

バイブルコーディング超えてバイブルプロイ

～**Cloudflare** MCPで実現する、未来のアプリケーションデリバリー～

Workers Tech Talks in Kyoto #1

# 自己紹介

- azukiazusa
- <https://azukiazusa.dev>
- FE（フロントエンド|ファイアーエムブレム）が好き



# 自己紹介

- azukiazusa
- <https://azukiazusa.dev>
- FE (フロントエンド開発者)  
　　→ Cloudflare Workers にホスト

バイブルコーディング

# バイブルコーディングとは？

AIエージェントが自律的にコードを生成・実行する技術

- AI に自然言語で指示を出す
- AI エージェントが主体になってアプリケーションの開発を進める
- 雰囲気・フィーリング・ノリ

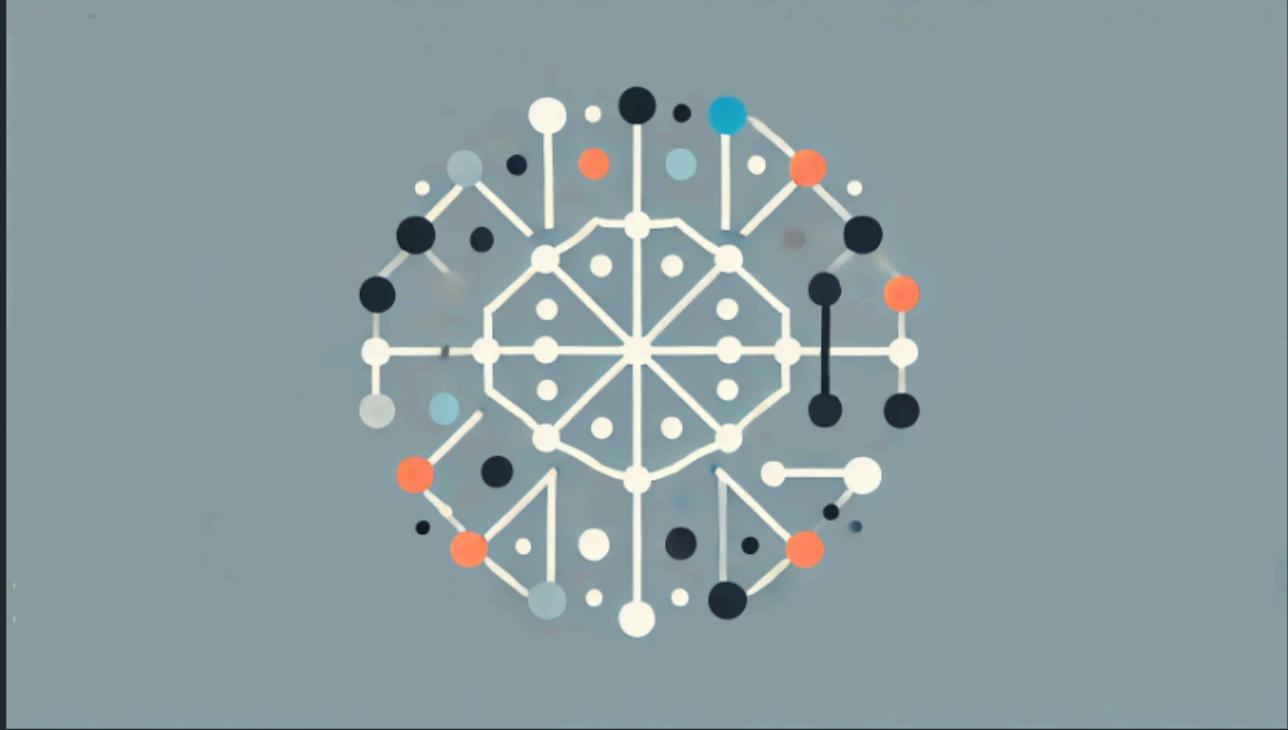
# Cloudflareの特徴

# Cloudflareの特徴

驚異的なデプロイの簡単さ

ドキュメントのボタン  
をクリックするだけで  
デプロイを開始

## Chat Agent Starter Kit



Deploy to Cloudflare

A starter template for building AI-powered chat agents using Cloudflare's Agent platform, powered by [agents](#). This project provides a foundation for creating interactive chat experiences with AI, complete with a modern UI and tool integration capabilities.

もっと気軽にデプロイしたい

バイブルコーディングのようなノリで...

