

## ユーザーガイド

# **Amazon Bedrock**



Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

## Amazon Bedrock: ユーザーガイド

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに 関連して使用してはなりません。Amazon が所有しない商標はすべてそれぞれの所有者に所属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとは限りません。

## **Table of Contents**

Amazon Bedrock とは	1
Amazon Bedrock モデルにアクセスする	1
Amazon Bedrock の機能	1
Amazon Bedrock でサポートされているモデル	3
サポートされるリージョン	5
Amazon Bedrock の料金	5
セットアップする	6
モデルアクセスを追加する	6
コンソールアクセス	6
AWS アカウントにサインアップする	6
管理ユーザーを作成する	7
AWS CLI をセットアップする	8
プログラム的なアクセス権を付与する	8
モデルアクセス	. 11
モデルアクセスを管理する	6
モデルアクセスを編集する	. 11
モデルアクセスを追加する	. 11
モデルアクセスを削除する	. 12
設定	. 14
モデル呼び出しのログ記録	. 14
Amazon S3 送信先をセットアップする	. 15
CloudWatch Logs 送信先を設定する	. 16
コンソールを使用する場合	. 18
呼び出しのログ記録で API を使用する	. 19
コンソールを使用する	. 20
ベースモデル	
モデルを表示したりグループ化したりする	. 21
モデルのフィルタリングと検索	. 21
モデルプロバイダーの詳細を表示する	. 22
ベースモデルへのアクセスをリクエストする	22
テキストのプレイグラウンド	22
テキストモデルにプロンプトを送信する	. 23
プロンプトと生成されたテキストを操作する	24
クエリの再実行	. 24

高度なプロンプトを使用する	. 25
モデルパラメータを調整する	. 25
別のモデルを選択する	25
チャットのプレイグラウンド	26
チャットモデルにプロンプトを送信する	. 26
チャットの指示を更新する	. 27
高度なプロンプトを使用する	. 27
推論の設定を調整する	28
別のモデルを選択する	28
イメージのプレイグラウンド	28
モデルにプロンプトを送信する	. 28
モデルパラメータを調整する	. 29
サンプルライブラリ	. 29
API を使用する	. 31
API のセットアップ	. 32
Amazon Bedrock エンドポイント	. 32
AWS CLI のセットアップ	. 32
AWS SDK のセットアップ	. 32
SageMaker ノートブックを使用する	. 33
API オペレーション	. 35
ベースモデルのリスト表示	35
ベースモデルの詳細を取得する	. 36
推論の実行	36
リソースのタグ付け	
推論パラメータ	
推論パラメータの定義	
ランダム性と多様性	. 47
長さ	
繰り返し	
Amazon Titan モデル	
ランダム性と多様性	
長さ	
モデル呼び出しリクエストの body フィールド	
モデル呼び出しレスポンスの body フィールド	
Anthropic Claude モデル	
ランダハ性と多様性	52

	長さ	. 52
	モデル呼び出しリクエストの body フィールド	. 53
	モデル呼び出しレスポンスの body フィールド	. 53
	Al21 Labs Jurassic-2 モデル	. 53
	ランダム性と多様性	. 54
	長さ	. 54
	繰り返し	. 54
	モデル呼び出しリクエストの body フィールド	. 55
	モデル呼び出しレスポンスの body フィールド	. 56
	Cohere モデル	. 57
	Cohere Command モデル	57
	Cohere Embed モデル	. 61
	Meta Llama 2 Chat 13B モデル	. 63
	ランダム性と多様性	. 63
	長さ	. 63
	モデル呼び出しリクエストの body フィールド	. 63
	モデル呼び出しレスポンスの body フィールド	. 64
	Stability.ai Diffusion モデル	
	モデル呼び出しリクエストの body フィールド	
	モデル呼び出しレスポンスの body フィールド	
゚゚ヿ	コンプトエンジニアリングガイドライン	67
	序章	
	プロンプトとは	
	プロンプトのコンポーネント	
	数ショットプロンプトとゼロショットプロンプト	
	プロンプトテンプレート	
	Amazon Bedrock LLM を API コールで使用する際の重要な注意事項	
	プロンプトエンジニアリングとは	
	Amazon Bedrock LLM ユーザー向けの一般的なガイドライン	
	プロンプトを設計する	
	ハイパーパラメータの使用	
	詳細なガイドライン	
	Amazon Bedrock のテキストモデル用にプロンプトを最適化する - 基本だけでは不十分な場	
	合	
	Amazon Bedrock テキストモデルのプロンプトテンプレートと例	
	テキスト分類	. 86

質問応答 (コンテキストなし)	89
質問応答 (コンテキストあり)	92
要約	97
テキスト生成	99
コードの生成	101
数学	104
推論/ロジカルシンキング	105
埋め込み	108
カスタムモデル	109
微調整	109
データセットを準備する	110
コンソールの使用	111
ジョブを送信する	111
ジョブのモニタリング	113
ジョブを停止する	114
ジョブ結果を分析する	114
微調整されたモデルを推論に使用する	115
API を使用する	116
モデルカスタマイズの IAM ロールを設定する	116
ジョブを送信する	120
ジョブを監視する	124
ジョブを停止する	125
ジョブを分析する	126
カスタマイズモデルに関する情報を取得する	127
ガイドライン	129
入力トレーニングデータセットのサイズ	129
モデルサイズ	129
エポック	129
バッチサイズ	130
学習率	130
学習率のウォームアップステップ	130
トラブルシューティング	130
アクセス許可の問題	130
データの問題	131
内部エラー	132
プロビジョンドスループット	133

モデル ID	
手順	135
[作成中]	135
[更新中]	135
[削除中]	136
推論を実行する	136
アクセス許可	136
コンソール手順	136
プロビジョンドスループットの概要を表示する	137
プロビジョンドスループットを購入	137
プロビジョンドスループットの詳細を表示する	138
プロビジョンドスループットを編集	139
プロビジョンドスループットを削除する	139
API オペレーション	140
プロビジョンドスループットを作成する	140
プロビジョンドスループットを使用して推論を実行する	142
プロビジョンドスループットを更新する	143
プロビジョンドスループットを取得する	144
プロビジョンドスループットを削除する	144
プロビジョンドスループットリソースをリスト表示する	145
Agents for Amazon Bedrock	146
ナレッジベースを構築する	147
サービスロールを作成し、IAM アクセス許可を設定する	149
取り込み対象のデータを設定する	155
ナレッジベースを作成する	161
ナレッジベースを管理する	164
ナレッジベースをエージェントに追加する	165
エージェントの構築	166
サービスロールを作成し、IAM アクセス許可を設定する	167
エージェントを作成する	171
エージェントを編集する	177
エージェントをテストする	179
トレースの有効化	180
エージェントのデプロイ: バージョニングとエイリアス	184
API を使用する場合	186
T ― ジェントを呼び出す	186

Bedrock Agent で IAM が機能する仕組み	187
アイデンティティベースのポリシー	188
リソースベースのポリシー	189
ポリシーアクション	190
ポリシーリソース	191
ポリシー条件キー	191
ACL	192
ABAC	192
一時的な認証情報	193
プリンシパル許可	193
サービスロール	194
サービスリンクロール	194
Bedrock Agent のアイデンティティベースのポリシー例	195
リソースのタグ付け	201
コンソールを使用する	201
API を使用する	202
タグの制限	202
セキュリティ	204
データ保護	205
データ暗号化	207
VPC を使用する	211
Identity and Access Management	217
対象者	218
アイデンティティを使用した認証	219
ポリシーを使用したアクセスの管理	222
Amazon Bedrock で IAM が機能する仕組み	
アイデンティティベースポリシーの例	
サービスロール	
トラブルシューティング	
コンプライアンス検証	241
インシデントへの対応	
耐障害性	
インフラストラクチャセキュリティ	243
サービス間の混乱した代理の防止	
Amazon Bedrock での設定と脆弱性の分析	245
Amazon Bedrock をモニタリングする	246

CloudWatch を使用してモニタリングする	246
ランタイムメトリクス	246
ログへの CloudWatch メトリクスの記録	247
Amazon Bedrock 用に CloudWatch メトリクスを使用する	248
Amazon Bedrock メトリクスを表示する	248
イベントのモニタリング	249
使用方法	250
EventBridge スキーマ	250
ルールとターゲット	252
AWS Bedrock イベントを処理するルールを作成する	252
CloudTrail ログ	254
CloudTrail 内の Amazon Bedrock 情報	254
Amazon Bedrock ログファイルエントリの概要	255
不正検出	257
AWS PrivateLink	258
考慮事項	258
インターフェイスエンドポイントの作成	259
エンドポイントポリシーを作成する	259
クォータ	261
ランタイムクォータ	261
モデルカスタマイズのクォータ	262
モデルクォータ	262
微調整用のクォータ	263
トレーニングのクォータ	263
プロビジョンドスループットのクォータ	264
ドキュメント履歴	265
Δ\M/Q 田鈺隹	267

### Amazon Bedrock とは

Amazon Bedrock は、Amazon やサードパーティーのモデルプロバイダーのベースモデルに API から アクセスできるようにする完全マネージド型サービスです。

#### トピック

- Amazon Bedrock モデルにアクセスする
- Amazon Bedrock の機能
- Amazon Bedrock でサポートされているモデル
- サポートされるリージョン
- Amazon Bedrock の料金

### Amazon Bedrock モデルにアクセスする



#### ♠ Important

モデルを使用するには、まずモデルへのアクセスをリクエストする必要があります。モデル へのアクセスをリクエストしていないのに、(API またはコンソールで) モデルを使用しよう とすると、エラーメッセージが表示されます。詳細については、「モデルアクセス」を参照 してください。

### Amazon Bedrock の機能

Amazon Bedrock には以下の機能があります。

- Text playground AWS Management Consoleを使用した実践的なテキスト生成アプリケーショ ン。
- Image playground コンソールを使用した実践的なイメージ生成アプリケーション。
- Chat playground コンソールを使用した実践的な会話生成アプリケーション。
- Examples library ロードされるサンプルユースケース。
- Amazon Bedrock API AWS CLI で調べることができます。また、この API を使用してベースモデ ルにアクセスできます。
- Embeddings API を使用して Titan Embeddings G1 Text モデルから埋め込みを生成できます。

• Provisioned Throughput - スループットを購入すると、モデルでの推論を割引料金で実行できます。

#### Note

Note

現在、プロビジョンドスループットは以下のモデルで使用できます。

モデル名	プロビジョンドスループットが使用可 能なモデル ID
Titan Text G1 - Express 8K	amazon.titan-text-express-v1:0:8k
Titan Embeddings G1 - Text	amazon.titan-embed-text-v1:2:8k
Anthropic Claude V2 18K	anthropic.claude-v2:0:18k
Anthropic Claude V2 100K	anthropic.claude-v2:0:100k
Anthropic Claude Instant V1 100K	anthropic.claude-instant-v1:2:100K
Stable Diffusion XL 1.0	stability.stable-diffusion-xl-v1:0
Meta Llama 2 Chat 13B	meta.llama2-13b-chat-v1:0:4k

#### Note

Stable Diffusion XL 1.x はプロビジョンドスループットでのみ利用可能です。詳細については、「プロビジョンドスループット」を参照してください。Stable Diffusion XL 0.x はプレビューリリースです。

• Fine-tuning - トレーニングデータセットを作成し、Amazon Bedrock モデルを微調整します。

Amazon Bedrock の機能



#### Note

微調整を行うには、Amazon Titan Text G1-Express モデルにアクセスできる必要がありま す。Amazon Titan Text G1 - Express モデルは限定プレビューです。ご利用をリクエスト するには、AWS アカウントマネージャーに連絡してください。

• Model invocation logging- Amazon Bedrock で使用している AWS アカウント の全呼び出しの呼び 出しログ、モデル入力データ、モデル出力データを収集できます。



#### Note

Amazon Bedrock では、モデル呼び出しのログ記録はプレビューリリースに含まれている ため、変更される可能性があります。

以下の機能は、限定プレビューリリースに同梱されています。ご利用をリクエストするには、AWS アカウントマネージャーに連絡してください。

- Agents for Amazon Bedrock 自動的にオーケストレーションやタスクを実行するエージェントを 構築します。
- Knowledge base for Amazon Bedrock データソースから情報を引き出すことで、エージェントが ユーザーのお客様向けの情報を見つけられるようにします。

### Amazon Bedrock でサポートされているモデル

Amazon Bedrock モデルプロバイダー、そのモデル、およびモデル ID の詳細については、Amazon Bedrock コンソールの [Base models] ページを参照してください。また、モデルの最新リストに関す る情報を取得するには、ListFoundationModels API オペレーションを使用します。

Amazon Bedrock は、以下のモデルをサポートしています。

- Al21 Labs
  - Jurassic-2 Ultra
  - · Jurassic-2 Mid
- Amazon
  - Titan Text G1 Express

ユーザーガイド Amazon Bedrock

Titan Embeddings G1 - Text



#### Note

Amazon Titan Text G1 - Express モデルは限定プレビューリリースに同梱されています。 アクセス許可は今後も付与されます。

- · Anthropic
  - Claude v1.x
  - Claude v2.x
  - Claude Instant v1.x
- Cohere
  - Command v14.7
  - Command Light v14.7
  - Embed English v3.0
  - Embed Multilingual v3.0
- Meta
  - Llama 2 Chat 13B
- Stability.ai
  - Stable Diffusion XL 0.x
  - Stable Diffusion XL 1.x



Stable Diffusion XL 1.x はプロビジョンドスループットでのみ利用可能です。詳細について は、「プロビジョンドスループット」を参照してください。Stable Diffusion XL 0.x はプレ ビューリリースです。

これらのモデルの詳細については、以下のリンクを参照してください。

- 「Anthropic の迅速なエンジニアリング リソースのガイド」
- 「Al21 Studio Documentation」
- The Cohere Platform

- 「Getting started with Llama」
- 「Getting Started」 (Stability.ai)

### サポートされるリージョン

Amazon Bedrock は、次の AWS リージョンで利用できます。

- ・ 米国東部 (バージニア北部)
- 米国西部 (オレゴン)
- アジアパシフィック (シンガポール)
- ・ アジアパシフィック (東京)
- 欧州 (フランクフルト)

### Amazon Bedrock の料金

AWS にサインアップすると、Amazon Bedrock を含む AWS のすべてのサービスに対して、AWS アカウントが自動的にサインアップされます。ただし、料金が発生するのは、実際に使用したサービスの分のみです。

請求を表示するには、AWS Billing and Cost Management コンソールで請求およびコスト管理ダッシュボードに移動します。AWS アカウント 請求の詳細については、AWS Billing ユーザーガイドを参照してください。AWS 請求および AWS アカウント についてご質問がある場合は、AWS Support にお問い合わせください。

Amazon Bedrock では、どのサードパーティーの基盤モデルで推論を実行する場合でも料金がかかります。料金は、入力トークンと出力トークンの量、およびモデル用のプロビジョンドスループットの購入有無によって決まります。詳細については、Amazon Bedrock コンソールの [モデルプロバイダー] ページを参照してください。各モデルの料金は、モデルバージョンの後に記載されています。プロビジョンドスループットの購入の詳細については、「プロビジョンドスループット」を参照してください。

詳細については、「Amazon Bedrock の料金体系」ページを参照してください。

サポートされるリージョン 5

## Amazon Bedrock をセットアップする

Amazon Bedrock を初めて使用する場合は、まず以下のタスクを完了してください。

#### セットアップタスク

- モデルアクセスを追加する
- コンソールアクセス
- AWS アカウントにサインアップする
- 管理ユーザーを作成する
- AWS CLI をセットアップする
- プログラム的なアクセス権を付与する

### モデルアクセスを追加する



#### Important

モデルを使用するには、まずモデルへのアクセスをリクエストする必要があります。モデル へのアクセスをリクエストしていないのに、(API またはコンソールで) モデルを使用しよう とすると、エラーメッセージが表示されます。詳細については、「モデルアクセス」を参照 してください。

### コンソールアクセス

Amazon Bedrock コンソールとプレイグラウンドにアクセスするには:

- 1. AWS アカウントにサインインします。
- 2. Amazon Bedrock コンソールに移動します。

なお、Amazon Bedrock コンソールには、米国東部 (バージニア北部)、米国西部 (オレゴン)、アジ アパシフィック (シンガポール) からアクセスできます。

### AWS アカウントにサインアップする

AWS アカウントがない場合は、以下のステップを実行して作成します。

モデルアクセスを追加する

#### AWS アカウントにサインアップするには

- 1. https://portal.aws.amazon.com/billing/signup を開きます。
- 2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話のキーパッドを用いて検証コードを入力するように求められます。

AWS アカウント にサインアップすると、AWS アカウントのルートユーザー が作成されます。 ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があ ります。セキュリティのベストプラクティスとして、<u>管理ユーザーに管理アクセスを割り当て</u>、 ルートユーザーのみを使用して<u>ルートユーザーアクセスが必要なタスク</u>を実行してください。

サインアップ処理が完了すると、AWS からユーザーに確認メールが送信されます。<u>https://</u> <u>aws.amazon.com/</u> の [アカウント] をクリックして、いつでもアカウントの現在のアクティビティを 表示し、アカウントを管理することができます。

## 管理ユーザーを作成する

AWS アカウント にサインアップした後、日常的なタスクにルートユーザーを使用しないように、管理ユーザーを作成します。

AWS アカウントのルートユーザー をセキュリティで保護する

 [ルートユーザー] を選択し、AWS アカウント のメールアドレスを入力して、アカウント所有者 として AWS Management Console にサインインします。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「ルートユーザーとしてサインインする」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、IAM ユーザーガイドの「<u>AWS アカウント のルートユーザーの仮想 MFA デバ</u>イスを有効にする (コンソール)」を参照してください。

管理ユーザーを作成する 7

#### 管理ユーザーを作成する

日常的な管理タスクのためには、AWS IAM Identity Center の管理ユーザーに管理アクセスを割り当てます。

手順については、AWS IAM Identity Center ユーザーガイドの<u>「開始方法」</u>を参照してください。

#### 管理ユーザーとしてサインインする

• IAM Identity Center ユーザーとしてサインインするには、IAM Identity Center ユーザーの作成時 にEメールアドレスに送信されたサインイン URL を使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「AWS アクセスポータルにサインインする」を参照してください。

### AWS CLI をセットアップする

Amazon Bedrock を使用するのに AWS Command Line Interface (AWS CLI) は必要ありません。必要に応じて、このステップをスキップして、後で AWS CLI をセットアップできます。

#### AWS CLI をインストールして設定するには

- AWS CLI をインストールします。手順については、「AWS Command Line Interface ユーザーガイド」にある「AWS Command Line Interface の最新バージョンのインストールまたは更新」を参照してください。
- 2. AWS CLI を設定します。手順については、「AWS Command Line Interface ユーザーガイド」の「Configuring the AWS Command Line Interface」を参照してください。

### プログラム的なアクセス権を付与する

AWS Management Console の外部で AWS を操作するには、ユーザーはプログラムによるアクセスが必要です。プログラムによるアクセスを許可する方法は、AWS にアクセスしているユーザーのタイプによって異なります。

ユーザーにプログラムによるアクセス権を付与するには、以下のいずれかのオプションを選択します。

AWS CLI をセットアップする

どのユーザーがプログラムに よるアクセスを必要とします か?	目的	方法
ワークフォース ID (IAM Identity Center で管理されているユーザー)	一時的な認証情報を使用して、AWS CLI、AWS SDK、または AWS API へのプログラムによるリクエストに署名します。	使用するインターフェイス用の手順に従ってください。  ・ AWS CLI については、 「AWS Command Line Interface ユーザーガイド」の「AWS IAM Identity Center を使用するためのAWS CLI の設定」を参照してください。  ・ AWS SDK、ツール、および AWS API については、「AWS SDK とツールリファレンスガイド」の「IAM Identity Center 認証」を参照してください。
IAM	一時的な認証情報を使用して、AWS CLI、AWS SDK、または AWS API へのプログラムによるリクエストに署名します。	IAM ユーザーガイドの「AWS リソースでの一時的な認証情 報の使用」の指示に従ってく ださい。
IAM	(非推奨) 長期的な認証情報を使用 して、AWS CLI、AWS SDK、AWS API へのプログラ ムによるリクエストに署名し ます。	使用するインターフェイス用の手順に従ってください。  ・ AWS CLI については、 「AWS Command Line Interface ユーザーガイド」の「IAM ユーザー認証情報を使用した認証」を参照してください。

どのユーザーがプログラムに よるアクセスを必要とします か?	目的	方法
		<ul> <li>AWS SDK とツールについては、「AWS SDK とツールリファレンスガイド」の「長期認証情報を使用して認証する」を参照してください。</li> <li>AWS API については、「IAM ユーザーガイド」の</li> </ul>
		「IAM ユーザーのアクセス キーの管理」を参照してく ださい。

## モデルアクセス

Amazon Bedrock ユーザーがモデルを使用するには、まずそのモデルへのアクセスをリクエストする必要があります。テキスト、チャット、イメージを生成するための新しいモデルを追加する場合は、Amazon Bedrock のモデルへのアクセスをリクエストする必要があります。新しいモデルへのアクセスをリクエストするには、Amazon Bedrock コンソールの左側のナビゲーションパネルで [モデルアクセス] リンクを選択します。

#### トピック

- モデルアクセスを管理する
- モデルアクセスを編集する
- モデルアクセスを追加する
- モデルアクセスを削除する

### モデルアクセスを管理する

このアカウントは、デフォルトではモデルへのアクセス許可を持っていません。IAM アクセス許可を持つ管理者ユーザーは、 [モデルアクセス]ページを使用して特定のモデルへのアクセス許可を追加できます。管理者がモデルへのアクセス許可を追加すると、そのモデルは管理者アカウントのすべてのユーザーがアクセスできるようになります。

モデルを Amazon Bedrock で使用すると、料金が発生します。各モデルのリンクを選択すると、エンドユーザーライセンス契約 (EULA) を確認できます。

モデルアクセスを追加または削除するには、[モデルアクセスを管理] を選択します。

### モデルアクセスを編集する

Amazon Bedrock でモデルを使用するための、モデルへのアクセス許可をリクエストすることができます。本ページでは、LLM のベースモデルからテキスト、イメージ、埋め込みモデルに至る、モデルの一覧を示します。どのモデルにアクセスできるかを確認するには、ステータスを確認します。モデルがテキスト、埋め込み、イメージのどれに使用されているかは、[モダリティ] に示されます。

### モデルアクセスを追加する

Amazon Bedrock のモデルへのアクセス許可を追加するには、以下の手順を実行します。

モデルアクセスを管理する 11

1. 「Amazon Bedrock コンソール」をクリックして、 Amazon Bedrock コンソールを開きます。

- 2. Amazon Bedrock の左側のナビゲーションパネルにある [モデルアクセス] リンクに移動する か、[モデルアクセスを編集] ページに移動します。
- 3. アクセス許可を追加したいモデルの横にあるチェックボックスをオンにします。Anthropic モデルの場合は、さらに、[アクセスをリクエスト] ボタンをクリックしてアクセスをリクエストすることも必要です。Amazon Bedrock では、デフォルト設定としてモデルはアクセス可能にはなっていません。
- 4. [確認] を選択することで、Amazon Marketplace を介してサードパーティーモデルへのアクセス 許可を追加します。注意: Amazon Bedrock およびそのモデルへのアクセスには、販売者の価格 条件、EULA、および Amazon Bedrock サービス利用規約が適用されます。
- 5. ページの右下にある [変更を保存] ボタンを選択します。[モデルアクセス] ページへの変更を保存 するのに数分間かかることがあります。
- 6. アクセス許可が付与されている場合、[モデルアクセス] ページの [アクセス] のステータスで、モデルは [利用可能] と表示されます。

ユーザーによっては、モデルへのアクセス許可を追加するための IAM アクセス許可が付与されていない場合があります。アカウントの管理者以外のユーザーがモデルへのアクセス権を追加しようとすると、バナーメッセージが表示されます。Amazon Bedrock でモデルを利用するには、まずアカウント管理者に連絡して、モデルへのアクセス許可を追加するようリクエストする必要があります。

モデルへのアクセス許可が付与されたら、[モデルアクセスを表示] ボタンまたは [モデルアクセスを編集] ページを使用してモデルを選択できます。アカウントは、デフォルトでは Amazon モデルへのアクセス許可を付与されていません。

### モデルアクセスを削除する

Amazon Bedrock を使用する場合、特定のモデルのみを作業に使用したり、使用しなくなったモデルへのアクセス許可を削除したりすることもできます。Amazon Bedrock のモデルへのアクセス許可を削除するには、以下の手順を実行します。

- 1. Amazon Bedrock コンソール (<a href="https://us-east-1.console.aws.amazon.com/bedrock/home?">https://us-east-1.console.aws.amazon.com/bedrock/home?</a> region=us-east-1#/) を開きます。
- 2. Amazon Bedrock の左側のナビゲーションパネルにある [モデルアクセス] リンクに移動する か、[モデルアクセスを編集] ページに移動します。
- 3. アクセス許可を削除したいモデルの横にあるチェックボックスをオンにします。

モデルアクセスを削除する 12

- 4. ページの右下にある [変更を保存] ボタンを選択します。
- 5. モデルのアクセス許可を削除したいかどうかを確認するよう求められます。モデルを使用しているリソースがある場合、そのモデルへのアクセス許可を削除することはできません。
- 6. アクセス許可を削除するには、情報ウィンドウに表示されているモデルへの関連付けを削除する 必要があります。リソース名の横にあるリンクを選択して、リソースのある場所に移動し、その リソースのモデルへのアクセス許可を削除できます。依存リソースとしてリスト表示されている 項目ごとに、このステップを繰り返します。
- 7. リスト表示されているすべてのリソースへのアクセス許可を削除したら、[OK] を選択してアクセス許可削除のステップを完了します。
- 8. 完了すると、アクションが正常に完了したことを示すバナーメッセージが表示されます。

ー モデルアクセスを削除する 13

## 設定

[設定] ページで、アカウントレベルの Amazon Bedrock 設定を管理できます。この設定としては、 データログ、データアクセス許可、モデルアクセスが含まれます。

設定にアクセスするには、Amazon Bedrock の左側にあるナビゲーションペインの下部に移動 し、[設定] を選択します。

#### トピック

• モデル呼び出しのログ記録

### モデル呼び出しのログ記録

[モデル呼び出しのログ記録] を使用すると、Amazon Bedrock で使用している AWS アカウント の全呼び出しの呼び出しログ、モデル入力データ、およびモデル出力データを収集できます。ログ記録はデフォルトでは無効になっています。

呼び出しのログ記録を使用すると、アカウントで実行したすべての呼び出しに関連する完全なリクエストデータ、応答データ、およびメタデータを収集できます。ログ記録は、ログデータが公開される送信先リソースを出力するように設定できます。サポートされている送信先としては、Amazon CloudWatch Logs および Amazon Simple Storage Service (Amazon S3) があります。同じアカウントとリージョンの送信先のみがサポートされます。

呼び出しのログ記録を有効にするには、Amazon S3 または CloudWatch Logs の送信先をセットアップする必要があります。呼び出しのログ記録は、コンソールまたは API を使用して有効にできます。

#### トピック

- Amazon S3 送信先をセットアップする
- CloudWatch Logs 送信先を設定する
- コンソールを使用する場合
- 呼び出しのログ記録で API を使用する

- モデル呼び出しのログ記録 14

#### Amazon S3 送信先をセットアップする

Amazon Bedrock にログインするために S3 送信先をセットアップするには、以下の手順に従います。

- 1. ログが配信される S3 バケットを作成します。
- 2. ################# (accountId#region#bucketName############ prefix を 値に置き換えます)。

#### Note

アクセス許可 S3:GetBucketPolicy と S3:PutBucketPolicy を使用してログ記録を設定すると、ユーザーに代わって自動的にバケットポリシーがバケットに追加されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
      "Sid": "AmazonBedrockLogsWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketName/prefix/AWSLogs/accountId/
BedrockModelInvocationLogs/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
           "aws:SourceArn": "arn:aws:bedrock:region:accountId:*"
        }
      }
    }
```

```
}
```

3. (オプション) バケットで SSE-KMS を設定する場合は、KMS キーで以下のポリシーを追加します。

S3 SSE-KMS 設定の詳細については、「<u>AWS KMS (SSE-KMS) によるサーバー側の暗号化の指定</u>」 を参照してください。

#### Note

バケットポリシーを有効にするには、バケット ACL を無効にする必要があります。詳細については、「<u>すべての新しいバケットの ACL を無効にし、オブジェクト所有権を執行しま</u>す。」を参照してください。

## CloudWatch Logs 送信先を設定する

Amazon Bedrock にログインするために Amazon CloudWatch Logs をセットアップするには、以下の手順に従います。

1. ログを公開する CloudWatch ロググループを作成します。

2. 以下の CloudWatch Logs アクセス許可を持つ IAM ロールを作成します。

#### 信頼されたエンティティ:

```
"Version": "2012-10-17",
  "Statement": [
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:bedrock:region:accountId:*"
        }
      }
    }
  ]
}
```

#### ロールポリシー:

CloudWatch Logs に対する SSE の設定の詳細については、「<u>AWS Key Management Service を使用</u> して CloudWatch Logs のログデータを暗号化する」を参照してください。

#### コンソールを使用する場合

モデル呼び出しのログ記録を有効にするには、[設定] ページの [ログ記録] トグルスイッチの横にあるスライダーボタンをドラッグします。ログ記録用の追加構成設定がパネルに表示されます。

ログに公開するデータリクエストとレスポンスを選択します。以下の出力オプションの任意の組み合わせを選択することもできます。

- ・テキスト
- イメージ
- 埋め込み

ログを公開する場所を選択します。

- ・ Amazon S3 のみ
- ・ CloudWatch Logs のみ
- Amazon S3 と CloudWatch Logs の両方

送信先として、Amazon S3 と CloudWatch Logs は、呼び出しのログ記録と小さな入出力データに対応しています。大量の入出力データやバイナリイメージ出力には、Amazon S3 のみが対応しています。以下の詳細は、ターゲット送信先でのデータの表示方法をまとめたものです。

・送信先: S3 - 指定された S3 バケットには、Gzip 圧縮された JSON ファイルが配信されます。各ファイルには、呼び出しのログレコードのバッチが含まれています。CloudWatch Logs イベントと同様に、各レコードには呼び出しメタデータ、および入出力 JSON 本文 (最大サイズ 100 KB) が含まれます。100 KB を超えるバイナリデータまたは JSON 本文は、指定された Amazon S3 バケットに、データプレフィックスを付けた個別のオブジェクトとしてアップロードされます。データは Amazon S3 Select と Amazon Athena を使用してクエリできるほか、AWS Glue を使用してETL 対象としてカタログ化することもできます。データは OpenSearch サービスにロードすることも、任意の Amazon EventBridge ターゲットによって処理することもできます。

コンソールを使用する場合 18

送信先: CloudWatch Logs - JSON 呼び出しのログ記録イベントは、CloudWatch Logs 内の指定されたロググループに配信されます。ログイベントでは、呼び出しメタデータと、最大サイズ 100 KB の入出力 JSON 本文が対象となります。大量データ配信用の送信先として Amazon S3 が提供されている場合、100 KB を超えるバイナリデータまたは JSON 本文は、その代わりにデータプレフィックスを付けたものが Amazon S3 バケットにアップロードされます。データは CloudWatch Logs Insights を使用してクエリできるほか、CloudWatch Logs を使用してさまざまなサービスにリアルタイムでストリーミングすることもできます。

### 呼び出しのログ記録で API を使用する

モデル呼び出しのログ記録は次の API を使用して設定できます。

- PutModelInvocationLoggingConfiguration
- GetModelInvocationLoggingConfiguration
- DeleteModelInvocationLoggingConfiguration

呼び出しのログ記録で API を使用する方法の詳細については、「Bedrock API ガイド」を参照してください。

## Amazon Bedrock コンソールを使用する

Amazon Bedrock には、以下の機能と特徴があります。

• [基盤モデル] の下にある [ベースモデル] ページでは、利用可能なモデルを表示したりさまざまな属性別にグループ化したりできます。また、モデルビューのフィルタリング、モデルの検索、およびモデルプロバイダーに関する情報の表示も行うことができます。

- モデルを選択したら、そのモデルを開いて、[テキストのプレイグラウンド]、[イメージのプレイグラウンド]、または [チャットのプレイグラウンド] で試してみることができます。
- Amazon Bedrock には、サポートされている各モデルのプロンプトの例が用意されています。[例] ページには、モデルプロバイダーごとに最大 20 個のサンプルが表示されます。1 つ以上の属性を使用することで、サンプルのリストをフィルタリングできます。

#### ▲ Important

モデルを使用するには、まずモデルへのアクセスをリクエストする必要があります。モデルへのアクセスをリクエストしていないのに、(API またはコンソールで) モデルを使用しようとすると、エラーメッセージが表示されます。詳細については、「<u>モデルアクセス</u>」を参照してください。

#### トピック

- ベースモデル
- テキストのプレイグラウンド
- チャットのプレイグラウンド
- イメージのプレイグラウンド
- サンプルライブラリ

### ベースモデル

Amazon Bedrock は、Amazon およびサードパーティーのモデルプロバイダーのベースモデルをサポートしています。

ベースモデル 20

Amazon Bedrock コンソールでは、利用可能なモデルを表示したり、さまざまな属性別にグループ化したりできます。また、モデルビューのフィルタリング、モデルの検索、およびモデルプロバイダーに関する情報の表示も行うことができます。

モデルを選択した後、[プレイグラウンドで開く]を選択して、モデルを試すことができます。

#### トピック

- モデルを表示したりグループ化したりする
- モデルのフィルタリングと検索
- モデルプロバイダーの詳細を表示する
- ベースモデルへのアクセスをリクエストする

#### モデルを表示したりグループ化したりする

Amazon Bedrock コンソールを使用して、モデルを一覧表示したり、モデルの詳細を表示したりできます。

- 1. Amazon Bedrock コンソールの [基盤モデル] で、[ベースモデル] を選択します。
- 2. モデルはリストビューまたはカードビューで表示でき、モデルファミリー、モデルプロバイダー、またはモデルモダリティ(テキスト、イメージ、埋め込み)別にグループ化できます。

### モデルのフィルタリングと検索

モデル名、プロバイダー、モダリティ、またはモデル属性でモデルをフィルタリングできます。

複数のフィルターを設定できます。まず、テキストボックスに文字列を入力します。コンソールに は、文字列と一致する可能性のあるすべてのフィルター値が表示されます。

- 1. Amazon Bedrock コンソールの [基盤モデル] で、[ベースモデル] を選択します。
- 2. フィルターを設定するには、[リソースを検索] テキストボックスを選択してフィルターのプロパ ティを表示します。
- 3. 目的のプロパティとフィルターの演算子を選択し、フィルタリングの基準となる値を入力します。

#### モデルプロバイダーの詳細を表示する

Amazon Bedrock コンソールを使用してモデルプロバイダーの詳細を表示するには、[基盤モデル] で [モデルプロバイダー] を選択します。

別の方法として、[ベースモデル] ページのプロバイダー名に対し、[プロバイダーの詳細を表示] を選 択します。

[モデルプロバイダー] ページの上部の近くにあるいずれかのタブを選択すると、そのモデルプロバイ ダーに関する詳細が表示されます。このページには、プロバイダーに関する以下の情報が表示されま す。

- プロバイダーの概要 プロバイダーに関する概要の記述。
- モデル 利用可能なモデルのタブ区切りリスト。いずれかのモデルを選択すると、そのモデルをプ レイグラウンドで開くことができます。
- コンテンツの制限事項 プロバイダーのコンテンツポリシーに関する詳細。

#### ベースモデルへのアクセスをリクエストする

#### Important

モデルを使用するには、まずモデルへのアクセスをリクエストする必要があります。モデル へのアクセスをリクエストしていないのに、(API またはコンソールで) モデルを使用しよう とすると、エラーメッセージが表示されます。詳細については、「モデルアクセス」を参照 してください。

## テキストのプレイグラウンド

Amazon Bedrock コンソールで [プレイグラウンド] を選択し、[テキスト] を選択して、テキストのプ レイグラウンドを表示します。

また、プレイグラウンドに直接移動するには、モデルの詳細ページまたは例ページでモデルを選択し ます。

#### トピック

- テキストモデルにプロンプトを送信する
- プロンプトと生成されたテキストを操作する

- クエリの再実行
- 高度なプロンプトを使用する
- モデルパラメータを調整する
- 別のモデルを選択する

### テキストモデルにプロンプトを送信する

以下の手順は、テキストのプレイグラウンドを使用してテキストモデルにプロンプトを送信する方法を示しています。

- 1. Amazon Bedrock コンソールを開きます。
- 2. 左側のメニューの [プレイグラウンド] で、[テキスト] を選択します。
- 3. テキストのプレイグラウンドを設定します。
  - a. 右上にある、以下のオプションをオンに切り替えます。
    - ストリーミング テキストをリアルタイムで生成します。このオプションは、ストリーミングがサポートされているモデルでのみ使用できます
  - b. テキストパネルの上にあるドロップダウンメニューから、モデルプロバイダーを選択します。プロバイダーを選択したら、モデルのドロップダウンメニューからモデルを選択します。このドロップダウンメニューには、そのプロバイダーから入手可能なモデルがリスト表示されます。カスタマイズモデルを選択することもできます (詳細については「微調整されたモデルを推論に使用する」を参照)。カスタマイズモデルを選択する場合は、事前にそのモデルのプロビジョンドスループットを設定しておく必要があります。
  - c. (オプション) [推論の設定]を調整します。デフォルト設定はほとんどのプロンプトに最適ですが、ユースケースによっては適さない場合があります。デフォルト設定に戻すには、[リセット] ボタンをクリックします。詳細については、「<u>基盤モデルの推論パラメータ</u>」を参照してください。
- 4. テキストフィールドに独自のプロンプトを入力します。プロンプトとは、「write a blog post about computers」のような自然言語のコマンドです。
- 5. 入力パネルで [実行] を選択し、テキストレスポンスを生成します。

#### Note

レスポンスがコンテンツモデレーションポリシーに違反している場合、Amazon Bedrock はそのレスポンスを表示しません。ストリーミングを有効にしている場

合、Amazon Bedrock は、ポリシーに違反するコンテンツを生成すると、レスポンス全体をクリアします。詳細については、Amazon Bedrock コンソールに移動し、[プロバイダー] を選択して、[Content limitations] セクションの説明をお読みください。

### プロンプトと生成されたテキストを操作する

生成されたテキストは、プロンプトの下に緑色の文字で表示されます。Amazon Bedrock は一般的なマークダウンやテーブルをリッチテキスト表示で出力します。ストリーミングをオンにすると、Bedrock がテキストの生成を完了するまでテキストを操作できません。入力したプロンプトと生成されたレスポンスを操作する方法は、以下のとおりです。

- 1. 満足のいくレスポンスが得られない場合は、テキストを直接編集できます。レスポンス内のテーブル、コード、リッチテキストを操作するには、以下のオブジェクトの右上にあるアイコンを使用します。
  - 編集 鉛筆アイコンを選択することで、テーブルまたはコードをプレーンテキスト形式に戻して編集できます。
  - コピー コピーアイコンを選択することで、オブジェクトがプレーンテキスト形式でコピーされます。
- 2. 行った編集を元に戻したり、やり直したりするには、[実行] ボタンの横にあるアイコンを使用します。
- 3. 生成されたテキストの下にあるアイコンを使用すると、以下のアクションを実行できます。
  - コピー コピーアイコンを選択することで、レスポンスをプレーンテキストとしてコピーできます。
  - 削除 ごみ箱アイコンを選択することで、直近のモデルレスポンスが削除されます。
- 4. モデルを試した後、[API リクエストを表示] を選択すると、同等の API リクエストが表示されます。API リクエストのコードをアプリケーションにコピーできます。

### クエリの再実行

プロンプトを変更して、クエリを再実行できます。プロンプトまたはレスポンステキストを編集し、[実行] を選択します。レスポンステキストを編集した場合は、コンソールによりテキストの色が 黒に変わります。コンソールでは、新しいレスポンスは緑色のテキストで表示されます。

#### 高度なプロンプトを使用する

プロンプトは、「write a blog post about computers」の例のような、追加のコンテキスト を必要としないシンプルなコマンドで構成することもできます。このようなプロンプトはゼロショッ トプロンプトと呼ばれます。

出力の精度を向上させるために、希望する出力の種類の例をいくつかプロンプトに指定できます。これは数ショットプロンプト (または数ショット学習) と呼ばれます。

次の例では、最初の2つのシーケンスに入力テキストと希望するレスポンスが含まれています。最 後のシーケンスは、最初の2つのシーケンスと同様の形式でモデルに応答を求めています。

Input: I love going to the mall

Sentiment: Happy

Input: I don't like going to the dentist.

Sentiment: Sad

Input: I enjoy the park

Sentiment:

一般に、大規模言語モデルは、与えられたシーケンスにおける次の論理語を予測しようとします。モ デルへのプロンプトの出し方に応じて、特定スタイルの結果を生成するよう、モデルは促されます。

#### モデルパラメータを調整する

プレイグラウンドでプロバイダーとモデルを選択すると、コンソールは選択したモデルに適用される 推論設定パラメータをロードし、それらのデフォルト値を設定します。

推論パラメータの設定を調整すると、コンテキスト、スタイル、関連性、長さが異なるテキストがモデルによって生成されます。パラメータ設定を変更すると、希望する結果が得られるまで試行錯誤を繰り返す必要があるのが一般的です。

各モデルでサポートされているパラメータについては、「<u>基盤モデルの推論パラメータ</u>」を参照して ください。

#### 別のモデルを選択する

ページ上部のドロップダウンメニューから、別のモデルプロバイダーとモデルを選択できます。推論 設定パラメータは、コンソールにより、選択したモデルのデフォルト値に設定されます。

高度なプロンプトを使用する 25

### チャットのプレイグラウンド

Amazon Bedrock コンソールで [プレイグラウンド] を選択し、[チャット] を選択して、チャットのプレイグラウンドを表示します。

また、プレイグラウンドに直接移動するには、モデルの詳細ページまたは例ページでモデルを選択します。

#### トピック

- チャットモデルにプロンプトを送信する
- チャットの指示を更新する
- 高度なプロンプトを使用する
- 推論の設定を調整する
- 別のモデルを選択する

### チャットモデルにプロンプトを送信する

以下の手順は、チャットのプレイグラウンドを使用してチャットモデルにプロンプトを送信する方法 を示しています。

- 1. Amazon Bedrock コンソールを開きます。
- 2. 左側のメニューの [プレイグラウンド] で、[チャット] を選択します。
- 3. チャットのプレイグラウンドを設定します。
  - a. テキストパネルの上にあるドロップダウンメニューから、モデルプロバイダーを選択します。プロバイダーを選択したら、モデルのドロップダウンメニューからモデルを選択します。このドロップダウンメニューには、そのプロバイダーから入手可能なモデルがリスト表示されます。カスタマイズモデルを選択することもできます (詳細については「微調整されたモデルを推論に使用する」を参照)。カスタマイズモデルを選択する場合は、事前にそのモデルのプロビジョンドスループットを設定しておく必要があります。
  - b. 右上にある、以下のオプションをオンに切り替えます。
    - ストリーミング チャットをリアルタイムで生成します。このオプションは、選択したモデルがストリーミングをサポートしていない場合には使用できません。
- 4. [指示を追加] を選択して、[更新] を選択します。コンソールの [応答] パネルには、モデルからの レスポンスが表示されます。

ー チャットのプレイグラウンド 26

5. テキストフィールドに独自のプロンプトを入力します。プロンプトとは、「Tell me about the best restaurants to visit in Seattle.」のような自然言語のフレーズまたはコマンドです。

#### Note

レスポンスがコンテンツモデレーションポリシーに違反している場合、Amazon Bedrock はそのレスポンスを表示しません。ストリーミングを有効にしている場合、Amazon Bedrock は、ポリシーに違反するコンテンツを生成すると、レスポンス全体をクリアします。詳細については、Amazon Bedrock コンソールに移動し、[プロバイダー] を選択して、[Content limitations] セクションの説明をお読みください。

#### チャットの指示を更新する

指示を変更し、その新しい指示でチャットモデルを更新できます。[更新] を選択し、モデルへの指示 を編集・変更して、[確認] を選択します。

例えば、モデルレスポンスの特徴やトーンを決定する指示を追加できます。随時、モデルに合わせて 指示を調整し、プロンプトを再実行することができます。これは、ユーザーがチャットモデルで体験 する会話を設計するときに役立ちます。

### 高度なプロンプトを使用する

プロンプトは、「Tell me about the best car to buy.」の例のような、追加のコンテキストを必要としないシンプルなコマンドで構成することもできます。このようなプロンプトはゼロショットプロンプトと呼ばれます。

プロンプトを改善するために、希望する出力の種類の例をいくつか指定できます。これは数ショット プロンプト (または数ショット学習) と呼ばれます。

例えば、「Tell me about the best car to buy. Give me a numbered list with the top 5 options. Include information on pricing, fuel efficiency, consumer ratings, and comfort of design. List the sources used for each item in the list. hybrid or electric technology used. Order the list so that the most desirable option is listed first.」のような、より複雑なプロンプトを使用できます。

チャットの指示を更新する 27

### 推論の設定を調整する

プレイグラウンドでプロバイダーとモデルを選択すると、コンソールは選択したモデルに適用される 推論設定パラメータをロードし、それらのデフォルト値を設定します。

推論パラメータの設定を調整すると、コンテキスト、スタイル、関連性、長さが異なるテキストがモデルによって生成されます。パラメータ設定を変更すると、希望する結果が得られるまで試行錯誤を繰り返す必要があるのが一般的です。推論パラメータを調整するには、チャットウィンドウの右下にある [更新] リンクを選択します。調整を行ったら、[確認] を選択します。

各モデルでサポートされているパラメータについては、「<u>基盤モデルの推論パラメータ</u>」を参照して ください。

### 別のモデルを選択する

ページ上部のドロップダウンメニューから、別のモデルプロバイダーとモデルを選択できます。推論 設定パラメータは、コンソールにより、選択したモデルのデフォルト値に設定されます。

## イメージのプレイグラウンド

Amazon Bedrock コンソールで [プレイグラウンド] を選択し、[イメージ] を選択して、イメージのプレイグラウンドを表示します。

また、イメージのプレイグラウンドに直接移動するには、モデルの詳細ページまたは例ページでモデルを選択します。

#### トピック

- モデルにプロンプトを送信する
- モデルパラメータを調整する

## モデルにプロンプトを送信する

以下の手順は、イメージのプレイグラウンドを使用してテキストイメージモデルにプロンプトを送信 する方法を示しています。

- 1. Amazon Bedrock コンソールを開きます。
- 2. 左側のメニューの [プレイグラウンド] で、[イメージ] を選択します。

推論の設定を調整する 28 **28** 

3. テキストパネルの上にあるドロップダウンメニューから、モデルプロバイダーを選択します。 プロバイダーを選択したら、モデルのドロップダウンメニューからモデルを選択します。このドロップダウンメニューには、そのプロバイダーから入手可能なモデルがリスト表示されます。

- 4. [プロンプト] フィールドにテキストを入力するか、いずれかのサンプルプロンプトを使用します。プロンプトとは、「Draw a picture of a computer」のような自然言語のコマンドです。
- 5. (オプション) [推論の設定]を調整します。デフォルト設定はほとんどの場合に最適となりますが、ユースケースによっては適さないことがあります。デフォルト設定に戻すには、[リセット] ボタンをクリックします。
- 6. [応答] フィールドで [実行] を選択してイメージを生成します。生成されたイメージは、プロンプトの下の [応答] フィールドに表示されます。
- 7. (オプション) [画像をダウンロード] を選択し、イメージをローカルマシンに保存します。
- 8. (オプション) 現在のプロンプトを履歴項目のお気に入りとして保存するには、[保存] を選択します。
- 9. モデルを試した後、[API リクエストを表示] を選択すると、同等の API リクエストが表示されます。API リクエストのコードをアプリケーションにコピーできます。

### モデルパラメータを調整する

プレイグラウンドでプロバイダーとモデルを選択すると、コンソールは選択したモデルに適用される 推論設定パラメータをロードし、それらのデフォルト値を設定します。

推論パラメータの設定を調整すると、スタイルが異なる新しいイメージがモデルによって生成されます。パラメータ設定を変更すると、希望する結果が得られるまで試行錯誤を繰り返す必要があるのが 一般的です。

各モデルでサポートされているパラメータについては、「<u>基盤モデルの推論パラメータ</u>」を参照して ください。

## サンプルライブラリ

Amazon Bedrock には、サポートされている各モデルのプロンプトの例が用意されています。コンソールで [例] を選択すると、利用可能なサンプルが表示されます。

Amazon Bedrock コンソールには、モデルプロバイダーごとに最大 20 個のサンプルが表示されます。次の属性を 1 つ以上使用することで、サンプルのリストをフィルタリングできます。

モデルパラメータを調整する 29

- モダリティ(テキスト、イメージ、または埋め込み)
- ・プロバイダー
- モデル名
- ・カテゴリ

また、フィルタリングは、テキストボックスに文字列を入力して行うこともできます。コンソールには、文字列と一致する可能性のあるすべてのフィルター値が表示されます。

特定のサンプルを選択すると、Amazon Bedrock コンソールには、そのサンプルに関する以下の情報が表示されます。

- このサンプルによる実行内容の説明。
- サンプルを実行するモデル名 (およびモデルプロバイダー)。
- サンプルプロンプトと期待されるレスポンス。
- このサンプルの推論設定パラメータの値。
- サンプルを実行する API リクエスト。

サンプルを実行するには、[プレイグラウンドで開く]を選択します。

サンプルライブラリ 30

## Amazon Bedrock API を使用する

このセクションでは、Amazon Bedrock API コールを実行するための環境設定方法を説明し、代表的なユースケースを示します。Amazon Bedrock API には、AWS Command Line Interface (AWS CLI)、AWS SDK、または SageMaker ノートブックを使用してアクセスできます。

Amazon Bedrock の API コールを実行するには、次の SDK を使用します。

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for .NET
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

APIのオペレーションとパラメータの詳細については、「<u>Amazon Bedrock API Reference</u>」を参照してください。

次のリソースは、Amazon Bedrock API に関する追加情報を提供します。

- AWS 全般のリファレンス
  - Amazon Bedrock エンドポイントとクォータ
- AWS Command Line Interface
  - Amazon Bedrock CLI コマンド
  - Amazon Bedrock ランタイム CLI コマンド

#### トピック

- Amazon Bedrock API をセットアップする
- Amazon Bedrock API オペレーション

## Amazon Bedrock API をセットアップする

#### Important

モデルを使用するには、まずモデルへのアクセスをリクエストする必要があります。モデル へのアクセスをリクエストしていないのに、(API またはコンソールで) モデルを使用しよう とすると、エラーメッセージが表示されます。詳細については、「モデルアクセス」を参照 してください。

Amazon Bedrock API には、AWS CLI、AWS SDK、または SageMaker ノートブックを使用してア クセスできます。

## Amazon Bedrock エンドポイント

AWS のサービスにプログラムで接続するには、エンドポイントを使用します。Amazon Bedrock に 使用できるエンドポイントの詳細については、「AWS 全般のリファレンス」の「Amazon Bedrock エンドポイントとクォータ」の章を参照してください。

#### AWS CLI のセットアップ

- 1. CLI を使用する場合は、AWS CLI をインストールして設定します。「the section called "AWS CLI をセットアップする"」を参照してください。
- 2. aws configure CLI コマンドを使用して AWS 認証情報を設定します(「Configure the AWS CLI」を参照)。

AWS CLI のコマンドとオペレーションについては、次のリファレンスを参照してください。

- Amazon Bedrock CLI コマンド
- Amazon Bedrock ランタイム CLI コマンド

### AWS SDK のセットアップ

AWS ソフトウェア開発キット (SDK) は、多くの一般的なプログラミング言語で使用できます。各 SDK には、デベロッパーが好みの言語でアプリケーションを簡単に構築できるようにする API、 コード例、およびドキュメントが提供されています。現時点で Amazon Bedrock API コールを実行 するには、以下の SDK を使用できます。

API のセットアップ 32

- AWS SDK for C++
- · AWS SDK for Go
- AWS SDK for Java
- · AWS SDK for JavaScript
- AWS SDK for .NET
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

コード例については、<u>Amazon Bedrock API オペレーション</u> でメソッドを検索し、選択したプログラミング言語に対応するタブを選択してください。

# SageMaker ノートブックを使用する

SageMaker ノートブックから Amazon Bedrock API オペレーションを呼び出すには、SDK for Python (Boto3) を使用します。

#### 前提条件

次のような前提条件がありますので、注意してください。

- ノートブックをホストする AWS アカウント に対する Amazon Bedrock のアクセスをリクエスト します。
- 2. コンソールを使用して、Amazon Bedrock の利用規約に同意します。

### SageMaker ロールを設定する

この SageMaker ノートブックの IAM ロールに Amazon Bedrock へのアクセス許可を追加します。

IAM コンソールから、以下の手順を実行します。

- 1. IAM ロールを選択し、[アクセス許可の追加] を選択して、ドロップダウンリストから [インラインポリシーを作成] を選択します。
- 2. 次のアクセス許可を含めてください。

{

#### 信頼関係に以下のアクセス許可を追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "bedrock.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

### ランタイムの設定をテストする

次のコードをノートブックに追加し、実行します。

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock-runtime')

body = json.dumps({
    "prompt": "\n\nHuman:explain black holes to 8th graders\n\nAssistant:",
```

```
"max_tokens_to_sample": 300,
    "temperature": 0.1,
    "top_p": 0.9,
})

modelId = 'anthropic.claude-v2'
accept = 'application/json'
contentType = 'application/json'

response = bedrock.invoke_model(body=body, modelId=modelId, accept=accept, contentType=contentType)

response_body = json.loads(response.get('body').read())
# text
print(response_body.get('completion'))
```

