

Kamal って便利? 社内プロジェクト 3 つを Kamal + AWS で運用した体験談

2025/09/26 @yappu



Partner with RubyStackNews^I

Independent Ruby & Rails publication for senior developers

Why RubyStackNews?

- Focused on Ruby and Ruby on Rails
- Long-form articles based on real conference talks
- Audience of senior developers and tech leads
- Readers from the US, Europe, and Asia

RubyStackNews turns conference talks and real-world experience into practical, production-focused technical articles.

Partnerships & Sponsorships

- Article sponsorships
- Inline placements inside articles
- Sidebar visibility

<https://rubystacknews.com/partner-with-rubystacknews/>

自己紹介

- yappu / 株式会社ソニックガーデン
- Rails4 年目、インフラ 1 年目
- 趣味は釣り、V の推し活、登山



今日話すこと

- Kamal って何?
- Kamal + AWS でどのように運用してるのか
- Kamal で良かったこと、困ったこと
- まとめ

質問です

**Kamal、使ったことあるよ！
って方👉**

Kamal、よく分からない
って方👉

Kamal って何?

Deploy web apps anywhere.

From bare metal to cloud VMs.

```
~/Work/demos/alpha
Neo-tree
├── ~/Work/demos/alpha
│   ├── app
│   ├── bin
│   └── config
│       ├── environments
│       ├── initializers
│       ├── locales
│       ├── application.rb
│       ├── boot.rb
│       ├── cable.yml
│       ├── cache.yml
│       ├── credentials.yml.enc
│       ├── database.yml
│       └── deploy.yml
│           ├── environment.rb
│           ├── importmap.rb
│           ├── puma.rb
│           ├── queue.yml
│           ├── recurring.yml
│           ├── routes.rb
│           └── storage.yml
└── db

routes.rb  × | ! deploy.yml  ×
21  - RAILS_MASTER_KEY
20  clear:
19  # Run the Solid Queue Supervisor inside the web server's Puma process to do job
18  # When you start using multiple servers, you should split out job processing to
17  SOLID_QUEUE_IN_PUMA: true
16
15  # Set number of processes dedicated to Solid Queue (default: 1)
14  # JOB_CONCURRENCY: 3
13
12  # Set number of cores available to the application on each server (default: 1).
11  # WEB_CONCURRENCY: 2
10
9  # Match this to any external database server to configure Active Record correct
8  # DB_HOST: 192.168.0.2
7
6  # Log everything from Rails
5  RAILS_LOG: debug
4
3  # Aliases are triggered with "bin/kamal <alias>". You can overwrite arguments on in
2  # "bin/kamal logs -r job" will tail logs from the first server in the job section.
1  aliases:
56  console: app exec --interactive --reuse "bin/rails console"
1  shell: app exec --interactive --reuse "bash"
```

**Web アプリを物理でもクラウドでも
どこでもデプロイできるツール**

Rails8 からデフォルトで入るように

Add Kamal by default to Rails 8 #51798

Merged


dhh merged 18 commits into `main` from `kamal-by-default` on May 14, 2024

Conversation 13

Commits 18


Checks 0


Files changed 9


dhh commented on May 13, 2024


Member


Rails should have a default answer to deploying applications out of the box to cloud or bare metal. Kamal is that answer. We'll configure it automatically as far as we can, but allow an opt-out with `--skip-kamal`.





