Microsoft Windows Azure cloud application development and security concepts

Cristian-Stefan Marin

Bachelor's Thesis of Degree Programme in Business Information Technology

Bachelor of Business Administration

TORNIO 2012

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Degree programme: Business Information Technology
Writer: Cristian-Stefan Marin
Thesis title: Microsoft Windows Azure cloud application development and security concepts
Pages (of which appendices): 82 (5)
Date:
Thesis instructor: Roger Niska

This thesis focuses on the area of cloud computing, an important and relevant research area in computer science. Considering the rapid evolution of cloud computing and the wide variety of cloud service providers, software developers need to stay updated on technology advancements. The focus of the thesis is on developing cloud applications.

In particular, this thesis analyzes the Microsoft Windows Azure (MWA) platform and the current services it provides. The research aims to emphasize the capabilities of the services provided by the MWA platform, whilst considering the business implications of developing cloud applications. Moreover, the research analyzes various security aspects and features implemented in the MWA platform. Finally, an application is built using open source templates and deployed into the MWA environment, illustrating the development process in MWA.

Exploratory research is the core research method employed in conducting this research. In an attempt to answer the proposed research questions and to achieve the research objectives, the data collected during the research process was primarily obtained from scientific sources such as printed or electronic literature published by established authors. Additionally, descriptive research is employed as secondary research methodology. The research is of theoretical nature, although it includes practical aspects regarding the use of the MWA platform and tools associated with it.

The main output of this thesis is to provide updated information about the MWA platform, considering that most previous research is deemed outdated or incomplete due to major improvements implemented during the year of 2012. Furthermore, recommended practices for increasing the security of MWA applications are provided. The sample application built and deployed during the research process represents a secondary output of this thesis. Further research on developing cloud applications can be conducted, as the MWA platform continues to be subject to modifications.

Keywords: cloud computing, cloud platform, cloud applications, Microsoft Windows Azure

CONTENTS

ABSTRACT

FIGURES

FIGURES

1 INTRODUCTION

The reasoning behind the topic selection is discussed in this chapter. Additionally, background information necessary for understanding the aim of the research is provided. Further, the objectives and structure of the thesis are discussed.

1.1 Motivation and background

Cloud computing is a concept that has been progressively receiving increased attention due to the popularity and advertising of cloud computing services. During the past decade, cloud computing has developed from a mere futuristic concept into reality, becoming a ubiquitous concept in the information technology (henceforth IT) field. Even though it is not a new concept itself and enough researches have been conducted in this field, cloud computing is not easily defined. Due to the broad area of services that it encompasses, cloud computing is perceived differently by different people (Krutz & Vines & Brunette 2010, 1), some even considering it a solution for every issue the IT field faces. However, this is not the case. Most businesses can benefit from adopting a cloud computing approach but cloud computing specifically addresses several crucial issues that independent software developers (henceforth ISD) and IT companies have to deal with.

Cloud computing is a concept that enables on-demand, worldwide access to shared IT resources that are managed by a cloud service provider, not by the company that employs the resources. The aim of this concept is to provide automated services for easily provisioning scalable, cheap resources in a timely manner and to minimize the need for managing these resources. Cloud computing adopts three service models, namely software as a service (henceforth SaaS), platform as a service (henceforth PaaS) and infrastructure as a service (henceforth IaaS). Furthermore, cloud environments can be deployed as either private or public models. (Mell & Grance 2011, 2.) Each combination of service and deployment model features its own benefits and targets different customers. The variety of cloud computing services sparks competition among cloud service providers, which consequently generates even wider variety in services and ultimately leads to increased benefits for customers.

Considering the growing interest for cloud computing, an increasing number of researches are conducted about how the software development field is affected by cloud computing. The assortment of cloud service solutions is astounding. Alongside with companies specializing in providing cloud services, IT industry giants such as Amazon, Microsoft and Google have developed their own cloud services solutions. A plethora of alternatives is available for every cloud service solution. The variety in services available in this field emphasizes the need for taking good decisions. Establishing the best development model is a difficult process to begin with and it becomes more troublesome as the alternatives increase in quantity. Software developers need to be well informed about their possible alternatives as the processes of taking good decisions regarding the implementation of the application and selecting suitable development tools need to be well thought through. Making a poor decision can severely impede or damage the development process of an application. (Lu & Zhang & Ruan 2007, 3-5.)

This research serves as guidance for software developers seeking information about cloud applications platforms. Specifically, the research aims to provide a detailed analysis of cloud service solutions, focusing on the Microsoft Windows Azure (henceforth MWA) platform, a cloud services platform developed by Microsoft. The main reasons behind selecting MWA as the focus of this research are the capabilities that the platform has. MWA is a cloud computing platform and it provides both PaaS and IaaS services. Unlike the solution offered by one of Microsoft's competitors and arguably the most popular cloud services provider, namely Amazon, MWA does not limit its services to only IaaS services. Rather than merely provisioning hardware resources, MWA also provides assistance in developing software applications through the tools made available. (Microsoft 2012a.) Moreover, in contrast with the cloud services solution offered by Google, MWA does not limit its development capabilities to only a few select programming languages. MWA is a highly scalable open platform which means that software developers are able to choose from a multitude of development technologies. In addition, Microsoft explicitly aims to provision MWA services to independent software vendors and enterprise developers alike (Chappell 2009, 2). This research attempts to provide sufficient information for software developers to consider the implications of employing the services provided by the MWA platform.

The need for my research is present because existing research is quickly outdated, as the MWA platform is constantly evolving. Microsoft releases significant service updates at least biannually. The MWA platform has been considerably modified since its initial public release in 2010. In fact, the utmost significant changes have been implemented during this year, when perhaps the most common method of customer-platform interaction, the management portal, has been entirely changed. Additionally, new technologies such as the PHP programming language became supported and the compute services have been expanded. Recent additions to the MWA platform are the Virtual Machines (henceforth VM), Websites and Mobile Service features that have been integrated during the year of 2012, albeit they are not yet fully implemented and tested. Furthermore, the tools associated with the MWA platform are updated as well. (Microsoft 2012b.) In addition, the official MWA documentation provided in existing research works has been updated or is no longer available as Microsoft brought changes to their websites.

1.2 Objectives

The objectives of this research are to study the technology behind the MWA platform, while also emphasizing the benefits, limitations and drawbacks of the MWA platform and tools associated with it. To accomplish the objectives, this thesis provides detailed information on current features implemented in or supported by the MWA platform. The research is beneficial for software developers interested in the MWA platform, aiding them in the process of making an informed decision of whether the capabilities of the MWA platform are suitable for their purposes. Furthermore, the research provides an analysis of the security features concerning the MWA platform, as trusting a third party company with critical business data can potentially damage the business.

As a practical output, the thesis provides a sample application, built using the Microsoft Visual Studio 2010 development environment and utilizing open source .NET templates, in order to illustrate the compute service of MWA and how to utilize the features associated with it. The sample application is deployed as a compute service, using a trial account subscription.

1.3 Structure of the thesis

To logically achieve the objectives of this research, the remainder of the thesis is structured as follows. Chapter 2 discusses the scope of the research, the research questions and the methodology used in conducting the research. Chapter 3 is an introductory chapter, as it provides necessary information about cloud computing and a brief analysis of the major providers of cloud computing services. The main topic of the research is discussed in chapter 4, as the chapter is dedicated for the analysis of the technology behind the MWA platform. Furthermore, chapter 4 gives suggestions as to how software developers can employ the services provided by the MWA platform. Additionally, a sample application is built using open source templates. Chapter 5 focuses on the security of the MWA environment, presenting various security features that are implemented in the MWA platform. Finally, chapter 6 concludes the research and presents the main findings of this research, while also offering suggestions for further research.

2 RESEARCH TOPIC, QUESTIONS AND METHODOLOGY

The research work explores the MWA platform for developing cloud applications and its utilization. The research also describes the concept of cloud computing and provides recommended security measures regarding cloud computing environments. The research provides background information on how cloud computing environments have functioned and evolved throughout computer history and more extensive information on current security vulnerabilities, threats, attacks, and prevention and protection methods that concern the MWA platform. Furthermore, this research discusses the process of developing software applications designed to be implemented and run in a cloud environment, using the MWA platform and associated tools. More specifically, this research focuses on the technology involved in the MWA platform and on how to operate the services associated with this specific platform. The research provides on information regarding professional use of cloud computing environments, from the point of view of ISD and businesses.

The outcome of the research is to describe how the MWA development platforms functions and to provide security guidelines and recommendations regarding the security of data stored in the MWA cloud environment. The research provides detailed technical recommendations that can assist in decision making. Finally, another outcome of the research is to develop and deploy a sample application using open source templates, to serve as an illustration of the development and deployment process. Furthermore, sections of the research can be applied to developing software applications in general, not only for MWA.

2.1 Research questions

Based on the objectives of this research, the research questions and the techniques which are used to answer them are further detailed below.

1. How are MWA cloud applications designed and developed?

The MWA platform technology and services are analyzed, explored and described, based on literature review and based on my experience with the MWA platform. MWA includes two areas of functionality: compute services and storage services (Dudley & Duchene 2010, 17). Both compute and storage services are discussed in this thesis. Furthermore, the research describes and discusses various aspects that need to be taken into consideration when designing and developing applications that are implemented in a cloud environment, using the MWA platform. Different design and development tools and practices are discussed.

2. How secure is the MWA cloud computing environment?

The concept of security regarding the MWA cloud computing environment is discussed. The research discusses several security vulnerabilities, threats and attacks, methods to improve security and how to address specific security challenges. Furthermore, the Microsoft Security Development Lifecycle (henceforth SDL) is discussed.

