Cloud-Computing-Projekt

Projektgruppe

15. Januar 2026

Inhaltsverzeichnis

1 Firebase Authentication

1.1 Zugriff auf den Authentication-Bereich

Die Konfiguration der Benutzerverwaltung erfolgt über die Firebase Console. Nach Auswahl des Projekts *Doku* wird im linken Navigationsmenü der Bereich *Authentication* ausgewählt. Dieser Bereich dient als zentrale Verwaltungsoberfläche für alle Aspekte der Authentifizierung.

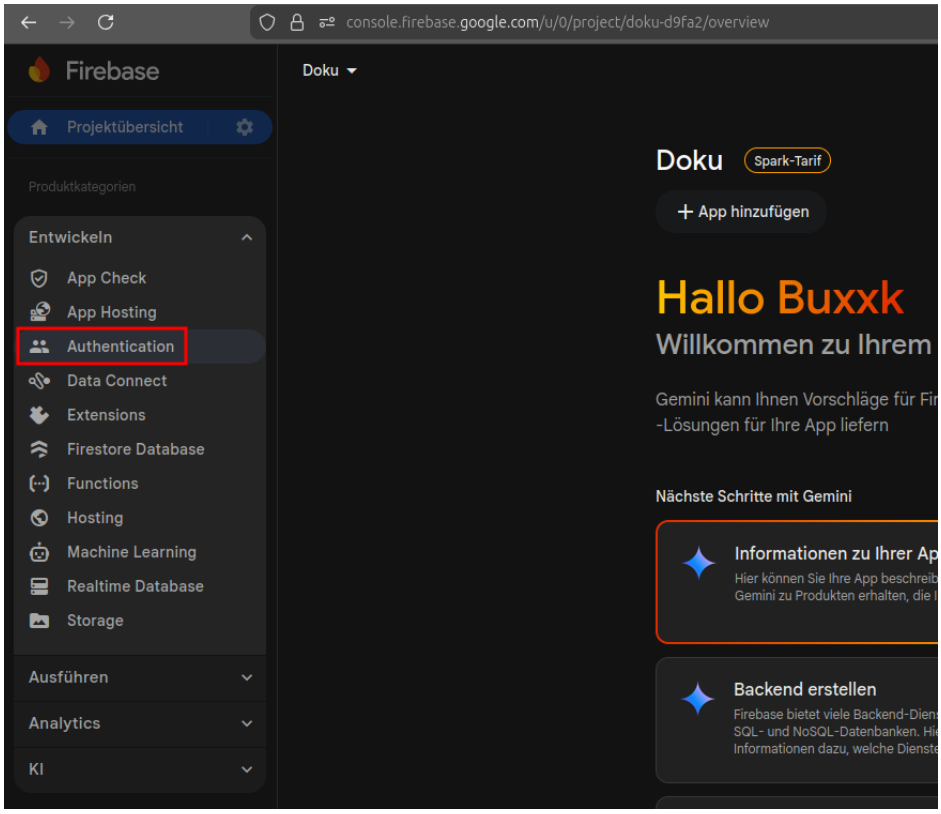

Abbildung 1: Aufruf des Authentication-Bereichs in der Firebase Console

1.2 Übersicht der Nutzerverwaltung

Nach dem Aufruf des Authentication-Moduls wird standardmäßig der Reiter *Nutzer* angezeigt. In diesem Bereich verwaltet Firebase alle registrierten Benutzerkonten der Anwendung.

Für jeden Nutzer speichert Firebase unter anderem:

- den verwendeten Authentifizierungsanbieter,

- den Zeitpunkt der Erstellung des Kontos,
- den Zeitpunkt der letzten Anmeldung,
- eine eindeutige Benutzer-ID (UID).

Die UID dient als stabiler Identifikator und wird für die Zuordnung von benutzerspezifischen Daten sowie für Autorisierungsentscheidungen in nachgelagerten Services verwendet.

