

全員 SRE ?

**CI/CD や運用まで完遂するプロダクトエンジニア
が実践する AWS CDK でのインフラ構築**

AlphaDrive Web Application Engineer & SRE / 畠山 嵩広

00 自己紹介

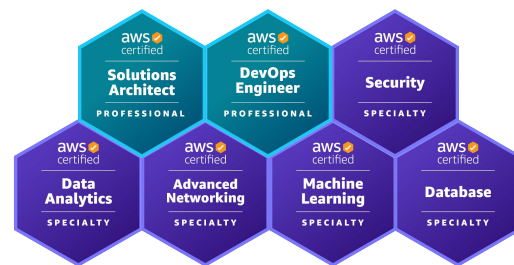


畠山 嵩広

AlphaDrive Webアプリケーションエンジニア & SRE



keyamin_



あと1...

アジェンダ

1. AlphaDriveについて
2. AWS CDKの紹介
3. 「全員SRE」のためにやっていること
4. まとめ

アジェンダ

1. AlphaDriveについて
2. AWS CDKの紹介
3. 「全員SRE」のためにやっていること
4. まとめ

01 AlphaDriveについて

AlphaDriveは、
新規事業創出支援のプロフェッショナルカンパニーかつ、
エンジニア第一のテクノロジーカンパニーです。



and more...

INCUBATION SUITE



▶ 対象課題

新規事業開発 / 事業開発人材育成対応

▶ 対象部署

事業開発・新規事業 / R&D・イノベーション
インキュベーション / 次世代事業開発
経営企画・総合計画 / 人事・人材開発

対象領域

事業創出

組織変革

人材育成

新規事業開発に特化したクラウド型総合支援ツール

3,500社以上の新規事業開発を支援してきたAlphaDriveの知見を余すことなく注ぎ込んだソリューション。新規事業制度を運営する事務局メンバーをサポートする管理機能から、起案者向け学習コンテンツ提供、起案ブラッシュアップ機能まで。新規事業開発に必要なサポートを1つのプラットフォームで実現します。

▶ ポイント



事務局業務の効率化を実現

新規事業プログラム参加メンバーの登録、研修/イベントの告知、事業アイデアの進捗管理、審査フィードバックなど多岐にわたる事務局業務に対応。
多忙な業務をサポートする機能を1つのツールに集約しました。



起案者の課題を先回りしたサポート機能

新規事業開発にあたって必要となる事業フレーム、事業案の管理、アイデアを仮説検証した履歴を自動アーカイブ、事業開発スキルの向上など多様な機能を取り揃えており、新規事業の推進・検討に専念することができます。



事業を前進させる支援オプション

リクルートで社内事業開発プログラムを立ち上げた麻生要一ら専門家によるメンタリングや顧客ヒアリング支援など人的なサポートも組み合わせることで、新規事業の進捗をバックアップします。

