

開発環境構築はAIで簡単に！

非エンジニアでも生成AIを活用して短時間で実現

困りごと



- ⚠ 環境構築に時間がかかる
- 🚫 途中でつまずき挫折しやすい
- 🕒 専門知識が必要で難しい

AIによる解決策

- 1 AIに環境構築手順を説明させる
初心者にもわかりやすい解説
▼
 - 2 セットアップスクリプトを自動生成
コードや設定ファイルを瞬時に作成
▼
 - 3 指示に従って実行するだけ
エラー時もAIが解決策を提案
- 🚩 今日のゴール：ClaudeとCursorを使って開発環境を自動セットアップ

AIでできる環境準備の範囲

AIが生成するスクリプト例

```
# AWS EC2インスタンス作成・設定スクリプト
import boto3

# AWS クライアントの初期化
ec2 = boto3.resource('ec2')

# EC2インスタンスの作成
instance = ec2.create_instances(
    ImageId='ami-0c55b159cbfafa1f0', # Amazon Linux 2 AMI ID
    MinCount=1,
    MaxCount=1,
    InstanceType='t2.micro',
    KeyName='my-key-pair', # 既存のキーペア名
    SecurityGroups=[
        'default', # デフォルトセキュリティグループ
    ],
    UserData='''#!/bin/bash
# インスタンスの初期化スクリプト'''
)
```

AIが自動化できる環境準備



AWS設定スクリプトの生成

EC2インスタンス作成からミドルウェア設定まで自動化



GitHubリポジトリ作成・初期化

リポジトリ設定からコミット・プッシュまでのコマンド生成



必要ライブラリの自動検出・インストール

プロジェクト要件から必要なパッケージを提案・導入



設定ファイル（config）のテンプレート生成

各種フレームワークやツールに最適な設定ファイルを作成

👤 人間が手動で行う必要がある作業

アカウント作成・認証

- AWSアカウント登録
- GitHubアカウント設定
- APIキー・認証情報の管理

セキュリティ対応

- アクセス権限設定
- セキュリティグループルール確認
- 機密情報の安全管理

最終確認・運用

- 生成されたコードの確認
- サービス稼働状態の確認
- コスト管理・リソース監視

Claude Codeによる環境準備依頼の進め方

プロンプト例と回答サンプル

Claude Code

AWS EC2にNode.js環境を構築するスクリプトを作成してください。ExpressとMongoDBをインストールして、シンプルなWebアプリケーションが動作するようにしたいです。

AWS EC2インスタンスでNode.js、Express、MongoDBの環境を構築するためのスクリプトを作成します。
このスクリプトは、EC2インスタンスにSSH接続した後に実行することを想定しています：

```
#!/bin/bash
```

```
# システムのアップデート
```

環境準備の依頼と実行の流れ

1

具体的な目標を明示したプロンプトを作成

「何を構築したいのか」「どのようなツールを使いたいのか」を具体的に伝えましょう。例えば「AWS EC2にNode.js環境を構築するスクリプトを作って」のような明確な指示が効果的です。

2

AIが生成したコードを入手

Claude Codeは目的に合わせたスクリプトやコマンドを生成します。必要に応じて追加の質問や修正依頼も可能です。「MongoDB追加」など要件を変更・追加できます。

3

安全確認後にコードを実行

生成されたコードをコピーして実行する前に、コードの意味を理解し安全性を確認しましょう。不明点はClaudeに「このコードの○○部分は何をしているの？」と質問できます。

実行前の安全確認ポイント

コードの理解

- コードが何をしているか理解する
- 不明な部分はAIに質問して確認
- 必要な変更点を明確にする

セキュリティ確認

- 権限設定は適切か
- 機密情報が含まれていないか
- 意図しない通信は発生しないか

段階的実行

- 全体を一度に実行しない
- ブロックごとに分けて実行
- 各ステップの結果を確認しながら進める

Cursorでの環境セットアップ自動化

Cursor AIの補完・修正機能



コード修正・自動提案の例

Cursor: 「Node.jsがインストールされていません。インストールスクリプトを追加しましょうか？」

あなた: 「はい、Node.js 16をインストールしてください」

Cursor: 「スクリプトに次の行を追加しました:
`curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -`
`sudo apt-get install -y nodejs`」

