"
      + resource_arn            = "arn:aws:rds:us-west-2:621074666262:cluster:my-aurora-serverless"
      + table_name              = "bedrock_integration.bedrock_kb"

      + field_mapping {
        + metadata_field     = "metadata"
        + primary_key_field = "id"
        + text_field         = "chunks"
        + vector_field       = "embedding"
      }
    }
  }
}
```

stack2_apply_output.png

aws_console_bedrock_kb.png

```
@sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution (main) $ python scripts/upload_s3.py
Successfully uploaded excavator-x950-spec-sheet.pdf to bedrock-kb-621074666262/spec-sheets/excavator-x950-spec-sheet.pdf
Successfully uploaded dump-truck-dt1000-spec-sheet.pdf to bedrock-kb-621074666262/spec-sheets/dump-truck-dt1000-spec-sheet.pdf
Successfully uploaded mobile-crane-mc750-spec-sheet.pdf to bedrock-kb-621074666262/spec-sheets/mobile-crane-mc750-spec-sheet.pdf
Successfully uploaded bulldozer-bd850-spec-sheet.pdf to bedrock-kb-621074666262/spec-sheets/bulldozer-bd850-spec-sheet.pdf
Successfully uploaded forklift-f1250-spec-sheet.pdf to bedrock-kb-621074666262/spec-sheets/forklift-f1250-spec-sheet.pdf
```

Script run to upload files

Images from First Attempt

```
● @sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1 (main) $ terraform output
aurora_arn = "arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
db_endpoint = "my-aurora-serverless.cluster-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
db_reader_endpoint = "my-aurora-serverless.cluster-ro-cfpccpqajvwgv.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
  "subnet-038815b4a2ca62185",
  "subnet-0bece1e27574099767",
  "subnet-053900ac4605e8e6a",
]
public_subnet_ids = [
  "subnet-0c6425ee4e387c5f0",
  "subnet-0ef903c2dab639a8",
  "subnet-06660667b5bbd9a47",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-HiTihc"
s3_bucket_name = "arn:aws:s3:::bedrock-kb-245403145430"
vpc_id = "vpc-03f451f5ad56437a1"
```

Outputs:

```
aurora_arn = "arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cfpqpqajvwgv.us-west-2.rds.amazonaws.com"
fb_endpoint = "my-aurora-serverless.cluster-cfpqpqajvwgv.us-west-2.rds.amazonaws.com"
fb_reader_endpoint = "my-aurora-serverless.cluster-ro-cfpqpqajvwgv.us-west-2.rds.amazonaws.com"
private_subnet_ids = [
    "subnet-038815b4a2ca62185",
    "subnet-0bec1e27574099767",
    "subnet-0539d0ac4605e8e6a",
]
public_subnet_ids = [
    "subnet-0c6425ee4e387c5f0",
    "subnet-0e7f903c2dab639a8",
    "subnet-06660667b5bb9a47",
]
rds_secret_arn = "arn:aws:secretsmanager:us-west-2:245403145430:secret:my-aurora-serverless-HiTihc"
s3_bucket_name = "arn:aws:s3::::bedrock-kb-245403145430"
vpc_id = "vpc-03f451f5ad56437a1"
@sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution/stack1 (main)
aurora_arn = "arn:aws:rds:us-west-2:245403145430:cluster:my-aurora-serverless"
aurora_endpoint = "my-aurora-serverless.cluster-cfpqpqajvwgv.us-west-2.rds.amazonaws.com"
```

The screenshot shows the AWS Aurora and RDS console interface. The left sidebar navigation includes 'Aurora and RDS', 'Dashboard', 'Databases', 'Query editor', 'Performance insights', 'Snapshots', 'Exports in Amazon S3', 'Automated backups', 'Reserved instances', 'Proxies', 'Subnet groups', 'Parameter groups', 'Option groups', 'Custom engine versions', 'Zero-ETL integrations', 'Events', 'Event subscriptions', 'Recommendations (0)', and 'Certificate update'. The main content area displays a success message: 'Connected to database my-aurora-serverless successfully.' Below this, the database identifier 'tf-20251110235911756000000002' is shown. A 'Related' section lists the database and its writer instance. The 'Connectivity & security' tab is selected, showing details for the endpoint, networking, and security groups.

DB identifier	Status	Role	Engine	Region	Size	Recom...	CPU	Current...
my-aurora-serverless	Available	Regional c...	Aurora Po...	us-west-2	1 instance	-	-	-
tf-20251110235911756000000002	Available	Writer ins...	Aurora Po...	us-west-2c	Serverless...	52.55%	0	0

Connectivity & security

Endpoint & port	Networking	Security
Endpoint: tf-20251110235911756000000002.cfpqpqajvwgv.us-west-2.rds.amazonaws.com Port: 5432	Availability Zone: us-west-2c VPC: bedrock-poc-vpc (vpc-03f451f5ad56437a1) Subnet group: my-aurora-serverless-subnet-group Subnets: subnet-0bec1e27574099767, subnet-0539d0ac4605e8e6a, subnet-038815b4a2ca62185	VPC security groups: my-aurora-serverless-sg (sg-01bdd43ae00f0d481) (Active) Publicly accessible: No Certificate authority: rds-ca-ra2048-g1 Certificate authority date: May 24, 2061, 15:59 (UTC-07:00) DB instance certificate expiration date: November 10, 2026, 16:03 (UTC-08:00)

Connected to database **my-aurora-serverless** successfully. X

Editor Recent Saved queries

Query editor: my-aurora-serverless

↔ ↵ ⚙️

Change database

