



CASE STUDY

European electronic lock manufacturer modernizes its identity access management system with Managed CockroachDB

An electronic lock manufacturer chose to run its identity access management platform on Managed CockroachDB due to its resiliency, regulatory compliance, and low operational lift

Overview

A European electronic lock manufacturer was looking to modernize its identity access management system by migrating from Microsoft SQL Server to a scalable database that did not require manual sharding. The manufacturer has customers across the globe that need always-available, low-latency reads. As a European company, they also require a database that complies with the EU's General Data Protection Regulation (GDPR). They turned to Cockroach Labs' database-as-a-service (DBaaS) offering, Managed CockroachDB, for its strong resiliency, scalability, and regulatory compliance. Cockroach Labs deployed a global cluster, which the manufacturer configured to pin data in specific locations. This geo-partitioning allowed the team to meet their performance and regulatory-compliance goals. Managed CockroachDB is now a critical part of the application's infrastructure, and the manufacturer's small engineering team has more free time to handle development.

Challenge

A European electronic lock manufacturer wanted to modernize the infrastructure of its identity access management system, which oversees admission to hospitals, airports, retail locations, and high-security facilities. The application relies on storing sensitive data from customers around the globe in a transactional database. Many use cases are time-sensitive (e.g., hospital room access) and require always-available records (e.g., military base access), making both low-latency reads and database resiliency important.

At the time the team decided to modernize the app, it was built on an on-premise deployment of Microsoft SQL Server. In the past they had also used MySQL. Both legacy databases required manual sharding, a practice the team found painstaking and labor-intensive. They wanted to move away from sharding and evolve the app's architecture from monolithic to microservices. Since they didn't have a site reliability team, and they didn't want to sink time into cloud operations, they decided to explore database-as-a-service (DBaaS) solutions.

Requirements

The team had multiple requirements for their new DBaaS. First, they needed a resilient solution that could survive datacenter or regional failures. In the event of a regional failure, they needed their database to remain available so customers could access important facilities. The team also wanted to scale globally to reach its widespread customer base, but they needed low-latency reads for their time-sensitive use cases. In addition, they wanted to achieve this global scale without manual sharding.

Another requirement was compliance with data domiciling laws, such as GDPR. The manufacturer has customers across the European, Asia, and North America, and they needed to ensure European customer data remained in Europe. If possible, they wanted to pin row-level data to individual countries. Finally, they needed the ability to stream a log of their transactional updates into their Elasticsearch, Logstash, and Kibana stack as an event bus for microservices.

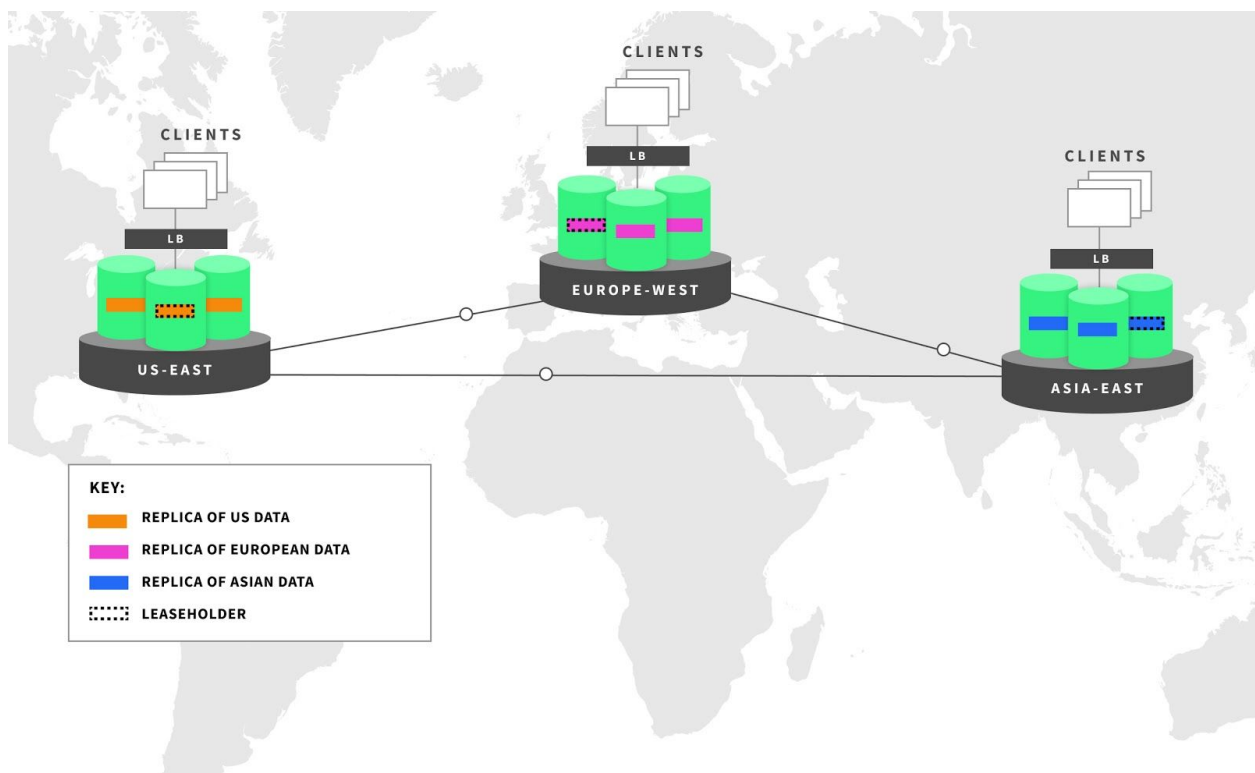
Solution: A Global Managed CockroachDB Deployment

