

Data Sources

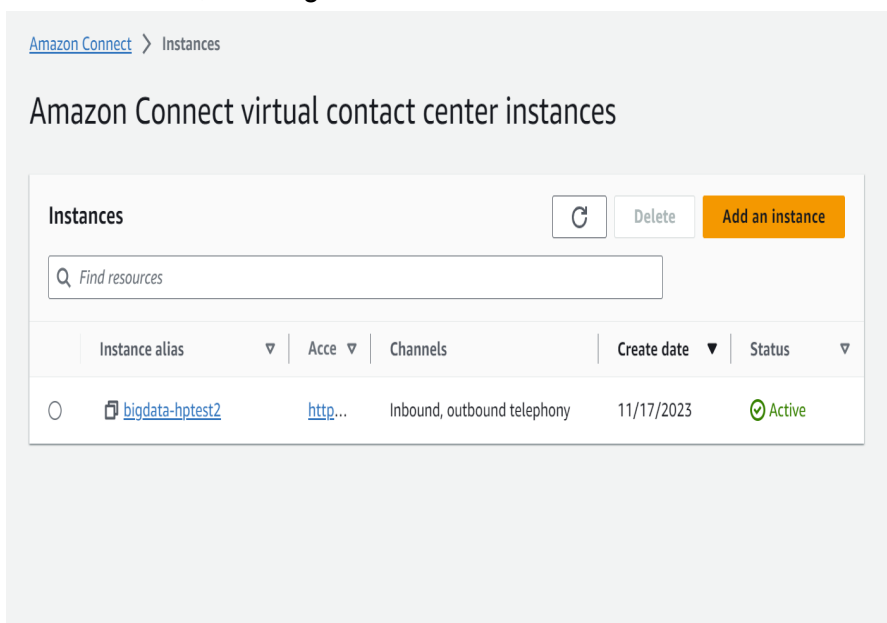
This project will be focused on helping customers interact with a customer support agent on the phone and we would be processing the customer's request in real time. For the call center experience we are using AWS Connect to be our cloud contact center.

First we would need to create an AWS Connect Instance.

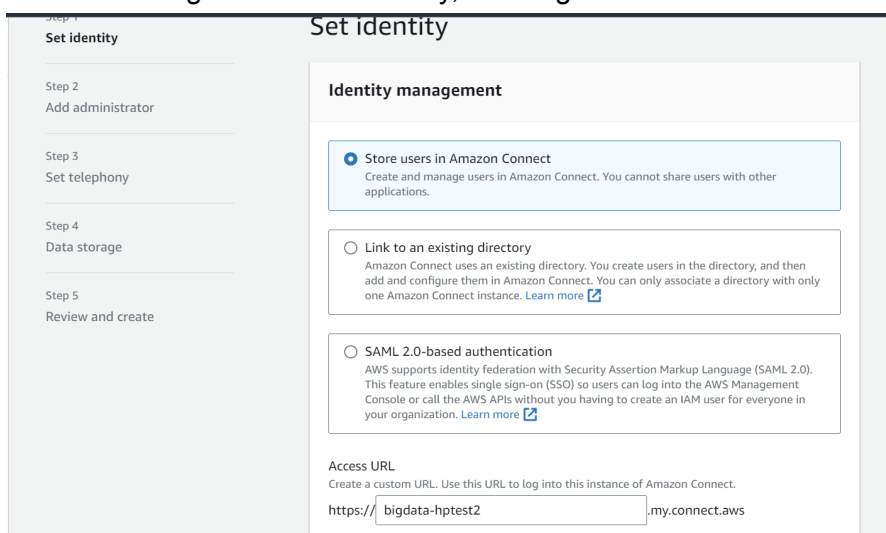
AWS Connect

First Creating an AWS Connect Instance

On the console, creating a instance.



The first setting is to set an identity, creating a name for the instance.



In Set telephony, this option allows our Connect Instance to handle both incoming and outgoing calls.

