

基本情報

プロフィール

項目	値
ハンドルネーム	つくぼし
最終学歴	早稲田大学 商学部卒
主要職種	クラウドインフラエンジニア

アカウント

- [DevelopersIO](#)
- [Zenn](#)
- [Github](#)
- [SpeakerDeck](#)
- [X](#)

業務スキル

クラウドインフラ設計

クラウドインフラの設計を専門としており、特にAWS のアーキテクチャ設計・構築が一番得意です。モノリシック、マイクロサービス、静的ウェブサイトホスティング、CICD パイプライン、マルチアカウント、イベント駆動といった様々なパターンの設計・構築経験があります。AWS アーキテクチャを構築するにあたり、CloudFormation、Terraform、CDK といった IaC の開発経験も多数あります。

生成AIの専門知識

LLMが流行し始めた頃から生成 AI、RAG、及び AI エージェントの分野に特化し、様々な情報を発信してきました。生成 AI を活用したチャットボットアプリの自社開発や導入支援に携わった経験があります。さらに、生成 AI インフラに特化した社内チームのリーダーを務めていたり、生成 AI コンピテンシーの取得に貢献した経験もあります。

サーバレス開発

きちんとコードを読んで様々な問題に対応できるエンジニアになりたいという思いから、インフラのみならずサーバレスのアプリケーション開発経験も積極的に積んできました。アーキテクチャとしては、主に Lambda や ECS Fargate、Cloud Run 上でのサーバレスアプリケーションを設計・開発してきています。言語としては主に Python 及び TypeScript を使用したバックエンド開発経験があります。

プロジェクトマネジメント

プロジェクトマネージャーを務めることが多く、チームマネジメント及び顧客折衝の経験が豊富です。小規模チームで(3 - 5 人程度)でリーダーを務めた事が多数あります。プロジェクトを成功に導くため、メンバーのモチベーション向上や業務効率化に繋がるような仕組みづくりを大切にしています。

アウトプット活動

多くの方と技術の面白さを語り合いたい思いから、ブログ執筆や勉強会登壇、個人開発等のアウトプット活動に積極的に取り組んできました。DevelopersIO、Zennなどで100本以上の技術記事を執筆すると共に、技術雑誌への寄稿実績もあり、AWSや生成AIを中心に発信しています。またJAWS-UGを主とした勉強会への登壇も20回以上実施しており、様々な形態での発信を心がけています。

技術スキル

実業務で使用した技術のみ列挙します。

AWS

IAM VPC ELB(ALB・NLB) EC2 EFS RDS(MySQL・Aurora) ElastiCache(Redis) S3 CloudFront Route 53 ACM WAF ECS(Fargate) App Runner ECR API Gateway Lambda(Node.js・Python) DynamoDB Step Functions CodeCommit CodeBuild CodeDeploy CodePipeline SQS SNS Data Firehose EventBridge CloudWatch (Alarm・Logs) Security Hub GuardDuty Control Tower Organizations Identity Center Service Catalog DevOps Guru Bedrock Kendra

Google Cloud

Cloud Run Alloy DB

IaC

Terraform CloudFormation SAM CDK

OS

LinuxWindows

Programming Language / Framework

PythonTypeScriptNode.jsFastifyPrisma

SaaS

GitHubGitHub Actions

保有称号

表彰

表彰名	対象年度
Japan AWS Top Engineer	2024/2025
AWS Community Builder	2024/2025

資格

資格名	取得日
AWS DOP	2023/6/15
AWS SAP	2022/8/3
LPIC-2	2021/6/28
ORACLE MASTER Silver Oracle Database 12c	2021/3/29
基本情報 技術者試験	2019/11/20

経歴

株式会社ゼスト(業務委託：2025/3 -2025/6、正社員：2025/7 - 2025/8)

業務要約

役割：バックエンドエンジニア

プロジェクト一覧

- 介護事業者向けZEST SCHEDULE開発
- 関係者向けZEST SCHEDULE ADMIN開発

業務内容

- フロントエンドのNext.jsと技術スタックを統一し開発チーム全体の開発チーム全体の保守性向上のためTypeScript/Fastify/Prismaを用いて介護事業者向けスケジューリングシステムのRESTful APIを設計・開発し、どのエンジニアでも比較的運用しやすいバックエンド基盤を構築
- 社内のインフラ運用担当者が少なかったため、Cloud Runを活用したコンテナデプロイやAlloy DBでのSQLオペレーションも兼任し、フルスタックな開発運用体制に貢献