 15

 3

 15

 11

  Add Kamal by default

b6bf5be

<https://github.com/rails/rails/pull/51798>

Kamal の特徴

- どこにでもデプロイできる
- Docker ベースのデプロイツール
- yaml ファイルでシンプルな設定
- ブルーグリーンデプロイ

とにかくシンプル

例えば、 **setup** コマンド

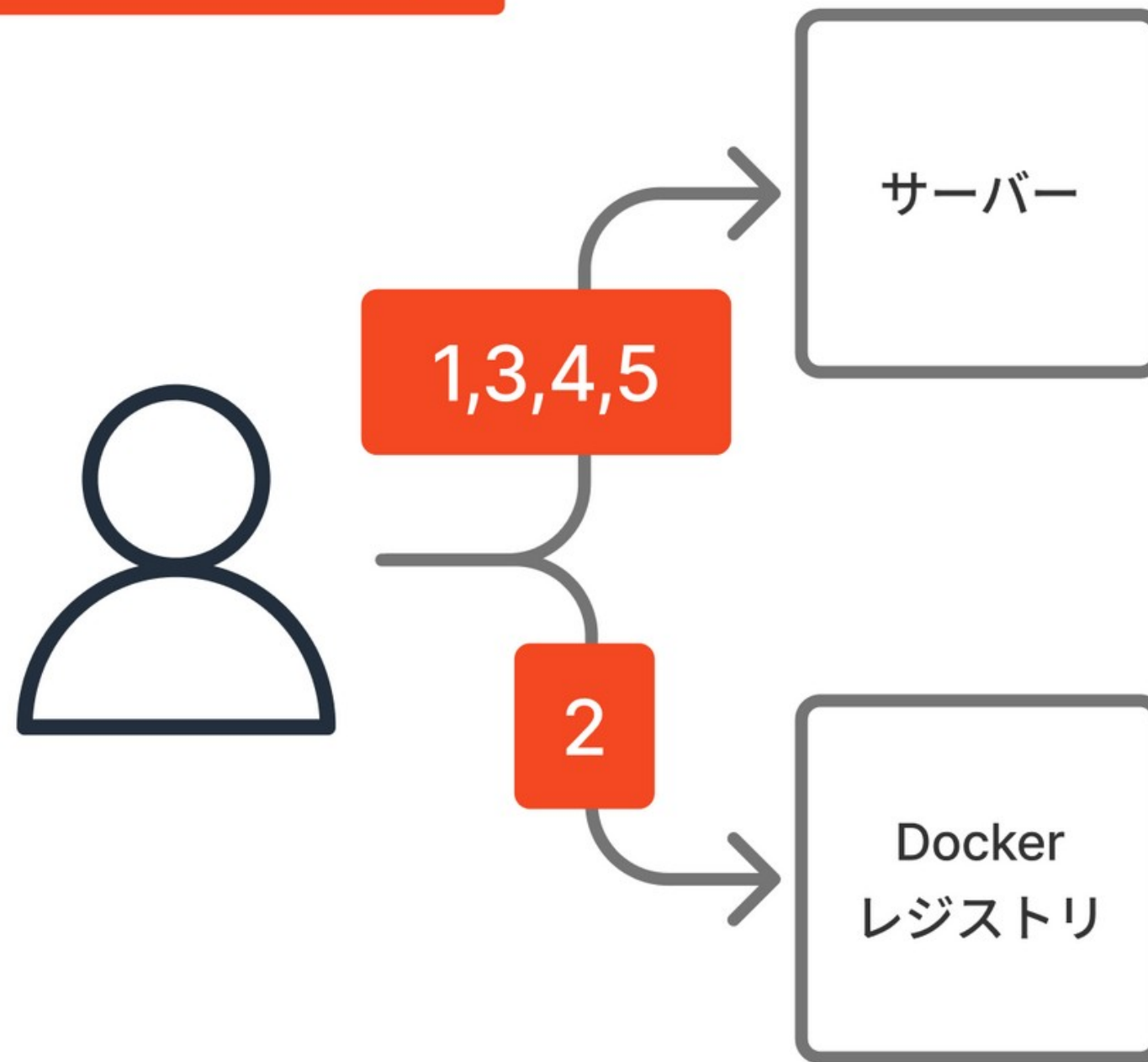
kamal setup

- Kamal でアプリを初めてデプロイする際に使用する
コマンド
- サーバーの初期設定からデプロイまでを一括で実行

kamal setup の処理フロー

1. サーバー準備 - SSH 接続して Docker をインストール
2. イメージ作成 - Dockerfile からビルドしてレジストリにプッシュ
3. デプロイ実行 - サーバーでイメージをプルして新コンテナを起動
4. トラフィック切替 - kamal-proxy が新コンテナの正常性を確認
後、トラフィックを切り替え
5. クリーンアップ - 旧コンテナを停止し、不要なイメージを削除

kamal setup



つまり

**サーバーと Docker を動かせる環境を
準備するだけでデプロイできる**

とてもシンプル！！

ちなみに ...

Kamal の由来

カマルは、船乗りが北極星で緯度を測定し航路を維持するために使用した古代アラブの航海用具に由来する名称



アラブ人が使った「カマール」



昔の人達は、GPS なしでどうやって現在位置を割り出していたのか | ログミー Business

というわけで本題です。

Kamal を使って 3 つの社内プロジェクトを 運用している話

まずは経緯をざっくりと

**自分がインフラ担当になって
Kamal を採用するまでの経緯**

とある日 ...

インフラ何も分からない
勉強しないとな～



上司

勉強も兼ねて若手で社内プロジェクト
を開発・運用することにしたから
インフラ担当を頼んだ！

了解です。頑張ります。
(マジっすか)



数日後 ...

(任されはしたけど)
インフラ何も分からない
勉強しないとな～



上司

社内プロジェクトをデプロイを
できるようにしておいて!
構成は任せるから

了解です。頑張ります。
(マジっすか)



heroku とか render とかの
お手軽デプロイしか知らないのに、
ど、どうしよう ...

そういや、Kamal ってのがあらし
ちょっと調べてみるか ...

