

AWS Builders Online Series

T2-1

はじめての生成系 AI 入門 ～AWS クラウドにおける 生成系 AI 活用法～

本橋 和貴

アマゾン ウェブ サービス ジャパン合同会社
AWS テクノロジーパートナーシップ
機械学習パートナーソリューションアーキテクト



AI/ML やっていますか？



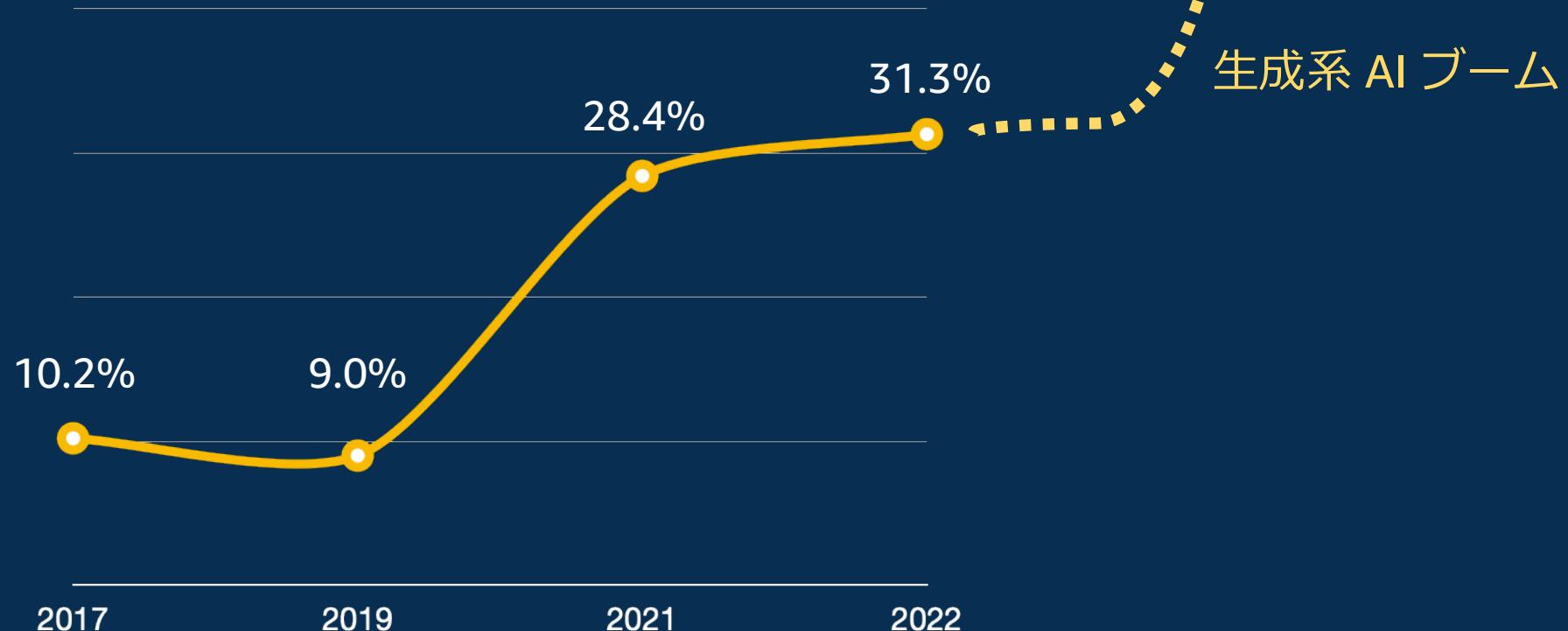
AI の民主化



日本企業における AI 導入率* の推移



* AI を導入している & 現在実証実験 (PoC) を行っている企業の割合

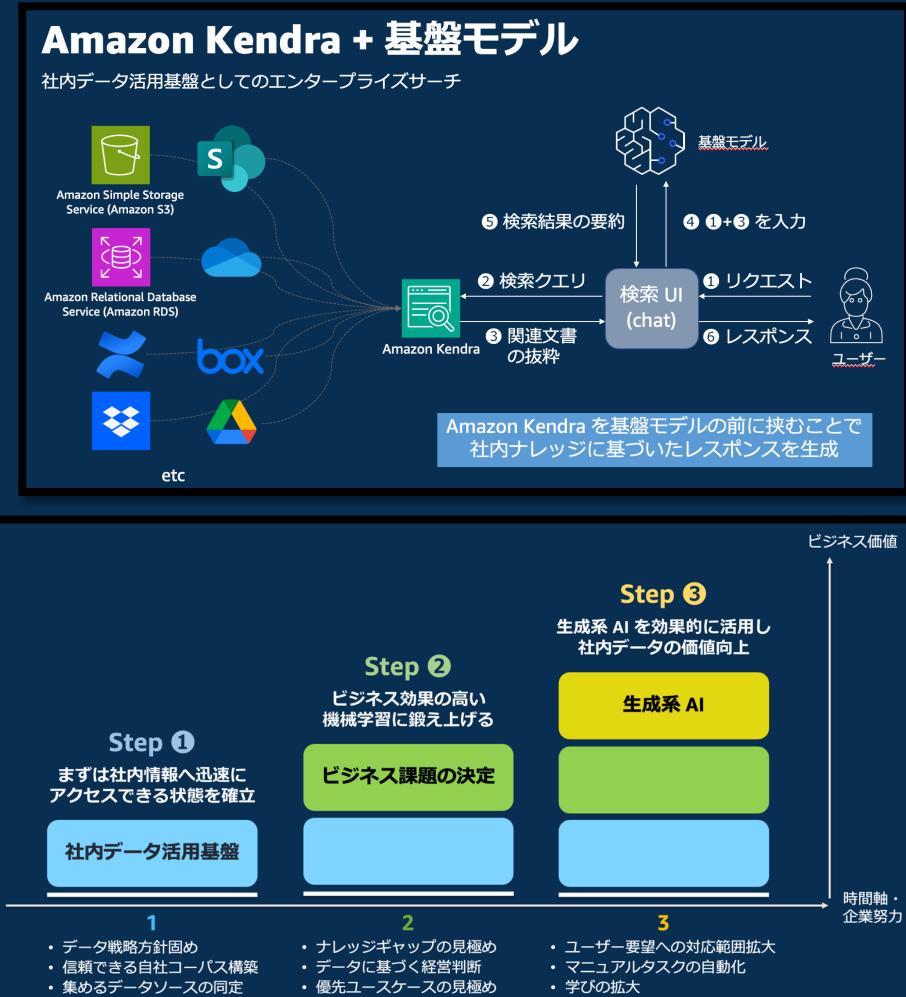
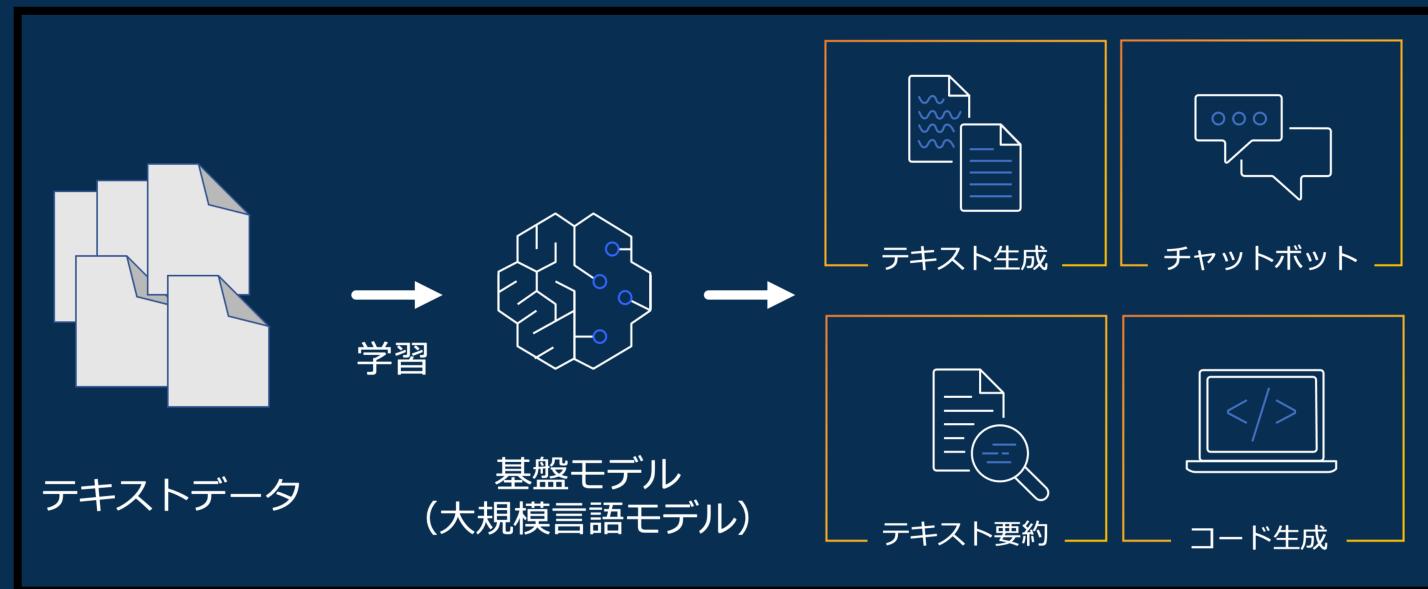


データ出典 : DX白書2023 | IPA 独立行政法人 情報処理推進機構



本セッションの概要

- 生成系 AI は何か？従来の機械学習とどう違うのか？
- AWS 上で生成系 AI を利活用していくための指針



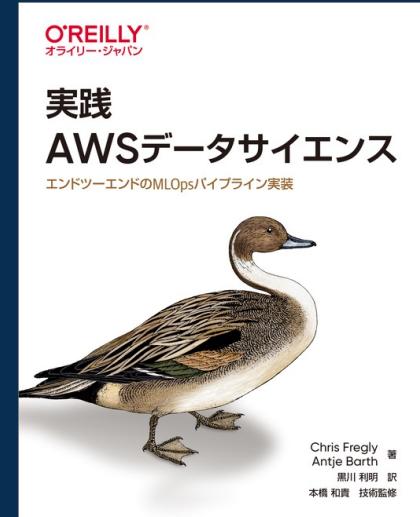
本セッションの想定視聴者像



- 最近流行っている生成系 AI がビジネスにどう活用できるか知りたい！
 - ビジネスリーダー
 - プロダクトマネージャー
- AWS の生成系 AI サービスを使ってみたい！
 - データサイエンティスト
 - アプリケーション開発者

本橋 和貴, Ph.D.