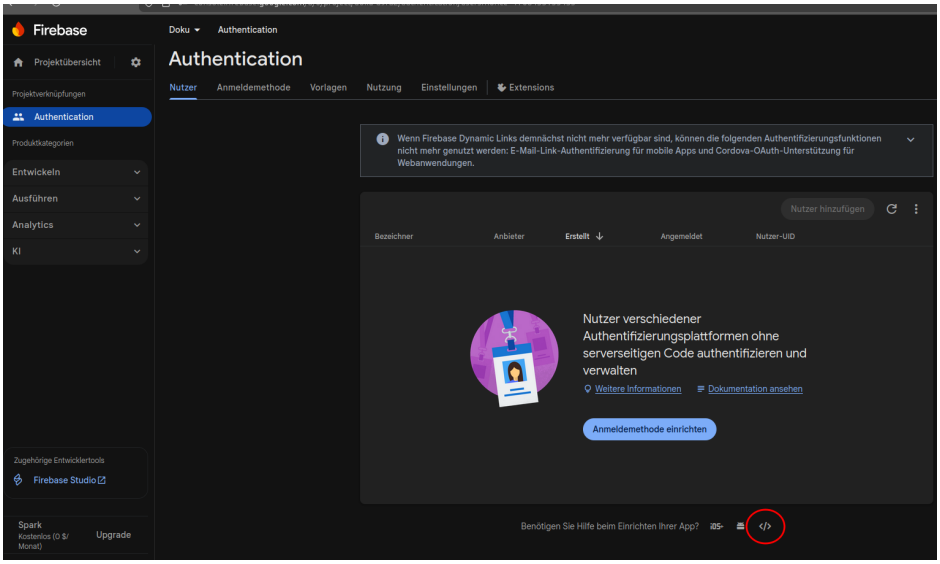

Abbildung 2: Nutzerübersicht innerhalb von Firebase Authentication

1.3 Konfiguration der Anmeldemethoden

Die verfügbaren Anmeldemethoden werden im Reiter *Anmeldemethode* konfiguriert. In diesem Projekt wurden bewusst mehrere Authentifizierungsanbieter aktiviert, um unterschiedliche Anmeldewege zu ermöglichen.

Konkret wurden folgende Anbieter eingerichtet:

- **E-Mail-Adresse / Passwort** als klassische Form der Anmeldung,
- **Google** als externer OAuth-2.0-Anbieter.

Die Aktivierung erfolgt über die Schaltfläche *Neuen Anbieter hinzufügen*. Jeder Anbieter kann unabhängig voneinander aktiviert oder deaktiviert werden.

