Untersuchung der AWS Mobile Hub

Michael Zipperle
259564
Hochschule Furtwangen
Michael.Zipperle@hs-furtwangen.de

***Abstract*—Mobile Anwendungen benötigen meistens ein Backend um beispielsweise die Nutzer der Anwendung zu verwalten oder Daten in einer Datenbank zu speichern. Die Konfiguration der traditionellen Backends benötigt i.d.R. ein Spezialisten und wenn sich die Anzahl der Nutzer drastisch ändert, können die zur Verfügung stehenden Ressourcen nicht mehr ausreichen. Ein Mobile Backend as a Service (MBaaS) soll dies ändern, dabei kann der Entwickler einfach Cloud Services wie Datenbank, Nutzermanagement und mehr in einfachen Schritten konfigurieren und muss sich nicht mehr um die Ressourcen kümmern. Dieser Artikel zeigt auf, welche MBaaS Anbieter aktuell auf dem Mark existieren und behandelt dabei Amazon Web Services (AWS) Mobile Hub genauer. Es werden die zur Verfügung stehenden Cloud Services beschrieben und anhand einer Android App wird gezeigt, wie AWS Mobile Hub eingesetzt und die Entwicklung vereinfacht werden kann.**

I. Einführung

Immer mehr Cloud Service Anbieter bieten ein MBaaS an. MBaaS bietet Entwicklern die Möglichkeit, ihre mobilen Anwendungen mit einem Cloud Backend zu verknüpfen. Dieses Backend erlaubt dem Entwickler, die einfache Konfiguration und das Hinzufügen von Cloud Services zu mobilen Anwendungen. Somit können in wenigen Schritten Cloud Services wie Authentifizierung, Datenspeicher, Benachrichtigung, Analystics und vieles mehr hinzugefügt werden. Die Services können über das vom Anbieter bereitgestellte Software Development Kit (SDK) in die mobilen Anwendungen integriert werden. Ein MBaaS wird von Entwicklern immer mehr verwendet, da der Einsatz Zeit spart für die Konfiguration und Wartung der IT-Infrastruktur einer mobilen Anwendung. Dazu kommen weitere Vorteile wie eine fast ein hundert prozentige Ausfallsicherheit und eine automatische Skalierung der Ressourcen. Aktuell gibt es mehrere Anbieter für ein MBaaS.

II. Verwandte Arbeit

Bevor auf AWS Mobile Hub genauer eingegangen wird, werden in diesem Kapitel die Anbieter für ein MBaaS kurz erläutert. Welcher daraus sich am besten für eine mobile Anwendung eignet, hängt einerseits von den Anforderungen der mobilen Anwendung und anderseits von den bisherigen Erfahrungen und Kenntnissen des Entwickler-Teams ab.

A. Apple Cloud Kit

Das Cloud Kit von Apple wurde 2015 als Teil von iOS 8 veröffentlicht und wird Entwicklern kostenlos zur Verfügung gestellt [10]. Dies soll die Entwicklung von Anwendungen für Apple's iOS, macOS, web, watchOS and Apple TV vereinfachen und vorantreiben. Das MBaaS unterstützt ausschließlich Plattformen von Apple und fokussiert sich unter anderem auf die Authentifizierung, Analyse und der Speicherung von Daten [11].

B. Progress Kinvey

Progress bietet mit Kinvey ein MBaaS, das mit allen gängigen mobilen Betriebssystemen und Technologien eingesetzt werden kann. Kinvey bietet ein Backend für Authentifizierung, Sicherheit und Compliance, Speicherung von Daten und mehr. Ebenso wirbt es mit einer Verkürzung der Entwicklungszeit um bis zu 75%. Für einen einzelnen Entwickler ist die Nutzung vorerst kostenlos, kommen mehrere Entwickler dazu und werden mehr Cloud Ressourcen benötigt, können bis zu €2000 pro Monat anfallen [12].

C. Google Firebase

Google veröffentlicht Firebase 2016 und ist aktuell einer der am meisten genutzten Anbieter. Dies liegt einerseits an der großen Anzahl an Services, die sich von Authentifizierung, Speicherung von Daten, Hosting bis hin zu maschinelles Lernen, Cloud Funktionen und einer Test Lab erstrecken, anderseits an der Unterstützung aller gängigen Plattformen sowie der großen Gemeinschaft von Entwicklern [10]. Auch Firebase bietet ein kostenloses Paket an, dieses ist jedoch nicht für eine Anzahl an Entwicklern beschränkt, sondern anhand des Bedarfs an Cloud Ressourcen. Braucht ein Entwickler-Team mehr Ressourcen, so wird nach dem gängigen "Pay As You Go" Prinzip abgerechnet [13].

