

[Event dashboard](#) > [Playground](#) > Invoking a model in Amazon Bedrock using AWS Step Functions

Invoking a model in Amazon Bedrock using AWS Step Functions

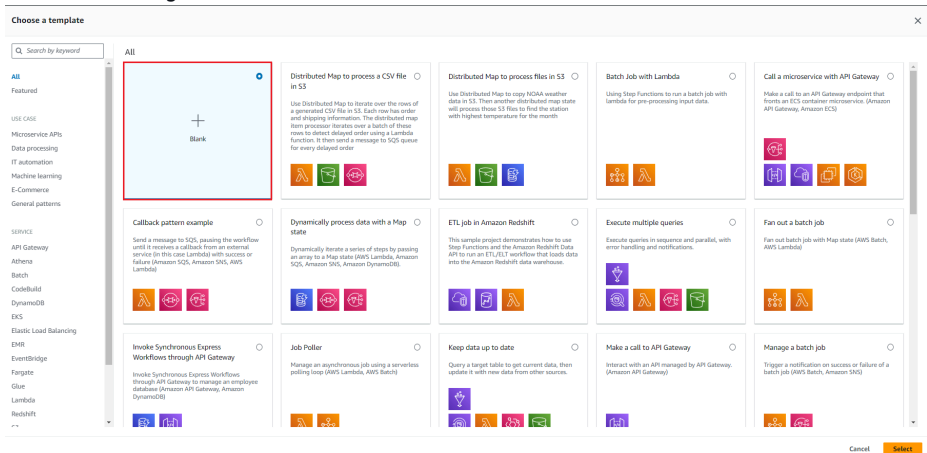
In this section, you will build a Step Function workflow to invoke an [Anthropic Claude](#) model in Amazon Bedrock and test in the Step Functions console. Then, you will run the same AWS Step Functions workflow to trigger the model from a user-facing AppSync API.

Create the Step Functions workflow

Console changes and new features!

Step Functions has launched a new feature to support JSONata and Variables on 11/22 along with console changes. While the new features simplify the creation of workflow, they are not integrated in to the workshop yet. Also you will see slight changes between the screenshots and the actual console.

1. Open the [Step Functions console](#)
2. Select **Create state machine** at the top right and choose the **Blank** template and choose **Select** on the bottom right.



3. You are now in the Step Functions visual designer studio.
4. Explore the visual designer studio. On the left, you can see tabs named **Actions**, **Flow**, and **Patterns**. You can search for actions using the search bar, drag and drop them in the graphical designer layout which is the middle part of the page. On the right of the page, you can see the options to configure the steps as you work on them.
5. In the top navigation bar, you will see tabs named **Design**, **Code**, and **Config**. You can edit the configurations of the workflow using these tabs.

Serverless x GenAI BKK Workshop

Generative AI at scale: Serverless workflows for enterprise-ready apps

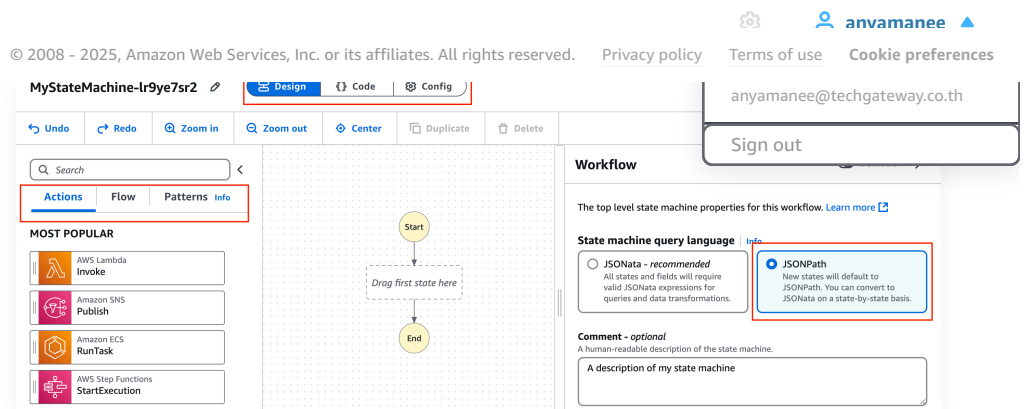
[Pre-requisite] Enable foundation model access in Amazon Bedrock

[Pre-requisite] Configuring the front-end application

Playground

[Invoking a model in Amazon Bedrock using AWS Step Functions](#)

Integrating an API with AWS Step Functions



Add Bedrock integration

1. In the left window, type **Bedrock InvokeModel** in the search box

Connecting a Client Application
to the Serverless GenAI Backend

(Optional) Deep Dive on AWS
AppSync

Summary

► Use cases

► Workshop Cleanup

▼ AWS account access

[Open AWS console](#)
(us-west-2) 

[Get AWS CLI credentials](#)