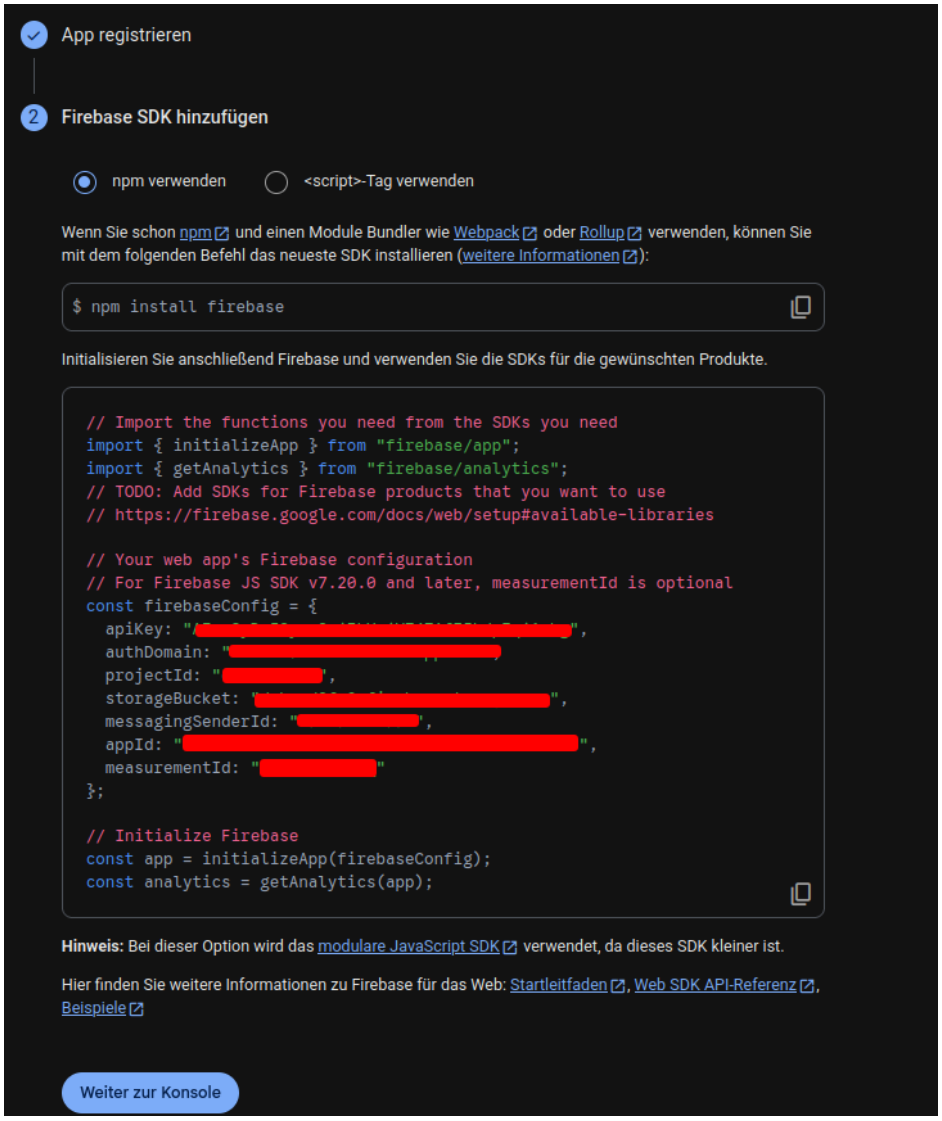

Abbildung 3: Auswahl und Aktivierung der Anmeldemethoden

1.4 Registrierung der Web-Anwendung

Um Firebase Authentication im Frontend nutzen zu können, wurde eine Web-Anwendung im Firebase-Projekt registriert. Dieser Schritt ist notwendig, damit das Frontend eindeutig dem Projekt zugeordnet werden kann.

Nach der Registrierung stellt Firebase eine projektspezifische Konfiguration zur Verfügung, die unter anderem folgende Parameter enthält:

- `apiKey`

- `authDomain`
- `projectId`
- `storageBucket`
- `appId`

Diese Konfigurationsdaten werden im Frontend verwendet, um das Firebase SDK zu initialisieren und eine Verbindung zu den Firebase-Diensten herzustellen.

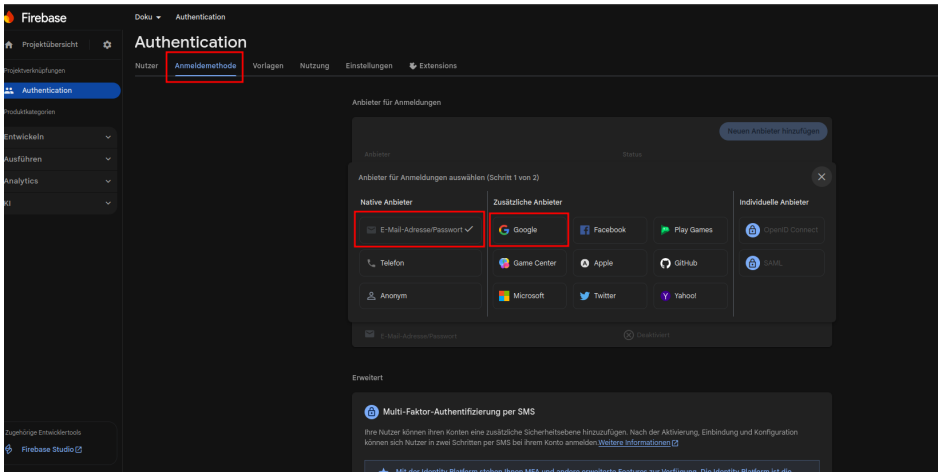

Abbildung 4: Firebase SDK Konfiguration für die Web-Anwendung

1.5 Einbindung des Firebase SDK

Für die Einbindung von Firebase im Frontend wurde das modulare JavaScript SDK verwendet. Die Installation erfolgt über den Node Package Manager (npm). Nach der Initialisierung des SDK kann die Authentication-Funktionalität direkt im Client genutzt werden.

Die Authentifizierung erfolgt vollständig über Firebase, wodurch keine Passwörter oder sicherheitskritischen Zugangsdaten im eigenen Backend gespeichert oder verarbeitet werden müssen.

1.6 Zusammenfassung

Firebase Authentication übernimmt in diesem Projekt die vollständige Benutzerverwaltung und Authentifizierung. Durch die Nutzung eines externen 3rd-Party-Dienstes wird eine sichere, skalierbare und wartbare Lösung bereitgestellt, die sich nahtlos in weitere Google-Cloud-Dienste integrieren lässt.

2 Backend-Services mit Cloud Run Functions

2.1 Ziel der Backend-Funktionalität

Zur Umsetzung der serverseitigen Logik wurde Google Cloud Run in Verbindung mit Cloud Functions eingesetzt. Ziel ist es, dem Frontend eine klar definierte REST-Schnittstelle bereitzustellen, über die sicher auf Anwendungsdaten, Datenbanken und Cloud-Storage-Ressourcen zugegriffen werden kann.

Direkte Zugriffe aus dem Frontend auf Datenbank oder Storage werden dabei bewusst vermieden. Sämtliche Datenmanipulationen erfolgen ausschließlich über die bereitgestellten Backend-Services.

2.2 Aufruf von Cloud Run in der Google Cloud Console

Die Erstellung und Verwaltung der Backend-Services erfolgt über die Google Cloud Console. Nach Auswahl des Projekts wird im Navigationsmenü der Dienst *Cloud Run* aufgerufen. Innerhalb dieses Bereichs werden alle laufenden Services zentral angezeigt und verwaltet.

