



# TiDB para DBAs MySQL

04 Outubro 2025

## AGENDA

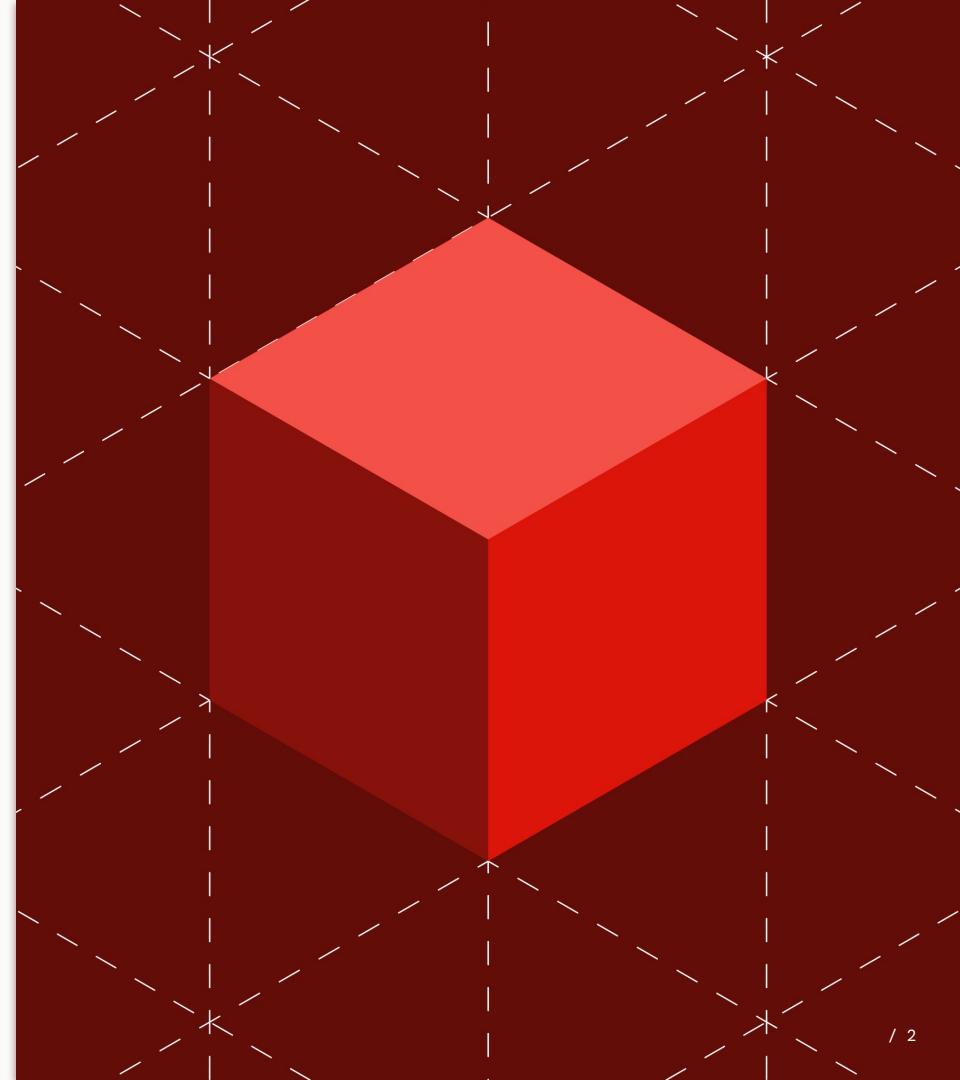
- 01 **Intro ao TiDB**
- 02 **Quando faz sentido**
- 03 **Como avaliar**
- 04 **Estudo de Caso**