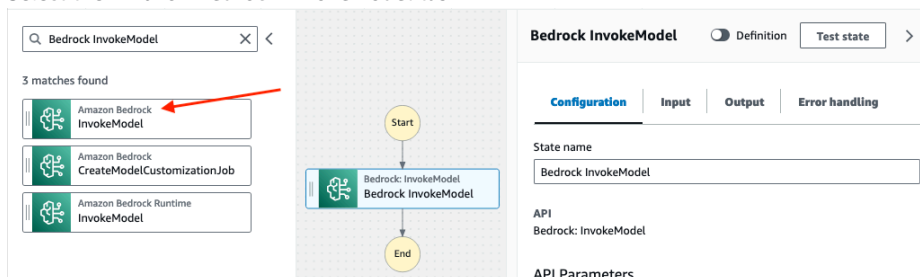
Exit event

2. Drag and drop the **InvokeModel** task under the **Amazon Bedrock** category to the designer. Notice how the search returned multiple results. *Amazon Bedrock Runtime InvokeModel* uses direct SDK/API integration, while the *Amazon Bedrock InvokeModel* uses an optimized integration. See the banner below for more information.

Do you know?

Step Functions integrates with AWS services in multiple ways. One common integration pattern is to invoke the AWS SDK API. Another integration pattern is the optimized call to the service with additional capabilities than offered in the SDK. For example, Bedrock optimized integration supports optionally retrieving and storing request/response data from S3. If you have use cases where you want to generate an image using LLM, you can use the native optimized integration to store the large image response from LLM in S3.

3. Select the **Amazon Bedrock InvokeModel** task



Search: Bedrock InvokeModel

3 matches found

- Amazon Bedrock InvokeModel
- Amazon Bedrock CreateModelCustomizationJob
- Amazon Bedrock Runtime InvokeModel

Bedrock InvokeModel

Configuration | Input | Output | Error handling

State name: Bedrock InvokeModel

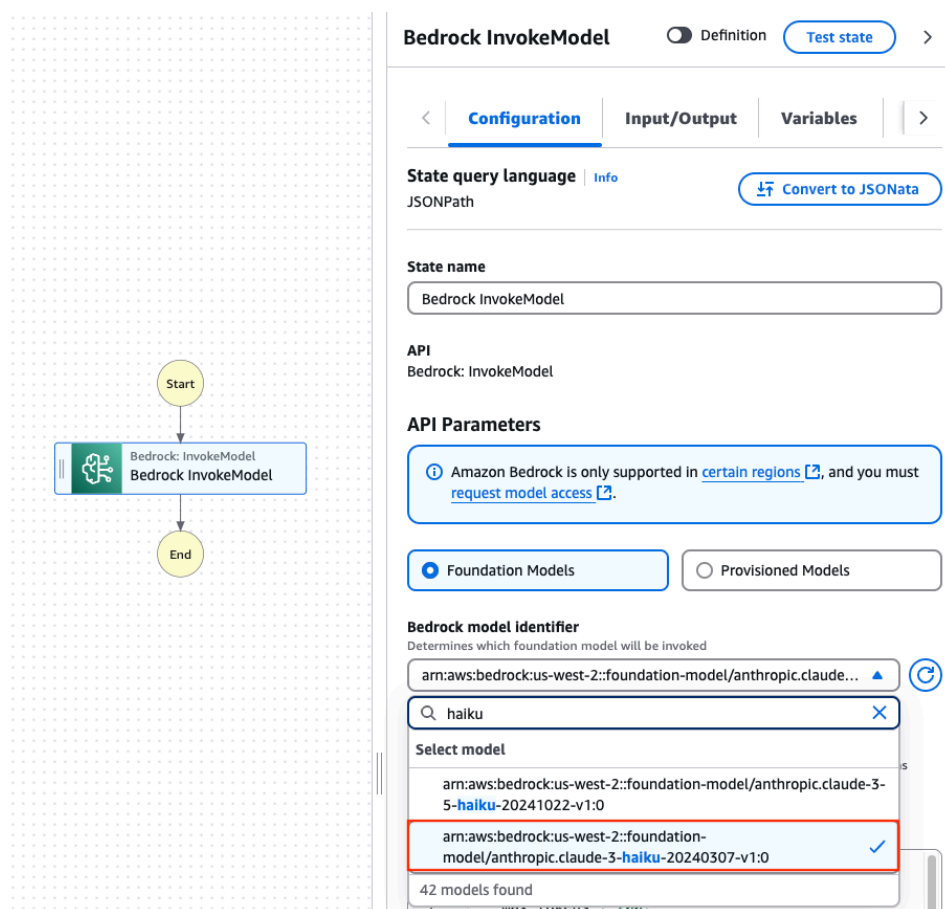
API: Bedrock: InvokeModel

API Parameters

Configure Bedrock Integration

1. Locate the configuration section on the right side of the studio. Type **haiku** under **Bedrock model identifier** and select **anthropic.claude-3-haiku**.

 Select **claude-3-haiku**, not **claude-3-5-haiku**.