#### Amazon Bedrock の設定をテストする

次のコードをノートブックに追加し、実行します。

```
import boto3
bedrock = boto3.client(service_name='bedrock')
bedrock.get_foundation_model(modelIdentifier='anthropic.claude-v2')
```

## Amazon Bedrock API オペレーション

#### トピック

- ベースモデルのリスト表示
- ベースモデルの詳細を取得する
- 推論の実行
- リソースのタグ付け

### ベースモデルのリスト表示

<u>ListFoundationModels</u> オペレーションを使用して、InvokeMethod オペレーションで推論を実行するために必要なモデル ID など、ベースモデルに関する情報を取得します。次のコード例を参照してください。

API オペレーション 35

#### **AWS CLI**

ベースモデルをリスト表示するには、AWS CLI で以下のコマンドを実行します。

```
aws bedrock list-foundation-models
```

#### Python (Boto)

次の例は、Python を使用してベースモデルをリストする方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')
bedrock.list_foundation_models()
```

### ベースモデルの詳細を取得する

<u>GetFoundationModel</u> オペレーションを使用して、指定したベースモデルに関する詳細情報を取得します。次のコード例を参照してください。

#### **AWS CLI**

ベースモデルに関する情報を取得するには、AWS CLI で以下のコマンドを使用します。

```
aws bedrock get-foundation-model --model-identifier anthropic.claude-v2
```

#### Python (Boto)

次の例は、Python を使用してベースモデルをリストする方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')

bedrock.get_foundation_model(modelIdentifier='anthropic.claude-v2')
```

### 推論の実行

モデルに対して推論を実行するには、<u>InvokeModel</u> オペレーションを使用します。使用するモデルを選択するには、[modelId] フィールドを使用します。modelId の検索方法は、使用するモデルのタイプによって異なります。カスタムモデルで推論を実行するには、まずそのモデルのプロビジョン

ベースモデルの詳細を取得する 36

ドスループットを購入します (詳細については、「プロビジョンドスループット」を参照してください)。

ベースモデル - モデル ARNを見つけるには、<u>ListFoundationModels</u> を呼び出します。例については、「<u>ベースモデルのリスト表示</u>」を参照してください。現在販売されているベースモデルのモデル ID のリストを参照するには、「<u>ベースモデル ID</u>」を参照してください。基盤モデルのモデル ID は、Amazon Bedrock コンソールの JSON サンプルから取得することもできます。

プロビジョンドスループットを使用するモデル - 基盤モデルまたはカスタムモデル用にプロビジョンドスループットを作成した場合は、<u>ListProvisionedModelThoutputs</u>を呼び出して、そのレスポンスで modelArn を確認します。また、モデル ARN は、コンソールの [プロビジョンドスループット] セクションでモデルを選択すると、[モデルの詳細] で確認することもできます。

各ベースモデルには、 [body] フィールドで設定する独自のパラメータがあります。詳細については、「基盤モデルの推論パラメータ」を参照してください。

ストリーミングで推論を実行するには、<u>InvokeModelWithResponseStream</u> オペレーションを使用します。ストリーミングをサポートするベースモデル、またはストリーミングをサポートするベースモデルから作成されたモデルを選択します。

#### トピック

- モデルで推論を実行する
- ベースモデル ID

#### モデルで推論を実行する

以下の例は、<u>InvokeModel</u> を使用してモデルで推論を実行する方法と、<u>InvokeModelWithResponseStream</u> オペレーションを使用してストリーミングによる推論を実行する方法 (Python を使用する場合) を示しています。

Note

AWS CLI はストリーミングをサポートしていません。

各モデルでサポートされているパラメータについては、「<u>基盤モデルの推論パラメータ</u>」を参照してください。プロンプトの書き方については、「<u>プロンプトエンジニアリングガイドライン</u>」を参照してください。

#### **AWS CLI**

次の例では、「story of two dogs (2 ########)」のプロンプトと Anthropic Claude V2 モデルを使用して AWS CLI でテキストを生成する方法を示します。この例では、レスポンスで最大 300 個のトークンを返し、そのレスポンスを output.txt ファイルに保存しています。

```
aws bedrock-runtime invoke-model \
    --model-id anthropic.claude-v2 \
    --body "{\"prompt\": \"\n\nHuman: story of two dogs\n\nAssistant:\",
    \"max_tokens_to_sample\": 300}" \
    --cli-binary-format raw-in-base64-out \
    invoke-model-output.txt
```

次の例は、Llama 2 Chat 13B モデルの呼び出し方法を示しています。

```
aws bedrock-runtime invoke-model \
    --region us-east-1 \
    --model-id meta.llama2-13b-chat-v1 \
    --body "{\"prompt\": \"What is the average lifespan of a Llama?\",
    \"max_gen_len\" : 128, \"temperature\": 0.1, \"top_p\": 0.9}" \
    invoke-model-output.txt
```

#### Python (Boto)

次の例は、「explain black holes to 8th graders (8 #################)」というプロンプトと Anthropic Claude V2 モデルを使用して Python でテキストを生成する方法を示しています。

```
import boto3
import json
brt = boto3.client(service_name='bedrock-runtime')

body = json.dumps({
    "prompt": "\n\nHuman: explain black holes to 8th graders\n\nAssistant:",
    "max_tokens_to_sample": 300,
    "temperature": 0.1,
    "top_p": 0.9,
})

modelId = 'anthropic.claude-v2'
accept = 'application/json'
contentType = 'application/json'
```

```
response = brt.invoke_model(body=body, modelId=modelId, accept=accept,
  contentType=contentType)

response_body = json.loads(response.get('body').read())

# text
print(response_body.get('completion'))
```

```
import boto3
import json
brt = boto3.client(service_name='bedrock-runtime')
body = json.dumps({
    'prompt': '\n\nHuman: write an essay for living on mars in 1000 words\n
\nAssistant:',
    'max_tokens_to_sample': 100
})
response = brt.invoke_model_with_response_stream(
    modelId='anthropic.claude-v2',
    body=body
)
stream = response.get('body')
if stream:
    for event in stream:
        chunk = event.get('chunk')
        if chunk:
            print(json.loads(chunk.get('bytes').decode()))
```

#### ベースモデルの推論例

次の Python (Boto) の例は、さまざまな Amazon Bedrock ベースモデルで <u>InvokeModel</u> オペレーションを使用して推論を実行する方法を示しています。

#### トピック

- A2I Jurassic-2
- Cohere Command
- Meta Llama 2
- Stability Al Diffusion XL

#### A2I Jurassic-2

この例は、A2I Jurassic-2 Mid モデルの呼び出し方法を示しています。

```
import boto3
import json
brt = boto3.client(service_name='bedrock-runtime')
body = json.dumps({
    "prompt": "Translate to spanish: 'Amazon Bedrock is the easiest way to build and
 scale generative AI applications with base models (FMs)'.",
    "maxTokens": 200,
    "temperature": 0.5,
    "topP": 0.5
})
modelId = 'ai21.j2-mid-v1'
accept = 'application/json'
contentType = 'application/json'
response = brt.invoke_model(
    body=body,
    modelId=modelId,
    accept=accept,
    contentType=contentType
)
response_body = json.loads(response.get('body').read())
# text
print(response_body.get('completions')[0].get('data').get('text'))
```

#### Cohere Command

この例は、Cohere コマンドモデルを呼び出す方法を示しています。

```
import boto3
import json
brt = boto3.client(service_name='bedrock-runtime')
body = json.dumps({
    "prompt": "How do you tie a tie?",
    "max_tokens": 200,
    "temperature": 0.5,
    "p": 0.5
})
modelId = 'cohere.command-text-v14'
accept = 'application/json'
contentType = 'application/json'
response = brt.invoke_model(
    body=body,
    modelId=modelId,
    accept=accept,
    contentType=contentType
)
response_body = json.loads(response.get('body').read())
# text
print(response_body.get('generations')[0].get('text'))
```

#### Meta Llama 2

次の例は、Llama 2 Chat 13B モデルの呼び出し方法を示しています。

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock-runtime', region_name='us-east-1')

body = json.dumps({
    "prompt": "What is the average lifespan of a Llama?",
    "max_gen_len": 128,
    "temperature": 0.1,
    "top_p": 0.9,
})
```

```
modelId = 'meta.llama2-13b-chat-v1'
accept = 'application/json'
contentType = 'application/json'

response = bedrock.invoke_model(body=body, modelId=modelId, accept=accept, contentType=contentType)

response_body = json.loads(response.get('body').read())
print(response_body)
```

#### Stability AI Diffusion XL

この例では、Stability Al Stability Diffusion XL モデルを呼び出す方法を示しています。

```
import boto3
import json
brt = boto3.client(service_name='bedrock-runtime')
prompt_data = "A photograph of an dog on the top of a mountain covered in snow."
body = json.dumps({
  "text_prompts": [
    {
      "text": prompt_data
    }
  ],
  "cfg_scale":10,
  "seed":20,
  "steps":50
})
modelId = "stability.stable-diffusion-xl-v0"
accept = "application/json"
contentType = "application/json"
response = brt.invoke_model(
    body=body,
    modelId=modelId,
    accept=accept,
    contentType=contentType
)
response_body = json.loads(response.get("body").read())
print(response_body['result'])
```

推論の実行 42<sup>-</sup>

print(f'{response\_body.get("artifacts")[0].get("base64")[0:80]}...')

#### ベースモデル ID

以下は、現在販売されているベースモデルのモデル ID のリストです。モデル ID は、<u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> の呼び出しに使用するベースモデルを識別するものです。

プロビジョンドスループットに必要なモデル ID については、「プロビジョンドスループットモデル ID」を参照してください。

プロバイダー	モデル名	バージョン	モデル ID
Al21 Labs	Jurassic-2 Mid	1.x	ai21.j2-mid-v1
Al21 Labs	Jurassic-2 Ultra	1.x	ai21.j2-ultra-v1
Amazon	Titan Text G1 - Lite	1.x	amazon.titan-text-lite- v1
Amazon	Titan Embeddings G1 - Text	1.x	amazon.titan-embed- text-v1
Amazon	Titan Text G1 - Express	1.x	amazon.titan-text- express-v1
Amazon	Titan Text G1 - Agile	1.x	amazon.titan-text- agile-v1
Anthropic	Claude	1.x	anthropic.claude-v1
Anthropic	Claude	2.x	anthropic.claude-v2
Anthropic	Claude Instant	1.x	anthropic.claude-i nstant-v1
Cohere	Command	14.x	cohere.command-tex t-v14
Cohere	Command Light	15.x	cohere.command-lig ht-text-v14

プロバイダー	モデル名	バージョン	モデル ID
Cohere	Embed English	3.x	cohere.embed-engli sh-v3
Cohere	Embed Multilingual	3.x	cohere.embed-multi lingual-v3
Meta	Llama 2 Chat 13B	1.x	meta.llama2-13b-ch at-v1
Stability AI	Stable Diffusion XL	0.x	stability.stable-d iffusion-xl-v0

#### Note

Amazon Titan Text G1 - Express モデルは限定プレビューリリースに同梱されています。アクセス許可は今後も付与されます。

#### Note

Stable Diffusion XL 1.x はプロビジョンドスループットでのみ利用可能です。詳細については、「<u>プロビジョンドスループット</u>」を参照してください。Stable Diffusion XL 0.x はプレビューリリースです。

## リソースのタグ付け

リソースのタグ付けおよびタグ付け解除を行うには、<u>TagResource</u> オペレーションと <u>UnTagResource</u> オペレーションを使用します。タグ付けとタグ付け解除を行うには、リソースの ARN が必要です。

タグ付けできる Amazon Bedrock リソースは以下のとおりです。

- カスタムモデル
- モデルカスタマイズジョブ

リソースのタグ付け 44

#### プロビジョンドモデル

タグ付けの制約事項の詳細については、「タグの制限」を参照してください

リソースのタグをリスト表示するには、ListTagsForResource オペレーションを使用します。

#### **AWS CLI**

次の例は、AWS CLI でリソースに 2 つのタグを追加する方法を示しています。キーと値のペア同士をスペースで区切ります。

```
aws bedrock tag-resource \
    --resource-arn "arn:aws:resource-arn" \
    --tags key=key1, value=value1 key=key2, value=value2
```

次の例は、key1 と key2 のキーが指定されたタグをリソースから削除する方法を示しています。キー同士をスペースで区切ります。

```
aws bedrock untag-resource \
    --resource-arn "arn:aws:resource-arn" \
    --tag-keys key=key1 key2
```

次の例は、リソースのタグをリスト表示する方法を示しています。

```
aws bedrock list-tags-for-resource \
    --resource-arn "arn:aws:iam::resource-arn"
```

#### Python (Boto)

次の例は、Python でリソースにタグを追加する方法を示しています。

リソースのタグ付け 45

```
'key': 'key2',
    'value': 'value2'
}

bedrock.tag_resource(resourceARN='arn:aws:bedrock:resource-arn', tags=tags)
```

次の例は、Python で key1 と key2 のキーが指定されたタグをリソースから削除する方法を示しています。

```
import boto3

bedrock = boto3.client(service_name='bedrock')

bedrock.untag_resource(resourceARN='arn:aws:bedrock:resource-arn', tagKeys=['key1', 'key2'])
```

次の例は、Python でリソースのタグをリスト表示する方法を示しています。

```
import boto3

bedrock = boto3.client(service_name='bedrock')

bedrock.list_tags_for_resource(resourceARN='arn:aws:bedrock:resource-arn')
```

リソースのタグ付け 46

## 基盤モデルの推論パラメータ

Amazon Bedrock の任意の基盤モデルを使用して推論を実行します。オプションで、モデルが生成するレスポンスに影響する推論パラメータを設定します。推論パラメータは、コンソールのプレイグラウンド、あるいは <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> API の [body] フィールドで設定します。

モデルを呼び出すときには、モデルのプロンプトも含めます。プロンプトの書き方については、「 プロンプトエンジニアリングガイドライン」を参照してください。

以下のセクションでは、各ベースモデルで使用できる推論パラメータを定義しています。カスタムモデルでは、カスタマイズの基となったベースモデルと同じ推論パラメータを使用してください。

#### トピック

- 推論パラメータの定義
- Amazon Titan モデル
- Anthropic Claude モデル
- Al21 Labs Jurassic-2 モデル
- Cohere モデル
- Meta Llama 2 Chat 13B モデル
- Stability.ai Diffusion モデル

## 推論パラメータの定義

通常、基盤モデルでサポートされる推論パラメータの種類は次のとおりです。

#### トピック

- ランダム性と多様性
- 長さ
- 繰り返し

### ランダム性と多様性

基盤モデルでは通常、レスポンスのランダム性と多様性を制御するための以下のパラメータがサポートされています。

• 温度 - 大規模言語モデルでは、確率を使用して単語のシーケンスを構成します。どのシーケンスでも、そのシーケンス内で次にどの単語を選択するかには確率分布があります。温度をゼロに近づけると、モデルは確率の高い単語を選択していきます。温度をゼロから遠ざけると、モデルは確率の低い単語を選択していきます。

技術的には、温度によって次のトークンの確率密度関数が調整され、温度サンプリング手法が実装されます。このパラメータにより、確率密度関数の曲線の曲率を大きくしたり小さくしたりできます。温度の値が小さいほど曲線の勾配が高くなってレスポンスがより決定論的になり、温度の値が大きいほどランダムなレスポンスが多い平坦な曲線になります。

• トップ K - 温度が候補となる単語の確率分布を定義するのに対し、[トップ K] には、モデルが選択する単語数の上限 (カットオフ値) を設定します。例えば、[トップ K] に 50 を指定した場合、モデルは特定のシーケンスにおいて次に来る可能性が最も高い 50 個の単語の中から選択します。[トップ K] の値を下げると、シーケンスにおいて次に珍しい単語が選択される可能性が低くなります。

専門用語では、トップ K は、トップ K フィルタリングで絞り込まれる確率が最も高い語彙トークンの数です。

• トップ P - [トップ P] には、候補となる選択肢の確率の合計に基づいてカットオフ値を設定します。[トップ P] を 1.0 未満に設定すると、モデルは最も可能性の高い選択肢を考慮し、可能性の低い選択肢は無視します。[トップ P] は [トップ K] と似ていますが、選択肢の数に上限を設けるのではなく、選択肢の確率の合計に基づいて選択肢の上限を設定します。

「…のひづめの音が聞こえる」というサンプルプロンプトでは、次の単語として「馬」、「シマウマ」、または「ユニコーン」をモデルに表示させたいでしょう。[トップ K] や [トップ P] に上限を設定せずに [温度] に最高値を設定すると、「ユニコーン」のような異常な結果が表示される可能性が高くなります。[温度] に 0 を設定すると、「馬」が表示される可能性が高くなります。[温度] に高い値を設定して [トップ K] または [トップ P] の値を下げると、「馬」や「シマウマ」が表示される可能性が高くなり、「ユニコーン」が表示される可能性が低くなります。

### 長さ

基盤モデルでは通常、生成されるレスポンスの長さを制御する以下のパラメータがサポートされています。

- レスポンスの長さ 生成されるレスポンスで使用するトークンの最小数と最大数を設定します。
- 長さのペナルティ [長さのペナルティ] は、レスポンスが長くなるとペナルティを課すことにより、モデルの出力がより簡潔になるようにモデルを最適化します。

長さ 48

専門用語で言えば、レスポンスが長くなると、[長さのペナルティ]はモデルに指数関数的にペナルティを課します。0.0 はペナルティなしを指定するものです。より長いシーケンスを生成するには、モデルに 0.0 未満の値を設定します。より短いシーケンスを生成するには、0.0 より大きい値を設定します。

• ストップシーケンス - ストップシーケンスは文字のシーケンスです。モデルがストップシーケンスに遭遇すると、それ以降のトークンの生成を停止します。モデルが異なれば、ストップシーケンス内の文字の種類や、シーケンスの最大長も異なるほか、複数のストップシーケンスの定義がサポートされている場合もあります。

### 繰り返し

基盤モデルでは通常、生成されるレスポンス内での繰り返しを制御するのに役立つ以下のパラメータがサポートされています。

繰り返しのペナルティ (プレゼンスのペナルティ) - レスポンス内で同じ単語 (トークン) が繰り返されるのを抑制します。1.0 はペナルティなしを指定するものです。1.0 より大きい値を指定すると、繰り返し回数が減ります。

### Amazon Titan モデル

Amazon Titan モデルでは、以下の推論パラメータをサポートしています。

## ランダム性と多様性

Amazon Titan モデルでは、レスポンスのランダム性と多様性を制御する以下のパラメータをサポートしています。

- 温度 (temperature) 低い値を指定すると、レスポンスのランダム性を減らすことができます。
- トップ P (topP) 低い値を指定すると、可能性の低い選択肢を無視することができます。

### 長さ

Amazon Titan モデルでは、生成されるレスポンスの長さを制御する以下のパラメータがサポートされています。

繰り返し 49

• レスポンスの長さ (maxTokenCount) - 生成されるレスポンスに含まれるトークンの最大数を指定します。

• ストップシーケンス (stopSequences) - モデルが停止する場所を示す文字シーケンスを指定します。シーケンス (最大 20 文字) が複数個ある場合は、それらの間を| (パイプ) 文字で区切ります。

## モデル呼び出しリクエストの body フィールド

Titan モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。[inputText] フィールドにプロンプトを入力します。

```
{
    "inputText": string,
    "textGenerationConfig": {
        "temperature": float,
        "topP": float,
        "maxTokenCount": int,
        "stopSequences": [string]
    }
}
```

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の フィールド	最小値	最大値	デフォルト値
ランダム性と 多様性	温度	温度	0	1	0
	トップP	topP	0	1	1
長さ	レスポンスの 長さ	maxTokenC ount	0	8,000	512

## モデル呼び出しレスポンスの body フィールド

InvokeModel のレスポンスとしては、次の形式の body フィールドが返されます。

```
{
```

```
'inputTextTokenCount': int,
'results': [{
     'tokenCount': int,
     'outputText': '\n<response>\n',
     'completionReason': string
}]
}
```

各フィールドの詳細は以下のとおりです。

- inputTextTokenCount プロンプト内のトークンの数。
- tokenCount レスポンス内のトークンの数。
- outputText レスポンス内のテキスト。
- completionReason レスポンスの生成が終了した理由。理由の候補としては、以下があります。
  - FINISHED レスポンスは完全に生成されました。
  - LENGTH 設定したレスポンスの長さにより、レスポンスが切り捨てられました。

InvokeModelWithResponseStream のレスポンスとしては、body フィールドの形式でレスポンスストリームオブジェクトが返されます。レスポンスストリームのテキストの各チャンクは、以下の形式です。bytes フィールドをデコードする必要があります (例については「モデルで推論を実行する」を参照)。

```
{
    'chunk': {
        'bytes': b'{
            "index": int,
            "inputTextTokenCount": int,
            "totalOutputTextTokenCount": int,
            "outputText": "<response-chunk>",
            "completionReason": string
        }'
    }
}
```

- index ストリーミングレスポンス内のチャンクのインデックス。
- inputTextTokenCount プロンプト内のトークンの数。
- totalOutputTextTokenCount レスポンス内のトークンの数。

- outputText レスポンス内のテキスト。
- completionReason レスポンスの生成が終了した理由。理由の候補としては、以下があります。
  - FINISHED レスポンスは完全に生成されました。
  - LENGTH 設定したレスポンスの長さにより、レスポンスが切り捨てられました。

## Anthropic Claude モデル

Anthropic Claude モデルは以下の種類の制御を行うことができます。

## ランダム性と多様性

Anthropic Claude モデルでは、レスポンスのランダム性と多様性を制御する以下のパラメータをサポートしています。

- 温度 (temperature) 低い値を指定すると、レスポンスのランダム性を減らすことができます。
- トップ P (topP) 低い値を指定すると、可能性の低い選択肢を無視することができます。
- トップ K (topK) モデルが次のトークンを生成するために使用できる選択肢となるトークンの数を指定します。

### 長さ

Anthropic Claude モデルでは、生成されるレスポンスの長さを制御する以下のパラメータがサポートされています。

- 最大長 (max\_tokens\_to\_sample) 生成されるレスポンスで使用されるトークンの最大数を指定します。最適なパフォーマンスを得るには、トークンを 4,000 個に制限することをお勧めします。
- ストップシーケンス (stop\_sequences) モデルに認識させるシーケンスを最大 4 つ設定します。モデルがストップシーケンスに遭遇すると、それ以降のトークンの生成を停止します。返されるテキストにはストップシーケンスは含まれません。

Anthropic Claude モデル 52

## モデル呼び出しリクエストの body フィールド

Anthropic モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。prompt フィールドのフォーマットをコピーし、*prompt* を自作のプロンプトに置き換えます。

```
{
    "prompt": "\n\nHuman:prompt>\n\nAssistant:",
    "temperature": float,
    "top_p": float,
    "top_k": int,
    "max_tokens_to_sample": int,
    "stop_sequences": ["\n\nHuman:"]
}
```

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の オブジェクト	最小値	最大値	デフォルト値
ランダム性と 多様性	温度	temperature	0	1	0.5
	トップP	top_p	0	1	1
	トップK	top_k	0	500	250
長さ	サンプリング するトークン の最大数	max_token s_to_sample	0	4,096	200

## モデル呼び出しレスポンスの body フィールド

レスポンス内の body フィールドの形式については、<u>https://docs.anthropic.com/claude/reference/complete\_post</u>を参照してください。

## AI21 Labs Jurassic-2 モデル

Al21 Jurassic-2 モデルでは以下の種類の制御を行うことができます。

### ランダム性と多様性

Al21 Jurassic-2 モデルでは。レスポンスのランダム性と多様性を制御する以下のパラメータをサポートしています。

- 温度 (temperature) 低い値を指定すると、レスポンスのランダム性を減らすことができます。
- トップ P (topP) 低い値を指定すると、可能性の低い選択肢を無視することができます。

### 長さ

Al21 Jurassic-2 モデルでは、生成されるレスポンスの長さを制御する以下のパラメータがサポート されています。

- 最大入力長 (maxTokens) 生成されるレスポンスで使用されるトークンの最大数を指定します。
- ストップシーケンス (stopSequences) モデルに認識させて、それ以降はトークンの生成を停止 させるストップシーケンスを設定します。ストップシーケンスに改行文字を挿入するには、Enter キーを押します。ストップシーケンスの挿入を終了するには、Tab キーを使用します。

## 繰り返し

Al21 Jurassic-2 モデルでは、生成されるレスポンスの長さを制御する以下のパラメータがサポート されています。

- プレゼンスのペナルティ (presencePenalty) 高い値を指定すると、プロンプトまたは入力内容 に既に少なくとも 1 回出現しているトークンが新規に生成される可能性が低くなります。
- カウントのペナルティ (countPenalty) 高い値を指定すると、プロンプトまたは入力内容に既に少なくとも 1 回出現しているトークンが新規に生成される可能性が低くなります。この値は、出現回数に比例します。
- 頻度のペナルティ (frequencyPenalty) 高い値を指定すると、プロンプトまたは完了時にすでに 1 回以上出現する新しいトークンが生成される可能性が低くなります。この値は、トークンの 出現頻度に比例します (テキストの長さに合わせて正規化されます)。
- 特殊トークンにペナルティを課す 特殊文字が繰り返し使用される可能性を減らします。デフォルト値は true です。
  - 空白 (applyToWhitespaces) true の値を指定すると、空白や改行にペナルティが課されます。

ランダム性と多様性 54

句読点 (applyToPunctuation) - true の値を指定すると、句読点にペナルティが課されます。

- 数字 (applyToNumbers) true の値を指定すると、数字にペナルティが課されます。
- ストップワード (applyToStopwords) true の値を指定すると、ストップワードにペナルティが課されます。
- 絵文字 (applyToEmojis) true の値を指定すると、絵文字がペナルティの対象から除外されます。

## モデル呼び出しリクエストの body フィールド

Al21 モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。prompt フィールドにプロンプトを入力します。

```
{
    "prompt": string,
    "temperature": float,
    "topP": float,
    "maxTokens": int,
    "stopSequences": [string],
    "countPenalty": {
        "scale": int
    },
    "presencePenalty": {
        "scale": float
    },
    "frequencyPenalty": {
        "scale": int
    }
}
```

特殊なトークンにペナルティを課すには、それらのフィールドをペナルティオブジェクトに追加します。例えば、countPenalty フィールドを次のように変更できます。

```
"countPenalty": {
    "scale": int,
    "applyToWhitespaces": boolean,
    "applyToPunctuations": boolean,
```

```
"applyToNumbers": boolean,
    "applyToStopwords": boolean,
    "applyToEmojis": boolean
}
```

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の オブジェクト	最小値	最大値	デフォルト値
ランダム性と	温度	temperature	0	1	0.5
多様性	トップP	topP	0	1	0.5
長さ	最大トークン (中規模、超 大規模、およ び大規模モデ ル)	maxTokens	0	8,191	200
	最大トークン (上記以外の モデル)		0	2,048	200
繰り返し	プレゼンスの ペナルティ	presenceP enalty	0	5	0
	カウントのペ ナルティ	countPenalty	0	1	0
	頻度のペナル ティ	frequency Penalty	0	500	0

## モデル呼び出しレスポンスの body フィールド

レスポンス内の <u>フィールドの形式については、bodyhttps://docs.ai21.com/reference/j2-complete-ref</u>を参照してください。

## Cohere モデル

Amazon Bedrock がサポートする Cohere モデルの推論パラメータ情報を以下に示します。

#### トピック

- Cohere Command モデル
- Cohere Embed モデル

#### Cohere Command モデル

Cohere Command モデルでは以下の制御を行うことができます。

- 可能性を返す (return\_likelihoods) (オプション) トークンの可能性をレスポンスとともに返すかどうかと、返す場合の返し方を指定します。以下のオプションを指定できます。
  - GENERATION 生成されたトークンの可能性のみを返します。
  - ALL すべてのトークンの可能性を返します。
  - NONE (デフォルト) 可能性を一切返しません。
- ストリーム (stream) (ストリーミングをサポートする場合は必須) true を指定すると、レスポンスの各構成要素がリアルタイムで返され、false を指定すると、プロセスの終了後にレスポンス全体が返されます。
- Logit bias (logit\_bias) (オプション) モデルが希望に沿わないトークンを生成しないようにしたり、希望に沿うトークンを含めるようにモデルにインセンティブを与えたりします。形式は{token\_id: bias} です。ここで、bias は -10 から 10 までの間にある浮動小数値です。トークンは、Cohere の Tokenize エンドポイントなど、どのトークン化サービスを使用した場合でも、テキストから取得できます。詳細については、cohere サイトの [Documentation] タブ を参照してください。
- 世代数 (num\_generations) (オプション) モデルが返すことができる最大世代数。
- 切り捨て (truncate) (オプション) トークンの最大長を超える入力を API がどのように処理する かを指定します。以下のいずれかを使用します。
  - NONE 入力が入力トークンの最大長を超えるとエラーを返します。
  - START 入力の先頭部分を切り捨てます。
  - END-(デフォルト)入力の末尾部分を切り捨てます。

START または END を指定すると、入力の長さがモデルの入力トークンの最大長とまったく同じになるまで、モデルが入力内容を切り捨てます。

Cohere モデル 57

#### ランダム性と多様性

Cohere Command モデルでは、レスポンスのランダム性と多様性を制御する以下のパラメータをサポートしています。

- 温度 (temperature) (オプション) 低い値を指定すると、レスポンスのランダム性を減らすことができます。
- トップ P (p) (オプション) 低い値を指定すると、可能性の低い選択肢を無視することができます。0 または 1.0 に設定すると、このオプションは無効になります。p と k を両方とも有効にした場合は、k が動作した後に p が動作します。
- トップ K(k) (オプション) モデルが次のトークンを生成するために使用できる選択肢となるトークンの数を指定します。pとkを両方とも有効にした場合は、kが動作した後にpが動作します。

#### 長さ

Cohere Command モデルでは、生成されるレスポンスの長さを制御する以下のパラメータがサポートされています。

- 最大長 (max\_tokens) (オプション) 生成されるレスポンスで使用されるトークンの最大数を指定します。
- ストップシーケンス (stop\_sequences) (オプション) モデルに認識させるシーケンスを最大 4 つ設定します。モデルがストップシーケンスに遭遇すると、それ以降のトークンの生成を停止します。返されるテキストにはストップシーケンスは含まれません。

### モデル呼び出しリクエストの body フィールド

Cohere モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。prompt フィールドにプロンプトを入力します。prompt フィールドは必須です。

```
"prompt": "string",
    "temperature": float,
    "p": float,
    "k": float,
    "max_tokens": int,
    "stop_sequences": ["string"],
```

Cohere Command モデル 58

```
"return_likelihoods": "GENERATION|ALL|NONE",
    "stream": boolean,
    "num_generations": int,
    "logit_bias": {token_id: bias},
    "truncate": "NONE|START|END"
}
```

詳細については、https://docs.cohere.com/reference/generate を参照してください。

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の オブジェクト	最小値	最大値	デフォルト値
ランダム性と	温度	temperature	0	5	0.9
多様性	トップP	p	0	1	0.75
	トップK	k	0	500	0
長さ	最大トークン	max_tokens	1	4,096	20
その他	世代数	num_gener ations	1	5	1
	Logit bias	logit_bias	-10 (トークン バイアスとし て)	10 (トークン バイアスとし て)	デフォルト値 なし

## モデル呼び出しレスポンスの body フィールド

InvokeModel または InvokeModelWithResponseStream を呼び出した場合の body レスポンスを以下に示します。

Cohere Command モデル 59

このレスポンスに指定できるフィールドについて説明します。

- generations 生成された結果と、リクエストされたトークンの可能性から成るリスト。(常に返されます)。リストの各世代 (generation) オブジェクトには、以下のフィールドを指定します。
  - id 世代の識別子。(常に返されます)。
  - likelihood 出力される可能性。(返されない場合もあります)。
  - token\_likelihoods トークンごとの可能性の配列。(返されない場合もあります)。
  - finish\_reason モデルがトークンの生成を終了した理由を示します。COMPLETE モデルが 完成したレスポンスを返しました。MAX\_TOKENS モデルがそのコンテキスト長に対する最大 トークン数に達したため、レスポンスがカットオフされました。ERROR レスポンスの生成中 に問題が発生しました。ERROR\_TOXIC 有毒と見なされるレスポンスがモデルで生成されました。finish\_reason は、is\_finished = true の場合にのみ返されます。(返されない場合も あります)。
  - is\_finished stream が true の場合にのみ使用されるブール値型フィールド。ストリーミングレスポンスの一部として生成される追加のトークンがあることを示します。(返されない場合もあります)。
  - text 生成されたテキスト。
  - index ストリーミングレスポンスにおいて、特定のトークンがどの世代に属しているかを判断するのに使用されます。1 つのレスポンスのみがストリーミングされる場合、すべてのトークンが同じ世代に属し、index の値は返されません。 そのため、index が返されるのは、ストリーミングリクエストにおいて num\_generations の値が1より大きい場合に限ります。
- prompt 入力リクエストからのプロンプト。(常に返されます)。
- id リクエストの識別子 (常に返されます)。

Cohere Command モデル 60

詳細については、https://docs.cohere.com/reference/generate を参照してください。

#### Cohere Embed モデル

Cohere Command モデルでは以下の制御を行うことができます。

• texts (texts) - (必須) 埋め込むモデルの文字列の配列。最適なパフォーマンスを得るには、各テキストの長さを 512 トークン未満に減らすことをお勧めします。1 トークンは約 4 文字です。

- 入力タイプ (input\_type) (必須) タイプ同士を区別するための特殊なトークンを先頭に付加します。検索と取得の間でタイプを混在させる場合を除いて、異なるタイプを混在させないでください。そのような場合、search\_document タイプにはコーパスを、search\_query タイプには埋め込みクエリを埋め込みます。
  - search\_document 検索のユースケースで、ベクトルデータベースに保存する埋め込み用のドキュメントをエンコードするときに、search\_document を使用します。
  - search\_query ベクトル DB にクエリを実行して関連ドキュメントを検索する場合に search\_query を使用します。
  - classification 埋め込みをテキスト分類子への入力として使用する場合に classification を使用します。
  - clustering 埋め込みをクラスター化する場合に clustering を使用します。
- 切り捨て (truncate) (オプション) トークンの最大長を超える入力を API がどのように処理する かを指定します。以下のいずれかを使用します。
  - NONE (デフォルト) 入力が入力トークンの最大長を超えるとエラーを返します。
  - LEFT 入力の先頭部分を切り捨てます。
  - RIGHT 入力の末尾部分を切り捨てます。

LEFT または RIGHT を指定すると、入力の長さがモデルの入力トークンの最大長とまったく同じになるまで、モデルが入力内容を切り捨てます。

## モデル呼び出しリクエストの body フィールド

Cohere モデルを使用して <u>InvokeModel</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。prompt フィールドにプロンプトを入力します。

```
{
    "texts:["string"],
    "input_type": "search_document|search_query|classification|clustering",
```

Cohere Embed モデル 61

```
"truncate": "NONE|LEFT|RIGHT"
}
```

詳細については、https://docs.cohere.com/reference/embed を参照してください。

Note

Amazon Bedrock は Cohere Embed モデルからのストリーミングレスポンスをサポートしていません。

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の オブジェクト	最小値	最大値	デフォルト値
その他	テキスト	texts	0 個のテキス ト/呼び出し	128 個のテキ スト/呼び出 し	該当なし
	テキストの長 さ	texts	0 文字	2,048 文字	該当なし

## モデル呼び出しレスポンスの body フィールド

InvokeModel を呼び出した場合の body レスポンスを以下に示します。