3. What are the recommended practices that need to be considered when designing and developing applications using MWA?

Several recommended practices and techniques for securing applications designed to be implemented into a cloud environment, through the use of the MWA platform, are researched upon and described throughout the thesis. The answer to this research question is not comprised into a specific chapter, instead recommendations are written in the chapter relevant to the features the recommendations apply to.

2.2 Research methodology

The methodology used in conducting the research includes descriptive and exploratory research based on literature analysis. Descriptive research is appropriate for this research as cloud computing security and Microsoft Windows Azure platform are not entirely new concepts themselves but do require further understanding and questioning, as both are evolving.

According to Sachdeva (2009, 15), the objective of descriptive research, also known as statistical research, is to describe things. This type of research describes the characteristics of the subject studied and provides answers to questions such as how, what, who. The description is precise and accurate. (Sachdeva 2009, 15.) This type of research is a "fact-finding investigation" (Krishnaswami & Satyaprasad 2010, 12). It is designed to gather detailed information about a particular subject and therefore it is relevant for the purpose of this research. Due to the theoretical nature of the research, descriptive research that relies on literature review as a primary source of data was conducted. Secondary data sources include journals, articles, white papers, websites or technical documentation articles.

Aside from descriptive research, the exploratory research method is used for the purposes of this research. The purpose of exploratory research is to gather information that generates solutions to existing problems and suggests new theories (Sachdeva 2009, 15). Exploratory research is conducted when the researcher has little or no knowledge about the topic in question. When conducting research work, exploratory research is considered the first stage, followed by descriptive research. (Krishnaswami & Satyaprasad 2010, 15.) The exploratory research is conducted regarding the Microsoft Windows Azure platform, based on literature review. Moreover, one other main source of information used in this research is the technical documentation provided by Microsoft on their support websites. Much of the previously conducted research is outdated therefore the gap between previous research and the current functionality of the MWA platform is covered by analyzing an extensive list of updated articles written by Microsoft or Microsoft partner authors. Additionally, the Windowsazure.com website is used for gathering current information regarding various stages of the MWA application development process.

The criteria for searching and using literature sources is that the sources need to be relevant to the research, current and published by established authors. The novelty aspect of the data sources needs to be emphasized as the research topic is constantly developing. The sources used for conducting descriptive and exploratory research were found by searching the electronic libraries to which the Kemi-Tornio University of Applied Sciences provides access to and by browsing relevant sections of the Microsoft

website, www.microsoft.com. Secondary sources were found by using the Google search engine.

3 CLOUD COMPUTING

The cloud computing world and the basics of cloud computing are discussed first. The chapter defines what cloud computing is, discusses its fundamental characteristics, describes the architecture of the cloud environment, various deployment models, and the most popular cloud computing services provider. Lastly, the security aspects of cloud computing environments are briefly discussed. As the MWA platform operates in a cloud computing environment, this chapter provides basic knowledge necessary for understanding the implications of running an application in a cloud environment.

3.1 Fundamentals of cloud computing

Cloud computing enables users to access the services offered by cloud service providers at any time and regardless of the users' physical location (Hurwitz & Bloor & Kaufman & Halper 2009, 8). According to Marks and Lozano (2010, 5-6), in an ideal world, computing would be easily scalable in size and utility, free or the costs would be close to none, and it operates appropriately at all times. The aim of cloud computing is to offer exactly the aforementioned attributes, together with many other features. The following definition of cloud computing is provided by the United States National Institute for Standards and Technology (henceforth NIST): "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell & Grance 2011, 2). Cloud computing can provide services that allow the users to fully operate a business in a cloud environment, which means that not only data can be stored but software applications and other operations can be run in such an environment. More definitions are provided by other organizations. However, for the purpose of this research the definition provided by NIST is used, as the definition comprises all the important aspects of cloud computing.

NIST (2011, 2) specifies that the cloud concept is based on five fundamental characteristics. First, cloud environments must have self-service capabilities. The most significant difference between traditional data centres and cloud environments is that in

case of the latter, services can be purchased, managed and scheduled without any human interaction between the customer and the provider. Provisioning services automatically results in increased efficiency and reduced costs for both parties, in comparison with the customers having to wait several days for the cloud service provider to analyze their request. Secondly, broad network access must be provided to the cloud environment. The customers must be able to access the cloud services and resources through standard mechanisms and by utilizing miscellaneous platforms, i.e. laptops, mobile devices etc. Thirdly, the resources of the cloud are shared between multiple customers according to their demands. Additionally, the resources can be physically located in separate data centres and the customer does not have exact control over the location of which resources are allocated to him. The customer can, however, specify an approximate geographical location. Resources include, but are not limited to, storage, memory and network bandwidth. The fourth fundamental attribute of cloud environments is elasticity. Cloud resources must be allocated and released automatically to provide an easily scalable environment. Lastly, resources usage must be automatically monitored, controlled and reported to both parties. (Mell & Grance 2011, 2; Krutz et al. 2010, 9-11.)

3.2 Cloud computing architecture

The architecture of cloud computing environments comprises two primary elements: the service delivery models and the service deployment models. Each service model can be delivered using a distinct deployment model. (Krutz et al. 2010, 33-43.)

The Software-Platform-Infrastructure (SPI) classification model represents the most broadly acknowledged cloud computing classification. As defined by Mell and Grance (2011, 2-3), cloud computing provides three primary service models: SaaS, PaaS and IaaS. The three service models are hierarchically classified according to the level of flexibility and security that the customer has over the cloud services. From the customer perspective, SaaS platforms offer the least amount of flexibility and the least amount of responsibility regarding the security of the services. In the middle of the hierarchy, PaaS offers some flexibility and some security responsibility lies on the customer. IaaS provides vast flexibility in terms of customization therefore the cloud service provider

cannot assure the security of the applications and only provides security at a platform level. (Krutz et al. 2010, 34-37.)

The SaaS model represents a model in which the provider offers software to the customer on request. The software is hosted on the provider's infrastructure and the provider is the sole party with access to the cloud infrastructure and application running parameters. The customer can usually access the applications through a web browser and, in some cases, has limited access to user-specific configuration settings. By adopting a SaaS solution, the customer eliminates the need to ensure software-hardware compatibility and the need to maintain the application, as the provider is responsible for providing maintenance and support. In addition, the SaaS model enables the service providers to control the distribution of unauthorized application copies. (Krutz et al. 2010, 37-39.) Examples of applications offered as SaaS are Google Docs and Gmail (Robinson & Valeri & Cave 2011, 17).

IaaS providers supply physical and virtual computing resources to the customer. The customer does not manage or control the fundamental cloud infrastructure. However, the customer is able to deploy, run and customize any software, including operating systems. The IaaS model provides a high level of scalability and can reduce the costs of purchasing, installing and maintaining hardware resources. Moreover, due to the nature of the automated cloud services, the time until the infrastructure is available for actual use is drastically reduced in comparison with purchasing and setting up physical infrastructure. (Mell & Grance 2011, 3.) Amazon is a provider of IaaS services, offering services such as Amazon Elastic Compute Cloud (henceforth EC2), Amazon SimpleDB and Amazon Simple Storage (Amazon S3) (Robinson et al. 2011, 17).

The third cloud service model, PaaS, represents the meeting point between SaaS and IaaS. PaaS provides more than just the use of an existing software application. PaaS services provide application design, development and management tools. Additionally, PaaS provides the cloud infrastructure necessary for storing and running applications. Similar to other cloud service models, the customer is not responsible for managing the fundamental cloud infrastructure. However, the customer has complete control over the applications deployed and customizing the application environment is possible, depending on the service provider. (Mell & Grance 2011, 2-3.) The major disadvantage

of PaaS is that many providers offer access only to proprietary development environments and tools, therefore restricting the customers to deploy applications coded in a specific programming language. As a solution to these limitations, some PaaS solutions are open source. An example of open PaaS solutions is the MWA platform that supports both Microsoft proprietary and non-proprietary programming languages and tools. (Sarna 2011, 206-212.) The MWA platform will be further discussed in chapter 4.

The second primary element of the cloud computing architecture is the deployment model. According to NIST, four cloud deployment models are distinguishable: public, private, community and hybrid clouds (Mell & Grance 2011, 3). Each of these deployment models can be paired with any service delivery model. (Krutz et al. 2010, 43).

The initial cloud deployment model, public clouds are common cloud environments in which the infrastructure is owned and managed by the cloud service provider. Moreover, the cloud infrastructure must be hosted at the location of the cloud service provider. Deriving from public clouds are private clouds. The essential difference between the two is that private clouds are provisioned for private use to a single organization or company, whereas public clouds are open to the general public. In private clouds, the cloud infrastructure can be owned, managed and operated by the same organization or by a cloud service provider. Private clouds are commonly hosted on an establishment owned by the same company that operates the cloud. The primary objective of private clouds is to enhance data security, obviously by allowing only the beneficiary company to control access to the cloud. The use of a public or private cloud is a decision made based on the interests of the company. When two or more companies share interests, i.e. operate in a specific industry branch, and agree to share the same cloud infrastructure, community clouds are formed. Plainly explained, community clouds are public clouds that are provisioned for private use by a particular community of customers or companies. (Marks & Lozano, 2010, 37-38; Mell & Grance 2011, 3.) An important aspect of community clouds is that they are not dependent on the existence of any of the companies involved in organizing the cloud (Krutz et al. 2010, 47).

A combination of any of the cloud deployment models discussed above represents a hybrid cloud. The two or more cloud infrastructures that comprise a hybrid cloud are connected through the use of technology that facilitates data and application portability. (Mell & Grance 2010, 3.) Hybrid clouds allow for an optimum cloud solution, without having to sacrifice costs for security or vice-versa, and are the most utilized type of deployment model in cases of enterprises or large corporations (Marks & Lozano 2010, 38.)

