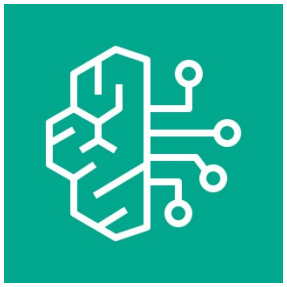
Spectrum Technology

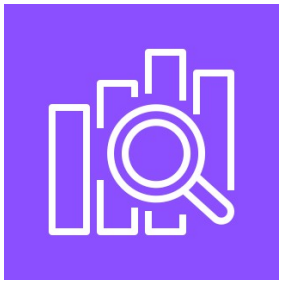
V1.0 2025/8/6

抜粋版

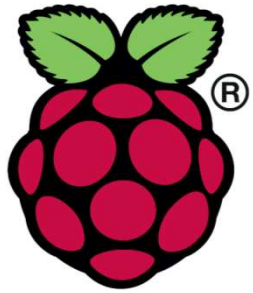
はじめてのAIエージェント学習・開発キット
～AWS bedrockを使った実践的AI活用キット～
開発編（Pi5版） その１



Amazon Bedrock



Amazon OpenSearch Service



Raspberry Pi

スペクトラム・テクノロジー株式会社

https://spectrum-tech.co.jp

sales1@spectrum-tech.co.jp

Spectrum Technology

開発編　目次

抜粋版のためページと内容は一致しません

開発編　目次

開発編その2

抜粋版のためページと内容は一致しません

Spectrum Technology

Pi運用マニュアル

Raspberry Pi

1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。LinuxベースのRasbianOSで動作しております。

2. Linux基本コマンド

① システム関係

- 起動：電源を入れると自動で起動します。
- 再起動：# reboot

  又は、アプリケーション>ログアウト>再起動；左上のメニューから
- 終了： # shutdown

  又は、アプリケーション>ログアウト>シャットダウン；左上のメニューから
- ログアウト # logout

  又は、アプリケーション>ログアウト>ログアウト；左上のメニューから
- 日本語／英語の入力切替：キーボードの全角/半角キーで切替ます。

Spectrum Technology

Pi運用マニュアル

Raspberry Pi

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

root@:~$ cd /root/Documents　　ディレクトリの切り替え
root@:/root/Documents# ls　　ファイルとディレクトリの表示（表示したら操作したいファイルを右クリックでコピペして操作します
root@:~# cp ファイル名　ディレクトリ　　配下のディレクトリのファイルを別のディレクトリへコピー
root@:~# mv ファイル名　ディレクトリ　　配下のディレクトリのファイルを別のディレクトリへ移動
root@:~# rm ファイル名　　ファイルの削除
便利な機能　rm –help　　コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

③ ユーザ権限、プロセス他

root@:~ $ su -　　スーパーユーザ（root)に切り替え、パスワードを入力
root@:~# ps a　　現状の動いているプロセスを表示
root@:~# kill　　特定のプロセスを強制終了
root@:~# apt-get install pkg　　パッケージのインストールなどに使用
root@:~# date　　日付、時間の設定を行います。
root@:~# mousepad /etc/network/interfaces　インタフェースに記述内容を変更します。Viよりも使いやすいです。

④ モジュール、usb、メモリ、HDDなどの表示

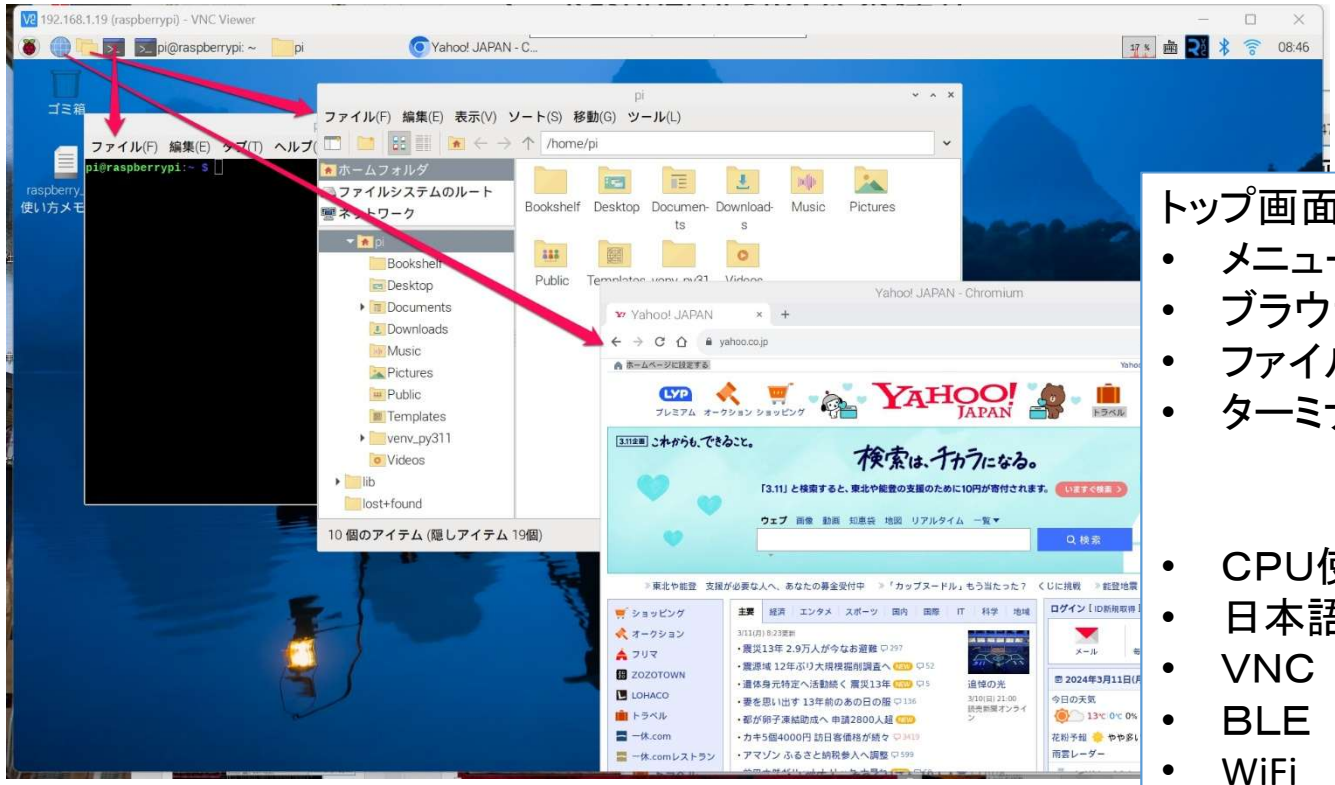
root@:~# lsmod　　linuxのモジュールリスト表示
root@:~# lsusb　　usbのデバイス表示
root@:~# free -mt　　メモリ使用状態表示
root@:~# df -h　　HDD（マイクロSD）の使用状態表示

pi@raspberrypi:~$ sudo
も同様のコマンドになります

RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

① 表示画面と内容



トップ画面（上段のタスクバーで選択）

- メニュー
- ブラウザ
- ファイルマネジャ
- ターミナル

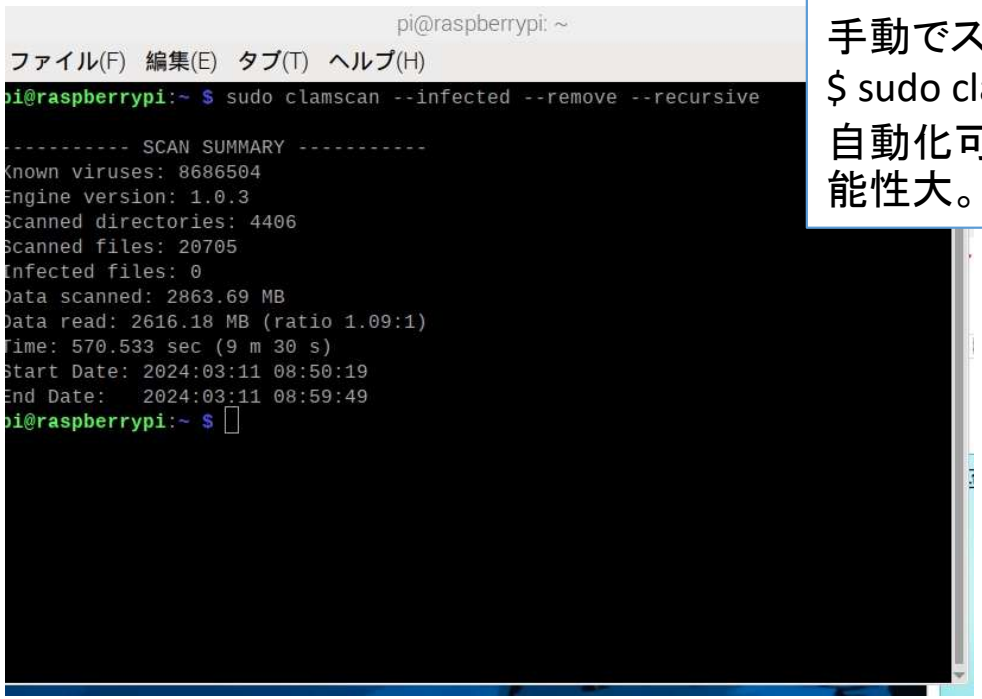
- CPU使用率
- 日本語入力
- VNC
- BLE
- WiFi
- 時刻

