



第7回（全7回） | 2026年06月18日

# Agentic AI勉強会

## AWS CLI & Kiro

セットアップと利用例

# 今日の内容

- 利用手続き
  - ✓ 登録時に必要な情報／提供される認証情報
  - ✓ パスワード再設定
- AWS CLIセットアップ
  - ✓ インストール
  - ✓ 認証情報設定
- Aws Kiro
  - ✓ 概要
  - ✓ インストール、サインイン設定、初期設定
  - ✓ 利用例

# 利用手続き

## 登録時に必要な情報

- メールアドレス
- 氏名
- 所属
- 利用サービス (AWS CLI / Kiro)

**i** AWS CLIは提供数に制限なし  
Kiroは最大20アカウント  
(1チーム2アカウント想定)

## 提供される認証情報

### AWS CLI用

- ユーザーID : `umpuserN` (*N* は2桁の数(01~))
- 初期パスワード
- コンソールサインイン URL
- Access key ID
- Secret access key

### Kiro用

- Kiro用ユーザーID : `umpkuserN`
- 初期パスワード
- AWS access portal URL

コンソールサインインURL :

<https://465239007752.signin.aws.amazon.com/console>

AWS access portal URL :

<https://d-906670dca7.awsapps.com/start>

# AWSパスワード再設定

## 手順

- 1 サインインページにアクセス  
`https://465239007752.signin.aws.amazon.com/console` にアクセスする
- 2 初期パスワードでサインイン  
ユーザーID → 初期パスワードでサインインする
- 3 新しいパスワードを設定  
画面の指示に従い新しいパスワードを入力・確認する
- 4 成功メッセージを確認  
”パスワードのリセットに成功しました” の表示を確認し【サインインを続行】を押す
- 5 サインアウト  
画面右上のユーザーIDをクリック → サインアウトする

**IAM ユーザーサインイン**

アカウント ID またはエイリアス  
(持っていませんか?)  
465239007752

このアカウントを記憶する

IAM ユーザー名  
umpuser00

パスワード  
.....

パスワードを表示 お困りですか?

**サインイン**

ルートユーザーの E メールを使用したサインイン

[新しい AWS アカウントを作成](#)

**パスワードのリセット**

アカウント (465239007752) のパスワードの有効期限が切れているか、リセットが必要です。

続行するには、古いパスワードを確認し、umpuser00 の新しいパスワードを設定してください。(お客様ではありませんか?)

古いパスワード  
.....

パスワードを表示

新しいパスワード  
.....

新しいパスワードの確認  
.....

パスワードを表示 一致

**パスワード変更の確認**

[別のアカウントにサインインする](#)

✔ パスワードのリセットに成功しました

**パスワードのリセット**

IAM ユーザー (umpuser00) パスワードが正常にリセットされました。

**サインインを続行**

# AWS Kiro用パスワード再設定

## 手順

- 1 AWS access portal URLにアクセス  
ブラウザで提供されたAWS access portal URLを開く
- 2 初期パスワードでサインイン  
KiroユーザーID → 初期パスワードでサインインする
- 3 新しいパスワードを設定  
新規パスワードを入力し、確認欄にも同じパスワードを入力する
- 4 サインアウト  
画面右上のアイコンをクリック → サインアウトする

**サインイン**

ユーザー名  
umpkuser00

次へ

**サインイン**

ユーザー名:  
umpkuser00 (ご自分ではありませんか?)

パスワード  
.....

パスワードの表示 [パスワードを忘れた](#)

サインイン

**新しいパスワードを設定**

ユーザー名: umpkuser00

新規パスワード  
.....

新しいパスワードの確認  
.....