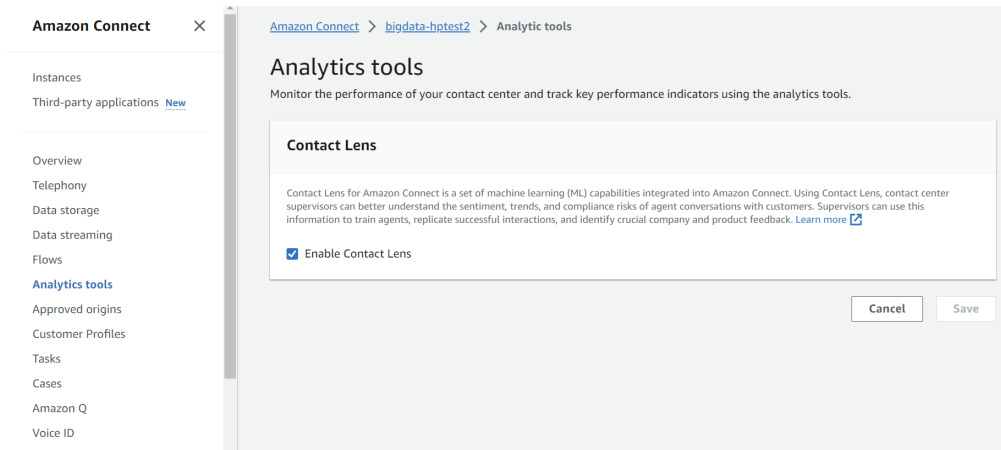
The screenshot shows the 'Set telephony' step in the 'Create Amazon Connect instance' wizard. On the left, a sidebar lists five steps: Step 1 'Set identity', Step 2 'Add administrator', Step 3 'Set telephony' (which is highlighted), Step 4 'Data storage', and Step 5 'Review and create'. The main content area is titled 'Set telephony' and contains a 'Telephony Options' box. This box instructs the user to 'Choose whether your contact center allows inbound calls, outbound calls, or both.' and includes two checked checkboxes: 'Allow incoming calls' and 'Allow outgoing calls'. At the bottom right of the main area are three buttons: 'Cancel', 'Previous', and 'Next'.

This will create the connect instance.

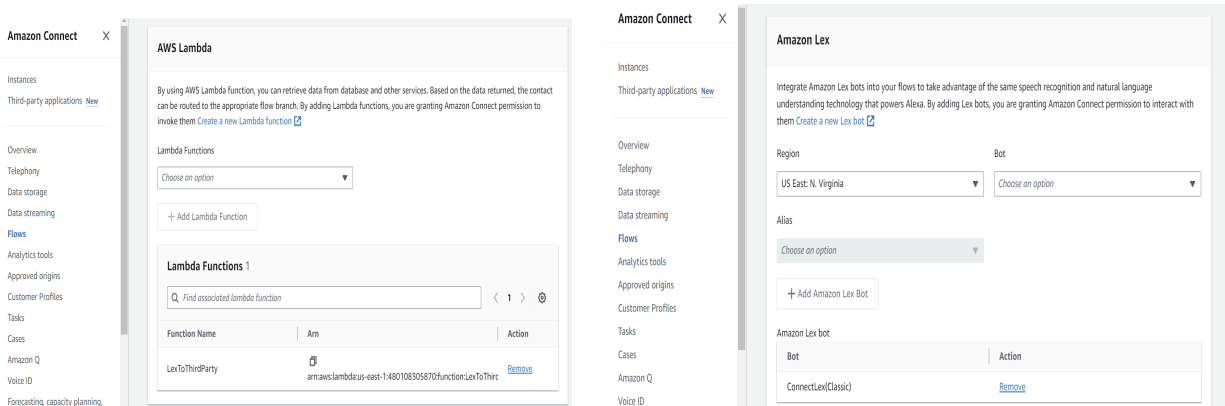
Once we have the base instance created, we can modify the remaining settings.

The screenshot displays the 'Data storage' configuration page within the Amazon Connect console. The breadcrumb trail at the top reads 'Amazon Connect > bigdata-hptest2 > Data storage'. The left-hand navigation menu includes links for 'Instances', 'Third-party applications', 'Overview', 'Telephony', 'Data storage' (the active link), 'Data streaming', 'Flows', 'Analytics tools', 'Approved origins', 'Customer Profiles', 'Tasks', 'Cases', 'Amazon Q', 'Voice ID', and 'Forecasting, capacity planning'. The main content area is titled 'Data storage' and includes a descriptive paragraph about saving data to an Amazon S3 bucket. Below this, there are two configuration sections: 'Call recordings' and 'Chat transcripts'. Each section shows the S3 bucket name (e.g., 'amazon-connect-b47ce2974165/connect/bigdata-hptest2/CallRecordings') and the encryption key ('aws/connect'). Each section also has an 'Edit' button.