手軽に AWS でデプロイできた！ 便利
これで試してみよう！

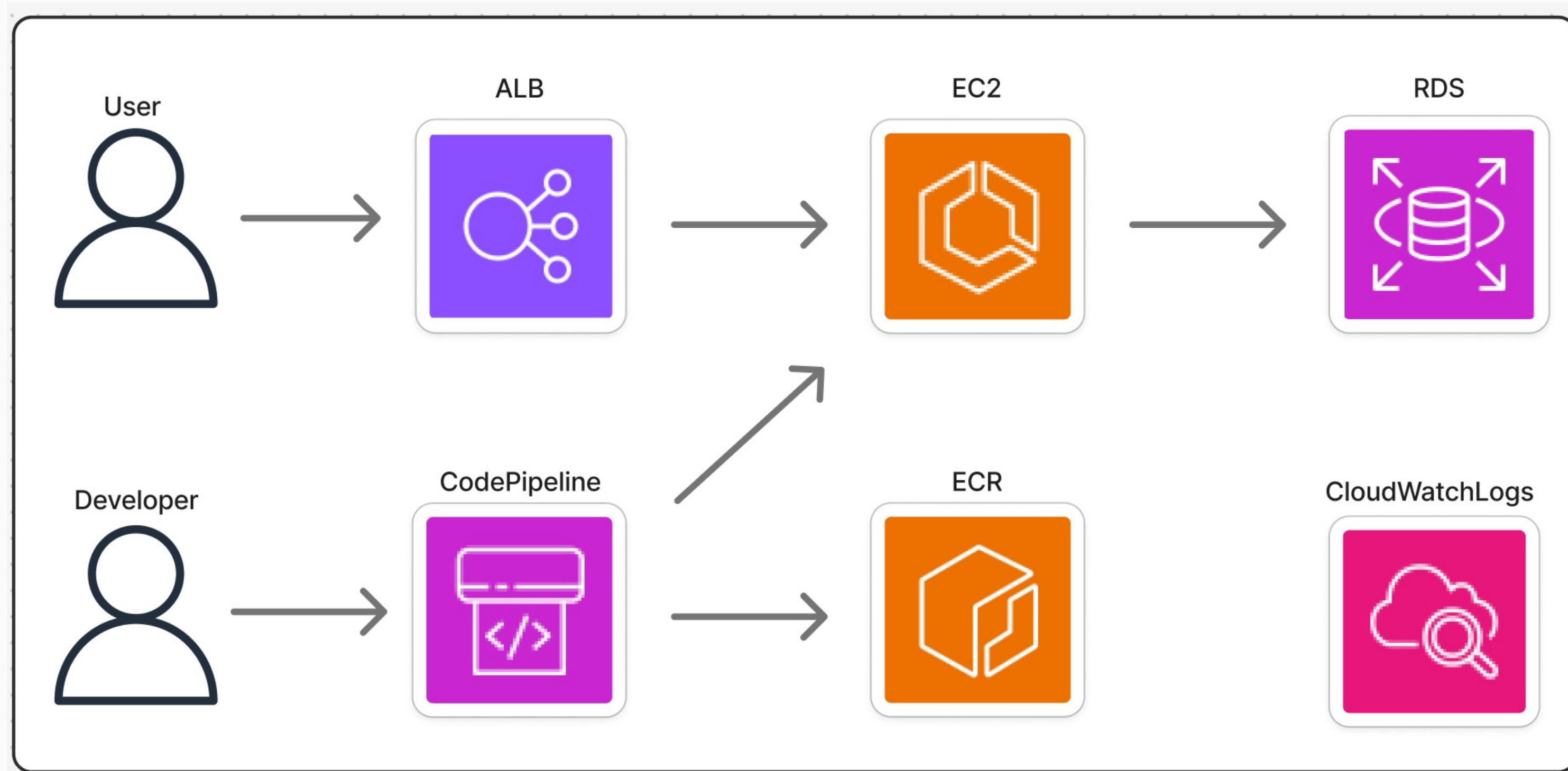


という経緯で Kamal を採用しました

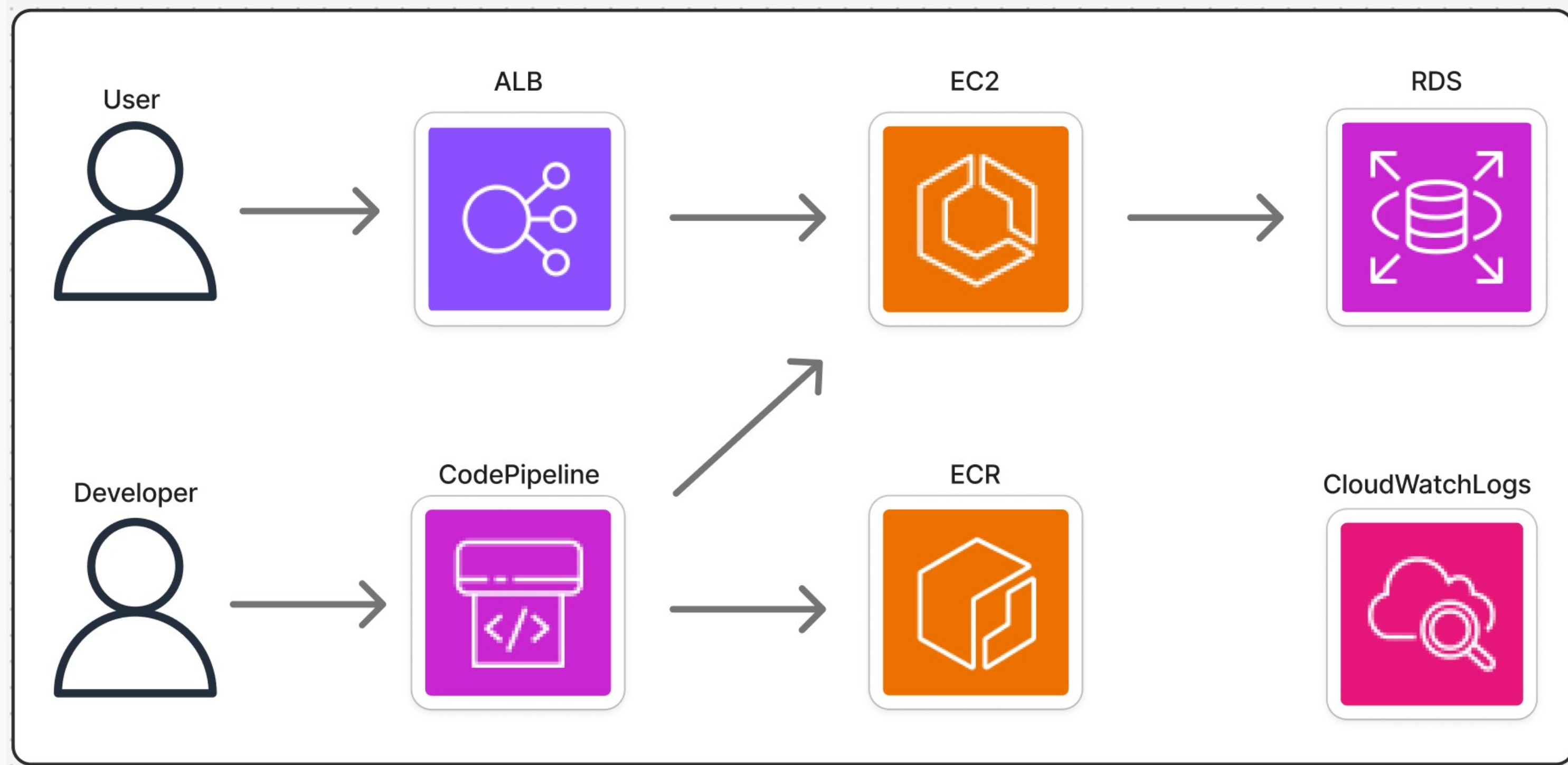
**AWS を使うことにしたのは
すぐに試せる環境が AWS だったから
です**

どんな構成でやってるのか

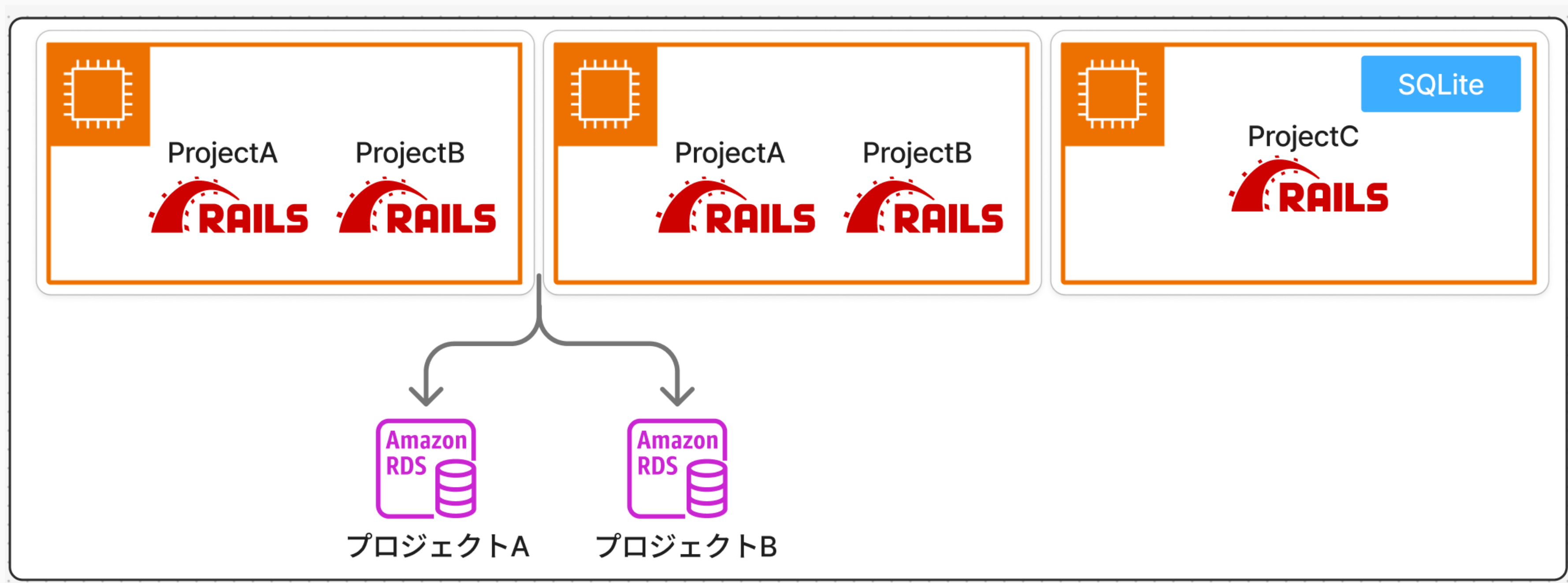
構成



構成



EC2



複数サーバーにデプロイする場合

deploy.yml

```
servers:  
  web:  
    hosts:  
      - 192.0.2.10  
      - 192.0.2.11
```