▶ 導入企業

TOYOTA

MUFG

docomo

NTT DATA

KIRIN

JR



▶ 対象課題

次世代型変革人材発掘 / 自律型人材育成 /
サイロ化打破 / エンゲージメント向上 /
ビジョン・理念浸透

▶ 対象部署

人事・人材育成 / 人財開発・組織開発
経営企画・総合計画 / 事業企画・経営管理 / 社内
広報 / 組織風土改革 / 総務・福利厚生 / 労働組合

対象領域

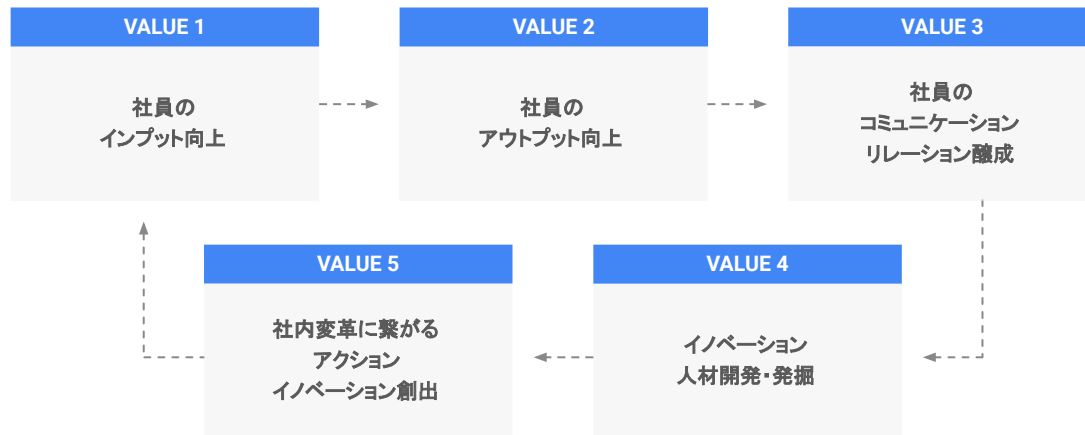
事業創出

組織変革

人材育成

学びと対話の空間

NewsPicksを法人向けの人材及び組織開発ツールとしてカスタマイズすることで、イノベーション人材を育て、組織の変革力を生み出す「人材と組織の成長プラットフォーム」です。



▶ 導入企業

docomo

PERSOL

三菱重工

マイナビ

dentsu

東急

JT

技術スタック紹介

AlphaDrive

Incubation Suite

利用頻度

低 ←



TypeScript



React
(WEB)

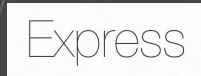
NEXT.JS



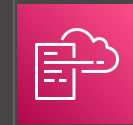
Figma



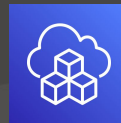
TypeScript



Express.js



AWS
CloudFormation



AWS CDK



フロント

デザイン

バックエンド

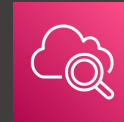
インフラ

監視

データベース



Aurora - Postgres



Amazon CloudWatch



NewRelic

AlphaDrive

開発中の新SaaS

利用頻度

低

フロント



TypeScript



Chakra UI



React
(WEB)



Storybook

NEXT.JS

デザイン



Figma

バックエンド



Go



Gin

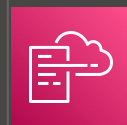


OpenAPI

インフラ



Amazon ECS /
Fargate



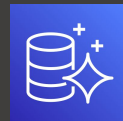
AWS
CloudFormation



AWS CDK

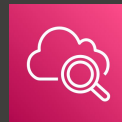


データベース



Aurora - Postgres

監視



Amazon CloudWatch



NewRelic

01 AlphaDriveのポリシー

私達AlphaDriveは組織の初期段階ではSREチームを組成せず、**プロダクト開発者全員がSRE**のスキルを持つことをポリシーに25名規模の組織で複数のSaaS開発を進めています。

- 赤澤 剛 (AlphaDrive CTO)



全員がSRE...? 🤔

01 そもそもSREとは？



**SREとは、ソフトウェアエンジニアに運用チームの設計を依頼したときに
できあがるものです。**

- 『SRE サイトリライアビリティエンジニアリング』より

「プロダクト開発者全員がSRE」とは...

**SREを職種ではなく方法論や文化として捉え、
運用まで開発者全員に浸透させること！** 🙋

01 AlphaDriveが目指す「プロダクト開発者全員がSRE」

自分が書いたコードは、
デプロイや運用まで自分で面倒を見る

👤「DevOpsエンジニアさん、パイプライン直して！」

👤「運用担当者さん、アラート鳴ってるよ〜！」

ではなく、**開発者自身**でソフトウェアエンジニアリングの知見を活かして開発者体験を向上させる！

それを実現できる仕組みを作る

そうは言っても、開発者全員にAWS(IAMポリシーの細かい書き方...)やTerraform(HCLのややこしいDynamicBlock...)を完璧に習得してもらうのは難しい

