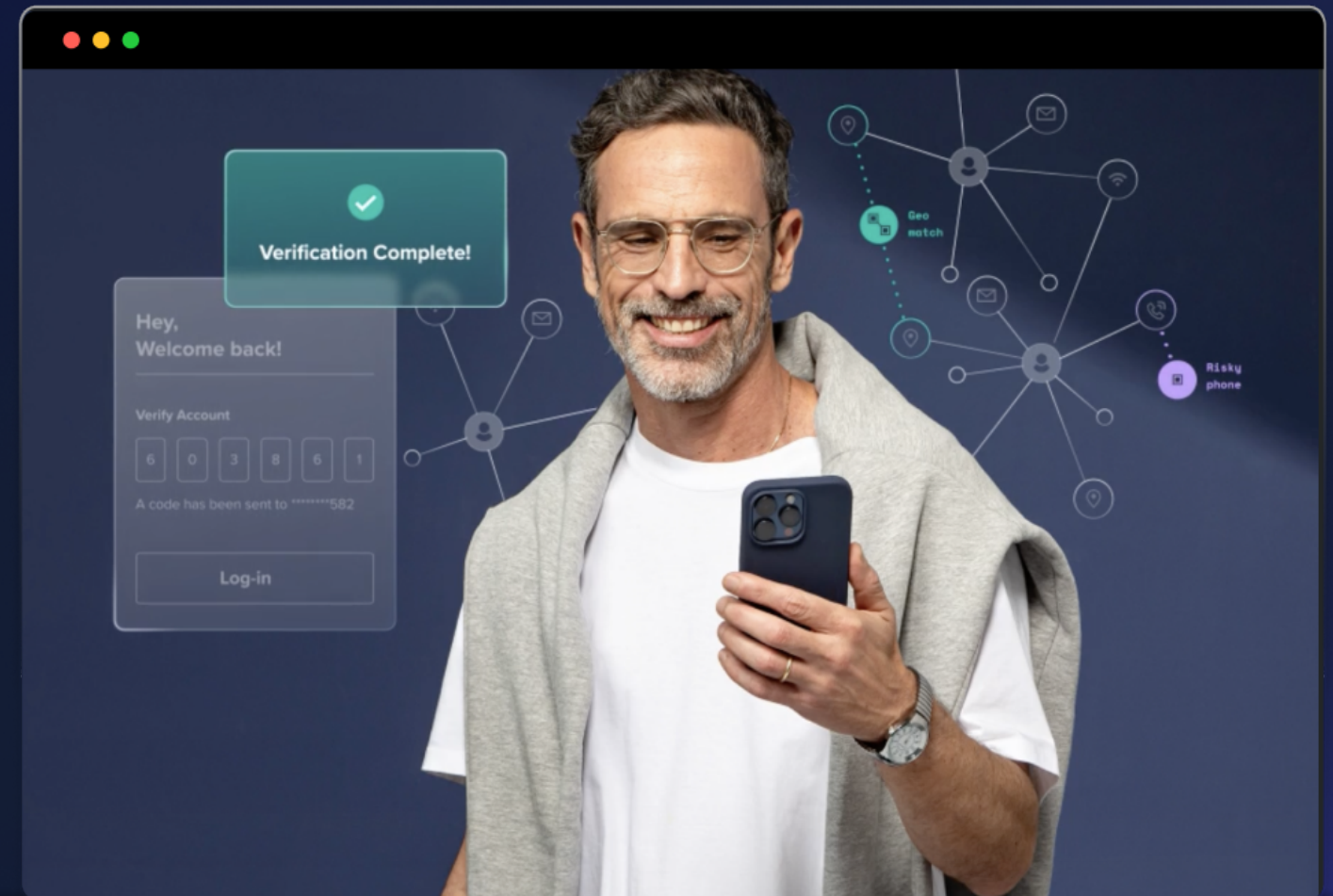


[CASE STUDY]



How Riskified Mitigated a Postgres Bottleneck with CockroachDB





How Riskified Mitigated a Postgres Bottleneck with CockroachDB

INDUSTRY :

eCommerce fraud detection

CHALLENGES :

Transaction bottleneck due to single writer limitation in PostgreSQL

SOLUTION :

Elastically scalable, PostgreSQL-compatible system of record that delivered a zero downtime migration

About Riskified

The risk of eCommerce fraud has been [growing in recent years](#): retailers' losses amounted to a whopping \$41 billion in 2022, and that number is expected to grow as fraudsters find new ways to take advantage of businesses' vulnerabilities. In service of the eCommerce community, Riskified has developed its all-in-one fraud management platform.