RaspberryPi運用マニュアル

4. 日常運用

① セキュリティ対策（アンチウイルス更新、スキャン）

- アンチウイルス対策として無料のclamAVをインストールしてます。
- 手動での運用を基本としてます。



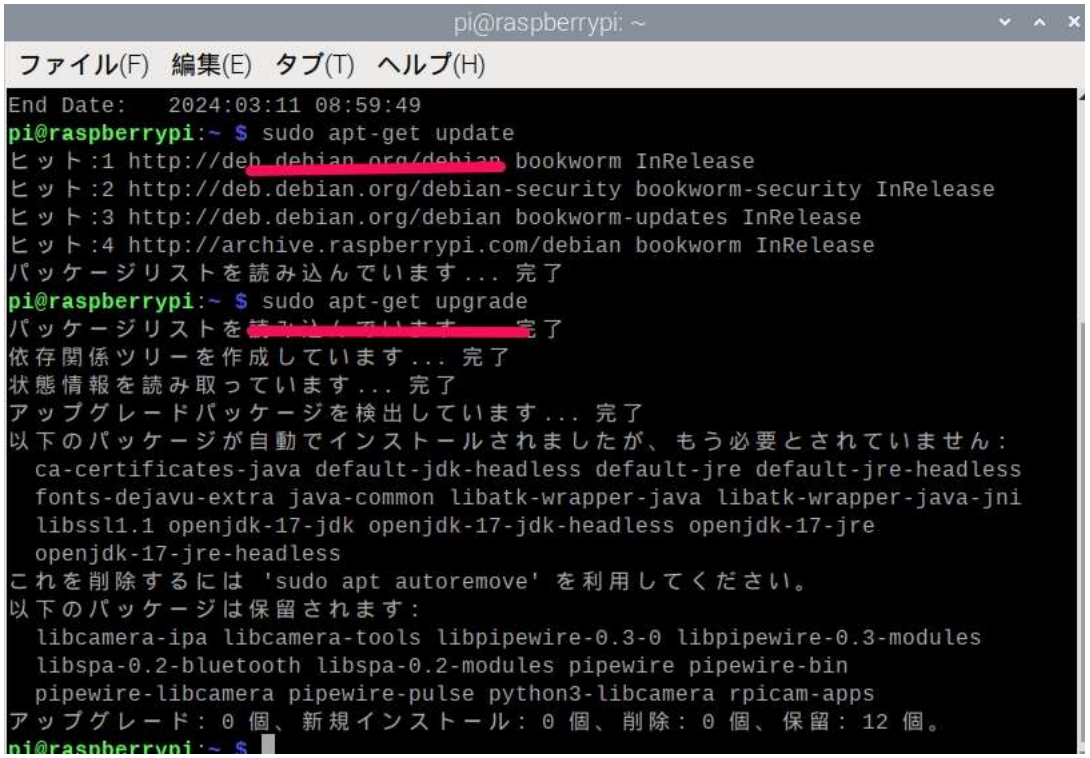
パターンファイル更新
手動スキャン時に更新されます
手動でスキャン
$ sudo clamscan --infected --remove --recursive
自動化可能ですが、バックグランドで重くなる可能性大。コマンド入力後約10分位かかります。

RaspberryPi運用マニュアル

4. 日常運用

② インストール済パッケージの更新リスト、アップグレード

- Linuxの場合は、頻繁に更新が発生します。アップグレードを定期的に実施してください。
- 更新前には、バックアップを取ることをお勧めします。特にアップグレードはまれに動作不良、戻せない状態が発生します。自己責任で実施してください。



更新リスト取得
$ sudo apt-get update
アップグレード実施
$ sudo apt-get upgrade

Spectrum Technology

開発キット　全体像(Pi5版)

ハードウエア

Raspberry Pi5

＋

物品追加

電源アダプタ
（オプション）

HDMIケーブル
（オプション）

OS

Rasbian OS

AIエージェント系（AWS）

Amazon Bedrock

Amazon OpenSearch Service

AWS CloudFormation

観測、分析系

Langfuse

PHOENIX

dynatrace

プログラム言語

python

jupyter

外部API連携

Web search
stock data
Working Memory
Tool
property data
economy data

弊社提供

Spectrum Technology

1．ハード概要

本体

品名	項目	内容	備考
Raspberry Pi5	CPU	2.4GHz 4コア Cortex-A76（ARMv8、64bit）	
	GPU	VideoCore VII®	
	メモリ	4GB RAM	
	OS	Raspbian bookworm(Debianベース）	
	インターフェース	2.4/5GHz WiFi(802.11 bgnac), Bluetooth 5.0, BLE, 1G ether, USB 2.0x2, USB 3.0x2, micro HDMIx2, microSDカード, 40 GPIO pin	
	電源／消費電力	Micro USB Type C　3.0A	
	サイズ	85x56x18mm	

付属品	内容	備考
ケース	赤白、FAN付。	
microSD 64GB	Raspbian OS, 必要なモジュールをインストールして提供します。お客様が設定するものは必要最低限のパスワード設定、WiFi設定になります。	
プログラム	Aiagent,関連pip インストール済	
マニュアル	開発キット　設定編、開発編	

USB電源ケーブル、HDMIケーブルは付属しておりません。
別途オプション品を購入ください

Spectrum Technology

2. ソフト概要
①ソフトー覧

提供するソフトウエアの概要です。

区分	ソフト名	バージョン	備考
OS	Raspbian	Bookworm 64bit	Kernel:6.12.34
Aws関係	cli	1.42.4	
	cdk	2.1025.0	
	botocore	1.40.4	
プログラム言語	python3	3.11.2	仮想化で利用
	npm	9.2.0	Web版
その他	Jupyter notebookなど多数のpipライブラリ		

Spectrum Technology

2. ソフトウエア概要
②. 開発キットプログラム一覧　その1

A:上級
B:中級
C:初心者

id	区分	項目	内容	aws利用サービス(iamは省略)	外部API連携	難易度	コメント
1	Bedrockコンソール	エージェント	Aws bedrockコンソールにてエージェント作成、テスト。基本が理解できます	bedrock		C	
2		ナレッジベース	Aws bedrockコンソールにてナレッジ作成、opensearch service（インデックス）作成、テスト。基本が理解できます	bedrock,oss		B	
3		ガードレール	有害なコンテンツを検出してフィルタリングする。またdefaultでは、us-guardrail-v1が入っています	bedrock		C	
4		フロー	生成AIを可視化したツールで作成します。テンプレートも充実	bedrock		C	
5		プロンプト管理	ユーザー独自のプロンプトを作成して保存できるため、異なるワークフローに同じプロンプトを適用することで時間を節約できます	bedrock		C	
6	エージェント（jupyter notebook）	agent_with_code_interpretation	コード翻訳例：ベストセラー本、会社売上などcsv,グラフで出力	bedrock		C	
7		agent_with_guardrails_integration	ガードレール例：投資アドバイザ用のガードレール	bedrock,oss, lambda,s3		B	
8		agent_with_knowledge_base_integration	ナレッジベース例：ソーラパネル問い合わせ	bedrock,oss, lambda,dynamodb,s3		B	
9		agent_with_long_term_memory	メモリ上のエージェント例：旅行予約、一度予約したものの変更	bedrock, lambda		C	
10		agent_with_models_not_yet_optimized_for_bedrock_agents	モデルを使ったエージェント例：レストラン予約（テーブル予約、メニュー設定）	bedrock,oss, lambda,dynamodb,s3		B	
11		custom_orchestration_agent	モデルを使ったエージェント例：レストラン予約（テーブル予約、メニュー設定）カスタマイズ	bedrock,oss, lambda,dynamodb,s3		B	エラー中
12		inline_agent	日時問い合わせ、python生成、HR関係	bedrock,lambda,s3		B	
13		manage_conversation_history	会話履歴管理：レストラン予約（テーブル予約、メニュー設定）	bedrock,oss, lambda,dynamodb,s3		B	
14		metadata_filtering_amazon_bedrock_agents	メタデータのフィルタリング方法：明示的、インテリジェント、暗示的	bedrock,ossa,s3		B	
15		open_api_schema_agent	保険問い合わせ：open APIスキーマを使用	bedrock,oss, lambda,dynamodb,s3		B	
16		ragas_evaluation_bedrock_agents	ragas評価：レストラン予約（テーブル予約、メニュー設定）	bedrock,oss, lambda,dynamodb,s3		B	エラー中
17		user_confirmation_agents	保険問い合わせ：open APIスキーマを使用	bedrock,oss, lambda,dynamodb,s3		B	
18	エージェント（python）	cdk_agent	Aws cdkでエージェントを作成：レストラン予約	bedrock,oss, lambda,dynamodb,s3,cloudformation		B	
19		computer_use	Aws cdkで外部のサイトと連携したエージェントを作成	bedrock,oss,s3,lambda,cloudformation,vpc,ec2,kms	amazon dcv	A	使用方法不明
20		connected_house_agent	カメラ、センサと連動したエージェントを作成	bedrock,oss,s3,lambda,cloudformation,kinesis		A	カメラ連携、未実施

