

地味に効く、ドキュメント作成の AI 活用術



ムツミックス / Mutsumix

2025-06-19 | 生成 AI と働くー エンジニア現場でのリアルな活用方法

自己紹介

梶原 睦 / かじはら むつみ

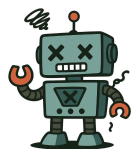
- 株式会社 シスマック
DX ソリューション事業部 部長
- Twitter (現 X) : [@Mutsumix_dev](#)
- Voicy: [Mutsumix の進捗どう？](#)
- 技術書典: [自宅水耕栽培で毎日サラダ生活](#)
- 最近は受託開発の提案やったり営業やったり研修講師やったり総務やったり



コードは AI に任せられるようになりました

開発ドキュメントは？

WordとExcelで設計書作ってる...



PowerPointでAWS構成図描いてる...

ドキュメント修正のたび、図の修正も...



従来ドキュメントの問題点

バージョン管理不可 (Fix_画面設計書_最終_250619.docx)

コードと別の場所にある (メンテコスト、陳腐化リスク増大)

コンテキストスイッチ (VSCode → PowerPoint → Word → VSCode)



図を描くのに 3 時間かった...

仕事した気になってしまいが、ビジネスは何も進んでいない

特に作図を含むドキュメントは脳への負担が大きい

どれも本質ではないことに集中してしまう

- レイアウトを考える
- 色を選ぶ
- アイコンを探す
- 線を綺麗に整える
- 文字サイズを調整
- 修正が入るたびに微調整

AI 時代、ドキュメントは Markdown で 書くことが一般化する

AI 時代、ドキュメントは Markdown で書くことが一般化する

と勝手に予想。文章を構造化して表現できるので、意図を適切に AI に汲み取らせやすい

現状：

README、Notion、Slack、Zenn、Qiita、GitHub Issues, Marp, ReVIEW

今後：

設計書、仕様書、議事録、マニュアル、提案書

そこで提案 💡

Markdown + Cursor + Claude + draw.io 拡張

- すべて一つのエディタで完結
- AI が図も文章も生成
- Git でバージョン管理
- コードと同じ場所に保管

実演

Terraform コードから構成図を生成

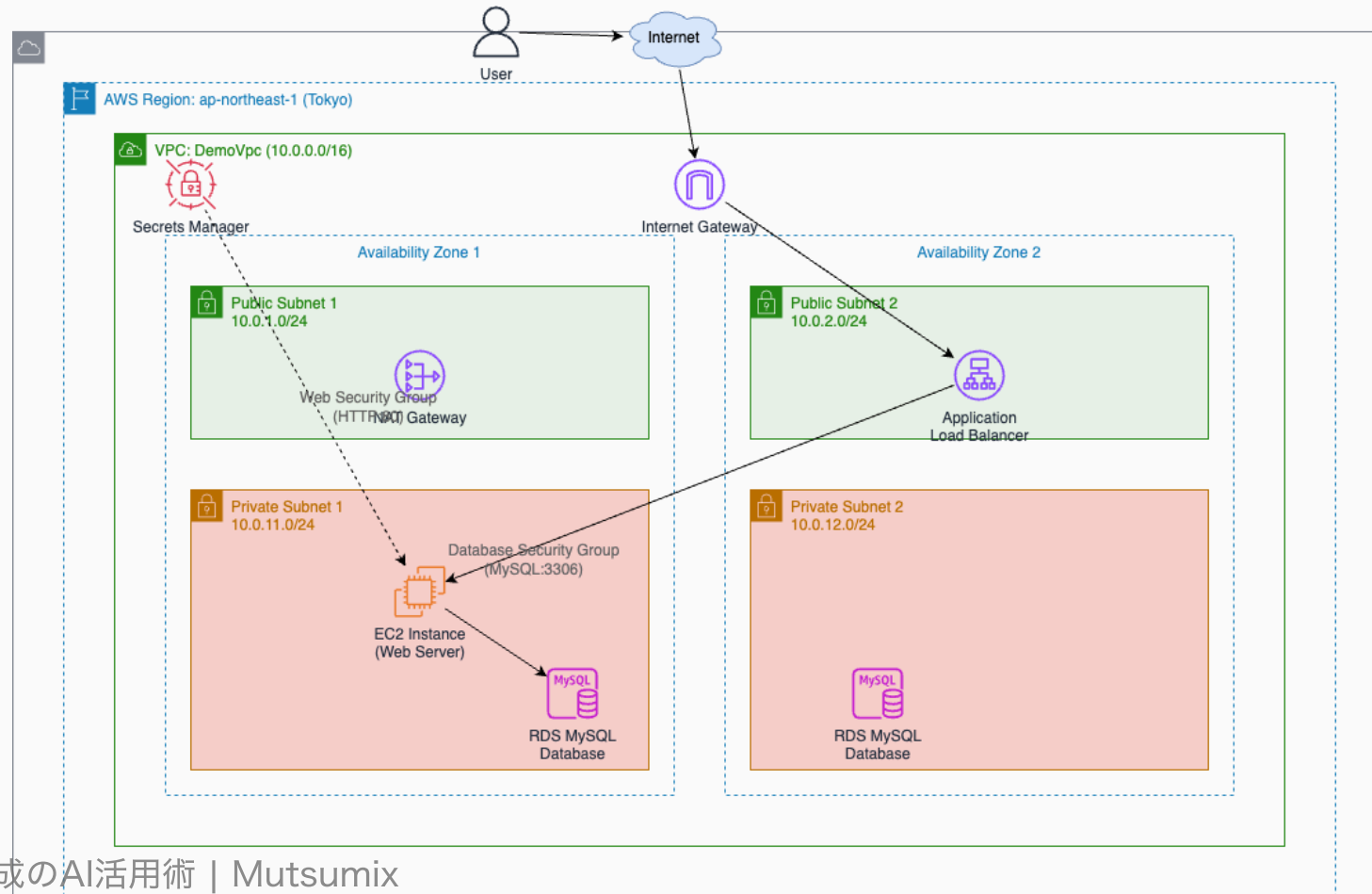
```
resource "aws_vpc" "demo_vpc" {  
  cidr_block          = "10.0.0.0/16"  
  enable_dns_hostnames = true  
  enable_dns_support  = true  
}  
resource "aws_instance" "web_server" {  
  ami              = data.aws_ami.amazon_linux.id  
  instance_type    = "t3.micro"  
  subnet_id        = aws_subnet.private_subnet_1.id  
  vpc_security_group_ids = [aws_security_group.web_sg.id]  
}  
# ... 以下、省略 (全381行)
```

