FORM3

Multi-cloud essentials: how we operate CockroachDB at Form3

Rogger Fabri, Lead Engineer
Mario Morgado, Senior Software Engineer

Oh Hai!

👋 Rogger Fabri

💻 Lead Engineer @ Form3
🇮🇪 Based in Dublin, Ireland
🛠️ Platform Data Storage team

 roggerfabri
 roggerfabri

👋 Mario Morgado

💻 Senior Engineer @ Form3
🇬🇧 Based in London, UK
🛠️ Platform Data Storage team

 mjvm

Oh Hai!

👋 Rogger Fabri

💻 Lead Engineer @ Form3
🇮🇪 Based in Dublin, Ireland
🛠️ Platform Data Storage team

 roggerfabri
 roggerfabri

👋 Mario Morgado

💻 Senior Engineer @ Form3
🇬🇧 Based in London, UK
🛠️ Platform Data Storage team

 mjvm

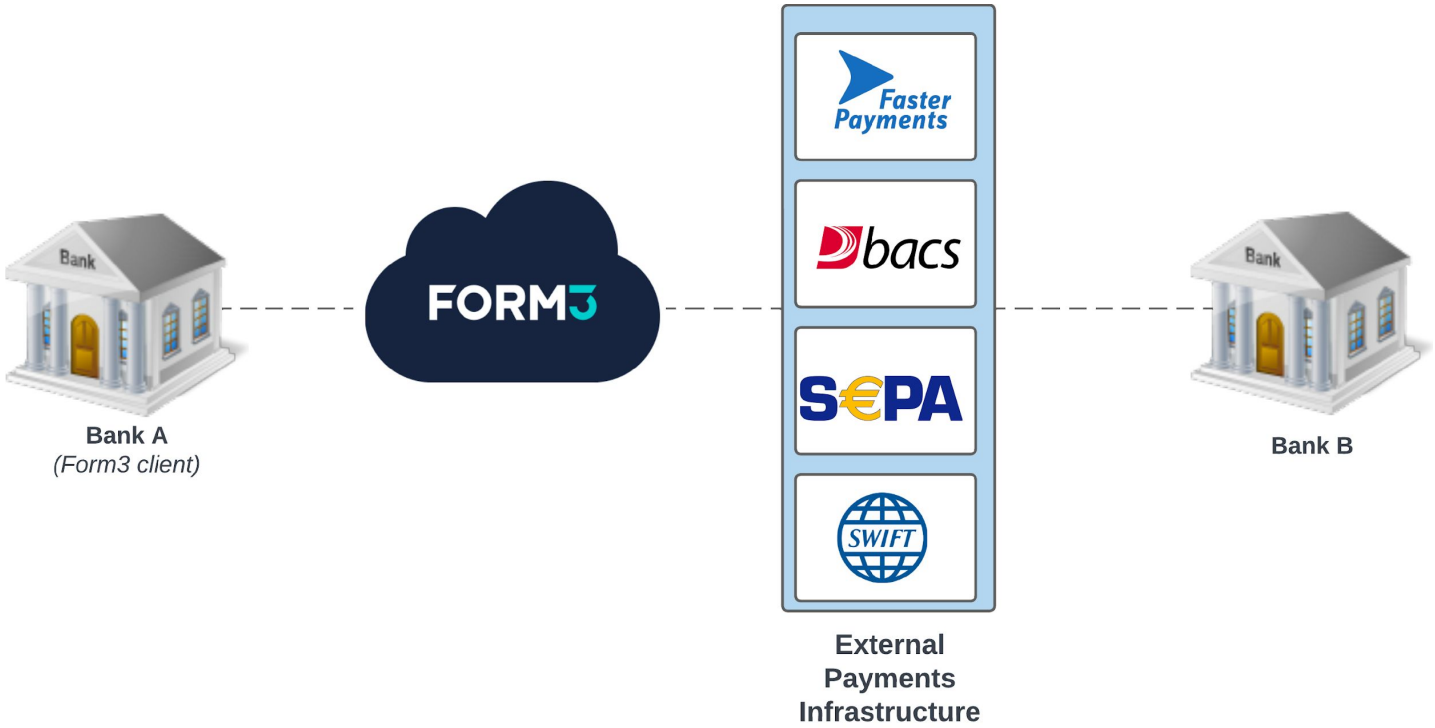
Oh Hai!

