

株式会社マイナビ

マイナビの内製開発を支えるコンテナ基盤設計開発の舞台裏 ～自動化で楽々デプロイ！でもアプリチームからはちょっと不評？～

2025.2.13

株式会社マイナビ
デジタルテクノロジー戦略本部

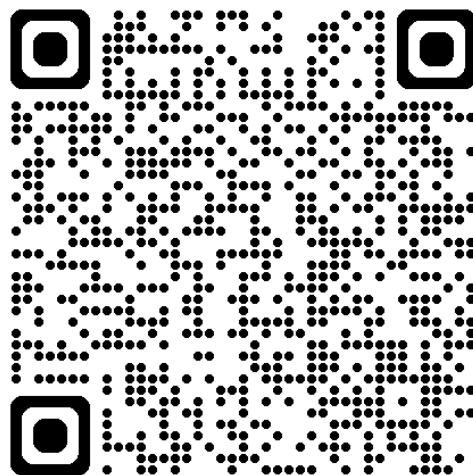


アジェンダ

- 01 | マイナビについて
- 02 | コンテナ集約基盤の構想背景
- 03 | 集約基盤の実現のためのアプローチ
- 04 | 集約基盤を支える技術と自動化の仕組み
- 05 | リリース後の裏話~アプリチームとの食い違い~
- 06 | 今後の展望
- 07 | おわりに

本日の資料やより詳細な取り組みを知りたい方は
エンジニアブログへ

マイナビ エンジニアブログ 🔍



01

マイナビについて



50年の歴史で培ったノウハウ

1973年8月15日

新聞の発行及び出版業、
絵画・美術品の輸入販売業等を
目的として設立

1990年代

PC系雑誌創刊
就職情報企画が
インターネット企画へ発展

2010年代

社名を（株）マイナビへ。
人材以外の情報メディア事業に進出。
積極的なM&Aでグループを拡大。

1980年代

就職情報誌発刊
一般書店での書籍販売開始
中国との交流事業等

2000年代

職業紹介事業、
転職、アルバイト情報事業等
多様な人材領域へ進出

2020年代

東アジアを中心とした企業の
M&Aを進める。
グローバル企業へ

50年の変遷の中で培ったノウハウは他社にない大きな強みです。

セグメント

サービス・企業名

キャリアデザイン

 マイナビ 2026 マイナビ 新卒紹介 マイナビ 進学 マイナビ 研修サービス

HR

 マイナビ 転職 マイナビ AGENT マイナビ バイト マイナビ ミドルシニア

ヘルスケア & ウェルネス

 マイナビ 看護師 マイナビ 薬剤師 マイナビ 健康経営 マイナビ 福祉・介護のシゴト

人材派遣BPO

 マイナビ スタッフ マイナビ クリエイター マイナビ キャリレーション

メディア & サービス

 マイナビ ニュース マイナビ ティーンズ マイナビ ウーマン マイナビ 学生の窓口

海外

DXデザイン事業(海外オフショア事業)／Mynavi USA Corporation／
Mynavi Solutions India Private Limited