**Kamal で
良かったこと、困ったこと**

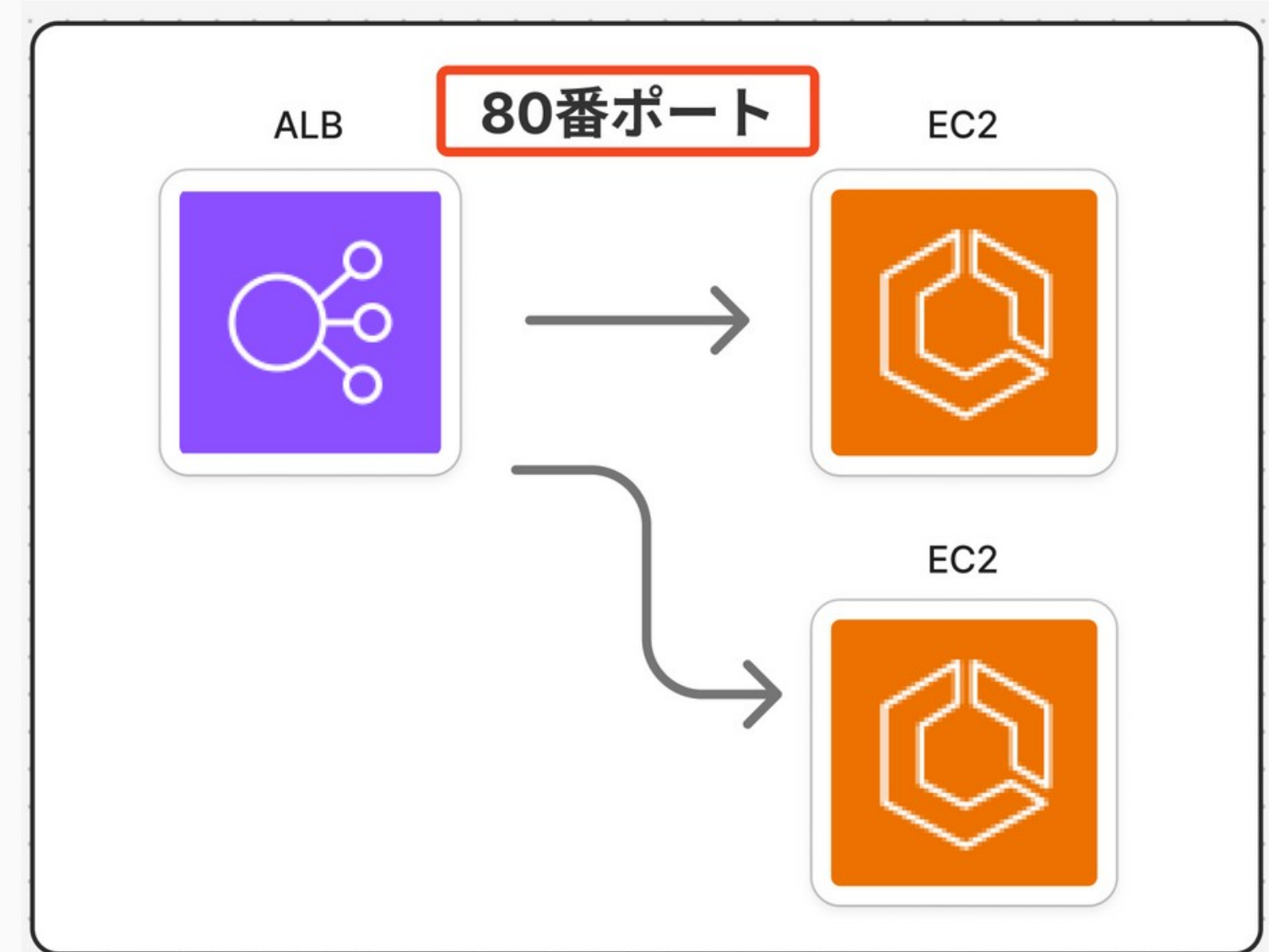
特に困ったこと 2 つ

- ALB ヘルスチェック問題
- 複数人開発でのデプロイ問題

ALB ヘルスチェック問題

ALB のヘルスチェック

- 各ターゲットに対しヘルスチェックを定期実行する
- 1つのインスタンスに1つのヘルスチェック
- プライベート IP が使われる



なんか失敗する ...

**Rails までリクエストが
来てない!**

なぜだ ...

Kamal Proxy が原因っぽい

Kamal Proxy とは

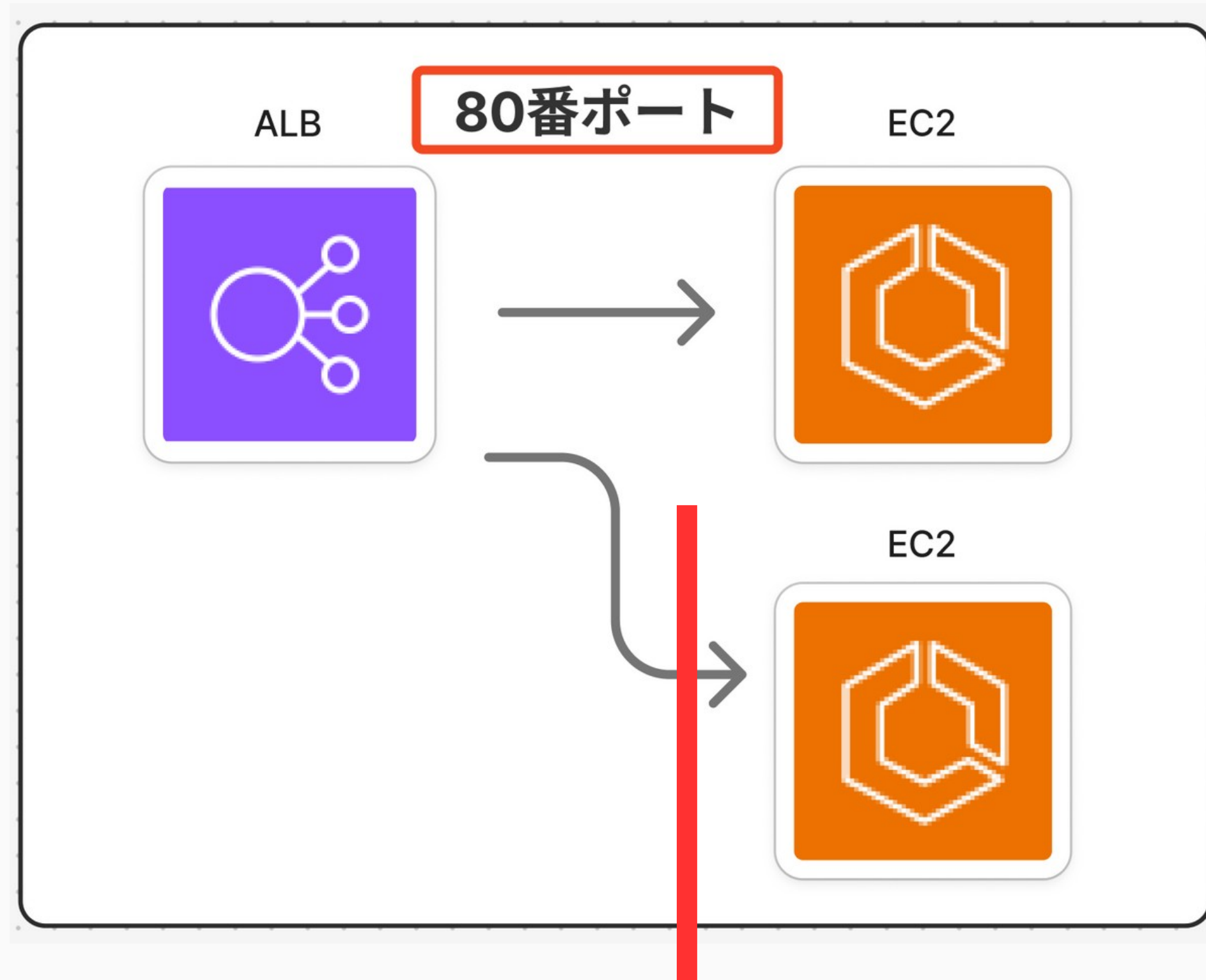
1. ゼロダウンタイムデプロイのための最小限の HTTP プロキシ
2. 外部からの HTTP リクエストを適切なコンテナにルーティング
3. 自動 SSL 機能
4. ヘルスチェック機能