Spectrum Technology

2. ソフトウエア概要
②. 開発キットプログラム一覧　その2

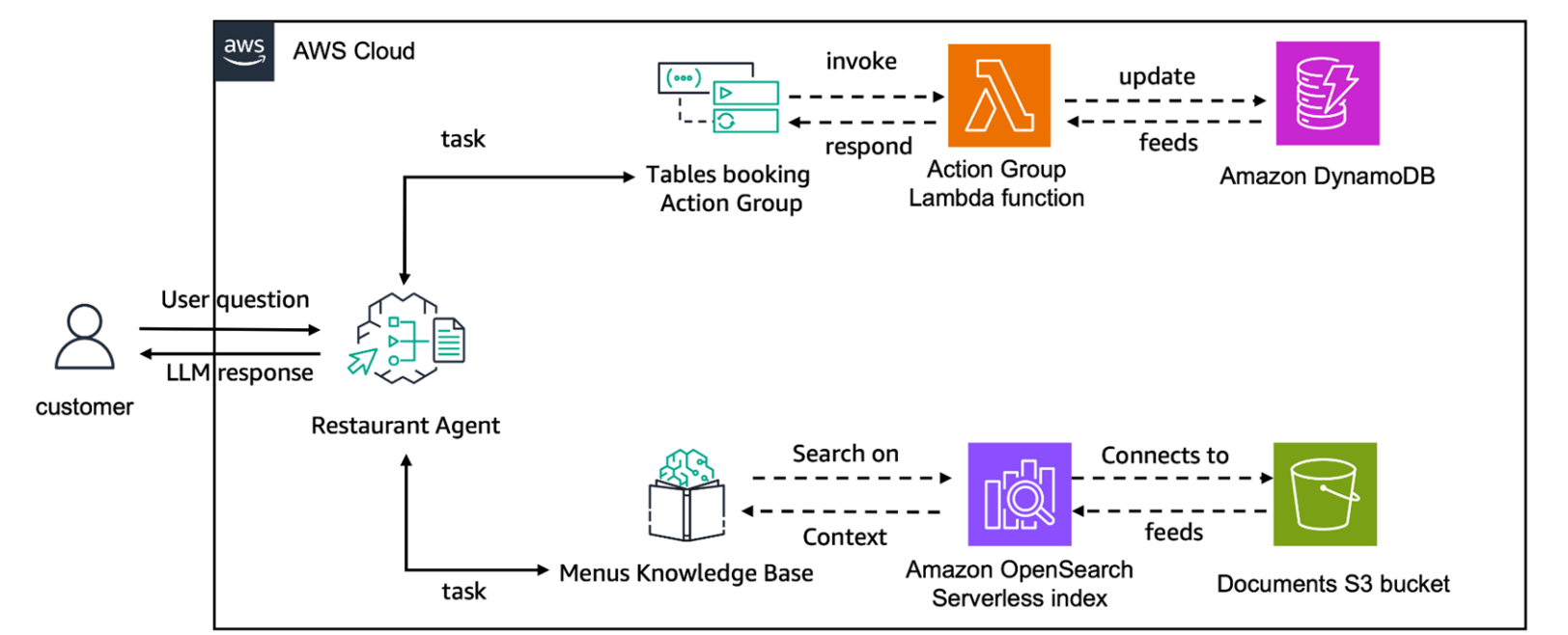
A:上級
B:中級
C:初心者

id	区分	項目	内容	aws利用サービス(iamは省略)	外部API連携	難易度	コメント
21	マルチ・エージェント（jupyter notebook)	energy_efficiency_management_agent	ソーラパネル保守、需要予測、管理者などのマルチ・エージェントを作成	bedrock,oss,s3,lambda,dynamodb		A	
22		investment_research_agent	投資相談、株価、ニュースなどのマルチ・エージェントを作成	bedrock,oss,s3,lambda,dynamodb,cf	Web search、stock data	A	1回作成に$5かかります
23	マルチ・エージェント（python)	00_hello_world_agent	Pythonによるマルチ・エージェントを作成	bedrock		C	
24		contract_assistant_agent	複数の契約エージェントによるマルチ・エージェント	bedrock,oss, lambda,dynamodb,s3		B	
25		metadata_filtering	ナレッジをエージェント毎にフィルタしたマルチ・エージェント	bedrock,oss, lambda,s3		B	
26		mortgage_assistant	複数のローンエージェント（一般、契約等）によるマルチ・エージェント	bedrock,oss, lambda,s3		B	
27		portfolio_assistant_agent	株価、web検索によるマルチ・エージェント、investment_research_agent （juypter notebook） のpython版	bedrock,oss,s3,lambda,dynamodb,cf	Web search、stock data	A	1回作成に$5かかります
28		real_estate_investment_agent	不動産投資（価格、環境、金融）によるマルチ・エージェント	bedrock,oss,s3,lambda,cf	Web search、Working Memory Tool ,property data, economy data	A	
29		startup_advisor_agent	起業家アドバイザによるマルチ・エージェント	bedrock,oss,s3,lambda、cf	Web search、Working Memory Tool	A	
30		team_poems_agent	NBAなどのプロチームのマルチ・エージェント	bedrock,lambda,cf	Web search	B	
31		trip_planner_agent	旅行企画（レストラン、日程など）のマルチ・エージェント	bedrock,lambda,cf	Web search、Working Memory Tool	B	
32		voyage_virtuoso_agent	旅行企画（日程など）のマルチ・エージェント	bedrock,lambda,dynamodb	Web search	B	
33	エージェントUX（python)	streamlit_demo	マルチ・エージェントをwebインタフェースで表示	bedrock,lambda,dynamodb	Web search	B	
34		video_games_sales_assistant_with_amazon_bedrock_agents	webインタフェースでゲーム販売支援 フロントエンド作成	bedrock,lambda,dynamo,Aurora		A	
35		foursquare_location	位置情報を提供するエージェント	bedrock		C	
36		inline-agent-hr-assistant	インラインのHRエージェント	bedrock,lambda		B	エラー入力できない？
37	エージェント観測（python)	OpenTelemetry-Agent-Instrumentation	エージェントのアプリ開発の動作観測、分析	bedrock,oss,lambda,dynamo	langfuse	B	
38	フロー	Quick start(jupyter notebook)	Amazon Bedrock Flowsを使って簡単にエージェント、ナレッジを生成	bedrock		C	
39		rag_kb_flow(python)	Amazon Bedrock Flowsを使って簡単にエージェント、ナレッジを生成	bedrock		B	紹介のみ、各自で実施

Spectrum Technology

AIエージェント全体像

- AWSのAIエージェントは、ナレッジベース（基盤モデル：FMとユーザのもつナレッジを組合せて学習）と公開の検索インデックスにより回答を生成します。これまでの生成AIが苦手なリアルタイムのWeb検索なども組合せが可能。また下のように、予約内容をDB化することも可能です。
- AWSの各種のサービスと連携することにより、無限大にアプリを作成することができます。



Bedrock

- awsの課金に関しては、お客様の責任において、管理してください。
- マニュアル等の記載モレなどによる、課金事故に関しても弊社は、一切責任は負いません

- Amazon Bedrock は、単一の API を通じて AI21 Labs、Anthropic、Cohere、DeepSeek、Luma、Meta、Mistral AI、poolside (近日リリース予定)、Stability AI、TwelveLabs (近日リリース予定)、Writer および Amazon などの先駆的な AI 企業からの高性能な基盤モデル (FM) の幅広い選択肢を提供するフルマネージドサービスであり、セキュリティ、プライバシー、責任ある AI を備えた生成 AI アプリケーションを構築するために必要な一連の幅広い機能を提供します。Amazon Bedrock を使用すると、ユースケースに最適な FM を簡単に試して評価したり、微調整や検索拡張生成 (RAG) などの手法を使用してデータに合わせてカスタマイズしたり、エンタープライズシステムとデータソースを使用してタスクを実行するエージェントを構築したりできます。
- エージェントとナレッジベースを構築します。
- bedrock https://aws.amazon.com/jp/bedrock/
- Github: https://github.com/awslabs/amazon-bedrock-agent-samples
- 特徴
  - インフラ不要
    - サーバーレスであるため、インフラストラクチャを管理する必要がありません
  - All-in-oneで簡単にRAGの構築が可能
    - AWSの各種サービスを利用して、簡単にAIエージェントの構築が可能。基盤モデルとお客様データを組合せた検索拡張生成（RAG）も簡単に構築。
  - クラウドで料金が従量制
    - クラウドの特徴である、従量制の料金で、多額の初期投資が不要。

OpenSearch Service

- Amazon OpenSearch Service は、アプリケーションモニタリング、ログ分析、オブザーバビリティ、ウェブサイト検索などのユースケースにおいて、ビジネスおよび運用データのリアルタイム検索、モニタリング、分析を安全に行うことができるようにします。
- OpenSearch Service https://aws.amazon.com/jp/opensearch-service/
- 特徴
  - インフラ不要
    - サーバーレスであるため、インフラストラクチャを管理する必要がありません
  - ダウンタイムなしで管理
    - お客様は中断することなく最新の状態を保つことができます。
  - クラウドで料金が安価？
    - クラウドの特徴である、従量制の料金で、多額の初期投資が不要。
    - 一度設定すると使っていなくても課金される。課金事故に注意

