

【Handson】標準フローを構築する

前提条件（など）

必要なツール/権限

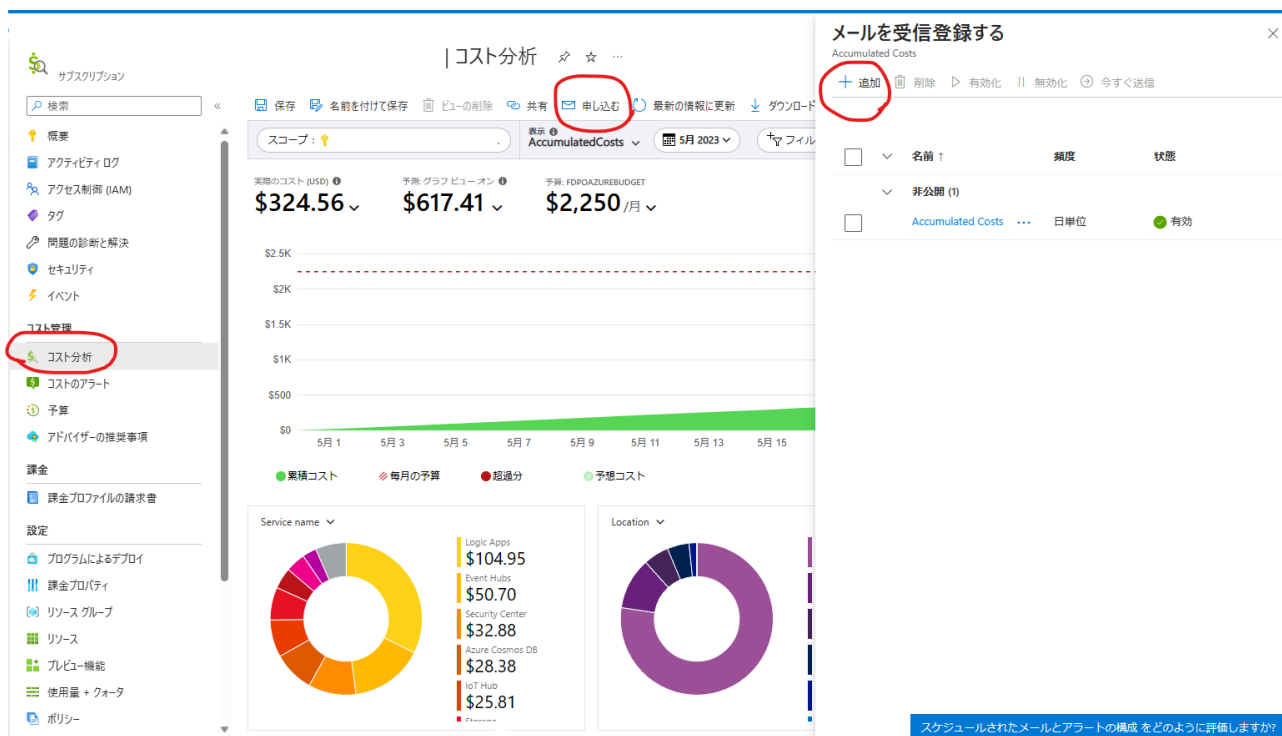
- Azure 管理ポータルにログインして、リソースを作成できる環境・権限が必要です。
手順は、サブスクリプションの「所有者」または「共同管理者」で実施する想定です。