3.3 Major providers of cloud computing services

The world of cloud computing is constantly developing, with new concepts and technologies being created at a rapid pace. Established and new service providers each offer different cloud solutions intended for different categories of customers, i.e. SaaS applications for individual users or IaaS services for enterprises. Diversity creates competition. The multitude of available cloud solutions and cloud services providers accelerates the rate at which cloud computing services are evolving, feature-wise. Furthermore, increased competition causes the services to become less expensive, leading to increased benefits for the customers. Despite targeting different customer types and utilizing different technologies, cloud service providers have one common goal: to offer stable, secure and scalable cloud computing that facilitate application development and reduce the costs of managing a local environment. (Dudley & Duchene 2010, 13.) Examples of major cloud computing service providers include Amazon, Google and Microsoft (Krishnaswamy 2010, 13-18).

Amazon's cloud computing solution is EC2. EC2 follows the IaaS service model. The EC2 services focus on providing an easy to use and scalable environment intended for application developers. As the name suggests, the IaaS platform is elastic enough to enable customers to scale capacity, up or down, within minutes. The elasticity of the services is further proven and enhanced by other features that Amazon offers through the EC2 platform: elastic IP addresses, elastic block storage and elastic load balancing. Aside from increased scalability, Amazon emphasizes the security of the platform through features such as Amazon Virtual Private Cloud, Amazon CloudWatch and various other security techniques. EC2 customers are charged for hourly usage of the

services and for the volume of data transferred. Another critical aspect is that EC2 services are designed for use with other Amazon services, such as S3 and Amazon SimpleDB. Amazon's EC2 services provide virtualized instances of a plethora of databases and operating systems, including Microsoft Windows and several Linux distributions, therefore allowing developers to build applications using any programming language supported by the operating systems available. (Amazon 2012.)

Google offers Google App Engine (henceforth GAE). It adopts the PaaS model and it focuses on developing web applications. Google's raw breakdown of the GAE services is explained as allowing customers to run their own web applications on Google's infrastructure. GAE provides integrated tools that offer support throughout the application development lifecycle. These tools allow for easy management of storage, distribution, load balancing and monitoring. While Amazon focuses on advertising their services as being highly scalable, Google emphasizes the ease of adopting their cloud solution. GAE has features such as automatic scalability and, conveniently for developers, several other cloud-based Google applications, i.e. Gmail and Google Docs, are integrated into the platform. Hosting and running an application on the GAE platform costs nothing, at first. Google maintains their policy of providing access to free resources. Non-paid user accounts can store up to 1 gigabyte (henceforth GB) of data and can use sufficient bandwidth to support up to 5 million visitors per day and various other free resources. Unlike the billing system present in EC2 and MWA in which the customers are charged on an hourly basis, the customers are billed only for the resources that are actually used, i.e. compute hours and data volume transfer. GAE supports only two programming environments, for Java and Python. Support for the Go programming language was added recently, however it is still in an experimental stage. (Google 2012.) The limited number of programming languages available could drive potential customers to adopt other cloud solutions, such as the ones offered by Amazon or Microsoft.

MWA is Microsoft's application platform available for the general public that provides resources in a public cloud infrastructure. MWA is offered as two service delivery models, PaaS and IaaS. Microsoft does not focus on only one characteristic of the MWA platform but instead several major features are highlighted. Most importantly for developers, MWA is an open platform, which means that customers can use any

operating system and can deploy applications developed using any programming language. Additionally, MWA supports three distinct storage types: structured query language (henceforth SQL) databases, noSQL tables and binary storage. The billing policy allows customers to register a trial account, free of charge, valid for a period of three months. The trial account includes one fully operational SQL database and data storage up to 35 GB, in addition other free resources. (Microsoft 2012c.) MWA adopts an open approach to application development. It gives the customers the possibility to tailor their applications based on their specifications, not on the limitations imposed by the cloud service provider. Furthermore, MWA has a couple of features that are not available on other cloud platforms, i.e. Access Control and Service Bus. (Dudley & Duchene 2010, 13.) The MWA platform is the focus of this research and will be discussed in Chapter 4, including technical details and business implications.

Table 1 presents a comparison between the three cloud services platforms discussed above. The data presented in the table are gathered from the official documentation provided by the three cloud service providers, as of November 2012. The comparison is made from technical and functional points of view. Furthermore, the current pricing model is included in the table. The data regarding the pricing model and resources available do not display the complete pricing policies as each cloud service provider implemented various other restrictions that affect both the charged and the free resources. For updated information on any of the cloud services described below, I recommend visiting the website of the appropriate cloud service provider.

Table 1. Comparison between different cloud services platforms

	Amazon EC2	**MWA**	**GAE**
Deployment type	IaaS	PaaS/IaaS	PaaS
OS supported	Red Hat Enterprise Linux, SUSE Linux Enterprise, Fedora, Oracle Enterprise Linux, various Linux distributions, Windows Server, Amazon Linux AMI	Windows Server, CentOS, Ubuntu, SUSE Linux Enterprise, openSUSE, Guest OS	Linux
Languages supported	.NET, C++, PHP, Ruby, Python, Java, Android, iOS	.NET, C++, PHP, Ruby, Python, Java, Windows Phone, iOS	Java, Python, Go
Storage	MySQL, MongoDB, Microsoft SQL Server, Postgre SQL, CouchDB + others	Azure SQL, non-relational tables, queues, blobs	Google Cloud SQL, Google Cloud Storage, App Engine Datastore
Applications run in VMs	Yes	Yes	No
Control Interface	API, Command Line, User Interface (UI)	API, Command Line, Management Portal	API
Load balancing	Yes	Yes	Yes
Suitable for developing small-sized applications	Yes	No	Yes
Suitable for developing highly scalable applications	Yes	Yes	Yes
Monitoring tools	CloudWatch	Online management portal, 3rd Party	Appstats for Java, Appstats for Python
Pricing model	Hourly, subscription plans	Hourly, subscription plans	Use based
Free resources	30 GB storage, 15 GB outbound bandwidth monthly, for one year	3-months trial with access to all features but limited resources	1 GB storage, 1 GB outbound bandwidth monthly

Another comparison between the three cloud service providers discussed above is a cloud speed test comparison, performed on a daily basis by a company specialized in cloud computing, CloudSleuth. The tests analyze data transmissions to and from the platforms and measure the average response times and availability of several cloud service providers. The methodology utilized in conducting these tests functions as follows. An identical application that imitates a small size e-commerce online shop is deployed to each cloud platform and then the response times and availability of the application is measured from different geographical regions. The results of these tests show that during the month of November 2012, the GAE platform has the lowest response times out of all the cloud platforms included in the tests. Out of the three providers analyzed in this subchapter, Microsoft's Europe main data centre located in Ireland ranks second, while Amazon EC2 services rank third. Concerning the availability of the services, the MWA platform has the highest availability for some geographical regions. GAE ranks second, followed by Amazon EC2, on the availability scale. For updated and more complete test results, visit https://cloudsleuth.net/global-provider-view. (Cloudsleuth.com 2012.)

3.4 Cloud computing security risks

In order to understand the concept of security in a cloud computing environment and what its implications are, three fundamental principles for information security need to be defined. The fundamental principles are defined broad enough to cover all the security policies an organization might employ. Furthermore, all the possible types of security breaches that an organization might be subjected to are included in the definitions provided below. These three fundamental principles can be referred to as the CIA triad, where CIA stands for confidentiality, integrity, availability. (Krutz et al. 2010, 125.)

First, confidentiality represents the capability of protecting information from being disclosed, including means through which confidentiality is achieved. A loss of confidentiality represents intentional or unintentional disclosure of confidential information. The second principle of security is integrity. Integrity represents the capability of preserving information in the original format, guaranteeing the authenticity of the information. Loss of integrity can result from intentionally or accidentally

altering information. The last principle included in defining the concept of security is availability. Availability ensures that information is accessible by authorized parties, from the specified locations and through the specified means. A loss of availability results in access to information being temporarily or permanently disrupted. (Standards for Security Categorization of Federal Information and Information Systems 2004, 2.)

Other principles need to be discussed as well, in order to thoroughly define security. Identification and authentication are the processes that ensure that only authorized, legitimate users have access to specific information. Authorization represents the permissions that establish the user's range in terms of access to information, commonly achieved through deployment of access control lists (henceforth ACL). (Krutz et al. 2010, 127.) Trust and reliability are also concepts difficult to define themselves but they need to be considered in defining the concept of security. Cloud services providers must create a solid trust relationship with the customers in order to maintain the cloud computing premise that the cloud resources are reliable and available at any moment. (Marks & Lozano 2010, 108.) Another relevant aspect of security is privacy. Privacy can be challenging to define due to various global organizations presenting different definitions. Privacy represents the level of confidentiality and protection of private information and is ensured in accordance with laws established regionally. (Krutz et al. 2010, 127.)

The security of cloud computing environments and information related to such environments is the responsibility of both the service provider and customer. In most cases, the service provider is responsible for maintaining the security of the cloud infrastructure, including physical, logical and network resources, whereas the customer is responsible for the security of the applications and information stored in the cloud environment. Security is vital for cloud computing environments and all security measures must be permanently employed, regardless if the customer is accessing the resources. Cloud computing environments are accessible through common Internet protocols thus they are susceptible to common attacks, vulnerabilities and threats that affect Internet communications.

Breaches in security can be defined in terms of loss of any of the CIA triad principles, as discussed above. Deploying applications into a cloud environment implies entrusting

a third party with data critical for the well-being of the business. Therefore, security breaches can have a devastating impact on an organization's resources. The impact of security breaches can be low, medium or high, depending on the damage caused to the physical resources of an organization. Furthermore, security breaches can have a negative impact an organization, in terms of credibility or public confidence loss. (Stoneburner & Goguen & Feringa 2002, 22-23.) In an attempt to ensure the security of the cloud environment, several security measures are recommended to be implemented by the cloud service provider, by the customer or both parties. General security measures should include maintaining up-to-date network security protocols, implementing network authentication and data encryption services and implementing firewalls and intrusion detection systems. (Krutz et al. 2010, 125-126.) Security principles recommended to be considered when designing applications for the cloud include the least privilege principle, separation of duties, the fail-safe principle and the weakest link principle (Krutz et al. 2010, 66-67).