使用技術

- 言語・フレームワーク：TypeScriptNode.jsFastifyPrisma
- Google Cloudサービス：Cloud RunAlloy DB
- SaaS：GitHubGitHub Actions

クラスメソッド株式会社(正社員：2022/3 - 2025/6)

参画したプロジェクトが多いため、以下で主要なもののみ個別で記載します。

SaaSログ連携・保管コンテナシステム設計構築 (2024/9 - 2025/6)

概要：SaaS から出力されるログを定期的に AWS にアップロードし、監査で使用するためのログ形式に加工するコンテナインフラ/アプリを設計・構築・開発

役割：プロジェクトマネージャー(3名チーム)/バックエンドエンジニア

業務内容

- タイムアウト制限を回避しつつ運用負荷を最小限にするためECS Fargateをインフラ/アプリの両面で設計・構築し、SaaSログの定期取得と加工処理を自動化
- マルチテナント環境での設定差分管理のためCloudFormation/SAMでIaC化し、パラメータファイルによる環境別デプロイを実現
- ログ欠損による業務影響を防ぐためStep Functionsの自動リトライ機構とCloudWatch Alarmsによる監視を設計・構築し、障害時の自動復旧を確立

使用技術

- AWS サービス：VPCECRECS(Fargate)Lambda(Python)SQSSNSS3CloudWatch (Alarm・Logs)EventBridgeStep Functions
- IaC：CloudFormationSAM
- SaaS：GitHub

AI-Starter内製開発(2024/1 - 2025/6)

概要：生成AIを活用した自社SaaSチャットボットアプリ基盤を設計・開発

役割：生成AIエンジニア/インフラエンジニア

業務内容

- React.js + Fastifyアプリと技術スタックを統一し開発チーム全体の保守性向上のためCDK（TypeScript）でIaCを刷新し、コンストラクト化により顧客要件に応じた多様なアーキテクチャバターンの提供を実現
- 顧客環境へのデプロイ作業を効率化するためCDKからMarketplace用CloudFormationテンプレートを生成する仕組みを実装し、CDKメタデータ削除とパラメータ最適化により標準化されたデプロイを確立
- 運用負荷削減のためApp Runner/Bedrock/Kendraでサーバレス RAGアーキテクチャを設計・構築し、コンソールからのAPIキー変更に対応するためLambda-backedカスタムリソースでSecrets Manager連携を実装

使用技術

- ・ AWS サービス: WAF App Runner DynamoDB Secrets Manager Lambda(Node.js) Bedrock Kendra S3
- ・ IaC: CDK(Typescript)

参考URL: [生成AI環境構築サービス「AI-Starter」 | サービス | クラスメソッド株式会社](#)

AIチャットボット設計構築 (2024/9 - 2025/3)

概要: 生成 AI を活用したOSSである[GenU](#)を用いたチャットボットを構築

役割: プロジェクトマネージャー(3名チーム)/生成AIエンジニア

業務内容

1. お客様の業務内容に適した生成AIユースケース選定のためGenUを提案し、セキュリティ要件を満たすためBedrock Knowledge BaseのRAGシステムをCDKで実装
2. 社内文書に含まれる図表も検索対象にする必要からAdvanced Parsing機能を有効化し、テキストだけでなく視覚的情報も含めた高精度な回答生成を実現
3. 知識レベルの異なる関係者への説明のため生成AI基礎からKendraとの技術比較、検索精度評価のRAGASやBedrock Guardrailの活用方法まで幅広く技術QAに対応

使用技術

- ・ AWS サービス: WAF DynamoDB Secrets Manager Lambda(Node.js) Bedrock Kendra S3
- ・ IaC: CDK(Typescript)

社内生成AIインフラチーム活動(2024/1 - 2025/6)