- 手順書の操作の結果、Azure 上にリソースが作成され、それぞれのサービスには**費用が発生します**。利用するサービスの内容と、費用をよくご確認ください。

以下のコスト管理の機能などを使って、適切なコストコントロールを実施してください。

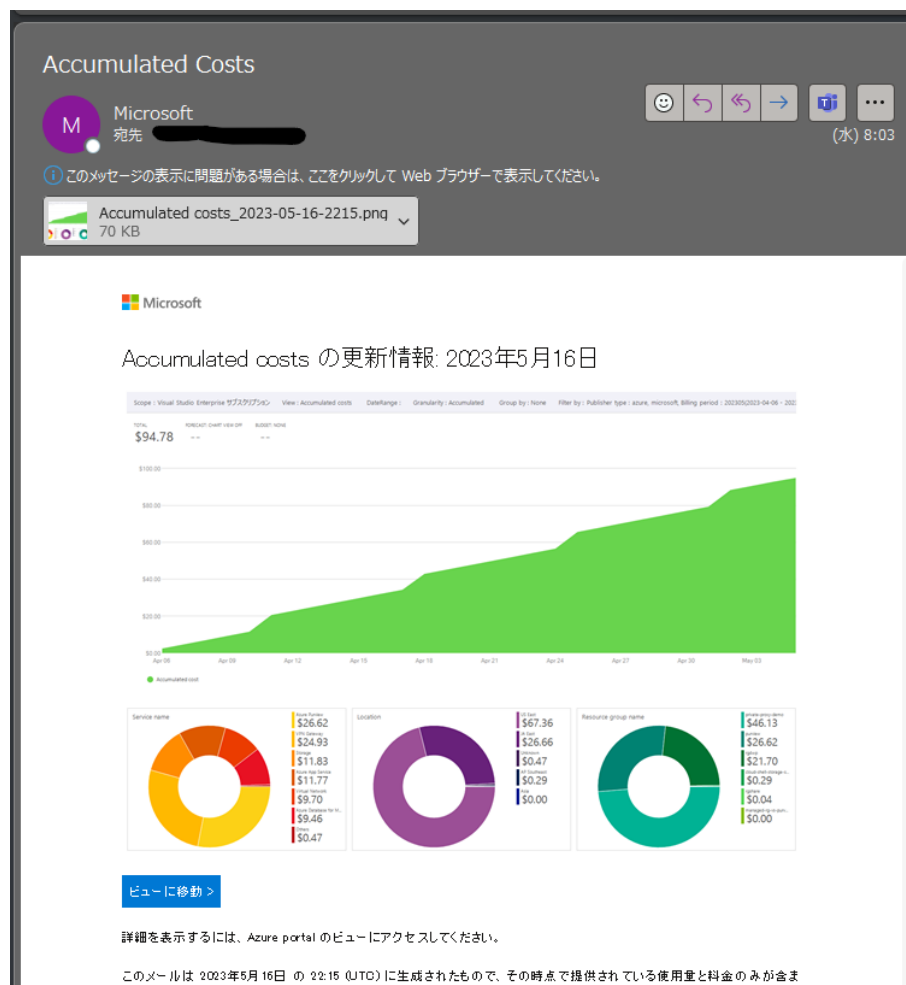
[Cost Management](#) での一般的なコスト分析の使用 - [Microsoft Cost Management](#) | [Microsoft Learn](#)

- **おすすめ機能**： 1 日に 1 回 メールでコスト管理情報を受信する設定が出来ます。



- 「頻度」設定を、カスタムに設定して、曜日に関わらず毎日とすることが出来ます。

メールのイメージはこちらです。



(注意・免責が長かったですね。。。すみません)

GPT モデルをデプロイするための Azure OpenAI サービスを作成する Create the Azure OpenAI service to deploy a GPT model

プロンプト フローで LLM モデルを使用するには、まず Azure OpenAI を使用してモデルをデプロイする必要があります。

Azure OpenAI リソースをプロビジョニングする

まだお持ちでない場合は、Azure サブスクリプションで Azure OpenAI リソースをプロビジョニングします。

1. で [Azure portal](https://portal.azure.com) にサインインします。<https://portal.azure.com>
2. 次の設定で **Azure OpenAI** リソースを作成します。
 - **サブスクリプション**: Azure OpenAI サービスへのアクセスが承認されている Azure サブスクリプションを選択します
 - **リソース グループ**: [rg-prompt-flow-labs](#)
 - **地域**: 利用可能な地域からランダムに選択します*
 - **名前**: 任意の一意の名前
 - **価格レベル**: Standard S0

* Azure OpenAI リソースは、リージョンのクォータによって制約されます。リージョンをランダムに選択すると、他のユーザーとサブスクリプションを共有しているシナリオで、1 つのリージョンがクォータ制限に達するリスクが軽減されます。演習の後半でクォータ制限に達した場合は、別のリージョンに別のリソースを作成する必要がある可能性があります。