など

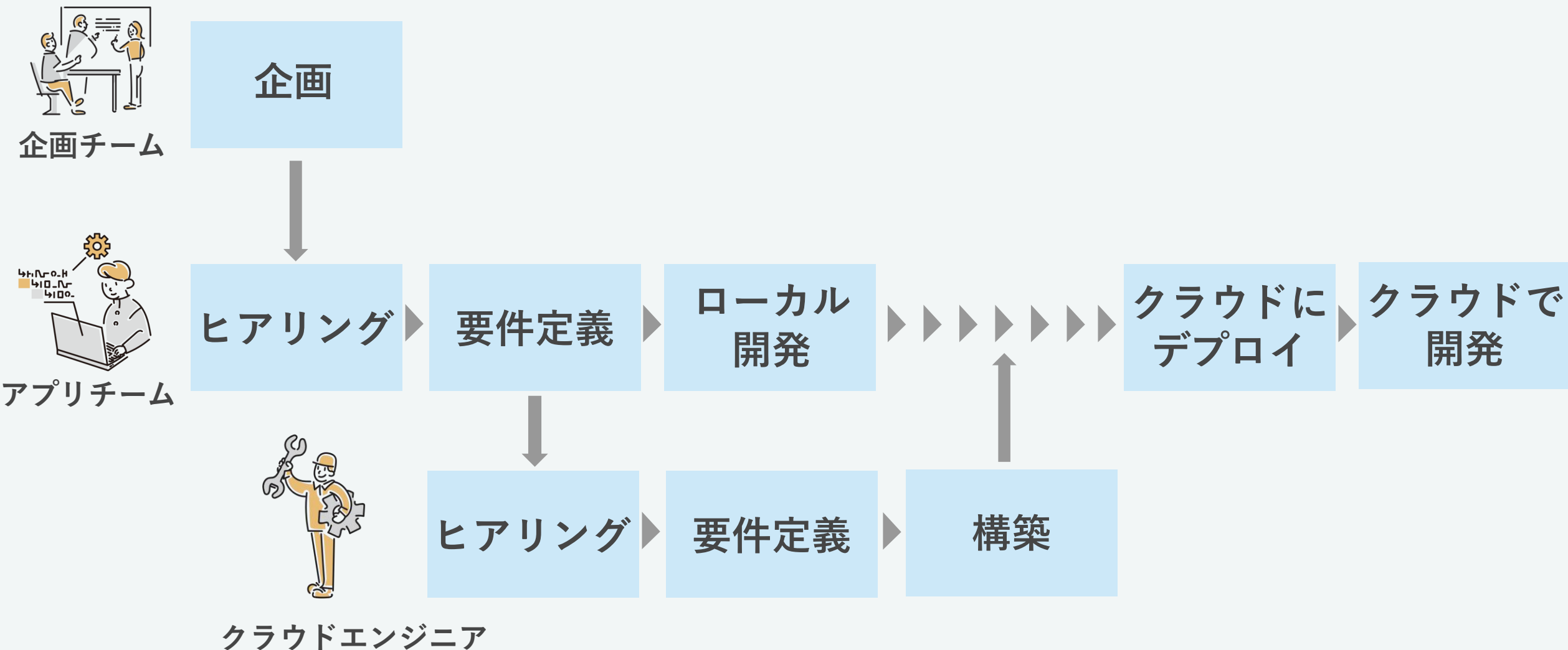
※2024年12月時点

02

コンテナ集約基盤の 構想背景



マイナビにおける内製開発の進め方



AWS CDKを利用したリソースのデプロイ

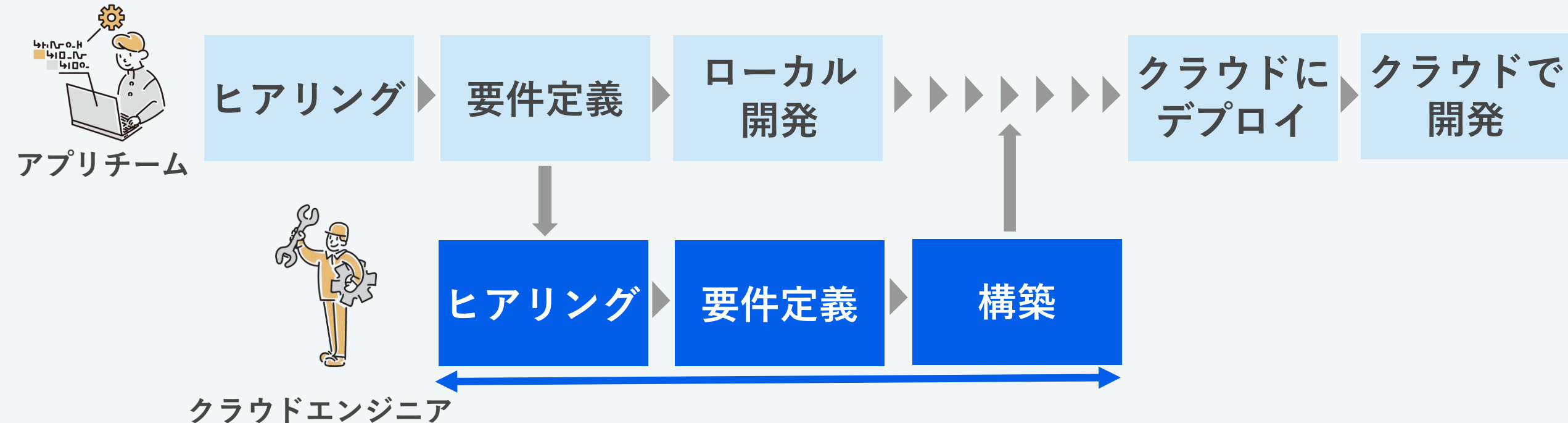


BLEA (Baseline Environment on AWS)
を自社向けにカスタマイズ

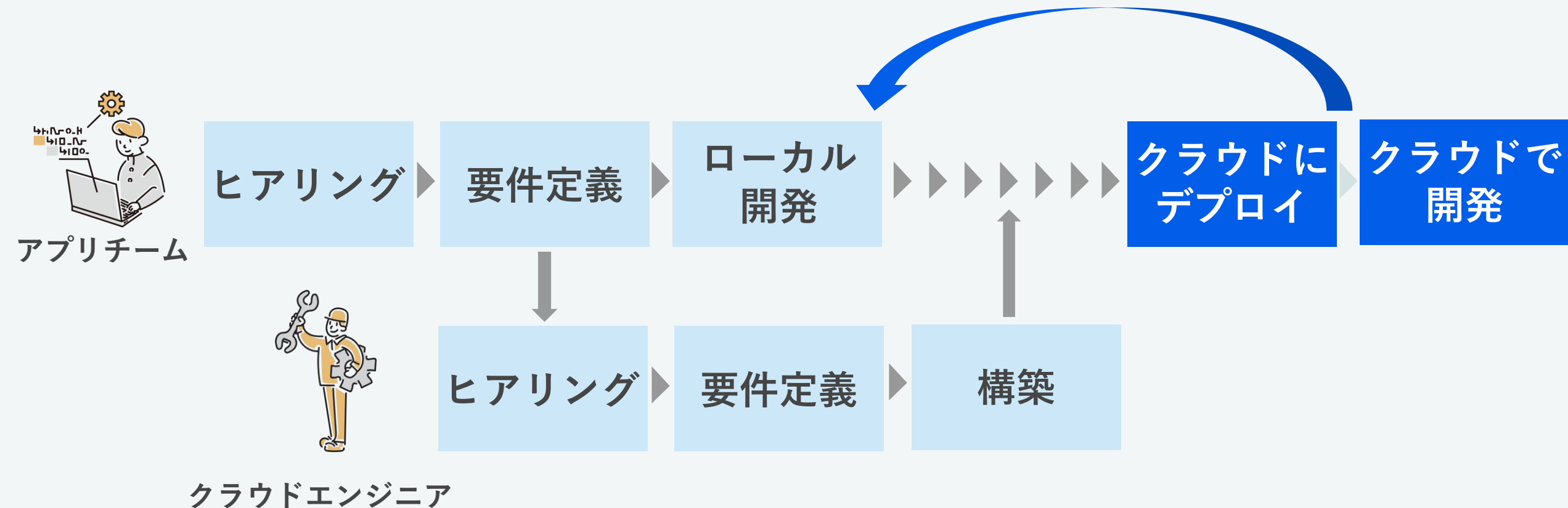


PJの要件に合わせてCDKコードを開発

①最低でも1ヵ月程度のリードタイムが発生

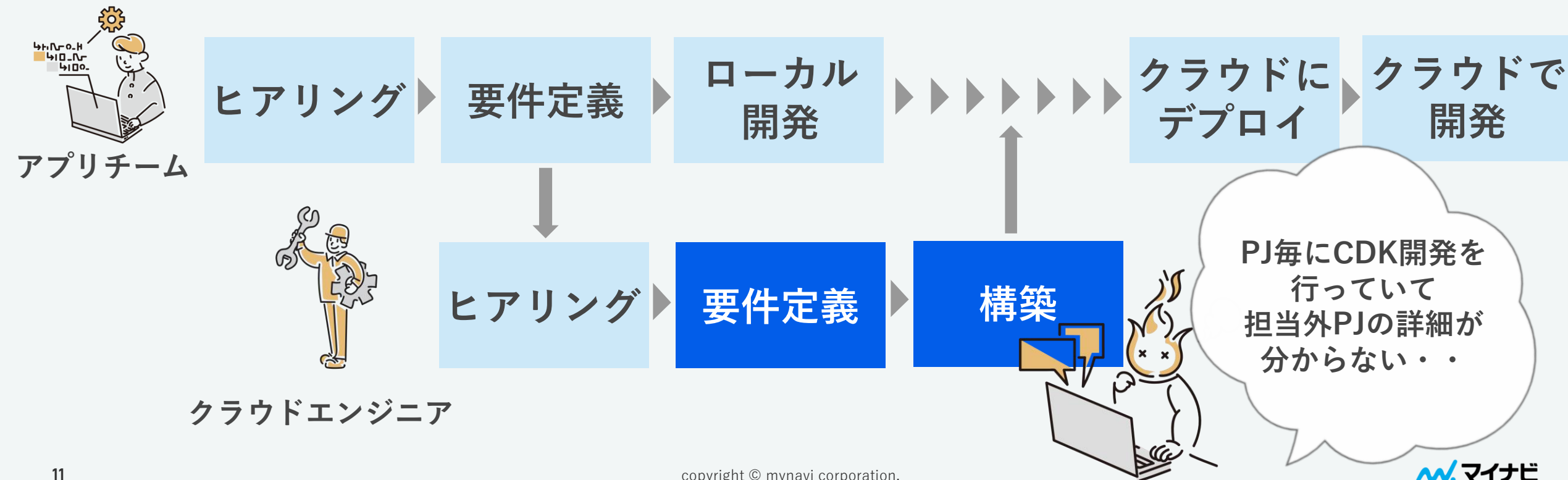


② インフラ環境を渡した後の手戻り



今までの進め方における課題点

③ 運用の属人化



こういった課題を解決するために
マイナビが目指したのは・・・

「数日のリードタイムで環境を提供」できて
「開発時の手戻りが少なく」、
「誰でも運用可能な」
プラットフォームの開発

As Is

最低でも1ヵ月程度の
リードタイムが発生

インフラ環境を渡した後に
手戻りが発生するケースが多い

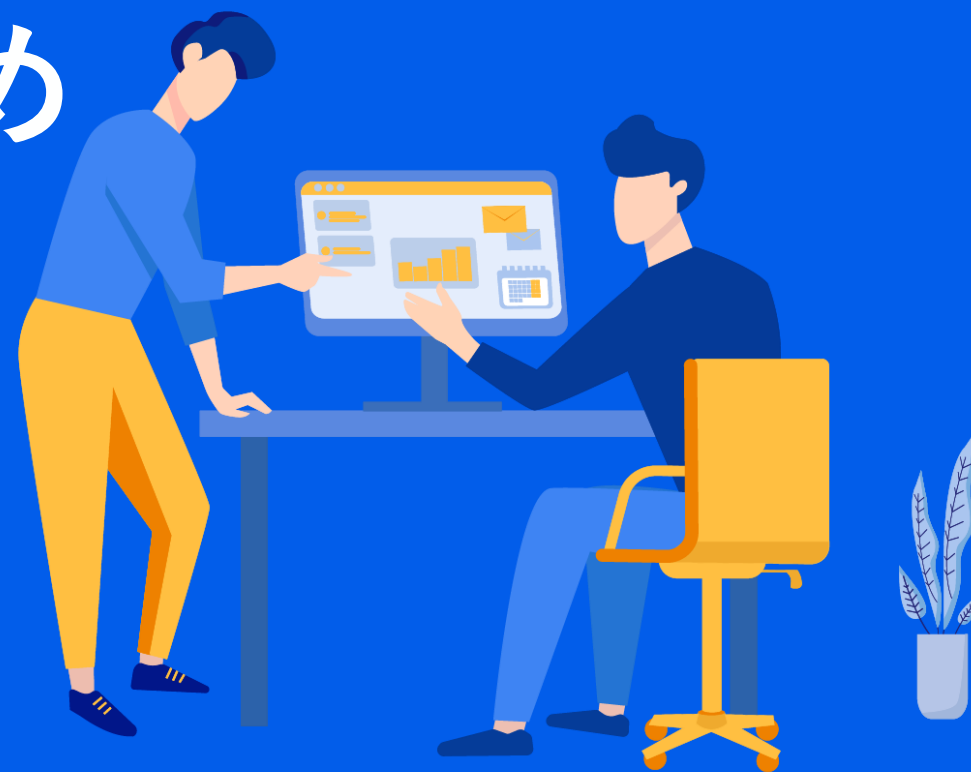
運用の属人化

To Be

- ✓ コンテナ基盤を用意して、
数日のリードタイムで提供
- ✓ 早期にクラウドでの開発へシフト
し、自由に開発を進めてもらう
- ✓ サービスを基盤に集約して、
全員で運用