---

**Airton Lastori (PingCAP/TiDB)**

**Cassio Perin (Bling)**

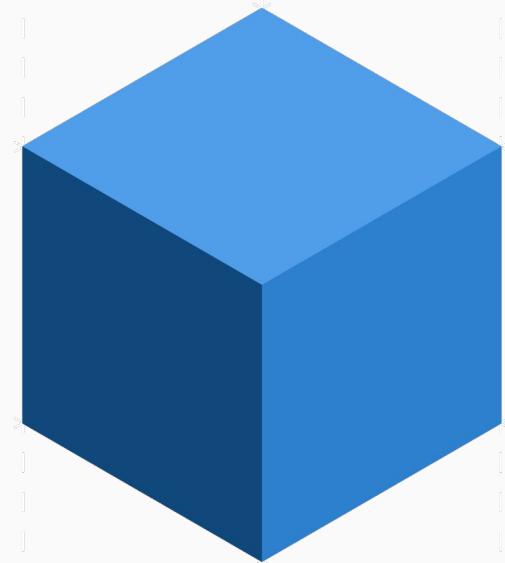
**Everton Silva (Bling)**



# 01 Intro

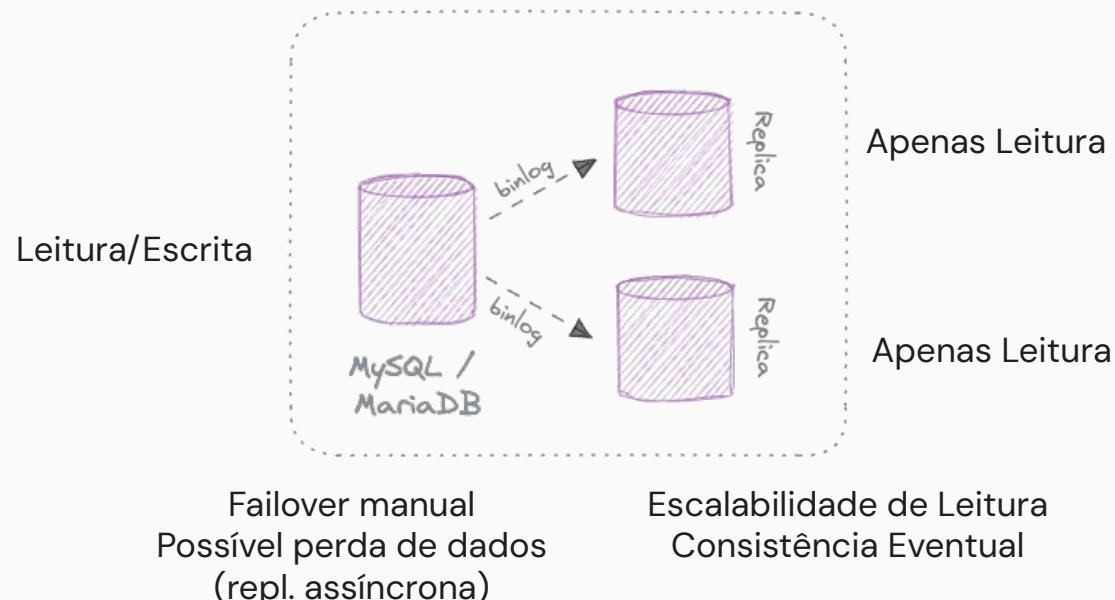
# Ti(tanium)DB

- **10 anos (PingCAP)**
  - Popular na Ásia, expandindo globalmente
- **Open Source (Apache 2.0) + Cloud**
- **Baseado em:**
  - Google Spanner + F1 (Papers)
  - GoLang + Rust
  - Facebook RocksDB (Key-Value storage engine)
- **Compatível com MySQL**
  - Não é um Fork
  - Mesmo protocolo/connectors
  - ACID/MVCC (SQL Distribuído e **escalável**)



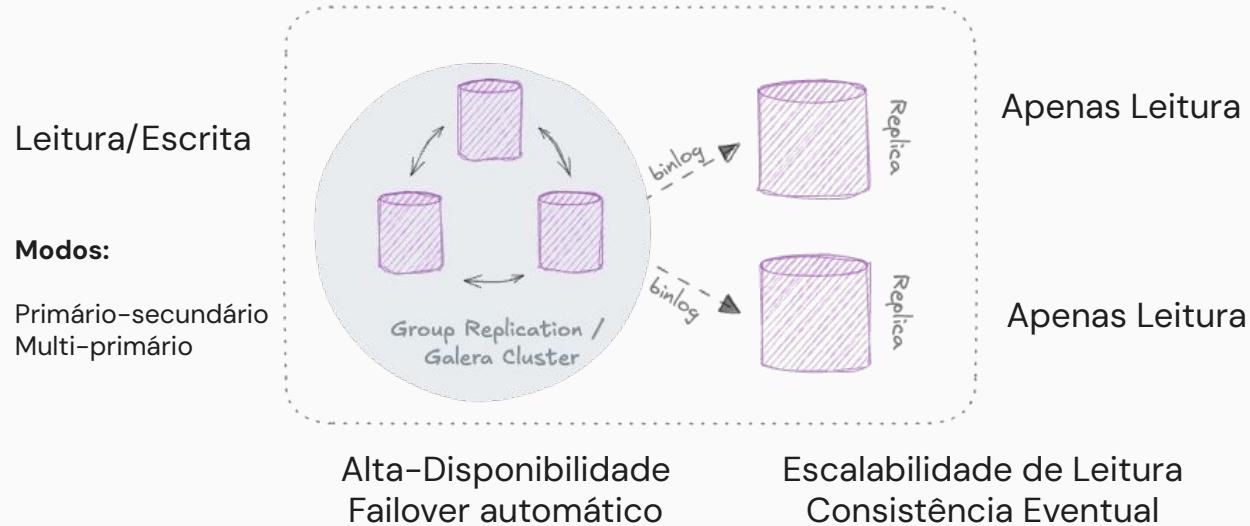
# Arquitetura MySQL (recap)