Since its founding in 2013, Riskified has been a pioneer in using technology as a tool to help businesses outsmart malicious actors online. [Riskified](#) now operates in 185 countries and counting and their client list includes over 50 publicly-traded companies, including Prada, Wayfair, REVOLVE, and Canada Goose. These businesses provide their various services to their own customer base, and then these businesses will send their transactions to Riskified, which will identify each one as valid or fraudulent.

In 2022, the team reviewed over \$105 billion in gross merchandise value (GMV), and today the team is excited to continue their growth. But a few years ago, the Data Platform Engineering team identified a performance bottleneck that would have prevented their business from reaching its potential.

Watch : [How Riskified used CockroachDB to mitigate a Postgres Bottleneck](#) ▶

10K

TRANSACTIONS PER SECOND
(TPS)

\$105B+

GROSS MERCHANDISE
VALUE REVIEWED IN 2022

185

COUNTRIES IN OPERATION



A single writer limits transactions

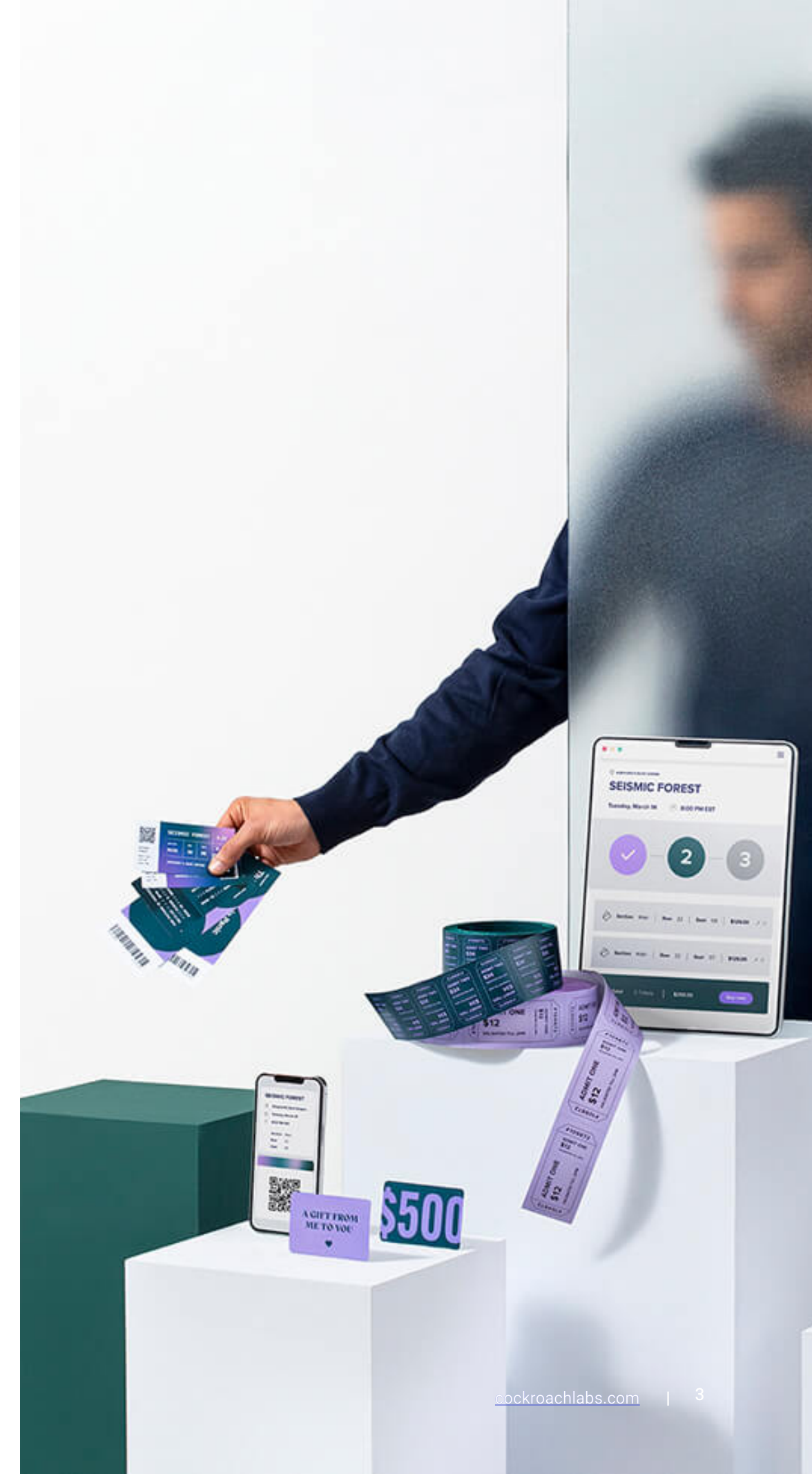
Fraudulent transactions put revenue and profits at risk for any business, and that's why Riskified's customers trust them to help protect their data, including personally identifiable information (PII). With a 99+% customer retention rate, Riskified's team knew they had found product-market fit and wanted to keep evolving their product to further support their clients.

