WEB DAY

27 MARZO 2024 **7A EDIZIONE**

BREAKING FREE FROM MONOLITHS: REVOLUTIONIZING DEVELOPMENT WITH LIVEWIRE AND SERVERLESS

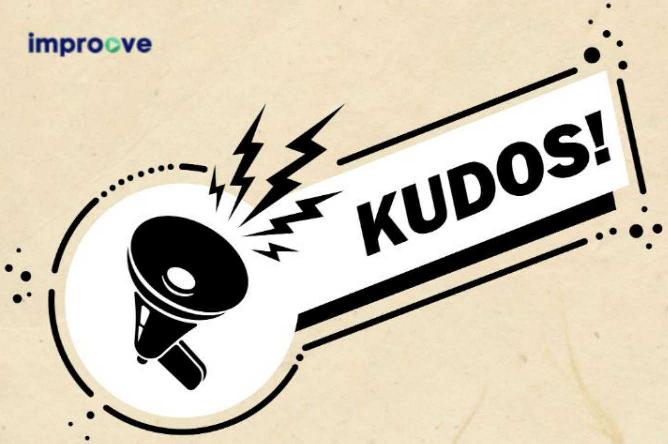
SIMONE D'AMICO

TECH LEAD

SPREAKER

improove

improove



WEB DAY

27 MARZO 2024 **7A EDIZIONE**

.NET on aws

software one

adesso.it

Spreaker From iHeart

UNIKEY TECH INSIDE

Agenda

What?

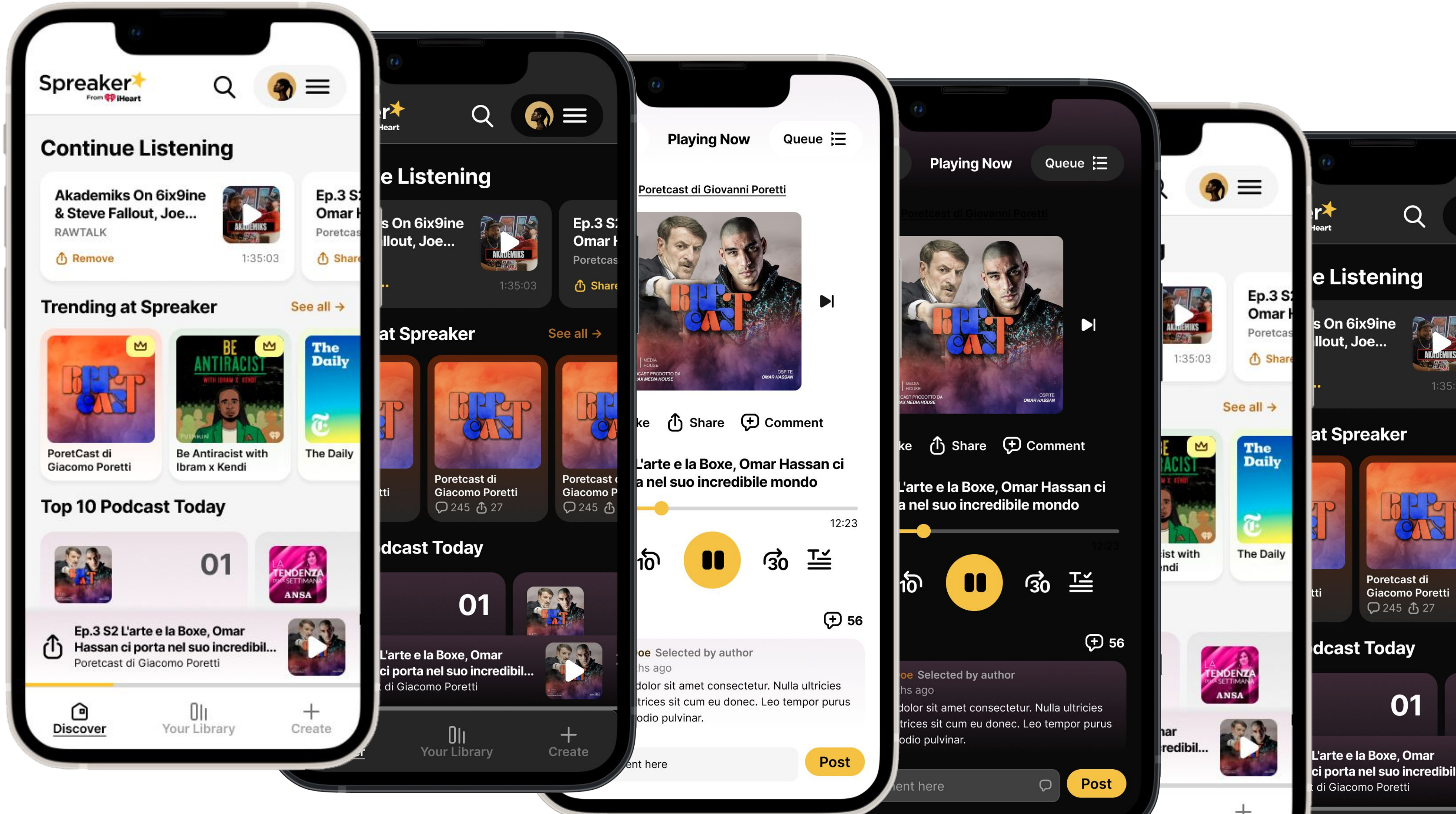
We leveraged the powerful combination of Laravel, Livewire, and Serverless to build a **modern web-application in PHP** currently named Spreaker Next

Why?

The need to **push the boundaries of our web listening and interaction experience**, that was stuck on a 15 years old monolith

How?

The process, technical choices, challenges we faced, and the results achieved.


Spreaker
From iHeart
Continue Listening
Akademiks On 6ix9ine & Steve Fallout, Joe...
RAWTALK
Remove
1:35:03
Trending at Spreaker
See all →
PoretCast di Giacomo Poretti
Be Antiracist with Ibram x Kendi
The Daily
Top 10 Podcast Today
01
Ep.3 S2 L'arte e la Boxe, Omar Hassan ci porta nel suo incredibil...
Poretcast di Giacomo Poretti
Discover
Your Library
Create
Playing Now
Queue
Poretcast di Giovanni Poretti
Share
Comment
L'arte e la Boxe, Omar Hassan ci porta nel suo incredibile mondo
12:23
56
Selected by author
Post
dolor sit amet consectetur. Nulla ultricies trices sit cum eu donec. Leo tempor purus odio pulvinar.

WHO I AM

Simone D’Amico

Technical Leader, Platform Team at Spreaker

Full-time dad of two, Lego addict and occasional speaker

@dymissy

https://www.linkedin.com/in/simonedamico/

simone.damico@spreaker.com

Spreaker From iHeart

Spreaker is a podcast platform that offers a comprehensive solution for podcast enthusiasts.

Spreaker is an all-in-one podcast platform for:

- Hosting
- Distributing
- Monetize
- Discover and Listen to podcasts

We serve 100M+ requests per day on spreaker.com only

Technologies adopted
AWS Services

- **AWS Cloudfront:** Content Delivery Network
- **AWS API Gateway**: Create, manage, and monitor scalable, secure APIs
- **AWS Lambda**: Run serverless code in response to events

- In addition to them, we used **Serverless Framework** to manage and deploy the application infrastructure

AWS Services

Bref

Octane

Laravel Livewire

Technologies adopted
Bref

Bref is a framework to write, deploy, and run serverless PHP applications on AWS Lambda.