概要: 生成 AI 関連の案件に対応する体制作りのため、生成 AI に特化したインフラを専門に部署を跨いだ交流を促進する活動を主導

役割: チームリーダー(5名チーム)/生成AIエンジニア

業務内容

1. 生成AI案件の急増に対応するため有志メンバーで生成AIインフラチームを立ち上げ、部門間の連携不足解消のため定期交流会を主導し効率的な業務フローを整備
2. 部門内の生成AI知識格差を解消するため継続的なブログ作成とレベル別勉強会を企画し、初学者から上級者まで段階的な学習環境を構築
3. AWSコンピテンシー取得要件を満たすため生成AI案件のドキュメント整備と監査対応準備を部門横断で実施し、コンピテンシー取得に貢献

参考URL: [40社の支援実績によりAWSの「生成AIコンピテンシー」を取得 | クラスメソッド株式会社](#)

RAG システム設計構築 (2023/11 - 2024/3)

概要: 生成 AI を活用したチャットボットアプリで使用する、社内ドキュメントを検索するための RAG システムを設計・構築

役割: 生成AIエンジニア/インフラエンジニア

業務内容

1. 運用負荷を最小化するため当時まだ実績の少なかったBedrock/Kendra/S3のフルマネージドサービスを組み合わせたサーバレスRAGアーキテクチャを設計・構築
2. インフラ担当者不足に対応するためApp Runnerのオートスケーリングを活用し、NWやOSの運用が不要なAIチャットボット基盤を確立

使用技術

- ・ AWS サービス: WAF App Runner Bedrock Kendra S3
- ・ IaC: CloudFormation

AWSマルチアカウント管理支援通知集約システム内製開発(2023/8 - 2024/2)

概要: 複数の AWS アカウントに対する通知設定を一括で付与できる、通知集約システムを構築

役割: インフラエンジニア

業務内容

1. 複数アカウントのセキュリティアラート見逃しを防ぐためSecurity Hub/GuardDutyの通知をEventBridge経由で集約し、Step Functionsで一元管理する構成を実装
2. アカウント追加の度に手動設定する運用負荷を削減するためCDK(TypeScript)でマルチアカウント対応のIaC実装を行い、パラメータ化による効率的なデプロイを実現

使用技術

- ・ AWS サービス: Security Hub GuardDuty Step Functions EventBridge
- ・ IaC: CDK(Typescript)

参考URL: [AWSマルチアカウント管理支援 | クラスメソッド株式会社](#)

SaaSログ連携・保管サーバレスシステム設計構築(2023/5 - 2024/6)

概要: SaaS から出力されるログを定期的に AWS にアップロードし、監査で使用するためのログ形式に加工するサーバレスインフラ/アプリ及びCICDシステムを設計・構築・開発

役割: プロジェクトマネージャー(3名チーム)/バックエンドエンジニア

業務内容

1. サーバレス構成で運用負荷を最小化するためLambdaをインフラ/アプリの両面で設計・構築し、SaaSログの定期取得と加工処理を自動化
2. 長期的な保守性確保のためPycodestyle準拠のコーディング規約を策定し、pytestによる品質担保とloggingモジュールによる障害時の原因特定を容易化
3. 複数メンバーでの並行開発を効率化するためGitHub Pull requests/Wiki/Actionsで開発プロセスを標準化し、自動テスト/デプロイでリリース品質を向上

使用技術

- ・ AWS サービス: EventBridge Lambda(Python) SQS SNS S3 CloudWatch (Alarm・Logs) DevOps Guru
- ・ IaC: CloudFormation SAM
- ・ SaaS: GitHub GitHub Actions

認証システム用コンテナ基盤技術支援(2022/11 - 2023/6)

概要: 顧客情報利用のための認証システムにおけるアプリケーションのコンテナ化に伴うインフラ基盤を設計・構築

役割: プロジェクトマネージャー(5名チーム)/インフラエンジニア

業務内容

1. アクセス数に応じたスケーリングの要求に対しALB/ECS Fargateを設計・構築しCloudFormationでコード化、PHPログの用途別振り分けのためFireLens + Fluent Bitによるログルーティングを実装
2. CI/CD未経験の顧客のためCodePipelineのローリングアップデートとブルー・グリーンの比較検証を実施し、シンプルな構成を重視してローリングアップデートを設計・構築
3. 社内ネットワーク限定アクセスとTLS暗号化要件を満たすため、API Gateway/NLB/ECSのプライベート統合にCloudFront/WAFを組み合わせたセキュアな認証基盤を実現