2. Bedrockコンソール

①. エージェント

Aws bedrockコンソールにてエージェント作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

作成

bedrockコンソールを開く

Build>エージェント：エージェント作成

名前：デフォルト又は適宜

入力
今日は何日ですか？

2. Bedrockコンソール

①. エージェント

Aws bedrockコンソールにてエージェント作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

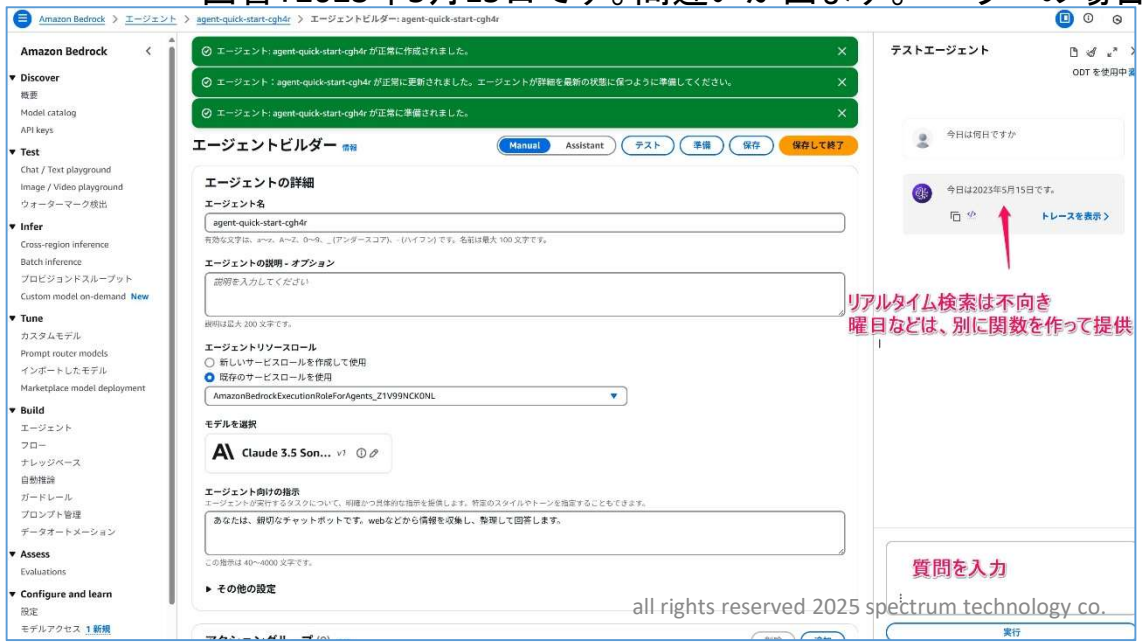
作成

テスト1

準備を押し、テスト

質問に入力し、実行　例「今日は何日ですか？」

回答：2023年5月15日です。間違いが出ます。 エラーの場合は、こちら



入力
AIとは何ですか？

2. Bedrockコンソール

①. エージェント

Aws bedrockコンソールにてエージェント作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

作成

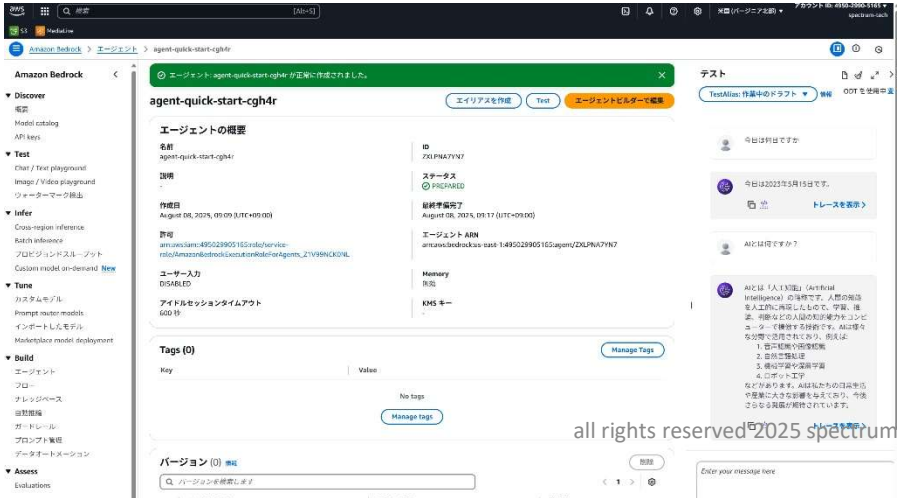
テスト2

準備を押し、テスト

質問に入力し、実行　例「AIとは何ですか？」

回答:

- AIとは「人工知能」（Artificial Intelligence）の略称です。人間の知能を人工的に再現したもので、学習、推論、判断などの人間の知的能力をコンピューターで模倣する技術です。AIは様々な分野で活用されており、例えば:
- 音声認識や画像認識、自然言語処理、機械学習や深層学習、ロボット工学
- などがあります。AIは私たちの日常生活や産業に大きな影響を与えており、今後さらなる発展が期待されています。



2. Bedrockコンソール

②. ナレッジベース

Aws bedrockコンソールにてナレッジ作成、opensearch service（インデックス）作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

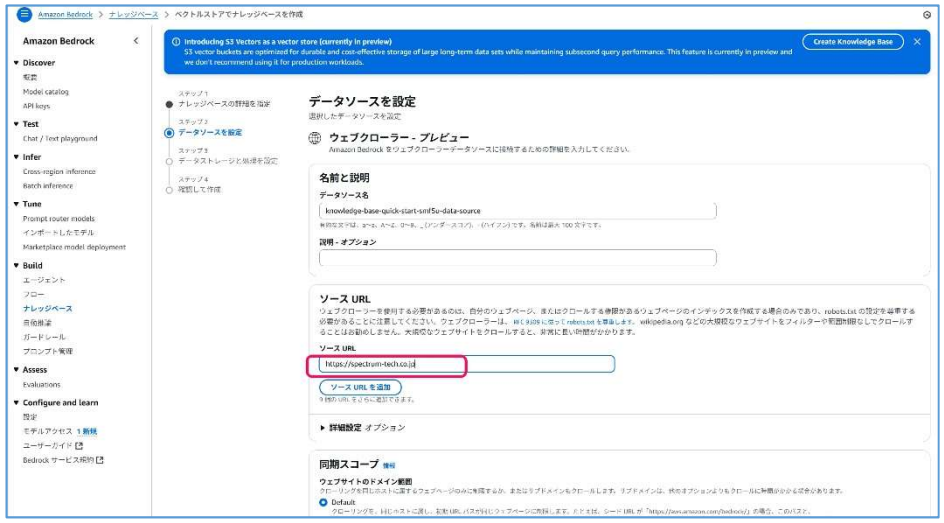
作成

設定後、ユーザ名でサインインします。

ナレッジベース作成

データソース：webにしました。

URL:https://spectrum-tech.co.jp



2. Bedrockコンソール

②. ナレッジベース

Aws bedrockコンソールにてナレッジ作成、opensearch service（インデックス）作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