パスワードの表示 一致

新しいパスワードを設定

# AWS CLI セットアップ

## 開発環境の事前準備

### ターミナル環境

WSL(Windows Subsystem for Linux)、PowerShell、Lima など

### プログラム言語

Python、Java、JavaScript / TypeScript など

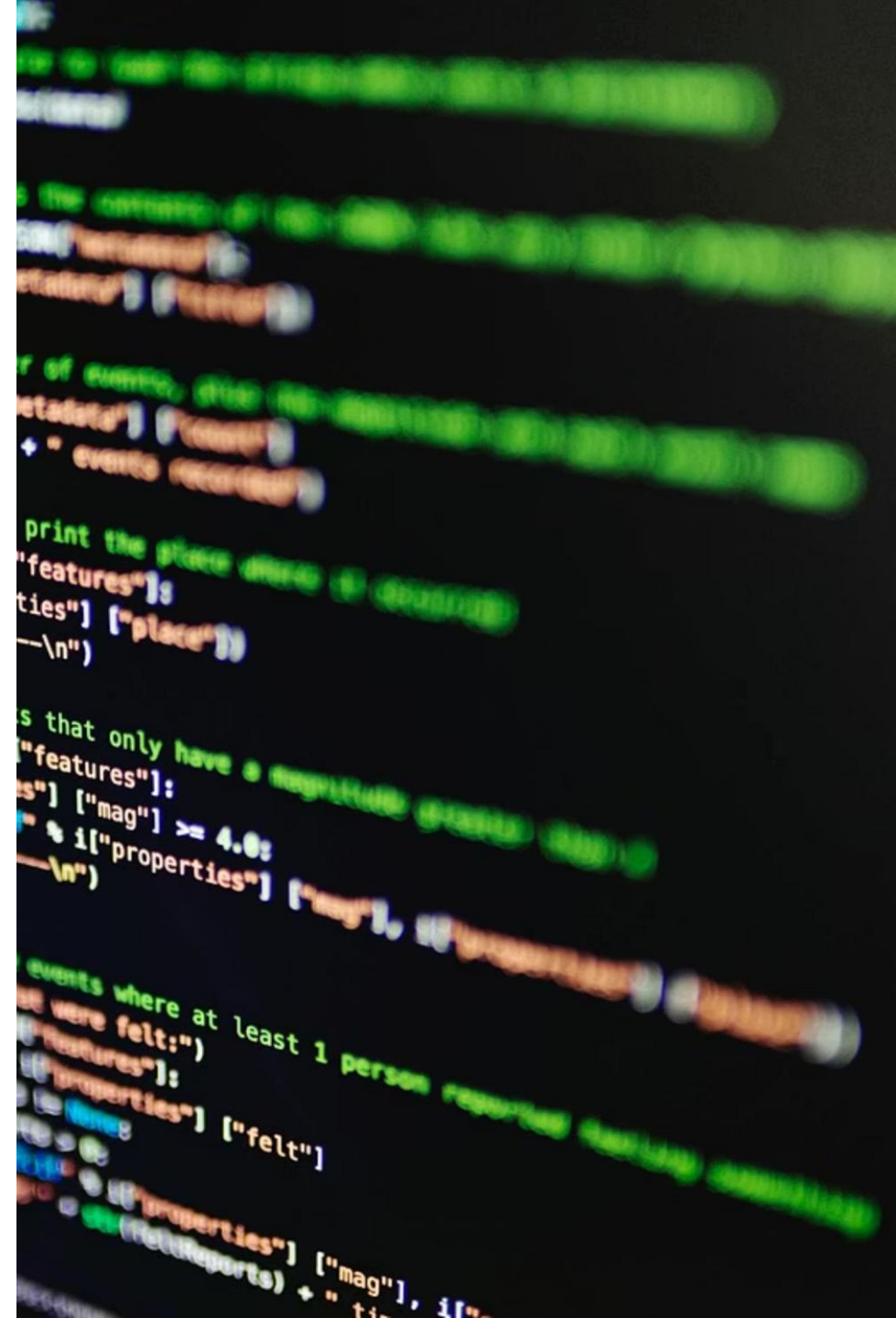
### エディタ

VSCode など

### AWS CLI

ローカル環境に合った最新バージョンをインストールする

- ① AWS Documentation: AWS CLI の最新バージョンのインストールまたは更新  
[https://docs.aws.amazon.com/ja\\_jp/cli/latest/userguide/getting-started-install.html](https://docs.aws.amazon.com/ja_jp/cli/latest/userguide/getting-started-install.html)



