# Agentic AI勉強会

## Amazon Bedrock 入門

第3回（全7回）　｜　2026年05月14日

Q:「AI AgentとAgentic AIって何か違いがありますか。それともここの定義は曖昧ですか」

情報科学的観点でいう **エージェント agent** とは、

“環境を観測し、その観測に基づいて行動を選び、環境に働きかける計算主体”

ChatGPT says:

たとえば、迷路を解く強化学習エージェントは AI agent です。しかし、それを通常 Agentic AI とはあまり呼びません。なぜなら、現在の Agentic AI という言葉は、LLMを使って複数の業務ステップを自律的に進め、外部ツールを使い、必要に応じて結果を見ながら計画を修正するようなシステムを指すことが多いからです。

逆に、ChatGPTのようなLLMが単に文章を返しているだけなら、それは生成AIではありますが、強い意味での Agentic AI ではありません。そこに、検索、ファイル読解、API実行、カレンダー確認、メール下書き、コード実行、反復的な計画修正などが加わると、Agentic AI に近づきます。

## AI agent

- 観測し、判断し、行動する主体
- 単体で成立
- “古典的”AIも含む

## Agentic AI ≒ “architecture of AI agents”

- 生成AI、LLM利用を基盤として、計画、ツール利用、外部システム操作、記憶、振り返りを組み合わせたシステム全体の性質・設計思想
- 強い自律性、複数agent、複数ツールの協働を指向する

# UMP-JUST 主催 ハッカソン

## 『AIエージェントの社会実装』

**2026年6月27日（土）・28日（日）開催**

AIエージェントの自律的問題解決能力を用いて、日常的な課題、日本の課題、世界が直面する課題など、
自由な発想で課題を設定し、ツールによる問題解決を進めていただきます。課題の制約はありません。

