







# Rust で実装したAPIをデプロイした

 安定の無料 

# 誰お前？

- 氏名: 大杉太郎
- Twitter: @tarosg
- 仕事: エンジニア, プログラミング講師
- 技術: Laravel, JS, Deno, (Rust)
- 好きなもの: , , ,  , 

# 今日のお題

- ジョジョのセリフを返す API を Rust で実装した.
- デプロイしたい.
- お金はかけたくない (重要)

# APIの実装

フレームワークがいくつかある。

- Axum
- Actix-web
- Rocket

好きなもので実装すれば OK. （割愛）

# デプロイ

せっかく実装したのでデプロイしたい！！

- AWS App Runner
- GCP Cloud Run
- Azure Functions

どれもできます．とはいえ．．．

めんどくさい

# めんどくさい

- 多分自分しか使わない.
  - スケールするわけでもない.
- 完全にオーバースペック. もっと手軽なのがほしい.





コマンド一発でデプロイできるサービス

# Shuttle is 何？

- Rust で実装した コード をデプロイできるサービス.
- 様々な Rust のフレームワークに対応.
- 無料枠がある.
- DB (Postgresなど) がある.
- Example が豊富.

# 作り方

# インストール

```
cargo install cargo-shuttle
```

# プロジェクトの作成

```
cargo shuttle init
```

# ローカルで動作

```
cargo shuttle run
```

# デプロイ

```
cargo shuttle deploy
```

# API 動作例 (Axum フレームワーク)

```
use axum::{routing::get, Router};

async fn hello_world() -> &'static str {
    "Hello, world!"
}

#[shuttle_runtime::main]
async fn main() -> shuttle_axum::ShuttleAxum {
    let router = Router::new().route("/", get(hello_world));

    Ok(router.into())
}
```

デモ

# 感じたこと

- インフラまでまとめて管理されているので，とても楽．
- AWS とかの設定や管理は結構たいへん．
- 初心者ほど活用できるサービスだと思う．

# 注意点

- 無料枠は 3 プロジェクトまで.
- Shuttle のアップデートでデプロイできなくなることがある.
  - プロジェクトのバージョンを修正すれば OK.
- プロジェクトが 30 分で寝る
  - `cargo shuttle project start --idle-minutes 0` の設定が必要.

# まとめ

**Rust のコードは Shuttle でデプロイしよう！**

- 無料でデプロイできる.
- インフラの設定が不要.
- 初心者も活用可能.





**Thanks!**

 alt text