# AWS CLI セットアップ

## AWSユーザー認証情報の設定

ターミナルで以下のコマンドを実行し、提供された情報を入力する。

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

提供された情報を入力

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

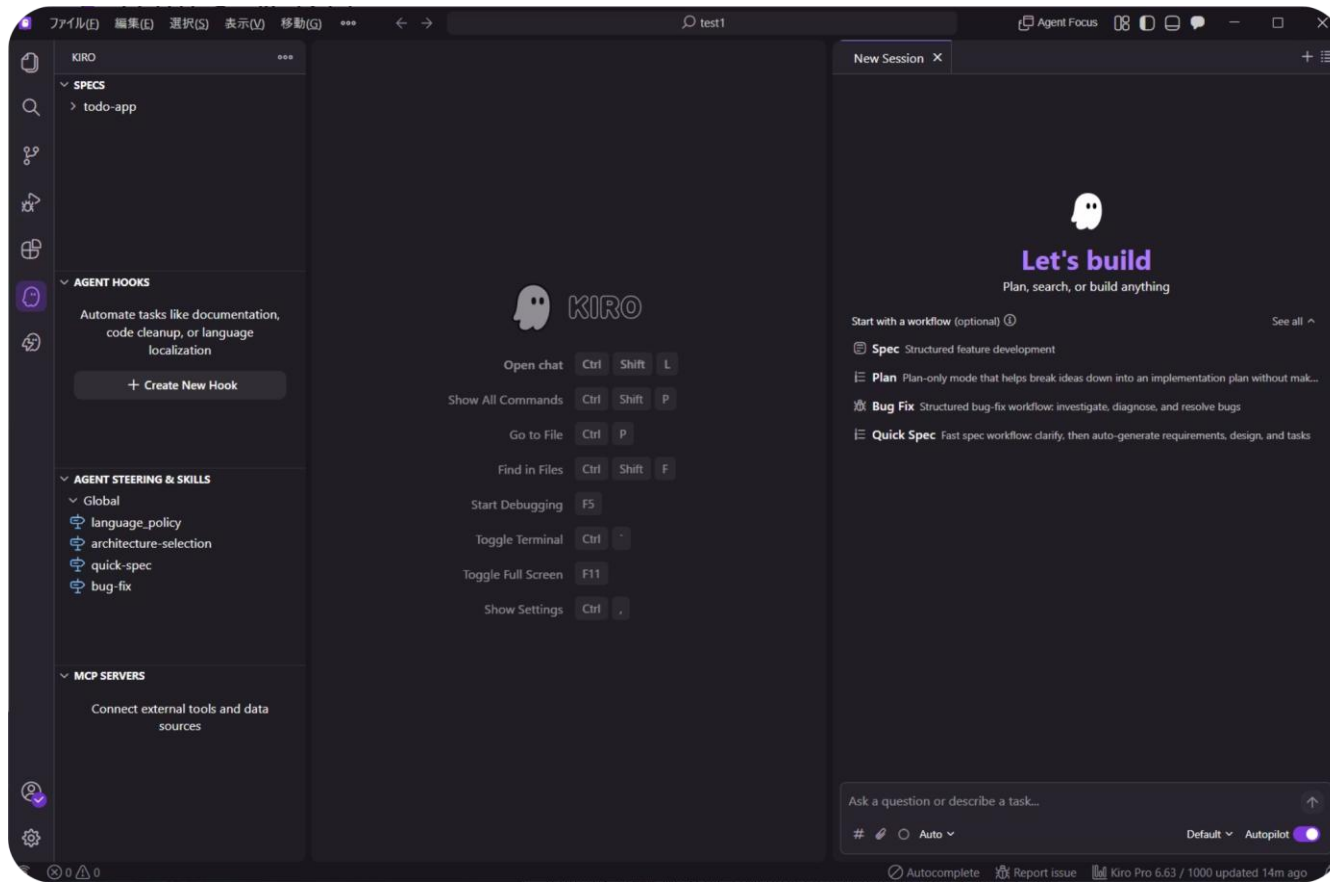
```
Default region name [None]: us-east-1
```

```
Default output format [None]: json
```

} 規定値



# AWS Kiro



Amazon(AWS)が開発・提供する、AIを組み込んだ  
Agentic IDE(統合開発環境)。2026年5月正式リリース。

VS Codeのオープンソース版(Code OSS)ベース  
であるため、見た目や操作感はVS Codeとほぼ同じ。  
既存のVS Code拡張機能やキーバインドも引き継げる。

✔️ VS Codeユーザーはほぼそのまま移行可能。  
拡張機能・キーバインドも継続利用できる。

# AWS Kiro の特長

## 仕様駆動開発 (Spec-Driven Development)

「仕様書(スペック)を作業の起点にする」という考え方。

新機能を実装する際、まず「何を作りたいか」を自然言語で説明すると、Kiroが以下の3つの構造化ファイルを自動生成する。



### requirements.md

ユーザーストーリーと受け入れ条件



### design.md

システム設計、コンポーネント構成、  
データフロー



### tasks.md

実装タスクの一覧

仕様書が正であり優先されるべき「真実」。コードはこれらの仕様書から生成される「成果物」という位置づけ。

- ❑ 他のAIコーディングツール(Cursor、GitHub Copilot、Claude Code など)は「チャットファースト」：プロンプトを打ち、AIがコードを書き、反復するアプローチを採用している。



