

Cloud Solutions for Mobile Applications

Anssi Soinu

Master's thesis
May 2014
Information Technology

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in Information Technology

Anssi Soinu:
Cloud Solutions for Mobile Applications

Thesis supervisor Mikkolainen Jari

Master's thesis 122 pages, appendices 10 pages
May 2014

This master's thesis starts with an introduction to how cloud solutions and mobile applications can be connected to each other. After the introduction, the ways in which different cloud solutions will be studied and compared are defined. After the definition of study methods, the selection of cloud solutions is made, followed by an actual study of the selected cloud solutions. Next, based on the results of this study, a comparison of the selected cloud solutions is made. This master's thesis ends with a conclusion, which takes account of the study and the comparison of the selected cloud solutions.

In this master's thesis, Amazon Web Services, Google Cloud Platform and Microsoft Azure were selected as study and comparison targets. As a conclusion of the study and comparison, it is possible to say that the compared cloud solutions offer different kinds of approaches on how to support mobile applications. This master's thesis presents these different approaches; based on the given information, picking the most suitable cloud solution for a certain mobile application is facilitated.

Key words: cloud solutions, mobile solutions, MBaaS, BaaS, Amazon Web Services, Google Cloud Platform, Microsoft Azure

CONTENTS

1	INTRODUCTION.....	7
2	STUDY AND COMPARISON CRITERIA	8
3	STUDY TARGETS.....	9
4	STUDY.....	11
4.1	Amazon Web Services.....	11
4.1.1	Setting up MBaaS solution	14
4.1.2	Pull Information to Cloud Solution.....	14
4.1.3	Store Information in Cloud Solution.....	15
4.1.4	Process Information in Cloud Solution.....	16
4.1.5	Push Information to Mobile Application	17
4.1.6	Security	18
4.1.7	Mobile Platform Support	20
4.1.8	Pricing Model.....	20
4.2	Google Cloud Platform.....	22
4.2.1	Setting up MBaaS Solution.....	26
4.2.2	Pull Information to Cloud Solution.....	27
4.2.3	Store Information in Cloud Solution.....	28
4.2.4	Process Information in Cloud Solution.....	29
4.2.5	Push Information to Mobile Application	30
4.2.6	Security	31
4.2.7	Mobile Platform Support	33
4.2.8	Pricing Model.....	33
4.3	Microsoft Azure.....	36
4.3.1	Setting up MBaaS solution	39
4.3.2	Pull Information to Cloud Solution.....	40
4.3.3	Store Information in Cloud Solution.....	41
4.3.4	Process Information in Cloud Solution.....	42
4.3.5	Push Information to Mobile Application	44
4.3.6	Security	47
4.3.7	Mobile Platform Support	50
4.3.8	Pricing Model.....	51
4.4	IMPLEMENTING AN EXAMPLE CLOUD SOLUTION.....	56
4.4.1	Amazon Web Services	56
4.4.2	Google Cloud Platform	74
4.4.3	Microsoft Azure	81
5	COMPARISON.....	95

5.1	Setting up the MBaaS solution	95
5.2	Pull Information to Cloud Solution	95
5.3	Store Information in Cloud Solution	95
5.4	Process Information in Cloud Solution.....	96
5.5	Push Information to Mobile Application	97
5.6	Security	98
5.7	Mobile Platform Support	99
5.8	Pricing Model	99
6	CONCLUSION	101
	REFERENCES.....	107
	APPENDICES	113
	Appendix 1. An example code that uses Amazon DynamoDB table from Windows Phone mobile application.	113
	Appendix 2. An example code that uses Google Cloud Datastore from Android mobile application.....	117
	Appendix 3. An example code that uses Microsoft Azure mobile services SDK in Windows Phone mobile application.	120

GLOSSARY

Amazon EBS	Amazon Elastic Block Store
Amazon EC2	Amazon Elastic Compute Cloud
Amazon EMR	Amazon Elastic MapReduce
Amazon FPS	Amazon Flexible Payments Service
Amazon IAM	Amazon Identity and Access Management
Amazon MFA	Amazon Multi-Factor Authentication
Amazon RDS	Amazon Relational Database Service
Amazon S3	Amazon Simple Storage Service
Amazon SES	Amazon Simple Email Service
Amazon SNS	Amazon Simple Notification Service
Amazon SQS	Amazon Simple Queue Service
Amazon SWF	Amazon Simple Workflow
Amazon VPC	Amazon Virtual Private Cloud
AD	Active Directory
ADM	Amazon Device Messaging
AMI	Amazon Machine Image
API	Application Programming Interface
APNs	Apple Push Notification service
AWS	Amazon Web Services
BaaS	Back-end as a Service
Blob	Binary Large Object
CORS	Cross-Origin Resource Sharing
CRUD	Create Read Update Delete
DB	Database
GB	Gigabyte
GCM	Google Cloud Messaging
HSM	Hardware Security Module
IaaS	Infrastructure as a Service
MBaaS	Mobile Back-end as a Service
MPNS	Microsoft Push Notification Service
PaaS	Platform as a Service
PNS	Platform Notification System

REST	Representational State Transfer
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SLA	Service-level agreement
SQL	Structured Query Language
SSL	Secure Sockets Layer
URI	Uniform Resource Identifier
TAMK	Tampere University of Applied Sciences
WNS	Windows Push Notification Service
VAT	Value Added Taxes

1 INTRODUCTION

In FIGURE 1 is presented common use cases how a mobile application and a cloud solution can be connected to each other. In the FIGURE 1 is presented that:

- a mobile application can pull information to a cloud solution,
- a cloud solution can push information to a mobile application,
- a cloud solution can process information received from a mobile application,
- a cloud solution can store information received from a mobile application,
- a cloud solution can have security related functionalities to authenticate and authorize a mobile application,
- a mobile application can be built on top of different mobile platforms (for example on top of iOS or Android) and
- usage of a cloud solution might cost something (pricing model)

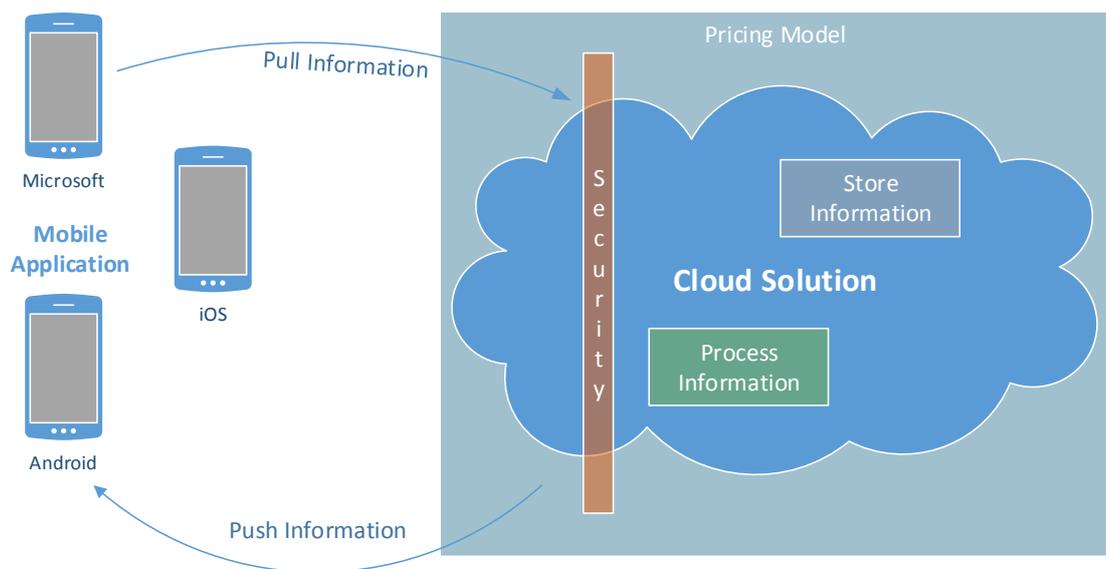


FIGURE 1. Common use cases how a mobile application and a cloud solution can be connected to each other

Pull information in the FIGURE 1 means that mobile application can send information to cloud solution. The information pull in the FIGURE 1 is triggered by the mobile application. Push information in the FIGURE 1 means that cloud solution can send information to mobile application. The information push in the FIGURE 1 is triggered by the cloud solution. This master's thesis is scoped to study and compare how cloud solutions support mobile applications based on the use cases mentioned in the FIGURE 1.

2 STUDY AND COMPARISON CRITERIA

This master's thesis studies and compares:

- how different cloud solutions makes possible to pull information from the mobile application to the cloud solution,
- how different cloud solutions stores information received from the mobile application,
- how different cloud solutions process information received from the mobile application,
- how different cloud solutions makes possible to push information from the cloud solution to the mobile application,
- how different cloud solutions handle security,
- how different cloud solutions support different mobile platforms and
- pricing models of different cloud solutions

Pull information in this master's thesis will be studied and compared based on how mobile application can send information to cloud solution. Pull information in this case means that the mobile application has triggered the information sending. Push information in this master's thesis is studied and compared based on how cloud solution can send information to mobile application. Push information in this case means that the cloud solution has triggered the information sending. Security of different cloud solution will be studied and compared in this master's thesis based on how authentication and authorization can be done. Authentication in this case means to verify that the mobile application or user of it is whom it claims to be. Authorization in this case means to verify that the mobile application can only access information in the cloud solution where it has access to.

3 STUDY TARGETS

It is possible to identify layers between a cloud solution and a mobile application as Michael Facemire (2012) from the Forrester Research has done in his mobile service triangle (see FIGURE 2). In the previously mentioned triangle, Michael Facemire has defined following layers: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Mobile Backend-as-a-Service (MBaaS) and Mobile Middleware. It is possible to compare all these layers to for example a computer: IaaS is the hardware, PaaS is the operating system (for example Windows 8), MBaaS hides/groups complex services of the operating system by having a simplified service on top of the operating system services and Mobile Middleware is a software which uses services of MBaaS.

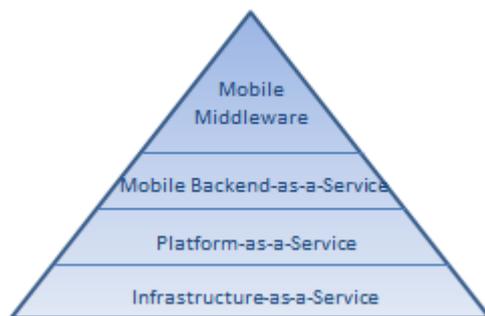


FIGURE 2. Michael Facemire's mobile service triangle

The Mobile Backend-as-a-Service layer (in FIGURE 2) is possible to see for example in the Internet referred also as a "Backend-as-a-Service" (BaaS). MBaaS or BaaS is something which has been introduced to for example accelerate development of mobile applications. Appcelerator and IDC have made a report about mobile trends (Michael King, John Jackson, 2013). The mobile trends report was made by surveying 6698 mobile developers. In the previously mentioned survey was a following question: "BaaS will overtake PaaS as the preferred cloud solution for mobile developers". Mobile developers answered to the previously mentioned question following:

- 14.8 % very likely,
- 59.4 % likely and
- 25.8 % not likely.

Based on the mobile trends report it is possible to see that 74.2 % of mobile developers that was surveyed expects BaaS to overtake PaaS as the preferred cloud solution for mobile developers.

Gartner and MarketsAndMarkets have made forecasts related to MBaaS. Gartner has forecasted in one of their press releases following: “By 2016, 40 Percent of Mobile Application Development Projects Will Leverage Cloud Mobile Back-End Services” (Meulen & Rivera 2013). MarketsAndMarkets has forecasted in one of their press releases BaaS market to grow between years 2012 and 2017 from 216.5 million dollar to 7.7 billion dollar (MarketsAndMarkets 2012). Based on forecasts from the Gartner and MarketsAndMarkets there exists at the moment a growing market for MBaaS.

Based on the reports and forecasts mentioned in this chapter, this master’s thesis will concentrate studying and comparing how Amazon Web Services, Google Cloud Platform and Microsoft Azure offers MBaaS cloud solutions. The selection of Amazon Web Services, Google Cloud Platform and Microsoft Azure has been done based on personal interests.

4 STUDY

4.1 Amazon Web Services

Amazon Web Services (AWS) is Amazon.com's cloud platform. AWS was released in year 2006 and it offers its products and services from following regions North Virginia, Oregon, North California, Ireland, Singapore, Tokyo, Sydney, São Paulo, GovCloud and Beijing (Amazon Web Services: Product and Services by Region).

From the AWS web site navigation it is possible to see that AWS categorizes its products to following categories: compute & networking, storage & CDN, database, analytics, application services, deployment & management and AWS Marketplace software. In the compute & networking category AWS provides following products: Amazon EC2, Auto Scaling, Elastic Load Balancing, Amazon WorkSpaces, Amazon VPC, Amazon Route 53 and AWS Direct Connect. In the storage & CDN category AWS provides following products: Amazon S3, Amazon Glacier, Amazon EBS, AWS Import/Export, AWS Storage Gateway and Amazon CloudFront. In the database category AWS provides following products: Amazon RDS, Amazon DynamoDB, Amazon ElastiCache and Amazon Redshift. In the analytics category AWS provides following products: Amazon EMR, Amazon Kinesis, AWS Data Pipeline and Amazon Redshift. In the application services category AWS provides following products: Amazon AppStream, Amazon CloudSearch, Amazon SWF, Amazon SQS, Amazon SES, Amazon SNS, Amazon FPS and Amazon Elastic Transcoder. In the deployment and management category AWS provides following products: AWS Management Console, AWS Identity and Access Management (IAM), AWS CloudTrail, Amazon CloudWatch, AWS Elastic Beanstalk, AWS CloudFormation, AWS OpsWork and AWS CloudHSM. In the Amazon Marketplace software category AWS provides different kind of partner software (for example developer tools).

From the AWS Solutions web page of the AWS web site it is possible to see that AWS claims that it can be used in building of following solutions: application hosting, backup and storage, content delivery, databases, e-commerce, enterprise IT, high performance computing, media hosting, on-demand workforce, search engines, web hosting, media, entertainment and life sciences (Amazon Web Services: AWS Solutions).

Sign up to Amazon Web Services is required in order to be able to use it. The sign up process contains five steps: account creation, payment method, identity verification, support plan and confirmation. In the account creation step it is required to input full name, country, address, city, state (or province or region), postal code (or ZIP) and phone number. In the account creation step it is also required to read and agree the terms of the Amazon Web Services Customer Agreement. In the payment method step it is required to input type of the credit card, credit card number, cardholder's name, expiration date of the credit card and billing address. After completing the credit card information and billing address in the payment method step, VAT registration number is asked if the billing address is located in a country that collects value added taxes (VAT). It is possible to define the VAT number also later in the AWS Account Activity page. In the identity verification steps, Amazon.com has an automated system that makes a call to given phone number. In the call it is required to input a PIN code which will be visible in the sign up web page while the call is in progress. After the correct PIN code is entered in the call, the sign up web page will automatically confirm that the identity verification is successfully completed. In the support plan step, Amazon.com will offer four plans: Basic, Developer, Business and Enterprise. From the FIGURE 3 it is possible to see details of different support plans. It is possible to see from the FIGURE 3 for example that the Basic support plan does not cost anything, Developer support plan will cost 49 dollars per month, Business support plan will cost at least 100 dollars per month and Enterprise support plan will cost at least 15 000 dollars per month. In the confirmation step, Amazon.com will send a sign up completed confirmation, a welcome and a getting started mail. From the sign up web page when in confirmation step, it is possible to for example launch the AWS Management Console, view developer resources or start exploring Amazon Web Services.



Select your AWS Support Plan

All customers receive free support. Choosing a paid support plan will allow you to receive one-on-one technical assistance from experienced engineers and access many other support features. [Click here to compare all Support plans.](#)

- Basic (Free)**
Contact Customer Service for account and billing questions, receive help for resources that don't pass system health checks, and access the AWS Community Forums.
- Developer (\$49/month)**
Get started on AWS – ask technical questions and get a response to your web case within 12 hours during local business hours.
- Business (Starting at \$100/month - Pricing example ) - Recommended**
24/7/365 real-time assistance by phone and chat, a 1 hour response to web cases, and help with 3rd party software. Access Trusted Advisor to increase performance, fault tolerance, security, and potentially save money. ([What's this !\[\]\(36e59bb2f5f69081e12d62e74933c4ce_img.jpg\)](#))
- Enterprise (Starting at \$15,000/month - Pricing example )**
15 minute response to web cases, an assigned technical account manager (TAM) who is an expert in your use case, and white-glove case handling that notifies your TAM and the service engineering team of a critical issue.

[Continue !\[\]\(39f5d6f07e075dacab8bb9790b2c07d2_img.jpg\)](#)

AWS Support Features

FIGURE 3. Support plan offers of AWS

Registered user of the Amazon Web Services is possible to access AWS Management Console (see FIGURE 4). In the AWS Management Console it is possible to access and manage Amazon Web Services. In the FIGURE 4 it is possible to see the console home of the AWS Management Console. From the console home it is possible for example to navigate to a certain AWS service. In the top of the AWS Management Console it is possible for example to see user account related actions, current region change action, navigation to different AWS services and navigation to help documentation.

From the Amazon Web Services web site it is possible to find “Mobile Applications on AWS” documentation (Amazon Web Services: Mobile Applications on AWS). By reading the “Mobile Applications on AWS” documentation and comparing content of it to Michael Facemire’s mobile service triangle (see FIGURE 2), it is not possible to find one service from the AWS which could be referred as a Mobile Backend-as-a-Service. From the “Mobile Application on AWS” documentation it is possible to find for example a service to store information in the AWS and a service to push information from the AWS to a mobile application. By combining different services of AWS it could be possible to create one service which could be called Mobile Backend-as-a-Service.

The screenshot displays the AWS Management Console interface. At the top, there is a navigation bar with the AWS logo, 'Services' dropdown, 'Edit' dropdown, and user information 'Anssi Soinu', 'Global', and 'Help'. The main content area is titled 'Amazon Web Services' and is organized into several columns of service cards, each with an icon and a brief description:

- Compute & Networking:** Direct Connect (Dedicated Network Connection to AWS), EC2 (Virtual Servers in the Cloud), Route 53 (Scalable Domain Name System), VPC (Isolated Cloud Resources).
- Database:** DynamoDB (Predictable and Scalable NoSQL Data Store), ElastiCache (In-Memory Cache), RDS (Managed Relational Database Service), Redshift (Managed Petabyte-Scale Data Warehouse Service).
- Analytics:** Data Pipeline (Orchestration for Data-Driven Workflows), Elastic MapReduce (Managed Hadoop Framework), Kinesis (Real-time Processing of Streaming Big Data).
- Storage & Content Delivery:** CloudFront (Global Content Delivery Network), Glacier (Archive Storage in the Cloud), S3 (Scalable Storage in the Cloud), Storage Gateway (Integrates On-Premises IT Environments with Cloud Storage).
- Deployment & Management:** CloudFormation (Templated AWS Resource Creation), CloudTrail (User Activity and Change Tracking), CloudWatch (Resource and Application Monitoring), Elastic Beanstalk (AWS Application Container), IAM (Secure AWS Access Control), OpsWorks (DevOps Application Management Service).
- App Services:** CloudSearch (Managed Search Service), Elastic Transcoder (Easy-to-use Scalable Media Transcoding), SES (Email Sending Service), SNS (Push Notification Service), SQS (Message Queue Service), SWF (Workflow Service for Coordinating Application Components).

On the right side, there are additional sections:

- Additional Resources:** Getting Started (See our documentation to get started and learn more about how to use our services.), Trusted Advisor (Best practice recommendations to save money, improve fault tolerance, increase performance, and close security gaps.).
- Service Health:** A green checkmark indicates 'All services operating normally.' It also shows the last update: 'Updated: Feb 11 2014 21:01:00 GMT+0200' and a link to the 'Service Health Dashboard'.
- Set Start Page:** A dropdown menu currently set to 'Console Home'.
- AWS Marketplace:** A promotional box with the text 'Find & buy software, launch with 1-Click and pay by the hour.'

At the bottom of the console, there is a footer with copyright information: '© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links for 'Privacy Policy' and 'Terms of Use'. A 'Feedback' button is also present in the bottom right corner.

FIGURE 4. AWS Management Console

4.1.1 Setting up MBaaS solution

In the Amazon Web Services it is not possible to setup a single service which would act as a Mobile Backend-as-a-Service. In the Amazon Web Services it is required to use several different services in order to create MBaaS solution.

4.1.2 Pull Information to Cloud Solution

In the Amazon Web Services it is possible to pull information from the mobile application to Amazon Web Service using AWS SDK. How to pull information from a mobile application to the Amazon Web Service varies based on where the information will be stored. In the Amazon Web Services it is required to use services of the selected storing solution

directly in order to be able to pull information from the mobile application to Amazon Web Services.

4.1.3 Store Information in Cloud Solution

In the Amazon Web Services it possible to use for example Amazon DynamoDB, Amazon RDS, Amazon SimpleDB or Amazon S3 to store information. It is also possible to setup own database on Amazon EC2 and Amazon EBS to make possible to store information in the Amazon Web Services.

Amazon DynamoDB is a non-relational database service. Amazon DynamoDB database consist of tables. A table in the Amazon DynamoDB is schema-less and can contain one or many items. An item in the Amazon DynamoDB can contain one or many attributes. A table in the Amazon DynamoDB is not limited to a certain data size. Amazon DynamoDB is not limited to a certain amount of tables. It is possible to manage Amazon DynamoDB using Amazon DynamoDB APIs or Amazon DynamoDB console. It is possible to for example create a new table to Amazon DynamoDB using Amazon DynamoDB APIs or Amazon DynamoDB console.

Amazon RDS is a relational database service. With the Amazon RDS it is possible to setup, scale and operate a relational database. Amazon RDS consist of following components: DB instances, regions, availability zones, security groups, DB parameter groups and DB option groups. DB instance is the database environment in the Amazon RDS. Every DB instance in the Amazon RDS has a DB engine. It is possible in the Amazon RDS to use products like MySQL, PostgreSQL, Oracle or Microsoft SQL Server as a DB engine. Region in the Amazon RDS means location of a data center in the world. One region in the Amazon RDS can contain multiple availability zones. It is possible to run a DB instance in availability zone. Access to DB instances are controlled with security groups. With a DB parameter group it is possible to manage configuration of a DB engine. With a DB option group it is possible to for example simplify management of a DB engine. Based on Amazon RDS documentation DB option groups are only supported for Oracle DB instances (AWS documentation: What is Amazon Relational Database Service?).

Amazon SimpleDB is a non-relational database service. Amazon SimpleDB consist of domains. A domain in the Amazon SimpleDB is schema-less and can contain one or many items. An item in the Amazon SimpleDB can contain one or many attributes. A domain in the Amazon SimpleDB can grow up to 10 GB. It is possible to divide domain to several domains if the size limit of a domain is achieved. Initially in the Amazon SimpleDB it is possible to use 250 domains. It is possible to get more than 250 domains to Amazon SimpleDB by creating a request to Amazon.com.

Amazon Simple Storage Service (S3) is a data storage service. Amazon S3 makes possible to read, write and delete objects in the Amazon Web Services. In the Amazon S3 one object can contain up to 5 terabytes of data. Every object in the Amazon S3 is stored to a bucket. It is possible to manage Amazon S3 using Amazon S3 APIs or Amazon S3 console. In the Amazon S3 console it is possible to for example create buckets.

4.1.4 Process Information in Cloud Solution

In the Amazon Web Services it is required to use Amazon EC2 in order to be able to process information in the AWS from mobile application point of view. In practice Amazon EC2 makes possible to run custom virtual servers in the Amazon Web Services. In the custom virtual server it is possible to run for example custom web services which can process information in the AWS and can be consumed by a mobile application.

Amazon EC2 offers Amazon Machine Images (AMIs) to provide templates that contains a software configuration. A software configuration can include for example operating system and a web server. From an AMI it is possible to launch one or many instances. An instance is a virtual server having a software configuration defined in the AMI.

It is possible to run scheduled jobs in the Amazon Web Services for example by using Amazon OpsWork. The Amazon OpsWork uses cron daemon to run scheduled jobs. The scheduled jobs are in practice cron jobs. For a cron job it is possible to define a command that will be executed and a time when it should be executed.

4.1.5 Push Information to Mobile Application

In the Amazon Web Services it is possible to push information to a mobile application by using Amazon Simple Notification Service (SNS). Amazon SNS supports following platform notification systems (PNSs): Apple Push Notification service (APNs), Google Cloud Messaging (GCM) and Amazon Device Messaging (ADM).

In the Amazon SNS it is possible to push information directly to a mobile application. In order to push information directly to a mobile application, it is required to register mobile application to mobile platform specific PNS. In the registration mobile application will get a device token (or equivalent depending on the used PNS) from the PNS. Mobile application needs to provide the device token to Amazon SNS. The device token is used by the Amazon SNS to create a mobile endpoint. Amazon SNS can use the created mobile endpoint to send information directly to the mobile application. In the FIGURE 5 Amazon presents overview how Amazon SNS is used to send information directly to mobile endpoint (AWS documentation: Amazon SNS Mobile Push Notifications).



FIGURE 5. Overview how Amazon SNS is used to send information directly to mobile endpoint

Amazon SNS supports also publishing information to a topic. By publishing information to a topic it is possible to push information to all subscribers of a topic. Subscriber of a topic can be one of the following endpoint: mobile application, Amazon SQS, SMS, HTTP/S or email. In the FIGURE 6 Amazon presents how Amazon SNS can be used to send information to topic subscribers (AWS documentation: Amazon SNS Mobile Push Notifications).

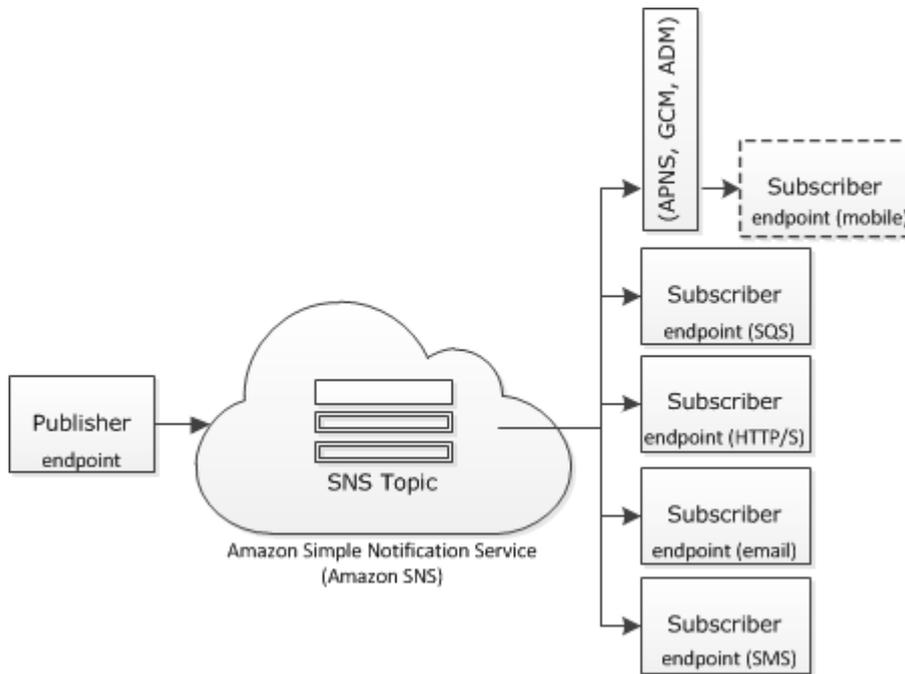


FIGURE 6. Amazon SNS topic subscribers

4.1.6 Security

Authentication and authorization of a mobile application can be done in the Amazon Web Services by using Amazon Identity and Access Management (IAM). Amazon IAM offers following features: IAM users, IAM groups, credentials, SAML providers, IAM roles, server certificates (public key certificates), IAM permissions and IAM policies.

IAM users feature of Amazon IAM can be used to authenticate and authorize an actual person in the Amazon Web Services. It is also possible to use IAM users feature to authenticate and authorize for example an application running in the Amazon EC2.

IAM groups feature makes possible to give permission to group of IAM users. As an example IAM user belonging to IAM group “administrator” is able to do administrative tasks.

IAM user is required to have credentials in order to be authenticated. Credentials in the IAM can be signing certificates, access keys or username and password. It is possible to enhance security of IAM user authentication by using Amazon Multi-Factor Authentication (MFA).

Amazon Multi-Factor Authentication (MFA) makes possible to add extra layer of protection on top of username and password authentication. Extra layer of protection means in practice that Amazon MFA authentication is done using two factors: what user knows and what user has. What user knows means in practice that user needs to provide username and password in the authentication. What user has means in practice that user needs to provide authentication code from an Amazon MFA device in the authentication.

SAML providers are used to establish trust between Amazon Web Services and identity provider like Microsoft Active Directory (AD), Facebook or Google. After the trust between AWS and identity provider has been established, users of the identity provider can access resources of AWS. Identity provider needs to support SAML 2.0 in order to be able to establish trust with Amazon IAM.

IAM roles makes possible to create temporary credentials for a user or application to access resources of Amazon Web Services. Example use of temporary credentials is a mobile application which needs to access resources in the AWS. By using temporary credentials it is possible to give mobile application access to resources in the AWS without storing credentials in the mobile applications. In the IAM role creation it is required to create two policies: trust and access. The trust policy defines who is allowed to act in the role. The access policy defines what actions or resources the trusted entity is allowed access.

IAM Permission makes possible to define who can access to AWS resources and what kind of actions they are allowed to do with the resources. It possible to assign permissions as user-based permission or as resource-based permission. User-based permission means that permission is given to IAM user, group or role. In the user-based permission it is defined what IAM user, group or role can done. Resource-based permission means that permission is given to AWS resource. In the resource-based permission it is defined who can access the resource and what kind of actions they can do with the resource.

IAM policy is a document which lists IAM permissions. The IAM policy is used to assign permissions to IAM user, group, role or resource. In the policy it is possible to define: actions, resources and effects. Actions describe what actions are allowed. Resources defines which resources are allowed for the actions. Effects makes possible to define is user allowed or denied to execute certain action in the resource. In the FIGURE 7 Amazon