### 審査基準

**問題着眼点・着想点**

- 何の問題を解いているかが明確で、着眼点が新しいこと
- 問題は既知でも、解き方が斬新・技術的に優れていること

**完成度・動作性**

- コンセプトで提示された機能が実際に動作し有効である
- AIエージェントの自律的問題解決能力が有効に活用されている

### 提供される環境

- **AWS Bedrock エージェント**（MARSFLAG 社 提供）
- Azure OpenAI エージェント（Givery 社 提供）

※ 上記のいずれも使わず、自ら他の環境を用意して利用しても構わない

# Amazon Bedrock

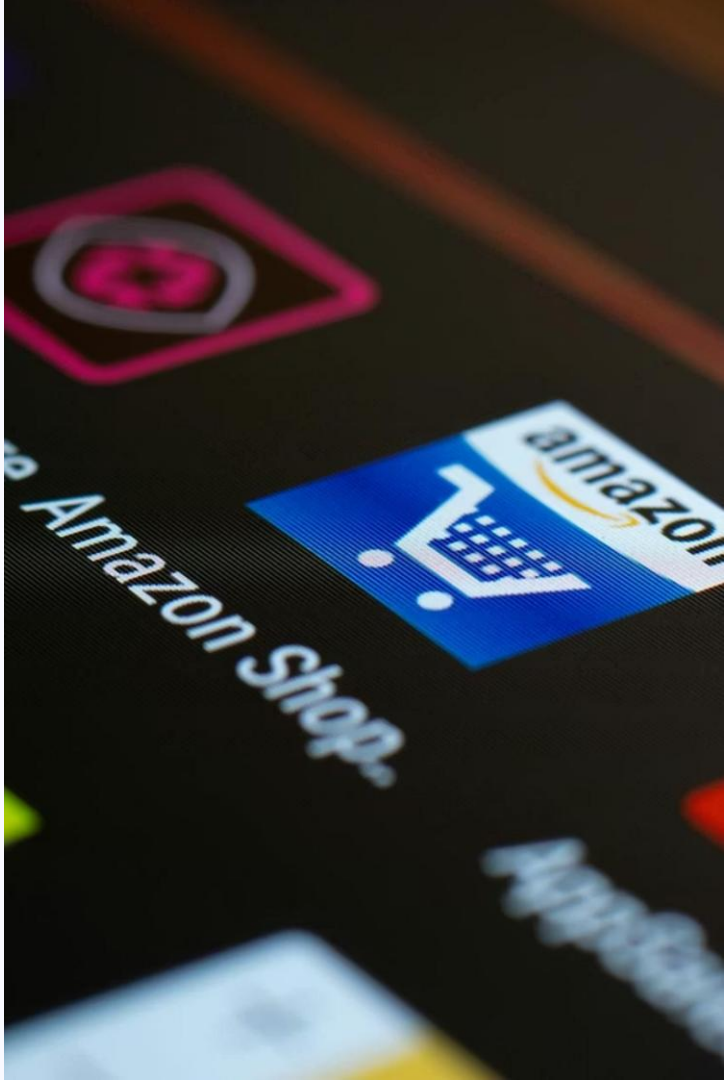
## 生成AIアプリとAIエージェントをAWS上で作るための基盤

### 基盤モデル
会話・要約・分類・生成の土台となるAIモデル

### Knowledge Base
社内文書を検索して回答する仕組み

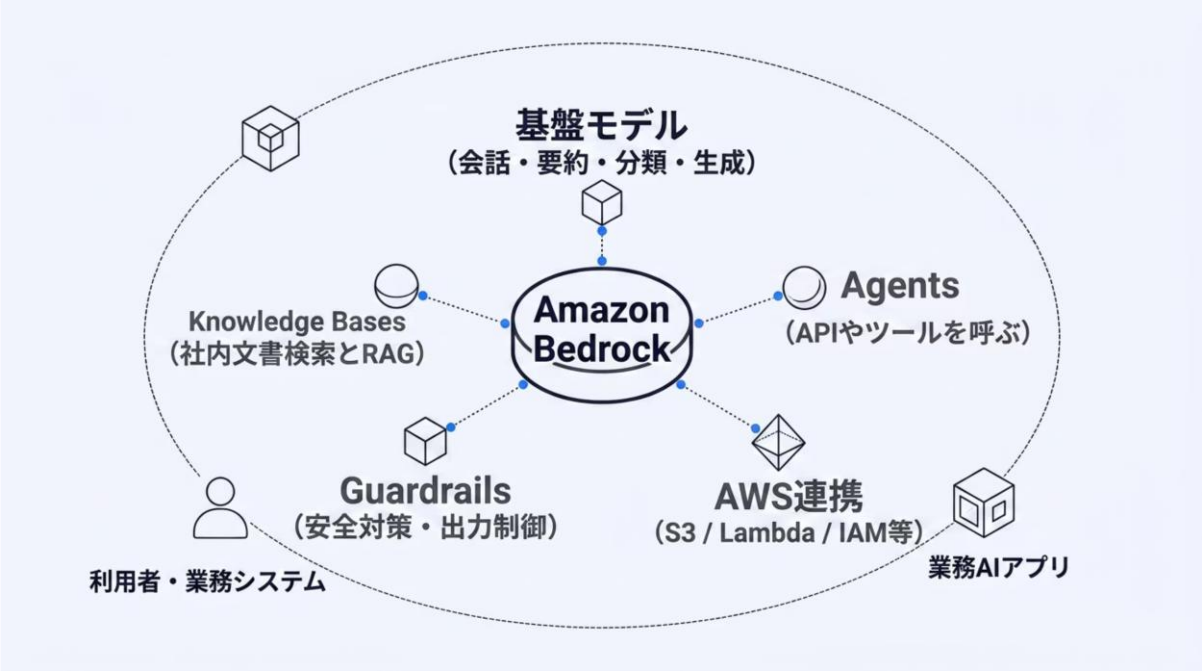
### Agents
APIやツールを呼び出して業務を実行

### Guardrails
安全対策・出力制御の仕組み



# Amazon Bedrock の位置づけ

単なるLLM呼び出しではなく、業務利用に必要な部品をまとめた基盤




**ポイント：**モデル選定・社内データ連携・アクション実行・安全対策を、AWSの管理基盤として扱える。

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# Bedrockは「AIそのもの」ではなく「AIを使う基盤」

