

Sistemas Distribuídos

MongoDB e Sharding

Mariane Oliveira
Rafaela Ramos

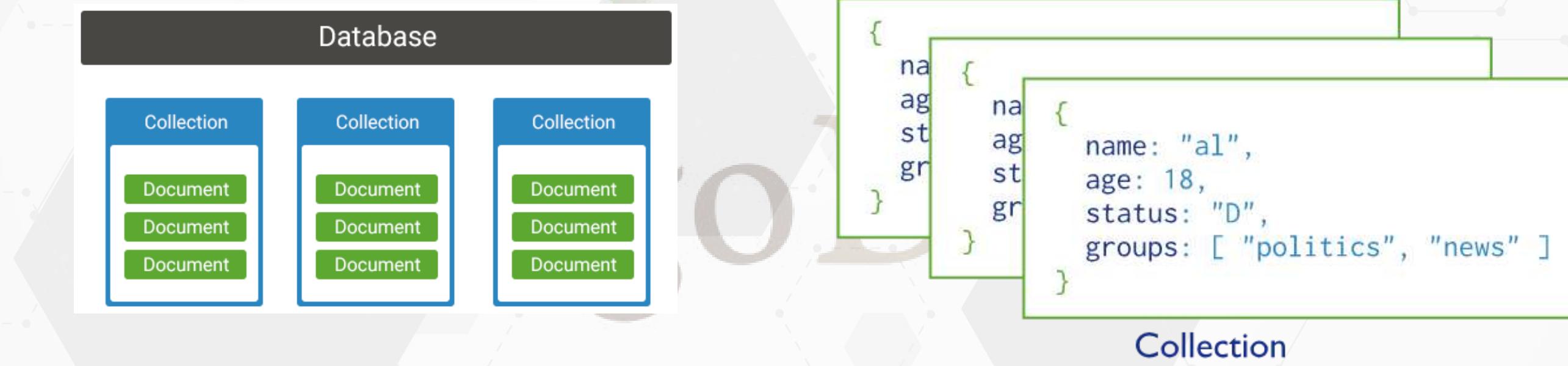
Estrutura Básica do MongoDB

- A estrutura fundamental do MongoDB difere dos bancos de dados relacionais tradicionais. Ela é baseada em:
 - **Documentos (Documents)**: Registros de dados.
 - Armazena dados como documentos JSON.
 - A estrutura é flexível: os campos podem variar de um documento para outro.
 - Suportam hierarquia e estruturas como arrays.
 - Internamente, o MongoDB converte esses documentos para um formato binário chamado BSON



Estrutura Básica do MongoDB

- A estrutura fundamental do MongoDB difere dos bancos de dados relacionais tradicionais (que usam tabelas e linhas). Ela é baseada em:
 - **Coleções (Collections)**: São grupos de documentos.
 - Funcionam de forma análoga às "tabelas" em bancos relacionais.
 - Não impõem um esquema rígido (schema), permitindo que documentos dentro da mesma coleção tenham campos diferentes.



Visão Geral e Recursos Principais

- O MongoDB é um banco de dados de documentos de propósito geral, ideal para desenvolvimento moderno e para a nuvem.

Recursos Principais:

- **Replica Sets:** Garante alta disponibilidade com cópias automáticas dos dados (mínimo de 3 instâncias).
- **Sharding:** Distribuição inteligente de dados em várias máquinas para escalabilidade.

Visão Geral e Recursos Principais

- O MongoDB é um banco de dados de documentos de propósito geral, ideal para desenvolvimento moderno e para a nuvem.

Recursos Principais:

- **Índices:** Aceleram consultas escaneando índices em vez de documentos inteiros.
- **Aggregation Pipelines:** Framework para processamento e análise de dados em escala
- **Versões:** Community (Open Source), Enterprise (Comercial) e Atlas (DBaaS na nuvem).

Diferença entre replicação e sharding

Replicação vs. Sharding

- **Sharding (Foco: Escalabilidade)**
 - **O que é:** Divisão dos dados em fragmentos menores (shards).
 - **Objetivo:** Reduzir a carga em um único servidor.
 - **Benefício:** Permite crescimento ilimitado do volume de dados (Escala Horizontal).

