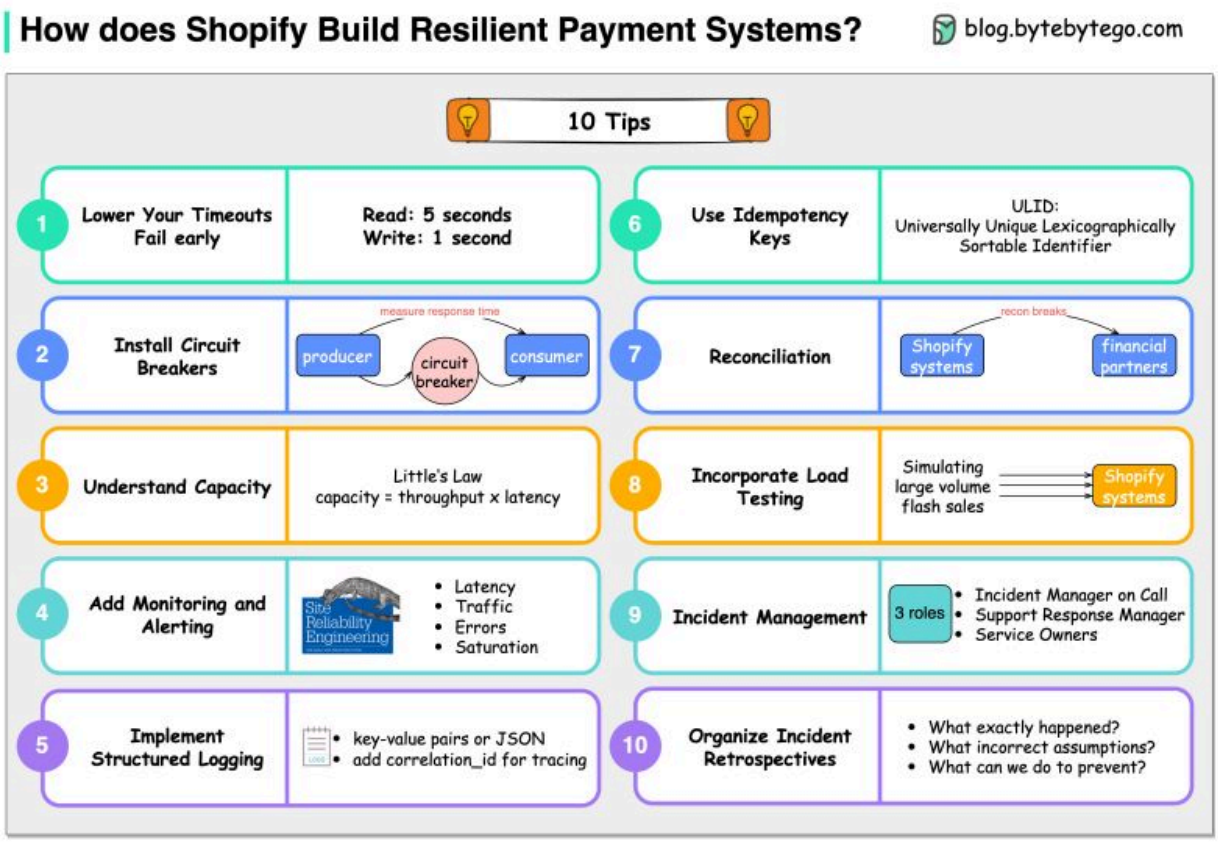
10 principles for building resilient payment systems (by Shopify).

Shopify has some precious tips for building resilient payment systems.



1. Lower the timeouts, and let the service fail early
   The default timeout is 60 seconds. Based on Shopify's experiences, read timeout of 5 seconds and write timeout of 1 second are decent setups.

2. Install circuit breaks
   Shopify developed Semian to protect Net::HTTP, MySQL, Redis, and gRPC services with a circuit breaker in Ruby.

3. Capacity management
   If we have 50 requests arrive in our queue and it takes an average of 100 milliseconds to process a request, our throughput is 500 requests per second.

4. Add monitoring and alerting
   Google's site reliability engineering (SRE) book lists four golden signals a user-facing system should be monitored for: latency, traffic, errors, and saturation.

5. Implement structured logging
   We store logs in a centralized place and make them easily searchable.

6. Use idempotency keys
   Use Universally Unique Lexicographically Sortable Identifier (ULID) for these idempotency keys instead of a random version 4 UUID.

7. Be consistent with reconciliation
   Store the reconciliation breaks with Shopify's financial partners in the database.

8. Incorporate load testing
   Shopify regularly simulates the large volume flash sales to get the benchmark results.

9. Get on top of incident management
   Each incident channel has 3 roles: Incident Manager on Call (IMOC), Support Response Manager (SRM), and service owners.

10. Organize incident retrospectives
    For each incident, 3 questions are asked at Shopify: What exactly happened? What incorrect assumptions did we hold about our systems? What we can do to prevent this from happening?

Reference: 10 Tips for Building Resilient Payment Systems