

MA-CHAT

Morhaf Alnhlawe
Mouayad Alnhlawe
Inf21_A
CsBe (Computerschule Bern)
Version: 1.0

Inhaltsverzeichnis

Beschreibung der Systemteile und Vorschläge wie die jeweiligen Systemkomponenten Ausfallsicherheit gebaut werden können.	2
Script.js.....	2
Beschreibung	2
Verbesserung des Ausfallsicherheit.....	2
Api.js.....	3
Beschreibung	3
Verbesserung des Ausfallsicherheit.....	3
Database.js.....	3
Beschreibung	3
Verbesserung des Ausfallsicherheit.....	4
Websocketserver.js	4
Beschreibung	4
Verbesserung des Ausfallsicherheit.....	4
App.js.....	5
Beschreibung	5
Verbesserung des Ausfallsicherheit.....	5
Docker-compose.yaml	5
Beschreibung	5
Verbesserung des Ausfallsicherheit.....	5
Dockerfile.....	6
Beschreibung	6
Verbesserung des Ausfallsicherheit.....	6
Quellen.....	7

1 Beschreibung der Systemteile und Vorschläge wie die jeweiligen Systemkomponenten Ausfallsicherheit gebaut werden können.

1.1 Script.js

1.1.1 Beschreibung

Das Code zeigt die Implementierung eines einfachen Chat-Systems, das WebSocket und REST-API verwendet, um Nachrichten und Benutzerdaten zu senden und zu empfangen.

- WebSocket: Im Code wird die WebSocket-Verbindung verwendet, um Nachrichten zu empfangen und anzuzeigen, daneben ist es verwendet, um die Liste der aktiven Benutzer auf der Seite zu aktualisieren.
- REST-API: Im Code wird die REST-API verwendet, um die Liste der Chat-Nachrichten abzurufen und Benutzerdaten zu aktualisieren.

1.1.2 Verbesserung des Ausfallsicherheit

WebSocket: Um Ausfallsicherheit der WebSocket-Verbindung zu verbessern, können folgende Schritte unternommen werden:

- Verwendung von Load Balancern, um die Verbindungen auf mehrere Server zu verteilen, um die Last zu verteilen und die Verfügbarkeit zu erhöhen
- Verwendung von redundanten Servern, um eine hohe Verfügbarkeit und eine Ausfallsicherheit zu gewährleisten, falls ein Server ausfällt.

REST-API: Um Ausfallsicherheit der REST-API zu verbessern, können folgende Schritte unternommen werden:

- Implementierung von Caching-Techniken, um häufig angeforderte Daten zwischenspeichern und die Serverlast zu reduzieren.
- Verwendung von Lastausgleichern und redundanten Servern, um die Verfügbarkeit zu erhöhen und den Datenverkehr auf mehrere Server zu verteilen.
- Implementierung von Backups und Wiederherstellungslösungen, um Datenverluste zu vermeiden und die Verfügbarkeit zu erhöhen, falls ein Server ausfällt.

Insgesamt können die Ausfallsicherheit des Systems verbessert werden, indem redundante Server, Load Balancer und Backup-System implementiert werden. Es ist auch wichtig, regelmässige Wartungsarbeiten durchzuführen, um sicherzustellen, dass alle Systemkomponenten ordnungsgemäss funktionieren und auf dem neuesten Stand sind.

1.2 Api.js

1.2.1 Beschreibung

Das Code enthält eine Node.js-Anwendung, die eine API bereitstellt. Es gibt vier Hauptkomponenten: die Datenbank, den WebSocket-Server, die API-Endpunkt und die Funktionen für das Hinzufügen neuer Nachrichten und Benutzer sowie für das Aktualisieren von Benutzerdaten.

1.2.2 Verbesserung des Ausfallsicherheit

Um die Ausfallsicherheit der Anwendung zu verbessern, gibt es verschiedene Ansätze, die für jede Komponente angewendet werden können:

Datenbank:

- Ein Backup-System kann eingerichtet werden, um im Falle eines Datenbankausfalls schnell eine Wiederherstellung durchführen zu können.
- Es ist ratsam, regelmäßige Backups durchzuführen und diese an einem sicheren Ort aufzubewahren, um Verluste von wichtigen Daten zu vermeiden.
- die Verwendung von Hochverfügbarkeits-Datenbanken oder Clustern, um sicherzustellen, dass die Datenbank immer verfügbar ist.