Diferença entre replicação e sharding

Replicação vs. Sharding

- **Replicação (Foco: Disponibilidade)**
 - **O que é:** Criação de cópias idênticas dos dados em vários servidores.
 - **Objetivo:** Redundância e segurança contra falhas.
 - **Benefício:** Garante que o sistema continue online se um servidor cair e agiliza leituras.

Benefícios

- Melhorar o tempo de resposta
- Evitar interrupção total do serviço
- Escalar com eficiência

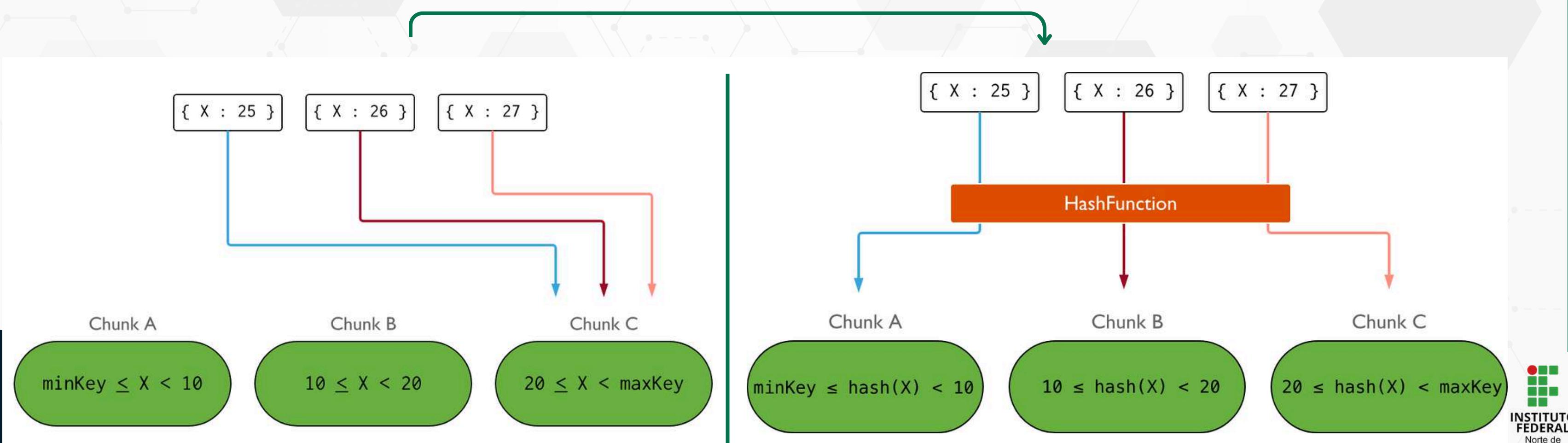
Desafios

- **Hotspots de dados (Desequilíbrio)**
 - Solução: Distribuição uniforme com chaves otimizadas.
- **Complexidade operacional**
 - Solução: Automação e ferramentas simplificando o gerenciamento da arquitetura (como na AWS).
- **Custos de infraestrutura**
 - Solução: Infraestrutura virtual em nuvem (como a EC2).
- **Complexidade da aplicação**
 - Solução: Bancos de dados criados especificamente com recursos de escalabilidade horizontal integrados.