## MySQL/MariaDB: Primary-Secondary



# Arquitetura MySQL (recap)

## MySQL/MariaDB: Group Replication/Galera



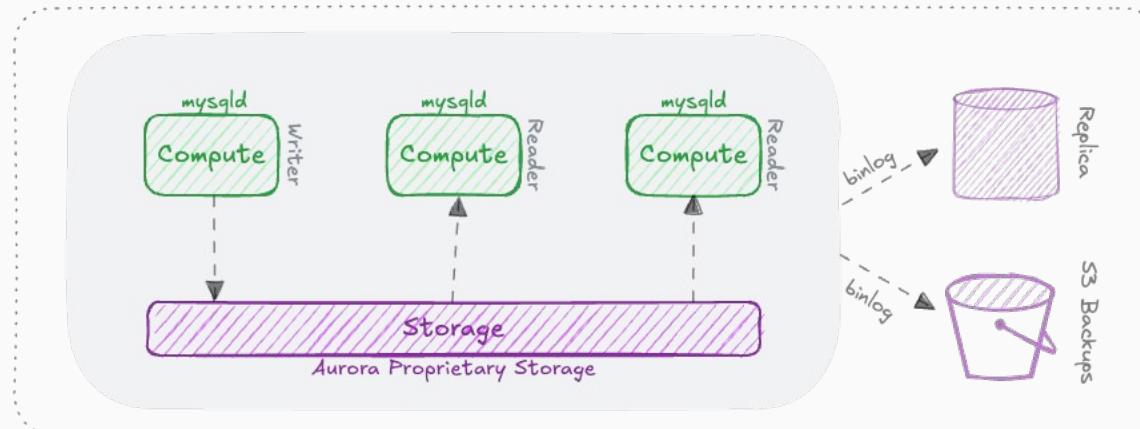
# Arquitetura Aurora

## AWS Aurora: Storage Compartilhado

Separação:  
Compute +  
Storage

Evita 1 cópia do  
dataset por nó  
computacional

Limits:  
1 nó de escrita  
15 nós leitura  
128 TiB

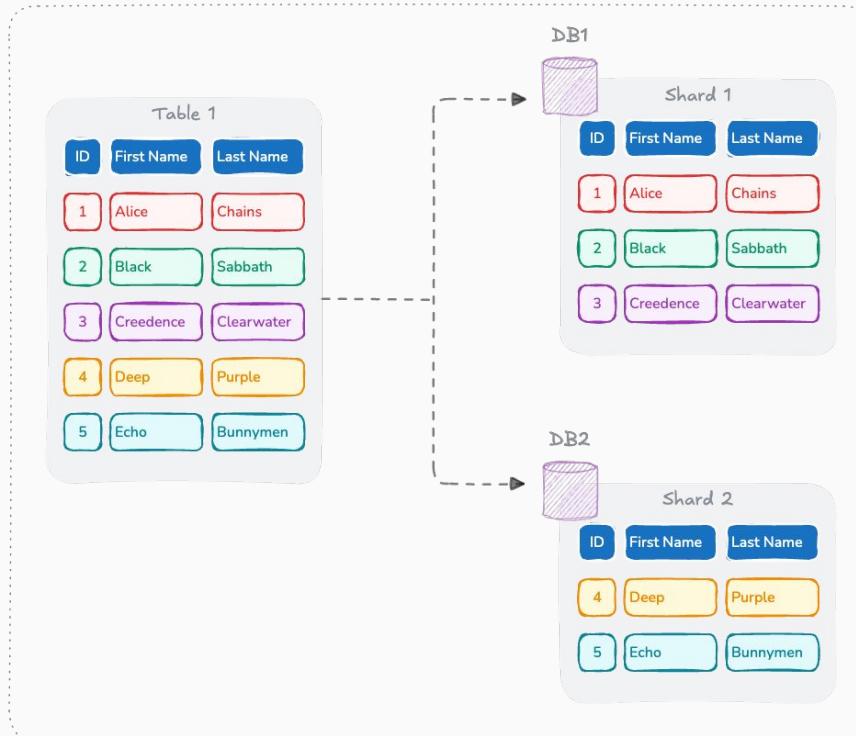


Alta-Disponibilidade  
Failover automático

Escalabilidade de Leitura  
Consistência Eventual  
(baixo lag replicação)

Réplicas opcionais  
(ex., cross-region)

# Sharding: escalabilidade de escrita



## Exemplos de solução:

- "na unha"
- Vitess (middleware)
- MySQL NDB Cluster (nativo)

## Desafios:

- **Manter o recursos do SQL:** cross-shard joins/FKs/UNIQUE e transações viram código da aplicação.
- **Rebalanceamento** (crescimento/hot keys) sem downtime é difícil.
- **Gestão de schema:** DDL em N shards.
- **Backups/PITR:** coordenados entre shards para evitar drift.
- **Monitoramento/custos:** troubleshooting, on-call e tooling sobem bastante.

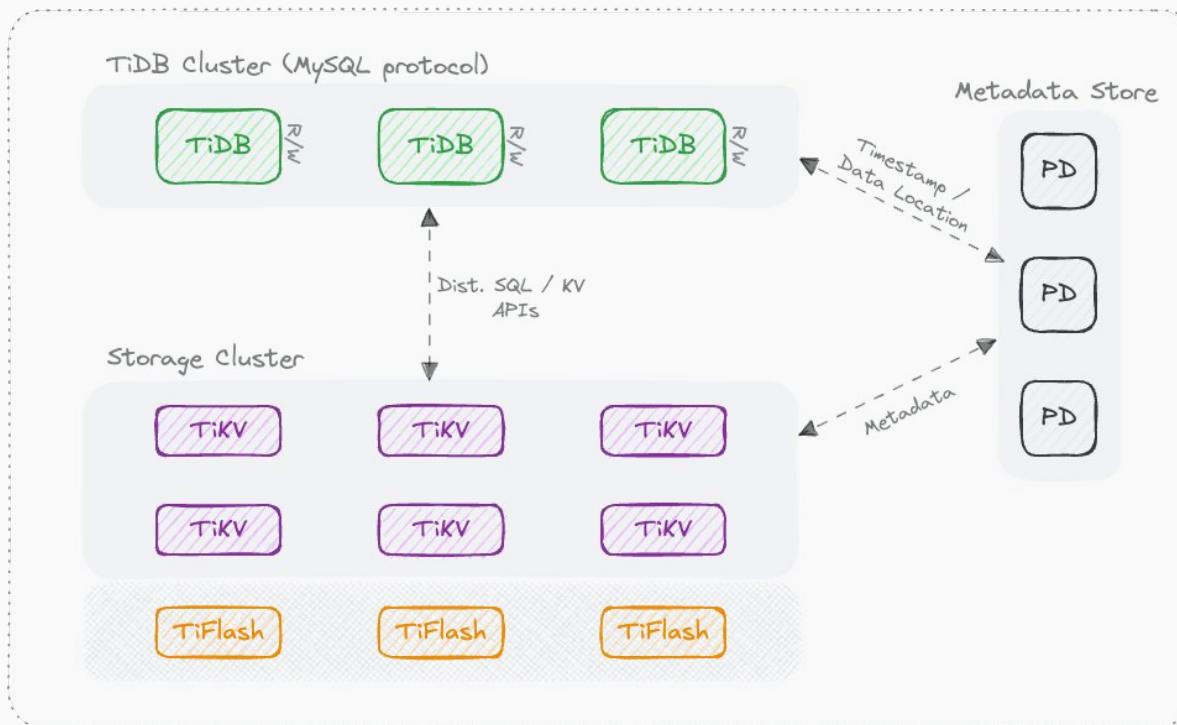
# Arquitetura TiDB

Leitura/Escrita  
em qualquer nó,  
abstraindo  
complexidade  
da distribuição  
de dados.

Storage  
escalável com  
compactação,  
HA e sharding  
granular.