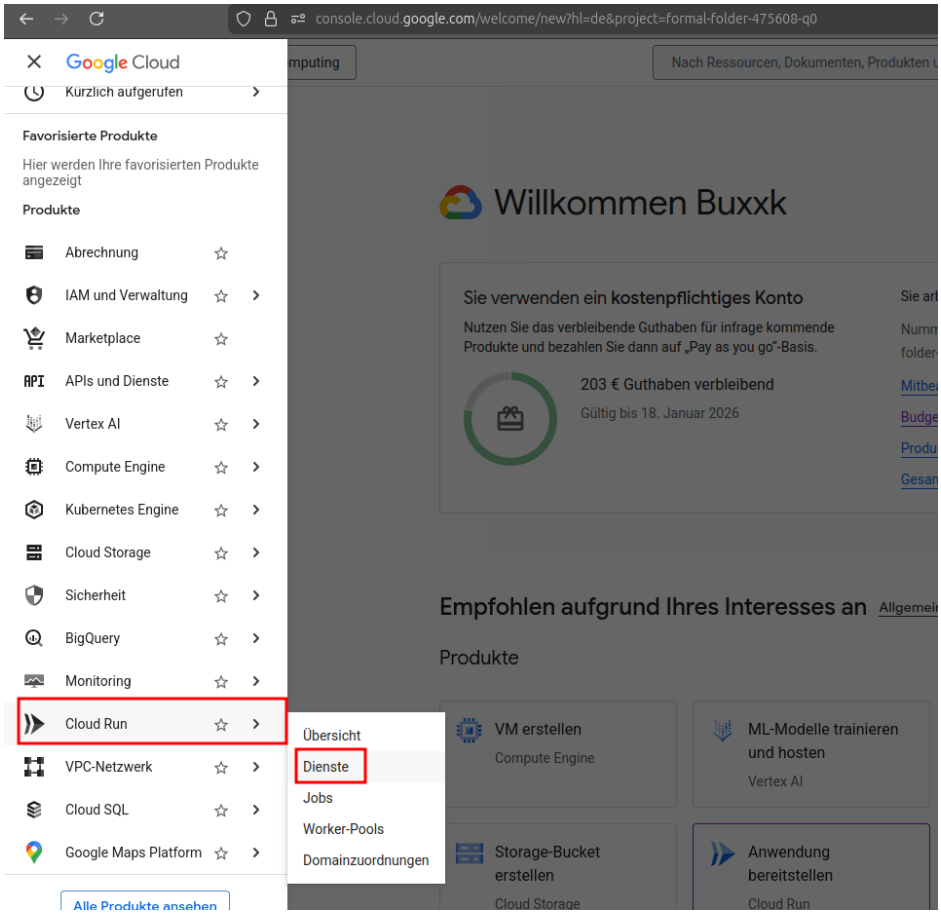

Abbildung 5: Navigation zu Cloud Run innerhalb der Google Cloud Console

2.3 Übersicht der bestehenden Services

In der Serviceübersicht werden alle bereitgestellten Cloud-Run-Services angezeigt. Jeder Service stellt einen eigenständigen HTTP-Endpunkt dar und kann unabhängig von anderen Services skaliert, aktualisiert oder neu bereitgestellt werden.

In diesem Projekt werden mehrere Services eingesetzt, darunter:

- API-Services zur Verarbeitung von Anwendungsdaten,
- Funktionen zum Upload von Dateien,
- Funktionen zum Abruf von Inhalten aus dem Cloud Storage.

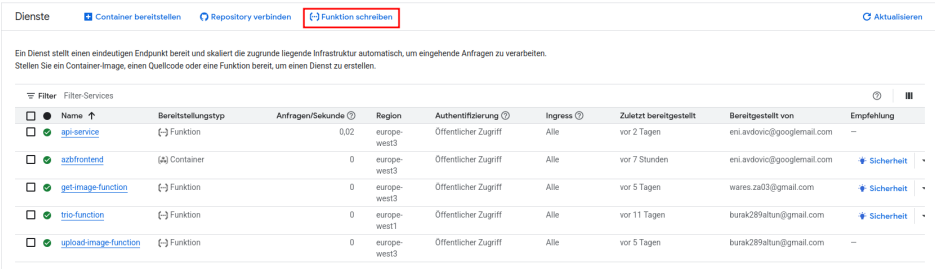

Abbildung 6: Übersicht der bestehenden Cloud-Run-Services

2.4 Erstellung einer neuen Cloud-Run-Function

Zur Erstellung eines neuen Backend-Services wurde die Option *Funktion schreiben* gewählt. Dabei handelt es sich um eine Cloud-Run-Function, die direkt über einen Inline-Editor erstellt und bereitgestellt werden kann.

Während der Erstellung werden grundlegende Parameter definiert, darunter:

- der eindeutige Name des Services,
- die Region (z. B. europe-west3),
- die verwendete Laufzeitumgebung (Node.js),
- die Authentifizierungsart,
- das Abrechnungsmodell.