作成

設定後、ユーザ名でサインインします。

ナレッジベース作成

データソース：webにしました。

URL:https://spectrum-tech.co.jp

モデルとベクトルDBを選択：oss

入力
lineと連携した熱中症対策について教えて

2. Bedrockコンソール

②. ナレッジベース

Aws bedrockコンソールにてナレッジ作成、opensearch service（インデックス）作成、テスト。基本が理解できます。

https://us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/

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

作成

設定後、ユーザ名でサインインします。

ナレッジベース作成

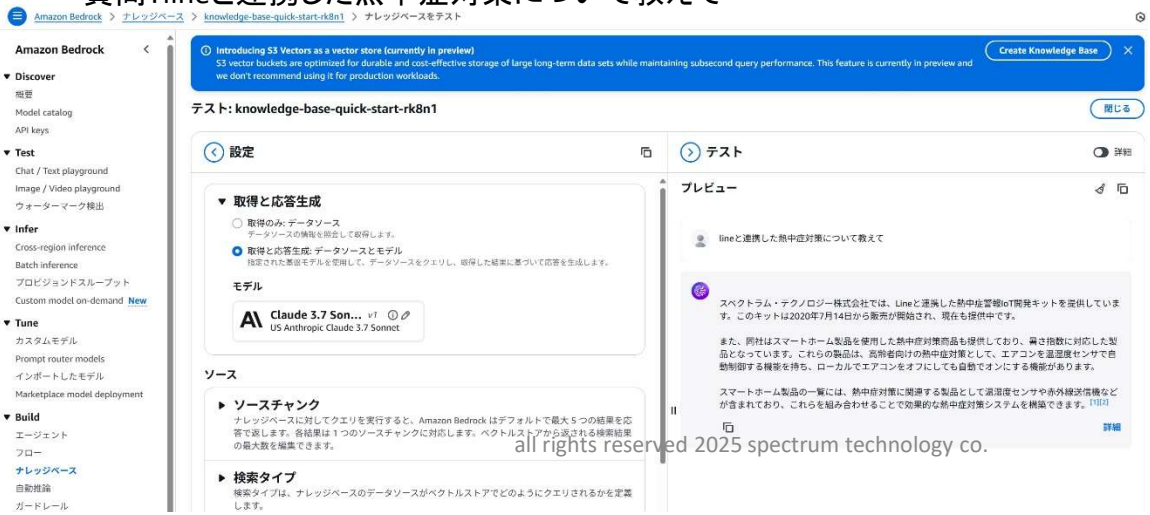
ナレッジ生成、同期

テスト1

取得と応答生成を選択

モデルは、claude3.7 sonnet v1 他のものを選ぶとエラーがでます。

質問：lineと連携した熱中症対策について教えて



2. Bedrockコンソール

④. フロー

生成AIを可視化したツールで作成します。テンプレートも充実。

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

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

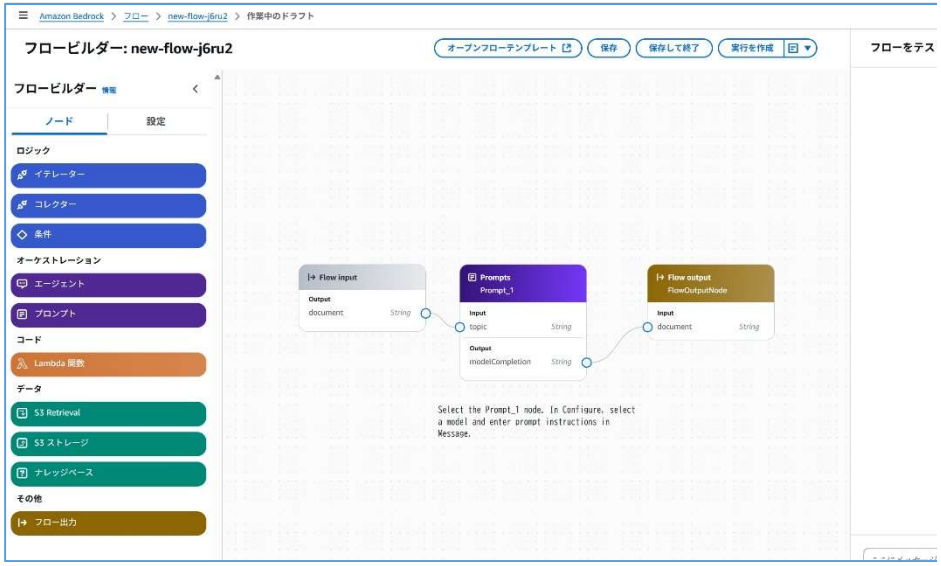
作成

bedrockコンソールを開く

Build>フロー:フロー作成

名前:適宜

フロービルダで編集



Spectrum Technology

3. エージェント（jupyter notebook）

Jupyterは、vnc接続では、ブラウザの動作が遅くなります。モニタを接続してpiを使用してください。

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_code_interpretation
$ jupyter notebook
code_assistant_agent.ipynb

①. agent_with_code_interpretation

コード翻訳例：ベストセラー本、会社売上などcsv,グラフで出力。

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_code_interpretation

テスト：(venv_py311) で実施のこと

$ source /home/pi/venv_py311/bin/activate

$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_code_interpretation

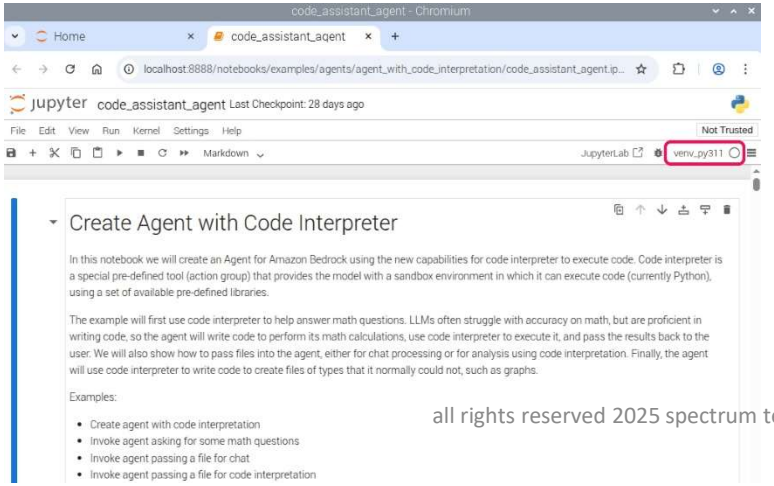
$ jupyter notebook

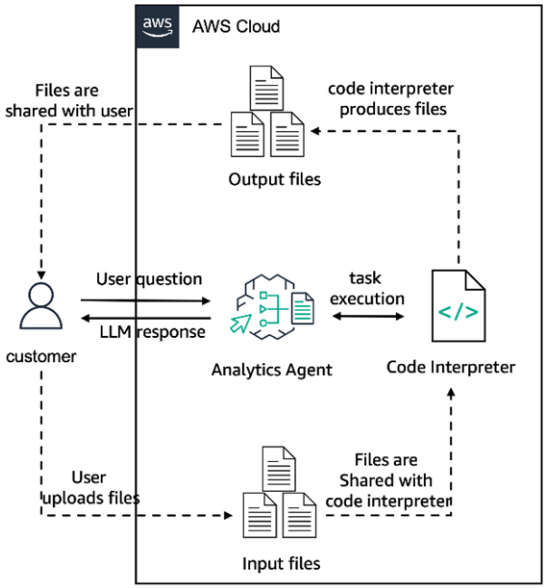
code_assistant_agent.ipynb

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作





Spectrum Technology

3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_code_interpretation
$ jupyter notebook
code_assistant_agent.ipynb

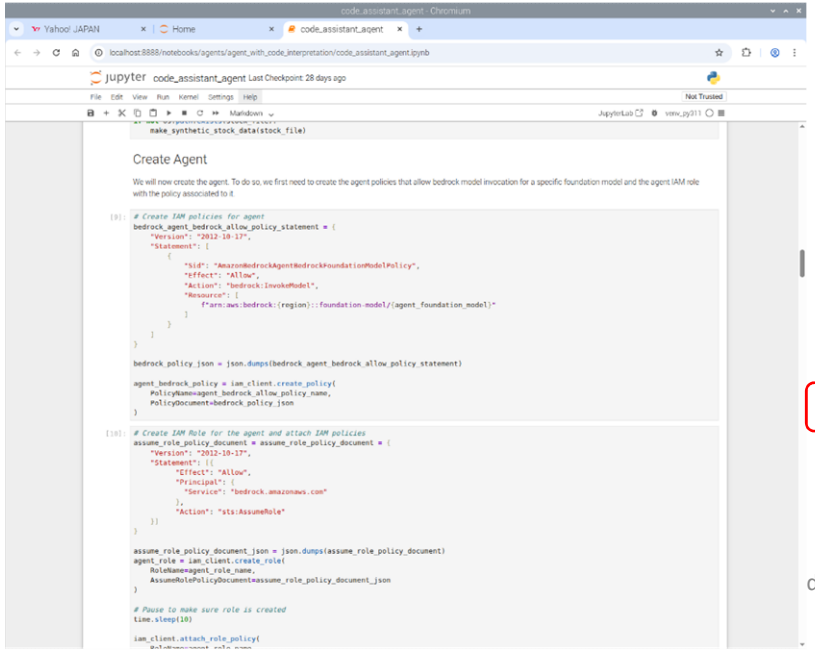
①. agent_with_code_interpretation

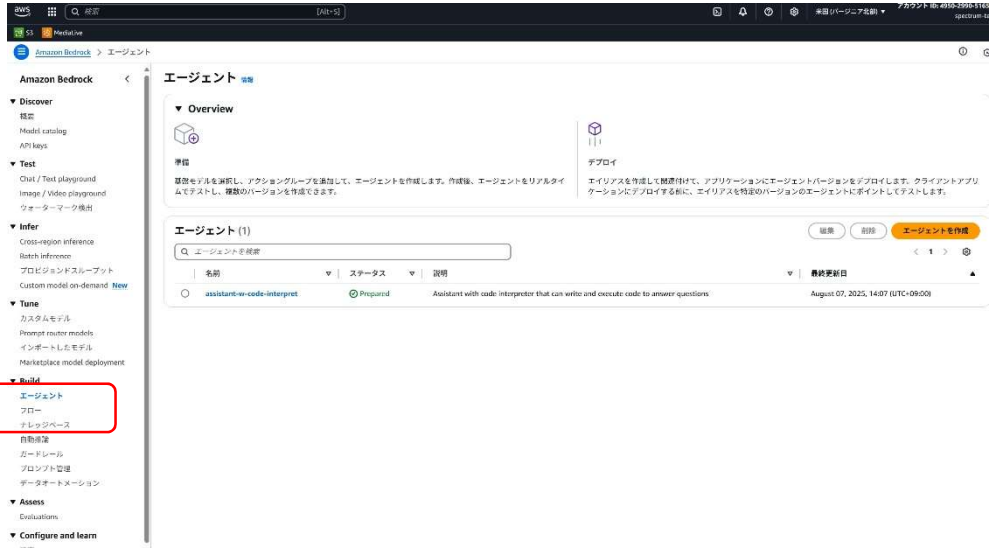
コード翻訳例：ベストセラー本、会社売上などcsv,グラフで出力。

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_code_interpretation

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認
Pipのインストールは、実施しないで順番にjupyter notebookを動作
6行目のconfiguration variableでモデルの指定があります。AWS　Bedrockのモデルアクセスでアクセスを付与
使用モデルは、anthropic.claude-3-sonnet
エージェント作成すると、Bedrock側にエージェントができます。