# AWS Kiro セットアップ

公式サイト: <https://kiro.dev/>

1

サイトのダウンロードページへ  
右上の【DOWNLOADS】ボタンをクリック

2

インストーラをDL  
手元のPC環境(Windows / macOS / Linux)に適したインストーラ  
をダウンロード

3

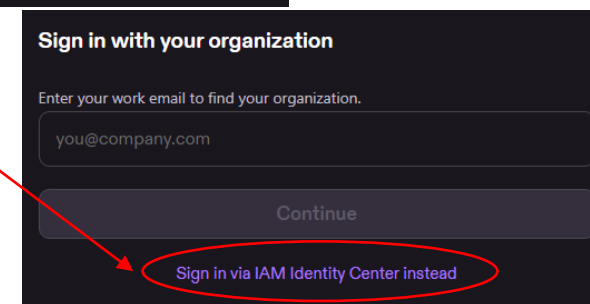
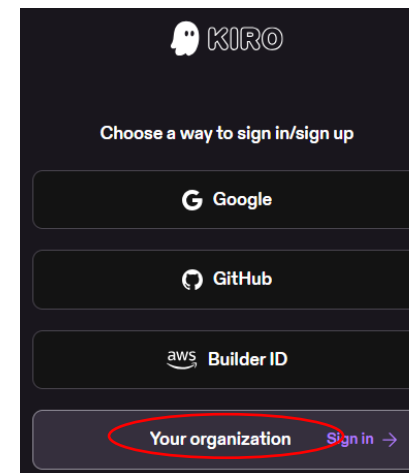
インストール実行  
ダウンロードしたインストーラを実行してインストールする

4

Kiroを起動  
インストール完了後、Kiroを起動する

# AWS Kiro サインイン設定

- 1 [Sign in] を押す  
webブラウザでサインイン手段を選択する画面が開くので  
[Your organization] を選択する
- 2 入力欄を無視して ”Sign in via IAM Identity Center instead”  
のリンクをクリックする
- 3 以下の接続情報を入力して [Continue]
  - Start URL: 提供されたAWS access portal URL
  - Region: us-east-1
- 4 AWSサインイン画面でサインイン  
KiroユーザーID → 先程変更したパスワードでサインインする
- 5 ”Kiro IDE がデータにアクセスすることを許可しますか？”  
と尋ねられるので【アクセスを許可】を押す
- 6 Kiro に戻る。ブラウザ画面は閉じて良い