# Cloudflare MCP

13のMCPサーバーを提供

<https://blog.cloudflare.com/ja-jp/thirteen-new-mcp-servers-from-cloudflare/>

サーバー	説明
<a href="#">Cloudflare ドキュメントサーバー</a>	Cloudflareの開発者向けドキュメントから最新の参照情報を入手する
<a href="#">Workers バインディングサーバー</a>	ストレージ、AI、コンピューティングのプリミティブを使用してWorkersアプリケーションを構築する
<a href="#">Workers Observabilityサーバー</a>	デバッグして、Workersアプリケーションのログと分析に関する洞察を得る
<a href="#">コンテナサーバー</a>	サンドボックス開発環境をスピニングアップする
<a href="#">ブラウザレンダリングサーバー</a>	Webページを取得し、マークダウンに変換して、スクリーンショットを作成する
<a href="#">Radarサーバー</a>	全世界のインターネットトラフィックに関する洞察、トレンド、URLスキャン、その他のユーティリティを入手する
<a href="#">Logpushサーバー</a>	Logpushジョブの正常性に関する簡単な概要を入手する
<a href="#">AI Gatewayサーバー</a>	ログを検索し、プロンプトとレスポンスの詳細を入手する
<a href="#">AutoRAGサーバー</a>	AutoRAG上のドキュメントを一覧表示し、検索する
<a href="#">監査ログサーバー</a>	監査ログを照会して、レビュー用レポートを生成する
<a href="#">DNS Analyticsサーバー</a>	現在の設定に基づいてDNSのパフォーマンスを最適化し、問題をデバッグする
<a href="#">Digital Experience Monitoringサーバー</a>	組織の重要なアプリケーションに関する洞察をすばやく入手する
<a href="#">Cloudflare One CASBサーバー</a>	SaaSアプリケーションのセキュリティの設定ミスを迅速に特定し、アプリケーション、ユーザー、データを保護する

# Model Control Protocol (MCP)とは？

- LLMが外部システムと連携するためのプロトコル
- AIエージェントが様々なサービスを直接操作可能

# Workers Bindings Server

- AIエージェントがKVやR2, D1などのCloudflareリソースを操作可能
- 自然言語で指示を出すだけでリソースの作成・削除が可能
- 真の「バイブルデプロイ」

# 実際にやってみよう！

URL短縮サービスを Cloudflare Workers + KV ストレージで実装

# 下準備

- Cloudflare Workers アカウントを作成
- Claude Code をインストール

```
npm install -g @anthropics/clause-code
```

# Claude Code に MCP サーバーを追加

```
claude mcp add --transport sse -s project "Cloudflare Workers" "https://bindings.mcp.cloudflare.com/sse"
```

# .mcp.json ファイルが生成される

```
{  
  "mcpServers": {  
    "Cloudflare Workers": {  
      "type": "sse",  
      "url": "https://bindings.mcp.cloudflare.com/sse"  
    }  
  }  
}
```

# 認証

/mcp コマンドで認証  
が必要

\* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/asai/sandbox/cloudflare-demo

Tips for getting started:

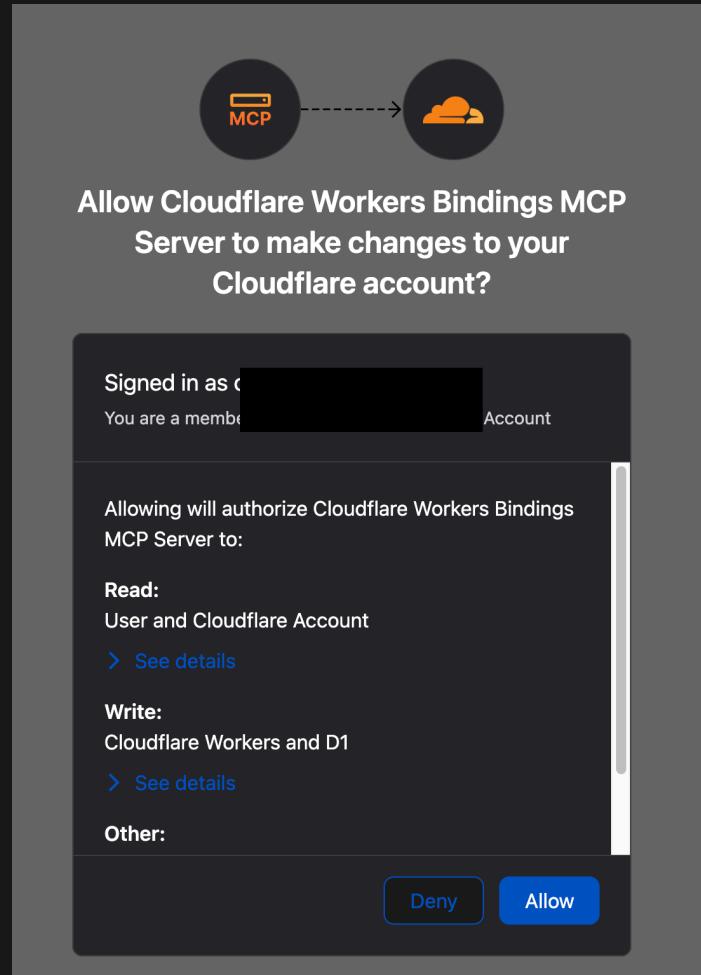
1. Run /init to create a CLAUDE.md file with instructions for Claude
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