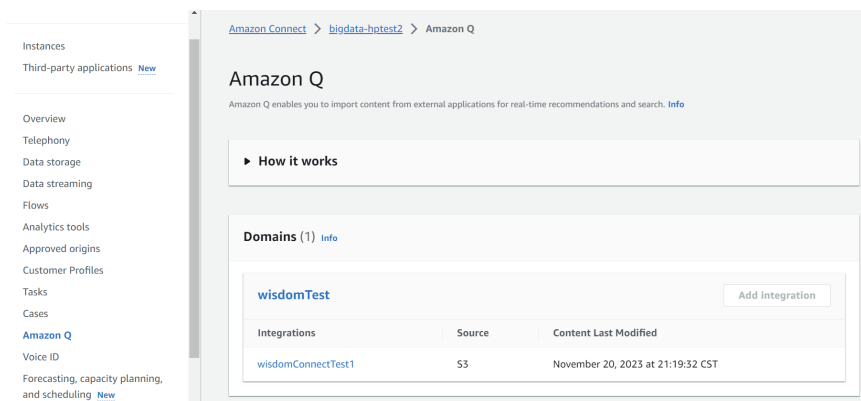
Once the instance is created, we can choose the instances and edit the data storage. This allows for the transcripts of the call to be stored in S3. Both recordings and chat data would be stored.



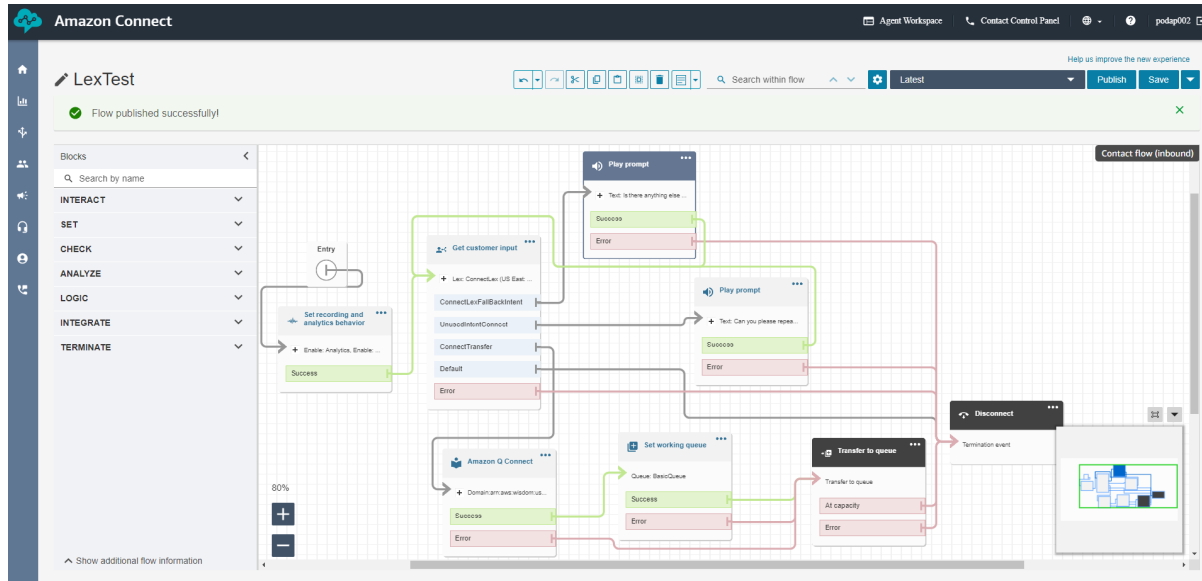
By allowing for Contact Lens in the Analytics Tools we can get access to post call analytics.



In the Flows sections, we would need to add AWS Lambda and Lex. Before we add the two services to the flow sections, we would need to create them separately first. Will look into creating these services in a later section. Please refer to them and create the resources and come back to this section.



The final setting in Connect would be to add in the Wisdom which allows to be used in the Agent's workspace.