Kamal Proxy とは

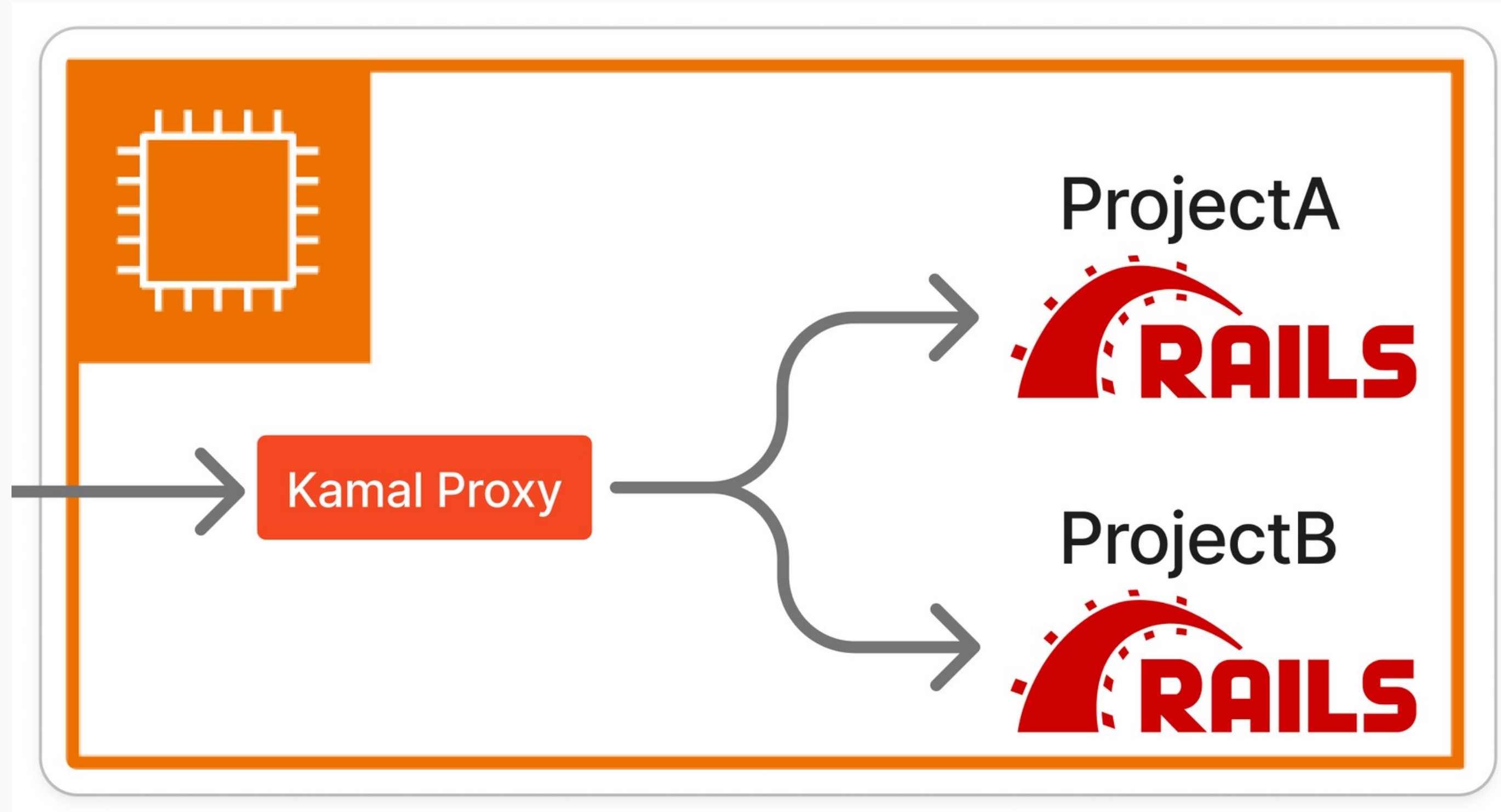
1. ゼロダウンタイムデプロイのための最小限の HTTP プロキシ
2. 外部からの HTTP リクエストを適切なコンテナにルーティング
3. 自動 SSL 機能
4. ヘルスチェック機能