The manufacturer evaluated various solutions, including Google Spanner and Amazon Aurora, and they decided Managed CockroachDB best met their requirements. The team is also a proponent of Golang and open source projects, and they were impressed by the quality of review processes and code they found on the CockroachDB repository.

Managed CockroachDB is a single-tenant, secure, fully managed and hosted service that runs Enterprise CockroachDB in Amazon Web Services (AWS) or Google Cloud Platform (GCP). The Managed CockroachDB team removes the operational complexity of setting up and maintaining a distributed database, so that application teams can focus on building

their business applications. Since removing operational burden was a key criteria for the manufacturer's team, using the hosted solution of CockroachDB was a no-brainer option.

Cockroach Labs created a global Managed CockroachDB deployment to meet the manufacturer's goals. The Cockroach Labs team set up a total of nine nodes, spread across three regions: US-East, Europe-West, and Asia-East. Within each region, there are three different availability zones that each house one node. The manufacturer's team then used a feature called [geo-partitioned replicas](#) to pin row-level data to specific regions; for example, European customer data is pinned to Europe-West. No additional work or manual sharding was required to create this global deployment, since Managed CockroachDB automatically scales and breaks data into sections called ranges. If the manufacturer decides to expand into new regions, scaling will be as simple as spinning up new nodes and pointing them at the cluster.



The manufacturer's Managed CockroachDB cluster uses the geo-partitioned replicas configuration, where all replicas for a set of data are constrained to a region, and each replica is pinned to separate datacenter.

This deployment, combined with CockroachDB's inherent resiliency, allowed the manufacturer to meet their requirement of an always-available database. By default, Managed CockroachDB replicates data three times and stores the replicas on machines that maximize geographic diversity. For the electronics manufacturer, all three replicas of

data are pinned to the same geographic region (depending on where the customer is located), and then the replicas are distributed among the three availability zones in that region. This means that even if one zone fails, two additional copies of data remain accessible in the other zones.

With their scale and resiliency goals met, the manufacturer wanted to ensure it could achieve local read performance and comply with GDPR. The geo-partitioned replica feature allowed the team to meet both goals. By placing data in the nearest region to the customer, Managed CockroachDB achieves low latencies and remains within the regional jurisdiction.

Finally, the team used CockroachDB's [change data capture](#) to stream their transactional data from Managed CockroachDB to their search system (ElasticSearch) thus completing this stack.

Results and What's Next

Managed CockroachDB is now a critical part of the manufacturer's user access control system around the world. With their global deployment, the team has achieved the availability, scale, and read performance they needed. Managed CockroachDB is also a key part of their regulatory compliance story. If the team identifies the need to expand in the future, their database can scale out easily to handle the demand. Most importantly, Cockroach Labs handles the database's day-to-day operational tasks, meaning that the company's small engineering team is now free to develop applications.