AWS AI AND DATA CONFERENCE 2025

100K Unique short links in 1 second:

Concurrency & DynamoDB in action

rebrandly.com

Index

Abstract

High-volume, high-velocity short link generation is the focus of this whitepaper. Rebrandly developed a solution to quickly create a massive amount of personalized links for SMS. The result surpassed our initial goals and led us to develop an infrastructure that can be embedded into platforms that require dynamic, unique link creation, such as marketing clouds, social media tools, agency solutions, and enterprise systems. Rebrandly Embedded provides a reliable, scalable, and performant architecture capable of generating hundreds of thousands of links per second. We delve into the technical details of leveraging AWS services, particularly DynamoDB, alongside Golang, Node.js, and Elasticache. We explore the problem to be solved, initial challenges, the iterative development process, architectural decisions, and resulting impact, with insights into building a resilient and efficient product for demanding enterprise applications.

Introduction

Rebrandly’s expertise in reliable **link generation at high volumes** attracts enterprises relying on efficient push solutions. Like IoT, eCommerce, and Fintech, **Communication Platforms as a Service (CPaaS) require a critical infrastructure layer for high-performance connections** between physical devices, data streams, and user interfaces at scale. Our recent collaboration with **WONDERCAVE** a Stagwell Marketing Cloud company and leading text messaging automation platform, led to a solution capable of generating millions of ready-to-use unique links with a single API request.

Using DynamoDB, Go, Node.js, and Elasticache, **we demonstrated the strength of a values-based velocity** that exemplifies the AWS Well-Architected principles of reliability, security, efficiency, and cost-effectiveness. Our achievement—Rebrandly Embedded—is now a proven solution that resulted in a 60% increase in click-through rates for our customers' clients. Successful implementation relied on technical expertise, a thorough understanding of AWS services, and **collaboration** with Wonder Cave, AWS support, and the dev community at large.

The goal:
generate high-volume, personalized links at scale

Wonder Cave is a leading peer-to-peer (P2P) and application-to-peer (A2P) messaging automation company. They approached us with a challenging task: create an embedded solution capable of generating millions of unique, personalized short links across hundreds of domains for SMS campaigns within 30 days. The key requirements were:

- Generate **100,000 links within ten seconds** (ultimately achieved in one)
- Provide **1:1 personalization** for each recipient
- Enable **dynamic link tracking** and **analytics**
- Ensure **robust architecture** for clean data transfer and **scalability**

Referred from another Rebrandly customer, Wonder Cave selected our engineering team for the project due to the API's proven robustness and reliability. This collaboration highlights the **growing importance of dependable link infrastructure for businesses seeking seamless customer interactions and efficient data flow.**

Iterating towards a solution

Initial attempts to solve the problem involved synchronous DynamoDB queries, generating short links without a destination, and batch operations. However, these approaches had limitations. Synchronous queries were too slow for the required throughput. **Short links without an assigned destination improved initial response times but did not address the scalability issue.** BatchWriteItem operations were efficient for smaller batches but were not as successful as TransactWriteItems in providing data integrity and uniqueness. Preventing duplicate links was also a concern.

We first attempted direct DynamoDB queries to generate 100,000 slashtags and used GetItem to check for uniqueness. The individual queries and checks resulted in high costs and slow response times, respectively. Another idea was to use SQS/SNS and Lambda to generate the links, return them to the client, and then insert records asynchronously into DynamoDB. The risks of collisions and broken or incorrect links were too high with this solution. **Pre-generating short links and assigning them dynamically left us with an unclear pre-allocation strategy, and managing multiple domains was challenging.**

We concluded that **DynamoDB and the Transaction API would ensure unique insertions and avoid duplicate links. We could insert 100 records per transaction using the Transaction API, guaranteeing uniqueness and consistency.** We also had support from the AWS team to increase our account-level write throughput limit.

Building Rebrandly Embedded

AWS SDK for Go's high concurrency through multi-threaded processes was invaluable. ECS workers were used to process chunks of short links for DynamoDB insertion, giving us parallel processing, scalability, and the ability to distribute the workload for improved scaling. Elasticache served as an efficient temporary buffer for batch processing, reducing the load on DynamoDB and improving overall performance.