アマゾン ウェブ サービス ジャパン合同会社
機械学習パートナーソリューションアーキテクト



2020年より AWS にジョイン
AWS 上で機械学習関連のソフトウェアを開発しているパートナー企業の技術支援を担当

オライリージャパン社「実践 AWS データサイエンス」翻訳・技術監修

SNS ハッシュタグ: #AWSBuilders



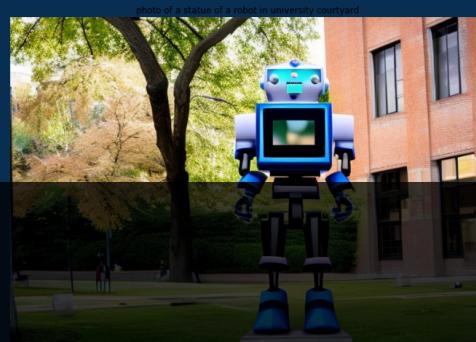
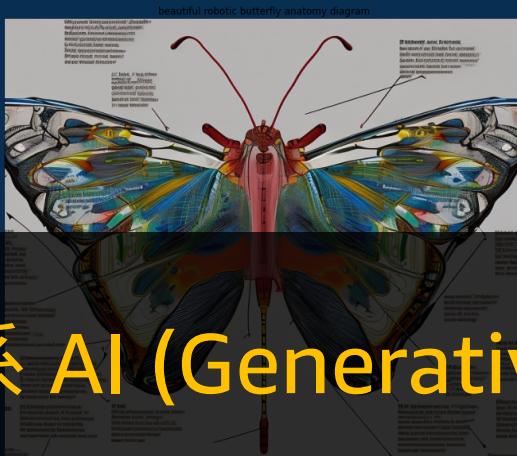
アジェンダ

1. 生成系 AI 入門
2. AWS の生成系 AI サービス
3. 生成系 AI のビジネス応用
4. まとめ

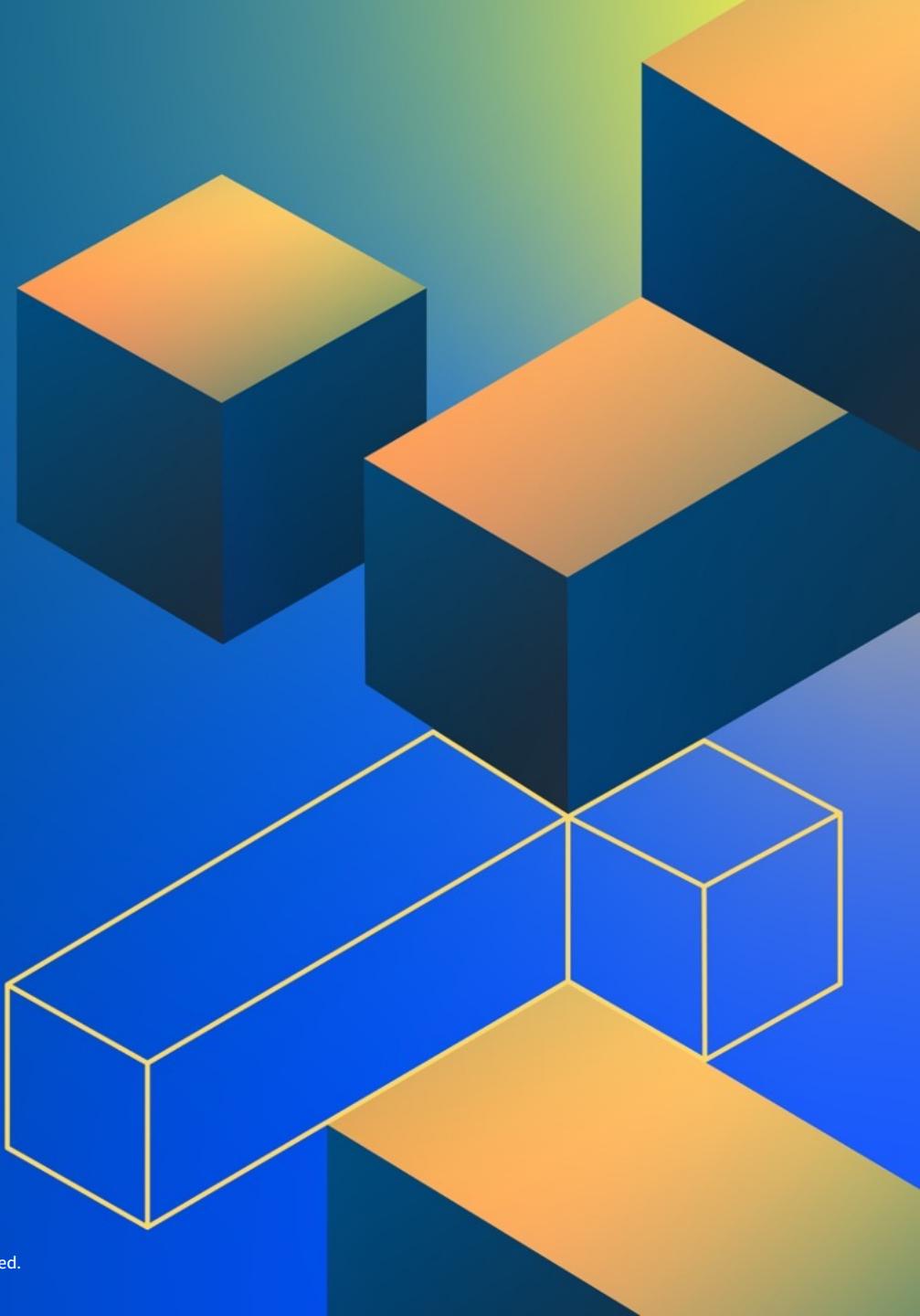


生成系 AI (Generative AI) とは？

- 画像、会話、文章、画像、音楽など、新しいコンテンツやアイデアを創造
- 一般に**基盤モデル (Foundation Model)** と呼ばれる、膨大なデータに基づいて事前にトレーニングされた大規模モデルを搭載



生成系 AI 入門



生成系 AI は深層学習の一種



人工知能 (Artificial Intelligence, AI)

人間の知的判断をコンピュータ上で実現するための技術全般
(ロジック、if-then 文、機械学習)



機械学習 (Machine Learning, ML)

AI の一種であり、知的モデルを構築するためにデータの中の傾向を学習する技術



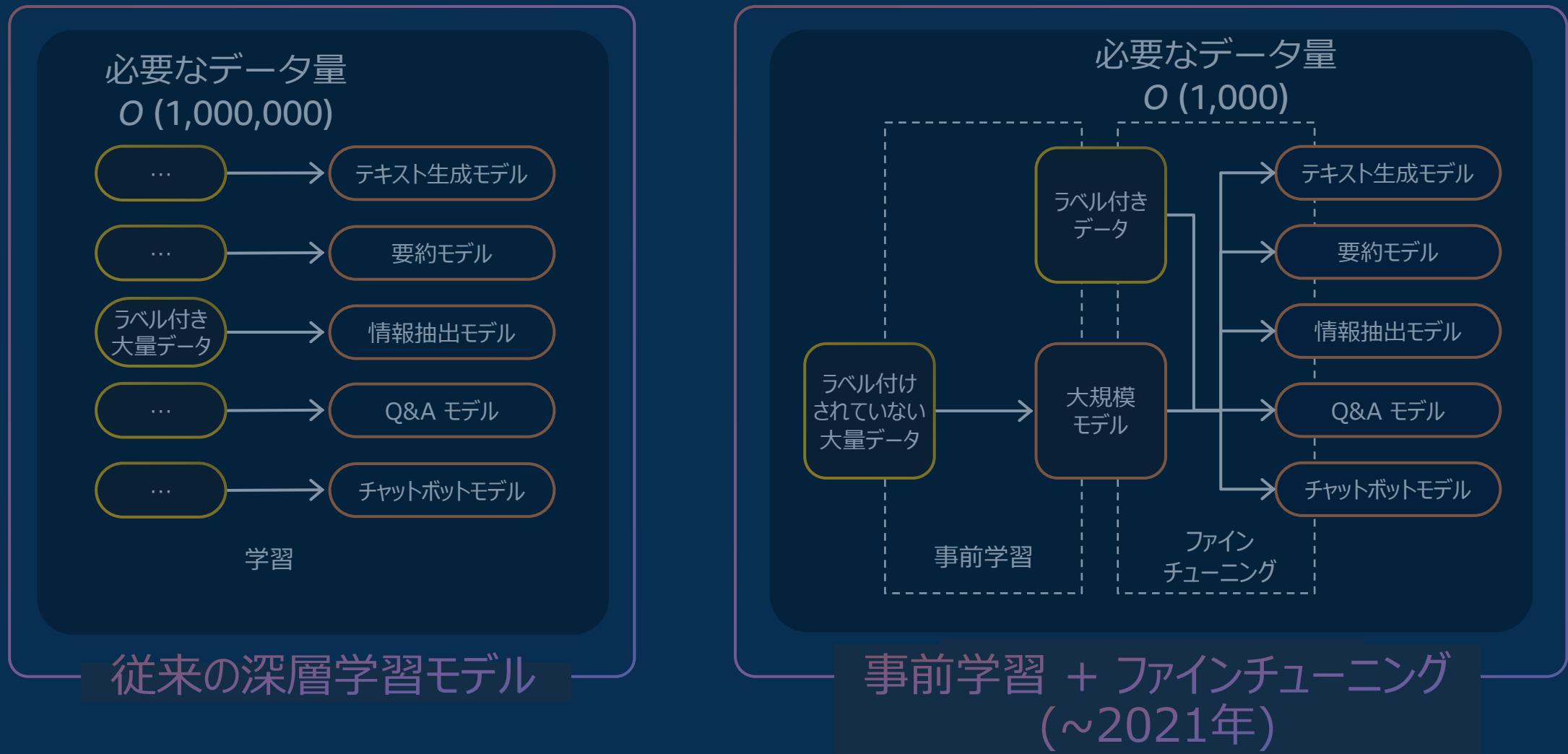
深層学習 (Deep Learning, DL)