While cloud computing environments create various benefits and opportunities for developers by successfully addressing issues such as high costs for high performance hardware, the security risks associated with deploying an application in a cloud environment are significantly increased when compared to running an application on-premise. However, by adopting several well-known, recommended policies and practices, developers can reduce security risks and minimize the impact in case the security of the application or that of the cloud environment is breached.

4 MICROSOFT WINDOWS AZURE

The MWA platform is explored, from technological and functional points of view, in this chapter. First, this chapter briefly introduces the MWA platform, including the technologies and services associated with it. Then, an analysis of the costs of the MWA services is presented. Further, this chapter provides information about developing cloud applications using the MWA platform, while also implementing and deploying a sample application. Moreover, recommendations regarding various technical aspects or business implications of the using the MWA services are given, where appropriate.

4.1 Introduction

MWA represents Microsoft's solution for cloud computing services. Launched in 2010, it is a platform which essentially provides the resources for running applications and storing data (Cunningham 2010, 1). However, through the MWA platform, Microsoft provides more than compute and storage services to customers. Unlike the solutions provided by Microsoft's competitors that offer either raw resources or very limited development tools, the MWA platform provides a more managed experience. The services and tools provided by MWA are designed to assist developers in developing easily scalable and easily manageable applications. The services and support offered by MWA are suitable for developing a wide range of applications, including SaaS or enterprise applications. (Microsoft 2012a.)

MWA adopts both IaaS and PaaS models. It is a collection of cloud services that provide for developers means for building highly scalable applications. The three main components of the MWA infrastructure are, as illustrated in figure 1, compute, storage and the fabric controller. (Krishnaswamy 2010, 18.)

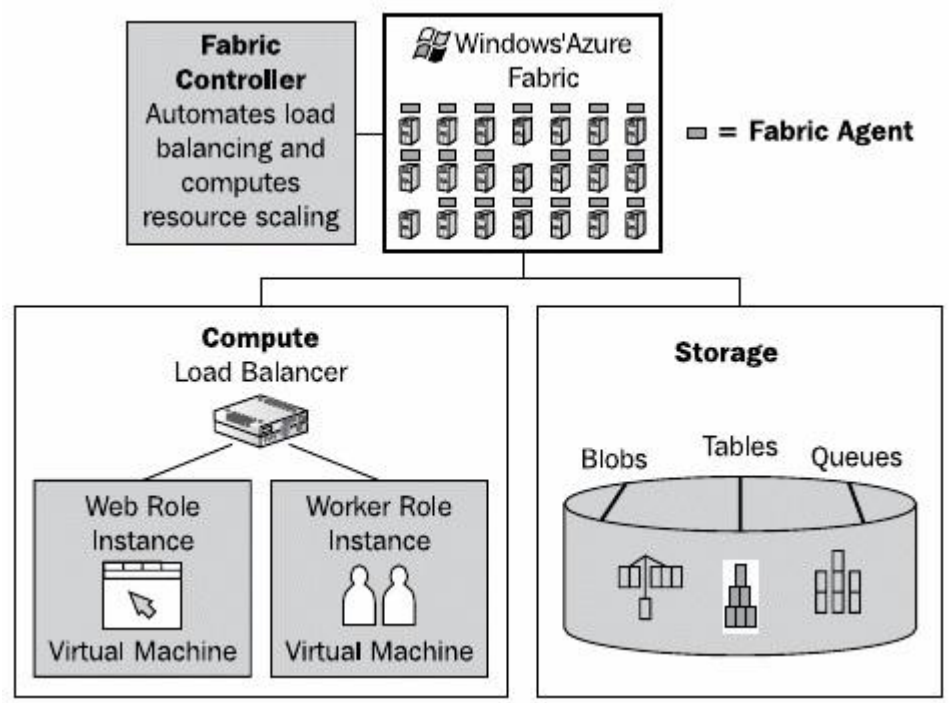

Figure 1. MWA components (Seroter & Fairweather & Ramani 2010, 121)

Developing applications using MWA requires a different approach due to the way the MWA services are designed. The technology and capabilities of the MWA platform can be summarized as follows. First, the compute service provides processing power to run the applications which are hosted in separate VMs that run the OS selected by the developer, i.e. Windows Server or Ubuntu. However, the VMs themselves run on the Windows Azure OS. Then, the storage service provides storage space on servers in data centres owned by Microsoft. Lastly, the fabric controller represents the application that manages the communication between the compute and storage services. (Chappel 2010.)

4.2 Pricing

Analyzing the pricing models of the MWA platform and services associated with it is necessary in order to highlight MWA's compliance with the core aspect of cloud computing of lowering infrastructure and management costs. Furthermore, the analysis of pricing models emphasizes the suitability of the platform for ISDs or small businesses, depending on the size and scope of the application.

The pricing of MWA services is complex enough to satisfy the needs of most customers, without compelling them to purchase unnecessary amounts of resources. MWA features several pricing models, based on the type of service and on the type of customer. Microsoft offers several size or performance options for most of the services available. However, for the purpose of this research, only the pricing of the most significant resources for relevant services are described. The pricing presented in this research is calculated based on the rates applied to regular customers, excluding special pricing rates that may apply to various Microsoft partners or affiliated members. In addition, the currency used to present the prices is Euro (€). The payments can be done in several different currencies, depending on the country the customer is purchasing from. The payments can be done by using a credit or debit card, or invoicing only under special conditions. (Microsoft 2012d.)

MWA offers two payment models: pay-as-you-go and subscription pricing. First, the pay-as-you-go pricing model charges customers for resources used, at an hourly rate. Partial service hours are billed as full hours. The billing time starts at the time of deploying the application and stops when the application is removed from the servers; this implies that the customer is charged even is the application is idle. In addition, the hourly billing rate is calculated taking into account clock hours. If the application is running from 1:50 PM to 2:10 PM, the customer is billed for two hours of usage. The billing rates used for this pricing model are the standard rates that are used to calculate all other payment options. (Microsoft 2012d.)

Table 2 presents the standard pricing rates used to calculate all the billing rates, regardless of the pricing model chose. The services included in Table 2 are relevant services for application developers. The prices are calculated in € and reflect the current prices available on the official Microsoft price calculator and documentation. The amount of resources for each service type is selected to fit the minimum requirements to run a small to medium application. First, the selected VMs have 1 central processing unit (henceforth CPU) core at 1.6 Gigahertz (henceforth GHz) frequency and 1.75 GB of random-access memory (henceforth RAM). The VMs are presented as small size and the prices are applicable for both Windows and non-Windows VMs. VMs are billed at 33% discounted standard rates for the moment, as the service is recently implemented

into the platform. Secondly, the cloud services selected are of small size and feature 1 CPU at 1.6 GHz frequency, 1.75 GB RAM, and allow access to 225 GB of storage. Then, the storage services, except relational databases, are billed in GB of the average daily amount stored over a monthly period. This means that if the customer stores 2 GB of data in the first half of the month and no data in the remaining of the month, the customer is billed for 1 GB of stored data at the end of the month. The storage type selected for the purpose of this research is geo-redundant storage, as it is the default storage type. Finally, the price of outbound bandwidth represents the billing price for bandwidth that exceeds 5 GB of traffic. The first 5 GB of outbound bandwidth and all inbound bandwidth are available free of charge. (Microsoft 2012e.)

Table 2. MWA standard pricing rates (Microsoft 2012e)

	Price in €
VM	0.0568 per hour
Cloud services	0.0852 per hour
Relational database, 0-100 MB size	3.5425 per month
Storage, up to 1 TB	0.0887 per GB
Service Bus	0.0071 per 10000 messages
Outbound bandwidth	0.0928 per GB

The second pricing model is based on subscription plans. Currently, there are two subscription plans available, for the duration of six or twelve months. Both subscription plans can be paid on a monthly basis or in full, at the beginning of the contract. Monthly subscription plans can only be purchased if a minimum of 350€ worth of services is selected, for either six or twelve months. Customers that purchase a six months subscription plan are charged for rates equal to 20% discounted standard rates, while twelve months subscribers are granted 22.5% discount applied against standard rates. (Microsoft 2012f; Microsoft 2012g.) The subscription plans can also be paid in advance, at the beginning of the contract, but the minimum subscription needs to be of at least 2100€ for six months prepaid subscriptions and of at least 4200€ for twelve months prepaid subscriptions. The discount rates that apply for the prepaid subscription plans

are of 22.5% and 25%, respectively. Various other rules and restrictions might apply, according to the policies presented by Microsoft. (Microsoft 2012h; Microsoft 2012i.)

Besides paid accounts, a trial account option is available at the moment. Trial account users have access to all the features of the MWA platform but the resources available are limited. Trial account users can utilize up to 750 hours of VM compute resources per month, same type and size as the VM example used as reference in Table 2. Additionally, users can utilize 1 SQL database, 35 GB of non-relational storage, unlimited inbound and 25 GB outbound bandwidth, and half a million service bus messages. Other resources are also available free of charge during the trial period. A valid credit or debit card, a valid Windows Live ID and a valid mobile phone number are required for creating a trial account. (Windowsazure.com 2012a.)

Support service is another service available for purchase. Currently, support features can be purchased at a monthly rate and are available in multiple categories. All accounts have access to support regarding billing and subscription management. Furthermore, all accounts have access to the community forums and service dashboard. Paid support features are available under the following categories and billing rates: developer, standard and professional direct available for 20.57€, 212.77€ and 709.20€, respectively. The highest level of support, named premier support, is only available on demand, after contacting Microsoft. A special offer, active until December 31st 2012, provides standard-type support free of charge, for all customers. (Microsoft 2012j.)