WebSocket-Server und API-Endpunkte:

- Verwendung von Lastausgleichern, um sicherzustellen, dass mehrere Serverinstanzen verfügbar sind, wenn eine Instanz ausfällt.

1.3 Database.js

1.3.1 Beschreibung

Der Code implementiert eine Datenbankverbindung und Methoden zu Ausführung von SQL-Abfrage sowie zur Initialisierung der Datenbankschemata. Es besteht aus den folgenden Systemkomponenten:

- Datenbankpool: Dies ist ein Pool von Verbindung zu Datenbank, der erstellt wird, wenn die Methode «initializeMariaDB» aufgerufen wird. Der Pool wird erstellt, um eine begrenzte Anzahl von Verbindungen zur Datenbank bereitzustellen und gleichzeitig eine Überlastung der Datenbank zu vermeiden.
- executeSQL-Funktion: Diese Funktion führt SQL-Abfrage über eine Verbindung aus dem Datenbankpool aus und gibt das Ergebnis zurück. Wenn ein Fehler auftritt, wird eine Fehlermeldung ausgegeben und die Verbindung wird geschlossen.
- initializeDBSchema-Funktion: Diese Funktion initialisiert das Datenbankschema, indem sie Tabellen für Benutzer und Nachrichten erstellt.

1.3.2 Verbesserung des Ausfallsicherheit

- Datenbankpool: Um den Datenbankpool ausfallsicherer zu gestalten, kann die Anzahl der Verbindungen erhöht werden, um den Bedarf zu decken, und es kann eine Überwachungsfunktion implementiert werden, um den Pool auf Probleme wie Verbindungsfehler und Ausnahmen zu überwachen und gegebenenfalls neu zu starten.
- executeSQL-Funktion: Um die executeSQL-Funktion ausfallsicherer zu gestalten, kann ein erneuter Verbindungsversuch durchgeführt werden, wenn die Verbindung unterbrochen wird, oder ein neuer Pool von Verbindungen kann erstellt werden, wenn der ursprüngliche Pool nicht mehr verfügbar ist.
- initializeDBSchema-Funktion: Um initializeDBSchema-Funktion ausfallsicherer zu machen, können Überprüfungen und Rückmeldungen für erfolgreiche Initialisierungen hinzugefügt werden, und es können Mechanismen zum automatischen Neuerstellen von Tabellen implementiert werden, wenn diese gelöscht oder beschädigt werden. Auch hier kann eine Überwachungsfunktion hinzugefügt werden, um die Integrität der Tabellen zu überwachen und auf Probleme zu reagieren.

Zusammenfassend kann die Ausfallsicherheit der Systemkomponenten durch Implementierung von Überwachungsfunktionen, erhöhte Anzahl von Verbindungen und automatisches neuerstellen von Tabellen verbessert werden.

1.4 Websoketserver.js

1.4.1 Beschreibung

Dieser Code ist für die Initialisierung eines WebSocket-Servers, der es den Benutzern ermöglicht, in Echtzeit miteinander zu kommunizieren. Das WebSocket-Protokoll ist eine besser Alternative zu http-Protokoll, da es eine Verbindung zwischen Server und Client aufrechterhält, was eine schnellere Kommunikation ermöglicht.

1.4.2 Verbesserung des Ausfallsicherheit

Für eine höhere Ausfallsicherheit können folgende Vorschläge umgesetzt werden:

- Skalierung des Servers: Ein WebSocket-Server kann horizontal skaliert werden, indem mehrere Instanzen gestartet werden. Dadurch kann die Last auf mehrere Server verteilt werden und die Ausfallzeit reduziert werden, falls ein Server ausfällt.
- Verwendung von Load Balancer: Ein Load Balancer kann dazu beitragen, dass eingehende Verbindungen auf mehrere Server verteilt werden, wodurch die Auslastung reduziert, und die Verfügbarkeit erhöht wird.

