

## F1E

Ich kann ein Konzept für die Skalierung einer NoSQL Datenbank erstellen.

### Lernziele

- Skalierung in NoSQL einer Datenbanken verstehen.
- Analyse der Skalierungsmöglichkeiten in NoSQL einer Datenbanken machen.
- Konzept für die Skalierung in NoSQL einer Datenbanken erstellen.

### Umsetzung

#### Skalierung in NoSQL-Datenbanken

In den letzten beiden Kapiteln haben wir die Grundlagen der Replikation und Skalierung in NoSQL-Datenbanken behandelt. Skalierung ist ein entscheidender Aspekt, um die Leistungsfähigkeit und Verfügbarkeit von NoSQL-Datenbanken zu gewährleisten. Replicasets ermöglichen es, Daten über mehrere Knoten zu verteilen und so die Last zu verteilen.

#### Analyse der Skalierungsmöglichkeiten

Die Skalierungsmöglichkeiten in NoSQL-Datenbanken können in zwei Hauptkategorien unterteilt werden:

- **Vertikale Skalierung (Scale Up):**

Hierbei wird die Kapazität eines einzelnen Knotens erhöht, indem mehr Ressourcen (CPU, RAM, Speicher) hinzugefügt werden. Dies ist einfach umzusetzen, hat aber Grenzen in Bezug auf Hardware und Kosten.

- **Horizontale Skalierung (Scale Out):**

Hierbei werden mehrere Knoten hinzugefügt, um die Last zu verteilen. Dies ermöglicht eine nahezu unbegrenzte Skalierung, da neue Knoten einfach hinzugefügt werden können. NoSQL-Datenbanken sind in der Regel für horizontale Skalierung optimiert.

#### Konzept für die Skalierung in NoSQL-Datenbanken

Ein Konzept für die Skalierung in NoSQL-Datenbanken sollte folgende Aspekte berücksichtigen:

- **Anforderung der Skalierung:**

Zunächst sollte die erwartete Last und das Wachstum der Datenbank analysiert werden. Dies hilft, die geeignete Skalierungsstrategie zu wählen.

- **Lastverteilung:**

Die Last sollte gleichmäßig auf alle Knoten verteilt werden, um Engpässe zu vermeiden. Dies kann

Die Last sollte gleichmässig auf alle Knoten verteilt werden, um Engpässe zu vermeiden. Dies kann durch Sharding erreicht werden, bei dem Daten in kleinere Teile (Shards) aufgeteilt und über mehrere Knoten verteilt werden.

- **Replikation:**

Replikation sollte implementiert werden, um die Verfügbarkeit und Fehlertoleranz zu erhöhen. Ein Replica Set ermöglicht es, Daten über mehrere Knoten zu replizieren, sodass bei einem Ausfall eines Knotens die Daten weiterhin verfügbar sind.

- **Monitoring und Skalierung:**

Ein effektives Monitoring-System sollte implementiert werden, um die Leistung der Knoten zu überwachen und bei Bedarf neue Knoten hinzuzufügen. Automatisierte Skalierung kann helfen, die Ressourcen dynamisch anzupassen, basierend auf der aktuellen Last.

Mit diesen Konzepten kann eine NoSQL-Datenbank effizient skaliert werden, um den Anforderungen moderner Anwendungen gerecht zu werden.

## Visualisierung