AWS Services

Bref

Octane

Laravel Livewire

Technologies adopted
Laravel Octane

Laravel Octane: High-performance, concurrent request handling for Laravel applications

AWS Services

Bref

Octane

Laravel Livewire

Technologies adopted
Laravel + Livewire

- **Laravel**: PHP framework
- **Livewire**: Full-stack framework for Laravel, enabling reactive, dynamic interfaces without JavaScript

AWS Services

Bref

Octane

Laravel Livewire

The challenge

THE CHALLENGE

A new product direction

🙉
Encourage **listeners engagement and interaction**

⛓️
Decoupling business logic from **listening experience**

📲
Appealing UI and mobile-first approach

💪
Build a **strong and scalable** frontend foundation

THE CHALLENGE

Technical Constraints

🗿
Over a **decade-old monolith** hindering product evolution

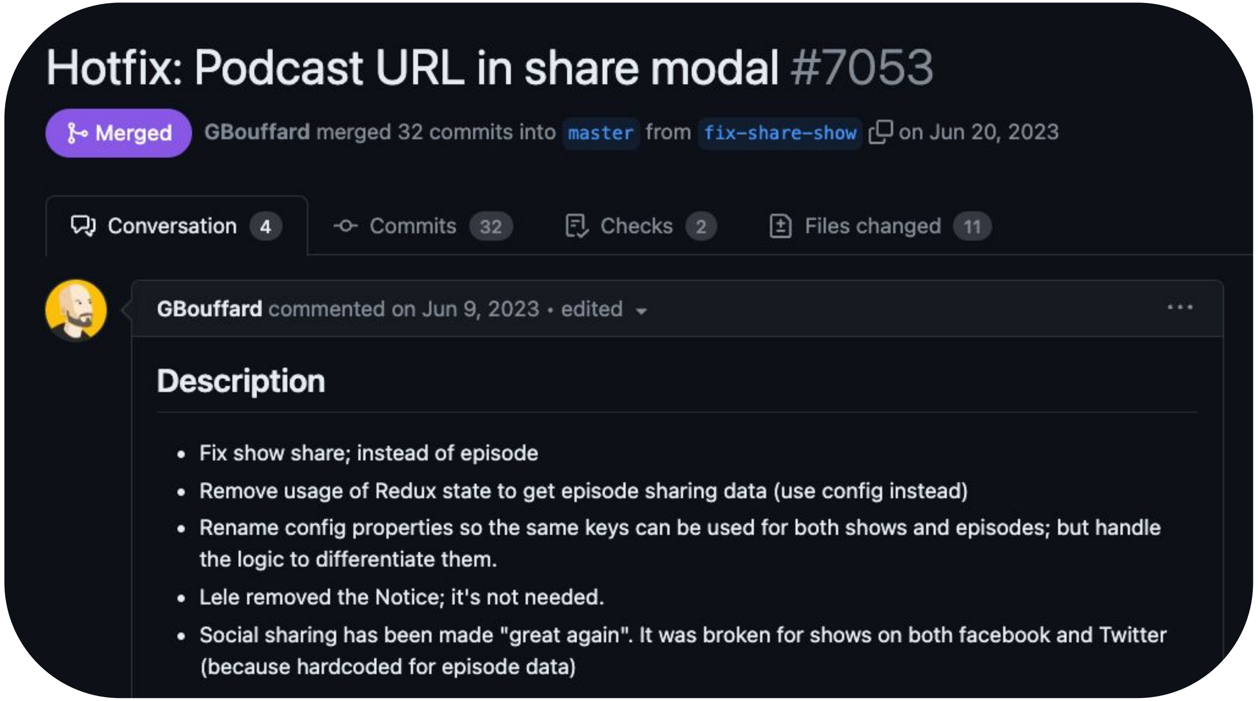
⏯️
Persistent mini-player across pages navigation

👨‍💻
Team primarily skilled in **PHP**

🔍
Product with heavily indexed contents

THE CHALLENGE

🗿Monolith hard to evolve



Hotfix: Podcast URL in share modal #7053

Merged GBouffard merged 32 commits into `master` from `fix-share-show` on Jun 20, 2023

Conversation 4 Commits 32 Checks 2 Files changed 11

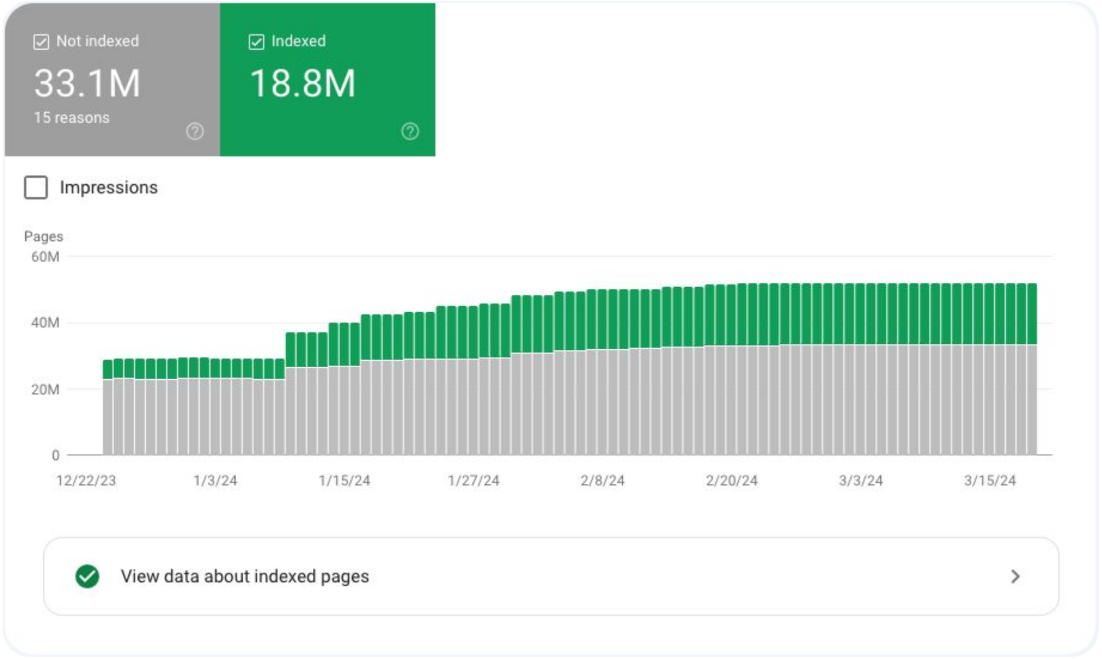
GBouffard commented on Jun 9, 2023 • edited

Description

- Fix show share; instead of episode
- Remove usage of Redux state to get episode sharing data (use config instead)
- Rename config properties so the same keys can be used for both shows and episodes; but handle the logic to differentiate them.
- Lele removed the Notice; it's not needed.
- Social sharing has been made "great again". It was broken for shows on both facebook and Twitter (because hardcoded for episode data)

THE CHALLENGE

🔍 Heavily indexed contents



THE CHALLENGE

▼ Functional Requirements

📣 **Makes all the full-stack engineers comfortable in working E2E**

🎨 Design

New design centered on listeners and their interactivity

Strong component model

Styleguide

🚀 Deployment and Infrastructure

Serverless based

Observability

Deployed through Jenkins

⚙️ Tech Stack

One main language (no duplication of state/templates)

💡 Flexibility, Maintenance and Evolution

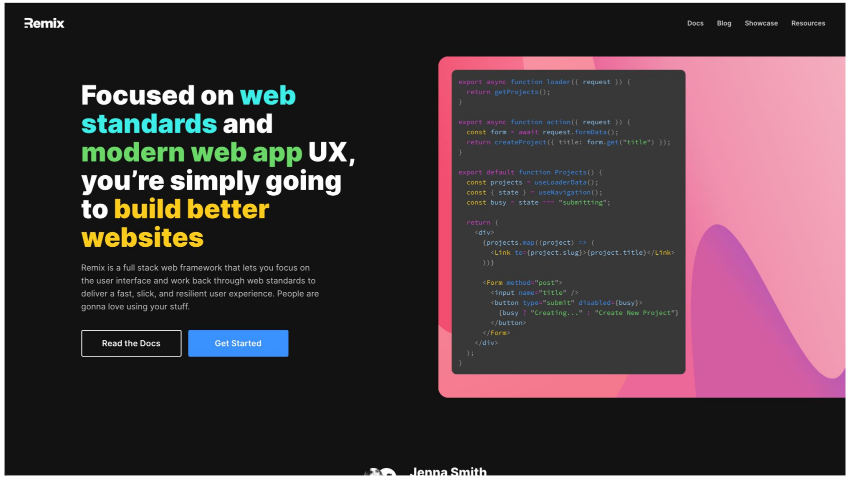
Pages / Features easy to add

The solution

... and how we got there

THE SOLUTION

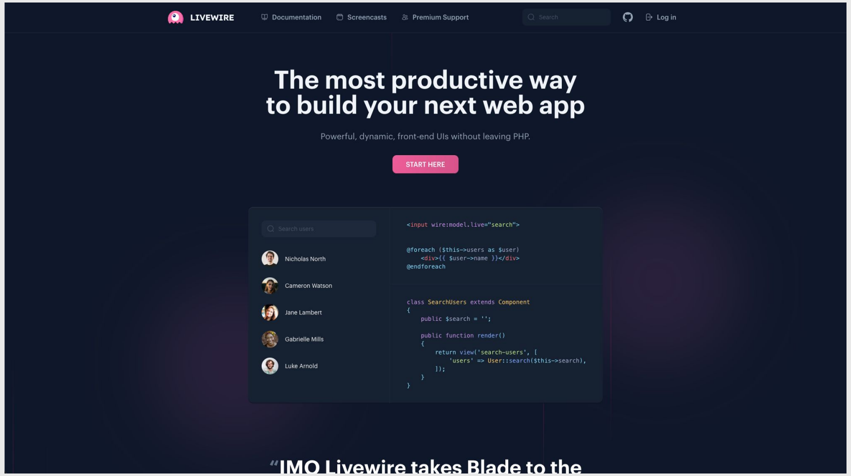
Proof of Concepts



Focused on web standards and modern web app UX, you're simply going to build better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web standards to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

```
export async function loader({ request }) {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({ title: form.get("title") });
}

export default function Projects() {
  const projects = useLoaderData();
  const { state } = useNavigation();
  const busy = state === "submitting";

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug}>{project.title}</Link>
      ))}

      <Form method="post">
        <input name="title" />
        <button type="submit" disabled={busy}>
          {busy ? "Creating..." : "Create New Project"}
        </button>
      </Form>
    </div>
  );
}
```