presents an example IAM policy which allows “ListBucket” action in the “example_bucket” (AWS documentation: Overview of AWS IAM Policies).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::example_bucket"
    }
  ]
}
```

FIGURE 7. Example IAM policy

4.1.7 Mobile Platform Support

Based on the Amazon Web Services documentation, AWS provides native SDKs for Android, iOS, Java, .NET, JavaScript (browser), JavaScript (Node.js), PHP, Python and Ruby. Amazon Web Services SDK for the .NET supports Windows Store and Windows Phone platforms.

Amazon Web Services supports pushing information to Android, iOS and Kindle Fire mobile applications. Based on the Amazon Web Services documentation pushing information to for example Windows Phone mobile application is not supported.

4.1.8 Pricing Model

Pricing in the Amazon Web Services will be defined based on the selected AWS services. Every AWS service has own pricing factors. For example Amazon Simple Storage Service (S3) has following pricing factors: storage size, reduced redundancy storage, amount of PUT/COPY/POST/LIST requests, amount of GET and other request, inter-region data transfer out, data transfer out and data transfer in.

Amazon Web Services provides a calculator which can be used to estimate how much certain setup of AWS services would cost in a month. The calculator is called: “Amazon Web Services Simple Monthly Calculator” (Amazon Web Services: Amazon Web Services Simple Monthly Calculator). From the FIGURE 8 it is possible to see the Amazon

Web Services Simple Monthly Calculator. In the FIGURE 8 is visible monthly calculation for the Amazon DynamoDB service. It is possible to see for example from the FIGURE 8 that the estimated monthly cost for the estimated use of Amazon DynamoDB is 11,20 \$.

The screenshot shows the Amazon Web Services Simple Monthly Calculator interface. At the top, there is a navigation bar with the Amazon logo and the text "amazon webservices SIMPLE MONTHLY CALCULATOR". Below this, a banner reads "NEW! - AWS lowers its pricing again - Reducing the price for [Amazon S3 by upto 22%](#) and [Amazon EBS b](#)". A yellow box below the banner states "FREE USAGE TIER: New Customers get free usage tier for first 12 months".

The main content area is titled "Services" and "Estimate of your Monthly Bill (\$ 11.20)". It features a "Choose region:" dropdown menu set to "US-East / US Standard (Virginia)". A sidebar on the left lists various AWS services, with "Amazon DynamoDB" highlighted in orange. The main panel for Amazon DynamoDB includes a description: "Amazon DynamoDB is a high performance non-relational database service that is easy to set up, operate, and scale. It is designed to address the core performance and low latency at scale." Below this, a yellow box states: "FREE TIER: Each month, Amazon DynamoDB users pay no charges on the first 100MB of storage, as well as 5 writes/second and 10 reads/second of on".

The configuration section includes the following fields:

- Indexed Data Storage:** Dataset Size: 1 GB
- Provisioned Throughput Capacity *:**
 - Item Size (All attributes): 1 KB
 - Number of items read per second: 100 Reads/Second
 - Read Consistency: Strongly Consistent Eventually Consistent (2x cheaper)
 - Number of items written per second: 10 Writes/Second
- Data Transfer:**
 - Data Transfer Out: 3 GB/Month
 - Data Transfer In: 1 GB/Month

Footnotes at the bottom state:

* Every Amazon DynamoDB table has pre-provisioned the resources it needs to achieve the throughput (read/write) rate you asked for. You are billed for these pre-provisioned resources irrespective of whether you have lower or higher throughput rate.

* You can also purchase Reserved Capacity save up to 53% with a 1-year term and up to 76% with a 3-year term. For more info visit the [Amazon DynamoDB Pricing Page](#)

FIGURE 8. Amazon Web Services Simply Monthly Calculator

At the writing time of this master's thesis Amazon.com had AWS Free Usage Tier offer (Amazon Web Services: AWS Free Usage Tier). From the FIGURE 9 it is possible to see details of the AWS Free Usage Tier offer. The AWS Free Usage Tier offer is only available to new Amazon Web Services customers. The AWS Free Usage Tier offer expires after 12 month from the customer's AWS sign-up date or if customer's AWS service usage exceeds the free usage tiers. After the AWS Free Usage Tier has expired customer will pay from the resources it has used. The AWS Free Usage Tier does not tie to a long-term contract.

AWS Free Usage Tier (Per Month):	
<p>Elastic Compute Cloud (EC2)</p> <ul style="list-style-type: none"> 750 hours of Amazon EC2 Linux/UNIX or RHEL or SLES Micro Instance usage (613 MB of memory and 32-bit and 64-bit platform support) – enough hours to run continuously each month* 750 hours of Amazon EC2 Microsoft Windows Server† Micro Instance usage (613 MB of memory and 32-bit and 64-bit platform support) – enough hours to run continuously each month* 750 hours of an Elastic Load Balancer plus 15 GB data processing* 30 GB of Amazon Elastic Block Storage, plus 2 million I/Os and 1 GB of snapshot storage* 	<p>Simple Workflow (SWF)</p> <ul style="list-style-type: none"> 1,000 Amazon SWF workflow executions can be initiated for free. A total of 10,000 activity tasks, signals, timers and markers, and 30,000 workflow-days can also be used for free**
<p>Simple Storage Service (S3)</p> <ul style="list-style-type: none"> 5 GB of Amazon S3 standard storage, 20,000 Get Requests, and 2,000 Put Requests* 	<p>Simple Queue Service (SQS) and Simple Notification Service (SNS)</p> <ul style="list-style-type: none"> 1,000,000 Requests of Amazon Simple Queue Service** 1,000,000 Requests, 100,000 HTTP notifications and 1,000 email notifications for Amazon Simple Notification Service**
<p>DynamoDB</p> <ul style="list-style-type: none"> 100 MB of storage, 5 units of write capacity, and 10 units of read capacity for Amazon DynamoDB.** 	<p>Amazon Elastic Transcoder</p> <ul style="list-style-type: none"> 20 minutes of SD transcoding or 10 minutes of HD transcoding**
<p>Relational Database Service (RDS)</p> <ul style="list-style-type: none"> 750 hours of Amazon RDS Single-AZ Micro DB Instances, for running MySQL, Oracle BYOL or SQL Server (running SQL Server Express Edition) – enough hours to run a DB Instance continuously each month* 20 GB of database storage 10 million I/Os 20 GB of backup storage for your automated database backups and any user-initiated DB Snapshots 	<p>CloudWatch</p> <ul style="list-style-type: none"> 10 Amazon Cloudwatch metrics, 10 alarms, and 1,000,000 API requests**
	<p>Data Transfer</p> <ul style="list-style-type: none"> 15 GB of bandwidth out aggregated across all AWS services*
	<p>Data Pipeline</p> <ul style="list-style-type: none"> 3 low frequency preconditions running on AWS per month* 5 low frequency activities running on AWS per month*
	<p>ElastiCache</p> <ul style="list-style-type: none"> 750 hours of Amazon ElastiCache - enough hours to run a Cache Node continuously each month.*

* These free tiers are only available to new AWS customers, and are available for 12 months following your AWS sign-up date. When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates (see each [service page](#) for full pricing details). Restrictions apply; see [offer terms](#) for more details.

** These free tiers do not expire after 12 months and are available to both existing and new AWS customers indefinitely.

† The following Windows variants are not eligible for the free usage tier: Microsoft Windows Server 2008 R2 with SQL Server Web, Microsoft Windows Server 2008 R2 with SQL Server Standard, Microsoft Windows 2008 R2 64-bit for Cluster Instances and Microsoft Windows 2008 R2 SQL Server 64-bit for Cluster Instances.

The AWS free usage tier applies to participating services across all AWS regions: US East (Northern Virginia), US West (Oregon), US West (Northern California), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), and South America (Sao Paulo). Your free usage is calculated each month across all regions and automatically applied to your bill – free usage does not accumulate.

FIGURE 9. AWS Free Usage Tier offer

4.2 Google Cloud Platform

Google Cloud Platform is Google's cloud platform. Google has data centers in following places: Berkeley County (South Carolina), Council Bluffs (Iowa), Douglas County (Georgia), Quilicura (Chile), Mayes County (Oklahoma), Lenoir (South Carolina), Dallas (Oregon), Hamina (Finland), St Ghislain (Belgium), Dublin (Ireland), Hongkong, Singapore and Taiwan (Google: Data center locations).

From the products web page of the Google Cloud Platform it is possible to see that Google Cloud Platform categorizes its products into the following categories: compute, storage, big data, services and developer tools (Google Cloud Platform: Products). In the compute category Google Cloud Platform provides the following products: Compute Engine and App Engine. In the storage category Google Cloud Platform provides the following products: Cloud SQL, Cloud Storage and Cloud Datastore. In the big data category Google Cloud Platform provides the following product: BigQuery. In the service Google Cloud Platform provides the following products: Cloud Endpoints, Translate API and Prediction API. In the developer tools category Google Cloud Platform provides the following products: Google Cloud SDK, Push-to-Deploy, Cloud Playground, Android Studio and Google Plugin for Eclipse. From the solutions web page of the Google Cloud Platform it is possible to see that the Google Cloud Platform claims that it can help for example in building of mobile, gaming or Hadoop solution (Google Cloud Platform: Solutions).

To be able to use Google Cloud Platform, a Google Account is required. When first time signing in to Google Cloud Platform with a Google Account it is possible to see the Google Cloud Console where “Get Started with Google Cloud Platform” dialog is open (see FIGURE 10). In the “Get Started with Google Cloud Platform” dialog it is possible to create a new project in the Google Cloud Platform. To be able to create a new project in the Google Cloud Platform it is required to define a name for the project and agree to all “Terms of Service” for the Google Cloud Platform products. After pressing “Get started” button in the “Get Started with Google Cloud Platform” dialog, a new project will be created in the Google Cloud Platform and a “Welcome” dialog will be visible in the Google Cloud Console (see FIGURE 11).

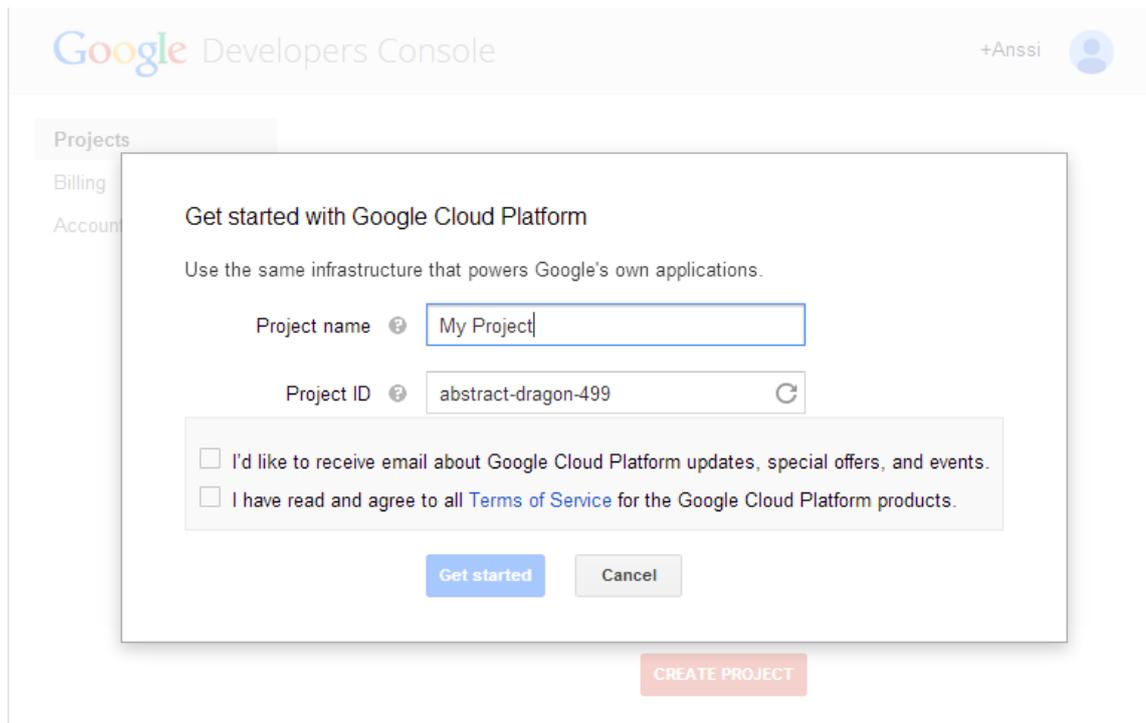


FIGURE 10. “Get started with Google Cloud Platform” dialog in the Google Cloud Console

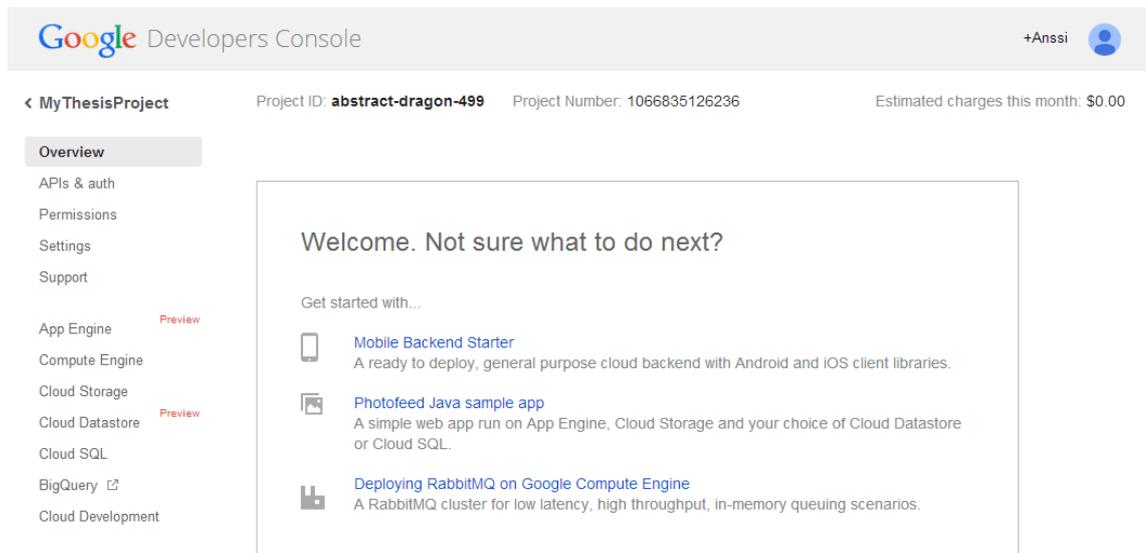


FIGURE 11. “Welcome” dialog in the Google Cloud Console

Registered user of the Google Cloud Platform can access Google Cloud Console. In the main page of the Google Cloud Console it is possible to manage projects, billing and account settings. In the projects page of the Google Cloud Console it is possible to see all registered user related projects in a table. In the table is visible following information about every project: name, id, amount of request, amount of errors and accumulated amount of charges for the current month’s bill. By selecting one project from the projects

table it is possible to navigate to a page in the Google Cloud Console from where it is possible to manage the selected project. In the FIGURE 11 it is possible to see the Google Cloud Console project page for the MyThesisProject project. In the Google Cloud Console project page it is possible to for example manage what kind of products (e.g. App Engine or Cloud Storage) is used from the Google Cloud Platform, add members for the project (Permissions) and manage settings (e.g. enable billing or rename project) of the project.

In the FIGURE 12 Google presents an example mobile solution on the Google Cloud Platform (Google Cloud Platform: Mobile Solutions on Google Cloud Platform). By comparing the example mobile solution to Michael Facemire’s mobile service triangle (see FIGURE 2), it is not possible to identify one service or product from the Google Cloud Platform which could be referred as a Mobile Backend-as-a-Service.

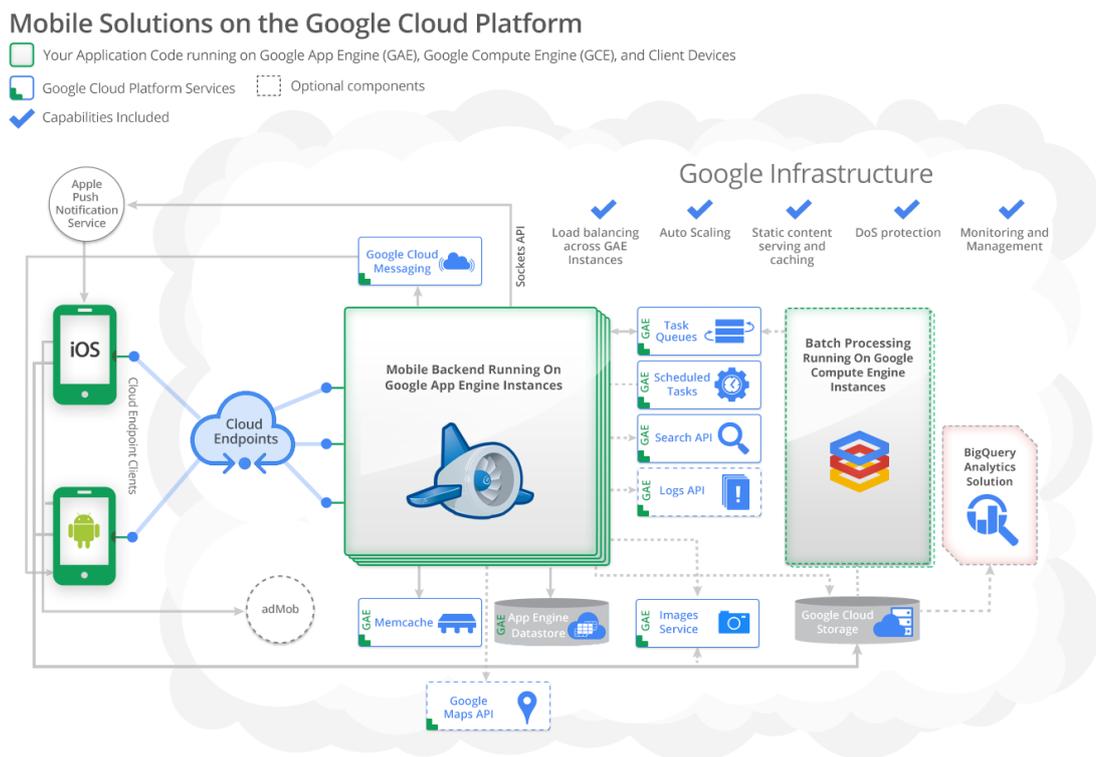


FIGURE 12. Example mobile solution on the Google Cloud Platform

In the article: “Mobile Backend Starter: The Best Mix of BaaS and PaaS” (Google Cloud Platform: Mobile Backend Starter: The Best Mix of BaaS and PaaS), Google explains that instead of providing a BaaS service or product they are providing a Mobile Backend Starter. Technically Mobile Backend Starter is not a BaaS service or product, it is an application that is possible to deploy and run on the Google App Engine with a single

click. Google claims in the article “Mobile Backend Starter: The Best Mix of BaaS and PaaS” that many BaaS providers are black boxes and does not allow developers to solve challenges like scalability, availability and extensibility. Based on the article “Mobile Backend Starter: The Best Mix of BaaS and PaaS” Google tries to solve previously mentioned challenges by having an application that can be deployed to a PaaS instead of creating a BaaS service or product. In the FIGURE 13 it is possible to see architecture of the Mobile Backend Starter.

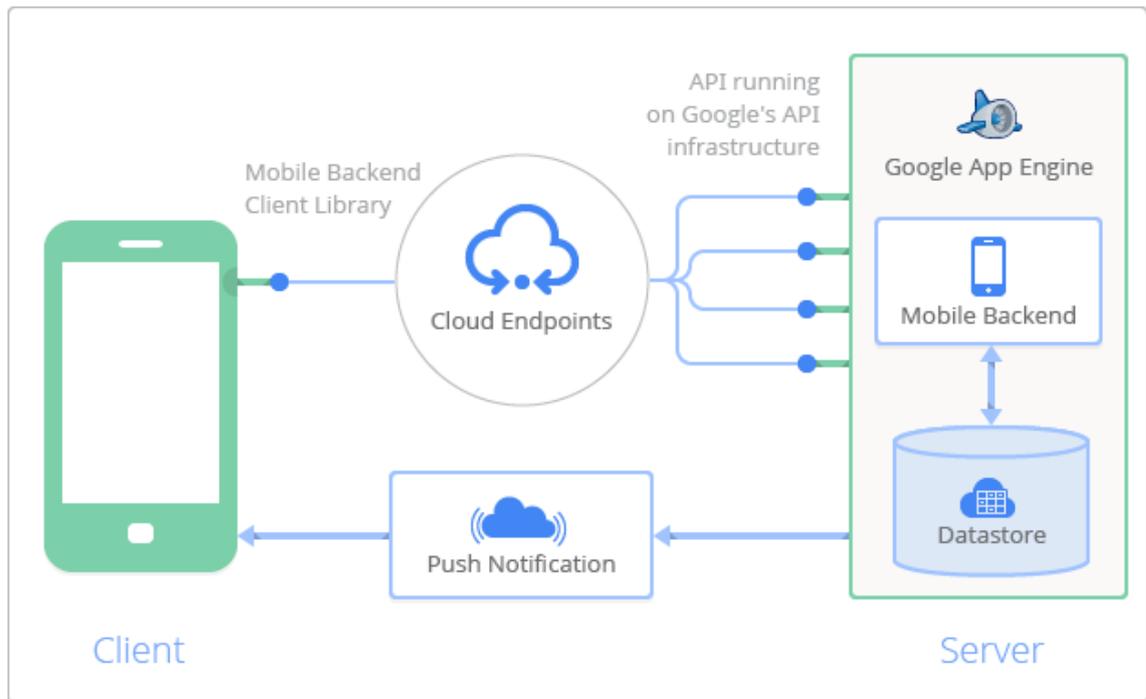


FIGURE 13. The Mobile Backend Starter architecture

4.2.1 Setting up MBaaS Solution

It is possible to set up MBaaS solution in the Google Cloud Platform for example by using the Mobile Backend Starter. It is possible to find the Mobile Backend Starter for example by opening an empty project in the Google Cloud Console and navigating to overview page. In the overview page it is possible to see link to the Mobile Backend Starter. By clicking the Mobile Backend Starter link it is possible to navigate to a page from where it is possible to deploy the Mobile Backend Starter (see FIGURE 14). By clicking the Deploy button in the Mobile Backend Starter deployment page, it is possible to start deployment of the Mobile Backend Starter. After the deployment is done Google

Cloud Console offers a possibility to redeploy the Mobile Backend Starter, change settings of the Mobile Backend Starter or download source codes of example mobile applications.

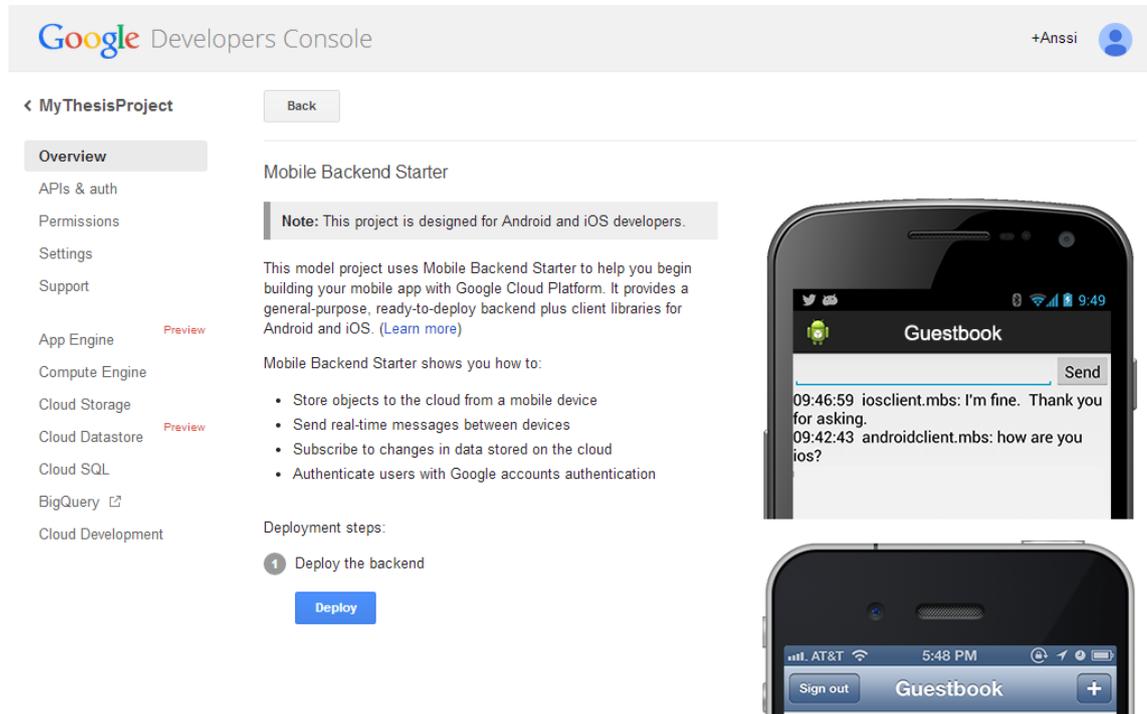


FIGURE 14. Deploy Mobile Backend Starter in the Google Cloud Console

4.2.2 Pull Information to Cloud Solution

The Mobile Backend Starter contains a Cloud Backend API which can be used to pull information from a mobile application to the Google Cloud Platform. The Cloud Backend API will in practice expose CRUD (create, read, update and delete) and query operations to Cloud Datastore. The Cloud Datastore contains entities which are represented as CloudEntity classes in the Cloud Backend API. A class that makes possible CRUD and query operations to Cloud Datastore is named as CloudBackend in the Cloud Backend API. In the FIGURE 15 is an example how it is possible to pull information from a mobile application to the Google Cloud Platform by using CloudEntity and CloudBackend classes of the Cloud Backend API.

```
1
2 CloudEntity entity = new CloudEntity("Persons");
3 entity.put("name", "Kalle Virtanen");
4 entity.put("address", "Mannerheimintie 29");
5
6 CloudBackend backend = new CloudBackend();
7 backend.insert(entity);
8
```

FIGURE 15. Example usage of CloudEntity and CloudBackend classes to pull information from mobile application to cloud solution

It is also possible to pull information from a mobile application to the Google Cloud Platform by using other APIs or services than Cloud Backend API of the Mobile Backend Starter. How to pull information from a mobile application to the Google Cloud Platform varies based on what is the product or service of the Google Cloud Platform that will be used to store the information. For example mobile application can use App Engine SDK to pull information to a Google Cloud SQL product.

It is also possible to pull information from the mobile application to the Google Cloud Platform by creating own APIs on top of Google App Engine using Google Cloud Endpoints. By creating own APIs it is possible to customize what kind of functionalities the Google App Engine will expose to for example mobile applications. By using Google Cloud Endpoints it is possible to have the logic in a place which can be used by different application platforms instead of having own logic for example in the mobile application and own logic in the web application. Google refers APIs build on top of the Google App Engine as API backend (Google Developers: Overview of Google Cloud Endpoints).

4.2.3 Store Information in Cloud Solution

The Mobile Backend Starter uses Cloud Datastore to store information. The Cloud Datastore is a non-relational database. The Cloud Datastore consists of entities. An entity in the Cloud Datastore is schema-less and can contain one or many properties. Property of a Cloud Datastore entity can be for example type of integer, floating-point number, string, date, binary data, link or google account user (Google Developers: Entities, Properties and Keys). It is possible to access Cloud Datastore for example by using the Cloud Backend API of the Mobile Backend Starter.

In the Google Cloud Platform it is possible to store information also to Google Cloud SQL or Google Cloud Storage. Google Cloud SQL is a relational database. Google Cloud SQL is in practice MySQL database with some additional features and some unsupported features. Google Cloud SQL does not for example support following features of the MySQL: SHA2() function, super privilege, user defined functions and LOAD_FILE() statement. Google Cloud SQL provides for example following additional features on top of MySQL features: support for Secure Sockets Layer (SSL) protocol, ability to host database in the cloud, choice of billing options and data encryption when on Google's internal networks. It is possible to use Google Cloud SQL from a mobile application for example by using App Engine SDK. Google Cloud Storage can be used for example as for storing large files and associated metadata. It is possible to use Google Cloud Storage from a mobile application for example by using Google Cloud Storage Client Library.

4.2.4 Process Information in Cloud Solution

The Cloud Backend API of the Mobile Backend Starter makes possible to execute queries against the Cloud Datastore entities. By executing queries against Cloud Datastore entities it is possible to process the entities of the Cloud Datastore and aim to get certain kind of result back from the Google Cloud Platform to the mobile application.

It is also possible to process information in the Google Cloud Platform by creating own APIs on top of Google App Engine using Google Cloud Endpoints. By creating an own API on top of Google App Engine it is possible for example to retrieve information from several sources of the Google Cloud Platform and process information in order to get certain kind of result back from the Google Cloud Platform to the mobile application.

The Google App Engine has Task Queue feature which makes possible to process information asynchronously in the Google App Engine. By using Task Queue it is possible to for example initiate the information processing from the mobile application and let the information processing continue in the Google App Engine. The Google App Engine provides two different task queue configurations: push queues and pull queues. Push queues is the default configuration of the Task Queue. In the push queues configuration tasks are processed automatically based on the processing rate. The processing rate can be configured in the queue configuration to match with the mobile application needs and available

resources. In the push queues configuration tasks are automatically deleted from the Task Queue after the processing of task is completed. The pull queue configured Task Queue can be used only inside the Google App Engine. The pull queues configuration compared to push queues configuration gives more control for the task consumer (e.g. mobile application) to for example decide when the task is executed and when the task is deleted from the task queue. The pull queues configuration also makes possible to use Task Queue feature outside the Google App Engine. It is possible to use pull queues configured Task Queue for example from the mobile application or from the Google Compute Engine using Task Queue REST API.

In the Google App Engine it is possible to schedule processing of information with the App Engine Cron Service. With the App Engine Cron Service it is possible to schedule a cron job to be run automatically for example at certain time or date. The App Engine Cron Service also support intervals (e.g. monthly or daily) to be used in the scheduling. It is possible to use cron job to for example to update cached data every night or send information to the registered users of the mobile application monthly.

4.2.5 Push Information to Mobile Application

The Mobile Backend Starter makes possible to push information from the Google Cloud Platform to the mobile application by using push notifications, continuous queries or publish/subscribe messaging. Continuous queries and publish/subscribe messaging uses push notifications to push the actual information from the Google Cloud Platform to the mobile application.

Push notifications in the Mobile Backend Starter are done using Google Cloud Messaging or Apple Push Notification services. Google Cloud Messaging is used to push information to Android mobile application. Apple Push Notification service is used to push information to iOS mobile application.

Continuous queries in the Mobile Backend Starter makes possible to create data queries to past and future. By making a query to future it is possible to push information from the Google Cloud Platform to a mobile application in the future if the defined query conditions are met. In the FIGURE 16 it is possible to see an example of continuous query. It

is possible to create a continuous query using `CloudQuery` class of the Cloud Backend API. By default `CloudQuery` class is set to make data query to past. To create a data query to future it is required to set scope of the `CloudQuery` class to future by using its `setScope` method. It is possible to define a duration for the continuous query. By defining a duration for the continuous query, it is possible to make the continuous query to be valid only for a certain time (e.g. one day).

```

1
2  CloudQuery continuousQuery = new CloudQuery("Work");
3  continuousQuery.setFilter(F.eq("location", "Turku"));
4  continuousQuery.setScope(Scope.FUTURE_AND_PAST);
5  List<CloudEntity> results = cloudBackendAsync.list(continuousQuery, handler);
6

```

FIGURE 16. Example implementation of continuous query using the Cloud Backend API of the Mobile Backend Starter

Publish/Subscribe messaging of the Mobile Backend Starter makes possible for the mobile application to subscribe to listen a certain topic in the Google Cloud Platform. By listening a topic it is possible to receive topic related messages from the Google Cloud Platform. Mobile application can also publish a message with a topic to the Google Cloud Platform. By publishing a messaging with a topic in the Google Cloud Platform, it is possible to push information through the Google Cloud Platform to all subscribers of a topic. In the FIGURE 17 Google presents an example how it is possible to publish a message having a topic to the Google Cloud Platform (Google Developers: Pub/Sub Messaging on the Mobile Backend Starter).

