

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
経験職種	ソリューションアーキテクト/バックエンドエンジニア/プロジェクトマネージャー

アカウント

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

業務スキル

クラウドアーキテクチャ設計

現在は AWS を業務で一番用いています。
モノリシック、静的ウェブサイトホスティング、マイクロサービス、CI/CD パイプライン、マルチアカウント、サーバーレスといった様々なアーキテクチャの設計・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。

サーバーレスバックエンド開発

クラウドインフラのみならず、サーバーレスアーキテクチャ上のバックエンド開発経験も持ち合わせているためフルスタックな活躍が可能です。
アーキテクチャとしては、主に Lambda や ECS Fargate 上での開発経験があります。
特に Python 及び TypeScript を用いたバックエンド開発が得意です。

生成AIの専門知識

最近では生成 AI、RAG、及び AI エージェントの技術支援に特化し、様々な情報を発信しています。
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経験もあります。

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ことが多数あります。
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管理や業務効率化にも力を入れています。

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

IaC

Terraform CloudFormation SAM CDK

OS

Linux Windows

Programming Language

Python JavaScript TypeScript

SaaS

GitHub GitHub Actions

保有称号

表彰

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

資格

資格名	取得日
AWS DOP	2023/6/15
AWS SOA	2023/2/15
AWS SAP	2022/8/3
AWS DVA	2022/3/24
AWS SAA	2021/11/11
AWS CLF	2021/7/21

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

クラスメソッド株式会社（2022/3 - 2025/6）

主にソリューションアーキテクトとして、AWS アーキテクチャの/設計/構築を担当しました。
またPJに応じて、バックエンドエンジニア兼任でサーバレスアプリケーションの開発や、プロジェクトマネージャー兼任でチームマネジメント及び顧客折衝も行っています。

以下では参画したプロジェクトの内、主要なもののみを記載します。

AIチャットボット設計構築（2024/9 - 2025/3）

概要

生成 AI を活用したチャットボットアプリを構築。

担当

- プリセールス
- 顧客折衝
- システム設計
- システム構築
- システム運用保守

業務内容

- 設計書作成
- 構築作業
- 動作確認試験

使用技術

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

SaaSログ連携・保管コンテナシステム設計構築（2024/9 - 2025/3）

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、特定用途で使用するためのログ形式に加工するコンテナアプリケーションを構築。

担当

- プリセールス
- 顧客折衝
- アプリケーション設計
- アプリケーション構築

5. アプリケーション運用保守

業務内容

1. 設計書作成
2. アプリケーションコード開発
3. 構築作業
4. 動作確認試験

使用技術

- AWS サービス： VPC ECR ECS(Fargate) Lambda(Python) SQS SNS S3 CloudWatch (Alarm・Logs) EventBridge Step Functions
- IaC： CloudFormation SAM
- SaaS： GitHub

[AI-Starter](#)内製開発 (2024/1 - 2025/6)

概要

生成 AI を活用したチャットボットアプリの開発

担当

1. チャットボットアプリ基盤設計
2. チャットボットアプリ基盤開発

業務内容

1. 設計書作成
2. IaC コード開発
3. デプロイ手順整備

使用技術

- AWS サービス： WAF App Runner 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 - 2024/8)

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実施

担当

1. リーダー

業務内容

1. 勉強会開催
2. 案件対応推進
3. 部署連携促進
4. 検証環境整備

使用技術

- AWS サービス： Amazon Bedrock Amazon Kendra Amazon S3 AWS App Runner Amazon DynamoDB AWS WAF

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 システムを構築

担当

1. RAG システム設計
2. RAG システム構築
3. RAG システム試験

業務内容

1. 設計書作成
2. IaC コード開発
3. 構築作業
4. 動作確認試験

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。

担当

1. 通知集約システム基盤設計
2. 通知集約システム基盤開発

業務内容

1. IaC コード開発
2. デプロイ手順整備

使用技術

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

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、特定用途で使用するためのログ形式に加工するサーバレスアプリケーションを構築。

担当

1. プリセールス
2. 顧客折衝
3. アプリケーション設計
4. アプリケーション構築

5. アプリケーション運用保守
6. CICD システム設計
7. CICD システム構築
8. CICD システム運用保守

業務内容

1. 設計書作成
2. アプリケーションコード開発
3. 構築作業
4. 動作確認試験

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いて稼働が必要な、アプリケーションのコンテナ化に伴う技術支援を実施。

担当

1. チームマネジメント (5 人)
2. 顧客折衝
3. コンテナ基盤設計
4. コンテナ基盤構築
5. コンテナ基盤試験
6. CICD システム設計
7. CICD システム構築
8. CICD システム試験

業務内容

1. 要件定義
2. 設計書作成
3. IaC コード開発
4. 構築作業
5. 動作確認試験
6. 技術質問対応

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 - 2023/3)

概要

部門・プロジェクト単位での権限/環境の分離や、一元的に統一されたガバナンスの確保が必要な、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。