D. Kumulos

Kumulos veröffentlichte bereits 2010 ihr MBaaS und richtet sich hauptsächlich an Freelancer und Agenturen. Es eignet sich unter anderem für Entwickler, die mehrere mobile Anwendungen verwalten und dafür eine zuverlässige Daten Plattform benötigen [15]. Kumulos hebt sich mit Features wie Crash Reporting, Automatisches Reporting & Analytics, App Store Optimierung, Agency Console und Push Benachrichtigungen hervor. Ebenso wirbt es mit der einfachen Konfiguration und dem günstigen Preiskonzept und unterstützt alle gängigen mobilen Plattformen und Technologien. Aktuell wird es von mehreren Tausend Entwicklern in mehr als 25 Ländern genutzt [14].

E. Parse

Parse ist ein OpenSource Projekt und stellt ein MBaaS bereit. Es umfasst Services wie Authentifizierung, Speicherung von Daten, Analystics und Push Benachrichtigungen. Alle gängigen mobilen Plattformen und Technologien werden unterstützt. Parse hat den großen Vorteil, dass der Entwickler selbst über die Konfiguration und den Standort der IT-Infrastruktur entscheiden kann und ist zudem kostenlos [17]. Doch es gibt auch Anbieter wie back4app, die Parse auf ihrer IT-Infrastruktur bereitstellen und im Rahmen eines Abos Entwicklern zur Verfügung stellen [16].

III. AWS Mobile Hub

Die meisten Menschen denken bei Amazon nur an einen Online Shop, bei dem zahlreiche Produkte gekauft werden können. Doch mit AWS stellt Amazon eine Cloud-Infrastruktur und Cloud Services bereit, welche aktuell zu einer Haupteinnahmequelle von Amazon zählen. AWS ist sehr erfolgreich, somit zählt Amazon aktuell zu den Marktführern der Cloud Anbieter [18]. Mit AWS Mobile Hub stellt Amazon ein MBaaS zur Verfügung, dass wir im folgenden genauer anschauen.

Mit AWS Mobile Hub lassen sich einfach und effizient Cloud Services zu einer mobilen Anwendungen hinzufügen. Im folgenden werden die unterstützten Cloud-Services kurz erläutert.

A. Benachrichtigungen und Nutzungsanylyse

Mit Amazon Pinpoint bietet AWS ein Cloud Service, um die Nutzer einer Anwendung via Push-Benachrichtigungen, E-Mail oder SMS mit Informationen zu versorgen. Eine Anwendung veröffentlicht neue Push-Benachrichtigungen an ein Thema und alle Abonnenten dieses Themas werden benachrichtigt. Dabei werden nicht die Endgeräte direkt benachrichtigt, da jedes mobile Betriebssystem seinen eigenen Service hat, um die mobilen Endgeräte zu benachrichtigen. Somit wird der Service des jeweiligen Betriebssystems benachrichtigt und dieser leitet diese Benachrichtigung weiter. Des Weiteren können durch Amazon Pinpoint Nutzungsdaten erfasst und analysiert werden. Dies ermöglicht den Entwicklern das Verhalten der Nutzer besser zu verstehen. Es kann unter anderem analysiert werden, wie ein Nutzer auf eine Benachrichtigung reagiert oder wie viele Nutzer aktuell eingeloggt sind [4].

B. User Sign-in

Mit Amazon Cognito lassen sich Registrierung, Anmeldung und Zugriffskontrolle eines Nutzers einfach verwalten. Es können sichere und skalierbare Benutzerverzeichnisse angelegt und soziale Identitätsanbieter wie Facebook und Google oder Unternehmens-Identitätsanbieter wie Microsoft Active Directory zur Anmeldung verwendet werden. Zudem beinhaltet es ein Sicherheitskonzept, das die Multifaktor-Authentifizierung sowie die Verschlüsselung der Daten im Speicher und auf dem Übertragungsweg unterstützt. Ebenso lassen sich mit Amazon Cognito die Zugriffsrechte für andere Cloud-Services im Rahmen der Anwendung festlegen [5].