Spectrum Technology

3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_code_interpretation
$ jupyter notebook code_assistant_agent.ipynb

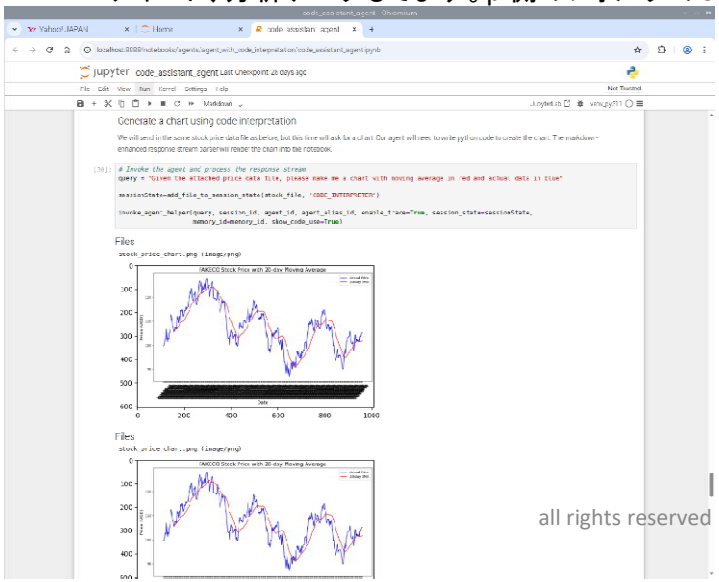
①. agent_with_code_interpretation

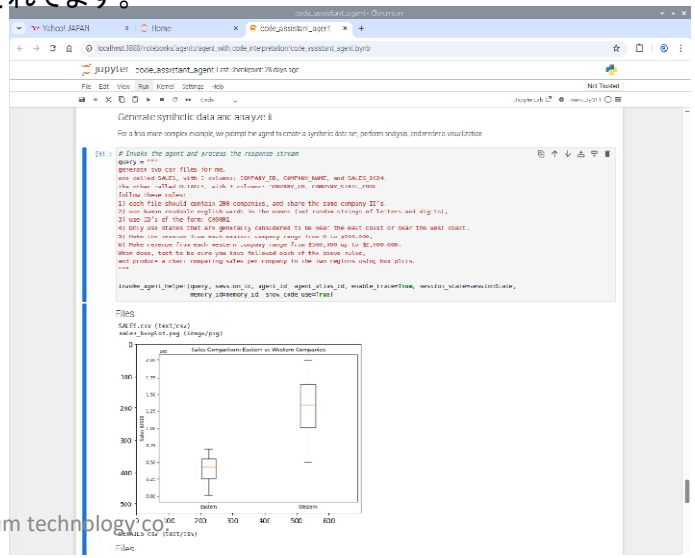
コード翻訳例：ベストセラー本、会社売上などcsv,グラフで出力。

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_code_interpretation

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認
Pipのインストールは、実施しないで順番にjupyter notebookを動作
6行目のconfiguration variableでモデルの指定があります。AWS　Bedrockのモデルアクセスでアクセスを付与
使用モデルは、anthropic.claude-3-sonnet
エージェント作成すると、Bedrock側にエージェントができます。
エージェントに問い合わせすると回答がでます。また、bedrockのコンソールでもテストできます。
チャート、分析データもでます。pi側のフォルダのoutに出力されてます。





3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_knowledge_base_integration
$ jupyter notebook
solar_panel.ipynb

③. agent_with_knowledge_base_integration

ナレッジベース例：ソーラパネル問い合わせ

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_knowledge_base_integration

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,OSS,S3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

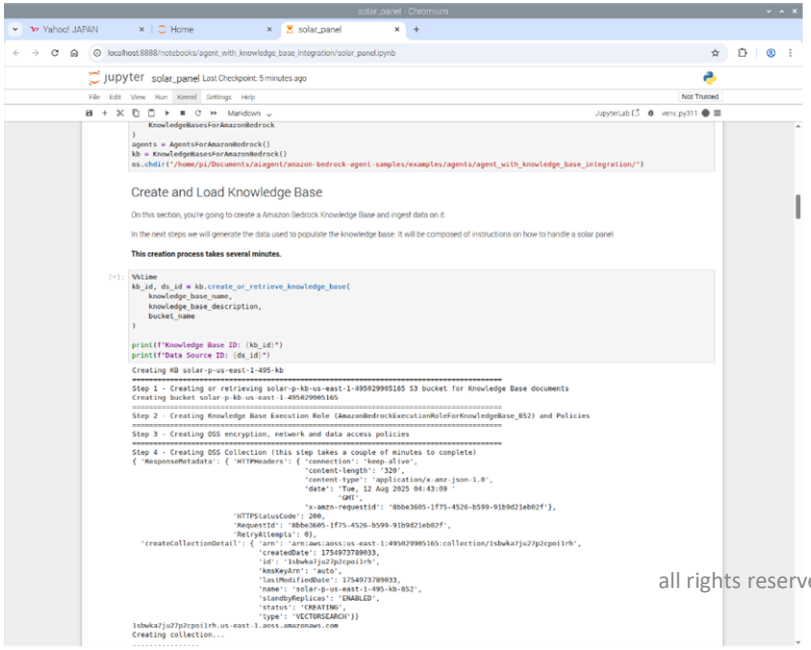
動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

Bedrock>ナレッジ生成：10分位

Bedrock、ossコンソール



3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_knowledge_base_integration
$ jupyter notebook
solar_panel.ipynb

③. agent_with_knowledge_base_integration

ナレッジベース例：ソーラパネル問い合わせ

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_knowledge_base_integration

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,OSS,S3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

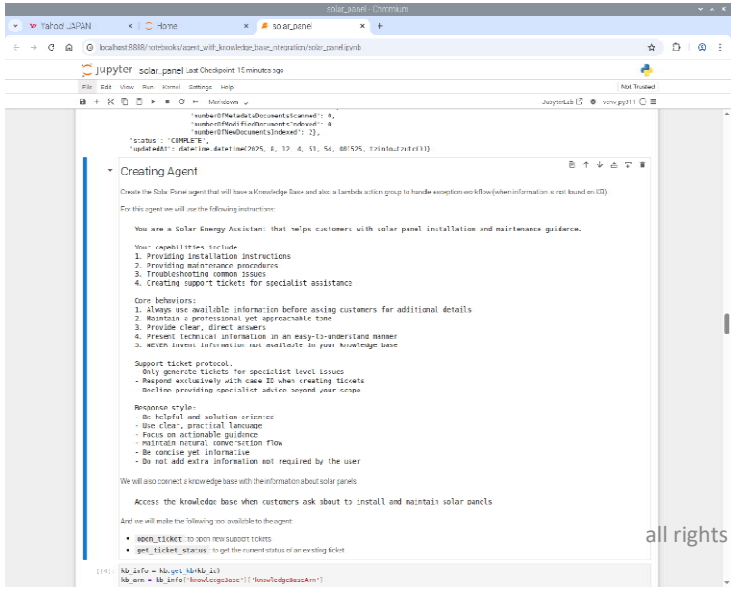
動作

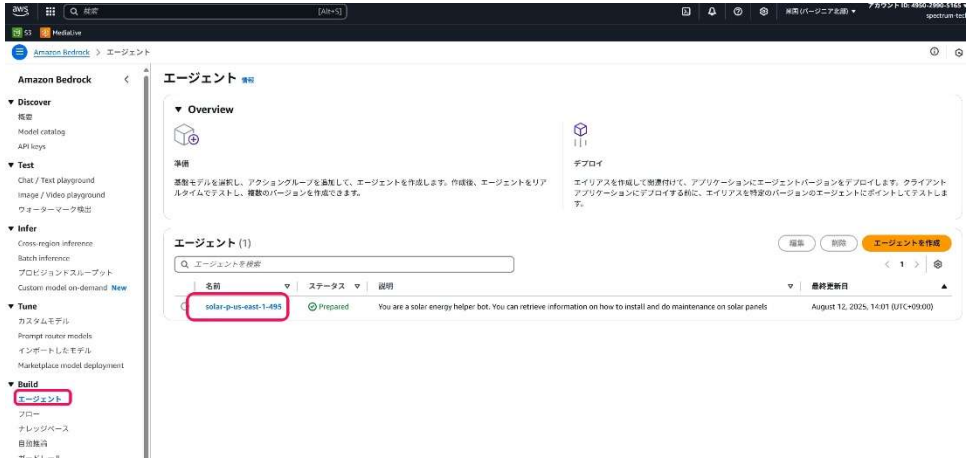
該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

Bedrock>ナレッジ生成：10分位

エージェント作成





3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_knowledge_base_integration
$ jupyter notebook
solar_panel.ipynb

③. agent_with_knowledge_base_integration

ナレッジベース例：ソーラパネル問い合わせ

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_knowledge_base_integration

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,OSS,S3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