Estratégias de particionamento

Fragmentação baseada em hash

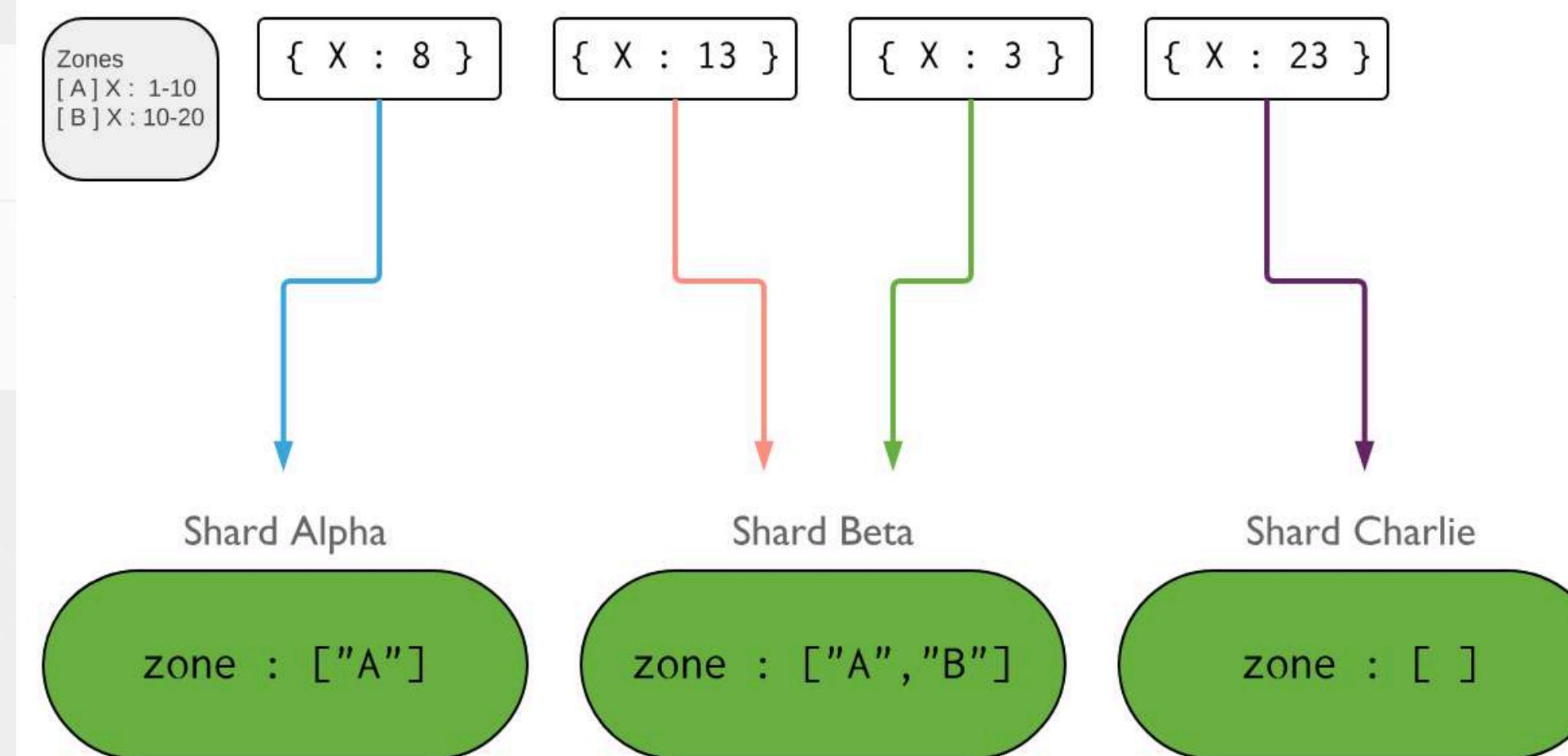
Distribui os dados de forma equilibrada entre os shards com base em uma função hash aplicada aos valores das chaves dos fragmentos.