ML の一種であり、音声・画像認識などのタスクを深い複数レイヤー構造のニューラルネットワークで実現する技術

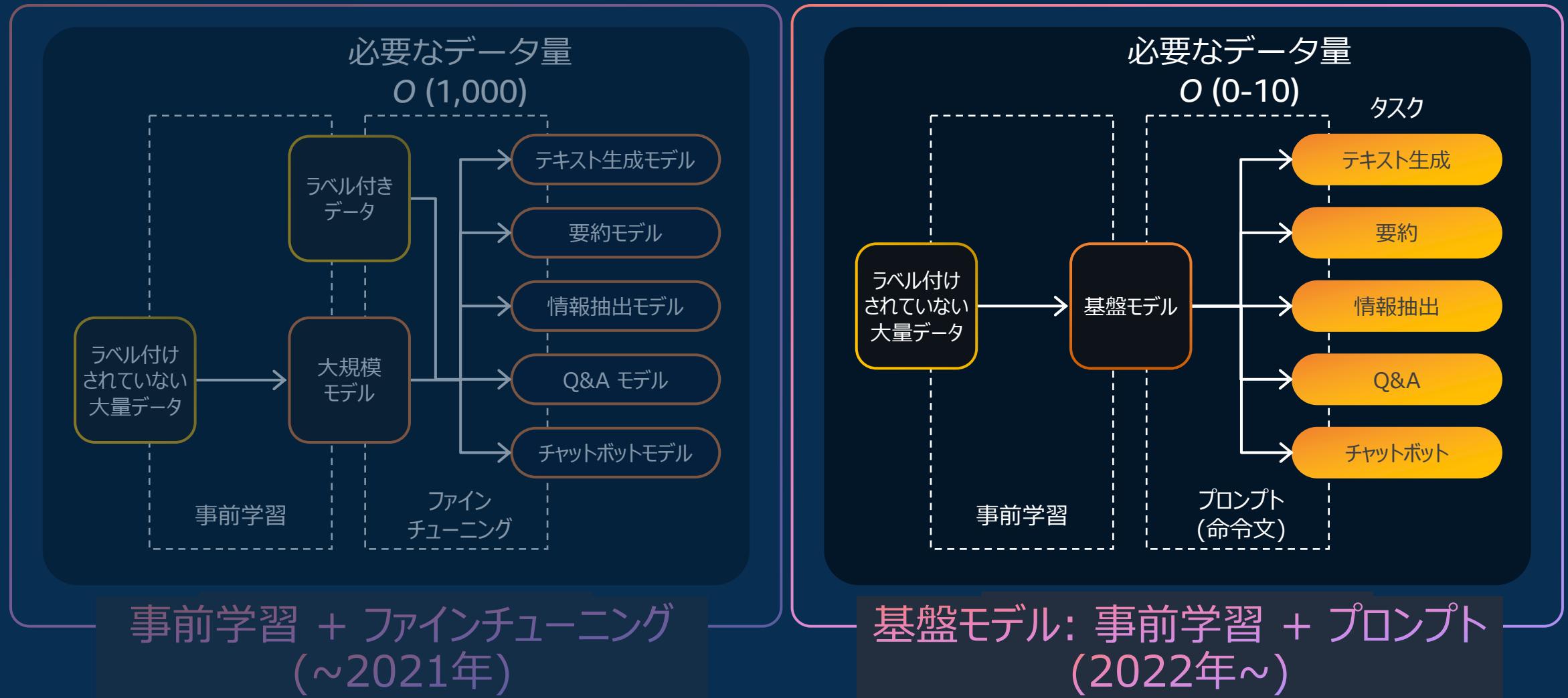
生成系 AI (Generative AI)

テラバイト規模のデータで数千億規模のパラメーターのモデルを学習することで、**追加学習なしに** 人間に近しい生成を実現する技術

基盤モデル以前の深層学習



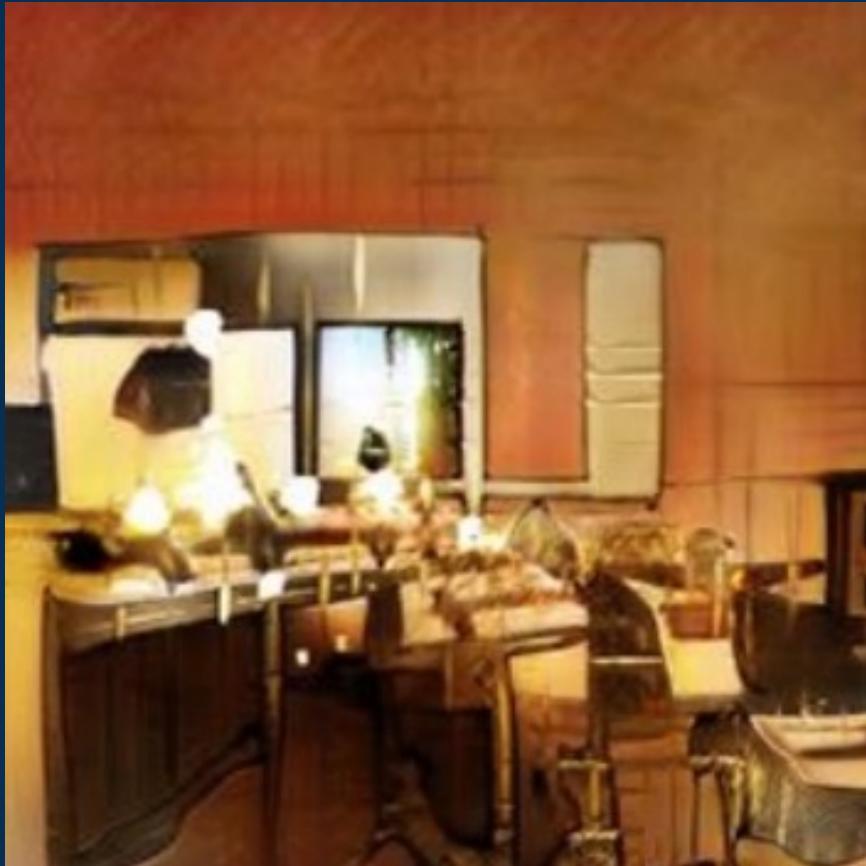
基盤モデルによるパラダイムシフト



基盤モデルの技術によりクオリティが向上



「『とても綺麗なリビングルームの写真』出して」



2017

StackGAN,
Zhang et al.



「おっけー」



2022

Stable Diffusion,
Rombach et al.