Dados colunares  
para Analytics

"Relógio" e  
"mapa" do  
cluster



# Arquitetura TiDB – Camadas

## Camada Stateless de SQL Distribuído (TiDB)

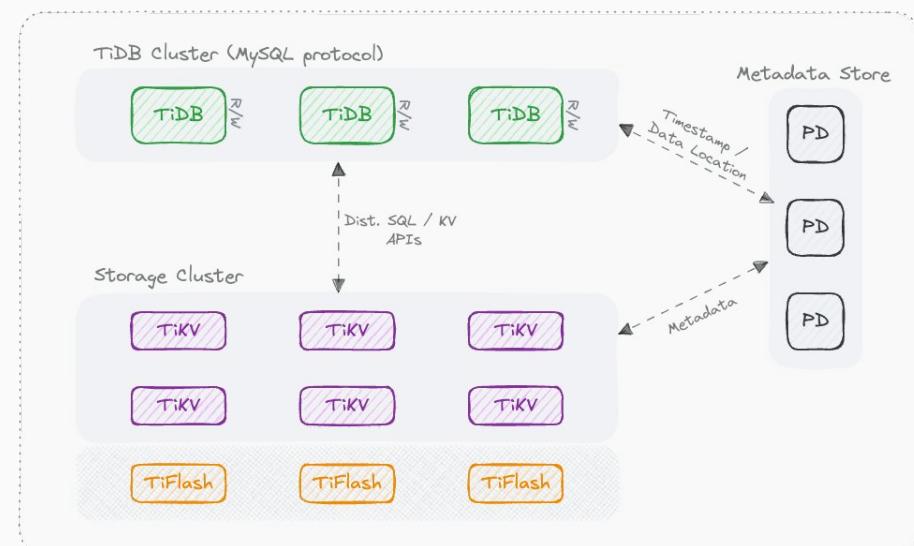
- Protocolo MySQL (drivers/ORMs existentes)
- Abstração via SQL (sharding transparente)
- Leitura/escrita em qualquer TiDB (roteamento automático)
- Milhões de tabelas/tenants no mesmo cluster

## Camada de Storage (TiKV + TiFlash)

- KV distribuído com Raft (replicação + failover)
- Alta compactação, baixa duplicação
- MVCC para transações ACID
- Scale-out/in linear de capacidade/IOPS
- TiFlash (colunar) para OLAP/HTAP opcional

## Coordenação (Placement Driver)

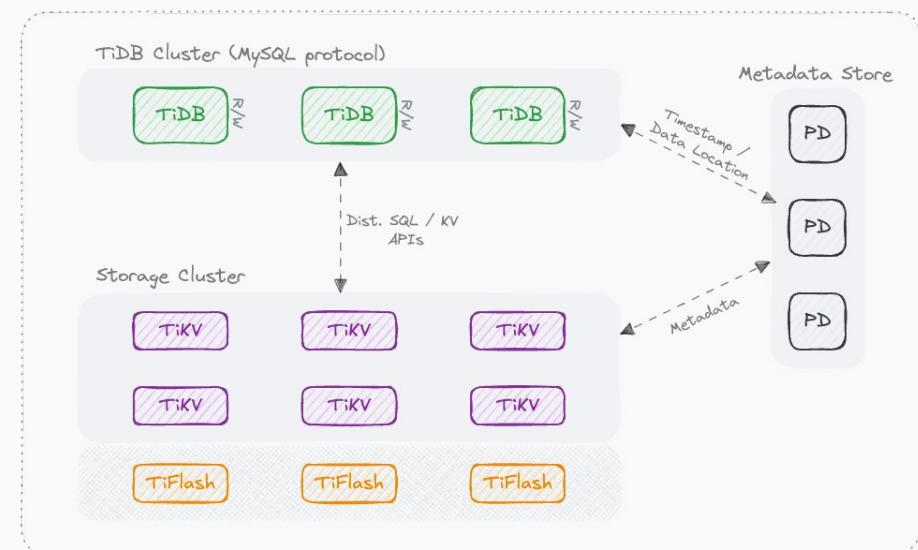
- Relógio do cluster: ordem entre transações.
- Administrador de shards (regiões): decide onde ficam os dados e redistribui para performance e alta disponibilidade.



# Arquitetura TiDB – Recursos

## Recursos para operar em larga escala

- Compute e Storage elásticos (independentes)
- Operações online: DDL, índices, rolling upgrade
- Migração de conexões (TiProxy opcional)
- Backups/Restore/PITR otimizados para grandes volumes
- Monitoramento integrado





# TiDB Cloud – AWS, Azure, Google, AliCloud



## TiDB Cloud Starter

Aplicações simples,  
requisitos básicos.



## TiDB Cloud Essential

Maior  
escalabilidade,  
performance e  
recursos de  
segurança.



## TiDB Cloud Premium

Aplicações de  
missão-crítica,  
escalabilidade  
ilimitada.



## TiDB BYOC

Instalado na sua  
conta cloud,  
co-gerenciado pela  
PingCAP.

Comece gratuitamente: [tidbcloud.com](https://tidbcloud.com)

## 02 Quando usar



# Fique com MySQL quando...

- **Seu single-node está “confortável” e sem dores reais de escala/HA:**  
→ Se precisar de HA/tolerância a falhas, use Group Replication/Galera.
- **Workloads majoritariamente de leitura e...**
  - Consegue separar leituras/escritas na aplicação;
  - Aceita consistência eventual;
  - Mantém número limitado de réplicas;
  - Tolera janelas de manutenção/failover.  
→ Réplicas de leitura resolvem sem complexidade extra.
- **Requisito de latência ultra-baixa em 1 máquina.**  
→ Network hops de sistemas distribuídos viram gargalo quando requer sub-ms em escrita.
- **App embarcado / footprint mínimo.**  
→ MySQL ou SQLite são mais simples e baratos.
- **Dependência pesada de recursos específicos/legados do MySQL.**  
→ Procedures/triggers, UDFs, storage engine.
- **Sem flexibilidade de alterar a aplicação.**  
→ Otimização de queries ou exigências rígidas do fornecedor para usar “MySQL/InnoDB”.
- **Somente OLAP e escala crescendo em direção data warehouse.**  
→ Prefira um MPP dedicado; não precisa de banco transacional distribuído.