C. Cloud Logik

Amazon Lambda bietet eine Plattform in der Cloud auf der Programme ausgeführt werden können, ohne dass der Nutzer den Server bereitstellen und verwalten muss. Dabei lädt der Entwickler lediglich den Programmcode hoch und Lambda übernimmt den Rest. Amazon Lambda skaliert automatisch die Anwendung je nach Verarbeitungslast und der Anzahl an Zugriffen. Zusätzlich lässt sich Amazon Lambda mit anderen Amazon Services verknüpfen, welche durch ein Event eine Anwendung von Amazon Lambda starten können. Daher ist Amazon Lambda perfekt um Logik einer mobilen Anwendung in die Cloud zu verlangen und somit Ressourcen auf dem mobilen Endgeräte einzusparen. Der Programmcode kann in den Programmiersprachen Javascript, Python, Java und C# entwickelt werden [1]. Des Weiteren lässt sich mit Hilfe von Amazon Application Programming Interface (API) Gateway eine Representational State Transfer (REST) API bereitstellen, mit der auf Lambda Funktionen zugegriffen werden kann. Dabei stellt Amazon API Gateway die Verwaltung des Datenverkehrs, Autorisierung und Zugriffskontrolle, Überwachung und Verwaltung der API-Version bereit [6].

D. NoSQL Database

Amazon DynamoDB ist eine NoSQL Datenbank, die für Anwendungen eine konsistente Antwortzeit im einstelligen Millisekundenbereich anbietet. Zusätzlich passt DynamoDB die Kapazität automatisch anhand der Anwendungsanfragen an. Des Weiteren muss der Nutzer keine Datenbankverwaltungsaufgaben vornehmen. DynamoDB stellt automatisch die Hardware und Software bereit und kümmert sich um den Betrieb eines zuverlässigen, verteilten Datenbank-Clusters oder die Skalierung der Daten über mehrere Instanzen. Dabei unterstützt DynamoDB sowohl das Dokumente- als auch Schlüssel-Wert-Speichermodell. DynamoDB kann mit Amazon Lambda verknüpft werden, sodass bei einer Datenänderung (Hinzufügen, Ändern oder Löschen eines Datensatzes) automatisch eine Lambda Anwendung ausgeführt werden kann. Die Lambda Anwendungen kann dann beispielsweise Nutzer über geänderte Daten in der Datenbank informieren [2].

E. Daten Speicher

Amazon S3 ist ein Objektspeicher, der das Speichern und Abrufen von beliebigen Daten für beliebige Anwendungen ermöglicht. Der Objektspeicher kann für Unternehmensanwendungen, mobile Anwendungen (Apps und Webseiten) sowie der Speicherung der Daten von IoT Geräten (Sensoren) eingesetzt werden. Dabei wird eine Verfügbarkeit der Daten von nahe zu 100% und eine automatische Skalierung der Datenmenge gewährleistet. Zudem wird ein umfassendes Sicherheitskonzept eingesetzt, um den neuesten Richtlinien für Sicherheit stand zu halten. Das BigData Konzept wird eingesetzt, um schnellst möglich einen bestimmten Datensatz im Speicher zu finden [7].

F. Konversations-Bot

Amazon Lex ist ein Konversations-Bot der ermöglicht, einen Chat in einer Anwendung zu erzeugen. Dieser Bot basiert auf einer fortgeschrittenen Deep-Learning-Funktion und erlaubt die Umwandlung von Sprache zu Text, Text zu Sprache und das Sprachverständnis. Dies erlaubt die Erzeugung eines realistischen Gesprächs für eine Anwendung und somit ein gutes Benutzererlebnis. Der Einsatz von Bots bietet zahlreiche Einsatzmöglichkeiten um eine Anwendung zu verbessern. Zum Beispiel werden Bots eingesetzt, um Nutzer auf einer Webseite Hilfe zu gewährleisten, falls diese Fragen haben. Ein anderes Beispiel sind Hotlines von großen Unternehmen. Um Mitarbeiterressourcen zu sparen, werden Bots eingesetzt, um die ersten Informationen des Anrufers auszuwerten und um einen passenden Mitarbeiter bereit zu stellen. Erwähnenswert ist auch, dass Amazon Lex die Grundlage für den Sprachassistenten Alexa ist [8].