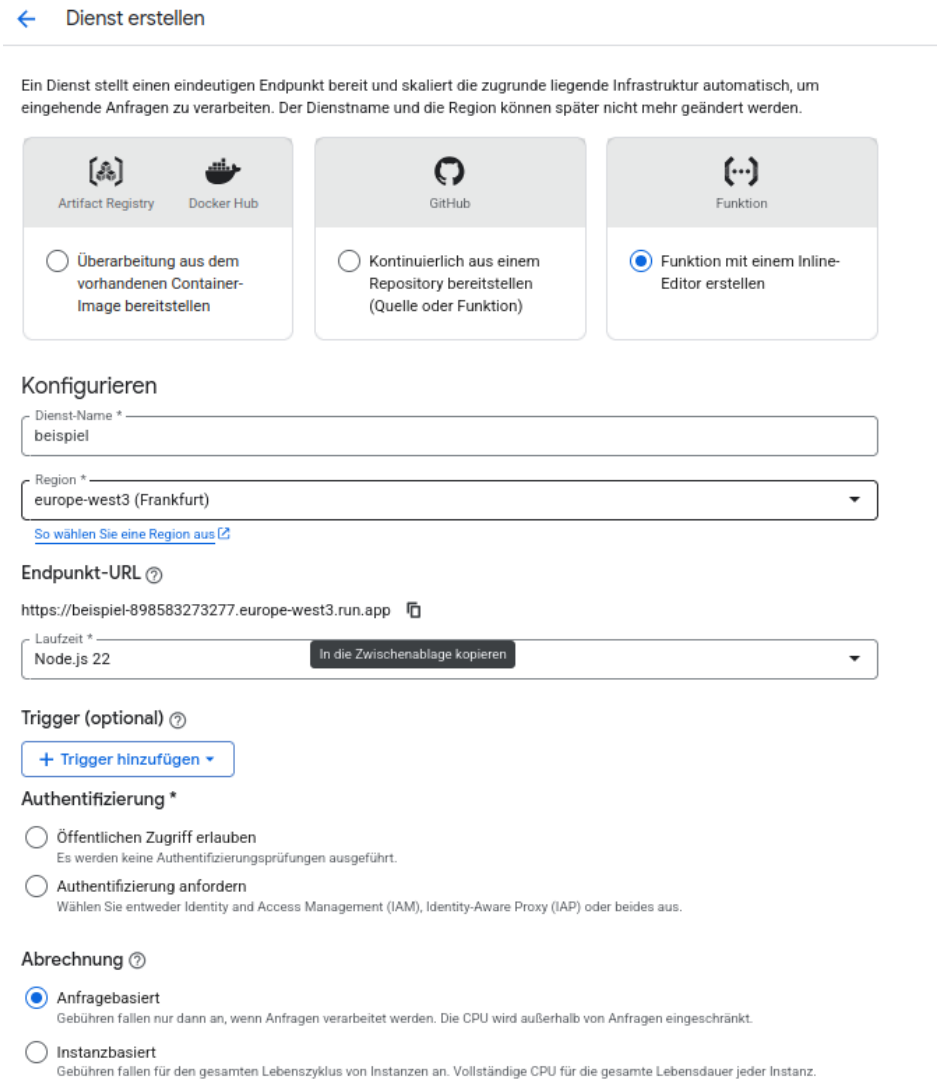

Abbildung 7: Konfiguration eines neuen Cloud-Run-Services

2.5 Authentifizierung, Skalierung und Abrechnung

Für den erstellten Service wurde ein öffentlicher HTTP-Endpunkt bereitgestellt, sodass das Frontend direkt mit der Cloud-Run-Function kommunizieren kann. Die Zugriffskontrolle kann optional über IAM oder Identity Tokens erweitert werden.

Die Skalierung erfolgt automatisch auf Basis der eingehenden Anfragen. Die minimale Anzahl an Instanzen wurde auf null gesetzt, wodurch im Leerlauf keine Kosten entstehen.

Die Abrechnung erfolgt anfragebasiert, sodass Kosten nur bei tatsächlicher Nutzung des Services anfallen.