Manage MCP servers

› 1. Cloudflare Workers △ disconnected · Enter to login

Esc to exit

# Cloudflare の認証画面



# プロンプト

Claude Codeを使って、URL短縮サービスをCloudflare Workers + KVストレージで実装してデプロイしてください。

要件：

- Hono を使ってHTTPサーバーを実装
- web UI で短縮URLを生成
- Cloudflare KVを使ってURL情報を保存
- 短縮コードは6文字のランダムな英数字

# Claude Code が実行計画を立てる

- web UI で短縮URLを生成
- Cloudflare KVを使ってURL情報を保存
- 短縮コードは6文字のランダムな英数字

- CloudflareのMCPが接続されているので、Cloudflare Workers + KVストレージを使ったURL短縮サービスを実装します。

- **Update Todos**
  - └ □ Cloudflareアカウントの確認とKVネームスペースの作成
    - プロジェクトの初期化とpackage.jsonの作成
    - Hono を使ったWorkerのAPIエンドポイント実装
    - Web UIの実装（HTML/CSS/JS）
    - wrangler.tomlの設定
    - デプロイとテスト

- **Update Todos**
  - └ □ Cloudflareアカウントの確認とKVネームスペースの作成
    - プロジェクトの初期化とpackage.jsonの作成
    - Hono を使ったWorkerのAPIエンドポイント実装
    - Web UIの実装（HTML/CSS/JS）
    - wrangler.tomlの設定
    - デプロイとテスト

- **Cloudflare Workers:kv\_namespaces\_list (MCP)(params: {})**

**Tool use**

```
Cloudflare Workers:accounts_list() (MCP)
List all accounts in your Cloudflare account
```

# KV が MCP を通じて生成される

```
□ デプロイとテスト
└ No (tell Claude what to do differently)

● Cloudflare Workers:kv_namespaces_list (MCP)(params: {})
└ Interrupted by user

> continue
└ {
    "accounts": [
        {
            ...
            ... +7 lines (ctrl+r to expand)
}

● Cloudflare Workers:kv_namespaces_list (MCP)(params: {})
└ {
    "namespaces": [
        {
            ...
            ... +22 lines (ctrl+r to expand)
}

● Cloudflare Workers:kv_namespace_create (MCP)(title: "URL_SHORTENER")
└ {
    "id": "c66c2c00057e4c3b960f9440d464f6fa",
    "title": "URL_SHORTENER",
    ...
    ... +3 lines (ctrl+r to expand)
}

● Update Todos
└ □ Cloudflareアカウントの確認とKVネームスペースの作成
  □ プロジェクトの初期化とpackage.jsonの作成
  □ Hono を使ったWorkerのAPIエンドポイント実装
  □ Web UIの実装（HTML/CSS/JS）
  □ wrangler.tomlの設定
  □ デプロイとテスト
```

# wrangler deploy で Worker をデプロイ

```
... +5 lines (ctrl+r to see all)

● Bash(npx wrangler login)
└ ⚡ wrangler 3.114.10 (update available 4.23.0)
-----
... +15 lines (ctrl+r to expand)

● Bash(npx wrangler deploy)
└ ⚡ wrangler 3.114.10 (update available 4.23.0)
-----
... +14 lines (ctrl+r to expand)
```

# Worker のデプロイが完了

## ■ アノロイとアベト

- URL短縮サービスのデプロイが完了しました！

サービス URL: <https://url-shortener.azukiazusa.workers.dev>

### 実装した機能

1. 短縮URL生成: 6文字のランダムな英数字で短縮コードを生成
2. Web UI: シンプルで使いやすいインターフェース
3. URL検証: 無効なURLを事前にチェック
4. クリック数追跡: 各短縮URLのアクセス数を記録
5. 統計表示: /stats/{shortCode} でクリック数等の統計を確認可能

### 使用技術

- **Cloudflare Workers:** サーバーレス実行環境
- **Hono:** 軽量なWebフレームワーク
- **Cloudflare KV:** URL情報の永続化ストレージ

サービスが正常に動作しているか確認してください。

> ■

? for shortcuts

# Worker の情報を取得

> 今回作成した worker の情報を取得

- **Cloudflare Workers:workers\_get\_worker (MCP)**(scriptName: "url-shortener")
  - └ name id url-shortener edd24fb00d0148cfa746af336720c737
- **Cloudflare Workers:workers\_get\_worker\_code (MCP)**(scriptName: "url-shortener")
  - └ --6a912d1f1c3ef1b79aab2bbeeb0316c620c968cb21d7b77f20fe89350049Content-Disposition: form-data; name="index.js"

