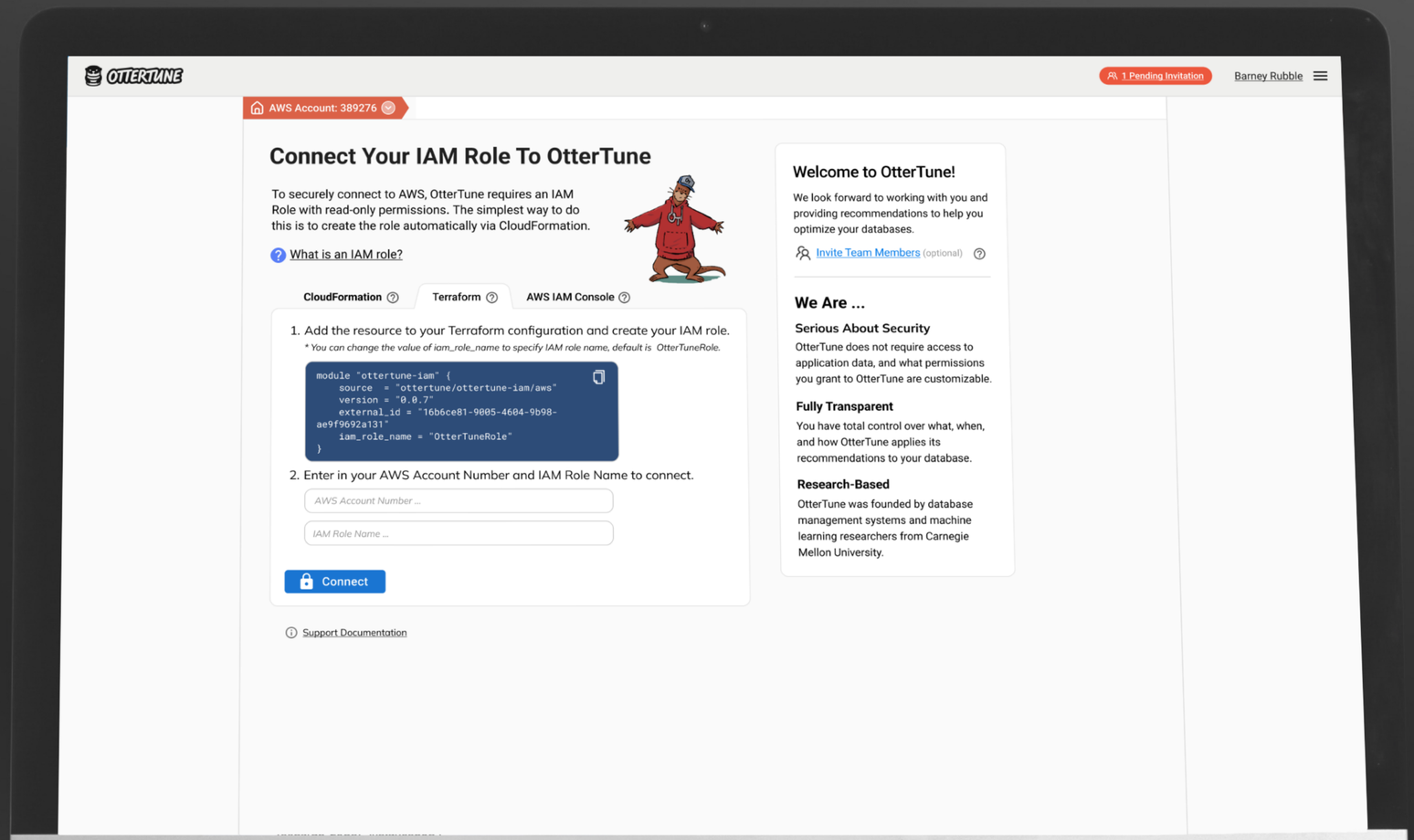
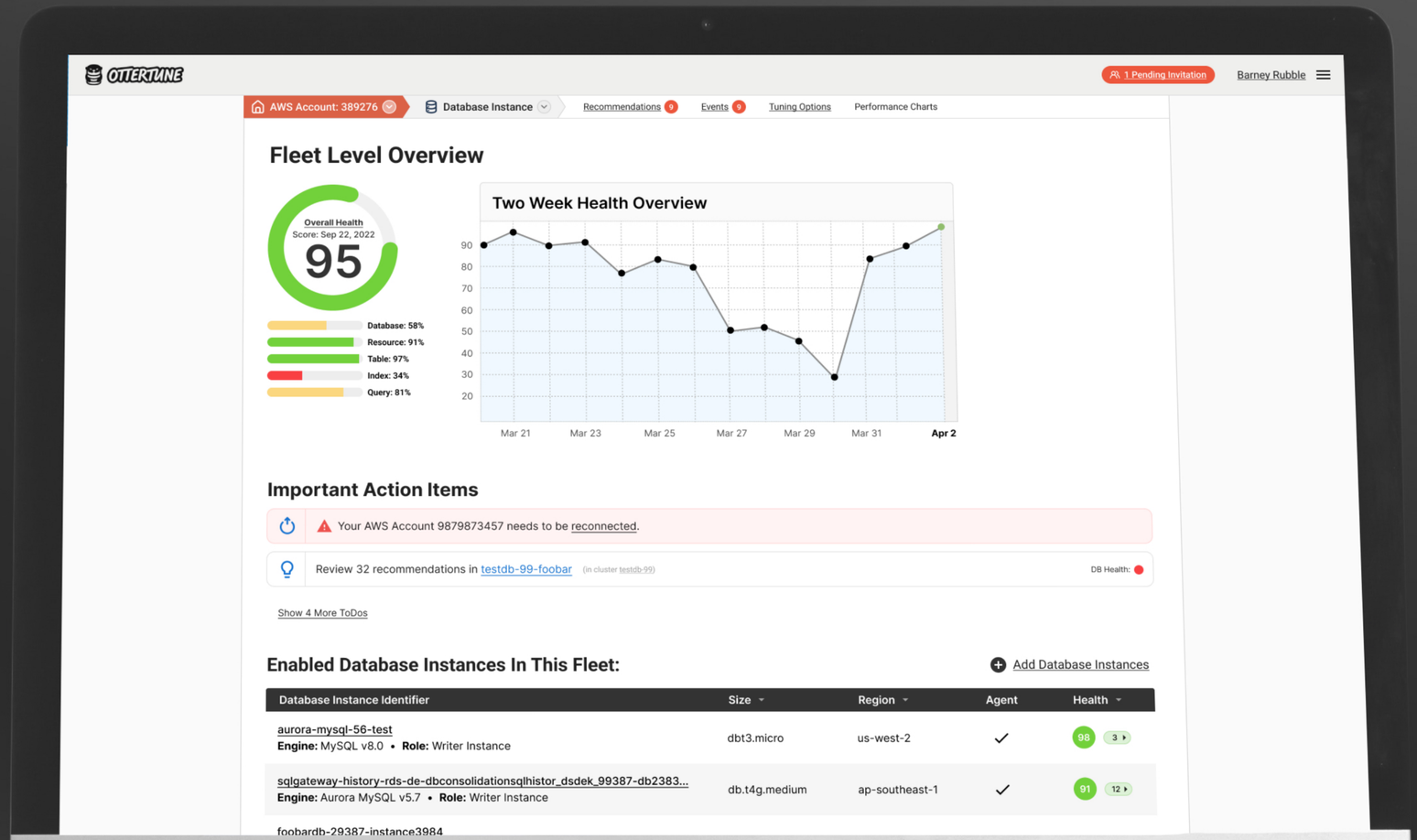
Application Redesign Case Study

- **Role:** Head of Design
- **Company:** OtterTune
- **Timeline:** v1 to v2 Redesign
- **Technology:** AI-powered advisor for relational databases (AWS MySQL and PostreSQL)
- **Location:** Remote (based in Pittsburgh, PA)



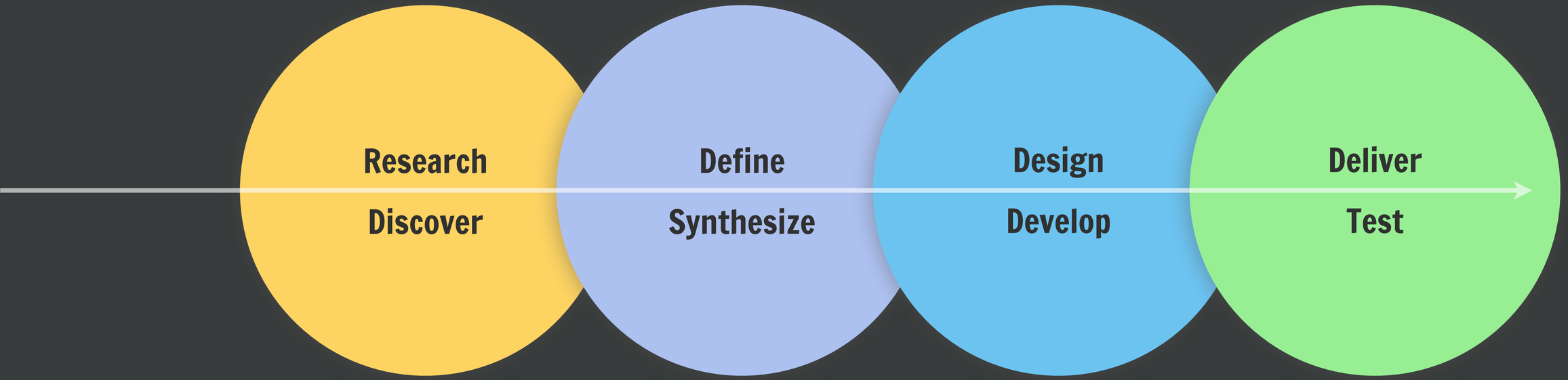
Intended Users:

- **General Partners (GPs):** Owners of accounts, responsible for inviting and approving investors.
- **Investors:** Primarily 60+ year old males, not very tech-savvy, used to manual processes.
- **Fund Administrators (FAs):** Tech-savvy individuals handling background checks and administrative tasks.



OTTERTUNE

The Design Process:



Meet with important stakeholders both internally and externally, scope out the competition, read documentation, etc.

Sketch out product ideas and concepts via FigJam, wire framing, drawing on paper, white boarding and more.