# Use TiDB quando...

- **Precisa escalar escritas/leituras sem sharding manual ou separação R/W.**  
→ SQL distribuído (*sharding transparente*) + Multi-primário (*R/W em qualquer nó*).
- **Escala de centenas de TB com economia de storage.**  
→ Compressão e redundância mínima necessária para tolerância a falhas (ex. 3 cópias).
- **Muitas tabelas/tenants/instâncias (SaaS) e/ou mudanças frequentes de schema.**  
→ Milhões de tabelas e *DDL online* eficiente em tabelas grandes.
- **Alta disponibilidade sem janela de manutenção ou drop de conexões.**  
→ Rolling upgrades, failover transparente, migração de conexões via *TiProxy*.
- **Operação mais simples com dezenas/centenas de nós.**  
→ Monitoramento integrado; backup/PITR consistentes; administração centralizada (*TiUP, Operator*).
- **Compatibilidade com MySQL e com liberdade de escolha.**  
→ Alterações mínimas na aplicação; Open-source; *TiDB Cloud* (gerenciado).
- **HTAP: OLTP + OLAP no mesmo cluster.**  
→ Rélicas colunares (*TiFlash*) sem ETL.



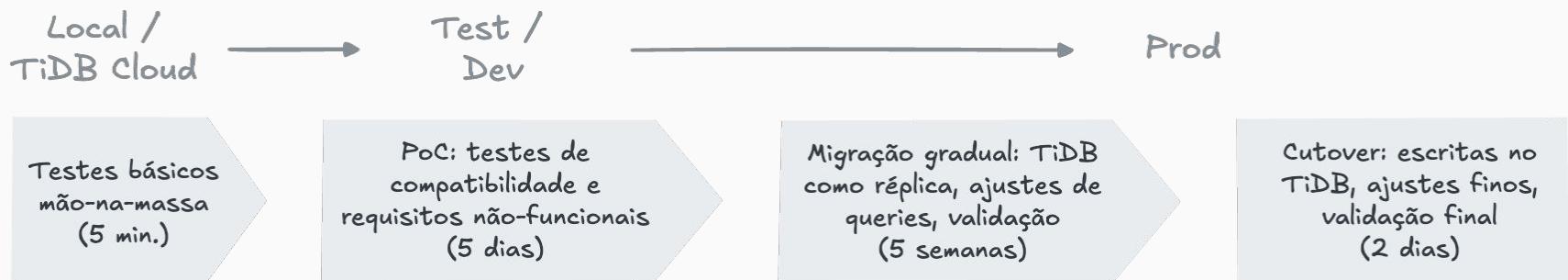
# Hora do DBA avaliar

*"Acho que dá para aguentar mais um pouco."*

## 03 Guia para avaliação

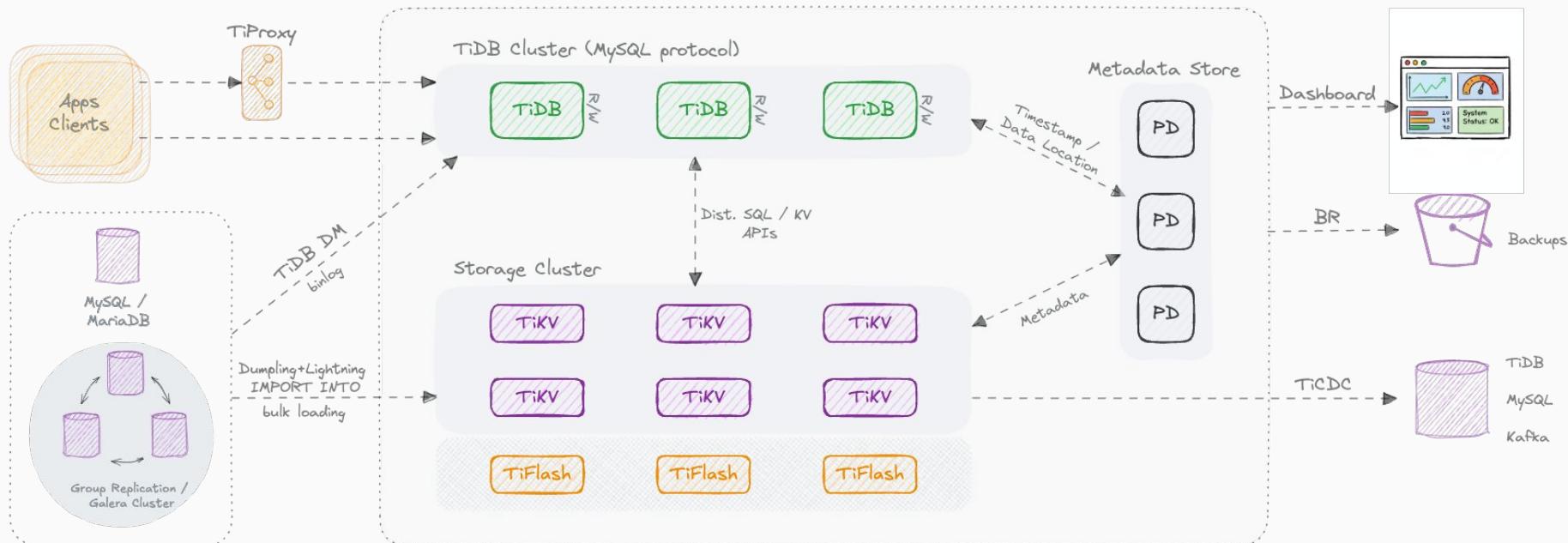
## MIGRAÇÃO

# Avalie e Migre



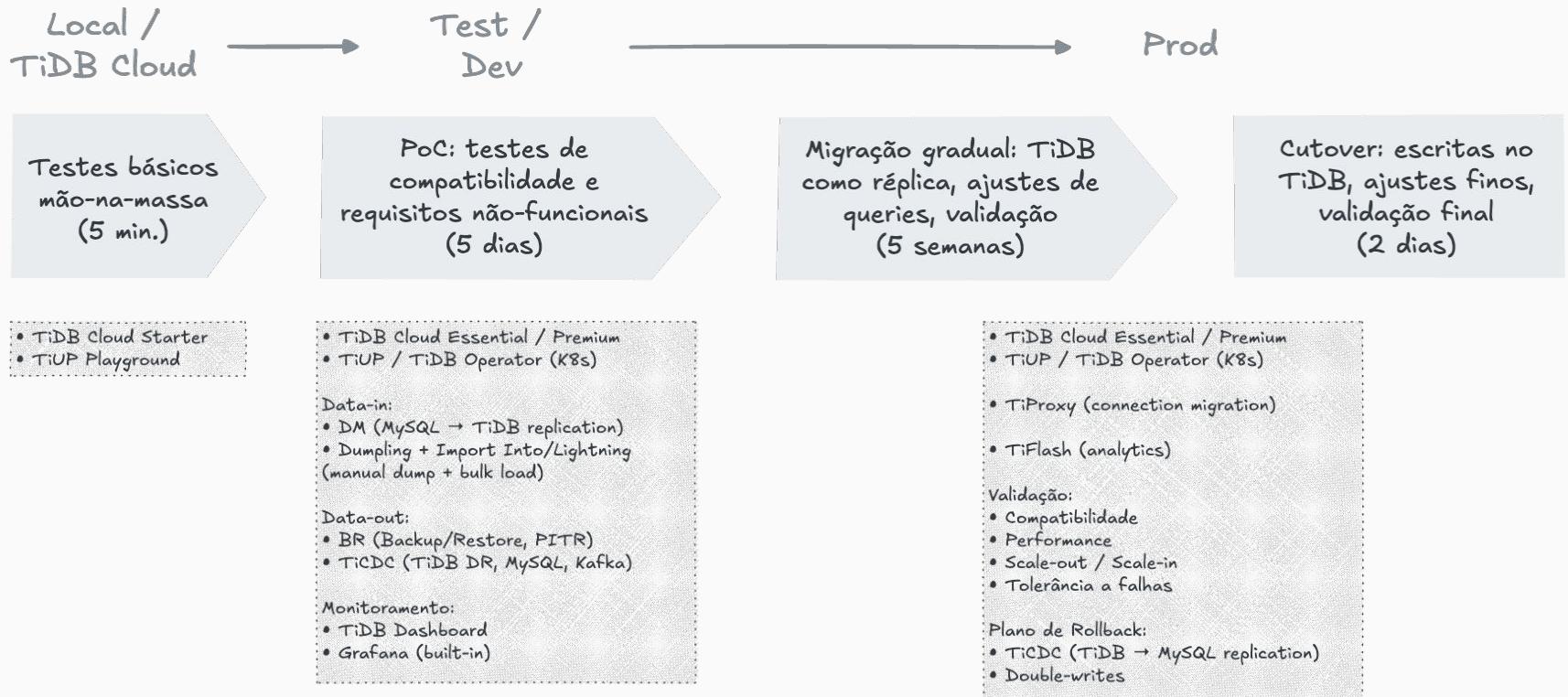
## MIGRAÇÃO

# Ferramentas



## MIGRAÇÃO

# Avalie e Migre



MIGRAÇÃO