詳しく調査

EC2 の中の Kamal Proxy を調査



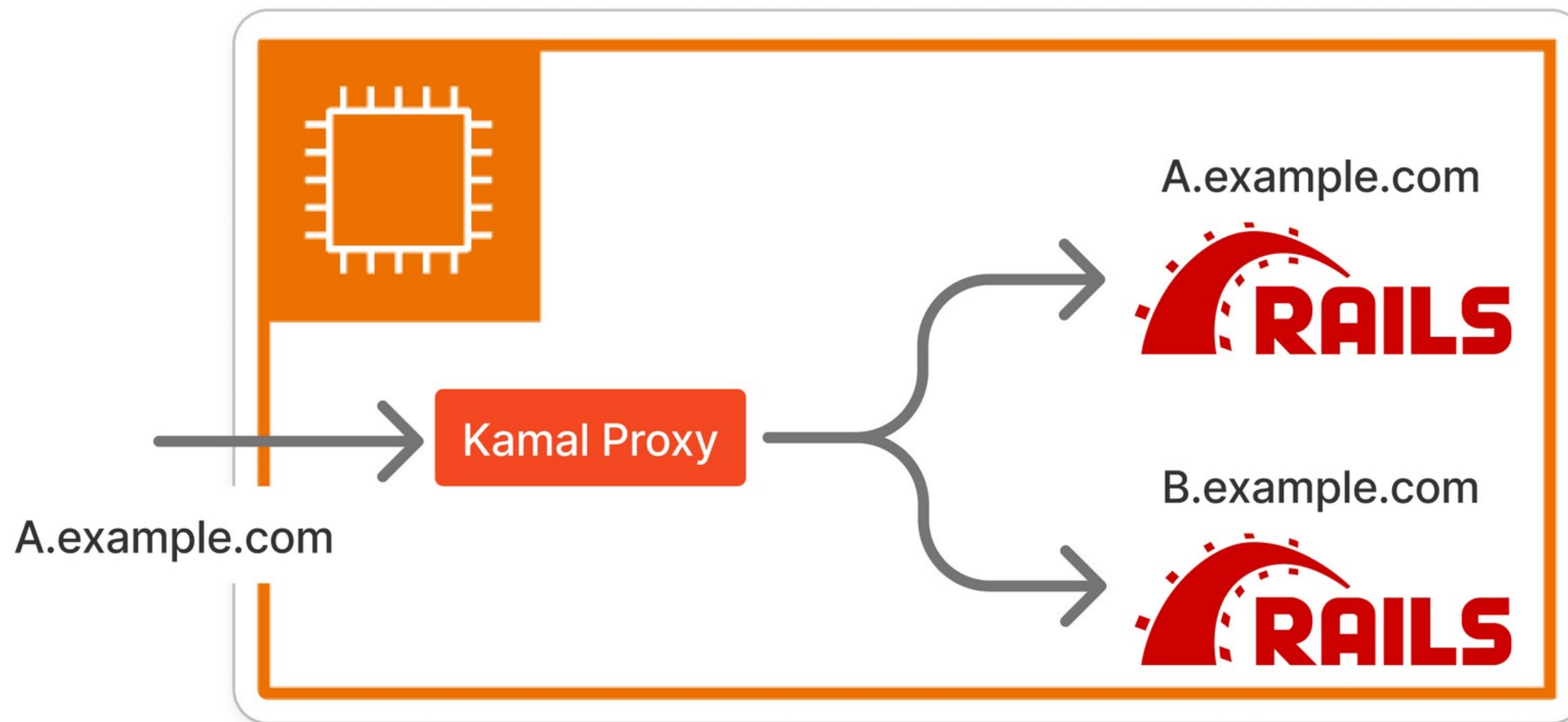
ここで止まってそう！



**Kamal Proxy は
どうやってリクエストを
各コンテナに振り分けているのか**

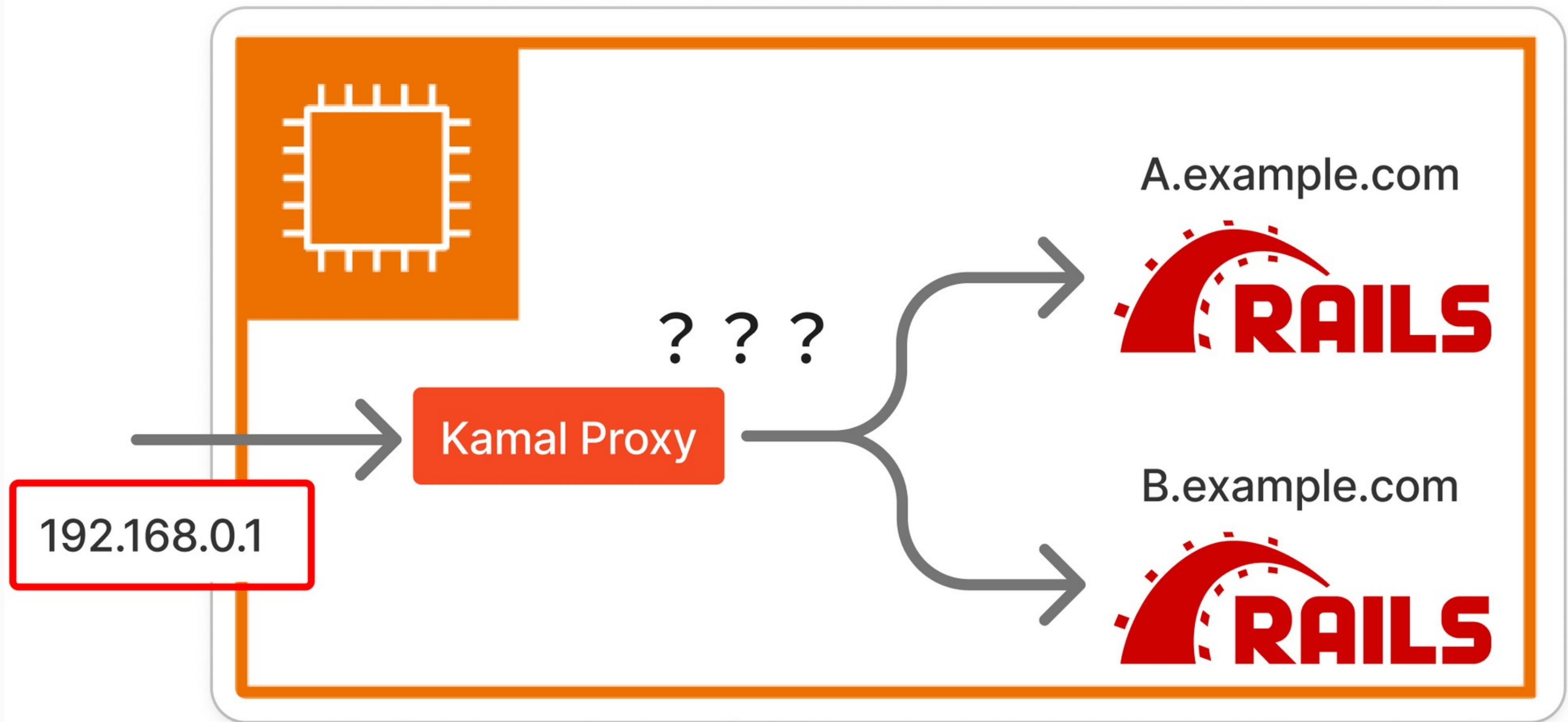
Kamal Proxy のリクエストの振り分け

- リクエストのホスト名を元にどのコンテナにリクエストを転送するか決めている



なるほど💡

**ALB のヘルスチェックは
プライベート IP なので、 Kamal が
リクエストを振り分けできない**



解決策

解決策

- 1つのアプリでプライベート IP を受け付けるように設定した!