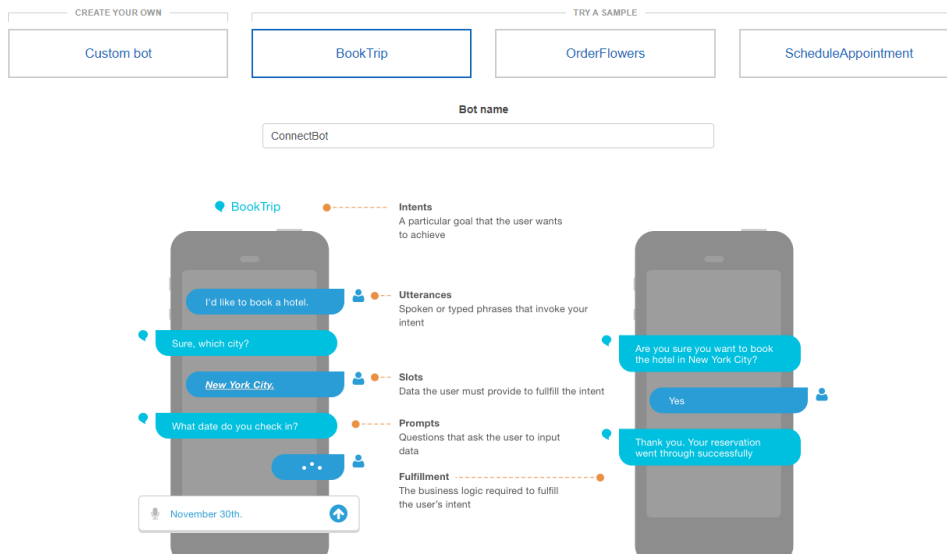
This is the flow of the phone conversation.

AWS Lex

In the V1 console, we will be building a custom bot.

Create your bot

Amazon Lex enables any developer to build conversational chatbots quickly and easily. With Amazon Lex, no deep learning expertise is necessary—you just specify the basic conversational flow directly from the console, and then Amazon Lex manages the dialogue and dynamically adjusts the response. To get started, you can choose one of the sample bots provided below or build a new custom bot from scratch.



Each Lex bot must have intents set up. These intents look for what the user inputs into the system and for each intent Lex will have a set of actions being defined.

The first intent will be a FallBack Intent, this intent will be triggered when all other intents fail and

Lex cannot activate the other intents. For our project this is the most important intent. As there will be no predefined intents and so each time the customer has a query the fallback intent will trigger.

The screenshot shows the AWS Lex console interface for the 'ConnectLex' project. The left sidebar lists the intents: 'ConnectLexFallbackIntent', 'ConnectLexFallbackIntent', and 'UnusedIntentConnect'. The main panel is titled 'ConnectLexFallbackIntent' and contains the following sections:

- Sample utterances:** A note stating that built-in intents provide an extensive set of pre-built utterances that cannot be viewed or edited.
- Fulfillment:** The 'AWS Lambda function' option is selected. The 'Lambda function' dropdown is set to 'bedrock39', and the 'Version or alias' dropdown is set to 'Latest'.
- Response:** An 'Add Message' button is visible, and the 'Enable response card' checkbox is unchecked.

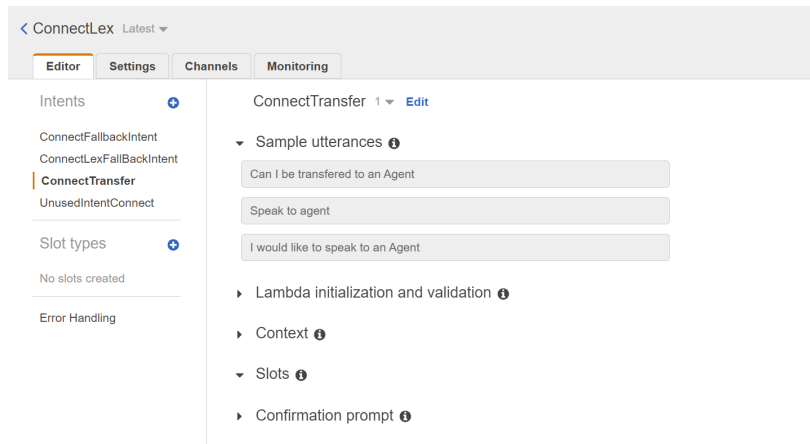
Within the Fallback Intent we will attach a Lambda function. This lambda function will take in the user inputs, process it and return a result back to the user.