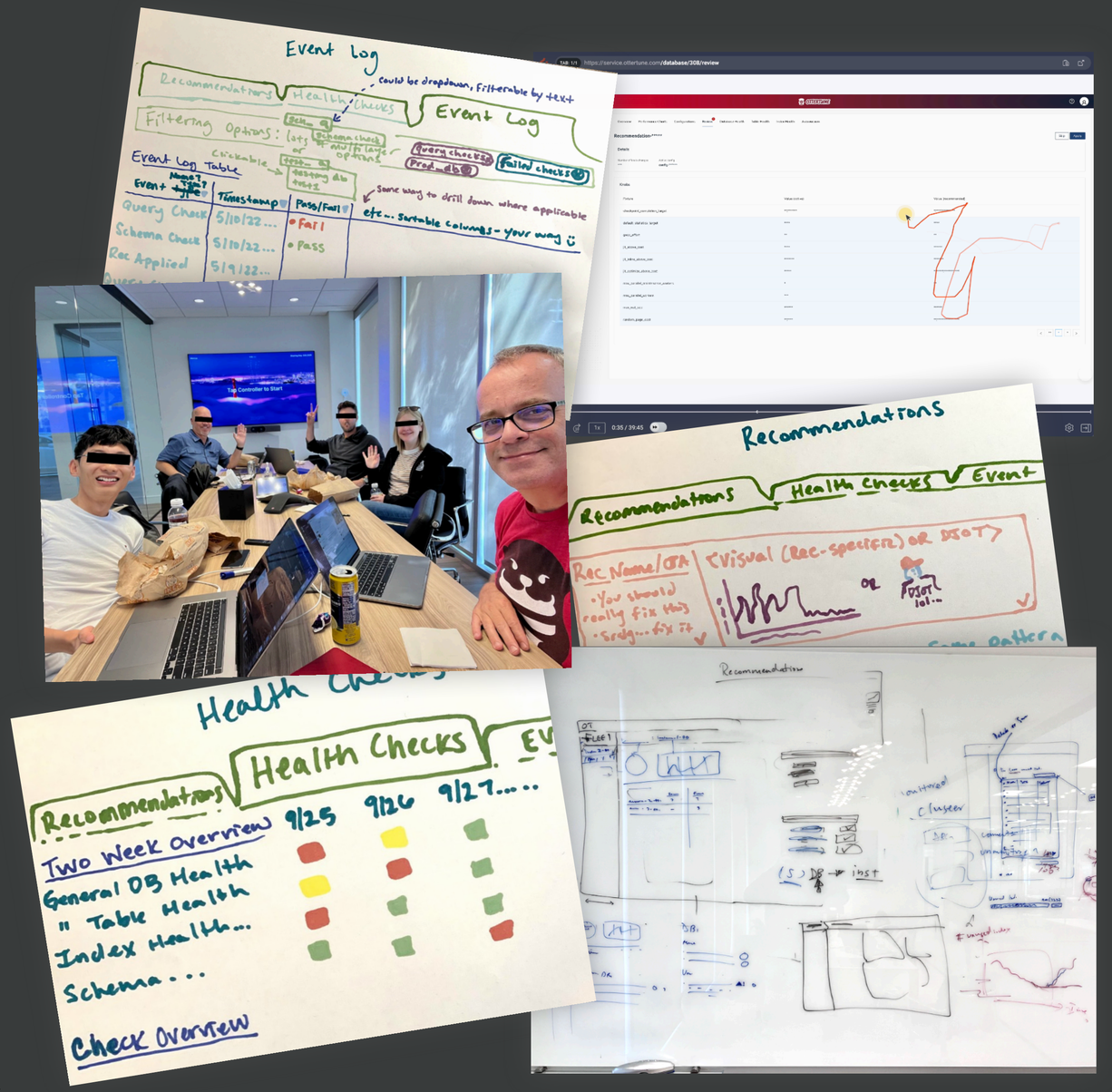
After sign off from stakeholders, develop high-fidelity mocks, prototypes and style guides for engineering in Figma

Release the product, but it doesn't end there. Testing and observation begins for incremental product improvements.

The Design Process:

1. Research & Discover

- Methods
  - Reviewed HotJar user videos and internal documentation.
  - Ongoing conversations on Google Meet and offsite meetings with product management, founders, engineering and marketing to white board, sketch and wire-frame ideas, concepts, requirements and goals for the redesign.
  - Engage in competitive analysis (e.g. Dynatrace, Datadog, EverSQL, pganalyze, Vantage, etc). Most were either all text dashboards or all charts. Not a healthy balance between the two.
  - Engaged with power users for feedback. This included both design partners as well as paying members of the application.

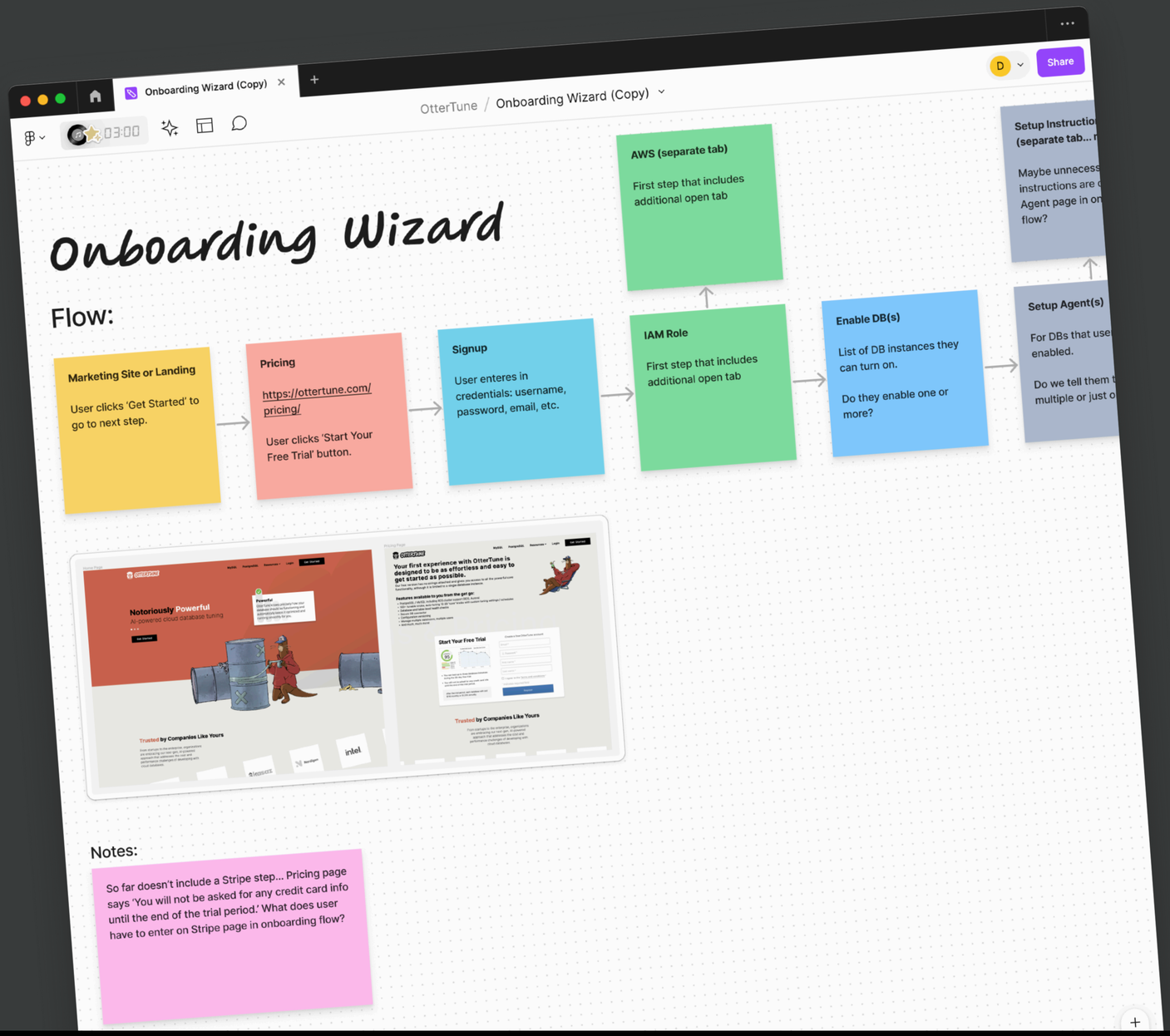


* Product discussions, HotJar video analysis, sketches, white boarding.

The Design Process:

2. Define & Synthesize:

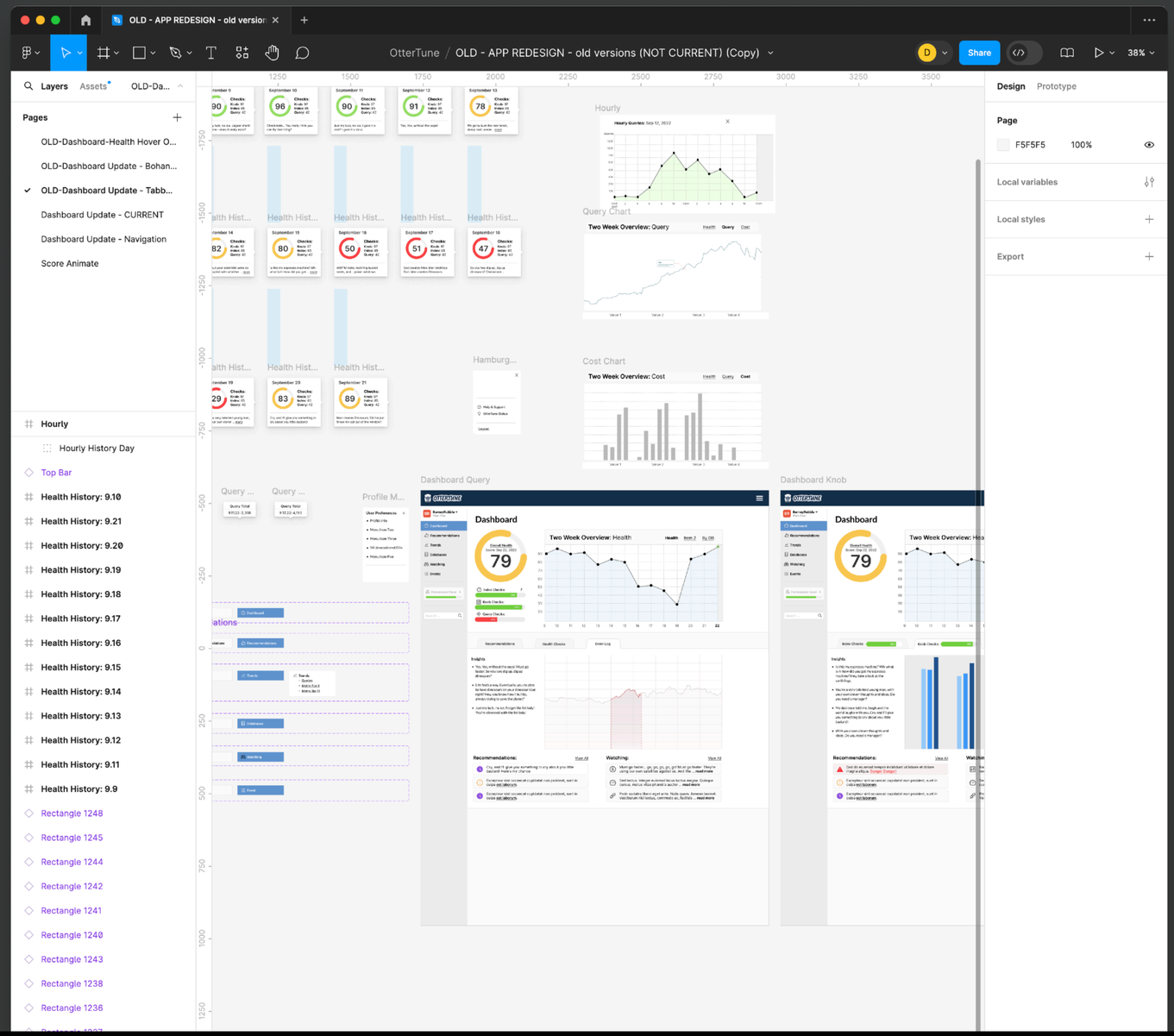
- Activities
  - After initial research process, began process of putting down ideas into Figma for basic prototyping.
  - Developed a site map for navigation and feature flows.
  - Iteratively refined design concepts and feedback, regularly presenting to stakeholders ensuring alignment with product goals and requirements.