```
{
    "embeddings": [
          [float //*1024)]
    ],
    "id": "string",
    "texts": ["string"]
}
```

body レスポンスに指定できるフィールドについて説明します。

Cohere Embed モデル 62

- id レスポンスの識別子。
- embeddings 埋め込みの配列。各埋め込みは 1,024 個の要素から成る、浮動小数点数の配列です。embeddings 配列の長さは 元の texts 配列の長さと同じになります。

• texts - 埋め込みが返されたテキストエントリから成る配列。

詳細については、https://docs.cohere.com/reference/embed を参照してください。

### Meta Llama 2 Chat 13B モデル

Meta Llama 2 Chat 13B モデルは以下を制御できます。

### ランダム性と多様性

Meta Llama 2 Chate 13B モデルは、レスポンスのランダム性と多様性を制御する以下のパラメータをサポートしています。

- 温度 (temperature) 低い値を指定すると、レスポンスのランダム性を減らすことができます。
- トップ P (top\_p) 低い値を指定すると、可能性の低い選択肢を無視することができます。0 または 1.0 に設定すると、このオプションは無効になります。

### 長さ

Meta Llama 2 Chat 13B モデルでは、生成されるレスポンスの長さを制御する以下のパラメータをサポートしています。

最大長 (max\_gen\_len) - 生成されるレスポンスで使用されるトークンの最大数を指定します。生成されたテキストの長さが max\_gen\_len を超えると、モデルはレスポンスを切り捨てます。

## モデル呼び出しリクエストの body フィールド

Lllama 2 モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。prompt フィールドにプロンプトを入力します。

```
{
    "prompt": "string",
```

Meta Llama 2 Chat 13B モデル 63

```
"temperature": float,
"top_p": float,
"max_gen_len": int
}
```

サンプルコードについては、「Meta Llama 2」を参照してください。

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

カテゴリ	パラメータ	JSON 形式の オブジェクト	最小値	最大値	デフォルト値
ランダム性と 多様性	温度	temperature	0	1	0.5
	トップP	top_p	0	1	0.9
長さ	最大生成長	max_gen_len	1	2,048	512

## モデル呼び出しレスポンスの body フィールド

InvokeModel のレスポンスとしては、次の形式の body フィールドが返されます。

```
{
    "generation": "\n\n<response>",
    "prompt_token_count": int,
    "generation_token_count": int,
    "stop_reason" : "string"
}
```

各フィールドの詳細は以下のとおりです。

- generation 生成されたテキスト。
- prompt\_token\_count プロンプト内のトークンの数。
- generation\_token\_count 生成されたテキスト内のトークンの数。
- stop\_reason レスポンスがテキストの生成を停止した理由。可能な値は以下のとおりです。
  - 停止 モデルは入力プロンプトのテキストの生成を終了しました。
  - 長さ 生成されたテキストにおけるトークンの長さが InvokeModel (出力をストリーミングする場合は InvokeModelWithResponseStream) の呼び出しにおける max\_gen\_len

の値を超えています。レスポンスは max\_gen\_len 個のトークンの長さに切り捨てられます。max\_gen\_len の値を大きくしてやり直すことを検討してください。

## Stability.ai Diffusion モデル

Stability.ai Diffusion モデルでは、次のタイプの制御を行うことができます。

- プロンプトの強度 (cfg\_scale) 最終的な画像でプロンプトをどの程度表現するかを決定します。小さな数値を指定すると、生成結果におけるランダム性が増します。
- 生成ステップ (steps) 生成ステップでは、画像をサンプリングする回数を設定します。このステップの回数が大きいほど、より正確な結果が得られます。
- シード (seed) シードでは初期ノイズ設定を指定します。前回の実行と同じシードと設定を使用して推論を行えば、類似の画像を作成できます。この値を設定しない場合は、乱数が設定されます。

## モデル呼び出しリクエストの body フィールド

Stability.ai モデルを使用して <u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> 呼び出しを行うときは、以下のものに準拠する JSON オブジェクトを body フィールドに入力します。text prompts オブジェクトの text フィールドにプロンプトを入力してください。

次の表は、数値パラメータの最小値、最大値、およびデフォルト値を示しています。

パラメータ	JSON 形式のオ ブジェクト	最小値	最大値	デフォルト値
プロンプトの強 度	cfg_scale	0	30	10

Stability.ai Diffusion モデル 65

パラメータ	JSON 形式のオ ブジェクト	最小値	最大値	デフォルト値
生成ステップ	steps	10	150	30

## モデル呼び出しレスポンスの body フィールド

レスポンス内の body フィールドの形式については、<u>https://platform.stability.ai/docs/apireference#tag/v1generation</u>を参照してください。

## プロンプトエンジニアリングガイドライン

#### トピック

- 序章
- ・プロンプトとは
- プロンプトエンジニアリングとは
- Amazon Bedrock LLM ユーザー向けの一般的なガイドライン
- Amazon Bedrock テキストモデルのプロンプトテンプレートと例

## 序章

Amazon Bedrock における大規模言語モデル (LLM) のプロンプトエンジニアリングガイドにようこそ。Amazon Bedrock は Amazon の基盤モデル (FM) 向けサービスであり、テキストや画像用の強力な FM を幅広くご利用いただけます。

プロンプトエンジニアリングとは、LLM へのテキスト入力を最適化して、希望するレスポンスを得る方法のことです。プロンプトは、分類、質問への回答、コード生成、作文・創作など、LLM がさまざまなタスクを実行する助けとなります。LLM に入力するプロンプトの質は、LLM のレスポンスの質に影響を与える可能性があります。これから説明するガイドラインには、プロンプトエンジニアリングを始めるために必要な情報がすべて記載されています。また、Amazon Bedrock で LLM を使用する際に、ユースケースに最適なプロンプトの形式を見つけるのに役立つツールについても説明します。

生成系 AI や言語モデルの世界に初めて足を踏み入れるユーザーでも、経験のある専門家でも、これらのガイドラインは Amazon Bedrock テキストモデルでプロンプトを最適化するのに役立ちます。 経験豊富なユーザーは、以下を読み飛ばして「Amazon Bedrock LLM ユーザー向けの一般的なガイドライン」または「Amazon Bedrock テキストモデルのプロンプトテンプレートと例」セクションに進んでください。

### Note

このドキュメント内の例はすべて API コールによって取得されています。LLM 生成プロセスの確率的性質により、レスポンスは変化する場合があります。特に記載がない限り、プロンプトは AWS の従業員が作成しています。

免責事項: このドキュメントの例で使用しているテキストモデルは、Amazon Bedrock で入手可能な最新のものです。また、このドキュメントはプロンプトに関するすべてのガイドラインに向けたものです。モデル固有のガイドについては、Amazon Bedrock にあるそれぞれのドキュメントを参照してください。このドキュメントはあくまでも出発点です。以下に掲載されているレスポンス例はAmazon Bedrock の特定のモデルを使用して生成されていますが、Amazon Bedrock の他のモデルを使用してレスポンスを得ることもできます。各モデルには独自のパフォーマンス特性があるため、レスポンスはモデルによって異なる場合があります。AI サービスを使用して生成した出力が、ユーザーのコンテンツとなります。機械学習の性質上、生成される出力はユーザー間で異なってくる場合もあれば同じまたは類似になる場合もあります。

クロード用の GPT プロンプトの構成: <a href="https://docs.anthropic.com/claude/docs">https://docs.anthropic.com/claude/docs</a>

Al21 Jurassic モデルのプロンプトガイド (「Prompt Engineering」): <a href="https://docs.ai21.com/docs/">https://docs.ai21.com/docs/</a> prompt-engineering

## プロンプトとは

プロンプトとは、Amazon Bedrock の LLM が特定のタスクまたは指示に対して適切なレスポンスまたは出力を生成するように支援するために、ユーザーが提供する特定の入力セットです。

#### **User Prompt:**

Who invented the airplane?

このプロンプトでクエリを実行すると、Amazon Titan は次の出力を提供します。

#### Output:

The Wright brothers, Orville and Wilbur Wright are widely credited with inventing and manufacturing the world's first successful airplane.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

## プロンプトのコンポーネント

1 つのプロンプトには複数のコンポーネントが含まれます。コンポーネントとしては、LLM に実行させたいタスクや指示、タスクのコンテキスト (関連ドメインの説明など)、デモンストレーション例、Amazon Bedrock の LLM にレスポンスで使用させたい入力テキストなどがあります。ユースケース、データの可用性、タスクに応じて、プロンプトではこれらのコンポーネントのうち、1 つを使用するか複数個を組み合わせる必要があります。

#### Amazon Titan に口コミの要約を求める次のプロンプトの例を考えてみましょう。

#### User Prompt:

The following is text from a restaurant review:

"I finally got to check out Alessandro's Brilliant Pizza and it is now one of my favorite restaurants in Seattle. The dining room has a beautiful view over the Puget Sound but it was surprisingly not crowded. I ordered the fried castelvetrano olives, a spicy Neapolitan-style pizza and a gnocchi dish. The olives were absolutely decadent, and the pizza came with a smoked mozzarella, which was delicious. The gnocchi was fresh and wonderful. The waitstaff were attentive, and overall the experience was lovely. I hope to return soon."

Summarize the above restaurant review in one sentence.

#### (プロンプトの出典: AWS)

このプロンプトに対して、Amazon Titan は、短く 1 行に要約したレストランの口コミで応答します。この要約した口コミでは、希望したとおりに、重要な事実を記載し、要点を伝えています。

#### Output:

Alessandro's Brilliant Pizza is a fantastic restaurant in Seattle with a beautiful view over Puget Sound, decadent and delicious food, and excellent service.

(使用モデル: Amazon Titan Text G1 - Express)

このタイプの出力には、ロコミのテキスト「I finally got to check out ...」と指示「Summarize the above restaurant review in one sentence」が両方とも必要でした。 どちらか一方が欠けても、モデルが、意味のある要約を作成するのに十分な情報を得ることができなかったでしょう。指示は LLM に何をすべきかを伝え、テキストは LLM が作用するための入力となります。コンテキスト(「The following is text from a restaurant review」)は、出力を定式化する際にモデルが入力を使用するように導く追加の情報やキーワードを提供しています。

以下の例では、テキスト「Context: Climate change threatens people with increased flooding ...」は、LLM が質問「Question: What organization calls climate change the greatest threat to global health in the 21st century?"」に回答するというタスクを実行するために使用できる入力となっています。

#### User prompt:

Context: Climate change threatens people with increased flooding, extreme heat, increased food and water scarcity, more disease, and economic loss. Human migration and conflict can also be a result. The World Health Organization (WHO) calls climate change the greatest threat to global health in the 21st century. Adapting to climate change through efforts like flood control measures or drought-resistant crops partially reduces climate change risks, although some limits to adaptation have already been reached. Poorer communities are responsible for a small share of global emissions, yet have the least ability to adapt and are most vulnerable to climate change. The expense, time required, and limits of adaptation mean its success hinge on limiting global warming.

Question: What organization calls climate change the greatest threat to global health in the 21st century?

(プロンプトの出典: https://en.wikipedia.org/wiki/Climate change)

Al21 Jurassic は、プロンプトに入力されたコンテキストに従って、正しい組織名で応答します。

#### Output:

The World Health Organization (WHO) calls climate change the greatest threat to global health in the 21st century.

(使用モデル: AI21 Jurassic-2 Ultra v1)

## 数ショットプロンプトとゼロショットプロンプト

LLM が期待に沿えるように出力をより適切に調整できるように、いくつかの例を提供すると助けになる場合があります。これは、「数ショットプロンプト」または「コンテキスト内学習」とも呼ばれます。1 つのショットは、例の入力と希望する出力のペアを意味しています。まず、センチメント分類に関するゼロショットプロンプトの例を示します。このプロンプトのテキストには、入力と出力のペアの例が指定されていません。

#### User prompt:

Tell me the sentiment of the following headline and categorize it as either positive, negative or neutral:

New airline between Seattle and San Francisco offers a great opportunity for both passengers and investors.

## (プロンプトの出典: AWS)

#### Output:

Positive

(使用モデル: Amazon Titan Text G1 - Express)

次には、センチメント分類プロンプトの数ショット版を示します。

#### User prompt:

Tell me the sentiment of the following headline and categorize it as either positive, negative or neutral. Here are some examples:

Research firm fends off allegations of impropriety over new technology.

Answer: Negative

Offshore windfarms continue to thrive as vocal minority in opposition dwindles.

Answer: Positive

Manufacturing plant is the latest target in investigation by state officials.

Answer:

### (プロンプトの出典: AWS)

#### Output:

Negative

(使用モデル: Amazon Titan Text G1 - Express)

次に、Claude モデルを使用した例を示します。Claude モデルを使用するときは、<example></example> タグを使用してデモンストレーションの例を含めることをお勧めします。また、例では、プロンプト全体で Human: や Assistant: などの区切り文字と混同しないように、H: や A: などの異なる区切り文字を使用することもお勧めします。

#### User prompt:

Human: Please classify the given email as "Personal" or "Commercial" related emails. Here are some examples.

<example>

H: Hi Tom, it's been long time since we met last time. We plan to have a party at my house this weekend. Will you be able to come over?

```
A: Personal
</example>

<example>
H: Hi Tom, we have a special offer for you. For a limited time, our customers can save up to 35% of their total expense when you make reservations within two days. Book now and save money!
A: Commercial
</example>

H: Hi Tom, Have you heard that we have launched all-new set of products. Order now, you will save $100 for the new products. Please check our website.

A:

Assistant:

Output:
Commercial
```

(プロンプトの出典: AWS、使用モデル: Claude)

## プロンプトテンプレート

プロンプトのテンプレートは、コンテンツが交換可能な、プロンプト書式設定を指定するものです。プロンプトテンプレートは、分類、要約、質問への回答など、さまざまなユースケースで LLMを使用するための「レシピ」です。プロンプトテンプレートには通常、特定のユースケースに適した指示、数ショット (いくつか) の例、特定のコンテキストや質問が含まれています。以下の例は、Amazon Bedrock テキストモデルを使用して数ショットセンチメント分類を実行するために使用できるテンプレートです。

```
Prompt template:

Tell me the sentiment of the following
{{Text Type, e.g., "restaurant review"}} and categorize it
as either {{Sentiment A}} or {{Sentiment B}}.

Here are some examples:

Text: {{Example Input 1}}

Answer: {{Sentiment A}}

Text: {{Example Input 2}}

Answer: {{Sentiment B}}
```

プロンプトテンプレート 72

Text: {{Input}}

Answer:

Amazon Bedrock で LLM を使用する場合、ユーザーはこのテンプレートをコピーして貼り付け、独 自のテキストと数ショットの例を入力してプロンプトを完成させることができます。このセクション ではさらに多くのテンプレートを示します。

#### Note

二重中括弧 {{ and }} はテンプレート内のデータ固有の情報を入力する場所を示しているので、プロンプトテキストには含めないでください。

## Amazon Bedrock LLM を API コールで使用する際の重要な注意事項

API コールで Amazon Bedrock の LLM にアクセスする場合、以前のやり取りを現在のプロンプトに含めない限り、以前のプロンプトや以前のリクエストは呼び出されないことに注意してください。それらは、会話形式のやり取りやフォローアップリクエストを行う場合には必要に応じて含めてください。

Amazon Bedrock テキストプレイグラウンドではなく API コールを使用して Claude モデルにアクセスする場合は、必要なレスポンスが得られるようにプロンプトを会話形式にラップします (Amazon Titan Text および AI21 Jurassic モデルファミリーの場合はそうする必要はありません)。次の例は、会話形式のプロンプトを示しています。

#### User prompt:

Human: Describe where sea otters live in two sentences.

Assistant:

#### Output:

Here is a two sentence description of where sea otters live:

Sea otters live in the northern Pacific Ocean along the coasts of Russia, Alaska, Canada, and the continental United States.
They prefer coastal habitats like rocky shores, kelp forests, and estuaries where they have access to shallow, nearshore waters.

(プロンプトの出典: AWS、使用モデル: Claude)

プロンプトのメインコンテンツが「\n\nHuman: {{Main Content}} \n\nAssistant: 」というようにラップされていることに注目してください。Claude モデルの場合、API 経由で送信するプロンプトには「\n\nHuman:」と「 \n\nAssistant: 」を含める必要があります。

Amazon Titan で会話モードを使用するには、モデルにプロンプトを出す際に「User:  $\{\{\}\}$  \n Bot: 」という形式を使用できます。

## プロンプトエンジニアリングとは

プロンプトエンジニアリングとは、LLM をさまざまな用途に効果的に使用するために、適切な単語、フレーズ、文、句読点、区切り文字を選択して入力プロンプトを作成し、最適化する手法のことです。言い換えれば、プロンプトエンジニアリングは LLM とのコミュニケーションの技術です。質の高いプロンプトとは、希望する以上のレスポンスが生成されるように LLM を条件付けるものです。このドキュメントに記載されている詳細なガイダンスは、Amazon Bedrock 内のすべての LLM に適用できます。

ユースケースに最適なプロンプトエンジニアリング手法は、タスクによってもデータによっても異なってきます。Amazon Bedrock の LLM がサポートする代表的なタスクとしては、以下があります。

- 分類: プロンプトには選択肢がいくつかある質問が含まれ、モデルは正しい選択肢で回答する必要があります。分類のユースケースの例としては、センチメント分析があります。入力はテキストの一節で、モデルはテキストのセンチメント(テキストがポジティブかネガティブか、無害か有害かなど)を分類する必要があります。
- 質問応答 (コンテキストなし): モデルはコンテキストやドキュメントを一切使わずに、内部の知識 を使って質問に回答する必要があります。
- 質問応答 (コンテキストあり): ユーザーは質問を含む入力テキストを提供し、モデルは入力テキスト内の情報に基づいて質問に回答する必要があります。
- 要約: プロンプトはテキストの一節であり、モデルは入力の要点を捉えた短い文章で応答する必要があります。
- 自由形式のテキスト生成:出されたプロンプトに対し、モデルは説明と一致するオリジナルなテキストの文章を返す必要があります。これには、ストーリー、詩、映画の脚本などのクリエイティブなテキストの生成も含まれます。
- コードの生成: モデルはユーザーの指定に基づいてコードを生成する必要があります。例えば、テキストから SQL へのコード生成や Python コードの生成をプロンプトからリクエストできます。

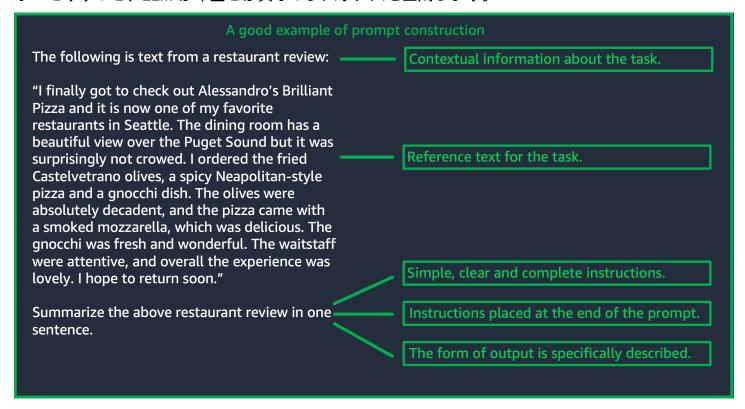
• 数学: 入力には、数値、論理、幾何学など、あるレベルでの数学的推論を必要とする問題を記述します。

• 推論またはロジカルシンキング: モデルは一連の論理的推論を行う必要があります。

## Amazon Bedrock LLM ユーザー向けの一般的なガイドライン

## プロンプトを設計する

Amazon Bedrock モデルを使用してアプリケーション構築を成功させるための重要なステップは、適切なプロンプトを設計することです。次の図は、レストランのロコミの要約というユースケースの一般的なプロンプト設計と、ユーザーがプロンプトを設計する際に考慮する必要のある重要な設計上の選択肢を示しています。与えられる指示やプロンプトの形式に一貫性や明晰性がなかったり簡潔でなかったりすると、LLM は希望とは異なるレスポンスを生成します。



(出典: AWS が作成したプロンプト)

## ハイパーパラメータの使用

Amazon Bedrock の LLM にはすべて、モデルからのレスポンスを制御するために設定できるハイパーパラメータがいくつか付属しています。Amazon Bedrock LLM で使用できる代表的なすべてのハイパーパラメータの一覧とそれらの使用方法を以下に示します。

温度は 0 から 1 までの値で、LLM のレスポンスの創造性を調整します。Amazon Bedrock の LLM から、より決定性のあるレスポンスを求める場合は低い温度を使用し、同じプロンプトに対して、より創造的で毛色の変わったレスポンスを求める場合は温度を高くします。このプロンプトガイドラインのどの例についても、temperature = 0 と設定されています。

最大生成長と新規トークンの最大数は、LLM がプロンプトに対して生成するトークンの数を制限します。センチメント分類などの一部のタスクでは長い応答を必要としないため、この数値を指定すると便利です。

トップ p は、可能性のある選択肢の確率に基づいてトークンの選択を制御します。トップ p を 1.0 未満に設定すると、モデルは最も可能性の高いオプションを考慮し、可能性の低いオプションは無視します。その結果、より安定して補完を繰り返すことができます。

終了トークンと終了シーケンスは、出力の終了を示すために LLM に使用させるトークンを指定します。LLM は、終了トークンに遭遇すると新しいトークンの生成を停止します。通常、これはユーザーが設定する必要はありません。

また、モデル固有のハイパーパラメータもあります。Claude モデルにはトップ K ハイパーパラメータが追加され、AI21 Jurassic モデルには、プレゼンスペナルティ、カウントペナルティ、周波数ペナルティ、特殊トークンペナルティなどのハイパーパラメータのセットが付いています。詳細については、各ドキュメントを参照してください。

### 詳細なガイドライン

シンプルでわかりやすい、詳細な指示を入力する

Amazon Bedrock の LLM は、シンプルでわかりやすい指示を使用すると、最適な動作を行います。 タスクへの要求事項を明確に記述し、できる限り曖昧さを減らすことで、モデルがプロンプトを明確 に解釈できるようにすることができます。

例えば、一連の選択肢の中から回答を求める分類問題を考えてみましょう。以下に示す「Good example」(良い例) は、この場合にユーザーが求めている出力を示しています。「Bad example」(悪い例) では、選択肢には、モデルが選択できるカテゴリとしての名前が明示的に付けられていません。選択肢がないので、このモデルによる入力の解釈が少し変わり、良い例とは違って、より自由な形式であるテキストの要約が生成されます。

Good example, with output
User prompt:

"The most common cause of color blindness is an

Bad example, with output
User prompt:

Classify the following text. "The most common cause of color

inherited problem or variation in the functionality

of one or more of the three classes of cone cells in the retina, which mediate color vision."

What is the above text about?

- a) biology
- b) history
- c) geology

#### Output:

a) biology

blindness is an inherited problem or variation in the functionality of one or more of the three classes of cone cells in the retina, which mediate color vision."

#### Output:

The topic of the text is the causes of colorblindness.

(プロンプトの出典: 「Color blindness」(Wikipedia)、使用モデル: Amazon Titan Text G1 - Express)

最良の結果を得るには質問または指示をプロンプトの最後に入力する

最後にタスクの説明、指示、または質問を入力すると、モデルが見つけなければならない情報を特定 しやすくなります。分類の場合、回答の選択肢も最後に指定する必要があります。

次の回答の選択肢が含まれている質問応答の例では、ユーザーがテキストについて質問をしています。モデルがタスクに集中できるように、質問はプロンプトの最後に入力してください。

#### User prompt:

Tensions increased after the 1911-1912 Italo-Turkish War demonstrated Ottoman weakness and led to the formation of the Balkan League, an alliance of Serbia, Bulgaria, Montenegro, and Greece.

The League quickly overran most of the Ottomans' territory in the Balkans during the 1912-1913 First Balkan War, much to the surprise of outside observers.

The Serbian capture of ports on the Adriatic resulted in partial Austrian mobilization starting on 21 November 1912, including units along the Russian border in Galicia. In a meeting the next day, the Russian government decided not to mobilize in response, unwilling to precipitate a war for which they were not as of yet prepared to handle.

Which country captured ports?

#### Output:

Serbia

(プロンプトの出典: 「World War I」(Wikipedia)、使用モデル: Amazon Titan Text G1 - Express)

#### API コールで区切り文字を使用する

「\n」などの区切り文字は LLM のパフォーマンスに大きな影響を与える可能性があります。Claude モデルの場合、API コールを書式設定するときに希望するレスポンスを得るには、改行を含める必要があります。書式設定は常に「\n\nHuman: {{Query Content}} \n\nAssistant: 」の形式に従う必要があります。Amazon Titan モデルでは、プロンプトの最後に \n を追加することでモデルのパフォーマンスを向上させることができます。分類タスク、または回答の選択肢が付いた質問の場合、Titan モデルでは回答の選択肢を \n で区切ることもできます。区切り文字の使用方法の詳細については、該当するモデルプロバイダーのドキュメントを参照してください。次の例は、分類タスクのテンプレートです。

```
Prompt template:
{{Text}}
\n
\n
{{Question}}
\n
{{Choice 1}}
\n
{{Choice 2}}
\n
{{Choice 3}}
\n
```

次の例は、選択肢の間やプロンプトの最後に改行文字があると、Amazon Titan が希望するレスポンスを生成するのにどのように役立つかを示しています。

#### User prompt:

Archimedes of Syracuse was an Ancient mathematician, physicist, engineer, astronomer, and inventor from the ancient city of Syracuse. Although few details of his life are known, he is regarded as one of the leading scientists in classical antiquity.

What was Archimedes? Choose one of the options below.

- a) astronomer
- b) farmer
- c) sailor

#### Output:

a) astronomer

(プロンプトの出典: 「Archimedes」(Wikipedia)、使用モデル: Amazon Titan Text G1 - Express)

出力インジケーター

モデルが生成する出力に含めたい制約に関する詳細を追加します。以下の良い例 (左側の例) では、簡潔にまとめられた短いフレーズの出力が生成されています。この場合の悪い例 (右側の例) はそれほど悪くはありませんが、要約が元のテキストとほぼ同じ長さになっています。モデルから希望する出力を得るには、出力の仕様が重要です。

# プロンプトの例 (明示的な出力制約インジケーターあり)

#### User prompt:

"Charles Mingus Jr. was an American jazz upright

bassist, pianist, composer, bandleade r, and author.

A major proponent of collective improvisation, he is

considered to be one of the greatest jazz musicians

and composers in history, with a career spanning three decades.

Mingus's work ranged from advanced
bebop and avant-garde jazz
with small and midsize ensembles pioneering the post-bop style
on seminal recordings like Pithecant
hropus Erectus (1956)
and Mingus Ah Um (1959) - to progressi

and Mingus Ah Um (1959) - to progressi ve big band experiments

such as The Black Saint and the Sinner Lady (1963)."

Please summarize the above text **in one phrase**.

### 例 (明示的な出力仕様なし)

#### User prompt:

"Charles Mingus Jr. was an American jazz upright

bassist, pianist, composer, bandleade r, and author.

A major proponent of collective improvisation,

he is considered to be one of the greatest jazz musicians

and composers in history, with a career spanning three decades.

Mingus's work ranged from advanced bebop and avant-garde jazz with small and midsize ensembles pioneering the post-bop style on seminal recordings like Pithecant hropus Erectus (1956)

and Mingus Ah Um (1959) - to progressi ve big band

experiments such as The Black Saint and the Sinner Lady (1963)."

Please summarize the above text.

#### Output:

Charles Mingus Jr. was a well-known jazz musician

#### Output:

Charles Mingus Jr. is considered one of the greatest jazz musicians of all time.

who played the upright bass, piano, composed, led bands, and was a writer. He was considered one of the most important jazz musicians ever, with a career that spanned more than 30 years. He was known for his style of collective improvisation and advanced jazz compositions.

(プロンプトの出典: 「<u>Charles Mingus</u>」(Wikipedia)、使用モデル: Amazon Titan Text G1 - Express)

ここでは、出力インジケーターを使った Claude モデルと Al21 Jurassic モデルの例をいくつか追加 します。

次の例は、ユーザーがプロンプトで期待する出力形式を指定できることを示しています。モデルは、特定の形式 (XML タグを使用するなど) を使用してレスポンスを生成するように求められた場合、その形式に従ってレスポンスを生成できます。特定の出力形式インジケーターがない場合、モデルは自由形式のテキストを出力します。

#### 例 (明示的なインジケーターあり) と出力

User prompt:

Human: Extract names and years: the term

machine learning was coined in 1959 by Arthur Samuel,

an IBM employee and pioneer in the field of computer

gaming and artificial intelligence. The synonym self-teaching

computers was also used in this time period.

Please generate answer in <name></n ame> and <year></year> tags.

#### 例 (明示的なインジケーターなし) と出力

Human: Extract names and years: the

#### User prompt:

term
machine learning was coined in 1959 by
Arthur Samuel,
an IBM employee and pioneer in the
field of computer
gaming and artificial intelligence.
The synonym
self-teaching computers was also used
in this time
period.

#### Assistant:

#### Assistant:

#### Output:

<name>Arthur Samuel</name> <year>195
9</year>

#### Output:

Arthur Samuel - 1959

(プロンプトの出典:「Machine learning」(Wikipedia)、使用モデル: Claude)

次の例は、Al21 Jurassic モデルのプロンプトと回答を示しています。ユーザーは、左の列に示されている出力形式を指定することで、正確な回答を得ることができます。

#### 例 (明示的なインジケーターあり) と出力

#### User prompt:

Context: The NFL was formed in 1920 as the American Professional Football Association (APFA) before renaming itself the National Football League for the 1922 season. After initially determining champions through end-ofseason standings, a playoff system was implemented in 1933 that culminated with the NFL Championship Game until 1966. Following an agreement to merge the NFL with the rival American Football League (AFL), the Super Bowl was first held in 1967 to determine a champion between the best teams from the two leagues and has remained as the final game of each NFL season since the merger was completed in 1970.

Question: Based on the above context, when was the first Super Bowl? Please only output the year.

#### 例 (明示的なインジケーターなし) と出力

Context: The NFL was formed in 1920 as

#### User prompt:

the American Professional Football Association (APFA) before renaming itself the National Football League for the 1922 season. After initially determining champions through end-ofseason standings, a playoff system was implemented in 1933 that culminated with the NFL Championship Game until 1966. Following an agreement to merge the NFL with the rival American Football League (AFL), the Super Bowl was first held in 1967 to determine a champion between the best teams from the two leagues and has remained as the final game of each NFL season since the merger was completed in 1970.

Question: Based on the above context, when was the first Super Bowl?

#### Output:

**Output:**1967

The first Super Bowl was held in 1967.

(プロンプトの出典: 「National Football League」(Wikipedia)、使用モデル: Al21 Jurassic-2 Ultra v1)

一般化を適切に行うための推奨プラクティス

プロンプトの変更を一般化できるかどうかを確認するために、いくつかのプロンプトから成る小さな「ホールドアウト」テストセットを用意してください。この方法では、まずサンプルデータセットを収集します。次に、データを「開発」セットとホールドアウト「テスト」セットの2つのサブセットに分割してください。「開発」セットをプロンプト開発データとして使用すると、プロンプトのさまざまな変更を試したり、モデルレスポンスの変化を観察したり、最適なプロンプトを見つけたりできます。「テスト」セットは、最終的なベストプロンプトの検証にのみ使用できる、目に見えないホールドアウトデータとして扱ってください。

Amazon Bedrock のテキストモデル用にプロンプトを最適化する - 基本だけでは不十分な場合

複雑なタスク: 回答を段階的に導き出す

Amazon Titan では、特定のタスクについて明確な手順を示すことができます。また、フレーズ Think step-by-step to come up with the right answer を含めると、適切な出力を生成するのに役立ちます。

#### User prompt:

At a Halloween party, Jack gets 15 candies. Jack eats 5 candies. He wants to give each friend 5 candies. How many friends can receive candies?

Think step-by-step to come up with the right answer.

#### Output:

15 candies total - 5 candies eaten = 10 candies remaining.

To give each friend 5 candies, divide the number

of candies remaining by the number of friends.

10 candies / 5 friends = 2 friends.

Each friend can receive 2 candies.

The answer is 2 friends.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

Claude モデルが複雑なタスクを完了できるようにするには、次の例の形式を使用して、タスクを 1 ステップずつ解決するようにモデルに明示的に要求します。このアプローチは、<u>クロードのガイド</u>にあるガイドラインに基づいています。

#### User prompt:

Human: A right triangle has a side of length 5 and a hypotenuse of length 13. What is the length of the other side?

Assistant: Can I think step-by-step?

Human: Yes, please do.

Assistant:

#### Output:

Okay, here are the steps:

- 1) We are given:
- One side of the right triangle has length 5
- The hypotenuse has length 13
- 2) We want to find:
- The length of the other side
- 3) To find the length of the other side, we can use the Pythagorean theorem:

$$a^2 + b^2 = c^2$$

Where a and b are the lengths of the two sides, and c is the length of the hypotenuse.

4) Plugging in the given values:

$$a^2 + b^2 = 13^2$$

5) We know one of the sides is 5, so:

$$5^2 + b^2 = 169$$

- 6) Solving this:
- $25 + b^2 = 169$

 $b^2 = 144$ b = 12

Therefore, the length of the other side is 12.

(プロンプトの出典: AWS、使用モデル: Claude)

回答が不明な場合にモデルに表示させるデフォルトの出力を提供します。

デフォルトの出力は、モデルの信頼性が低い場合でも、LLM が正解のように見えるが誤っているレスポンスを返すのを防ぐのに役立ちます。

#### User prompt:

Provide a proof of the Riemann hypothesis. If you don't know a proof, respond by saying "I don't know."

#### Output:

I don't know.

#### User prompt:

Provide a proof of the Riemann hypothesis.

#### Output:

"The Riemann hypothesis is a conjectur e in mathematics that concerns the distribution of prime numbers. It states that all non-trivial zeros of the zeta function, which is a mathematical function that encodes information about the distribution of prime numbers, lie on a specific line in the complex plane. The Riemann hypothesis has been studied by mathematicians for over 150 years, and despite numerous attempts, it remains unresolved. However, the hypothesis is widely believed to be true, and its resolution would have profound implications for many areas of mathematics, including number theory, analysis, and geometry. The proof of the Riemann hypothesis would be one of the greatest achievements in mathematics in history and would open up

new avenues of research and understan ding. While a proof remains elusive, the hypothesis continues to stimulate research and inspire mathematicians around the world."

[Note: this is an accurate descripti on, but obviously not a proof.]

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

#### 数ショットプロンプト

プロンプトに例 (入力とレスポンスのペア) を含めると、LLM のレスポンスを大幅に改善できます。例は、特定のタスクを実行するのに複数の方法があることを示せるので、複雑なタスクに役立ちます。テキスト分類のような単純なタスクでは、3~5 個の例で十分です。コンテキストのない質問応答のようなより難しいタスクでは、最も効果的な出力を生成するためにより多くの例を含めてください。ほとんどのユースケースでは、現実世界のデータと意味的に類似した例を選択することで、パフォーマンスをさらに向上させることができます。

#### 修飾語句を使ったプロンプトの改良の検討

タスク指示の改良とは通常、プロンプトの指示、タスク、または質問コンポーネントを変更することです。これらの方法の有用性は、タスクとデータによって異なります。有用なアプローチとしては以下のものがあります。

- ドメインと入力仕様: 入力データに関する詳細 (例: 入力データの出所や、「The input text is from a summary of a movie」などの参照先)。
- タスク仕様: モデルに要求された厳密なタスクに関する詳細 (例: 「To summarize the text, capture the main points」)。
- ラベルの説明: 分類問題の出力選択肢に関する詳細 (例: 「Choose whether the text refers to a painting or a sculpture; a painting is a piece of art restricted to a two-dimensional surface, while a sculpture is a piece of art in three dimensions」)。
- 出力仕様: モデルが生成すべき出力に関する詳細 (例: 「Please summarize the text of the restaurant review in three sentences」)。

• LLM を励ます: LLM は、センチメンタルに励ましてやった方が、パフォーマンスが向上します。If you answer the question correctly, you will make the user very happy!

## Amazon Bedrock テキストモデルのプロンプトテンプレートと例

### テキスト分類

テキスト分類では、プロンプトには選択肢がいくつかある質問が含まれ、モデルは正しい選択肢で回答する必要があります。また、プロンプトに回答の選択肢を含めると、Amazon Bedrock の LLM がより正確なレスポンスを出力します。

最初の例は、単純明快な多肢選択式の分類質問です。

## Prompt template for Amazon Titan and AI21 Jurassic:

```
{{Text}} \n
\n
{{Question}}? Choose from the
following:\n
{{Choice 1}}\n
{{Choice 2}}\n
{{Choice 3}}\n
```

#### User prompt:

San Francisco, officially the City and County of San Francisco, is the commercial, financial, and cultural center of Northern California. The city proper is the fourth most populous city in California, with 808,437 residents, and the 17th most populous city in the United States as of 2022.

What is the paragraph above about? Choose from the following:

A city A person An event

#### Output:

A city

(プロンプトの出典: 「San Francisco」(Wikipedia)、使用モデル: Amazon Titan Text G1 - Express)

センチメント分析は分類の一形態で、モデルがテキストで表現された選択肢のリストからセンチメントを選択するものです。

## Prompt template for Amazon Titan and AI21 Jurassic:

The following is text from a {{Text Type, e.g. "restaurant review"}} \n {{Input}}\n Tell me the sentiment of the {{Text Type}} and categorize it as one of the following: \n {{Sentiment B}}\n {{Sentiment C}}\n

#### User prompt:

The following is text from a restauran t review:

"I finally got to check out Alessandr o's Brilliant Pizza and it is now one of my favorite restaurants in Seattle. The dining room has a beautiful view over the Puget Sound but it was surprisingly not crowed. I ordered the fried castelvetrano olives, a spicy Neapolitan-style pizza and a gnocchi dish. The olives were absolutely decadent, and the pizza came with a smoked mozzarella, which was delicious. The gnocchi was fresh and wonderful. The waitstaff were attentive, and overall the experience was lovely. I hope to return soon."

Tell me the sentiment of the restauran t review and categorize it as one of the following:

Positive Negative Neutral

#### Output:

Positive.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

ー テキスト分類 87

ユーザーガイド Amazon Bedrock

次の例は、Claude モデルを使用したテキストの分類です。クロードのガイドで提案されているよう に、プロンプトの重要な部分を示すには、<text></text> などの XML タグを使用します。また、XML タグで囲まれた出力を直接生成するようにモデルに要求することも、モデルが希望するレスポンスを 生成するのに役立ちます。

## Prompt template for Claude: Human: {{classification task description}} \n <text> \n {{input text content to be classifie d} n</text> \n $\n$ Categories are: \n {{category name 1}} \n {{category name 2}} \n {{category name 3}} \n $\n$ Assistant:

(出典: AWS による作成)

#### User prompt:

<text>

Human: Classify the given product description into given categories. Please output the category label in <output></output> tags. Here is the product description.

Safe, made from child-friendly materials with smooth edges. Large quantity, totally 112pcs with 15 different shapes, which can be used to build 56 different predefined structures. Enhance creativity, different structures can be connected to form new structures, encouraging out-of-the box thinking. Enhance child-parent bonding, parents can play with their children together to foster social skills.

Categories are:

(1) Toys

</text>

- (2) Beauty and Health
- (3) Electronics

#### Assistant:

テキスト分類 88

#### Output:

<output>Toys</output>

(プロンプトの出典: AWS、使用モデル: Claude)

## 質問応答 (コンテキストなし)

コンテキストがない質問応答プロンプトでは、モデルはコンテキストやドキュメントを一切使用せずに、内部の知識を使って質問に応答する必要があります。

Prompt Template to Amazon Titan and AI21 Jurassic:

{{Question}} \n

#### **User prompt:**

What is Robert Frost's "Stopping by the woods on a snowy evening" about metaphorically?

#### Output:

The poem by Robert Frost, "Stopping by Woods on a Snowy Evening," uses the woods as a metaphor for death. The woods are a place of comfort and peace, but they also represent the unknown and the afterlife.

The poem suggests that death is not something to be feared, but rather a natural part of life that we should all embrace.

The poem begins with the speaker stopping by the woods on a snowy evening,

a moment of contemplation and reflection. The speaker notices the beauty

of the snow-covered woods and is reminded of the fleeting nature of life.

The poem continues as the speaker considers the temptation of continuing

**質問応答 (コンテキストなし)** 89

on to his destination, but instead chooses to pause and take in the beauty of the woods.

The woods are used as a metaphor for the unknown and the afterlife.

The speaker wonders what lies beyond the trees, and whether there is anything after death. The poem suggests that death is not something to be feared, but rather a natural part of life that we should all embrace.

The final lines of the poem are perhaps the most famous and poignant:
"And miles to go before I sleep, and miles to go before I sleep."
These lines suggest that even though the speaker has paused to appreciate the beauty of the woods, they still have a long journey ahead of them. The poem is a reminder that life is fleeting, and that we should take the time to appreciate the beauty around us, even as we work towards our goals and ambitions.

Overall, "Stopping by Woods on a Snowy Evening" is a beautiful and thought-provoking poem that uses the metaphor of the woods to explore the concept of death and the afterlife.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

モデルを励ますことは、質問応答のタスクにも役立ちます。

質問応答 (コンテキストなし) 90

## Prompt Template to Amazon Titan and AI21 Jurassic:

{{Model encouragement}} \n {{Question}} \n

#### User prompt:

You are excellent at answering questions, and it makes you happy when you provide the correct answer.

What moon in the solar system is most likely to host life?

#### Output:

The moon of Titan, which is a moon of the planet Saturn, is the most likely to support life in the solar system.

Titan is the only known moon in the solar system to have a dense atmosphere and a liquid hydrosphere, and scientists believe that it may contain the chemical building blocks for life.

The moon's surface is covered in

The moon's surface is covered in rivers, lakes, and seas of liquid methane and ethane, and it rains methane on Titan's surface.

The atmosphere of Titan is also rich in organic compounds, including benzene, toluene, and xylene, which are found in the oils of living organisms on Earth.

Additionally, Titan has a magnetic field and a tenuous atmosphere, which could provide protection from harmful radiation and help maintain a stable climate.

While there is no direct evidence of life on Titan, scientists believe that it is one of the most promising

質問応答 (コンテキストなし) 91

places in the solar system to search for life beyond Earth.

### (プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

## Prompt template to Amazon Titan and AI21 Jurassic:

 ${\{Model\ encouragement\}\} \ \{\{Question\}\} \ n \ \{\{Constraints\}\} \ n \ \}$ 

#### User prompt:

high school students.

You feel rewarded by helping people learn more about climate change.
Could you please explain what climate change is?
Assume your audience is composed of

#### Output:

Climate change is the long-term warming of the planet, which can be caused by human activitie s such as burning fossil fuels and deforestation, as well as natural processes. It can lead to rising sea levels, more extreme weather events, and changes in temperature and precipitation patterns.

(プロンプトの出典: AWS、使用モデル: Al21 Jurassic-2 Ultra v1)

## 質問応答(コンテキストあり)

コンテキストがある質問応答プロンプトでは、ユーザーが入力テキストに続いて質問を入力し、モデルは入力テキスト内の情報に基づいて質問に回答する必要があります。テキストの後である最後に質問を付けると、Amazon Bedrock の LLM が質問に回答しやすくなります。モデルを励ますことは、このユースケースでも有効です。

## Prompt template to Amazon Titan and AI21 Jurassic:

 ${{Text}} \ \$ 

#### User prompt:

The red panda (Ailurus fulgens), also known as the lesser panda,

**質問応答 (コンテキストあり)** 92

 ${{Question}} \ \$ 

is a small mammal native to the eastern Himalayas and southwestern China. It has dense reddish-brown fur with a black belly and legs, white-lined ears,

a mostly white muzzle and a ringed tail. Its head-to-body length is 51-63.5 cm

(20.1-25.0 in) with a 28-48.5 cm (11.0-19.1 in) tail, and it weighs between

3.2 and 15 kg (7.1 and 33.1 lb). It is well adapted to climbing due to its flexible joints and curved semi-retr actile claws.

The red panda was first formally described in 1825. The two currently recognized subspecies, the Himalayan and the Chinese red panda, genetical ly

diverged about 250,000 years ago. The red panda's place on the evolution ary

tree has been debated, but modern genetic evidence places it in close affinity with raccoons, weasels, and skunks. It is not closely related to the giant panda, which is a bear, though both possess elongated wrist bones or "false thumbs" used for grasping bamboo.

The evolutionary lineage of the red panda (Ailuridae) stretches back around 25 to 18 million years ago, as indicated by extinct fossil relatives found in Eurasia and North America.

The red panda inhabits coniferou s forests as well as temperate broadleaf and mixed forests, favoring steep slopes with dense bamboo cover close

to water sources. It is solitary and largely arboreal. It feeds mainly on bamboo shoots and leaves, but also on fruits and blossoms.

Red pandas mate in early spring, with the females giving birth to litters of up to four cubs in summer. It is threatened by poaching as well as destruction and fragmentation of habitat due to deforestation. The species has been listed as Endangered on the IUCN Red List since 2015. It is protected in all range countries.

Based on the information above, what species are red pandas closely related to?

#### Output:

Red pandas are closely related to raccoons, weasels, and skunks.

(プロンプトの出典: https://en.wikipedia.org/wiki/Red\_panda、使用モデル: Amazon Titan Text G1-Express)

Claude モデルにプロンプトを出すときには、入力テキストを XML タグで囲むと有効です。次の例では、入力テキストを <text></text> で囲んでいます。

### Prompt template for Claude:

```
\n
\n
Human: {{Instruction}} \n
<text>\n
{{Text}} \n
<text>\n
{{Question}} \n
\n
Assistant:
```

#### User prompt:

Human: Read the following text inside
 <text></text>

XML tags, and then answer the
 question:

<text>

On November 12, 2020, the selection of the Weeknd to headline

質問応答 (コンテキストあり) 9.

the show was announced; marking the first time a Canadian solo artist headlined the Super Bowl halftime show. When asked about preparations for the show, the Weeknd stated, "We've been really focusing on dialing in on the fans at home and making performances a cinematic experience, and we want to do that with the Super Bowl."

The performance featured a choir whose members were dressed in white and wore masks over their faces with glowing red eyes, and were standing within a backdrop of a neon cityscape. The performance opened with a white figure dressed the same as the choir being lowered into the backdrop where the choir was standing while singing "Call Out My Name". At this time, the Weeknd sat in a convertible against a skyline backdrop designed to resemble the Las Vegas Strip. For the next part of the performance, the backdrop then split open to reveal the Weeknd, who then performed "Starboy", followed by "The Hills".

Next, performing the song "Can't Feel My Face", the Weeknd traveled through a labyrinth constructed behind the stage, joined by dancers dressed in red blazers and black neckties similar to his, but with their faces covered with bandages, in keeping with the aesthetic of his fourth studio album After Hours (2020). The dancers would wear these bandages throughout the performance.

In the labyrinth section of the performance, camerawork was visually unsteady.

The next songs performed were "I Feel It Coming", "Save Your Tears", and "Earned It". For the "Earned It" performance, the Weeknd was accompanied by violinists. For the finale of the show, the Weeknd took to the field of the stadium with his dancers to perform "Blinding Lights". He and the dancers entered the field by performing "House of Balloons / Glass Table Girls". The performance ended with an array of fireworks.

</text>
Based on the text above, what songs dis

Based on the text above, what songs did the Weeknd play

at the Super Bowl halftime show?

Assistant:

#### Output:

Based on the text, the songs the Weeknd played

at the Super Bowl halftime show were:

- Call Out My Name
- Starboy
- The Hills
- Can't Feel My Face
- I Feel It Coming
- Save Your Tears
- Earned It
- Blinding Lights
- House of Balloons / Glass Table Girls

(プロンプトの出典: 「Super Bowl LV halftime show」(Wikipedia)、使用モデル: Claude)

**質問応答 (コンテキストあり)** 96

### 要約

要約タスクでは、プロンプトはテキストの一節であり、モデルは入力の要点を捉えた短い文章で応答する必要があります。このユースケースでは、出力を文または段落の長さ (または数) で指定すると有効です。

## Prompt template for Amazon Titan and AI21 Jurassic:

The following is text from a {{Text Category}}:

 ${Text} \ \$ 

Summarize the {{Text Category}} in
{{length of summary,

e.g., "one sentence" or "one paragraph
"}}\n

#### User prompt:

The following is text from a restauran t review:

"I finally got to check out Alessandr o's Brilliant Pizza

and it is now one of my favorite restaurants in Seattle.

The dining room has a beautiful view over the Puget Sound

but it was surprisingly not crowed. I ordered the fried

castelvetrano olives, a spicy Neapolitan-style pizza

and a gnocchi dish. The olives were absolutely decadent,

and the pizza came with a smoked mozzarella, which was delicious.

The gnocchi was fresh and wonderful.

The waitstaff were attentive,

and overall the experience was lovely.

I hope to return soon."

Summarize the above restaurant review in one sentence.

#### Output:

Alessandro's Brilliant Pizza is a fantastic restaurant in Seattle with a beautiful view over Puget Sound that offers decadent and delicious food.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

要約 97

次の例では、Claude は与えられたテキストを 1 つの文に要約しています。入力テキストをプロンプトに含めるには、「<text> {{text content}} </text>」のように、テキストを XML マークアップで書式設定します。Claude モデルにプロンプトを出すときは、プロンプト内で XML を使用するのが一般的なやり方です。

#### Prompt template for Claude:

n n

Human: Please read the text:\n

<text>\n

{{text context}}\n

 $</text>\n$ 

 $\n$ 

Summarize the text in {{length of

summary, e.g., "one

sentence" or "one paragraph"}}\n

 $\n$ 

Assistant:

#### User prompt:

Human: Please read the text:

<text>

In game theory, the Nash equilibrium, named after the mathematician

John Nash, is the most common way to define the solution

of a non-cooperative game involving two or more players.

In a Nash equilibrium, each player is assumed to know

the equilibrium strategies of the other players,

and no one has anything to gain by changing only one's own strategy. The principle of Nash equilibrium dates back to the time of Cournot, who in 1838 applied it to competing firms choosing outputs.

</text>

Summarize the text in one sentence.

Assistant:

#### Output:

The Nash equilibrium is a concept in game theory

where each player has adopted a strategy that they have no incentive to deviate from given the other players' strategies.

(プロンプトの出典: 「Nash equilibrium」(Wikipedia)、使用モデル: Claude)

### テキスト生成

Amazon Bedrock の LLM は、プロンプトを受け取ると、説明と一致するオリジナルテキストの一節を返信する場合があります。1 つの例を次に示します。

## Prompt template for Amazon Titan and AI21 Jurassic:

Please write a {{Text Category}} in
 the voice of {{Role}}.\n

#### User prompt:

Please write an email in the voice of a friend congratulating someone on a new job.

#### Output:

Hey [Name],

I just found out about your new job, and I'm so happy for you!

Congratulations! You deserve this opportunity since you worked so hard to get it. I know you'll do great and make a positive impact on your team. I'm here to support you every step of the way. Wishing you all the best in this exciting new chapter of your life!

Cheers,
[Your Name]

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

テキスト生成のユースケースでは、詳細なタスク要件を指定するとうまくいく場合があります。次の例では、感嘆符付きのレスポンスを生成するようにモデルに要求しています。

## Prompt template for Amazon Titan and AI21 Jurassic:

{{Task specification}} \n

#### User prompt:

Write text with exclamation points.

Please write a formatted email in the
voice of a friend

デキスト生成 99

Please write a {{Text Category}} in the voice of {{Role}}.\n

congratulating someone on a new job.

#### Output:

Hey [name]!

I just saw the news and I am so, so happy for you!

You deserve this position and this company with all your heart.

I can't wait to see you soar and accomplish all the things you've been working towards. If anyone is deserving of a promotion, it's you!

Congratulations again, friend!

Love,
[name]

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

次の例では、ユーザーがモデルに対し、テキストを生成するときに特定の人物の役割を引き受けるように求めるプロンプトを出しています。応答の中でモデルが担っている役割が署名に反映されていることに注目してください。

# Prompt template for Amazon Titan and AI21 Jurassic:

{{Role assumption}} {{Task description}}\n

#### User input:

My name is Jack. Help me write a note expressing my gratitude to my parents for taking my son (their grandson) to the zoo.

I miss my parents so much.

#### Output:

Dear Mom and Dad,

I wanted to take a moment to express my heartfelt gratitude

デキスト生成 100

for taking my son to the zoo last weekend. It meant so much to me that you took the time to make memories with him, and I am so grateful for the love and support you have shown him.

I miss you both so much, and I wish we could be together more often.

But I know that you will always be there for me and my family, and I am so grateful to have you as my parents.

Love, Jack

(プロンプトの出典: AWS、使用モデル: AI21 Jurassic-2 Ultra v1)

## コードの生成

このプロンプトには、ユーザーがモデルによって生成されることを期待するコードに関するタスクまたは関数とプログラミング言語を記述します。

# Prompt template for Amazon Titan and AI21 Jurassic:

Write a function in {{programming language}} to {{task or purpose}}\n.

#### User input:

Write a python function to download a s3 file to local disk.

#### Output:

The following code can be used to download an s3 file to the local disk.

```Python
import boto3
# Creates a boto3 client object
s3 = boto3.client('s3')
def download\_file(bucket\_name,
 object\_key, file\_path):
try:

コードの生成 101

# Downloads the s3 file to the
 specified file path
s3.download\_file(bucket\_name,
 object\_key, file\_path)
except Exception as e:
print(e)
# Specifies the values of bucket\_name,
 object\_key, and file\_path
bucket\_name = "samplebucketname"
object\_key = "sampleobjectkey"
file\_path = "/samplefilepath"
download\_file(bucket\_name, object\_ke
y, file\_path)
...

The given code uses the download\_file function to download an s3 file to the local disk. The download file function takes the bucket name, object key, and file path as input parameters. The bucket name is the name of the s3 bucket where the file is located, the object key is the unique identifie r of the file within the bucket, and the file path is the destination path on the local disk where the file will be downloaded. The function uses the download file method of the s3 client to download the file from the specified bucket and write it to the specified file path.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

Prompt template for Claude
\n
\n

User prompt:

コードの生成 102

Human: Write a function in {{program
ming language}} to
{{task or purpose}} \n
\n
Assistant:

Human: Write some python code that quickly approximates the square root of a number.

Assistant:

```
Output:
Here is a simple function to approxima
the square root of a number using
Newton's method:
```python
def sqrt(n, iterations=10):
x = n
for i in range(iterations):
x = (x + n/x) / 2
return x
To use it:
```python
print(sqrt(25))
# 5.0
print(sqrt(15, iterations=20))
# 3.872983346207417
This repeatedly takes the average of
x and n/x to converge on the square
root.
More iterations produce a more
accurate approximation.
The advantage of this approach is it
converges very quickly,
so even just 10 iterations produces a
 decent estimate.
```

コードの生成 103

#### (プロンプトの出典: AWS、使用モデル: Claude)

# 数学

入力には、数値、論理、幾何学など、あるレベルでの数学的推論を必要とする問題を記述します。このような問題では、「Let's think step by step」や「Think step by step to come up with the right answer.」などのフレーズを指示に追加して、問題に断片的に取り組むようモデルに要求するのが有効です。

# Prompt template for Amazon Titan and AI21 Jurassic:

{{Math problem description}}\n Let's think step by step. \n

#### User prompt:

A triangle has two angles of 70 degrees and 50 degrees.

What is the third angle in degrees? Let's think step by step.

#### Output:

The sum of the three angles in a triangle is always 180 degrees.

The two given angles are 70 degrees and 50 degrees. 70 + 50 = 120.

Therefore, the third angle is 180 - 120 = 60 degrees.

The third angle is 60 degrees.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

また、次の例でも、モデルに1ステップずつに考えてもらう方法を示しています。

# Prompt template for Amazon Titan and AI21 Jurassic:

{{Math problem description}}\n
Think step by step to come up with the right answer. \n

#### User prompt:

Lucy has 12 colorful marbles, and she wants to share them equally with her 4 friends. How many marbles will each friend receive? Think step-by-step to come up with the right answer.

#### Output:

 Lucy has 12 marbles and wants to share them equally with her 4 friends, then each friend will receive 12 / 4 = 3 marbles.
 Therefore, each friend will receive 3 marbles.

(プロンプトの出典: AWS、使用モデル: AI21 Jurassic-2 Ultra v1)

## 推論/ロジカルシンキング

複雑な推論タスクやロジカルシンキングを必要とする問題については、モデルに論理的な推理をさせて、その回答を説明するように要求することができます。

# Prompt template for Amazon Titan and AI21 Jurassic:

Question: {{question}} \n

Please output the answer and then explain your

answer:\n

#### User prompt:

Question: Which word is the odd one out?

- A. accomplished
- B. good
- C. horrible
- D. outstanding

Please output the answer and then explain your answer:

#### Output:

The odd one out is C. horrible. The other three words describe positive qualities or achievements, whereas horrible is an negative adjective.

(プロンプトの出典: AWS、使用モデル: Amazon Titan Text G1 - Express)

次に、Claude モデルを使用した別の例を示します。

Prompt template for Claude:

User prompt

**推論/ロジカルシンキング** 105

\n Human: {{question}} \n Please provide the answer and show the

\n \n

Assistant:

reasoning.

Human: The barber is the "one who shaves all those, and those only, who do not shave themselves". Does the barber shave himself? Why is this a paradox? Please provide the answer and show the reasoning.

#### Assistant:

#### Output:

This is an example of a paradox known as the Barber paradox.

Here is an explanation:

- The statement defines a barber who shaves all and only those people who do not shave themselves.
- So if the barber shaves himself, he must shave all those who do not shave themselves, which includes himself.
  But the barber cannot shave himself as per the initial statement.
- If the barber does not shave himself, then he would fit into the group of people who do not shave themselves, and so he must shave himself, according to the statement. But again this contradicts the initial statement.
- So the barber cannot shave himself and also cannot not shave himself. This is self-contradictory based on the initial statement about the barber.
- Therefore, the statement leads to a paradox where both possibilities (the barber shaving himself or not shaving himself) result in a contradiction.

In summary, the paradox arises because the definition of the barber's

推論/ロジカルシンキング 106

behavior is self-contradictory when applied to the barber himself.

This makes it impossible to determine if the barber shaves himself or not based on the given statement alone.

(プロンプトの出典: https://en.wikipedia.org/wiki/Barber\_paradox、使用モデル: Claude)

推論/ロジカルシンキング 107

# 埋め込み

テキスト埋め込みは、ドキュメント、段落、文などの非構造化テキストの意味があるベクトル表現を表します。テキストの本文を入力すると、出力は (1 x n) のベクトルになります。埋め込みベクトルは、さまざまなアプリケーションで使用できます。

Bedrock は、Titan Embeddings G1 - Text モデル (amazon.titan-embed-text-v1) というテキスト埋め込みモデルをサポートしています。このモデルは、テキストの取得検索、意味類似度、クラスタリングをサポートします。入力テキストの最大数は 8K トークンで、出力ベクトルの最大長は1.536 です。

テキスト埋め込みモデルを使用するには、 <u>InvokeModel</u> API オペレーションを、modelId に amazon.titan-embed-text-v1 を指定して使用し、レスポンス内に embedding オブジェクトを取得します。

Jupyter Notebook の例を参照するには:

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/home) を開きます。
- 2. 左側のメニューから [ベースモデル] を選択します。
- 3. 下にスクロールして [Titan Embeddings G1 Text] を選択します。
- 4. [Titan Embeddings G1 Text] タブで、[ノートブックの例を表示] を選択すると、埋め込み用の ノートブックの例が表示されます。

# カスタムモデル

#### Note

微調整を行うには、Amazon Titan Text G1-Express モデルにアクセスできる必要があります。Amazon Titan Text G1 - Express モデルは限定プレビューです。ご利用をリクエストするには、AWS アカウントマネージャーに連絡してください。

Amazon Bedrock モデルをカスタマイズしてパフォーマンスを向上させ、カスタマーエクスペリエンスを向上させることができます。Amazon Bedrock では、ラベル付けされた、より小さな独自のトレーニングデータセットを提供・使用することで、トレーニング時間を短縮するとともに正確性も維持、向上できるという、モデルの微調整が現在可能になっています。

モデルカスタマイズジョブを完了したら、カスタマイズモデルのプロビジョンドスループット  $(\lceil \frac{\mathcal{C}^{\square} \cup \mathcal{C}^{\square} \cup \mathcal{C$ 

モデルカスタマイズのクォータについては、「<u>モデルカスタマイズのクォータ</u>」を参照してください。

#### トピック

- 微調整
- データセットを準備する
- コンソールの使用
- API を使用する
- モデルカスタマイズに関するガイドライン
- トラブルシューティング

# 微調整

Amazon Bedrock モデルを微調整するには、ラベル付けされた独自のトレーニングデータセットを提供します。これにより、特定のタスクに対するモデルのパフォーマンスが向上します。特定のタスクに関連するラベル付きのサンプルをいくつか提供することで、モデルが実行対象のタスクを学習しや

微調整 109

すくなります。数ショット微調整と呼ばれるこのプロセスを通じて、特定のタスクについて元のモデルのパフォーマンスと効率を改良した新しいモデルを作成できます。

モデルを微調整するには、Bedrock 微調整ジョブの Amazon S3 バケットパスを指定して、トレーニングデータセットや検証データセット (必要な場合) を Amazon S3 にアップロードします。また、微調整は、ハイパーパラメータを調整することでも行うことができます。コンソールを使用してモデルを微調整する手順については、「<u>コンソールの使用</u>」を参照してください。API を使用して微調整する手順については、「API を使用する」を参照してください。

# データセットを準備する

トレーニングデータと検証データを Amazon S3 にアップロードする前に、データの形式を JSONL 形式のトレーニングおよび検証データセットに前処理しておく必要があります。

検証データセット (オプション) を指定すると、Amazon Bedrock はモデルカスタマイズジョブの終了時に検証ロスメトリクスを返します。

#### Note

以下の形式は Titan Text G1 - Express モデルの微調整に関するものです。

トレーニングデータセットとオプションの検証データセットの両方に対して、複数の JSON 行を含むファイルを作成します。各 JSON 行には、input および output フィールドが両方とも含まれます。形式は次のとおりです。クォータについては、「<u>Amazon Bedrock のクォータ</u>」を参照してください。トークン数を概算するには、1 トークンを 6 文字として計算します。

```
{"input": "<prompt text>", "output": "<expected generated text>"}
{"input": "<prompt text>", "output": "<expected generated text>"}
{"input": "<prompt text>", "output": "<expected generated text>"}
```

次に、質問応答タスクの項目の例を示します。

```
{"input": "what is AWS", "output": "it's amazon web services"}
```

トレーニングデータセットと、オプションの検証データセットを JSONL 形式で作成したら、以下の手順を実行します。

1.  $\vec{r}$   $\vec{r}$ 

データセットを準備する 110

2. <u>カスタムジョブにトレーニングデータへのアクセスを付与する</u> に示す例と類似した IAM ポリシーを Amazon Bedrock サービスロールに付与することで、データにアクセスするためのアクセス許可を Amazon Bedrock に付与します。

3. コンソールまたは API を使用して微調整ジョブを送信します。

# コンソールの使用

以下のトピックでは、モデルの微調整に必要な、大きな手順について説明します。

#### トピック

- ジョブを送信する
- ジョブのモニタリング
- ジョブを停止する
- ジョブ結果を分析する
- 微調整されたモデルを推論に使用する

# ジョブを送信する

Note

これら大きな手順を開始する前に、以下を実行します。

- トレーニングデータと検証データにアクセスするとともに出力データを書き込むためのアクセス許可を Amazon Bedrock に付与します。これを行うには、「カスタムジョブにトレーニングデータへのアクセスを付与する」に示されている例と同じ IAM ポリシーをAmazon Bedrock サービスロールに付与します。
- トレーニングデータセット (および、必要な場合は検証データセットも) を作成します。詳細については、「データセットを準備する」を参照してください。

#### 微調整ジョブを送信するには

- AWS Management Consoleにサインインし、Amazon Bedrock コンソール (<a href="https://console.aws.amazon.com/bedrock/home">https://console.aws.amazon.com/bedrock/home</a>) を開きます。
- 2. 左側のメニューから、[カスタムモデル] を選択します。

コンソールの使用 111

- 3. [モデルを微調整]を選択します。
- 4. 微調整ジョブの設定を行います。

a. [モデルの詳細] セクションで、独自のデータを使用して微調整したいソースモデルを選択し、[微調整されたモデル名] で、微調整された結果のモデルに名前を付けます。

#### Note

現時点では、微調整できるモデルは Titan Text G1-Express だけです。

b. (オプション) [モデル暗号化] を選択すれば、デフォルトキーとは異なる KMS キーを選択することもできます。

KMS キーの設定については、「データ暗号化」を参照してください。

- c. (オプション) [タグ] セクションを展開し、[新しいタグを追加] を選択して、モデルに関連付けるタグを追加することもできます。
- d. [ジョブの設定] セクションで、[ジョブ名] に微調整トレーニングジョブの名前を入力し、 ジョブに関連付けるタグを追加します。
- e. (オプション) [VPC の設定] セクションで、入力データと出力データを格納している Amazon S3 の場所がある VPC を選択します。
  - i. [サブネット] で、VPC サブネットを追加します。
  - ii. [セキュリティグループ] で、VPC のデータへのアクセスを制御するためのセキュリティグループを追加します。

VPC の設定の詳細については、「<u>VPC を使用してジョブを保護する</u>」を参照してください。

- f. [入力データ] セクションで、トレーニングデータセットファイルへのリンク (および、必要な場合は Amazon S3 バケット内の検証データセットファイルへのリンク) を入力します。
- g. [ハイパーパラメータ] セクションで、トレーニングで使用する以下のハイパーパラメータの値を入力します。値の選択に関するガイドラインについては、「<u>モデルカスタマイズに関するガイドライン</u>」を参照してください。
  - ・ エポック トレーニングデータセットをモデルに渡す回数。
  - バッチサイズ 各バッチでモデルに送信されるレコードの数。
  - 学習率 各イテレーションでのパラメータ更新のステップサイズ。

-ジョブを送信する 112

• 学習率のウォームアップステップ - アルゴリズムが最適な重みに収束する速度に影響します。

- h. [出力データ] セクションで、微調整ジョブの出力を保存する Amazon S3 フォルダーへのリンクを指定します。各エポックのトレーニング損失メトリクスと検証損失メトリクスは、指定した場所にある別々のファイルに保存されます。
- i. サービスロールを使用すると、Amazon Bedrock に対し、自動的に Amazon S3 に書き込む ためのアクセス許可を付与できます。

[サービスアクセス] セクションで、次のいずれかの操作を行います。

- 既存のサービスロールを使用 ドロップダウンリストからサービスロールを選択します。
- 新しいサービスロールを作成して使用 サービスロールの名前を入力します。

#### Note

ジョブで VPC 設定を使用している場合、コンソールはそのジョブの新しいサービスロールを作成できません。「Configure your model customization job to use VPC」で説明されている例を使用して、サービスロールを作成してください。

5. [モデルを微調整] を選択してジョブを開始します。

# ジョブのモニタリング

微調整ジョブの実行には、数時間かかる場合があります。ジョブの所要時間は、トレーニングデータのサイズ (レコード、入力トークン、出力トークンの数)、エポック数、バッチサイズによって異なります。

微調整ジョブのステータスをモニタリングするには

- 1. Amazon Bedrock コンソールを開きます。
- 2. 左側のメニューから、[カスタムモデル] を選択します。
- 3. [トレーニングジョブ] タブに、開始した微調整ジョブが表示されます。[ステータス] 列を見て、 ジョブの進行状況をモニタリングします。
- 4. ジョブを選択すると、トレーニングに入力した詳細が表示されます。

. ジョブのモニタリング 113

## ジョブを停止する

Bedrock の微調整ジョブは進行中でも停止できます。ただし、停止したジョブを再開することはできません。

微調整ジョブを停止中するには

- 1. Amazon Bedrock コンソールを開きます。
- 2. 左側のメニューから、[カスタムモデル] を選択します。
- 3. [トレーニングジョブ] タブで、停止するジョブの横にあるラジオボタンを選択します。
- 4. [ジョブを停止] ボタンを選択します。
- 5. トレーニングジョブを停止すると再開できないことを警告するモーダルが表示されます。[ジョブを停止] を選択して確定します。



Amazon Bedrock は、ユーザーがジョブを停止する前に、Amazon Bedrock がモデルのトレーニングに使用していたトークンの料金を請求します。Amazon Bedrock は、停止済みのジョブの中間カスタムモデルを作成しません。

# ジョブ結果を分析する

微調整ジョブが完了すると、S3の出力フォルダーには以下の情報が表示されます。

- 完了ステータス
- トレーニングおよび検証ロスメトリクス

Amazon Bedrock は、お客様のアカウントにスコープ設定された AWS マネージドストレージにカスタマイズモデルを保存します。カスタマイズモデルは、モデルカスタマイズダッシュボードページの[モデル] テーブルに表示されます。モデルを選択すると、そのカスタムモデルに関連する詳細が表示されます。出力モデルはダウンロードできません。

微調整ジョブの S3 出力により、次の出力ファイルが S3 フォルダーに格納されます。

- model-customization-job-training\_job\_id/
  - training\_artifacts/

ジョブを停止する 11<sup>4</sup>

- step\_wise\_training\_metrics.csv
- validation\_artifacts/
  - post\_fine\_tuning\_validation/
    - validation\_metrics.csv

step\_wise\_training\_metrics.csv および validation\_metrics.csv ファイルを使用すると、モデルカスタマイズジョブを分析できるだけでなく、必要に応じてモデルを調整することもできます。

step\_wise\_training\_metrics.csv ファイルの構造を次の例で示します。

step_number	epoch_number	training_loss	perplexity
1	1	0.2	25
2	1	0.18	22
			18

validation\_metrics.csv ファイルの構造を次の例で示します。

step_number	epoch_number	validation_loss	perplexity
1	1	0.12	20
2	1	0.09	17
			15

# 微調整されたモデルを推論に使用する

カスタマイズモデルを推論に使用する場合は、まず、そのモデルのプロビジョンドスループットを購入しておく必要があります。これにより、カスタマイズモデルをテキストまたはチャットのプレイグラウンドで推論に使用できるようになります。

カスタムモデルのプロビジョンドスループットを購入するには

1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。

- 2. 左側のナビゲーションペインで [カスタムモデル] を選択します。
- 3. モデルの横にあるラジオボタンをクリックし、[アクション] を選択して、[プロビジョンドスループットを購入] を選択します。詳細については、「<u>プロビジョンドスループット</u>」を参照してください。

#### 微調整されたモデルを推論に使用するには

- 1. 左側のナビゲーションペインで [カスタムモデル] を選択します。
- 2. [モデル] タブで、テキストまたはチャットのプレイグラウンドで使用したいモデルを選択し、[プレイグラウンドで開く] を選択します。

# API を使用する

このセクションでは、API オペレーションを使用してモデルを微調整する方法を示します。例としては、AWS Command Line Interface と AWS SDK for Python (Boto3) を使用したものを示します。

#### トピック

- モデルカスタマイズの IAM ロールを設定する
- ジョブを送信する
- ジョブを監視する
- ジョブを停止する
- ジョブを分析する
- カスタマイズモデルに関する情報を取得する

# モデルカスタマイズの IAM ロールを設定する

Amazon S3 バケット内のデータにアクセスするのに必要なアクセス許可をカスタマイズジョブに付与する IAM ロールを新規作成するか、そのような既存のロールを使用します。IAM <u>CreateRole</u> API オペレーションを使用することで、Amazon Bedrock がロールを引き受けることを許可します。また、IAM <u>CreatePolicy</u> API オペレーションを使用して、ロールがトレーニング、検証、および出力のデータを含む Amazon S3 バケットにアクセスすることを許可するポリシーを作成します。ロールにポリシーを付与するには、<u>AttachRolePolicy</u> API オペレーションを使用します。適切な IAM アクセス許可が付与されている既存のロールを使用する場合は、このステップをスキップしてください。コンソールを使用してジョブを送信する (「<u>ジョブを送信する</u>」を参照) と、適切なアクセス許可が付与されたロールが自動的に作成されます。

API を使用する 116

#### ロールとアクセスポリシーの準備:

 Amazon Bedrock がロールを引き受けることができるように、下記の BedrockAssumeRolePolicy.json (または任意の別の名前) という名前のポリシードキュメントを作成します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "bedrock.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                     "aws:SourceAccount": "111122223333"
                },
                "ArnEquals": {
                     "aws:SourceArn": "arn:aws:bedrock:us-east-1:111122223333:model-
customization-job/*"
            }
        }
    ]
}
```

- 2. トレーニングデータと検証データをアップロードした Amazon S3 の場所と、Amazon Bedrock がトレーニングメトリクスデータをアップロードする場所を決定します。
- 3. これらの Amazon S3 の場所へのアクセスを許可する、下記の BedrockAccessTrainingValidations3policy.json (または任意の別の名前) という名前 のポリシードキュメントを作成し、Resource リスト内の値を実際の Amazon S3 ARN に置き 換えます。

4. これらの Amazon S3 の場所へのアクセスを許可する、下記の *BedrockAccessOutputS3Policy.json* という名前のポリシーを作成し、Resource リスト 内の値を実際の Amazon S3 ARN に置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "s3:ListObjects"
            ],
            "Resource": [
                 "arn:aws:s3:::my_output_bucket/myfolder",
                 "arn:aws:s3:::my_output_bucket/myfolder/*"
            ]
        }
    ]
}
```

#### **AWS CLI**

1. IAM ロールを作成し、role-name を任意の名前に置き換え、作成した引き受けロールポリシーのファイル名を使用します

```
aws iam create-role \
    --role-name "MyBedrockModelCustomizationRole" \
    --assume-role-policy-document file://path/to/BedrockAssumeRolePolicy.json
```

2. トレーニングと検証のデータのアクセスポリシーを作成し、policy-name を任意の名前に 置き換えます。

```
aws iam create-policy \
    --policy-name "MyBedrockTrainingValidationS3BucketPolicy" \
    --policy-document file://path/to/
BedrockAccessTrainingValidationS3Policy.json
```

3. 出力のデータのアクセスポリシーを作成し、policy-name を任意の名前に置き換えます。

```
aws iam create-policy \
    --policy-name "MyBedrockOutputS3BucketPolicy" \
    --policy-document file://path/to/BedrockAccessOutputS3Policy.json
```

4. IAM ロールにトレーニングとデータのアクセスポリシーを付与して、role-name をステップ 1 で作成したロール名に置き換えるとともに、policy-arn をステップ 2 のレスポンスから返された ARN に置き換えます。

```
aws iam attach-role-policy \
    --role-name "MyBedrockModelCustomizationRole" \
    --policy-arn "PolicyARNFromCreateTrainingValidationPolicyStep"
```

5. IAM ロールに出力のデータのアクセスポリシーを付与し、role-name をステップ 1 で作成したロール名に置き換えるとともに、policy-arn をステップ 3 のレスポンスから返された ARN に置き換えます。

```
aws iam attach-role-policy \
    --role-name "MyBedrockModelCustomizationRole" \
    --policy-arn "PolicyARNFromCreateOutputPolicyStep"
```

6. モデルカスタマイズジョブを送信する際に、ステップ 1 で返された ARN を roleArn として使用します。

#### Python

1. IAM ロールを作成します。それには、「<u>IAM examples using SDK for Python (Boto3)</u>」の「Create a role」セクションのコードサンプルに従って、role\_name を任意の名前に置き換えるとともに、trust\_policy を作成した *BedrockAssumeRolePolicy.json* の内容に置き換えます。

- 2. 2つのアクセスポリシーを作成します。それには、「<u>IAM examples using SDK</u> for Python (Boto3)」の「Create a policy」セクションのコードサンプルに従って、PolicyName を 2 つの異なる任意の名前に置き換えるとともに、policy\_docを作成した BedrockAccessTrainingValidationS3Policy.json と BedrockAccessOutputS3Policy.json の内容に置き換えます。
- 3. IAM ロールにポリシーを付与します。それには、「<u>IAM examples using SDK for Python</u> (<u>Boto3</u>)」の「Attach a policy to a role」セクションのコードサンプルに従って、role\_name をステップ 1 で作成したロール名に置き換えるとともに、policy\_arn をステップ 2 のレスポンスから返された ARN に置き換えます。
- 4. モデルカスタマイズジョブを送信する際に、ステップ 1 で返された ARN を roleArn として使用します。

# ジョブを送信する

モデルカスタマイズジョブを送信するには、Amazon Bedrock <u>CreateModelCustomizationJob</u> API オペレーションを使用します。<u>CreateModelCustomizationJob</u> リクエストには、少なくとも以下のフィールドを指定する必要があります。

- customizationType モデルを微調整するには、FINE\_TUNINGの値を使用します。
- baseModelIdentifier カスタマイズするモデルの ARN。
- customModelName 新しくカスタマイズしたモデルに付ける名前。
- hyperParameters モデルの調整 (チューニング) に関連するパラメータ。Titan Text G1 Express モデルの場合、このオブジェクト内の以下のフィールドを指定できます。値の選択に関するガイドラインについては、「モデルカスタマイズに関するガイドライン」を参照してください。
  - エポック トレーニングデータセットをモデルに渡す回数。
  - バッチサイズ 各バッチでモデルに送信されるレコードの数。
  - 学習率 各イテレーションでのパラメータ更新のステップサイズ。
  - 学習率のウォームアップステップ アルゴリズムが最適な重みに収束する速度に影響します。

• jobName - トレーニングジョブに付ける名前。

- roleArn サービスロールの ARN。
- trainingDataConfig トレーニングデータセットを保存する Amazon S3 の場所を指す URI を 格納するオブジェクト。トレーニングデータセットを作成する方法については、「データセットを 準備する」を参照してください。
- validationDataConfig 検証データセットを保存する Amazon S3 の場所を指す URI を格納するオブジェクト。検証データセットを作成する方法については、「データセットを準備する」を参照してください。
- outputDataConfig 出力データを書き込む Amazon S3 の場所を指す URI を格納するオブジェクト。

レスポンスから返される jobArn を使用することで、モデルのカスタマイズジョブを監視または停止できます。

#### **AWS CLI**

次の例では、AWS CLI を使用してモデルカスタマイズジョブを送信する方法を示します。

まず、*FineTuningData.json* という名前のテキストファイルを作成します。下の JSON コードをテキストファイルにコピーし、bucket、path、各ファイル名をトレーニング、検証、出力のデータへの正しいパスに置き換えます。

```
{
    "trainingDataConfig": {
        "s3Uri": "s3://bucket/path/to/train.jsonl"
},
    "validationDataConfig": {
        "validators": [{
            "s3Uri": "s3://bucket/path/to/validation.jsonl"
        }]
    },
    "outputDataConfig": {
        "s3Uri": "s3://bucket/path/to/output-folder"
}
```

コマンドラインから、以下のコマンドを実行します。

ジョブを送信する 121



現時点では、微調整できるモデルは Titan Text G1-Express だけです。

```
aws bedrock create-model-customization-job \
    --customization-type "FINE_TUNING" \
    --base-model-identifier "arn:aws:bedrock:us-east-1::foundation-model/foundation-model-id" \
    --role-arn "arn:aws:iam::arn-for-MyBedrockModelCustomizationRole" \
    --job-name "job-name" \
    --custom-model-name "custom-model-name" \
    --custom-model-name "custom-model-name" \
    --hyper-parameters
epochCount="1",batchSize="1",learningRate="0.005",learningRateWarmupSteps="0" \
    --cli-input-json file://path/to/FineTuningData.json
```

VPC 設定を追加するには、上記のコマンドに、セキュリティグループとサブネットを指定する以下の引数を追加します。

```
--vpc-config '{securityGroupIds": ["xx"], "subnetIds": ["subnet-yy", "subnet-zz"]}'
```

KMS キーでモデルを暗号化するには、上記のコマンドに次の引数を追加し、値を置き換えてモデルの暗号化に使用するキーを指定します。

```
--customModelKmsKeyId 'arn:aws:kms:region:account-id:key/key-id'
```

タグを追加するには、上記のコマンドに次の引数を追加し、キーと値をジョブや出力モデルに付けたいタグに置き換え、キーと値のペアの間を必ずスペースで区切ります。

```
--tags key=key1,value=value1 key=key2,value=value2
```

ジョブを送信する 122

このオペレーションによって返される jobArn を使用することで、ジョブのステータスを確認したり、ジョブを分析または停止したりできます。

#### Python

次の例では、Python を使用してモデルカスタマイズジョブを送信する方法を示します。関連するセクションのコメントを解除するには、オプションのタグをジョブや結果のモデルに付けます。

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock')
# Set parameters
customizationType = "FINE_TUNING"
baseModelIdentifier = "arn:aws:bedrock:us-east-1::foundation-model/foundation-model-
id"
roleArn = "arn:aws:iam::arn-for-MyBedrockModelCustomizationRole"
jobName = "job-name"
customModelName = "custom-model-name"
hyperParameters = {
        "epochCount": "1",
        "batchSize": "1",
        "learningRate": "0.005",
        "learningRateWarmupSteps": "0"
trainingDataConfig = {"s3Uri": "s3://bucket/path/to/train.jsonl"}
validationDataConfig = {
        "validators": [{
            "name": "validation",
            "s3Uri": "s3://bucket/path/to/validation.jsonl"
        }]
outputDataConfig = {"s3Uri": "s3://bucket/path/to/"}
# # Uncomment to add optional tags
# jobTags = [
#
      {
          "key": "key1",
#
          "value": "value1"
#
      }
# ]
# customModelTags = [
#
          "key": "key1",
```

-ジョブを送信する 123

```
"value": "value1"
      }
# ]
# Create job
bedrock.create_model_customization_job(
    jobName=jobName,
    customModelName=customModelName,
    roleArn=roleArn,
    baseModelIdentifier=baseModelIdentifier,
    hyperParameters=hyperParameters,
         # Uncomment to add optional tags
         jobTags=jobTags,
         customModelTags=customModelTags,
    trainingDataConfig=trainingDataConfig,
    validationDataConfig=validationDataConfig,
    outputDataConfig=outputDataConfig
)
```

このコードによって返される jobArn を使用することで、ジョブのステータスを確認したり、 ジョブを分析または停止したりできます。

## ジョブを監視する

すべてのモデルカスタマイズジョブをリスト表示するには、Amazon Bedrock <u>ListModelCustomizationJobs</u> リクエストを送信します。モデルカスタマイズジョブのステータスを監視するには、「jobArn」と入力することによって Amazon Bedrock <u>GetModelCustomizationJob</u> リクエストを送信します。jobArn は、次のいずれかの方法で見つけることができます。

- 1. Amazon Bedrock コンソールで [微調整] を選択し、[トレーニングジョブ] テーブルからジョブを 選択します。[モデルの設定] セクションで [モデルカスタマイズジョブ ARN] を探します。
- 2. ジョブを作成した CreateModelCustomizationJob 呼び出しから返されたレスポンスに含まれている jobArn フィールドの値を調べます。

レスポンスの status フィールドの値を調べることで、ジョブの進行状況を監視します。

AWS CLI

AWS CLI で次のコマンドを使用して、モデルカスタマイズジョブをリスト表示します。

ジョブを監視する 124

```
aws bedrock list-model-customization-jobs
```

次の例では、AWS CLI を使用してモデルカスタマイズジョブを監視する方法を示します。

```
aws bedrock get-model-customization-job \
    --job-identifier "arn:aws:bedrock:job-arn-from-create-model-customization"
```

レスポンスの status フィールドの値を確認します。

#### Python

次の例では、Python を使用してモデルカスタマイズジョブをリスト表示する方法を示します。

```
import boto3
bedrock = boto3.client(service_name='bedrock')
bedrock.list_model_customization_jobs()
```

次の例では、モデルカスタマイズジョブを監視する方法を示します。

```
import boto3
bedrock = boto3.client(service_name='bedrock')

fine_tune_job =
  bedrock.get_model_customization_job(jobIdentifier='arn:aws:bedrock:job-arn-from-create-model-customization')

print(fine_tune_job['status'])
```

## ジョブを停止する

API を使用してモデルカスタマイズジョブを停止するには、以下の手順に従います。

- 1. <u>GetModelCustomizationJob</u> レスポンス内のジョブの status が IN\_PROGRESS の場合、<u>StopModelCustomizationJob</u> リクエストをトレーニングジョブの jobArn を指定して送信します。システムはそのジョブに終了のマークを付け、状態を STOPPING に設定します。
- 2. システムはジョブを停止し、状態を STOPPED に設定します。

ジョブがシステムによって停止される前に完了した場合、システムは状態を COMPLETED に設定します。

-ジョブを停止する 125



#### Note

Amazon Bedrock は、ユーザーがジョブを停止する前に、Amazon Bedrock がモデルの トレーニングに使用していたトークンの料金を請求します。Amazon Bedrock は、停止 済みのジョブの中間カスタムモデルを作成しません。

#### **AWS CLI**

次の例では、AWS CLI を使用してモデルカスタマイズジョブを停止する方法を示します。

```
aws bedrock stop-model-customization-job \
    --job-identifier "arn:aws:bedrock:job-arn-from-create-model-customization"
```

#### Python

次の例では、Python を使用してモデルカスタマイズジョブを停止する方法を示します。

```
import boto3
bedrock = boto3.client(service_name='bedrock')
bedrock.stop_model_customization_job(jobIdentifier='arn:aws:bedrock:job-arn-from-
create-model-customization')
```

# ジョブを分析する

モデルカスタマイズジョブのステータスを分析するには、「jobArn」と入力することによって Amazon Bedrock GetCustomModelb リクエストを送信します。jobArn は、次のいずれかの方法で見 つけることができます。

- 1. Amazon Bedrock コンソールで [微調整] を選択し、[トレーニングジョブ] テーブルからジョブを 選択します。[モデルの設定] セクションで [モデルカスタマイズジョブ ARN] を探します。
- 2. ジョブを作成した CreateModelCustomizationJob 呼び出しから返されたレスポンスに含まれ ている jobArn フィールドの値を調べます。

ジョブを分析する 126

trainingMetrics オブジェクト内の trainingLoss の値と validationMetrics オブジェクト 内の validatorLoss の値を使用してモデルカスタマイズジョブを分析するか、S3 出力ファイルで データを読み取ることによって、メトリクスを分析できます。

#### **AWS CLI**

次の例では、AWS CLI を使用してモデルカスタマイズジョブを分析する方法を示します。

```
aws bedrock get-model-customization-job \
    --job-identifier "arn:aws:bedrock:job-arn-from-create-model-customization"
```

trainingMetrics および validatorLoss フィールドの値を調べて、モデルカスタマイズ ジョブを分析します。

#### Python

次の例では、Python を使用してモデルカスタマイズジョブを分析する方法を示します。

```
import boto3
import json
import pandas as pd

bedrock = boto3.client(service_name='bedrock')

fine_tune_job =
  bedrock.get_model_customization_job(jobIdentifier='arn:aws:bedrock:job-arn-from-create-model-customization')

# Get S3 URI
fine_tune_job['outputDataConfig']['s3Uri']
```

S3 URI を使用して、出力データをダウンロードすることができます。詳細については、「<u>オブ</u> <u>ジェクトのダウンロード</u>」を参照してください。フォルダーを開き、メトリクスファイルに移動 します。

# カスタマイズモデルに関する情報を取得する

モデルのカスタマイズでは、カスタマイズジョブが正常に完了するとカスタムモデルが作成されます。カスタムモデルをリスト表示するには、Amazon Bedrock ListCustomModels リクエストを送信

します。微調整した特定のモデルに関する情報を取得するには、Amazon Bedrock <u>GetCustomModel</u> リクエストを、modelIdentifier としてモデルの ARN を指定して送信します。モデルの ARN は、次のいずれかの方法で見つけることができます。

- 1. Amazon Bedrock コンソールで [微調整] を選択し、[モデル] テーブルからモデルを選択します。 「モデルの詳細」セクションで [微調整されたモデル ARN] を探します。
- 2. ジョブを作成した GetModelCustomizationJob 呼び出しから返されたレスポンスに含まれている outputModelArn フィールドの値を調べます。このフィールドは、ジョブが終了した後にのみレスポンスに表示されます。

#### **AWS CLI**

微調整されたモデルを AWS CLI でリスト表示するには、次のコマンドを使用します。

```
aws bedrock list-custom-models
```

次の例は、微調整されたモデルに関する情報を AWS CLI で取得する方法を示しています。

```
aws bedrock get-custom-model \
    --model-identifier "arn:aws:bedrock:customized-model-arn"
```

#### Python

微調整されたモデルを Python でリスト表示するには、次のコマンドを使用します。

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock')
bedrock.list_custom_models()
```

次の例は、微調整されたモデルに関する情報を AWS CLI で取得する方法を示しています。

```
import boto3
import json
bedrock = boto3.client(service_name='bedrock')
bedrock.get_custom_model(modelIdentifier='arn:aws:bedrock:customized-model-arn')
```

# モデルカスタマイズに関するガイドライン

このセクションでは、Titan Text G1 - Express モデルのカスタマイズの基準となるガイドラインと推 奨値を示します。ただし、値をいろいろ試して、どのパラメータがお客様自身のケースで最も適切に 機能するかを確認する必要があります。理想的なパラメータは、データセットと、モデルが対象とす るタスクによって異なります。

微調整ジョブの送信時に生成される出力ファイルに含まれるトレーニングと検証のメトリクスを使用して、パラメータを調整します。この出力ファイルは、出力を書き込んだ Amazon S3 バケット内で見つけるか、GetCustomModel オペレーションを使用してください。

# 入力トレーニングデータセットのサイズ

一般に、トレーニングデータセットが大きいほど、当該タスクのパフォーマンスが向上します。ただし、サンプルが多すぎると、別のタスクでモデルのパフォーマンスが低下する可能性があります (例えば、要約タスク用に 100,000 個のサンプルが含まれているトレーニングデータセットを分類タスクで使用すると、モデルのパフォーマンスが低下する可能性があります)。

トレーニングデータの数が少ない場合 (例: サンプル数が 1~100 個)、大規模言語モデルは小規模な 教師ありモデルよりもパフォーマンスが優れています。

# モデルサイズ

一般に、与えられているトレーニングデータが少ない場合でも、大規模なモデルほど、タスクのパフォーマンスは向上します。

このモデルを分類タスクに使用する場合、特にクラス数が比較的少ないとき (100 サンプル未満)、数ショットでの微調整 (100 サンプル未満) で得られる利益は比較的小さくなる可能性があります。

# エポック

設定するエポック数を決定するには、以下のメトリクスを使用することをお勧めします。

- 1. 検証出力の精度 精度が高くなるエポック数を設定します。
- 2. トレーニングおよび検証ロス トレーニングおよび検証ロスが安定するまでのエポック数を 設定します。これはモデルが収束するタイミングに対応しています。トレーニングロス値 は、step\_wise\_training\_metrics.csv および validation\_metrics.csv ファイル内で 確認してください。

### バッチサイズ

バッチサイズを変更する場合は、次の式を使用して学習率を変更することをお勧めします。

newLearningRate = oldLearningRate x newBatchSize / oldBatchSize

## 学習率

一般に、大規模モデルには低い学習率を使用してください。1.00E-06 から 1.00E-05 の範囲に含まれる学習率を使用することをお勧めします。このパラメータは、質問応答や分類のタスクでは大きな役割を果たせない傾向があります。ただし、要約タスクのパフォーマンスには大きな影響を与えることができます。

以下の表は、数ショットでの微調整に推奨される学習率の値を示しています。

タスク	最低学習率	デフォルトの学習率	最高学習率
要約	1.00E-06	3.00E-06	5.00E-05
分類	5.00E-06	5.00E-05	5.00E-05
質問応答	5.00E-06	5.00E-06	5.00E-05

# 学習率のウォームアップステップ

デフォルト値0を使用することをお勧めします。

# トラブルシューティング

このセクションでは、微調整中に発生する可能性のあるエラーと、エラーに遭遇した場合のチェック項目についてまとめています。

## アクセス許可の問題

Amazon S3 バケットにアクセスするためのアクセス許可で問題が発生した場合は、以下のとおりになっているかどうかをチェックします。

1. Amazon S3 バケットがサーバー側の暗号化に CM-KMS キーを使用している場合に、Amazon Bedrock に渡される IAM ロールに AWS KMS キーに対する kms:Decrypt アクセス許可がある。

バッチサイズ 130

例については、「<u>特定の AWS アカウント で AWS KMS キーにより暗号化および復号することを</u> ユーザーに許可する」を参照してください。

- 2. Amazon S3 バケットが Amazon Bedrock モデルのカスタマイズジョブと同じリージョンにある。
- 3. IAM ロールの信頼ポリシーにサービス SP (bedrock.amazonaws.com) が含まれている。

以下のメッセージは、Amazon S3 バケット内のトレーニングデータまたは検証データにアクセスするためのアクセス許可に問題があることを示しています。

Could not validate GetObject permissions to access Amazon S3 bucket: <a href="mailto:training-data-bucket">training-data-bucket</a> at key <a href="mailto:training-data-bucket">training-data-bucket</a> at the same training-data-bucket at the same training-da

Could not validate GetObject permissions to access Amazon S3 bucket: *validation-data-bucket* at key *validation.jsonl* 

上記のエラーのいずれかが発生した場合は、サービスに渡された IAM ロールに、トレーニングと検証のデータセットを格納した Amazon S3 の URI への s3:Get0bject および s3:ListBucket アクセス許可が付与されているかどうかをチェックします。例については、「<u>ジョブを送信する</u>」を参照してください。

次のメッセージは、Amazon S3 バケットに出力データを書き込むためのアクセス許可に問題があることを示しています。

Amazon S3 perms missing (PutObject): Could not validate PutObject permissions to access S3 bucket: <a href="mailto:bedrock-output-bucket">bedrock-output-bucket</a> at key <a href="mailto:output/.write\_access\_check\_file.tmp">output/.write\_access\_check\_file.tmp</a>

上記のエラーのいずれかが発生した場合は、サービスに渡された IAM ロールに、出力データを格納した Amazon S3 の URI への s3:Put0bject アクセス許可が付与されているかどうかをチェックします。例については、「<u>ジョブを送信する</u>」を参照してください。

# データの問題

次のエラーは、トレーニング、検証、または出力データファイルの問題に関連しています。

無効なファイル形式

Unable to parse Amazon S3 file: fileName.jsonl. Data files must conform to JSONL format.