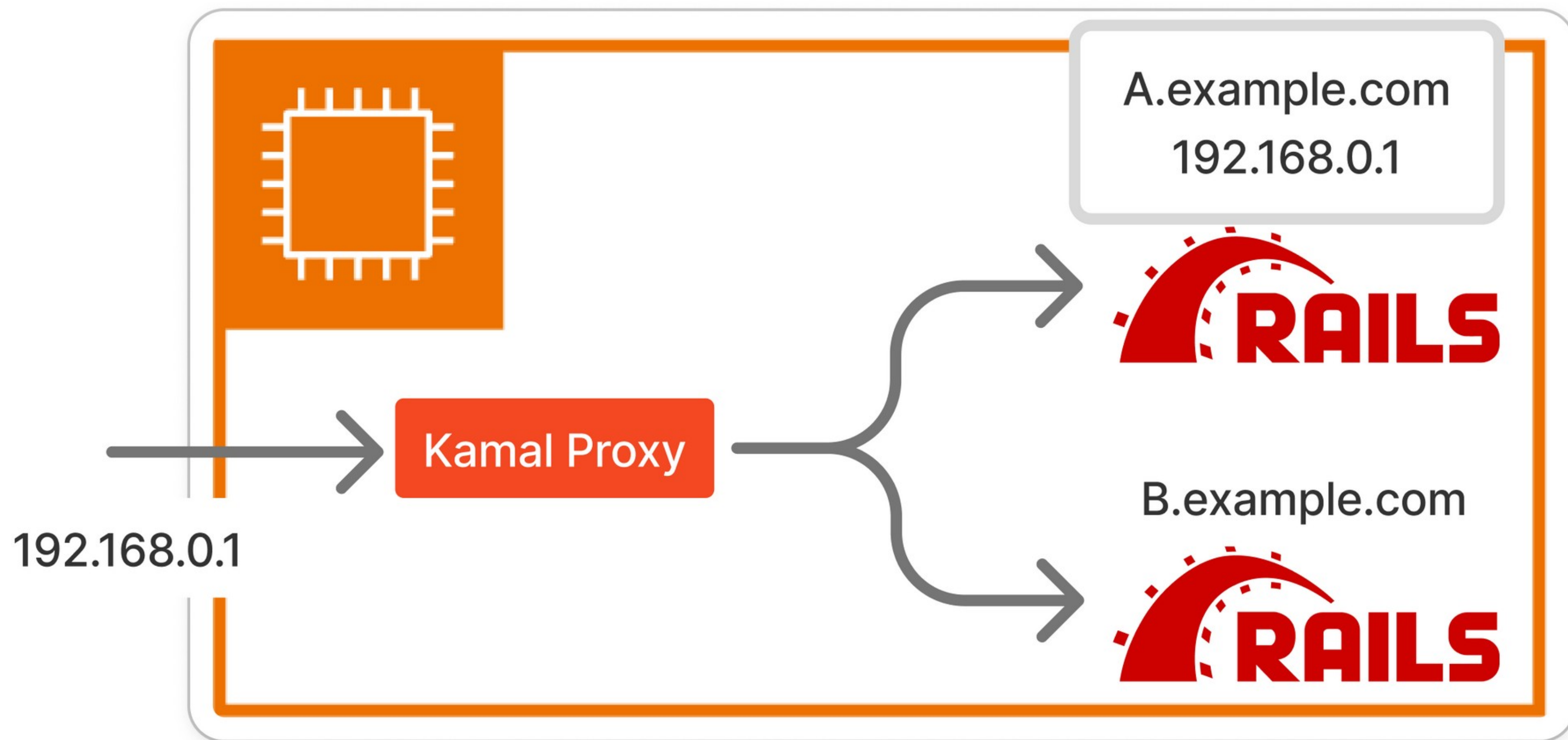
```
proxy:
```

```
  host:
```

```
    - A.example.com
```

```
    - 192.168.0.1
```





解決🎉

複数人開発でのデプロイ問題

**構築初期の頃は自動デプロイを
作っていなかった**

デプロイのたびに
連絡してもらっていた

開発
メンバ

デプロイお願いしますー

デプロイしました



開発
メンバ

不具合あったので修正しました。
もう一度お願いします。

デプロイしました～



めんどい!

