

# Learning Windows Azure Mobile Services for Windows 8 and Windows Phone 8

**Geoff Webber-Cross**



## Chapter No. 1 "Preparing the Windows Azure Mobile Services Portal"

## In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.1 "Preparing the Windows Azure Mobile Services Portal"

A synopsis of the book's content

Information on where to buy this book

## About the Author

**Geoff Webber-Cross** has commercial and personal experience of developing Windows 8 and Windows Phone applications and using Azure for websites, mobile services, web services, and Windows services. He enjoys learning about new technologies and solving difficult software problems.

---

I'd like to thank my wife for putting up with me tapping away on my laptop every night for months on end while writing this book.

---

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)

# Learning Windows Azure Mobile Services for Windows 8 and Windows Phone 8

Windows Azure offers a wide range of cloud-based services, which are hosted on a robust, well-managed infrastructure, and can be easily scaled to meet our business demands. Windows Azure Mobile Services is a fantastic member of the Azure family, which allows mobile developers to quickly build web-connected applications and enhance user experience with push notifications.

Using traditional web technology, we will need to think about creating databases and web services, deciding what security mechanisms to use; build tools to administer the data and services; and write backend services to interface with the different Push Notification Service providers we want to use.

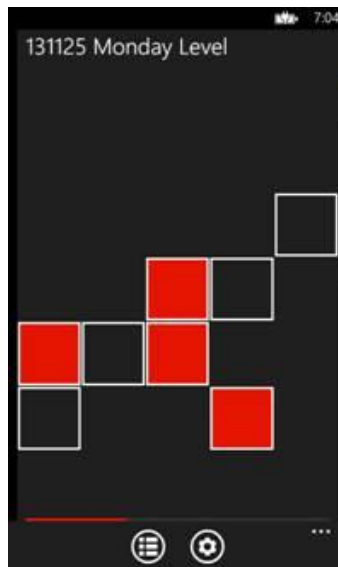
With Windows Azure Mobile Services, we can build model-first services without touching the database schema, get a fully managed and REST API for our data without writing a line of code, and modify the database API methods using scripts. Using scripts, we can also create API methods to access data, send push notifications, and make HTTP requests.

This book aims to investigate all that Windows Azure Mobile Services has to offer with practical examples, which can be used in real applications. Also, it covers areas of application development to enhance user experience, help with store certification, and improve development efficiency. I've created a simple game named TileTapper in C#/XAML for Windows 8 and Windows Phone to help illustrate use cases for all the service features and keep the book real!

The TileTapper game consists of a grid board seeded from a simple Boolean array of active or inactive tiles. When the app launches, it prompts the user to log in using the Windows Live connect authentication provider, downloads levels and current high score from our backend service, and then begins the game. The phone app game grid looks like the following screenshot.

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)



The user has to tap on all the tiles before the time runs out to complete the level. The score at the end of the game as well as high score are stored in the service, if needed. Levels are generated automatically using a scheduled script and push notifications are sent about new high scores achieved and new levels created. Both apps have settings pages for managing notifications.

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)

## What This Book Covers

*Chapter 1, Preparing the Windows Azure Mobile Services Portal*, explains how to choose a subscription, set up an Azure account, and create a Mobile Service; it also talks about the current Mobile Services portal features.

*Chapter 2, Start Developing with Windows Azure Mobile Services*, covers what software and hardware you need to develop Windows 8 and Windows Phone 8 apps using Windows Azure Mobile Services. We'll also learn about creating preconfigured apps from the portal and connecting existing apps from scratch.

*Chapter 3, Securing Data and Protecting the User*, looks at permission options for tables and APIs and different authentication methods for protecting our data and users' personal information. We'll also look at developing code to log in users with an authentication provider and storing their credentials for subsequent app usage.

*Chapter 4, Service Customization with Scripts*, covers customizing scripts to perform validation, manipulate data, and make HTTP requests. We'll also look at installing a Node npm package and using it in our scripts and finally, using the Git version control to pull a copy of our scripts to work locally and as a backup.