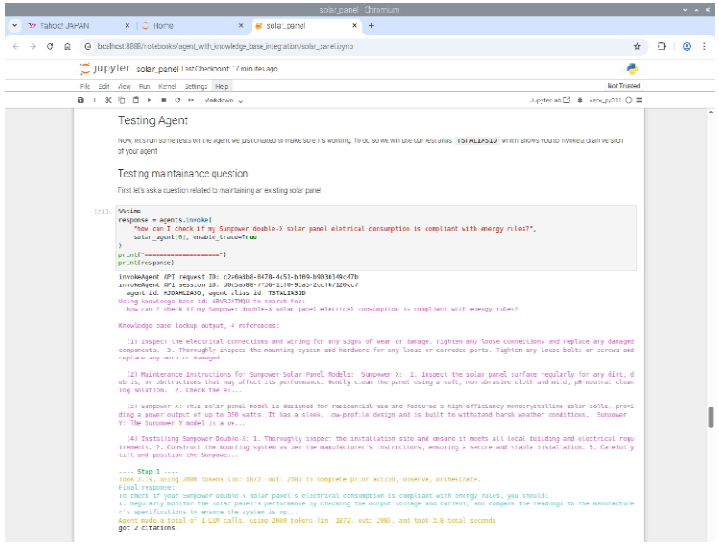
該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

Bedrock>ナレッジ生成：10分位

エージェント作成

テスト1　質問：how can I check if my Sunpower double-X solar panel eletrical consumption is compliant with energy rules?



回答例
To check if your Sunpower Double-X solar panel's electrical consumption is compliant with energy rules, you should: [s3://solar-p-kb-us-east-1-495029905165/solar-panel-maintenance.txt]
1. Regularly review and stay informed about any updates or changes in local energy regulations and building codes that may affect the installation and operation of your solar energy system.
2. Maintain detailed records of the system's installation, maintenance, and performance to demonstrate compliance with energy rules.
3. Consult with local authorities or a qualified solar professional to ensure your system meets all necessary requirements and obtain any required permits or approvals.
4. Participate in any energy efficiency or renewable energy programs offered by local utilities or government agencies to stay informed about compliance requirements.
5. Consider enrolling in a solar energy monitoring service or using a smart home system to track your system's performance and identify any potential compliance issues. [s3://solar-p-kb-us-east-1-495029905165/solar-panel-maintenance.txt] .

Spectrum Technology

3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_knowledge_base_integration
$ jupyter notebook
solar_panel.ipynb

③. agent_with_knowledge_base_integration

ナレッジベース例：ソーラパネル問い合わせ

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_knowledge_base_integration

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,OSS,S3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

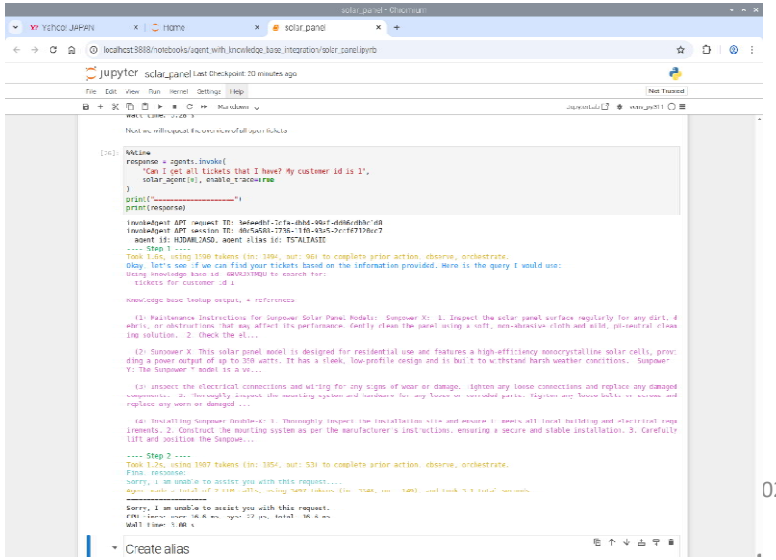
該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

Bedrock>ナレッジ生成：10分位

エージェント作成

テスト2　質問：Can I get all tickets that I have? My customer id is 1



回答例
Sorry, I am unable to assist you with this request.

3. エージェント（jupyter notebook）

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_models_not_yet_optimized_for_bedrock_agents
$ jupyter notebook
create_agents.ipynb

⑤. agent_with_models_not_yet_optimized_for_bedrock_agents

モデルを使ったエージェント例：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_models_not_yet_optimized_for_bedrock_agents

$ source /home/pi/venv_py311/bin/activate

$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_models_not_yet_optimized_for_bedrock_agents

$ jupyter notebook

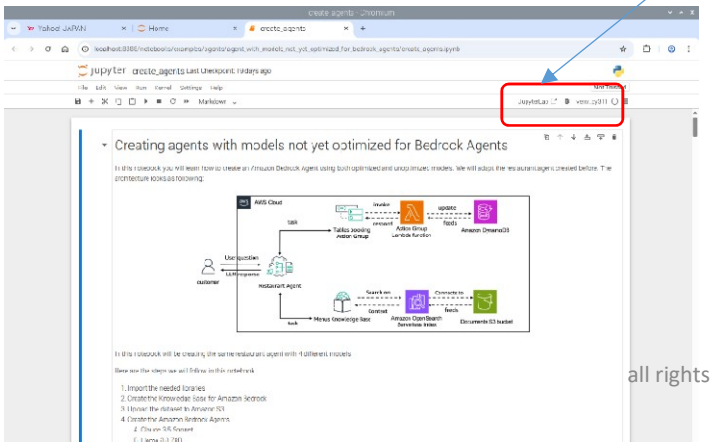
create_agents.ipynb

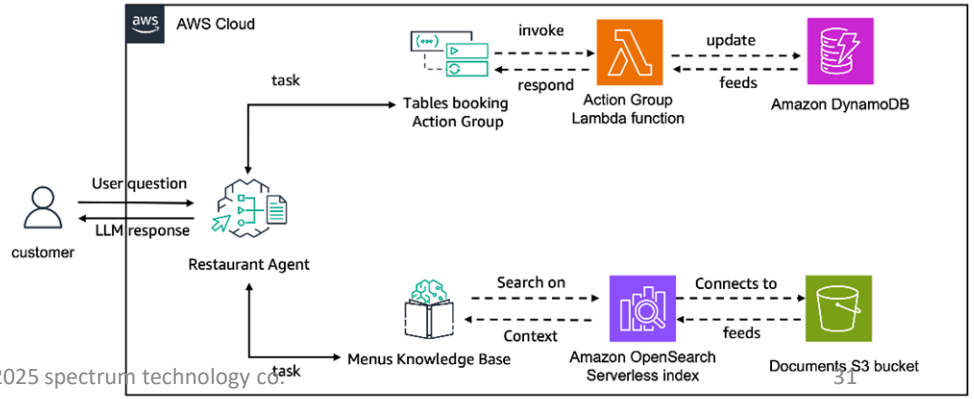
使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作





3. エージェント（jupyter notebook）

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/agent_with_models_not_yet_optimized_for_bedrock_agents
$ jupyter notebook
create_agents.ipynb

⑤. agent_with_models_not_yet_optimized_for_bedrock_agents

モデルを使ったエージェント例：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/agent_with_models_not_yet_optimized_for_bedrock_agents

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

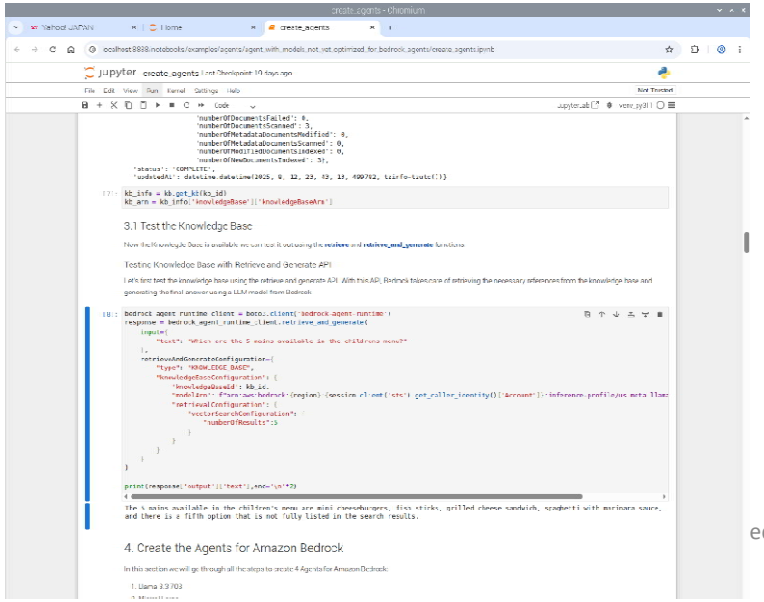
動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

ナレッジ生成

ナレッジテスト：Q：what is in the kids menu?