The Design Process:

3. Design & Develop

- Activities
  - Built high-definition feature prototypes in Figma, incorporating feedback through Figma's commenting feature, Slack, Google Meet and offsite meetings.
  - Final feature flows were added to Jira for reference either as screenshots and/or video walk throughs of Figma prototypes.
  - Updated designs based on ongoing discussions and feedback.



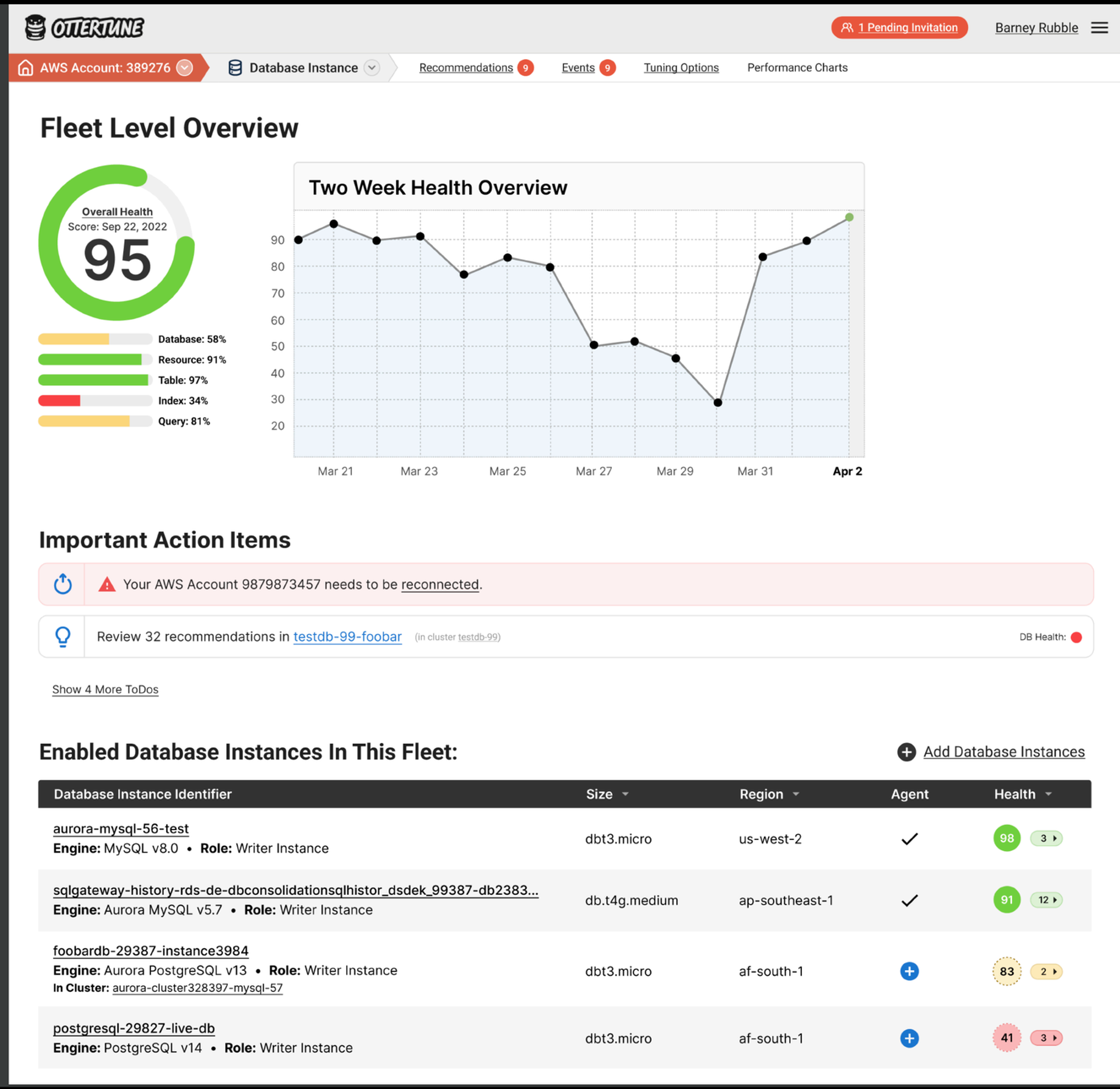
The Design Process:

4. Deliver & Test

- Activities
  - Collaborated closely with engineers to ensure fidelity to Figma mocks.
  - Figma served as a UI style guide for engineering to refer to throughout the implementation process.
  - Conducted regular reviews and weekly bug bashes to test the product before release.
  - **THE PRODUCT IS RELEASED!!**
  - Continued to analyze and monitor live product through HotJar user activity videos and traffic log data to make ongoing incremental improvements.

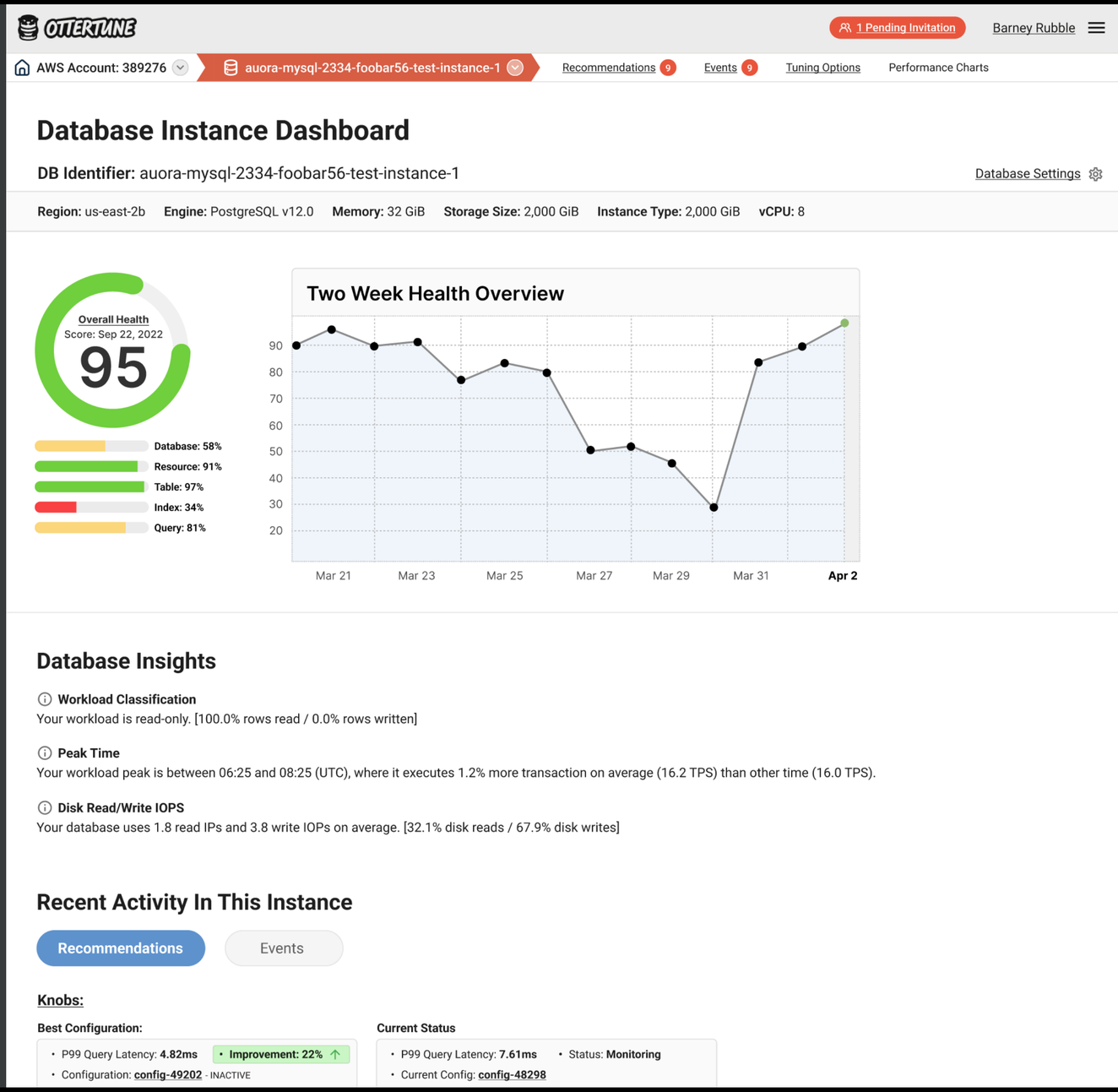
Key Redesign Features:

- **Fleet Dashboard**
  - Transitioned from a simple list to a comprehensive overview with animated health scores, time charts, aggregated fleet database recommendations and database listings.
  - Health scores would animate in, both the circle and bars below, similar to a car dashboard turning on.
  - Round up of most critical recommendations from various fleet databases for user to immediately jump to.
  - List of databases, with active databases listed first followed by databases not added to OtterTune from AWS.
  - Each database listed number of recommendations for user to jump straight to.