*Chapter 5, Implementing Push Notifications*, explains configuring Windows Store and Windows Phone 8 apps to implement push notifications; create a channels table to manage push channel URIs; and send Tile, Toast, and Badge notifications using the MPNS (Windows Phone) and WNS (Windows Store) providers.

*Chapter 6, Scaling Up with the Notifications Hub*, looks at the benefits of using the Notifications Hub from the service bus family of services, building on *Chapter 5, Implementing Push Notifications*. We adapt our code to register the push channel URI with the Notifications Hub, create scripts for sending notifications using the Azure for Node SDK, and use the Windows Azure Service Bus SDK to send notifications from .NET backend services.

*Chapter 7, Best Practices for Web-connected Apps*, looks at what we need to do to get our apps certified with respect to our Windows Azure Mobile Services implementation. We'll look at the app certification requirements for the Windows Store and UX guidelines and then talk about privacy statements, checking the cost impact of using the Internet connection and managing push notifications.

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)

# 1

## Preparing the Windows Azure Mobile Services Portal

Before we get down to any coding or even looking at development tools, we need to do some work on getting things prepared in Windows Azure. In this chapter, we're going to talk about the following:

- Choosing a pricing plan for services you wish to implement
- Creating a Windows Azure account that allows you to use any Windows Azure services
- Creating our first mobile service
- Exploring the Mobile Service portal

To use Windows Azure Services and create application store accounts, you're going to need a Microsoft account (formerly known as Microsoft Live ID). If you haven't already got one, go and create one here <https://signup.live.com/signup.aspx>.

### Choosing a subscription

To get started, go to <http://www.windowsazure.com> and first check out the pricing options; there will be a **PRICING** tab and a **Mobile Services** option under the **COMPUTE** header. Take a look at the pricing calculator for **mobile services** at <http://www.windowsazure.com/en-us/pricing/calculator/?scenario=mobile> and have a quick look to make sure you have an idea of how much the services you want to use might cost. If you don't know what you want, just sign up for a free account.

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)

## Pay-as-you-go subscription

Small apps and a bit of experimentation are unlikely to cost you anything. At the time of writing this, you get the following for free. But check for yourself so that you're not in for a nasty surprise if you sign up for a **Pay-as-you-go (PAYG)** account and exceed your usage:

- 10 Mobile Services
- 20 MB SQL database for 12 months
- 500 K API calls per month
- Send push notifications via the Notification Hubs to up to 500 active devices
- 1,00,000 Notification Hubs operations per month

Throughout the book, I'll try to point out where you need to be careful to make sure you don't start incurring costs if you want to maintain free service usage.

## Basic and Standard subscriptions

**Basic** and **Standard** subscriptions need you to buy units (service instances) for the number of API calls you expect to make. If you can calculate how many API calls your apps are likely to be making and how much storage you need, you can decide if either of these subscriptions will be the most economical for you.

## Free trial

The free trial allows you to use 200 USD worth of any services (not just mobile) you like per month.

## Creating a Windows Azure account

If you already have a Windows Azure account, skip to the next section; otherwise, click on the **Portal** tab (<https://manage.windowsazure.com/>). It will take you to log in using your Microsoft account if you are not already logged in. Once you have logged in, you will see a page saying you have no subscription. Click on the **SIGN UP FOR WINDOWS AZURE** link, <https://account.windowsazure.com/SignUp>. You should end up on the **Sign up** page (There are a number of routes to get to this page through the website, but this seemed to be the least clicks for me!). Your personal details should appear on your details in the account info page and you'll need to verify it with an SMS message or a call verification:

The screenshot shows the Windows Azure sign-up process. On the left, there's a 'Sign up' sidebar with a 'Free Trial' link. The main content area is titled 'Windows Azure' and shows a user profile with an '@outlook.com' email. The process is broken down into four steps:

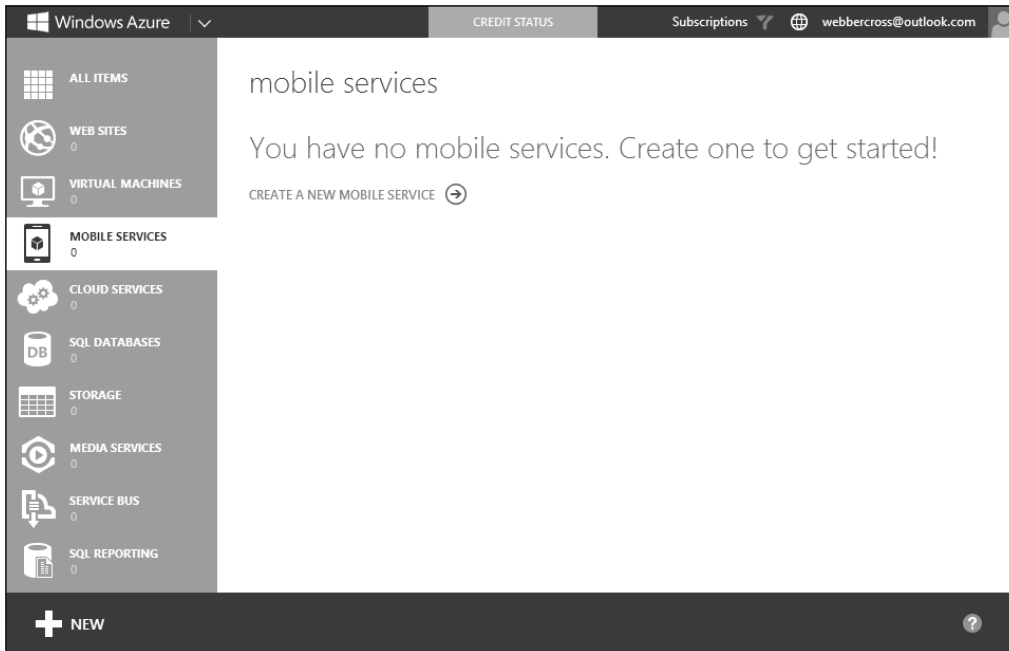
- About you:** Fields for First Name (Geoff), Last Name (Webber-Cross), Country/Region (United Kingdom), VATID (Optional), Contact Email (@outlook.com), and Company Name (Optional).
- Mobile verification:** Radio buttons for 'Send text message' (selected) and 'Call me'. A dropdown menu shows 'United Kingdom (+44)'. Below is a phone number field and buttons for 'Send text message' and 'Verify code'.
- Payment information:** This section is currently empty.
- Agreement:** Two checkboxes: 'I agree to the Windows Azure Agreement, Offer Details, and Privacy Statement.' and 'Microsoft may use my email and phone to provide special Windows Azure offers.'

Once verified, you can enter your credit card details. You can also sign up for a free trial or a pay-as-you-go account. Don't panic, you don't get automatically signed up for any premium subscriptions; however, 1 USD will be charged to your credit card for verification. Accept the agreement and click on the **Purchase** button, your card details will be validated and you will be taken to the subscriptions page where you'll be pleased to find you already have a free trial! From here, you can add subscriptions to meet your own requirements. If you have chosen a trial subscription, there is a spending limit feature so you don't incur any costs; once you reach the offer limits, services will be disabled and data will be available as read only.



## Creating a mobile service

Now we've got all the boring sign up stuff out of the way, we can get to the bit we're interested in. Go to the portal (<https://manage.windowsazure.com>) and it's probably a good idea to bookmark the page in your browser as we'll be here quite a bit. The portal should look something like the following screenshot, displaying all the Windows Azure services on the left available to us:



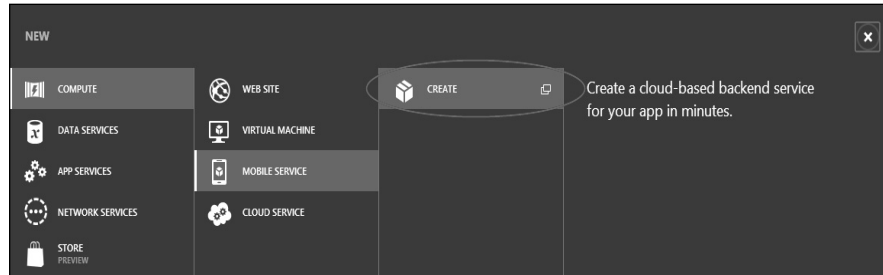
The Windows Azure portal offers a plethora of services, but we're obviously going to concentrate on Windows Azure Mobile Services and will touch upon Windows Azure SQL Databases and Windows Azure Service Bus when we look at the Notification Hubs.

To create a new mobile service perform the following steps:

1. Click on the **+ NEW** toolbar button shown as follows:



2. Select **CREATE** from the pop-up menu shown as follows:



3. Fill in the details for the service. I'm going to opt to use my PAYG subscription, **Create a free 20MB SQL database**, and target **North Europe**.

A screenshot of the 'NEW MOBILE SERVICE' form in the Azure portal. The form is titled 'Create a Mobile Service' and has a close button (X) in the top right corner. The form contains the following fields:

- URL: A text input field containing 'TileTapper|' and '.azure-mobile.net' below it.
- DATABASE: A dropdown menu with 'Create a free 20 MB SQL database' selected.
- SUBSCRIPTION: A dropdown menu with 'Pay-As-You-Go' selected.
- REGION: A dropdown menu with 'North Europe' selected.

At the bottom right of the form, there is a right arrow button and the number '2'.

At this point, if we choose the **Create a new SQL database** instance, we will start incurring costs for the new database. If we had already created a database, we would see this as an option too. Choose a region close to where your target audience is likely to be so that the service is hosted as close to them as possible. Mobile Services does not use affinity groups, so you have to specify a region.

- The next screen will show us options for creating a database instance:

NEW MOBILE SERVICE

### Specify database settings

NAME  
TileTapper\_db

SERVER  
New SQL database server

SERVER LOGIN NAME  
geoff

SERVER LOGIN PASSWORD  
.....

CONFIRM PASSWORD  
.....

REGION  
North Europe

CONFIGURE ADVANCED DATABASE SETTINGS

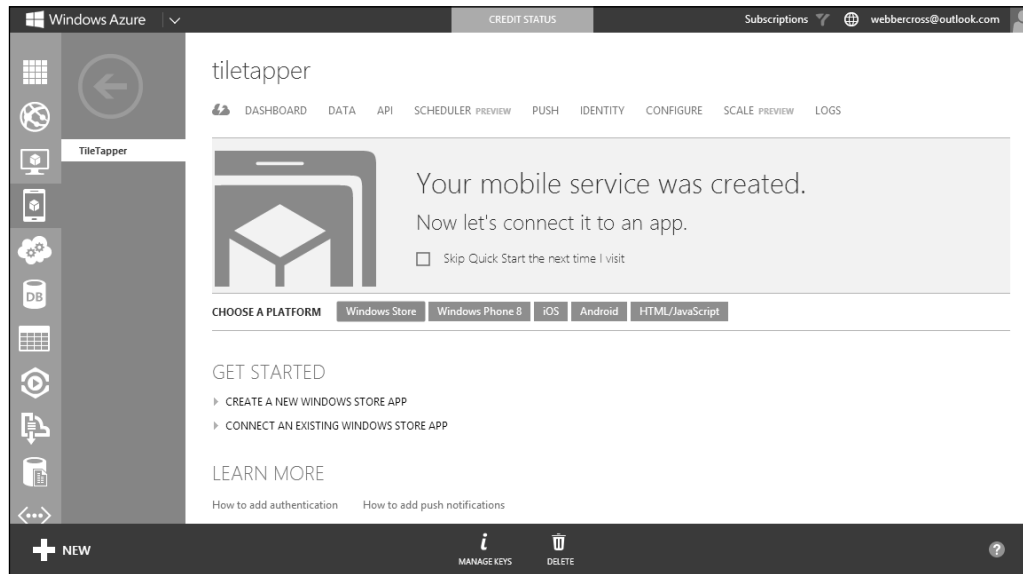
At this point, we need to choose a name for the database, set the login credentials (make a note of them for future reference), and choose a location for the database server. By default, the mobile service with the suffix `_db` is set as the database name; this is fine for me as I only want to use it for one service. However, if you don't want to spend money on more databases and want to use it for multiple applications, you may want to choose a more generic database name, something like `AppsDatabase`. It is sensible to host the database server at the same location as your mobile service instance, so that additional transfer costs are not incurred and they don't have to talk to each other across the world every time a request is made!

