



Why Squarespace adopted CockroachDB for better global performance at scale

Gavin McQuillan
Senior Staff Software Engineer





What is Squarespace?

Domains

Websites

Marketing

Commerce





Agenda

- 01 Breaking up with Postgres**
- 02 The Great Migration**
- 03 Lessons Learned**
- 04 Closing Remarks**



Breaking Up With Postgres

Scale



Running low on Disk

A satellite image of a tropical cyclone, showing a well-defined eye and spiral cloud bands over a vast expanse of the ocean. The image is taken from a high altitude, providing a top-down view of the storm's structure. The ocean surface is visible as a textured blue area, and the white clouds of the storm are the dominant feature in the center and surrounding areas.

Scale

Need improved resilience

Scale

Need to authoritatively serve traffic from multiple continents



Operational complexity

Postgres requires a flotilla of supporting services





Operational complexity

- Postgres (the datastore [v9.5,10,11,14, ...])
- Repmgr (replication management/failover)
- Postgres-exporter (monitoring)
- PgBackrest (backups and archival)
- PgBouncer (connection pooling)
- PgHero (Query analysis, perf tuning)
- Debezium (Change Data Capture)

An aerial photograph of a building's floor plan, showing a complex layout of rooms and corridors. The image is overlaid with two semi-transparent white text boxes. The top box contains the text 'Operational complexity' and the bottom box contains the text 'Not always easy to see what's going on inside the cluster'. The floor plan shows various rooms, some with 'EXIT' signs and room numbers like '2'. The walls and floor are shown in shades of grey and brown.

Operational complexity

Not always easy to see what's going on inside the cluster

An aerial photograph of a port terminal. The image shows a dense grid of shipping containers in various colors (blue, red, orange, white) stacked in neat rows. A yellow gantry crane is visible in the lower right, positioned over a stack of containers. The ground is marked with white lines and numbers, indicating different sections or lanes. The overall scene conveys a sense of organized industrial activity and complex logistics.

Operational complexity

With PG, Upgrades are complex logistically, usually requiring extensive downtimes

Why Not Managed Postgres?

Feature	LOE: CRDB	LOE: Managed SQL
Scaling	L	M
Migration Rollback	M	H
Regional Failover	L	H/M*
Geo-partitioning	L	N/A
Upgrades	L	H**

* with advanced DR

** mandatory hours of downtime per upgrade

L = low effort
M = medium effort
H = high effort

CockroachDB decision

An aerial photograph of a bridge under construction. The bridge features tall, grey concrete piers. A large yellow gantry crane is positioned on the bridge deck, spanning across several piers. The surrounding landscape is a mix of green fields and brown earth, with some buildings visible in the distance. The sky is clear and blue.

