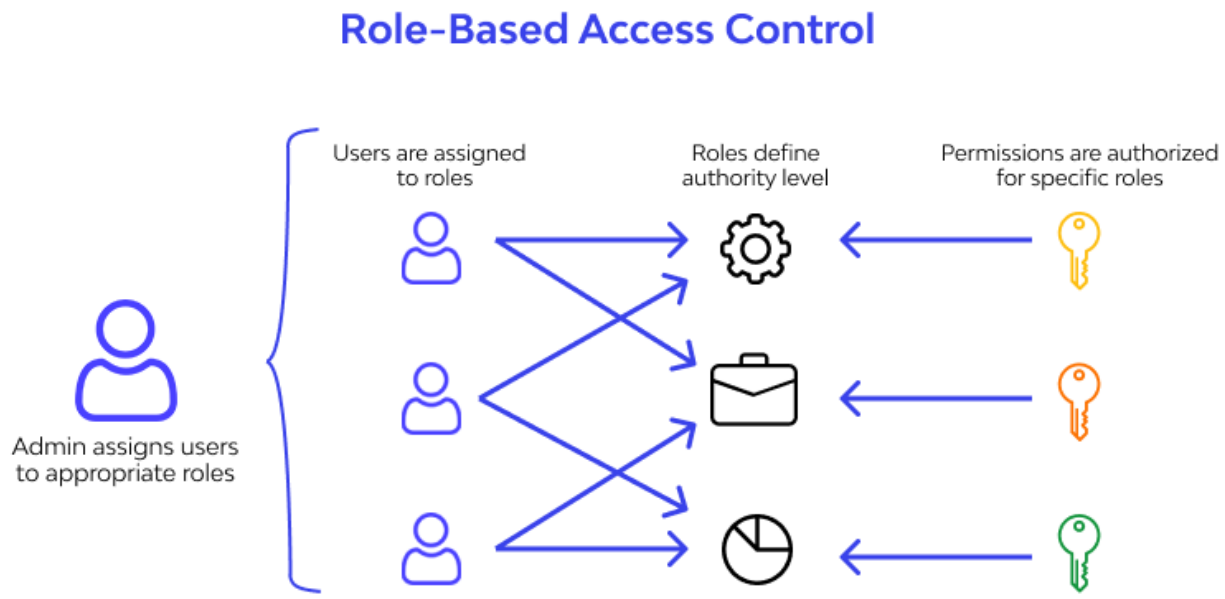
D1G

Ich kann die Funktion von Zugriffsberechtigungen in einer NoSQL Datenbank erläutern. (Benutzer, Rollen, Zugriffsrechte)

Fragenstellung und Lernziele

- **Was sind Zugriffsberechtigungen in einer NoSQL Datenbank?** Verstehen, wie Benutzer, Rollen und Zugriffsrechte zusammenspielen, um den Zugriff auf Daten zu steuern.
- **Wie werden Benutzer in einer NoSQL Datenbank angelegt und authentifiziert?** Erlernen, wie individuelle Benutzerkonten erstellt werden und welche Authentifizierungsmethoden eingesetzt werden.
- **Welche Funktion haben Rollen und Zugriffsrechte?** Erkennen, wie Rollen zur Bündelung von Berechtigungen verwendet werden und wie Zugriffsrechte konkret festlegen, was ein Benutzer oder eine Rolle tun darf.
- **Wie unterscheidet sich die Zugriffsverwaltung in NoSQL Datenbanken von relationalen Systemen?** Die speziellen Anforderungen und Flexibilitäten der NoSQL-Zugriffsverwaltung im Vergleich zu traditionellen, schema-gebundenen Systemen verstehen.

Umsetzung



In NoSQL Datenbanken erfolgt die Zugriffskontrolle über ein mehrschichtiges Sicherheitsmodell. Dieses Modell basiert auf der Verwaltung von Benutzern, der Zuweisung von Rollen und der Definition spezifischer Zugriffsrechte.

Benutzerverwaltung

Definition und Bedeutung: Benutzer sind individuelle Konten, die zur Authentifizierung benötigt werden. Jeder Benutzer erhält ein Passwort oder andere Authentifizierungsmerkmale, um seine Identität zu bestätigen.

Authentifizierung:

Methoden umfassen Passwörter, Tokens oder Zertifikate, die den sicheren Zugang gewährleisten.

Beispiel:

Ein Benutzerkonto kann folgendermassen definiert sein:

```
{
  "user": "alice",
  "pwd": "sicheresPasswort123",
  "roles": ["readWrite"]
}
```

Dieses Beispiel zeigt, wie ein Benutzer mit einem Passwort und einer Rolle in der Datenbank definiert wird.

```
db.createUser({
  user: "alice",
  pwd: "sicheresPasswort123",
  roles: [{ role: "readWrite", db: "exampleDB" }]
});
```

Dieses Beispiel zeigt, wie ein Benutzer zusammen mit seinen zugewiesenen Rollen in der Datenbank definiert wird.

Rollen:

- Rollen fassen mehrere Zugriffsrechte zusammen und können mehreren Benutzern zugewiesen werden.
- Sie vereinfachen die Verwaltung, indem sie Berechtigungen zentral definieren.

Zugriffsrechte:

- Zugriffsrechte bestimmen, welche Operationen (wie Lesen, Schreiben, Aktualisieren oder Löschen) ein Benutzer oder eine Rolle ausführen darf.
- Typische Rechte sind beispielsweise `read`, `write`, `dbAdmin` und `clusterAdmin`.

Beispiel: Eine Rolle mit Lese- und Schreibrechten könnte folgendermassen aussehen:

```
{
  "role": "readWrite",
  "db": "exampleDB",
  "privileges": [
    { "resource": { "db": "exampleDB", "collection": "documents" }, "acti
  ],
  "roles": []
}
```

Dieses Beispiel verdeutlicht, wie Rollen Zugriffsrechte bündeln und gezielt zugewiesen werden können.

Zugriffskontrolle in NoSQL Datenbanken

Implementierung:

- Die Zugriffskontrolle erfolgt durch interne Sicherheitsmodelle, wie zum Beispiel den Authentifizierungsmechanismus in MongoDB.
- Administratoren definieren, welche Benutzer und Rollen existieren und welche spezifischen Rechte ihnen zugewiesen werden.

Unterschied zu relationalen Datenbanken:

- Relationale Datenbanken besitzen oft ein festes, tabellenbasiertes Rollenkonzept.
- NoSQL Datenbanken bieten mehr Flexibilität, um den verteilten und dynamischen Charakter moderner Anwendungen zu unterstützen.

Wichtige Begriffe und Strukturen

Benutzer: Individuelle Konten, die zur Authentifizierung und Autorisierung in der Datenbank verwendet werden.

Rollen: Gruppen von Zugriffsrechten, die mehreren Benutzern zugeordnet werden können.

Zugriffsrechte: Bestimmen, welche Aktionen (z. B. lesen, schreiben, administrieren) ein Benutzer oder eine Rolle durchführen darf.

Authentifizierung: Der Prozess, durch den die Identität eines Benutzers überprüft wird.

Autorisierung: Der Prozess, der festlegt, welche Operationen ein authentifizierter Benutzer ausführen darf.