The screenshot shows the AWS Lex console interface for the 'UnusedIntentConnect' project. The left sidebar lists the intents: 'ConnectLexFallbackIntent', 'ConnectLexFallbackIntent', and 'UnusedIntentConnect'. The main panel is titled 'UnusedIntentConnect' and contains the following sections:

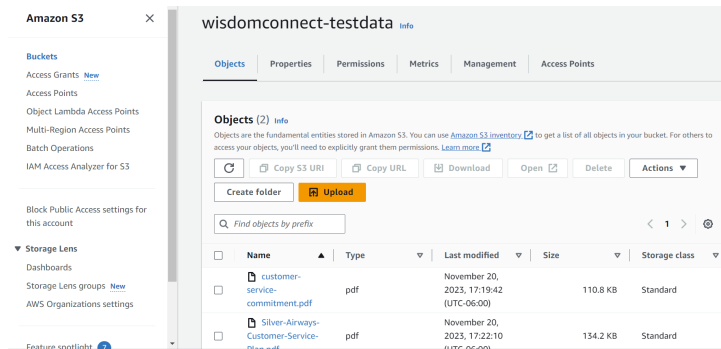
- Sample utterances:** Two sample utterances are listed: 'e.g. I would like to book a flight' and 'I am having network issues'.
- Lambda initialization and validation:** A section for configuring the Lambda function.
- Context:** A section for configuring the context.
- Slots:** A table with columns: Priority, Required, Name, Slot type, Version, Prompt, and Settings. The table contains one row with the following values: Priority (empty), Required (empty), Name ('e.g. Location'), Slot type ('e.g. AMAZON_US_CITY'), Version (empty), Prompt ('e.g. What city?'), and Settings (empty).
- Confirmation prompt:** A section for configuring the confirmation prompt.
- Fulfillment:** The 'Return parameters to client' option is selected.
- Response:** An 'Add Message' button is visible, and the 'Enable response card' checkbox is unchecked.

AWS Lex requires to have a defined intent and while our project does not require it. So the sample intent will just be a placeholder for now.

The final intent is to listen for any requests from the customer to speak to a human Agent. So this would transfer the call to a human agent also handled by AWS Connect.