3. デプロイが完了するのを待ちます。次に、デプロイされた Azure OpenAI リソースを Azure portal で移動します。

エンドポイントとキーを保存する

Azure OpenAI でデプロイされたモデルの使用を Azure Machine Learning に承認するには、エンドポイントとキーが必要です。

1. Azure OpenAI リソースの **[キーとエンドポイント]** ページに移動します。
2. **エンドポイント**の値をコピーし、後で使用するためにメモ帳に保存します。値は のようになります。

<https://<your-openai-service-name>.openai.azure.com/>

3. **キー**の 1 つの値をコピーし、後で使用するためにメモ帳に保存します。

モデルをデプロイする

Azure OpenAI には、モデルのデプロイ、管理、探索に使用できる **Azure OpenAI Studio** という名前の Web ベースのポータルが用意されています。Azure OpenAI Studio を使用してモデルをデプロイします。

1. Azure OpenAI リソースの **【概要】** ページで、[Azure OpenAI Studio に移動] ボタンを使用して、新しいブラウザー タブで **Azure OpenAI Studio** を開きます。
2. Azure OpenAI Studio で、**【デプロイ】** ページに移動します。
3. 次の設定で gpt-35-turbo/gpt-4/ gpt-4-1106-preview の新しいデプロイを作成します。
 - **モデル:** `gpt-35-turbo`
 - **モデル バージョン:** 既定値に自動更新
 - **デプロイ名:** `gpt-35-turbo`

LLM モデルがデプロイされたので、デプロイされたモデルを呼び出すフローを Azure Machine Learning で作成できます。

フローを作成するために必要なリソースをプロビジョニングする

プロンプト フローを使用するには、まず Azure Machine Learning ワークスペースをプロビジョニングする必要があります。

Azure Machine Learning リソースをプロビジョニングする

まだお持ちでない場合は、Azure サブスクリプションで Azure Machine Learning リソースをプロビジョニングします。

1. の [Azure portal](https://portal.azure.com) に移動します。 `https://portal.azure.com`
2. 次の設定で新しい **Azure AI Studio** リソースを作成します (それ以外はすべて既定のままにします)。
 - **サブスクリプション:** Azure サブスクリプション
 - **リソース グループ:** `rg-prompt-flow-labs`
 - **ワークスペース名:** `mlw-prompt-flow-labs`
 - **地域:** 任意のリージョンを選択してください
3. ワークスペースとそれに関連するリソースが作成されるのを待ちます - 通常、これには約 5 分かかります。

コンピュータ・インスタンスランタイムの作成

プロンプトフローを実行するには、ノードを実行するためにコンピュータ・インスタンスにインストールされている必要なパッケージを含むランタイムが必要です。単純なフローを作成しているので、既定の環境を使用して必要なランタイムを作成できます。

1. Azure Machine Learning を起動します。
2. 「ビルド」から「新しいプロジェクト（任意の名前）」を作成します。
3. 上記で作成したプロジェクトに入り「設定」を選択します。
4. [プロンプトフローランタイム]の「すべて表示」に進み、ランタイムを設定します。
5. 「コンピュータ・インスタンス」タブで、次の設定で新しい**コンピュータ・インスタンスランタイム**を作成します。
 - **コンピューティング名**: 名前を入力します
 - **仮想マシンの種類**: CPU
 - **仮想マシンのサイズ**: Standard_E4ds_v3

コンピュータ・インスタンスランタイムが作成されて実行が開始されるのを待っている間に、次のステップに進むことができます。

note: コンピュータ・インスタンスをランタイムとして使用するには、**認証**が必要な場合があります。

1. コンピューティング インスタンスの自動再起動を求められたら、**[確認]** を選択します。

接続を作成する

プロンプト フローで NER を実行するには、Azure OpenAI でデプロイした GPT モデルを呼び出します。デプロイされたモデルを呼び出すには、Azure OpenAI リソースのエンドポイントとキーを格納する接続を作成する必要があります。

1. 「設定」「接続」と進みます。
2. 次の設定で新しい Azure OpenAI 接続を作成します。
 - **名前**: aoai
 - **プロバイダー**: Azure OpenAI
 - **サブスクリプション ID**: サブスクリプションを選択します
 - **Azure OpenAI アカウント名**: Azure Open AI リソースの名前を選択します
 - **API キー**: 保存したキーを入力します
 - **API ベース**: 保存したエンドポイントを入力します
 - **API タイプ**: azure