- I'm going to choose default database settings, but you can check **CONFIGURE ADVANCED DATABASE SETTINGS** and you will be able to change the collation of the database.

This page actually displays a message stating that we won't be charged for the database configuration we've chosen, But if you change the database size, it will become a paid database.

## Mobile Services features

Now that we've created a mobile service, we can explore the features available to us in the portal. From the main portal, select **Mobile Services** and then click on the service you have just created to navigate to the Mobile Services portal:



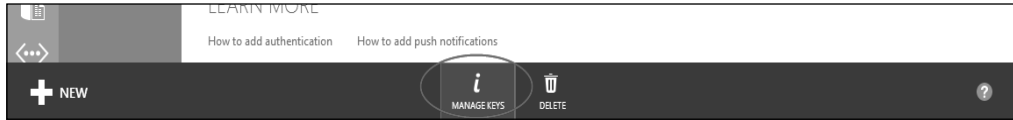
Along the top are all features available to us to help build our services and applications. The bottom toolbar is context sensitive and has actions for the selected feature.

At the time of writing this, a number of features had a **PREVIEW** tag next to them; you may also see beta and prerelease features. These features are likely to become fully supported. However, if you use them, you need to bear in mind that they may be changed or be completely removed. There is a terms of use article here, which is worth a read: <http://www.windowsazure.com/en-us/support/legal/preview-supplemental-terms/>. We'll investigate all the features, even the preview ones just for completion. When you are reading this, there are likely to be more features.

## Managing keys

Windows Azure Mobile Services have an **application key** and a **master key**, which limit access to the API. Tables and APIs can be set to only grant access to calls from application requests bearing the application key embedded in the application code. However, it is not encrypted so is not considered secure. This means it is important to authenticate users before accessing services.

The master key is used for administrator access and should not be distributed with the app. These keys can be managed from the **MANAGE KEYS** button on the bottom toolbar, which appears on the main portal and various tabs:

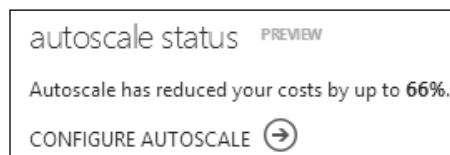


The keys can be regenerated if there has been a security compromise, but should not be changed unless absolutely necessary as it will stop all published apps from accessing services and will mean they need to be republished with the new key.

## Mobile service dashboard

This is an overview of what's going on with our service. The top section displays a chart of our API and data usage; there are filters to change the time period and y-axis scaling. The dashboard displays the following sections:

- **Mobile service endpoint status:** This allows us to monitor the availability of our service (this is only available for premium subscriptions) when endpoint monitoring has been configured. If you have a critical system, this is an important feature for making sure the service is meeting your SLA.
- **Usage overview:** This is an overview of your API call, active device, and out data usage.
- **Autoscale status:** If you have scaling enabled, this displays the current scaling status. This can be set up by clicking on the **CONFIGURE AUTO SCALE** link or going to the **Scale** tab. When enabled, the dashboard tells us about how much cost reducing scaling is being achieved (depending on the demand):



This is a round about way of saying we've got one of three possible instances running.

- **Quick glance:** This section on the right and has a quick summary of the service's current status.