ChatGPTのような単一サービス名ではなく、複数のモデルと周辺機能を扱うAWSサービス

## よくある誤解

- → ~~Bedrock = 1つのAIチャット~~
- → ~~Bedrock = 1つのモデル名~~

## 正しい捉え方

- 複数の基盤モデルを呼び出す
- 業務アプリ化のための共通基盤

## 開発者の見え方

- API経由で使い分ける
- AWSサービスと組み合わせる

**説明の一文**

Amazon Bedrock は、AWS上で基盤モデル(FM)を選び、社内データや業務API、安全対策と組み合わせて、生成AIアプリを作るためのマネージドサービスです。

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# なぜ Bedrock を使うのか

生成AIを業務に入れるときの「面倒な部分」をAWSに寄せられる

1

## モデルを選べる

性能・コスト・用途で選択

2

## インフラ運用を減らす

GPU管理を意識しにくい

3

## 社内データとつなぐ

RAGを作りやすい

4

## 業務APIを呼ぶ

エージェント化しやすい

**業務利用で重要なのは、モデルの賢さだけではありません。**「社内情報を使えるか」「権限を管理できるか」「ログを追えるか」「安全対策を入れられるか」まで含めて考える必要があります。

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# 1. 基盤モデルをAPIで使う

用途に応じて、テキスト・画像・マルチモーダルなどのモデルを選択

### 会話・QA
問い合わせ対応、ナレッジ検索

### 要約・分類
議事録要約、メール分類

### 生成・編集
提案書案、文章リライト

### 画像・動画
画像理解、コンテンツ生成

**使い分けの観点**

**精度**
複雑な推論・日本語品質・長文処理が必要か

**速度**
対話UIで待ち時間を短くしたいか

**コスト**
大量処理するため、単価を抑えたいか

**機能**
テキストだけでなく画像・表・PDFも扱うか

**LLM（Large Language Model）**：自然言語の理解・生成を担う大規模言語モデル。BedrockではLLMを含む複数のFMを利用できる。

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# 2. Knowledge Bases：社内情報を読ませて答える

Retrieval-Augmented Generation（RAG）をマネージドに構築するための機能


## 向いている例

- 社内規程・FAQ・製品資料・手順書
- 「最新ルールに基づいて答える」用途

## Bedrockでの役割

データソース接続、検索、回答生成までの仕組みを構成

## 注意点

- 文書品質の確保
- アクセス権の設計
- 検索精度の調整
- 根拠表示の設計

**要するに：LLMに「思い出させる」のではなく、必要な情報を検索してから答えさせる。**

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# 3. Agents：外部APIや業務処理を使って作業する