- **API バージョン:** 2023-07-01-preview

ランタイムを作成する

フローを実行するには、ノードを実行するためにコンピュート・インスタンスにインストールされている必要なパッケージを含むランタイムが必要です。単純なフローを作成しているので、既定の環境を使用して必要なランタイムを作成できます。

2. 「プロンプト・フロー」ページで、「ランタイム」タブを選択します。
3. 次の設定で新しいランタイムを作成します。
 - **ランタイム名:** 名前を入力します
 - **AzureML コンピューティング インスタンスの選択:** 前のセクションで作成したコンピューティング インスタンスを選択します
 - **カスタムアプリケーション:** New
 - **環境:** Use default environment

ノート: コンピュート・インスタンスをランタイムとして使用するには、**認証**が必要な場合があります。

4. コンピューティング インスタンスの自動再起動を求められたら、**[確認]** を選択します。

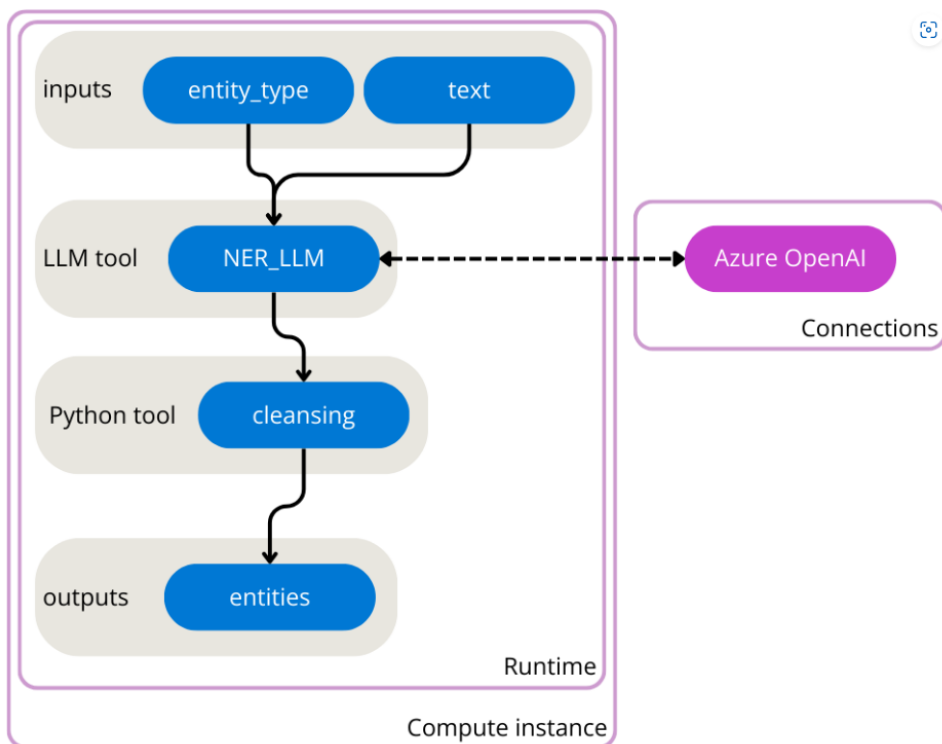
コンピューティング インスタンスが再起動し、新しいランタイムが作成されます。コンピュート・インスタンスの再起動中に、次のステップに進むことができます。

必要なリソースがすべてプロビジョニングされたので、フローを作成できます。

■ 演習 1

大規模言語モデル(LLM)を使用して固有表現抽出 (NER) を実行します。テキストを入力として受け取り、エンティティを出力するアプリケーションを作成するには、プロンプト・フローで LLM ノードを使用するフローを作成します。

この演習では、Azure Machine Learning のプロンプト フローを使用して、エンティティ型とテキストを入力として想定する LLM アプリケーションを作成します。LLM ノードを介して Azure OpenAI から GPT モデルを呼び出し、指定されたテキストから必要なエンティティを抽出し、結果をクリーンアップして、抽出されたエンティティを出力します。



新しいフローを作成する

テンプレートを使用して新しいフローを作成するには、開発するフローの種類を 1 つ選択します。

1. 「プロンプト・フロー」ページで、「フロー」タブを選択します。
2. 新しい標準フローを作成し、フォルダ名として入力します。entity-recognition