```

15
16  CloudEntity ce = cloudBackendMessaging.createCloudMessage("dogs");
17  ce.put("dog-name", "Fido");
18  ce.put("dog-age", "5 days");
19  ce.put("message", "I have a new puppy!");
20  cloudBackendMessaging.sendCloudMessage(ce);
21

```

FIGURE 17. Example implementation of publishing a messaging with a topic “dogs” to Google Cloud Platform using the Cloud Backend API of the Mobile Backend Starter

4.2.6 Security

Authentication and authorization of the mobile application can be done in the Mobile Backend Starter using Client ID, Google Account Authentication and Access control for

Cloud Entities. The Mobile Backend Starter has also Authentication/Authorization setting (see FIGURE 18). It is possible to set Authentication/Authorization setting to “Locked Down”, “Open” or “Secured by Client IDs”. Setting “Locked Down” will reject all requests. Setting “Open” will allow all requests. Setting “Secured by Client IDs” will allow only authenticated requests. The Mobile Backend Starter will require two different authentication in order to make request authenticated: Client ID and Google Account Authentication. Client ID makes possible to ensure that only mobile applications having correct Client ID can use the mobile backend. Google Account Authentication makes possible to identify the user of the mobile application.

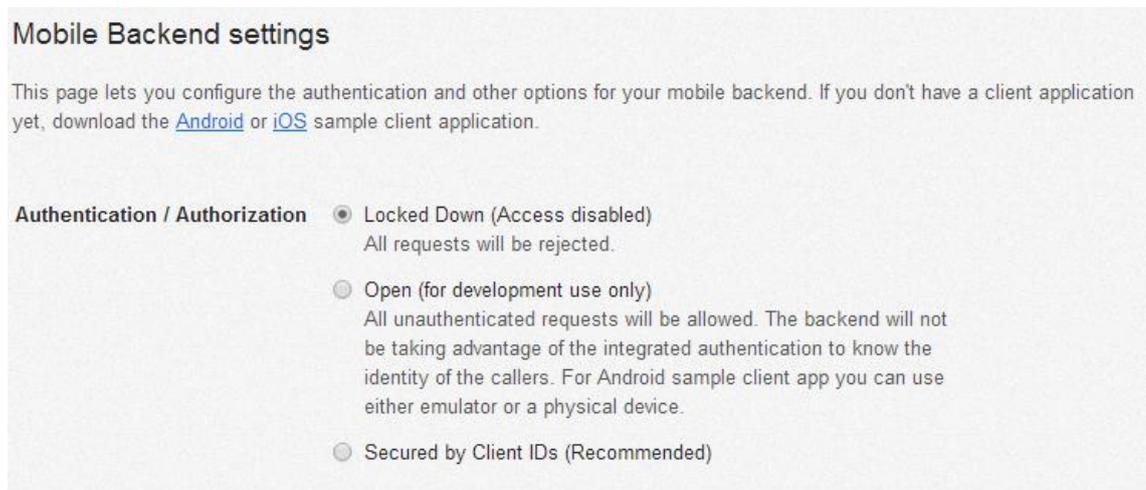


FIGURE 18. Authentication / Authorization settings of the Mobile Backend Starter

Access Control for Cloud Entities is a feature of the Cloud Backend API. Access Control for Cloud Entities makes in practice possible to define different kind of access levels for cloud entities. Access level for a Cloud Entity feature can be taken in to use by adding a prefix on the kind name of a cloud entity. In the FIGURE 19 Google has defined all prefixes available for cloud entities (Google Developers: Authentication on the Mobile Backend Starter). From the FIGURE 19 it is possible see that there exist three different kind name prefixes: [private], (no prefix) and [public]. By having a prefix [private] on the cloud entity kind name it is possible to define that only entity owner can get, query, update or delete cloud entity. By having entity kind name without any prefixes it is possible to define that only entity owner can update or delete cloud entity. By having a prefix [public] on the cloud entity kind name it is possible to define that anyone can get, query, update, delete or insert cloud entity.

Kind Name Prefix	get/query	update/delete	insert
[private]	entity owner	entity owner	anyone
(no prefix)	anyone	entity owner	anyone
[public]	anyone	anyone	anyone

FIGURE 19. Kind name prefixes for cloud entities

Authentication and authorization in the Google Cloud Endpoints is done using Client IDs and Google Account Authentication. To protect a method of custom API which has been done using Google Cloud Endpoints, it is required to do following steps: add User parameter to the method which needs to be protected, add email scope to @Api annotation and specify clientIds whitelist in the @Api annotation. The protected API method needs to provide a custom logic to handle case where the user is authenticated or is not authenticated. If the Google Cloud Endpoints is not able to authenticate the user of the method, the parameter User will be null. If the Google Cloud Endpoints is able to authenticate the user of the method, Google Cloud Endpoints will set a valid user to User parameter. By having custom logic in the API method it is possible to provide for example custom authorization for the authenticated user.

4.2.7 Mobile Platform Support

The Mobile Backend Starter supports Android and iOS mobile platforms. Google does not support in the Mobile Backend Starter for example Windows Phone mobile platform.

Google App Engine provides native SDKs for following languages: Java, Python, PHP and Go (Google Developers: Downloads). Google App Engine does not provide native SDK for example for Microsoft's C#.

Google Cloud Endpoints API libraries are available for following languages: Java, Python, PHP and GO (Google Cloud Platform: Cloud Endpoints). Google Cloud Endpoints does not provide native APIs for example for Microsoft's C#.

4.2.8 Pricing Model

Pricing in the Google Cloud Platform is defined based on used Google Cloud Platform products. Every product in the Google Cloud Platform has own pricing models and factors.

Google Cloud Platform provides a calculator called “Google Cloud Pricing Calculator”. With the Google Cloud Pricing Calculator it is not possible to calculate cost estimates for every Google Cloud Platform product. By using the Google Cloud Pricing Calculator it is possible to calculate cost estimate for Google Compute Engine, Google Cloud Storage and Google Cloud SQL. It is not possible for example to calculate cost estimate for Google App Engine usage. The cost estimate in the Google Cloud Pricing Calculator is calculated for one month usage.

The Mobile Backend Starter uses Google App Engine. In the FIGURE 20 Google has defined pricing categories of the Google App Engine (Google Cloud Platform: App Engine Pricing). In the FIGURE 20 is also visible what kinds of features from the Google App Engine are available for a certain pricing category. From the FIGURE 20 it is possible to see for example that by upgrading pricing category from “Free” to “Paid” it is possible to get usage based pricing, infinitely scalable and SLA.

Google has defined usage quotas for the Google App Engine in the FIGURE 21 (Google Cloud Platform: App Engine Pricing). Google has defined usage quotas so that every application is free within certain usage quota. The usage quotas in the App Engine are defined for one day usage. By upgrading pricing category of the Google App Engine from “Free” to “Paid” it is possible to use more hosting and APIs than it is possible in Free pricing category. From the FIGURE 21 it is possible to see for example that Google App Engine will give 1GB storage for free and will charge 0.18\$ in a month for every gigabyte that goes over the free limit.

Pricing

	Free	Paid	Premier
Price			Starts at \$150/account/month
Dynamic scaling	Yes	Yes	Yes
Java Runtime	Yes	Yes	Yes
Python Runtime	Yes	Yes	Yes
Go Runtime	Yes	Yes	Yes
Usage based pricing		Yes	Yes
Infinitely scalable		Yes	Yes
SLA		Yes	Yes
Operational support			Yes
Tools			
Google Plugin for Eclipse	Yes	Yes	Yes
Code upload/download	Yes	Yes	Yes
Graph History	Yes	Yes	Yes
Request Logs	Yes	Yes	Yes
Developer Access Control	Yes	Yes	Yes

FIGURE 20. Pricing categories of the Google App Engine

Usage Quotas		
All applications are free within a usage quota that is reset daily. Paid applications can use more at the prices below.		
Hosting	Free quota per app per day	Pricing if you exceed your free quota
On-demand Frontend Instances	28 free instance hours	\$0.08 / hour
Reserved Frontend Instances		\$0.05 / hour
High Replication Datastore Storage	1 GB total limit	\$0.18 / GB / month
Outgoing Network Traffic	1 GB	\$0.12 / GB
Incoming Network Traffic	1 GB	Free
APIs		
Datastore API	50k free read/write/small	\$0.09/100k write ops \$0.06/100k read ops \$0.01/100k small ops
Search API	1000 basic operations 0.01 GB indexing documents 0.25 GB document storage 100 complex searches 1000 simple searches	\$0.10 / 10k basic operations \$2.00 / GB indexing documents \$0.18 / GB / month document storage \$0.60 / 10k complex searches \$0.13 / 10k simple searches
Blobstore Storage	5 GB total limit	\$0.13 / GB / month
Email API	100 recipients	\$0.01 / 100 recipients
XMPP API	10k stanzas	\$0.10 / 100k stanza
Channel API	100 channels opened	\$0.01 / 100 channels opened
Image Manipulation API	Yes	Yes
Shared Memcache	Yes	Yes
Dedicated Memcache	No free quota	\$0.12 / GB / hour
Users API	Yes	Yes
Task Queue	Yes	Yes
Files API	Yes	Yes
URL Fetch API	Yes	Yes
Cron	Yes	Yes
SNI SSL Certificates	No free quota	\$9.00 / set of five SNI certificate slots / month
SSL Virtual IPs	No free quota	\$39.00 / virtual IP / month
Pagespeed Network Traffic	No free quota	\$0.39 / GB

FIGURE 21. Usage Quotas of the Google App Engine

Google Cloud Endpoints is free for applications that run on top of Google App Engine or Google Compute Engine. From the FIGURE 21 is possible to see for example the free incoming and outgoing traffic quotas for the Google App Engine. If the pricing category of the Google App Engine is “Free” and Google Cloud Endpoints is running on top of it, it is possible to use the Google Cloud Endpoints having maximum of 1GB daily outgoing and incoming traffic. By upgrading from “Free” pricing category to “Paid” it is possible to have more than 1 GB outgoing and incoming traffic per day. It also good to notice that although the Google Cloud Endpoints as a product is free, the usage of the Google App Engine or Google Compute Engine might cost something.

4.3 Microsoft Azure

Microsoft Azure is Microsoft’s cloud platform. Microsoft Azure was announced October 2008 at Microsoft’s Professional Developer Conference (Microsoft Unveils Windows Azure at Professional Developers Conference). First of February 2010 Microsoft announced general availability of Microsoft Azure (Doug Hauger, 2010). Previous name of the Microsoft Azure was Windows Azure. Microsoft changed name from Windows Azure to Microsoft Azure in April 2014 (Microsoft Azure: Upcoming Name Change for Windows Azure).

From the documentation of the Microsoft Azure it is possible to see that Microsoft Azure categories its services to following categories: compute, data services, app services and networks (Microsoft Azure: Documentation). In the compute category Microsoft Azure provides following services: Virtual Machines, Cloud Services, Web Sites and Mobile Services. In the data services category Microsoft Azure provides following services: Storage, SQL Database, HDInsight, Cache and Recovery Services. In the app services category Microsoft Azure provides following services: Media Services, Service Bus, Notification Hubs, Scheduler, BizTalk Services, Active Directory and Multi-Factor Authentication. In the networks category Microsoft Azure provides following services: Express-Route, Virtual Networks and Traffic Manager.

Sign up to Microsoft Azure is required in order to be able to use it. To be able to sign up for the Microsoft Azure, Microsoft Account is required. The sign up process of Microsoft Azure contains four steps: about you, mobile verification, payment information and

agreement (see FIGURE 22). The about you step collect information like first and last name of the person signing up to the Microsoft Azure. The mobile verification steps makes sure that person signing up to the Microsoft Azure is who he/she claims to be. The payment information steps collects information like credit card number in order to make possible payments between person signing up to the Microsoft Azure and Microsoft Azure. The agreements steps makes sure that person signing up to the Microsoft Azure will agree “Windows Azure Agreement”, “Offer Details” and “Privacy Statement”. The agreements steps also makes possible for the person signing up to the Microsoft Azure to get special Microsoft Azure offers.

FIGURE 22. Microsoft Azure sign up web page

Registered user of the Microsoft Azure is able to access Microsoft Azure management portal from where it is possible to manage the Microsoft Azure account (see FIGURE 23). In the left side of the management portal is a navigation to different services / features of the Microsoft Azure. In the right side of the management portal is displayed details of the currently selected service / feature. From the FIGURE 23 it is possible for example to see that in the right side of the management portal is displayed details of all items feature.

In the bottom of the management portal is displayed actions related to currently selected service / feature. From the FIGURE 23 it is possible to see for example that there is “New” action available for the all items feature. In the top of the management portal it is possible for example to see user account related actions like: sign out, change password and view my bill (FIGURE 24).

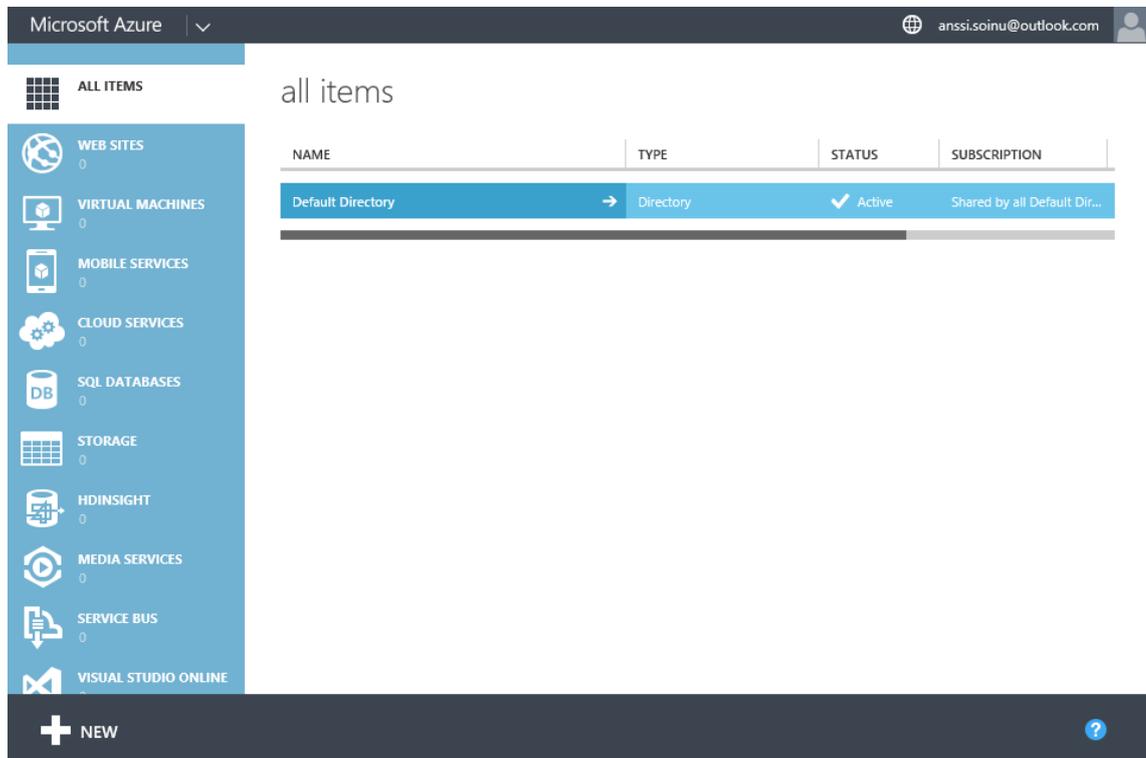


FIGURE 23. Screen capture from the Microsoft Azure management portal

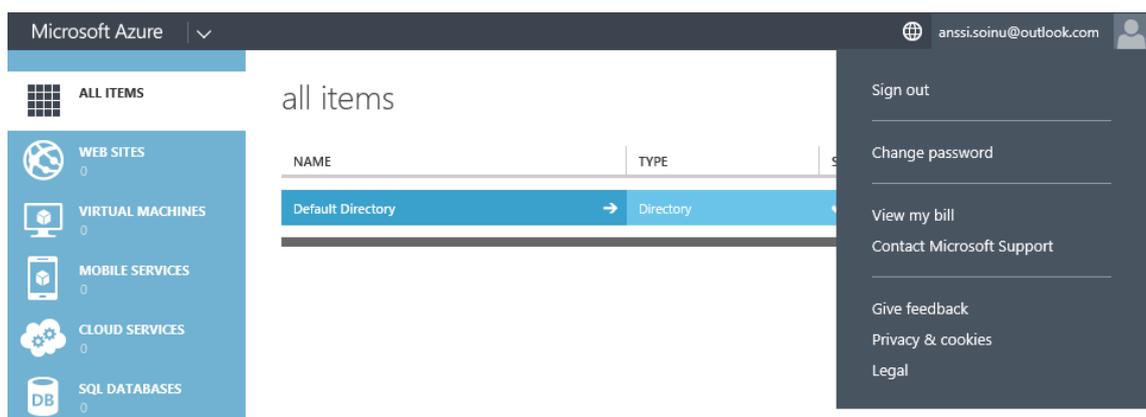


FIGURE 24. Screen capture of user account related actions in the Microsoft Azure management portal

Microsoft Azure has own documentation for mobile solutions (Microsoft Azure: Mobile Apps). In the previously mentioned documentation is said as follows: “Azure gives you

flexibility to power your mobile apps using Virtual Machines, Cloud Services or Mobile Services” (Microsoft Azure: Mobile Apps). By comparing previously mentioned options to Michael Facemire’s mobile service triangle (see FIGURE 2) it is possible to identify that virtual machines are in the Infrastructure-as-a-Service layer, cloud services belongs to Platform-as-a-Service layer and mobile services belongs to the Mobile Backend-as-a-Service layer. Since this master’s thesis is scoped to study how Microsoft Azure can offer MBaaS cloud solution, this master’s thesis will scope study of Microsoft Azure mainly to Mobile Services.

4.3.1 Setting up MBaaS solution

To create Mobile Backend-as-a-Service solution in the Microsoft Azure, a Mobile Service needs to be created. It is possible to create a new Mobile Service using Microsoft Azure management portal. The creation of a Mobile Service can be initiated by clicking the “New” action in the bottom of the Microsoft Azure management portal (see FIGURE 25). The creation of Mobile Service will contain two steps. In the first step will be defined location (URL), region, database and backend type (JavaScript or .NET) for the new Mobile Service. In the last step will be defined database related configuration. Configuration of the database can vary based on the selected database in the first step. It is possible to configure Mobile Service to use an existing SQL database, to create a new SQL database or to create a new free 20 MB SQL database.

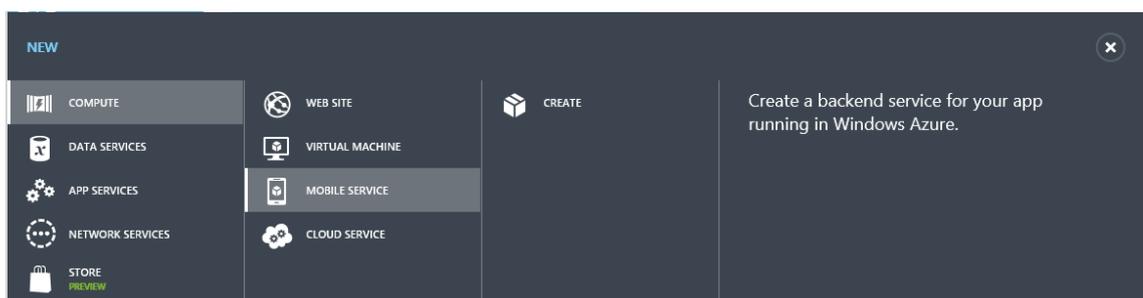


FIGURE 25. Creating a new Mobile Service in the Microsoft Azure management portal

The .NET backend type for the Mobile Service had at the written time of this master’s thesis a preview status. The preview status in the Microsoft Azure means a feature / product that is new in the Microsoft Azure and has been made available for the users to try out. A feature / product that has a preview status in the Microsoft Azure is not a final

product / feature. This master's thesis is scoped mainly to study JavaScript backend type of the Microsoft Azure Mobile Services because at the written time of this master's thesis .NET backend type of a Mobile Service was not a final product / feature of the Microsoft Azure.

4.3.2 Pull Information to Cloud Solution

In order to make possible to pull information from the mobile application to Mobile Service, a new data table needs to be created in the Mobile Service. The data table in the Mobile Service is an actual database table in the Microsoft Azure SQL Database. Mobile application can access the created data table by using Mobile Service SDK. By accessing the data table using Mobile Service SDK it is possible to pull information from the mobile application to the Mobile Service. In the Mobile Service a new data table can be created for example using Microsoft Azure management portal (see FIGURE 26). In the creation of a new data table it is required to define name and permissions (insert, update, delete and read) for it.

It is also possible to pull information from the mobile application to the Microsoft Azure Mobile Service by creating custom APIs. With a custom API it is possible to create an endpoint in the Mobile Service which does not map to insert, update, delete or read operation of a data table.

MOBILE SERVICES: DATA

Create New Table

TABLE NAME

Question

Tables have changed, existing client code may need to be modified. [Learn more](#)

You can set a permission level against each operation for your table. ?

INSERT PERMISSION

Anybody with the Application Key

UPDATE PERMISSION

Anybody with the Application Key

DELETE PERMISSION

Anybody with the Application Key

READ PERMISSION

Anybody with the Application Key

✓

FIGURE 26. “Create New Table” pop-up in the Microsoft Azure management portal

4.3.3 Store Information in Cloud Solution

Microsoft Azure Mobile Service uses Microsoft Azure SQL Database to store information. The database which will be used by the Mobile Service is initially defined and created when the Mobile Service is created. After the Mobile Service has been created it is possible to change the current SQL database of it to another SQL database. The change of the current SQL database can be done using Microsoft Azure management portal.

Microsoft Azure provides in its data service category service called Storage. The Storage service contains two sub services: Blob storage and Table service. Blob storage is a service for storing large objects (for example videos or images). Table service is a non-relational datastore. Table service consist of following components: storage account, tables and entities. One storage account can contain one or many tables. One table can contain one or many entities. One entity can have one or many properties.

Microsoft Azure Mobile Services uses Microsoft Azure SQL Database by default to store information. It is possible also to use Microsoft Azure Storage service or some other stor-

age / database in the Mobile Service. In practice to use something else for storing information than the Microsoft Azure SQL Database, it is required to introduce custom code to the Mobile Service.

4.3.4 Process Information in Cloud Solution

In the Microsoft Azure Mobile Service it is possible to add custom JavaScript code to a data table related insert, update, delete or read operation. The custom JavaScript code can be created in the Microsoft Azure management portal (see FIGURE 27). It is also possible to create the custom JavaScript in a local development environment by using tools like Microsoft Visual Studio or Sublime Text. To make possible to deploy locally created custom JavaScript code to the Mobile Service, source control needs to be enabled in the Mobile Service. It is possible to enable the source control for the Mobile Service for example in the Dashboard of the Mobile Service (see FIGURE 28). In the local development environment needs to be Git installed in order to make possible to deploy the locally created custom JavaScript from the local development environment to the Mobile Service.

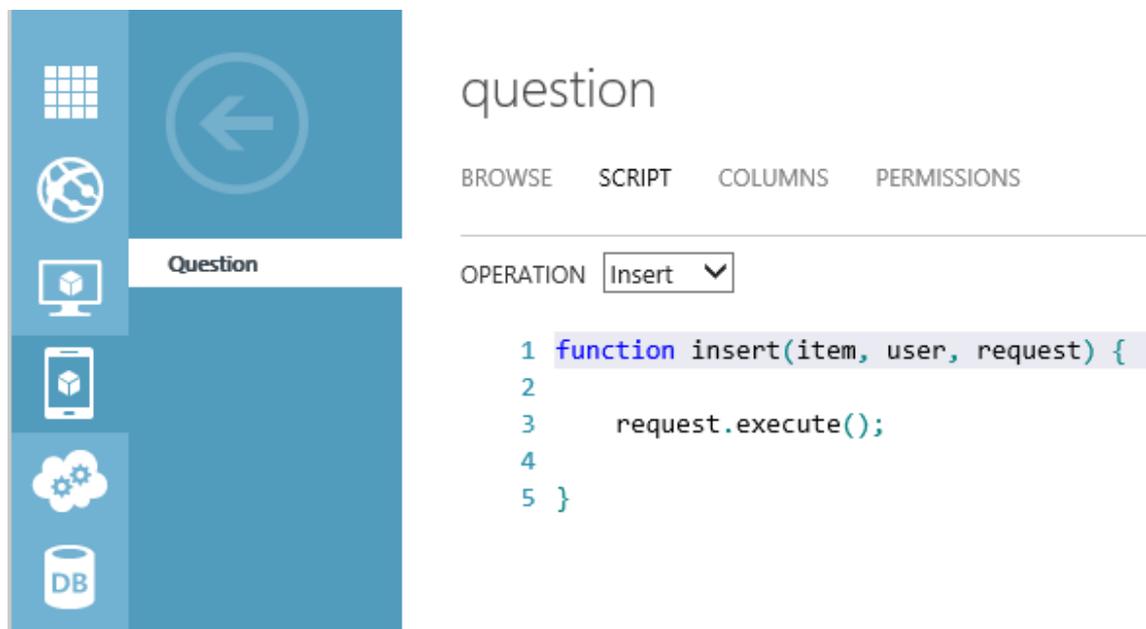


FIGURE 27. Custom script in the insert operation of the Question data table

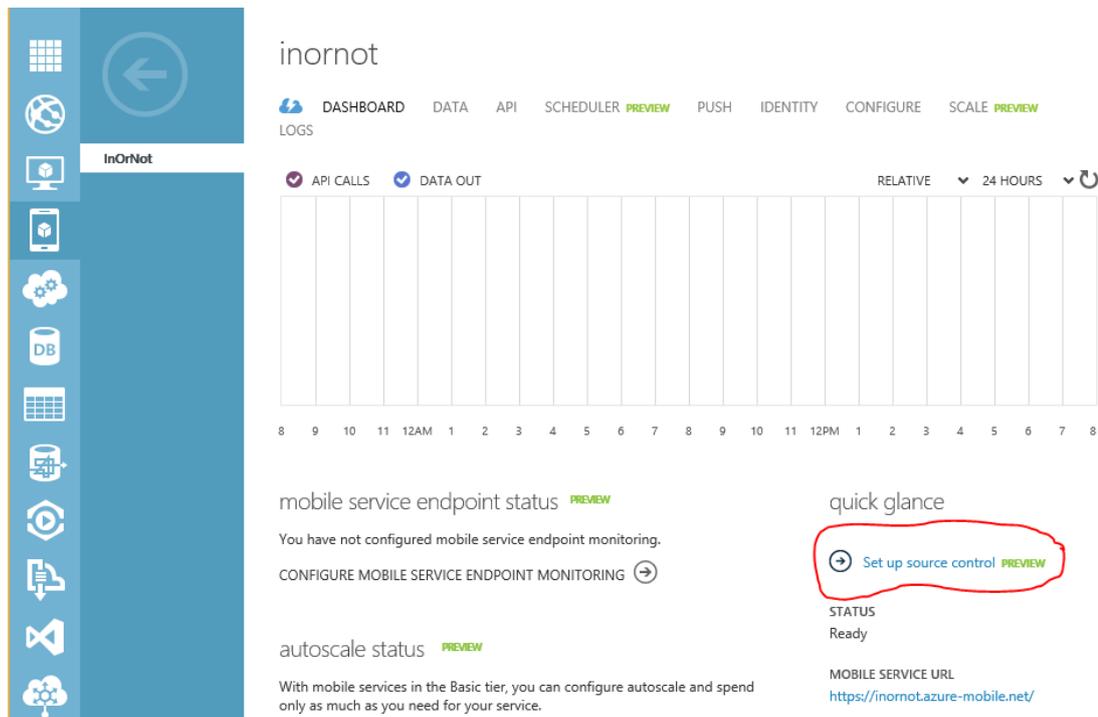


FIGURE 28. Mobile Service Dashboard

In the Microsoft Azure Mobile Service it is possible to create a custom API. With a custom API it is possible to create an endpoint in the Mobile Service which does not map to insert, update, delete or read operation of a data table. With a custom API it is possible to for example access to all data in the Mobile Service related database and create statistics out of that data. The logic behind a custom API needs to be written with JavaScript. It is possible to create the custom API related JavaScript code with the Microsoft Azure management portal (see FIGURE 29). It is also possible to create the custom API related JavaScript code in a local development environment same way as custom JavaScript can be created for the data table related insert, update, delete or read operation.



FIGURE 29. Custom API in the Mobile Service

In the Microsoft Azure Mobile Service exists feature called Scheduler which makes possible to define tasks that will be executed periodically or on demand. It is possible to use scheduler task to for example to update cached data every night or send information to the registered users of the mobile application monthly. The logic in the scheduled task needs to be written with JavaScript. It is possible to create the logic for the scheduled task using Microsoft Azure management portal or in a local development environment same way as custom JavaScript can be created for the data table related insert, update, delete or read operation.

4.3.5 Push Information to Mobile Application

In order to make possible to push information from the Microsoft Azure Mobile Service to a mobile application, a channel needs to be created between the Mobile Service and the mobile application. The channel creation is mobile platform specific. In the FIGURE 30 Windows Apps Team (2013) presents an example push notification lifecycle for the Windows Phone and Windows 8 device. From the FIGURE 30 is possible to see that there exists a platform notification system (PNS) between the mobile application and the app backend (Mobile Service). The PNS is used to create the channel between mobile application and the app backend. The PNS is different for different mobile platforms. For example:

- to make possible to push information to Android mobile application, the channel needs to be created using Google Cloud Messaging (GCM),
- to make possible to push information to Windows Phone mobile application, the channel needs to be created using Microsoft Push Notification Service (MPNS) and
- to make possible to push information to iOS mobile application, the channel needs to be create using Apple Push Notification service (APNs).

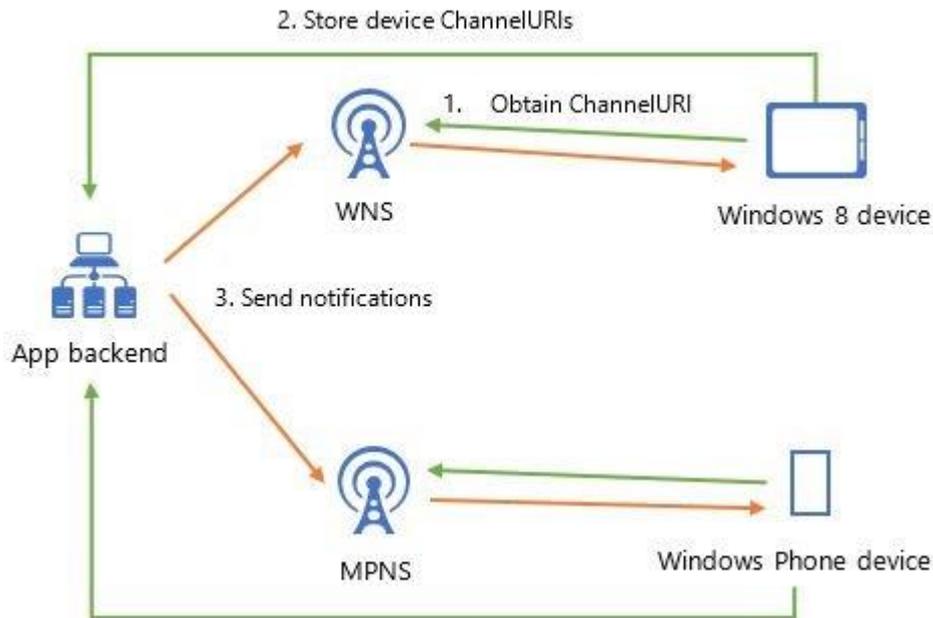


FIGURE 30. Push notification lifecycle

After the channel is created, information about the created channel needs to be stored to a place which can be accessed by the Mobile Service. The channel information can be stored for example to a data table in the Mobile Service. In the FIGURE 31 is an example of a data table which stores channel information. In the channel data table is stored for example the URI of the channel.

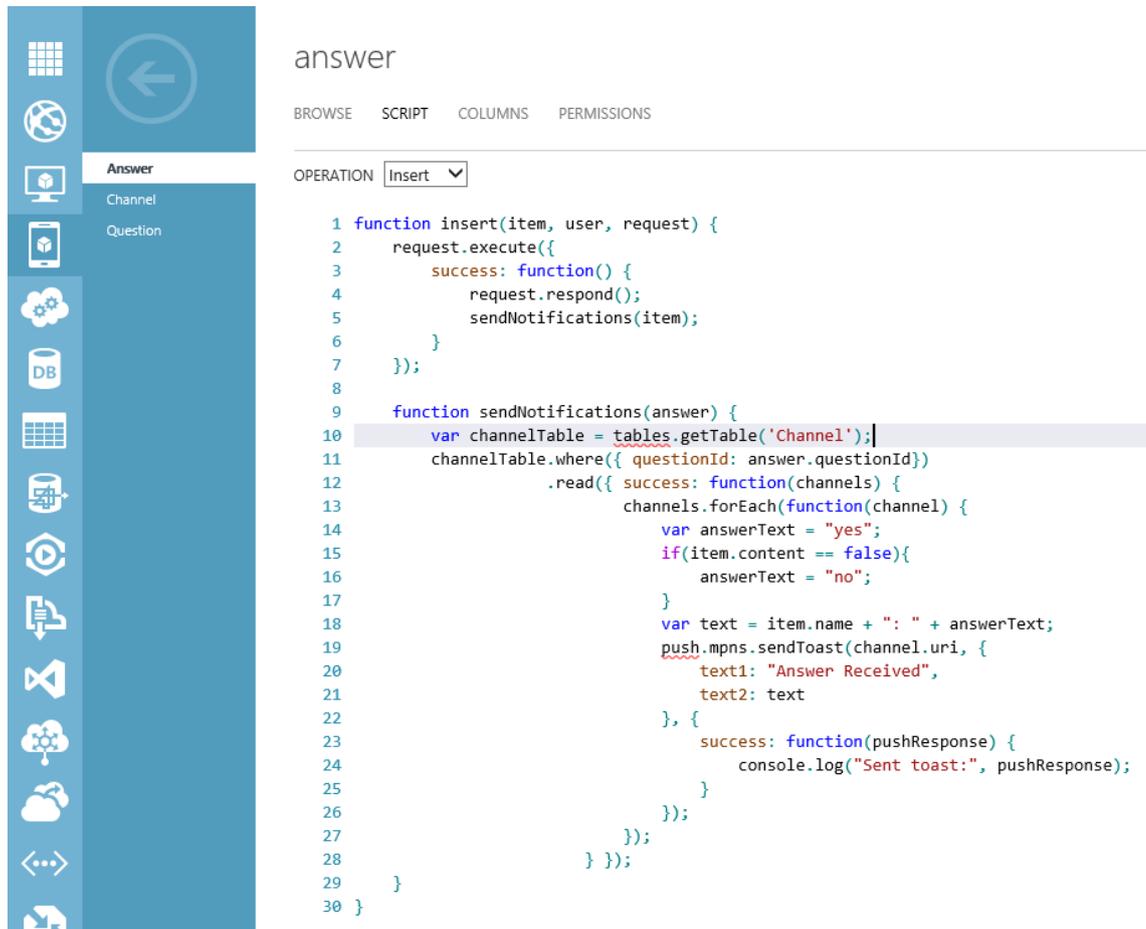
The screenshot shows a data table named "channel" with the following columns and types:

COLUMN NAME	TYPE	INDEX
id	string	✓ Indexed
_createdAt	date	✓ Indexed
_updatedAt	date	
_version	timestamp (MSSQL)	
uri	string	
questionid	string	

FIGURE 31. Example of a data table which will store channel information

In the Mobile Service it is possible to push information to an existing channel for example by creating a script which is part of a certain data table operation (insert, update, delete or create) or part of a custom API. The push notification logic needs to be written using JavaScript. In the FIGURE 32 is an example of script which will push information as a

toast to a Windows Phone mobile application (`push.mpns.sendToast`). The example script in the FIGURE 32 is part of answer data table insert operation.



The screenshot shows the Azure portal interface for a table named 'answer'. The left sidebar contains navigation icons for various services, including 'Answer', 'Channel', 'Question', 'DB', and 'Question'. The main area displays the 'SCRIPT' tab for the 'insert' operation. The script is as follows:

```

1 function insert(item, user, request) {
2   request.execute({
3     success: function() {
4       request.respond();
5       sendNotifications(item);
6     }
7   });
8 }
9
10 function sendNotifications(answer) {
11   var channelTable = tables.getTable('Channel');
12   channelTable.where({ questionId: answer.questionId})
13     .read({ success: function(channels) {
14       channels.forEach(function(channel) {
15         var answerText = "yes";
16         if(item.content == false){
17           answerText = "no";
18         }
19         var text = item.name + ": " + answerText;
20         push.mpns.sendToast(channel.uri, {
21           text1: "Answer Received",
22           text2: text
23         }, {
24           success: function(pushResponse) {
25             console.log("Sent toast:", pushResponse);
26           }
27         });
28       });
29     });
30 }

```

FIGURE 32. Example script to push information from the Mobile Service to a Windows Phone mobile application

In the Microsoft Azure it is also possible to use Notification Hubs to push information to a mobile application. It is possible to use Notification Hub from a Mobile Service. The Notification Hub makes possible to push notification to a mobile application without handling the PNS related logic in the Mobile Service. In the FIGURE 33 Microsoft presents that the App-back-end (Mobile Service) uses Notification Hub instead of directly using PNS to push information to a mobile application (Microsoft Azure: Notification Hubs Overview).

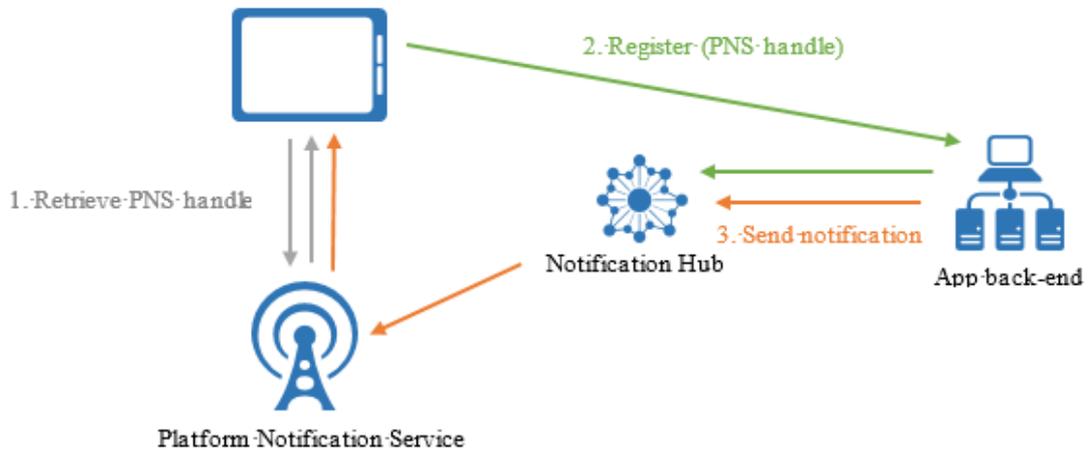


FIGURE 33. Push information using Microsoft Azure Notification Hub

Notification Hubs makes possible to push (publish) information to all registered subscribers or to certain subscriber(s). In order to publish information to a certain subscriber(s) it is required to define a tag to information. It is possible to use tags to for example publish information to subscribers which are currently located in a certain location (e.g. “location_Helsinki”). Notification Hubs supports also tag expressions. By using tag expression it is possible to publish information or subscribe to listen information based on an expression. A tag expression could be for example: “(brand_Volvo && (model_V70 || model_S60)) || brand_BMW” (information related to Volvo V70 or Volvo S60 or BMW).

Notification Hubs provides possibility for the subscriber to define a template in the registration. By using a template subscriber can tell to Notification Hub that it would like to have the information published in a certain format. By using templates it is possible to handle information in the Notification Hub without having subscriber platform related logic in the Notification Hub. Usage of templates can make possible also for example personalization or localization of information.

4.3.6 Security

In order to authenticate users in the Mobile Service, the mobile application needs to be registered with an identity provider. Microsoft Azure Mobile Service supports use of following identity providers: Microsoft, Facebook, Twitter, Google and Microsoft Azure

Active Directory. After the mobile application has been registered with an identity provider, the generated client secret of the identity provider needs to be registered to Mobile Service.

In the Microsoft Azure Mobile Service it is possible to set the permissions for data tables. It is possible to define permissions to all operations (insert, update, delete and read) of a data table. In FIGURE 34 is an example of permission settings of a data table. From the FIGURE 34 it is possible to see that the permission can be: Everyone, Anybody with the Application Key, Only Authenticated User or Only Scripts and Admins. Permission setting “Everyone” means that the operation of the data table is not restricted (it can be used by everyone). Permission setting “Anybody with the Application Key” means that the operation of the data table is restricted to applications using correct application key. Permission setting “Only Authenticated User” means that the operation of the data table is restricted to applications that are used by authenticated users. Permission setting “Only Scripts and Admins” means that the operation of the data table is restricted to Mobile Service scripts and applications using correct master key.

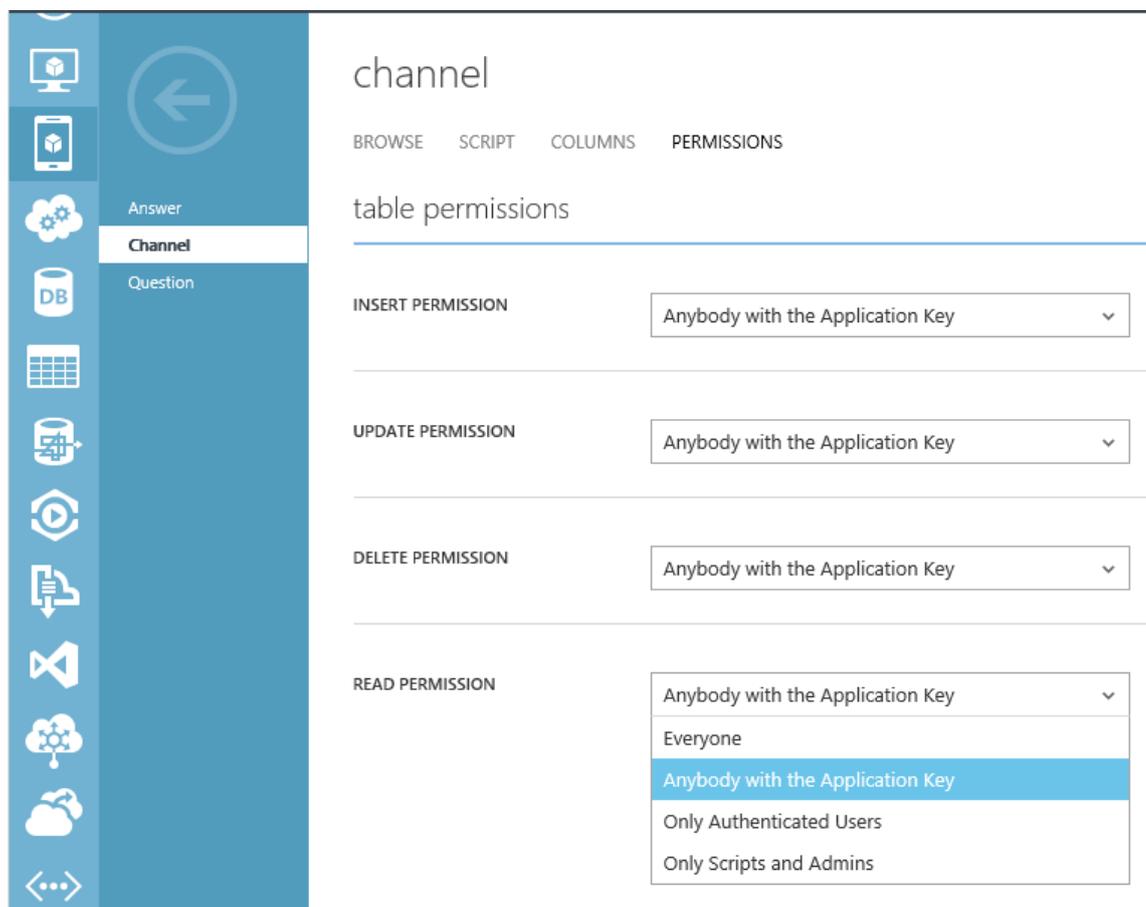


FIGURE 34. Permission settings of the channel data table

In the Microsoft Azure Mobile Service it is possible to set the permissions for custom APIs. It is possible to define permissions to all operation (get, post, put, patch and delete) of a custom API. In FIGURE 35 is an example of permission settings of a custom API. From the FIGURE 35 it is possible to see that the permission can be: Everyone, Anybody with the Application Key, Only Authenticated User or Only Administrators. Permission setting “Everyone” means that the operation of the custom API is not restricted (it can be used by everyone). Permission setting “Anybody with the Application Key” means that the operation of the custom API is restricted to applications using correct application key. Permission setting “Only Authenticated User” means that the operation of the custom API is restricted to applications that are used by authenticated users. Permission setting “Only Administrators” means that the operation of the custom API is restricted to applications using correct master key.

The screenshot displays the 'getstatistics' API configuration page in the Azure Mobile Service console. The left sidebar contains navigation icons for various services. The main content area shows the 'api permissions' section for the 'getstatistics' API. The permissions are configured as follows:

Operation	Permission
GET PERMISSION	Anybody with the Application Key
POST PERMISSION	Anybody with the Application Key
PUT PERMISSION	Anybody with the Application Key
PATCH PERMISSION	Anybody with the Application Key
DELETE PERMISSION	Anybody with the Application Key

The 'DELETE PERMISSION' dropdown menu is expanded, showing the following options:

- Anybody with the Application Key
- Everyone
- Anybody with the Application Key (highlighted)
- Only Authenticated Users
- Only Administrators

FIGURE 35. Permission settings of the getStatistics custom API

In Microsoft Azure Mobile Service it is possible to improve authorization of a user or group of users for example by using a data table. It is possible to create a data table in the Mobile Service which will define authorization rules for users or group of users. The authorization rules data table can be used in the Mobile Service scripts (data table or custom API) for example to check that can a certain user execute a certain operation.

In the Microsoft Azure Mobile Service it is possible to define from which host it is possible to use the Mobile Service. This cross-origin resource sharing configuration applies only to JavaScript based applications running in Internet browser. The cross-origin resource sharing configuration can be found from the configuration of the Mobile Service. From the FIGURE 36 it is possible to see example of cross-origin resource sharing configuration.

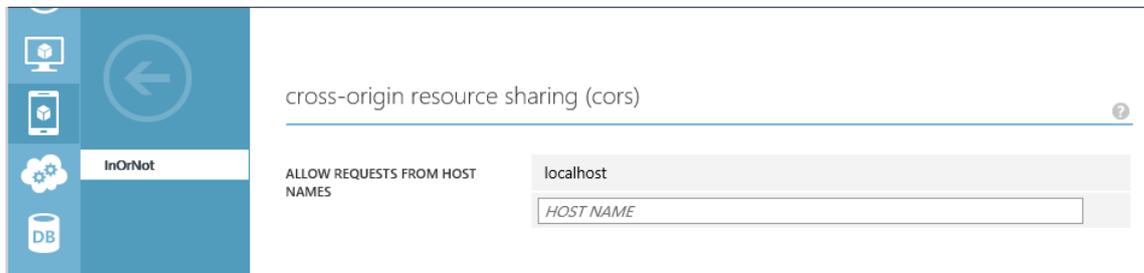


FIGURE 36. Cross-origin resource sharing (cors) configuration of the InOrNot Mobile Service

4.3.7 Mobile Platform Support

Microsoft Azure Mobile Service provides native SDKs for Windows Store, Windows Phone, Android and iOS mobile applications (Microsoft Azure: Developer tools for Mobile Services). Microsoft Azure Mobile Service provides also SDK for Xamarin. It is possible with Xamarin to create native mobile applications for example for Windows Store, Windows Phone, Android and iOS using Microsoft's C# language. Xamarin makes also possible to share code between different mobile platforms.

Microsoft Azure Mobile Service provides also a REST API. With the Mobile Service REST API it is possible to do data table related operations like insert, update, delete and query. With the Mobile Service REST API it is also possible to do client-directed or ser-

vice-directed login operation. Client-directed login operation means that the mobile application will provide an existing authentication token to Mobile Service. The existing authentication token might be retrieved for example by using the identity providers own SDK. Service-directed login operation means that the Mobile Service will initiate authentication using a specific identity provider.

Microsoft Azure Mobile Service documentation contains tutorials and examples for following platforms: Windows Store (C#, JavaScript and C++), Windows Phone, iOS, Android, HTML, Xamarin (iOS and Android) and Sencha. It is possible to find also getting started tutorial for PhoneGap from the get started page of a Mobile Service.

4.3.8 Pricing Model

Microsoft Azure Mobile Service offers two pricing plans: “pay-as-you-go plan” and “6 or 12-month plan”. In the FIGURE 37 Microsoft presents details for the “pay-as-you-go plan” (Microsoft Azure: Mobile Services Pricing Details). In the FIGURE 38 Microsoft presents details for the “6 or 12-month plan” (Microsoft Azure: Mobile Services Pricing Details). Both previously mentioned pricing plans contains three categories inside the plan: free, basic and standard. By comparing “pay-as-you-go plan” and “6 or 12-month plan” it is possible to see that the main difference between the plans is the price. By using the “6 or 12-month plan” it is possible to save 20 – 32 % compared to “pay-as-you-go plan”. From the pricing plans it is possible to see that for example the free category includes: 500 000 API calls, 500 active devices, no scaling, limited scheduled jobs (one job and one execution per hour) and 20 MB SQL Database (for the first twelve months). By changing the category from free to basic it is possible to: get 1 500 000 API calls / unit, unlimited amount of active devices, scaling up to 6 units, 10 scheduled jobs and 50 000 scheduled job executions. By changing the category from basic to standard it is possible to: get 15 000 000 API calls / unit, scaling up to 10 units and 500 000 scheduled job executions.

Pricing Details

Euro (€)

Windows Azure Mobile Services is offered in three tiers: **Free**, **Basic** and **Standard**.

	FREE ¹	BASIC	STANDARD
Price ²	Free (up to 10 services / month)	€19 / month per unit	€149 / month per unit
API Calls ²	500K	1.5M per unit	15M per unit
Active Devices ³	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Up to 10 units
Scheduled jobs ⁴	Limited	Included	Included
SQL Database ⁵ (required)	20 MB included, Standard rates apply for additional capacity	20 MB included, Standard rates apply for additional capacity	20 MB included, Standard rates apply for additional capacity

¹ Quotas for the Mobile Services Free tier apply at the subscription level.
² Billing and the quotas for API calls are prorated daily.
³ Active devices refers to the daily number of physical devices and emulators that make at least one call to or receive a push notification from your mobile service.
⁴ The scheduled jobs feature is currently in preview. The Free tier is limited to one job and one execution per hour. Basic and Standard tiers include 50k and 500k job executions, respectively, and can accommodate up to 10 jobs.
⁵ One 20MB Azure SQL Database is available per subscription for the first twelve months of use; standard rates apply thereafter.

Data transfers are included in Mobile Services prices and are subject to the [Use Terms](#).

FIGURE 37. Microsoft Azure Mobile Service pricing details for the pay-as-you-go plan

Pricing Details

Euro (€)

Windows Azure Mobile Services is offered in three tiers: Free, Basic and Standard.

	FREE ¹	BASIC	STANDARD
Price ²	Free (up to 10 services / month)	€13 - €15 / month per unit	€101 - €119 / month per unit
API Calls ²	500K	1.5M per unit	15M per unit
Active Devices ³	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Up to 10 units
Scheduled jobs ⁴	Limited	Included	Included
SQL Database ⁵ (required)	20 MB included, Standard rates apply for additional capacity	20 MB included, Standard rates apply for additional capacity	20 MB included, Standard rates apply for additional capacity

* Actual price is determined by your committed spend level, commitment duration and whether you choose to pay monthly or pre-pay.
¹ Quotas for the Mobile Services Free tier apply at the subscription level.
² Billing and the quotas for API calls are prorated daily.
³ Active devices refers to the daily number of physical devices and emulators that make at least one call to or receive a push notification from your mobile service.
⁴ The scheduled jobs feature is currently in preview. The Free tier is limited to one job and one execution per hour. Basic and Standard tiers include 50k and 500k job executions, respectively, and can accommodate up to 10 jobs.
⁵ One 20MB Azure SQL Database is available per subscription for the first twelve months of use; standard rates apply thereafter.

Data transfers are included in Mobile Services prices and are subject to the [Use Terms](#).

Make a monthly commitment to Windows Azure for 6 months or 12 months and receive a significant discount of 20-32%. The discount you earn is determined by the monthly monetary amount you commit. We offer two options:

- Pay Monthly: Minimum commitment €350/month billed monthly