ユーザーの依頼を分解し、Knowledge BasesやAction Groupsを使ってタスクを進める


## Knowledge Bases

社内文書・FAQ・規程などから情報を取得

## Action Groups

LambdaやAPIを呼び、予約・照会・更新などを実行

## 指示と権限

何をしてよいか、どこまで自動化するかを設計

# 4. Guardrails：安全に使うための制御

ユーザー入力とモデル出力を評価し、不適切な内容や機密情報リスクを抑える

### 入力チェック
危険な依頼や不適切な内容を検出

### 出力チェック
望ましくない回答を抑制・調整

### 機密情報対策
個人情報・秘密情報の扱いを制御

### 一貫した体験
モデルをまたいだ安全方針を設計

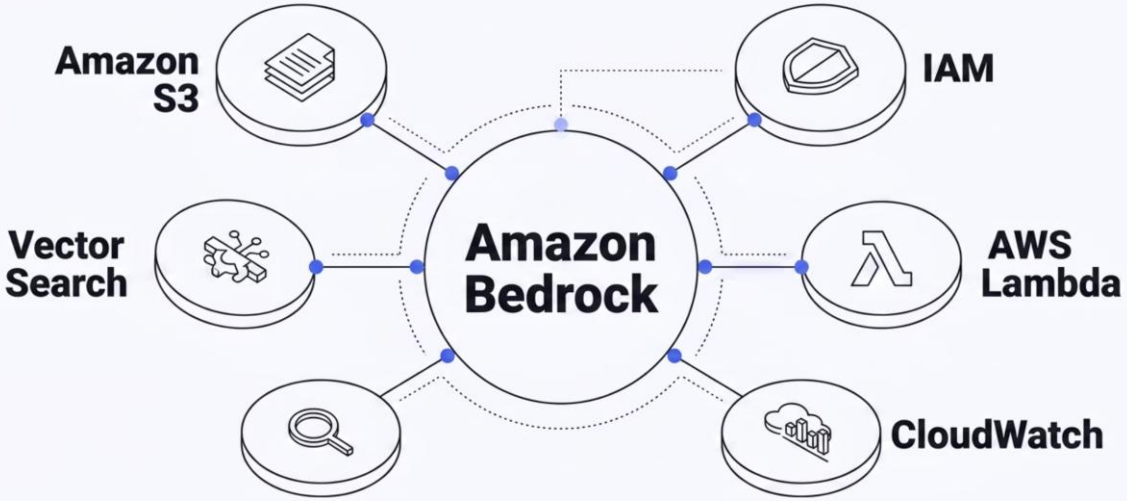
⚠ **エージェント化すると、1回の誤答が「実行」につながる可能性がある。**
そのため、Bedrockの安全機能に加えて、最小権限・人間確認・ログ監査・段階導入を組み合わせる設計が重要。

- 最小権限
- 人間確認
- ログ監査
- 段階導入
- 接続先管理

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# Bedrock は AWSサービスと組み合わせて強くなる

AI単体ではなく、データ・処理・監視・権限管理と接続して業務アプリにする




Amazon Bedrockを中核とする5つのAWSサービス連携

**業務AI化の本質は、LLMに「知識」と「手段」と「制限」を渡すこと。**

出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）

# Bedrock と AgentCore の関係

Bedrockは生成AIアプリの土台、AgentCoreはエージェント運用を強化する層

## Amazon Bedrock

生成AIアプリを作る基盤

- 基盤モデルの利用
- Knowledge Bases / RAG
- Agents / Guardrails
- AWSサービス連携

## Amazon Bedrock AgentCore

エージェントを実行・運用する基盤

- Runtime / Memory
- Gateway / Identity
- Code Interpreter / Browser
- Observability など

ハッカソンでは、まずBedrockの基本機能で「小さく動くAI」を作り、
必要に応じてAgentCoreで運用・実行面を広げる、という順番が現実的です。

# まとめ：Bedrockは業務AI化の土台

モデルを呼ぶだけでなく、データ・ツール・安全対策を組み合わせる

1. **生成AIアプリの基盤**
   複数の基盤モデルを使い分けられる
2. **社内情報とつなぐ**
   Knowledge BasesでRAGを構築できる
3. **業務実行に近づける**
   AgentsでAPIや処理を呼び出せる
4. **安全対策を入れる**
   GuardrailsやAWSの権限管理と組み合わせる

REFERENCES

# 参考情報

公式情報をもとに作成

### Amazon Bedrock 公式ページ

https://aws.amazon.com/bedrock/

### Amazon Bedrock ユーザーガイド：What is Amazon Bedrock

https://docs.aws.amazon.com/bedrock/latest/userguide/what-is-bedrock.html

### Amazon Bedrock Knowledge Bases

https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base.html

### Amazon Bedrock Agents

https://docs.aws.amazon.com/bedrock/latest/userguide/agents.html

### Amazon Bedrock Guardrails

https://docs.aws.amazon.com/bedrock/latest/userguide/guardrails.html

### Amazon Bedrock security / observability

https://docs.aws.amazon.com/bedrock/latest/userguide/security.html

⚠ AWSサービスの提供機能・モデル一覧・リージョン対応は更新されるため、利用前に必ず最新の公式ページで確認してください。

# サンプルアプリを作ってみよう！

Amazon Bedrock の新機能マルチエージェントで「わが家の AI 技術顧問」を作ろう！
https://aws.amazon.com/jp/builders-flash/202503/create-ai-advisor-with-bedrock/