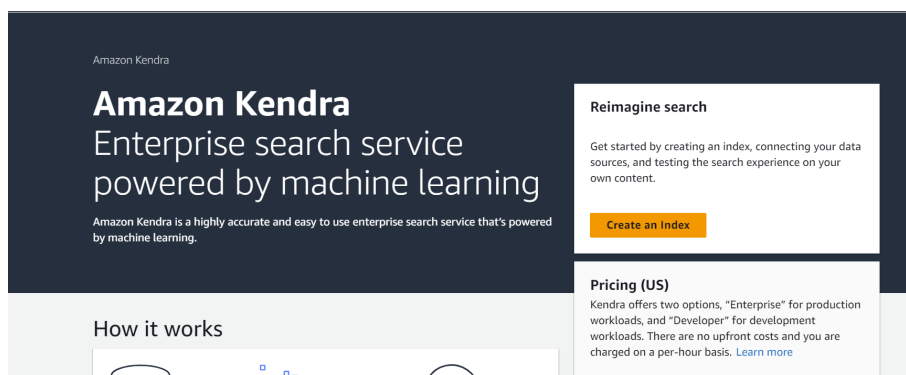


AWS S3

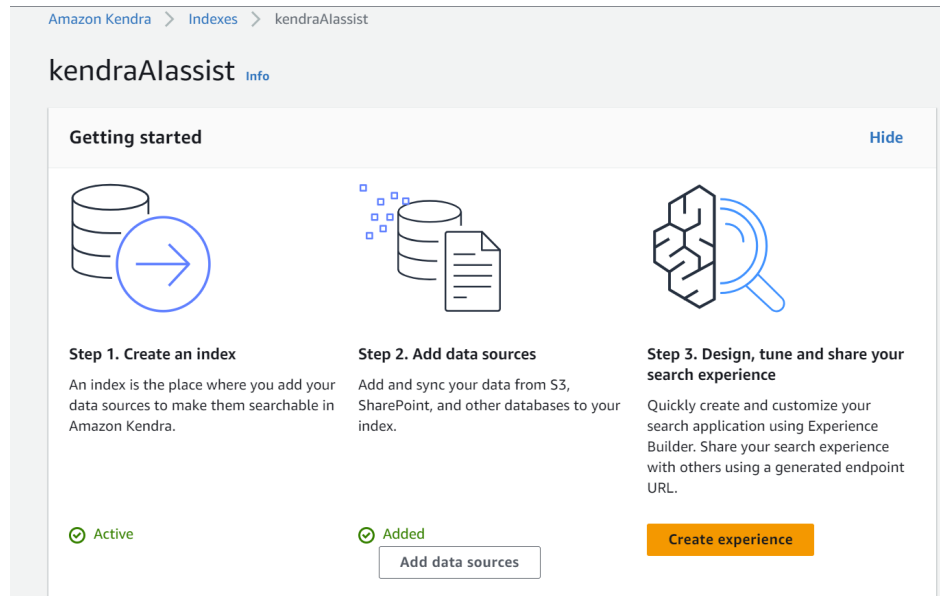


Created a S3 bucket to store the company policies

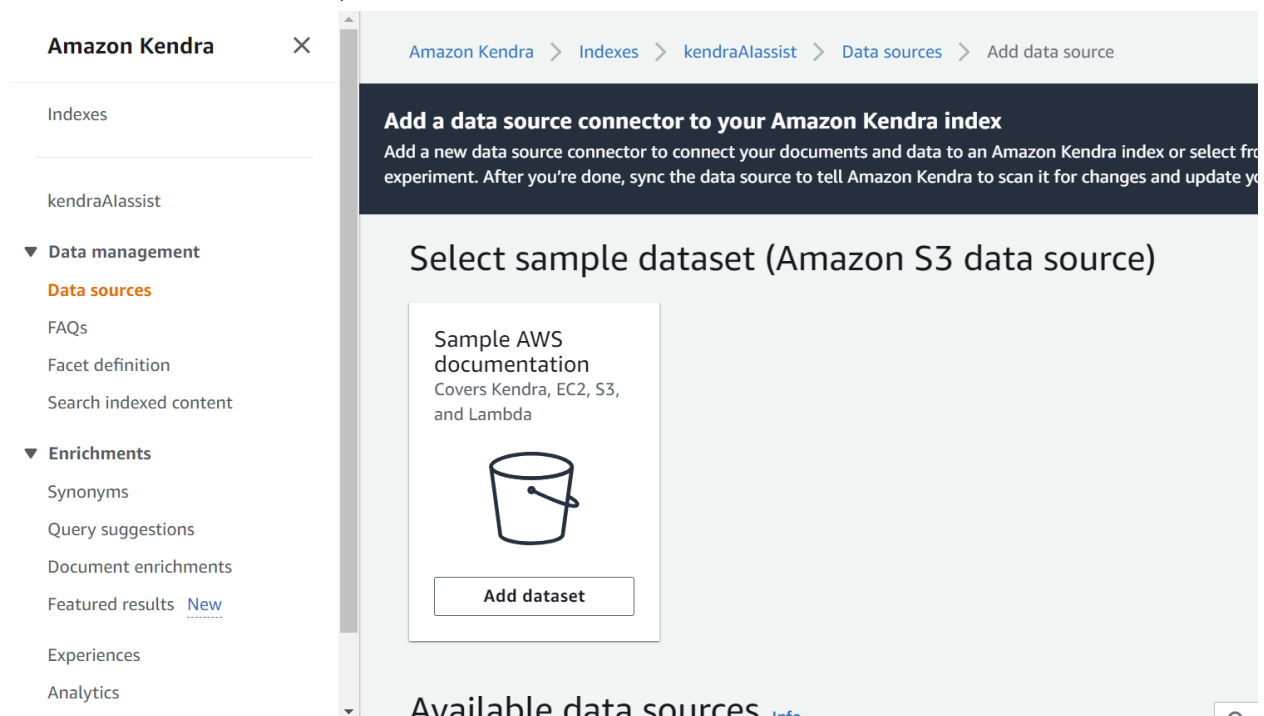
AWS Kendra



Creating a new Kendra Index, to use to search among the S3 company policies.



Once the index is created, we can add the data sources to connect to our S3 bucket



AWS Bedrock

Amazon Bedrock

✕

✓ Getting started

Overview

Examples

Providers

✓ Foundation models

Base models

Custom models

✓ Playgrounds

Text

Chat

Image

✓ Orchestration

Knowledge base

Agents

✓ Assessment & deployment

Model Evaluation

Preview

Amazon Bedrock > Overview

Overview

Explore & LearnBuild & Test

Foundation models

Amazon Bedrock supports foundation models from industry-leading providers. Choose the model that is best suited to achieving your unique goals.

AI21 labs

Jurassic-2 series

By AI21 Labs

a

Titan

By Amazon

AI

Claude

By Anthropic

Command

By Cohere

Llama 2

By Meta

S.

Stable Diffusion

By Stability AI

Spotlight

ANTHROPIC

Anthropic offers the Claude family of large language models purpose built for conversations, summarization, Q&A, workflow automation, coding and more.

Open in chat playground

<

>

For bedrock, we got access to Jupyter.