G. Hosting und Streaming

Amazon CloudFront stellt ein Content Delivery Network (CDN) bereit und erlaubt Daten, Videos, Anwendungen und API's weltweit mit wenig Wartezeit, hoher Sicherheit und hohen Übertragungsraten bereitzustellen [9].

H. Amazon CloudWatch

Amazon CloudWatch ist ein Service, der die anderen Cloud Services von AWS überwacht. CloudWatch kann verschiedene Protokolldateien und Metriken von Cloud Services sammeln und überwachen. Des Weiteren kann der Administrator Alarme festlegen, um zeitnah auf evtl. vorliegende Probleme eines Service zu reagieren. Es können Events definiert werden, die bestimmen, wie das Problem eines ausgelösten Signals automatisch behoben werden kann. Zusätzlich bietet CloudWatch einen Einblick in die Auslastung der Ressourcen, die Anwendungsleistung und die Integrität ihrer Betriebsabläufe [3].

IV. Tutorial: Cow Tracker

Nachdem im letzten Kapitel die Services von AWS Mobile Hub erläutert wurden, geht es jetzt darum, diese Services zu nutzen und eine kleine Android App zu entwickeln, die beispielhaft deren Einsatz aufzeigt. Folgende Anforderungen gilt es mit der App umzusetzen:

- Login via E-Mail/Passwort und Facebook
- Anzeige der Kühe eines Nutzers auf einer Karte
- Anzeige der aktuellen Position des Nutzers auf der Karte
- Zu jeder Kuh soll die Entfernung vom Nutzer angezeigt werden

Die Entwicklung der Android App erlaubt erste Erfahrungen mit AWS Mobile Hub zu sammeln. Nachfolgend werden die einzelnen Schritte beschrieben.

A. Erstellung eines Android Projekts mit Android Studio

Der Quellcode für die Android App lässt sich von GitHub "https://github.com/MichZipp/Exploring-AWS-Mobile-Hub.git" herunterladen. Nun öffnet man das heruntergeladene Android Projekt in Android Studio und konfigurieren das Backend.

B. Erstellung eines AWS Mobile Hub Projekts

Um ein neues Projekt anzulegen, logen wir uns bei der "AWS Console" ein und wählen den Service "Mobile Hub" aus. Dann kann dem Wizard gefolgt werden, bei dem wir den Projektnamen und die Projektplattform, in unserem Fall Android, auswählen. Standardmäßig ist Amazon Pipeline zur Analyse der Nutzerdaten aktiviert und unser Dashboard sollte wie auf Abbildung 1 aussehen. Als nächstes können wir Cloud Services hinzufügen, die wir zur Erfüllung unserer Anforderung benötigen. Jedes mal wenn wir im Dashboard neue Cloud Services hinzufügen oder Einstellungen verändern, müssen wir "Intergrate" klicken, um eine neue Konfigurationsdatei zu erhalten. Diese Konfigurationsdatei fügen wir bei unserem Android Projekt unter "res/raw" ein. Wenn wir in unserer Android App auf die Cloud Services zugreifen, holt sich die SDK automatisch die nötigen Information über den Service aus dieser Konfigurationsdatei.

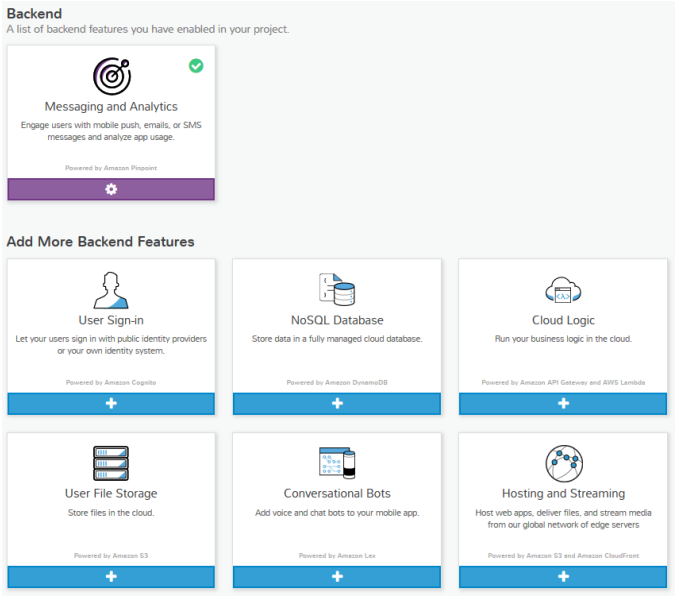


Fig. 1: Dashboard von AWS Mobile Hub

1) Login: Ein Nutzer soll sich entweder mit E-Mail und Passwort oder Facebook authentifizieren können.