ただ開発者の負担が増えるだけでは、組織拡大の敷居も高くなる
→ **持続可能性**が低い👤

アジェンダ

1. AlphaDriveについて
- 2. AWS CDKの紹介**
3. 「全員SRE」のためにやっていること
4. まとめ

02 「プロダクト開発者全員がSRE」実現のためのIaCツール

開発中のSaaSでは、IaCを実現するツールとしてAWS CDKを採用しています



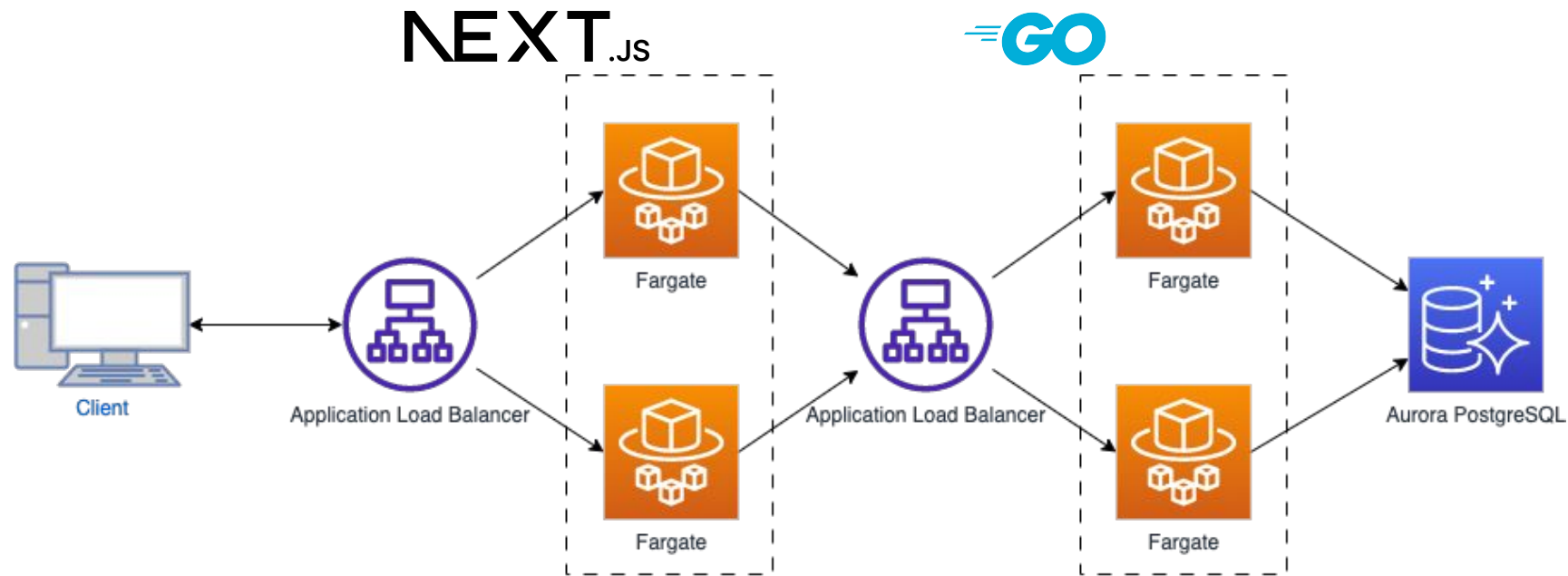
AWS Cloud Development Kit
(AWS CDK)

- 馴染みのあるプログラミング言語を使いつつ、IaCのメリットを享受できる
- L2 Constructなどによる抽象化で、直感的にインフラをコードで記述できる
- 内部でCloudFormationが使用されており、公式サービスならではの手厚いサポートや豊富な機能提供を受けられる

```
const topic = new aws_sns.Topic(scope: this, id: "SampleTopic");  
topic.grantPublish(grantee: new aws_iam.ServicePrincipal(service: "events.amazonaws.com"));
```