- **Data:** The **Data** tab lists all the tables configured in our database, shows us an overview, and allows us to browse the data, modify the operation scripts, edit columns, and change the permissions. These features will be discussed in detail in subsequent chapters.
- **API:** The **API** tab allows us to manage custom APIs implemented in our service. Each table has a default set of operation scripts that can be modified. APIs allow us to create any operation that can make HTTP requests and perform database operations. Each API has a standard set of HTTP methods that can be implemented as required.
- **Scheduler:** From here, we can create and manage scheduled jobs that can run scripts on a timed schedule or on demand.
- **Push:** For me, this is one of the coolest features of Windows Azure Mobile Services that allows us to manage push notifications to our applications, without having to manually create and host our own services, which interface with **Windows Push Notification Services (WNS)**, **Apple Push Notification Service (APNs)**, and **Google Cloud Messaging (GCM)** service. We'll also look at the Notification Hubs, which is a more scalable way of achieving push notifications; however, it's not configured directly from the Mobile Services portal.
- **Identity:** Windows Azure Mobile Services delegates its authentication to providers such as Microsoft account, Facebook, Twitter, and Google. This means we don't need to worry about storing and managing user credentials or manually dealing with authentication mechanisms such as OAuth2. This tab is where we configure the identity provider used to authenticate our application.

## Configure

The **Configure** tab contains miscellaneous settings for Windows Azure Mobile Services as follows:

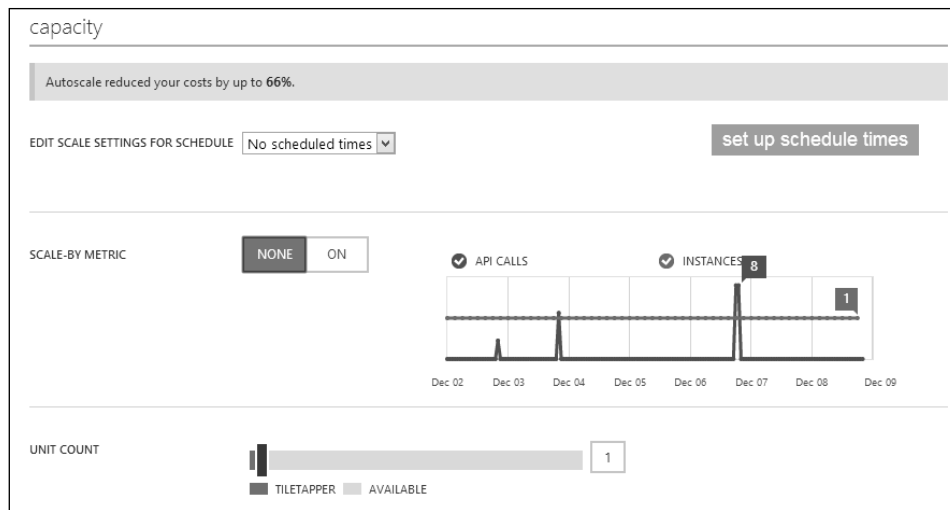
- **Database settings:** This section contains two links for configuring the database and database server that Mobile Services use. Both of these links take us out of the Mobile Services portal and into the SQL Databases portal.
- **Source control:** It's possible to manage the scripts used by the service (we'll discuss these later in the book) using Git version control, by initially pulling the repository to your machine, working locally, and then pushing back updates you have made, instead of working in the portal. Once this is set up, the dashboard displays the source control username.

- **Dynamic schema:** This setting allows you to enable or disable the **Dynamic Schema** feature. The feature allows the service to automatically add columns to tables as they appear through the API so that you don't have to constantly modify your database schema while you develop your services. It is recommended that this feature is disabled once development is finished and your app is in production.
- **Cross-origin resource sharing (CORS):** This section allows you to create a list of hosts that are permitted to interact with your mobile service (including wildcards such as `*.example.com`). Client-side JavaScript originating from hosts in the list will be granted access to the service, otherwise they will be denied. This does not affect native apps using the APIs.
- **Developer analytics:** This section allows you to set up the application performance analytics.
- **App settings:** These are key-value pair values you can use and access in scripts to help with things such as string settings, which you may want to change from the dashboard rather than in the script. This is similar to the `AppSettings` section in `web.config` and `app.config` files.
- **Monitoring:** If you have a premium subscription, up to two monitoring endpoints can be configured from here, allowing you to monitor the service availability from up to three geo-distributed locations.