Horizontal scaling built-in

CockroachDB decision

Amazing console to introspect
what's happening inside clusters





CockroachDB decision

**Straight forward operation
patterns: online upgrades, online
schema changes**



The Great Migration

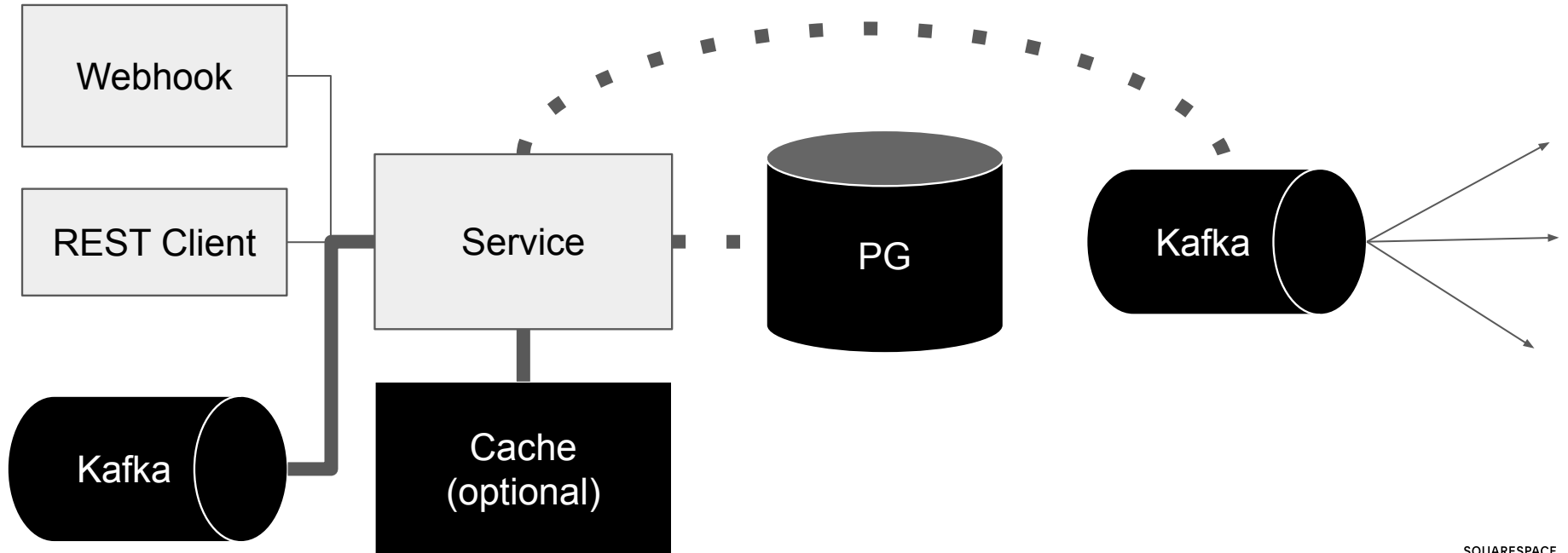


The Great Migration: Numbers

50 cluster to migrate
X 2 environments
= 100 total cutovers



The Great Migration: Average Microservice



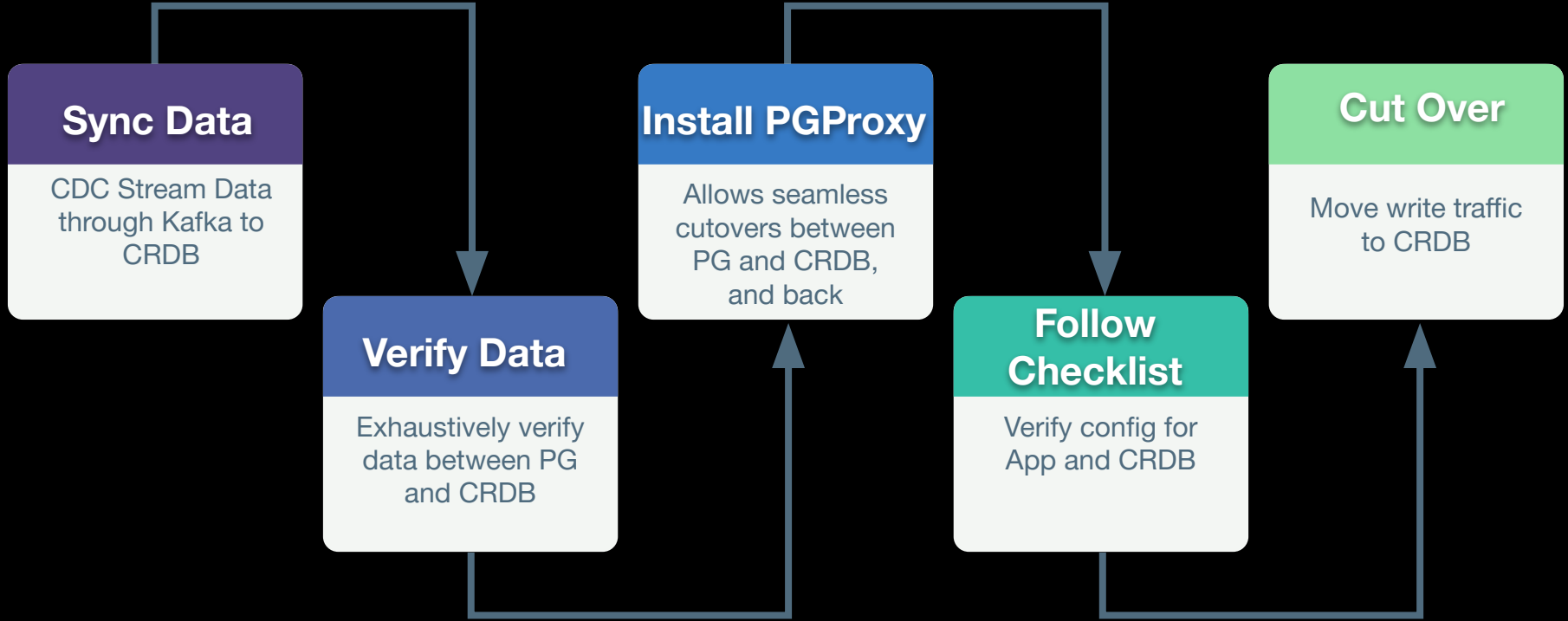


The Great Migration: Constraints

**0 downtime during these
migrations and reversible?**



Migration





The Great Migration: Process

Phase 1:

**Begin syncing data from PG →
CRDB using Change Data
Capture (CDC)**



The Great Migration: Process

Phase 2:
Verify the data

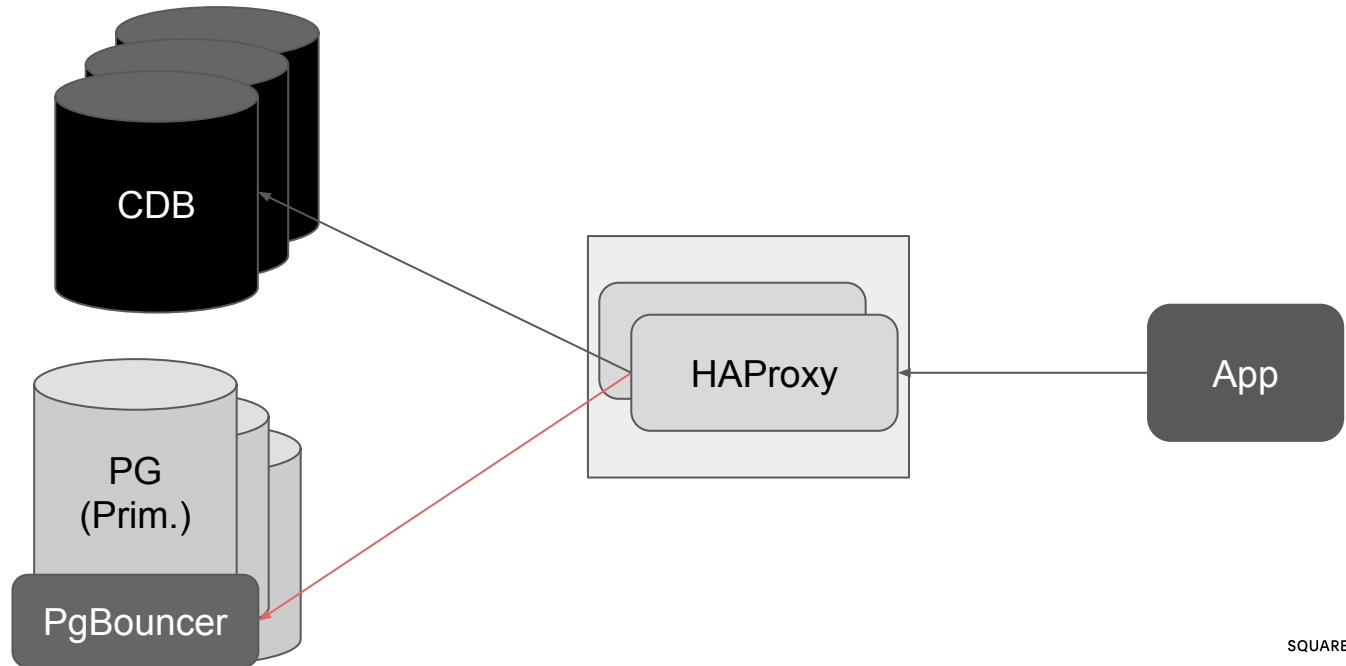


The Great Migration: Process

Phase 3: Setup Proxies



The Great Migration: Process





The Great Migration: Process

Phase 4:

**Prep for cutover; setup failback,
schedule with team.**



The Great Migration: Process

Phase 5:

**Do cutover; exercise failback if
need be**



The Great Migration: Process

Each of these phases is done:

- Staging
- Production

Other environments do not
require seamless cutovers



Lessons Learned

Lessons Learned: Team

**99.99% comes down to the
team doing the migration**





Lessons Learned: Code Changes

**A vast majority of applications
were easy – some were not!**

Lessons Learned: Performance Differences

A man is rock climbing a vertical cliff face. He is shirtless, wearing olive green cargo pants and climbing shoes. He is positioned in the lower right quadrant of the frame, reaching up. The cliff is light-colored and textured. The background shows a clear blue sky and some green foliage on the left side.

**Some applications were not designed
to be distributed systems – these were
a challenge**

Lessons Learned: Buy-in

Uneven buy-in from
teams about the value
for the effort





Lessons Learned: Self Discovery

**Developers will take advantage
of the new tooling**



Lessons Learned: Adapting to Sequences

- **Hash-sharded indexes**
 - **Can help with hotspots**
 - **Can cause read fanout**



Lessons Learned: Read After Write

- **Using READ_COMMIT in nearly all cases fixed these issues.**



Closing Remarks



SQSP is on track to exceed its business goals

- **Global reach**
- **Horizontal Scale**
- **Improved recovery time**
- **Reduced complexity**



Developer Experience has Improved

- **Online schema changes**
- **Console and Observability**
- **Time Traveling Queries**



Security Stance has Improved

- **Seamless upgrades**
- **mTLS for replication**
- **Event log**



Thank you!