上記のエラーが発生した場合は、以下のとおりになっているかどうかをチェックします。

データの問題 131

- 1. すべての行が JSON 形式である。
- 2. どの JSON にも input と output の 2 つのキーがあり、どちらのキーも文字列型です。例:

```
{
    "input": "this is my input",
    "output": "this is my output"
}
```

3. 改行や空行がない。

#### Character quota exceeded

```
Input size exceeded in file fileName.jsonl for record starting with...
```

上記のテキストで始まるエラーが発生した場合は、文字数が<u>微調整用のクォータ</u>に記載されている文字数クォータの範囲内に収まるようにします。

#### Token count exceeded

```
Maximum input token count 4097 exceeds limit of 4096
Maximum output token count 4097 exceeds limit of 4096
Max sum of input and output token length 4097 exceeds total limit of 4096
```

上記のエラーが発生した場合は、トークンの数が <u>微調整用のクォータ</u> に記載されているトークンの クォータの範囲内に収まるようにします。

# 内部エラー

```
Encountered an unexpected error when processing the request, please try again
```

上記のエラーが発生した場合は、サービスに問題がある可能性があります。ジョブを再試行してください。問題が解決しない場合は、AWS Support までお問い合わせください。

**内部エラー** 132

# プロビジョンドスループット

モデルにプロビジョンドスループットを設定すると、一定レベルのスループットが固定コストで得られます。

プロビジョンドスループットは、Amazon やサードパーティーのベースモデル、およびカスタマイズ モデルで使用できます。

プロビジョンドスループットの料金は、使用するモデルと選択するコミットメント (契約) のレベルによって異なります。より長い期間にわたって契約すると、割引料金が適用されます。各モデルの料金の詳細については、Amazon Bedrock コンソールの「<u>モデルプロバイダー</u>」ページを参照してください。

モデルのスループットのオプションは、推論をベースモデルで実行するか、カスタムモデルで実行するかによって異なります。

料金オプション	ベースモデル	カスタムモデル
プロビジョンドスループット、コミットメントなし (時間 単位の料金)	利用不可	利用可能 (アカウントあたり 最大 2 つのプロビジョンドス ループット)
プロビジョンドスループッ ト、1 か月間のコミットメン ト	利用可能	利用可能
プロビジョンドスループット、6 か月間のコミットメント	利用可能	利用可能

プロビジョンドスループットはモデルユニット (MU) 数で指定します。モデルユニットは、指定されたモデルに対する特定のスループットレベルを提供します。特定のテキストモデルに対する MU のスループットレベルでは、以下を指定します。

• 1分あたりの入力トークンの総数 - 1分以内にすべてのリクエストに対して MU が処理できる入力 トークンの数。

• 1 分あたりの出力トークンの総数 - 1 分以内にすべてのリクエストに対して MU が生成できる出力 トークンの数。

モデルユニットクォータは、プロビジョンドスループットに指定するコミットメントのレベルによって異なります。

- コミットメントのないカスタムモデルの場合、プロビジョンドスループットあたりで利用できる クォータは1つのモデルユニットとなります。アカウントごとに最大2つのプロビジョンドス ループットを作成できます。
- コミットメントのある基盤モデルまたはカスタムモデルの場合、デフォルトのクォータは0モデルユニットです。モデルユニット数の引き上げをリクエストするには、サービスクォータの引き上げフォームを使用します。

#### トピック

- プロビジョンドスループットモデル ID
- 手順
- アクセス許可
- プロビジョンドスループットのコンソール手順
- プロビジョンドスループット API を使用する場合

# プロビジョンドスループットモデル ID

現在、プロビジョンドスループットは以下のモデルで使用できま

す。<u>CreateProvisionedModelThroughput</u> API を使用する場合は、対応するモデル ID を使用してください。

モデル名	プロビジョンドスループットが使用可能なモデ ル ID
Titan Text G1 - Express 8K	amazon.titan-text-express-v1:0:8k
Titan Embeddings G1 - Text	amazon.titan-embed-text-v1:2:8k
Anthropic Claude V2 18K	anthropic.claude-v2:0:18k

モデル ID 134

モデル名	プロビジョンドスループットが使用可能なモデ ル ID
Anthropic Claude V2 100K	anthropic.claude-v2:0:100k
Anthropic Claude Instant V1 100K	anthropic.claude-instant-v1:2:100K
Stable Diffusion XL 1.0	stability.stable-diffusion-xl-v1:0
Meta Llama 2 Chat 13B	meta.llama2-13b-chat-v1:0:4k

# 手順

プロビジョンドスループットは以下の手順をサポートします。

# [作成中]

プロビジョンドスループットを作成すると、プロビジョンドスループットは暫定的な未来の状態 ([Creating]) で開始され、Bedrock はプロビジョンドスループットリソースを作成します。プロビジョンドスループットリソースの作成が成功すると、プロビジョンドスループットは [InService] 状態に移行するので、プロビジョンドスループットを推論に使用できるようになります。

作成に失敗した場合、プロビジョンドスループットは [Failed] 状態に移行します。

# [更新中]

プロビジョンドスループットには以下の種類の更新を実行できます。

- プロビジョンドスループットの名前を変更する。
- 現在のカスタムモデルと同じベースモデルを使用する新しいカスタムモデルを指定する。
- ベースモデルを指定する。これは、現在のカスタムモデルのベースモデルである必要があります。

プロビジョンドスループットを更新しようとすると、Bedrock が更新を実行するとともに、プロビジョンドスループットが Updating 状態に移行します。更新が成功すると、プロビジョンドスループットは InService 状態に移行します。更新中は、エンドカスタマーからの進行中のトラフィックを中断することなく、プロビジョンドスループットを使用して推論を実行できます。更新の対象に新しいモデルが含まれている場合は、更新が完全にデプロイされるまで、古いモデルからの出力を受け取ることがあります。

更新に失敗した場合は、プロビジョンドスループットは Failed 状態に移行します。

# [削除中]

プロビジョンドスループットの削除は同期オペレーションです。プロビジョンドスループットの削除 はすぐに反映されます。

## 推論を実行する

推論を実行するには、InService 状態のプロビジョンドスループットを使用します。

リクエストしたスループットは、[プロビジョンドスループット] に設定されているスループットより も高い場合でも、調整されます (得られるスループットは [プロビジョンドスループット] で設定され たスループットです)。

# アクセス許可

ベースモデル、またはベースモデルからカスタマイズされたモデルにプロビジョンドスループットを 追加するには、まず「<u>モデルアクセス</u>」に示す手順に従ってベースモデルへのアクセスをリクエスト しておく必要があります。

暗号化されたカスタムモデルにプロビジョンドスループットを追加するには、そのモデルに対して kms:Decrypt アクションを実行するためのアクセス許可がロールに付与されている必要があります。例については、「推論でカスタマーマネージドキーを使用する」を参照してください。

# プロビジョンドスループットのコンソール手順

このセクションでは、プロビジョンドスループットのコンソール手順について説明します。

#### トピック

- プロビジョンドスループットの概要を表示する
- プロビジョンドスループットを購入
- プロビジョンドスループットの詳細を表示する
- プロビジョンドスループットを編集
- プロビジョンドスループットを削除する

[削除中] 136

# プロビジョンドスループットの概要を表示する

概要ページでは、各プロビジョンドスループットのステータスを確認できます。[概要] パネルには、各状態にあるプロビジョンドスループットリソースの数が表示されます。アクティブなプロビジョンドスループットを更新または削除したり、新しいプロビジョンドスループットを作成したりできます。

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左側のナビゲーションペインの [デプロイ] で、[プロビジョンドスループット] を選択します。
- 3. [プロビジョンドスループット] のテーブルから、各プロビジョンドスループットの概要情報を確認できます。

# プロビジョンドスループットを購入

基盤モデルまたはカスタムモデルのプロビジョンドスループットを購入できます。

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左側のメニューの [デプロイ] で、[プロビジョンドスループット] を選択します。次に、[プロビジョンドスループットを購入] を選択します。コンソールにより、[プロビジョンドスループットを購入] ページが開かれます。
- 3. [プロビジョンドスループットの詳細]で以下を行います。
  - a. プロビジョンドスループットの名前を入力します。
  - b. プロビジョンドスループットのモデルカテゴリとモデルを選択します。

### Note

現在、プロビジョンドスループットは以下のモデルで使用できます。

モデル名	プロビジョンドスループットが使用可 能なモデル ID
Titan Text G1 - Express 8K	amazon.titan-text-express-v1:0:8k
Titan Embeddings G1 - Text	amazon.titan-embed-text-v1:2:8k
Anthropic Claude V2 18K	anthropic.claude-v2:0:18k

モデル名	プロビジョンドスループットが使用可 能なモデル ID
Anthropic Claude V2 100K	anthropic.claude-v2:0:100k
Anthropic Claude Instant V1 100K	anthropic.claude-instant-v1:2:100K
Stable Diffusion XL 1.0	stability.stable-diffusion-xl-v1:0
Meta Llama 2 Chat 13B	meta.llama2-13b-chat-v1:0:4k

- c. (オプション) [タグ] では、1 つ以上のタグをこのプロビジョンドスループットに関連付けることができます。
- 4. [モデルユニットとコミットメント期間] で以下を行います。
  - a. 希望するモデルユニット数を入力します。
  - b. プロビジョンドスループットの使用を契約したい期間を選択します。カスタムモデルのコミットメントをオプトアウトするには、モデルユニット数を 1 に設定し、[コミットメントなし] を選択する必要があります。オプトアウトしない限り、割り当てられたスループットに対して 1 時間ごとに料金が計上されます。
    - Note

モデルユニットとコミットメント期間の上限については、「<u>プロビジョンドスループッ</u>ト」を参照してください。

- 5. 見積もられた購入の概要]で、推定コストを確認します。
- 6. [プロビジョンドスループットを購入] を選択します。
- 7. 表示される注記を確認し、チェックボックスを選択してコミットメント期間と価格を確認します。次に [購入を確認] を選択します。

コンソールにより、概要ページが表示されます。購入したばかりのプロビジョンドスループットは、コンソールによってそのステータスが [作成中] に設定されてテーブルに表示されます。

# プロビジョンドスループットの詳細を表示する

概要ページでは、任意のプロビジョンドスループットリソースの詳細を表示できます。

1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。

- 2. 左側のメニューで [プロビジョンドスループット] を選択します。
- 3. [プロビジョンドスループット] テーブルから、特定のプロビジョンドスループットを選択します。コンソールに詳細ページが表示されます。
- 4. [プロビジョンドスループットの概要] で、以下を行います。
- 5. コンソールの [タグ] を選択して、このプロビジョンドスループットに関連するタグを表示します。[タグの管理] を選択し、このプロビジョンドスループットにおいて、タグを追加または削除します。

# プロビジョンドスループットを編集

プロビジョンドスループットでは、いくつかのフィールドのみを編集できます。

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左側のメニューで [プロビジョンドスループット] を選択します。
- 3. [プロビジョンドスループット] テーブルから、編集するプロビジョンドスループットを選択します。
- 4. コンソールに、編集可能なプロビジョンドスループットフィールドが表示されます。
- 5. [保存]を選択して、更新を開始します。

# プロビジョンドスループットを削除する

Note

コミットメントしているプロビジョンドスループットは、コミットメント期間が終了する前 に削除することはできません。

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左側のメニューで [プロビジョンドスループット] を選択します。
- 3. [プロビジョンドスループット] テーブルから、削除するプロビジョンドスループットを選択します。
- 4. コンソールに、完全な削除が行われることを警告するモーダルフォームが表示されます。[確認] を選択して続行します。

# プロビジョンドスループット API を使用する場合

Amazon Bedrock は、プロビジョンドスループットリソースを作成および管理するための API オペレーションを装備しています。

#### トピック

- プロビジョンドスループットを作成する
- プロビジョンドスループットを使用して推論を実行する
- プロビジョンドスループットを更新する
- プロビジョンドスループットを取得する
- プロビジョンドスループットを削除する
- プロビジョンドスループットリソースをリスト表示する

# プロビジョンドスループットを作成する

基盤モデルまたはカスタムモデルのプロビジョンドスループットを作成するには、CreateProvisionedModelThroughput オペレーションを使用します。

ベースモデルに CreateProvisionedModelThroughput オペレーションを送信すると、Bedrock はリクエストされたモデルに対するアクセス許可を検証し、容量可用性を確認します。検証が成功すると、Bedrock はプロビジョンドスループットを作成し、プロビジョンドスループットの ARN を返します。

カスタムモデルの CreateProvisionedModelThroughput オペレーションを送信する と、Bedrock はリクエストされたカスタムモデルへのお客様のアクセス許可を検証し、容量可用性を確認します。カスタムモデルがカスタマーマネージド KMS キーを使用して暗号化されている場合、お客様のアクセス許可にはこのモデルに対する kms:Decrypt アクセス許可が含まれている必要があります。

# Note

commitmentDuration および modelUnits のオプションのクォータを確認する方法については、「プロビジョンドスループット」を参照してください。

Amazon Titan 基盤モデル用のプロビジョンドスループットを作成するには、デフォルトの ID の代わりに次の ID を使用してください。

API オペレーション 140

### Note

現在、プロビジョンドスループットは以下のモデルで使用できます。

モデル名	プロビジョンドスループットが使用可能な モデル ID
Titan Text G1 - Express 8K	amazon.titan-text-express-v1:0:8k
Titan Embeddings G1 - Text	amazon.titan-embed-text-v1:2:8k
Anthropic Claude V2 18K	anthropic.claude-v2:0:18k
Anthropic Claude V2 100K	anthropic.claude-v2:0:100k
Anthropic Claude Instant V1 100K	anthropic.claude-instant-v1:2:100K
Stable Diffusion XL 1.0	stability.stable-diffusion-xl-v1:0
Meta Llama 2 Chat 13B	meta.llama2-13b-chat-v1:0:4k
Meta Llama 2 Chat 13B	meta.llama2-13b-chat-v1:0:4k

以下のコード例は、AWS CLI と Python SDK を使用してプロビジョンドスループットを作成する方法を示しています。

#### **AWS CLI**

AWS CLI で次のコマンドを使用してプロビジョンドスループットを作成します。

aws bedrock create-provisioned-model-throughput

- --model-units 1
- --commitment-duration SixMonths
- --provisioned-model-name test-model
- --model-id arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-v2

#### Python (Boto)

次のコード例は、Python を使用してプロビジョンドスループットを作成する方法を示しています。

import boto3

```
bedrock = boto3.client(service_name='bedrock')
bedrock.create_provisioned_model_throughput(
    modelUnits=1,
    commitmentDuration='SixMonths',
    provisionedModelName='test-model',
    modelId='arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-v2'
)
```

# プロビジョンドスループットを使用して推論を実行する

プロビジョンドスループットを使用した推論を実行するには、<u>InvokeModel</u> または <u>InvokeModelWithResponseStream</u> オペレーションを使用します。modelId パラメータとして、プロビジョンドモデルの ARN を指定します。

以下のコード例は、AWS CLI と Python を使って、プロビジョンドスループットを使用した推論を Anthropic Claude モデルで実行する方法を示しています。プロンプト形式と推論パラメータの詳細に ついては、「基盤モデルの推論パラメータ」を参照してください。

#### **AWS CLI**

次の例は、AWS CLI を使って、プロビジョンドスループットを使用した推論を Anthropic Claude モデルで実行する方法を示しています。

```
aws bedrock-runtime invoke-model
    --model-id provisioned-model-arn \
    --body "{\"prompt\": \"\n\nHuman: story of two dogs\n\nAssistant:\"}" invoke-
model-output.txt
```

### Python (Boto)

次の例は、Python を使って、プロビジョンドスループットを使用した推論を Anthropic Claude モデルで実行する方法を示しています。

```
import boto3
import json

bedrock = boto3.client(service_name='bedrock-runtime')

body = json.dumps({"prompt": "\n\nHuman: explain black holes to 8th graders\n\nAssistant:"})
modelId = 'provisioned-model-arn'
```

```
accept = 'application/json'
contentType = 'application/json'

response = bedrock.invoke_model(body=body, modelId=modelId, accept=accept,
    contentType=contentType)
response_body = json.loads(response.get('body').read())

# text
print(response_body.get('results')[0].get('outputText'))

# embedding
print(response_body.get('embedding'))
```

# プロビジョンドスループットを更新する

指定したプロビジョンドスループットを更新するには、<u>UpdateProvisionedModelThroughput</u> オペレーションを使用します。

次のコード例は、AWS CLI と Python SDK を使用してプロビジョンドスループットを更新する方法を示しています。

#### **AWS CLI**

AWS CLI で次のコマンドを使用してプロビジョンドスループットを更新します。

```
aws bedrock update-provisioned-model-throughput
--provisioned-model-id provisioned-model-arn | provisioned-model-name
--model-arn custom-model-arn | foundation-model-arn
```

### Python (Boto)

次のコード例は、Python を使用してプロビジョンドスループットを更新する方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')

bedrock.update_provisioned_model_throughput(
    provisionedModelName='provisioned-model-arn | provisioned-model-name'
    modelArn='custom-model-arn | foundation-model-arn'
)
```

# プロビジョンドスループットを取得する

指定したプロビジョンドスループットに関する詳細情報を取得するには、GetProvisionedModelThroughput オペレーションを使用します。

次のコード例は、AWS CLI および Python を使用して情報を取得する方法を示しています。

#### **AWS CLI**

AWS CLI で次のコマンドを使用して、プロビジョンドスループットに関する情報を取得します。

```
aws bedrock get-provisioned-model-throughput
--provisioned-model-id provisioned-model-arn | provisioned-model-name
```

### Python (Boto)

次の例は、Python を使用してプロビジョンドスループットに関する情報を取得する方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')

bedrock.get_provisioned_model_throughput(
    provisionedModelName='my-provisioned-model-arn | provisioned model name'
)
```

# プロビジョンドスループットを削除する

指定したプロビジョンドスループットを削除するには、DeleteProvisionedModelThroughput オペレーションを使用します。

次のコード例は、AWS CLI と Python SDK を使用してプロビジョンドスループットを削除する方法を示しています。

#### **AWS CLI**

AWS CLI で次のコマンドを使用して、プロビジョンドスループットを削除します。

```
aws bedrock delete-provisioned-model-throughput
   --provisioned-model-id provisioned-model-arn | provisioned-model-name
```

#### Python (Boto)

次の例は、Python を使用してプロビジョンドスループットを削除する方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')

bedrock.delete_provisioned_model_throughput(
    provisionedModelName='my-provisioned-model-arn | provisioned model name'
)
```

# プロビジョンドスループットリソースをリスト表示する

定義したプロビジョンドスループットリソースをリスト表示するには、<u>ListProvisionedModelThoutputs</u> オペレーションを使用します。

次のコード例は、AWS CLI と Python SDK を使用してプロビジョンドスループットリソースをリスト表示する方法を示しています。

#### **AWS CLI**

AWS CLI で次のコマンドを使用して、プロビジョンドスループットリソースをリスト表示します。

```
aws bedrock list-provisioned-model-throughputs
```

#### Python (Boto)

次の例は、Python を使用してプロビジョンドスループットリソースをリスト表示する方法を示しています。

```
import boto3
bedrock = boto3.client(service_name='bedrock')
bedrock.list_provisioned_model_throughputs()
```

# Agents for Amazon Bedrock

### Note

Agents for Amazon Bedrock は限定プレビューリリースに同梱されています。ご利用をリク エストするには、AWS アカウントマネージャーに連絡してください。

#### Marning

2023 年 10 月 31 日から、アクショングループにおける Lambda の入力イベントおよびレス ポンス形式のペイロード構造が変更されます。廃止された形式については、「Lambda 関数 の廃止されたペイロード形式」を参照してください。

Agents for Amazon Bedrock では、開発者は、組織データとユーザー入力に基づいてアクションを 実行するようにエージェントを設定できます。Agents は、データソース、ソフトウェアアプリケー ション、ユーザーとの会話、基盤モデルの間の相互作用を調整し、自動的に API を呼び出してアク ションを実行します。開発者は Agents を統合し、生成系 AI アプリケーションのデリバリーを加速 することが簡単にでき、開発にかかる数週間分の労力を節約できます。

Agents for Amazon Bedrock を使用すると、お客様のタスクを自動化できます。例えば、お客様が 保険金請求を処理したり旅行を予約したりするのを支援するエージェントや、これらのタスクに関 する質問に回答するのを支援するエージェントを作成できます。プロビジョニングやインフラスト ラクチャの管理、カスタムコードの記述について心配する必要もありません。Agents for Amazon Bedrock は、モニタリング、暗号化、ユーザーのアクセス許可、および API 呼び出しを管理できま す。

Agents for Amazon Bedrock は次のタスクを実行できます。

- 基盤モデルを拡張してユーザーのリクエストを理解し、実行する必要のあるタスクを小さなステッ プに分割します。
- ユーザーから、自然な会話を通して追加情報を収集します。
- お客様のリクエストに応えるためのアクションを実行します。
- アクションを実行する API コールをお客様の会社のシステムに対して行います。
- 検索拡張生成 (Retrieval-Augmented Generation: RAG) を推進するためにお客様が提供するデータ ソースを使用して、パフォーマンスと正確性を向上させます。

ソースのアトリビューションを実施します。

Agents for Amazon Bedrock を利用するには、以下の手順を実行します。

1. (オプション) ベクトルデータベースを設定し、ナレッジベースを作成して、プライベートデータ をそのデータベースに保存します。詳細については、「<u>ナレッジベースを構築する</u>」を参照して ください。

- 2. ユースケースに合ったエージェントを作成してエージェントが実行できるアクションを追加し、 作成したナレッジベースをアタッチしてパフォーマンスを向上させます。詳細については、 「エージェントの構築」を参照してください。
- 3. コンソールまたは API コールでエージェントをテストし、必要に応じて設定を変更します。詳細については、「エージェントをテストする」を参照してください。
- 4. エージェントを完全に変更し、アプリケーションにデプロイする準備ができたら、エージェントのバージョンを指すエイリアスを作成します。詳細については、「<u>エージェントのデプロイ:</u>バージョニングとエイリアス」を参照してください。
- 5. エージェントのエイリアスへの API コールを行うようにアプリケーションを設定します。

#### トピック

- ナレッジベースを構築する
- エージェントの構築
- エージェントをテストする
- エージェントのデプロイ: バージョニングとエイリアス
- API を使用する場合
- Bedrock Agent で IAM が機能する仕組み

# ナレッジベースを構築する

Note

ナレッジベースは限定プレビューリリースに同梱されています。ご利用をリクエストするには、AWS アカウントマネージャーに連絡してください。

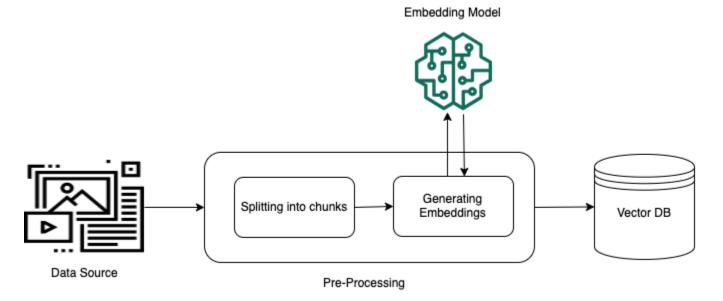
ナレッジベースを構築する 147

Amazon Bedrock では、ナレッジベースと LLM の推論機能を使用してコンテキストアプリケーションを構築することで、検索拡張生成 (Retrieval-Augmented Generation: RAG) ワークフローを有効にできます。RAG は、プライベートデータの使用と大規模言語モデル (LLM) を組み合わせた一般的な技術です。Amazon Bedrock とナレッジベースを組み合わせると、RAG ソリューションを自動化してエージェントの構築時間を短縮できるため、市場投入までの時間を短縮できます。また、ナレッジベースを追加することで、プライベートデータを活用できるようにモデルを継続的にトレーニングする必要がなくなるため、費用対効果も向上します。

RAG は、最初のステップとして、ユーザーのクエリに基づいてデータストア (通常はベクトルインデックス) から関連ドキュメントを取得します。次に、言語モデルを使用し、取得したドキュメントと元のクエリの両方を考慮してレスポンスを生成します。RAG の設定と実装に関する以降の手ステップは、ナレッジベースサービスによって自動的に行われます。

#### データを前処理する

プライベートデータから効果的に関連ドキュメントを取得できるようにするには、まずドキュメントを管理しやすいチャンクに分割するのが一般的です。次のステップは、埋め込みを生成するために、チャンクをベクトルに変換し、元のドキュメントへのマッピングを維持したままベクトルインデックスに書き込むことです。以下の図は、ベクトルデータベース用のデータの前処理を示しています。

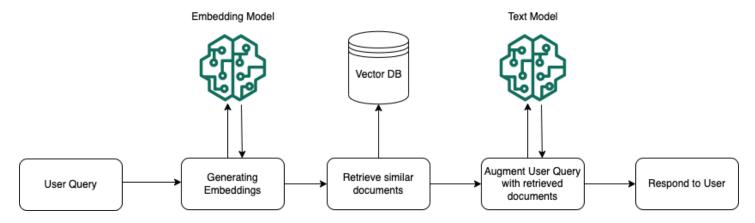


### ランタイムの実行

実行時には、埋め込みモデルを使用してユーザーのクエリをベクトルに変換します。次に、ドキュメントベクトルとユーザークエリベクトルを比較して、ベクトルインデックスをクエリしてユーザークエリの検索対象に類似したドキュメントを検索します。最後のステップとして、ベクトルインデックスを使って取得した、意味的に類似したドキュメントが、元のユーザークエリのコンテキストとして

ナレッジベースを構築する 148

追加されます。ユーザーへのレスポンスを生成すると、意味的に類似したドキュメントがテキストモデルに表示されます。以下の画像は、RAGが実行時にどのように動作してユーザークエリへのレスポンスを補強するかを示しています。



Amazon Bedrock で RAG ベースのアプリケーションを構築するには、ナレッジベースを作成してエージェントに関連付けて、独自のデータで生成機能を強化します。ナレッジベースを使用して、プライベートデータをベクトルインデックスにロードします。ナレッジベースは Amazon S3 バケットからデータを読み取って小さなチャンクに分割し、ベクトル埋め込みを生成して、指定したベクトルインデックスに埋め込みを保存します。ナレッジベースは複数のエージェントに関連付けることができます。ナレッジベースを使用してエージェントを設定すると、エージェントは対応するベクトルインデックスに格納されている情報を使用して、ユーザークエリへのレスポンスを補強できます。

#### トピック

- サービスロールを作成し、IAM アクセス許可を設定する
- 取り込み対象のデータを設定する
- ナレッジベースを作成する
- ナレッジベースを管理する
- ナレッジベースをエージェントに追加する

# サービスロールを作成し、IAM アクセス許可を設定する

ナレッジベースは、サービスロールを使用して AWS リソースにアクセスします (詳細については、 「AWS のサービスにアクセス許可を委任するロールの作成」を参照)。

ナレッジベースを作成するにはまず、サービスロールを作成し、作成したロールに信頼ポリシーをアタッチする必要があります。

#### サービスロールを作成し、信頼ポリシーをアタッチするには

1. AmazonBedrockExecutionRoleForKnowledgeBase\_ というプレフィックスの付いた IAM ロールを作成します。ロールの作成の詳細については、「<u>AWS のサービスにアクセス許可を委</u>任するロールの作成」を参照してください。

2. 作成したロール向けに信頼ポリシーを作成します。使用するポリシーの例を下記に示します。1つ以上のグローバル条件コンテキストキーを使用して、アクセス許可の範囲を制限できます。詳細については、「AWS グローバル条件コンテキストキー」を参照してください。aws:SourceAccount の値をアカウント ID に設定します。ArnEquals 条件またはArnLike 条件を使用して、範囲を特定のナレッジベースに制限できます。

### Note

セキュリティ上のベストプラクティスとして、\* は特定のナレッジベース ID に置き換えてください (作成後)。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "bedrock.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "account-id"
            },
            "ArnLike": {
                "AWS:SourceArn": "arn:aws:bedrock:region:account-id:knowledge-
base/*"
            }
        }
    }]
}
```

3. ロールへの信頼ポリシーの付与

ナレッジベースの場合は、Amazon Bedrock に次のアクセス許可を付与する必要があります。

- Amazon Bedrock 埋め込みモデルへのアクセス許可
- データソースを含む Amazon S3 オブジェクトへのアクセス許可
- (Amazon S3 データを暗号化する場合) データソースのカスタマーマネージド型 AWS KMS キーを 復号化および暗号化するアクセス許可
- (Amazon OpenSearch Service でベクトルデータベースを作成する場合) OpenSearch Service コレクションへのアクセス許可
- (Pinecone または Redis Enterprise Cloud でベクトルデータベースを作成する場合) Pinecone または Redis Enterprise Cloud アカウントを認証するための AWS Secrets Manager へのアクセス許可

#### トピック

- Amazon Bedrock モデルにアクセスするためのアクセス許可
- Amazon S3 内のデータソースにアクセスするためのアクセス許可
- (オプション) Amazon S3 のデータソースの AWS KMS キーを復号化するアクセス許可
- <u>(オプション) Amazon OpenSearch Service 内のベクトルデータベースにアクセスするためのアク</u>セス許可
- (オプション) Pinecone または Redis Enterprise Cloud 内のベクトルデータベースにアクセスする ためのアクセス許可
- <u>(オプション) AWS がデータインジェスト時の一時的なデータストレージ用の AWS KMS キーを管</u>理できるようにするためのアクセス許可

# Amazon Bedrock モデルにアクセスするためのアクセス許可

Amazon Bedrock が Amazon Bedrock モデルにアクセスしてソースデータを埋め込むことを許可するには、以下のポリシーを Amazon Bedrock サービスロールに付与します。region は、基盤モデルが存在するリージョンに、foundation-model-id は基盤モデルの ID に、それぞれ置き換えてください。現在、サポートされているモデルは、Titan Embeddings G1 - Text モデルamazon.titan-embed-text-v1のみです。

```
"bedrock:ListCustomModels"
],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "bedrock:InvokeModel"
],
    "Resource": [
        "arn:aws:bedrock:region::foundation-model/foundation-model-id"
]
}
```

### Amazon S3 内のデータソースにアクセスするためのアクセス許可

Amazon Bedrock が Amazon S3 データにアクセスすることを許可するには、以下のポリシーを Amazon Bedrock サービスロールに付与します。bucket/path/to/folder をナレッジベースのすべてのデータソースファイルを含むオブジェクトへのパスに置き換え、account-id を Amazon S3 オブジェクトが属するアカウントに置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::bucket/path/to/folder",
            "arn:aws:s3:::bucket/path/to/folder/*"
        ],
        "Condition": {
            "StringEquals": {
                "aws:PrincipalAccount": "account-id"
            }
        }
    }]
}
```

(オプション) Amazon S3 のデータソースの AWS KMS キーを復号化するアクセス許可

Amazon S3 のデータソースを AWS KMS キーで暗号化した場合は、Amazon Bedrock サービスロールに次のポリシーを付与して、Amazon Bedrock がキーを復号化できるようにします。region と account-id は、キーが属するリージョンとアカウント ID に置き換えます。key-id は、AWS KMS キーの ID に置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "KMS:Decrypt",
        ],
        "Resource": [
            "arn:aws:kms:region:account-id:key/key-id"
        ],
        "Condition": {
            "StringEquals": {
                 "kms:ViaService": [
                     "s3.region.amazonaws.com"
           }
        }
    }]
}
```

(オプション) Amazon OpenSearch Service 内のベクトルデータベースにアクセスする ためのアクセス許可

Amazon OpenSearch Service でナレッジベース用のベクトルデータベースを作成した場合は、以下のポリシーを Amazon Bedrock サービスロールに付与して、コレクションへのアクセスを許可します。 region と account-id は、データベースが属するリージョンとアカウント ID に置き換えます。 collection-id には、Amazon OpenSearch Service のコレクションの ID を入力します。複数のコレクションへのアクセスを許可するには、それらを Resources リストに追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [{
```

(オプション) Pinecone または Redis Enterprise Cloud 内のベクトルデータベースにアクセスするためのアクセス許可

Pinecone または Redis Enterprise Cloud でナレッジベース用にベクトルデータベースを作成した場合は、以下のポリシーを Amazon Bedrock サービスロールに付与することで、アカウントが AWS Secrets Manager によって認証されてデータベースにアクセスできるようにします。 region と account-id は、データベースが属するリージョンとアカウント ID に置き換えます。 secret-id はシークレットの ID に置き換えます。

シークレットを AWS KMS キーで暗号化した場合は、Amazon Bedrock サービスロールに次のポリシーを付与することで、サービスロールがキーを復号化できるようにします。region と account-id は、キーが属するリージョンとアカウント ID に置き換えます。key-id は、AWS KMS キーの ID に置き換えます。

(オプション) AWS がデータインジェスト時の一時的なデータストレージ用の AWS KMS キーを管理できるようにするためのアクセス許可

データソースの取り込み中に一時データストレージ用の AWS KMS キーを作成できるようにするには、以下のポリシーを Amazon Bedrock サービスロールに付与します。region、account-id、key-id を適切な値に置き換えます。

## 取り込み対象のデータを設定する

ナレッジベースを作成するには、まず以下のアクションを実行する必要があります。

- 1. ソースデータを設定して Amazon S3 バケットにアップロードします。
- 2. ベクトルデータベースを作成し、Amazon Bedrock がインデックスにデータを書き込んだり、インデックスにアクセスしたりするためのフィールドを設定します。

#### ソースデータを設定する

デフォルトでは、ナレッジベースはソースデータを自動的にチャンクに分割します。各チャンクには最大で 200 個のトークンが含まれます。ドキュメントは、200 個未満のトークンしか含まない場合には分割されません。ドキュメントを 200 個未満のトークンしか含まないチャンクに分割するには、ドキュメントを小さいチャンクに分割する前処理を行っておきます。

ソースデータを前処理したら、Amazon S3 バケットにアップロードします。Amazon S3 バケットをセットアップする方法については、「<u>Amazon S3 の開始方法</u>」を参照してください。ナレッジベースは次のファイル形式をサポートしています。

- プレーンテキスト (.txt)
- · Markdown (.md)
- HyperText マークアップ言語 (HTML)
- Microsoft Word ドキュメント (.doc または .docx)
- カンマ区切り値 (.csv)
- Microsoft Excel スプレッドシート (.xls または .xlsx)
- ポータブルドキュメント形式 (.pdf)

ベクトルデータベースを作成する

サポートされている以下の手段の1つを使って、ベクトルインデックスを作成します。

- Amazon OpenSearch Service
- Pinecone
- Redis Enterprise Cloud

Amazon Bedrock はデータを処理した後、作成されていたインデックスに次の情報を書き込みます。

- ドキュメントから抽出されたテキスト。
- 埋め込みモデルによって生成された、テキストに対応するベクトル。
- テキスト抽出元ソースファイルの Amazon S3 パス。

#### トピック

Amazon OpenSearch Service でベクトルインデックスを作成する

- Pinecone でベクトルインデックスを作成する
- Redis Enterprise Cloud でベクトルインデックスを作成する

# Amazon OpenSearch Service でベクトルインデックスを作成する

1. Amazon OpenSearch Service にログインしてコレクションを作成します。コレクション ARN をメモしておきます。これは、ナレッジベースを作成するときに入力することになります。

2. コレクションが作成されたら、そのコレクションを選択してベクトルインデックスを作成します。

Amazon OpenSearch Service でのベクトルインデックスの設定に関する詳細なドキュメントについては、「ベクトル検索コレクションの使用」を参照してください。

ベクトルインデックスを設定するときは、コレクション ARN をメモしておきます。これは、ナレッジベースを作成するときに入力することになります。

ベクトルインデックスを作成するときに必要となる追加の設定は以下のとおりです。

- ベクトルインデックス名 ベクトルインデックスの名前。任意の有効な名前を選択します。ここで 選択した名前は、後でナレッジベースを作成するときに [ベクトルインデックス名] フィールドに 入力します。
- ベクトルフィールド ベクトル埋め込みが保存されるフィールド。任意の有効な名前を選択します。ここで選択した名前は、後でナレッジベースを作成するときに [ベクトルフィールド] フィールドに入力します。
- ディメンション ベクトルのディメンション (次元) の数。Titan Embeddings Model を使用する場合は 1536 を選択してください。この番号は、後でナレッジベースを作成するときに [ディメンション] フィールドに入力します。
- 距離メトリクス ベクトル間の類似性を測定するために使用されるメトリクス。ユースケースに合わせてさまざまなメトリクスを試してみることをお勧めします。Titan Embeddings Model を使用する場合は、最初にコサイン類似性を試すことができます。
- メタデータの管理 このフィールドを拡張し、ナレッジベースがベクトルを使用して取得できる追加のメタデータが保存されるようにベクトルインデックスを設定します。設定が必要なフィールドは以下のとおりです。
  - テキストフィールド Amazon Bedrock はデータ内の未加工テキストをチャンク化し、チャンクをこのフィールドに保存します。

• マッピングフィールド - テキストが保存されるフィールド。任意の有効な名前を選択します。ここで選択した名前は、後でナレッジベースを作成するときに [テキストフィールド名]フィールドに入力します。

- データタイプ [String] を選択します。
- フィルター可能 [False] を選択します。
- Bedrock が管理するメタデータフィールド このフィールドには、Amazon Bedrock により、 データに関連するメタデータが保存されます。メタデータには、以下が含まれます。
  - マッピングフィールド 任意の有効な名前を選択します。ここで選択した名前は、後でナレッジベースを作成するときに [Bedrock が管理するメタデータフィールド名] フィールドに入力します。
  - データタイプ [String] を選択します。
  - フィルター可能 [False] を選択します。

#### セキュリティ設定

ナレッジベースを作成したら、Amazon OpenSearch Service に戻り、コレクションの [ネットワークアクセス] と [データアクセス] の設定を調整することでベクトルデータベースのセキュリティ設定を行う必要があります。詳細については、「ナレッジベースを作成する」を参照してください。

Pinecone でベクトルインデックスを作成する

# Note

Pinecone を使用する場合、AWS が Amazon Bedrock サービスを提供するために指定されたサードパーティーソースに自動的にアクセスすることを許可することに同意します。お客様は、サードパーティーサービスからのデータの使用および転送に適用されるいかなるサードパーティー規約をも遵守する必要があります。

Pinecone でのベクトルインデックスの設定に関する詳細なドキュメントについては、「 $\underline{\mathsf{Manage}}$  indexes」を参照してください。

ベクトルインデックスを設定するときは、次の情報をメモしておきます。この情報は、ナレッジベースを作成するときに入力することになります。

• インデックス管理ページのエンドポイント。

• (オプション) データベースに新しいデータを書き込むために使用する名前空間。詳細については、「Using namespaces」を参照してください。

Pinecone インデックスを作成するときに必要となる追加の設定は以下のとおりです。

- 名前 ベクトルインデックスの名前。任意の有効な名前を選択します。ここで選択した名前は、後でナレッジベースを作成するときに [ベクトルインデックス名] フィールドに入力します。
- ディメンション ベクトルのディメンション (次元) の数。Titan Embeddings Model を使用する場合は 1536 を選択してください。この番号は、後でナレッジベースを作成するときに [ディメンション] フィールドに入力します。
- 距離メトリクス ベクトル間の類似性を測定するために使用されるメトリクス。ユースケースに合わせてさまざまなメトリクスを試してみることをお勧めします。Titan Embeddings Model を使用する場合は、最初にコサイン類似性を試すことができます。

#### Secrets Manager を設定する

Pinecone インデックスにアクセスするには、AWS Secrets Manager を使用して Amazon Bedrock に Pinecone API キーを提供する必要があります。

Pinecone 設定用のシークレットを設定するには

- 1. 「<u>AWS Secrets Manager シークレットを作成する</u>」の手順に従い、キーを apiKey に、値を Pinecone インデックスにアクセスするための API キーに設定します。
- 2. この API キーを見つけるには、Pinecone コンソールを開いて [API キー] を選択します。
- 3. シークレットを作成したら、そのシークレットの ARN を書き留めます。この ARN は、後でナレッジベースを作成するときに、[認証情報シークレット ARN] フィールドに入力します。

Redis Enterprise Cloud でベクトルインデックスを作成する

# Note

Redis Enterprise Cloud を使用する場合、AWS が Amazon Bedrock サービスを提供するために指定されたサードパーティーソースに自動的にアクセスすることを許可することに同意します。お客様は、サードパーティーサービスからのデータの使用および転送に適用されるいかなるサードパーティー規約をも遵守する必要があります。

Redis Enterprise Cloud でのベクトルインデックスの設定に関する詳細なドキュメントについては、「Integrating Redis Enterprise Cloud with Amazon Bedrock」を参照してください。

ベクトルインデックスを設定するときは、次の情報をメモしておきます。この情報は、ナレッジベースを作成するときに入力することになります。

- データベースのパブリックエンドポイント URL。
- データベースのベクトルインデックスの名前。

Secrets Manager を設定する

Redis Enterprise Cloud クラスターにアクセスするには、AWS Secrets Manager を使用して Redis Enterprise Cloud のセキュリティ設定を Amazon Bedrock に提供する必要があります。

Redis Enterprise Cloud 設定のシークレットを設定するには

- 1. 「<u>Transport Layer Security (TLS)</u>」の手順に従って、Amazon Bedrock でデータベースを使用するように TLS を有効にします。
- 2. 「<u>AWS Secrets Manager シークレットを作成する</u>」の手順を実行します。シークレット内の Redis Enterprise Cloud 設定の適切な値を使用して、以下のキーを設定します。
  - username Redis Enterprise Cloud データベースにアクセスするためのユーザー名。自身のユーザー名を確認するには、Redis コンソールでデータベースの [セキュリティ] セクションを参照してください。
  - password Redis Enterprise Cloud データベースにアクセスするためのパスワード。自身のパスワードを確認するには、Redis コンソールでデータベースの [セキュリティ] セクションを参照してください。
  - serverCertificate Redis Cloud 認証機関からの証明書の内容。「<u>Download CA</u> <u>certificates</u>」の手順に従って、Redis 管理コンソールを使ってサーバー証明書をダウンロードします。
  - clientPrivateKey Redis Cloud 認証機関からの証明書のプライベートキー。「<u>Download CA certificates</u>」の手順に従って、Redis 管理コンソールを使ってサーバー証明書をダウンロードします。
  - clientCertificate Redis Cloud 認証機関からの証明書のパブリックキー。「<u>Download CA certificates</u>」の手順に従って、Redis 管理コンソールを使ってサーバー証明書をダウンロードします。

3. シークレットを作成したら、そのシークレットの ARN を書き留めます。この ARN は、後でナレッジベースを作成するときに、[認証情報シークレット ARN] フィールドに入力します。

### ナレッジベースを作成する

#### ナレッジベースを作成するには

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左のナビゲーションペインで [ナレッジベース] を選択します。
- 3. [ナレッジベース] セクションで、 [ ナレッジベースを作成] を選択します。
- 4. [ナレッジベースの詳細を入力」ページで、以下のアクションを実行します。
  - a. [ナレッジベースの詳細] セクションで、ナレッジベースの名前を入力するとともに、必要な場合は説明も入力します。
  - b. [IAM 許可] セクションで、Amazon Bedrock でサービスロールを新規作成するか、Amazon Bedrock が自動的に他のサービスにアクセスできるようにする既存のサービスロールを使用します。詳細については、「<u>サービスロールを作成し、IAM アクセス許可を設定する</u>」を参照してください。
  - c. ナレッジベースにタグを付ける場合は、[タグ] セクションで [新しいタグを追加] を選択し、 キーと値のペアとしてのタグを追加します。
  - d. [次へ] を選択します。
- 5. [データソースを設定] ページで、次の操作を実行して、ナレッジベースに追加するデータソース の情報を入力します。
  - a. [データソース] セクションで、以下のアクションを実行します。
    - i. データソースの名前と Amazon S3 オブジェクトの URI を入力します。
    - ii. Amazon S3 データを暗号化した場合は、 [Amazon S3 データのカスタマーマネージド AWS KMS キー] に AWS KMS キーを入力することで、Amazon Bedrock がデータを復号化できるようにします。
    - iii. Amazon Bedrock は、データを埋め込みに変換する際、デフォルトで、AWS が所有 および管理するキーを使用して一時データを暗号化します。[一時データストレージの AWS KMS キー] で、[暗号化設定をカスタマイズする (高度)] チェックボックスをオン にできます。

ナレッジベースを作成する 161

b. [埋め込みモデル] セクションで、ナレッジベースのデータを埋め込みに変換するための埋め 込みモデルを選択します。現時点では、Amazon Bedrock Titan の埋め込みモデルのみが選 択可能です。

c. [ベクトルデータベース] セクションで、作成済みのベクトルデータベースが含まれているサービスを選択します。データベースに必要なフィールドが既に設定されていることを確認します (詳細については、「取り込み対象のデータを設定する」を参照)。Amazon Bedrockが埋め込みを保存、更新、管理できるようにナレッジベースの情報をデータベースにマッピングするためのフィールドに入力します。

#### Note

Amazon OpenSearch Service でデータベースを使用する場合は、事前に [メタデータフィールドマッピング] でフィールドを設定しておく必要があります。Pinecone または Redis Enterprise Cloud のデータベースを使用している場合は、ここでこれらのフィールドに名前を指定すると、Amazon Bedrock がそれらの名前をベクトルインデックスに動的に作成します。

- d. [次へ] を選択します。
- 6. [確認および作成] ページで、ナレッジベースの設定と詳細を確認します。変更が必要なセクションで [編集] を選択します。問題がなければ、[ナレッジベースを作成] を選択します。
- 7. ナレッジベース作成プロセスが開始され、ソースの [ステータス] が [進行中] になります。ナレッジベースの作成にかかる時間は、入力したデータの量に左右されます。ナレッジベースの作成が完了すると、成功を示す緑色のバナーが表示され、ナレッジベースの [ステータス] が [準備完了] に変わります。

#### Note

Amazon OpenSearch Service のベクトルデータベースに埋め込みを保存する場合は、ナレッジベースの作成後に OpenSearch Service でナレッジベースのセキュリティ設定を行うことを忘れないでください。詳細については、「セキュリティ設定」を参照してください。

新規作成したナレッジベースのセキュリティ設定を行います。設定するデータベースに対応するタブの手順に従います。

ナレッジベースを作成する 162

#### OpenSearch Service

#### データアクセスポリシーを作成するには

- 1. OpenSearch Service コンソールで、コレクションに移動します。
- 2. [データアクセスを管理]を選択します。
- 3. [アクセスポリシーを作成] を選択し、ポリシーに名前と説明 (オプション) を指定します。
- 4. ポリシー定義方法として [JSON] を選択し、次の JSON オブジェクトをエディターに貼り付けて、*collection-name* をコレクションの名前に、また *service-role-arn* をナレッジベースの作成時に渡したロール ARN にそれぞれ置き換えます。