## アプリの概要

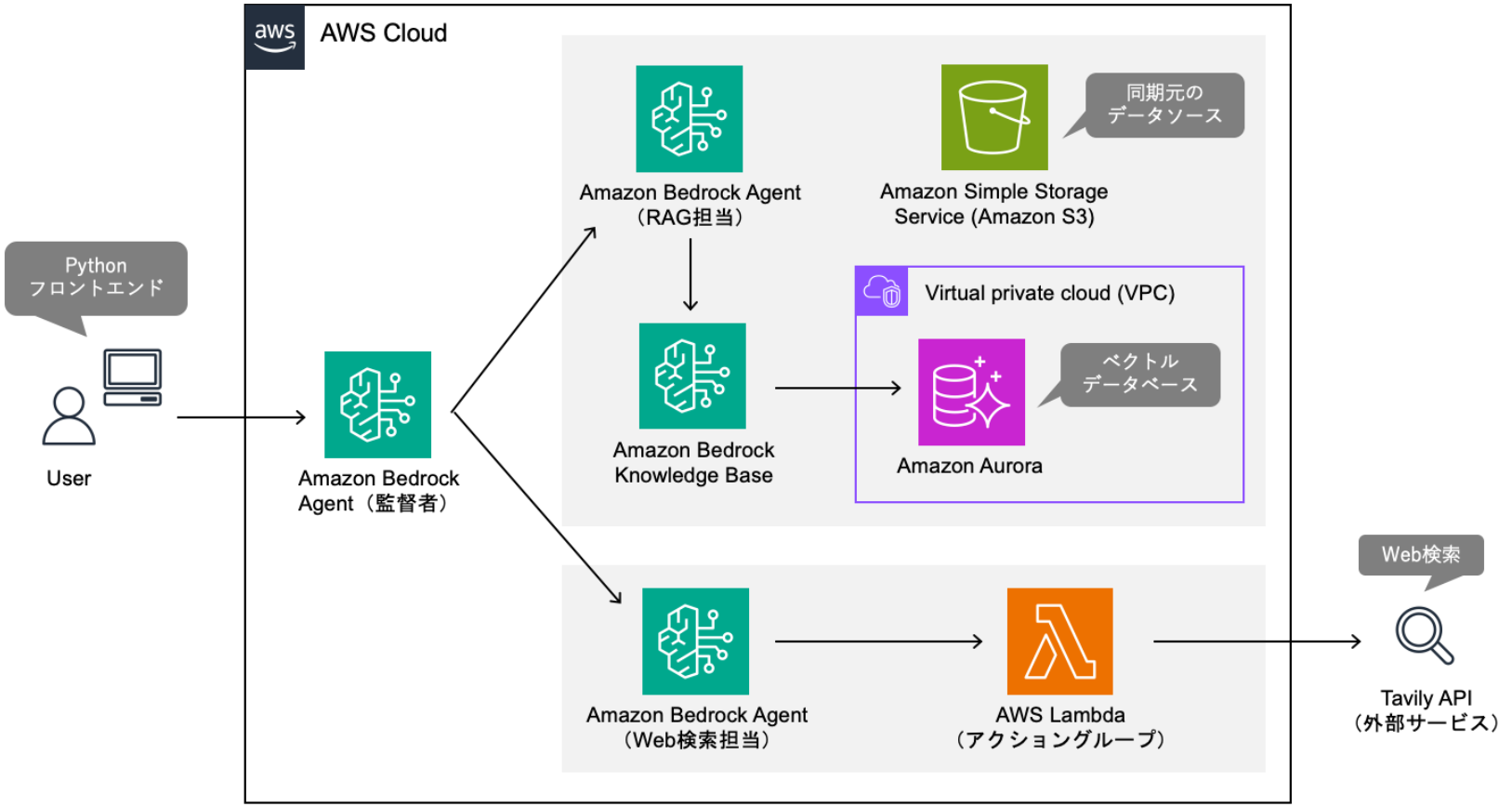
このアプリケーションに質問すると裏で 3 体の AI エージェントが動作します。