Another part of the challenge was to overcome a hot partition problem. We implemented a Global Secondary Index (GSI) and designed the partition key using workspace_id + a random uuid4. This approach distributed the load evenly across partitions, preventing any single partition from becoming a bottleneck. Load distribution was crucial to achieving high throughput and low latency.

Solution details

1/4

DynamoDB and Transaction API

We chose DynamoDB for its scalability, performance, and managed nature. Transaction API was important for ensuring uniqueness and consistency during high-volume writes. Each transaction could insert up to 100 records, guaranteeing that all or none of the records were written. This was essential for preventing duplicate links and upholding data integrity.

Solution details

2/4

Go and consistency

Go provides concurrency and efficiency, handling link generation and database interaction logic.

Go's concurrency model, based on goroutines and channels, was ideal for this use case. Goroutines are lightweight threads managed by the Go runtime, allowing for high concurrency without the overhead of traditional threads. **The Go API with Goroutines provided a low memory footprint and multi-threaded execution for the high concurrency.**

Go enabled the development of highly concurrent applications that could efficiently process large volumes of data. The multi-threaded execution model allowed for parallel processing, maximizing the utilization of available resources.

Solution details

3/4

ECS workers and scalability

We used ECS workers to **distribute the workload and scale horizontally**. ECS allowed for the deployment and management of container applications, making it easy to **scale up or down on demand**.

Each ECS worker processed a chunk of short links for DynamoDB insertion. This approach allowed for parallel processing, reducing the overall processing time and increasing throughput.

The ability to scale the number of ECS workers dynamically ensured the system could handle high loads.

Solution details

4/4

Elasticache & Buffering

Elasticache in-memory data store was used as a temporary buffer for batch processing. This reduced the load on DynamoDB and improved overall performance. Elasticache’s fast read and write speeds made it ideal for buffering data before writing it to DynamoDB. This approach reduced latency and improved responsiveness.

Ultimately, **the large task of generating millions of links was broken down into smaller, manageable chunks** to facilitate efficient scaling and distribute the workload evenly across resources. It also made the system more resilient, as any individual worker failures would only affect individual parts of the larger task.

Results: growth without limitations

The successful development and launch of Rebrandly Embedded demonstrate the effectiveness of this approach. On average, **we handle the demands of 200 million monthly links and 588 million at peak**. The changes we've implemented will enable us to **grow that number without limitation**.

Wonder Cave's satisfaction also validates our efforts. Now, Wonder Cave can offer its clients link customization and dynamic link tracking at scale. 97% of the campaigns that run through their system now use Rebrandly generated links. Rebrandly Relay™ is now a fully automated messaging solution that drives superior performance and actionable insights for text campaigns. It is also a testament to the effectiveness of growth-minded partnerships and our underlying architecture's ability to deliver real-world results.

The achievement was building a high-performance, reliable, and scalable service through teamwork, leveraging AWS, and the expertise of talented engineers. This project demonstrated the importance of a thoughtful, well-architected approach.

Why Rebrandly

Rebandly's robust API, reputation for reliability, extensive experience, security, and SOC2 compliance were crucial factors in making it the right platform for the project. These attributes draw enterprise engineers, CTOs, and business leaders of all kinds to our platform, and we work hard to ensure that every solution meets their current and forward-looking enterprise-grade requirements.

Through our engineering innovation and thought leadership in building scalable and responsive systems, we've proven that we can meet customers' needs to execute their business goals and drive differentiated innovation and growth strategies.

The Rebrandly API and Rebrandly Embedded are key to delivering effective link infrastructure for enterprise customers needing millions of customized links at scale.

Discover how Rebrandly Embedded can transform your tech stack. Whether you need to generate thousands or millions of smart links, our scalable architecture delivers unmatched performance and reliability.

Learn more about Rebrandly Embedded or contact our team. Our engineers are ready to work with you to build a customized solution that meets your current needs and supports your future growth.

Speakers

Gonzalo Rios

DevOps Lead @Rebrandly

Alex Ischenko

Senior VP of Engineering @Rebrandly

Contacts

- rebrandly.com
- Partner with us
- Contact sales
- LinkedIn
- Facebook
- Instagram
- Youtube