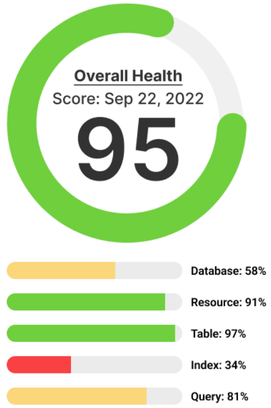
Key Redesign Features:

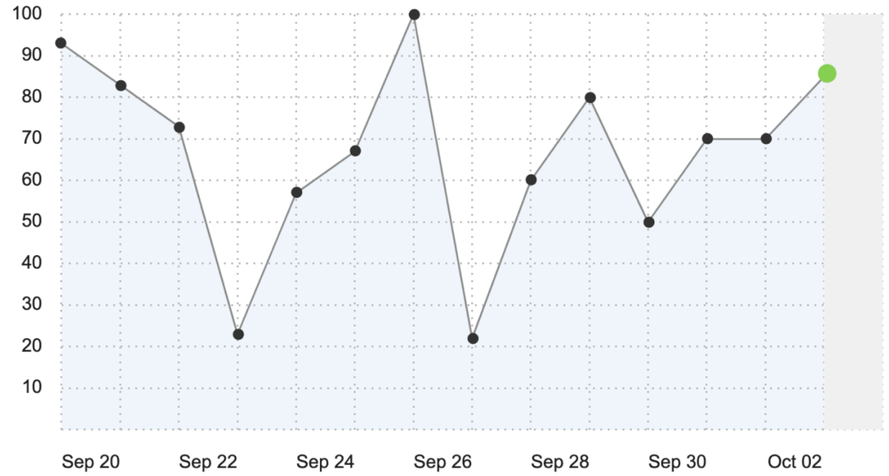
- **Database Dashboard**
  - Enhanced from text-heavy notifications to a visually engaging page with overall health scores and recommendations previews.
  - Page consisted of immediate database insights (e.g. workload activity, active or inactive, etc)., recommendations specifically for knobs, indexes and tables.
  - Page had same health score presentation and animation, but health score was specific to the database.

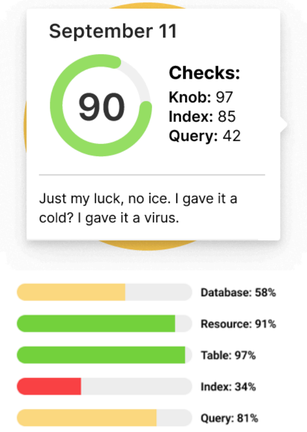


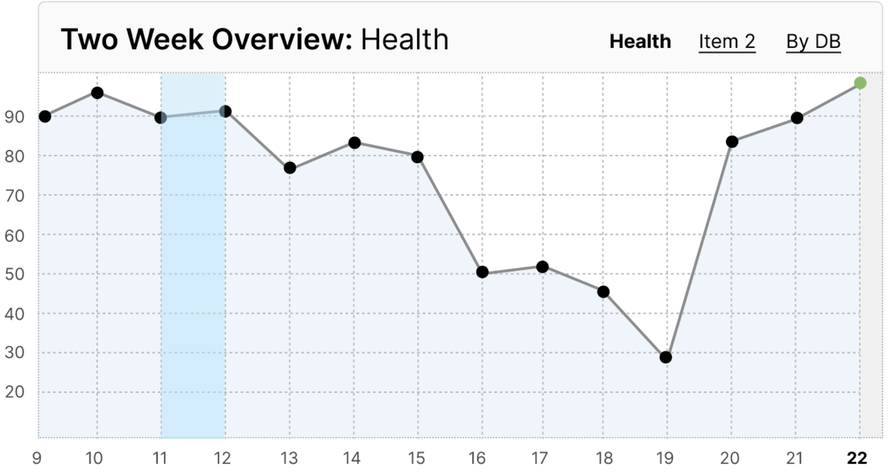
Key Redesign Features:

- **Health Scores**
  - Added visual health scores across various views for quick assessment and access to recommendations.
  - As the dashboard would load, health scores would animate in, both the circle and bars below, similar to a car dashboard turning on.
  - Goal was to immediately provide the user with a visual cue on the health of one's fleet and dashboard.
    - 85-100 was green (healthy)
    - 70-84 was yellow (warning)
    - 0-69 was red (emergency, must fix)
  - Chart showed a two-week history of a fleet and dashboards health scores. Hovering over a date column showed overall health and individual health scores.









OTTERTUNE

Key Redesign Features:

- **Knob Recommendations**
  - Introduced new feature for recommending database knob adjustments with options for automatic updates, approval, or manual implementation.
  - OtterTune performed DB checks every 10-15 minutes. Either a user could dismiss the recommendation or the recommendation would no longer appear once the recommendation was applied and checked.

Recommendations

DB Identifier: auora-mysql-2334-foobar56-test-instance-1

Region: us-east-2b **Engine:** PostgreSQL 12 **Memory:** 32 GiB **Storage Size:** 2,000 GiB **Instance Type:** 2,000 GiB **vCPU:** 8

Knobs | Indexes | Queries

Recommendation Type	Instance & Visual Overview

☐ **Tune Knob max_connection**

aurora-mysql-56-test-insance-5

Related check: Connection utilization (DB Check) (> details: Your database uses 10 of its 200 available connections on average (5%). This value can indicate efficient connection management. You can increase your max connection.)

Apply Selected Recomendations

Applied Recommendations

More Configs: 1

config-58933: 2 Knob Recommendations Applied - P99 Query Latency: **4.3** (ms) *Processing recommendation...*

Knob Name:	From	To
• tmp_table_size	XX	YY
• max_heap_table_size	YY	YY

config-49369: 12 Knob Recommendations Applied - P99 Query Latency: **4.3** (ms) View Configuration

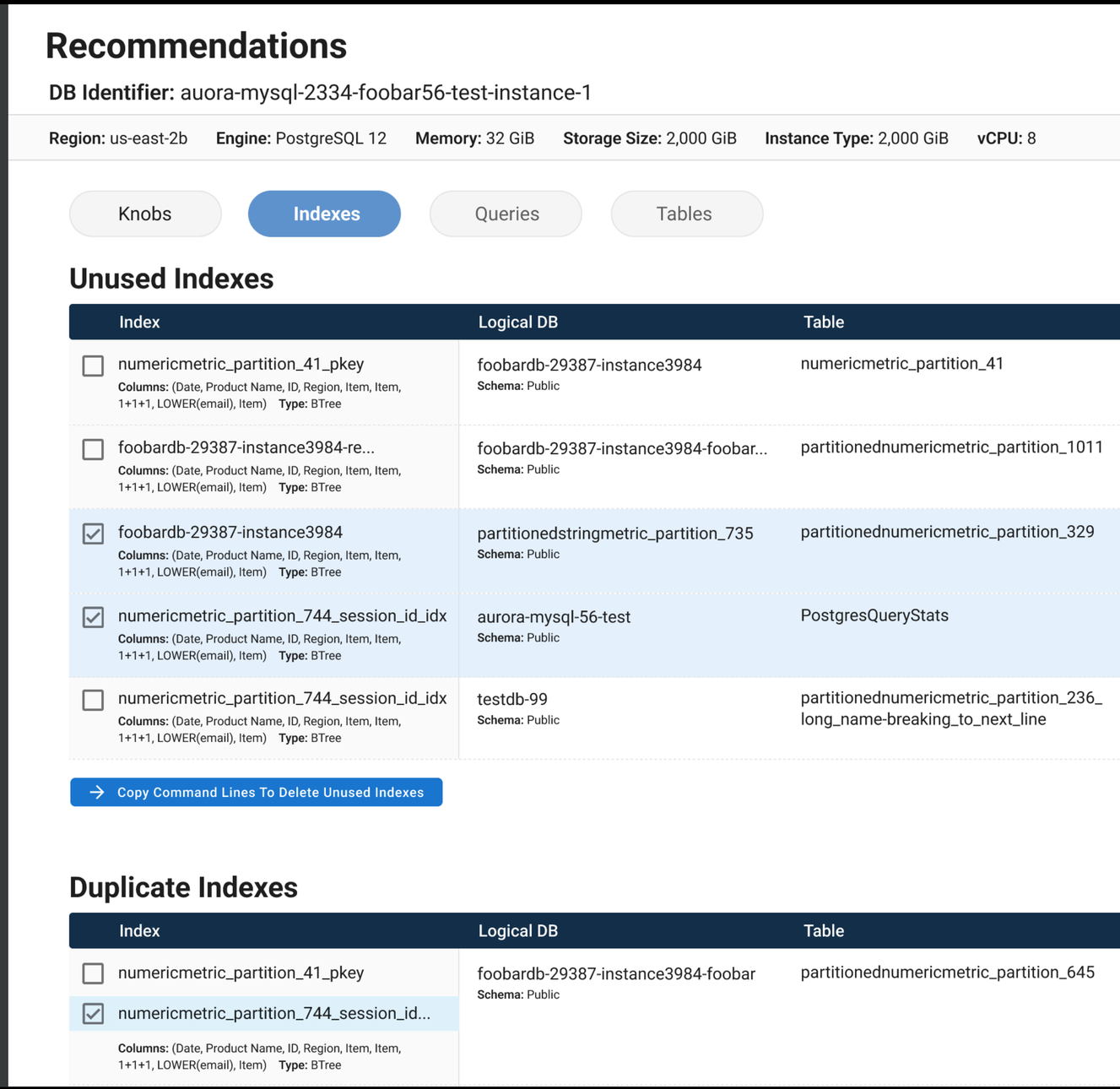
Feb 25, 2022 - 08:45 GMT

Knob Name:	From	To
• autovacuum_analyze_scale_factor	0.01	0.05
• autovacuum_analyze_threshold	27	50
• array_nulls	off	on
• cpu_operator_cost	0.0017	0.0025
• cursor_tuple_fraction	0.09	0.1
• default_statistics_target	83	100

OTTERTUNE

Key Redesign Features:

- **Index and Query Recommendations**
  - Transformed informational content into actionable recommendations with SQL code for easy implementation.
  - Query recommendations were observations and analysis only. It did not (yet) provide corrected SQL code for the user to take, although that was the ultimate plan.