- Pre-Pay: Pre-Pay for the entire term and get an additional discount

For details on discount pricing, please see the 6-Month and 12-Month plan [details](#).

FIGURE 38. Microsoft Azure Mobile Service pricing details for the 6 or 12-month plan

Microsoft Azure provides pricing calculator which can be used to estimate monthly cost for a certain kind of setup in the Microsoft Azure (Microsoft Azure: Pricing Calculator). In the calculator it is possible to calculate monthly cost for example for a Mobile Service or monthly cost estimate for all services that will be used from the Microsoft Azure. From the Microsoft Azure pricing calculator it is possible to see that different services of Microsoft Azure have different cost factors. For example Mobile Service has following cost factors: selected pricing category (free, basic or standard), selected pricing plan (pay-as-you-go, 6-month plan or 12-month plan), required database size, use of bandwidth, use of notification hubs and use of Microsoft Azure support (see FIGURE 39).

Microsoft Azure

SALES: 0800-552-049 | MY ACCOUNT | PORTAL | Search

Features Pricing Documentation Downloads Add-ons Community Support

FREE TRIAL

No upfront costs. Pay only for what you use.

web sites virtual machines **mobile services** cloud services data management full calculator

Mobile Services

Free Basic Standard

Run up to 10 Mobile Services with a 20MB SQL database for free. Seamlessly upgrade as your traffic grows. (up to 500K API calls)

Free

SQL Database (Web and Business)

Database Size: 1GB €7.44

Bandwidth

United States + Europe egress: 0GB €0.00

Notification Hubs

Free Basic Standard

Send 100,000 Pushes per month to up to 500 active devices.

Free

Support

Included: €0.00

€7.44/mo

Currency: Euro (€)

Region: US East

Estimated Price

Pay-as-you-go
 6-Month Plans*
 12-Month Plans*

Purchase

pricing quick links

Explore our [Pay-As-You-Go](#), [6 and 12-Month](#), [Member Offers](#), and [Enterprise Agreements](#).
 Learn more about our [pricing](#), [billing](#) and [metering](#) details.
 See how you can pay by [invoice](#) instead of credit card.
 Learn about our [license mobility](#) policies.

FIGURE 39. Microsoft Azure pricing calculator

At the written time of this master's thesis Microsoft Azure offered free one month trial including 150 € to spend on all Microsoft Azure services (Microsoft Azure: Microsoft Azure Free Trial: Try Azure). In the FIGURE 40 is a screen capture of the Microsoft

Azure one month trial offer. Microsoft Azure had also special offers for MSDN subscribers, start-up companies and academic use (Microsoft Azure: Microsoft Azure Free Trial: Try Azure). In the FIGURE 41 is a screen capture of the Microsoft Azure offer for MSDN subscribers, start-up companies and academic use.

Microsoft Azure

SALES: 0800-552-049 | MY ACCOUNT | PORTAL | Search

Features Pricing Documentation Downloads Add-ons Community Support

FREE TRIAL →

Free one month trial

Sign-up for free and get €150 to spend on all Azure services

Try it now →

Or buy now ▶
Frequently Asked Questions ▶
More questions? Call us: 0800-552-049

- ✓ Virtual Machines
- ✓ SQL Databases
- ✓ Web Sites
- ✓ Hadoop
- ✓ Mobile Push
- ✓ Media Streaming
- ✓ Active Directory
- ✓ Everything else...

FIGURE 40. Microsoft Azure one month trial offer

Microsoft Azure

SALES: 0800-552-049 | MY ACCOUNT | PORTAL | Search

Features Pricing Documentation Downloads Add-ons Community Support

FREE TRIAL →

21013-Production	10258	ERNSH	1	7/17/1996 1..	8/14/1996 1..	7/2
fearlesspenguin-DB	10259	CENTC	4	7/18/1996 1..	8/15/1996 1..	7/2
Content-24hours	10260	OTTIK	4	7/19/1996 1..	8/16/1996 1..	7/2
mysqlserverinstance	10261	QUEDE	4	7/19/1996 1..	8/16/1996 1..	7/3
useroicefeedback	10262	RATTC	8	7/22/1996 1..	8/19/1996 1..	7/3
windowsAecXUm2po	10263	ERNSH	9	7/23/1996 1..	8/20/1996 1..	7/3
Storage	10264	FOLKO	6	7/24/1996 1..	8/21/1996 1..	8/2
Virtual Machines	10265	BLONP	2	7/25/1996 1..	8/22/1996 1..	8/1
Web Sites	10266	WARTH	3	7/26/1996 1..	8/23/1996 1..	7/3
	10267	FRANK	4	7/29/1996 1..	8/26/1996 1..	8/6
	10268	GROSR	8	7/30/1996 1..	8/27/1996 1..	8/2
	10269	WHITC	5	7/31/1996 1..	8/14/1996 1..	8/9

MSDN subscriber?
Get up to €1,380 per year of additional Azure services
Activate Now →

Startup?
Join the BizSpark program and get free Azure services
Learn More →

Academic use?
Schools and institutions can receive special benefits
Contact Us →

FIGURE 41. Microsoft Azure offer for MSDN subscriber, start-up companies and academic use

4.4 IMPLEMENTING AN EXAMPLE CLOUD SOLUTION

In this chapter an example MBaaS cloud solution will be created to Amazon Web Services, Google Cloud Platform and Microsoft Azure. The aim of this chapter is to present concrete example how it is possible to create MBaaS cloud solution to Amazon Web Services, Google Cloud Platform and Windows Azure.

The MBaaS cloud solution that will be created in this chapter will be used by a notes mobile application. The notes mobile application gives for the user of it possibility to create, modify, delete and view notes. The notes mobile application will also make possible for the user of it to register listening changes of a certain note. The notes mobile application is implemented as native application for Android, iOS and Windows Phone.

The MBaaS solution that will be created in this chapter needs to support the notes mobile application so that it is:

- able to store notes,
- able to get notes,
- able to add a note,
- able to delete a note,
- able to update a note,
- able to register listening changes of certain note,
- able to inform registered listeners about changes of a certain note,
- able to authenticate and authorize the notes mobile application and
- usable from Android, iOS and Windows Phone.

4.4.1 Amazon Web Services

In the Amazon Web Services it is required to setup several services in order to create MBaaS solution for the notes mobile application. Amazon Web Services does not provide single service that could be identified as MBaaS.

Amazon Web Services provides several different possibilities for storing notes in the cloud solution. It is possible to use for example Amazon DynamoDB for storing the notes. It is possible to setup Amazon DynamoDB by doing for example following steps:

- 1) Navigate to AWS Management Console.
- 2) Click “DynamoDB” under the Database section (see FIGURE 42).
 - Getting started page for the Amazon DynamoDB will be visible (see FIGURE 43).
- 3) Click “Create Table” action.
 - “Create Table” dialog will be visible (see FIGURE 44).
- 4) Write name for the table.
- 5) Set primary key type to “Hash”.
- 6) Write name for the hash key (for example “Notes ID”)
- 7) Click “Continue” action in the “Create Table” dialog.
 - Add indexes part of the “Create Table” dialog will be visible (see FIGURE 45).
- 8) Click “Continue” action in the “Create Table” dialog.
 - It is not required to add indexes for the Notes table.
 - Provisioned throughput capacity part of the “Create Table” dialog will be visible (see FIGURE 46).
- 9) Click “Continue” action in the “Create Table” dialog.
 - It is not required to change the default values of the provisioned throughput capacity for the Notes table.
 - Throughput Alarms part of the “Create Table” dialog will be visible (see FIGURE 47).
- 10) Remove selection from the “Use Basic Alarms”.
 - It is not required to use the basic alarm in this example.
- 11) Click “Continue” action in the “Create Table” dialog.
 - Summary part of the “Create Table” dialog will be visible (see FIGURE 48).
- 12) Click “Create” action in the “Create Table” dialog.
 - Amazon Web Services starts creating the table.
 - Amazon DynamoDB tables will be visible (see FIGURE 49).
 - After the table is created, the status of the table changes from “creating” to “active”.
 - Amazon DynamoDB setup is now completed.

The screenshot shows the AWS Management Console interface. At the top, there are navigation elements: 'Services', 'Edit', and user information 'Anssi Soinu', 'Oregon', and 'Help'. The main content area is titled 'Amazon Web Services' and is organized into several columns of service cards. The 'Database' column is circled in red and contains the following services: **DynamoDB** (Predictable and Scalable NoSQL Data Store), **ElasticCache** (In-Memory Cache), **RDS** (Managed Relational Database Service), and **Redshift** (Managed Petabyte-Scale Data Warehouse Service). Other columns include 'Compute & Networking', 'Storage & Content Delivery', 'Deployment & Management', 'Analytics', and 'App Services'. On the right side, there are sections for 'Additional Resources' (Getting Started, Trusted Advisor), 'Service Health' (All services operating normally), and 'Set Start Page' (Console Home). At the bottom right, there is a 'AWS Marketplace' banner.

FIGURE 42. AWS Management Console where DynamoDB is highlighted

The screenshot shows the 'Amazon DynamoDB Getting Started' page. The main content area has a yellow background and contains the following elements:

- A description: "Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability. [Learn More](#) about Amazon DynamoDB."
- A call to action: "To start using Amazon DynamoDB, create a table" with a **Create Table** button.
- A note: "Note: Your table will be created in the EU West (Ireland) region."
- A section titled "How do I create a table?" with a three-step process:
 - 1 Pick Primary Key. (Learn More)
 - 2 Set Provisioned Throughput. (Learn More)
 - 3 Create your table with alarms. (Learn More)
- An "Additional Resources" section with links to:
 - Getting Started Guide
 - FAQ
 - Release Notes
 - Developer Guide
 - Forums
 - Report an issue
- A "Watch the video" section with the text "Learn more about how to setup and use DynamoDB databases." and a video player showing a video titled "Amazon DynamoDB".

FIGURE 43. Getting started page of the Amazon DynamoDB

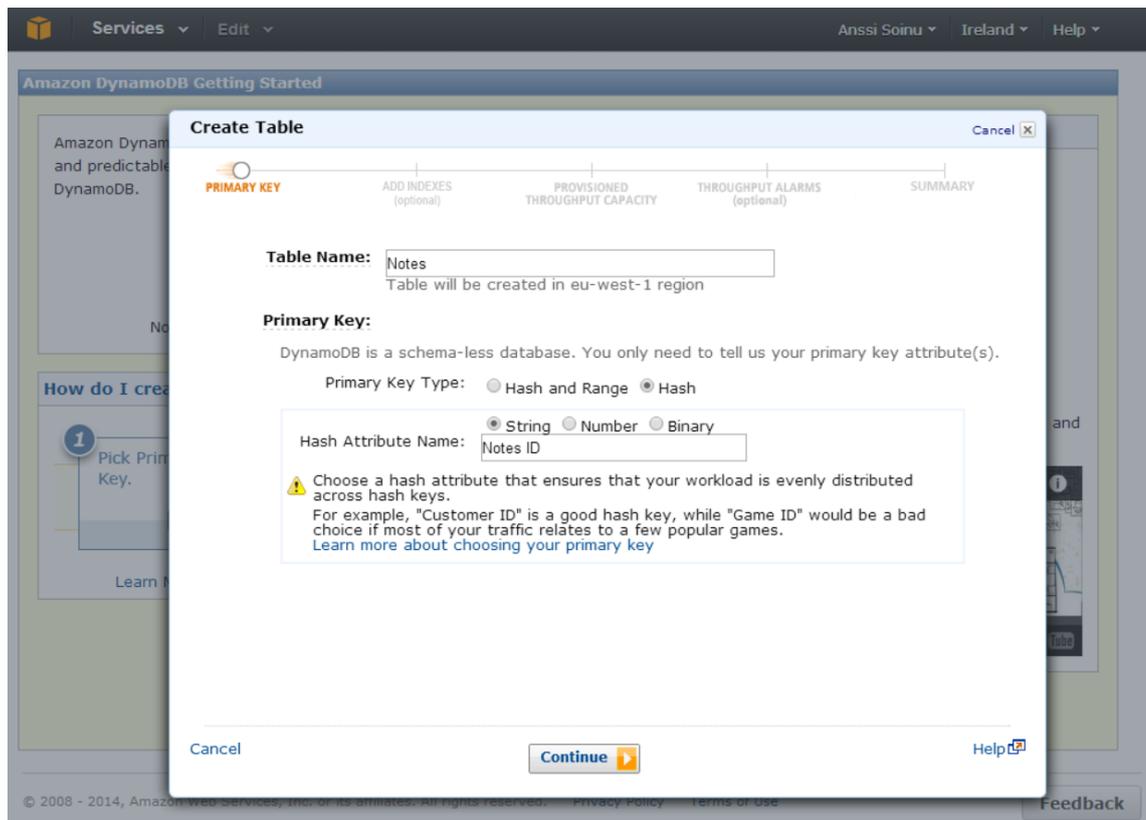


FIGURE 44. “Create Table” dialog in the AWS Management Console

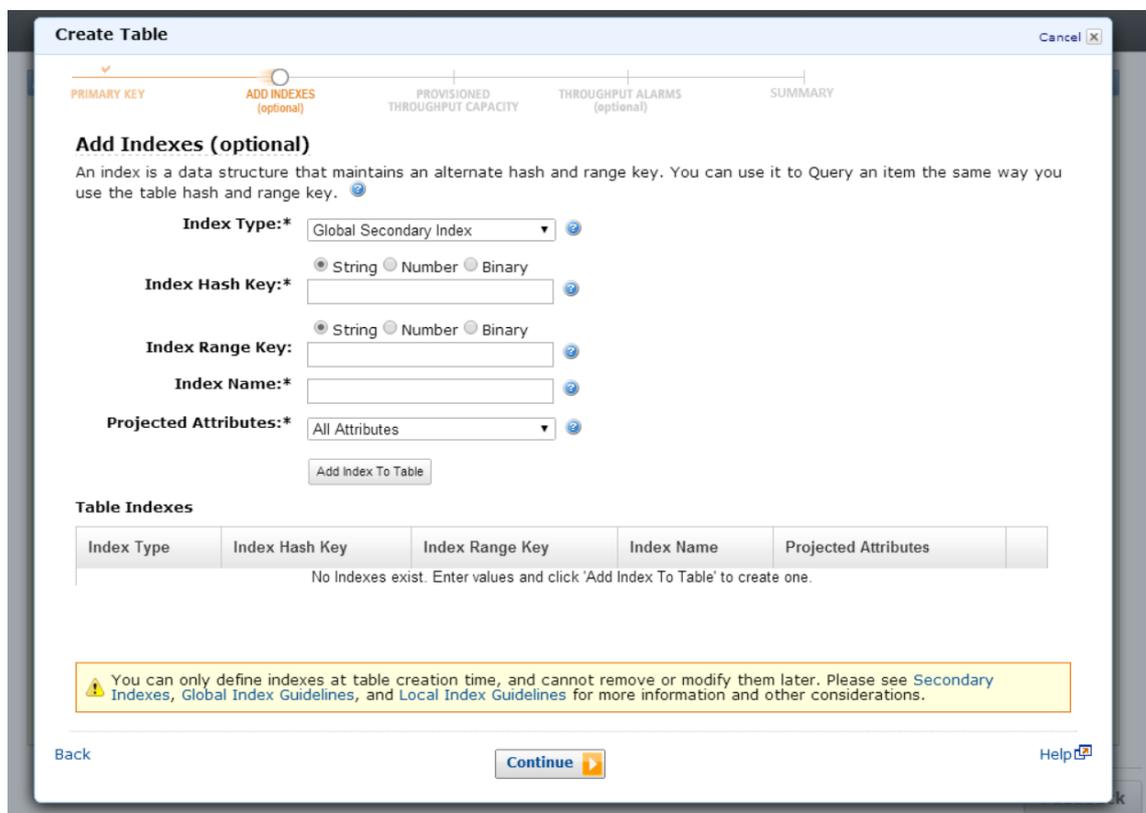


FIGURE 45. Add indexes part of the “Create Table” dialog in the AWS Management Console

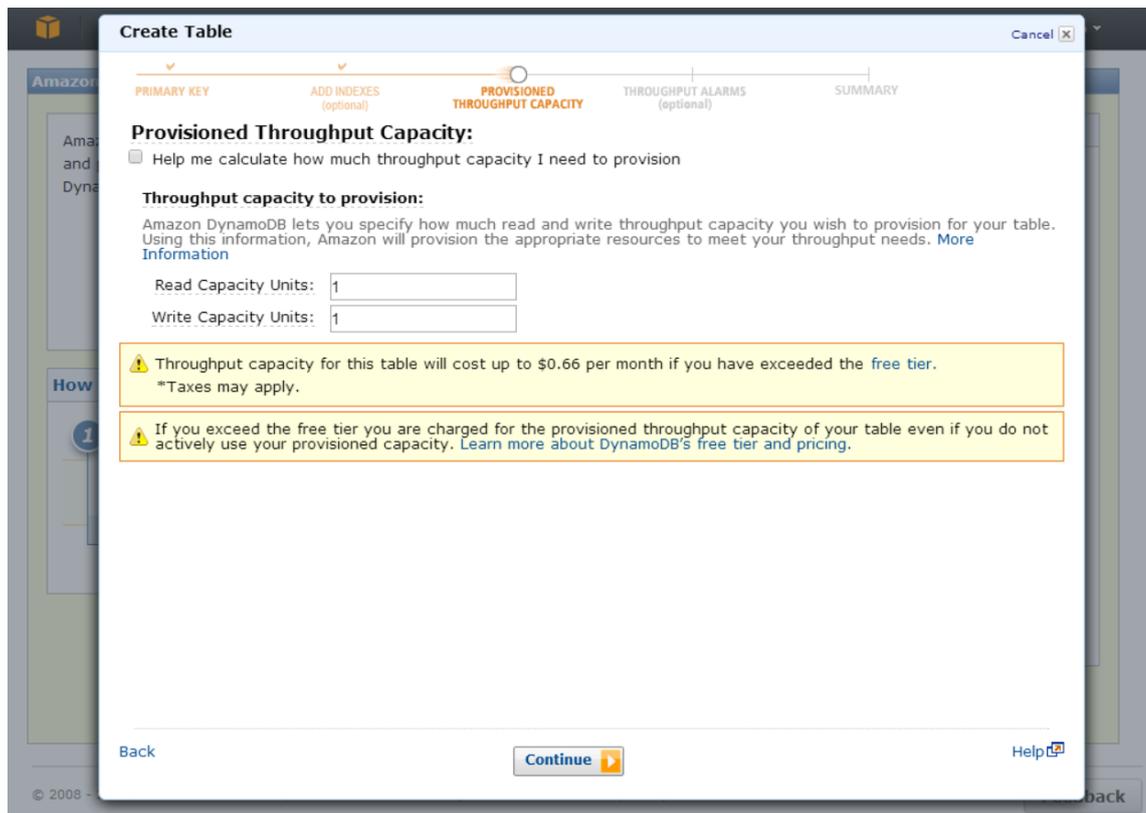


FIGURE 46. Provisioned throughput capacity part of the “Create Table” dialog in the AWS Management Console

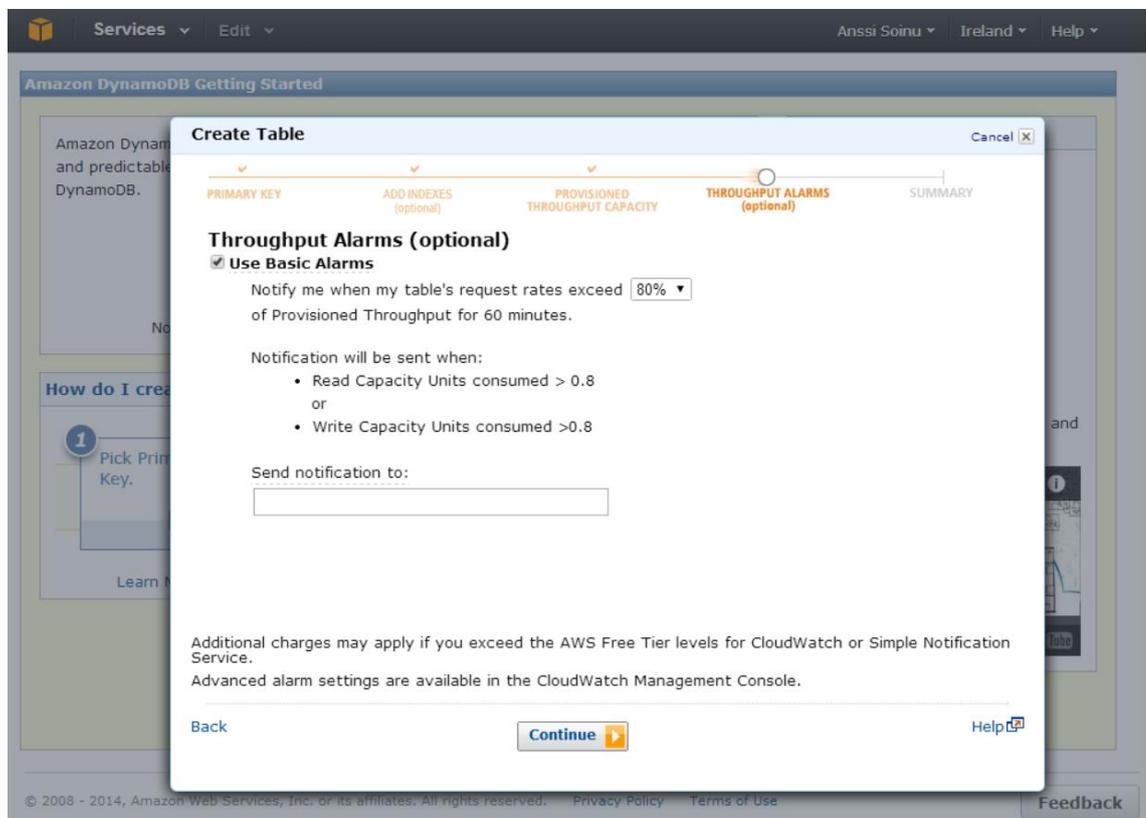


FIGURE 47. Throughput Alarms part of the “Create Table” dialog in the AWS Management Console

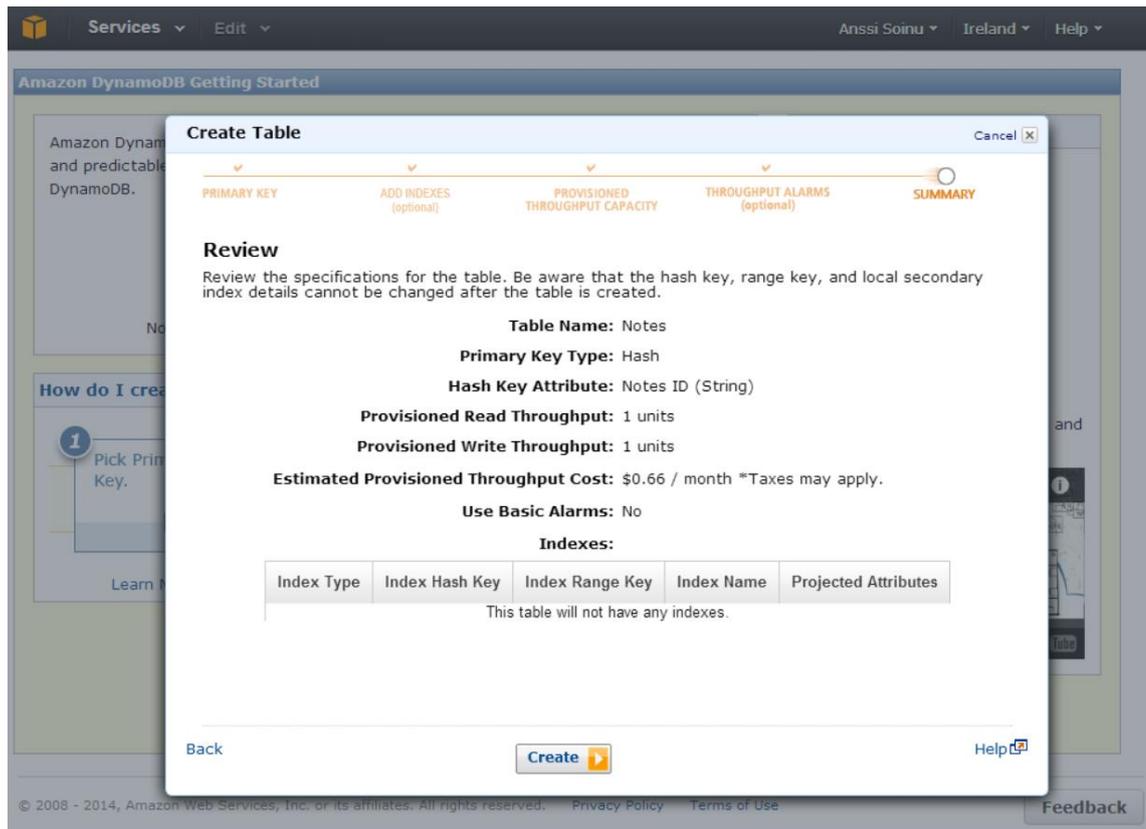


FIGURE 48. Summary part of the “Create Table” dialog in the AWS Management Console

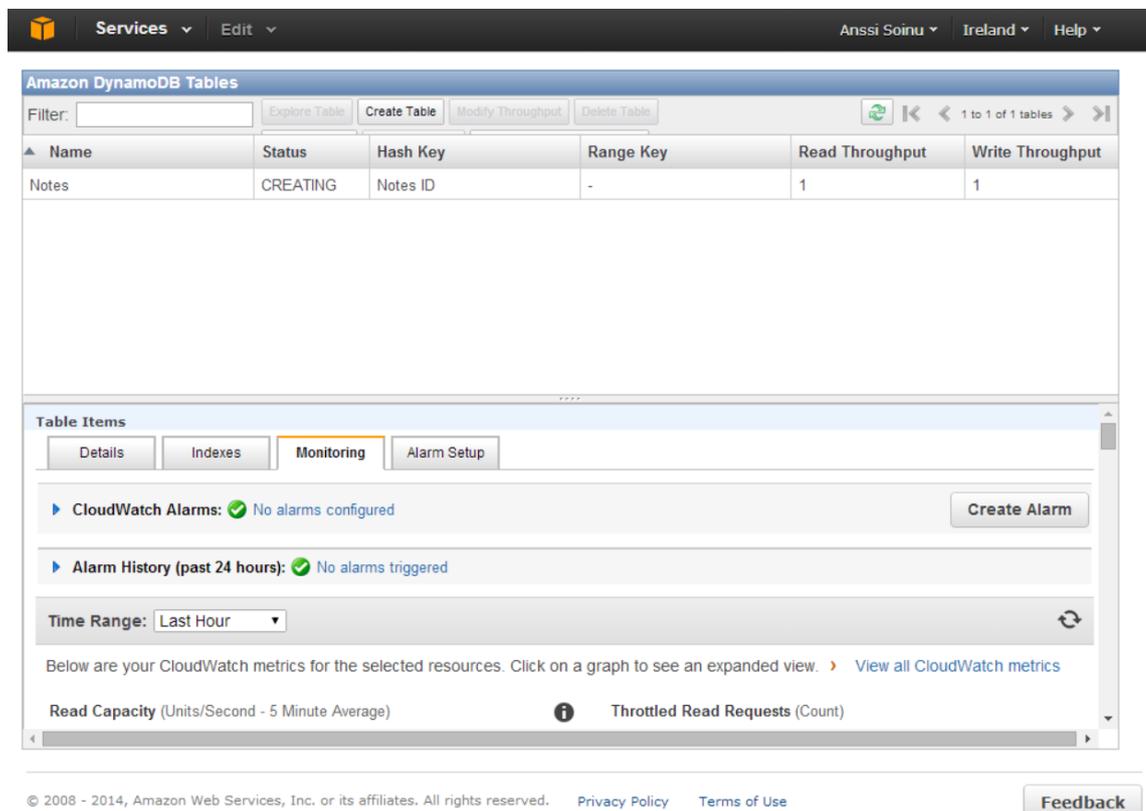


FIGURE 49. Amazon DynamoDB Tables in the AWS Management Console

AWS SDK makes possible to connect to an existing Amazon DynamoDB table from the mobile application. By connecting to the notes Amazon DynamoDB table from the notes mobile application it is possible to for example view, create, update or delete notes. How in practice the connection can be made and how Amazon DynamoDB table is used, it is required to use mobile platform specific code. Amazon Web Services provides own AWS SDK for iOS, Android and Windows Phone. In the appendix 1 it is possible to see an example code how AWS SDK can be used in the Windows Phone notes mobile application to connect to notes Amazon DynamoDB table.

Amazon SNS can be used to send note changed messages to notes mobile application. The problem with the Amazon SNS is that it does not support push notifications to Windows Phone mobile applications. The Amazon SNS supports sending of note changed messages to Android and iOS mobile applications. It is possible to setup Amazon SNS for the notes mobile application by doing for example following steps:

- 1) Navigate to AWS Management Console.
- 2) Click “SNS” under the App Services section (see FIGURE 50).
→ Getting started page of the Amazon SNS will be visible (see FIGURE 51).
- 3) Click “Add a New App” action.
→ “Add your application to Amazon SNS” dialog will be visible (see FIGURE 52).
- 4) Write name for the app (for example “NotesAndroid”).
- 5) Select push platform as “Google Cloud Messaging (GCM)”.
- 6) Provide the API Key.
- 7) Click “Add new app” action.
→ Amazon SNS setup for the Android notes mobile is now completed.
→ It is possible to see the created application in the Amazon SNS (see FIGURE 54).
- 8) Repeat the steps 3 – 7 again in order to make possible to send note changed messages for the notes iOS mobile application.
→ Select push platform as “Apple Push Notification Service (APNS)”
→ Provide APNS Provider certificate.
→ It is now possible to send note changed messages for the notes Android and iOS mobile application.

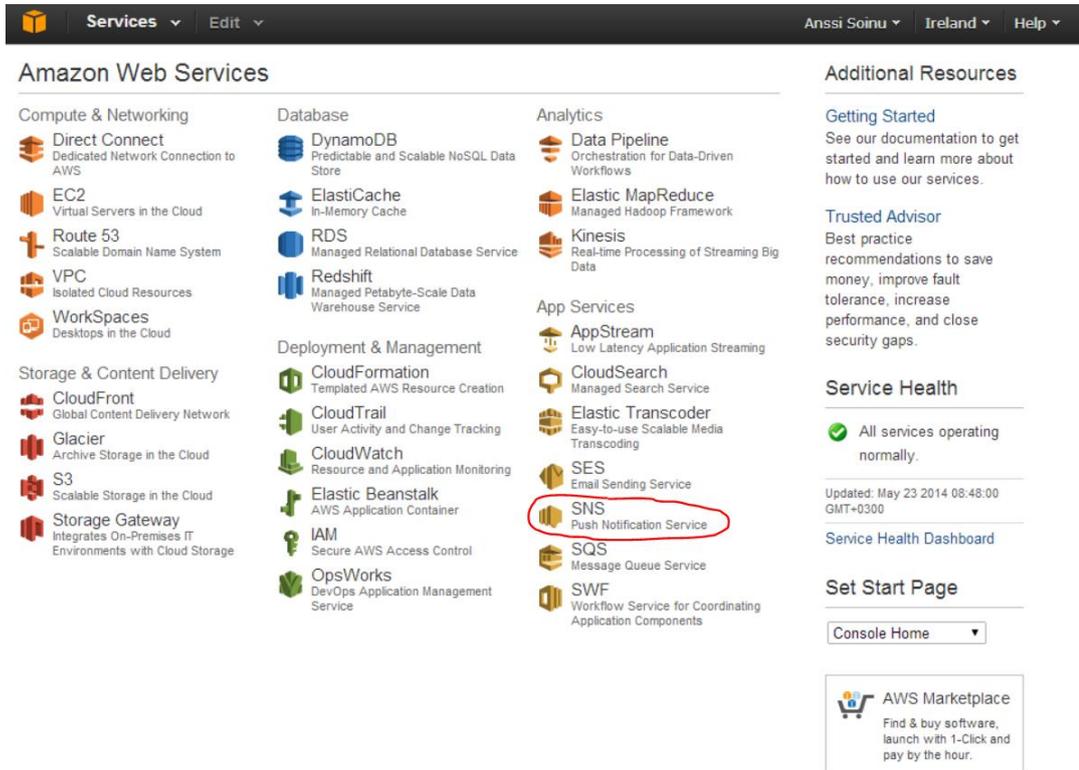


FIGURE 50. AWS Management Console where Amazon SNS is highlighted

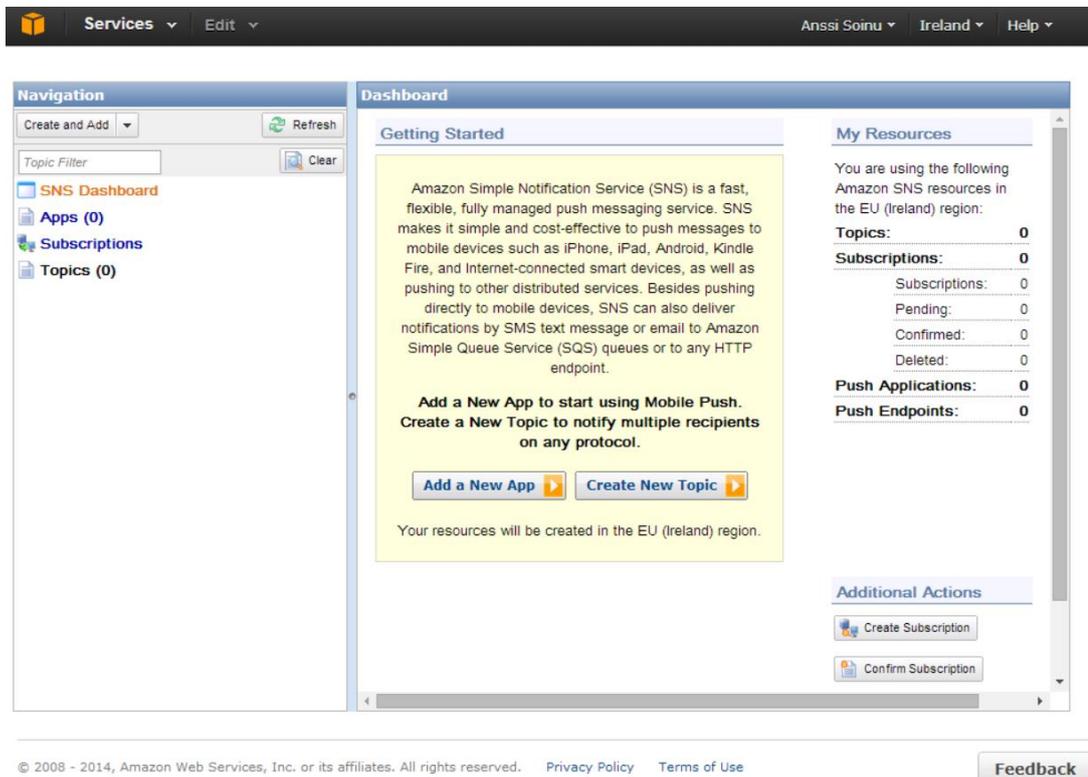


FIGURE 51. Getting started page of the Amazon SNS

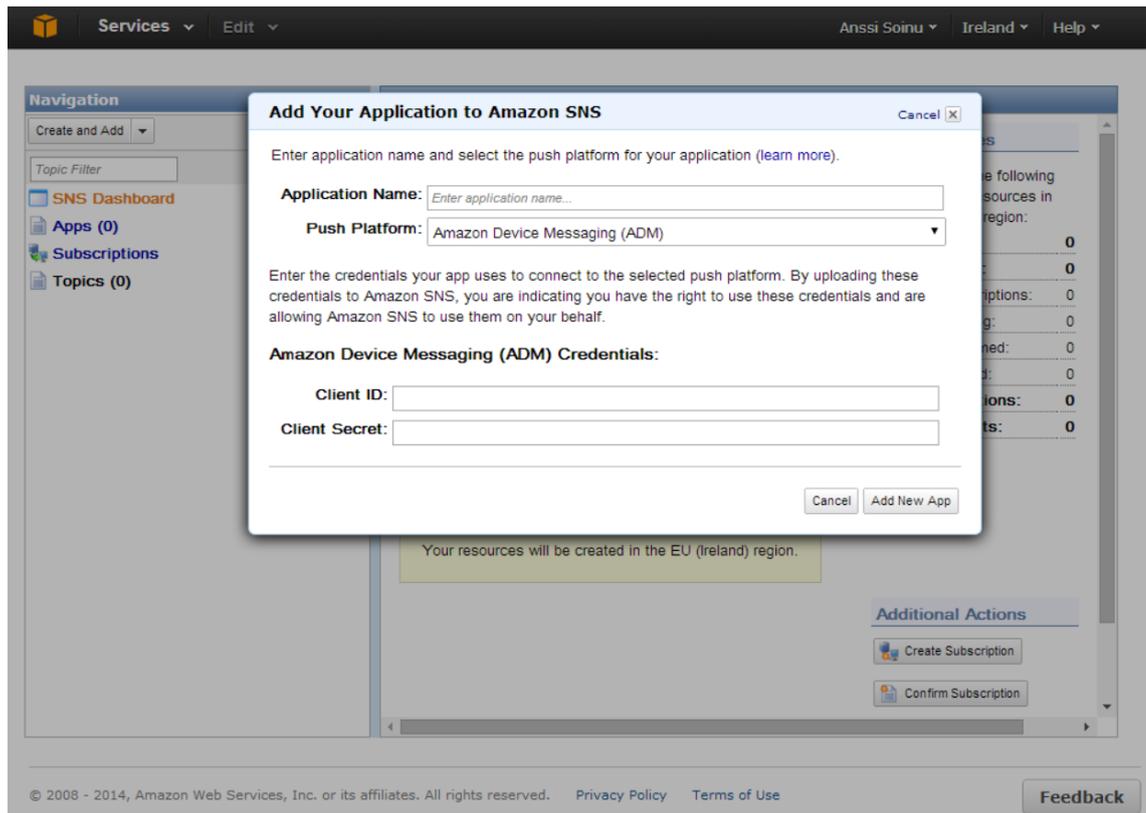


FIGURE 52. “Add your Application to Amazon SNS” dialog in the AWS Management Console

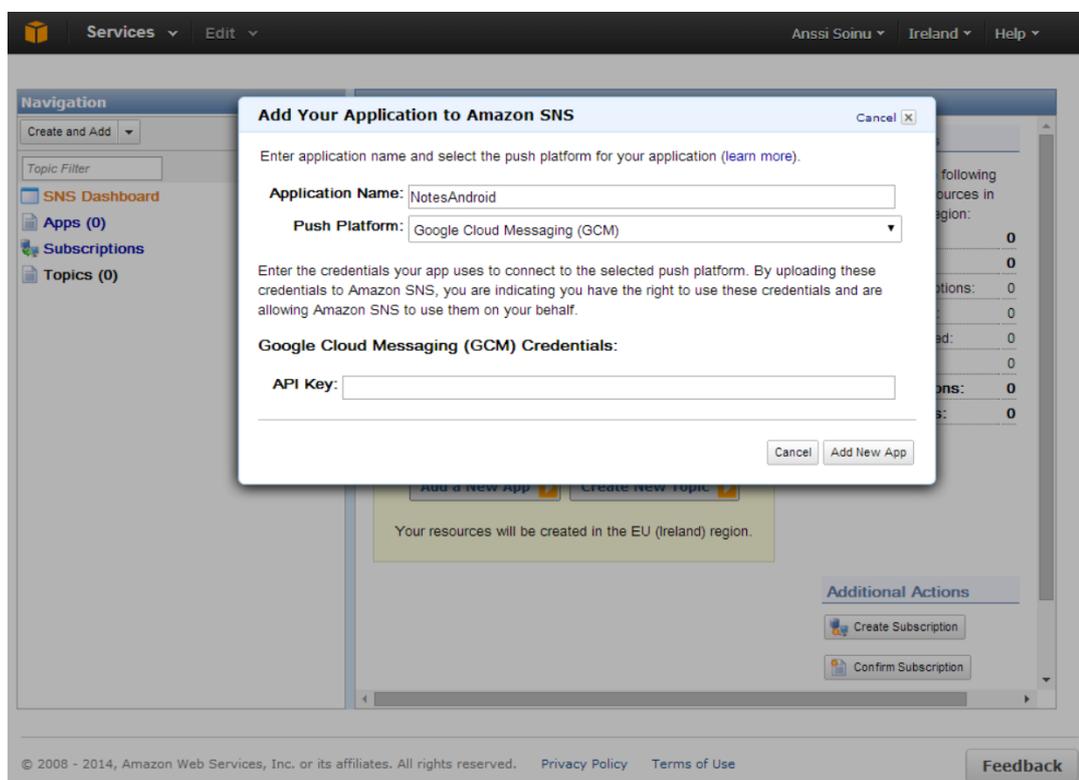


FIGURE 53. “Add Your Application to Amazon SNS” dialog in the AWS Management Console for the notes Android mobile application

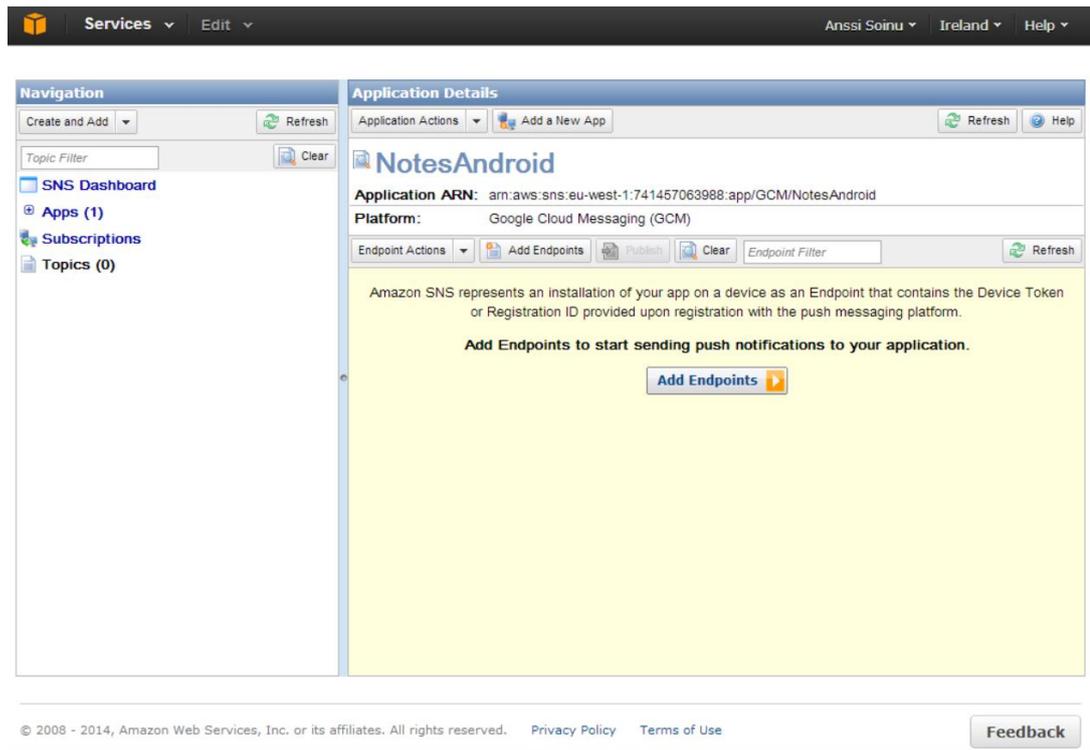


FIGURE 54. Application details for the NotesAndroid application in the Amazon SNS

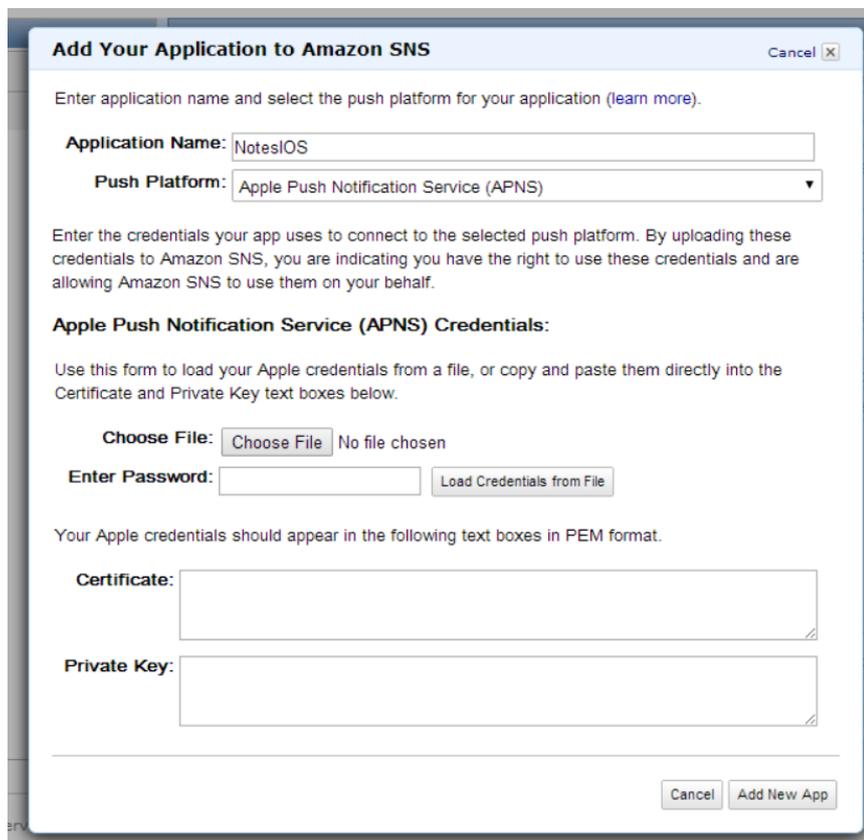