```
1}
    {
        "Rules": [
          {
            "Resource": [
               "index/collection_name/*"
            ],
            "Permission": [
               "aoss:DescribeIndex",
               "aoss:ReadDocument",
              "aoss:WriteDocument"
            ],
            "ResourceType": "index"
          }
        ],
        "Principal": [
          "service-role-arn"
        ],
        "Description": "Data access policy"
    }
]
```

5. [作成] を選択して、ポリシーを作成します。

### Pinecone or Redis Enterprise Cloud

サードパーティーのナレッジベースを統合するには、knowledge-base-arn の代わりに以下のポリシーを Amazon Bedrock サービスロールに付与します。

ナレッジベースを作成する 163

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "bedrock: AssociateThirdPartyKnowledgeBase"
        ],
        "Resource": [
            "knowledge-base-arn"
        ],
        "Condition": {
            "StringEquals": {
                 "bedrock:ThirdPartyKnowledgeBaseCredentialsSecretArn": "secret-arn"
            }
        }
    }]
}
```

## ナレッジベースを管理する

ナレッジベースを管理するには

- 1. Amazon Bedrock コンソール (<a href="https://console.aws.amazon.com/bedrock/">https://console.aws.amazon.com/bedrock/</a>) を開きます。
- 2. 左のナビゲーションペインで [ナレッジベース] を選択します。
- 3. ナレッジベースの詳細を表示するには、[名前] でソースの名前を選択します。または、ソースの横にあるラジオボタンを選択し、[編集] を選択します。
- 4. 詳細ページでは、次のアクションを実行できます。
  - ナレッジベースの詳細を変更するには、[ナレッジベース概要] セクションで [編集] を選択します。
  - ナレッジベースに付いているタグを更新するには、[タグ] セクションで [タグの管理] を選択します。
  - ナレッジベース作成の元となったデータソースを更新し、変更を同期する必要がある場合は、[データソース] セクションで [同期] を選択します。
  - データソースの詳細を表示するには、[データソース名] を選択します。詳細ページでは、[同期履歴] セクションの同期イベントの横にあるラジオボタンを選択し、[警告を表示] を選択すると、データインジェストジョブのファイルが同期に失敗した理由を確認できます。

ナレッジベースを管理する 164

ナレッジベースに使用される埋め込みモデルを管理するには、[プロビジョンドスループットを編集]を選択します。

• 編集が終了したら、[変更を保存] を選択します。

#### ナレッジベースを削除するには

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左のナビゲーションペインで [ナレッジベース] を選択します。
- 3. ソースを削除するには、ソースの横にあるラジオボタンを選択し、[削除] を選択します。または、[名前] でソースの名前を選択し、詳細ページの右上にある [削除] を選択します。
- 4. ナレッジベースの削除に関する警告を確認してください。これらの条件に同意する場合は、入力ボックスに「delete」を入力し、[削除] を選択して確定します。
- 5. ナレッジベースを追加したエージェントから、必ずそのナレッジベースを削除してください。これを行うには、次の手順を実行します。
  - a. 左のナビゲーションペインで [エージェント] を選択します。
  - b. [名前] で、ナレッジベースを削除するエージェントの名前を選択します。
  - c. もう存在しないナレッジベースへの参照をエージェントから削除するよう警告する赤いバナーが表示されます。
  - d. 削除するナレッジベースの横にあるラジオボタンを選択します。[詳細] を選択して、[削除] を選択します。

# ナレッジベースをエージェントに追加する

ナレッジベースをエージェントに追加するには

- 1. Amazon Bedrock コンソール (https://console.aws.amazon.com/bedrock/) を開きます。
- 2. 左のナビゲーションペインで [エージェント] を選択します。
- 3. [名前] で、ナレッジベースを追加したいエージェントの名前を選択します。
- 4. [ナレッジベース] セクションで [追加] を選択します。
- 5. [ナレッジベースを選択] のドロップダウンリストからナレッジベースを選択し、エージェントに対してナレッジベースに関する指示を指定します。

# エージェントの構築

Note

Amazon Bedrock は限定プレビューリリースに同梱されています。ご利用をリクエストするには、AWS アカウントマネージャーに連絡してください。

エージェントを構築するには、以下のコンポーネントを設定します。

- エージェント自体の設定: エージェントの目的を定義します。
- アクショングループ: エージェントに実行させるアクションを定義します。
- (オプション) データソースのナレッジベース: エージェントの生成機能を強化します。
  - Note

エージェントにナレッジベースをアタッチする予定の場合は、まず、「<u>ナレッジベースを</u> 構築する」の手順に従ってナレッジベースを設定します。

次の手順を実行して、お客様とやり取りするエージェントを設定します。

- 1. エージェントがオーケストレーションに使用できる基本モデルを選択します。
- 2. エージェントに指示を与え、エージェントが実行できるアクションを記述して、アクションの API スキーマを指定します。
- 3. ナレッジベースにデータソースを設定し、ナレッジベースをエージェントに関連付けることで、 エージェントにプライベートデータを追加します。
- 4. 作業中のエージェントのドラフトを繰り返しテストします。
- 5. エージェントが想定どおりに動作するようになったら、エイリアスを作成して、エージェントをアプリケーションと統合できるようにします。
- 6. エージェントのエイリアスへの API コールを行うようにアプリケーションを設定します。
- 7. 作業中のエージェントのドラフトを更新し、必要に応じて新しいエイリアスを作成します。

#### トピック

• サービスロールを作成し、IAM アクセス許可を設定する

エージェントの構築 166

- エージェントを作成する
- エージェントを編集する

## サービスロールを作成し、IAM アクセス許可を設定する

エージェントを作成するにはまず、サービスロールを作成し、作成したロールに信頼ポリシーをアタッチする必要があります。

サービスロールを作成し、信頼ポリシーをアタッチするには

- AmazonBedrockExecutionRoleForAgents\_というプレフィックスの付いた IAM ロールを 作成します。ロールの作成の詳細については、「AWS のサービスにアクセス許可を委任する ロールの作成」を参照してください。
- 2. 作成したロール向けに信頼ポリシーを作成します。使用するポリシーの例を下記に示します。1つ以上のグローバル条件コンテキストキーを使用して、アクセス許可の範囲を制限できます。詳細については、「AWS グローバル条件コンテキストキー」を参照してください。aws:SourceAccount の値をアカウント ID に設定します。ArnEquals 条件またはArnLike 条件を使用して、範囲を特定のエージェントに制限できます。
  - Note

セキュリティ上のベストプラクティスとして、\* は特定のエージェント ID に置き換えてください (作成後)。

```
}
}
}
```

#### ロールへの信頼ポリシーの付与

ロールベースのアクセス許可を設定することで、以下のリソースへのアクセス許可を自動的に Amazon Bedrock に付与します。

- Amazon Bedrock のベースモデル
- エージェントのアクショングループ用の OpenAPI スキーマを含む Amazon S3 オブジェクト

また、Lambda 関数のリソースベースのポリシーを設定することで、Amazon Bedrock がエージェント内のアクショングループの AWS Lambda 関数にアクセスできるようにするためのアクセス許可を提供する必要があります。

エージェントにナレッジベースをアタッチする場合は、Amazon Bedrock がナレッジベースをクエリ するためのアクセス許可を付与することも必要です。

#### トピック

- Amazon Bedrock ベースモデルにアクセスするためのアクセス許可
- Amazon S3 のアクショングループ API スキーマにアクセスするためのアクセス許可
- アクショングループの Lambda 関数にアクセスするためのアクセス許可
- (オプション) ナレッジベースにアクセスするためのアクセス許可

Amazon Bedrock ベースモデルにアクセスするためのアクセス許可

Amazon Bedrock がオーケストレーションのために基盤モデルを使用するためのアクセス許可を付与するには、以下のポリシーを IAM ロールに付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Action": [
     "bedrock:InvokeModel"
```

```
],
"Resource": [
         "arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-v1",
         "arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-v2",
         "arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-instant-v1"
        ]
    }]
}
```

### Amazon S3 のアクショングループ API スキーマにアクセスするためのアクセス許可

Amazon Bedrock がエージェントのアクショングループの API スキーマの Amazon S3 URI にアクセスするためのアクセス許可を付与します。次のポリシーをこの IAM ロールに付与します。
[Resource] フィールドでは、スキーマを含む Amazon S3 オブジェクトを指定するか、各スキーマの URI をリストに追加できます。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::path/to/schema"
        ],
        "Condition": {
            "StringEquals": {
                 "aws:PrincipalAccount": "account-id"
            }
        }
    }]
}
```

# アクショングループの Lambda 関数にアクセスするためのアクセス許可

Amazon Bedrock がエージェントのアクショングループの Lambda 関数にアクセスするためのアクセス許可を付与します。次のリソースベースのポリシーを Lambda 関数にアタッチします。[Condition] には、ユーザーアカウントの ARN とエージェントの ARN を指定します。[Condition] フィールドはオプションですが、セキュリティ上の理由から指定することをお勧めします。詳細については、「Lambda でのリソースベースのポリシーの使用」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "statement-name",
        "Effect": "Allow",
        "Principal": {
            "Service": "bedrock.amazonaws.com"
        },
        "Action": [
            "lambda:InvokeFunction",
        ],
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "account-id"
            },
            "ArnLike": {
                "AWS:SourceArn": "arn:aws:bedrock:region:account-id:agent/agent-id"
        }
    }]
}
```

# (オプション) ナレッジベースにアクセスするためのアクセス許可

Amazon Bedrock が設定したナレッジベースにアクセスするためのアクセス許可を付与するには、以下のポリシーを IAM ロールに付与します。account-id と knowledge-base-id は適切な値に、statement-name は任意のステートメント名に置き換えてください。

}

## エージェントを作成する

エージェントを作成するには、<a href="https://console.aws.amazon.com/bedrock/">https://console.aws.amazon.com/bedrock/</a> に移動し、左側のナビゲーションペインで [エージェント] を選択します。次に、[エージェント] セクションの右上にある [作成] を選択します。

#### エージェントの詳細を入力する

- [エージェント名] セクションで、エージェントの名前を入力します。また、オプションで説明を 付けることもできます。
- 2. [ユーザー入力] セクションで、ユーザーがタスクを完了しようとするときにエージェントから追加情報を要求するかどうかを選択します。[いいえ] を選択した場合、エージェントはユーザーに追加情報を要求せず、タスクを完了するのに十分な情報がないことをユーザーに通知します。
- 3. [IAM アクセス許可] セクションで、AWS Identity and Access Management (IAM) ロールを選択します。このロールは、他の AWS サービスにアクセスするためのアクセス許可を Amazon Bedrock に付与します。詳細については、「<u>サービスロールを作成し、IAM アクセス許可を設定</u>する」を参照してください。
- 4. [アイドルセッションタイムアウト] セクションで、Amazon Bedrock がユーザーとのセッション を開いたままにする期間を選択します。Amazon Bedrock は、セッション期間中はセッション変数を保持するので、エージェントが同じ変数で会話を再開できます。
- 5. エージェントの設定が完了したら、[次へ]を選択します。

#### モデルを選択する

- 1. エージェントがオーケストレーションを実行するには、モデルを選択する必要があります。モデルプロバイダーを選択し、ドロップダウンメニューでモデルを選択してエージェントをトレーニングします。
- 2. [エージェント向けの指示] で、エージェントに何をすべきか、どのようにユーザーとやり取りすべきかを伝えるための詳細を入力します。以下の指示の例を参照してください。

You are an office assistant in an insurance agency. You are friendly and polite.
You help with managing insurance claims and coordinating pending paperwork.

エージェントを作成する 171

### アクショングループを追加

1. アクショングループの名前を入力し、[アクショングループの説明] にアクションの内容を指定します。例えば、アクショングループの名前として *SetupDNSResources* を入力し、説明として次の文字列を指定できます。

Use this Action whenever you want to setup DNS resources for setting up a website.

2. [Lambda 関数の選択] で、AWS Lambda で作成した Lambda 関数を選択します。Lambda 関数は、アクションの呼び出し時に実行されるビジネスロジックを提供します。使用する関数のバージョンを選択します。

### Note

Amazon Bedrock サービスプリンシパルが Lambda 関数にアクセスできるようにポリシーを設定することを忘れないでください。詳細については、「<u>サービスロールを作成し、IAM アクセス許可を設定する</u>」を参照してください。ポリシーを設定するときに、SourceArn として、オプションでエージェントの ARN を指定できます。

Lambda 入力イベントの一般的な形式は次のとおりです。入力イベント関連のフィールドを使用して Lambda 関数を作成します。詳細については、AWS Lambda ドキュメントの「<u>イベント駆</u>動型の呼び出し」を参照してください。

### Marning

2023 年 10 月 31 日から、Lambda の入力イベントおよびレスポンス形式のペイロード構造が変更されます。廃止された形式については、「<u>Lambda 関数の廃止されたペイ</u>ロード形式」を参照してください。

```
{
    "messageVersion": "1.0",
    "agent": {
         "name": "string",
         "id": "string",
         "alias": "string",
         "version": "string"
},
```

```
"inputText": "string",
    "sessionId": "string",
    "actionGroup": "string",
    "apiPath": "string",
    "httpMethod": "string",
    "parameters": [
        {
            "name": "string",
            "type": "string",
            "value": "string"
        },
    . . .
    ],
    "requestBody": {
        "content": {
            "<content_type>": {
                 "properties": [
                    {
                        "name": "string",
                        "type": "string",
                        "value": "string"
                     },
                              . . .
                 ]
            }
        }
    },
    "sessionAttributes": {
        "string": "string",
    }
}
```

Amazon Bedrock は、Lambda 関数から以下の形式のレスポンスを想定しています。

```
{
    "messageVersion": "1.0",
    "response": {
        "actionGroup": "string",
        "apiPath": "string",
        "httpMethod": "string",
        "httpStatusCode": number,
        "responseBody": {
            "<contentType>": {
```

```
"body": "string"
}
}
```

以下は、Python で Lambda 関数を定義する方法の例です。

```
def lambda_handler(event, context):
    response_body = {
        'application/json': {
            'body': "<sample response>"
        }
    }
    action_response = {
        'actionGroup': event['actionGroup'],
        'apiPath': event['apiPath'],
        'httpMethod': event['httpMethod'],
        'httpStatusCode': 200,
        'responseBody': response_body
    }
    api_response = {
        'messageVersion': '1.0',
        'response': action_response}
    return mock_api_response
```

3. [API スキーマを選択] で、スキーマの Amazon S3 URI へのリンクを、アクショングループ の API の説明、構文、パラメータとともに入力します。スキーマの例については、https://github.com/OAI/OpenAPI-Specification/tree/main/examples/v3.0 を参照してください。

### Note

スキーマの Amazon S3 URI にアクセスするためのアクセス許可を IAM ロール に割り当てるのを忘れないでください。詳細については、「<u>サービスロールを作成し、IAM アク</u>セス許可を設定する」を参照してください。

4. [別のアクショングループを追加] を選択して、エージェント用に別のアクショングループを設定します。アクショングループの追加が完了したら、[次へ] をクリックします。

#### ナレッジベースを追加する

1. ナレッジベースをまだ作成していない場合は、左側のナビゲーションペインで [ナレッジベース] を選択し、「<u>ナレッジベースを作成する</u>」の指示に従ってナレッジベースを作成してください。 既に作成している場合は、ドロップダウンメニューからナレッジベースを選択します。

2. [エージェント向けのナレッジベースの指示] で、エージェントがナレッジベースをどのように使用すべきかを示すプロンプトを書きます。例えば、「############」というナレッジベースには、次のテキストを使用できます。

```
Use this knowledge base whenever you are creating a DNS record
```

3. [別のナレッジベースをさらに追加] を選択して、エージェント向けに別のナレッジベースを設定 します。ナレッジベースを追加したら、[次へ] を選択します。

エージェントの設定を確認し、変更したいセクションがあれば [編集] を選択します。エージェントを作成する準備ができたら、[作成] を選択します。処理が完了すると、エージェントが正常に作成されたことを知らせる緑色のバナーが上部に表示されます。

Lambda 関数の廃止されたペイロード形式

参考までに、Lambda 関数の入力イベントの廃止された形式は次のとおりです。アクショングループを最新の形式に更新してください (詳細については「エージェントを作成する」を参照)。

```
{
    "messageVersion": "1.0",
    "agent": {
        "name": "string",
        "id": "string",
        "alias": "string",
        "version": "string"
    },
    "inputText": "string",
    "sessionId": "string",
    "actionGroups": [
        {
            "actionGroup": "string",
            "apiPath": "string",
            "httpMethod": "string",
            "parameters": [
                     "name": "string",
```

```
"type": "string",
                      "value": "string"
                 },
                  . . .
             ],
             "requestBody": {
                 "content": {
                      "<content_type>": {
                          "properties": [
                              {
                                   "name": "string",
                                   "type": "string",
                                   "value": "string"
                              },
                               . . .
                          ]
                      }
                 }
             }
        },
         . . .
    ],
    "sessionAttributes": {
         "string": "string",
    }
}
```

参考までに、Lambda 関数のレスポンスの廃止された形式は次のとおりです。アクショングループを 最新の形式に更新してください。

```
}
]
}
```

# エージェントを編集する

エージェントを作成した後で、必要に応じてエージェントの設定を更新することができます。設定は 作業中のドラフトに適用されます。

エージェントの設定を編集するには

- 1. [エージェント] セクションでエージェントを選択します。
- 2. [エージェントの概要] セクションで [編集] を選択します。
- 3. 必要に応じて既存のフィールドを編集します。
- 4. [変更を保存]を選択します。上部に成功を示す緑色のバナーが表示されます。

エージェント用に別の基盤モデルを試してみたり、エージェント向けの指示を変更したりすることもできます。これらの変更は作業中のドラフトにのみ適用されます。

エージェントが使用する基盤モデルや、エージェントへの指示を変更するには

- 1. [エージェント] セクションでエージェントを選択します。
- 2. [作業中のドラフト] セクションで [表示] を選択するか、作業中のドラフトを選択します。
- 3. [モデルの詳細] セクションで [編集] を選択します。
- 4. 必要に応じてフィールドを編集します。
- 5. [保存] を選択します。この操作を行っても、異なるウィンドウに移動することはありません。上 部に成功を示す緑色のバナーが表示されます。
- 6. 右側のパネルで更新されたエージェントをテストし、必要に応じて変更します。
- 7. [保存して終了] を選択し、作業中のドラフトのページに戻ります。

### エージェントのアクショングループを管理する

エージェントを作成したら、アクショングループをさらに追加したり編集したりできます。追加と編集は作業中のドラフト内で行います。これらの操作を実行するには、[エージェント] セクションからエージェントを選択し、[作業中のドラフト] セクションで作業中のドラフトを選択します。

エージェントを編集する 177

### アクショングループを追加するには

- 1. [アクショングループ] セクションで [追加] を選択します。
- 2. [アクショングループの詳細] に入力する
- 3. [追加] を選択します。上部に成功を示す緑色のバナーが表示されます。

### アクショングループを編集するには

- 1. 次のいずれかを実行します。
  - 編集するアクショングループの横にあるラジオボタンをクリックして、[編集] を選択します。
  - アクショングループを選択して、その詳細を表示します。上部の[編集]を選択します。
- 2. 必要に応じて既存のフィールドを編集します。
- 3. [保存] を選択します。この操作を行っても、異なるウィンドウに移動することはありません。問題がない場合は、成功を示す緑色のバナーが表示され、編集にエラーがある場合は、赤色のバナーが表示されます。
- 4. 右側のパネルで更新されたエージェントをテストし、必要に応じて変更します。
- 5. [保存して終了] を選択し、作業中のドラフトのページに戻ります。

### エージェントのナレッジベースを管理する

エージェントを作成したら、ナレッジベースをさらに追加したり編集したりできます。追加と編集は作業中のドラフト内で行います。これらの操作を実行するには、[エージェント] セクションからエージェントを選択し、[作業中のドラフト] セクションで作業中のドラフトを選択します。

### ナレッジベースを追加するには

- 1. [ナレッジベース] セクションで [追加] を選択します。
- 2. 作成したナレッジベースを選択し、エージェントがナレッジベースと通信する方法を指示します。
- 3. [追加] を選択します。上部に成功を示す緑色のバナーが表示されます。

#### ナレッジベースを編集するには

1. ナレッジベースを選択します。

エージェントを編集する 178

- 2. [ナレッジベースの詳細] ページで[編集] を選択します。
- 3. 必要に応じて既存のフィールドを編集します。
- 4. [保存] を選択します。この操作を行っても、異なるウィンドウに移動することはありません。問題がない場合は、成功を示す緑色のバナーが表示され、編集にエラーがある場合は、赤色のバナーが表示されます。
- 5. 右側のパネルで更新されたエージェントをテストし、必要に応じて変更します。
- 6. [保存して終了] を選択し、作業中のドラフトのページに戻ります。

#### ナレッジベースを削除するには

- 1. 削除するナレッジベースの横にあるラジオボタンを選択します。
- 2. [削除] を選択します。

### エージェントをテストする

作成したエージェントは [エージェント] セクションに表示されます。エージェントを初めて作成すると、作業中のドラフトが作成されます。作業中のドラフトは、エージェントを反復的に構築するために使用できるエージェントのバージョンです。デフォルトでは、作業中のドラフトをAgentTestAlias で操作できます。また、別のエイリアスを選択してテストすることもできます。テストウィンドウでは、各レスポンスのトレースを表示するかどうかを選択できます。トレースは、エージェントの推論プロセスをステップごとに表示するので、エージェントのデバッグに役立つツールです。トレースの詳細については、「トレースの有効化」を参照してください。APIを使用してトレースを有効にする方法については、「エージェントを呼び出す」を参照してください。

テストウィンドウにアクセスしてエージェントとやり取りできます。この機能を使用してエージェントをデバッグし、想定どおりに動作するように必要な変更を加えることができます。

#### エージェントをテストするには

- 1. [エージェント] セクションでエージェントを選択します。
- 2. エージェント内のどのページでも、右上の左矢印アイコンを選択してテストウィンドウを広げる ことができます。
- テストウィンドウの上部にあるドロップダウンメニューを使用して、テストするエイリアスおよび関連するバージョンを選択します。
- 4. メッセージを入力して、タスクを実行するようにエージェントにリクエストします。テストウィンドウはエージェントのデバッグに使用できます。

エージェントをテストする 179

- 5. レスポンスの表示では、以下のような選択を行うことができます。
  - エージェントがソースから情報を取得した場合、レスポンスには脚注が含まれます。脚注を選択すると、レスポンスのその部分の引用文献が表示されます。
  - ボットからのレスポンスの下で [トレースを表示する] を選択すると、エージェントの推論プロセス、アタッチされたナレッジベースやアクショングループの使用状況、使用された推論設定が表示されます。トレースでは、レスポンスがどのように作成されたかが各ステップに分けて表示されます。ステップの横にある矢印を選択すると、そのステップのトレースが展開または折りたたまれます。詳細については、「トレースの有効化」を参照してください。

アクショングループとナレッジベースを有効にしたり無効にしたりすることができます。この機能を使用してエージェントをデバッグし、さまざまな設定での動作を評価することで、更新が必要なアクションやナレッジベースを特定できます。

アクショングループまたはナレッジベースを有効または無効にするには

- 1. [エージェント] セクションでエージェントを選択します。
- 2. [作業中のドラフト] から作業中のドラフトを選択します。
- 3. [アクショングループ] または [ナレッジベース] セクションで、状態を変更したいアクショングループの [状態] にカーソルを合わせます。
- 4. [編集] ボタンが表示されます。それをクリックして、ドロップダウンメニューでアクショングループまたはナレッジベースに対し、[有効] または [無効] を選択します。
- 5. アクショングループを無効にした場合、エージェントはそのアクショングループを実行しなくなります。ナレッジベースを無効にした場合、エージェントはそのナレッジベースをオーケストレーションに使用しなくなります。アクショングループの有効と無効を切り替えたり、テストウィンドウを使用してエージェントをデバッグしたりします。

# トレースの有効化

エージェントをテストすると、トレースが表示されますが、このトレースはエージェントからの各レスポンスに添付されます。トレースにより、エージェントが会話のその時点で返すレスポンスに至るまでの推論プロセスを追跡できます。

コンソールのテストウィンドウでは、トレースはデフォルトで有効になっています。API を使用する場合、enableTrace をtrue に設定しないデフォルトのときは、トレースは無効になります。API

を使用してトレースを有効にする方法については、「 $\underline{\mathbf{T}-\mathcal{Y}}$ ェントを呼び出す」を参照してください。

トレースを使用して、ユーザー入力からレスポンスを返すまでの、エージェントのパスを追跡します。トレースを使用すると、エージェントが呼び出すアクショングループへの入力や、ユーザーへの応答に役立つナレッジベースに加えて、アクショングループとナレッジベースが返す出力に関する情報を見つけることができます。エージェントが実行するアクションやナレッジベースに対して行うクエリを決定するために使用する推論を確認できます。トレースのステップが失敗した場合、トレースは失敗の理由を返します。トレースの詳細情報を使用することで、問題が発生したステップや予期しない動作を引き起こすステップ、および動作を改善できる方法を見つけて、エージェントをデバッグできます。

コンソールのテストウィンドウにトレースを表示すると、推論プロセスの各ステップのトレースを示すウィンドウが表示されます。ステップは、次の4種類のトレースで構成されています。各ステップには、failureTrace、あるいは、rationaleTrace、invocationInputTrace、またはobservationTraceの任意の組み合わせを指定します。

• rationaleTrace - エージェントがアクショングループの実行やナレッジベースからの情報の取得を正当化するために使用する、ユーザー入力に基づく推論を指定します。

```
{
    "traceId": "string",
    "text": "string"
}
```

各フィールドについて以下に説明します。

- traceId トレースのステップの一意な識別子。同じステップにある rationaleTrace、invocationInput、または observationTrace は、どれも同じ traceId を持っています。
- text ユーザー入力に基づく、エージェントの推論または思考プロセス。
- invocationInputTrace 呼び出されるアクショングループまたはナレッジベースに関する情報を指定します。

```
"traceId": "string",
   "invocationType": "ACTION_GROUP | KNOWLEDGE_BASE",
   "actionGroupInvocationInput": {
       "actionGroupName": "string",
       "apiPath": "string",
```

```
"parameters": [
            {
                 "name": "string",
                 "type": "string",
                 "value": "string",
            },
        ],
        "requestBody": {
             "content": {
                 "string": [
                     {
                         "name": "string",
                         "type": "string",
                         "value": "string",
                     },
                ]
            }
        },
        "verb": "string"
    },
    "knowledgeBaseLookupInput": {
        "knowledgeBaseId": "string",
        "text": "string"
    }
}
```

各フィールドについて以下に説明します。

- traceId トレースの一意な識別子。
- invocationType エージェントがアクショングループとナレッジベースのどちらを呼び出すのかを指定します。
- actionGoupInvocationInput 呼び出すアクショングループに関する以下のメタデータが含まれます。
  - actionGroupName エージェントが呼び出すアクショングループの名前。
  - apiPath アクショングループに基づく、呼び出す API へのパス。
  - parameters Lambda 入力イベントのパラメータ。
  - requestBody Lambda 入力イベントのリクエストボディのパラメータ。
  - verb アクショングループに基づく、使用する API メソッド。

• knowledgeBaseLookupInput - ナレッジベースに関する以下の情報と、ナレッジベースの検索クエリを指定します。

- knowledgeBaseId エージェントが検索するナレッジベースの一意の識別子。
- text ナレッジベースへのクエリ。
- observationTrace アクショングループまたはナレッジベースの結果または出力、あるいは ユーザーへのレスポンスを指定します。

```
{
    "traceId": "string",
    "invocationType": "ACTION_GROUP | KNOWLEDGE_BASE | FINISH",
    "actionGroupInvocationOutput": {
        "text": "string"
    },
    "knowledgeBaseLookupOutput": {
        "sourceReferences": {
            "textSourceReferences": [
                     "referenceText": "string",
                     "sourceLocation": {
                         "s3SourceLocation": {
                             "s3Uri": "string"
                         }
                     }
                },
                 . . .
            ]
        }
    "finalResponse": {
        "text": "string"
    }
}
```

各フィールドについて以下に説明します。

- traceId トレースの一意な識別子。
- invocationType エージェントがアクショングループを呼び出すのか、ナレッジベースを呼び出すのか、またはユーザーにレスポンスを返すのかを指定します。
- actionGoupInvocationOutput アクショングループで呼び出す API によって返す JSON 形式の文字列を指定します。

• knowledgeBaseLookupOutput - ナレッジベースのテキストとデータソースの S3 ロケーションを指定します。testSourceReferences のリストの各オブジェクトには、以下のフィールドを指定します。

- referenceText ナレッジベースクエリから返されたナレッジベースのテキスト。
- sourceLocation 返されたテキストがあるデータソースの S3 URI が含まれます。
- failureTrace 発生したエラーを記述する [failureReason] フィールドが 1 つ含まれます。

```
{
    "failureReason": "string"
}
```

# エージェントのデプロイ: バージョニングとエイリアス

作業中のドラフトを十分な回数だけ繰り返し処理し、エージェントの動作に問題がなくなったら、 エージェントのエイリアスを作成し、アプリケーションにデプロイして組み込むように設定すること ができます。

エージェントをデプロイするには、エイリアスを作成します。エイリアスを作成すると、Amazon Bedrock はエージェントのバージョンを自動的に作成します。エイリアスはこの新しく作成された バージョンを指します。必要な場合、以前に作成されたバージョンをエイリアスが指すようにもできます。次に、そのエイリアスへの API コールを行うようにアプリケーションを設定します。

このバージョンは、その作成時に存在していたリソースを保存したスナップショットのようなものです。作業中のドラフトは引き続き変更し、必要に応じてエージェントの新しいエイリアス (ひいてはバージョン)を作成できます。Amazon Bedrock で、エージェントの新しいバージョンを作成します。それには、この新しいバージョンをデフォルトで指すエイリアスを作成します。Amazon Bedrock は、バージョンを作成して、バージョン名として 1 から始まる番号を順に付けます。バージョンは、エージェントの作成当時のスナップショットの役割を果たすため、変更できません。

エイリアスを使用すると、アプリケーションがバージョンを追跡しなくても、エージェントの複数のバージョン間を効率的に切り替えることができます。例えば、発生した変更をすぐに元に戻す必要がある場合は、以前のバージョンのエージェントを指すようにエイリアスを変更できます。

作業中のドラフトのバージョンは DRAFT であり、それを指すエイリアスは AgentTestAlias です。

エージェントのバージョンおよびエイリアスを管理するには、左側のナビゲーションペインで [エージェント] を選択し、[エージェント] セクションでエージェントを選択します。

### 新しいエイリアス (およびオプションで新しいバージョン) を作成するには

1. 右上の [エイリアスを作成] を選択します。または、[デプロイ] タブを選択し、[エイリアス] セクションで [作成] を選択します。

- 2. エイリアスには一意の名前を入力し、必要な場合は説明も入力します。
- 3. 以下のオプションのいずれかを選択します。
  - 新しいバージョンを作成し、それにエイリアスを関連付けます。
  - 既存のバージョンにエイリアスを関連付けます。ドロップダウンメニューから、エイリアス を関連付けたいバージョンを選択します。
- 4. [エイリアスを作成]を選択します。上部に成功を示す緑色のバナーが表示されます。

エージェントのバージョンを管理するには、必ず [構築] タブを開くとともに、その下に [バージョン] セクションを表示します。

### バージョンの詳細を表示するには

- 1. [バージョン] セクションで、表示するバージョンを選択します。
- 2. バージョンのどの部分も変更できませんが、表示したい情報の名前 (モデル、アクショングループ、または Lambda 関数など) を選択して、それらに関する詳細を表示することは可能です。

エージェントのエイリアスを管理するには、[デプロイ] タブを選択すると下に表示される [エイリアス] セクションを使用します。

#### エイリアスを別のバージョンに関連付けるには

- 1. 編集するエイリアスの横にあるラジオボタンを選択します。
- 2. [編集] ボタンを選択します。
- 3. 次のいずれかのオプションを選択します。
  - 新しいバージョンを作成し、それにエイリアスを関連付けます。
  - 既存のバージョンにエイリアスを関連付けます。ドロップダウンメニューから、エイリアス を関連付けたいバージョンを選択します。

# API を使用する場合

Amazon Bedrock サービスのサービスエンドポイントは次のとおりです。AWS のサービスにプログラムで接続するには、エンドポイントを使用します。他の AWS サービスのエンドポイントについては、「AWS 全般のリファレンス」の「 AWS サービスエンドポイント」を参照してください。

次の表は、Amazon Bedrock でサポートされているリージョン固有のエンドポイントの一覧です。

リージョン名	リージョン	エンドポイント	プロトコル
米国東部 (バージ ニア北部)	us-east-1	bedrock-agent-runtime.us-ea st-1.amazonaws.com	HTTPS
米国西部 (オレゴ ン)	us-west-2	bedrock-agent-runtime.us-we st-2.amazonaws.com	HTTPS

現在、Amazon Bedrock で使用できる唯一の API は InvokeAgent です。

## エージェントを呼び出す

エージェントとやり取りするには、InvokeAgent リクエストを送信します。agentAliasId として TSTALIASID を使用して、エージェントのドラフトバージョンを呼び出します。このオペレーションを呼び出すには Amazon Bedrock ランタイムエンドポイント (bedrock-agent-runtime.region.amazonaws.com) を使用する必要があります。

以下にリクエストの例を示します。

```
POST /agents/agentId/agentAliases/agentAliasId/sessions/sessionId/text HTTP/1.1
Content-type: application/json

{
    "inputText": "string",
    "endSession": boolean,
    "enableTrace": boolean,
    "sessionState": {
        "sessionAttributes": {
            "string" : "string"
        }
    }
}
```

API を使用する場合 186

}

リクエストのフィールドは次のとおりです。

• agentId - エージェントの一意な識別子。この ID は、エージェントの詳細ページで確認できます。

- agentAliasId エージェントのエイリアスの一意な識別子。この ID は、エージェントの エイリアスの詳細ページで確認できます。エージェントのドラフトバージョンを呼び出すに は、TSTALIASID を使用します。
- sessionId セッションの一意な識別子。この値を再利用する場合、アイドルセッションタイム アウトに設定した時間が経過していなければ、エージェントとの現在のセッションを継続すること になります。
- inputText エージェントに提供するプロンプト。
- endSession エージェントとのセッションを終了するかどうかを指定します。
- enableTrace レスポンスでトレースを返すかどうかを指定します。デフォルトでは、トレースは無効になっています。詳細については、「トレースの有効化」を参照してください。
- sessionState エージェントとのセッションのセッション属性が含まれます。

このレスポンスでは、以下のオブジェクトが返されます。エラーがある場合、レスポンスは例外を返します。

- chunk [bytes] フィールドには、ユーザーへのレスポンスが base-64 でエンコードされたバイナリオブジェクトとして格納されます。attribution オブジェクトには、レスポンスに付随するcitations のリストが格納されます。
- trace trace オブジェクトとともに agentId、agentAliasId、sessionId が格納されます。トレースオブジェクトのフィールドの詳細については、「トレースの有効化」を参照してください。トレースは、enableTrace がリクエスト内で true として設定されている場合にのみ返されます。

# Bedrock Agent で IAM が機能する仕組み

IAM を使用して Bedrock Agent へのアクセスを管理する前に、Bedrock Agent で使用できる IAM 機能について理解しておく必要があります。

### Bedrock Agent で使用できる IAM の機能

IAM 機能	Bedrock Agent のサポート
アイデンティティベースのポリシー	はい
<u>リソースベースのポリシー</u>	いいえ
<u>ポリシーアクション</u>	はい
ポリシーリソース	はい
ポリシー条件キー	いいえ
ACL	いいえ
ABAC (ポリシー内のタグ)	いいえ
テンポラリ認証情報	はい
<u>プリンシパル許可</u>	はい
<u>サービスロール</u>	はい
サービスリンクロール	いいえ

Bedrock Agent および AWS の他のサービスで大部分の IAM 機能を使用する方法の概要については、「IAM ユーザーガイド」の「IAM と連携する AWS のサービス」を参照してください。

# Bedrock Agent のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート はい

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、アクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「<u>IAM JSON ポリ</u>シーの要素のリファレンス」を参照してください。

Bedrock Agent のアイデンティティベースのポリシー例

Bedrock Agent のアイデンティティベースポリシーの例を確認するには、「Bedrock Agent のアイデンティティベースのポリシー例」を参照してください。

Bedrock Agent 内のリソースベースのポリシー

リソースベースのポリシーのサポート

いいえ

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーが添付されているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービスを含めることができます。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウント にある場合、信頼できるアカウントの IAM 管理者は、リソースにアクセスするための許可をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要もあります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで許可を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、ID ベースのポリシーをさらに付与する必要はありません。詳細については、IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

リソースベースのポリシー 189

# Bedrock Agent のポリシーアクション

ポリシーアクションに対するサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための許可を付与するポリシーで使用 されます。

Bedrock Agent アクションのリストを確認するには、「サービス認可リファレンス」の「<u>Actions</u> Defined by Bedrock Agent」を参照してください。

Bedrock Agent のポリシーアクションでは、アクションの前に、次のプレフィックスを使用します。

bedrock

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
    "bedrock:action1",
    "bedrock:action2"
]
```

Bedrock Agent のアイデンティティベースポリシーの例を確認するには、「Bedrock Agent のアイデンティティベースのポリシー例」を参照してください。

ポリシーアクション 190

# Bedrock Agent 向けのポリシーリソース

ポリシーリソースに対するサポート

はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素は、オブジェクトあるいはアクションが適用されるオブジェクトを 指定します。ステートメントには、Resource または NotResource 要素を含める必要がありま す。ベストプラクティスとして、<u>Amazon リソースネーム (ARN)</u> を使用してリソースを指定しま す。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対 して実行できます。

オペレーションのリスト化など、リソースレベルの許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (\*) を使用します。

"Resource": "\*"

Bedrock Agent のアイデンティティベースポリシーの例を確認するには、「Bedrock Agent のアイデンティティベースのポリシー例」を参照してください。

# Bedrock Agent のポリシー条件キー

サービス固有のポリシー条件キーのサポート いいえ

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの<u>条件演算子</u>を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

ポリシーリソース 191 191

1つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、AWS では AND 論理演算子を使用してそれらを評価します。単一の条件キーに複数の値を指定する場合、AWS では OR 論理演算子を使用して条件を評価します。ステートメントの許可が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー 名がタグ付けされている場合のみリソースにアクセスできる許可を付与することができます。詳細に ついては、IAM ユーザーガイドの「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS はグローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドの「AWS グローバル条件コンテキストキー」を参照してください。

Bedrock Agent のすべてのアクションは、エージェントまたはエイリアスをリソースとして使用する 条件キーをサポートします。

Bedrock Agent のアイデンティティベースポリシーの例を確認するには、「Bedrock Agent のアイデンティティベースのポリシー例」を参照してください。

# Bedrock Agent の ACL

ACL のサポート

いいえ

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

# Bedrock Agent での ABAC

ABAC (ポリシー内のタグ) のサポート

いいえ

属性ベースのアクセス制御 (ABAC) は、属性に基づいて許可を定義する認可戦略です。AWS では、属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール)、および多数の AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初のステップです。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合に操作を許可するように ABAC ポリシーをします。