For updated prices and resource sizes available for purchase, I recommend visiting the official MWA price calculator, https://www.windowsazure.com/en-us/pricing/calculator/. Furthermore, regional restrictions are applicable, as the MWA platform services are available only in 89 countries (Laing 2012).

4.3 Development set up

The application development tools provided by Microsoft are compatible with a multitude of operating systems, Internet browsers and development environment versions. For the purposes of this research, I utilized the following freeware or

shareware software applications and their corresponding versions to serve as the development environment:

- Google Chrome Internet browser, version 23.0.1271.64 m
- Microsoft Visual Studio 2010 Premium, version 10.0.40219.1 SP1Rel
- Microsoft Visual Basic 2010
- Microsoft .NET Framework, version 4.5.50709 SP1Rel
- Windows Azure SDK for .NET (VS 2010 SP1), version 1.8
- Windows 7 Professional x64, Service Pack 1 and all the available updates from the Windows Live Update centre, as of December 2012
- Microsoft Web Platform Installer 4.0.

The first step for utilizing the MWA platform services is creating an account. A trial account is used as an example; however, the subscription plan can be changed after creating the initial account. A valid credit or debit card, a valid Windows Live ID and a valid mobile phone number are prerequisites prior to registering for an MWA account. After meeting the requirements, a trial account can be created by accessing the Internet address https://www.windowsazure.com/en-us/pricing/free-trial/ and clicking on the *try it free* button displayed in the middle of the page. A new window that contains all the information regarding trial accounts is displayed further. Figure 2 represents the first form displayed during the registration process. The customer is required to fill in two additional forms, successively displayed in the same window.

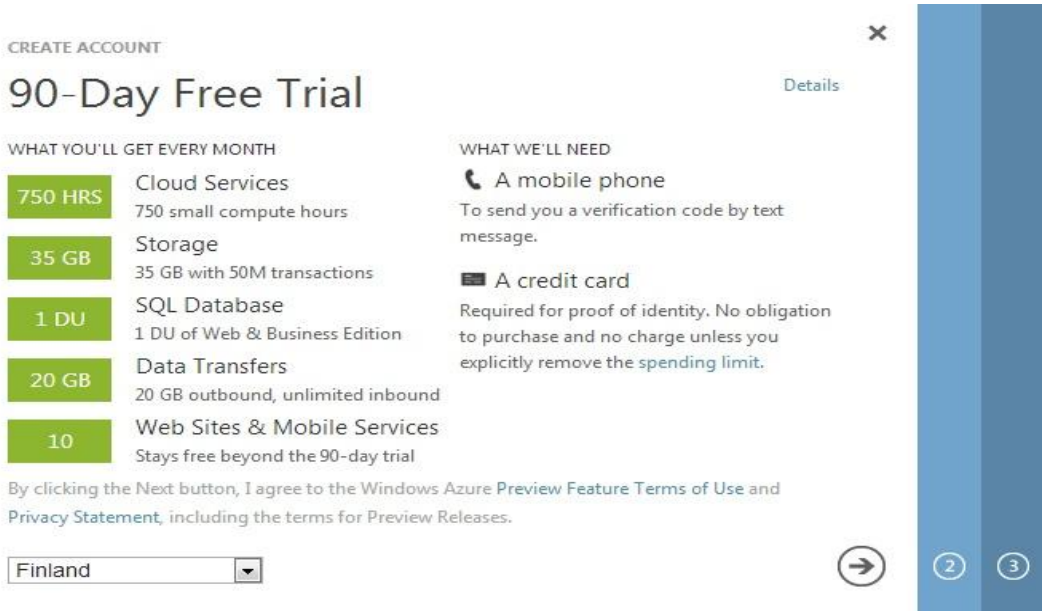

Figure 2. First form of the trial account registration process (Windowsazure.com 2012b)

A successful registration process results in the customer being redirected to the Microsoft Windows Azure management portal, also available at https://manage.windowsazure.com. The management portal is the interface through which the user can manage most resources and perform basic operations such as creation and deletion of services. Figure 3 illustrates the latest version of the management portal interface.

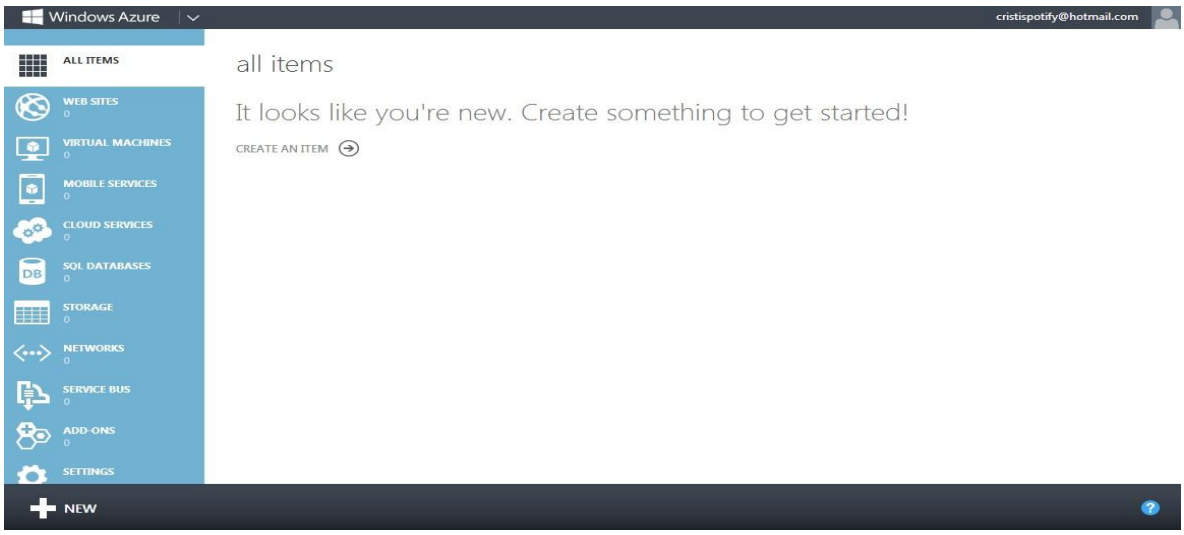

Figure 3. Windows Azure management portal (Windowsazure.com 2012c)

Further, the local development environment needs to be set up. First, the tools necessary for designing and developing the application need to be downloaded. All the development tools available for MWA can be found at https://www.windowsazure.com/en-us/develop/downloads/. For the purpose of this research, I downloaded the Visual Studio 2010 .NET tools, full install version.

Afterward, Internet Information Services (henceforth IIS) need to be enabled. This step requires administrator access rights on the local system. Enabling IIS requires opening the Control Panel directory on the local system and selecting Programs, Turn windows features on or off. Further, IIS, World Wide Web Services, and Application Development Features categories need to be expanded successively and the boxes corresponding to .NET extensibility, ASP.NET and CGI, respectively, need to be checked, displayed in Figure 4. (Dudley & Duchene 2010, 29-30.)

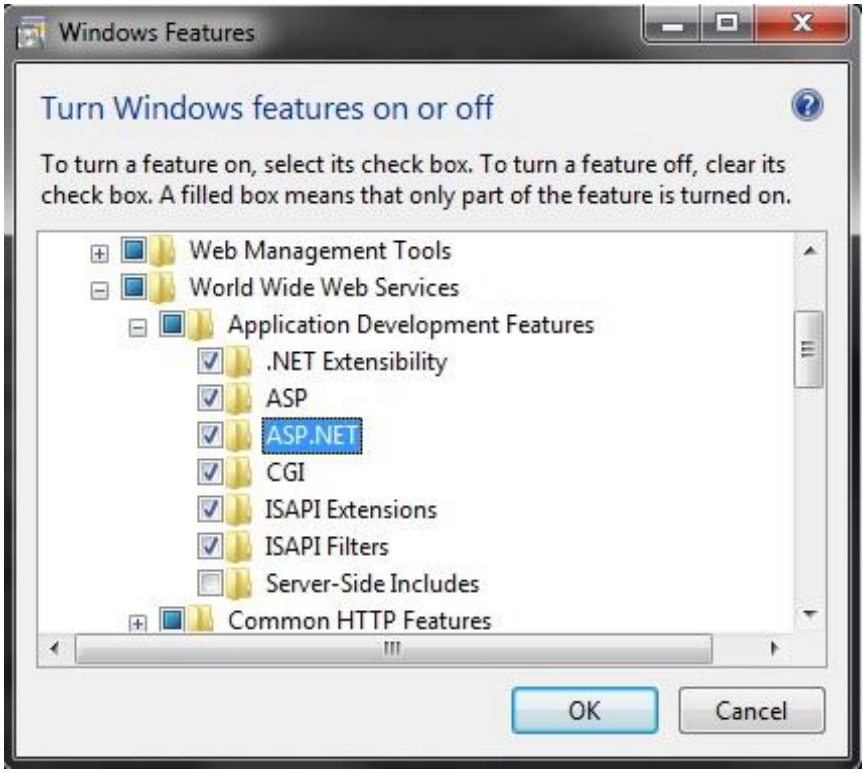

Figure 4. Customizing Windows features

Further, the following applications and their corresponding versions need to be installed:

- SQL Server Express 2008 R2
- Microsoft .NET Framework, version 4.5
- Windows Azure Tools for Microsoft Visual Studio 2010 – October 2012, version 1.8; note that there are two different instances of this application
- Windows Azure SDK 1.8
- Windows Azure SDK for PHP – October 2012
- ASP.NET MVC 4 with Language Packs (August 2012).