指示文

「この Terraform コードを元に、AWS 構成図を.drawio ファイルで作成してください。最新 AWS のアイコンを利用してください。ファイル名は aws_structure.drawio.png としてください。」

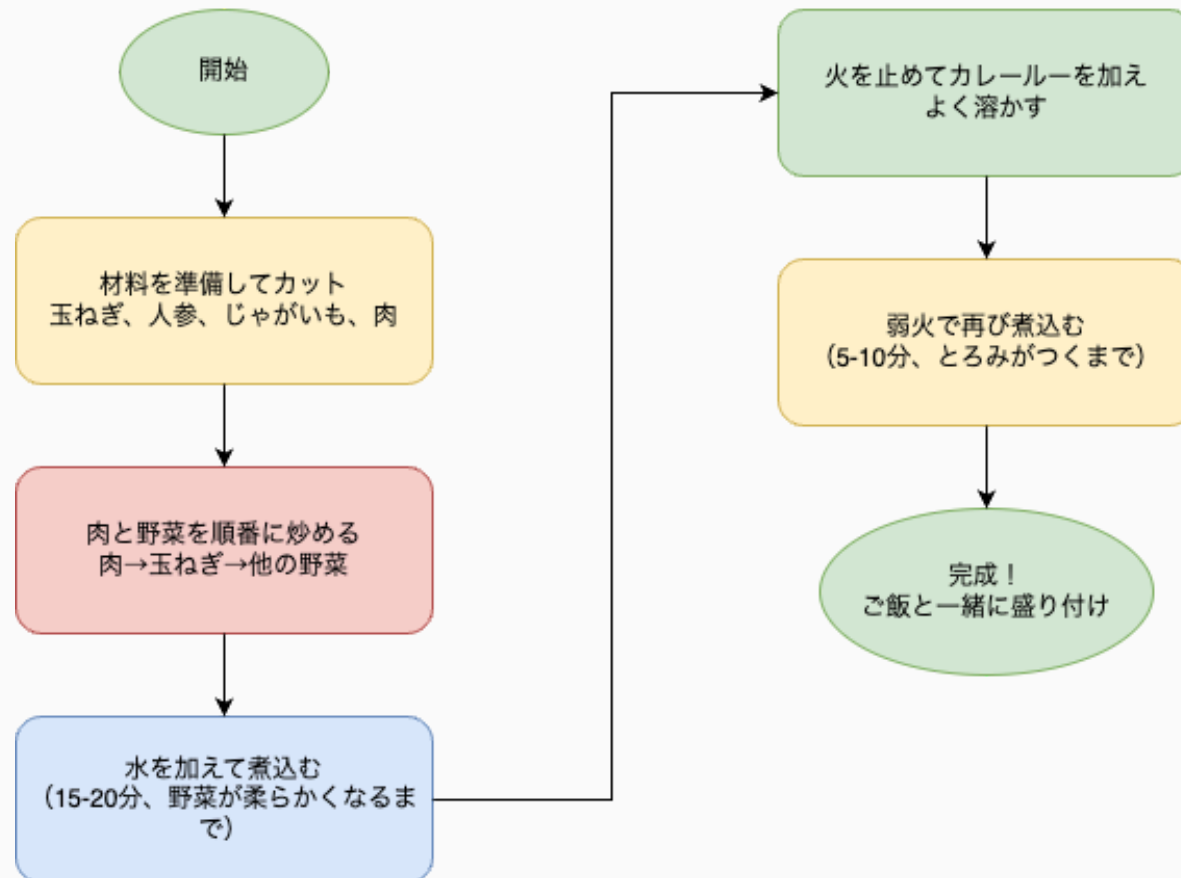
結果（例）

手直しすれば、使えそう



待っている間に

カレーの作り方



さらなるメリット

- 図の修正も一瞬（AI に再指示するだけ）
- 多言語対応も簡単（翻訳 AI と連携）
- 検索性抜群（テキストベース）
- チーム共有簡単（GitHub/GitLab）

ありがとうございました 🙏