The most productive way to build your next web app

Powerful, dynamic, front-end UIs without leaving PHP.

```
<input wire:model.live="search">

@foreach ($this->users as $user)
    <div>{{ $user->name }}</div>
@endforeach
```

```
class SearchUsers extends Component
{
    public $search = '';

    public function render()
    {
        return view('search-users', [
            'users' => User::search($this->search),
        ]);
    }
}
```

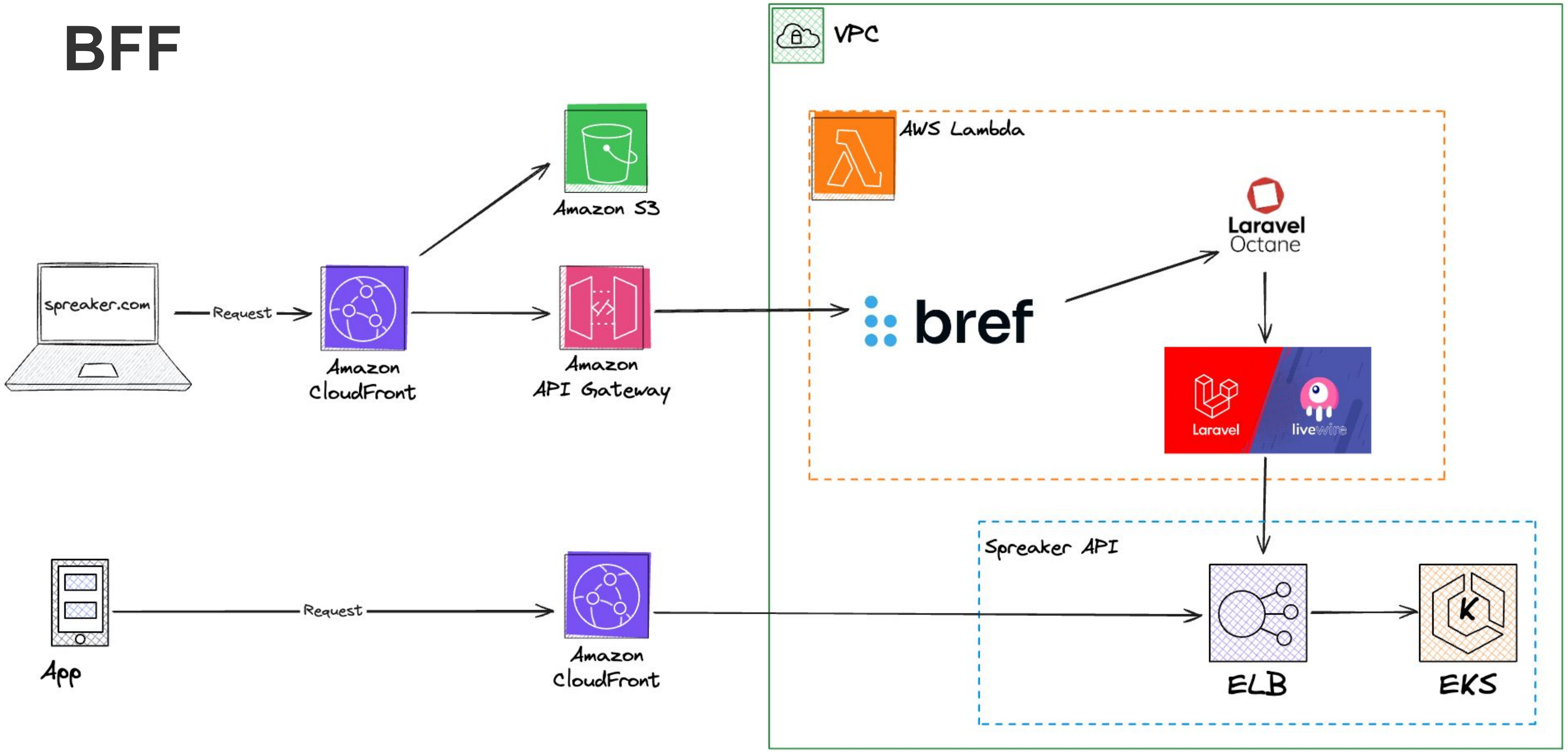
"IMO Livewire takes Blade to the

PROs: SSR, fast, JS-based
CONs: steep learning curve, HTTP requests needed to change page's state, team with PHP-oriented skills

PROs: SSR, huge community, frontend "PHP" based
CONs: HTTP requests needed to change page's state

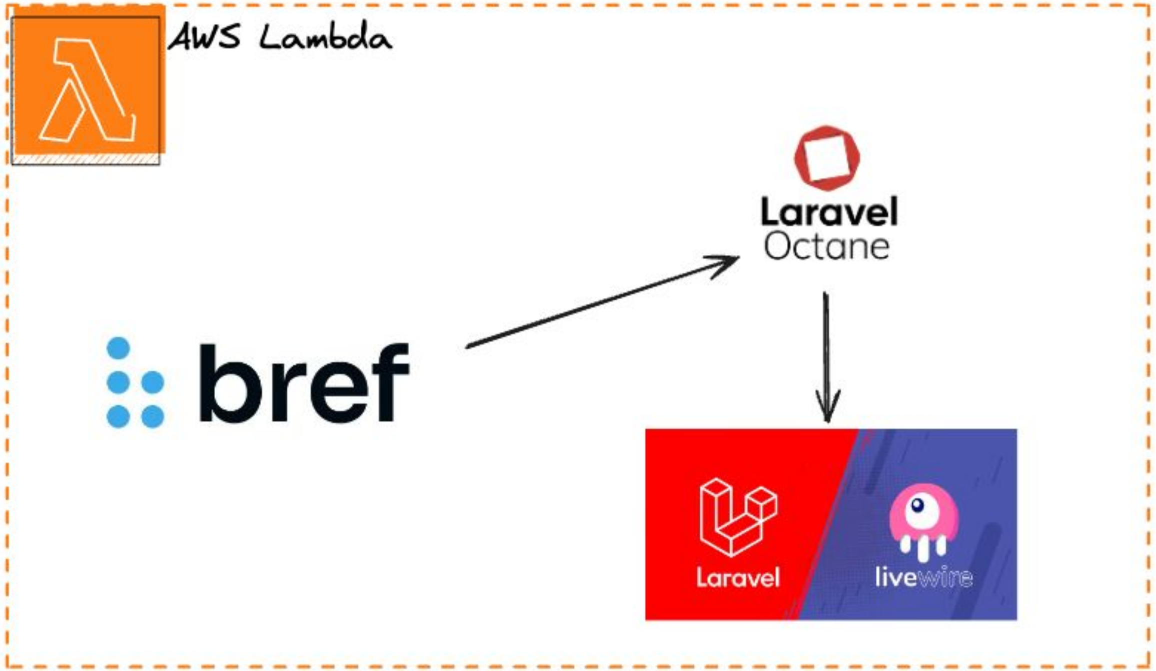
THE SOLUTION

BFF



THE SOLUTION

The application



Why Bref + Octane?

Bref

To deploy PHP applications and setup the infrastructure on Lambda

Octane vs PHP-FPM

Octane caches the Laravel application in RAM instead of reading the files from the hard drive and restarting the framework with each subsequent request

Network opt.