A few years ago, the Data Platform Engineering Team had a complex tech stack, including several databases like Aurora Postgres and RDS. As the company grew, and the data grew, the team faced the following challenges:

- ✓ Managing a de facto transaction limit with a single writer server
- ✓ Breaking up a monolithic system into multiple clusters to support more traffic
- ✓ Managing patches and database versions across multiple clusters
- ✓ Traffic reaching the cluster limit, resulting in transactions needing to be flushed

The key limitation that spurred the search for a new solution is common in legacy relational databases: a single writer. Since all writes were served by a single instance, there was a limit on the number of transactions that could be received per second. This issue primarily affected Riskified's chargeback-guaranteed eCommerce application which is central to their platform.

As with their own product, the Data Platform team took a rigorous, methodical approach towards evaluating potential solutions. The team examined a variety of modern database solutions, including Yugabyte and CockroachDB.



Beyond long-term scalability, Harel Safra, Data Platform Engineering Team Lead, shared that Postgres compatibility was the most important factor to minimize adjustments made to the customer-facing application. This would save engineering time, and ensure a successful migration. Ideally, the solution would also provide more automated features, such as server and node replacement and backups.

Additionally, security and data integrity were of the utmost importance both due to the sensitive nature of the data stored, as well as the responsibility Riskified owes to its customers if data transactions are incorrectly marked as valid when they are fraudulent.

In summary, the team evaluated various databases on the following criteria:

- ✔ Postgres compatibility to minimize changes to their application
- ✔ Guaranteed security given the personally identifiable information (PII) stored
- ✔ Highly efficient query delivery in a scalable, resilient system

Safra's team put CockroachDB and other databases through a rigorous proof of concept, which included testing different environments, loading in half of the production data, and bombarding the database with queries. With measurements from [Apache JMeter™](#), the engineers found that CockroachDB was able to perform as closely as possible to their existing PostgreSQL database. In addition, the team saw the benefits of CockroachDB's elastic scalability and resilience against availability zone (AZ) failure.

“ ”

“I heard about CockroachDB at the beginning of my time at Riskified, and originally I didn't understand why we used a database with such a name. But now I understand the reason – because it survives. It can't be killed, even though I tried to kill it myself.”

– **Yoav Shemesh**

Senior Data Platform Engineer, Riskified

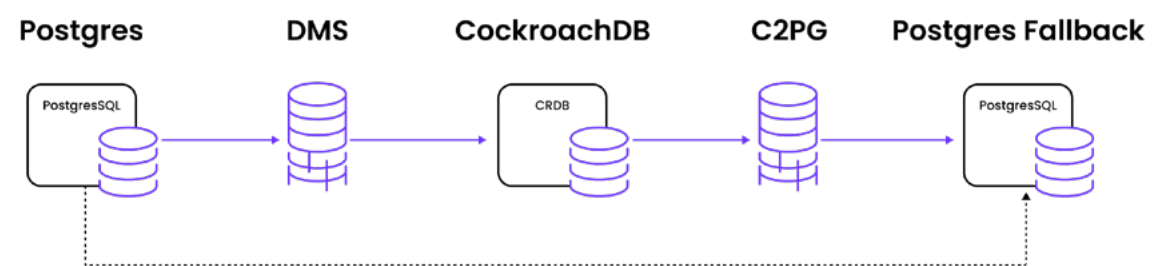


Zero-downtime migration

In order to actually migrate the data from the existing infrastructure to CockroachDB, Yoav Shemesh, Senior Data Platform Engineer at Riskified, shared the importance of a no-downtime migration. Since Riskified's client-facing application runs 24/7, the team needed the guarantee of a [zero-downtime solution](#) – lost time in the eCommerce space can mean lost revenue or, even worse, lost customers.

First, the team started by creating CockroachDB instances, including the schemas, users, and scripts. Then, the engineers used [AWS Data Migration Service \(DMS\)](#) and [CockroachDB Replicator](#) to move the data.

Migration with AWS DMS and Replicator (belt & suspenders)



1. Create schemas, users
2. Full load + CDC with AWS Data Migration Service
3. Create a copy from source
4. CDC using C2PG (partitioned tables need custom script)

In addition, Shemesh and his colleagues used Cockroach Labs' [Change Data Capture \(CDC\)](#) feature to replace Debezium and ensure that ongoing changes to the data were replicated properly. Lastly, the team set up a clone of the production clusters, and implemented custom scripts from CockroachDB's professional services team to ensure any issues could be resolved as quickly as possible.