# Ajuda

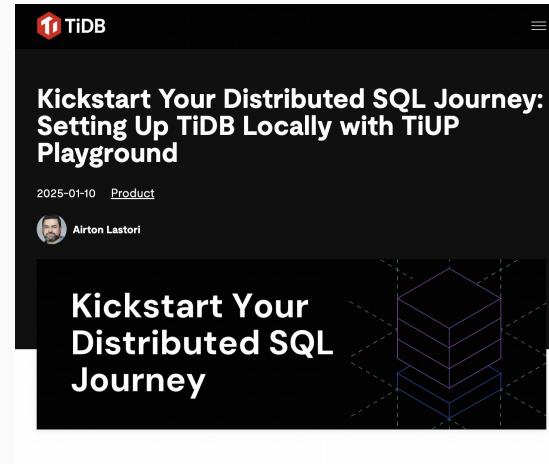
 TiDB | Community

 slack  Discord

 TiDB | AI Assistant

[tidb.ai](https://tidb.ai) → [TiDB Docs](#)

Tutorial mão-na-massa (5–20 min.)





October 7-8 | Computer History Museum, Mountain View, CA



# TiDB SC<sup>A</sup>LE 2025

Scaling Data. Accelerating AI.  
Enabling You.

Register Now



## 04 Casos de Sucesso

# Customer Validation

Trusted and Verified by **4000+** Industries Leaders Globally

NA/EMEA



dailymotion



Opera Ads



JAPAN



PLAID ele.style

SB Payment Service

LINE

CAPCOM®

DMM.com

APAC



DELHIVERY



bookmyshow



PING AN  
Finance · Technology



Bank



WeBank

招商銀行  
CHINA MERCHANTS BANK

浙商银行  
CZBANK

北京银行  
BANK OF BEIJING

ailo bank



China unicom 中国联通

YumChina

浦发银行  
SPD BANK

ZTO:  
Global Logistics

(SF) EXPRESS

广发银行  
CGB

小红书

GCR

# Bling



# Bling - TiDB

Outubro | 2025

## Case Bling

### Profile



**Cássio Perin**  
Especialista em Infraestrutura



**Everton Silva**  
Engenheiro de Software



## Case Bling

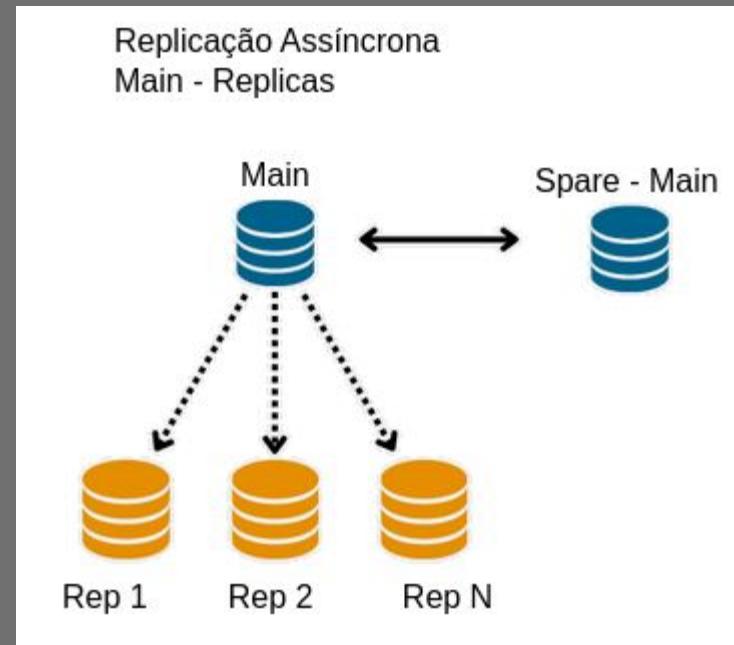
# Bling é um SaaS ERP focado e-commerce

- Bling fundada em 2009, localizada em Bento Gonçalves / RS
- Parte do Grupo LWSA desde de 2021
- Líder de mercado como ERP com foco em e-commerce para PME no Brasil
- Integração com diversos marketplaces: Amazon, Shopee, Mercado Livre, etc
- + 300K usuários online diariamente