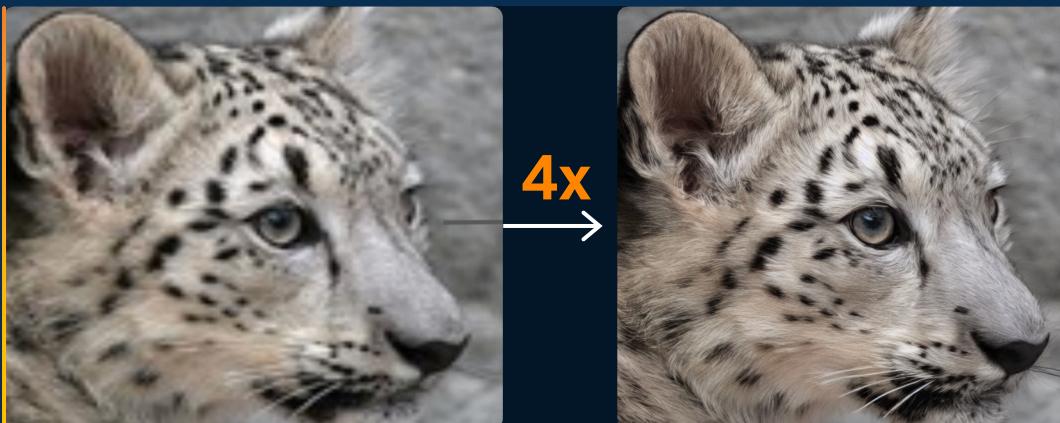
画像生成 AI の応用 – Stable Diffusion



Generated by Stable Diffusion 2.0

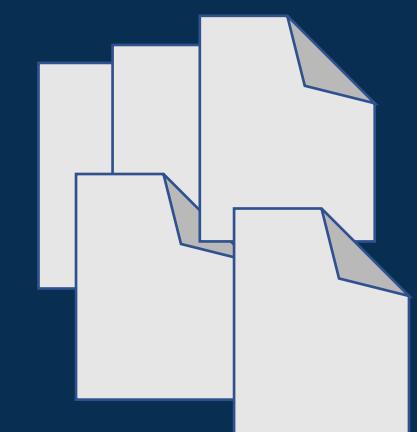


Image transformation
画像変換



Upscaling
超解像

画像生成だけではないさまざまなユースケース



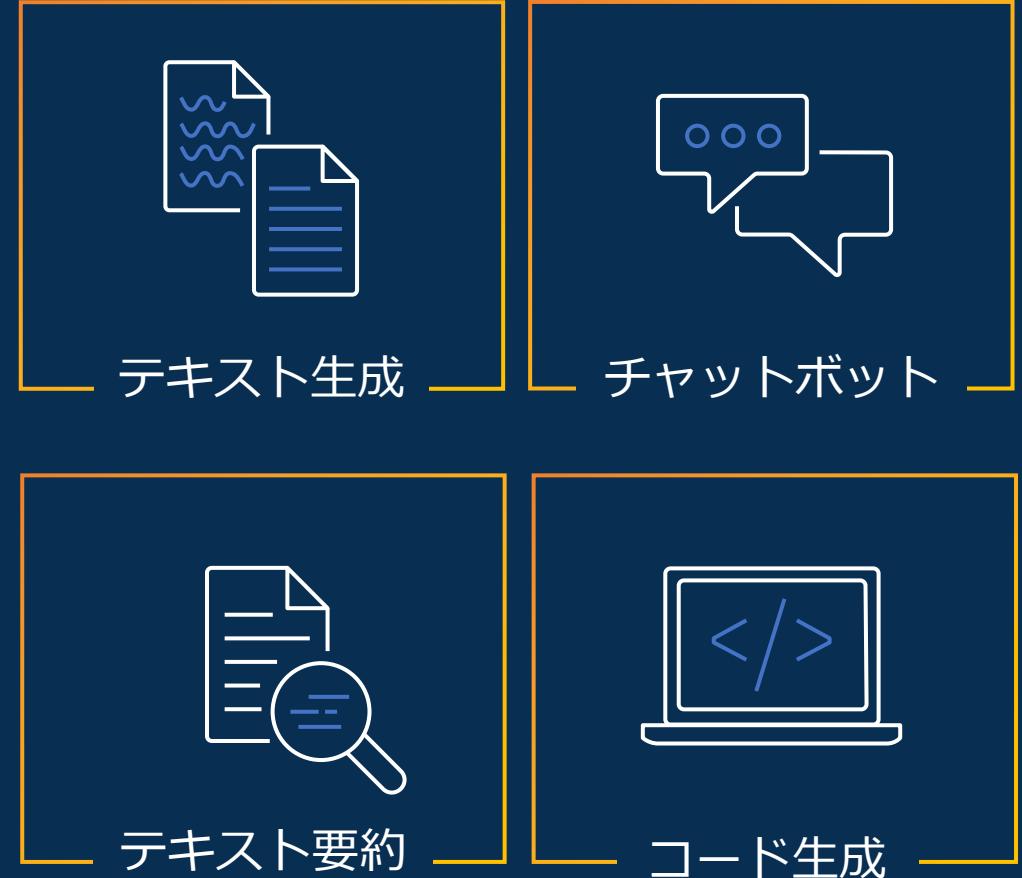
テキストデータ

→
学習

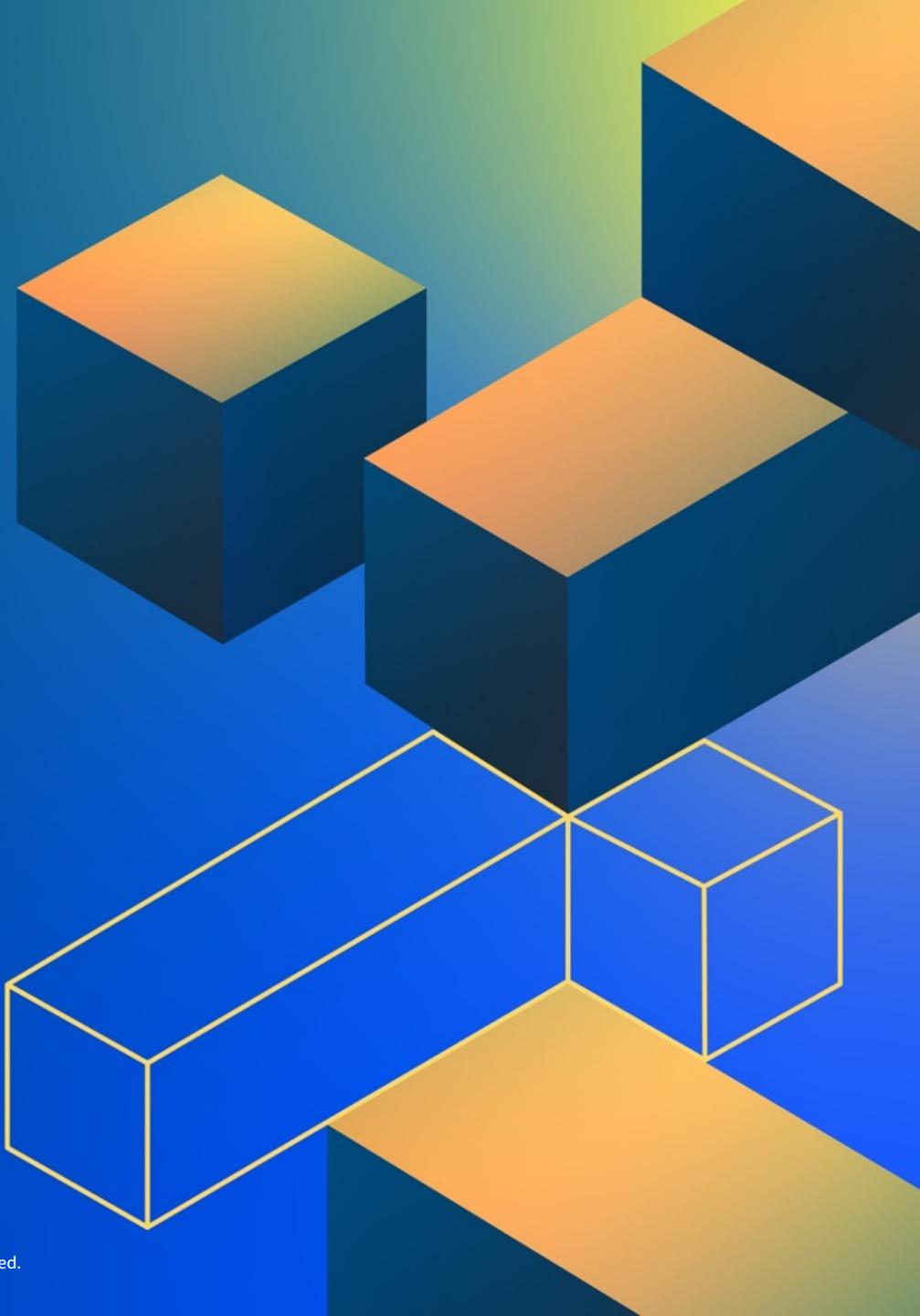


基盤モデル
(大規模言語モデル)

→



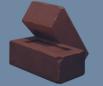
生成系 AI on AWS



これから生成系 AI を始める 4 つのアプローチ

生成系 AI
コンシューマー

① すぐに使いたい
商用の生成系 AI
サービスを利用 

② ありものに付加価値を
複数の API を
組み合わせる 

③ カスタマイズしたい
生成系 AI モデルを
チューニング 

④ 独自の優位性に積極投資
生成系 AI モデルを
独自開発 



これから生成系 AI を始める 4 つのアプローチ

	Level ① / ② 商用サービス / API 利用	Level ③ モデルチューニング	Level ④ 独自モデル開発
必要な専門知識	小	中	大
Time-to-Value	小	中	大
コスト (モデル学習)	ゼロ	中	大
コスト (推論)	大	中	中
カスタマイズ性	小	中	大
関連する AWS サービス	Amazon CodeWhisperer Amazon Bedrock (限定プレビュー)	SageMaker JumpStart	AWS Trainium AWS Inferentia



これから生成系 AI を始める 4 つのアプローチ

	Level ① / ② 商用サービス / API 利用	Level ③ モデルチューニング	Level ④ 独自モデル開発
必要な専門知識	小	中	大
Time-to-Value	小	中	大
コスト (モデル学習)	ゼロ	中	大
コスト (推論)	大	中	中
カスタマイズ性	小	中	大
関連する AWS サービス	Amazon CodeWhisperer Amazon Bedrock (限定プレビュー)	SageMaker JumpStart	AWS Trainium AWS Inferentia



これから生成系 AI を始める 4 つのアプローチ

	Level ① / ② 商用サービス / API 利用	Level ③ モデルチューニング	Level ④ 独自モデル開発
必要な専門知識	小	中	大
Time-to-Value	小	中	大
コスト (モデル学習)	ゼロ	中	大
コスト (推論)	大	中	中
カスタマイズ性	小	中	大
関連する AWS サービス	Amazon CodeWhisperer Amazon Bedrock (限定プレビュー)	SageMaker JumpStart	AWS Trainium AWS Inferentia



NEW

Amazon Bedrock

基盤モデルを使用して
生成系 AI アプリケーションを
構築・スケーリングする
最も簡単な方法

限定プレビュー



Amazon Bedrock: 複数の基盤モデルから用途に最適なものを選択

Amazon が提供



Titan Text



Titan
Embeddings

最先端スタートアップ企業が提供

AI21 Labs



Jurassic-2



Claude



Stable
Diffusion

あとでデモ

これから生成系 AI を始める 4 つのアプローチ

	Level ① / ② 商用サービス / API 利用	Level ③ モデルチューニング	Level ④ モデルプロバイダー
必要な専門知識	小	中	大
Time-to-Value	小	中	大
コスト (モデル学習)	ゼロ	中	大
コスト (推論)	大	中	中
カスタマイズ性	小	中	大
関連する AWS サービス	Amazon CodeWhisperer Amazon Bedrock (限定プレビュー)	SageMaker JumpStart	AWS Trainium AWS Inferentia



Amazon CodeWhisperer



Amazon CodeWhisperer で開発効率を向上



ソースコード自動生成 AI、個人開発者は無料で利用可能



&

Go, Rust, PHP, Ruby, Kotlin, C, C++,
Shell scripting, SQL, and Scala



&

JupyterLab, CLion, GoLand, WebStorm,
Rider, PhpStorm, RubyMine, and DataGrip



IDE がコードの
文脈情報を自動的に
CodeWhisperer へ送信

コード候補を受信



リアルタイム
コード候補サジェスト

出典の追跡

セキュリティスキャン



① リアルタイムコード候補サジェスト

```
import java.util.Map.Entry;
import java.util.stream.Collectors;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.SQSEvent;
import com.amazonaws.services.lambda.runtime.events.SQSMessageAttribute;
import com.amazonaws.services.lambda.runtime.events.SQSEvent.MessageAttribute;
import com.amazonaws.services.sqs.model.MessageAttributeValue;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.regions.Region;
/* キューの Amazon SQS に送信されたメッセージのボディーを
Amazon DynamoDB テーブルに保管する AWS Lambda の関数を書いて */
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.regions.Region;
/*Create a lambda function that stores the body of the SQS message
into a hash key of a DynamoDB table. */
public class SQSIdentityFunction implements RequestHandler<SQSEvent, String> {

    private static final String TABLE_NAME = "SQSMessage";
    private static final String HASH_KEY = "MessageId";

    private AmazonDynamoDBClientBuilder builder = AmazonDynamoDBClientBuilder.standard();
    private AmazonDynamoDB client = builder.withRegion(Regions.US_EAST_1).build();

    @Override
    public String handleRequest(SQSEvent event, Context context) {
        for (SQSMessage message : event.getRecords()) {
            String messageId = message.getMessageId();
            String messageBody = message.getBody();
            context.getLogger().log("MessageId: " + messageId + " MessageBody: " + messageBody);
            storeMessage(messageId, messageBody);
        }
        return "Success";
    }

    private void storeMessage(String messageId, String messageBody) {
        client.putItem(TABLE_NAME, HASH_KEY, messageId, messageBody);
    }
}
```

② 出典の追跡

```
Key={'url': 'http://tinyurl.com/3yjw3c8'} # return response['Item']
```

PROBLEMS 1 OUTPUT TERMINAL

CODEWHISPERER REFERENCE LOG:

```
/* Create a lambda function that stores the body of the SQS message
into a hash key of a DynamoDB table. */
public class SQSIdentityFunction implements RequestHandler<SQSEvent, String> {

    private static final String TABLE_NAME = "SQSMessage";
    private static final String HASH_KEY = "MessageId";

    private AmazonDynamoDBClientBuilder builder = AmazonDynamoDBClientBuilder.standard();
    private AmazonDynamoDB client = builder.withRegion(Regions.US_EAST_1).build();

    @Override
    public String handleRequest(SQSEvent event, Context context) {
        for (SQSMessage message : event.getRecords()) {
            String messageId = message.getMessageId();
            String messageBody = message.getBody();
            context.getLogger().log("MessageId: " + messageId + " MessageBody: " + messageBody);
            storeMessage(messageId, messageBody);
        }
        return "Success";
    }

    private void storeMessage(String messageId, String messageBody) {
        client.putItem(TABLE_NAME, HASH_KEY, messageId, messageBody);
    }
}
```

「こんなコードでどうでしょう」

- > CodeCatalyst
- > CDK
- ▽ CodeWhisperer (Preview)...
 - II Pause Auto-Suggestions
 - ▷ Run Security Scan
 - ▷ Open Code Reference L...

