

# Diff of the conference-app from 2023 to 2024

unasuke

東京Ruby会議12 前夜祭

2025-01-17

# 自己紹介

---

- Name: うなすけ
- Work: フリーランス
- Kaigi on Rails organizer
- GitHub <https://github.com/unasuke>
- Fediverse <https://mstdn.unasuke.com/@unasuke>
- X (Twitter) [https://twitter.com/yu\\_suke1994](https://twitter.com/yu_suke1994)
- <https://unasuke.com>



# さっきのワークショップで書いたコード

```
CANVAS_SIZE = 800
DRAW_COUNT = 0

class Circle
  def initialize
    @x = rand(0..CANVAS_SIZE)
    @y = rand(0..CANVAS_SIZE)
    loop do
      @speed = rand(-10.0..10.0)
      break if abs(@speed) > 2.0
    end
    @angle = rand(0..Math::PI)
    @timing = [10, 20, 30, 40, 50].shuffle.first
  end

  def next_position(count)
    if count % @timing == 0
      @angle = rand(0..(Math::PI*2))
    end

    @x = @x + cos(@angle) * @speed
    @y = @y + sin(@angle) * @speed
    if @x >= CANVAS_SIZE || @y >= CANVAS_SIZE
      @angle += Math::PI
    end
  end

  def draw(count)
    next_position(count)
    fill("#07786")
    ellipse(@x, @y, 50, 50)
  end
end

def setup
  createCanvas(CANVAS_SIZE, CANVAS_SIZE)
  background("#CAF8F8")

  # noLoop
  @circles = 100.times.map { Circle.new }
  @draw_count = 0
end

def draw
  noStroke
  grid_size = 30
  background("#CAF8F8")
  @circles.each { |c| c.draw(@draw_count) }
  @draw_count += 1
end
```

東京Ruby会議12では「Rubyと暮らす」発表をけっこう幅広く募集しています!!!!!!!!!!!!!!(いました)

---

東京Ruby会議12では

**「Rubyと暮らす」**

発表を幅広く募集しています!!!!!!!!!!!!!!



Tokyo  
RubyKaigi 12

<https://note.com/tokyorubykaigi12/n/n14803ce700dd>

---

「Rubyと暮らす」

東京Ruby会議12がめざす方向性をあえて一言で示すならば、こうなるでしょうか。Rubyを日々書いて暮らす、あなたの話を聞きたいです！ 小粒なものでも、大掛かりなものでも、どちらも歓迎です。

<https://note.com/tokyorubykaigi12/n/n14803ce700dd> より

## 「暮らし」

---

- カンファレンス(コミュニティ)は暮らしの一部
  - ですよね？

# 「暮らし」

---

- カンファレンス(コミュニティ)は暮らしの一部
  - ですよね？
  - 特に運営メンバーともなると、暮らしですよね？

## Kaigi on Railsは暮らし、そういうことにします

---

というわけで暮らしの話をしてします



# Kaigi on Rails 2024

---

- 参加していただけましたか？
- 楽しんでもらえましたか？

## Kaigi on Rails conference-app

---

- <https://app.kaigionrails.org>
  - <https://github.com/kaigionrails/conference-app>

それはとあるafter eventでの出来事であった

---

「へ～、そんなアプリがあったんですね」

## 2025年の抱負

---

2025年はもっとconference-appを宣伝する年にします

## Diff of the conference-app from 2023 to 2024

---

1. DB移行
2. Redis移行
3. Heroku脱出
4. Rails 8移行
5. Solid Queue移行

# 2024、一番やりたかったこと

## Heroku脱出！

 kaigionrails 

 NEW POST

 HOME

 POSTS

 TAGS

 SETTINGS

#248 rfc /

### 【完了】 HerokuからAWSに移行したい

 Star 0 |  Unwatch 2  |  Comments 2 

 Created by [unasuke](#)  
2022-10-25 14:22 | 2年前

 Updated by [unasuke](#)  
2024-06-16 03:18 | 7ヶ月前 [± 移行した！](#)

☰ 目次 ▾

### なぜやるのか

現在、cfp-appとsponsor-appとconference-appはHeroku上に運用されています。しかしHerokuには次のような課題があり、別のPaaSに移行したほうが運営としてより良いと判断できる状況になってきています。

# Why 脱Heroku?

---

- ログインセッションが短い
- インスタンスがUSにある
- HTTP/1.1しかサポートされていない
  - ※ 2024-11-21にHTTP/2がGAになりました
  - <https://devcenter.heroku.com/changelog-items/3066>

## どうすれば脱Herokuできるのか？

---

- 小さいマネージドのPostgreSQLがあって
- 小さいマネージドのRedisがあって
- アクセスのない時はインスタンスを止めてくれて

そんなPaaS、IaaSがあればなあ……



## 移行先のPaaS候補

---

- Render.com
  - <https://render.com>
- Fly.io
  - <https://fly.io>

## 結局AWSに移行しました

---

- AWS App Runnerがある
  - Google CloudでいうCloud Run的な
- RubyKaigi側で使っているAWSアカウントを親にできる
  - (thanks sorah)
- GitHubアカウントでOIDCできる (thanks sorah)
  - <https://speakerdeck.com/sorah/serverless-idp-for-small-team-himari>
- 実はRubyKaigiのsponsor-appがApp Runnerに移行している