Estratégias de particionamento

Fragmentação baseada em intervalos/alcance

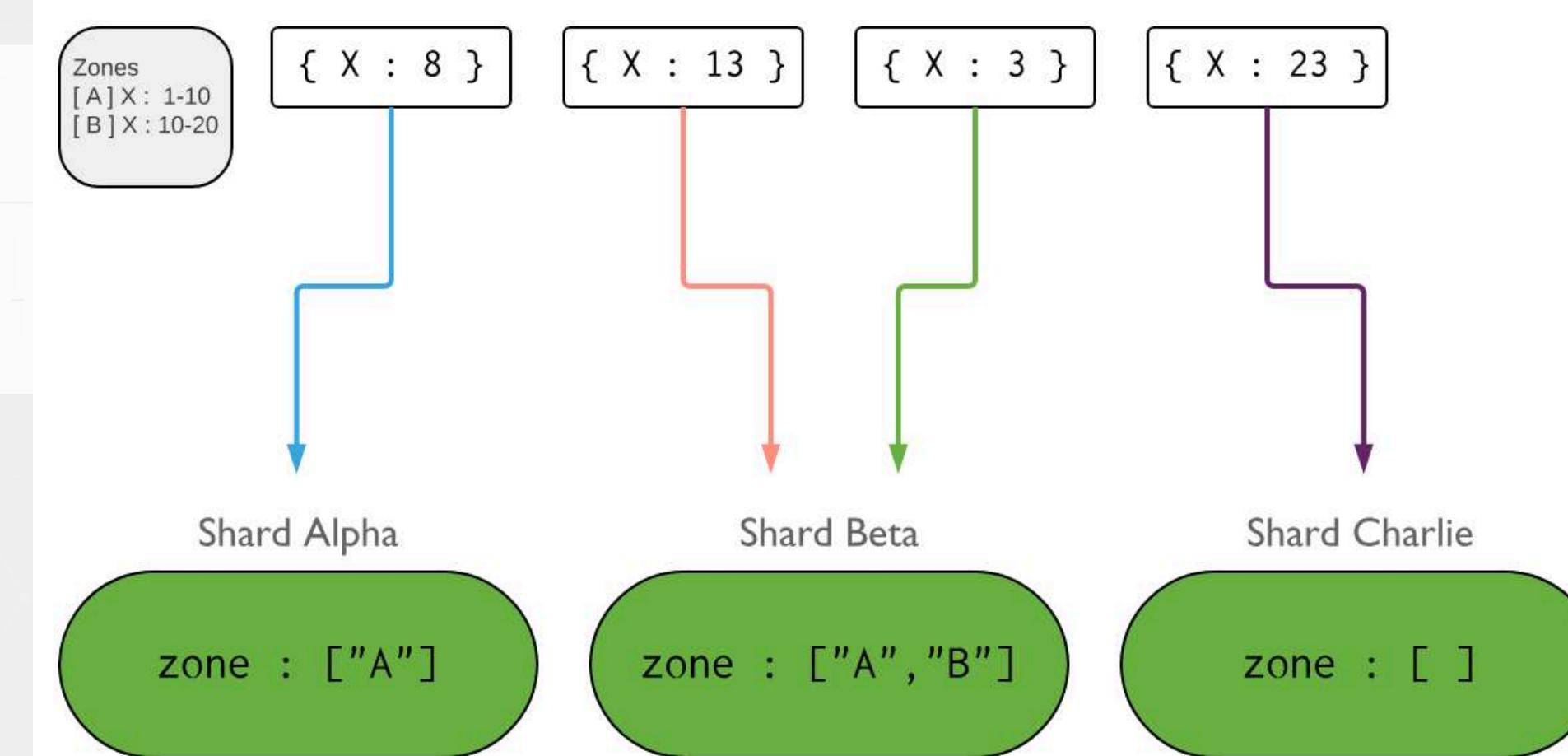
Divide os dados com base em um intervalo definido de valores de chave de fragmentação.



Estratégias de particionamento

Fragmentação baseada em zonas/localização

Organização geográfica dos dados em “zonas”, onde cada zona é associada a um ou mais shards.



Casos práticos

Útil para aplicações que gerenciam dados em larga escala e/ou que tenham dados acessados continuadamente:

1

Plataformas de Comércio Eletrônico (e-commerce)

2

Redes Sociais

3

Dispositivos IOT

4

Serviço de streaming.



INSTITUTO
FEDERAL
Norte de
Minas Gerais

Implementação Prática

Cenário: Implementação de Arquitetura Distribuída e Escalável com MongoDB na AWS.