- Log injection: User-provided inputs must be sanitized before they are logged. An attacker can use unsanitized input to break a log's integrity, forge log entries, or bypass log monitors. Detected by CodeWhisperer
- View Problem (F8) No quick fixes available
- codewhisperer
- Showing 5 of 21
- ▽ data_aggregator.py src
 - Hardcoded cred... Detected by CodeWhisperer [Ln 12, Col 1]
 - ⚠ Log injection: Us... Detected by CodeWhisperer [Ln 39, Col 1]
 - Missing S3 buck... Detected by CodeWhisperer [Ln 46, Col 1]
 - Unrestricted upl... Detected by CodeWhisperer [Ln 58, Col 1]
 - Resource leak: A... Detected by CodeWhisperer [Ln 59, Col 1]



① リアルタイムコード候補サジェスト

```
import java.util.Map.Entry;
import java.util.stream.Collectors;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.SQSEvent;
import com.amazonaws.services.lambda.runtime.events.SQSEvent.MessageAttribute;
import com.amazonaws.services.lambda.runtime.events.SQSEvent.SQSMessage;
import com.amazonaws.services.sqs.model.MessageAttributeValue;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.regions.Regions;

/*Create a lambda function that stores the body of the SQS message
into a hash key of a DynamoDB table. */
public class SQSIdentityFunction implements RequestHandler<SQSEvent, String> {

    private static final String TABLE_NAME = "SQSMessage";
    private static final String HASH_KEY = "MessageId";

    private AmazonDynamoDBClientBuilder builder = AmazonDynamoDBClientBuilder.standard();
    private AmazonDynamoDB client = builder.withRegion(Regions.US_EAST_1).build();

    @Override
    public String handleRequest(SQSEvent event, Context context) {
        for (SQSMessage message : event.getRecords()) {
            String messageId = message.getMessageId();
            String messageBody = message.getBody();
            context.getLogger().log("MessageId: " + messageId + " MessageBody: " + messageBody);
            storeMessage(messageId, messageBody);
        }
        return "Success";
    }

    private void storeMessage(String messageId, String messageBody) {
        client.putItem(TABLE_NAME, HASH_KEY, messageId, messageBody);
    }
}
```

② 出典の追跡

```
12 |     Key={'url': ...
13 |     # return ...
|     return response['Item']
```

「今回のコード生成の元ネタは
MIT ライセンスで公開されてました」

[3/23/2023, 10:31:49 AM] Accepted recommendation with code
response = client.get_item(TABLE_NAME='tinyurl-table', Key={'url': {'S': url}})
provided with reference under **MIT** from repository [Scout](#). Added to
/Users/brejenk/Desktop/tesload/serverless-python-demo/src/put_product/app.py (line at 12).

③ セキュリティスキャン

- > CodeCatalyst
- > CDK
- ▽ CodeWhisperer (Preview)...
 - II Pause Auto-Suggestions
 - ▷ Run Security Scan
 - ▷ Open Code Reference L...

filename - Input: Enter a filename. "	
39	logger.info("Processing %s", filename)
(function) info: Any	
See Real World Examples From GitHub	
Log injection:	User-provided inputs must be sanitized before they are logged. An attacker can use unsanitized input to break a log's integrity, forge log entries, or bypass log monitors. Detected by CodeWhisperer
View Problem (F8)	No quick fixes available
codewhisperer	Showing 5 of 21
5	
▽	data_aggregator.py src
	Hardcoded cred... Detected by CodeWhisperer [Ln 12, Col 1]
⚠	Log injection: Us... Detected by CodeWhisperer [Ln 39, Col 1]
	Missing S3 buck... Detected by CodeWhisperer [Ln 46, Col 1]
	Unrestricted upl... Detected by CodeWhisperer [Ln 58, Col 1]
	Resource leak: A... Detected by CodeWhisperer [Ln 59, Col 1]



① リアルタイムコード候補サジェスト

```
import java.util.Map.Entry;
import java.util.stream.Collectors;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.SQSEvent;
import com.amazonaws.services.lambda.runtime.events.SQSEvent.MessageAttribute;
import com.amazonaws.services.lambda.runtime.events.SQSEvent.SQSMessage;
import com.amazonaws.services.sqs.model.MessageAttributeValue;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.regions.Regions;

/*Create a lambda function that stores the body of the SQS message
into a hash key of a DynamoDB table. */
public class SQSIdentityFunction implements RequestHandler<SQSEvent, String> {

    private static final String TABLE_NAME = "SQSMessage";
    private static final String HASH_KEY = "MessageId";

    private AmazonDynamoDBClientBuilder builder = AmazonDynamoDBClientBuilder.standard();
    private AmazonDynamoDB client = builder.withRegion(Regions.US_EAST_1).build();

    @Override
    public String handleRequest(SQSEvent event, Context context) {
        for (SQSMessage message : event.getRecords()) {
            String messageId = message.getMessageId();
            String messageBody = message.getBody();
            context.getLogger().log("MessageId: " + messageId + " MessageBody: " + messageBody);
            storeMessage(messageId, messageBody);
        }
        return "Success";
    }

    private void storeMessage(String messageId, String messageBody) {
        client.putItem(TABLE_NAME, HASH_KEY, messageId, messageBody);
    }
}
```

② 出典の追跡

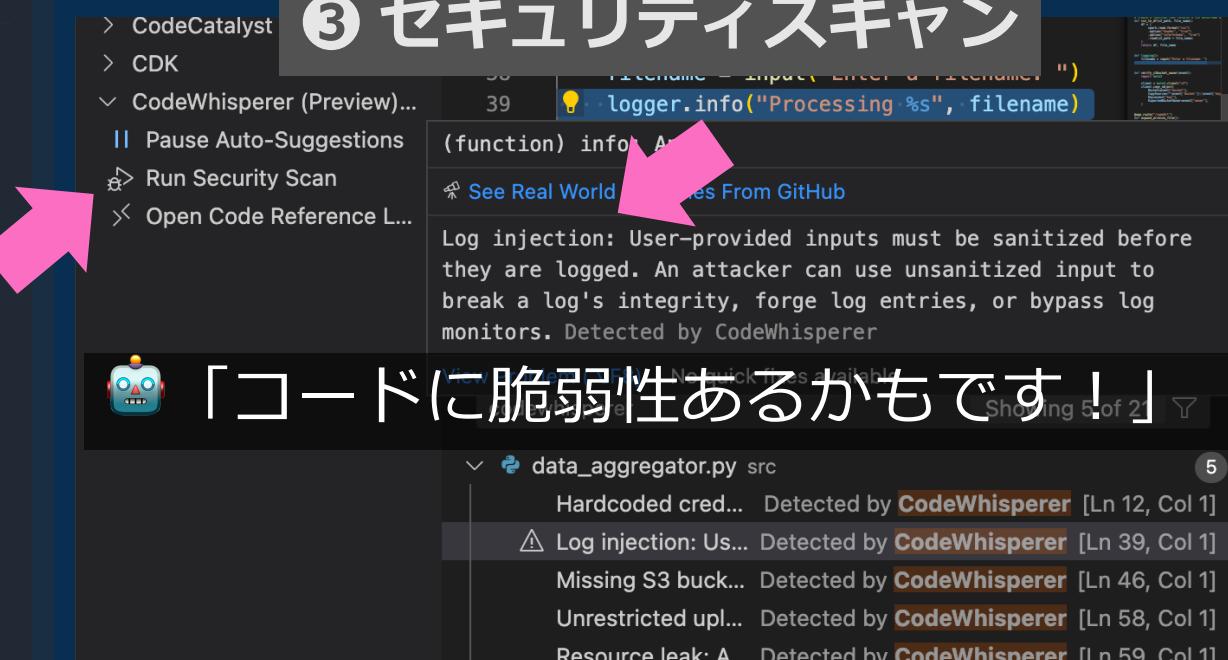
```
12 |     Key={'url': ...
13 |     # return ...
|     return response['Item']

PROBLEMS 1 OUTPUT TERMINAL

CODEWHISPERER REFERENCE LOG:
Don't want suggestions that include code with references? Uncheck this option in CodeWhisperer Settings
[3/23/2023, 10:31:49 AM] Accepted recommendation with code
response = client.get_item(TableName='tinyurl-table', Key={'url': {'S': url}})

provided with reference under MIT from repository Scout. Added to
/Users/brejenk/Desktop/tesload/serverless-python-demo/src/put_product/app.py (line at 12).
```

③ セキュリティスキャン



CodeCatalyst
CDK
CodeWhisperer (Preview)...
Pause Auto-Suggestions
Run Security Scan
Open Code Reference L...

logger.info("Processing %s", filename)

See Real World Examples From GitHub

Log injection: User-provided inputs must be sanitized before they are logged. An attacker can use unsanitized input to break a log's integrity, forge log entries, or bypass log monitors. Detected by CodeWhisperer

Hardcoded cred... Detected by CodeWhisperer [Ln 12, Col 1]
Log injection: Us... Detected by CodeWhisperer [Ln 39, Col 1]
Missing S3 buck... Detected by CodeWhisperer [Ln 46, Col 1]
Unrestricted upl... Detected by CodeWhisperer [Ln 58, Col 1]
Resource leak: A... Detected by CodeWhisperer [Ln 59, Col 1]

「コードに脆弱性あるかもです！」



① リアルタイムコード候補サジェスト

```
J sc
12 Key={'url': {'S': 'http://tinyurl.com/4h9lqjy'}} ② 出典の追跡
13 # return response['Item']
return response['Item']

PROBLEMS 1 OUTPUT TERMINAL
CODEWHISPERER REFERENCE LOG:
Don't want suggestions that include code with references? Uncheck this option in CodeWhisperer Settings
[3/23/2023, 10:31:49 AM] Accepted recommendation with code
response = client.get_item(TableName='tinyurl-table', Key={'url': {'S': 'http://tinyurl.com/4h9lqjy'}})
provided with reference under MIT from repository Scout. Added to
/Users/brejenk/Desktop/tesload/serverless-python-demo/src/put_product/app.py (line at 12).

12 Key={'url': {'S': 'http://tinyurl.com/4h9lqjy'}} ③ セキュリティスキャン
13 # return response['Item']

PROBLEMS 1 OUTPUT TERMINAL
CODEWHISPERER REFERENCE LOG:
CodeCatalyst
CDK
CodeWhisperer (Preview)...
II Pause Auto-Suggestions
Run Security Scan
Open Code Reference L...
(function) info: Any
See Real World Examples From GitHub
Log injection: User-provided inputs must be sanitized before they are logged. An attacker can use unsanitized input to break a log's integrity, forge log entries, or bypass log monitors. Detected by CodeWhisperer
View Problem (F8) No quick fixes available
codewhisperer
Showing 5 of 21
data_aggregator.py src
Hardcoded cred... Detected by CodeWhisperer [Ln 12, Col 1]
⚠ Log injection: Us... Detected by CodeWhisperer [Ln 39, Col 1]
Missing S3 buck... Detected by CodeWhisperer [Ln 46, Col 1]
Unrestricted upl... Detected by CodeWhisperer [Ln 58, Col 1]
Resource leak: A... Detected by CodeWhisperer [Ln 59, Col 1]
```



Amazon SageMaker JumpStart



Amazon SageMaker JumpStart

既存の基盤モデルに
アクセスして開発・
運用を始める

Products / Machine Learning / Amazon SageMaker JumpStart

Getting started with Amazon SageMaker JumpStart

Amazon SageMaker JumpStart is a machine learning (ML) hub that can help you accelerate your ML journey. Explore how you can get started with built-in algorithms with pretrained models from model hubs, pretrained foundation models, and prebuilt solutions to solve common use cases. To get started, see documentation or example notebooks that you can quickly execute.