💼 About Form3

- Real-time payments processing platform
- Multi-cloud (AWS, GCP, Azure)
- Go, IaC (Terraform)
- SecDevOps culture
- Fully remote

Introduction to multi-cloud

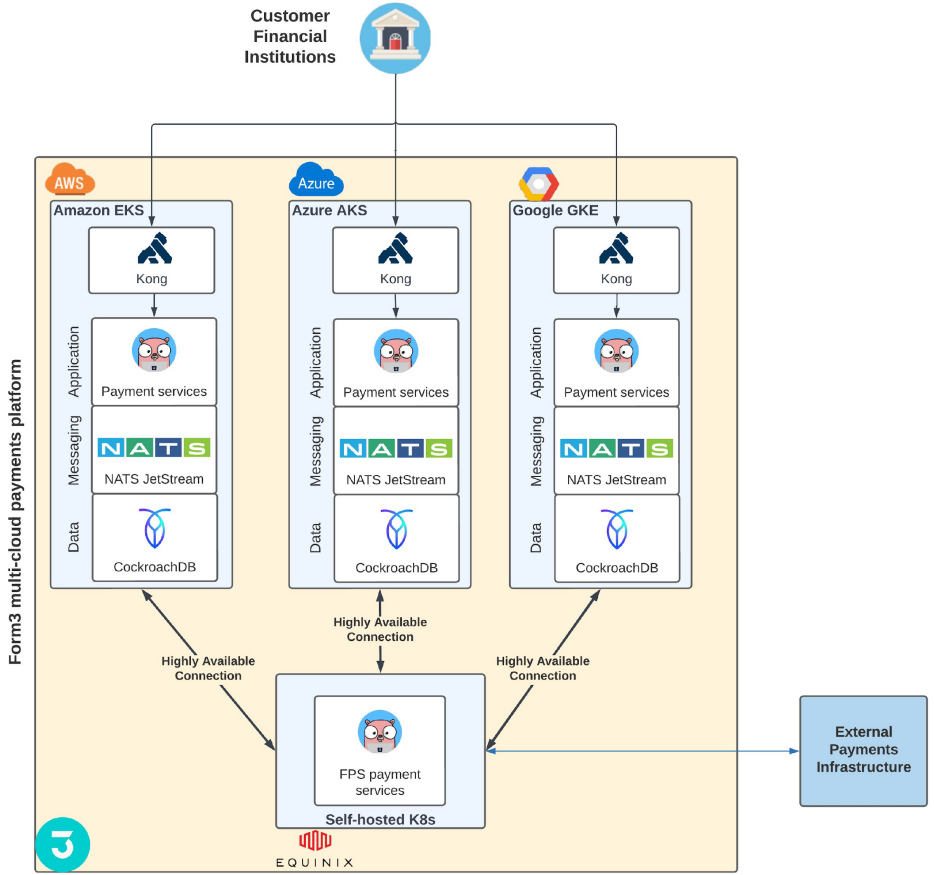
What is a payments processing platform?



Intro to multi-cloud

The technologies behind multi-cloud

Our customers can connect to the Form3 platform through endpoints in each cloud. The multi-cloud project is in the final stages of development. 🚀



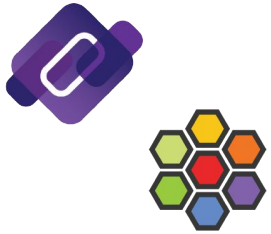
Multi-cloud CockroachDB Cluster

Challenges and solutions

FORM3

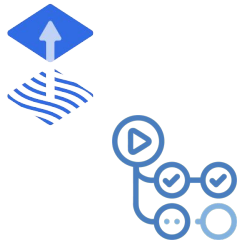
Multi-cloud CockroachDB Cluster

- Pod to pod connectivity:
  - Cilium CNI
  - Routing between the 3 Clouds and our physical DCs
  - CoreDNS w/ DNS Forwarding