- 1 体目は監督者エージェントで、残りの 2 体のサブエージェントへ専門的な仕事をルーティングします。
- 2 体目は RAG (Retrieval-Augmented Generation、検索によって強化された生成) 担当のサブエージェントで、Amazon Bedrock のドキュメントを検索して情報提供を行います。
- 3 体目は Web 検索担当のサブエージェントで、Tavily という検索エンジンの API を使って最新情報を取得します。

## 環境の前準備

- Linux 実行環境
  - ➢ Python 3系
  - ➢ AWS CLI 2系
- AWS アカウント　※別途アナウンスします
  - ➢ IAMユーザー登録
    - ✓ AWS CLI 初期設定

# サンプルアプリ　アーキテクチャ図



# Claude Haiku / Sonnet / Opus の違い

ざっくり言うと **性能・速度・コスト** の階層名

| **種類** | **位置づけ** | **向いている用途** |
| --- | --- | --- |
| Haiku | 軽量・高速・低コスト | 要約、分類、簡単なチャット、大量処理 |
| Sonnet | バランス型 | 文章作成、分析、コード、業務利用全般 |
| Opus | 最上位・高性能 | 複雑な推論、長い作業、難しいコード、エージェント的作業 |

(余談：ファミリーの名称の由来)

Haiku：俳句。「5-7-5の17音」という短い形式

Sonnet：ソネット。ヨーロッパの伝統的な定型詩で、14行で構成される

Opus：オーパス。ラテン語で「作品」や「仕事」を意味する。詩や音楽の偉大な作品を指す

HACKATHON SCOPE

# 2日間ハッカソンで狙うなら

「全部使う」より、1つの業務課題に絞って動くデモを作る

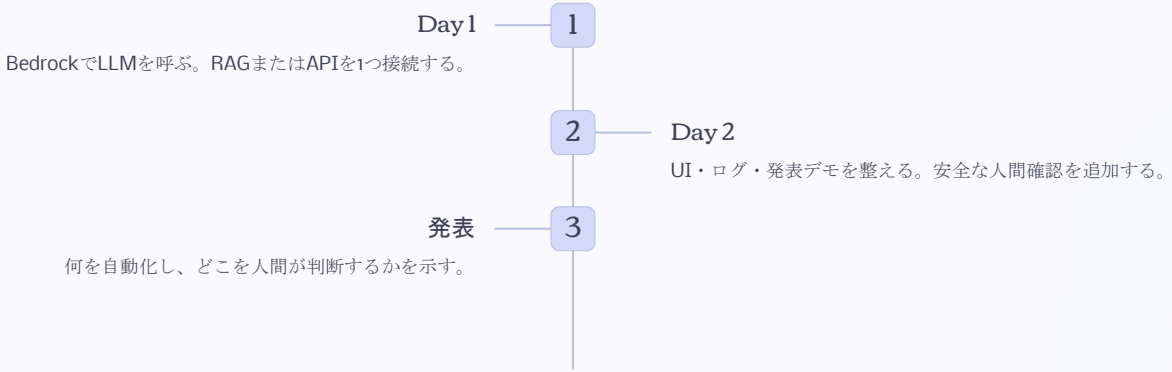
**社内FAQ AI**
Knowledge Bases + 基盤モデル

**調査メモ作成**
LLM + 検索/API + 出力整形

**申請補助AI**
Agents + Lambda/API

**提案書たたき台**
RAG + 生成 + 人間確認

## おすすめ構成




出典：AWS公式ドキュメント / AWS公式サイトをもとに作成（2026年5月時点）