## AWS移行、DBとRedisはどうするの？

---

- ElastiCache Serverlessはあるけど……
- Aurora Serverlessは最小を0にできないし(最小は0.5 ACU)
  - (後述)

1年で1週間くらいしか稼動しないのにずっとリソース確保しっぱなしはしたくない！

# Serverless Postgres

---

- Neon
  - <https://neon.tech>
- Nile
  - <https://www.thenile.dev>

これならDBにアクセスしている時間のみの課金にできる！

# ヨッシャ移行するぞ

---

2024年6月4日 ▾



**unasuke** 21:37

さて、やるか

2024年6月8日 ▾



**unasuke** 22:58

deployどうすっかな.....



**unasuke** 23:04

う～ん！

めんどくせ！

めんどくせ～



**unasuke** 23:44

っていうかまずcfp-appのDocker化やんなきゃじゃん！！！！！！

アオ～ン

2024年6月9日 ▾



# ヨッシャ

```
aws/conference-app/apprunner.tf  +117 ●●●●● ●●●●● ●●●●● ...
... @@ -0,0 +1,117 @@
1 + resource "aws_apprunner_service" "conference_app" {
2 +   provider      = aws.usw2
3 +   service_name = "conference-app"
4 +
5 +   source_configuration {
6 +     image_repository {
7 +       image_configuration {
8 +         port = "3000"
9 +
10 +         # See also kaigionrails/conference-app/deploy/task_definition.jsonnet
11 +         runtime_environment_variables = {
12 +           APPLICATION_URL      = "https://app.kaigionrails.org"
13 +           GITHUB_OAUTH_REDIRECT_URI = "https://app.kaigionrails.org/auth/github/callback"
14 +           LANG                  = "en_US.UTF-8"
15 +           RACK_ENV              = "production"
16 +           RAILS_ENV            = "production"
17 +           RAILS_LOG_TO_STDOUT  = "enabled"
18 +           RAILS_SERVE_STATIC_FILES = "enabled"
19 +           SCOUT_NAME           = "conference-app"
20 +           SENTRY_ENV           = "production"
21 +           VAPID_SUBJECT_MAILTO  = "mailto:info@kaigionrails.org"
22 +         }
23 +       }
24 +     }
25 +   }
26 + }
```

<https://github.com/kaigionrails/terraform/pull/4>

## Diff of the conference-app from 2023 to 2024

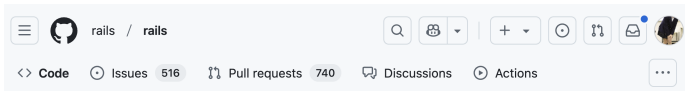
---

1. DB移行
2. Redis移行
3. Heroku脱出 ← ココまで終わり
4. Rails 8移行
5. Solid Queue移行



# Rails 8移行

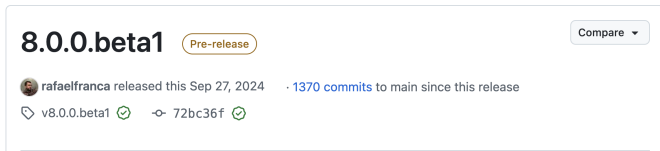
## Rails 8 beta1 2024-09-27 released




rails / rails





<> Code Issues 516 Pull requests 740 Discussions Actions

[Releases](#) / v8.0.0.beta1



**8.0.0.beta1** Pre-release Compare

 rafaelfranca released this Sep 27, 2024 · [1370 commits](#) to main since this release

 v8.0.0.beta1   72bc36f 

ところでKaigi on Rails 2024は10/25

## Rails 8 upgrade

---

conference-appの場合、以下の順でupdateをした

1. v7.1.3.4

2. v7.2.1

3. v8.0.0.beta1

4. v8.0.0.rc1 (Kaigi on Rails 2024開催中はここ)

5. v8.0.0.rc2 (10/30 released)

6. v8.0.0 (11/07 released)

# Is rails 8 upgrade hard?

[https://guides.rubyonrails.org/upgrading\\_ruby\\_on\\_rails.html#upgrading-from-rails-7-2-to-rails-8-0](https://guides.rubyonrails.org/upgrading_ruby_on_rails.html#upgrading-from-rails-7-2-to-rails-8-0)



Guides

Version: 8.0



Guides Index

Contribute

## Ruby on Rails 8.0 Release Notes

Highlights in Rails 8.0:

### 1 Upgrading to Rails 8.0

If you're upgrading an existing application, it's a great idea to have good test coverage before going in. You should also first upgrade to Rails 7.2 in case you haven't and make sure your application still runs as expected before attempting an update to Rails 8.0. A list of things to watch out for when upgrading is available in the [Upgrading Ruby on Rails](#) guide.