Reset Filters

Product Type

Text Tasks

- End-to-end Solution
- Text Classification
- Text Embedding
- Text Generation
- Text Summarization
- Named Entity Recognition
- Question Answering

Sort By Popularity

foundation models

FOUNDATION MODEL PREVIEW

Text Generation Proprietary Models Various Providers Models from AI21 Labs, Cohere, and LightOn in preview. Sign-up for preview with JumpStart in us-east-1 or eu-west-1 SageMaker Console.

FOUNDATION MODEL FEATURED

Text to Image stability.ai

Stable Diffusion 2 Stabilityai Model ID: model-txt2img-stabilityai-stable-diffusion-v2. This is a text-to-image model from Stability AI and downloaded from HuggingFace. It takes a textual description as Fine-tunable Deploy Only

Text Generation Alexa alexa

AlexaTM (20b) Pytorch Model ID: pytorch-textgeneration1-alexatm20b. AlexaTM 20B is a multitask, multilingual, large-scale sequence-to-sequence (seq2seq) model, trained on a mixture of CommonCrawl Deploy Only

Text Generation Bloom 1b7 Huggingface Model ID: huggingface-textgeneration-bloom-1b7. This is a Text Generation model built upon a Transformer model from Hugging Face. It takes a text string as input and predicts next words in the sequence. This model has BigScience Responsible AI License v1.0. Please read the [terms] (<https://huggingface.co/spaces/bloom-1b7>) Deploy Only

The screenshot shows the Amazon SageMaker JumpStart interface. At the top, there's a search bar with 'foundation models' typed in. Below it, there are filters for 'Product Type' (set to 'Text Tasks') and 'Sort By' (set to 'Popularity'). A dashed orange arrow points from the 'foundation models' search term down to a highlighted section of the interface. This section displays a list of foundation models, each with a thumbnail, name, provider, and a brief description. The models shown are: 'Text Generation Proprietary Models' (stability.ai), 'Text to Image' (stability.ai), 'Text Generation' (Alexa), 'Text Generation' (Bloom 1b7), and 'Text Generation' (LightOn). Each model entry includes a 'PREVIEW' button, a 'FEATURED' badge, and a 'Deploy Only' button.

SageMaker JumpStart



Bloom 560m

By Hugging Face | Ver 1.3

TEXT GENERATION, INFORMATION
EXTRACTION, QUESTION ANSWERING,
SUMMARIZATION

BigScience Large Open-science Open-access
Multilingual Language Model

[View model](#)



Bloom 1b1

By Hugging Face | Ver 1.3

TEXT GENERATION, INFORMATION
EXTRACTION, QUESTION ANSWERING,
SUMMARIZATION

BigScience Large Open-science Open-access
Multilingual Language Model

[View model](#)



Bloom 1b7

By Hugging Face | Ver 1.3

TEXT GENERATION, INFORMATION
EXTRACTION, QUESTION ANSWERING,
SUMMARIZATION

BigScience Large Open-science Open-access
Multilingual Language Model

[View model](#)



Stable Diffusion 1.4

By StabilityAI | Ver 1.4

IMAGE GENERATION (FROM TEXT)

Stable Diffusion v1.4 is a latent text-to-image
diffusion model capable of generating photo-
realistic images given any text input.

[View model](#)



Stable Diffusion 2.1 base

By StabilityAI | Ver 2.1

IMAGE GENERATION (FROM TEXT)

Stable Diffusion v2.1 greatly improves the quality
of the generated images compared to earlier V1
releases.

[View model](#)

Amazon models



AlexaTM 20B

By Alexa

MACHINE TRANSLATION, QUESTION AND
ANSWERING, SUMMARIZATION, ANNOTATED,
DATA GENERATION

Alexa Teacher Model 20B parameter seq2seq is an
encoder-decoder multilingual large scale language
model. AlexaTM 20B can learn new tasks from just
a handful of examples (few-shot learning).

[View model](#)



Stable Diffusion 2.1 base
By StabilityAI

Try a product demo of the capabilities of this model from StabilityAI. Do not upload any confidential or sensitive information. Use of this feature is for demonstration purposes only. This demo may not accurately represent the actual response times of the product.

Prompt

cyberpunk forest by Salvador Dali

Negative prompt (optional)

trees, green

Generate image

Output



[Download Image](#)

Example

Artist style

▼

General info

Width

51

Height

51

Number of inference
steps

50

Guidance scale

7

Seed

50

SageMaker JumpStart で利用可能* な基盤モデル

*2023年5月現在

OSS 公開モデル

stability.ai



モデル

Text2Image
Upscaling

タスク

テキスト入力から
写真のようにリアル
な画像を生成
画像の鮮明化

特徴

SD 2.1 モデルは
ファインチューニング
可能

モデル

AlexaTM
20B

タスク

機械翻訳
質問応答
要約
アノテーション
データ生成



モデル

Flan T-5 models
(8 variants)

DistilGPT2,
GPT2

Bloom models
(3 variants)

タスク

機械翻訳
質問応答
要約
アノテーション
データ生成

プロプライエタリ (非公開) モデル

co:here



AI21labs

モデル

Cohere
Command-Light

タスク

テキスト生成
情報抽出
質問応答
要約

モデル

Lyra-Fr
10B

タスク

テキスト生成
キーワード抽出
情報抽出
質問応答
要約
感情分析
テキスト分類

モデル

Jurassic-2
Grande 17B

タスク

テキスト生成
長文生成
要約
パラフレーズ
チャット
情報抽出
質問応答
テキスト分類



なんか似たような名前のモデルが複数あるけど...🤔

▼ JumpStart

- Foundation models NEW
- Computer vision models
- Natural language processing models
- ▶ Governance
- ▶ Ground Truth
- ▶ Notebook
- ▶ Processing
- ▶ Training
- ▶ Inference
- ▶ Edge Manager
- ▶ Augmented AI

AI21 labs

AI21 Jurassic-2 Grande
By AI21 Labs | Ver 1.0.043

PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).

Best-in-class large language model with optimal quality-latency trade-off. Ideal for generating text using plain English.

[View model](#)

AI21 labs

AI21 Jurassic-2 Grande Instruct
By AI21 Labs | Ver 1.1.043

PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).

Best-in-class instruction following model with optimal quality-latency trade-off. Ideal for generating text using plain instructions.

[View model](#)



大規模言語モデルとインストラクションチューニング

「あの鐘を鳴らすのは？」 →



言語モデル



「わたし」 20%

「あなた」 70%

「アキ子」 10%

次の単語を予測できるよう事前学習するだけでなく、
多数のタスクのデータセット（インストラクション）で微調整する

→ 例示なし (zero-shot) での生成能力が向上

AI21 Lab: Jurassic-2 Grande モデルの場合

Write a tweet about the future of NLP:

(自然言語処理の未来についてのツイートを書いてください)

インストラクションチューニングなし

We want to know what you think NLP will be like in 10 years.
Send a tweet to @iNLP Center or @iNLPTech with the hashtag
#NLPFuture: Tell us what NLP will be like in 10 years.

Tag one friend to do the same: The person you tag will be
entered into a drawing to win a free gift.

1. Tweet: 10 years from now, what will NLP be like?
2. Tag a friend: The person you tag will be entered into a
drawing to win a free gift.
3. Tweet: I tagged @JohnSmith.
4. Tag another friend: The person you tag will be entered into
a drawing to win a free gift.
5. Tweet: I tagged @JaneDoe.

(あまり意味をなしていない文章)

インストラクションチューニングあり

The future of NLP is bright. Using NLP, we can train
machines to understand language, reason about
knowledge, and interact with humans naturally. #END

(自然言語処理の未来は明るいでしょう。自然言語処理を
用いると、言語を理解し、知識に基づく推論を行い、人
間と自然に対話する機械を育てることができます。)

Tips : 新しいモデルが登場したときはどのように
学習されたものなのか確認しましょう！



基盤モデルの選定基準

- ・ まず深呼吸して解きたい課題が何か考える
- ・ 複数の軸のトレードオフを評価する
 - ・ 解きたいタスクに対する**性能**
 - ・ 推論の**スピード** / レイテンシー
 - ・ **コスト**
 - ・ フайнチューニング可能か
 - ・ 対応**言語**は何か
 - ・ **セキュリティ**
 - ・ データは社内環境に閉じるか
 - ・ **ライセンス**
 - ・ 商用利用可能か / 研究利用のみか
- ・ HELM の評価を参考にする



HELM Models Scenarios Results Raw runs

Core scenarios

The scenarios where we evaluate all the models.

[[Accuracy](#) | [Calibration](#) | [Robustness](#) | [Fairness](#) | [Efficiency](#) | [General information](#)]

Accuracy

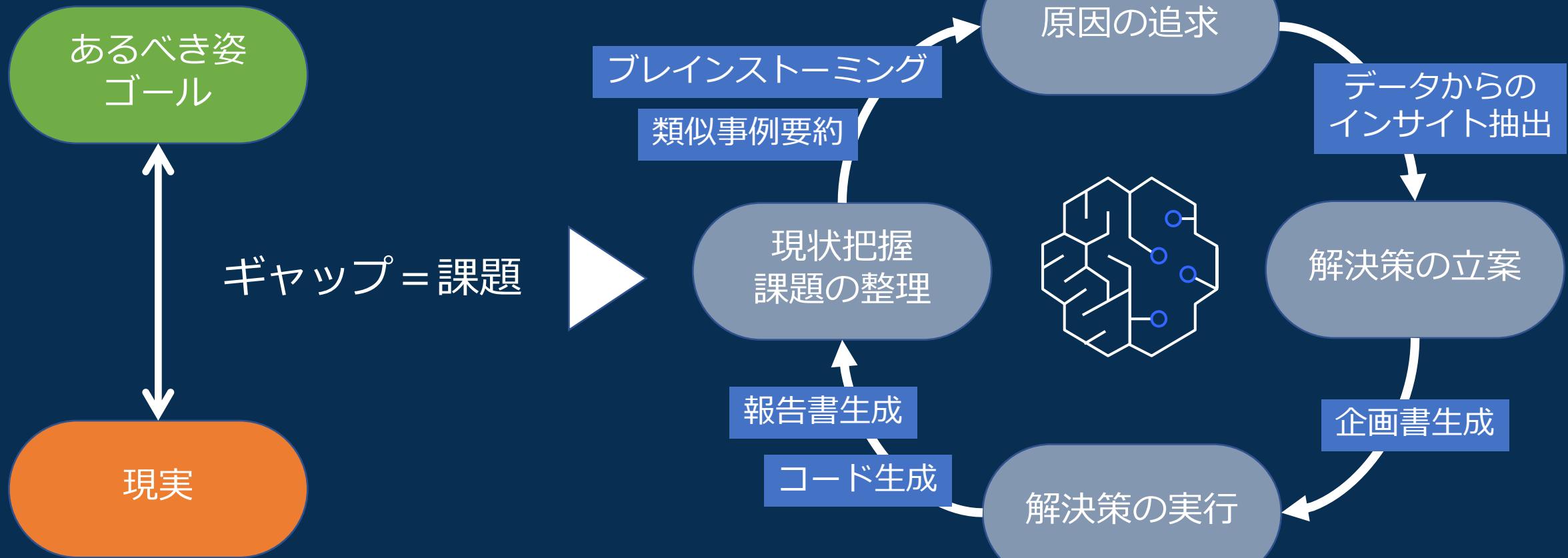
Model/adapter	Mean win rate ↑ [sort]	MMLU - EM ↑ [sort]	BoolQ - EM ↑ [sort]	NarrativeQA - F1 ↑ [sort]	NaturalG (closed- F1 ↑ [sort]
Cohere Command beta (52.4B)	0.93	0.452	0.856	0.752	0.372
text-davinci- 002	0.93	0.568	0.877	0.727	0.383
text-davinci- 003	0.898	0.569	0.881	0.727	0.406
TNLG v2 (530B)	0.855	0.469	0.809	0.722	0.384
Anthropic-LM v4-s3 (52B)	0.842	0.481	0.815	0.728	0.288
J1-Grande v2 beta (17B)	0.806	0.445	0.812	0.725	0.337



