# Technical Questions

## Architecture & Scaling

1. What does the current PostgreSQL architecture look like (e.g. single cluster, sharded, horizontally scaled, I/O Optimised? What engine class? What Major Version? What data volumes? How many Clusters? AWS or on-prem (if on-prem what OS?)
2. How do you handle high throughput? Are you using connection pooling? PgBouncer / RDS-Proxy? Application pooling? Any custom partitioning, logical replication, or queue-based ingestion layers?
3. Is the application write or read heavy?
4. Are reads mostly real-time, analytical, or a mix? Do you split these out to read-replicas?
5. Are you using PL/pgSQL?
6. How do you handle schema migrations? Flyway?

---

## Tooling & Observability

1. What tooling do you use for performance monitoring? (e.g. pganalyze, pg_stat_statements, CloudWatch, Datadog?, Grafana? PgBadger)
2. How is alerting set up around Postgres infrastructure? (Teams / Slack alerts?) Is there an SRE team involved?
3. Log monitoring? ElasticSearch / OpenSearch?

---

## Operational Practices

1. How do you currently manage minor and major version upgrades for Postgres (especially if zero-downtime is expected ?) Logical replication? Blue/Green deployments?
2. Do you require 100% uptime or is there time available for maintenance windows?
3. Do you use Infrastructure as Code for your database provisioning - e.g. CDK / Teraform / Ansible?
4. How do you currently deploy? Do you do direct CloudFormation stack deployment (AWS) or run through CI/CD pipelines such as GitLab Runners

---

## Data Movement & Integration

1. How is data ingested into the platform? (e.g. app-driven inserts, Batch or scheduled ETL pipelines?)
2. Is there any cross-region or multi-cloud replication happening?
3. Are there event-driven or streaming components (e.g. Kafka, Kinesis) connected to the database

---

## Data Management

1. What’s your current backup and disaster recovery setup?
2. Is there a data archival or cold storage strategy? Data Warehouse?

---

## Culture, Ownership & Future Plans

1. How is ownership of the database layer shared between Data Engineering, SREs, and Developers?
2. Are there any upcoming challenges around scaling, migrations, or re-architecting?
3. Is there room to shape technical direction for Postgres or propose new tooling?