### Chapters

1. Upgrading to Rails 8.0
2. Major Features
3. Raities
  - Removals
  - Deprecations
  - Notable changes
4. Action Cable
  - Removals
  - Deprecations
  - Notable changes
5. Action Pack
  - Removals

全然長くない！

# Diff of the conference-app from 2023 to 2024

---

1. DB移行
2. Redis移行
3. Heroku脱出
4. Rails 8移行 ← ココまで話した
5. Solid Queue移行

# Solid Queue移行

Navigation bar for the GitHub repository `kaigionrails / conference-app`. It includes a search icon, repository icons, and navigation tabs for `Code`, `Issues` (2), `Pull requests` (highlighted), `Actions`, `Projects`, `Wiki`, and `Security`.

## Solid queue #283

Edit

<> Code

Merged

unasuke merged 12 commits into `kaigionrails:main` from `unasuke:solid-queue` on Oct 6, 2024

Conversation 0

Commits 12

Checks 3

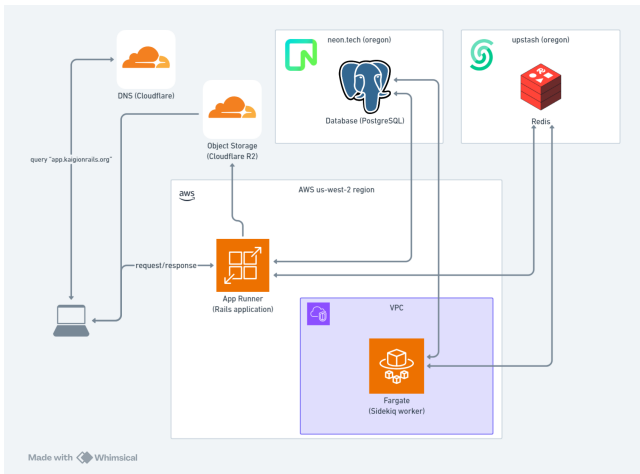
Files changed 31

+3,381 -98

やるだけ

(デカいdiffの大半はmigrationとrbsです)

# 現在の構成



# Why 脱Heroku?

---

- ログインセッションが短い
- インスタンスがUSにある
- HTTP/1.1しかサポートされていない
  - ※ 2024-11-21にHTTP/2がGAになりました
  - <https://devcenter.heroku.com/changelog-items/3066>

## 結局オレゴン(アメリカ)やん！

---

- はい……



## 2024-11-20 Amazon Aurora Serverless v2 update

---

Amazon Aurora Serverless v2 がゼロキャパシティへのスケーリングをサポート

Amazon Aurora Serverless v2 では、0 個の Aurora 容量ユニット (ACU) へのスケーリングがサポートされるようになりました。このリリースにより、データベース接続に基づいて非アクティブ状態が一定期間続くと、データベースを自動的に一時停止できます。最初の接続が要求されると、データベースは自動的に再開し、アプリケーションの要求に合わせてスケールします。Aurora Serverless v2 は ACU 単位で容量を計算します。1 つの ACU は、約 2 ギビバイト (GiB) のメモリ、対応する CPU、およびネットワークから構成されます。容量範囲を指定すると、データベースはアプリケーションのニーズに合わせてこの範囲内でスケーリングされます。

<https://aws.amazon.com/jp/about-aws/whats-new/2024/11/amazon-aurora-serverless-v2-scaling-zero-capacity/>

つまり……？

---

全てがap-northeast-1になる時代

## 要は？

---

- Kaigi on Rails 2025でもconference-appを使う予定です！
- ポートフォリオ置き場にAWSという選択はいかが？
  - 無理があるのではないか？

# たすきを渡す

---



# たすきを渡す

## Signage #323

[Edit](#)[Code](#)[Merged](#)unasuke merged 20 commits into [kaigionrails:main](#) from [unasuke:signage](#) on Oct 25, 2024[Conversation](#) 0[Commits](#) 20[Checks](#) 3[Files changed](#) 71

+2,178 -0

[Changes from all commits](#) [File filter](#) [Conversations](#) [Settings](#)

0 / 71 files viewed

[Ask Copilot](#)[Review in codespace](#)[Review changes](#)[Gemfile](#)[Gemfile.lock](#)[app](#)[controllers](#)[admin](#)[signage\\_device\\_ass...](#)[signage\\_devices\\_co...](#)[signage\\_pages\\_con...](#)2 [Gemfile](#)[Code](#)[File](#) Viewed[Comments](#) [More](#)

```
↑ ... @@ -80,6 +80,8 @@ gem "web-push"
80 80 gem "solid_queue"
81 81 gem "mission_control-jobs"
82 82
83 + gem "alba"
84 +
83 85 group :development, :test do
84 86   # See
   https://guides.rubyonrails.org/debugging_rails_application
   s.html#debugging-with-the-debug-gem
85 87   gem "debug", platforms: %i[mri mingw x64_mingw]
```

<https://github.com/kaigionrails/conference-app/pull/323>

# たすきを渡す

18:15



## Alba: Why, How and What's So Interesting

おおくらまさふみ



@okuramasafumi



@okuramasafumi