生成系 AI のビジネス応用

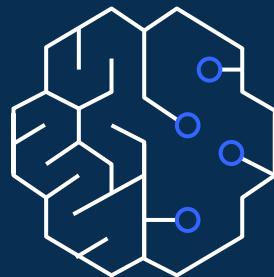


問題解決サイクルと生成系 AI



基盤モデルですべて解決、ではありません

- 基盤モデルは一般的な意味理解はできるものの、苦手領域もある
 - 数値計算
 - 複雑な論理の理解
 - 事実とは異なる内容を出力することがある
 - 基盤モデルはデータベースではありません
 - **社内に閉じたナレッジ**
 - 大規模モデルゆえの**推論コスト**



- + 計算ツール
- + Web 検索
- + エンタープライズサーチ

基盤モデルと外部ツールと組み合わせて苦手領域をカバーするアプローチがトレンド

Amazon Kendra + 基盤モデル

Amazon Kendra エンタープライズサーチ



Amazon Kendra search

Q Amazon Kendraって何ですか？

X

Frequently asked questions

Amazon Kendra とは何ですか?

Amazon Kendra は、機械学習 (ML) を利用する高精度で使いやすいエンタープライズ検索サービスです。デベロッパーはアプリケーションに検索機能を追加できます。これにより、その企業全体に散在する膨大な量のコンテンツ内に保存されている情報をエンドユーザーが見つけられるようになります。これには、マニュアル、調査報告書、よくある質問、人事 (HR) 関連ドキュメント、カスタマーサービスガイドのデータが含まれます。Amazon Simple Storage Service (S3)、Microsoft SharePoint、Salesforce、ServiceNow、RDS データベース、Microsoft OneDrive などの様々なシステムに存在している場合があります。質問が入力されると、このサービスは機械学習アルゴリズムを使用してその内容を理解し、質問の直接の回答であれ、ドキュメント全体であれ、最も適切な回答を返します。例えば、「企業クレジットカードのキャッシュバック率はどれくらいですか?」といった質問をすることができ、Amazon Kendra は関連するドキュメントにマッピングして具体的な回答（「2% です」など）を返します。Kendra はサンプルコードを提供するため、ユーザーは迅速に使用を開始し、新規または既存のアプリケーションに極めて正確な検索を簡単に統合できます。



Amazon Kendra はどの言語をサポートしていますか?

>

Amazon Kendra にはどのようなタイプの質問ができますか?

>

Amazon Kendra はどのリージョンで利用できますか?

>

Amazon Kendra + テキスト要約 AI 事実に基づく自然なチャットボット



Amazon Kendraって何ですか？



Amazon Kendraは、企業が大量の非構造化テキストデータ(PDF、Wordドキュメント、PowerPointプレゼンテーションなど)から知識を抽出し、従業員が必要な情報をすばやく見つけるのに役立つAI対応のエンタープライズサーチサービスです。

Kendraは機械学習と自然言語処理を使用して、テキスト内の関連性と意味を理解します。これにより、ユーザーは複雑な質問に対して正確でコンテキストに応じた回答を得ることができます。

Kendraは高度にカスタマイズ可能で、企業のニーズとデータに合わせて最適化できます。Kendraはオンデマンドでスケーリングできるので、テキストデータの量が増えてもパフォーマンスが低下することはありません。



どんなファイルタイプがサポートされますか？



Amazon Kendraは、次のファイル形式をサポートしています:

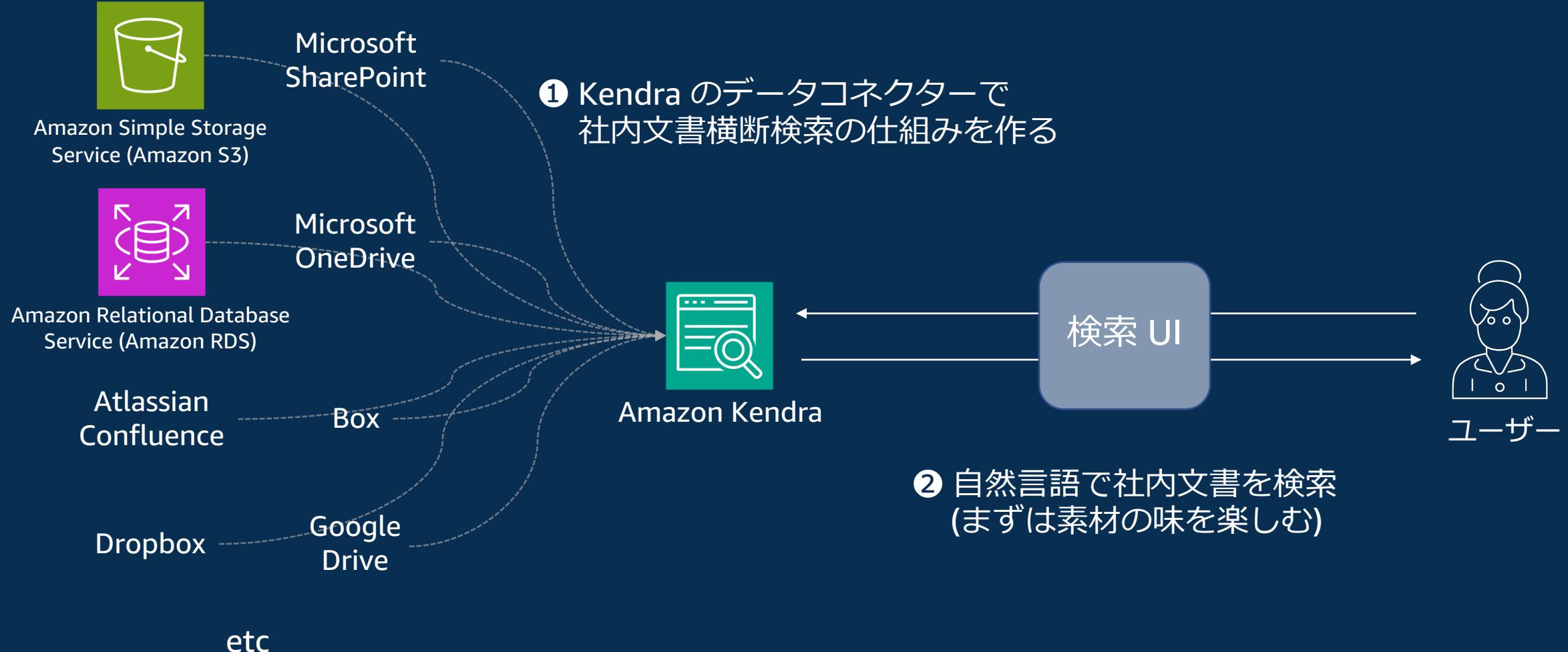
- Microsoft Office(Word、Excel、PowerPoint) • PDF • HTML • XML • JSON • CSV • TXT

これらのファイル形式は、Amazon Kendraインデックスの構築に使用できます。Amazon Kendraは、これらのファイルからテキストとメタデータを抽出し、インデックス内のドキュメントとして保存します。