03

集約基盤の実現のための アプローチ



マルチテナント型集約基盤へのアプローチ

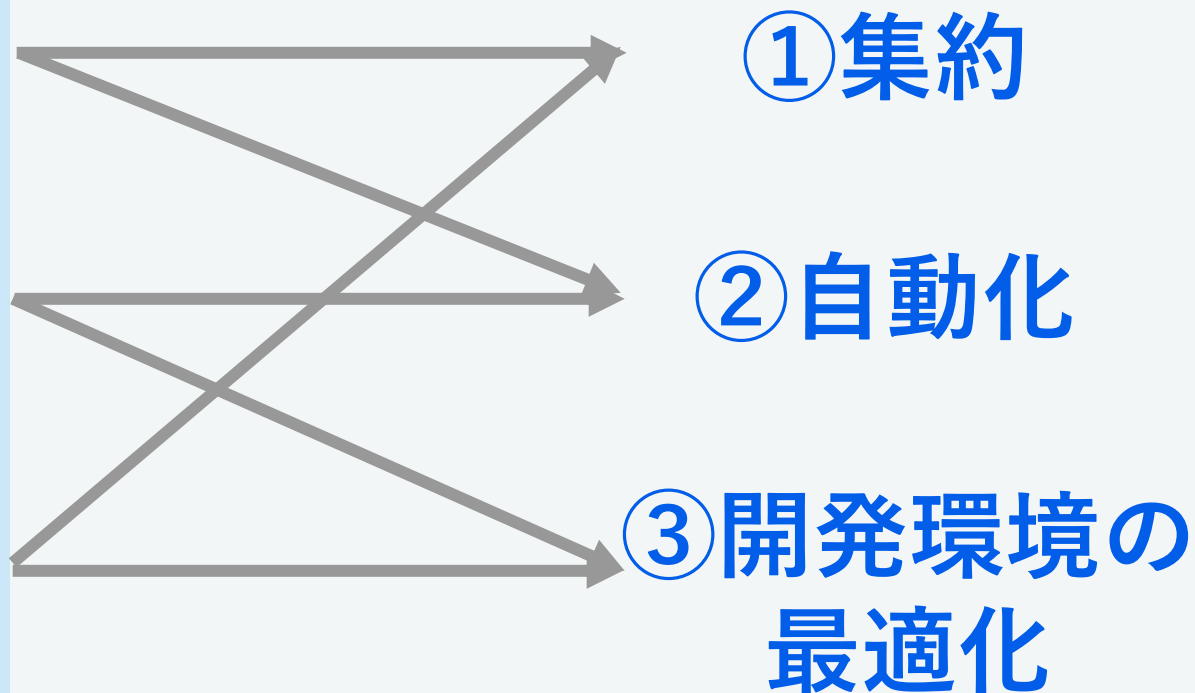
03.集約基盤の実現のためのアプローチ

To Be

✓ コンテナ基盤を用意して、
数日のリードタイムで提供

✓ 早期にクラウドでの開発へシフト
し、自由に開発を進めてもらう

✓ サービスを基盤に集約して、
全員で運用



マルチテナント型集約基盤へのアプローチ

03.集約基盤の実現のためのアプローチ

To Be

✓ コンテナ基盤を用意して、
数日のリードタイムで提供

✓ 早期にクラウドでの開発へシフト
し、自由に開発を進めてもらう

✓ サービスを基盤に集約して、
全員で運用

①集約

②自動化

③開発環境の
最適化

①集約

AWSアカウントとリソースの集約

- ・ アカウント作成にかかるリードタイム
- ・ 大量のアカウント管理が必要
- ・ サービス毎にNAT GatewayやVPCエンドポイントなどのリソースを作るとコストが高い



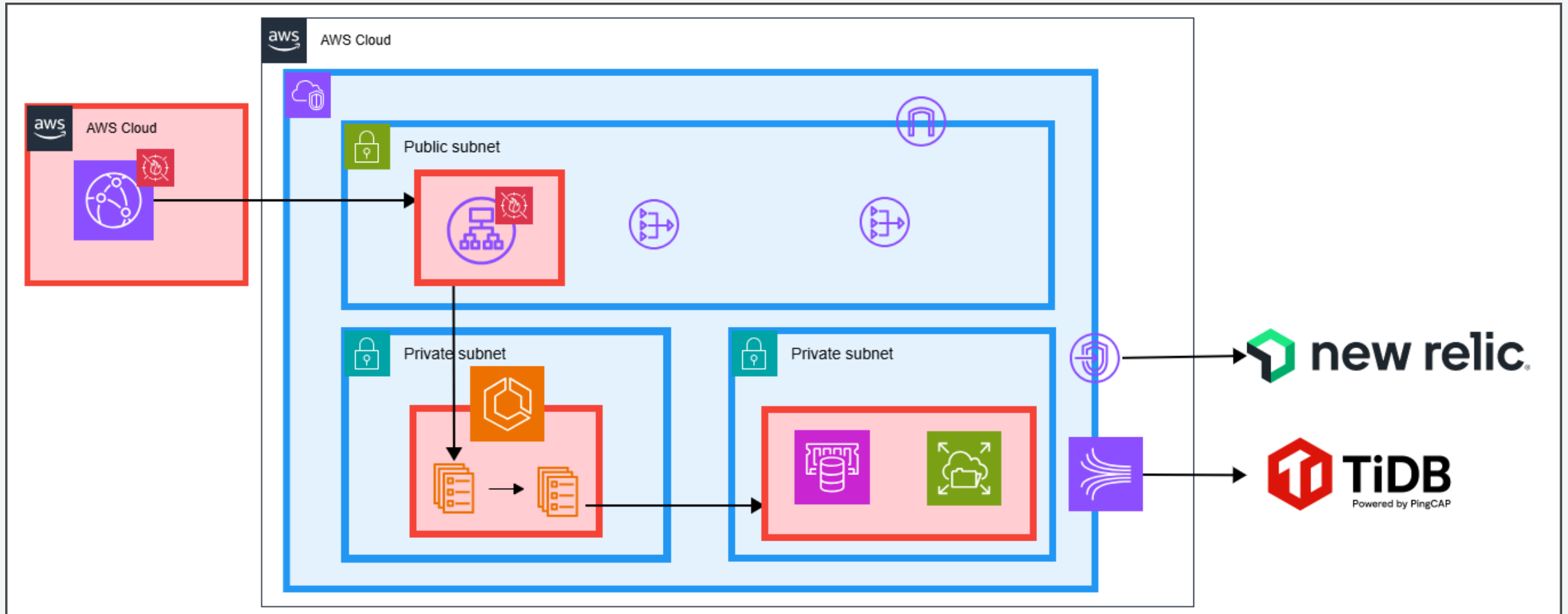
- ・ アカウントを事前に用意することで事務手続きが不要に
- ・ アカウントが集約されているため一部のアカウントのみ管理すればOK
- ・ 共有リソース/専有リソースに分け、一部の共有可能なリソースを節約



①集約

共有リソース：環境単位で作成しサービスで共有

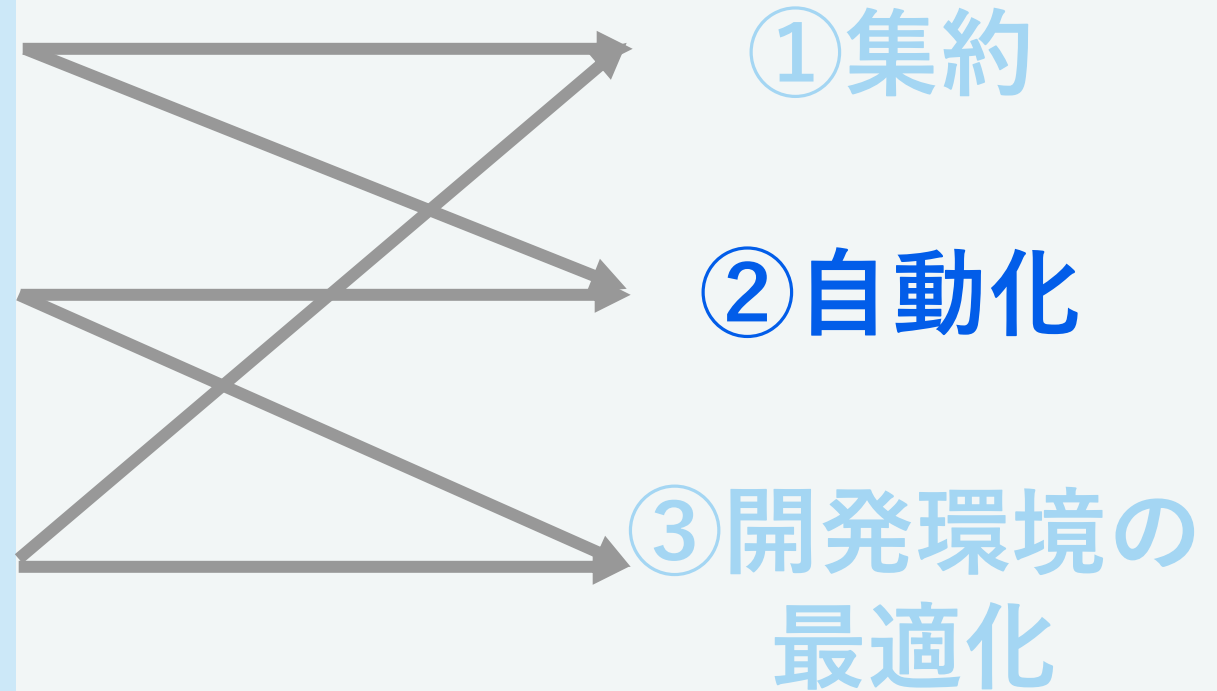
専有リソース：各環境にサービス単位で作成



解消するためのアプローチ

To Be

- ✓ コンテナ基盤を用意して、数日のリードタイムで提供
- ✓ 早期にクラウドでの開発へシフトし、自由に開発を進めてもらう
- ✓ サービスを基盤に集約して、全員で運用