We consume our internal APIs via HTTPS using curl/curl multi, and we aim to avoid incurring costs associated with TLS handshake

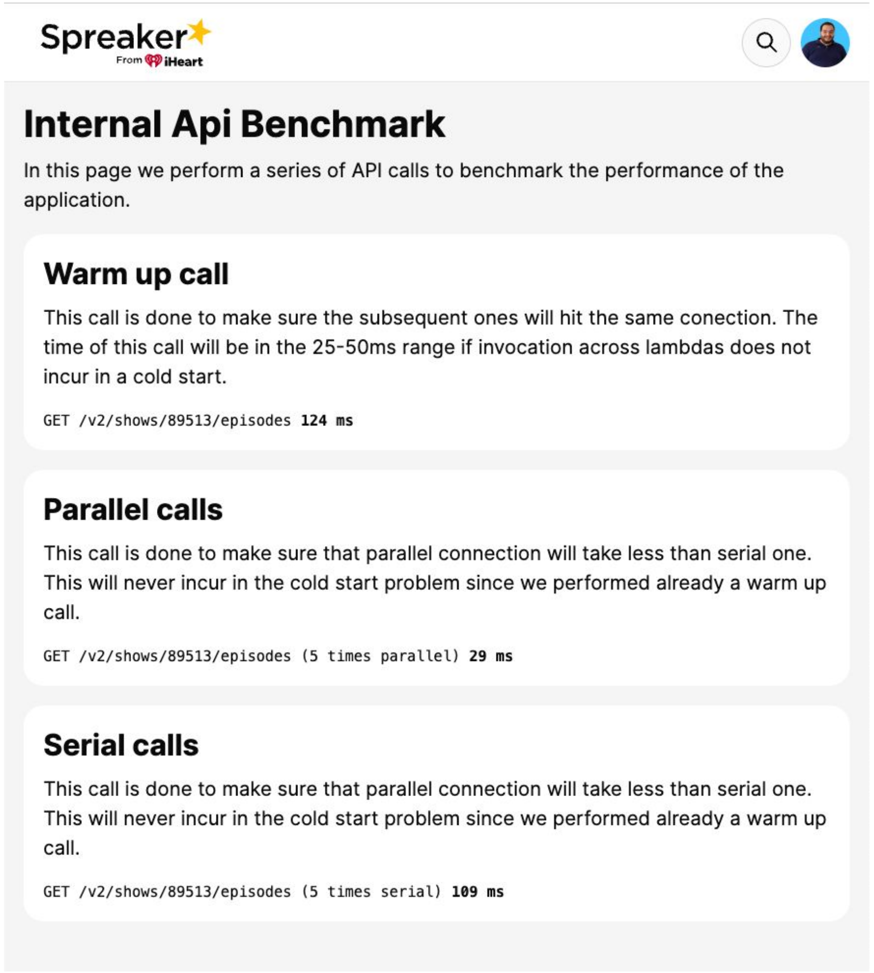
https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/

THE SOLUTION

Parallel cUrl with Connection Keep-Alive

With persistent connections, there's no need to repeatedly establish new connections for each request, resulting in reduced overhead and faster response times.

Parallel cURL with Keep-Alive maximizes network utilization, reducing latency and speeding up data retrieval.



Internal Api Benchmark

In this page we perform a series of API calls to benchmark the performance of the application.

Warm up call

This call is done to make sure the subsequent ones will hit the same conection. The time of this call will be in the 25-50ms range if invocation across lambdas does not incur in a cold start.

`GET /v2/shows/89513/episodes` **124 ms**

Parallel calls

This call is done to make sure that parallel connection will take less than serial one. This will never incur in the cold start problem since we performed already a warm up call.

`GET /v2/shows/89513/episodes (5 times parallel)` **29 ms**

Serial calls

This call is done to make sure that parallel connection will take less than serial one. This will never incur in the cold start problem since we performed already a warm up call.

`GET /v2/shows/89513/episodes (5 times serial)` **109 ms**

Why Livewire?

Frontend with PHP

Write interactive web interfaces using only PHP, eliminating the need for complex JavaScript frameworks

Real-Time Updates

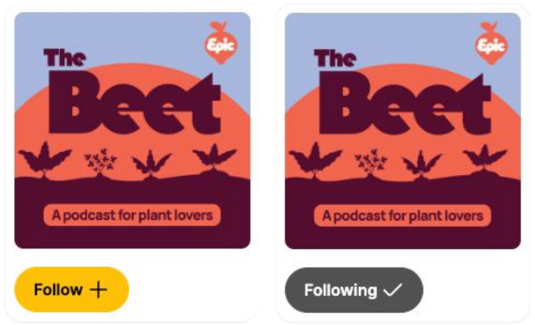
It leverages the power of Laravel's server-side logic to handle user interactions and update the UI dynamically, providing a seamless user experience.

Integration with Laravel

This ensures consistency and coherence throughout your application, while also benefiting from Laravel's robust ecosystem and community support.

THE SOLUTION

Why Livewire?

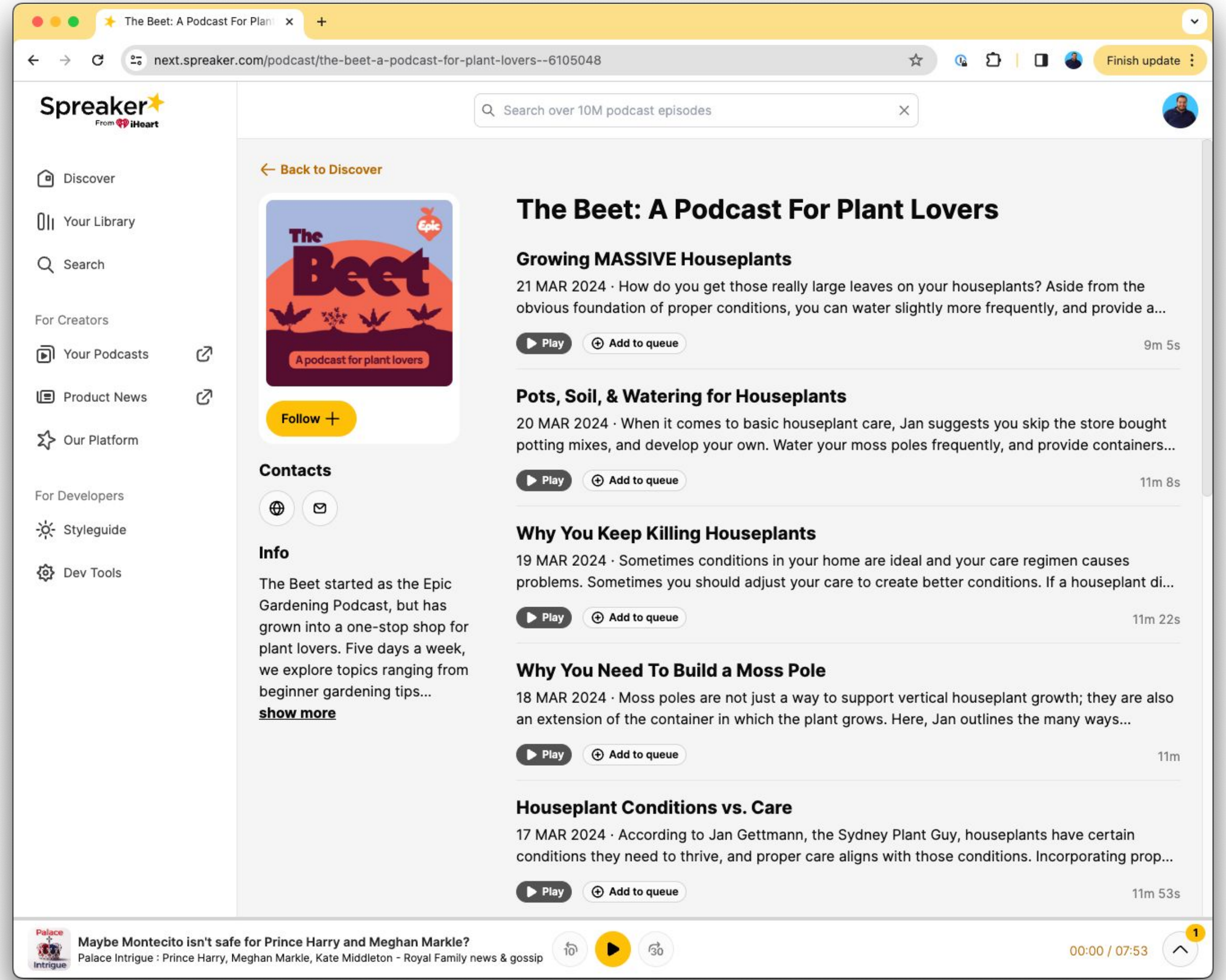


```
<div>
    @auth
        <x-common.button
            ...
            wire:loading.attr="disabled"
            wire:click="follow">
            {{ $isFollowing
                ? __('app.podcast.following')
                : __('app.podcast.follow') }}
        </x-common.button>
    @endauth
</div>
```

```
class Follow extends Component
{
    public int $podcastId;

    #[Reactive]
    public bool $isFollowing;

    public function mount(
        int $podcastId,
        bool $isFollowing
    ): void {
        $this->podcastId = $podcastId;
        $this->isFollowing = $isFollowing;
    }

    public function follow(): void
    {
         try {
             $gateway->podcast->follow($user->getAuthIdentifier(), $this->podcastId);
             $this->dispatch('podcast-follow-updated', isFollowing: true);
        } catch (SpreakerApiException) {
        }
    }

    public function render(Request $request, SymfonyUrlGenerator $symfonyRoute): View
    {
        return view('livewire.podcast.common.follow');
    }
}
```