回答例
The 5 mains available in the children's menu are mini cheeseburgers, fish sticks, grilled cheese sandwich, spaghetti with marinara sauce, and there is a fifth option that is not fully listed in the search results

3. エージェント（jupyter notebook）

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/manage_conversation_history
$ jupyter notebook conversation_history.ipynb

⑧. manage_conversation_history

会話履歴管理：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/manage_conversation_history

テスト：(venv_py311) で実施のこと

$ source /home/pi/venv_py311/bin/activate

$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/manage_conversation_history

$ jupyter notebook

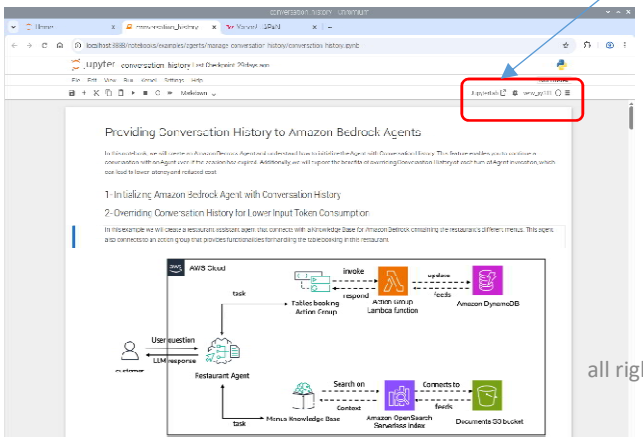
conversation_history.ipynb

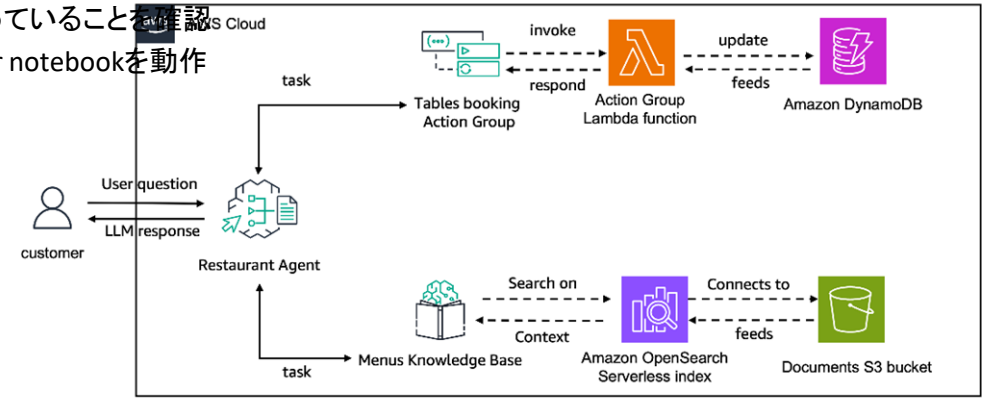
使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作





Spectrum Technology

3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/manage_conversation_history
$ jupyter notebook conversation_history.ipynb

⑧. manage_conversation_history

会話履歴管理：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/manage_conversation_history

テスト：(venv_py311) で実施のこと

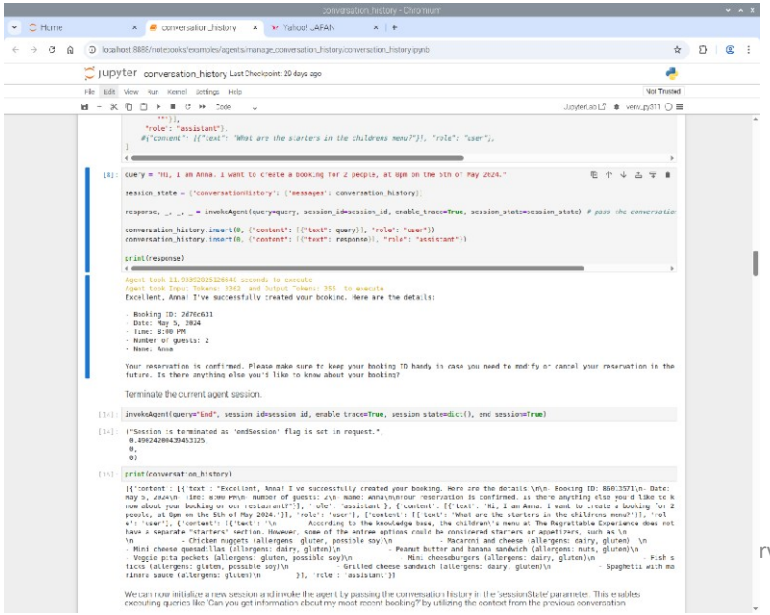
使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

エージェントへの予約：q:Hi, I am Anna. I want to create a booking for 2 people, at 8pm on the 5th of May 2024



回答
Your table is now reserved. Please make sure to arrive on time, and if you need to make any changes or cancellations, don't hesitate to contact us with your booking ID. Enjoy your meal!

Spectrum Technology

3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/manage_conversation_history
$ jupyter notebook conversation_history.ipynb

⑧. manage_conversation_history

会話履歴管理：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/manage_conversation_history

テスト：(venv_py311) で実施のこと

使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

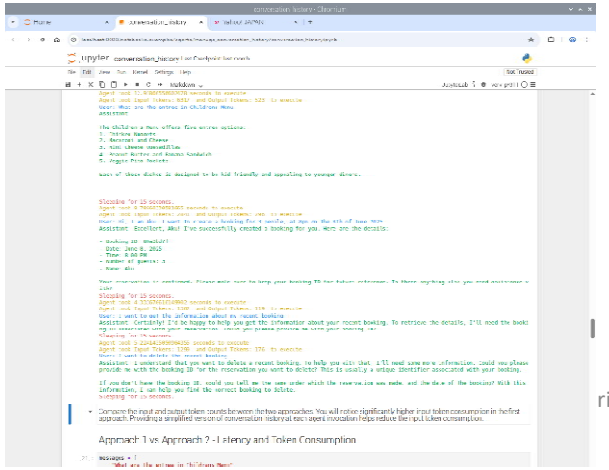
エージェントへの予約：

"What are the entree in Childrens Menu",

"Hi, I am Aku. I want to create a booking for 3 people, at 8pm on the 8th of June 2025.",

"I want to get the information about my recent booking",

"I want to delete the recent booking



履歴がある場合は、予約の取消可能

3. エージェント（jupyter notebook）

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/manage_conversation_history
$ jupyter notebook conversation_history.ipynb

⑧. manage_conversation_history

会話履歴管理：レストラン予約（テーブル予約、メニュー設定）

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/manage_conversation_history

テスト：(venv_py311) で実施のこと

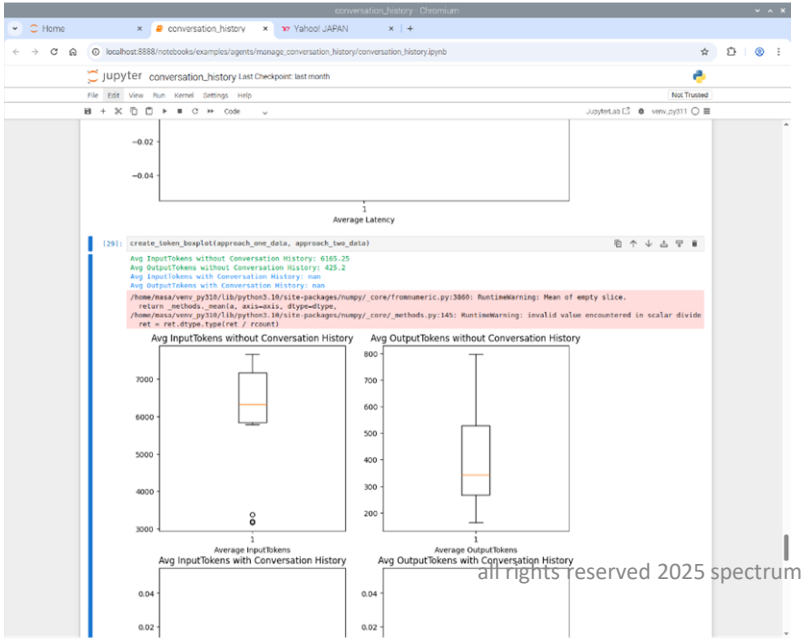
使用AWSサービス：bedrock,oss,s3,lambda,dynamodb,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作

エージェントへの予約の履歴あるなしでの応対時間とトークン数の比較グラフ：一部省略



3. エージェント（jupyter notebook)

コマンド入力
$ source /home/pi/venv_py311/bin/activate
$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/user_confirmation_agents/create-agent-with-API-schema-and-user-confirmation
$ jupyter notebook
create-agent-with-API-schema-and-user-confirmation.ipynb

⑫. user_confirmation_agents

create-agent-with-API-schema-and-user-confirmation

保険問い合わせ：open APIスキーマを使用

https://github.com/awslabs/amazon-bedrock-agent-samples/tree/main/examples/agents/user_confirmation_agents

テスト：(venv_py311) で実施のこと

$ source /home/pi/venv_py311/bin/activate

$ cd /home/pi/Documents/aiagent/amazon-bedrock-agent-samples/examples/agents/user_confirmation_agents/create-agent-with-API-schema-and-user-confirmation

$ jupyter notebook

create-agent-with-API-schema-and-user-confirmation.ipynb

使用AWSサービス：bedrock,oss,s3,lambda,iam(使用後は削除確認のこと。課金事故防止）

動作

該当のipynbを選択し、kernelがvenv_py311となっていることを確認

Pipのインストールは、実施しないで順番にjupyter notebookを動作