②自動化

コンテナ構成のパターン化

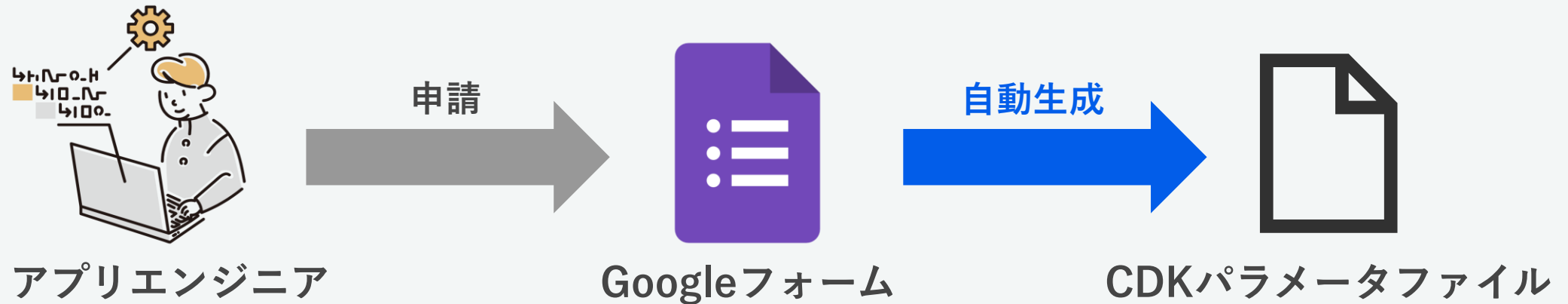
- ・PJ毎にCDK開発をするため
リードタイムや属人化が発生する

- ・コンテナ構成をパターン化し
リードタイムを短縮
- ・構成をパターン化することで
誰でも運用が可能に◎



②自動化

✓ 申請内容からパラメータファイルを自動生成

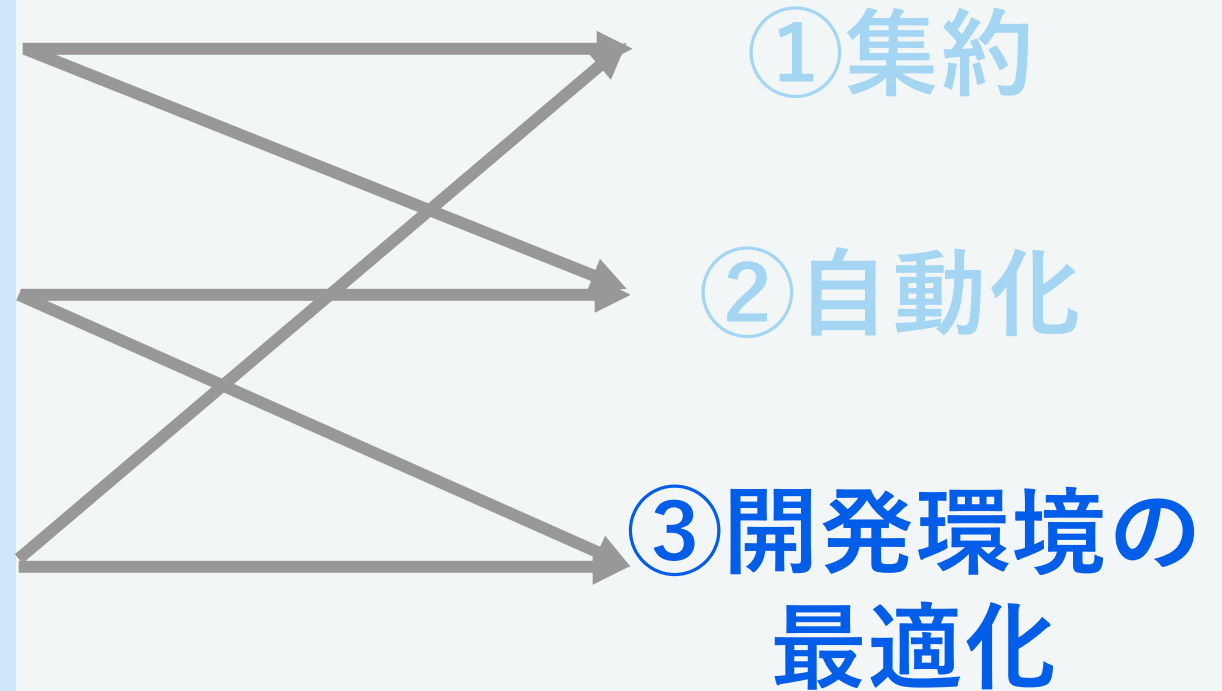


自動化による工数削減とリードタイム短縮

解消するためのアプローチ

To Be

- ✓ コンテナ基盤を用意して、数日のリードタイムで提供
- ✓ 早期にクラウドでの開発へシフトし、自由に開発を進めてもらう
- ✓ サービスを基盤に集約して、全員で運用



③開発環境の最適化

コンテナのデプロイ方式のパターン化

ecspressoを使用したローリングデプロイ

- ・ ecspressoを使用
- ・ 環境変数はCodeBuild内で設定
- ・ アプリチーム側でサービスの設定更新対応が可能

CodeDeployを使用したBlue/Greenデプロイ

- ・ CDKでECSサービス/タスクを作成
- ・ 環境変数はCDK内のpropsで設定
- ・ BGデプロイが必要な要件のために用意

③開発環境の最適化

オブザーバビリティツールの選定

特徴

- インフラのメトリクスからアプリのログ監視までを一元管理できる
- 独自にカスタマイズ可能なダッシュボード
- 異常検知と通知機能が充実
- 多くのツールやサービスとシームレスに連携

選定理由

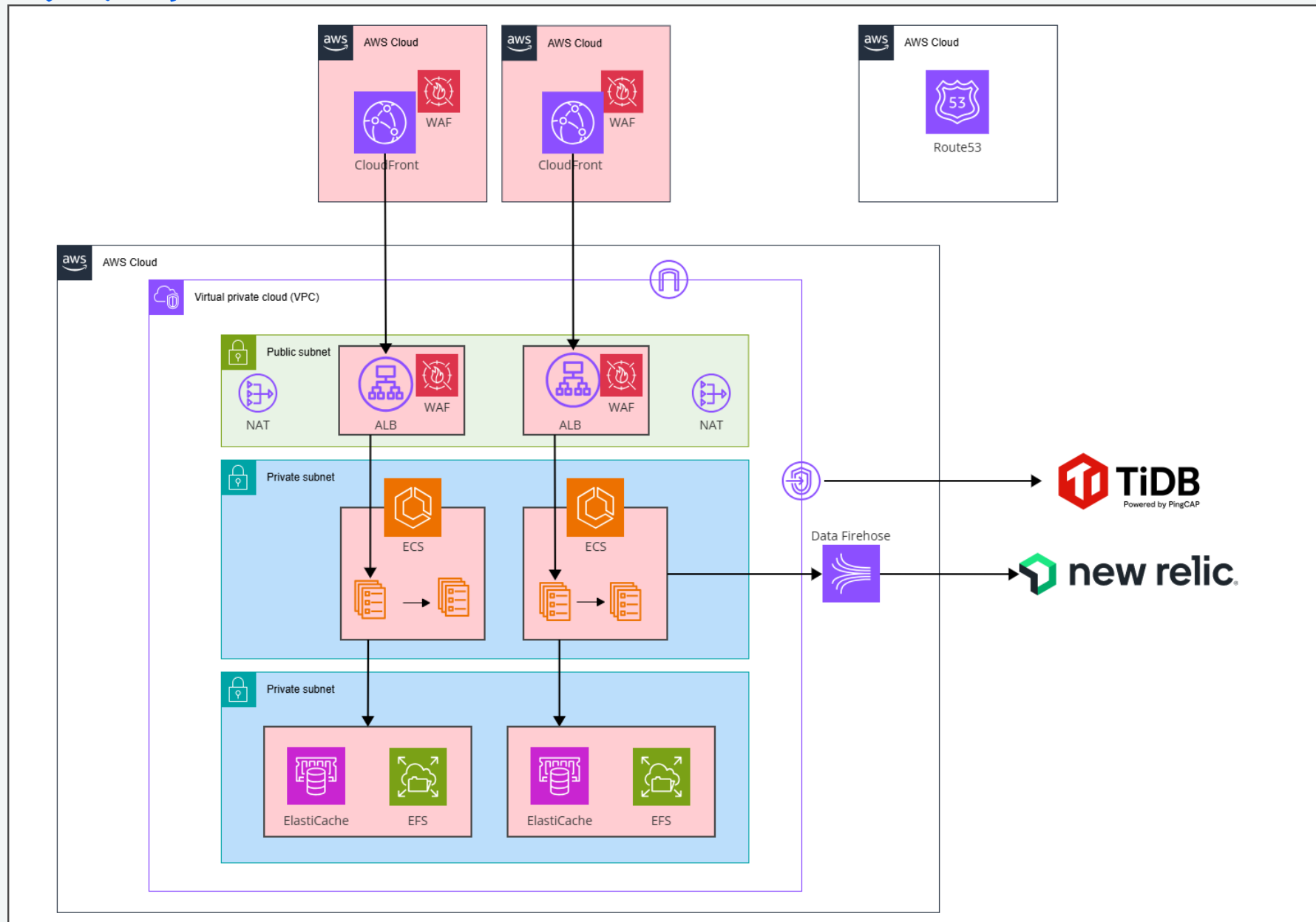
- AWSコンソール外でログやCloudWatchメトリクスが確認できる
- ユーザーライセンス数＋データ転送量に応じた料金体系
→ **コンテナを多く稼働させる集約基盤では、コスト効率が良い**