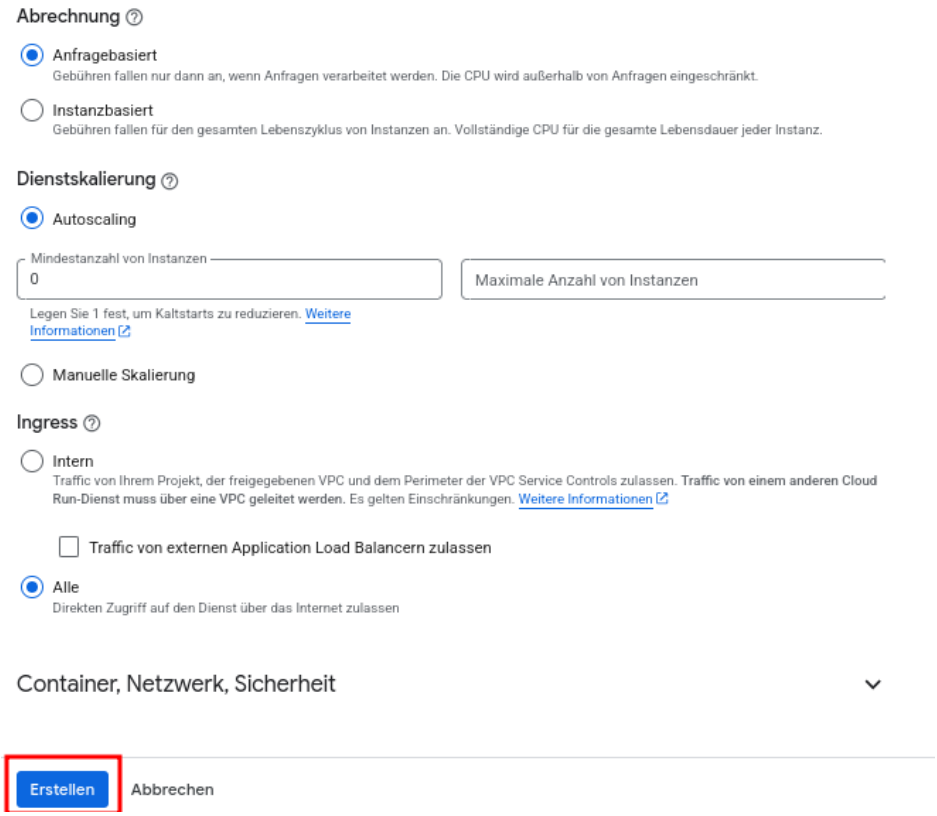

Abbildung 8: Abrechnungs- und Skalierungseinstellungen des Cloud-Run-Services

2.6 Einbindung in die Gesamtarchitektur

Die Cloud-Run-Functions bilden die zentrale Schnittstelle zwischen Frontend, Datenbank und Cloud Storage. Das Frontend kommuniziert ausschließlich über diese Services mit dem Backend.

Durch diese Architektur wird sichergestellt, dass:

- keine direkten Datenbankzugriffe aus dem Client erfolgen,
- sensible Operationen serverseitig kontrolliert werden,
- die Anwendung horizontal skalierbar bleibt,
- sicherheitsrelevante Logik zentral im Backend gebündelt ist.

3 Identity and Access Management (IAM)

3.1 Ziel des IAM-Konzepts

Zur Absicherung der Cloud-Infrastruktur wurde Google Cloud Identity and Access Management (IAM) eingesetzt. Ziel ist es, Zugriffe auf Cloud-Ressourcen granular zu steuern und sicherzustellen, dass jede Komponente der Anwendung nur über die minimal notwendigen Berechtigungen verfügt.

IAM bildet damit die Grundlage für:

- kontrollierten Zugriff auf Cloud-Run-Services,
- abgesicherte Datenbank- und Storage-Zugriffe,
- Trennung von Frontend, Backend und Infrastruktur,
- Nachvollziehbarkeit von Zugriffen über Audit-Logs.

3.2 Aufruf des IAM-Bereichs

Die Konfiguration der Zugriffsrechte erfolgt über den Bereich *IAM und Verwaltung* innerhalb der Google Cloud Console. Nach Auswahl des Projekts wird der Menüpunkt *IAM* geöffnet.