To simplify the set up processes, I recommend downloading the Microsoft Web Platform Installer (henceforth WPI) by accessing the website available at http://www.microsoft.com/web/downloads/platform.aspx. The application needs to be installed and run. If successful, the application interface, as presented in Figure 5, should be displayed. Tools can be searched through the search feature available in the WPI software, by using the full name and version of the tool as keywords, i.e. Windows Azure SDK 1.8 keywords for the "Windows Azure SDK 1.8" tool. The selected applications can be installed by pressing the *Add* button corresponding to each of the applications and then pressing the *Install* button. Alternatively, each application can be downloaded on its corresponding website. I recommend thoroughly reading the Terms and Conditions agreement, where applicable.

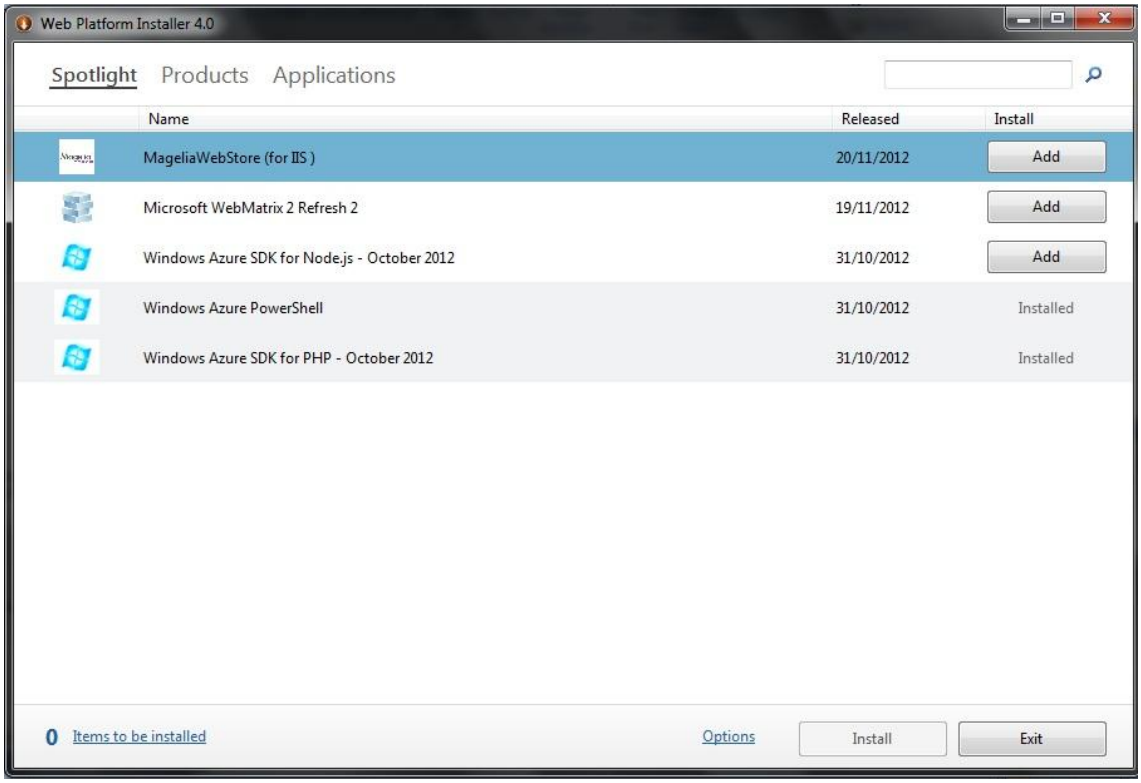

Figure 5. WPI interface

Additional MWA resources, including applications and documentation, are available for download from the Microsoft download centre, at http://www.microsoft.com/en-us/download/search.aspx?q=azure**.** The applications discussed above are necessary to be installed in order to be able to start developing MWA applications. Furthermore, the ASP.NET MVC 4 with Language Packs (August 2012) tool provides templates for developing web and worker roles and is not imperative to be installed.

I recommend restarting the system after all the resources have been installed successfully. Even if not required during the installation processes, some features might not be available for use or might not be fully operational unless the system is restarted.

4.4 Cloud services

The cloud service layer represents the resources that provide the processing power for the application deployed in the Windows Azure environment (Seroter et al. 2010, 122). The cloud service itself represents the service which provides processing power and

hosts the actual application code. In MWA, applications are divided into three components, called roles, specifically web roles, worker roles or VM roles. Web roles represent the front end of the application with which the users can interact, whereas worker roles are the back end, the part of the application that runs in the background and is not visible from the users' point of view. (Dudley & Duchene 2010, 17.) Any MWA application must be implemented as a single role or a combination of roles and multiple instances of any application role can be running simultaneously (Chappell 2010, 4). Moreover, web roles and worker roles function independently and each role is running in its own distinct VM. Despite running in separate VMs, different roles can interact with each other but application users are unable to interact directly with a worker role. (Dudley & Duchene 2010, 17.) Figure 6 presents the various roles that an application can function in and how communication is performed between roles and the MWA platform.

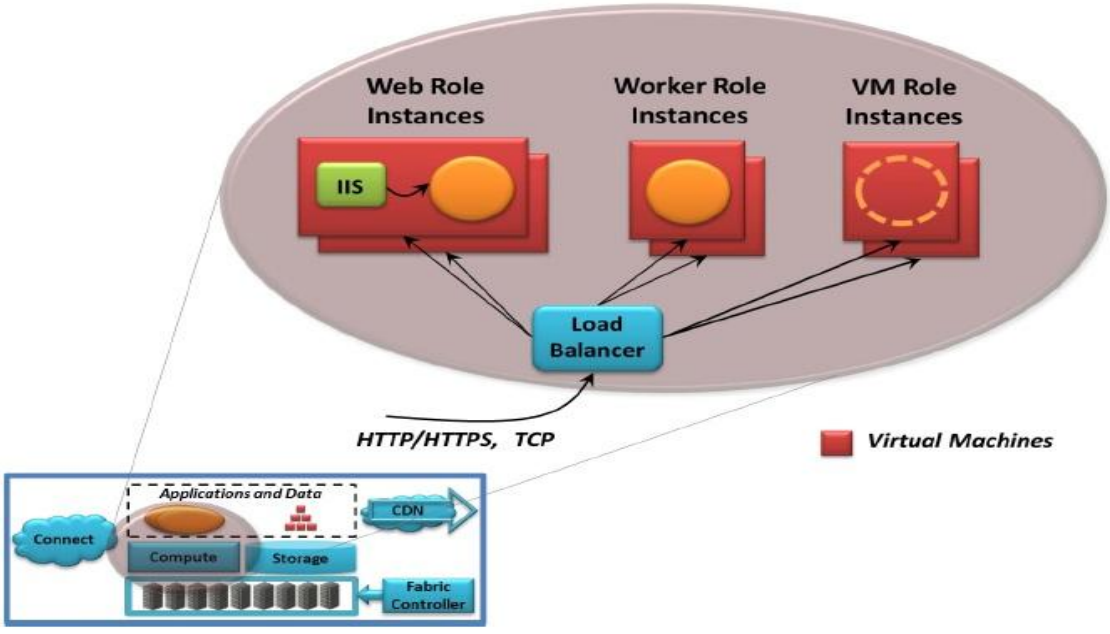

Figure 6. MWA application roles (Chappell 2010, 4)

The requests sent by application users are made using hypertext transfer protocol (henceforth HTTP), HTTP secure (HTTPS) or transmission control protocol (TCP) protocols. As applications role instances run in separate VMs, MWA does not provide any control over which instance handles a specific request. Instead, MWA provides automated load balancing, sending requests to all the role instances. (Chappell 2010, 4.)

4.4.1 Web roles

A web role represents the front end of the application hosted in the cloud environment. Web roles are similar to websites and may also incorporate web services. They can initiate or receive communication requests through IIS. The web role represents the service that allows for user interaction and serves as the connection point between the user and worker roles, commonly the graphical interface of a website. (Dudley & Duchene 2010, 120.) Furthermore, web roles can be utilized to host applications developed in any of the languages supported by MWA, be it ASP.NET, PHP or others. (Seroter et al. 2010, 122).

The process of creating a web role is described further. For the purpose of this research, I do not instruct on how to create application code nor do I create it myself. Instead, I utilized application templates provided through the use of the ASP.NET MVC 4 application. Figure 7 partially illustrates the process of creating a web role. The first step in creating a web role is to run Visual Studio 2010. After doing so, subsequently new project must be created. If all the tools discussed in chapter 4, subchapter 3 have been installed successfully, the project creation window should have available an option named Cloud, under the Installed Templates menu. .NET Framework 4 should be selected by default as the development framework. Further, the Windows Azure Cloud Services project type needs to be selected. The project and solution names are irrelevant for the purpose of creating the web role application as they are not related to the name of the web role application itself.