New Relicを採用





マルチテナント型の集約基盤



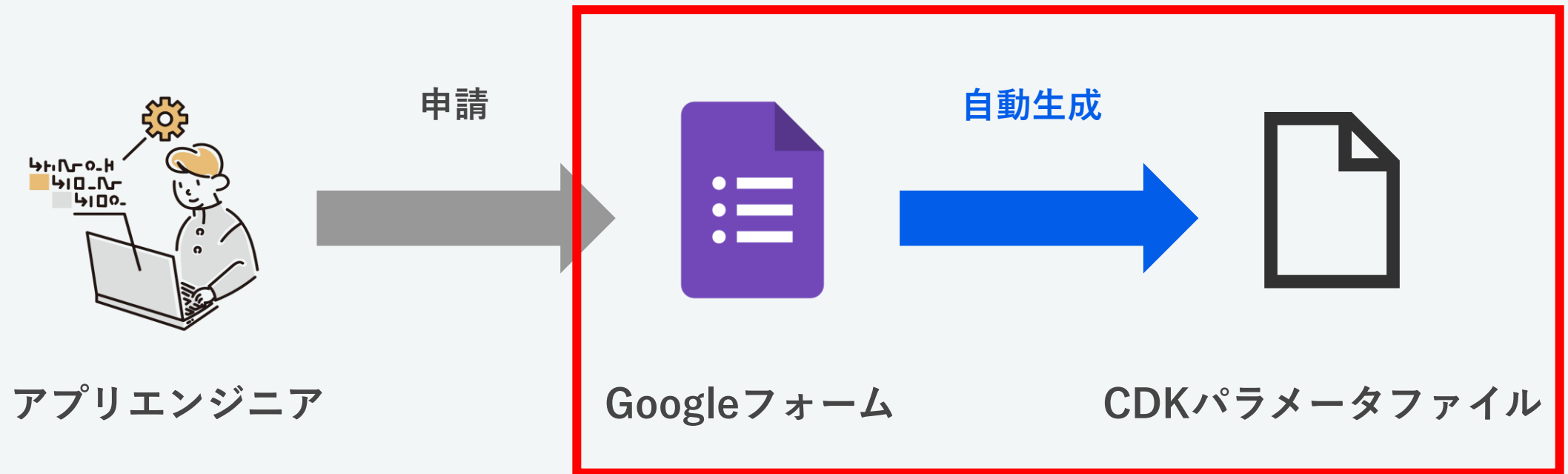
04

集約基盤を支える技術 と自動化の仕組み



デプロイ自動化に向けた 開発アプローチ

GoogleフォームからCDK用パラメータファイルを作成



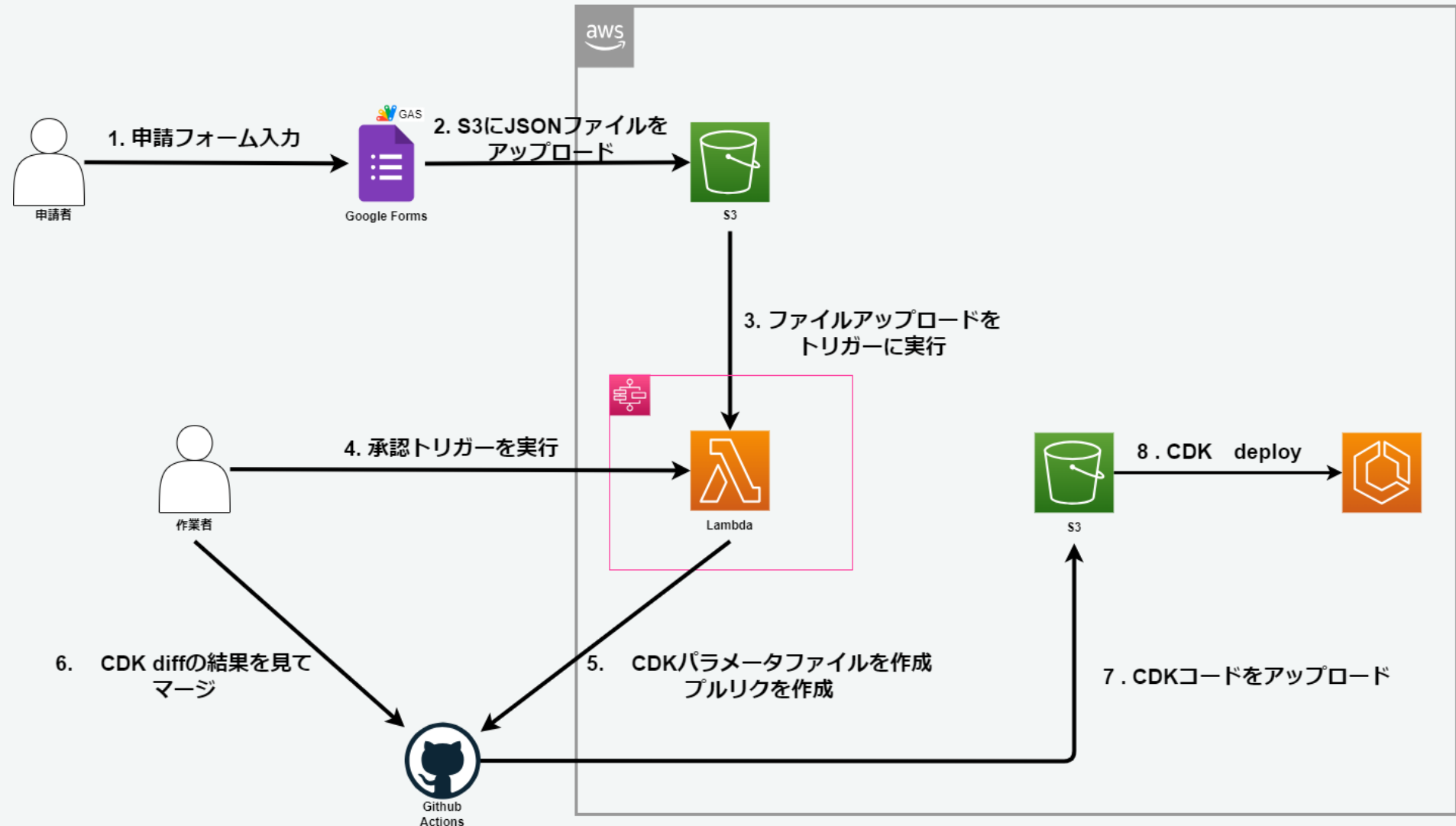
なぜパラメータファイルが必要なのか

集約基盤では複数サービスを管理する必要がある

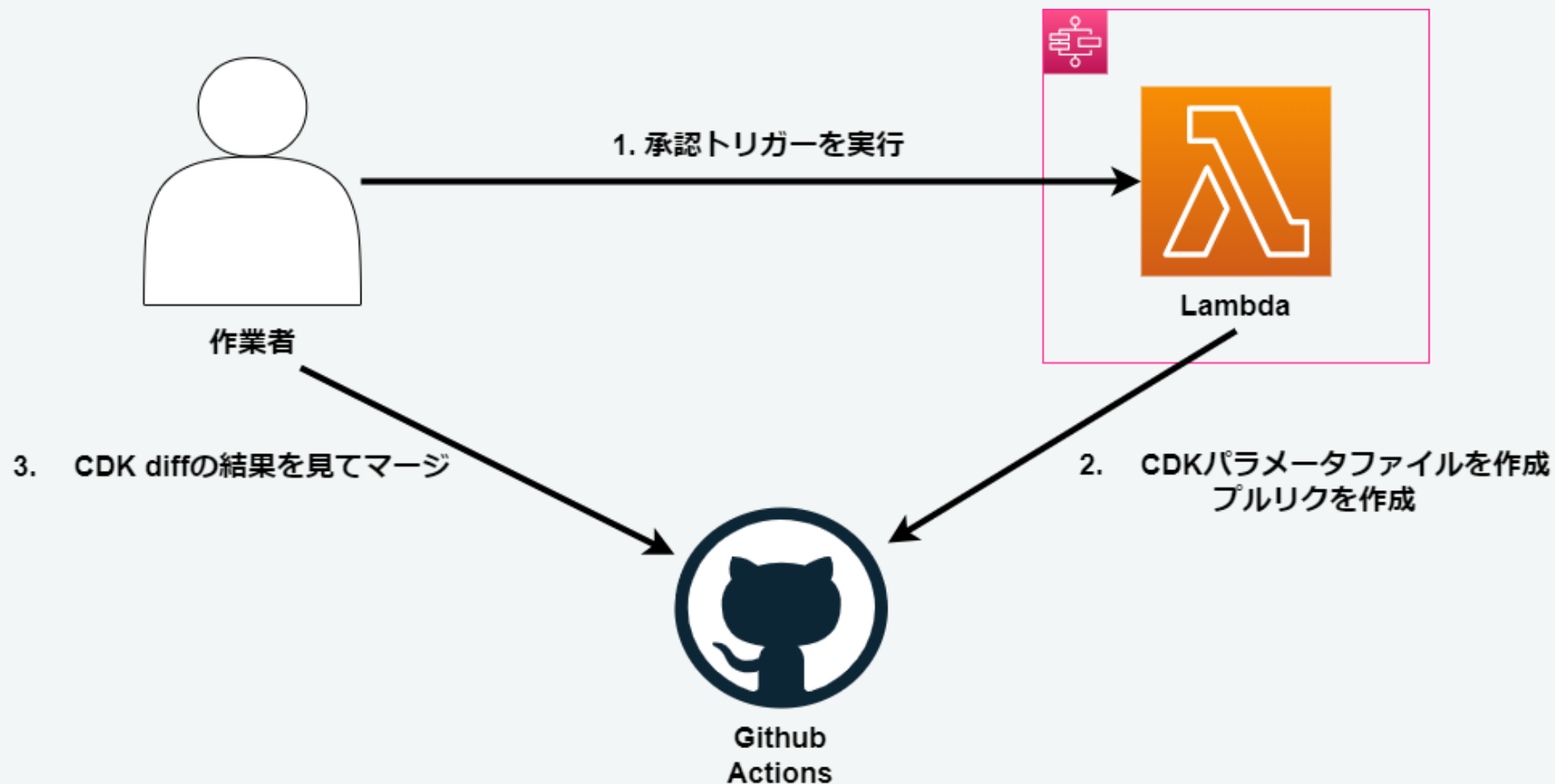
各サービスで異なるパラメータ設定が必要となる

上記の条件でサービスを管理する最適な手法がCDKパラメータファイルでの管理

パラメータファイル作成時の構成図



手動作業を自動化！



CDKで管理するためのファイル作成は
自動化できたが・・・
ACMのDNS検証という未サポート動作があった

要件：
クロスアカウントで
ACMのDNS検証を
CDKで実行

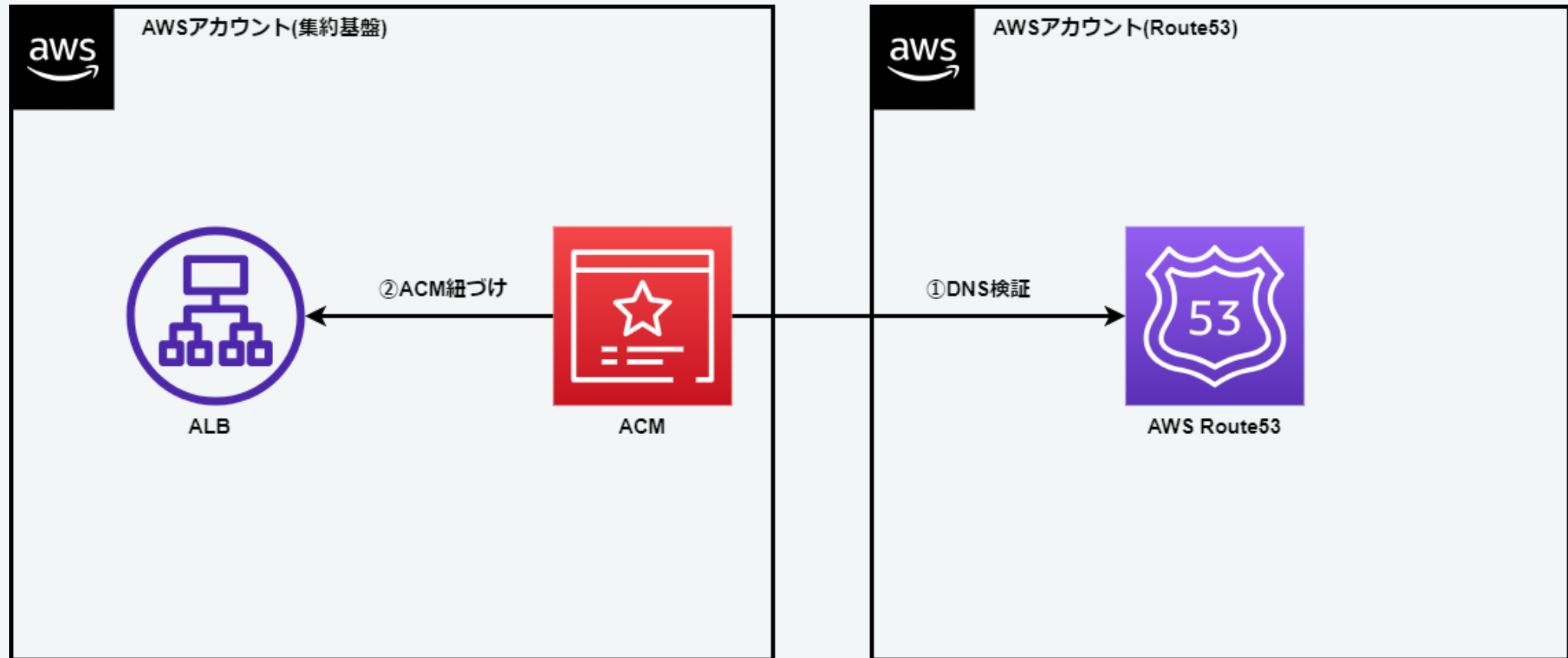
問題点：
CDKで実行したいが
要件の動作はCDKでは
未サポート

方針：
カスタムリソースを
活用する

なぜクロスアカウントアクセスが必要なのか

04.集約基盤を支える技術と自動化の仕組み

集約基盤アカウントとRoute53アカウントが分かれているため

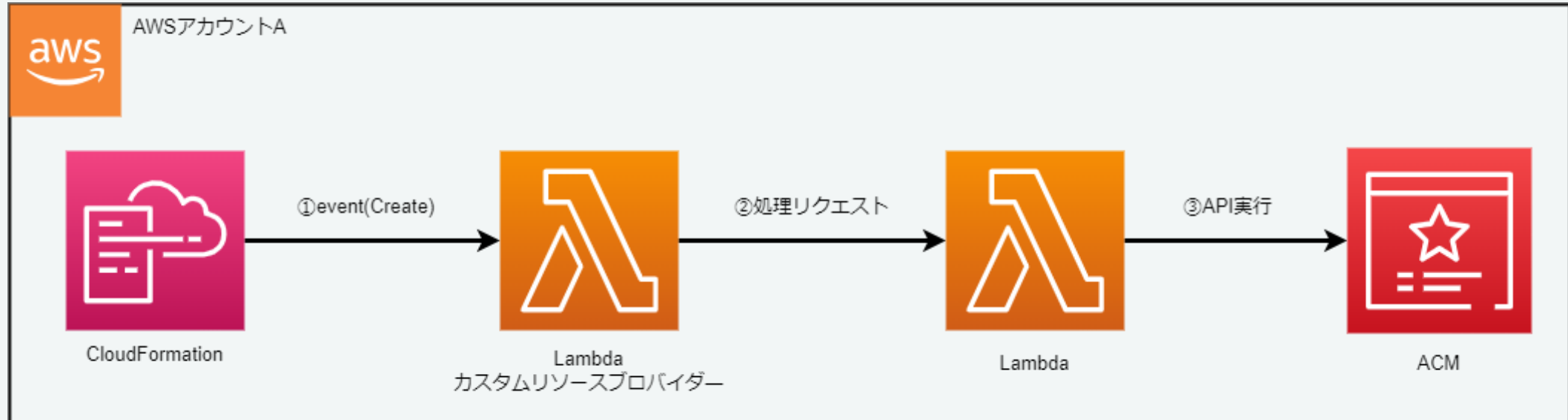


カスタムリソースとは？

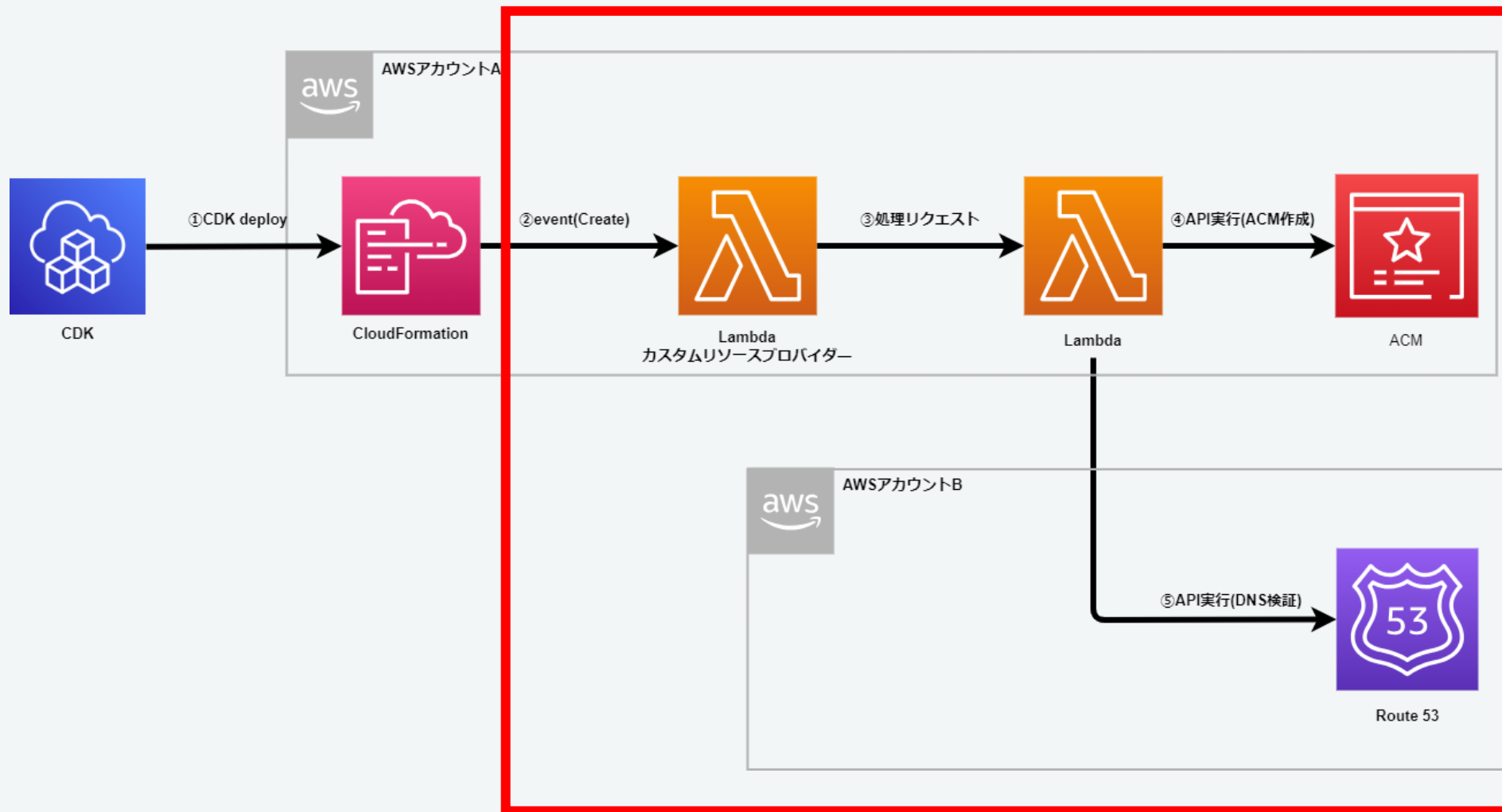
カスタムリソースの仕組み

CloudFormationの処理を拡張する仕組み

イベントCreate・Update・Deleteをトリガーに、任意のAWS Lambdaを実行し処理を実現する



カスタムリソース構成図