使用技術

- ・ AWS サービス: VPC ELB(ALB・NLB) ECS EFS RDS(Aurora) ElastiCache(Redis) S3 CloudFront WAF ECS(Fargate) ECR API Gateway CodeCommit CodeBuild CodeDeploy CodePipeline EventBridge
- ・ IaC: CloudFormation

基幹システム用マルチアカウント基盤技術支援 (2022/7 - 2024/3)

概要: 部門・プロジェクト単位での権限/環境の分離やガバナンス統一に必要な、社内システム用のマルチアカウント管理基盤の設計・構築に伴う技術支援を実施。

役割: プロジェクトマネージャー(3名チーム)/インフラエンジニア

業務内容

- 1. 部門・プロジェクト単位での権限分離要求に対しControl Tower/Organizationsの設計支援を実施し、セキュリティとガバナンスを統一した管理基盤を実現
- 2. 各部門が安全にリソース作成できる仕組みのためService Catalogでの標準テンプレート作成を支援し、承認済みリソースのみ展開可能な環境を確立
- 3. 複数アカウントのログ分析効率化のためWAF/Data Firehose/CloudWatch Logsによる集約基盤の設計支援を行い、インシデント調査時間の短縮を実現

使用技術

- ・ AWS サービス： Control Tower Organizations Identity Center Service Catalog WAF Data Firehose CloudWatch (Logs)
- ・ IaC： CloudFormation
- ・ SaaS： GitHub

ECサイト用AWS環境構築(2022/6 - 2022/7)

概要：新規 EC サイトにおけるランディングページに必要な静的ウェブサイトホスティング基盤を設計・構築

役割：インフラエンジニア

業務内容

- 1. 3週間の短納期要求に対しインフラ構築を高速化するためTerraformでIaC化し、CloudFront/S3による高可用性の静的ホスティング基盤を設計・構築
- 2. 公開前のランディングページ保護のためCloudFront Functions(Node.js)によるBasic認証を開発し、開発段階での不正アクセスを防止
- 3. 企業ブランディング確立のためRoute 53/ACMでカスタムドメインとHTTPS配信を設定し、信頼性の高いWebサイト環境を実現

使用技術

- ・ AWS サービス： IAM CloudFront S3 Lambda(Node.js) Route 53 ACM
- ・ IaC： Terraform

CMSインフラ用AWS環境構築(2022/5 - 2022/8)

概要：動画サイト構築 CMS 新規導入のためのウェブサイト基盤を設計・構築

役割：インフラエンジニア

業務内容

- 1. 2週間の短納期と環境別構成要求に対応するためTerraformでIaC化し、ALB/EC2/RDS/ElastiCacheのCMS基盤を迅速に設計・構築
- 2. 検証・本番環境の差異による不具合を防ぐためTerraform Moduleで共通部分を抽出し、パラメータのみの変更で環境差分を吸収
- 3. 夜間のトラフィック減少に着目しLambda/EventBridgeでRDSインスタンスタイプの定期切り替えを自動化し、ランニングコストを削減

使用技術

- ・ AWS サービス： IAM VPC ELB(ALB) EC2 RDS(MySQL) ElastiCache(Redis) S3 Route 53 ACM Lambda
- ・ IaC： Terraform

兼松エレクトロニクス株式会社(正社員：2019/4 - 2022/2)

業務要約

役割：インフラエンジニア

プロジェクト一覧

- ・ 基幹システム用vSphere基盤更改(2021/10 - 2022/2): Oracle19c DBサーバの設計・構築
- ・ 情報システム用vSphere基盤更改(2021/10 - 2021/12): RHEL8・Apache2.4環境の構築
- ・ 基幹システム用z/VM基盤更改(2021/6 - 2021/12): SLES12 SP5へのバージョンアップ
- ・ 基幹システム用VIOS基盤更改(2021/1 - 2021/7): WAS9.0.5 APサーバの設計・構築
- ・ 社内コンテナ検証チーム活動(2020/6 - 2022/1): Kubernetes・OpenShift検証環境構築