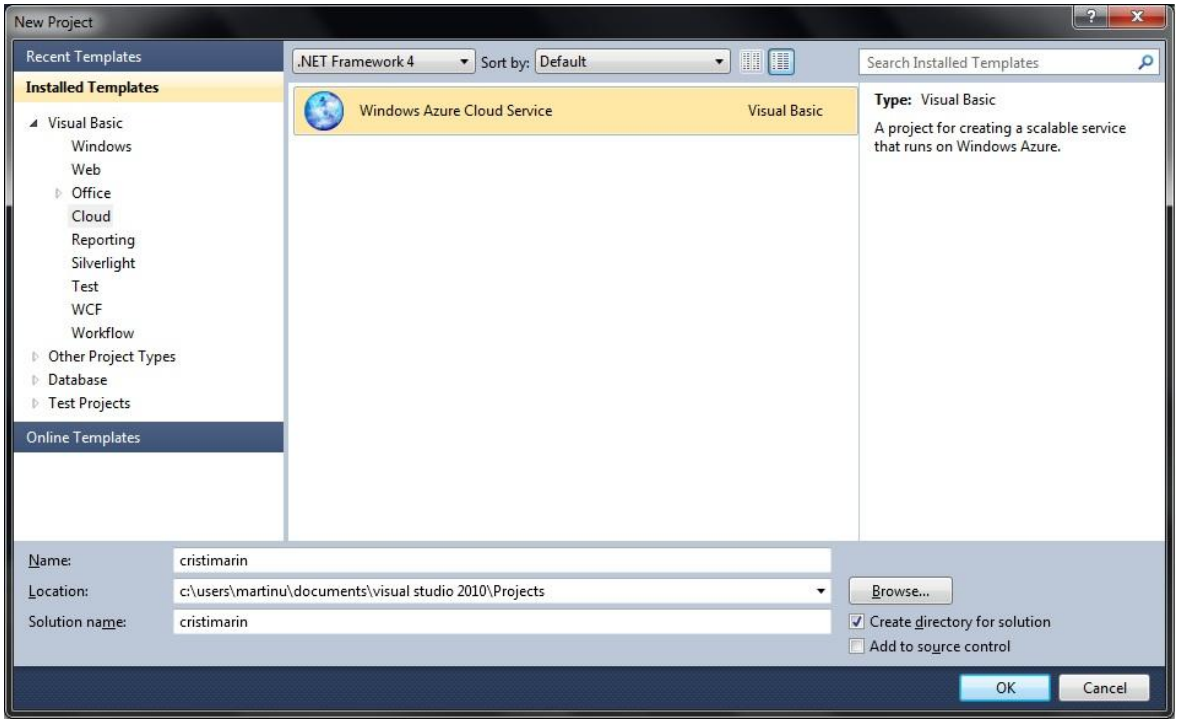

Figure 7. Web role creation

After selecting the project type, the new window allows for selecting the type of cloud service to be created. ASP.NET creates a web role using a template coded in the corresponding version of .NET Framework. To act as the template for the web role, I selected *ASP.NET MVC 4 Web Role*. Further, selecting the *Empty* option creates an empty project template that allows for custom applications to be designed and developed as part of the project. Inside the new window, the *Internet Application* option needs to be selected. This represents the project template that is used to create the web role. Additionally, the *Razor view engine* needs to be selected, if not selected as the default view engine. The Create a unit test project feature is not needed for this example as it creates an additional test project that permits for advanced testing of the application roles (Microsoft 2012k). Figure 8 shows the result of these last few stages of the process.

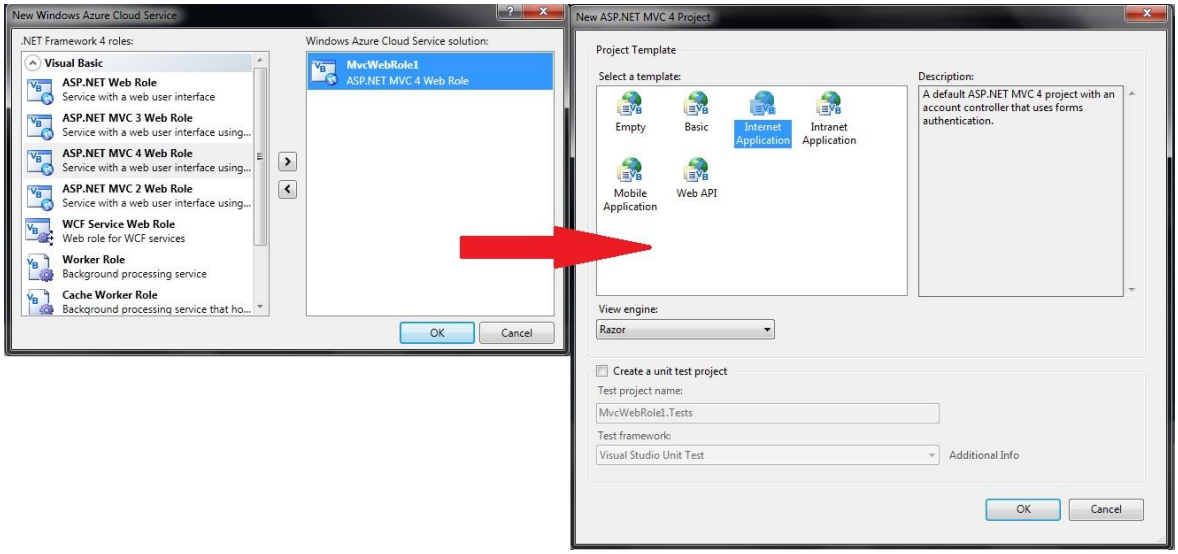

Figure 8. Advanced stage of web role creation

If the creation process is performed successfully, the web role can be configured further. The configuration of a MWA project includes two files, namely ServiceDefinition.csdef and ServiceConfiguration.cscfg. Configuring a role implies modifying the default parameters set into these two files. The configuration menu is accessible by selecting the role itself on the right side menu, inside the expanded MWA project tree, right-clicking and selecting the *Properties* option. (Microsoft 2012l.) Alternatively, double-clicking the role yields the same result, as presented in Figure 9. A third method of modifying the values of most parameters is to directly edit the configuration files discussed above, which can be done using Visual Studio or a regular text editor. The configuration files are coded using extensible markup language (XML).

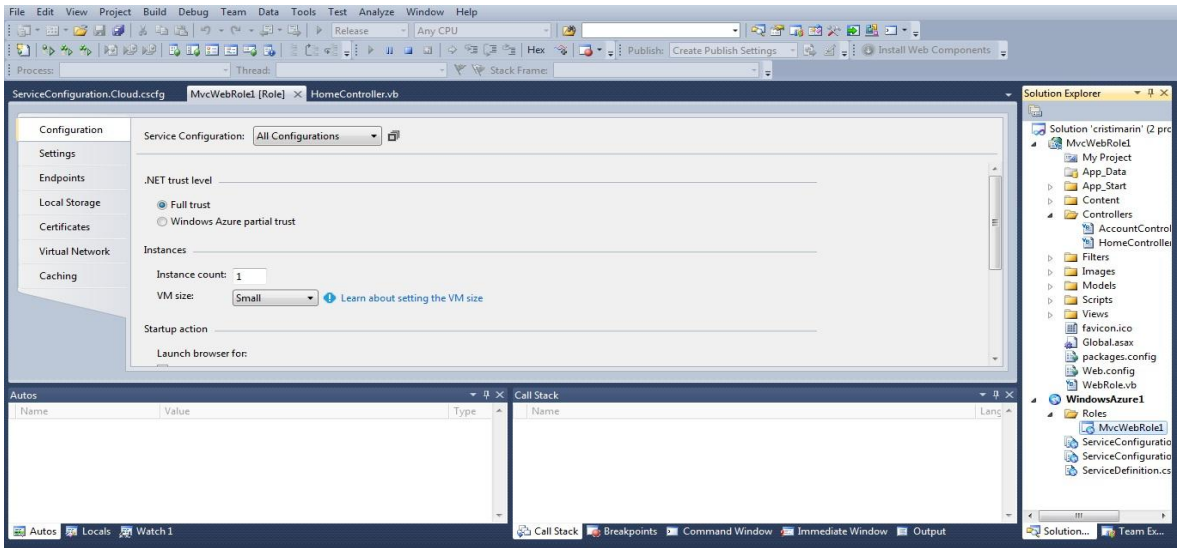

Figure 9. Web role configuration

The configuration settings menu displayed on the left side in Figure 6 allows the user to control the running parameters of the web role. First, the Service Configuration selected option must be *All Categories*, as the other options do not allow for most parameters to be modified. Then, the .NET trust level represents the rights level assigned to the web role. It is set to Full trust by default and should not be changed unless the application to be deployed requires restricted access to resources. (Microsoft 2012l.)

Further, the number of instances that the application can run in can be modified. As discussed at the beginning of this chapter, any role can run in multiple instances. MWA deploys and maintains three instances of the web role if a value of 3 is set to the Instance count parameter. This is an excellent method to highlight the automated services of the MWA platform. By adjusting the value of the Instance count parameter, the developer can run an identical application multiple times without having to repeat the creation process or other steps, such as installation or configuration, in case of a more complex application. Setting at least two instances for every role is recommended; in case one instance crashes, the second one handles all the requests until the first instance is restarted. The application instances can be viewed in a simulated environment, the MWA Compute Emulator. To view these simulated instances, the *Debug* menu and *Start Debugging* option subsequently need to be selected or the F5 key, by default, needs to be pressed in Visual Studio. The MWA compute and storage simulators are started if all the steps are performed successfully. Then, right-clicking on the Windows Azure notification icon displayed on the Windows taskbar and selecting *Show Compute Emulator UI* launches the compute emulator interface which displays information about the running instances of the web role created. (Brunetti 2011, 50-54.) Figure 10 illustrates the simulated difference between one and three instances of the web role. The application can be terminated by returning to the Visual Studio interface and selecting the *Stop Debugging* option or pressing the SHIFT + F5 keys, by default.