THE SOLUTION

Spreaker Next



The Beet: A Podcast For Plant Lovers

next.spreaker.com/podcast/the-beet-a-podcast-for-plant-lovers--6105048

Finish update

Spreaker From iHeart

Search over 10M podcast episodes

- Discover
- Your Library
- Search

For Creators

- Your Podcasts
- Product News
- Our Platform

For Developers

- Styleguide
- Dev Tools

Back to Discover

Follow +

Contacts

Info

The Beet started as the Epic Gardening Podcast, but has grown into a one-stop shop for plant lovers. Five days a week, we explore topics ranging from beginner gardening tips...
show more

The Beet: A Podcast For Plant Lovers

Growing MASSIVE Houseplants
21 MAR 2024 · How do you get those really large leaves on your houseplants? Aside from the obvious foundation of proper conditions, you can water slightly more frequently, and provide a...
Play | Add to queue | 9m 5s

Pots, Soil, & Watering for Houseplants
20 MAR 2024 · When it comes to basic houseplant care, Jan suggests you skip the store bought potting mixes, and develop your own. Water your moss poles frequently, and provide containers...
Play | Add to queue | 11m 8s

Why You Keep Killing Houseplants
19 MAR 2024 · Sometimes conditions in your home are ideal and your care regimen causes problems. Sometimes you should adjust your care to create better conditions. If a houseplant di...
Play | Add to queue | 11m 22s

Why You Need To Build a Moss Pole
18 MAR 2024 · Moss poles are not just a way to support vertical houseplant growth; they are also an extension of the container in which the plant grows. Here, Jan outlines the many ways...
Play | Add to queue | 11m

Houseplant Conditions vs. Care
17 MAR 2024 · According to Jan Gettmann, the Sydney Plant Guy, houseplants have certain conditions they need to thrive, and proper care aligns with those conditions. Incorporating prop...
Play | Add to queue | 11m 53s

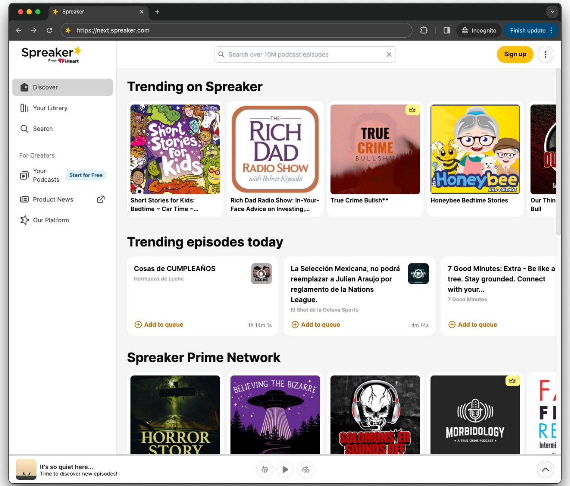
Maybe Montecito isn't safe for Prince Harry and Meghan Markle?
Palace Intrigue : Prince Harry, Meghan Markle, Kate Middleton - Royal Family news & gossip
00:00 / 07:53

Did everything go smoothly?

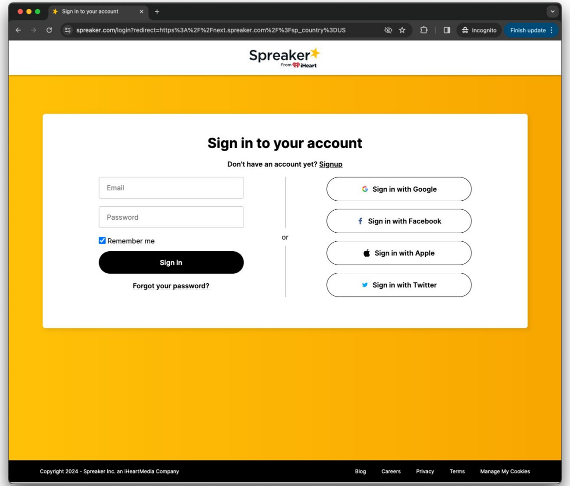
Of course it didn’t!

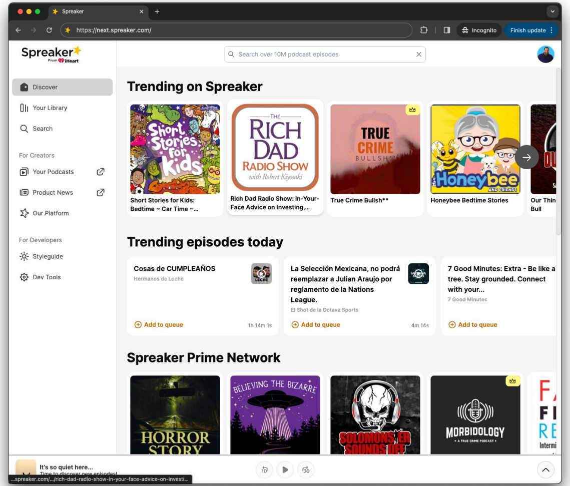
CHALLENGES FACED

Coexistence current and new website



https://next.spreaker.com
Spreaker
Search over 10M podcast episodes
Sign up
Discover
Your Library
Search
For Creators
Your Podcasts
Start for Free
Product News
Our Platform
Trending on Spreaker
Short Stories for Kids: Bedtime ~ Car Time ~...
Rich Dad Radio Show: In-Your-Face Advice on Investing,...
True Crime Bullsh**
Honeybee Bedtime Stories
Trending episodes today
Cosas de CUMPLEAÑOS
La Selección Mexicana, no podrá reemplazar a Julian Araujo por reglamento de la Nations League.
7 Good Minutes: Extra - Be like a tree. Stay grounded. Connect with your...
Add to queue
Spreaker Prime Network
It's so quiet here...
Time to discover new episodes!

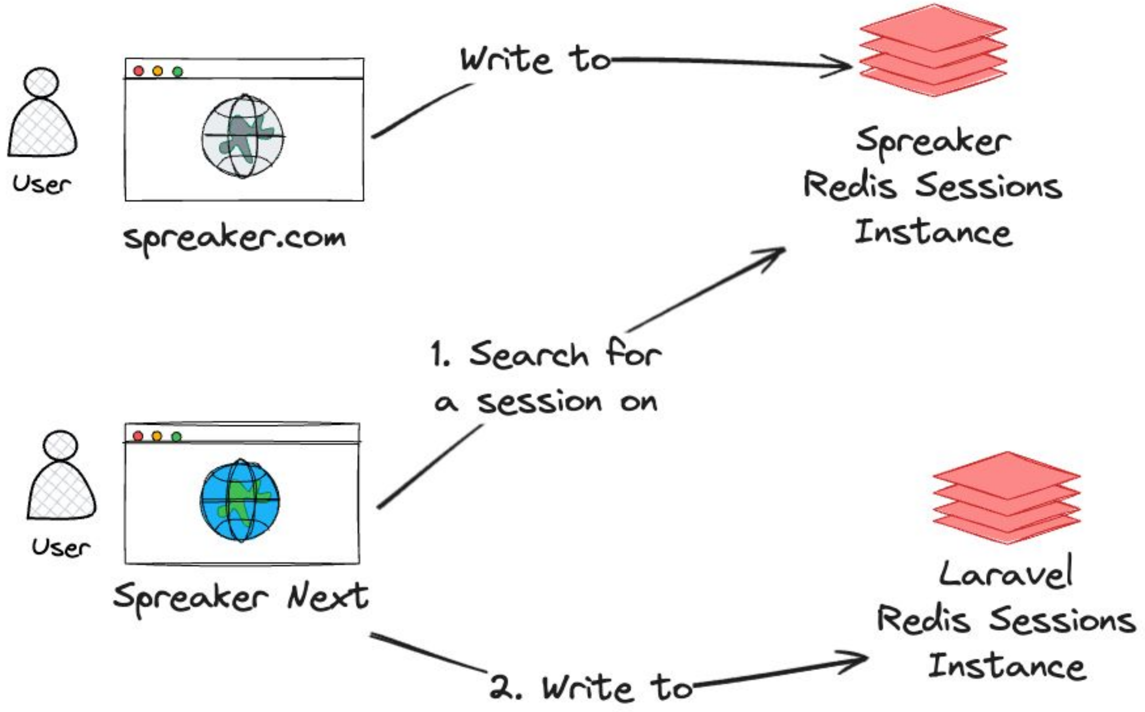




CHALLENGES FACED

Shared Sessions

Although OAuth2 authentication is available through the Spreaker API, we aimed to reduce networking costs.