業務内容

- 1. 基幹システムの老朽化対応のためVMware vSphere/zVM/VIOS上での更改を担当し、Oracle19c/RHEL8/SLES12/WAS9.0.5の設計・構築により安定稼働を継続
- 2. 社内のコンテナ技術導入推進のためDocker/Kubernetes/OpenShiftの検証チームに参加し、従来の仮想化環境からの移行検討と技術標準の策定に貢献

使用技術: Linux(RHEL/SLES), VMware vSphere, Oracle, WebSphere, Apache, Docker, Kubernetes, OpenShift

業務外活動

個人開発

以下では開発した自作リポジトリの内、主要なものを記載します。

リポジトリ名	言語・フレームワーク	概要
dotfiles	Bash	自身の PC セットアップ用の dotfiles
honox-resume-template	TypeScript HonoX Vite	Markdown職務経歴書をWebサイトとPDFに変換し自動デプロイするテンプレート
sam-notify-aws-billing	SAM Python	AWS 利用料金を通知するサーバレスシステムコード
sam-ai-cloud-engineer	SAM Python	Amazon Bedrock を活用し構成図を CloudFormation テンプレートとパラメータシートに変換するサーバレスシステムコード
nextjs-mastra-mcp-with-aws	CDK TypeScript	AWS とMastraで実装したMCP Serverを呼び出すAIエージェント
gas-count-schedule-time	Apps Script TypeScript	Googleカレンダーの予定から色ごとに工数をカウントし集計するアプリコード
sam-bedrock-kb-aurora-rag	SAM Python	Amazon Aurora と Amazon Bedrock Knowledge Bases を用いた RAG システムデプロイするためのコード
terraform-microservices-template	Terraform	AWS を用いたマイクロサービス構成用 Terraform テンプレート

ブログ投稿

DevelopersIO

2022/3から2025/6までに、100本以上の記事を作成していました。代表的な記事は以下の通りです。

- ・ [AWS入門ブログリレー2024～Amazon Bedrock編～](#)
- ・ [CloudFormation一撃で作るAWS料金通知ツール\(Email/Slack/LINE対応\)](#)
- ・ [Knowledge Bases for Amazon Bedrock \(with OpenSearch Serverless\)をSAMで実装してみた](#)
- ・ [S3にアップロードしたAWS構成図をCloudFormationに変換するシステムを作ってみた](#)

- [Kendraのアクセスコントロール設定をEntra IDのOpen ID Connectを用いて有効化してみた](#)
- [AWSの構成図をChatGPT\(GPT-4V\)に読み込ませてIaCコードを生成してみた](#)
- [ECSとCodePipelineのブルー/グリーンデプロイ構成をCDKで実装してみた](#)
- [AuroraとRDSの違いを一覧表でまとめてみた](#)
- [FireLens\(Fluent Bit\)におけるログルーティングの仕組みについて調査してみた](#)
- [TerraformでCloudFront Functionsを環境ごとに有効化/無効化してみた](#)

Zenn

2023/5から現在まで投稿している主要のブログサイトになります。代表的な記事は以下の通りです。

- [Macの環境をdotfilesでセットアップしてみた改](#)
- [Google Calenderの予定をTypeScriptを用いて色別で工数集計してみた](#)
- [GmailをTypeScriptとLINE Messaging APIを用いてLINEに自動転送してみた](#)
- [MCP Serverを呼び出すAIエージェントをMastra、Next.js、AWS CDKで実装してみた](#)
- [Markdownで書いた職務経歴書を良い感じのWebサイトとPDFに自動変換するツールを作ってみた](#)
- [Claude Codeとghコマンドで作るカスタムスラッシュコマンドn選](#)

イベント登壇

以下では登壇したイベントの内、主要な資料及び動画を記載します。

Mita.ts #5 : **Mastra**に入門してみた ~**AWS CDK**を添えて~

- [Mastraに入門してみた ~AWS CDKを添えて~ - Speaker Deck](#)

Classmethod AI Talks(CATs) #14 : **AWS**エンジニアに捧ぐ**LangChain**の歩き方

- [AWSエンジニアに捧ぐLangChainの歩き方 - Speaker Deck](#)
- [AWSエンジニアに捧ぐLangChainの歩き方 - YouTube](#)