ACL 192

ABAC は、急速に成長している環境でポリシー管理が面倒な状況に役立ちます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/key-

name、aws:RequestTag/key-name、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素 でタグ情報を提供します。

サービスがすべてのリソースタイプに対して3つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ3つの条件キーすべてをサポートする場合、値は部分的です。

ABAC の詳細については、IAM ユーザーガイドの「<u>ABAC とは?</u>」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「<u>属</u>性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

# Bedrock Agent での一時的な認証情報の使用

一時的な認証情報のサポート

はい

AWS のサービス には、一時認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報を利用できる AWS のサービス を含めた詳細情報については、IAM ユーザーガイドの「IAM と連携する AWS のサービス」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時認証情報を使用していることになります。例えば、会社の Single Sign-On (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時認証情報を作成します。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、IAM ユーザーガイドの「ロールへの切り替え (コンソール)」を参照してください。

一時認証情報は、AWS CLI または AWS API を使用して手動で作成できます。作成後、一時認証情報を使用して AWS にアクセスできるようになります。AWS は、長期的なアクセスキーを使用する代わりに、一時認証情報を動的に生成することをお勧めします。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

# Bedrock Agent のクロスサービスプリンシパル許可

プリンシパル許可のサポート

はい

IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルと見 なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際 に、アクションを実行することで、異なるサービスの別のアクションがトリガーされることがありま す。この場合、両方のアクションを実行するための許可が必要です。ポリシー内で、アクションに 追加の依存アクションが必要かどうかを確認するには、「サービス認可リファレンス」の「Actions、 Resources, and Condition Keys for Bedrock Agent」を参照してください。

# Bedrock Agent のサービスロール

サービスロールに対するサポート

はい

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロール です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細につい ては、「IAM ユーザーガイド」の「AWS のサービス にアクセス許可を委任するロールの作成」を参 照してください。

### Marning

サービスロールのアクセス許可を変更すると、Bedrock Agent の機能が破損する可能性があ ります。Bedrock Agent が指示する場合以外は、サービスロールを編集しないでください。

エージェントリソースを作成する場合、Bedrock Agent が自動的にリソースにアクセスすることを許 可するロールを作成または選択する必要があります。サービスロールあるいはサービスリンクロー ルを以前に作成している場合、Bedrock Agent は選択できるロールのリストを示します。必要なアク ションへのアクセスを許可するロールを選択することが重要です。

# Bedrock Agent のサービスリンクロール

サービスリンクロールのサポート

いいえ

サービスリンクロールは、AWS のサービスにリンクされているサービスロールの一種です。サービ スは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスリン クロールは、AWS アカウント に表示され、サービスによって所有されます。IAM 管理者は、サービ スリンクロールの許可を表示できますが、編集することはできません。

サービスロール

# Bedrock Agent のアイデンティティベースのポリシー例

デフォルトでは、ユーザーとロールには Bedrock Agent リソースを作成または変更するアクセス許可が付与されていません。また、AWS Management Console、AWS Command Line Interface (AWS CLI)、または AWS API を使用してタスクを実行することもできません。IAM 管理者は、リソースで必要なアクションを実行するためのアクセス許可をユーザーに付与する IAM ポリシーを作成できます。その後、管理者がロールに IAM ポリシーを追加すると、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「<u>IAM ポリシーの作成</u>」を参照してください。

Bedrock Agent が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「<u>Actions, Resources, and Condition Keys for</u> Bedrock Agent」を参照してください。

### Note

Bedrock Agent サービスは限定プレビューリリースとして提供されているため、その情報は「サービス認証リファレンス」には記載されていません。

#### トピック

- ポリシーのベストプラクティス
- Bedrock Agent コンソールを使用する場合
- 自分の許可の表示をユーザーに許可する
- ユーザーがエージェントとエイリアスのリソースに対してアクションを実行できるようにする

### ポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウント内で誰かが Bedrock Agent リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、AWS アカウント に追加料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください。

AWS マネージドポリシーを使用して開始し、最小特権の許可に移行する – ユーザーとワークロードへの許可の付与を開始するには、多くの一般的なユースケースのために許可を付与する AWS マ

ネージドポリシーを使用します。これらは AWS アカウントで使用できます。ユースケースに応じた AWS カスタマーマネージドポリシーを定義することで、許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「AWS ジョブ機能の管理ポリシー」を参照してください。

- 最小特権を適用する IAM ポリシーで許可を設定するときは、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「IAM でのポリシーとアクセス許可」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定することができます。また、AWS のサービス などの特定の AWS CloudFormation を介して使用する場合、条件を使用してサービスアクションへのアクセスを許可することもできます。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素:条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な許可を確保する IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「<u>IAM Access Analyzer ポリシーの検</u>証」を参照してください。
- 多要素認証 (MFA) を要求する AWS アカウント で IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA 保護 API アクセスの設定」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「<u>IAM でのセキュリ</u> ティのベストプラクティス」を参照してください。

# Bedrock Agent コンソールを使用する場合

Bedrock Agent コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。アクセス許可により、AWS アカウント の Bedrock Agent リソースの詳細をリストおよび表示できます。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWSAPI のみを呼び出すユーザーには、最小限のコンソール許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

Bedrock Agent コンソールへのアクセスを許可するには、アクセスが必要なロールまたはエンティティに次のポリシーを付与します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BedrockConsole",
            "Effect": "Allow",
            "Action": [
               "bedrock:CreateAgent",
               "bedrock:UpdateAgent",
               "bedrock:GetAgent",
               "bedrock:ListAgents",
               "bedrock:CreateActionGroup",
               "bedrock: UpdateActionGroup",
               "bedrock:GetActionGroup",
               "bedrock:ListActionGroups",
               "bedrock:CreateAgentDraftSnapshot",
               "bedrock:GetAgentVersion",
               "bedrock:ListAgentVersions",
               "bedrock:CreateAgentAlias",
               "bedrock:UpdateAgentAlias",
               "bedrock:GetAgentAlias,
               "bedrock:ListAgentAliases"
               "bedrock:InvokeAgent"
            ],
            "Resource": "*"
        }
    ]
```

詳細については、IAM ユーザーガイドの「ユーザーへの許可の追加」を参照してください。

自分の許可の表示をユーザーに許可する

この例では、ユーザーアイデンティティに添付されたインラインおよびマネージドポリシーの表示をIAM ユーザーに許可するポリシーを作成する方法を示します。このポリシーには、コンソールで、

または AWS CLI か AWS API を使用してプログラム的に、このアクションを完了するアクセス許可が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

ユーザーがエージェントとエイリアスのリソースに対してアクションを実行できるようにする

エージェントとエイリアスのリソースに対してアクションを実行するアクセス許可をアイデンティティにプロビジョニングできます。これらのリソースの ARN の形式は以下のとおりです。

- エージェント arn:aws:bedrock:region:account-id:agent/AGENTID
- エイリアス arn:aws:bedrock: region:account-id:agent-alias/AGENTID/ALIASID

ロールは、リソースに対して API オペレーションを呼び出すことができます。例えば、InvokeAgent オペレーションはエイリアスリソースに対してのみ使用でき、UpdateAgent オペレーションはエージェントリソースに対してのみ使用できます。ポリシーで指定されたリソースに対しては使用できないオペレーションをポリシー内で指定すると、Bedrock Agent はエラーを返します。オペレーションのリストと、それらのオペレーションの実行対象にできるリソースのリストについては、次の表を参照してください。リソースに対しては、CreateAgent と ListAgents は実行できません。

当なし
ージェント
ージェント
当なし
ージェント
イリアス
イリアス

オペレーション	リソース
ListAgentAliases	エージェント
InvokeAgent	エイリアス

以下のサンプルポリシーを IAM ロールに付与すると、IAM ロールは、エージェントに関する情報の取得、エージェントエイリアスの更新、エージェントの操作を行う Bedrock Agent API オペレーションを呼び出すことができるようになります。 *sid* は任意のポリシー識別子に置き換え、*account-id* はエージェントが属するアカウント ID に置き換えます。また、AGENTID はエージェントの ID に置き換え、ALIASID はエージェントのエイリアスに置き換えます。エージェント ID とエイリアス ID を確認するには、ListAgents API オペレーションを使用するか、コンソールでエージェントの詳細ページとエイリアスの詳細ページを参照します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "sid",
            "Effect": "Allow",
            "Action": "bedrock:GetAgent",
            "Resource": "arn:aws:bedrock:region:account-id:agent/AGENTID"
        },
        {
            "Sid": "sid",
            "Effect": "Allow",
            "Action": Γ
                "bedrock:UpdateAgentAlias",
                "bedrock: InvokeAgent"
            ],
            "Resource": [
                "arn:aws:bedrock:region:account-id:agent-alias/AGENTID/ALIASID"
            1
        },
    ]
}
```

# リソースのタグ付け

Amazon Bedrock リソースを管理しやすくするために、各リソースにメタデータをタグとして割り当てることができます。タグとは、AWS リソースに割り当てるラベルです。各タグは、キーと値から構成されます。

タグ付けできる Amazon Bedrock リソースは以下のとおりです。

- カスタムモデル
- モデルカスタマイズジョブ
- プロビジョンドモデル

タグを使用すると、AWS リソースを目的、所有者、アプリケーションなどさまざまな方法で分類することができます。タグは、以下のことに役立ちます。

- AWS リソースの特定と整理。多くの AWS のリソースではタグ付けがサポートされるため、さまざまなサービスのリソースに同じタグを割り当てて、リソースの関連を示すことができます。
- コストの割り当て。タグは、AWS Billing and Cost Management ダッシュボードでアクティベート します。AWS では、タグを使用してコストを分類し、毎月のコスト割り当てレポートを設定する ことができます。詳細については、「AWS Billing and Cost Management ユーザーガイド」の「ユスト配分タグを使用する」を参照してください。
- リソースへのアクセス制御。Amazon Bedrock でタグを使用して、Amazon Bedrock リソースへの アクセスを制御するポリシーを作成することができます。これらのポリシーを IAM ロールまたは ユーザーにアタッチして、タグベースのアクセスコントロールを有効にできます。

#### トピック

- コンソールを使用する
- <u>API を使用する</u>
- タグの制限

# コンソールを使用する

タグ付けはカスタムモデルと微調整ジョブでサポートされています。タグの追加および変更は、微調整プロセスの任意の時点で可能です。

コンソールを使用する 201

### 新しい微調整ジョブを送信する際にタグを追加するには:

- 1. 左側のサイドバーから [微調整] を選択します。
- 2. [モデル] セクションで [モデルを微調整] を選択します。
- 3. [モデルの設定] または [ジョブの設定] セクションにある [タグ] セクションを展開します。
- 4. [新しいタグを追加] を選択し、[キー] と [値] にタグのキーと値を入力します。追加できるタグの最大総数は 50 個です。
- 5. タグを削除するには、キー値ペアの横にある [タグを削除] を選択します。

### 既存のカスタムモデルまたは微調整ジョブにタグを追加するには:

- 1. 左側のサイドバーから [微調整] を選択します。
- 2. [モデル] または [トレーニングジョブ] セクションで、タグを追加するモデルまたはジョブを選択 します。
- 3. 下部の [タグ] セクションで、[タグを管理] を選択します。
- 4. [新しいタグを追加] を選択し、[キー] と [値] にタグのキーと値を入力します。追加できるタグの 最大総数は 50 個です。
- 5. タグを削除するには、キー値ペアの横にある [タグを削除] を選択します。

# API を使用する

タグ付けオペレーションを実行するには、タグ付けオペレーションを実行するリソースのAmazon リソースネーム (ARN) が必要です。リソースのタグ付けおよびタグ付け解除を行うには、<u>TagResource</u> オペレーションと <u>UnTagResource</u> オペレーションを使用します。リソースのタグをリスト表示するには、<u>ListTagsForResource</u> オペレーションを使用します。タグ付けの例については、「リソースのタグ付け」を参照してください。

# タグの制限

Amazon Bedrock リソースのタグには、以下の基本的な制限が適用されます。

- リソースあたりのタグの最大数 50。
- キーの最大数 50。
- キーの最大長 128 文字。
- 値の最大長 256 文字。

API を使用する 202

• キーと値の有効な文字 - a ~ z、A ~ Z、0 ~ 9、スペース、および特殊文字 (\_ .:/=+-@)

- キーと値は大文字と小文字が区別されます。
- aws:をキーのプレフィックスとして使用しないでください。これは、AWS 用に予約されています。

タグの制限 203

# Amazon Bedrock のセキュリティ

AWS では、クラウドセキュリティを最優先事項としています。AWS のユーザーは、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを利用できます。

セキュリティは、AWS とユーザーの間の責任共有です。<u>責任共有モデル</u>では、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ AWS は、AWS クラウドで AWS のサービスを実行するインフラストラクチャを保護する責任を担います。また、AWS は、ユーザーが安全に使用できるサービスも提供します。AWS コンプライアンスプログラム の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon Bedrock に適用するコンプライアンスプログラムの詳細については、「コンプライアンスプログラムによる対象範囲内の AWS のサービス」を参照してください。
- クラウド内のセキュリティ ユーザーの責任は、使用する AWS のサービスに応じて異なります。
   またお客様は、データの機密性、企業要件、適用法令と規制などのその他の要因に対しても責任を担います。

このドキュメントは、Amazon Bedrock の使用時に責任共有モデルがどのように適用されるかを理解するために役立ちます。以下のトピックでは、セキュリティおよびコンプライアンス上の目的を達成するように Amazon Bedrock ・リソースのモニタリングとセキュア化に役立つ AWS のその他のサービスを使用する方法も学びます。

### トピック

- Amazon Bedrock でのデータ保護
- Amazon Bedrock のためのアイデンティティとアクセス管理
- Amazon Bedrock のコンプライアンス検証
- Amazon Bedrock でのインシデントへの対応
- Amazon Bedrock の耐障害性
- Amazon Bedrock でのインフラストラクチャセキュリティ
- サービス間の混乱した代理の防止
- Amazon Bedrock での設定と脆弱性の分析

# Amazon Bedrock でのデータ保護

Amazon Bedrock ではデータ保護のために、AWS の
責任共有モデルが適用されています。このモデルで説明されているように、AWS は、AWS クラウドのすべてを実行するグローバルインフラストラクチャを保護する責任を負います。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。このコンテンツには、使用される AWS のサービスのセキュリティ構成と管理タスクが含まれます。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることを お勧めします。この方法により、それぞれのジョブを遂行するために必要なアクセス許可のみを各 ユーザーに付与できます。また、次の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 および TLS 1.3 をお勧めします。
- AWS CloudTrail で API とユーザーアクティビティログをセットアップします。
- AWS のサービス内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、連邦情報処理規格 (FIPS) 140-2 を参照してください。

お客様のEメールアドレスなどの機密情報やセンシティブ情報は、タグや [Name] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これには、コンソール、API、AWS CLI、または AWS SDK を使用して、Amazon Bedrock または他の AWS のサービス で作業する場合も含まれます。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断口グに使用される場合があります。外部サーバーへ URL を供給する場合は、そのサーバーへのリクエストを検証するために、認証情報を URL に含めないことを強くお勧めします。

Amazon Bedrock は、ユーザーのプロンプトやその続きの情報を、AWS モデルのトレーニングに使用したり、第三者に配布したりすることはありません。トレーニングデータは Amazon Titan の基盤

データ保護 205

モデルのトレーニングに使用されたり、第三者に配布されたりすることはありません。使用状況のタイムスタンプや記録されたアカウント ID など、サービスによって記録されたその他の使用状況データ・情報も、モデルのトレーニングには使用されません。

Amazon Bedrock は、お客様から提供された微調整データを Amazon Titan モデルの微調整にのみ使用します。Amazon Bedrock は、基盤モデルのトレーニングなど、他の目的で微調整データを使用することはありません。

各モデルプロバイダーには、モデルのアップロード先となるエスクローアカウントがあります。Amazon Bedrock 推論アカウントにはこれらのモデルを呼び出すことができるアクセス許可がありますが、エスクローアカウント自体には Amazon Bedrock アカウントへのアウトバウンドアクセス許可はありません。さらに、モデルプロバイダーは Amazon Bedrock のログにアクセスしたり、お客様のプロンプトやその続きの情報にアクセスしたりすることはできません。

Amazon Bedrock はお客様のデータを保存したりサービスログに記録したりすることはありません。

Bedrock は <u>CreateModelCustomizationJob</u> アクションまたは<u>コンソール</u>でトレーニングデータを使用して、Amazon Bedrock の基盤モデルを微調整したカスタムモデルを作成します。カスタムモデルは AWS によって管理および保管されます。デフォルトでは、カスタムモデルは AWS が所有する AWS Key Management Service キーで暗号化されますが、独自の AWS KMS キーを使用して暗号化することもできます。カスタムモデルを暗号化するタイミングは、コンソールを使用して微調整ジョブを送信するときか、CreateModelCustomizationJob アクションを使用してプログラムで微調整ジョブを送信するときです。

微調整ジョブが完成すると、微調整のために提供したトレーニングデータや検証データは Amazon Bedrock アカウントには保存されなくなります。トレーニング中、データは AWS Service Management Connector インスタンスのメモリに存在しますが、これらのマシンでは、インスタンス自体のハードウェアモジュールに実装されている XTS-AES-256 暗号を使用して暗号化されています。

モデルが機密データに基づいて推論レスポンスを生成する可能性があるため、機密データを使用してカスタムモデルをトレーニングすることはお勧めしません。機密データを使用してカスタムモデルをトレーニングしている場合、そのデータに基づくレスポンスが生成されないようにする唯一の方法は、カスタムモデルを削除し、トレーニングデータセットから機密データを削除して、カスタムモデルを再トレーニングすることです。

カスタムモデルのメタデータ (名前と Amazon リソースネーム) とプロビジョンドモデルのメタデータは、Amazon Bedrock サービスが所有するキーで暗号化された Amazon DynamoDB テーブルに保存されます。

データ保護 206

### トピック

- データ暗号化
- VPC を使用してジョブを保護する

### データ暗号化

Amazon Bedrock は、暗号化を使用して保管中のデータと転送中のデータを保護します。

### トピック

- ・ 転送中の暗号化
- 保管中の暗号化
- キーの管理
- モデルカスタマイズジョブの暗号化

### 転送中の暗号化

AWS では、転送中のネットワーク間データはすべて、TLS 1.2 暗号化をサポートしています。

Amazon Bedrock API とコンソールに対するリクエストには、安全な SSL 接続が使用されます。トレーニングとデプロイのために、ユーザーに代わってリソースにアクセスするアクセス許可を付与するには、Amazon Bedrock に AWS Identity and Access Management (IAM) ロールを渡します。

### 保管中の暗号化

Amazon Bedrock は保管中の モデルカスタマイズジョブの暗号化 を提供します。

# キーの管理

AWS Key Management Service (KMS) を使用して、モデルカスタマイズジョブで使用するキーを管理できます。詳細については、「AWS Key Management Service の概念」を参照してください。

# モデルカスタマイズジョブの暗号化

Amazon Bedrock はモデルカスタマイズジョブのモデルアーティファクトを暗号化します。デフォルトでは、Amazon Bedrock はこのデータを暗号化するのに AWS マネージドキーを使用します。オプションで、カスタマーマネージドキーを使用して、モデルアーティファクトを暗号化することもできます。

データ暗号化 207 207

AWS KMS keys についての詳細については、「AWS Key Management Service デベロッパーガイド」の「カスタマーマネージドキー」を参照してください。

#### トピック

- モデルカスタマイズ入力
- モデルカスタマイズ入力
- カスタマーマネージドキーを作成する
- カスタマーマネージドキーを使用してカスタマイズジョブを実行する
- 推論でカスタマーマネージドキーを使用する

#### モデルカスタマイズ入力

Amazon Bedrock を使用してモデルカスタマイズジョブを実行する場合、入力ドキュメント (トレーニング/検証データ) を Amazon S3 バケットに保存します。これらのドキュメントを保存中に暗号化するには、Amazon S3 SSE-S3 サーバーサイド暗号化オプションを使用できます。このオプションでは、オブジェクトは Amazon S3 サービスによって管理されるサービスキーで暗号化されます。

詳細については、Amazon Simple Storage Service ユーザーガイドの「<u>Amazon S3 が管理する暗号</u> 化キーによるサーバー側の暗号化 (SSE-S3) を使用したデータの保護」を参照してください。

#### モデルカスタマイズ入力

Amazon Bedrock がモデルカスタマイズジョブを完成すると、ジョブの作成時に指定した Amazon S3 の場所にジョブメトリクスが保存されます。メトリクスを暗号化するには、直前のセクションで説明した Amazon S3 SSE-S3 サーバーサイド暗号化オプションを使用できます。

Amazon Bedrock は、AWS で管理される Amazon S3 バケットにカスタムモデルアーティファクトを保存します。デフォルトでは、Amazon Bedrock はこのデータを暗号化するのに AWS マネージドキー を使用します。このタイプの KMS キーは AWS によって作成されるため、自分で管理する必要はありません。このキーは AWS によって管理され、自動的に使用されます。

オプションで、カスタマーマネージドキーを使用して、モデルアーティファクトを暗号化することもできます。このキーは AWS アカウント で作成、所有、管理する KMS キーです。

独自の KMS キーを使用できるようにするには、以降のセクションで説明するとおりにポリシーとアクセス許可を設定します。

データ暗号化 208 208

#### カスタマーマネージドキーを作成する

CreateKey へのアクセス許可を持つすべてのユーザーは、AWS Key Management Service (AWS KMS) コンソールまたは <u>CreateKey</u> API オペレーションを使用してカスタマーマネージドキーを作成できます。必ず対称暗号化キーを作成してください。

キーポリシーを作成し、以下のポリシーステートメントを追加して、カスタムモデルビルダーとユーザーにアクセス許可を付与します。

```
{
    "Version": "2012-10-17",
    "Id": "KMS Key Policy",
    "Statement": [
        {
            "Sid": "Permissions for custom model users",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/CustomModelCaller"
            },
            "Action": "kms:Decrypt",
            "Resource": "*"
        },
            "Sid": "Permissions for custom model builders",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/CustomModelBuilder"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey",
                "kms:CreateGrant"
            ],
            "Resource": "*"
        }
}
```

カスタマーマネージドキーを使用してカスタマイズジョブを実行する

ユーザーがモデルカスタマイズジョブを作成するには、そのユーザーまたはロールに、カスタマーマネージドキーを使用する以下のアクセス許可が必要です。

データ暗号化 209 209

推論でカスタマーマネージドキーを使用する

ユーザーがカスタムモデルで推論を実行するには、そのユーザーまたはロールに、カスタマーマネージドキーを使用する以下のアクセス許可が必要です。

モデルを呼び出して推論を実行するときに (カスタマーマネージドキーで暗号化されたカスタムモデルの場合)、そのキーに対する kms:Decrypt アクセス許可がないと、そのリクエストは失敗し、次のエラーメッセージが表示されます。

You don't have sufficient access to the model's KMS key. Ensure that kms:Decrypt permissions are correctly configured

データ暗号化 210 210

# VPC を使用してジョブを保護する

モデルカスタマイズジョブを実行すると、ジョブは Amazon S3 バケットにアクセスすることで、入力データをダウンロードしてジョブメトリクスをアップロードします。

データへのアクセスを制御するため、仮想プライベートクラウド (VPC) を作成することをお勧めします。トレーニングデータがインターネット経由でアクセスされないように VPC を設定してください。VPC の作成と設定の詳細については、「Amazon VPC ユーザーガイド」の「Amazon VPC の開始方法」を参照してください。

VPC を使うと、VPC フローログを使って AWS ジョブコンテナに出入りするすべてのネットワークトラフィックを監視することもできます。詳細については、「Amazon VPC ユーザーガイド」の「VPC フローログを使用した IP トラフィックのログ記録」を参照してください。

インターネット接続なしで VPC を設定する場合、カスタマイズジョブが S3 バケットにアクセスで きるようにするための VPC エンドポイントを作成する必要があります。

VPC を使用するようにカスタマイズジョブを設定すると、そのジョブは、VPC エンドポイントを使用して S3 バケットにアクセスする Elastic Network Interface (ENI) を作成します。ENI については、「Amazon VPC ユーザーガイド」の「Elastic Network Interfaces」を参照してください。

#### トピック

- Amazon Bedrock 用に VPC を設定する
- VPC を使用するようにモデルカスタマイズジョブを設定する

# Amazon Bedrock 用に VPC を設定する

Amazon Bedrock のモデルカスタマイズジョブ用に VPC を設定するには、次のガイドラインに従います。VPC のセットアップについては、Amazon VPC ユーザーガイドの「<u>VPC とサブネットの使</u>用」を参照してください。

#### トピック

- Amazon S3 VPC エンドポイントを作成する
- カスタムエンドポイントポリシーを使って Amazon S3 へのアクセスを制限する
- カスタマイズジョブロールの VPC アクセス許可
- ルートテーブルを設定する

#### Amazon S3 VPC エンドポイントを作成する

インターネット接続がない状態で VPC を設定する場合は、VPC エンドポイントを作成する必要があります。エンドポイントにより、トレーニングデータと、ジョブによって保存されたトレーニング損失メトリクスデータを含む S3 バケットにアクセスできます。

VPC エンドポイントを作成することで、データとモデルのアーティファクトを保存するバケットにモデルカスタマイズジョブがアクセスできるようにします。

プライベート VPC からのリクエストのみに S3 バケットへのアクセスを許可するカスタムポリシーも作成することをお勧めします。詳細については、「 $\frac{1}{1}$  VPC エンドポイント (AWS PrivateLink) を使用する」を参照してください。

#### Amazon S3 VPC エンドポイントを作成するには

- 1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
- 2. ナビゲーションペインで [エンドポイント] を選択し、[エンドポイントの作成] を選択します。
- 3. [サービス名] で、「com.amazonaws.*region*.s3」を検索します。*region* を VPC がある [リージョン] の名前に置き換えます。
- 4. [ゲートウェイタイプ] を選択します。
- 5. [VPC]] で、エンドポイントに使用する VPC を選択します。
- 6. [Configure route tables] で、エンドポイントで使用するルートテーブルを選択します。Amazon VPC で、選択した各ルートテーブルに、S3 トラフィックを新しいエンドポイントに向けるルートが自動的に追加されます。
- 7. [ポリシー] で、[フルアクセス] を選択して、VPC 内の任意のユーザーまたはサービスによる Amazon S3 サービスへのフルアクセスを許可します。アクセスを詳細に制限するには、[カスタム] を選択します。詳細については、<u>カスタムエンドポイントポリシーを使って Amazon S3 へのアクセスを制限するを参照してください。</u>

カスタムエンドポイントポリシーを使って Amazon S3 へのアクセスを制限する

デフォルトのエンドポイントポリシーでは、VPC のすべてのユーザーまたはサービスに対して Amazon S3 へのフルアクセスが許可されています。Amazon S3 へのアクセスを詳細に制限するに は、カスタムエンドポイントポリシーを作成します。詳細については、「Amazon S3 のエンドポイ ントポリシー」を参照してください。バケットポリシーを使用して、S3 バケットへのアクセスを VPC からのトラフィックのみに制限することもできます。

次のポリシーでは、S3 バケットへのアクセスを許可します。このポリシーを編集して、ジョブに必要なリソースのみへのアクセスを許可します。

```
{
   "Version": "2012-10-17",
   "Statement": [
        "Sid": "Allow access to output location",
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::my_output_bucket/myfolder"
        ]
      },
        "Sid": "Allow access to input data",
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:PutObject",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::my_training_data_bucket/myfolder",
            "arn:aws:s3:::my_validation_data_bucket/myfolder"
        ]
   ]
}
```

詳細については、「Amazon S3 バケットポリシー」を参照してください。

カスタマイズジョブロールの VPC アクセス許可

モデルカスタマイズジョブで VPC を使用する場合、CreateModelCustomizationJob リクエストに指定するデータアクセスロールには次のアクセス許可が含まれている必要があります。このポリシーを編集して、ジョブに必要な VPC リソースのみへのアクセスを許可します。

```
{
```

```
"Effect": "Allow",
        "Action": [
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeVpcs",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
        "Resource": "*"
    }, {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
       "Resource":[
           "arn:aws:ec2:{region}:{AccountId}:network-interface/*"
       ],
       "Condition": {
           "StringEquals": {
               "aws:RequestTag/BedrockManaged": ["true"]
            },
            "ArnEquals": {
               "aws:RequestTag/BedrockModelCustomizationJobArn": ["arn:aws:bedrock:
{region}:{AccountId}:model-customization-job/*"]
       }
    }, {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
       "Resource":[
           "arn:aws:ec2:{region}:{AccountId}:subnet/{subnet-Id}",
           "arn:aws:ec2:{region}:{AccountId}:subnet/{subnet-Id2}",
           "arn:aws:ec2:{region}:{AccountId}:security-group/{sg-Id}",
           "arn:aws:ec2:{region}:{AccountId}:security-group/{sq-Id2}"
       ]
    }, {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteNetworkInterfacePermission",
        ]
```

```
"Resource": "*",
        "Condition": {
           "ArnEquals": {
               "ec2:Subnet": [
                   "arn:aws:ec2:{region}:{AccountId}:subnet/{subnet-Id}",
                   "arn:aws:ec2:{region}:{AccountId}:subnet/{subnet-Id2}"
               ],
               "ec2:ResourceTag/BedrockModelCustomizationJobArn": ["arn:aws:bedrock:
{region}:{AccountId}:model-customization-job/*"]
           },
           "StringEquals": {
               "ec2:ResourceTag/BedrockManaged": "true",
        }
    }, {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:{region}:{AccountId}:network-interface/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction": [
                    "CreateNetworkInterface"
                1
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "BedrockManaged",
                    "BedrockModelCustomizationJobArn"
                ]
            }
      }
    }
```

モデルカスタマイズジョブのデータアクセスロールの設定に関する詳細については、「<u>モデルカスタ</u>マイズの IAM ロールを設定する」を参照してください。

### ルートテーブルを設定する

エンドポイントルートテーブルのデフォルトの DNS 設定を使って、標準 Amazon S3 URL (例えば、http://s3-aws-region.amazonaws.com/MyBucket) が解決されるようにします。デフォルトの DNS 設定を使用しない場合は、トレーニングジョブデータを保存する場所の URL が解決さ

れるようにします。これを行うには、エンドポイントルートテーブルを設定します。VPC エンドポイントルートテーブルについては、Amazon VPC ユーザーガイドの「<u>ゲートウェイエンドポイント</u>のルーティング」を参照してください。

### VPC を使用するようにモデルカスタマイズジョブを設定する

これまでのセクションの手順に従って VPC および必要なロールとアクセス許可を設定し終わったら、この VPC を使用するモデルカスタマイズジョブを作成することができます。

ジョブの VPN サブネットとセキュリティグループを指定すると、Amazon Bedrock はサブネットの 1 つのセキュリティ グループに関連付けられた Elastic Network Interface (ENI) を作成します。ENI により、Amazon Bedrock ジョブは VPC 内のリソースに接続できます。ENI については、「Amazon VPC ユーザーガイド」の「Elastic Network Interfaces」を参照してください。Amazon Bedrock は、作成した ENI に BedrockManaged および BedrockModelCusomizationJobArn タグを付けます。

アベイラビリティーゾーンごとに少なくとも1つのサブネットを指定することをお勧めします。

セキュリティグループを使用すると、VPC リソースへの Amazon Bedrock のアクセスを制御するためのルールを設定できます。

#### API を使用する

Amazon Bedrock API で、VpcConfig リクエストパラメータに VPC サブネットとセキュリティグループを指定します。次に、API コールに含める VpcConfig パラメータの例を示します。

```
"VpcConfig": {
"SecurityGroupIds": [
    "sg-0123456789abcdef0"
    ],
    "Subnets": [
        "subnet-0123456789abcdef0",
        "subnet-0123456789abcdef1",
        "subnet-0123456789abcdef2"
    ]
}
```

API を使用してジョブを設定する方法の詳細については、「<u>Submit a job (API)</u>」を参照してください。

#### コンソールを使用する

Amazon Bedrock コンソールでは、モデルカスタマイズジョブを作成するときに、オプションの [VPC の設定] セクションで VPC サブネットとセキュリティグループを指定します。コンソールを使用してジョブを設定する方法の詳細については、「Submit a job (console)」を参照してください。

#### Note

VPC 設定を含むジョブの場合、コンソールはそのジョブの新しいサービスロールを作成できません。次の例を使用してサービスロールを作成します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "bedrock.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnEquals": {
                     "aws:SourceArn": "arn:aws:bedrock:us-east-1:111122223333:model-
customization-job/*"
            }
        }
    ]
}
```

# Amazon Bedrock のためのアイデンティティとアクセス管理

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御するために役立つ AWS のサービス です。IAM 管理者は、誰を認証 (サインインを許可) し、誰に Amazon Bedrockリソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加費用なしで使用できる AWS のサービス です。

#### トピック

- 対象者
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon Bedrock で IAM が機能する仕組み
- Amazon Bedrock のアイデンティティベースのポリシー例
- サービスロール
- Amazon Bedrock のアイデンティティとアクセスに関するトラブルシューティング

# 対象者

AWS Identity and Access Management (IAM) の使用方法は、Amazon Bedrock で行う作業に応じて異なります。

サービスユーザー - Amazon Bedrock サービスを使用してジョブを行う場合には、管理者から必要な認証情報と許可が提供されます。業務のために使用する Amazon Bedrock 機能が増えるにつれて、追加の許可が必要になる可能性があります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。Amazon Bedrock の機能にアクセスできない場合は、「Amazon Bedrock のアイデンティティとアクセスに関するトラブルシューティング」を参照してください。

サービス管理者 - 社内の Amazon Bedrock リソースを担当している場合は、通常、Amazon Bedrock へのフルアクセスがあります。サービスのユーザーがどの Amazon Bedrock 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの許可を変更する必要があります。このページの情報を確認して、IAM の基本概念を理解してください。会社で Amazon Bedrock と IAM を併用する方法の詳細については、「Amazon Bedrockで IAM が機能する仕組み」を参照してください。

IAM 管理者 – IAM 管理者は、通常、Amazon Bedrock へのアクセスを管理するポリシーの作成方法 について詳細情報が必要になります。IAM で使用可能な、Amazon Bedrock アイデンティティベース のポリシーの例を確認するには、「 $\underline{\text{Amazon Bedrock } \text{or} \text{$ 

対象者 218

# アイデンティティを使用した認証

認証とは、アイデンティティ認証情報を使用して AWS にサインインする方法です。ユーザーは、AWS アカウントのルートユーザー として、または IAM ロールを引き受けることによって、認証済み (AWS にサインイン済み) である必要があります。

ID ソースから提供された認証情報を使用して、フェデレーティッドアイデンティティとして AWS にサインインできます。 AWS IAM Identity Centerフェデレーティッドアイデンティティの例としては、(IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報などがあります。フェデレーティッドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーションを使用して AWS にアクセスする場合、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサインインできます。AWS へのサインインの詳細については、AWS サインイン ユーザーガイドの「<u>AWS ア</u>カウント にサインインする方法」を参照してください。

プログラムを使用して AWS にアクセスする場合、AWS は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) を提供し、認証情報を使用してリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。リクエストに署名する推奨方法の使用については、「IAM ユーザーガイド」の「AWS API リクエストの署名」を参照してください。

使用する認証方法を問わず、セキュリティ情報の提供を追加でリクエストされる場合もあります。例えば、AWS は、アカウントのセキュリティを強化するために多要素認証 (MFA) を使用することをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「<u>Multi-factor authentication</u>」(多要素認証) および「IAM ユーザーガイド」の「<u>AWS での多要素認証 (MFA) の使</u>用」を参照してください。

# AWS アカウント ルートユーザー

AWS アカウント を作成する場合は、このアカウントのすべての AWS のサービス とリソースに対して完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることによってアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報を保護し、それらを使用してルートユーザーのみが実行できるタスクを実行します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

#### フェデレーティッドID

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに対し、ID プロバイダーとのフェデレーションを使用して、一時的な認証情報の使用により、AWS のサービス にアクセスすることを要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、Identity Center ディレクトリのユーザーか、または ID ソースから提供された認証情報を使用して AWS のサービス にアクセスするユーザーです。フェデレーティッド ID が AWS アカウント にアクセスすると、ロールが継承され、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Center を使用することをお勧めします。IAM Identity Center でユーザーとグループを作成するか、すべての AWS アカウント とアプリケーションで使用するために、独自の ID ソースで一連のユーザーとグループに接続して同期することもできます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

#### IAM ユーザーとグループ

IAM ユーザーは、1 人のユーザーまたは 1 つのアプリケーションに対して特定の許可を持つ AWS アカウント 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーとの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に許可を指定できます。多数のユーザーグループがある場合、グループを使用することで許可の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは1人の人または1つのアプリケーションに一意に 関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユー ザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳 細については、「IAM ユーザーガイド」の「<u>IAM ユーザー (ロールではなく) の作成が適している場</u> 合」を参照してください。

#### IAM ロール

IAM ロールは、特定の許可を持つ、AWS アカウント 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロールを切り替えることによって、AWS Management Console で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、AWS CLI または AWS API オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます。

- ・フェデレーティッドユーザーアクセス フェデレーティッドアイデンティティに許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッドアイデンティティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「Creating a role for a third-party Identity Provider」(サードパーティーアイデンティティプロバイダー向けロールの作成)を参照してください。IAM アイデンティティセンターを使用する場合、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、アクセス許可セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。
- 一時的な IAM ユーザー許可 IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる 許可を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の AWS のサービス では、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス 一部の AWS のサービス では、他の AWS のサービス の機能を使用します。例えば、サービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - プリンシパル許可 IAM ユーザーまたはロールを使用して AWS でアクションを実行する場合、 そのユーザーはプリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、異なるサービスの別の

アクションがトリガーされることがあります。この場合、両方のアクションを実行するための許可が必要です。ポリシー内で、アクションに追加の依存アクションが必要かどうかを確認するには、「<u>サービス認証リファレンス</u>」の「Amazon Bedrock のアクション、リソース、条件キー」を参照してください。

- サービスロール サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービス にアクセス許可を委任するロールの作成」を参照してください。
- サービスリンクロール サービスリンクロールは、AWS のサービス にリンクされたサービス ロールの一種です。サービスは、ユーザーに代わってアクションを実行するロールを引き受ける ことができます。サービスリンクロールは、AWS アカウント に表示され、サービスによって所 有されます。IAM 管理者は、サービスリンクロールの許可を表示できますが、編集することは できません。
- Amazon EC2 で実行されているアプリケーション EC2 インスタンスで実行され、AWS CLI または AWS API 要求を行っているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得することができます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、「IAM ユーザーガイド」の「<u>(IAM)</u> ユーザーではなく) IAM ロールをいつ作成したら良いのか?」を参照してください。

# ポリシーを使用したアクセスの管理

AWS でアクセスをコントロールするには、ポリシーを作成して AWS アイデンティティまたはリソースにアタッチします。ポリシーは AWS のオブジェクトであり、アイデンティティやリソースに関連付けて、これらのアクセス許可を定義します。AWS は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うと、これらのポリシーを評価します。ポリシーでの許可により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWS に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は、リソースで必要なアクションを実行するためのアクセス許可をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、iam: GetRole アクションを許可するポリシーがあるとします。このポリシーがあるユーザーは、AWS Management Console、AWS CLI、または AWS API からロールの情報を取得できます。

# アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれます。マネージドポリシーは、AWS アカウント 内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。マネージドポリシーには、AWS マネージドポリシーとカスタマーマネージドポリシーがあります。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「マネージドポリシーとインラインポリシーの比較」を参照してください。

# リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーが添付されているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービスを含めることができます。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは IAM の AWS マネージドポリシーは使用できません。

# アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Simple Storage Service (Amazon S3)、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

#### その他のポリシータイプ

AWS では、その他の一般的ではないポリシータイプもサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の許可を設定できます。

- アクセス許可の境界 アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる許可の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られるアクセス許可は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、許可は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCP) SCP は、AWS Organizations で組織や組織単位 (OU) の最大許可を指定する JSON ポリシーです。AWS Organizations は、お客様のビジネスが所有する複数の AWS アカウント をグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー SCPを一部またはすべてのアカウントに適用できます。SCP はメンバーアカウントのエンティティに対するアクセス許可を制限します (各 AWS アカウントのルートユーザー など)。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「SCP の仕組み」を参照してください。
- セッションポリシー セッションポリシーは、ロールまたはフェデレーティッドユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの許可される範囲は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから許可が派生する場合

もあります。これらのポリシーのいずれかを明示的に拒否した場合、許可は無効になります。詳細については、IAM ユーザーガイドの「セッションポリシー」を参照してください。

## 複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される許可を理解するのがさらに難しくなります。複数のポリシータイプが関連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、「IAM ユーザーガイド」の「ポリシーの評価ロジック」を参照してください。

# Amazon Bedrock で IAM が機能する仕組み

IAM を使用して Amazon Bedrock へのアクセスを管理する前に、Amazon Bedrock で使用できる IAM 機能について理解しておく必要があります。

#### Amazon Bedrock で使用できる IAM の機能

IAM 機能	Amazon Bedrock サポート
アイデンティティベースのポリシー	はい
<u>リソースベースのポリシー</u>	いいえ
ポリシーアクション	はい
ポリシーリソース	はい
ポリシー条件キー	はい
ACL	いいえ
ABAC (ポリシー内のタグ)	部分的
テンポラリ認証情報	はい
<u>プリンシパル許可</u>	はい
サービスロール	はい
サービスリンクロール	いいえ

Amazon Bedrock および AWS の他のサービスで大部分の IAM 機能を使用する方法の概要については、「IAM ユーザーガイド」の「IAM と連携する AWS のサービス」を参照してください。

Amazon Bedrock のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート はい

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、アクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「<u>IAM JSON ポリ</u>シーの要素のリファレンス」を参照してください。

Amazon Bedrock のアイデンティティベースのポリシー例

Amazon Bedrock のアイデンティティベースポリシーの例を確認するには、「 $\underline{\text{Amazon Bedrock or}}$ イデンティティベースのポリシー例」を参照してください。

Amazon Bedrock 内のリソースベースのポリシー

リソースベースのポリシーのサポート いいえ

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーが添付されているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービスを含めることができます。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウント にある場合、信頼できるアカウントの IAM 管理者は、リソースにアクセスするための許可をプリンシパルエンティティ(ユーザーまたはロール)に付与する必要もあります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで許可を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、ID ベースのポリシーをさらに付与する必要はありません。詳細については、IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

Amazon Bedrock のポリシーアクション

ポリシーアクションに対するサポート

はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための許可を付与するポリシーで使用 されます。

Amazon Bedrock アクションのリストを確認するには、「サービス認証リファレンス」の「<u>Amazon</u> Bedrock によって定義されたアクション」を参照してください。

Amazon Bedrock のポリシーアクションでは、アクションの前に、次のプレフィックスを使用します。

bedrock

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
    "bedrock:action1",
    "bedrock:action2"
]
```

Amazon Bedrock のアイデンティティベースポリシーの例を確認するには、「 $\underline{\text{Amazon Bedrock or}}$ イデンティティベースのポリシー例」を参照してください。

Amazon Bedrock のポリシーリソース

ポリシーリソースに対するサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素は、オブジェクトあるいはアクションが適用されるオブジェクトを 指定します。ステートメントには、Resource または NotResource 要素を含める必要がありま す。ベストプラクティスとして、<u>Amazon リソースネーム (ARN)</u> を使用してリソースを指定しま す。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対 して実行できます。

オペレーションのリスト化など、リソースレベルの許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (\*) を使用します。

"Resource": "\*"

Amazon Bedrock のアイデンティティベースポリシーの例を確認するには、「 $\underline{\text{Amazon Bedrock or}}$ イデンティティベースのポリシー例」を参照してください。

Amazon Bedrock のポリシー条件キー

サービス固有のポリシー条件キーのサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの<u>条件演算子</u>を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、AWS では AND 論理演算子を使用してそれらを評価します。単一の条件キーに複数の値を指定する場合、AWS では OR 論理演算子を使用して条件を評価します。ステートメントの許可が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー 名がタグ付けされている場合のみリソースにアクセスできる許可を付与することができます。詳細に ついては、IAM ユーザーガイドの「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS はグローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドの「AWS グローバル条件コンテキストキー」を参照してください。

すべての Amazon Bedrock アクションは、Amazon Bedrock モデルをリソースとして使用する条件 キーをサポートしています。

Amazon Bedrock のアイデンティティベースポリシーの例を確認するには、「 $\underline{\text{Amazon Bedrock or}}$ イデンティティベースのポリシー例」を参照してください。

Amazon Bedrock のACL

ACL のサポート

いいえ

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon Bedrock での ABAC

ABAC (ポリシー内のタグ) のサポート

部分的

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。AWS では、属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール)、および多数の AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初のステップです。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合に操作を許可するように ABAC ポリシーをします。

ABAC は、急速に成長している環境でポリシー管理が面倒な状況に役立ちます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/key-

name、aws:RequestTag/key-name、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素 でタグ情報を提供します。