Multi-cloud CockroachDB Cluster

- Deployment
  - GitHub Actions
  - Flux w/ OCI registries

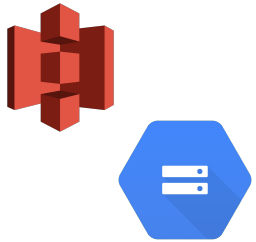


Multi-cloud CockroachDB Cluster

- Security
  - TLS Certificates w/ Cert Manager + Hashicorp Vault
  - Encryption at Rest Key w/ External Secrets Operator (ESO)
  - Maintain our own CockroachDB operator

Multi-cloud CockroachDB Cluster

- Backups
  - Locality-restricted backups, S3 and Google Cloud Storage



Multi-cloud CockroachDB Cluster

- Observability
  - Prometheus + Grafana

PostgreSQL to CockroachDB

Performance optimisation and application design

FORM3

PostgreSQL to CockroachDB

- Zero-downtime upgrades with rolling updates
- Can sustain a full cloud outage without performance impact

PostgreSQL to CockroachDB

Performance optimisation and application design

Indexes

- Tree like coverage
- Partial indexes
- Sequential data must use shard-based indexes

Cardinality

- Using UUIDv4 is not enough (data needs to have high cardinality)

MVCC

- A row is rewritten on update
- Frequent updates on the same row creates hot ranges

FORM3

PostgreSQL to CockroachDB

Performance optimisation and application design

Indexes

- Tree like coverage
- Partial indexes
- Sequential data must use shard-based indexes

Cardinality

- Using UUIDv4 is not enough (data needs to have high cardinality)

MVCC

- A row is rewritten on update
- Frequent updates on the same row creates hot ranges

FORM3

PostgreSQL to CockroachDB

Performance optimisation and application design

Objectives:

- Workflow 1 – 100ms P99
- Workflow 2 – 200ms P99

Constraints:

- Highly concurrent access
- Transient data
- Cross cloud communication

FORM3

PostgreSQL to CockroachDB

Performance optimisation and application design

Range operations locality

- Schema was redesigned to use a single table instead of 3
- Column holding the blob is dropped after workflow finishes
- Provision nodes with as much memory as possible