1 つの入力、2 つのノード、および 1 つの出力を持つ標準フローが作成されます。フローを更新して、2 つの入力を受け取り、エンティティを抽出し、LLM ノードからの出力をクリーンアップして、エンティティを出力として返します。

入力を構成する

作成するフローは、テキストと、テキストから抽出するエンティティの種類の 2 つの入力を受け取ります。

1. 名前を付けた新しいフローを作成すると、フローがスタジオで開きます。`entity-recognition`
2. **[入力]** で、`topicstring` という名前の 1 つの入力が構成されます。既存の入力を変更し、次の設定で更新します。

`topicstring`

- 名前: `entity_type`
 - タイプ: `string`
 - 値: `job title`
3. **[入力の追加]** を選択します。
 4. 2 番目の入力を次の設定に構成します。
 - 名前: `text`
 - タイプ: `string`
 - 値: `The software engineer is working on a new update for the application.`

LLM ノードの設定

標準フローには、LLM ツールを使用するノードがすでに含まれています。ノードはフローの概要にあります。デフォルトのプロンプトでは、ジョークが要求されます。LLM ノードを更新して、前のセクションで指定した 2 つの入力に基づいてエンティティを抽出します。

1. `joke` という名前の **LLM ノード** に移動します。
2. 名前を `NER_LLM`
3. **[接続]** で、前に作成した接続を選択します。 `aoai`
4. `deployment_name` の場合は、Azure OpenAI にデプロイしたモデルを選択します。 `gpt-35-turbo`

5. プロンプト フィールドを次のコードで置き換えます。

コード写し

system:

Your task is to find entities of a certain type from the given text content.
If there're multiple entities, please return them all with comma separated, e.g.
"entity1, entity2, entity3".
You should only return the entity list, nothing else.
If there's no such entity, please return "None".

user:

Entity type: {{entity_type}}
Text content: {{text}}
Entities:

6. [入力の検証と解析] を選択します。
7. LLM ノードの [Inputs] セクションで、次のように構成します。

- には、値 を選択します。entity_type\${inputs.entity_type}
- には、値 を選択します。text\${inputs.text}

LLM ノードは、エンティティの種類とテキストを入力として受け取り、指定したプロンプトに含めて、Azure OpenAI にデプロイされたモデルに要求を送信します。

Python ノードの構成

モデルの結果から重要な情報のみを抽出するには、[Python] ツールを使用して LLM ノードの出力をクリーンアップします。

1. という名前の Python ノードに移動します。echo

2. 名前を に置き換えます。cleansing
3. コードを次のように置き換えます。

コード写し

```
from typing import List
from promptflow import tool

@tool
def cleansing(entities_str: str) -> List[str]:
    # Split, remove leading and trailing spaces/tabs/dots
    parts = entities_str.split(",")
    cleaned_parts = [part.strip(" ¥t.¥""") for part in parts]
    entities = [part for part in cleaned_parts if len(part) > 0]
    return entities
```

4. [入力の検証と解析] を選択します。
5. Python ノードの[Inputs]セクションで、の値を に設定します。entities_str\${NER_LLM.output}

出力を構成する

最後に、フロー全体の出力を構成できます。フローへの出力は 1 つだけで、抽出されたエンティティである必要があります。

1. フローの [出力] に移動します。
2. [名前] に「」と入力します。entities
3. [値] で を選択します。\${cleansing.output}

フローを実行する

フローを開発したので、フローを実行してテストできます。入力にデフォルト値を追加したので、スタジオでフローを簡単にテストできます。

1. 前のセクションで作成したランタイムを **【ランタイム】** で選択します。
2. **【実行】** を選択してフローをテストします。
3. 実行が完了するまで待ちます。
4. **【出力の表示】** を選択します。ポップアップが表示され、デフォルトの入力の出力が表示されます。必要に応じて、ログを調べることもできます。
5. **Azure リソースを削除する**

■ 追加演習

① バリエーション機能を実行し、フローの精度を高める

② バルク評価の実行

[名前付け規則を定義する - Cloud Adoption Framework | Microsoft Learn](#)

[Azure リソースの省略形の例 - Cloud Adoption Framework | Microsoft Learn](#)