Bedrock InvokeModel

Definition | Test state

Configuration | Input/Output | Variables

State query language: JSONPath

Convert to JSONata

State name: Bedrock InvokeModel

API: Bedrock: InvokeModel

API Parameters

Amazon Bedrock is only supported in [certain regions](#), and you must [request model access](#).

Foundation Models | Provisioned Models

Bedrock model identifier

Determines which foundation model will be invoked

am:aws:bedrock:us-west-2::foundation-model/anthropic.claude-3-haiku-20241022-v1:0

Search: haiku

Select model

am:aws:bedrock:us-west-2::foundation-model/anthropic.claude-3-5-haiku-20241022-v1:0

am:aws:bedrock:us-west-2::foundation-model/anthropic.claude-3-haiku-20240307-v1:0

42 models found

Important

As you are creating the workflow, you will see a red banner at the top saying **workflow not created**. You may ignore this warning as it will resolve once you save the workflow

2. Move down to **Bedrock Model Parameters** section and replace the text in the text area with the json below, then click **Save**.

i What is \$ for?

\$ symbol in the request below indicates that the value is evaluated at run time. text.\$ expects a runtime variable by the name prompt.

```

1  {
2    "anthropic_version": "bedrock-2023-05-31",
3    "max_tokens": 200,
4    "messages": [
5      {
6        "role": "user",
7        "content": [
8          {
9            "type": "text",
10           "text.$": "$.prompt"
11         }
12       ]
13     }
14   ]
15 }
```



Configure the State machine settings

1. Select the **Config** tab at the top of the State Machine visual studio
2. Name your State Machine ChatbotPlaygroundWorkflow
3. Select the Type as **Express**. You will be using this same state machine as a data source for a synchronous API call in a few steps. Express workflows are suitable for high burst and low latency use cases which can complete within 5 minutes, while Standard workflows can run for 365 days. Learn more about the difference [here](#) .
4. Select the IAM role starting with chatbotstack-PlaygroundSfnRole for your Step Function Execution Role. This role is pre-created with the policy to invoke the Bedrock model.
5. Review that your settings look as the screenshot below and select **Create**

ChatbotPlaygroundWorkflow [Design](#) [Code](#) [Config](#) Workflow not created [Cancel](#) [Actions](#) [Create](#)

State machine configuration [Feedback](#)

Details

State machine name
State machine name cannot be changed after creation.

Must be 1-80 characters. Can use alphanumeric characters, dashes, and underscores.

Type [Info](#)
State machine type cannot be changed after creation.

☐ **Standard**
Durable workflows for ETL, ML, e-commerce and automation. They can run for up to 1 year, and history is stored in Step Functions for auditing and playback. Supported by a feature-rich console debugger. Recommended for new users.

☒ **Express**
Low cost, high scale workflows for streaming data processing and microservice APIs. They can run for up to 5 minutes, and history can be streamed to CloudWatch Logs.

Permissions

Execution role
The IAM role that defines which resources your state machine has permission to access during execution. To create a custom role, go to the [IAM console](#).

[Refresh](#) [View in IAM](#)

Logging

You can log your state machine's execution history to CloudWatch Logs. For Express state machines, you must enable logging to inspect and debug executions. CloudWatch Logs charges apply. [Learn more](#)

Log level
Indicates which execution history events to log

☒ **Include execution data**
Log execution input, data passed between states, and execution output

CloudWatch log group

Maximum 512 alphanumeric characters. Can include hyphens, underscores, periods, and forward slashes

☐ **Encrypt log group with customer managed key - new** [Info](#)
Step Functions encrypts your new log group with the AWS KMS key that you choose.

Run the workflow

1. Select **Execute** from the next page. This will open a new Start Execution tab in your browser
2. Insert the below JSON in the input field

```
1 {
2   "prompt": "What is express workflow in AWS Step Functions?"
3 }
```



3. Select **Start execution**
4. Select the **Bedrock InvokeModel** task in the execution and explore the **Input/Output** tab on the right side.
5. If you do not see any input, turn on **Advanced view**

Graph view [Actions](#)

Bedrock InvokeModel [Test state](#)

bedrock:invokeModel

[Input/Output](#) [Variables](#) [Details](#) [Definition](#) [Events](#)

☐ Advanced view

State input [Learn more](#)

```
1 {
2   "prompt": "What is express workflow in AWS Step Functions?"
3 }
```

State output [Learn more](#)

```
1 {
2   "Body": {
3     "id": "msg_bdrk_01MXAZFvczjH7BhaMCF2mJM7",
4     "type": "message",
5     "role": "assistant",
6     "model": "claude-3-haiku-20240307",
7     "content": [
8       {
9         "type": "text",
```

[Formatted](#) [Copy](#)

[In progress](#) [Failed](#) [Caught error](#) [Canceled](#) [Succeeded](#)

6. You may now close this browser tab to return to the State Machine design editor

✔ **Congratulations!**

You have successfully invoked a foundation model in Amazon Bedrock using Step Functions.

[Previous](#)

[Next](#)