担当

1. AWS 基盤設計/構築に伴う技術支援

業務内容

1. 設計に必要な情報提供
2. 構築に必要な動作検証
3. IaC コード開発
4. 技術質問対応

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

Webサイト用RDSインスタンスタイプ自動変更システム構築（2022/7 - 2022/8）

概要

Web サイトリニューアルに伴い、RDS インスタンスタイプを定期的に自動変更するためのサーバレスアプリケーションを構築。

担当

1. プリセールス
2. 顧客折衝
3. サーバレスアプリケーション設計
4. サーバレスアプリケーション構築
5. サーバレスアプリケーション試験

業務内容

1. 設計書作成
2. アプリケーションコード開発
3. 構築作業
4. 動作確認試験

使用技術

- AWS サービス： EventBridge Lambda(Python) SNS

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構築。

担当

1. チームマネジメント（3 人）
2. 顧客折衝
3. インフラ基盤設計
4. 静的ウェブサイトホスティング基盤構築

5. 静的ウェブサイトホスティング基盤試験

業務内容

1. 要件定義
2. 設計書作成
3. IaC コード開発
4. 構築作業
5. 動作確認試験
6. 技術質問対応

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構築。

担当

1. 顧客折衝
2. ウェブサイト基盤設計
3. ウェブサイト基盤構築

業務内容

1. 設計書作成
2. IaC コード開発
3. 構築作業
4. 動作確認試験

使用技術

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

兼松エレクトロニクス株式会社（2019/4 - 2022/2）

インフラエンジニアとして、主にオンプレミスにおける Linux サーバの設計/構築を担当しました。
またコンテナ技術（Docker/Kubernetes）の検証も実施しています。

以下では参画したプロジェクトの内、主要なもののみを記載します。

基幹システム用vSphere基盤更改（2021/10 - 2022/2）

概要

基幹システムで稼働している OS / ミドルウェア の EOL が切れるため、インフラ基盤の更改を実施

担当

1. DB サーバ設計
2. DB サーバ構築

業務内容

1. 設計書作成
2. 構築作業
3. 動作確認試験

使用技術

- ミドルウェア： Oracle19c

情報システム用vSphere基盤更改（2021/10 - 2021/12）

概要

会社で利用している情報システムの老朽化にともない、新システムのインフラ構築及び既存環境からのデータ移行を実施

担当

1. OS / Web サーバ設計
2. OS / Web サーバ構築

業務内容

1. 設計書作成
2. 構築作業
3. 動作確認試験

使用技術

- OS： RHEL8
- ミドルウェア： Apache 2.4

基幹システム用z/VM基盤更改（2021/6 - 2021/12）

概要

基幹システムで稼働している OS が EOL を迎えるため、システム全体のバージョンアップを実施

担当

1. OS 設計
2. OS 構築

業務内容

1. 設計書作成
2. 構築作業
3. 動作確認試験

使用技術

- OS： SLES12 SP5

基幹システム用VIOS基盤更改（2021/1 - 2021/7）

概要

基幹システムで稼働している OS / ミドルウェア が EOL を迎えるため、インフラ基盤（H/W, S/W)の 更改を実施

担当

1. AP サーバ設計
2. AP サーバ構築

業務内容

1. 設計書作成
2. 構築作業
3. 動作確認試験

使用技術

- ミドルウェア： WAS9.0.5

社内コンテナ検証チーム活動 (2020/6 - 2022/1)

概要

DX 推進に関連した新ビジネスを確立するべく、コンテナ技術の検証を実施。自身はコンテナ検証/環境構築を担当

担当

1. コンテナ基盤 構築

業務内容

1. 検証
2. 構築作業

使用技術

- ミドルウェア： 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	概要
dotfiles	Bash	自身の PC セットアップ用の dotfiles
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
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-kendra-rag	SAM Python	Amazon Kendra と Amazon Bedrock を用いた RAG システムデプロイするためのコード
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
cdk-microservices-bluegreendeployment-template	CDK TypeScript	AWS を用いた マイクロサービス構成用 CDK テンプレート

ブログ投稿

以下では投稿したブログの内、主要なものを記載します。

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](#)

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。

[Amazon Bedrock GenUハンズオン #1](#)：生成AI解説資料

資料

- [Amazon Bedrock GenUハンズオン座学資料 #1 GenU環境で生成AIを体験してみよう](#)

[Amazon Bedrock GenUハンズオン #2](#)：RAG解説資料

資料

- [Amazon Bedrock GenUハンズオン座学資料 #2 GenU環境でRAGを体験してみよう](#)

[Classmethod AI Talks\(CATs\) ～生成AI 新年LT大会～](#)：AWSエンジニアに捧ぐLangChainの歩き方

資料

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

動画

- [AWSエンジニアに捧ぐLangChainの歩き方 - 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

資料

- [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

資料

- [AWSで構築するパターン別RAG構成解説 - Speaker Deck](#)

動画

- [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

資料

- [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](#)

[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

資料

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

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

資料

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

[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

資料

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

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

資料

- [インフラエンジニアのためのLambda実践入門](#)

[DevelopersIO 2023](#) : AWSとGitHubを用いたパターン別CI/CD構成解説

資料

- [AWSとGitHubを用いたパターン別CI/CD構成解説](#)

動画

- [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てみる

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

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

資料

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

[DevelopersIO 2022](#) : AWS初心者に捧ぐコスト可視化のススメ

資料

- [AWS初心者に捧ぐコスト可視化のススメ](#)

動画

- [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？

資料

- [それ、t2.micro選んで大丈夫？](#)

[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

資料

- [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](#)

その他

以下では特記すべき経歴について記載します。

[Software Design 2024年11月号](#) 寄稿

「第1特集 新世代の開発スタイル はじめてのAI駆動開発」における「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」を担当

[JAWS-UG 茨城](#) 運営

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