FIGURE 55. “Add Your Application to Amazon SNS” dialog in the AWS Management Console for the notes iOS mobile application

AWS SDK can be used in the notes mobile application to create a topic in the Amazon SNS for a certain note. After the topic is created it is possible to subscribe the notes mobile application to listen messages of it. It is possible to subscribe to listen certain Amazon SNS topic in the notes mobile application by using the AWS SDK. When a certain note is changed it is possible to publish message from the notes mobile application to Amazon SNS by using the note as a topic identifier. Amazon SNS will take care that all subscribed notes mobile applications will receive the note changed message.

Amazon IAM can be used to authenticate and authorize the notes mobile application. In the Amazon IAM exists roles feature which can be used to grant temporary access for example to a table in the Amazon DynamoDB. Temporary access means for example that it is possible to access AWS resources without storing the credentials in the notes mobile application. The process of accessing AWS resources using Amazon IAM roles is as follows:

- 1) Authenticate user in the mobile application using a supported identity provider. Amazon IAM roles feature supports following identity providers: Login with Amazon, Facebook and Google.
- 2) In the mobile application, use the authentication token provided by the identity provider to request a temporary security credentials from the Amazon IAM.
- 3) Use the received security credentials in the mobile application to access AWS resources.

In the AWS it is required to define Amazon IAM role for every supported identity provider. It is required to create for example a role which defines that the Notes DynamoDB table can be accessed by notes mobile application users that have been authenticated with the Google identity provider. It is possible in practice to create previously mentioned role by doing for example following steps:

- 1) Navigate to AWS Management Console.
- 2) Click “IAM” under the Deployment and Management section (see FIGURE 56).
→ Getting started page of the Amazon IAM will be visible (see FIGURE 57).
- 3) Click “Roles” from the navigation of the Amazon IAM.
→ Amazon IAM roles pages will be visible (see FIGURE 58).
- 4) Click “Create New Role” action.
→ “Create Role” dialog will be visible (see FIGURE 59).

- 5) Write name for the role (for example “NotesGoogleRole”).
- 6) Click “Continue” action.
 - Select role type question will be visible (see FIGURE 60).
- 7) Select “Role for Identity Provider Access”.
- 8) Select “Grant access to web identity provider”.
 - Identity provider question will be visible (see FIGURE 61).
- 9) Select “Google” as identity provider.
- 10) Input the unique identifier of your application (created by the identity provider) to “Audience” field.
- 11) Click “Continue” action.
 - Verify “Verify Role Trust” part of the “Create Role” dialog will be visible (see FIGURE 62).
 - It is possible to modify the IAM policy document manually in the “Verify Role Trust” part of the “Create Role” dialog.
- 12) Click “Continue” action.
 - It is not required to modify the given IAM policy document.
 - “Set Permissions” part of the “Create Role” dialog will be visible (see FIGURE 63).
- 13) Select “Policy Generator”.
- 14) Click the “Select” action in the “Policy Generator” section.
 - Policy generator will be visible in the “Create Role” dialog (see FIGURE 64).
- 15) Select “Amazon DynamoDB” as AWS service.
- 16) Select “DeleteItem”, “PutItem”, “GetItem”, “Query” and “UpdateItem” as actions.
- 17) Write the ARN of the Notes DynamoDB into the Amazon Resource Name field.
 - It is possible to find the Amazon Resource Name of the Notes DynamoDB by navigating to details of the Notes DynamoDB.
- 18) Click “Add Statement” action.
- 19) Click “Continue” action.
 - The generated policy document will be visible (see FIGURE 65).
- 20) Click “Continue” action.
 - Review part of the “Create Role” dialog will be visible (see FIGURE 66).
- 21) Click “Create Role” action
 - Role is now created for accessing Notes DynamoDB table.

→ It is possible to see the new role in the Amazon IAM roles page (see FIGURE 67).

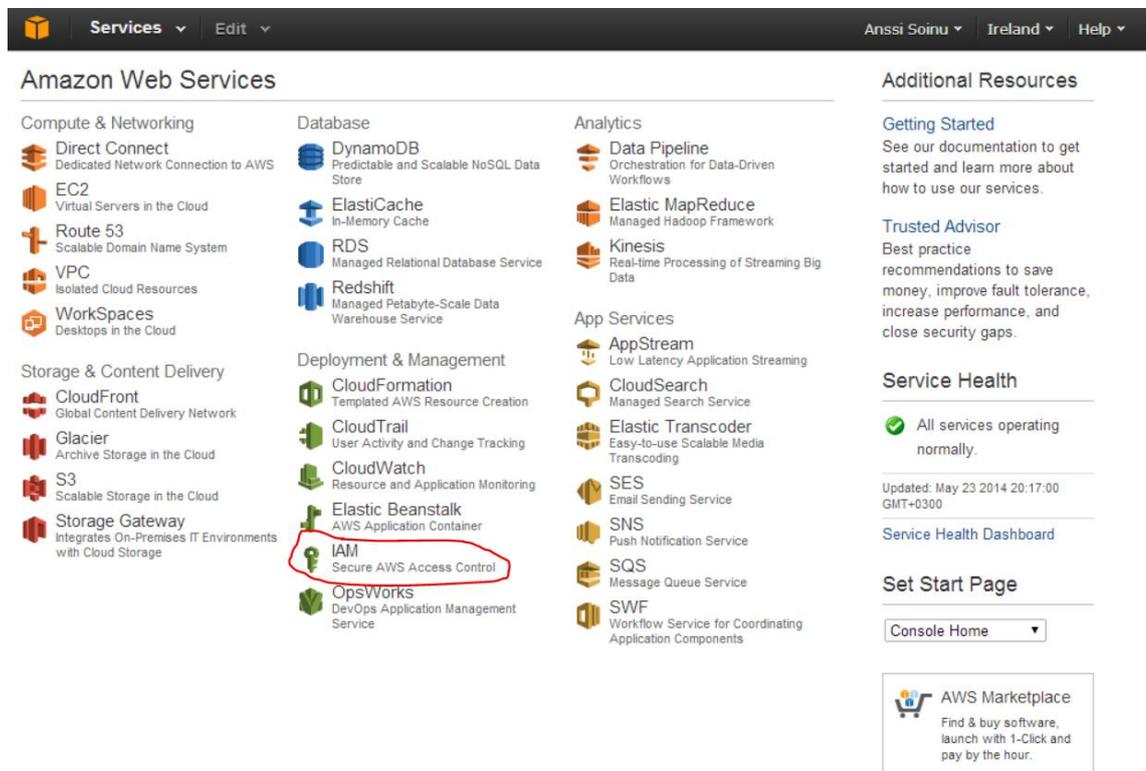


FIGURE 56. AWS Management Console where Amazon IAM is highlighted

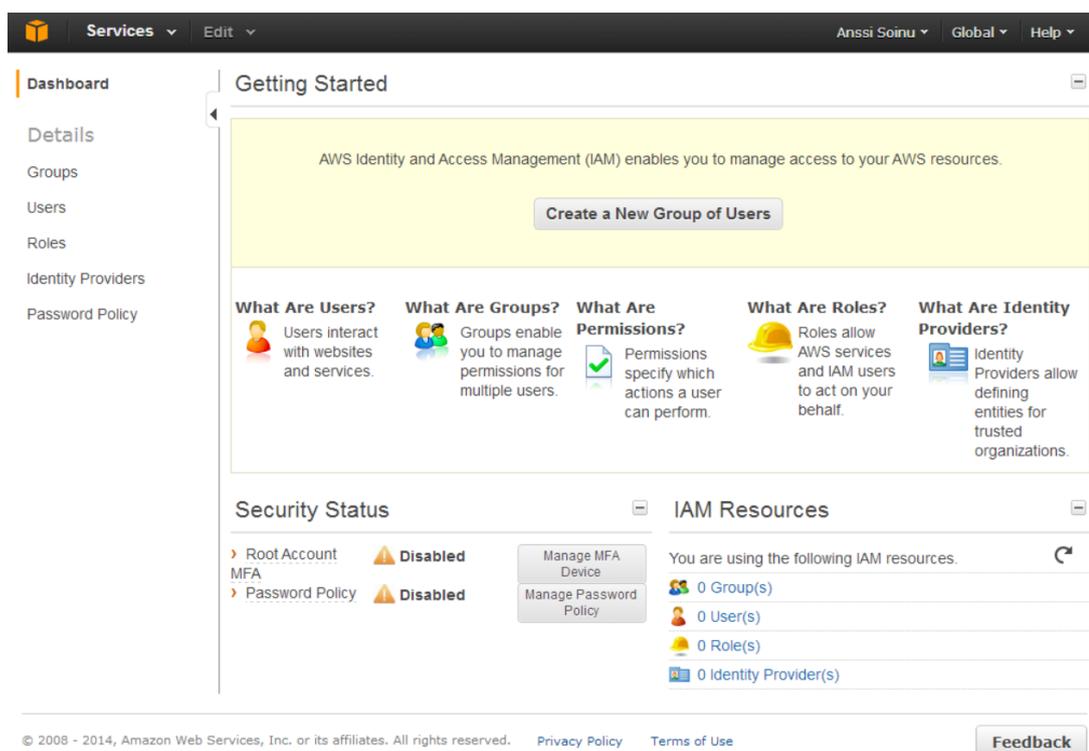


FIGURE 57. Getting started page of the Amazon IAM

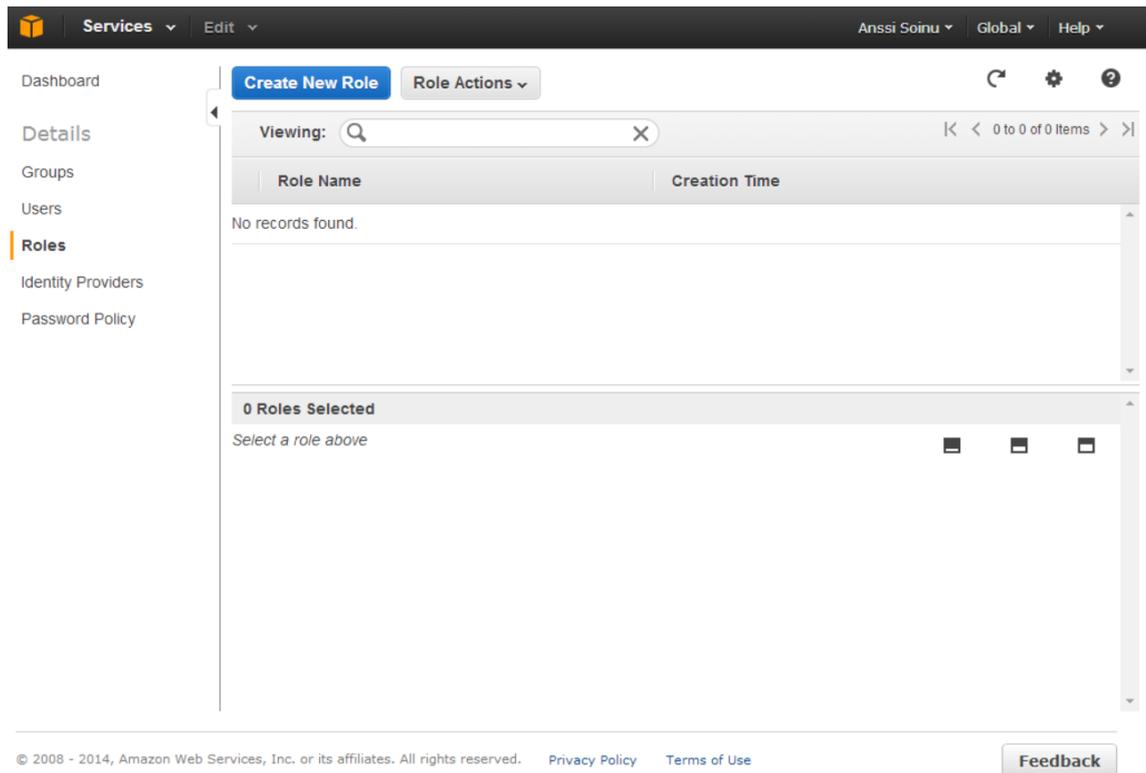


FIGURE 58. Amazon IAM roles page in the AWS Management Console

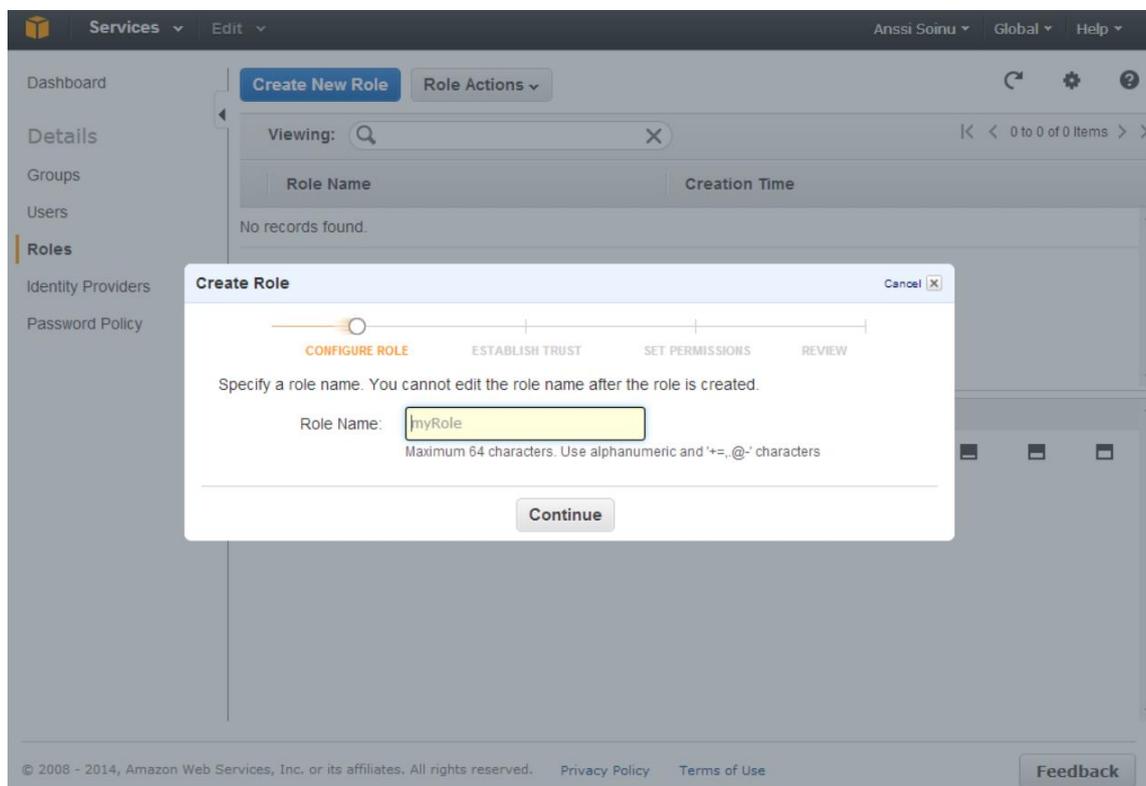


FIGURE 59. “Create Role” dialog in the AWS Management Console

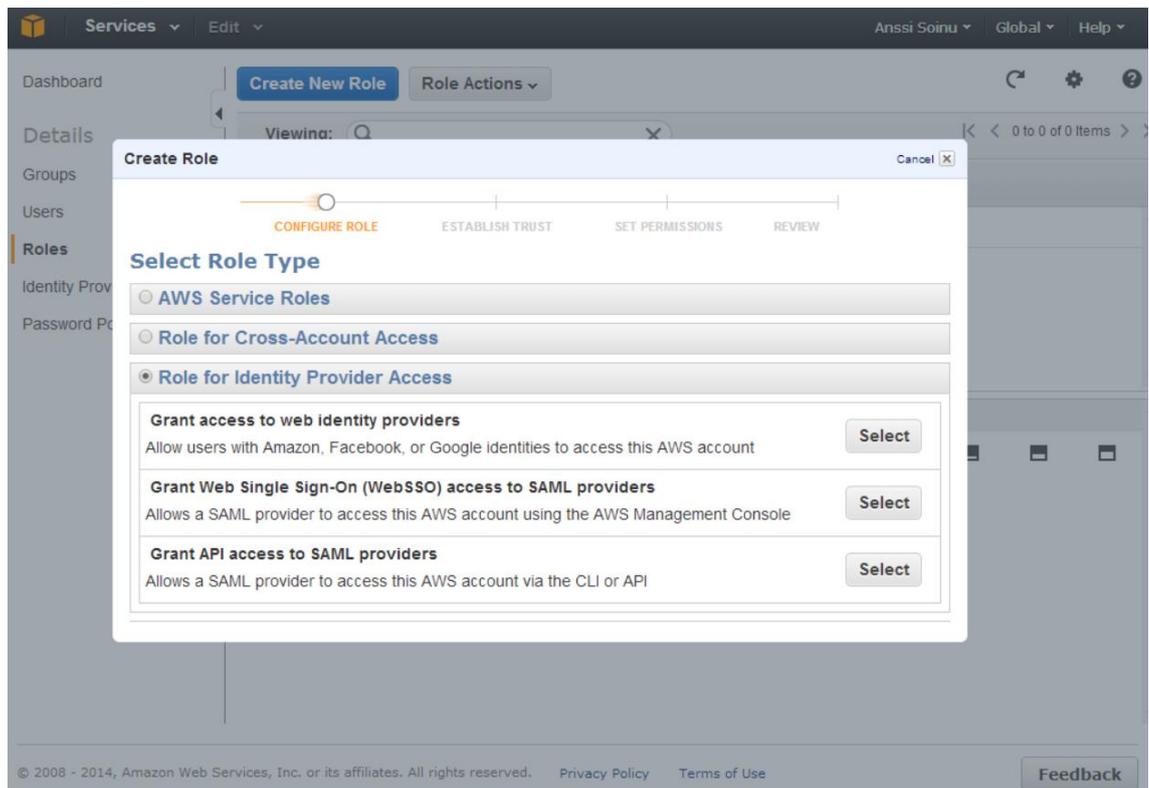


FIGURE 60. “Create Role” dialog with a selection of “Role for Identity Provider Access”

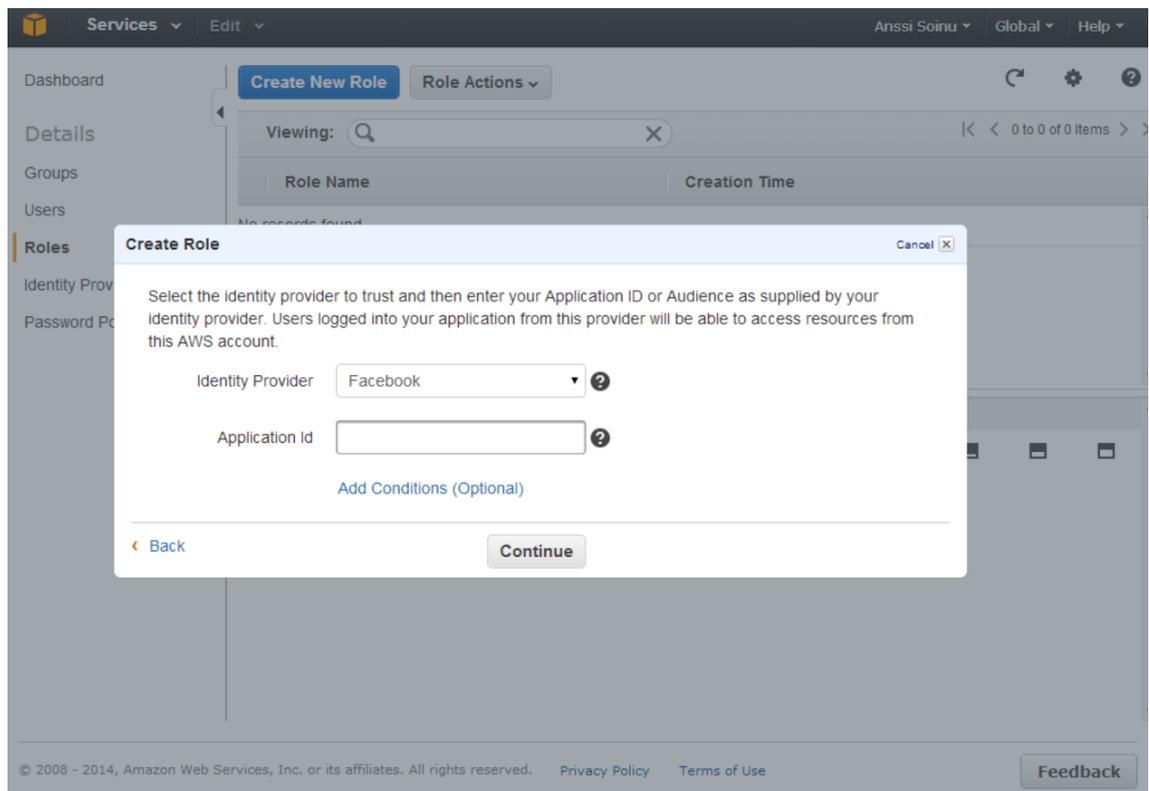


FIGURE 61. Definition of the identity provider in the “Create Role” dialog

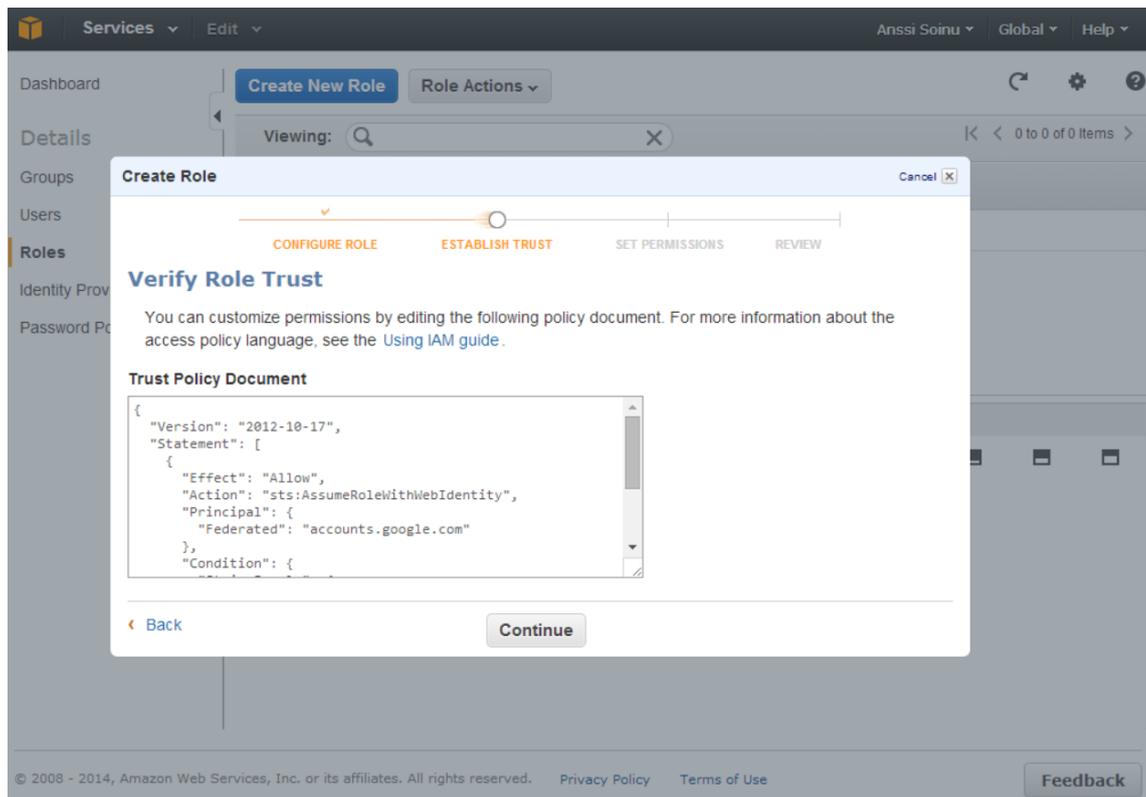


FIGURE 62. “Verify Role Trust” part of the “Create Role” dialog

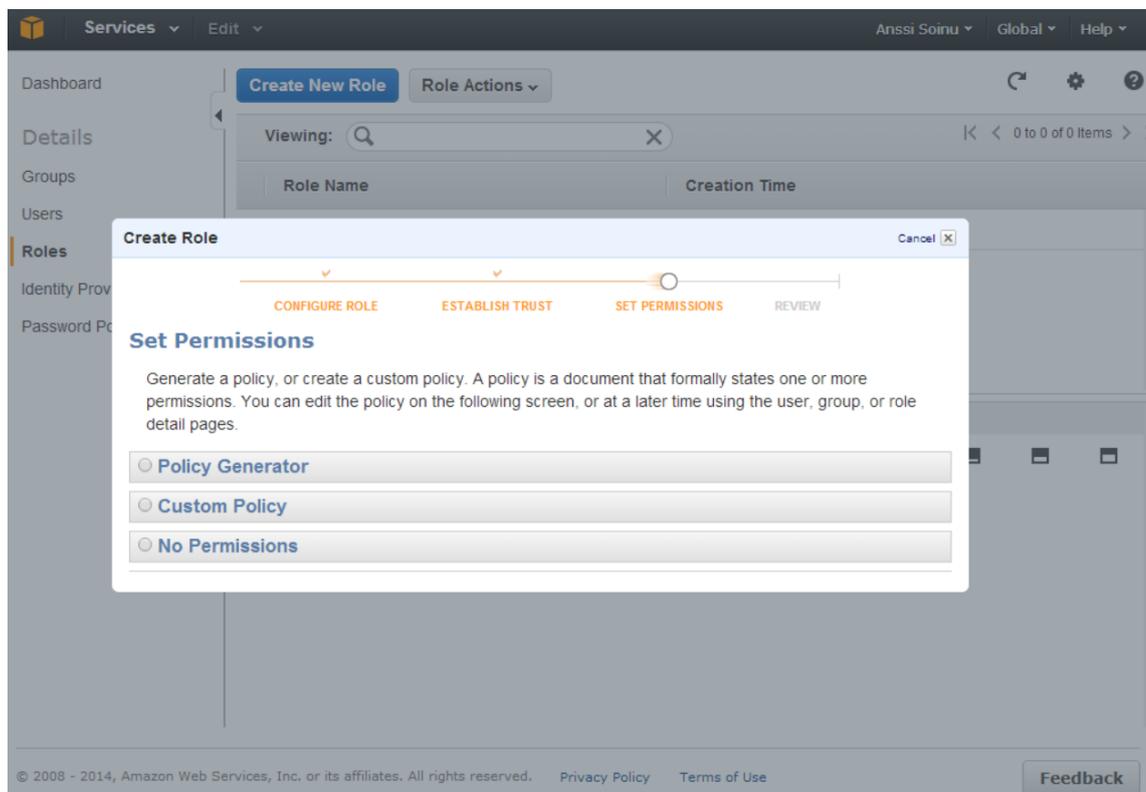


FIGURE 63. “Set Permissions” part of the “Create Role” dialog

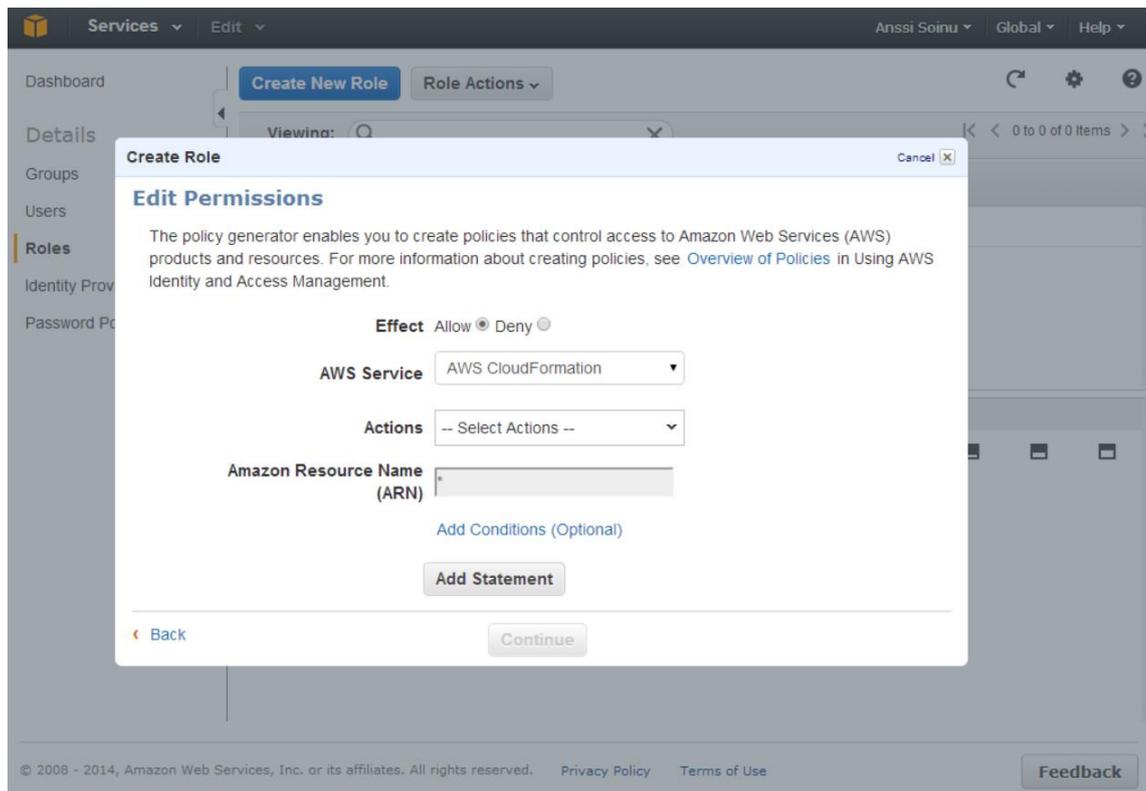


FIGURE 64. Policy generator in the “Create Role” dialog

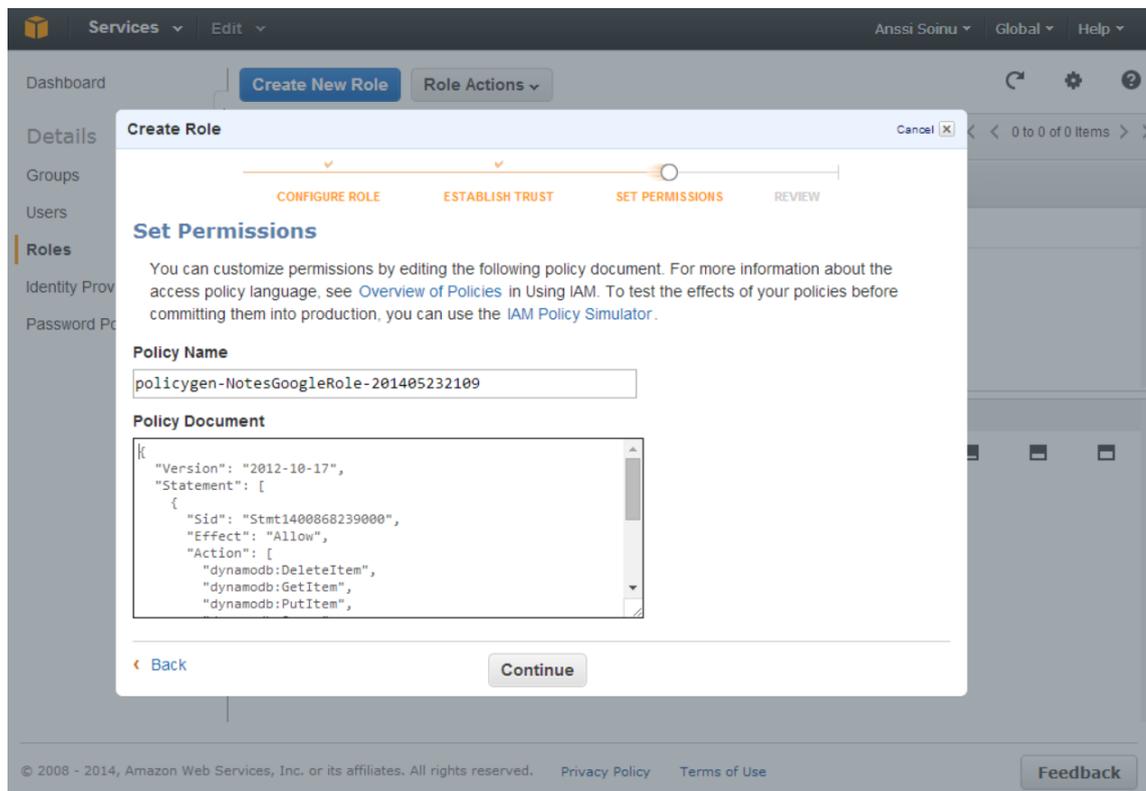


FIGURE 65. Generated policy document in the “Create Role” dialog

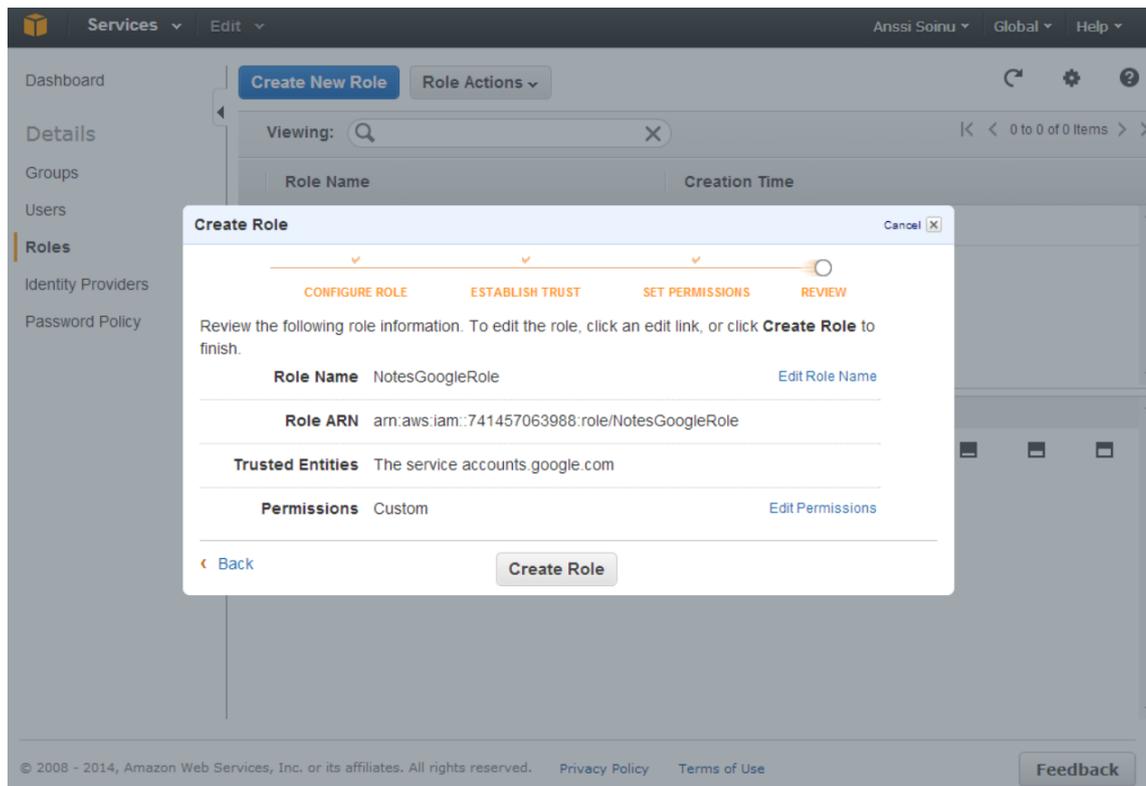


FIGURE 66. Review part of the “Create Role” dialog

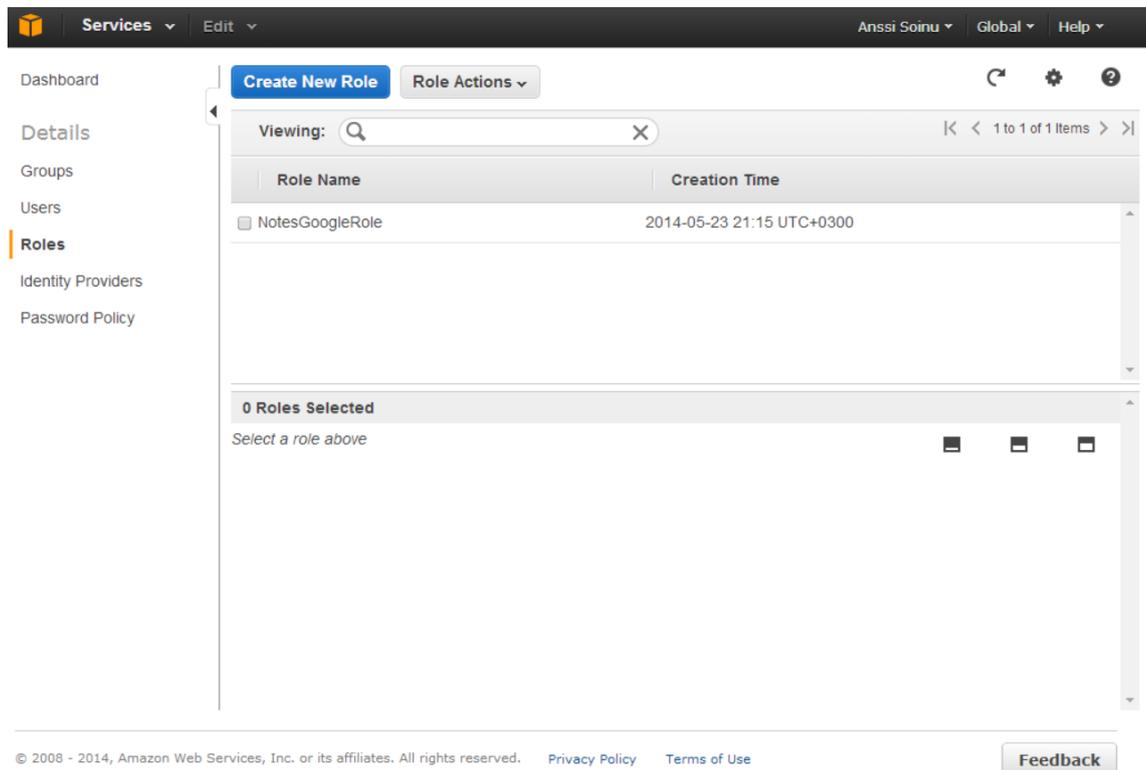


FIGURE 67. NotesGoogleRole in the Amazon IAM Roles

4.4.2 Google Cloud Platform

In order to setup MBaaS solution in the Google Cloud Platform, a new project needs to be created in the Google Cloud Platform. It is possible to create a new project in the Google Cloud Platform by doing for example following steps:

- 1) Browse to Google Developers Console.
→ see FIGURE 68.
- 2) Click “CREATE PROJECT” action.
→ “New Project” dialog will be visible.
- 3) Input name for the project.
- 4) Click the “Create” action.
→ Google Developers Console starts creating a new project.
→ After the project is created, Google Developers Console will navigate to project dashboard.
→ New project is now created in the Google Cloud Platform.

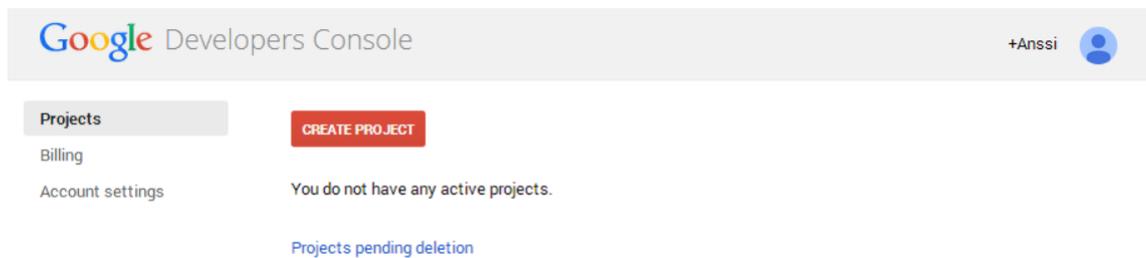


FIGURE 68. Google Developers Console.

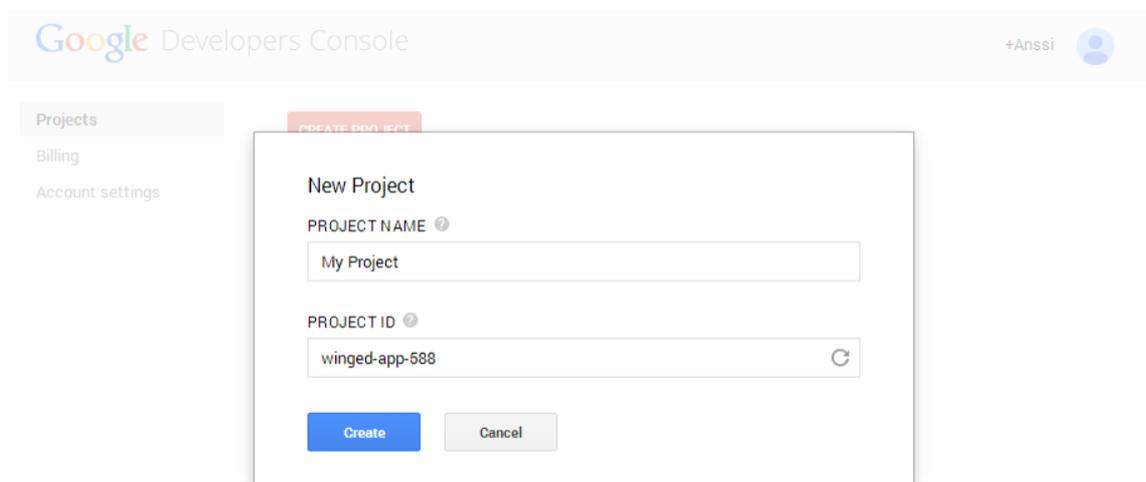


FIGURE 69. “New Project” dialog in the Google Developers Console

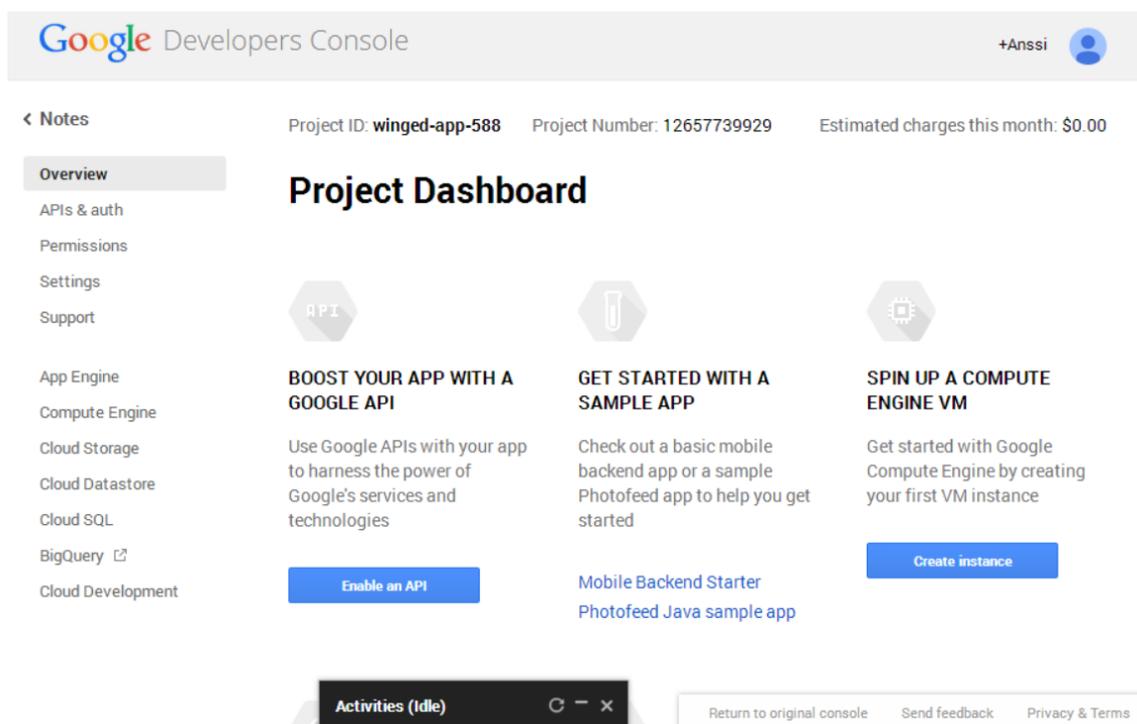


FIGURE 70. Project Dashboard in the Google Developers Console

After the project in the Google Cloud Platform is created, it is possible to use for example Mobile Backend Starter to setup MBaaS solution. The problem with the Mobile Backend Starter is that it will support only Android and iOS mobile applications. Google does not provide supported MBaaS solution that could be used by the Windows Phone mobile applications at the moment. It is possible to setup Mobile Backend Starter by doing for example following steps:

- 1) Navigate to Project Dashboard in the Google Developers Console.
- 2) Click the link “Mobile Backend Starter” in the “GET STARTED WITH A SAMPLE APP” section.
 - Google Developer Console navigates to Mobile Backend Start (see FIGURE 71).
- 3) Click “Deploy” action.
 - Google Developer Console starts deploying the Mobile Backend Starter to the Google Cloud Platform.
 - After the deployment is done, the Google Developer Console will suggest to open the backend for incoming request and download an example mobile client (see FIGURE 72).
 - The setup of the MBaaS solution in the Google Cloud Platform is done.

Google Developers Console +Anssi

< Notes Back

Overview

- APIs & auth
- Permissions
- Settings
- Support
- App Engine
- Compute Engine
- Cloud Storage
- Cloud Datastore
- Cloud SQL
- BigQuery [↗](#)
- Cloud Development

Mobile Backend Starter

Note: This project is designed for Android and iOS developers.

This model project uses Mobile Backend Starter to help you begin building your mobile app with Google Cloud Platform. It provides a general-purpose, ready-to-deploy backend plus client libraries for Android and iOS. ([Learn more](#))

Mobile Backend Starter shows you how to:

- Store objects to the cloud from a mobile device
- Send real-time messages between devices
- Subscribe to changes in data stored on the cloud
- Authenticate users with Google accounts authentication

Deployment steps:

- 1 Deploy the backend

[Deploy](#)

09:46:59 iosclient.mbs: I'm fine. Thank you for asking.

09:42:43 androidclient.mbs: how are you ios?

AT&T 5:48 PM

Sign out Guestbook +

Return to original console Send feedback Privacy & Terms

FIGURE 71. Mobile Backend Starter in the Google Developers Console

Google Developers Console +Anssi

< Notes

Overview

- APIs & auth
- Permissions
- Settings
- Support
- App Engine
- Compute Engine
- Cloud Storage
- Cloud Datastore
- Cloud SQL
- BigQuery [↗](#)
- Cloud Development

libraries for Android and iOS. ([Learn more](#))

Mobile Backend Starter shows you how to:

- Store objects to the cloud from a mobile device
- Send real-time messages between devices
- Subscribe to changes in data stored on the cloud
- Authenticate users with Google accounts authentication

Deployment steps:

- 1 Deploy the backend
 - ✔ The mobile backend is now deployed. [Redeploy](#)
- 2 Open (for development use only) the backend to accept incoming requests via [Settings](#)
- 3 Open a mobile client:

[Android client](#)
[iOS client](#)

Read the [Android](#) or [iOS](#) getting started guides for more information.

09:46:59 iosclient.mbs: I'm fine. Thank you for asking.

09:42:43 androidclient.mbs: how are you ios?

AT&T 5:48 PM

Sign out Guestbook +

I'm fine. Thank you for asking.

07/10/2013 05:46 PM iosclient.mbs@gmail.com >

how are you ios?

07/10/2013 05:42 PM androidclient.mbs@gmail.com >

Return to original console Send feedback Privacy & Terms

FIGURE 72. Mobile Backend Starter deployment completed in the Google Developers Console

Mobile Backend Starter deployment will setup Cloud Datastore in the Google Cloud Platform. It is possible to use the Cloud Datastore for storing the notes in the Google Cloud Platform. The Cloud Datastore is a non-relational database and it will contain entities. It is possible to think note as an entity in the Cloud Datastore. Because the Cloud Datastore is schema-less, it is not required to define properties for the note entity beforehand. The Cloud Datastore will define automatically properties for the entity when the entity is inserted.

It is possible to connect to the Cloud Datastore from the notes mobile application by using the Cloud Backend API of the Mobile Backend Starter. By connecting to the Cloud Datastore from the notes mobile application it is possible to for example view, create, update or delete notes. In the appendix 2 it is possible to see an example code how Cloud Backend API can be used in the Android notes mobile application to connect to Cloud Datastore.

It is possible to use pub/sub messaging feature of the Mobile Backend Starter to inform notes mobile application about changes of a certain note. In the pub/sub messaging notes mobile application needs to subscribe to listen messages having note identifier as the topic identifier. When a note is changed in the notes mobile application, the notes mobile application will publish a cloud message with the identifier of the changed note as topic identifier. Pub/sub messaging feature of the Mobile Backend Starter can be used from the notes mobile application by using the Cloud Backend API of the Mobile Backend Starter.

In order to pub/sub messaging feature of the Mobile Backend Starter to work with the iOS and Android notes mobile application it is required to enable Google Cloud Messaging and iOS Push Notification in the Mobile Backend settings. In order to push notification for iOS mobile application to work, it is also required that the billing is enabled for the Google Cloud Platform project. It is possible to enable Google Cloud Messaging and iOS Push Notification by doing for example following steps:

- 1) Navigate to settings of the mobile backend with an Internet browser.
 - Address of your mobile backend settings is <https://your-project-id.appspot.com/admin/configure.jsp> (replace your-project-id in the address with id of your project).
 - For example notes mobile backend settings can be found from: <https://winged-app-588.appspot.com/admin/configure.jsp>.

- 2) Change Google Cloud Messaging and iOS Push Notification setting to “Enabled”.
 - Google Cloud Messaging API Key setting will be visible.
 - APNS Provider Certificate setting will be visible.
 - see FIGURE 73.
- 3) Provide Google Cloud Messaging API Key.
- 4) Provide the APNS Provider Certificate.
- 5) Click “Save” action.
 - Google Cloud Messaging and iOS Push Notification is now enabled.

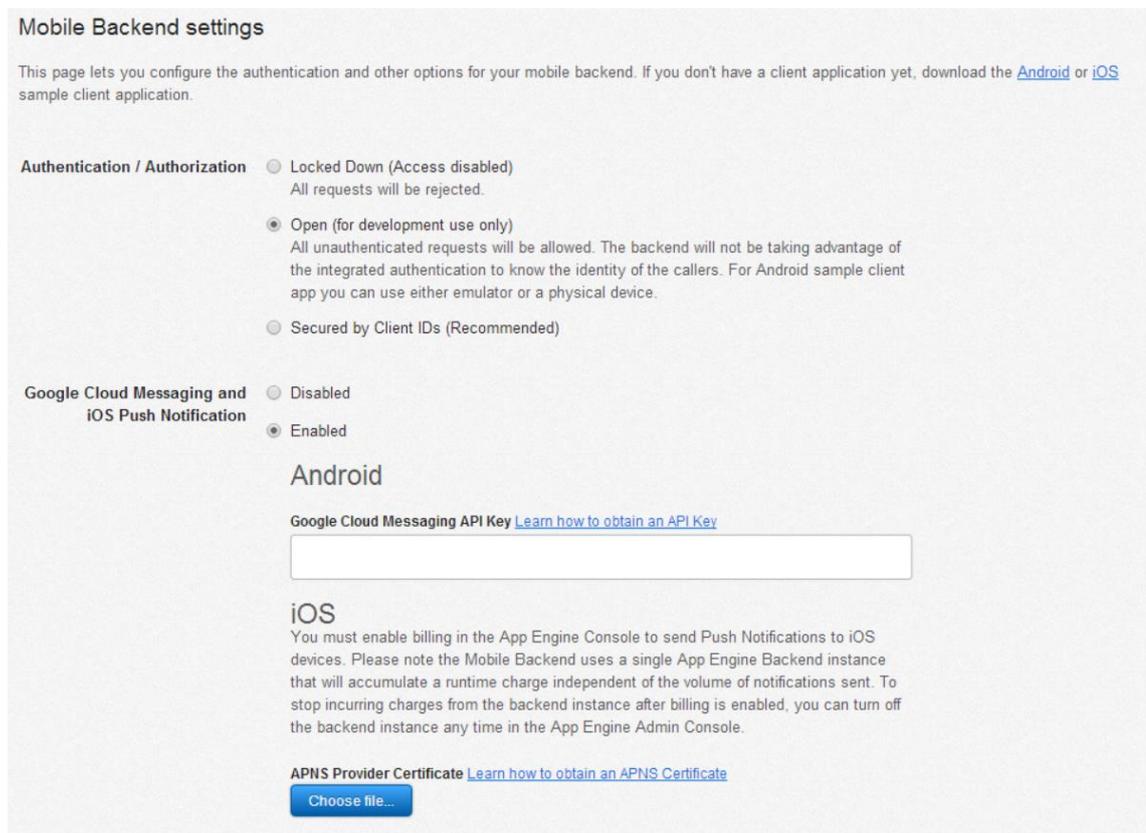


FIGURE 73. Mobile Backend settings where the Google Cloud Messaging and iOS Push Notification is enabled

It is possible to enable billing for the Google Cloud Platform project by doing for example following steps:

- 1) Navigate to Project Dashboard in the Google Developers Console.
- 2) Click “Settings” from the project navigation.
 - Settings of the Notes project will be visible (see FIGURE 74).
- 3) Click “Enable billing” action.
 - Setup billing profile page will be visible.
- 4) Define country or territory where your billing address is located.

- 5) Click “Confirm” action.