この2行でEventBridgeの通知先となるSNSトピックを記述できる👍

02 大まかなアプリケーション構成



CDKを書いてて最高なポイント

①圧倒的に少ないコードで書けて最高 🎉

この1行で...

```
new ec2.Vpc(scope: this, id: "VPC", props: {});
```

これ全部できあがります



↓diff

```
Resources
[+] AWS::EC2::VPC VPC VPCB9E5F0B4
[+] AWS::EC2::Subnet VPC/PublicSubnet1/Subnet
[+] AWS::EC2::RouteTable VPC/PublicSubnet1/Ro
[+] AWS::EC2::SubnetRouteTableAssociation VPC/
[+] AWS::EC2::Route VPC/PublicSubnet1/Default
[+] AWS::EC2::EIP VPC/PublicSubnet1/EIP VPCPub
[+] AWS::EC2::NatGateway VPC/PublicSubnet1/NAT
[+] AWS::EC2::Subnet VPC/PublicSubnet2/Subnet
[+] AWS::EC2::RouteTable VPC/PublicSubnet2/Ro
[+] AWS::EC2::SubnetRouteTableAssociation VPC/
[+] AWS::EC2::Route VPC/PublicSubnet2/Default
[+] AWS::EC2::EIP VPC/PublicSubnet2/EIP VPCPub
[+] AWS::EC2::NatGateway VPC/PublicSubnet2/NAT
[+] AWS::EC2::Subnet VPC/PrivateSubnet1/Subnet
[+] AWS::EC2::RouteTable VPC/PrivateSubnet1/Ro
[+] AWS::EC2::SubnetRouteTableAssociation VPC/
[+] AWS::EC2::Route VPC/PrivateSubnet1/Defaul
[+] AWS::EC2::Subnet VPC/PrivateSubnet2/Subnet
[+] AWS::EC2::RouteTable VPC/PrivateSubnet2/Ro
[+] AWS::EC2::SubnetRouteTableAssociation VPC/
[+] AWS::EC2::Route VPC/PrivateSubnet2/Defaul
[+] AWS::EC2::InternetGateway VPC/IGW VPCIGWB
[+] AWS::EC2::VPCGatewayAttachment VPC/VPCGW
```

Terraformとの比較

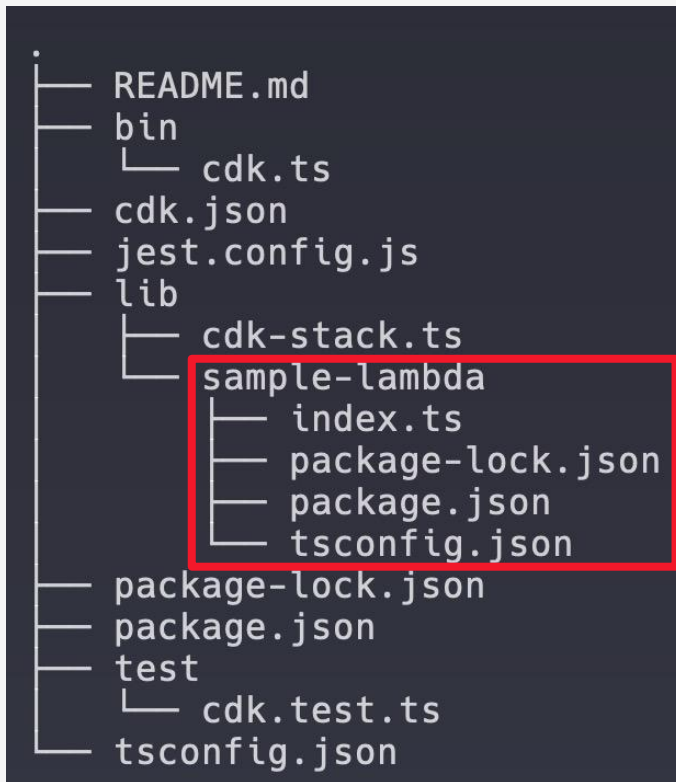
```
module "vpc" {  
  source = "terraform-aws-modules/vpc/aws"  
  version = "3.19.0"  
  
  cidr = "10.0.0.0/16"  
  azs = ["ap-northeast-1a", "ap-northeast-1c"]  
  private_subnets = ["10.0.1.0/24", "10.0.2.0/24"]  
  public_subnets = ["10.0.101.0/24", "10.0.102.0/24"]  
  enable_nat_gateway = true  
  enable_dns_hostnames = true  
}
```

terraform-aws-modules/vpc(*1)を利用した場合

(*1)<https://registry.terraform.io/modules/terraform-aws-modules/vpc/aws/latest>

②Lambdaの開発体験が良くて最高 🎉

このようなディレクトリ構造の場合...



パスを渡すだけでLambdaをデプロイできる！ 🚀

```
new cdk.aws_lambda_nodejs.NodejsFunction(scope: this, id: "SampleFunction", props: {  
  entry: "lib/sample-lambda/index.ts",  
  depsLockFilePath: "lib/sample-lambda/package-lock.json",  
  runtime: cdk.aws_lambda.Runtime.NODEJS_18_X,  
});
```

※これだけでデプロイした場合、destroyしてもCloudWatchのロググループが残るので注意

02 Lambdaの開発体験が良くて最高 🎉

aws-cdk-lib.aws-lambda-nodejs.NodejsFunction(*1)を使うと、エントリーポイントやlockファイルのpathなどを渡すだけでTypeScriptで書いたLambdaコードをいい感じにアップロードしてくれるので楽