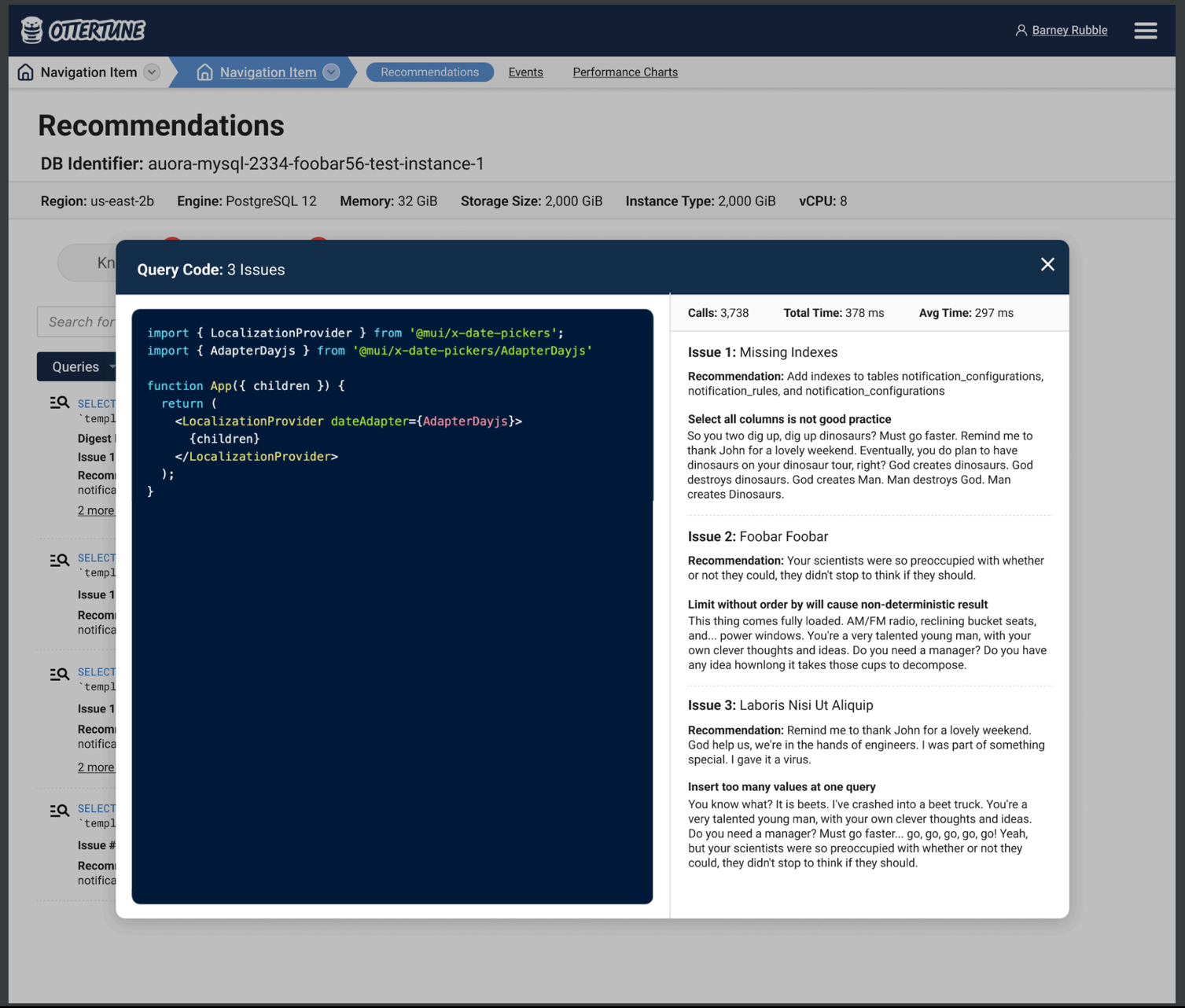
OTTERTUNE

ures:

mendations

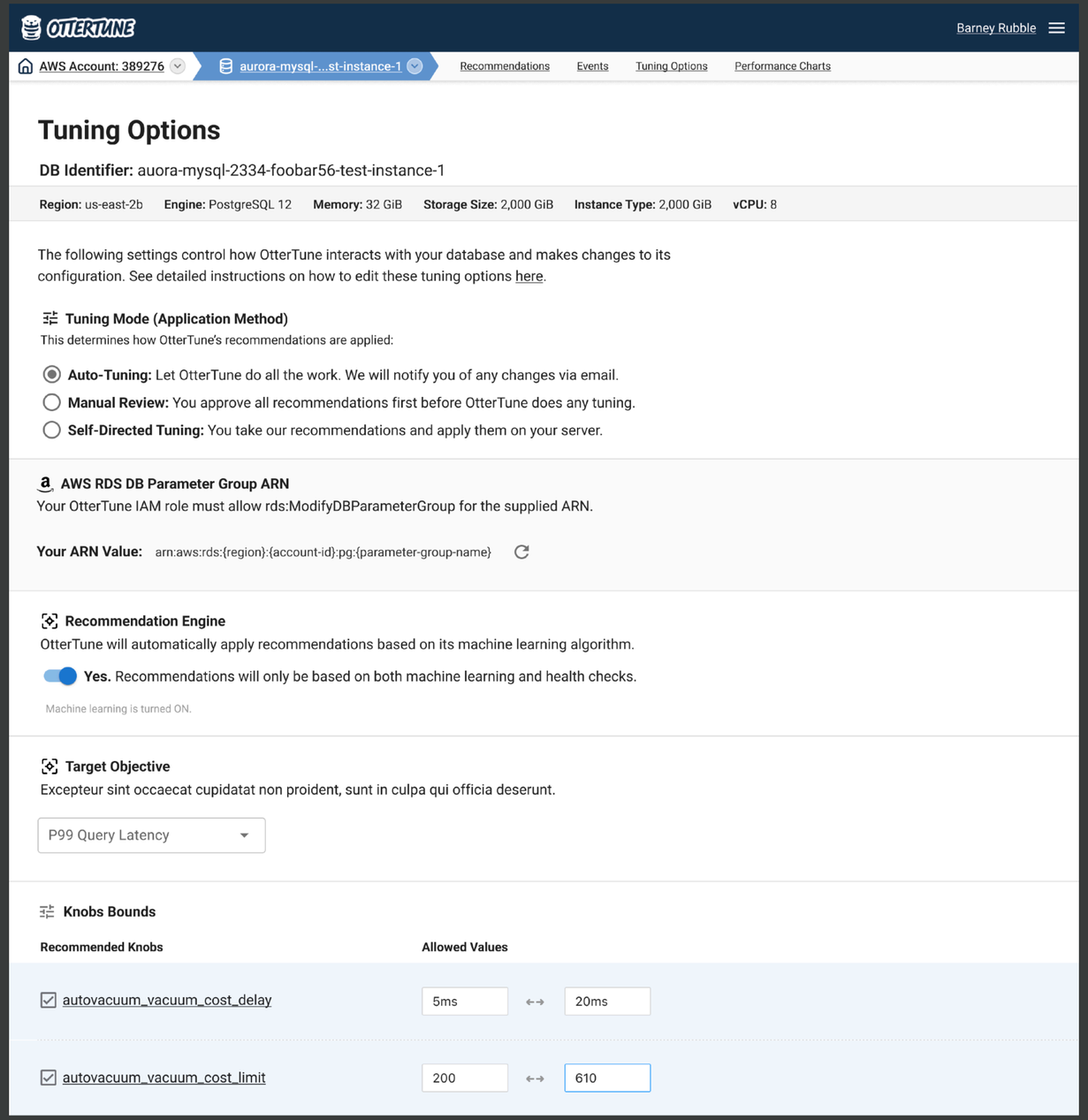
ontent into actionable
code for easy implementation.

e observations and analysis only.
cted SQL code for the user to take,
e plan.



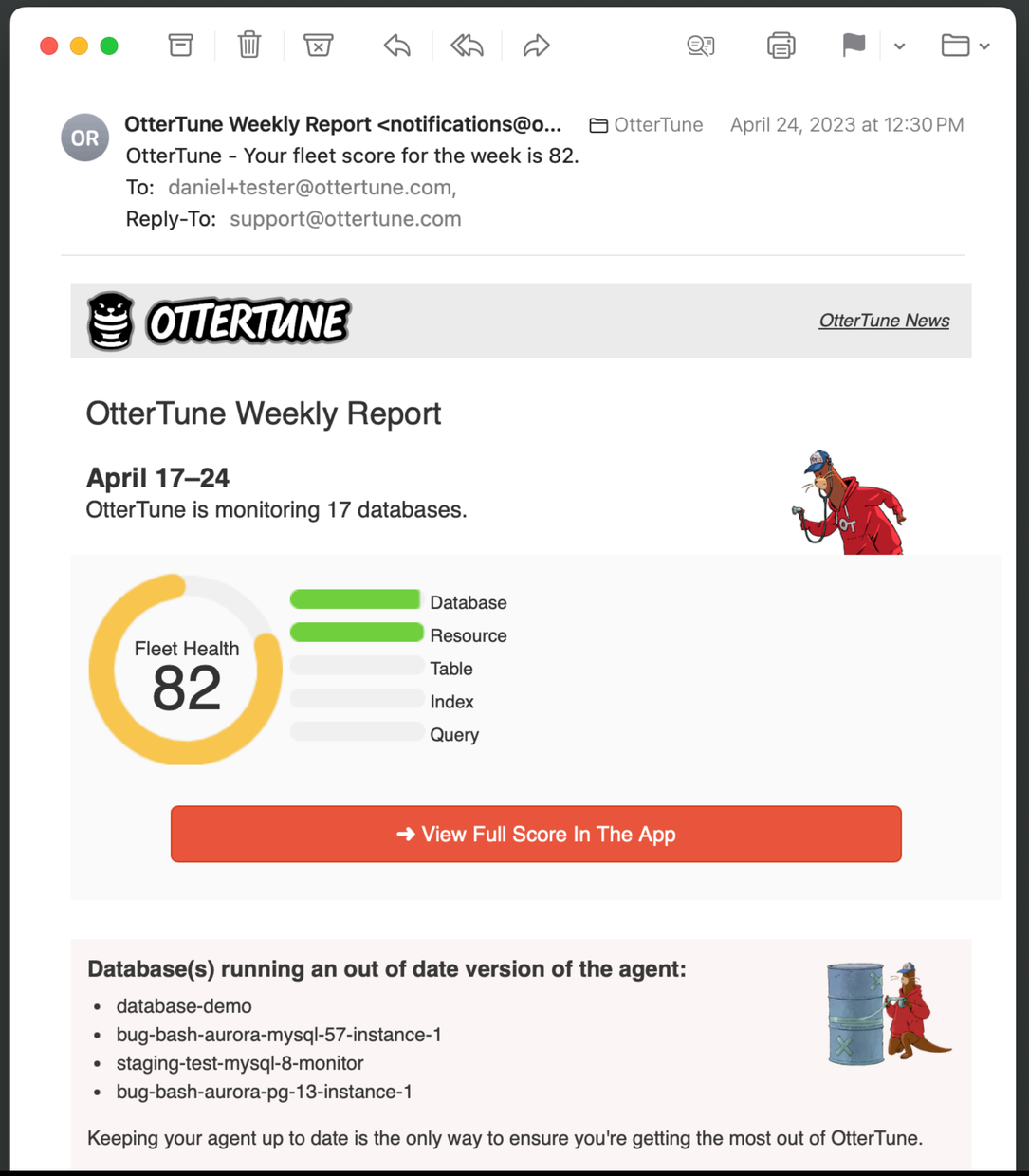
Key Redesign Features:

- **Tuning Options**
  - Provided flexible tuning options, including automatic tuning, approval-based tuning, and manual implementation.
  - User could select from the following options:
    - **Auto-Tuning:** user would allow OtterTune to automatically change settings for knobs, indexes, tables and more. They would receive email notifications of any change made.
    - **Manual Review:** OtterTune would request permission to make changes on their database, but after user approved first.
    - **Self-Directed:** OtterTune would present the user with recommendations for the user to do on their own in their database console. If needed, SQL code was provided.
  - Users at first typically selected Manual Review in order to gain trust from the OtterTune application. If it did what they wanted and expected, they often switched to Auto-Tuning mode.



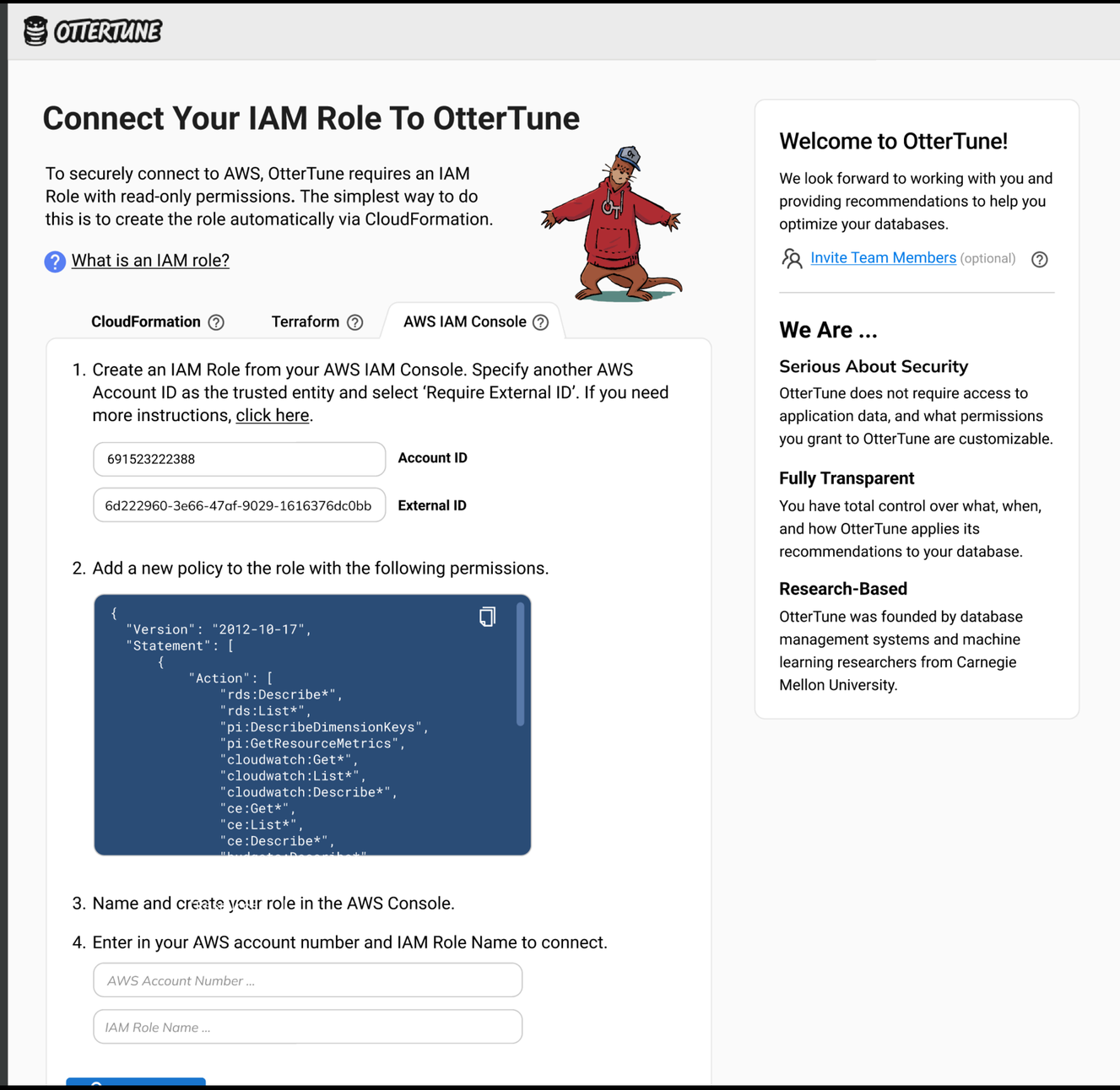
Key Redesign Features:

- **Email Notifications**
  - Redesigned email notifications to better match with company branding, health score UI and easier ability to read.
  - I started the design process in Figma for the email redesign but then finished it by coding out the email in HTML and (inline) CSS to get the true feel for how it would look and delivered to various email clients.



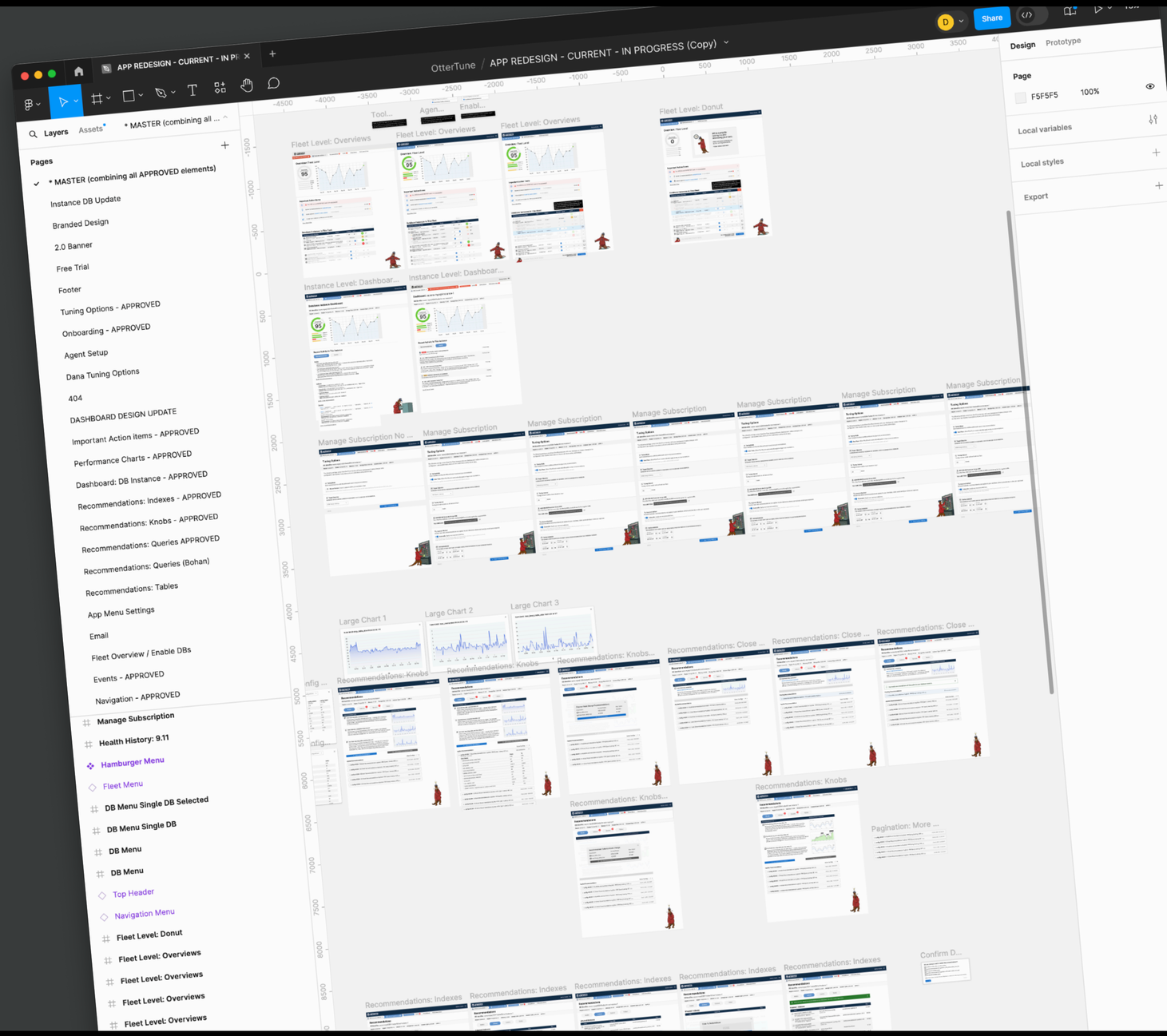
Key Redesign Features:

- **Onboarding**
  - Improved onboarding flow, increasing success rates from 10% to over 30%.
  - Connecting a user's IAM permissions on AWS was the most critical access point in onboarding customers.
  - Without IAM, OtterTune was useless to a user. There was nothing to report on without access to their account.
  - CloudFormation was by far the most popular method used and was visible by default when accessing the page. More advanced, high-end DBAs tended to prefer using Terraform or the AWS IAM Console.
  - The redesign allowed for messaging about why they could use and trust OtterTune to make changes to improve their database performance.