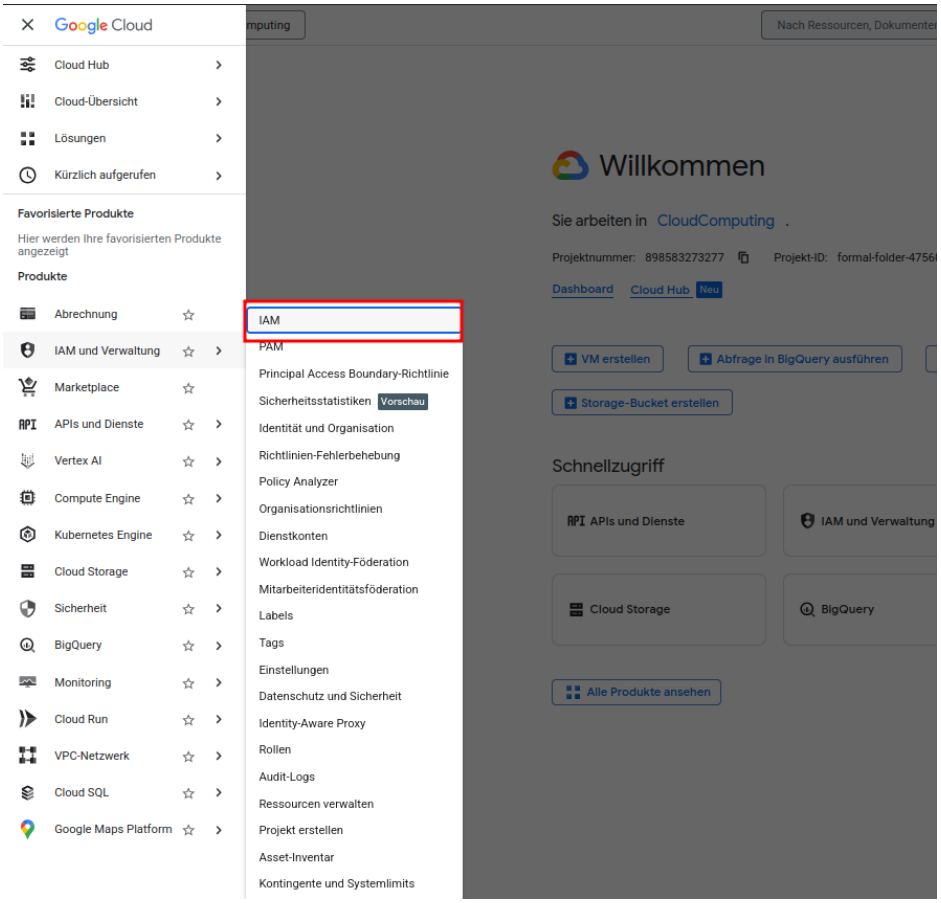

Abbildung 9: Navigation zum IAM-Bereich in der Google Cloud Console

3.3 Vergabe von Zugriffsrechten

Innerhalb der IAM-Übersicht können Hauptkonten, Gruppen und Dienstkonten mit spezifischen Rollen ausgestattet werden. Die Vergabe neuer Berechtigungen erfolgt über die Funktion *Zugriffsrechte erteilen*.

Dabei wird zunächst ein Hauptkonto oder Dienstkonto ausgewählt und anschließend eine oder mehrere Rollen zugewiesen, die definieren, welche Aktionen auf Projektebene erlaubt sind.

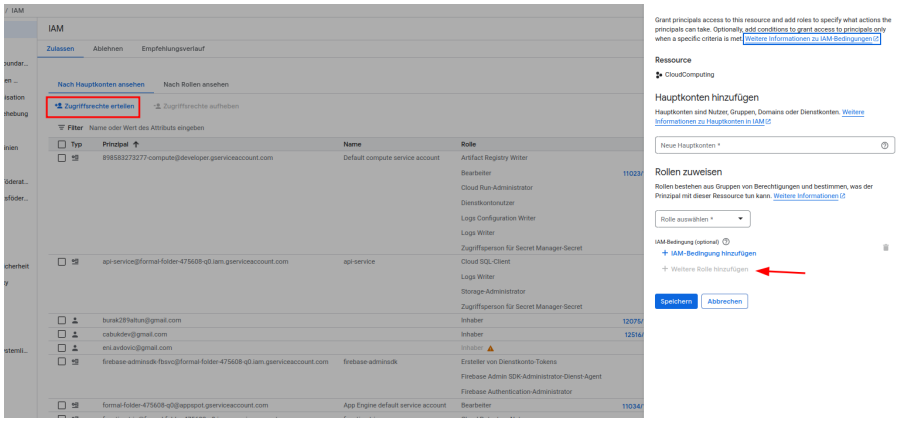

Abbildung 10: Erteilen von Zugriffsrechten über IAM

3.4 Service Accounts für Backend-Services

Für die Backend-Services wurde ein dediziertes Dienstkonto (*api-service*) verwendet. Dieses Dienstkonto wird von den Cloud-Run-Services genutzt, um kontrolliert auf weitere Cloud-Ressourcen zugreifen zu können.

Dem Dienstkonto wurden gezielt Rollen zugewiesen, unter anderem:

- Zugriff auf Cloud Storage,
- Zugriff auf Cloud SQL,
- Schreibrechte für Logs,
- Zugriff auf Secrets im Secret Manager.

Durch diese Trennung wird vermieden, dass Cloud-Run-Services mit übergeordneten oder globalen Rechten ausgeführt werden.

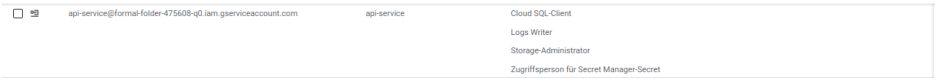

Abbildung 11: Zugewiesene Rollen für das Service Account des API-Services

3.5 Einbindung des Service Accounts in Cloud Run

Das konfigurierte Dienstkonto wird direkt dem Cloud-Run-Service zugewiesen. Dies erfolgt bei der Bereitstellung oder Aktualisierung eines Services im Reiter *Sicherheit*.