→ Page with billing information fields is visible.
- 6) Define your billing information.
- 7) Click “Submit and enable billing” action.
→ Billing is now enabled for a Google Cloud Platform project.

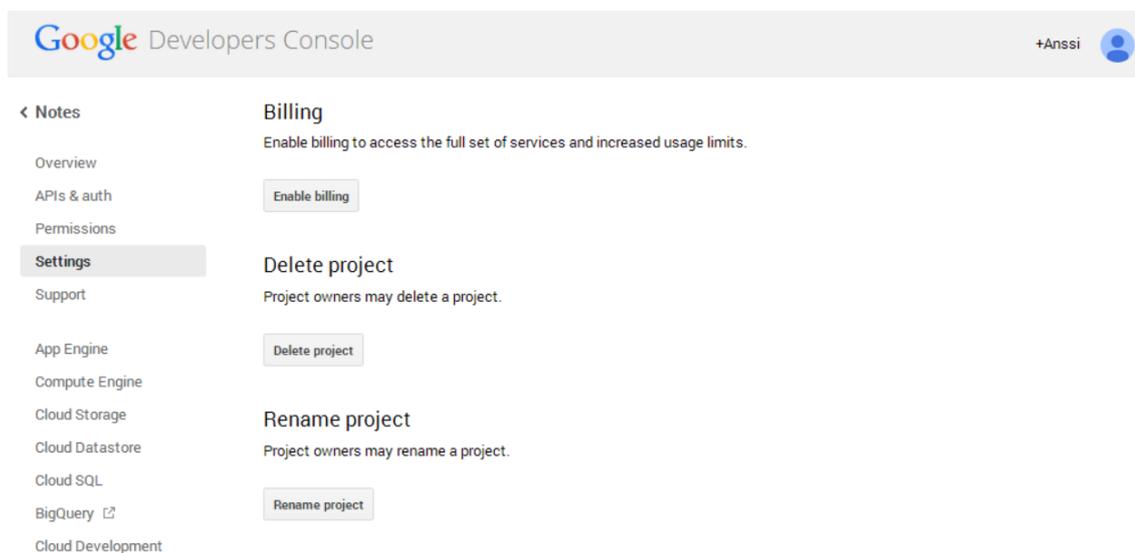


FIGURE 74. Settings of the Notes Google Cloud Platform project

In order to authenticate notes mobile application, it is required in the Mobile Backend Starter to use Client ID and Google Account. Client ID is used to authenticate that the notes mobile application is whom it claims to be. Google Account is used to authenticate that the user of the notes mobile application is whom it claims to be. Notes mobile application needs to provide both Client ID and Google Account when connecting to the mobile backend in order to be authenticated.

Authentication and authorization is not enabled by default in the Mobile Backend Starter. It possible to enable authentication and authorization in the Mobile Backend Starter by doing for example following steps:

- 1) Navigate to settings of the mobile backend with an Internet browser.
- 2) Change authentication / authorization setting to “Secured by Client IDs (Recommended)”.
→ See FIGURE 75.
- 3) Provide Android Client ID.
- 4) Provide Web Client ID.

5) Provide iOS Client ID.

6) Click “Save” action.

→ Authentication / authorization settings are now defined so that it is possible to authenticate and authorize notes mobile application in the mobile backend.

Mobile Backend settings

This page lets you configure the authentication and other options for your mobile backend. If you don't have a client application yet, download the [Android](#) or [iOS](#) sample client application.

Authentication / Authorization

- Locked Down (Access disabled)
All requests will be rejected.
- Open (for development use only)
All unauthenticated requests will be allowed. The backend will not be taking advantage of the integrated authentication to know the identity of the callers. For Android sample client app you can use either emulator or a physical device.
- Secured by Client IDs (Recommended)
Only authenticated calls using the registered Client IDs will be allowed. For Android sample client app you need to use a physical device.

Android

Android Client ID [Learn how to obtain an Android Client ID](#)

Web Client ID [Learn how to obtain a Web Client ID](#)

iOS

iOS Client ID [Learn how to obtain an iOS Client ID](#)

Google Cloud Messaging and iOS Push Notification Disabled

FIGURE 75. Authentication / Authorization settings of the notes mobile backend

The authentication / authorization setting “Secured by Client IDs” will authorize all authenticated call to be able to access notes in the Cloud Datastore. It is possible improve authorization of notes by adding prefix in to the kind name of the notes entity. It is possible for example to make sure that only entity owner can update or delete a note by adding a [private] kind name prefix to the notes entity. In the notes mobile application case it is required only to authorize all authenticated calls.

4.4.3 Microsoft Azure

In order to setup MBaaS solution in the Microsoft Azure, a mobile service needs to be created. It is possible to create a mobile service in the Microsoft Azure management portal by doing for example following steps:

- 1) Browse to Microsoft Azure management portal with an Internet browser.
- 2) Sign in to Microsoft Azure management portal.
- 3) Click the “New” action in the bottom of the Microsoft Azure management portal.
→ List of actions will be visible.
- 4) Select Compute / Mobile Service / Create from the actions list.
→ “New mobile service” dialog will be visible.
- 5) Input URL, database, region and backend type of the MBaaS solution to the “New mobile service” dialog.
→ In the FIGURE 76 is presented a screen capture of the “New mobile service” dialog with an example inputs for URL, database, region and backend type questions.
- 6) Click the “Next” action in the bottom right corner of the “New mobile service” dialog.
→ Database settings will be visible.
- 7) Specify the database settings.
→ In the FIGURE 77 is presented a screen capture of the “New mobile service” dialog with an example database settings specified.
- 8) Click the “Complete” action in the bottom right corner of the “New mobile service” dialog.
→ “New mobile service” dialog is closed.
→ Microsoft Azure management portal navigates to mobile services management (see FIGURE 78).
→ The mobile service defined in the “New mobile service” dialog is visible in the mobile services list with a status of “Creating”.
→ When the creation of the mobile service is done, the status of the mobile service changes to “Ready”.
→ The setup of the MBaaS solution in the Microsoft Azure is done.

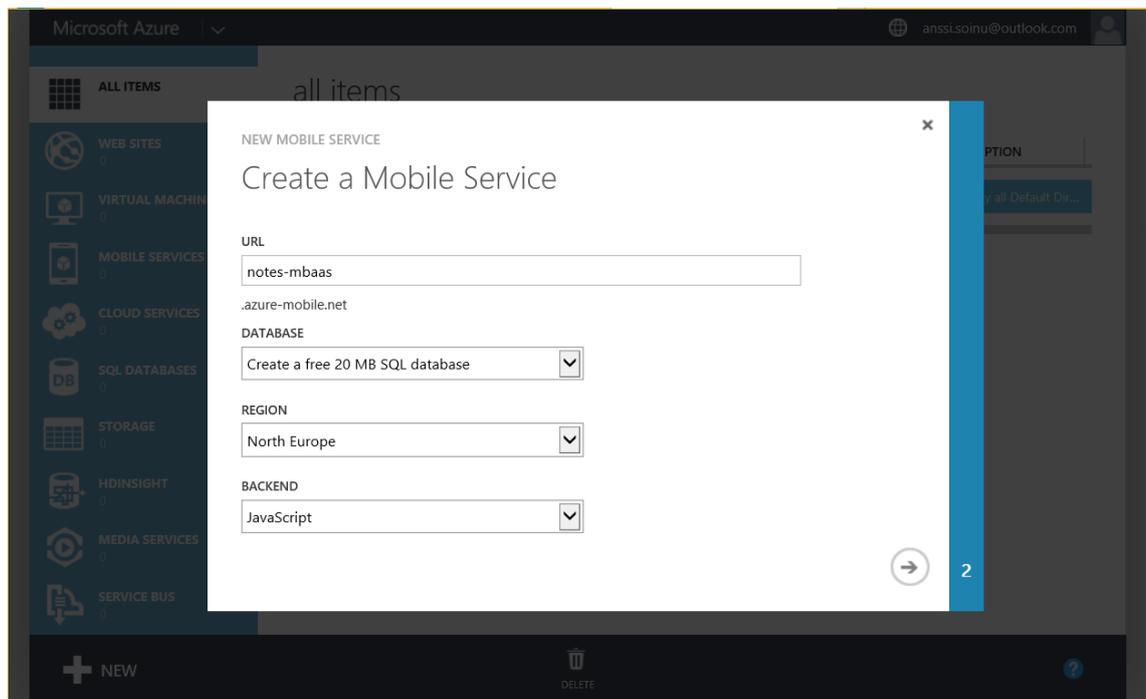


FIGURE 76. “New mobile service” dialog in the Microsoft Azure management portal with an example inputs for URL, database, region and backend type questions

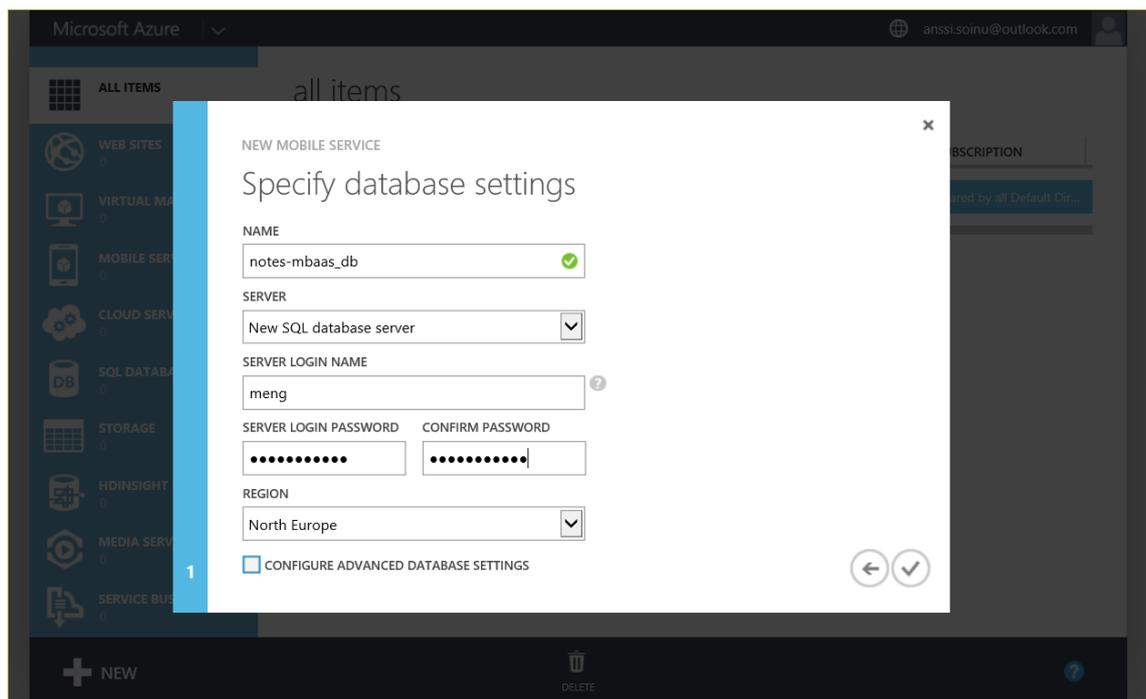


FIGURE 77. Specify database settings step of the “New mobile service” dialog in the Microsoft Azure management portal with an example database settings specified

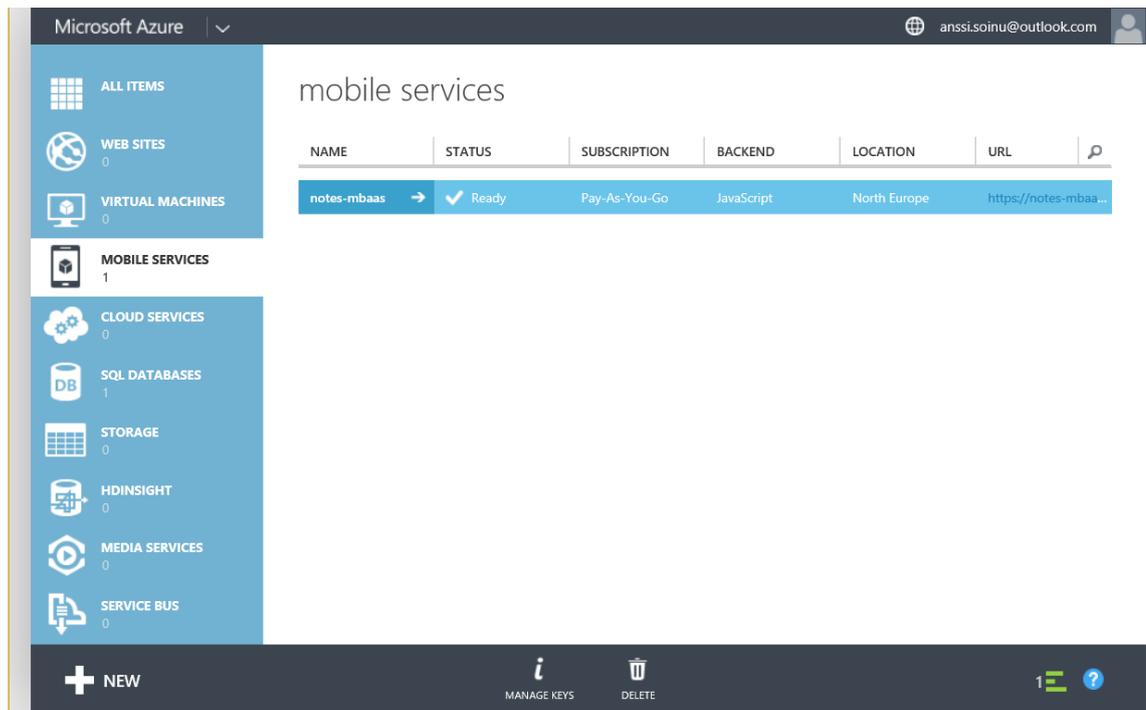


FIGURE 78. Mobile services management in the Microsoft Azure management portal

In order to be able to store notes in the cloud solution, a data table needs to be created in the mobile service. It is possible to create a data table in the mobile service for example by doing following steps:

- 1) Navigate to mobile service management in the Microsoft Azure management portal.
- 2) Select the mobile service where you would like to add a data table.
→ Microsoft Azure management portal will navigate to mobile service management (see FIGURE 79).
- 3) Click “DATA” in the mobile service management navigation.
→ Management of the mobile service data tables is visible (see FIGURE 80).
- 4) Click “ADD A TABLE” or “Create” action.
→ “Create New Table” dialog is visible (see FIGURE 81).
- 5) Give name for the new data table.
- 9) Click “Complete” action in the bottom right corner of the “Create New Table” dialog.
→ “Create New Table” dialog will be closed.
→ Microsoft Azure navigates to management of data tables.
→ Microsoft Azure starts creating the new data table.
→ After the data table creation is completed, the new data table will be visible in the data table management (see FIGURE 82).

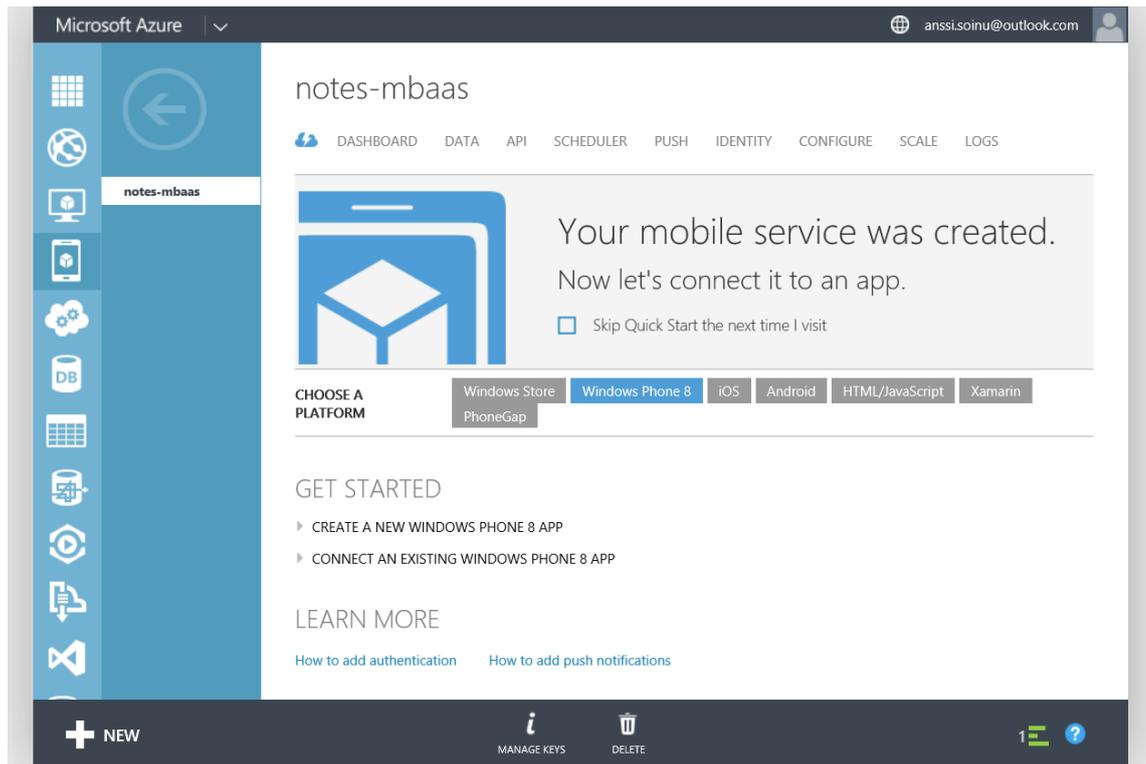


FIGURE 79. Mobile service management in the Microsoft Azure management portal

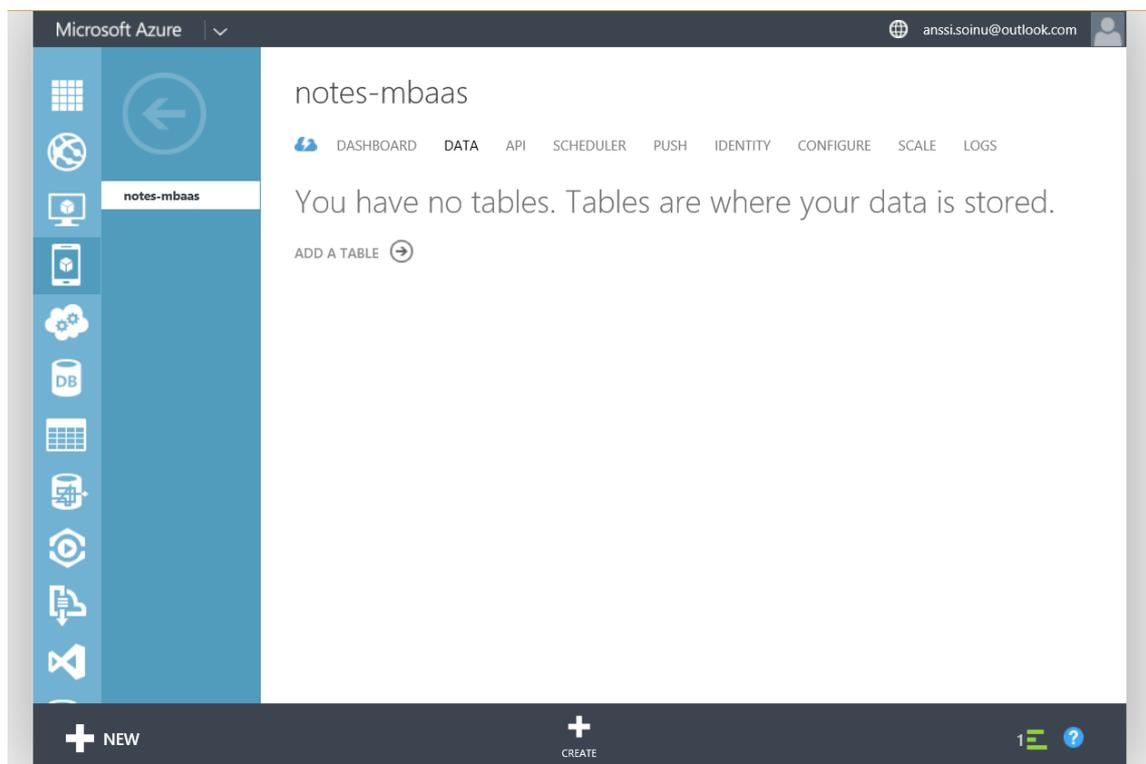


FIGURE 80. Management of mobile service data tables

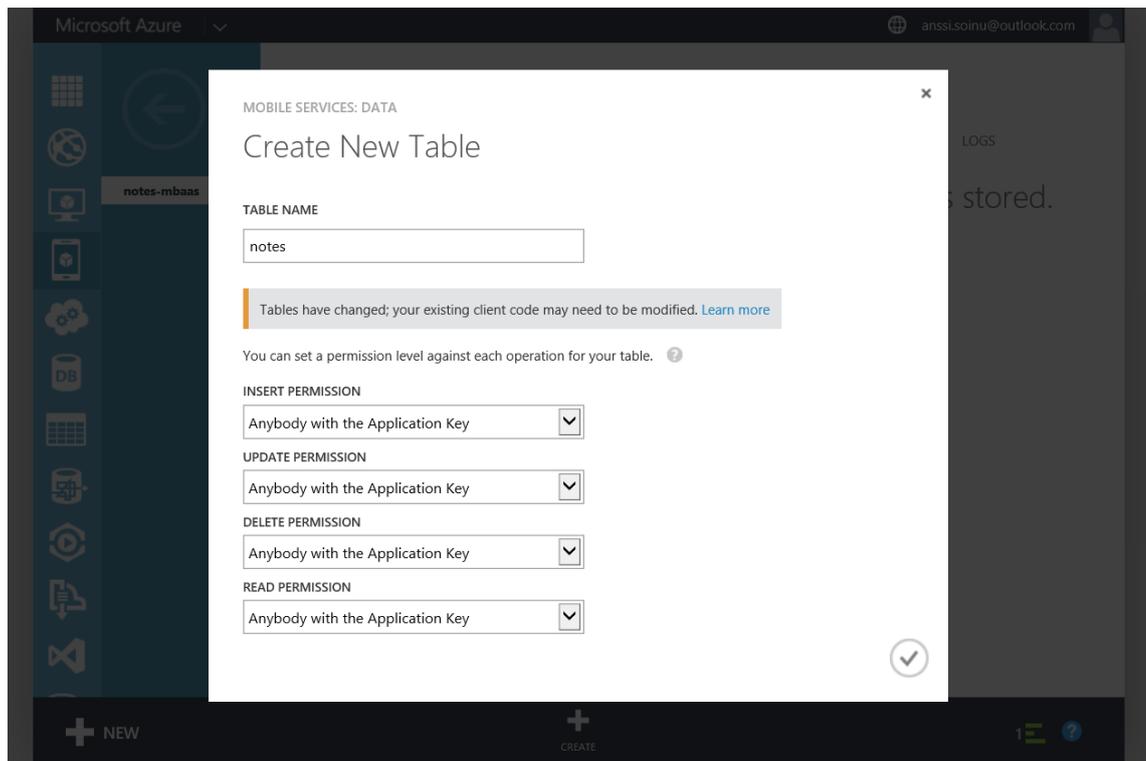


FIGURE 81. “Create New Table” dialog in the Microsoft Azure management portal

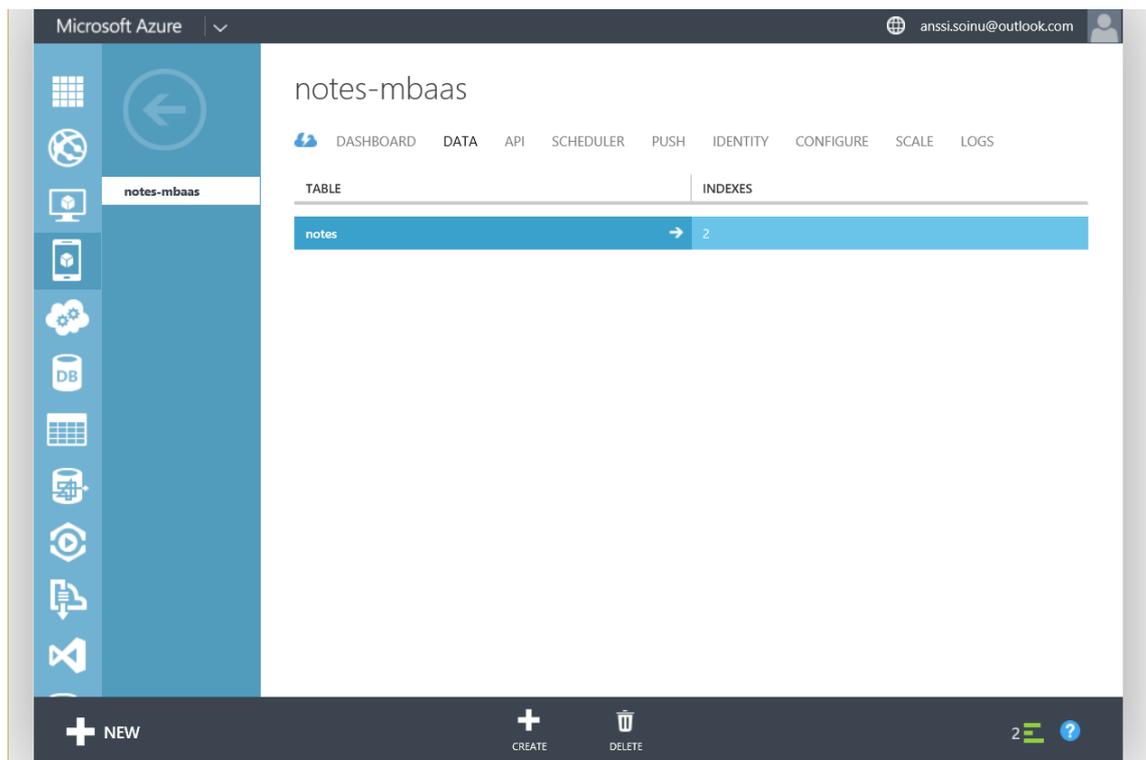


FIGURE 82. Notes data table listed in the data table management

Microsoft Azure will create the data table with following columns: id, _createdAt, _updatedAt and _version. In order to be able to store text of a note it is required to add new

column to the notes data table. It is possible to add new column to a data table for example by doing following steps:

- 1) Navigate to management of the data tables in the mobile service.
- 2) Select the data table where you would like to add a new column.
→ Data table management will be visible (see FIGURE 83).
- 3) Click “COLUMNS” from the data table management navigation.
→ Columns of the data table will be listed (see FIGURE 84).
- 4) Click “ADD COLUMN” action from the bottom of the Microsoft Azure management portal.
→ “Add Column” dialog will be visible (see FIGURE 85).
- 5) Define the data type for the column.
→ It is possible to define following types: string, number, date or boolean.
- 6) Define the name for the column.
- 7) Click “Complete” action in the bottom right corner of the “Add Column” dialog.
→ “Add Column” dialog will be closed.
→ Microsoft Azure starts adding the new column to the data table.
→ After the column is added, the new column will be visible in the list of data table columns (see FIGURE 86).

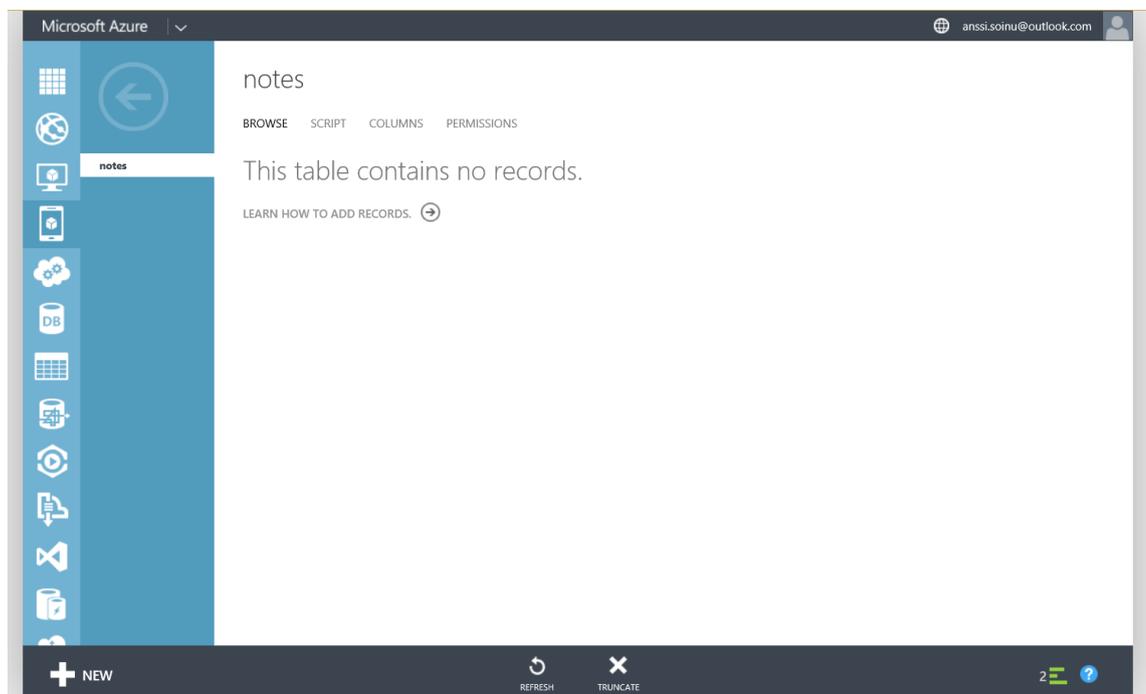


FIGURE 83. Data table management for the notes data table

Microsoft Azure | anssi.soinu@outlook.com

notes

BROWSE SCRIPT COLUMNS PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	string	✓ Indexed
_createdAt	date	✓ Indexed
_updatedAt	date	
_version	timestamp (MSSQL)	

+ NEW | + ADD COLUMN | 2 ?

FIGURE 84. Columns of the notes data table

Microsoft Azure | anssi.soinu@outlook.com

notes

BROWSE SCRIPT COLUMNS PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	string	Indexed
_createdAt	date	Indexed
_updatedAt	date	
_version	timestamp (MSSQL)	

MOBILE SERVICES: DATA Add Column

DATA TYPE: String

COLUMN NAME:

✓

+ NEW | + ADD COLUMN | 2 ?

FIGURE 85. “Add Column” dialog in the Microsoft Azure management portal

Microsoft Azure

notes

BROWSE SCRIPT COLUMNS PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	string	✓ Indexed
_createdAt	date	✓ Indexed
_updatedAt	date	
_version	timestamp (MSSQL)	
text	string	

+ NEW

+ ADD COLUMN

3 ?

FIGURE 86. Notes data table with the new text column

Microsoft Azure mobile services SDK makes possible to connect to an existing data table from the mobile application. By connecting to the notes data table from the notes mobile application it is possible to for example view, create, update or delete notes. How in practice the connection is made and how a data table is used, it is required to use mobile platform specific code. Microsoft Azure provides own mobile services SDK for iOS, Android and Windows Phone. In the appendix 3 it is possible to see an example code how Microsoft Azure mobile services SDK can be used in the Windows Phone notes mobile application to connect to notes data table.

In order to make possible for the notes mobile application to register listening changes of a certain note, it is required to create a data table in the mobile service for the registrations. In the registration data table needs to be defined following columns: handle (String), devicetype (String) and noteid (String). In the handle column will be stored the device PNS handle. The PNS handle is required in order to be able to push note change messages to the correct notes mobile application. In the devicetype column will be stored the type of device (e.g. Android, iOS or Windows Phone). It is required to store the device type in order to be able to use correct PNS. In the noteid column will be stored id of the note. Id of the note needs to be stored to specify which note changes the listener wants to listen. In the FIGURE 87 it is possible to see the note change registration data table in the Microsoft Azure management portal.

Microsoft Azure | anssi.soini@outlook.com

notechangeregistrations

BROWSE SCRIPT COLUMNS PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	string	Indexed
_createdAt	date	Indexed
_updatedAt	date	
_version	timestamp (MSSQL)	
handle	string	
devicetype	string	
noteid	string	

+ NEW + ADD COLUMN 7 ?

FIGURE 87. Data table columns of the note change registrations data table

In order to send note change messages to registered note change listeners, it is required to add custom script to update operation of the notes data table. The custom scripts needs to be written with JavaScript. The custom script needs to be added to update operation since it will be executed always when a note is changed. It is possible to add custom script to update operation of the notes data table for example by doing following steps:

- 1) Navigate to notes data table management.
- 2) Select “SCRIPT” from the navigation of the notes data table management.
 - Insert operation script of the notes data table is visible.
- 3) Change operation to “Update”.
 - Update operation script of the notes data table is visible (see FIGURE 88).
- 4) Write the script to send note change messages
 - In the FIGURE 89 and FIGURE 90 it is possible to see an example script how it is possible to send note change messages to registered listeners.
- 5) Click “Save” action in the bottom of the Microsoft Azure management portal.
 - Notes data table is now modified to send note change messages to registered listeners.

The custom script in the FIGURE 89 and FIGURE 90 will first complete the request received from the notes mobile application. The request in this case will contain the updated note. After the note has been updated successfully, function `sendNotifications` will be executed. The `sendNotifications` function will first search all registered listeners related to the updated note. After finding all registered listeners, the `sendNotifications` function will check the device type of the registered listener. The device type needs to be checked in order to use the correct PNS for sending the note change message. After the device type is recognized, as a last thing the `sendNotifications` function will execute device specific script for sending the note change message. In the device specific scripts the PNS handle (defined by the registered listener) will be used to make sure that the note change message will be sent to the correct notes mobile application.

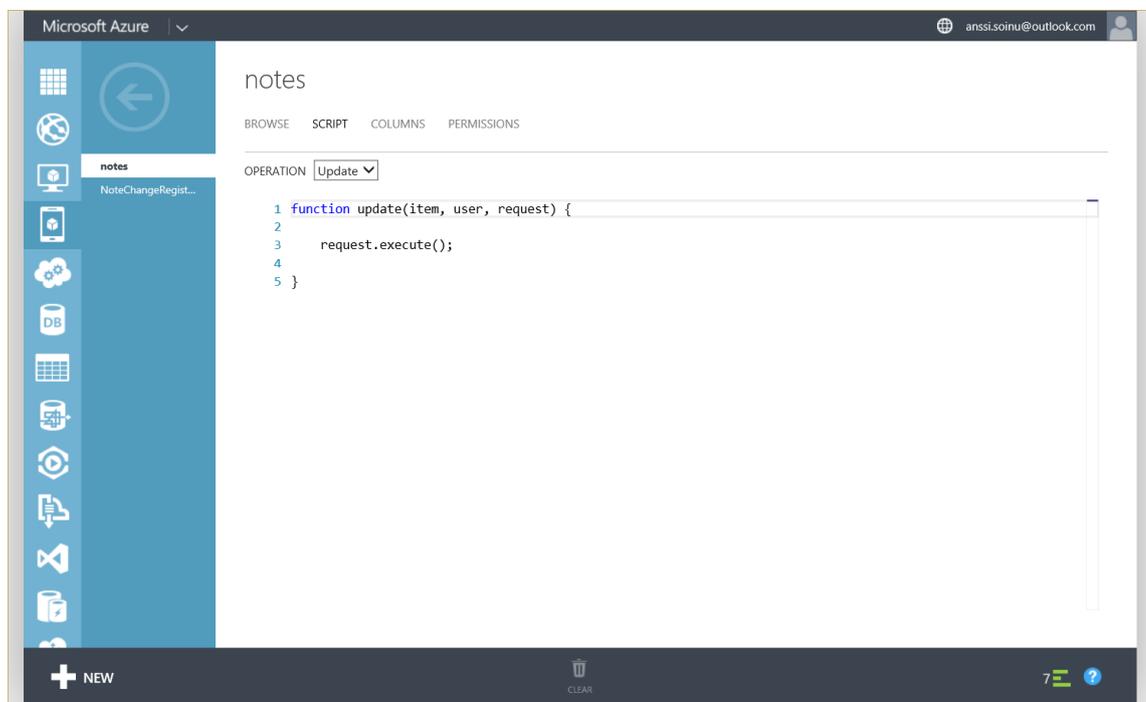


FIGURE 88. Update operation script of the notes data table

Microsoft Azure | anssi.soinu@outlook.com

notes

BROWSE SCRIPT COLUMNS PERMISSIONS

OPERATION Update