ACMの設定をパラメータで制御し、設定を自動化できること

```
8   export const ACMParam: inf.IACMParam = {
9     AcmDomainName: '',
10    HostZoneId: '',
11    AssumeRoleArn: '',
12  };
```

```
131  if (config.ACMParam.AcmDomainName) {
132    new AIDAliasStack(app, `${prefix}-AlbAlias`, {
133      AliasDomainName: config.ACMParam.AcmDomainName,
134      Route53HostZoneId: config.ACMParam.HostZoneId,
135      AssumeRoleArn: config.ACMParam.AssumeRoleArn,
136      appAlb: ecs.app.frontAlb.appAlb,
137      env: getProcEnv(),
138      crossRegionReferences: true,
139      providerServiceToken: shareResources.AliasCustomResourceProviderServiceToken,
140    });
141  }
```

カスタムリソースを利用した成果



AWSリソースについてCDKで
一元管理することが可能に



今まで手動で実施していた
ACMのDNS検証設定を自動化

**手動作業を自動化し、
申請から1日で環境の引き渡しが可能となりました。**

05

リリース後の裏話 ～アプリチームとの食い違い～



集約基盤が形になり
内製開発で利用が始まったが・・・

アプリチームからは数々の不満の声が



マネコンに入れず
開発し辛い

デプロイ手順が
複雑で分かりにくい

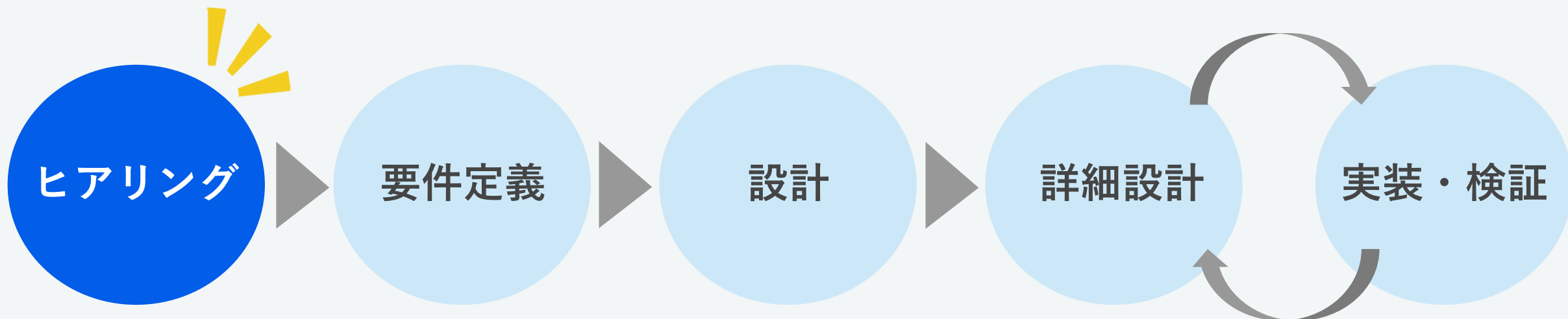
新しい
ソリューションの
使い方が分からない



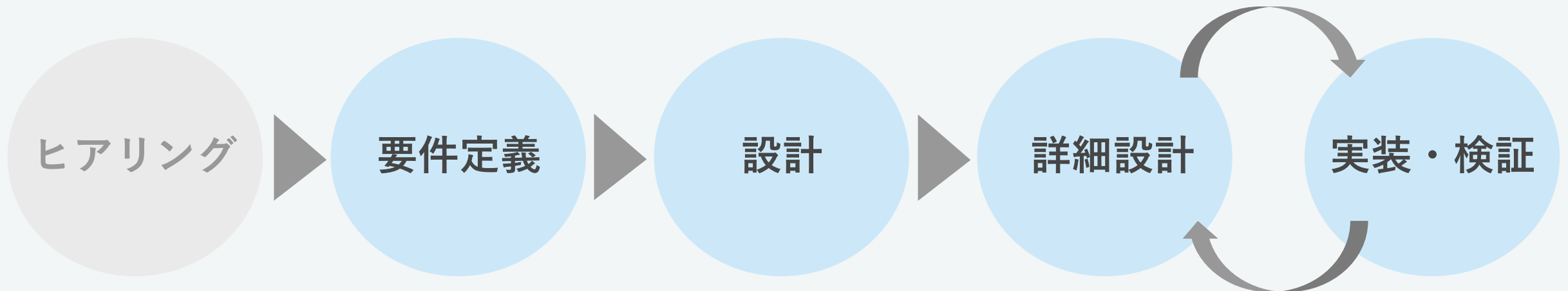
Q.いったい何が問題だったのか？

プラットフォームエンジニアリングの考えの元、
開発者が効率的に開発を行える基盤を作ることが理想・・

集約基盤開発の流れ



開発者に対するヒアリングが不足していた



A.ヒアリングが不足しており、
開発者ライクな集約基盤になっていなかった

実際にアプリチームから寄せられた FBの内容をご紹介します

事例1



デプロイ手順が複雑で分かりにくい

要因1: 集約したことで考慮点が増えた結果、
従来のデプロイフローより複雑化してしまった

①初回デプロイ

- ・デプロイするための初期設定
 - 必要リソースインストール
 - 実行権限付与
 - 設定ファイルの事前準備

②本デプロイ

- ・デプロイ設定
 - 環境変数設定 (CLI or ファイルアップロード)
 - GitHub Actionsの実行

事例1



デプロイ手順が複雑で分かりにくい

要因2: デプロイに色々な知見が必要

前述のデプロイ作業を行うために必要となる知識が多く、初見でのハードルが高い



aws



AWS CLI



Amazon ECS





デプロイ手順が複雑で分かりにくい



ハンズオン形式と一緒にデプロイ手順をレクチャー



デプロイ設計を見直し、手順を簡素化



マネジメントコンソールへログインができず、開発がやりにくい

集約基盤ではAWSアカウントがサービスで共通となるため、
利用者のマネジメントコンソールへのログインを制限している。
そのため、従来の内製案件で可能だったことができなくなってしまった。