User
spreaker.com
Write to
Spreaker Redis Sessions Instance
1. Search for a session on
User
Spreaker Next
2. Write to
Laravel Redis Sessions Instance

CHALLENGES FACED

Octane Memory Leaks

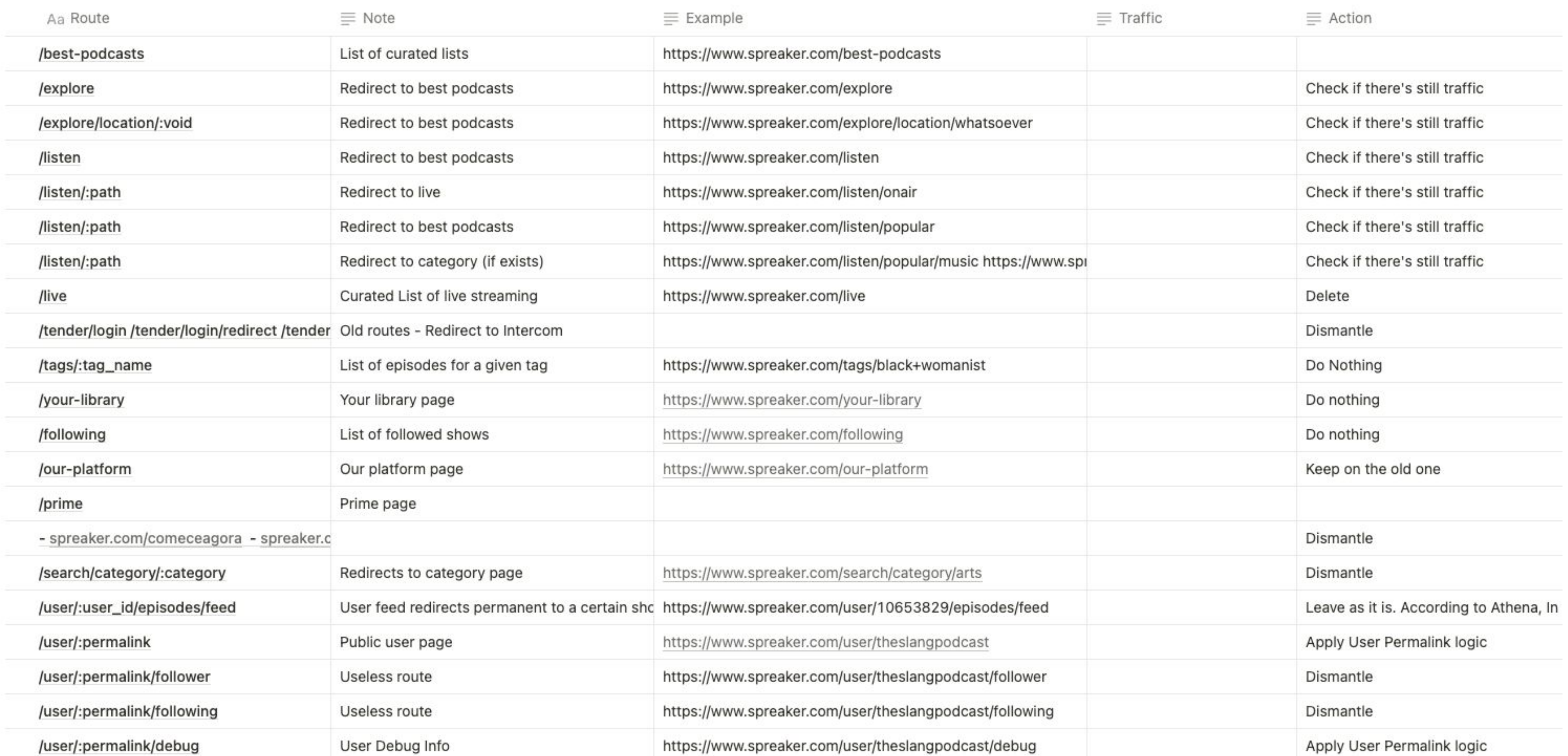
Octane keeps the application in memory between requests; therefore, adding data to a statically variables may result in a memory leak.

It is the intended behaviour for certain services (eg. the Redis client or the client that performs requests to Spreaker API)

```php
readonly class SetUserTokenInSpreakerApiClient
{
    public function handle(Request $request, \Closure $next): Response
    {
        /** @var ?SymfonyUser $user */
        $user = \auth()->user();

        // if the user is not logged in we reset the token
        $this->apiClient->withAccessToken($user?->getAccessToken());

        return $next($request);
    }
}
```

CHALLENGES FACED

Contents routing revamped

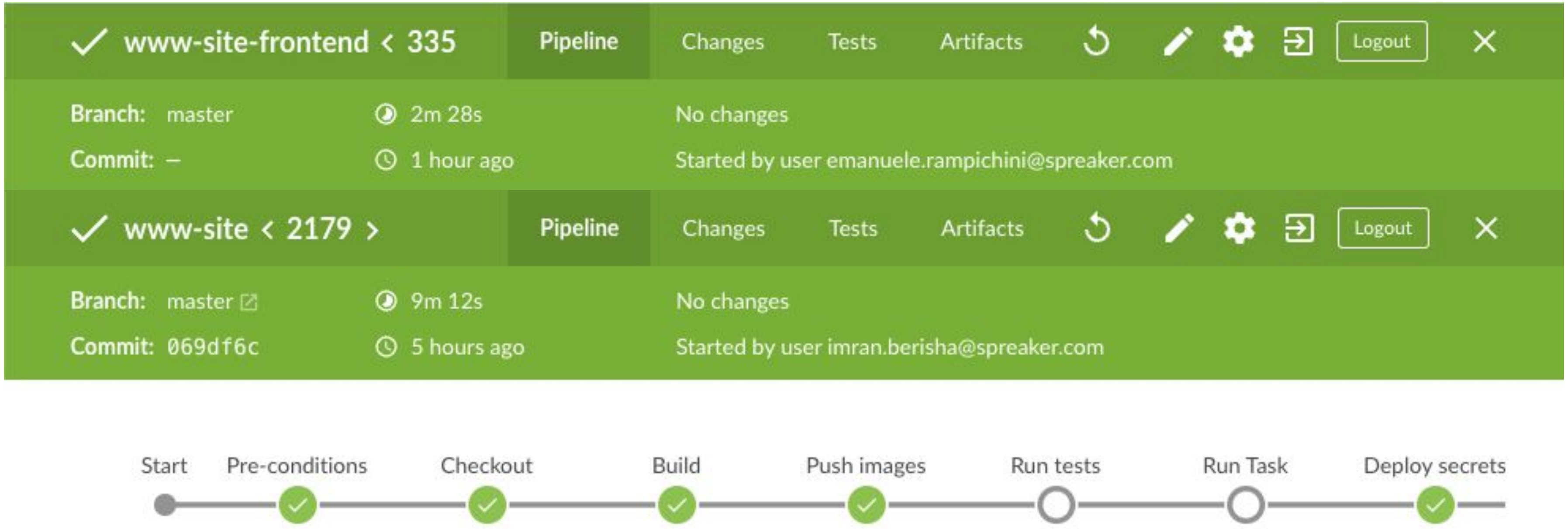


Route	Note	Example	Traffic	Action
/best-podcasts	List of curated lists	https://www.spreaker.com/best-podcasts		
/explore	Redirect to best podcasts	https://www.spreaker.com/explore		Check if there's still traffic
/explore/location/:void	Redirect to best podcasts	https://www.spreaker.com/explore/location/whatsoever		Check if there's still traffic
/listen	Redirect to best podcasts	https://www.spreaker.com/listen		Check if there's still traffic
/listen/:path	Redirect to live	https://www.spreaker.com/listen/onair		Check if there's still traffic
/listen/:path	Redirect to best podcasts	https://www.spreaker.com/listen/popular		Check if there's still traffic
/listen/:path	Redirect to category (if exists)	https://www.spreaker.com/listen/popular/music https://www.spr		Check if there's still traffic
/live	Curated List of live streaming	https://www.spreaker.com/live		Delete
/tender/login /tender/login/redirect /tender	Old routes - Redirect to Intercom			Dismantle
/tags/:tag_name	List of episodes for a given tag	https://www.spreaker.com/tags/black+womanist		Do Nothing
/your-library	Your library page	https://www.spreaker.com/your-library		Do nothing
/following	List of followed shows	https://www.spreaker.com/following		Do nothing
/our-platform	Our platform page	https://www.spreaker.com/our-platform		Keep on the old one
/prime	Prime page			
- spreaker.com/comeceagora - spreaker.c				Dismantle
/search/category/:category	Redirects to category page	https://www.spreaker.com/search/category/arts		Dismantle
/user/:user_id/episodes/feed	User feed redirects permanent to a certain sho	https://www.spreaker.com/user/10653829/episodes/feed		Leave as it is. According to Athena, In
/user/:permalink	Public user page	https://www.spreaker.com/user/theslangpodcast		Apply User Permalink logic
/user/:permalink/follower	Useless route	https://www.spreaker.com/user/theslangpodcast/follower		Dismantle
/user/:permalink/following	Useless route	https://www.spreaker.com/user/theslangpodcast/following		Dismantle
/user/:permalink/debug	User Debug Info	https://www.spreaker.com/user/theslangpodcast/debug		Apply User Permalink logic

Is there anything else worth sharing?