... +1796 lines (ctrl+r to expand)

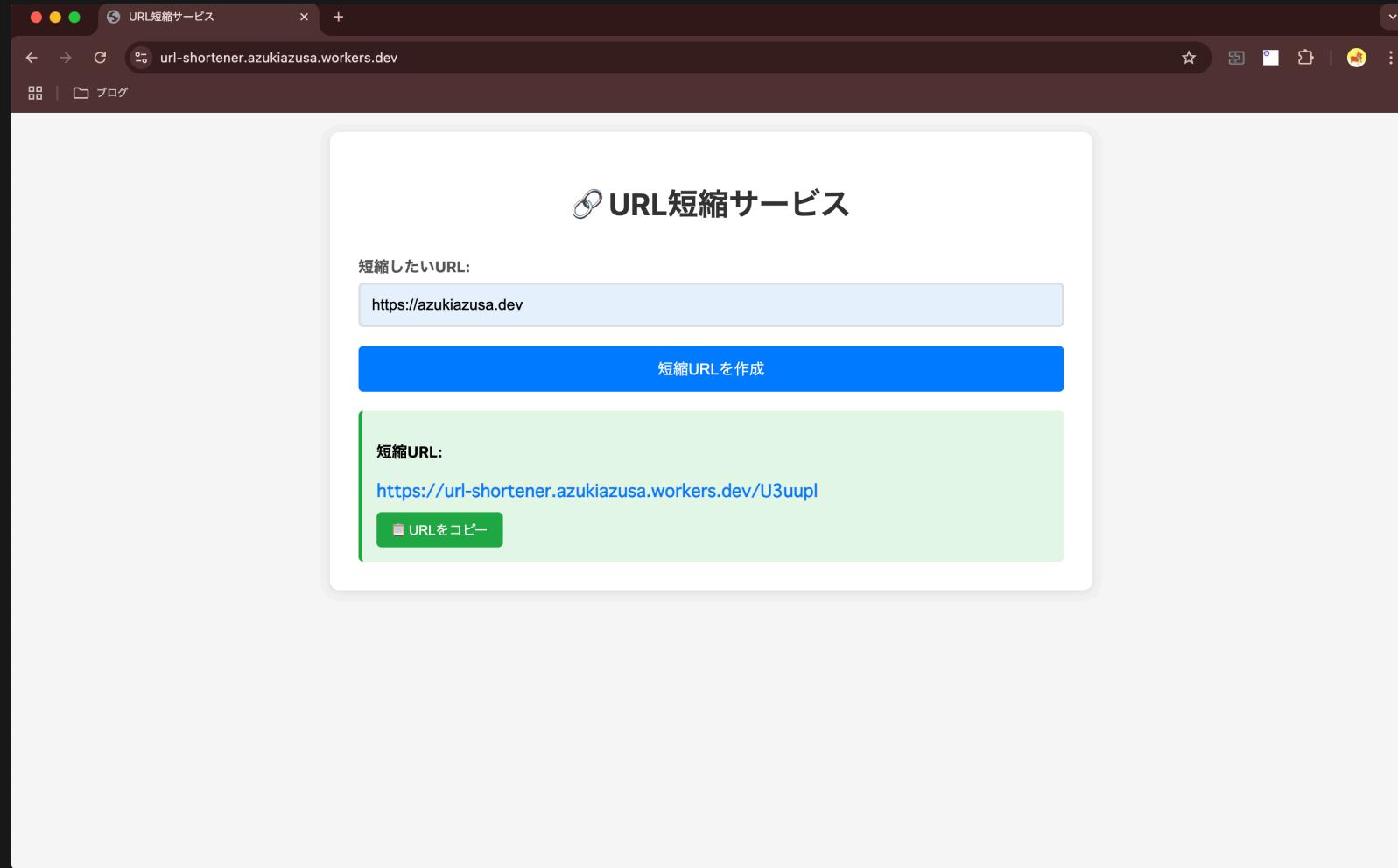
## ● Worker情報取得完了

- Worker名: url-shortener
- Worker ID: edd24fb00d0148cfa746af336720c737
- デプロイ済みURL: <https://url-shortener.azukiazusa.workers.dev>
- KVバインディング: URL\_SHORTENER (ID: c66c2c00057e4c3b960f9440d464f6fa)

Worker は正常にデプロイされ、Honoフレームワークを使用してURL短縮機能とWeb UIが実装されています。

>

# URL短縮サービスの完成



# まとめ

- バイブコーディングはAIエージェントが主体となる開発スタイル
- Cloudflare MCP は LLM から Cloudflare のリソースにアクセスしたり、作成・削除するためのツールを提供
- Cloudflare MCP の Workers Bindings Server により、AIエージェントが直接Workerをデプロイ可能