PostgreSQL to CockroachDB

Performance optimisation and application design

Reduce contention

- Column Families
- TTL jobs

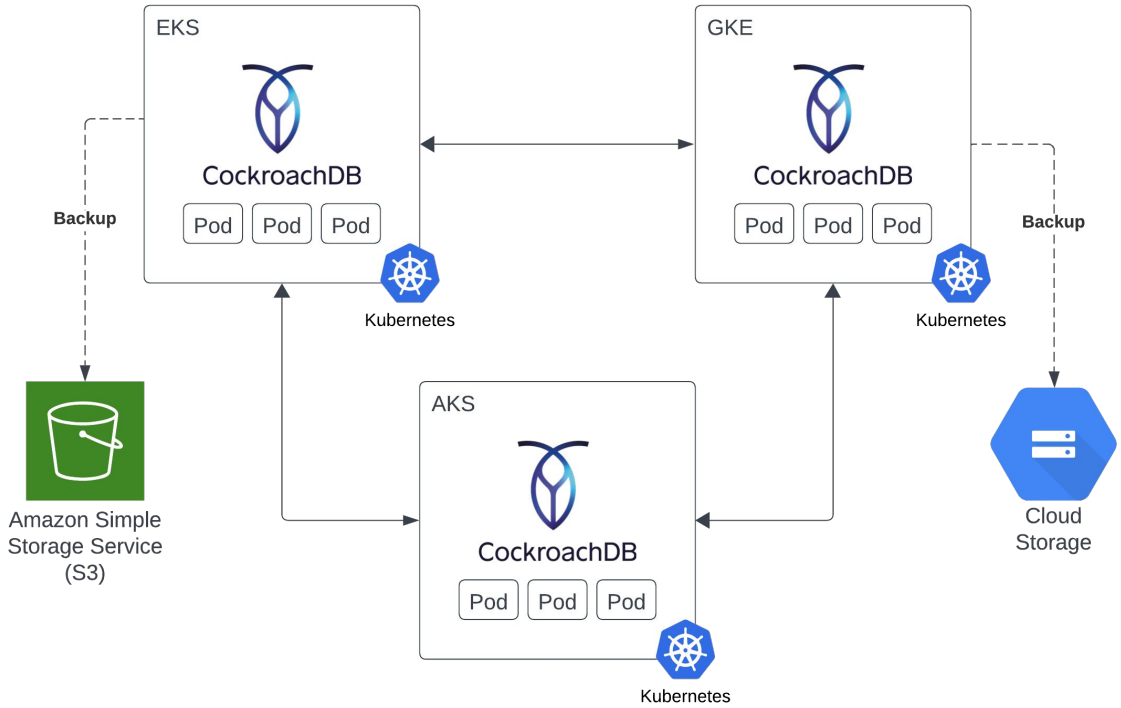
FORM3

Managing multi-cloud backups

Managing multi-cloud backups

- We use locality-restricted backups, introduced in v23.1
  - Backups in two different clouds (GCP, AWS), this gives us more resiliency in case of cloud outages
  - Implicit authentication via IAM Roles and Service Accounts
- Schedules for backups
  - Full cluster backups every 12 hours
  - Incremental backups every 5 minutes
  - In case of a DB corruption:
    - Recovery Point Objective (RPO) is 5 min
- In case a backups fails we get paged due to observability in place

Managing multi-cloud backups

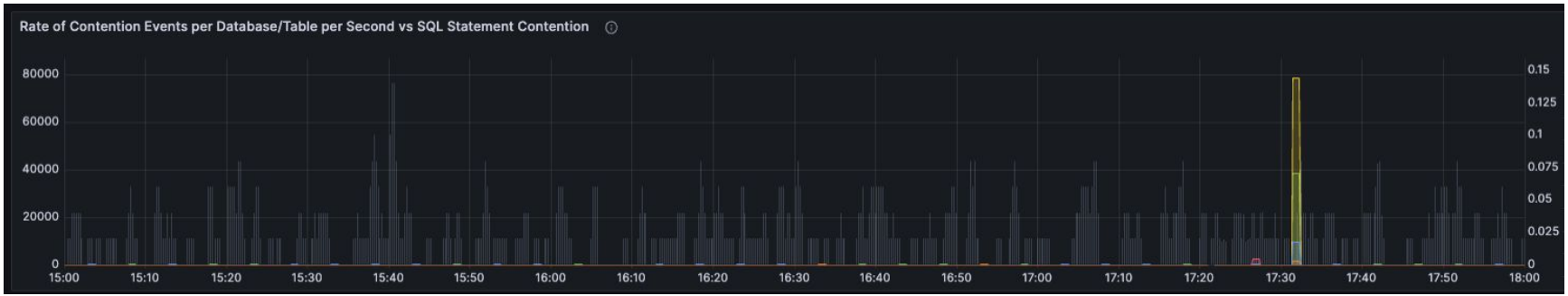


Observability & monitoring

Observability & monitoring

- Grafana Dashboards and Prometheus Metrics, logz.io for logs
- Custom dashboards created based on our experience and needs
  - End-to-end monitoring and metrics for application teams
  - Platform monitoring for operational observability
- Alerts set up on Grafana based on Prometheus metrics
  - When triggered and engineer gets paged on PagerDuty
- Alerts set up on logz.io for log-based alerts
- We are using **visus** (https://github.com/cockroachlabs/visus) for custom metrics such as:
  - Rate of contention events
  - Database/table size
- SLOs/SLIs based on latency buckets

Observability & monitoring



FORM3

Thank you!

🎧 Podcast: techpodcast.form3.tech

📖 Blog: form3.tech/engineering/content