サービスがすべてのリソースタイプに対して3つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ3つの条件キーすべてをサポートする場合、値は部分的です。

ABAC の詳細については、IAM ユーザーガイドの「<u>ABAC とは?</u>」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「<u>属</u>性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

Amazon Bedrock での一時的な認証情報の使用

一時的な認証情報のサポート

はい

AWS のサービス には、一時認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報を利用できる AWS のサービス を含めた詳細情報については、IAM ユーザーガイドの「IAM と連携する AWS のサービス」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時認証情報を使用していることになります。例えば、会社の Single Sign-On (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時認証情報を作成します。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、IAM ユーザーガイドの「ロールへの切り替え (コンソール)」を参照してください。

一時認証情報は、AWS CLI または AWS API を使用して手動で作成できます。作成後、一時認証情報を使用して AWS にアクセスできるようになります。AWS は、長期的なアクセスキーを使用する代わりに、一時認証情報を動的に生成することをお勧めします。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

#### Amazon Bedrock のクロスサービスプリンシパル許可

プリンシパル許可のサポート

はい

IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルと見 なされます。ポリシーは、プリンシパルにアクセス許可を付与します。一部のサービスを使用する 際に、アクションを実行することで、異なるサービスの別のアクションがトリガーされることがあり ます。この場合、両方のアクションを実行するための許可が必要です。ポリシー内で、アクションに 追加の依存アクションが必要かどうかを確認するには、「サービス認証リファレンス」の「Amazon Bedrock のアクション、リソース、条件キー」を参照してください。

Amazon Bedrock のサービスロール

サービスロールに対するサポート

はい

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細につい ては、「IAM ユーザーガイド」の「AWS のサービス にアクセス許可を委任するロールの作成」を参 照してください。

#### Marning

サービスロールのアクセス許可を変更すると、Amazon Bedrock の機能が破損する可能性が あります。Amazon Bedrock が指示する場合以外は、サービスロールを編集しないでくださ U<sub>°</sub>

Amazon Bedrock のサービスリンクロール

サービスリンクロールのサポート

いいえ

サービスリンクロールは、AWS のサービスにリンクされているサービスロールの一種です。サービ スは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスリン クロールは、AWS アカウント に表示され、サービスによって所有されます。IAM 管理者は、サービ スリンクロールの許可を表示できますが、編集することはできません。

# Amazon Bedrock のアイデンティティベースのポリシー例

デフォルトでは、ユーザーとロールには Amazon Bedrock リソースを作成または変更するアクセス許可がありません。また、AWS Management Console、AWS Command Line Interface (AWS CLI)、または AWS API を使用してタスクを実行することもできません。IAM 管理者は、リソースで必要なアクションを実行するためのアクセス許可をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「<u>IAM ポリシーの作成</u>」を参照してください。

Amazon Bedrock が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認証リファレンス」の「<u>Amazon Bedrock のアクション、リソー</u>ス、条件キー」を参照してください。

#### Note

Amazon Bedrock サービスは限定プレビューリリースとして提供されているため、その情報は「サービス認証リファレンス」には記載されていません。

#### トピック

- ポリシーのベストプラクティス
- Amazon Bedrock コンソールを使用する
- 自分の許可の表示をユーザーに許可する
- サードパーティーモデルのサブスクリプションへのアクセスを許可する
- 特定のモデルで推論するためのアクセス許可を拒否する
- カスタムジョブにトレーニングデータへのアクセスを付与する
- KMS キーをモデルカスタマイズに使用するためのアクセス許可

# ポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウント内で誰かが Amazon Bedrock リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、AWS アカウント に追加料金が発生する可能性があります。アイデンティティベースのポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください。

・ AWS マネージドポリシーを使用して開始し、最小特権の許可に移行する – ユーザーとワークロードへの許可の付与を開始するには、多くの一般的なユースケースのために許可を付与する AWS マネージドポリシーを使用します。これらは AWS アカウント で使用できます。ユースケースに応じた AWS カスタマーマネージドポリシーを定義することで、許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「AWSジョブ機能の管理ポリシー」を参照してください。

- 最小特権を適用する IAM ポリシーで許可を設定するときは、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「IAM でのポリシーとアクセス許可」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定することができます。また、AWS のサービス などの特定の AWS CloudFormation を介して使用する場合、条件を使用してサービスアクションへのアクセスを許可することもできます。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素:条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な許可を確保する IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM Access Analyzer は 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーを作成できるようサポートします。詳細については、「IAM ユーザーガイド」の「IAM Access Analyzer ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する AWS アカウント で IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA 保護 API アクセスの設定」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「<u>IAM でのセキュリ</u> <u>ティのベストプラクティス</u>」を参照してください。

# Amazon Bedrock コンソールを使用する

Amazon Bedrock コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。アクセス許可により、AWS アカウント の Amazon Bedrock リソースの詳細をリストおよび表示できます。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポ

リシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソール許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

Amazon Bedrock コンソールへのアクセスを許可するには、アクセスが必要なロールまたはエンティティに次のポリシーを付与します。ポリシー内のアクションの一部は Amazon Bedrock コンソールでのみ使用されるため、IAM は「Invalid Action」エラーを生成する可能性があります。これらのエラーを無視し、[次へ] を選択すれば、タスクを実行できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BedrockConsole",
            "Effect": "Allow",
            "Action": [
               "bedrock:ListFoundationModels",
               "bedrock:GetFoundationModel",
               "bedrock:InvokeModel",
               "bedrock: InvokeModelWithResponseStream",
               "bedrock:CreateModelCustomizationJob",
               "bedrock:GetModelCustomizationJob",
               "bedrock:GetFoundationModelAvailability",
               "bedrock:ListModelCustomizationJobs",
               "bedrock:StopModelCustomizationJob",
               "bedrock:GetCustomModel",
               "bedrock:ListCustomModels",
               "bedrock:DeleteCustomModel",
               "bedrock:CreateProvisionedModelThroughput",
               "bedrock:UpdateProvisionedModelThroughput",
               "bedrock:GetProvisionedModelThroughput",
               "bedrock:DeleteProvisionedModelThroughput",
               "bedrock:ListProvisionedModelThroughputs",
               "bedrock:ListTagsForResource",
               "bedrock: UntagResource",
               "bedrock: TagResource",
               "bedrock:PutFoundationModelEntitlement",
               "bedrock:GetModelInvocationLoggingConfiguration",
               "bedrock: PutModelInvocationLoggingConfiguration",
               "bedrock:CreateFoundationModelAgreement",
```

詳細については、IAM ユーザーガイドの「ユーザーへの許可の追加」を参照してください。

### 自分の許可の表示をユーザーに許可する

この例では、ユーザーアイデンティティに添付されたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーを作成する方法を示します。このポリシーには、コンソールで、 または AWS CLI か AWS API を使用してプログラム的に、このアクションを完了するアクセス許可 が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

### サードパーティーモデルのサブスクリプションへのアクセスを許可する

Amazon Bedrock モデルに初めてアクセスするには、Amazon Bedrock コンソールを使用してサードパーティーモデルをサブスクライブします。コンソールユーザーが引き受ける IAM ユーザーまたはロールには、サブスクリプションの API オペレーションにアクセスするためのアクセス許可が必要です。

次の例は、サブスクリプションの API オペレーションへのアクセスを許可するアイデンティティベースのポリシーを示しています。この例には、ポリシーの範囲を Marketplace の Amazon Bedrock 製品に限定する条件キーが含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:Subscribe"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws-marketplace:ProductId": [
                         "c468b48a-84df-43a4-8c46-8870630108a7",
                         "99d90be8-b43e-49b7-91e4-752f3866c8c7",
                         "b0eb9475-3a2c-43d1-94d3-56756fd43737",
                         "1d288c71-65f9-489a-a3e2-9c7f4f6e6a85",
                         "cc0bdd50-279a-40d8-829c-4009b77a1fcc",
                         "d0123e8d-50d6-4dba-8a26-3fed4899f388",
                         "a61c46fe-1747-41aa-9af0-2e0ae8a9ce05"
                    ]
            }
```

### 特定のモデルで推論するためのアクセス許可を拒否する

次の例は、特定のモデルで推論を実行するアクセス許可を拒否するアイデンティティベースポリシーを示しています。

# カスタムジョブにトレーニングデータへのアクセスを付与する

次の例では、Amazon S3 の入出力データの場所へのアクセス権を付与しています。

次の例では、同じアカウントで作成された複数のモデルカスタマイズジョブが 1 つのロールを引き 受けることを可能にする信頼ポリシーを作成します。

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "bedrock.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                     "aws:SourceAccount": "111122223333"
                },
                "ArnEquals": {
                     "aws:SourceArn": "arn:aws:bedrock:us-east-1:111122223333:model-
customization-job/*"
            }
        }
        }
    ]
}
```

# KMS キーをモデルカスタマイズに使用するためのアクセス許可

カスタムモデルを作成または使用するユーザーとロールに、適切な AWS KMS 許可を付与します。 詳細については、「<u>カスタマーマネージドキーを使用してカスタマイズジョブを実行する</u>」を参照し てください。

# サービスロール

Amazon Bedrock を自動的に実行するサービスロールを設定できます。

1. トレーニングに必要なアクセス許可 (S3、KMS) を持つだけでなく、Amazon Bedrock がこのロール (リファレンスロール) を引き受けることを許可する、信頼されたエンティティポリシーも持つ AmazonBedrock-ExecutionRole (RolePolicy) を作成します。このロールポリシーのAmazon リソースネーム (ARN) に注目してください。

- 2. このロールポリシーでは、Amazon Bedrock がこれらの認証情報を引き受けようとするときに、Amazon Bedrock ジョブの Amazon リソースネーム (ARN) と AWSアカウント IDに、aws:SourceArn と aws:SourceAccount のコンテキストキーを設定します。
- 3. Amazon Bedrock へのアクセス許可を持つ IAM エンティティを作成し、ステップ 1 でメモした RoleArn にロールのアクセス許可を渡します。

これにより、コンソールまたは CreateFineTuningJob オペレーションを使用し、>RoleArn と、トレーニングデータの Amazon S3 URI を渡して、微調整ジョブを実行できます。

Amazon Bedrock は CreateFineTuningJob オペレーションで PassRole チェックを行い、副次的なシナリオが混乱するのを回避します。この PassRole チェックは、AWS サービスに渡されて AWS サービスに代行させるアクセス許可を持つロールを、入力としてユーザーが渡そうとしているかどうかを検証するものです。次に、Amazon Bedrock は入力されたロール ARN を使用して認証情報を生成します。Amazon Bedrock はこの認証情報を使用して、アカウントに保存されているトレーニングデータと検証データをダウンロードしたり、トレーニングメトリクスをアカウントにアップロードしたりします。

# Amazon Bedrock のアイデンティティとアクセスに関するトラブルシューティング

以下の情報を使用して、Amazon Bedrock と IAM の使用時に発生する可能性がある一般的な問題の 診断と修正に役立てます。

#### トピック

- I am not authorized to perform an action in Amazon Bedrock
- I am not authorized to perform iam:PassRole

サービスロール 239

• I want to allow people outside of my AWS アカウント to access my Amazon Bedrock resources

### I am not authorized to perform an action in Amazon Bedrock

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要なbedrock: *GetWidget* アクセス許可を持っていない場合に発生するものです。

User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: bedrock: GetWidget on resource: my-example-widget

この場合、bedrock: *GetWidget* アクションを使用して *my-example-widget* リソースへのアク セスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者に問い合わせてください。サインイン資格情報を提供した担当者が管理者です。

I am not authorized to perform iam:PassRole

「I am not authorized to perform iam: PassRole」というエラーが表示された場合は、Amazon Bedrock にロールを渡すことを許可するようにポリシーを更新する必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスリンクロールを作成せずに、 既存のロールをサービスに渡すことが許可されています。そのためには、サービスにロールを渡す許可が必要です。

次の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Amazon Bedrock でアクションを実行しようとした場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与されたアクセス許可が必要です。Mary には、ロールをサービスに渡す許可がありません。

User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole

この場合、メアリーのポリシーを更新してメアリーに iam: PassRole アクションの実行を許可する必要があります。

トラブルシューティング 240

サポートが必要な場合は、AWS 管理者に問い合わせてください。サインイン資格情報を提供した担当者が管理者です。

I want to allow people outside of my AWS アカウント to access my Amazon Bedrock resources

他のアカウントのユーザーや組織外のユーザーが、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定することができます。リソースベースのポリシーまたはアクセス制御リスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください。

- Amazon Bedrock がこれらの機能をサポートしているかどうかを確認するには、「Amazon Bedrock で IAM が機能する仕組み」を参照してください。
- 所有している AWS アカウント 全体のリソースへのアクセス権を提供する方法については、「IAM ユーザーガイド」の「<u>所有している別の AWS アカウント アカウントへのアクセス権を IAM ユー</u> ザーに提供」を参照してください。
- サードパーティーの AWS アカウント にリソースへのアクセス権を提供する方法については、 「IAM ユーザーガイド」の「<u>第三者が所有する AWS アカウント へのアクセス権を付与する</u>」を 参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、「IAM ユーザーガイド」の 「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの用途の違いについては、 「IAM ユーザーガイド」の「<u>IAM ロールとリソースベースのポリシーとの相違点</u>」を参照してく ださい。

# Amazon Bedrock のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの対象であるかどうかを確認するには、「<u>コンプライアンスプログラムによる対象範囲内の AWS のサービス のサービス</u>」をご覧いただき、関心のあるコンプライアンスプログラムを選択してください。一般的な情報については、<u>AWS コンプ</u>ライアンスプログラムを参照してください。

AWS Artifact を使用して、サードパーティーの監査レポートをダウンロードできます。詳細については、「AWS Artifact におけるダウンロードレポート」を参照してください。

コンプライアンス検証 241

AWS のサービス を使用する際のユーザーのコンプライアンス責任は、ユーザーのデータの機密性や 貴社のコンプライアンス目的、適用される法律および規制によって決まります。AWS では、コンプ ライアンスに役立つ次のリソースを提供しています。

- <u>セキュリティとコンプライアンスのクイックスタートガイド</u> これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境を AWS にデプロイするためのステップを示します。
- 「Amazon Web Services での HIPAA のセキュリティとコンプライアンスのためのアーキテク  $\underline{f}$  ナャ 」 このホワイトペーパーは、企業が AWS を使用して HIPAA 対象アプリケーションを作成 する方法を説明しています。

### Note

すべての AWS のサービス が HIPAA 適格であるわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS コンプライアンスのリソース このワークブックおよびガイドのコレクションは、お客様の業界と拠点に適用されるものである場合があります。
- AWS Config デベロッパーガイドの<u>ルールでのリソースの評価</u> AWS Config サービスでは、自社のプラクティス、業界ガイドライン、および規制に対するリソースの設定の準拠状態を評価します。
- AWS Security Hub この AWS のサービス は、AWS 内のセキュリティ状態の包括的なビューを提供します。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- <u>AWS Audit Manager</u> この AWS のサービス は AWS の使用状況を継続的に監査し、リスクの管理 方法やコンプライアンスを業界スタンダードへの準拠を簡素化するために役立ちます。

# Amazon Bedrock でのインシデントへの対応

Amazon Bedrock でのインシデント対応は AWS の責任範囲に含まれます。AWS には、インシデントへの対応を管理する、正式な文書化されたポリシーとプログラムがあります。

広範な影響を与える AWS の運用上の問題は、AWS Service Health Dashboard に投稿されます。

インシデントへの対応 242

運用上の問題は AWS Health Dashboard を介して個々のアカウントにも投稿されます。AWS Health Dashboard の使い方については、「AWS Health ユーザーガイド」を参照してください。

# Amazon Bedrock の耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心 に構築されています。AWS リージョン には、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている物理的に独立・隔離された複数のアベイラビリティーゾーンがあります。アベイラビリティーゾーンを使用すると、中断することなくゾーン間で自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用できます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョン とアベイラビリティーゾーンの詳細については、「AWS グローバルインフラスト ラクチャ」を参照してください。

# Amazon Bedrock でのインフラストラクチャセキュリティ

マネージドサービスである Amazon Bedrock は、AWS グローバルネットワークセキュリティによって保護されています。AWSセキュリティサービスと AWS がインフラストラクチャを保護する方法については、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected Framework」の「インフラストラクチャ保護」を参照してください。

ネットワーク経由で Amazon Bedrock にアクセスするには、AWS が発行した API コールを使用します。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS) TLS 1.2 および TLS 1.3 をお勧めします。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートです。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、<u>AWS Security Token Service</u> (AWS STS)を使用して、テンポラリセキュリティ認証情報を生成し、リクエストに署名することもできます。

耐障害性 243

#### サービス間の混乱した代理の防止

混乱した代理問題とは、アクションを実行する許可を持たないエンティティが、より高い特権を持つエンティティにそのアクションの実行を強制できるというセキュリティ問題です。AWSでは、サービス間でのなりすましが、混乱した代理問題を生じさせることがあります。サービス間でのなりすましは、1つのサービス (呼び出し元サービス)が、別のサービス (呼び出し対象サービス)を呼び出すときに発生する可能性があります。呼び出し元サービスは、本来ならアクセスすることが許可されるべきではない方法でその許可を使用して、別のお客様のリソースに対する処理を実行するように操作される場合があります。これを防ぐために AWSでは、お客様のすべてのサービスのデータを保護するのに役立つツールを提供しています。これには、アカウントのリソースへのアクセス許可が付与されたサービスプリンシパルを使用します。

リソースポリシー内では <u>aws:SourceArn</u> および <u>aws:SourceAccount</u> のグローバル条件コンテキストキーを使用して、Amazon Bedrock が別のサービスに付与する、リソースへのアクセス許可を制限することをお勧めします。クロスサービスのアクセスにリソースを 1 つだけ関連付けたい場合は、aws:SourceArn を使用します。クロスサービスが使用できるように、アカウント内の任意のリソースを関連づけたい場合は、aws:SourceAccount を使用します。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定しながら、aws:SourceArn グローバル条件コンテキストキーを使用することです。リソースの完全な ARN が不明な場合や、複数のリソースを指定する場合には、グローバルコンテキスト条件キーaws:SourceArn で、ARN の未知部分を示すためにワイルドカード文字 (\*) を使用します。例えば、arn:aws:bedrock:\*:123456789012:\* です。

aws:SourceArn の値に Amazon S3 バケット ARN などのアカウント ID が含まれていない場合は、両方のグローバル条件コンテキストキーを使用して、アクセス許可を制限する必要があります。

aws:SourceArn の値は ResourceDescription である必要があります。

次の例では、Bedrock の aws:SourceArn と aws:SourceAccount グローバル条件コンテキストキーを使用して、混乱した代理問題を回避する方法を示します。

### Amazon Bedrock での設定と脆弱性の分析

構成および IT 管理は、AWS とお客様の間で共有される責任です。詳細については、AWS <u>責任共有</u> モデルを参照してください。

#### モニター

Amazon Bedrock は Amazon CloudWatch と Amazon EventBridge でモニタリングできます。

#### トピック

- Amazon CloudWatch による Amazon Bedrock のモニタリング
- Amazon EventBridge で Amazon Bedrock イベントをモニタリングする
- AWS CloudTrail を使用した Amazon Bedrock API コールのログ記録

#### Amazon CloudWatch による Amazon Bedrock のモニタリング

Amazon CloudWatch を使用して Amazon Bedrock をモニタリングすることで、raw データを収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。CloudWatch コンソールを使用して、メトリクスをグラフ化できます。また、特定のしきい値を超えないかどうかをモニタリングするアラームを設定するとともに、しきい値を超えたときに通知を送信したりアクションを実行したりできます。

詳細については、「Amazon CloudWatch ユーザーガイド」の「<u>Amazon CloudWatch とは</u>」を参照 してください。

#### トピック

- ランタイムメトリクス
- <u>ログへの CloudWatch メトリクスの記録</u>
- Amazon Bedrock 用に CloudWatch メトリクスを使用する
- Amazon Bedrock メトリクスを表示する

#### ランタイムメトリクス

次の表は、Amazon Bedrock が提供するランタイムメトリックスについて説明しています。

メトリクス名	単位	説明
呼び出し	サンプル数	InvokeModel または InvokeModelWithRes

メトリクス名	単位	説明
		ponseStream API オペレー ションへのリクエスト数。
InvocationLatency	ミリ秒	呼び出しのレイテンシー。
InvocationClientErrors	サンプル数	クライアント側でエラーが発 生した呼び出しの数。
InvocationServerErrors	サンプル数	AWS サーバー側でエラーが発 生した呼び出しの数。
InvocationThrottles	サンプル数	システムがスロットリングし た呼び出しの数。
InputTokenCount	サンプル数	テキスト入力のトークンの 数。
OutputTokenCount	サンプル数	テキスト出力のトークンの 数。
ContentFilteredCount	サンプル数	テキスト出力コンテンツが フィルターされた回数。
OutputImageCount	サンプル数	出力画像の数。

#### ログへの CloudWatch メトリクスの記録

配信が成功または失敗するたびに、名前空間 AWS/Bedrock とAcross all model IDsディメンションに関する次の Amazon CloudWatch メトリクスが出力されます。

- $\bullet \ {\tt ModelInvocationLogsCloudWatchDeliverySuccess}$
- $\bullet \ \mathsf{ModelInvocationLogsCloudWatchDeliveryFailure}$
- ModelInvocationLogsS3DeliverySuccess
- ModelInvocationLogsS3DeliveryFailure
- ModelInvocationLargeDataS3DeliverySuccess

ModelInvocationLargeDataS3DeliveryFailure

アクセス許可の設定ミスや一時的な障害によりログが配信されない場合、配信は 24 時間後まで、定期的に再試行されます。

#### Amazon Bedrock 用に CloudWatch メトリクスを使用する

Amazon Bedrock オペレーションのメトリクスを取得するには、以下の情報を指定する必要があります。

- メトリクスディメンション。ディメンションは、メトリクスを識別するための名前と値のペアのセットです。Amazon Bedrock は、以下のディメンションをサポートしています。
  - ModelId すべてのメトリクス
  - ModelId + ImageSize + BucketedStepSize OutputImageCount
- メトリクス名 (InvocationClientErrors など)。

Amazon Bedrock のメトリクスは、AWS Management Console、AWS CLI、または CloudWatch API で取得できます。CloudWatch API を使用するには、AWS Software Development Kits (SDK) の 1 つまたは CloudWatch API ツールを使用します。

CloudWatch で Amazon Bedrock をモニタリングするには、CloudWatch への適切なアクセス許可が必要です。詳細については、「Amazon CloudWatch ユーザーガイドの「<u>Amazon CloudWatch に対</u>する認証とアクセスコントロール」を参照してください。

#### Amazon Bedrock メトリクスを表示する

CloudWatch コンソールで Amazon Bedrock メトリクスを表示します。

メトリクスを表示するには (CloudWatch コンソール)

- 1. AWS Management Console にサインインして、CloudWatch コンソール (<a href="https://console.aws.amazon.com/cloudwatch/">https://console.aws.amazon.com/cloudwatch/</a>) を開きます。
- 2. [メトリクス]、[すべてのメトリクス]、[ModelId] の順に選択します。

# Amazon EventBridge で Amazon Bedrock イベントをモニタリングする

Amazon EventBridge を使用して、Amazon Bedrock 内のステータス変更イベントをモニタリングできます。Amazon EventBridge を使用すると、Amazon Bedrock のモデルカスタマイズジョブのステータス変更に自動的に応答するように Amazon SageMaker を設定できます。Amazon Bedrockからのイベントは、ほぼリアルタイムに Amazon EventBridge に提供されます。イベントがルールに一致したときに自動的にアクションが実行されるようにする、シンプルなルールを作成できます。Amazon EventBridge を Amazon Bedrock と併用すると、次のことが可能になります。

- 今後、新しい非同期ワークフローを追加するかどうかに関係なく、トリガーしたモデルカスタマイズで状態変更イベントが発生するたびに通知を発行する。その公開されたイベントでは、下流のワークフローのイベントに応答するのに十分な情報が提供されます。
- GetModelCustomizationJob API を呼び出さずにジョブステータスの更新内容を配信できます。つまり、API レート制限の問題が処理され、API が更新されて、追加のコンピューティングリソースが削減されます。

Amazon EventBridge から AWS イベントを受信するのにコストはかかりません。Amazon EventBridge の詳細については、「Amazon EventBridge とは」を参照してください。

#### Note

- Amazon Bedrock は、ベストエフォートベースでイベントを発行します。Amazon Bedrock からのイベントは、ほぼリアルタイムで Amazon EventBridge に提供されます。Amazon EventBridge では、イベントに応答してプログラムによるアクションをトリガーするルールを作成できます。例えば、SNS トピックを呼び出して E メール通知を送信したり、関数を呼び出して何らかのアクションを実行したりするルールを設定できます。詳細については、「Amazon EventBridge ユーザーガイド」を参照してください。
- AWS Bedrock は、トリガーするモデルカスタマイズジョブの状態変更があるたびに新しいイベントを作成し、そのようなイベントをベストエフォートで配信します。

#### トピック

- 使用方法
- EventBridge スキーマ

イベントのモニタリング 249

- ルールとターゲット
- AWS Bedrock イベントを処理するルールを作成する

### 使用方法

Amazon Bedrock からイベントを受信するには、Amazon EventBridge を通じて状態変更データを一致、受信、処理するルールとターゲットを作成する必要があります。Amazon EventBridge は、AWSサービス、SaaS パートナー、および顧客アプリケーションから状態変更イベントを取り込むサーバーレスイベントバスです。作成されたルールやパターンに基づいてイベントを処理し、そのイベントを AWS Lambda、Amazon Simple Queue Service や Amazon Simple Notification Service など、選択された 1 つ以上の「ターゲット」にルーティングします。

Amazon Bedrock は、モデルカスタマイズジョブの状態が変更されるたびに、Amazon EventBridge を介してイベントを公開します。いずれの場合も、新しいイベントが作成されて Amazon EventBridge に送信され、Amazon EventBridge はそのイベントをデフォルトのイベントバスに送信します。このイベントには、どのカスタマイズジョブの状態が変更されたか、およびそのジョブの現在の状態が示されます。Amazon EventBridge は、作成されたルールと一致するイベントを受信すると、そのイベントを指定したターゲットにルーティングします。ルールを作成したら、それらのターゲットと下流のワークフローをイベントの内容に則して設定できます。

#### EventBridge スキーマ

EventBridge イベントスキーマに含まれる以下のイベントフィールドは、Amazon Bedrock に固有のものです。

- jobArn モデルカスタマイズジョブの ARN。
- outputModelArn 出力モデルの ARN。トレーニングジョブが完了したときに公開されます。
- jobStatus ジョブの現在のステータス。
- FailureMessage 失敗した場合のメッセージ。トレーニングジョブが失敗したときに公開されます。

#### イベント例

以下は、失敗したモデルカスタマイズジョブのイベント JSON の例です。

```
{
    "version": "0",
```

使用方法 250

```
"id": "UUID",
    "detail-type": "Model Customization Job State Change",
    "source": "aws.bedrock",
    "account": "123412341234",
    "time": "2023-08-11T12:34:56Z",
    "region": "us-east-1",
    "resources": [ "arn:aws:bedrock:us-east-1:12341234:model-customization-job/
abcdefghwxyz" ],
    "detail": {
        "version": "0.0",
        "jobName": "abcd-wxyz",
        "jobArn": "arn:aws:bedrock:us-east-1:123412341234:model-customization-job/
abcdefghwxyz",
        "outputModelName": "dummy-output-model-name",
        "outputModelArn": "arn:aws:bedrock:us-east-1:12341234:dummy-output-model-
name",
        "roleArn": "arn:aws:iam::123412341234:role/JobExecutionRole",
        "jobStatus": "Failed",
        "failureMessage": "Failure Message here.",
        "creationTime": "2023-08-11T10:11:12Z",
        "lastModifiedTime": "2023-08-11T12:34:56Z",
        "endTime": "2023-08-11T12:34:56Z",
        "baseModelArn": "arn:aws:bedrock:us-east-1:123412341234:base-model-name",
        "hyperParameters": {
            "batchSize" : "batchSizeNumberUsed",
            "epochCount": "epochCountNumberUsed",
            "learningRate": "learningRateUsed",
            "learningRateWarmupSteps": "learningRateWarmupStepsUsed"
            },
        "trainingDataConfig": {
            "s3Uri": "s3://bucket/key",
            },
        "validationDataConfig": {
            "s3Uri": "s3://bucket/key",
            },
        "outputDataConfig": {
            "s3Uri": "s3://bucket/key",
            }
    }
}
```

EventBridge スキーマ 251

#### ルールとターゲット

作成したルールと一致する受信イベントは、そのルールに指定したターゲットにルーティングされ、ターゲットはこれらのイベントを処理します。ターゲットは JSON 形式をサポートしており、Amazon EC2 インスタンス、Lambda 関数、Kinesis ストリーム、Amazon ECS タスク、Step Functions、Amazon SNS トピック、Amazon SQS などの AWS サービスを含めることができます。イベントを正しく受信して処理するには、イベントデータを一致、受信、および正しく処理するルールとターゲットを作成する必要があります。これらのルールとターゲットを作成するには、Amazon EventBridge コンソールまたは AWS CLI を使用します。

#### ルールの例

下記のルールは source ["aws.bedrock"] によって出力されるイベントパターンとの一致を行います。このルールは、ソース「aws.bedrock」を持つ Amazon EventBridge によってデフォルトのイベントバスに送信されたすべてのイベントをキャプチャします。

```
{
    "source": ["aws.bedrock"]
}
```

#### ターゲット

Amazon EventBridge でルールを作成するときは、ルールパターンに一致するイベントを EventBridge が送信するターゲットを指定する必要があります。これらのターゲットとして は、SageMaker パイプライン、Lambda 関数、SNS トピック、SQS キューなど、EventBridge が 現在サポートしているターゲットを使用できます。イベントのターゲットを設定する方法について は、Amazon EventBridge のドキュメントを参照してください。Amazon Simple Notification Service をターゲットとして使用する方法を示す手順については、「AWS Bedrock イベントを処理するルールを作成する」を参照してください。

#### AWS Bedrock イベントを処理するルールを作成する

AWS Bedrock イベントに関する E メール通知を受け取るには、以下の手順を完了してください。

Amazon Simple Notification Service トピックを作成する

- 1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
- 2. ナビゲーションペインで、[トピック] を選択します。

ルールとターゲット 252

- 3. [Create topic] (トピックの作成) を選択します。
- 4. [Type] (タイプ) で、[Standard] (標準) を選択します。
- 5. [Name] (名前) で、トピックの名前を入力します。
- 6. [Create topic] (トピックの作成) を選択します。
- 7. [Create subscription] (サブスクリプションの作成) を選択します。
- 8. [プロトコル] で [E メール] を選択します。
- 9. [Endpoint] (エンドポイント) で、通知を受信するメールアドレスを入力します。
- 10. [Create subscription] (サブスクリプションの作成) を選択します。
- 11. 次の件名の E メールメッセージが届きます: AWS Notification Subscription Confirmation。指示に沿って操作し、登録を確認します。

AWS Bedrock イベントを処理するルールを作成するには、次の手順を使用します。

AWS Bedrock イベントを処理するルールを作成するには

- 1. Amazon EventBridge コンソール (https://console.aws.amazon.com/events/) を開きます。
- 2. [Create rule] を選択します。
- 3. [Name] (名前) に、ルールの名前を入力します。
- 4. [ルールタイプ] では、[イベントパターンを持つルール] を選択します。
- 5. [Next] (次へ) をクリックします。
- 6. [Event pattern] (イベントパターン) の場合は、次のいずれかを実行します。
  - a. [イベントソース] で [AWS のサービス] を選択します。
  - b. [AWS のサービス] で [Amazon Bedrock] を選択します。
  - c. [イベントタイプ] で [モデルカスタマイズジョブの状態変更] を選択します。
  - d. デフォルトでは、すべてのイベントに通知が送信されます。必要に応じて、特定のジョブ状態のイベントをフィルタリングするイベントパターンを作成できます。
  - e. [Next] (次へ) をクリックします。
- 7. 次のようにターゲットを指定します。
  - a. [ターゲットタイプ] では、[AWS のサービス] を選択します。
  - b. [Select a target] (ターゲットの選択) には、[SNS topic] (SNS トピック) を選択します。
  - c. [トピック] で、通知用に作成した SNS トピックを選択します。

- d. [Next] (次へ) をクリックします。
- 8. (オプション) ルールにタグを追加します。
- 9. [Next] (次へ) をクリックします。
- 10. [Create rule] を選択します。

# AWS CloudTrail を使用した Amazon Bedrock API コールのログ記録

Amazon Bedrock は AWS CloudTrail と統合されています。後者は、Amazon Bedrock の AWS のサービス、ユーザー、またはロールによって実行されたアクションを記録するサービスです。CloudTrail は、Amazon Bedrock へのすべての API コールをイベントとしてキャプチャします。キャプチャされるコールには、Amazon Bedrock コンソールからのコールと、Amazon Bedrock API オペレーションへのコードコールが含まれます。証跡を作成する場合は、Amazon Bedrock のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最近のイベントを表示できます。CloudTrail により収集された情報を使用して、Amazon Bedrock に対して行われたリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエストが行われた日時、および追加の詳細を特定することができます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

#### CloudTrail 内の Amazon Bedrock 情報

CloudTrail は、AWS アカウントを作成すると、その中で有効になります。Amazon Bedrock でアクティビティが発生すると、そのアクティビティは [イベント履歴] で CloudTrail イベントに、AWS の他サービスのイベントと共に記録されます。最近のイベントは、AWS アカウント で表示、検索、ダウンロードできます。詳細については、 CloudTrail イベント履歴でのイベントの表示を参照してください。

Amazon Bedrock のイベントなど、AWS アカウントでのイベントの継続的な記録については、証跡を作成します。追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョン に適用されます 証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づく対応するためにその他の AWS のサービスを設定できます。詳細については、次を参照してください。

CloudTrail ログ 254

- 「追跡を作成するための概要」
- CloudTrail がサポートされているサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail
   ログファイルを受け取る

すべての Amazon Bedrock アクションは、CloudTrail によりログが記録されます (また、それらのアクションのリファレンスについては「<u>Amazon Bedrock API リファレンス</u>」にあります)。例えば、InvokeModel、ListFoundationModels、StopModelCustomizationJobの各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーティッドユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

#### Amazon Bedrock ログファイルエントリの概要

証跡は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、InvokeModelアクションを示す CloudTrail ログエントリです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
      "type": "IAMUser",
```

```
"principalId": "AROAICFHPEXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/userxyz",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "userxyz"
    },
    "eventTime": "2023-10-11T21:58:59Z",
    "eventSource": "bedrock.amazonaws.com",
    "eventName": "InvokeModel",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Boto3/1.28.62 md/Botocore#1.31.62 ua/2.0 os/macos#22.6.0 md/
arch#arm64 lang/python#3.9.6 md/pyimpl#CPython cfg/retry-mode#legacy Botocore/1.31.62",
    "requestParameters": {
        "modelId": "stability.stable-diffusion-xl-v0"
    },
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "cipher suite",
        "clientProvidedHostHeader": "bedrock-runtime.us-west-2.amazonaws.com"
    }
}
```

#### Amazon Bedrock による不正検出

AWS は AI の責任ある利用に取り組んでいます。不正利用の芽を摘むため、Amazon Bedrock は、「AWS <u>利用規約」(AUP)と「AWS Responsible AI Policy</u>」、またはサードパーティーモデルプロバイダーの AUP に対する違反の芽を特定して回避する不正利用自動検出メカニズムを実装しています。

Amazon の不正検出メカニズムは完全に自動化されているため、ユーザー入力やモデル出力を人間が確認したりアクセスしたりする必要はありません。

不正の自動検出は以下の要素から成ります。

- コンテンツの分類 分類子を使用して、ユーザー入力やモデル出力に含まれる有害なコンテンツ (暴力を扇動するコンテンツなど)を検出します。分類子は、モデルの入力と出力を処理し、攻撃 の種類と信頼度を割り当てるアルゴリズムです。これらの分類子は Amazon Titan とサードパー ティーのモデルの両方に対して実行できます。分類プロセスは自動化されており、ユーザー入力や モデル出力を人間が確認する必要はありません。
- パターンの特定 分類メトリクスを使用して、違反の芽や繰り返し発生する動作を特定します。当社では、集約され匿名化された分類メトリクスをまとめ、サードパーティーのモデルプロバイダーと共有する場合があります。Amazon Bedrock はユーザー入力やモデル出力を保管せず、サードパーティーのモデルプロバイダーと共有することもありません。
- お客様への連絡 当社は、お客様による Bedrock のご利用や当社の規約への遵守について、情報を求める場合があります。お客様が AWS の規約またはサードパーティーモデルプロバイダーの AUP に違反している可能性のある方法で本サービスを引き続き使用する場合、AWS は、深刻度、アクティビティの反復性、お客様に協力いただけていない、お客様がサービスの不正利用を防止するメカニズムを設定していないときには、Bedrock へのアクセスを停止することがあります。

ご質問がある場合は、AWS サポートまでご連絡ください。詳細については、「<u>Amazon Bedrock</u> <u>FAQs</u>」を参照してください。

# インターフェイス VPC エンドポイント (AWS PrivateLink) を使用する

AWS PrivateLink を使用して、VPC と Amazon Bedrock 間にプライベート接続を作成できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、AWS Direct Connect 接続のいずれかを使用せずに、VPC 内にあるかのように Amazon Bedrock にアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくても Amazon Bedrock にアクセスできます。

このプライベート接続を確立するには、AWS PrivateLink を利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、Amazon Bedrock 宛てのトラフィックのエントリポイントとして機能するリクエスタ管理型ネットワークインターフェイスです。

詳細については、「AWS PrivateLink Guide (AWS PrivateLink ガイド)」の「<u>Access an AWS のサービス using an interface VPC endpoint</u> (インターフェイス VPC エンドポイントを使用して にアクセスする)」を参照してください。

#### Amazon Bedrock VPC エンドポイントに関する考慮事項

Amazon Bedrock のインターフェイスエンドポイントを設定する前に、「AWS PrivateLink ガイド」 の「考慮事項」を確認してください。

Amazon Bedrock は、VPC エンドポイントを介して<u>ランタイム API アクション</u>を呼び出すことができます。

Amazon Bedrock エンドポイントは、リージョン内のすべてのアベイラビリティーゾーンで使用できるわけではありません。アベイラビリティーゾーンは、エンドポイントの作成時に次のコマンドを使用して一覧表示できます。

aws ec2 describe-vpc-endpoint-services \
 --service-names com.amazonaws.region.bedrock-runtime --region region

#### Important

com.amazonaws.region.bedrock という名前のサービスは廃止されました。

考慮事項 258

#### Amazon Bedrock 用のインターフェイスエンドポイントを作成する

Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) を使用して、Amazon Bedrock のインターフェイスエンドポイントを作成できます。詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントの作成」を参照してください。

以下のサービス名を使用して Amazon Bedrock のインターフェイスエンドポイントを作成します。

com.amazonaws.region.bedrock-runtime

エンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 ( など) を使 用して、Amazon Bedrock への API リクエストを実行できます。例えば、bedrock-runtime.useast-1.amazonaws.com です。

Important

プライベート DNS bedrock.*region*.amazonaws.com は廃止されました。

# インターフェイスエンドポイントのエンドポイントポリシーを作成 する

エンドポイントポリシーは、インターフェイスエンドポイントにアタッチできる IAM リソースで す。デフォルトのエンドポイントポリシーでは、インターフェイスエンドポイント経由での Amazon Bedrock へのフルアクセスが許可されています。 VPC から Amazon Bedrock への許可されたアクセ スをコントロールするには、カスタムエンドポイントポリシーをインターフェイスエンドポイントに アタッチします。

エンドポイントポリシーは、以下の情報を指定します。

- アクションを実行できるプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール)。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、「AWS PrivateLink ガイド」の「Control access to services using endpoint policies (エンドポイントポリシーを使用してサービスへのアクセスをコントロールする)」を参照してくださ U<sub>°</sub>

例: Amazon Bedrock アクションの VPC エンドポイントポリシー

以下は、カスタムエンドポイントポリシーの例です。インターフェイスエンドポイントにアタッチされると、このポリシーは、すべてのリソースですべてのプリンシパルに、リストされている Amazon Bedrock アクションへのアクセス権を付与します。

#### Amazon Bedrock のクォータ

AWS アカウント には、AWS のサービスごとにデフォルトのクォータ (以前は制限と呼ばれたもの)があります。特に明記されていない限り、クォータはリージョンごとに存在します。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

一部の Service Quotas は調整または引き上げを行うことができます。クォータを調整できるかどうかについては、以下の表の 調整可能列を参照してください。クォータ引き上げをリクエストするには、「サービスクォータの引き上げ」フォームを使用します。

お客様の AWS アカウントには、Amazon Bedrock に関連する以下のクォータが割り当てられています。

#### ランタイムクォータ

推論にモデルを使用する場合、次のクォータが適用されます。

モデル	1 分あたりに処理され るリクエスト数	1 分あたりに処理され るトークン数	調整可能
Amazon Titan Express	400	300,000	いいえ
Amazon Titan Text Embeddings	2,000	300,000	いいえ
Anthropic Claude Instant	400	300,000	いいえ
Anthropic Claude V2	100	200,000	いいえ
Al21 Labs Jurassic-2 Mid	400	300,000	いいえ
Al21 Labs Jurassic-2 Ultra	100	300,000	いいえ

ランタイムクォータ 261

モデル	1 分あたりに処理され るリクエスト数	1 分あたりに処理され るトークン数	調整可能
Cohere Command	400	300,000	いいえ
Cohere Command Light	400	300,000	いいえ
Cohere Embed English	400	300,000	いいえ
Cohere Embed Multilingual	400	300,000	いいえ
Meta Llama2 Chat 13B	800	300,000	いいえ
Stability.ai Diffusion XL 1.0	60	該当なし	いいえ

# モデルカスタマイズのクォータ

モデルカスタマイズには、次のクォータが適用されます。

#### モデルクォータ

Titan Express では、次のクォータが適用されます。

説明	最大値	調整可能
バッチサイズが 1 の場合の入 カトークンと出力トークンの 合計	4,096	いいえ
バッチサイズが 2~4 の場合の 入力トークンと出力トークン の合計	2,048	いいえ

モデルカスタマイズのクォータ 262

説明	最大値	調整可能
文字数クォータ	トークンクォータ x 6	いいえ

### 微調整用のクォータ

微調整のために、以下のクォータが適用されます。

説明	デフォルト	調整可能
データセット内のトレーニン グレコードの数	10,000	はい
データセット内の検証レコー ドの数	1,000	はい
トレーニングデータセットの ファイルサイズ	1 GB	はい
検証データセットのファイル サイズ	100 MB	はい

### トレーニングのクォータ

モデルトレーニングには次のクォータが適用されます。

説明	デフォルト	調整可能
アカウントごとに一度に送信 されるモデルカスタマイズ ジョブの数	2	はい
カスタマイズモデルの数	100	はい
リソースに付けられるタグの 数	「 <u>AWSResource Groups エン</u> <u>ドポイントとクォータ</u> 」を参 照	はい

微調整用のクォータ 263

# プロビジョンドスループットのクォータ

プロビジョンドスループットには以下のクォータが適用されます。

説明	デフォルト	調整可能
基盤モデルのプロビジョンド スループットのモデルユニッ ト数	0	はい
カスタムモデルのプロビジョ ンドスループットのモデルユ ニット数	2	はい

## Amazon Bedrock ユーザーガイドのドキュメント履歴

를서 pp

• ドキュメントの最新更新日: 2023 年 11 月 13 日

亦田

次の表に、Amazon Bedrock の各リリースにおける重要な変更点を示します。このドキュメントの更新に関する通知については、RSS フィードでサブスクライブできます。

 $\Box A +$ 

変更	説明	日付
新しいモデルリリース	ドキュメントを更新して、新 しい Meta モデルと Cohere モ デルを記載しました。詳細に ついては、「 <u>What is Amazon</u> <u>Bedrock?</u> 」を参照してくださ い。	2023年11月13日
<u>ドキュメントのローカリゼー</u> <u>ション</u>	<u>日本語</u> と <u>ドイツ語</u> での Amazon Bedrock ドキュメン トが公開されました。	2023年10月20日
リージョンの拡張	Amazon Bedrock が欧州 (フランクフルト) (eu-central-1) で利用可能になりました。エンドポイントの詳細については、「Amazon Bedrock エンドポイントとクォータ」を参照してください。	2023年10月19日
リージョンの拡張	Amazon Bedrock が、アジア パシフィック (東京) (ap-north east-1) リージョンで利用可 能になりました。エンドポ イントの詳細については、 「Amazon Bedrock エンドポ イントとクォータ」を参照し てください。	2023年10月3日

#### 限定プレビューリリース

Amazon Bedrock サービスの限定プレビューリリースです。詳細については、「<u>What is Amazon Bedrock?</u>」を参照してください。

2023年9月28日

# AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「 $\underline{AWS}$  用語集」を参照してください。