```

1 function update(item, user, request) {
2
3   request.execute({
4     success: function() {
5       request.respond();
6       sendNotifications(item);
7     }
8   });
9
10  function sendNotifications(note) {
11    var registrationsTable = tables.getTable('NoteChangeRegistrations');
12    registrationsTable.where({noteid: note.id})
13    .read({success: function(registrations) {
14      registrations.forEach(function(registration) {
15        if(registration.devicetype == "windowsPhone") {
16          pushChangeMessageToWindowsPhone(registration);
17        } else if (registration.devicetype == "Android") {
18          pushChangeMessageToAndroid(registration);
19        } else if (registration.devicetype == "ios") {
20          pushChangeMessageToIOS(registration);
21        }
22      });
23    }
24  });
25

```

+ NEW CLEAR ?

FIGURE 89. First part of the Update operation script of the notes data table

Microsoft Azure | anssi.soinu@outlook.com

notes

BROWSE SCRIPT COLUMNS PERMISSIONS

OPERATION Update

```

23   }
24   });
25 }
26
27 function pushChangeMessageToWindowsPhone(registration) {
28   push.mpn.sendToast(registration.handle, {
29     text1: "Note changed",
30     text2: "Note id: " + registration.noteid + ""
31   });
32 }
33
34 function pushChangeMessageToAndroid(registration) {
35   push.gcm.send(registration.handle, "Note (" + registration.noteid + ") changed!");
36 }
37
38 function pushChangeMessageToIOS(registration) {
39   push.apns.send(registration.handle, {
40     alert: "Note changed!",
41     payload: {
42       inAppMessage: "Note id: " + registration.noteid + ""
43     }
44   });
45 }
46

```

+ NEW CLEAR ?

FIGURE 90. Last part of the Update operation script of the notes data table

In order to send note changed messages to Android and iOS mobile applications, it is required that the push notification settings are set correctly in the Microsoft Azure mobile service configuration. It is possible to find Android and iOS push notification settings for example by doing following steps:

- 1) Navigate to mobile service management
- 2) Select “PUSH” from the mobile service management navigation
 - apple push notification settings are visible
 - google cloud messaging settings are visible

In order to send note change messages to Android mobile application it is require that API KEY setting is defined in the google cloud messaging settings. In order to push note change messages to iOS mobile application it is required that the certificate is defined in the apple push notification settings.

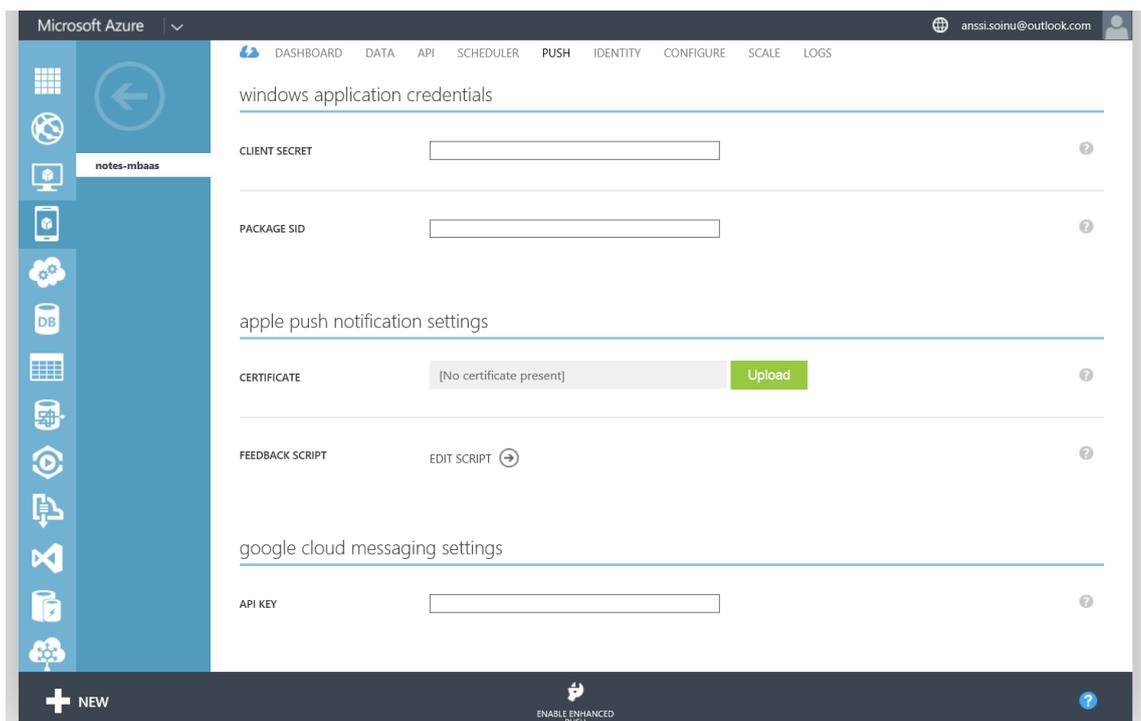


FIGURE 91. Mobile service push notification settings for the apple push notification and google cloud messaging

Authentication of the notes mobile application can be done in the mobile service by using application key. The application key is defined in the mobile service. The mobile application needs to use the application key when it connects to the mobile service using the mobile services SDK. It is possible to define new application key or see the current application key of the mobile service for example by doing following steps:

- 1) Navigate to mobile service management in the Microsoft Azure management portal.
- 2) Click “MANAGE KEYS” action in the bottom of the Microsoft Azure management portal.
 - “Manage Access Keys” dialog will be visible.
 - Current application key is visible.
- 3) Click the “regenerate” action to create a new application key.

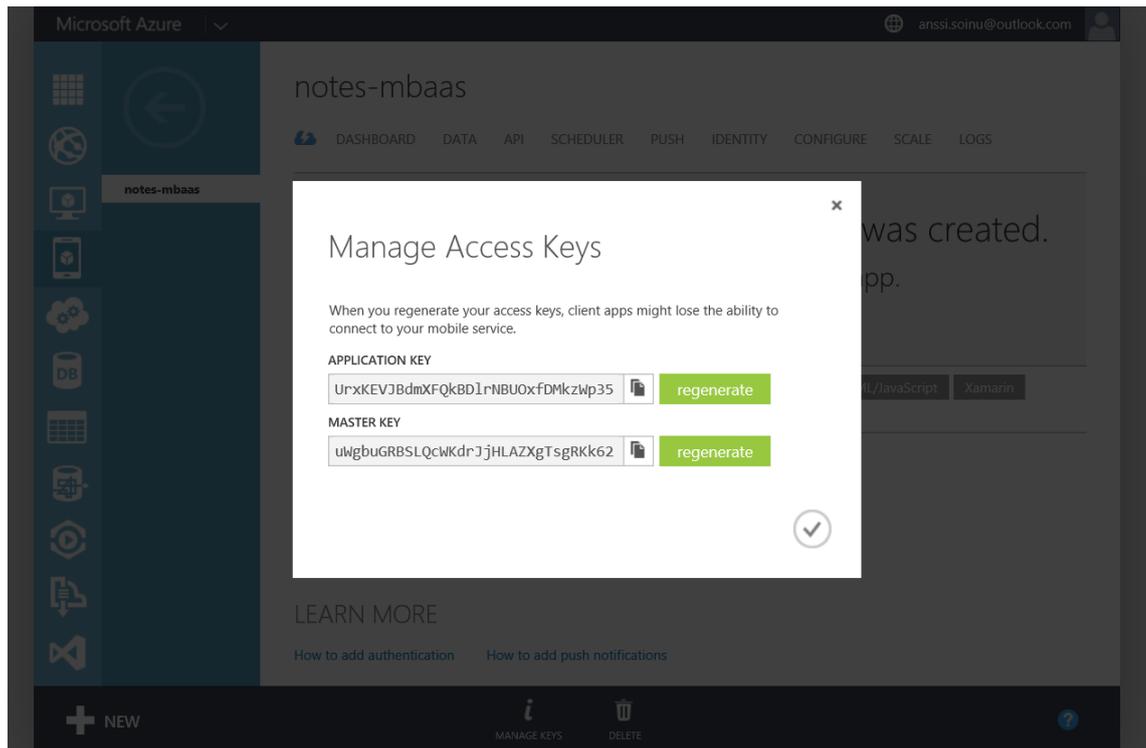


FIGURE 92. “Manage Access Keys” dialog in the Microsoft Azure management portal

Authorization of the notes mobile application can be done by setting permissions to the data tables. The notes mobile application will use following data tables from the mobile service: notes and NoteChangeRegistrations. To make sure that only the authenticated notes mobile application can access to previously mentioned data tables, it is required to adjust permissions of the notes and NoteChangeRegistrations data tables. It is possible to set permission to the notes data table for example by doing following steps:

- 1) Navigate to notes data table management in the Microsoft Azure management portal.
- 2) Select “PERMISSION” from the navigation of the notes data table management.
 - Permissions of the notes data table will be visible.

- 3) Select “Anybody with the Application Key” for every operation of the notes data table.
- 4) Click “Save” from the bottom of the Microsoft Azure management portal
 - Microsoft Azure starts saving the defined permissions for the notes data table.
 - After the saving is completed the notes data table can be accessed only when the application key is provided.

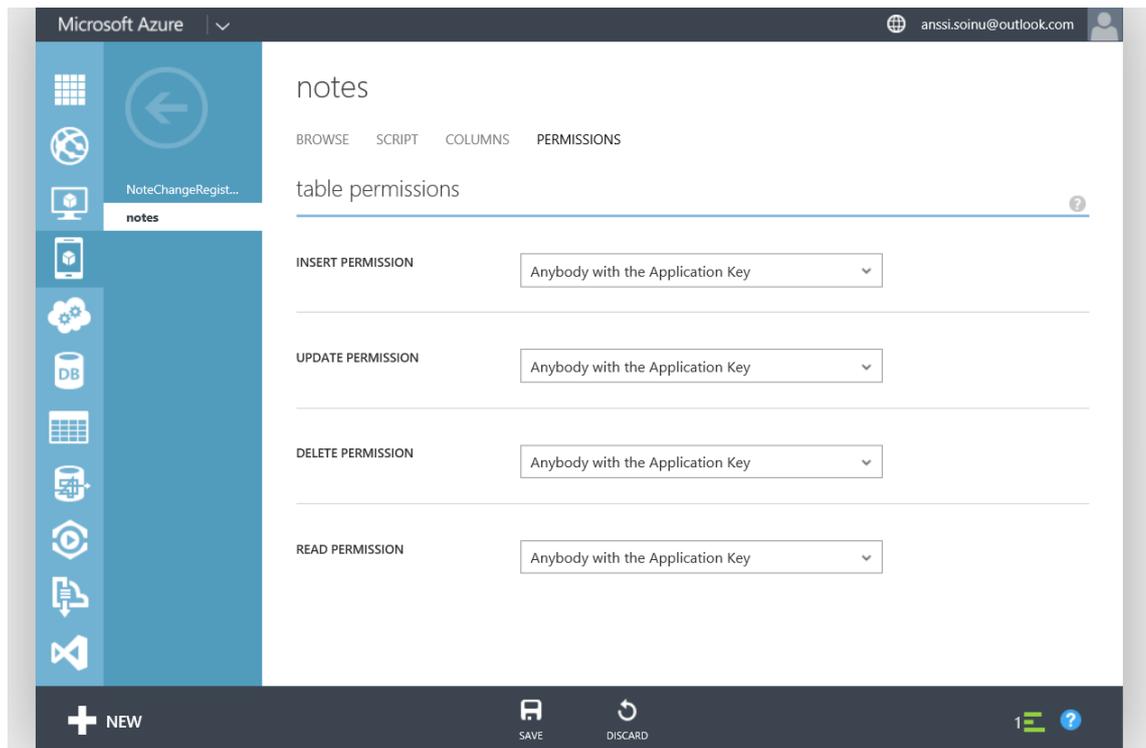


FIGURE 93. Permissions of the notes data table

5 COMPARISON

5.1 Setting up the MBaaS solution

Amazon Web Services, Google Cloud Platform and Microsoft Azure have different approaches on how to setup MBaaS solution in the cloud platform. One of the biggest difference is that Amazon Web Services does not provide one service or solution which could act as MBaaS solution. In the Amazon Web Services it is required to setup each required services as an own service in order to create MBaaS cloud solution. Google Cloud Platform does not either have MBaaS solution in the cloud platform. Google's approach is to give a starter solution (Mobile Backend Starter) which can be customized and deployed to Google's Cloud Platform (App Engine). Microsoft Azure has MBaaS solution (Mobile Services) in the cloud platform. Microsoft Azure is also giving possibility to customize the MBaaS solution in a local development environment or directly from the Microsoft Azure management portal.

5.2 Pull Information to Cloud Solution

Amazon Web Services, Google Cloud Platform and Microsoft Azure have all SDKs for making possible to pull information from the mobile application to the cloud solution. How in practice the SDK is used to pull information from the mobile application to the cloud solution in Amazon Web Services, Google Cloud Platform or Microsoft Azure can depend on where the information will be stored or what kind of API/functionality is used from the cloud solution.

5.3 Store Information in Cloud Solution

Amazon Web Services does not provide MBaaS solution with a default datastore. MBaaS solutions of Google Cloud Platform and Microsoft Azure have a default datastore which can be used to store information in the cloud solution. In the Mobile Backend Starter of the Google Cloud Platform the default datastore is Cloud Datastore. In the Mobile Services of the Microsoft Azure the default datastore is Microsoft Azure SQL Database. One

big difference between Microsoft Azure SQL Database and Cloud Datastore is that Microsoft Azure SQL Database is relational database and Cloud Datastore is non-relational database. To use in the Mobile Services of the Microsoft Azure or in the Mobile Backend Starter of the Google Cloud Platform other datastore than the default, customization of the cloud solution is required.

Amazon Web Services, Google Cloud Platform and Microsoft Azure provides possibility to store information to a relational or non-relational database. It is also possible with Amazon Web Services, Google Cloud Platform and Microsoft Azure to store objects which size can be large (for example videos or images).

5.4 Process Information in Cloud Solution

Amazon Web Services, Google Cloud Platform and Microsoft Azure provides possibility to create custom APIs which can be used by the mobile application to process information in the cloud solution. To create a custom API in the Amazon Web Services it is required to setup a custom virtual server with a custom web service. Custom APIs in the Google Cloud Platform can be done using Google Cloud Endpoints. The Mobile Backend Starter of the Google Cloud Platform uses Google Cloud Endpoints. In the Microsoft Azure custom APIs is a feature of Mobile Service.

Amazon Web Services, Google Cloud Platform and Microsoft Azure provides possibility to schedule jobs to be executed in the cloud solution. In the Amazon Web Services it is required to take into use specific service to make possible to schedule jobs. In the Google Cloud Platform scheduling jobs is a feature of App Engine. The Mobile Backend Starter of the Google Cloud Platform uses App Engine. In the Microsoft Azure scheduling jobs is a feature of Mobile Services.

In the Mobile Services of the Microsoft Azure it is possible to add custom logic to process information in every insert, update, delete or read operation of a data table. Google Cloud Platform or Amazon Web Services does not provide similar possibility.

In the App Engine of the Google Cloud Platform it is possible to use Task Queue feature. With the Task Queue feature it is possible to put tasks in to a queue. Mobile Services of the Microsoft Azure or Amazon Web Services does not provide similar possibility.

5.5 Push Information to Mobile Application

Amazon Web Services, Google Cloud Platform and Microsoft Azure supports information push from the cloud solution to mobile applications which supports Apple Push Notification service or Google Cloud Messaging. With Amazon Web Services it is also possible to push information to mobile applications which supports Amazon Device Messaging. With Microsoft Azure it is also possible to push information to mobile applications which support Microsoft Push Notification Service or Windows Push Notification Service.

With Amazon Web Services, Google Cloud Platform and Microsoft Azure it is possible to push information to a certain user of a mobile application or group of mobile application users. The terms (tag, topic) used to describe / filter information differs between the compared cloud solutions but publish / subscribe functionality in all compared cloud solutions is similar.

Microsoft Azure makes possible to push information to a mobile application using a template defined by the mobile application. Amazon Web Services or Google Cloud Platform does not provide similar functionality.

Google Cloud Platform has a feature called Continuous Queries. Similar functionality does not exist out of the box in the Amazon Web Services or in the Microsoft Azure. It is possible to build also similar functionality using for example publish / subscribe functionality to Amazon Web Services or to Microsoft Azure.

5.6 Security

In the Amazon Web Services it is possible to authenticate user for example using identity providers which support SAML 2.0 (for example using Microsoft Active Directory, Facebook or Google). In the Amazon Web Services it is possible to authenticate user also using users feature of Amazon Identity and Access Management. In the Mobile Backend Starter of the Google Cloud Platform it is possible to authenticate user using Google Account Authentication. In the Microsoft Azure Mobile Service it is possible to authenticate user using following identity providers: Microsoft, Facebook, Twitter, Google and Microsoft Azure Active Directory.

Amazon Web Services offers also multi-factor authentication to add extra layer of protection on top of IAM authentication. Mobile Backend Starter of the Google Cloud Platform or Microsoft Azure Mobile Services does not provide similar possibility out of the box.

In the Amazon Web Services it is possible to authenticate mobile application using users feature of Amazon Identity and Access Management. In the Mobile Backend Starter of the Google Cloud Platform it is possible to authenticate mobile application using Client ID. In the Microsoft Azure Mobile Services it is possible to authenticate mobile application using master or application key. The difference between compared cloud solutions is that Amazon Web Services and Mobile Backend Starter of the Google Cloud Platform makes possible to have several different authentication keys for mobile applications. In the Microsoft Azure it is possible to define two authentication keys (master and application).

In all compared cloud solutions it is possible to use existing authorization functionality and / or implement custom functionality for handling authorization. Amazon Web Services provides Identity and Access Management groups, roles, permissions and policies as an existing authorization functionality. Mobile Backend Starter of the Google Cloud Platform provides Access Control for Cloud Entities as an existing authorization functionality. Microsoft Azure Mobile Services provides as an existing authorization functionality a possibility to add permissions to data table and custom API operations.

5.7 Mobile Platform Support

Amazon Web Services provides native SDKs for Android, iOS, Java, .NET, JavaScript (browser), JavaScript (Node.js), PHP, Python and Ruby. Amazon Web Services SDK for the .NET supports Windows Phone. The Mobile Backend Starter of the Google Cloud Platform supports Android and iOS mobile platforms. Microsoft Azure Mobile Services provides native SDKs for Windows Store, Windows Phone, Android and iOS mobile platforms.

Amazon Web Services supports information push to Android, iOS and Kindle Fire mobile applications. Mobile Backend Starter of the Google Cloud Platform supports information push to Android and iOS mobile applications. Microsoft Azure Mobile Services supports information push to Android, iOS, Windows Store and Windows Phone mobile applications.

5.8 Pricing Model

Amazon Web Services, Google Cloud Platform and Microsoft Azure provides pricing calculators. With Amazon Web Services pricing calculator it is possible to calculate monthly cost estimate. With Google Cloud Platform pricing calculator it is not possible to calculate complete monthly cost estimate for all Google Cloud Platform products which are used by the Mobile Backend Starter. Google Cloud Platform pricing calculator can be used to calculate cost estimates only for certain Google Cloud Platform products. With Microsoft Azure pricing calculator it is possible to calculate monthly cost estimate for Mobile Services.

In the Amazon Web Services, Google Cloud Platform and Microsoft Azure pricing goes based on usage of a selected service of it. Every service in the compared cloud solutions have own pricing factors. Amazon Web Services does not provide pricing categories like App Engine of Google Cloud Platform (free, paid and premium) or Mobile Service of Microsoft Azure (free, basic and standard) provides. In the Microsoft Azure on top of the pricing categories exists following pricing plans: “pay-as-you-go plan” and “6 or 12-month plan”. By selecting the “6 or 12-month plan” in the Microsoft Azure it is possible

to save 20 – 32 % compared to “pay-as-you-go plan”. Amazon Web Services or Google Cloud Platform does not offer a time based pricing plans like Microsoft Azure offers.

At the writing time of this master’s thesis Amazon Web Services offered a free usage tier. With the free usage tier it is possible to use Amazon Web Services for free in a certain given limits. The free usage tier of the Amazon Web Services is valid only for 12 months starting from the user sign up date to Amazon Web Services. Mobile Backend Starter of the Google Cloud Platform uses Google Cloud Endpoints and App Engine. Google Cloud Platform offers Google Cloud Endpoints as a product free for applications running on top of App Engine. The Mobile Backend Starter of the Google Cloud Platform runs on top of the App Engine so the Google Cloud Endpoints is free as a product in the Mobile Backend Starter. Google has defined usage quotas for the App Engine applications so that every application is free within certain usage quota. The usage quotas in the App Engine is defined per day basis. At the written time of this master’s thesis Microsoft Azure offered free one month trial including 150 € to spend on all Microsoft Azure services. Microsoft Azure offers also free pricing category which makes possible to use Microsoft Azure for free in certain limits.

6 CONCLUSION

Based on this master's thesis study and comparison of Amazon.com's, Google's and Microsoft's cloud platforms it is possible to see that all previously mentioned cloud platforms offers different kind of approach on what comes to mobile application support. It is possible to see that Amazon Web Services does not offer one product or service which could be identified as MBaaS product. In the Amazon Web Services it is required to use different Amazon Web Services products or services in order to create MBaaS solution. Microsoft Azure provides one product (Mobile Services) which can identified as MBaaS solution. The Mobile Services can be customized and exists in the Microsoft Azure cloud platform. Google Cloud Platform provides a "code project" called Mobile Backend Starter which can be customized and deployed to Google Cloud Platform. A deployed Mobile Backend Starter can be identified as MBaaS solution.

To pick the right cloud solution for a mobile application can depend on several factors like is the development team able to understand or customize a given MBaaS code project, does the development team have time to learn different products or services offered by the selected cloud solution, is all required mobile platforms supported by the selected cloud solution, what kind of features from the cloud solution are required by the mobile application, is the customization of the MBaaS required and how much does costs to run the selected cloud solution. There can many factors based on for example the development team, the actual mobile application or investors of the mobile application. Every case of building a mobile application can be unique situation with unique requirements. It is not even a bad thing to switch from one cloud solution to another during the creation of a mobile application if the current requirements requires the switch.

In the FIGURE 94, FIGURE 95 and FIGURE 96 is presented pros and cons for the studied and compared cloud solutions. Pros and cons are listed based on this master's thesis study and comparison. The pros and cons can make easier to choose the most suitable cloud solution for a certain case of building a mobile application.

Amazon Web Services	
+	<p>Wide range of services</p> <p>Amazon Web Services provides wide range of services. From the Amazon Web Services it is possible to find for example several services just for storing information in the Amazon Web Services.</p>
+	<p>Pricing calculator</p> <p>Amazon Web Services provides pricing calculator which makes possible to calculate cost estimates for the MBaaS solution.</p>
+	<p>Scalable services</p> <p>From the Amazon Web Services it is possible to pick services which can support for example scalability of the data model and gives possibility to introduce new features for the MBaaS.</p>
+	<p>Authentication</p> <p>Amazon Web Services provides several different possibilities to authenticate user and mobile application. It is possible to authenticate user for example using SAML 2.0 provider like Facebook or Google.</p>
+ / -	<p>Authorization</p> <p>Amazon Web Services can provide based on selected service of it possibility to add authorization rules or logic.</p>
-	<p>Several services needs to be used in order to create a MBaaS solution</p> <p>Amazon Web Services does not provide one service which could be identified as MBaaS. It is required to use several services from the Amazon Web Services in order to have MBaaS solution.</p>

-	<p>Setup time of MBaaS</p> <p>Because of the wide range of services in the Amazon Web Services, it is required use time to be able to choose the correct services. From a person having previous experience of Amazon Web Service it might not take as much time as it can take from a person without any previous experience of Amazon Web Service. In the Amazon Web Service all used services needs to be set up separately.</p>
-	<p>Support of mobile platform</p> <p>Amazon Web Services offers SDK for Android, iOS and Windows Phone mobile platforms. Amazon Web Services supports only pushing information to Android and iOS mobile applications. It is not possible to push information from the Amazon Web Services to for example Windows Phone mobile application.</p>

FIGURE 94. Amazon Web Services pros and cons

Google Cloud Platform	
+	<p>Provides MBaaS solution</p> <p>Google provides Mobile Backend Starter code project which can be deployed to Google App Engine. The code project will provide basic functionality in order to create MBaaS solution in the Google Cloud Platform.</p>
+	<p>Scalability of the backend</p> <p>It is possible to modify the Mobile Backend Starter of the Google Cloud Platform to match as good as possible to needs of the mobile application. Modifications to Mobile Backend Starter needs to be done by changing or introducing new code to Mobile Backend Starter.</p>
+	<p>Scalability of the default datastore</p> <p>By default Mobile Backend Starter of the Google Cloud Platform uses Cloud Datastore which is a non-relational database. With the non-relational database it can be easier to adapt to future needs of the mobile application compared to relational database.</p>

+	<p>Setup time of MBaaS</p> <p>The setup of the Mobile Backend Starter can be done in the Google Cloud Console by browsing to Mobile Backend Starter page and pressing “Deploy” action. In the Google Cloud Platform it is not required to setup several services separately in order to setup MBaaS.</p>
+	<p>Continuous Queries</p> <p>Continuous Queries feature of the Google Cloud Platform makes possible to create queries to past and future. By making a query to future it is possible to push information from the Google Cloud Platform to a mobile application in the future if the defined query conditions are met.</p>
-	<p>Authentication</p> <p>Mobile Backend Starter of the Google Cloud Platform supports only authentication using Google Account and Client IDs. Mobile Backend Starter does not support authentication using for example Twitter, Facebook or Microsoft.</p>
-	<p>Authorization</p> <p>In the Mobile Backend Starter of the Google Cloud Platform it is required to add a prefix on the cloud entity kind name in order to define authorization rule for a cloud entity. It is required to add custom logic to handle authorization in a custom API. To make authorization work it is required always that the authentication has been done using Google Account and Client ID.</p>
-	<p>Support of mobile platform</p> <p>Mobile Backend Starter of the Google Cloud Platform supports only Android and iOS mobile platform. Mobile Backend Starter does not support for example Windows Phone mobile platform.</p>
-	<p>Pricing calculator</p> <p>With the pricing calculator of the Google Cloud Platform it is not possible to calculate price estimates for Mobile Backend Starter.</p>

FIGURE 95. Google Cloud Platform pros and cons

Microsoft Azure	
+	<p>Provides MBaaS solution</p> <p>Microsoft Azure provides one service (Mobile Services) in the cloud which can be identified as MBaaS.</p>
+	<p>Setup time of MBaaS</p> <p>Mobile Services setup can be done in the Microsoft Azure management console. In the Microsoft Azure it is not required to setup several different services separately in order to setup MBaaS solution.</p>
+	<p>Scalability of the backend</p> <p>Mobile Services of the Microsoft Azure is built on top Node.js. It is possible to access sources of the created Mobile Service and add functionality to it. New functionalities needs to implemented using JavaScript. In the Mobile Service it is possible add script to every operation of a data table or a custom API. The script that will be added to operation of a data table or custom API needs to be implemented using JavaScript.</p>
+	<p>Authentication</p> <p>Mobile Services of the Microsoft Azure supports user authentication using Microsoft, Facebook, Twitter, Google or Microsoft Azure Active Directory. It is possible authenticate mobile application using application key.</p>
+	<p>Authorization</p> <p>Mobile Services of the Microsoft Azure provides possibility to define authorization rules for every operation of data table or custom API. It is possible also to add custom authorization logic in every operation of data table or custom API.</p>
+	<p>Support of mobile platforms</p> <p>Mobile Services provides SDKs for Android, iOS and Windows Phone mobile platforms. It is possible to push information from the Mobile Services to Android, iOS or Windows Phone mobile application.</p>

+	<p>Push notification templates</p> <p>Microsoft Azure makes possible to push information to a mobile application using a template defined by the mobile application. By using templates it is possible to for example handle localization of push notifications.</p>
+	<p>Pricing calculator</p> <p>Microsoft Azure provides pricing calculator which makes possible to calculate cost estimates for the MBaaS solution.</p>
+	<p>Documentation</p> <p>Mobile Services provides example mobile applications to get starter with using Mobile Services from the mobile application. The example mobile application are provider for different mobile platforms.</p>
-	<p>Scalability of the default datastore</p> <p>By default Mobile Services of the Microsoft Azure uses Microsoft Azure SQL database which is a relational database. With the relational database it can be harder to adapt to future needs of the mobile application compared to non-relational database.</p>

FIGURE 96. Microsoft Azure pros and cons

REFERENCES

Amazon Web Services, About AWS, Read 1.1.2014, <http://aws.amazon.com/about-aws/>

Amazon Web Services, Amazon Web Services Simple Monthly Calculator, Amazon.com Inc., Read 19.2.2014, <http://calculator.s3.amazonaws.com/index.html>

Amazon Web Services, AWS Free Usage Tier, Amazon.com Inc., Read 19.2.2014, <http://aws.amazon.com/free/>

Amazon Web Services, AWS Identity and Access Management (IAM), Amazon.com Inc., Read 18.2.2014, <http://aws.amazon.com/iam/>

Amazon Web Services, AWS Solutions, Amazon.com Inc., Read 11.2.2014, <http://aws.amazon.com/solutions/aws-solutions/>

Amazon Web Services, Amazon S3 Product Details, Amazon.com Inc., Read 15.2.2014, <http://aws.amazon.com/s3/details/>

Amazon Web Services, Amazon SimpleDB Product Details, Amazon.com Inc., Read 15.2.2014, <http://aws.amazon.com/simpledb/details/>

Amazon Web Services. Amazon SNS, Amazon.com Inc., Read 17.2.2014, <http://aws.amazon.com/sns/>

Amazon Web Services, Manage User Credentials, Amazon.com Inc., Read 18.2.2014, <http://aws.amazon.com/iam/details/managing-user-credentials/>

Amazon Web Services, Mobile Applications on AWS, Amazon.com Inc., Read 13.2.2014, <http://aws.amazon.com/mobile/>

Amazon Web Services, Multi-Factor Authentication, Amazon.com Inc., Read 18.2.2014, <http://aws.amazon.com/iam/details/mfa/>

Amazon Web Services, Product and Services by Region, Amazon.com Inc., Read 15.3.2014, <http://aws.amazon.com/about-aws/globalinfrastructure/regional-product-services/>

Amazon Web Services, Sample Code & Libraries, Amazon.com Inc., Read 15.2.2014, <http://aws.amazon.com/code/>

AWS documentation, Amazon SNS Mobile Push Notifications, Amazon.com Inc., Read 17.2.2014, <http://docs.aws.amazon.com/sns/latest/dg/SNSMobilePush.html>

AWS documentation, Credentials (Passwords, Access Keys, MFA, and Certificates), Amazon.com Inc., Read 18.2.2014, http://docs.aws.amazon.com/IAM/latest/UserGuide/Using_ManagingLogins.html

AWS documentation, DynamoDB Data Model, Amazon.com Inc., Read 14.2.2014, <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DataModel.html>

AWS documentation, Getting Started with Amazon RDS, Amazon.com Inc., Read 15.2.2014,

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.html

AWS documentation, IAM Users and Groups, Amazon.com Inc., Read 18.2.2014,

http://docs.aws.amazon.com/IAM/latest/UserGuide/Using_WorkingWithGroupsAndUsers.html

AWS documentation, Instances and AMIs, Amazon.com Inc., Read 15.2.2014,

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instances-and-amis.html>

AWS documentation, Managing Server Certificates, Amazon.com Inc., Read 18.2.2014,

<http://docs.aws.amazon.com/IAM/latest/UserGuide/ManagingServerCerts.html>

AWS documentation, Overview of AWS IAM Permissions, Amazon.com Inc., Read 18.2.2014,

<http://docs.aws.amazon.com/IAM/latest/UserGuide/PermissionsOverview.html>

AWS documentation, Overview of AWS IAM Policies, Amazon.com Inc., Read 18.2.2014, <http://docs.aws.amazon.com/IAM/latest/UserGuide/PoliciesOverview.html>

AWS documentation, Roles (Delegation and Federation), Amazon.com Inc., Read 18.2.2014, <http://docs.aws.amazon.com/IAM/latest/UserGuide/WorkingWithRoles.html>

AWS documentation, Running Cron Jobs - AWS OpsWorks, Amazon.com Inc., Read 15.3.2014,

<http://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-extend-cron.html>

AWS documentation, Step 2: Create Example Tables, Amazon.com Inc., Read 15.2.2014, <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted-CreateTables.html>

AWS documentation, Using IAM to Control Access to DynamoDB Resources, Amazon.com Inc., Read 29.3.2014,

<http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/UsingIAM-WithDDB.html>

AWS documentation, What is Amazon DynamoDB?, Amazon.com Inc., Read 14.2.2014, <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>

AWS documentation, What is Amazon EC2?, Amazon.com Inc., Read 15.2.2014,

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

AWS documentation, What is Amazon Relational Database Service?, Amazon.com Inc., Read 15.2.2014,

<http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>

Doug Hauger, 2010, Windows Azure General Availability, Microsoft Inc., Published 1.2.2010, Read 28.1.2014,

http://blogs.technet.com/b/microsoft_blog/archive/2010/02/01/windows-azure-general-availability.aspx

Facebook Reports Second Quarter 2013 Results, Facebook Inc., Published 24.7.2013
Read 1.1.2014, <http://investor.fb.com/releasedetail.cfm?ReleaseID=780093>

Google, Data center locations, Google Inc., Read 22.2.2014,
<http://www.google.com/about/datacenters/inside/locations/index.html>

Google Cloud Platform, App Engine Pricing, Google Inc., Read 6.3.2014,
<https://cloud.google.com/products/app-engine/#pricing>

Google Cloud Platform, Cloud Endpoints, Google Inc., Read 6.3.2014,
<https://cloud.google.com/products/cloud-endpoints/#pricing>

Google Cloud Platform, Google Cloud Platform Pricing Calculator, Google Inc., Read
6.3.2014, <https://cloud.google.com/products/calculator/>

Google Cloud Platform, Products, Google Inc., Read 21.2.2014,
<https://cloud.google.com/products/>

Google Cloud Platform, Mobile Backend Starter: The Best Mix of BaaS and PaaS,
Google Inc., Read 21.2.2014,
<https://cloud.google.com/developers/articles/mobile-backend-starter>

Google Cloud Platform, Mobile Backend Starter - API Guide, Google Inc., Read
21.2.2014,
<https://cloud.google.com/developers/articles/mobile-backend-starter-api-reference>

Google Cloud Platform, Mobile Solutions on Google Cloud Platform, Google Inc., Read
24.2.2014,
<https://cloud.google.com/developers/articles/mobile-application-solutions#architecture-diagram>

Google Cloud Platform, Solutions, Google Inc., Read 21.2.2014,
<https://cloud.google.com/solutions/>

Google Developers, About Google Cloud SQL, Google Inc., Read 28.2.2014,
<https://developers.google.com/cloud-sql/docs/introduction>

Google Developers, Adding a Method Protected by OAuth, Google Inc., Read 5.3.2014,
https://developers.google.com/appengine/docs/java/end-points/getstarted/backend/auth#why_the_user_param_is_required

Google Developers, Authentication on the Mobile Backend Starter, Google Inc., Read
4.3.2014, <https://developers.google.com/cloud/samples/mbs/authentication>

Google Developers, Continuous Queries on the Mobile Backend Starter, Google Inc.,
Read 1.3.2014, https://developers.google.com/cloud/samples/mbs/continuous_query

Google Developers, Downloads, Google Inc., Read 1.3.2014,
<https://developers.google.com/appengine/downloads>

Google Developers, Enable Authentication, Google Inc., Read 4.3.2014,
https://developers.google.com/cloud/samples/mbs/android/enable_auth

Google Developers, Entities, Properties and Keys, Google Inc., Read 28.2.2014,
<https://developers.google.com/appengine/docs/java/datastore/entities>

Google Developers, Java Datastore API, Google Inc., Read 28.2.2014,
<https://developers.google.com/appengine/docs/java/datastore/>

Google Developers, Java Task Queue Configuration, Google Inc., Read 1.3.2014,
https://developers.google.com/appengine/docs/java/config/queue#Defining_Push_Queues_and_Processing_Rates

Google Developers, Mobile Backend Starter, Google Inc., Read 21.2.2014,
<https://developers.google.com/cloud/samples/mbs/>

Google Developers, Overview of Google Cloud Endpoints, Google Inc., Read 1.3.2014,
<https://developers.google.com/appengine/docs/java/endpoints/>

Google Developers, Pub/Sub Messaging on the Mobile Backend Starter, Google Inc.,
Read 1.3.2014, https://developers.google.com/cloud/samples/mbs/pubsub_messaging

Google Developers, Quotas, Google Inc., Read 28.2.2014,
<https://developers.google.com/appengine/docs/quotas>

Google Developers, Scheduled Tasks With Cron for Java, Google Inc., Read 1.3.2014,
<https://developers.google.com/appengine/docs/java/config/cron>

Google Developers, Storing Data, Google Inc., Read 28.2.2014,
https://developers.google.com/appengine/docs/java/storage#alternative_storage_solutions

Google Developers, Task Queue Java API Overview, Google Inc., Read 1.3.2014,
<https://developers.google.com/appengine/docs/java/taskqueue/>

Google Developers, Using Pull Queues in Java, Google Inc., Read 1.3.2014,
<https://developers.google.com/appengine/docs/java/taskqueue/overview-pull>

Google Developers, Using Push Queues in Java, Google Inc., Read 1.3.2014,
https://developers.google.com/appengine/docs/java/taskqueue/overview-push#Java_URL_endpoints

Glenn Block, 2013, Azure Mobile Services Deep Dive into Node.js Scripting, Microsoft Inc., Published 6.9.2013, Read 5.2.2014,
<http://channel9.msdn.com/Events/TechEd/Australia/2013/AZR334>

Ilya Sukhar, 2013, The Future of Parse, Parse Inc., Published 25.4.2013, Read 1.1.2014,
<http://blog.parse.com/2013/04/25/the-future-of-parse/>

Lawrence Pingree, 2013, Bring Your Own Device: Mobile Trends and Securing the Transition, Gartner Inc., Published 15.1.2013, Read 27.12.2013.

<http://my.gartner.com/portal/server.pt?open=512&objID=202&mode=2&PageID=5553&ref=webinar-rss&resId=2272223&srcId=1-2920760825>
http://www.gartner.com/it/content/2272200/2272223/january_15_bring_your_own_device_lpingree.pdf

MarketsAndMarkets 2012, Backend as a Service (BaaS) Market worth \$7.7 Billion by 2017, MarketsAndMarkets, Published October 2012, Read 28.12.2013
<http://www.marketsandmarkets.com/PressReleases/baas.asp>

Michael Facemire, 2012, Mobile Backend-as-a-Service: The New Lightweight Middleware?, Forrester Research, Published 25.4.2012, Read 28.12.2013,
http://blogs.forrester.com/michael_facemire/12-04-25-mobile_backend_as_a_service_the_new_lightweight_middleware

Michael King, John Jackson, 2013, Q4 2013 MOBILE TRENDS REPORT, Appcelerator and IDC, Read 1.1.2014, <http://www.appcelerator.com.s3.amazonaws.com/pdf/q4-2013-devsurvey.pdf>

Microsoft Azure, Developer tools for Mobile Services, Microsoft Inc., Read 12.3.2014,
<http://www.windowsazure.com/en-us/develop/mobile/developer-tools/>

Microsoft Azure, Documentation, Microsoft Inc., Read 22.2.2014,
<http://www.windowsazure.com/en-us/documentation/>

Microsoft Azure, How to use table storage, Microsoft Inc., Read 11.3.2014,
<http://www.windowsazure.com/en-us/documentation/articles/storage-dotnet-how-to-use-table-storage-20/#what-is>

Microsoft Azure, Microsoft Azure Free Trial: Try Azure, Microsoft Inc., Read 11.4.2014,
<http://azure.microsoft.com/en-us/pricing/free-trial/>

Microsoft Azure, Mobile Apps, Microsoft Inc., Read 3.2.2014,
<http://www.windowsazure.com/en-us/solutions/mobile/>

Microsoft Azure, Mobile Services Pricing Details, Microsoft Inc, Read 8.2.2014,
<http://www.windowsazure.com/en-us/pricing/details/mobile-services/>

Microsoft Azure Mobile Services REST API Reference, Microsoft Inc., Published 17.7.2013, Read 8.2.2014,
<http://msdn.microsoft.com/en-us/library/windowsazure/jj710108.aspx>

Microsoft Azure, Mobile Services server script reference, Microsoft Inc., Published 10.10.2013, Read 8.2.2014,
<http://msdn.microsoft.com/en-us/library/windowsazure/jj554226.aspx>

Microsoft Azure, Notification Hubs Overview, Microsoft Inc., Published 22.1.2014, Read 8.2.2014, <http://msdn.microsoft.com/en-us/library/windowsazure/jj927170.aspx>

Microsoft Azure, Pricing Calculator, Microsoft Inc., Read 22.2.2014,
<http://www.windowsazure.com/en-us/pricing/calculator/>

Microsoft Azure, Routing and Tag Expressions, Microsoft Inc., Read 18.3.2014, <http://msdn.microsoft.com/en-us/library/windowsazure/dn530749.aspx>

Microsoft Azure, Schedule Backend Tasks with Scheduler - Mobile Services, Microsoft Inc., Read 15.3.2014, <http://www.windowsazure.com/en-us/documentation/articles/mobile-services-schedule-recurring-tasks/>

Microsoft Azure, Solutions, Microsoft Inc., Read 28.1.2014, <http://www.windowsazure.com/en-us/solutions/>

Microsoft Azure, Storage Service, Microsoft Inc., Read 11.3.2014, <http://www.windowsazure.com/en-us/services/storage/>

Microsoft Azure, Store server scripts in source control, Microsoft Inc., Read 5.2.2014, <http://www.windowsazure.com/en-us/documentation/articles/mobile-services-store-scripts-source-control/>

Microsoft Azure, Templates, Microsoft Inc., Read 18.3.2014, <http://msdn.microsoft.com/en-us/library/windowsazure/dn530748.aspx>

Microsoft Azure, Upcoming Name Change for Windows Azure, Microsoft Inc., Read 27.3.2014, <http://blogs.msdn.com/b/windowsazure/archive/2014/03/25/upcoming-name-change-for-windows-azure.aspx>

Microsoft Azure, Work with Data, Microsoft Inc., Published 25.1.2014, Read 4.2.2014, <http://msdn.microsoft.com/en-us/library/windowsazure/jj631634.aspx>

Microsoft Unveils Windows Azure at Professional Developers Conference, Microsoft Inc., Published 27.10.2008, Read 28.1.2014, <https://www.microsoft.com/en-us/news/press/2008/oct08/10-27pdcdelay1pr.aspx>

node.js, Read 4.2.2014, <http://nodejs.org/>

Rob van der Meulen, Janessa Rivera, 2013, Gartner Says By 2016, 40 Percent of Mobile Application Development Projects Will Leverage Cloud Mobile Back-End Services, Gartner Inc., Published 30.4.2013, Read 28.12.2013 <http://www.gartner.com/newsroom/id/2463615>

Rob van der Meulen, Janessa Rivera, 2013, Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013, Gartner Inc., Published 14.11.2013, Read 27.12.2013. <http://www.gartner.com/newsroom/id/2623415>

Windows Apps Team. 2013, Delivering push notifications to millions of devices with Windows Azure Notification Hubs, Microsoft Inc., Published 16.9.2013, Read 8.2.2014, <http://blogs.windows.com/windows/b/appbuilder/archive/2013/09/16/delivering-push-notifications-to-millions-of-devices-with-windows-azure-notification-hubs.aspx>

APPENDICES

Appendix 1. An example code that uses Amazon DynamoDB table from Windows Phone mobile application.

In this appendix it is possible to see an example code how AWS SDK can be used in the Windows Phone mobile application to communicate with Amazon DynamoDB table. In this example Windows Phone mobile application will contain a page with four buttons: insert, update, get and delete. By pressing insert button it is possible to insert new note to notes data table which is located in the notes mobile service. By pressing update button it is possible to update note in the notes data table. By pressing get button it is possible to get note from the notes data table. By pressing delete it is possible to delete a note from the notes data table.

In the FIGURE 1 it is possible to see C# class NotesAmazonDynamoDBAdapter which is used to insert note to Amazon DynamoDB table, get a note from the Amazon DynamoDB table, update note in the Amazon DynamoDB table and delete note from the Amazon DynamoDB table. The NotesAmazonDynamoDBAdapter needs to be constructed with an instance of AmazonDynamoDBClient class. The AmazonDynamoDBClient is instantiated in the App.xaml.cs file of the notes mobile application. In the FIGURE 2 it is possible to see how AmazonDynamoDBClient is instantiated in practice. AmazonDynamoDBClient is instantiated in this example by giving following parameters: awsAccessKeyId, awsSecretAccessKey and region. In practice the awsAccessKeyId and awsSecretAccessKey are used to authenticate and authorize the example mobile application in the Amazon. Region is provide for the AmazonDynamoDBClient to define which AWS region will received the requests. In this example EUWest1 region is used. The AmazonDynamoDBClient class is defined in the AWS SDK. In order to communicate with a table in the Amazon DynamoDB, it is required to create a class that will represent the table in the Amazon DynamoDB. In the FIGURE 3 it is possible to see the definition of the Notes class which represent notes table in the Amazon DynamoDB. It is possible to insert note to Amazon DynamoDB table or update note in the Amazon DynamoDB table by creating a DynamoDBContext and using SaveAsync method of it. The SaveAsync method will take in as parameter the note that will be created or updated. It is possible to get note from the Amazon DynamoDB by creating a DynamoDBContext and using LoadAsync method of it. The LoadAsync method is generic and will required a

type which represent the table in the Amazon DynamoDB. The LoadAsync method will also require as a parameter the hash key of the row that will be loaded from the Amazon DynamoDB table. It is possible to delete a note from the Amazon DynamoDB by creating a DynamoDBContext and using DeleteAsync method of it. The DeleteAsync method will require as a parameter the note that will be deleted. In the FIGURE 4 it is possible see how NotesAmazonDynamoDBAdapter is instantiated and used in the main page of the Notes mobile application.

```
using Amazon.DynamoDBv2;
using Amazon.DynamoDBv2.DataModel;
using MEng.Examples.Notes;
using System.Threading.Tasks;

namespace NotesWindowsPhone
{
    class NotesAmazonDynamoDBAdapter
    {
        private readonly AmazonDynamoDBClient _client;

        public NotesAmazonDynamoDBAdapter(AmazonDynamoDBClient client)
        {
            _client = client;
        }

        public async Task Insert(NotesDynamoDBModel note)
        {
            using(var context = new DynamoDBContext(_client))
            {
                await context.SaveAsync(note);
            }
        }

        public async Task<NotesDynamoDBModel> Get(string noteId)
        {
            using (var context = new DynamoDBContext(_client))
            {
                return await context.LoadAsync<NotesDynamoDBModel>(noteId);
            }
        }

        public async Task Update(NotesDynamoDBModel note)
        {
            using (var context = new DynamoDBContext(_client))
            {
                await context.SaveAsync(note);
            }
        }

        public async Task Delete(NotesDynamoDBModel note)
        {
            using (var context = new DynamoDBContext(_client))
            {
                await context.DeleteAsync(note);
            }
        }
    }
}
```

FIGURE 1. NotesAmazonDynamoDBAdapter C# class which is used to communicate with notes table in the Amazon DynamoDB.

```

using Amazon;
using Amazon.DynamoDBv2;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using Microsoft.WindowsAzure.MobileServices;
using NotesWindowsPhone.Resources;
using System;
using System.Diagnostics;
using System.Windows;
using System.Windows.Markup;
using System.Windows.Navigation;

namespace NotesWindowsPhone
{
    public partial class App : Application
    {
        /// <summary>
        /// Provides easy access to the root frame of the Phone Application.
        /// </summary>
        /// <returns>The root frame of the Phone Application.</returns>
        public static PhoneApplicationFrame RootFrame { get; private set; }

        public static AmazonDynamoDBClient DynamoDBClient = new AmazonDynamoDBClient(
            "AKIAIQKGJBTSL3TWLSZA",
            "cwD0k/GV092HZFnh6W8U/+uxNB6s511sQDvwcdit",
            RegionEndpoint.EUWest1);
    }
}

```

FIGURE 2. Amazon DynamoDB client defined in the App.xaml.cs file.

```

using Amazon.DynamoDBv2.DataModel;

namespace NotesWindowsPhone
{
    [DynamoDBTable("Notes")]
    public class NotesDynamoDBModel
    {
        [DynamoDBHashKey("Notes ID")] // hash key
        public string Id { get; set; }
        [DynamoDBProperty]
        public string Text { get; set; }
    }
}

```

FIGURE 3. NotesDynamoDBModel class which represents notes table in the Amazon DynamoDB.

```
using MEng.Examples.Notes;
using Microsoft.Phone.Controls;
using System;
using System.Windows;

namespace NotesWindowsPhone
{
    public partial class MainPage : PhoneApplicationPage
    {
        private readonly NotesAmazonDynamoDBAdapter _notesService;
        private NotesDynamoDBModel _note;

        public MainPage()
        {
            InitializeComponent();
            _notesService = new NotesAmazonDynamoDBAdapter(App.DynamoDBClient);
        }

        private async void ButtonInsert_Click(object sender, RoutedEventArgs e)
        {
            _note = new NotesDynamoDBModel
            {
                Id=Guid.NewGuid().ToString(),
                Text = "Inserted new note"
            };
            await _notesService.Insert(_note);
        }

        private async void ButtonUpdate_Click(object sender, RoutedEventArgs e)
        {
            _note.Text = "Updated note";
            await _notesService.Update(_note);
        }

        private async void ButtonGet_Click(object sender, RoutedEventArgs e)
        {
            _note = await _notesService.Get(_note.Id);
        }

        private async void ButtonDelete_Click(object sender, RoutedEventArgs e)
        {
            await _notesService.Delete(_note);
        }
    }
}
```

FIGURE 4. Code behind of the Notes mobile application main page.

Appendix 2. An example code that uses Google Cloud Datastore from Android mobile application.

In this appendix it is possible to see an example code how Google Cloud Datastore can be used from Android mobile application. In this example Android mobile application will contain a page with four buttons: insert, update, get and delete. By pressing insert button it is possible to insert new note entity to the Google Cloud Datastore. By pressing update button it is possible to update note entity in the Google Cloud Datastore. By pressing get button it is possible to get note entity from the Google Cloud Datastore. By pressing delete it is possible to delete a note entity from the Google Cloud Datastore.

In the FIGURE 1 it is possible to see an example code how new note entity can be inserted to Google Cloud Datastore. The example code in the FIGURE 1 will be executed when insert button is pressed in the Android mobile application. In the previously mentioned example code, new note is created using `CloudEntity` class as a first thing. The `CloudEntity` class is defined in the Cloud Backend API of the Mobile Backend Starter. In the example code the new note will be given a property called “Text” and value of it will be set to “Inserted a note”. After the new note is created, a handler will be created for handling response from the Google Cloud Datastore. In the handler is handled what should be done when the note has been inserted to Google Cloud Datastore (`onComplete`) and what should be done when the insert of note will fail (`onError`). In this example when the note has been inserted it will be stored to `mNote` instance variable. It is possible to see in the FIGURE 2 the definition of the `mNote` instance variable. In this example nothing will be done in case the insert of note will fail. As a last thing in this example the new note will be inserted asynchronously to the Google Cloud Datastore. In this example the creation of backend class (`CloudBackendAsync`) is done in own method (`getBackend`). It is possible to see the definition of the `getBackend` method in the FIGURE 3. The `CloudBackendAsync` class is defined in the Cloud Backend API of the Mobile Backend Starter. The `CloudBackendAsync` class will require in the constructor the context of the current mobile application. It is possible to get the context of the current mobile application for example by calling `getApplicationContext` method.

```

public void onInsertClick(View view) {
    mNote = new CloudEntity("Notes");
    mNote.put("Text", "Inserted a note");

    CloudCallbackHandler<CloudEntity> handler = new CloudCallbackHandler<CloudEntity>() {
        @Override
        public void onComplete(final CloudEntity result) {
            mNote = result;
        }

        @Override
        public void onError(final IOException exception) {
        }
    };

    getBackend().insert(mNote, handler);
}

```

FIGURE 1. Code example for inserting a new note entity to the Google Cloud Datastore.

```

public class NotesActivity extends Activity {

    private CloudEntity mNote;
}

```

FIGURE 2. Instance variable mNote that will be used in the Android mobile application example to store a note in memory.

```

private CloudBackendAsync getBackend()
{
    return new CloudBackendAsync(this.getApplicationContext());
}

```

FIGURE 3. Code example how to create the backend class for communicating with Google Cloud Datastore.

In the FIGURE 4 it is possible to see an example code how note entity can be retrieved from the Google Cloud Datastore. It is possible to use get method from the CloudBackendAsync class for retrieving an entity from the Google Cloud Datastore. In the FIGURE 5 it is possible to see an example code how note entity can be updated in the Google Cloud Datastore. It is possible to use update method from the CloudBackendAsync class for updating a note entity in the Google Cloud Datastore. In the FIGURE 6 it is possible to see an example code how note entity can be removed from the Google Cloud Datastore. It is possible to use delete method of the CloudBackendAsync class for removing a note from the Google Cloud Datastore.

```

public void onGetClick(View view) {
    CloudEntity entityQuery = new CloudEntity("Notes");
    entityQuery.setId(mNote.getId());
    CloudCallbackHandler<CloudEntity> handler = new CloudCallbackHandler<CloudEntity>() {
        @Override
        public void onComplete(final CloudEntity result) {
            mNote = result;
        }

        @Override
        public void onError(final IOException exception) {
        }
    };
    getBackend().get(entityQuery, handler);
}

```

FIGURE 4. Code example for retrieving a note entity from the Google Cloud Datastore.

```

public void onUpdateClick(View view) {
    mNote.put("Text", "Updated a note");
    CloudCallbackHandler<CloudEntity> handler = new CloudCallbackHandler<CloudEntity>() {
        @Override
        public void onComplete(final CloudEntity result) {
            mNote = result;
        }

        @Override
        public void onError(final IOException exception) {}
    };
    getBackend().update(mNote, handler);
}

```

FIGURE 5. Code example for updating a note entity in the Google Cloud Datastore.

```

public void onDeleteClick(View view) {
    CloudCallbackHandler<Void> handler = new CloudCallbackHandler<Void>() {
        @Override
        public void onComplete(Void results) {
            mNote = null;
        }
    };
    getBackend().delete(mNote, handler);
}

```

FIGURE 6. Code example for removing a note entity from the Google Cloud Datastore.

Appendix 3. An example code that uses Microsoft Azure mobile services SDK in Windows Phone mobile application.

In this appendix it is possible to see an example code how Microsoft Azure mobile service SDK can be used in the Windows Phone mobile application. In this example Windows Phone mobile application will contain a page with four buttons: insert, update, get and delete. By pressing insert button it is possible to insert new note to notes data table which is located in the notes mobile service. By pressing update button it is possible to update note in the notes data table. By pressing get button it is possible to get note from the notes data table. By pressing delete it is possible to delete a note from the notes data table.

In the FIGURE 1 it is possible to see C# class `NotesMobileServiceAdapter` which is used to insert note to data table, get a note from the data table, update note in the data table and delete note from the data table. The `NotesMobileServiceAdapter` needs to be constructed with an instance of `MobileServiceClient` class. The `MobileServiceClient` is instantiated in the `App.xaml.cs` file of the notes mobile application. In the FIGURE 2 it is possible to see how `MobileServiceClient` is instantiated in practice. `MobileServiceClient` is instantiated in this example by giving following parameters: `applicationUrl` and `applicationKey`. The `applicationUrl` is in practice the location of the mobile service. The `applicationKey` needs to match with the application key defined in the mobile service. The `MobileServiceClient` is a class that is defined in the Microsoft Azure mobile services SDK. It is possible to connect to a data table by using `GetTable` method of the `MobileServiceClient` class. The `GetTable` method is a generic and needs a type that represents the data table in the mobile service. In this example `Notes` class is used to represent the notes data table in the mobile service. In the FIGURE 3 it is possible to see the definition of the `Notes` class. It is possible to insert note to notes data table by using `InsertAsync` method of the `MobileServiceClient` class. It is possible to get a note from the notes data table by using `LookupAsync` method of the `MobileServiceClient` class. It is possible to update a note in the notes data table by using `UpdateAsync` method of the `MobileServiceClient` class. It is possible to delete note from the notes data table by using `DeleteAsync` method of the `MobileServiceClient` class. In the FIGURE 4 it is possible see how `NotesMobileServiceAdapter` is instantiated and used in the main page of the Notes mobile application.

```

using Microsoft.WindowsAzure.MobileServices;
using System.Threading.Tasks;
using MEng.Examples.Notes;

namespace MEng.Examples.Notes
{
    class NotesMobileServiceAdapter : INotesService
    {
        private readonly MobileServiceClient _client;

        public NotesMobileServiceAdapter(MobileServiceClient mobileServiceClient)
        {
            _client = mobileServiceClient;
        }

        public Task Insert(Notes note)
        {
            return NotesTable.InsertAsync(note);
        }

        public Task<Notes> Get(string noteId)
        {
            return NotesTable.LookupAsync(noteId);
        }

        public Task Update(Notes note)
        {
            return NotesTable.UpdateAsync(note);
        }

        public Task Delete(Notes note)
        {
            return NotesTable.DeleteAsync(note);
        }

        private IMobileServiceTable<Notes> NotesTable
        {
            get { return _client.GetTable<Notes>(); }
        }
    }
}

```

FIGURE 1. NotesMobileServiceAdapter C# class which is used to communicate with notes data table in the Microsoft Azure mobile service.

```

using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using Microsoft.WindowsAzure.MobileServices;
using NotesWindowsPhone.Resources;
using System;
using System.Diagnostics;
using System.Windows;
using System.Windows.Markup;
using System.Windows.Navigation;

namespace NotesWindowsPhone
{
    public partial class App : Application
    {
        /// <summary>
        /// Provides easy access to the root frame of the Phone Application.
        /// </summary>
        /// <returns>The root frame of the Phone Application.</returns>
        public static PhoneApplicationFrame RootFrame { get; private set; }

        public static MobileServiceClient MobileService = new MobileServiceClient(
            "https://notes-mbaas.azure-mobile.net/",
            "UrnxKEVJBdmXFQkBD1rNBU0xFDMkzWp35"
        );

        /// <summary>
        /// Constructor for the Application object.
        /// </summary>
        public App()
        {

```

FIGURE 2. Mobile Service client defined in the App.xaml.cs file.

```

using Newtonsoft.Json;

namespace MEng.Examples.Notes
{
    public class Notes
    {
        public string Id { get; set; }

        [JsonProperty(PropertyName = "text")]
        public string Text { get; set; }
    }
}

```

FIGURE 3. Notes class which represents a data table in the mobile service.

```

using MEng.Examples.Notes;
using Microsoft.Phone.Controls;
using System.Windows;

namespace NotesWindowsPhone
{
    public partial class MainPage : PhoneApplicationPage
    {
        private readonly INotesService _notesService;
        private Notes _note;

        public MainPage()
        {
            InitializeComponent();
            _notesService = new NotesMobileServiceAdapter(App.MobileService);
        }

        private async void ButtonInsert_Click(object sender, RoutedEventArgs e)
        {
            _note = new Notes { Text = "Inserted new note" };
            await _notesService.Insert(_note);
        }

        private async void ButtonUpdate_Click(object sender, RoutedEventArgs e)
        {
            _note.Text = "Updated note";
            await _notesService.Update(_note);
        }

        private async void ButtonGet_Click(object sender, RoutedEventArgs e)
        {
            _note = await _notesService.Get(_note.Id);
        }

        private async void ButtonDelete_Click(object sender, RoutedEventArgs e)
        {
            await _notesService.Delete(_note);
        }
    }
}

```

FIGURE 4. Code behind of the Notes mobile application main page.