環境構築の効率化ステップ

1

Claudeで作成したスクリプトをCursorに貼り付け

Claudeが生成したセットアップコードをCursorのエディタやターミナルに直接貼り付けて実行できます。

2

エラーが発生したらAIに修正依頼

エラーが出た場合、そのエラーメッセージを選択して「Fix」ボタンを押すだけで、Cursorが解決策を提案します。

- 依存関係の不足を自動検出・インストール
- 構文エラーの修正案を表示
- 環境固有の問題に対応するコード修正

3

フィードバックループで継続改善

AIの提案を実行し、結果をフィードバックすることで、さらに最適化された環境を構築できます。

Cursorの自動化機能



必要ライブラリの自動検出

コード内の依存関係を分析し、不足しているパッケージを提案



コマンド提案

目的に合わせたコマンドやフラグの提案機能



コード補完・生成

入力中のコードを予測し、補完や次のステップを提案



環境最適化提案

コードの実行環境を分析し、効率化の提案を行う

AWS・GitHubをAI経由でサクッと設定

サービスの役割と連携

サービスの役割

 生成AI	 AWS	 GitHub
- スクリプト・コード生成 - 設定手順の提案 - トラブルシューティング - 最新技術の知識提供	- クラウドインフラ提供 - 計算リソース管理 - スケーラビリティ確保 - サービスのホスティング	- コード管理・バージョン管理 - チーム協業基盤 - CI/CD連携 - ソースコード保存

サービス間の連携

連携パターン	連携内容
AI → AWS	- EC2インスタンス作成スクリプト生成 - CloudFormationテンプレート作成 - セキュリティグループ設定支援
AI → GitHub	- リポジトリ初期化コマンド作成 - GitHubActions設定ファイル生成 - READMEやドキュメント生成
AWS ↔ GitHub	- コード変更時の自動デプロイ - AWSリソースへのGitHubからのアクセス - GitHub上のソースからAWSへのCI/CDパイプライン
ユーザー → AI	- 自然言語での要件伝達 - エラー時の解決策リクエスト - コード改善・最適化の依頼

AI経由で自動化できること

-  **AWSインスタンス立ち上げ**
EC2インスタンスのセットアップスクリプトを生成。AMI選択からユーザーデータ設定までを自動化
-  **GitHubリポジトリ初期化**
リポジトリ作成、README作成、初期コミット、プッシュまでのコマンドセットを一括生成
-  **CI/CDパイプライン設定**
GitHubActionsのワークフローファイルやAWS CodePipelineの設定ファイルを自動生成
-  **トラブルシューティング**
エラー発生時のデバッグ手順や修正コマンドを提案

⚙️ 自動化と手動作業の区分

AIに任せられる部分

- ✔️ コード・スクリプト生成
- ✔️ 設定ファイルテンプレート作成
- ✔️ コマンドラインの生成
- ✔️ 基本的なトラブルシューティング

手動で行う必要がある部分

- ❗ 認証情報の入力・管理
- ❗ アクセス許可・権限設定
- ❗ 生成されたコードの検証
- ❗ 最終的な実行判断

まとめと応用例



時間の大幅短縮

数時間～数日かかる環境構築が数分で完了。
学習コストも削減できる



エラー削減

手順の見落としやタイプミスによるトラブル
を防止。AIが整合性を確認



理解しながら進行

AIに説明を求めることで、ただコピペするだ
けでなく仕組みも学べる

💡 応用例



社内ツールの素早い試作

「シンプルなデータ集計ツールをNode.jsで」といった依頼だけで、
基本機能を持ったアプリケーションを短時間でセットアップ



新人教育用テンプレート

新入社員が取り組むプロジェクト環境を統一して用意。説明付きで
学習しやすい開発環境を提供



複数環境の同時構築

開発・テスト・検証など複数環境を同時に、同じ条件でセットアッ
プ。環境差異によるバグを予防



小さく始めてスキルアップ

シンプルな環境から始め、徐々に複雑な構成にステップアップ。AI
の補助で自信をつけながら進められる

ワーク：Claude＋Cursorで環境構築体験

1

Claudeに環境構築を依頼

「Node.jsとExpressが動くAWS EC2環境をセットアップするスクリプト」をClaudeに依頼してみましょう

2

Cursorで実行と修正

生成されたスクリプトをCursorに貼り付け、実行してみましょう。エラーが出た場合はCursorに修正を依頼

3

GitHubへコード保存

初期設定ができれば、リポジトリを作成し、コードをpushしてみましょう。コマンドはAIに生成してもらいましょう

📝 振り返り：AIに任せて楽になった部分を書き出そう

1 AIに任せて楽になったこと・時間短縮できたこと

2 AIを使って学べたこと・理解が深まったこと

3 次回、AIを活用したい環境構築やタスク

※実際の環境へのデプロイは、アカウント認証情報などが必要です。ローカル環境やテスト用アカウントでお試してください。