Durch diese Zuweisung werden alle Zugriffe des Cloud-Run-Services automatisch im Kontext des gewählten Dienstkontos ausgeführt.

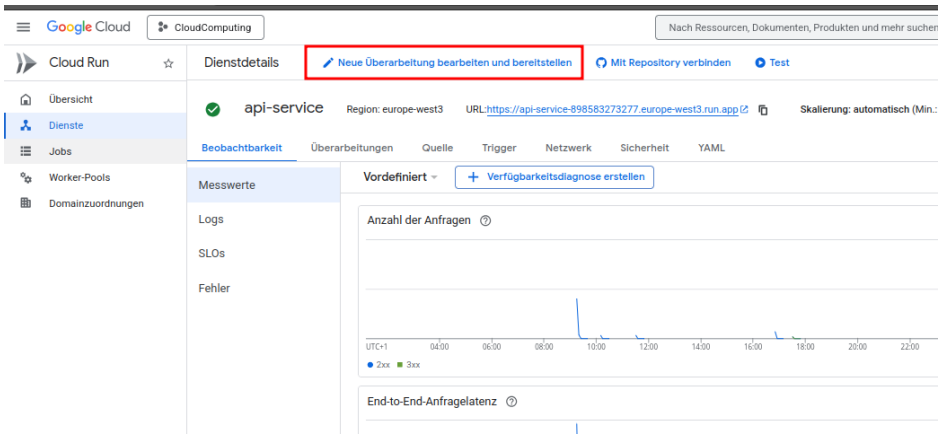

Abbildung 12: Zuweisung des Service Accounts zu einem Cloud-Run-Service

3.6 Revisionen und sichere Aktualisierung

Jede Änderung an der Konfiguration eines Cloud-Run-Services führt zur Erstellung einer neuen Revision. Diese Revision enthält neben dem Container-Image auch die Sicherheits- und IAM-Einstellungen.

Durch dieses Revisionsmodell bleibt jederzeit nachvollziehbar, welche Konfiguration aktiv ist und welche Berechtigungen verwendet werden.

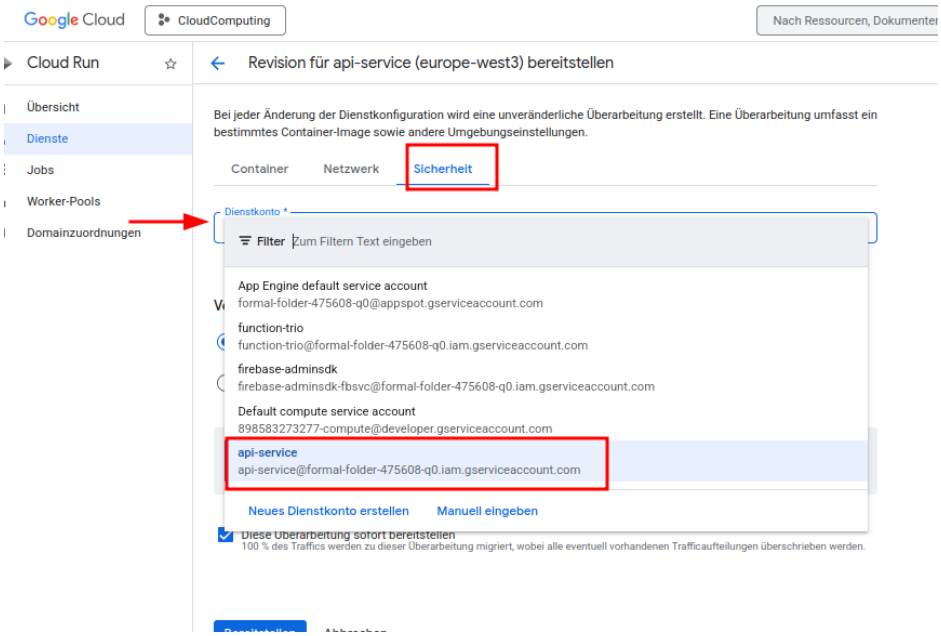

Abbildung 13: Bereitstellung einer neuen Revision mit aktualisierten Sicherheitsparametern

3.7 Sicherheitsbewertung

Durch den Einsatz von IAM in Kombination mit dedizierten Service Accounts wird das Prinzip der minimalen Rechtevergabe (Least Privilege) umgesetzt. Frontend-Komponenten besitzen keinerlei direkte Zugriffsrechte auf Cloud-Ressourcen.

Alle sicherheitskritischen Operationen werden ausschließlich durch Backend-Services durchgeführt, die klar definierte Rollen und Berechtigungen besitzen.