## Case Bling

# Cenário

- Estratégia Single Main / Multiple Read Replicas
- Ambiente AWS | US-East-1
- Estrutura Multi-Tenant
- Instâncias: EC2 16~48xlarge + EBS
- Replicação assíncrona por binlog
- Tabelas mais que dobrando de tamanho a cada ano



Main: MariaDB 10.2.14

+650 Tabelas

Replicas: MariaDB e MySQL 5.7

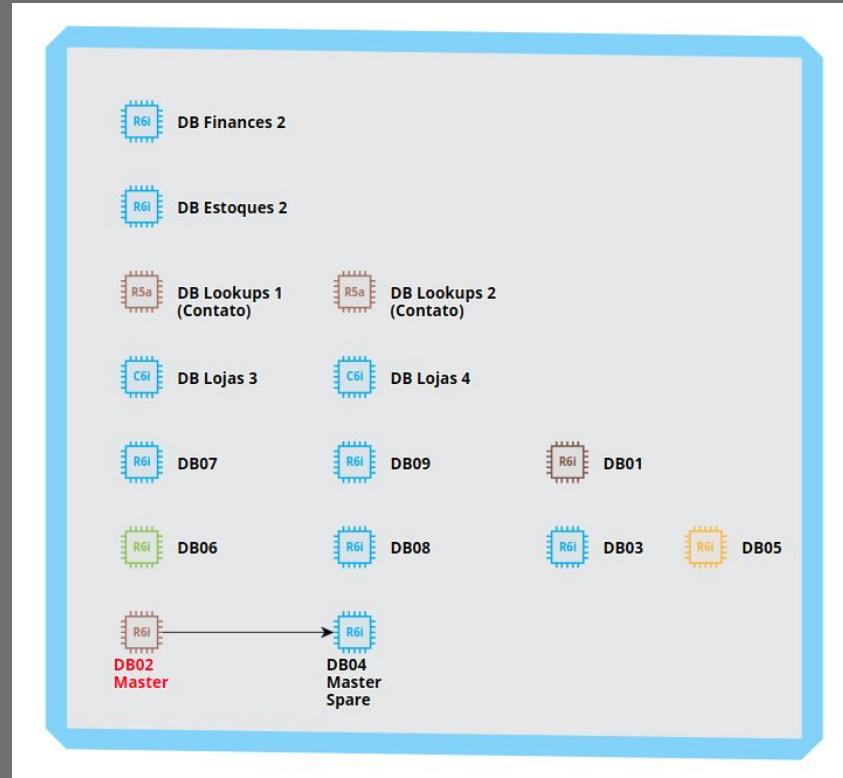
InnoDB

## Case Bling

### Stats do Banco de dados

**25 TB**  
p / instância

**2~5 Bi** rows nas maiores tabelas  
**2 TB** na maior tabela



## Case Bling

### Pico médio diário QPS

Main

80k

Replicas

300k

## Case Bling

# Problema

Principais **pain points** identificados:

- Escrita centralizada
- Projeção de crescimento de dados insustentável com tecnologias atuais
- Pontos únicos de falha
- Inconsistência de dados (replicação assíncrona)
- Alto custo de sustentação da infraestrutura
- *Diminishing return* evidente
- Alta complexidade de manutenção da estrutura
- Operações analíticas ineficientes ou não suportadas na estrutura atual
- Dificuldade em reestruturação da aplicação por impactar a experiência do cliente

# Escolha do TiDB

- Banco de dados distribuído altamente compatível com MySQL / MariaDB
- Separação das camadas de compute e storage
- Resiliente a falhas com auto recover
- Replicação síncrona automática
- Sharding automático (regions e particionamento)
- Operações de Alters não bloqueantes, rápidos e online
- Operações analíticas
- Deploy em ambiente de homologação simplificado com TiUP
- Fácil implantação, manutenção e scale-in/out com K8S
- Compressão de dados muito eficiente
- Ferramentas de monitoramento pré-construídas (observabilidade)
- Enorme quantidade de Materiais, documentações claras, IA e comunidade

# Configuração

- Implantação do cluster
  - AWS EKS com TiDB Operator
- Reestruturação de schema
  - Adequação a charset UTF-8
  - Chaves primárias compostas (tenantId + id)
  - Particionamento baseado em tenantId
- Importação e sincronização e um único dia com ferramentas
  - Dumpling - dump da base direto para S3
  - IMPORT INTO - importação da tabela direto do S3 com único comando
  - Data Migration (DM) - replicação

# Implantação

- Introduzido na aplicação como uma réplica de leitura
- Virada gradual da leitura para o TiDB, reduzindo uso das bases MariaDB
- Ajustes e otimizações
  - Uso de hints para otimizar queries
  - Resultados indeterminísticos (necessário ordenação)
  - Adição de índices cover para obter performance aceitável
  - Sizing do cluster - maior do que o esperado inicialmente (facilitado)
  - Configuração de AUTO\_INCREMENT para ser sequencial
  - Criação de resource\_groups (noisy neighbors)
- Treinamentos internos

# Virada da escrita

- Após virada majoritária da carga de leitura para o TiDB
- Downtime planejado de 6 horas na aplicação
  - Mudança do banco primário de escrita para o TiDB
  - Replicação do TiDB para réplicas de fallback utilizando TiCDC
  - Testes internos em todos os módulos assegurando funcionamento da aplicação
- Sem incidentes ou problemas de comportamento na escrita
- Hypercare (período de acompanhamento para assegurar funcionamento adequado)