## Kiro IDE がデータにアクセスすることを許可しますか？

【アクセスを許可】を選択すると、Kiro IDE以下へのアクセスを許可することに同意したものとみなされます。

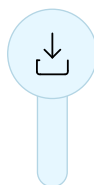


アクセスを拒否

アクセスを許可

# AWS Kiro 初期設定

サインイン完了後、以下の手順でKiroの初期設定を行う。



VS Code設定のインポート

”Import from VS Code” か ”Skip” を押す。既存のVS Code設定を引き継ぐ場合はインポートを選択する。



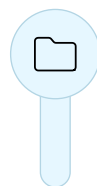
テーマを選択

好みのカラーテーマを選択して [Continue] を押す



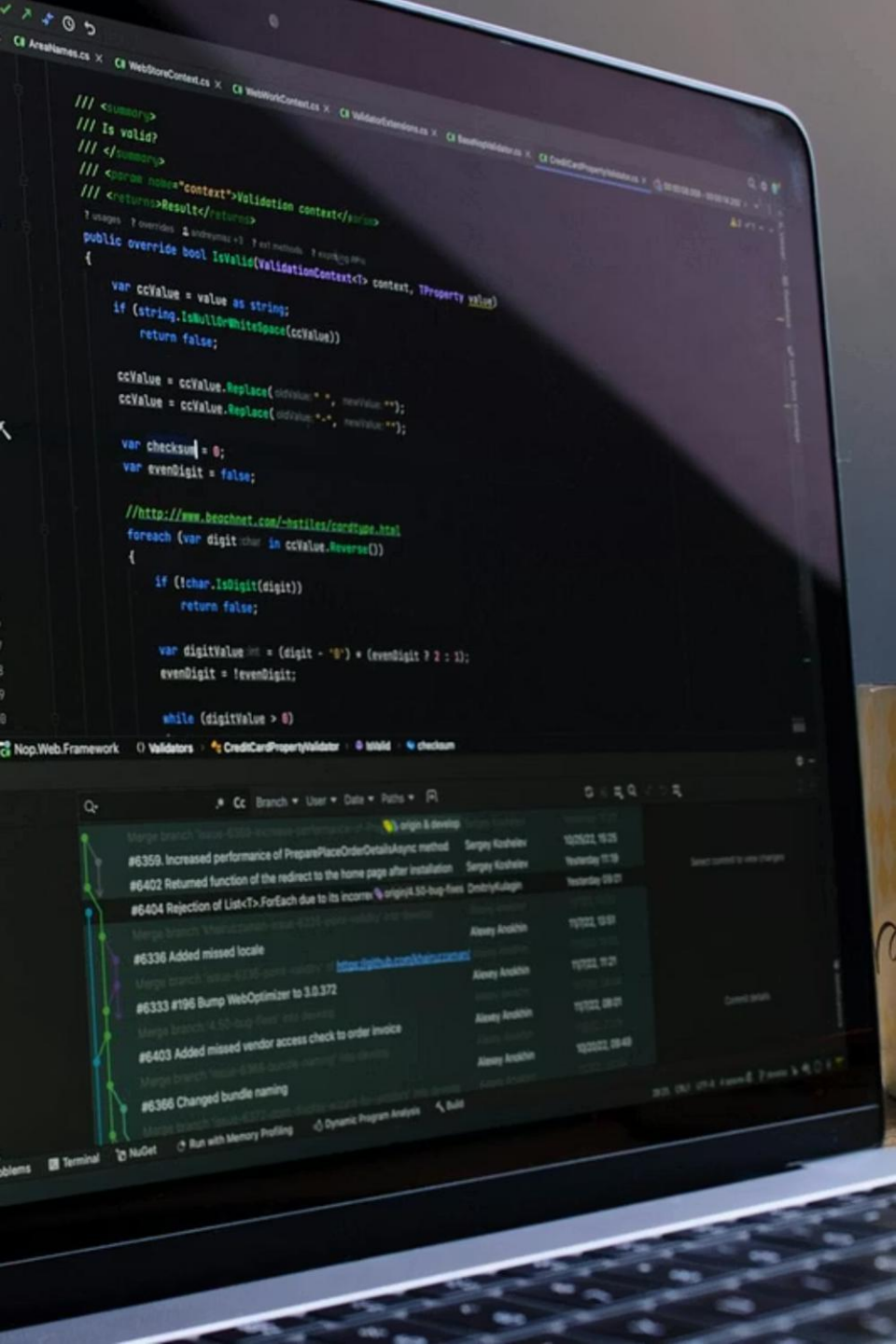
セットアップ完了

[Done] を押してセットアップを完了する



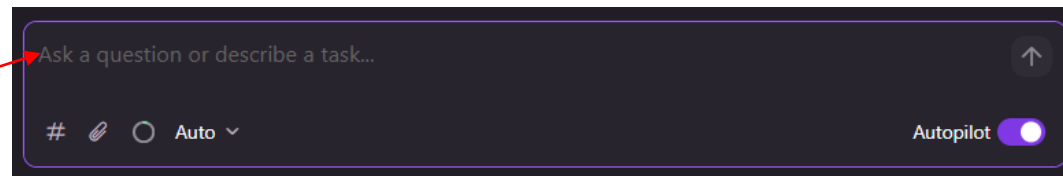
プロジェクトを開く

[Open a project] で、新しいプロジェクト用のフォルダを選択して開発を開始する



# Kiro を試してみる

1. 勉強会第4回で行なった、Bedrock上のエージェントを作成しておく
2. [Open a project] → フォルダを指定
3. ターミナルを1つ開く
4. Kiroのチャットに下記プロンプトを入力:



(step 1)

“Pythonを使って簡単なTODO管理のwebアプリを作成したい。段階的に進めるようガイドして”

(step 2) ※勉強会第4回で行なった、Bedrock上のエージェントが既に構築済である前提

- i. “Pythonを使って、Amazon Bedrock上で作成したAI技術顧問エージェントとのやり取りを行なうwebアプリを作成したい。AWS CLIは実装済み、AIエージェントとの通信では .env ファイルで指定する AGENT\_ID, AGENT\_ALIAS\_ID を利用する。ガイドを頼む。”
- ii. “実装。.envは後程手動で編集する前提でテンプレートだけ用意”