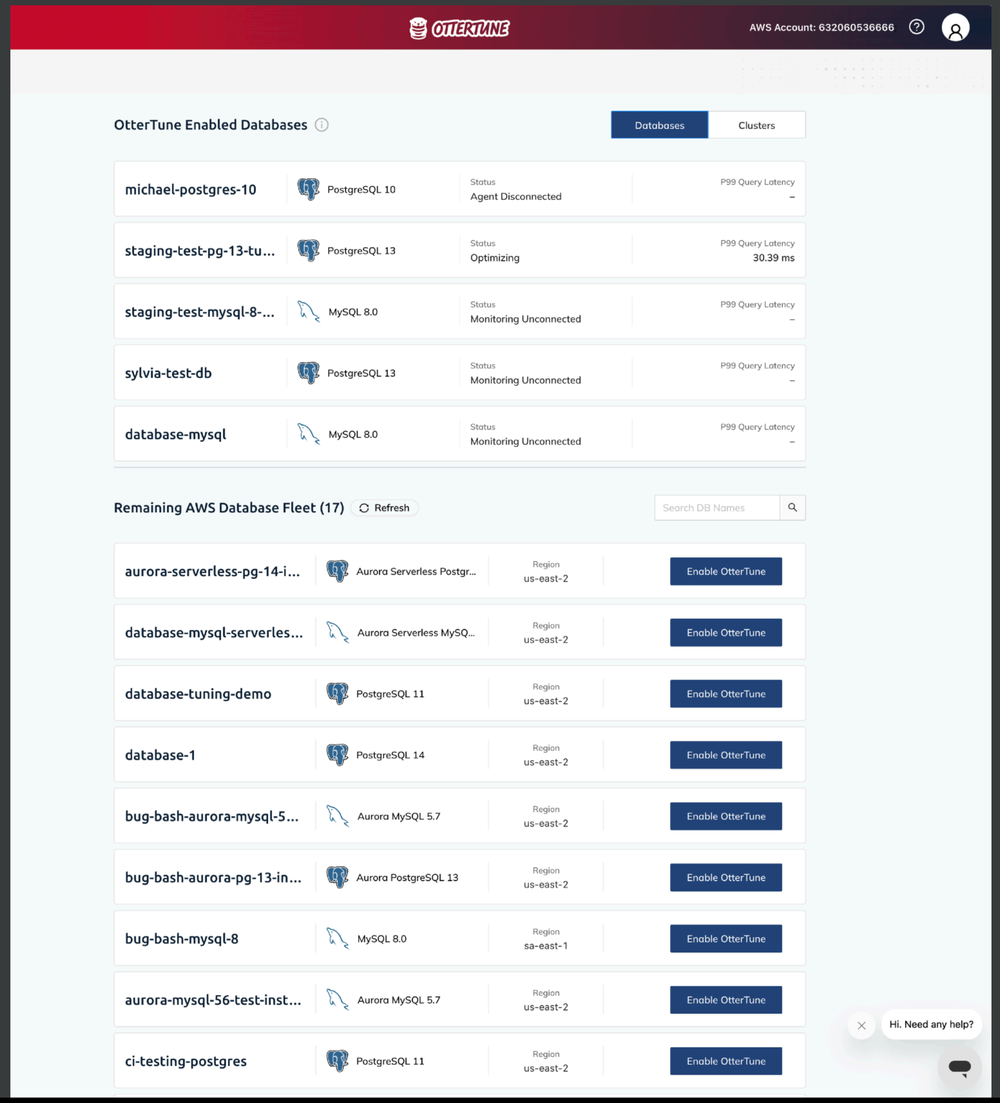


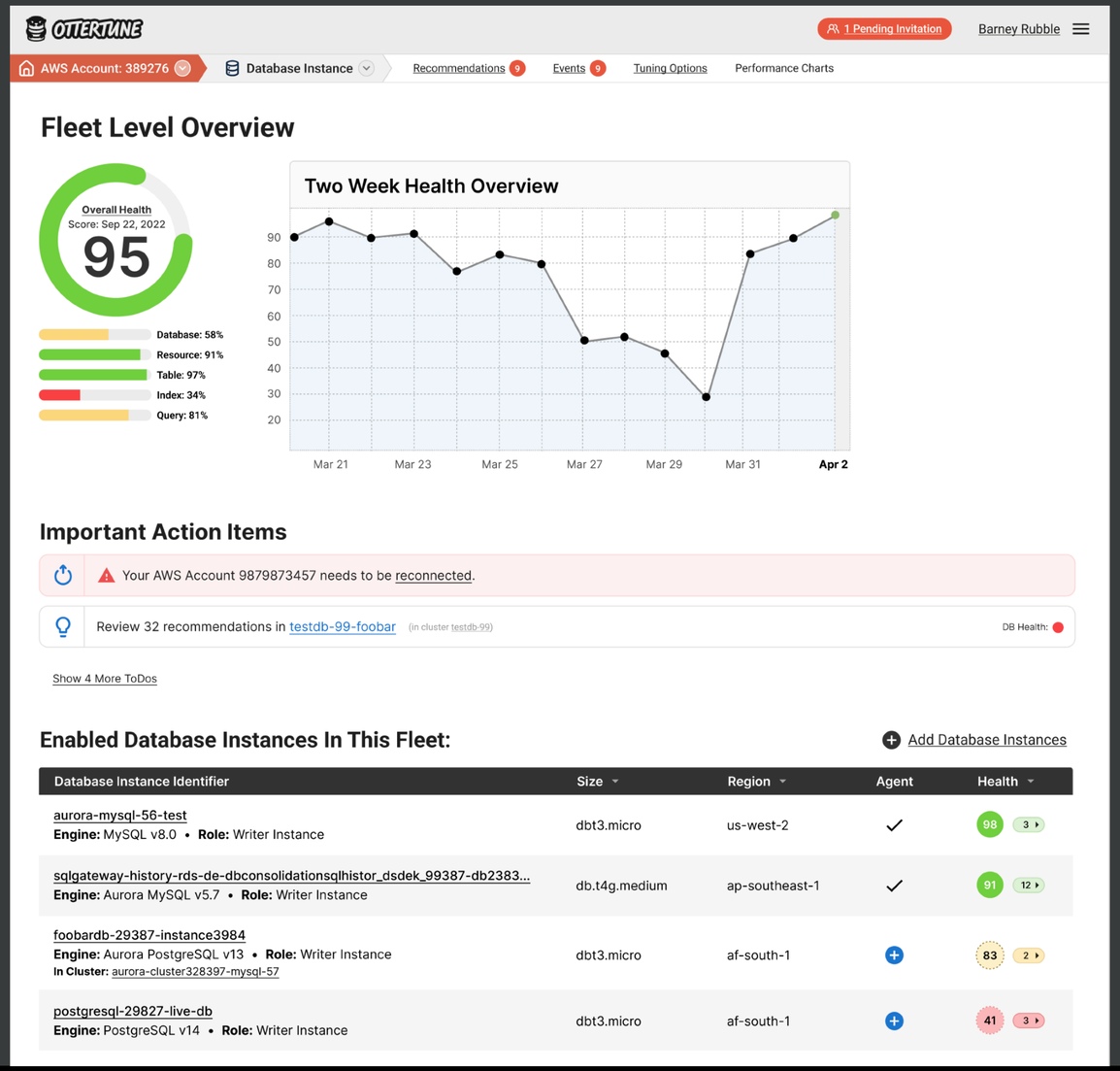
Outcome:

- **Results of Redesign**
  - **Efficiency Gains:** Users were able to complete tasks more quickly and efficiently, thanks to the streamlined interface and improved navigation.
  - **Revenue Growth:** The improved onboarding flow led to a higher success rate in user setup. Connecting a user's AWS IAM permissions improved from a completion rate of 10% to 25%, with later additional improvements bringing that to 35%.
  - **Scalability:** The redesign included a complete rewrite of the backend and frontend, making the application more scalable and easier to maintain and update.
  - **Positive Feedback From Users and Stakeholders:** The redesign received positive feedback from stakeholders, including the executive team, engineering team, and power users, validating the design decisions.
  - **Brand Refresh:** The redesign was accompanied by a site rebranding, enhancing the company's image and aligning with its new strategic focus.



Redesign Comparisons: Fleet Dashboard





Redesign Comparisons: Onboarding AWS IAM

Add IAM Role

OtterTune requires an IAM role to establish a trusted relationship with your Amazon account, so that it can collect info about your database. The role policy includes an External ID to designate who can assume the role. This is the preferred method for interacting with 3rd party accounts according to the AWS IAM User Guide. OtterTune currently only supports AWS databases, so this step is required.

You only have to create the IAM role once for your organization. Don't have the required permissions? Invite a teammate who does.

IAM Role ARN

1 Create IAM Role

To create an IAM role for usage with OtterTune service, you can either use AWS CloudFormation, or use Terraform. See our Documentation for more instructions.

Choose an IAM Setup Method

CloudFormation | Terraform

Create an IAM Role Using AWS CloudFormation

To create the IAM Role you will need the External ID (generated by OtterTune) to create the IAM Role in the next step. Copy the External ID below:

External ID
36d68077-0877-462d-bb0b-b2257dac0803 Copy

Use the 'Quick Create IAM Role' AWS CloudFormation form to quickly create an IAM Role.

If you prefer to manually create your IAM Role as opposed to generating it with CloudFormation, you can follow the steps outlined in our Documentation, then jump back to the 'Submit ARN' step on this screen (Step 3).

Quick Create IAM Role

2 Submit ARN

Once you have created the IAM Role, submit the 'Role ARN' of the created role to complete the connection. Use the following steps to copy the 'Role ARN':

1. Go to AWS IAM roles dashboard.
2. Search for the IAM Role you just created and click on the role.
3. Copy the 'Role ARN' and jump back to OtterTune to paste the 'Role ARN' below:

Enter generated IAM Role ARN.

arn:aws:iam::691523222388:role/OtterTuneRole-DanielSwartz

For help, please read our documentation and getting started guides or reach out to our slack community.
Invite otters from your team.

Connect Your IAM Role To OtterTune

To securely connect to AWS, OtterTune requires an IAM Role with read-only permissions. The simplest way to do this is to create the role automatically via CloudFormation.

What is an IAM role?

CloudFormation | Terraform | AWS IAM Console

1. Create an IAM Role from your AWS IAM Console. Specify another AWS Account ID as the trusted entity and select 'Require External ID'. If you need more instructions, click here.