THE SOLUTION

Serverless deploy


✓ www-site-frontend < 335

Branch: master
Commit: –
2m 28s
1 hour ago
No changes
Started by user emanuele.rampichini@spreaker.com

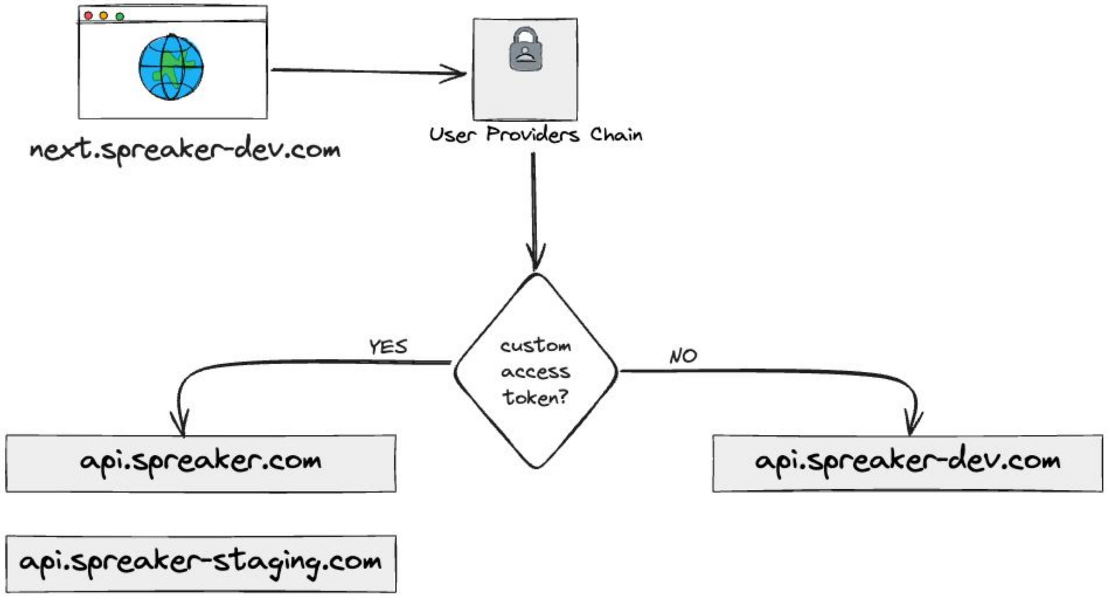
✓ www-site < 2179 >

Branch: master
Commit: 069df6c
9m 12s
5 hours ago
No changes
Started by user imran.berisha@spreaker.com

THE SOLUTION

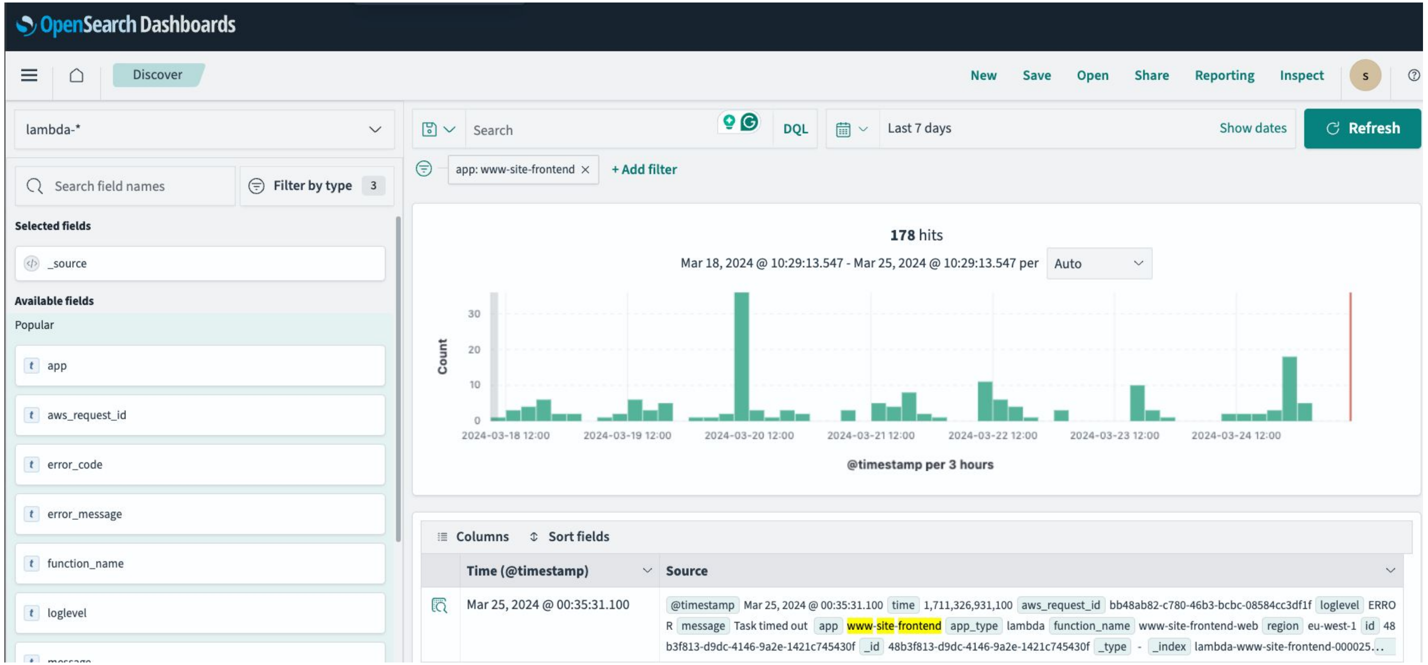
Local env against staging and prod data

To speed up local development and debugging, we built a user provider able to make requests against staging and production environments given a proper authentication token



THE SOLUTION

Observability layer



Index pattern: lambda-*

Last 7 days

Filter: app: www-site-frontend

Selected fields: _source

Available fields (Popular): app, aws_request_id, error_code, error_message, function_name, loglevel, message