- E-Mail und Passwort: Dazu klicken wir in unserem Dashboard auf "User Sign-in" und wählen "E-Mail und Passwort" aus. Wir setzen das Häkchen zur Erzeugung eines neuen Benutzerverzeichnisses und legen anschließend fest, welche Anforderungen ein Passwort haben soll.

Es kann festgelegt werden, wie viele und welche Zeichen das Passwort beinhalten muss. Um den Vorgang abzuschließen, klicken wir einfach ”Create user pool”.

- Facebook: Um Facebook als Authentifizierungsmethode zu verwenden, müssen wir uns zuerst bei der ”Facebook Developer Console” anmelden und ein neues Projekt erstellen. Nach der Erzeugung des Projekts, finden wir unter ”Einstellungen - Allgemeines” eine App-ID, welche wir kopieren. Nun gehen wir zurück zum AWS Mobile Hub Dashboard, klicken auf ”User Sign-in” und wählen Facebook aus. Anschließend werden wir aufgefordert, die kopierte App-ID einzugeben. Ist dies erledigt, klicken wir auf ”Enable Facebook Login”.

Als nächstes wollen wir uns den Login in unserer Android Projekt anschauen, dazu öffnen wir die Klasse ”AuthenticatorActivity” und betrachten die Methode ”showSignIn()”. Wir können sehen, dass sich mit nur wenigen Zeilen Code ein Login Fenster erstellen lässt, dass mit unserem Backend kommuniziert.

2) NoSQL Datenbank: Nun wollen wir eine Datenbank hinzufügen, in der die GPS-Daten der Kühe eines Nutzers gespeichert sind. Dazu klicken wir im Dashboard auf ”NoSQL Database” und erstellen eine neue Datenbank mit dem Namen ”Locations”. Die Struktur der Datenbank wollen wir selbst festlegen und wählen deshalb ”Costum Schema” aus. Anschließend legen wir die Datenbank Struktur wie in Abbildung 2 fest und klicken anschließend ”Create Table”, um die Datenbank zu erstellen. Über ”Download Models” kann eine Java Klasse heruntergeladen werden, die wir in unser Android Projekt einbinden. Somit können abgefragte Datensätze zu einem Java Objekt geparsed werden und wir können einfach Attribute abfragen, setzen oder ändern.

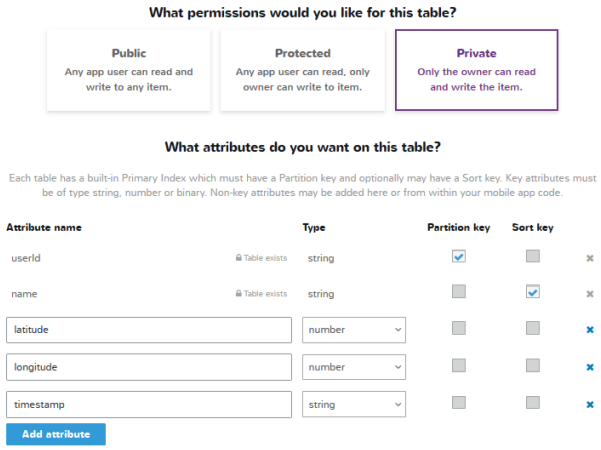


Fig. 2: Datenbank Konfiguration

Schauen wir uns noch den Zugriff auf die Datenbank in der Android App an: Dazu öffnen wir die Klasse ”DBHandler”. Durch die Methode ”createLocation()” kann eine neue Position einer Kuh in der Datenbank abgespeichert werden. Mit der Methode ”queryLocations()” können die Positionen aller Kühe des aktuellen Nutzers abgefragt werden. Damit wir nicht manuell neue Datensätze in die Datenbank eingeben müssen, gibt es die Methode ”generateTestLocations()”, die fünf Test Datensätze in die Datenbank einfügt. In der ”MapsActivity” der Android App wird der ”DBHandler” genutzt, um die Position der Kühe abzufragen. Diese Positionen werden dann auf einer Google Map mit einem Marker markiert. Zusätzlich wird noch via GPS der Standort des Nutzer ermittelt und ebenso mit einem Marker auf der Karte angezeigt.