691523222388 **Account ID**

6d222960-3e66-47af-9029-1616376dc0bb **External ID**

2. Add a new policy to the role with the following permissions.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Action": [
              "rds:Describe*",
              "rds:List*",
              "pi:DescribeDimensionKeys",
              "pi:GetResourceMetrics",
              "cloudwatch:Get*",
              "cloudwatch:List*",
              "cloudwatch:Describe*",
              "ce:Get*",
              "ce:List*",
              "ce:Describe*",
```

3. Name and create your role in the AWS Console.
4. Enter in your AWS account number and IAM Role Name to connect.

AWS Account Number ...

IAM Role Name ...

Welcome to OtterTune!

We look forward to working with you and providing recommendations to help you optimize your databases.

Invite Team Members (optional)

We Are ...

Serious About Security

OtterTune does not require access to application data, and what permissions you grant to OtterTune are customizable.

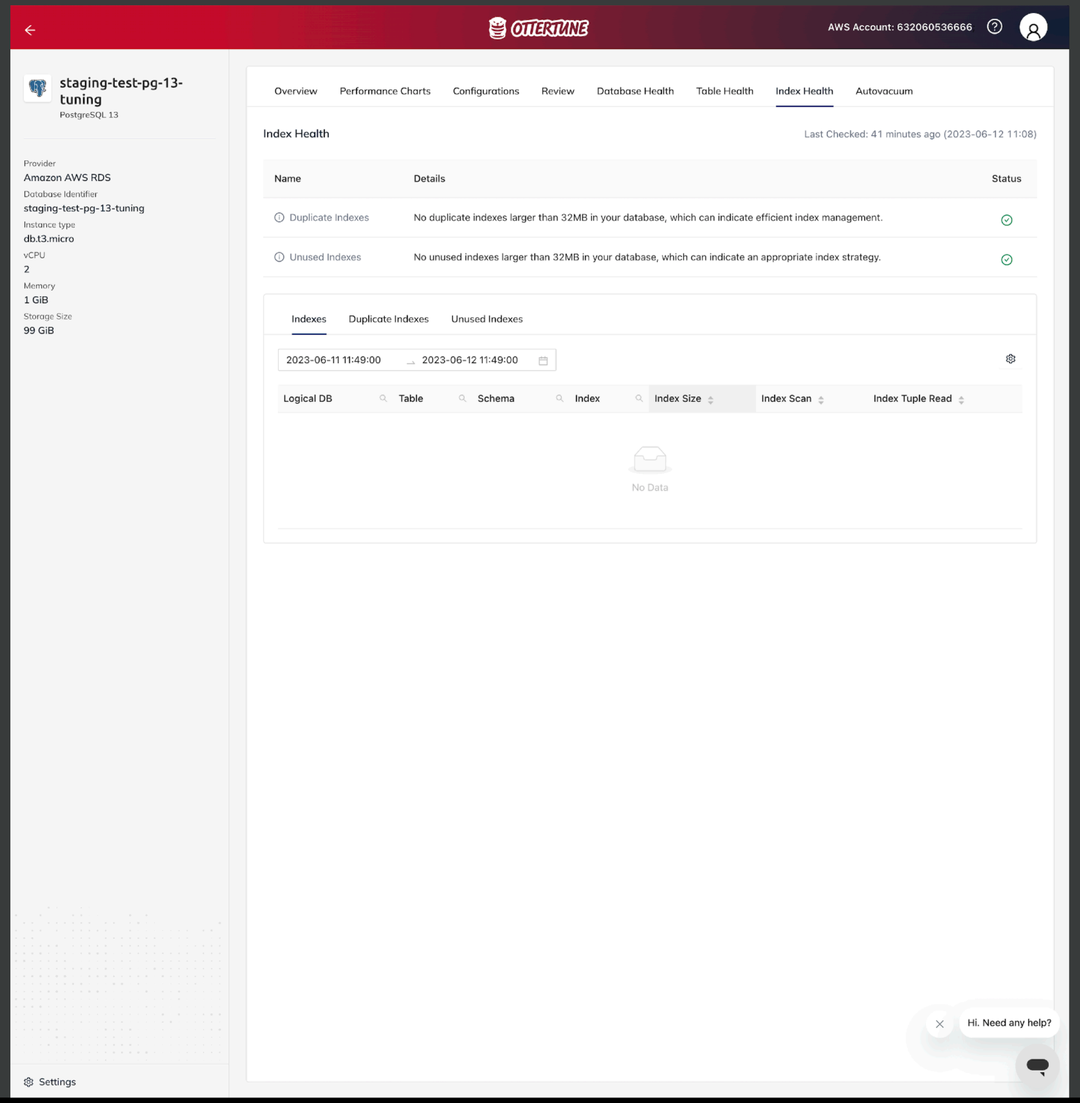
Fully Transparent

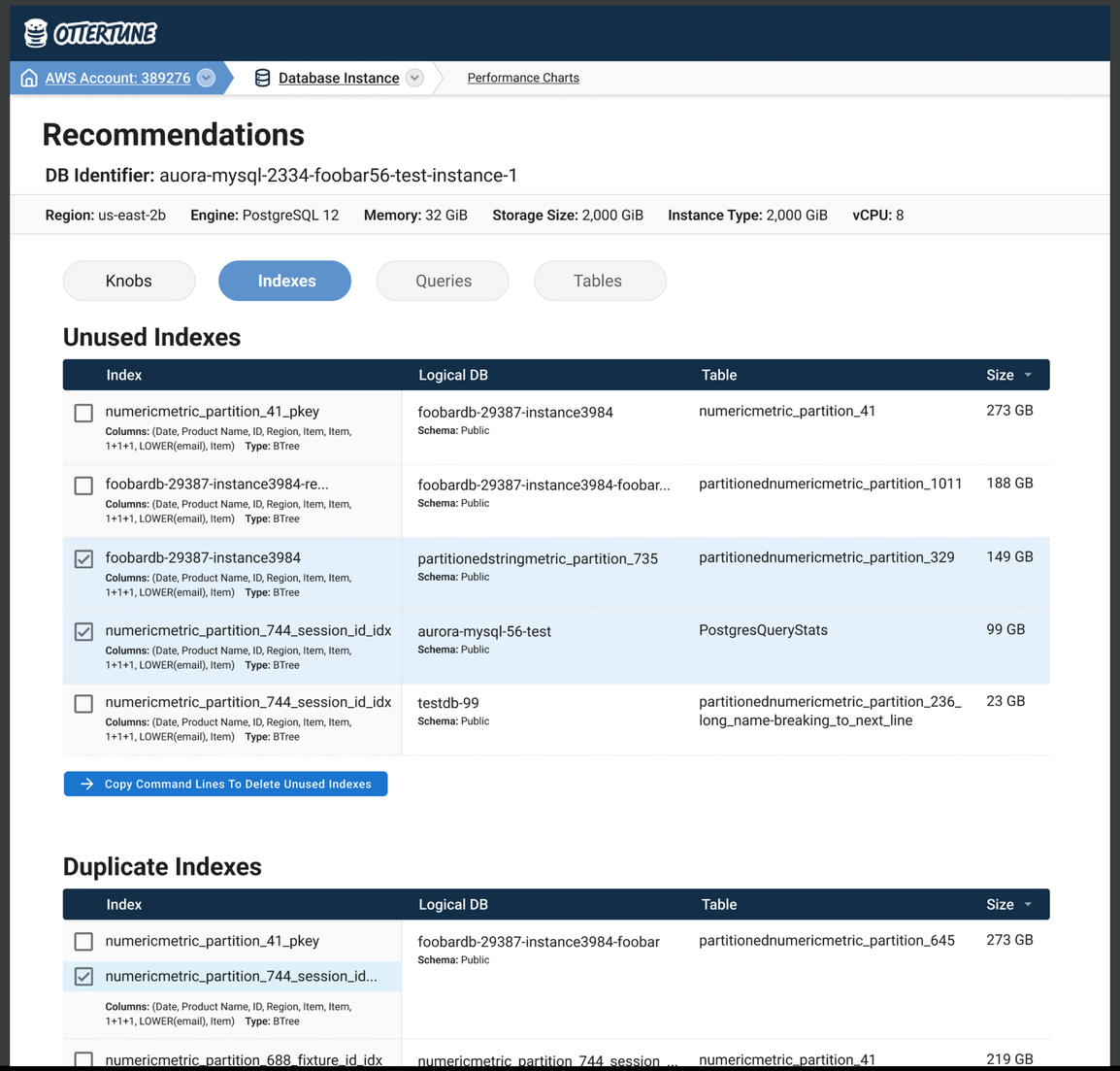
You have total control over what, when, and how OtterTune applies its recommendations to your database.

Research-Based

OtterTune was founded by database management systems and machine learning researchers from Carnegie Mellon University.

Redesign Comparisons: Index Health (originally no recommendations)





Redesign Comparisons: Email Notifications

OR **OtterTune Weekly Report <notifi...** Inbox - iCloud September 26, 2022 at 12:33 PM
OtterTune - 0 databases in your fleet need attention

OTTERTUNE

WEEKLY REPORT
September 19–26

OtterTune is monitoring 1 database

Congratulations your database passed its health check!

View your Dashboard →

Health Report:

daniel-test-database

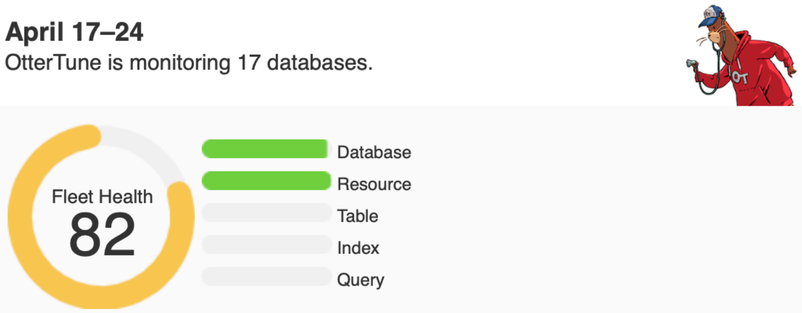
CPU Utilization

OR **OtterTune Weekly Report <notifications@o...** OtterTune April 24, 2023 at 12:30 PM
OtterTune - Your fleet score for the week is 82.
To: daniel+tester@ottertune.com,
Reply-To: support@ottertune.com

OTTERTUNE *OtterTune News*

OtterTune Weekly Report

April 17–24
OtterTune is monitoring 17 databases.



→ View Full Score In The App

Database(s) running an out of date version of the agent:

- database-demo
- bug-bash-aurora-mysql-57-instance-1
- staging-test-mysql-8-monitor
- bug-bash-aurora-pg-13-instance-1

Keeping your agent up to date is the only way to ensure you're getting the most out of OtterTune

Questions