Toranomon Tech Hub #2 : 世界の中心で**App Runner**を叫ぶ ~**Aurora DSQL**を添えて~

- [世界の中心でApp Runnerを叫ぶ ~Aurora DSQLを添えて~ - Speaker Deck](#)

Classmethod AI Talks(CATs) #5 : 初めての**GPTs** ~ネコ派を〇〇派に変える技術~

- [初めてのGPTs ~ネコ派を〇〇派に変える技術~ - Speaker Deck](#)
- [クラムソエンジニア陣が執筆！書籍『ビジネスのためのChatGPT活用ガイド』でChatGPTの業務活用を始めてみよう - YouTube](#)

JAWS FESTA 2024 in 広島 : **Amplify Gen 2**ではじめる 生成**AI**アプリ開発入門

- [Amplify Gen 2ではじめる 生成AIアプリ開発入門 - Speaker Deck](#)

Classmethod Odyssey : **AWS**で構築するパターン別**RAG**構成解説

- [AWSで構築するパターン別RAG構成解説 - Speaker Deck](#)
- [AWSで実現するRAG構成！Retrieverの種類と最適なインフラパターン解説 - YouTube](#)

Bedrock Claude Night 2 : **AWS**構成図から**CloudFormation**とパラメータシートを自動生成するシステムを作ってみた

- [AWS構成図からCloudFormationとパラメータシートを自動生成するシステムを作ってみた - Speaker Deck](#)

JAWS-UG東京 ランチタイムLT会 #11 : 5分で分かる(かもしれない)**Vector engine for OpenSearch Serverless**

- [5分で分かる\(かもしれない\)Vector engine for OpenSearch Serverless - Speaker Deck](#)

JAWS-UG朝会 #55 : 君は**Application Composer**というサービスを知っているか

- [君はApplication Composerというサービスを知っているか - Speaker Deck](#)

JAWS-UG CDK支部 #12 : **CDK**を**CloudFormation**テンプレートとして利用する際の注意点n選

- [CDKをCloudFormationテンプレートとして利用する際の注意点n選 - Speaker Deck](#)

JAWS-UG朝会 #51 : インフラエンジニアのための**Lambda**実践入門

- [インフラエンジニアのためのLambda実践入門 - Speaker Deck](#)

DevelopersIO 2023 : **AWS**と**GitHub**を用いたパターン別**CI/CD**構成解説

- [AWSとGitHubを用いたパターン別CI/CD構成解説 - Speaker Deck](#)
- [AWS基盤でのCI/CD構成を徹底解説！CodeシリーズとGitHubの使い分けも解消 - YouTube](#)

JAWS-UG朝会 #39 : (今更ながら)**AWS**のコンテナサービスについてざっくりまとめる

- [\(今更ながら\)AWSのコンテナサービスについてざっくりまとめる - Speaker Deck](#)

JAWS-UG CLI専門支部 #273R : **Lambda**と**Line Messaging API**で湯婆婆botを作ってみた

- [LambdaとLine Messaging APIで湯婆婆botを作ってみた - Speaker Deck](#)

DevelopersIO 2022 : **AWS**初心者に捧ぐコスト可視化のススメ

- [AWS初心者に捧ぐコスト可視化のススメ - Speaker Deck](#)
- [AWS初心者に捧ぐコスト可視化のススメ - YouTube](#)

AKIBA.AWS ONLINE #08 : それ、**t2.micro**選んで大丈夫？

- [それ、t2.micro選んで大丈夫？ - Speaker Deck](#)

JAWS-UG CLI専門支部 #256M : **CLI**専門支部に参加したら**AWS**エンジニアに転職できた件について

- [CLI専門支部に参加したらAWSエンジニアに転職できた件について - Speaker Deck](#)

特筆事項

Software Design 2024年11月号寄稿

概要 : 「第1特集 新世代の開発スタイル はじめてのAI駆動開発」内の「第4章：Infrastructure as Codeで生成AIを活用するアーキテクチャ図⇔IaCコードの変換も自由自在」を寄稿

参考URL : [Software Design 2024年11月号 | 技術評論社](#)

JAWS-UG 茨城運営

概要：JAWS-UG茨城の初期メンバーとして2024/11からの立ち上げに貢献し、運営メンバーとして活動中

参考URL：[JAWS-UG 茨城 - connpass](#)