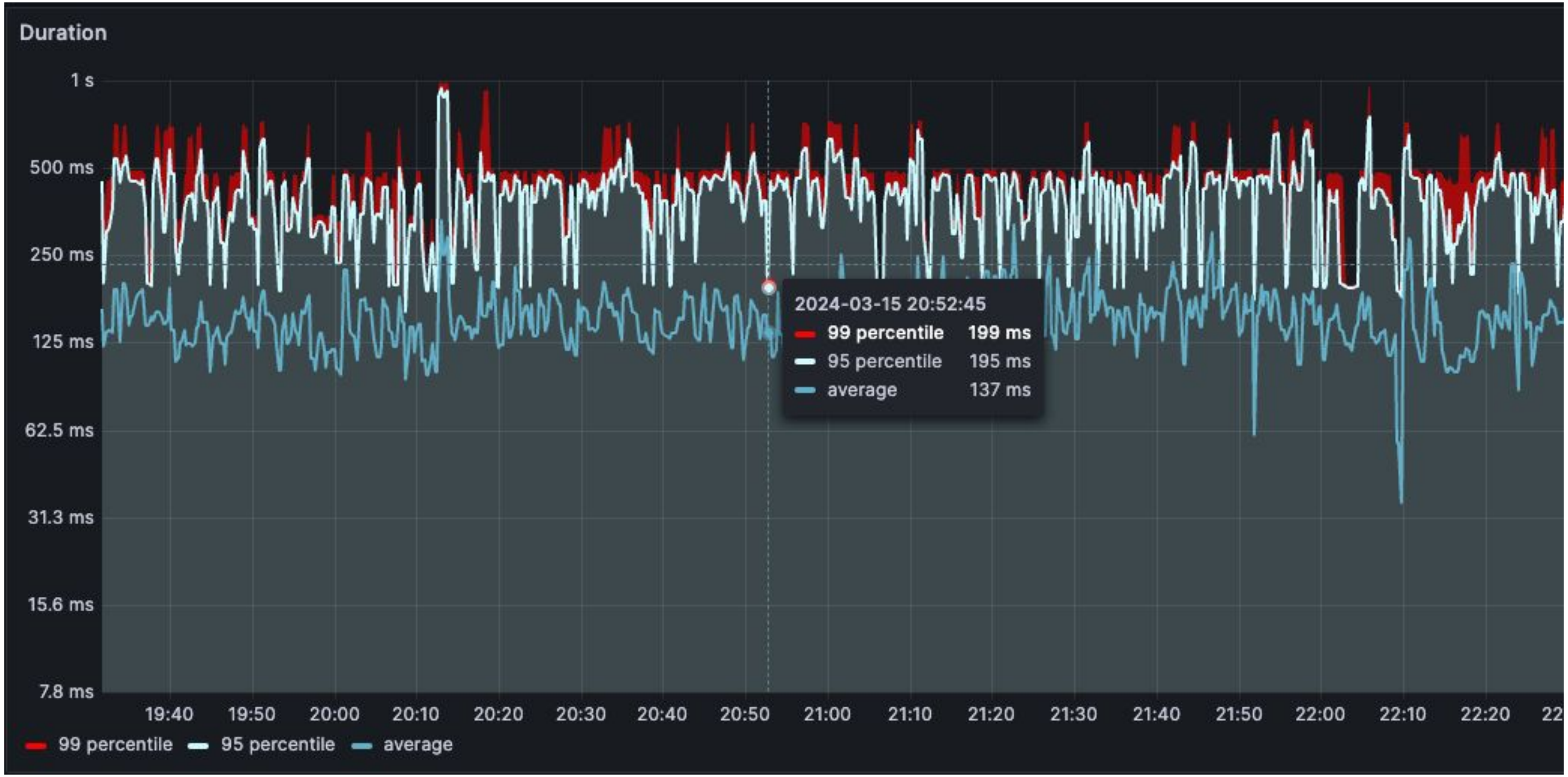
178 hits

Mar 18, 2024 @ 10:29:13.547 - Mar 25, 2024 @ 10:29:13.547 per Auto

Time (@timestamp)	Source
Mar 25, 2024 @ 00:35:31.100	@timestamp Mar 25, 2024 @ 00:35:31.100 time 1,711,326,931,100 aws_request_id bb48ab82-c780-46b3-bcbc-08584cc3df1f loglevel ERROR message Task timed out app www-site-frontend app_type lambda function_name www-site-frontend-web region eu-west-1 id 48b3f813-d9dc-4146-9a2e-1421c745430f _id 48b3f813-d9dc-4146-9a2e-1421c745430f _type - _index lambda-www-site-frontend-000025...

THE SOLUTION

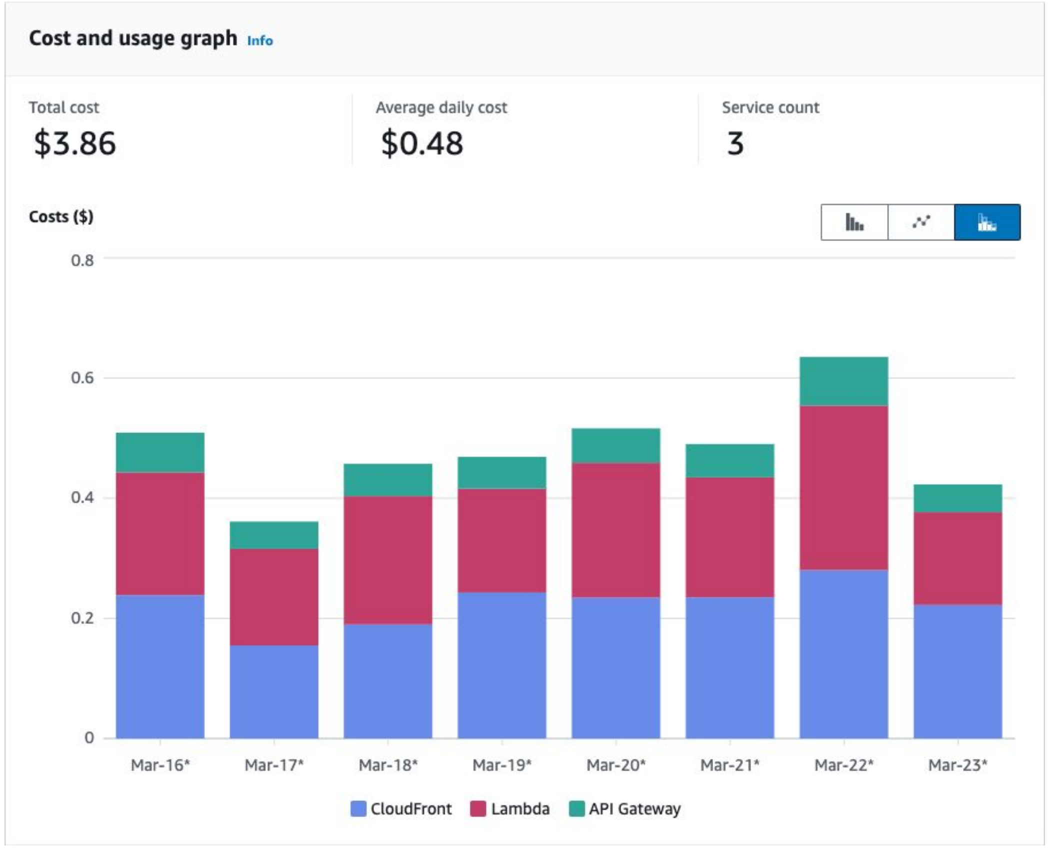
Metrics



THE SOLUTION

Cost Forecasts

Based on the current traffic-generated costs, we anticipate an average daily cost of around $21, with a projected annual cost of approximately $5k.



Cost and usage graph Info

Total cost	Average daily cost	Service count
$3.86	$0.48	3

THE SOLUTION

Timeline

Originally scheduled for early Q4 2024, we moved up the switch date to early Q2 as soon as we observed encouraging development speed and stickiness metrics.

POCs, Foundational
September - December, 2023

New Routes
December, 2023

Iteration 0
January-February, 2024

Iteration 1
March, April 2024

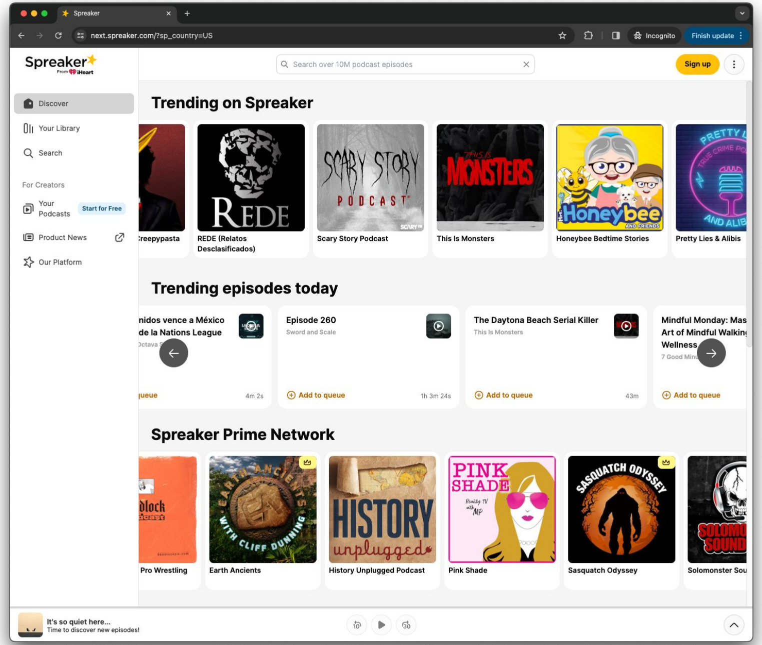
Release
April 2024

KEY TAKEAWAYS

Is it possible to build modern and scalable apps with PHP in 2024?

We think so, but try it out and let us know your POV!

🔗 next.spreaker.com



improove

WEB DAY

27 MARZO 2024 7A EDIZIONE

SLIDE + VIDEO

HTTPS://WWW.IMPROOVE.TECH/APP