1.5 App.js

1.5.1 Beschreibung

Der Code zeigt die Implementierung des Web-Servers mit einem WebSocket-Server und einer REST-API. Der Server wird mit Express erstellt und auf einem http-Server ausgeführt. Eine MariaDB-Datenbank wird initialisiert und ein Datenbankschema wird erstellt.

1.5.2 Verbesserung des Ausfallsicherheit

Um die Ausfallsicherheit des Servers zu erhöhen, können folgende Vorschläge umgesetzt werden:

- Verwendung von Load Balancer: Ein Load Balancer kann dazu beitragen, dass eingehende Verbindungen auf mehrere Server verteilt werden. Wenn ein Server ausfällt, können die anderen Server den Traffic aufnehmen und die Verfügbarkeit des Services aufrechterhalten.
- Durch die Verwendung von redundanter Hardware oder virtuellen Maschinen kann ein Ausfall eines Servers ohne Auswirkungen auf den Service verhindert werden.
- Eine passende Überwachung des Servers und der Anwendungen kann frühzeitig Ausfälle erkennen und möglicher Probleme beheben, bevor sie zu grösseren Problemen führen.
- Eine Skalierung der Infrastruktur kann dazu beitragen, Engpässe zu beseitigen und eine bessere Verfügbarkeit des Services zu gewährleisten. Durch die Verwendung von Cloud-Services können schnell und einfach zusätzliche Ressourcen hinzugefügt werden, um die Skalierbarkeit des Systems zu erhöhen.
- Regelmässige Backups können dazu beitragen, den Datenverlust in Falle eines Ausfalls zu minimieren. Die Backups sollten an einem sicheren Ort gespeichert werden und regelmässig überprüft werden, um sicherzustellen, dass sie funktionieren.
- Eine umfassende Risikoanalyse und ein Plan für die Notfallwiederherstellung können jedoch dazu beitragen, Ausfälle zu minimieren und Verfügbarkeit des Services zu erhöhen.

1.6 Docker-compose.yaml

1.6.1 Beschreibung

Das Datei beschreibt, wie zwei Services in einer Anwendung miteinander verknüpft werden können: MariaDB und phpMyAdmin.

1.6.2 Verbesserung des Ausfallsicherheit

Um die Ausfallsicherheit der Komponenten zu erhöhen, können mehrere Massnahmen ergriffen werden.

- Die Datenbankdaten regelmässig zu sichern und die Backups auf einem separaten Server oder in einem Cloud-Speicher zu speichern.
- Eine Verteiltes Datenbankarchitektur zu verwenden, um mehrere Datenbankinstanzen zu haben, die die Last gemeinsam tragen können und eine höhere Verfügbarkeit bieten.
- Die Container-Images regelmässig aktualisieren und Sicherheitsupdates installieren, um bekannte Schwachstellen zu beheben und die Systeme gegen Angriffe zu schützen.

1.7 Dockerfile

1.7.1 Beschreibung

Dieses Dockerfile definiert den Aufbau eines Docker-Images, das eine Node.js-Anwendung enthält.

1.7.2 Verbesserung des Ausfallsicherheit

Um die Ausfallsicherheit dieser Anwendung zu erhöhen, könnten verschieden Massnahmen ergriffen werden:

- Verwendung eines Load-Balancers: Indem mehrere Instanzen der Anwendungen gestartet werden und der Traffic auf diese Instanzen verteilt wird, kann die Ausfallsicherheit erhöht werden.
- Verwendung von Containern-Orchestrierungswerkzeugen wie Kubernetes oder Docker Swarm: Diese Tools ermöglichen es, eine Gruppe von Container zu verwalten und bei Bedarf automatisch neue Instanzen zu starten oder ausfallende Instanzen zu ersetzen.
- Verwendung von automatisierten Tests und Überwachung: Durch das regelmässige Testen der Anwendung und die Überwachung ihrer Leistung können Potenzielle Probleme frühzeitig erkannt und behoben werden.

Die Umsetzung der Applikation ist erfolgreich abgeschlossen, aber die Änderung des Benutzer-Name kann nur in Datenbank angezeigt, leider wird der Name in das Frontend (clientseitig) nicht aktualisiert bzw. nicht angezeigt.

2 Quellen

(Corey, 2023)

(stackoverflow, 2023)

(medium, 2023)