- デプロイの状況を確認したかった
- CloudWatchでログやメトリクスを確認したい
- Secretsをマネジメントコンソールで管理したい

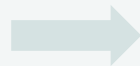
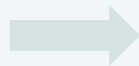




マネジメントコンソールへログインができず、開発がやりにくい



Build Statusを連携する仕組みを追加



New Relicのダッシュボードをカスタマイズして
必要なログ/メトリクスの収集を効率化

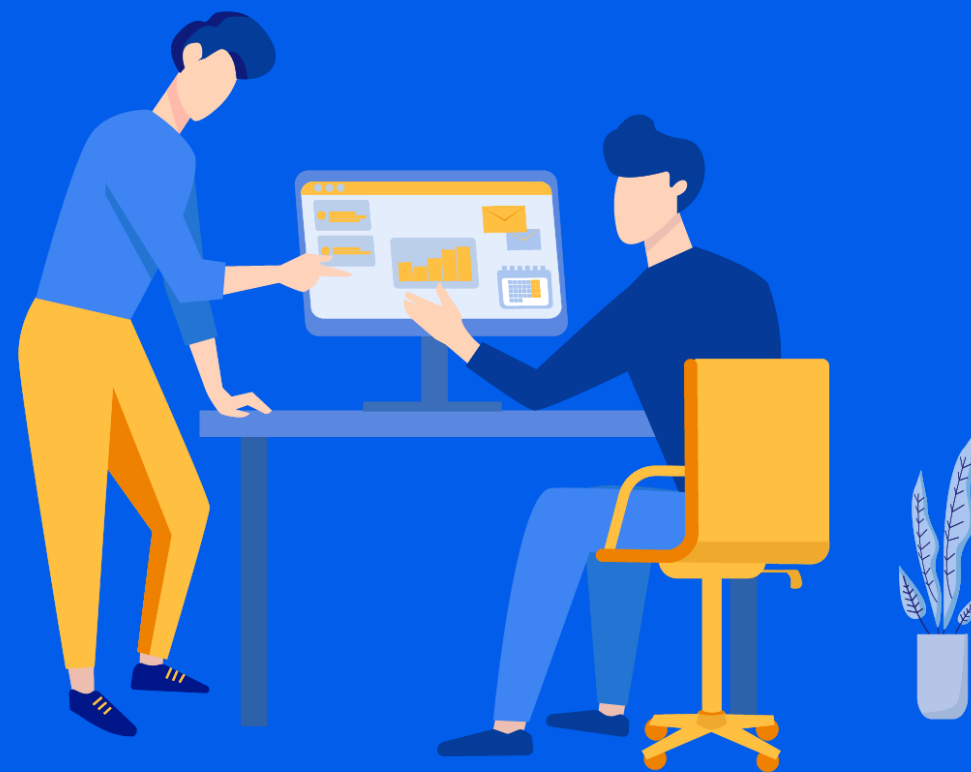


Secretsの管理方法をより簡素に

**利用者の立場に寄り添った
開発を行うことが重要！**

06

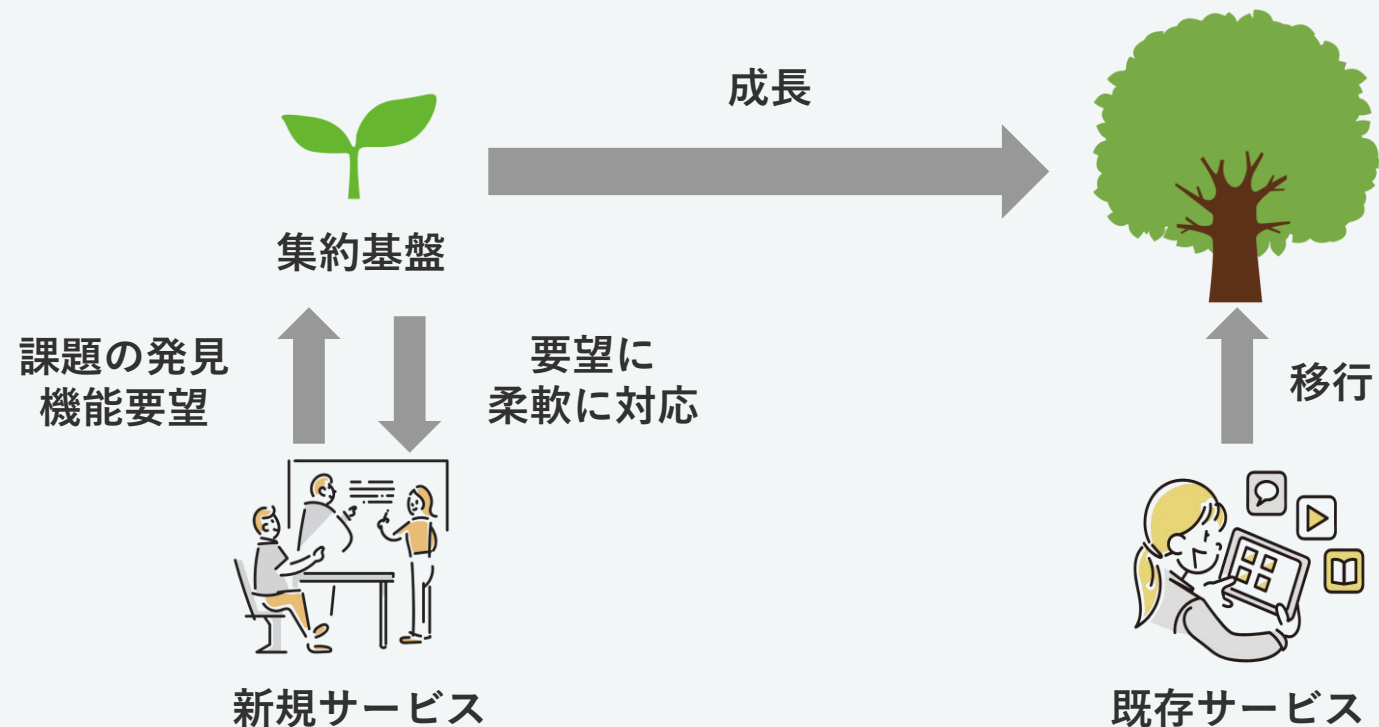
今後の展望



✓ 既存サービスの集約

集約基盤はまだまだ成長過程

利用実績を増やし、既存サービスも集約していく



 ビジネス指標を可視化し、アプリチーム以外へもメリットを提供

APMを導入することで
KPI/KGIをダッシュボード上で可視化し、
SE職や経営層など開発エンジニア以外にも
価値を提供する。

 多くのサービスが稼働する集約基盤へ

追加開発やマルチクラウド化を検討し
よりスピーディーでフレキシブルな基盤に進化し
新たなイノベーションへ貢献



07

おわりに



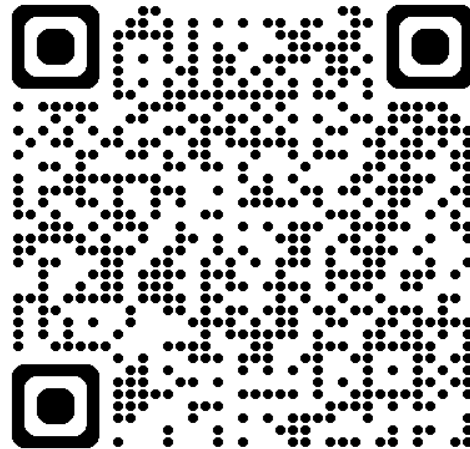
**マイナビではエンジニア採用
強化しております！**

募集ポジション

- クラウドエンジニア(SRE)
- 開発エンジニア
- WEBアプリケーションエンジニア
- データエンジニア／データサイエンティスト など

詳細はぜひ以下にて検索ください！

マイナビ IT採用サイト



ご清聴ありがとうございました

Thank you for listening!