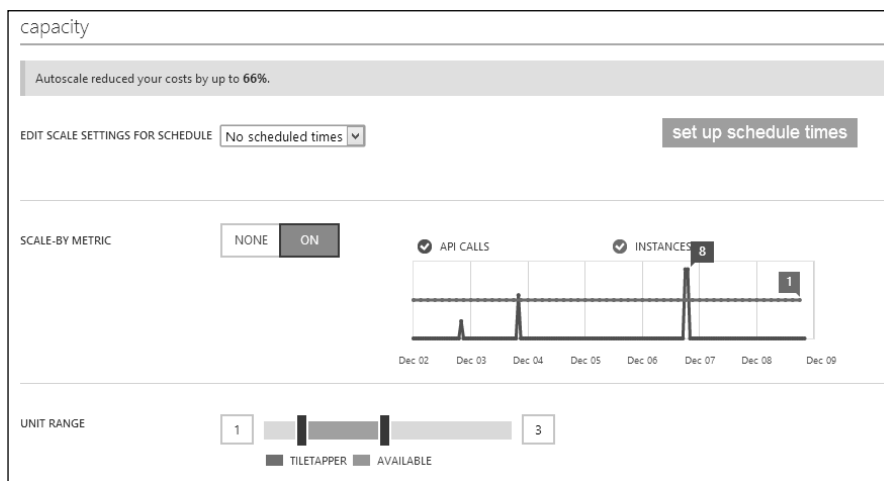
## Scale

One of the major features of Windows Azure is scalability . Windows Azure websites, web services, windows services, mobile services, and so on run in virtual machine instances managed by the **Windows Azure Fabric Controller**. This not only provides us with resilience but also allows a service to be scaled across multiple instances to meet the required capacity. We can configure the following features from here:

- **General:** This section allows us to change the **MOBILE SERVICE TIER**, which determines whether certain features can be used. In the **BASIC** and **STANDARD** mode, we can adjust the number of units in operation or auto scaling.
- **Capacity:** If we use a basic or standard service tier, we can configure the number of live units when **SCALE-BY METRIC** is set to **OFF**. These units are always active and will cost a fixed amount all the time.



**SCALE-BY METRIC** is a feature that allows the number of mobile service instances to increase and decrease automatically to meet the demand on the service. When **SCALE-BY METRIC** is set to **ON**, we can set the upper and lower unit thresholds:



With this configuration, we will incur the highest costs on peak demand when the system scales-up automatically, but it should be more economical than having a fixed number of units always active.

- **SQL Database:** Here, we can change the database capacity if required. Once we move away from the free 20 MB database, we will start incurring costs.



## Logs

The logs tab allows us to view logs created by script errors or logging during debugging. We will cover more on this in *Chapter 4, Service Customization with Scripts*.

## Summary

So far, we've chosen a subscription, signed up for a Windows Azure account, created our first Windows Azure's Mobile Service, and got a taste of what a mobile service has to offer us. Throughout the book, we'll be looking at these features in a lot more detail and learning how to use them in our applications.

In the next chapter, we're going to start setting up our development environment, get all the tools we will need, look at the portal starter solutions, and hook up an app from scratch.

## Where to buy this book

You can buy Learning Windows Azure Mobile Services for Windows 8 and Windows Phone 8 from the Packt Publishing website:  
<http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book>.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](#).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



[www.PacktPub.com](http://www.PacktPub.com)

**For More Information:**

[www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book](http://www.packtpub.com/learning-windows-azure-mobile-services-for-windows-8-and-windows-phone-8/book)