Today, Riskified is [self-hosting CockroachDB on AWS](#) autoscaling groups (ASGs) in a single region. Although the chargeback guarantee application is Ruby-based, there were many small microservices that were written in other languages, which were also supported by CockroachDB. Riskified's CockroachDB instances handle over 10,000 online transactions per second.

“ ”

“The main benefit that we’re seeing with CockroachDB is around scale. We can scale up and we can scale down. We have elasticity in the infrastructure to allow us to serve whatever application needs in case demand changes. For example, if there’s a busier season, we can scale up, and then we can scale down after to save money.”

– Harel Safra

Data Platform Engineering Team Lead,
Riskified



Details and documentation

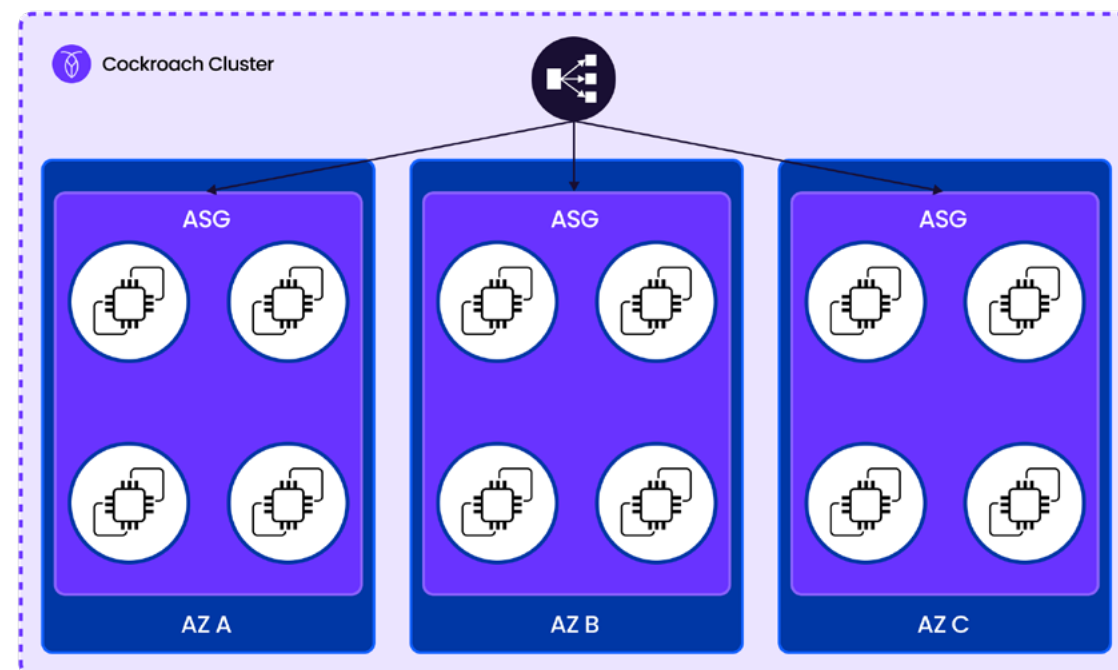
The Data Platform Engineering team was happy to share that the migration went very smoothly, and also wanted to share some tips and tricks, particularly for anyone planning their own database migration or working with Ruby-based applications.

With regard to their migration, the Riskified team found value in CockroachDB's cloud migration tool – [MOLT Fetch](#), but they suggest that other engineers pay attention to any preexisting `IDENTITY` columns in the database, as those should be replaced with CockroachDB's built-in functions like `unordered_unique_rowid`, to prevent hot shards.

For those building Ruby-based applications, the engineering team shared that Ruby on Rails works out-of-the-box on CockroachDB when using the Postgres adaptor, but they cautioned that it is important to use the SSL flag for production deployment.

The team shared many more [technical tips and tricks](#) with us, but their most important tip was to reference the [CockroachDB documentation](#), and to leverage our support as needed, for custom solutions or any questions you may have for your unique tech stack and use case.

Easy management using ASG per AZ



“ ”

“Read the documentation because most of the things that you’ll be hung up on are all written there. Additionally, the CockroachDB support team is great.”

– Yoav Shemesh

Senior Data Platform Engineer, Riskified



Scaling to safeguard eCommerce businesses

Looking into the future, Riskified as a company has plans to continue growing their core product offerings as well as to expand into more regions and capabilities, such as preventing threats like claims abuse, voucher abuse, and account takeovers. In support of these goals, the team at Riskified will be moving additional systems from Postgres to CockroachDB.

Check out the [Riskified website](#) to learn about their platform, fraud prevention strategies, and more exciting updates on how they are protecting businesses.

Ready to get started?

Go hands-on with 100% free CockroachDB Serverless. Spin up your first cluster in just a few clicks.