# 10 meses de projeto

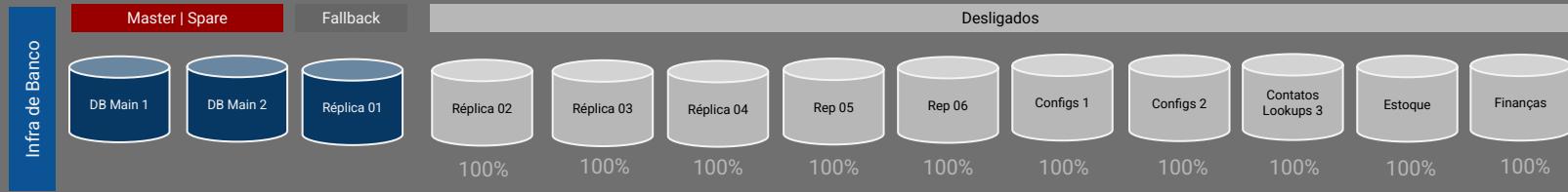
para concluir o processo de migração da escrita

De Outubro de 2024 até Agosto de 2025

- 2 dias para POC
- 3 meses de testes em ambientes controlado
- 2 semanas para subir um cluster completo em produção
- 1 mês para virar a carga da primeira réplica de leitura
- 5 meses de trabalho em adaptações e otimização de queries em conjunto com times de desenvolvimento
- 1 mês até virar a escrita (sem rollback)

## Case Bling

# Cenário final



Current storage size ⓘ

35.5 TB

Normal stores ⓘ

73

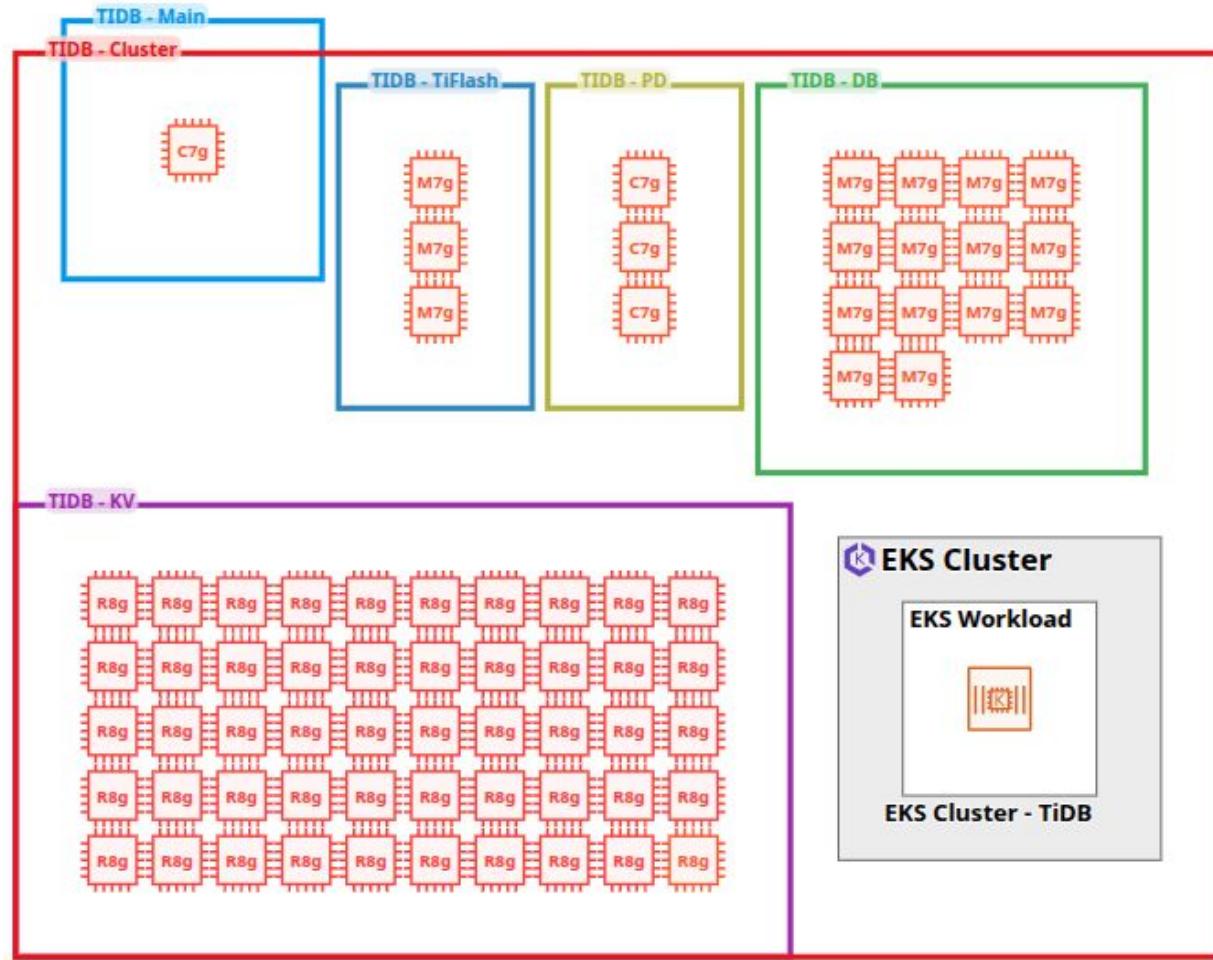
Number of Regions ⓘ

207375

# Conclusão

- Maiores desafios
  - Queries de baixa performance e dívidas técnicas
  - Realizar otimizações sem alterar comportamentos
  - Nova tecnologia para toda a empresa
  - Mudança de paradigma
- Ganhos
  - Escalabilidade
  - Tolerância a falhas
  - Milhares de queries otimizadas
  - Garantia de continuidade do negócio
  - Aumento de Performance
  - Redução de custo e gerenciamento simplificado

# Case Bling



Obrigado