Topologia do Cluster:

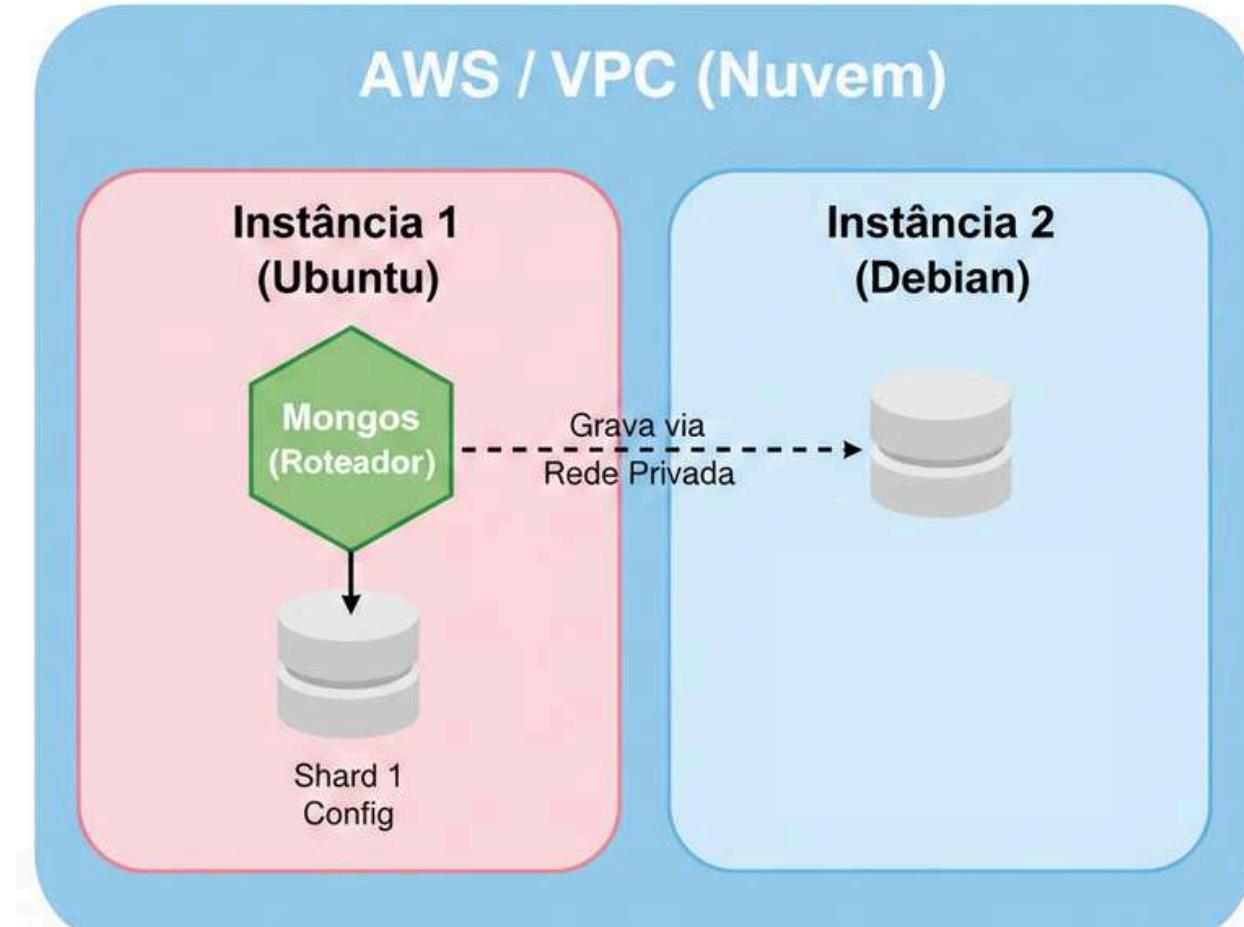
- **instancia1:** Mongos (Roteador), Config Server e Shard 1.
- **instancia2:** Shard 2 (Armazenamento Remoto).

Engenharia de Rede:

- **Containers Docker isolam redes por padrão (bridge).**
- **Solução:** Implementação de network_mode: "host". Isso permitiu que o Cluster utilizasse a infraestrutura de rede da Amazon para comunicação segura e direta entre as instâncias.



Implementação Prática



Obrigada!