- アルファ版ですが、PythonやGo版もあります
- Terraformだと、terraform_dataやnull_resourceを使うなどしてトランスパイル・ビルドを記述する必要がある

(*1)https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws_lambda_nodejs.NodejsFunction.html

③CDKとStepFunctionsの相性が最高🎉

**StepFunctions、GUIでワークフロー作るのは簡単だけど
IaCだとASL(*1)書くのが大変...😓**

(*1)Amazon States Language: <https://docs.aws.amazon.com/step-functions/latest/dg/concepts-amazon-states-language.html>

CDKならTypeScriptで書けます！👍

```
const random = new tasks.LambdaInvoke(scope: this, id: "Random", props: {
  lambdaFunction: randomFunc,
  payloadResponseOnly: true,
});
const task1 = new tasks.LambdaInvoke(scope: this, id: "Task1", props: {
  lambdaFunction: task1Func,
});
const task2 = new tasks.LambdaInvoke(scope: this, id: "Task2", props: {
  lambdaFunction: task2Func,
});

const choice = new stepfunctions.Choice(scope: this, id: "Choice")
  .when(condition: stepfunctions.Condition.booleanEquals(variable: "$.isEven", value: true), next: task1)
  .otherwise(def: task2)
  .afterwards();
const success = new stepfunctions.Succeed(scope: this, id: "Success");

const definition = random.next(next: choice).next(next: success);

new stepfunctions.StateMachine(scope: this, id: "StateMachine", props: {
  definition,
});
```

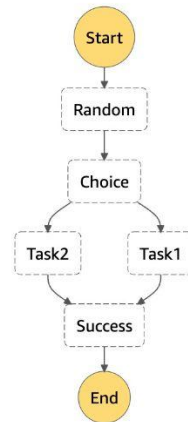
作成されたStateMachine

定義

エクスポート ▼

Layout ▼

```
21 ▾ "Choice": {
22   "Type": "Choice",
23   "Choices": [
24     {
25       "Variable": "$.isEven",
26       "BooleanEquals": true,
27       "Next": "Task1"
28     }
29   ],
30   "Default": "Task2"
31 },
32 ▾ "Task2": {
33   "Next": "Success",
34   "Retry": [
35     {
36       "ErrorEquals": [
37         "Lambda.ServiceException",
38         "Lambda.AWSLambdaException",
39         "Lambda.SdkClientException"
40       ],
41       "IntervalSeconds": 2,
42       "MaxAttempts": 6,
43       "BackoffRate": 2
44     }
45   ],
46   "Type": "Task",
47   "Resource": "arn:aws:states:::lambda:invoke",
48   "Parameters": {
```



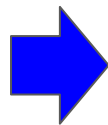
アジェンダ

1. AlphaDriveについて
2. AWS CDKの紹介
- 3. 「全員SRE」のためにやっていること**
4. まとめ

「TypeScriptでインフラ書ける」だけでは
「インフラ得意な人がTypeScriptで書く」だけ😭

日々の開発フローや構造から、「全員SRE」を目指す

03 CDKコードまで含めたモノレポで開発



いつもと違うリポジトリをさわる
のが億劫...



アプリケーションリポジトリ

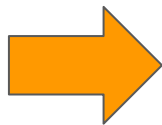
実装PR



CDKリポジトリ

バケット作成PR

03 CDKコードまで含めたモノレポで開発



単一のリポジトリ

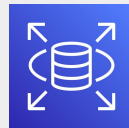
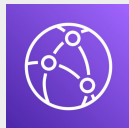
機能単位のPR

アプリケーションコードの開発と
同じようにIaC開発できる！

03 積極的にモブプロする



RemovalPolicyってなんだ...?



わからん...

03 積極的にモブプロする



既存のインフラを取り込むなら
cdk importでいけますよ！

あ、私  ったことあります！

アジェンダ

1. AlphaDriveについて
2. AWS CDKの紹介
3. 「全員SRE」のためにやっていること
- 4. まとめ**

04 まとめ

- SREはロールではなく方法論・文化・スキル👏
- AWS CDKを使うと、使い慣れたプログラミング言語で、アプリケーション開発と近い感覚でインフラをコードで書くことができます👤
- (例えば)モノレポやモブプロで、開発者がインフラにさわるハードルを下げていきましょう💪

ご清聴ありがとうございました