



Digital bank Nubank migrates to CockroachDB to keep pace with rapid growth and improve infrastructure resiliency

Nubank migrated its credit card authorization service from in-memory storage to CockroachDB to achieve scalable, always-on infrastructure and more efficient operations.

Overview

Nubank, the leading financial technology company in Latin America, sought a scalable SQL database for several critical services. The fintech initially stored data on-prem using an in-memory data store, but as the business rapidly grew, it needed a transactional solution for some applications. Nubank's team chose CockroachDB due to its ultra-resilience, simple scalability, strong consistency, and easy maintenance. They set up on-prem CockroachDB deployments and migrated their credit card authorization application. As the company continues to expand into new markets, they may migrate some apps to Amazon Web Services (AWS) and create a hybrid CockroachDB deployment that spans on-prem servers and cloud regions.

“

We have a highly complex transaction environment where we are continually searching for simplicity, scalability, and robustness

—Gabriel Melo, Engineering Manager at Nubank

”

Challenge

Nubank is a Brazilian digital bank. Founded in 2013, it quickly grew to become the largest independent digital bank in the world, with over 13 million customers. Nubank was built to address Brazil's excessively high borrowing rates and poor banking customer experience. The company offers fee-free credit cards, digital savings accounts, and personal loans, which are all accessible from mobile devices.

As Nubank's business grows and its customer base expands into Latin America - the company announced recently the launch of new offices in Mexico and Argentina - it must also evolve its application architecture. The digital bank built the first iteration of its infrastructure with Java Heap using an in-memory data store. However, this infrastructure was operationally complex and poorly scalable. For example, the company's credit card authorization app (the service that decides whether or not to approve transactions) was spread across four instances of Java Heap, and a Kafka message communicated data changes between the instances. The team found that Java Heap was not sufficient to store this kind of data.

Nubank has initially deployed CockroachDB on-prem in three locations, and if they choose to migrate to the cloud, CockroachDB's flexibility and scalability will allow them to seamlessly transition.

"We have a highly complex transaction environment where we are continually searching for simplicity, scalability, and robustness," said Gabriel Melo, Engineering Manager at Nubank. The team decided they needed a transactional SQL database and began evaluating options. Their goal was to migrate one service at a time, starting with their authorization app.

Requirements

Nubank's team had several requirements for their new solution. Resiliency was important, as they need to provide always-on service for their customers. They also required a horizontally scalable database that could meet a guaranteed level of consistency, in order to support their large volume of transactions.

As Nubank is always looking for simplicity, it wanted its new database to improve the efficiency of its developer and infrastructure teams. To increase productivity, the database ideally would be a SQL database that supports standard drivers. The bank also sought a solution with minimal operational upkeep.

Solutions

Nubank's team evaluated multiple databases, including Cassandra and CouchDB. Ultimately, they decided that CockroachDB best met their requirements. "In our environment, CockroachDB brings us the confidence and simplicity to follow Nubank's growth," said Melos. The database's combination of consistency and scalability provides a solid foundation for Nubank's credit card authorization system and for their future expansion.

CockroachDB was built to support transactional workloads like Nubank's financial ledger. It achieves strong consistency and guarantees the highest level of isolation (serializable isolation). At the same time, CockroachDB scales automatically. Data is broken into 64-megabyte chunks called ranges that distribute and balance across nodes. If a user wants to add an additional node, he or she simply spins it up and points it at the cluster. Nubank has initially deployed CockroachDB on-prem in three locations, and if they choose to migrate to the cloud, CockroachDB's flexibility and scalability will allow them to seamlessly transition.

Resiliency is also built into CockroachDB, helping Nubank achieve the always-on service it requires. CockroachDB replicates all ranges three times and stores the replicas in a way that maximizes geo-diversity. The replicas are spread across nodes of a cluster, which together forms a single logical database. Depending on the cluster's topology and how many machines and regions are involved, the replicas could be located across multiple regions, datacenters, machines or all of the above. This replication ensures survival in the event of failure.

In addition, CockroachDB increases the productivity of Nubank's developer and infrastructure teams. Its standard SQL interface, which is PostgreSQL wire compatible, allows developers to communicate easily with the database. The infrastructure team has benefitted from CockroachDB's automated backups, automated garbage collection, and rolling upgrades. These features give them more freedom to focus on tasks like maintaining the hardware of their on-prem servers.

Results and What's Next

CockroachDB is now the mission-critical database backing Nubank's credit card authorization system. It provides the availability, scale, consistency, and simplicity the digital bank needs. The company is evaluating additional apps to run on CockroachDB in the future. As the company continues to grow and expand internationally, they may migrate some data to AWS. In this case, they would run CockroachDB in a hybrid deployment across on-prem datacenters and cloud regions supported by AWS or any other cloud vendor.