3) Cloud Logik: Wenn wir unsere Android App starten, haben wir bereits ein Login Fenster bei dem wir uns anmelden können und nach dem Login wird uns unsere Position und die der Kühe angezeigt. Nun wäre es für uns noch wichtig zu wissen, wie weit eine Kuh von uns entfernt ist. Wir gehen jetzt davon aus, dass die Berechnung der Entfernung sehr viel Ressourcen in Anspruch nimmt und wollen deshalb die Berechnung in der Cloud durchführen. Dazu klicken wir im Dashboard auf ”Cloud Logic” und anschließend auf ”Create a new API”. Wir füllen das Formular wie in Abbildung 3 aus und erstellen die API mit ”Create API”.

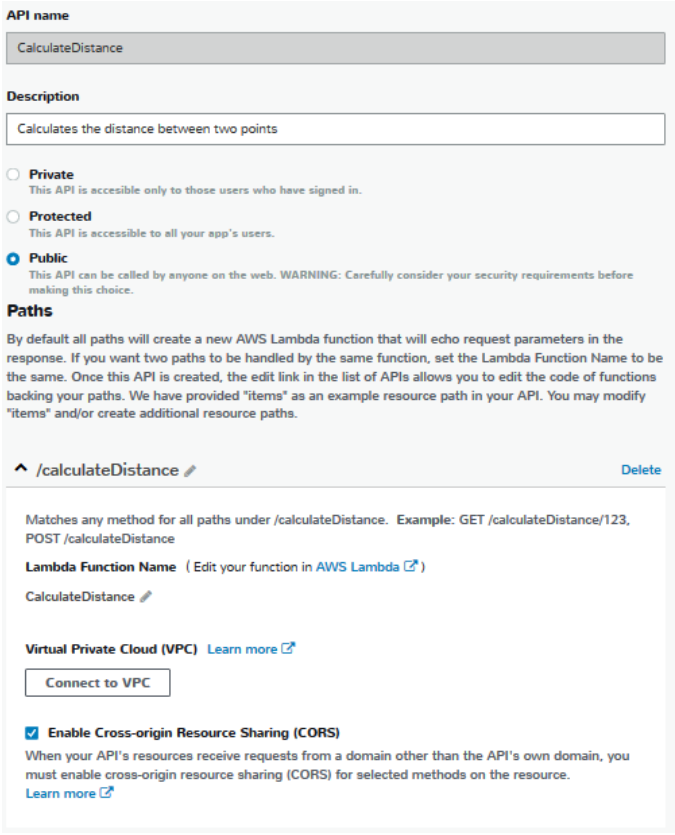


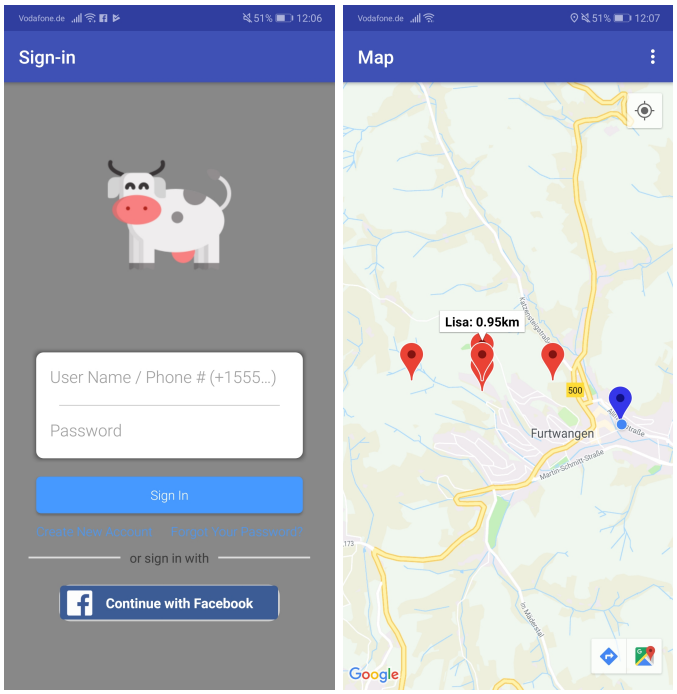
Fig. 3: Cloud Logik Konfiguration

Es wurde automatisch eine Lambda Funktion generiert. Um diesen Vorgang zu prüfen, wählen wir den Service ”Lambda Function” in unserer AWS Console aus. Wenn alles funktioniert, sollten wir eine Lambda Funktion mit dem Namen ”CalculateDistance” sehen. Nun können wir unseren Quellcode für die Berechnung der Entfernung hinzufügen.

Standardmäßig kann im Browser mit JavaScript entwickelt werden, wir bevorzugen aber Java und nutzen dafür die Integrated Development Environment (IDE) Elipse mit dem AWS Plugin, um den Code später in die Cloud hochladen zu können. Als erstes nutzen wir das AWS Plugin um eine neues Java Lambda Projekt zu erstellen. Anschließend fügen wir den Quellcode aus dem Ordner ”CowTrackerLambda” unseres heruntergeladenen Git Repository ein. Nun müssen wir den Quellcode nur noch hochladen. Dazu machen wir ein Rechtsklick auf das Projekt und klicken auf ”Amazon Web Services - Upload function to lambda...”. Es öffnet sich ein Wizard, bei dem wir die Lambda Funktion auswählen müssen, dementsprechend wählen wir ”CalculateDistance” aus und klicken auf ”Upload”. Damit haben wir Cloud Logik zu unserer Android App hinzugefügt. Die Cloud Logik wird in der Klasse ”DistanceCalculator” der Android App über ein HTTP Anfrage abgerufen.

C. Ergebnis

Nun haben wir alle nötigen Schritte beendet, um unsere Anforderung umzusetzen. Starten wir die App, können wir uns einloggen und die App zeigt eine Karte mit unserer Position und der Position der Kühe. Klicken wir auf einen Marker, so zeigt es uns den Kuhnamen und unsere aktuelle Entfernung zu dieser Kuh an. Abbildung 4 zeigt zwei Screenshots, wie das Login-Fenster und die Kartenansicht aussieht.



(a) (b)

Fig. 4: Screenshots der Cow Tracker Android App

Als nächsten wollen wir noch einen kleinen Einblick in die Nutzungsstatistik werfen. Dazu wählen wir in der AWS Console den Service Pinpoint und anschließend unser Projekt aus. Nun werden uns nützliche Statistiken zu unserer Android App angezeigt. Abbildung 5 zeigt uns einen kleinen Ausschnitt davon. Es wird uns beispielsweise angezeigt, wie viel Nutzer die App starten oder sich pro Tag einloggen.