なぜこうなったのか

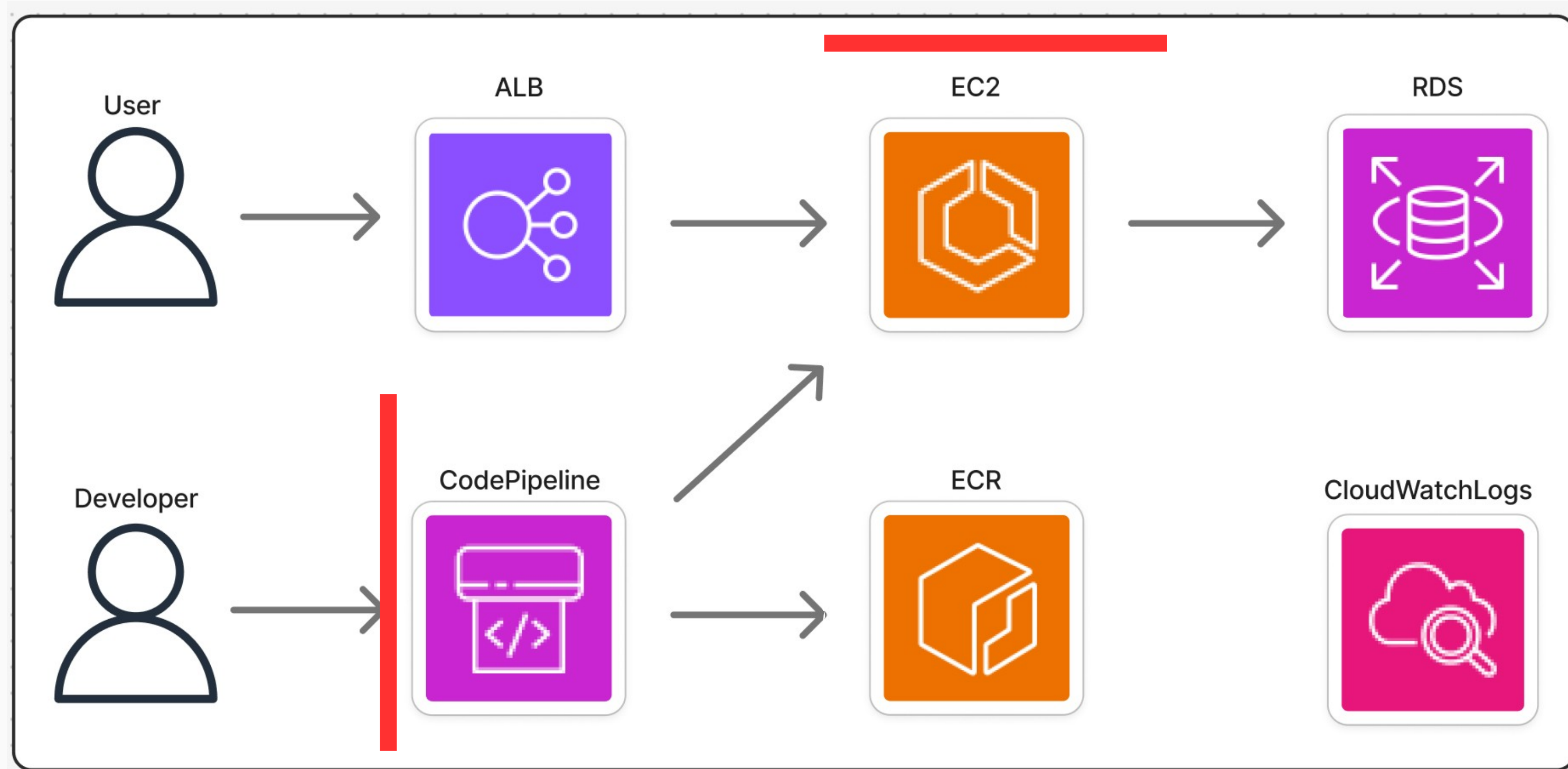
kamal でデプロイする時

- kamal deploy コマンド
- ローカル PC からサーバーに SSH 接続できる
 - 公開鍵、秘密鍵が必要
- ECR に docker イメージをプッシュする
 - ローカル PC から ECR にプッシュする権限が必要
- デプロイする人を増やすたびにキーを発行するか、共有する必要がある

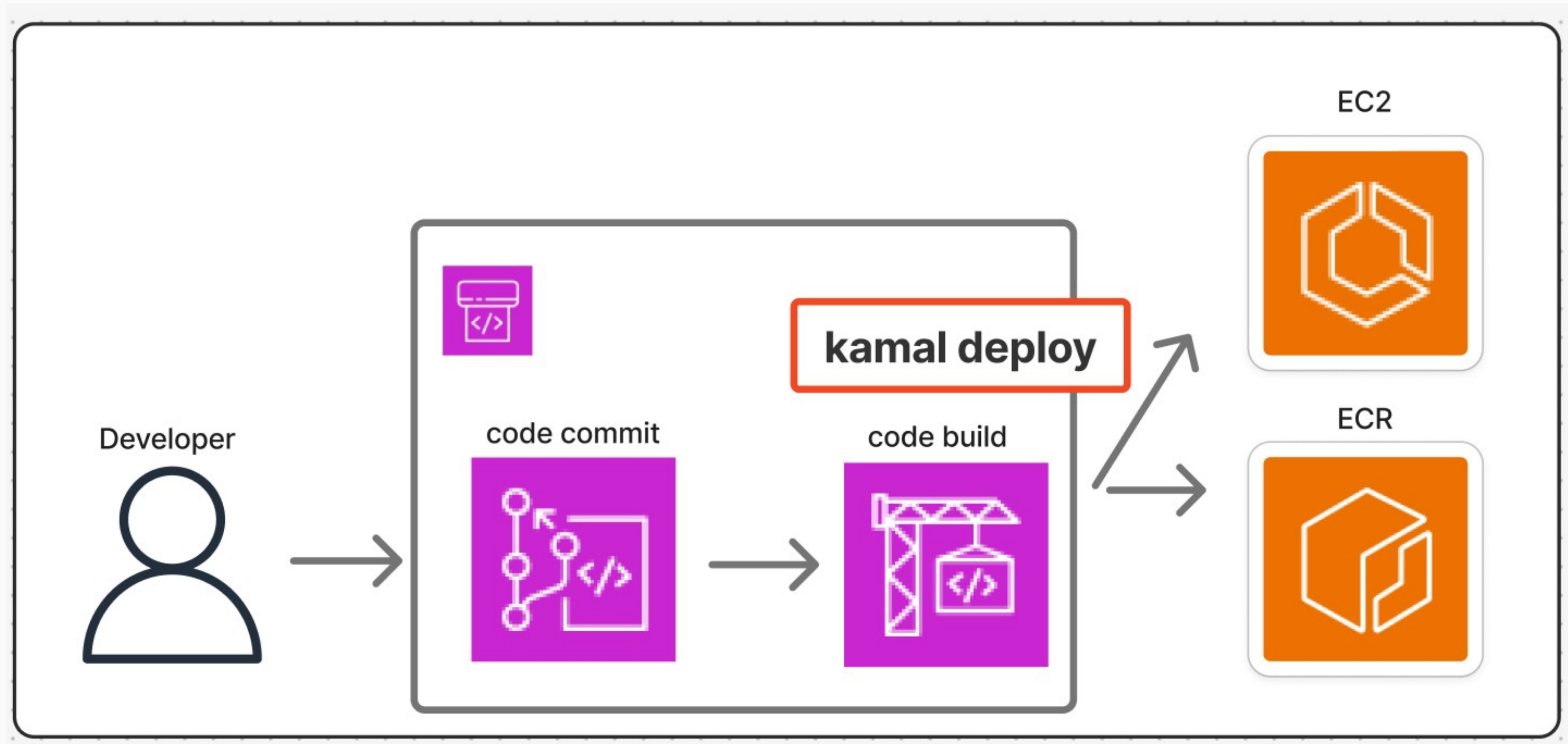
簡単に増やしたくない

ということで自動デプロイを
作ることにしました

構成



CodePipeline



解決🎉

困ったことを話しましたが

Kamal、いいところもある

自動 SSL 対応

- SSL を true にするだけ
 - Let's Encrypt + Go の autocert
 - サイトにアクセス => 証明書チェック => 期限切れを検出 → 新証明書取得開始

```
proxy:  
  ssl: true  
  host: app.example.com
```

データメンテ、デバッグが楽

- エイリアスが設定できてすぐにサーバーでコマンド実行できる
 - kamal console
 - kamal shell
 - kamal logs

```
aliases:
```

```
console: app exec --interactive --reuse "bin/rails console"
```

```
shell: app exec --interactive --reuse "bash"
```

```
logs: app logs -f
```

```
dbc: app exec --interactive --reuse "bin/rails dbconsole"
```

環境を分けてデプロイも簡単

- ファイルを分けたいだけ
 - `deploy.yml`
 - `deploy.staging.yml`
 - `deploy.production.yml`
- 環境ごとのデプロイ
 - `kamal deploy -d staging`
 - `kamal deploy -d production`

Docker のログ収集の設定も簡単

- logging に設定を書くだけ
- Amazon CloudWatch Logs の設定例

```
logging:  
  driver: awslogs  
  options:  
    awslogs-region: ap-northeast-1
```

他にもいろいろありますが、
設定がシンプルで簡単！

まとめ

- サーバーと Docker を動かせる環境を準備するだけでデプロイできる
- 設定がシンプルで簡単
- Kamal Proxy と ALB の組み合わせは若干苦戦する
- 複数人開発のデプロイには気をつけて
- それを踏まえてもメリットたくさん

以上です！

ありがとうございました！ 😊