```
1 select * from information_schema.tables;
2 # Press run and see the current database tables below
```

Run

Save

Clear

Output

Result set 1 (197)

Rows returned (197)

Export to csv

Search rows

< 1 2 3 4 5 6 7 ... 10 >

⚙️

table_catalog	table_schema	table_name	table_type	self_referencing_column_name
postgres	pg_catalog	pg_stats_ext	VIEW	NULL
postgres	pg_catalog	pg_stats_ext_exprs	VIEW	NULL
postgres	pg_catalog	pg_type	BASE TABLE	NULL
postgres	pg_catalog	pg_foreign_table	BASE TABLE	NULL

Connected to database my-aurora-serverless successfully.

Query editor: my-aurora-serverless

```
1 select * from information_schema.tables;
2 # Press run and see the current database tables below
```

Output **Result set 1 (197)**

Rows returned (197)

table_catalog	table_schema	table_name	table_type	self_referencing_column_name
postgres	pg_catalog	pg_stats_ext	VIEW	NULL
postgres	pg_catalog	pg_stats_ext_exprs	VIEW	NULL
postgres	pg_catalog	pg_type	BASE TABLE	NULL
postgres	pg_catalog	pg_foreign_table	BASE TABLE	NULL

Connected to database my-aurora-serverless successfully.

Query editor: my-aurora-serverless

```
1 CREATE EXTENSION IF NOT EXISTS vector;
2 SELECT * FROM pg_extension;
3
```

Output **Result set 2 (2)**

Rows returned (2)

oid	extname	extowner	extnamespace	extrelatable	extversion	extconfig	extconfig
14498	plpgsql	10	11	false	1.0	NULL	NULL
16450	vector	10	2200	true	0.8.0	NULL	NULL

aws.amazon.com/rds/home?region=us-west-2#query-editor:

```

1 SELECT
2   table_schema || '.' || table_name AS show_tables
3 FROM
4   information_schema.tables
5 WHERE
6   table_type = 'BASE TABLE'
7   AND table_schema = 'bedrock_integration';
8
9

```

Run Save Clear

Output Result set 1 (1)

Rows returned (1)

Export to csv

Search rows

show_tables
bedrock_integration.bedrock_kb

```

@ @sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution (main) $ python scripts/upload_s3.py
Successfully uploaded excavator-x950-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/excavator-x950-spec-sheet.pdf
Successfully uploaded dump-truck-dt1000-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/dump-truck-dt1000-spec-sheet.pdf
Successfully uploaded mobile-crane-mc750-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/mobile-crane-mc750-spec-sheet.pdf
Successfully uploaded bulldozer-bd850-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/bulldozer-bd850-spec-sheet.pdf
Successfully uploaded forklift-f1250-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/forklift-f1250-spec-sheet.pdf

```

Screenshot the result showing bedrock_integration.bedrock_kb

https://us-west-2.console.aws.amazon.com/s3/buckets/bedrock-kb-245403145430?region=us-west-2&tab=objects

Amazon S3 Buckets bedrock-kb-245403145430

Objects (0)

No objects
You don't have any objects in this bucket.

Upload

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

```

@ @sukhmani → /workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution (main) $ python scripts/upload_s3.py
Successfully uploaded excavator-x950-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/excavator-x950-spec-sheet.pdf
Successfully uploaded dump-truck-dt1000-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/dump-truck-dt1000-spec-sheet.pdf
Successfully uploaded mobile-crane-mc750-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/mobile-crane-mc750-spec-sheet.pdf
Successfully uploaded bulldozer-bd850-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/bulldozer-bd850-spec-sheet.pdf
Successfully uploaded forklift-f1250-spec-sheet.pdf to bedrock-kb-245403145430/spec-sheets/forklift-f1250-spec-sheet.pdf

```

```
@sukhmani ~/workspaces/cd13926-Building-Generative-AI-Applications-with-Amazon-Bedrock-and-Python-project-solution (main) $ cd stack2
terraform init
Initializing the backend...
Initializing modules...
- bedrock_kb in ../modules/bedrock_kb
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/time...
- Installing hashicorp/aws v6.20.0...
- Installed hashicorp/aws v6.20.0 (signed by HashiCorp)
- Installing hashicorp/time v0.13.1...
- Installed hashicorp/time v0.13.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!
```

```
# module.bedrock_kb.aws_bedrockagent_knowledge_base.main will be created
+ resource "aws_bedrockagent_knowledge_base" "main" {
    + arn          = (known after apply)
    + created_at   = (known after apply)
    + failure_reasons = (known after apply)
    + id           = (known after apply)
    + name         = "my-bedrock-kb"
    + region        = "us-west-2"
    + role_arn      = "arn:aws:iam::245403145430:role/my-bedrock-kb-role"
    + tags_all      = {}
    + updated_at    = (known after apply)

    + knowledge_base_configuration {
        + type = "VECTOR"

        + vector_knowledge_base_configuration {
            + embedding_model_arn = "arn:aws:bedrock:us-west-2::foundation-model/amazon.titan-embed-text-v1"
        }
    }
}
```

Connected to database my-aurora-serverless successfully.

X

Editor Recent Saved queries

Query editor: my-aurora-serverless

↶ ↽ ⚙ Change database

```
1 CREATE SCHEMA bedrock_integration;
2
3 CREATE TABLE bedrock_integration.bedrock_kb (
4     id UUID PRIMARY KEY,
5     chunks TEXT,
6     embedding VECTOR(1536),
7     metadata JSONB
8 );
9
10
11
```

Run

Save

Clear

Output

Statements (2)

Export to csv

Search rows

< 1 >

⚙

ID	Start	Statement
1	17:34:49	CREATE SCHEMA bedrock_integration
2	17:34:50	CREATE TABLE bedrock_integration.bedrock_kb (id UUID PRIMARY KEY, chunks TEXT, embedding

stack2_apply_success.png