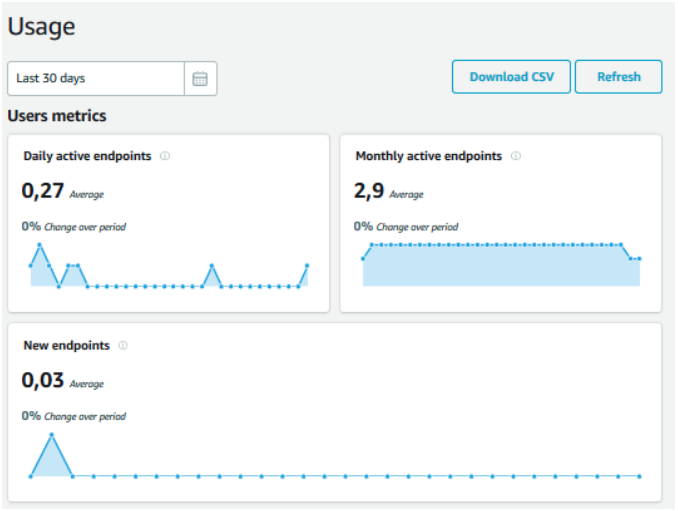


Fig. 5: Pinpoint Nutzungsanalyse

V. Erfahrung

Obwohl ich bereits mit AWS Services gearbeitet habe, fiel mir am Anfang die Konfiguration der AWS Mobile Hub schwer. Amazon stellt eine Dokumentation und Tutorials zur Entwicklung von mobilen Anwendungen mit AWS Mobile Hub als MBaaS bereit. Jedoch unterschieden sich die verwendeten SDK Versionen in der Dokumentationen mit der der Tutorials. Einfache Beispielanwendungen ließen sich einfach nach den Tutorials erstellen, sollten diese erweitert werden mit Funktion anhand der Dokumentation, musste genau drauf geachtet werden, welche SDK Version genutzt wurde. Hat man mit AWS Mobile Hub eine gewisse Zeit gearbeitet und Erfahrung gesammelt, lassen sich sehr einfach Anwendungen entwickeln mit AWS als MBaaS. Ich war fasziniert, wie einfach sich AWS Cloud Services in eine Anwendungen integrieren ließen. Mit nur wenigen Zeilen Code kann eine Nutzerstatistik oder ein Login zu einer Anwendungen hinzugefügt werden. Ich werde AWS Mobile Hub in der Zukunft in den Augen behalten und falls Bedarf, für meine nächsten Anwendungen einsetzen.

VI. Zusammenfassung

Dieser Artikel beschreibt wie durch den Einsatz eines MBaaS die Entwicklung mobiler Anwendungen vereinfacht wird. Aktuell sind die beliebtesten MBaaS Anbieter Apple Cloud Kit, Progress Kinvey, Google Firebase, Kumulos, Parse und AWS Mobile Hub. Es wird kurz der Funktionsumfang jedes Anbieters beschrieben. Welche Anbieter sich am besten eignet, hängt von den Anforderungen der Anwendung sowie

der bisherigen Erfahrung der Entwickler ab. Im Detail wird dann AWS Mobile Hub analysiert und die zu verknüpfenden Cloud Services beschrieben. Des Weiteren wird eine Android App als Beispielanwendung entwickelt. Die App zeigt, wie AWS Mobile Hub eingesetzt werden kann. Dabei wird der App Quellcode zur Verfügung gestellt und erklärt, wie sich das Backend und die Cloud Services konfigurieren lassen. Mit der App wird aufgezeigt, wie ein MBaaS die Entwicklung einer Anwendung vereinfacht und sich die Entwicklungskosten senken lassen.

REFERENCES

[1] ”Amazon Lambda” https://aws.amazon.com/de/lambda/ Accessed 18.06.2018

[2] ”Amazon DynamoDB” https://aws.amazon.com/de/dynamodb/ Accessed 18.06.2018

[3] ”Amazon Cloud Watch” https://aws.amazon.com/de/cloudwatch/ Accessed 18.06.2018

[4] ”Amazon Pinpoint” https://aws.amazon.com/de/pinpoint/ Accessed 18.06.2018

[5] ”Amazon Cognito” https://aws.amazon.com/de/cognito/ Accessed 19.06.2018

[6] ”Amazon API Gateway” https://aws.amazon.com/de/api-gateway/ Accessed 20.06.2018

[7] ”Amazon E3” https://aws.amazon.com/de/s3 Accessed 20.06.2018

[8] ”Amazon Lex” https://aws.amazon.com/de/lex Accessed 20.06.2018

[9] ”Amazon Cloud Front” https://aws.amazon.com/de/cloudfront/ Accessed 20.06.2018

[10] ”TechBeacon - How choose right MBaas” https://techbeacon.com/how-choose-right-mbaas-google-firebase-apple-icloud-or-kinvey Accessed 18.06.2018

[11] ”Apple Cloud Kit” https://developer.apple.com/documentation/cloudkit Accessed 18.06.2018

[12] ”Progress Kinvey” https://www.progress.com/kinvey/ Accessed 18.06.2018

[13] ”Google Firebase” https://firebase.google.com/ Accessed 18.06.2018

[14] ”Kumulos” https://www.kumulos.com/ Accessed 19.06.2018

[15] ”Karnival - 5 best MBaaS alternatives to parse” http://carnival.io/mobile-insights/5-best-mbaas-alternatives-to-parse-2/ Accessed 19.06.2018

[16] ”back4app” https://www.back4app.com/ Accessed 19.06.2018

[17] ”Parse” https://www.back4app.com/ Accessed 19.06.2018

[18] ”Statista - Cloud Market Share” https://www.statista.com/chart/7994/cloud-market-share/