Amazon Kendra + 基盤モデル

社内データ活用基盤としてのエンタープライズサーチ



Amazon Kendra + 基盤モデル

社内データ活用基盤としてのエンタープライズサーチ



Amazon Kendra + Anthropic Claude デモ

Amazon Kendra 概要 特徴 コネクタ 料金 リソース よくある質問 お客様

製品 / Machine Learning / Amazon Kendra

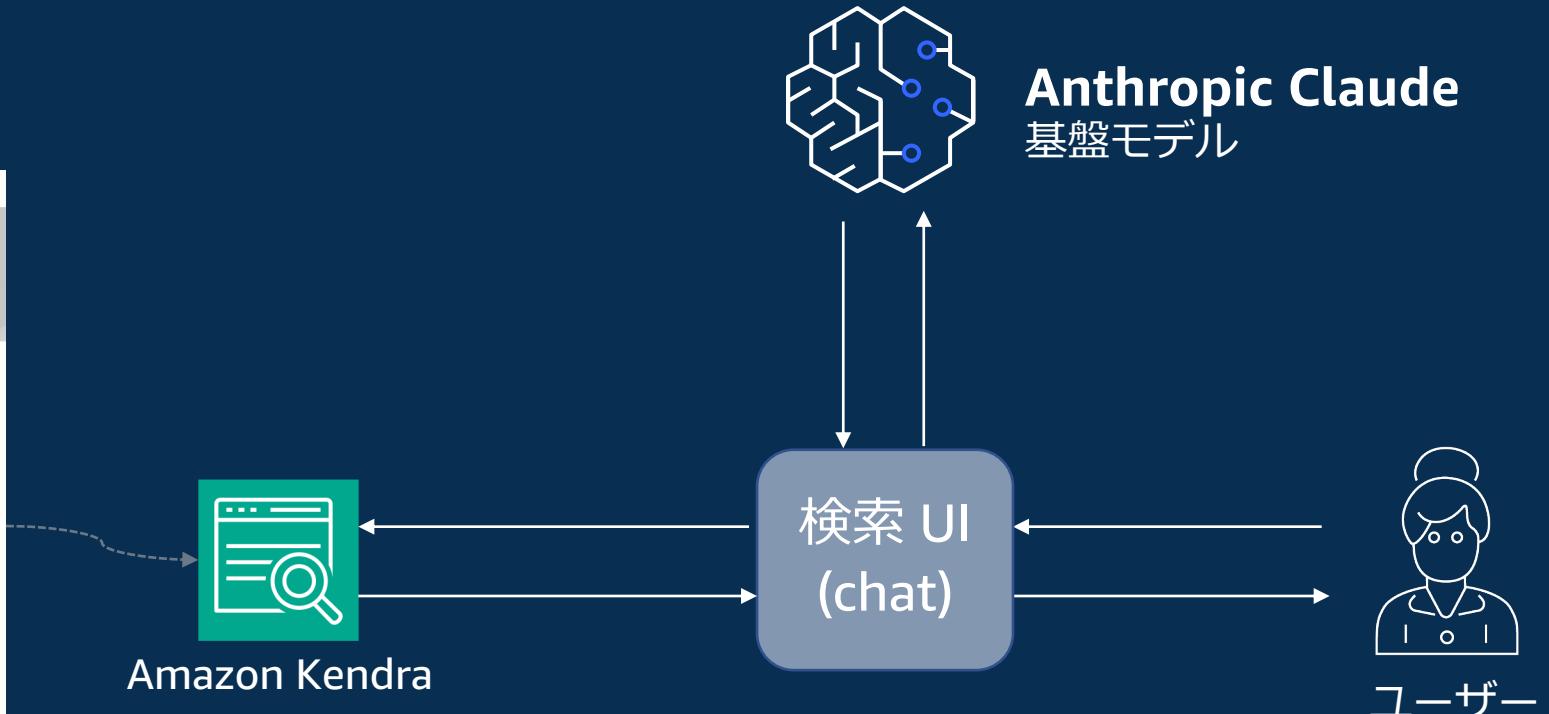
Amazon Kendra のよくある質問

Q: Amazon Kendra とは何ですか?

Amazon Kendra は、機械学習 (ML) を利用する高精度で使いやすいエンタープライズ検索サービスです。デベロッパーはアプリケーションに検索機能を追加できます。これにより、その企業全体に散在する膨大な量のコンテンツ内に保存されている情報をエンドユーザーが見つけられるようになります。これには、マニュアル、調査報告書、よくある質問、人事 (HR) 関連ドキュメント、カスタマーサービスガイドのデータが含まれます。Amazon Simple Storage Service (S3)、Microsoft SharePoint、Salesforce、ServiceNow、RDS データベース、Microsoft OneDriveなどの様々なシステムに存在している場合があります。質問が入力されると、このサービスは機械学習アルゴリズムを使用してその内容を理解し、質問の直接の回答であれ、ドキュメント全体であれ、最も適切な回答を返します。例え、「企業クレジットカードのキャッシュバック率はどれくらいですか?」といった質問をすることができ、Amazon Kendra は関連するドキュメントにマッチングして具体的な回答（「2% です」など）を返します。Kendra はサンプルコードを提供するため、ユーザーは迅速に使用を開始し、新規または既存のアプリケーションに極めて正確な検索を簡単に統合できます。

Q: Amazon Kendra は他の AWS のサービスとどのように連携しますか?

Amazon Kendra は、お客様が AWS に保存するすべての非構造化データについて、機械学習を利



AWS サービスの FAQ





生成系 AI アプリケーション powered by Amazon Kendra & Anthropic!

Clear Chat

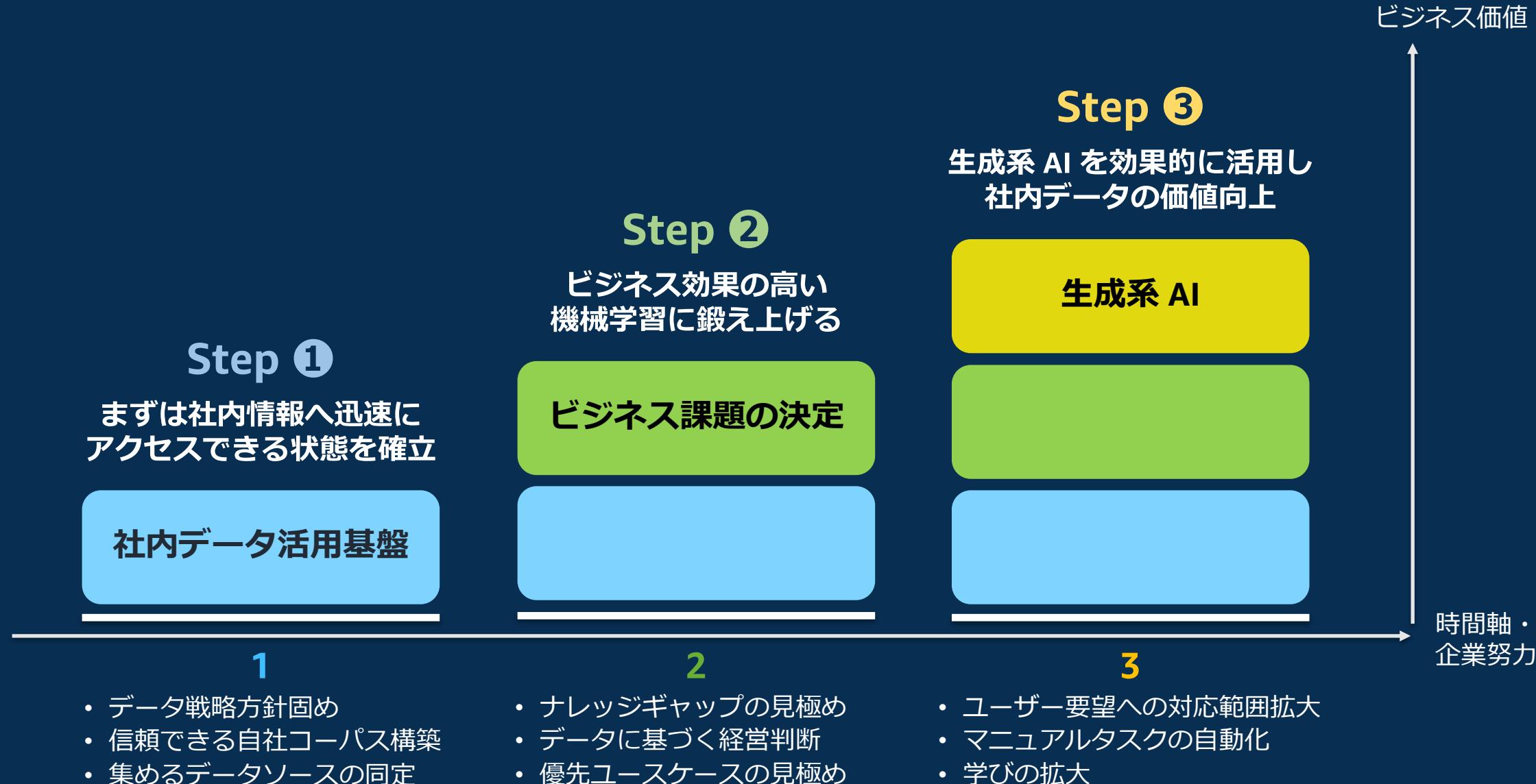
AI に質問してみましょう 😊

I

まとめ



生成系 AI の社内活用のためのロードマップを策定



今後に向けて

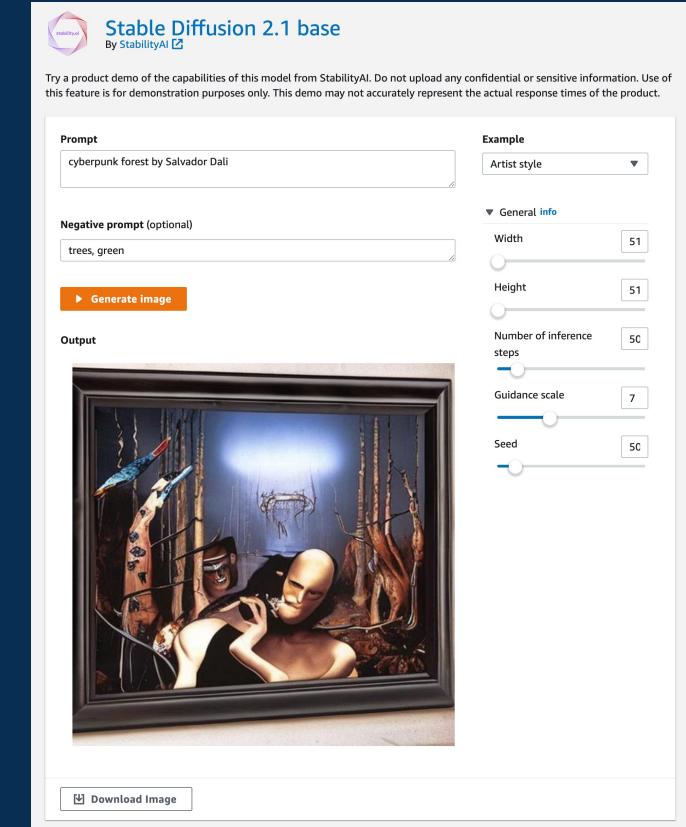
- まずは、Amazon CodeWhisperer のコード生成、
- そして、Amazon SageMaker JumpStart の基盤モデルのプレイグラウンドをお試しください

【ブログ】

- 高精度な生成系 AI アプリケーションを Amazon Kendra、LangChain、大規模言語モデルを使って作る

【ハンズオン】

- Amazon Kendra で簡単に検索システムを作ってみよう！



AWS TRAINING & CERTIFICATION

600+ ある AWS Skill Builder の無料デジタルコースで学ぼう

30 以上の AWS ソリューションの中から、自分にもっとも関係のあるクラウドスキルとサービスにフォーカスし、自習用のデジタル学習プランとランプアップガイドで学ぶことができます。

自分に合ったスキルアップ方法で学ぼう

[EXPLORE.SKILLBUILDER.AWS »](https://explore.skillbuilder.aws)



あなたのクラウドスキルを AWS 認定で証明しよう

業界で認められた資格を取得して、スキルアップの一歩を踏み出しましょう。AWS Certified Cloud Practitioner の取得方法と、準備に役立つ AWS のリソースをご覧ください。

[受験準備のためのリソースにアクセスしよう »](#)



AWS Builders Online Series に ご参加いただきありがとうございます

楽しんでいただけましたか? ぜひアンケートにご協力ください。
本日のイベントに関するご意見/ご感想や今後のイベントについての
ご希望や改善のご提案などがございましたら、ぜひお聞かせください。



aws-apj-marketing@amazon.com



twitter.com/awscloud_jp



facebook.com/600986860012140



<https://www.youtube.com/user/AmazonWebServicesJP>



<https://www.linkedin.com/showcase/aws-careers/>



twitch.tv/aws



Thank you!



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