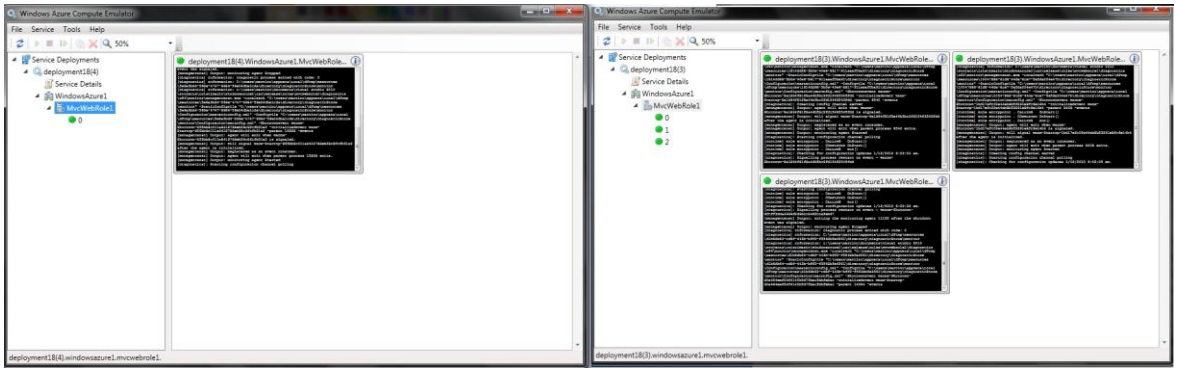

Figure 10. Compute Emulator displaying one and three running application instances

The VM size parameter displayed under the Instances subcategory allows the user to select the size of the VM that the role runs in. For information on VM sizes, refer to chapter 4, subchapter 2 and to the official documentation available at http://msdn.microsoft.com/en-us/library/windowsazure/ee814754.aspx.

The following subcategory of the Configuration settings menu allows the user to modify the Startup action parameters. This parameter specifies whether the local browser should launch the application using HTTP or HTTPS protocols. HTTPS endpoints can only be selected if they have been previously defined through the online management portal. (Microsoft 2012l.) The HTTPS creation process is not included in this research. The Startup action parameter is available only for web roles.

The Diagnostics settings are enabled by default to use the Windows Azure storage emulator. When the application is deployed to the MWA platform, the Diagnostic settings need to be updated to specify the name of the real storage account. The last parameter available is the Connection Strings one. This parameter specifies if the application should automatically update the storage connection strings whenever the value of the string is changed. I recommend checking the box corresponding to this parameter. Not updating the connection string results in severe web role malfunctions. (Microsoft 2012l.)

The second configuration menu, named Settings, contains the parameters required to access the storage account used by MWA. Connection strings can be configured in three distinct manners: to connect to the local emulated storage account, to connect to the MWA storage account by using an automatically generated configuration file provided

by Microsoft or to connect to the MWA storage account by manually providing account credentials. (Microsoft 2010a.)

The Endpoints and Certificates configuration menus are directly related to each other. The Endpoints parameters permit the user to add or remove endpoint connections. These connections are used to define the connection type between a VM and any local or Internet resources, and can be of two types: internal or available only to other roles, input or available for external requests only. Setting up an HTTPS endpoint requires selecting an SSL-type security certificate, through the Certificates configuration menu or through the online management portal. If IIS services have been enabled, an IIS SSL certificate should be available for testing purposes. Figure 11 illustrates the IIS certificate available for local development. (Microsoft 2012l.)

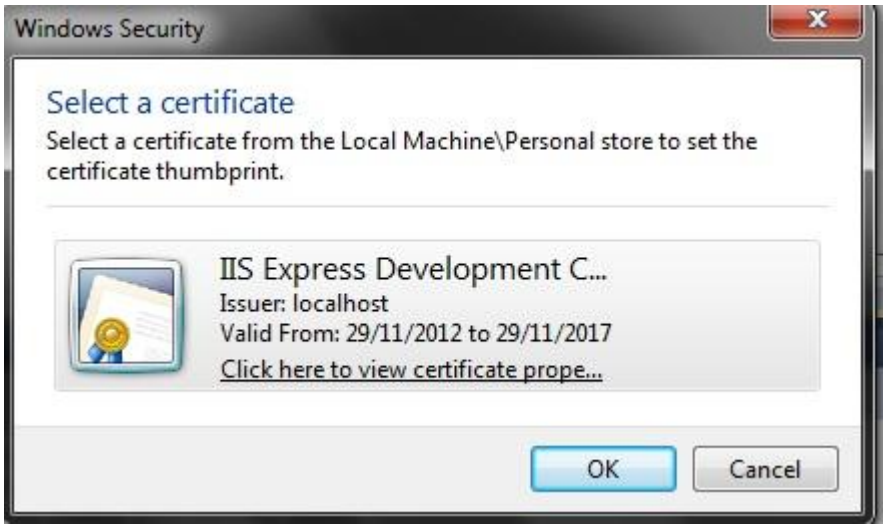

Figure 11. IIS Express Development Certificate

Local storage is the following category in the web role configuration menu. Local storage resources represent reserved storage space on the VM the role instance is running on. The minimum value of this parameter is 1 MB and the maximum value is limited by the size of the total VM storage space available. (Microsoft 2011a.) The following category addresses parameters related to Virtual Networks. A virtual network must have been previously created through the online management portal in order to be able to activate Windows Azure Connect. Virtual Networks allow setting up an IP address to serve as the endpoint between the selected role and other applications that run on computers outside of the MWA environment. (Microsoft 2012l.) The last

Configuration category is caching and it allows enabling the caching feature in MWA. Caching allows applications to store frequently used data, therefore reducing the delivery times. (Microsoft 2012m.)

4.4.2 Worker roles

A worker role represents the part of the cloud service that runs in the background. It does not have a UI and its purpose is to perform background actions, such as data or video processing. Worker roles are similar to Windows services that run in the background of the local system. This type of role can be utilized to host a server for any programming language supported by MWA, however worker roles can only be developed using .NET. (Dudley & Duchene 2010, 159-160).

The process of creating a worker role using Visual Studio 2010 is similar to the web role creation process. Repeating the process described in chapter 4, subchapter 4.1, regarding creating a web role and, selecting *Worker Role* instead of *ASP.NET 4 MVC Web Role* creates a worker role for the project. Alternatively, a worker role can be added to an existing MWA project by right-clicking on the current project name, selecting *New Worker Role* and, in the new window, selecting Worker Role, as illustrated in Figure 12.

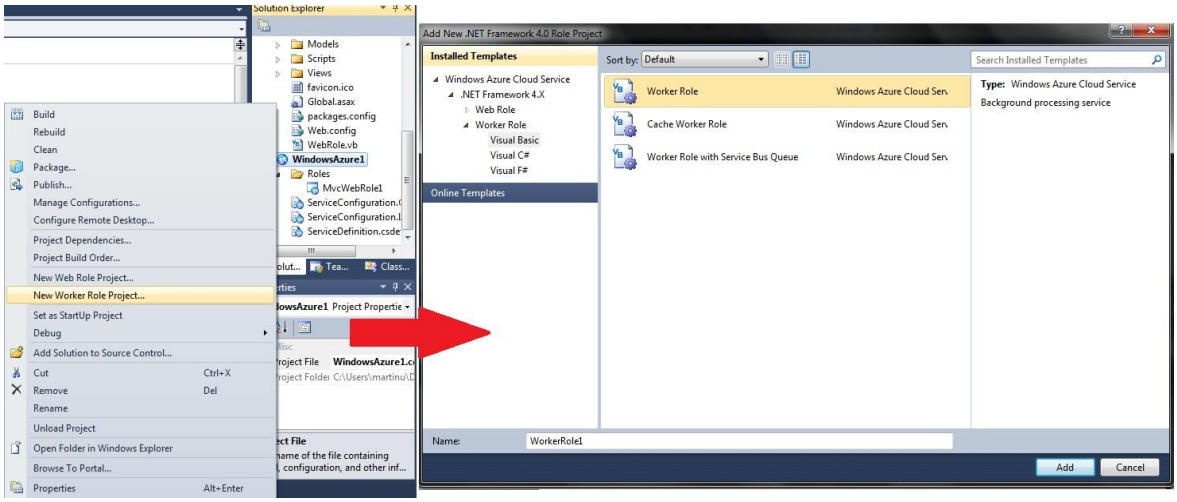

Figure 12. Adding a worker role

The configuration of a worker role is similar to configuring a web role. The configuration menu and parameters are identical, with the exception of web role specific parameters.

An example of an application that utilizes at least one web role and one worker role is a website with a search engine. The web role represents the front end of the website, which is the interface through which the user can interact with the website itself. The worker role represents the search engine function. When the user initiates a search query, the worker role processes the data and returns the results to the web role. If one of the roles cannot handle all the user-generated requests, the developer can easily scale up a role by increasing the number of instances in which the corresponding role runs, as discussed in chapter 4, subchapter 4.1. To further emphasize the scalability properties of MWA, an application could assign two distinct worker roles for two actions, even if the actions are identical. (Seroter et al. 2010, 122.) When increasing the number of instances for a role, consider the cost implications, as each role instance consumes additional resources.

4.4.3 Virtual machines

VM services highlight the IaaS capabilities provided by the MWA platform. VMs provide powerful processing resources for developing new applications or migrating new ones. This compute service can run OSs other than Windows Azure, such as Windows Server 2012 or various Linux distributions. Multiple VMs can be created at once and each VM can be configured to work with other roles. (Microsoft 2012n.)

Creating a VM is an easy process, performed by selecting the Virtual Machines category the management portal available at https://manage.windowsazure.com. Pressing the *New* button at the bottom of the page and selecting From Gallery displays the VM creation form. In the new window, an OS can be selected, either provided by Microsoft or from an image or disk, having previously created one. The second configuration page requires assigning a name to the VM, creating a password for the default Administrator account and selecting a VM size from the predefined options available. Further, the developer can select the standalone VM mode or connect the VM

to an existing VM, for distributing the requests between two or more VMs. Also, a domain name (henceforth DNS) can be provided and a new or an existing storage account can be assigned to the VM. Further, an availability set with an appropriate name needs can be selected. Clicking *Complete* ends the creation process and the new VM should be ready in a few minutes. Additionally, advanced information regarding the selected VM is displayed when clicking on the VM name. Figure 13 illustrates the management portal and a running VM.

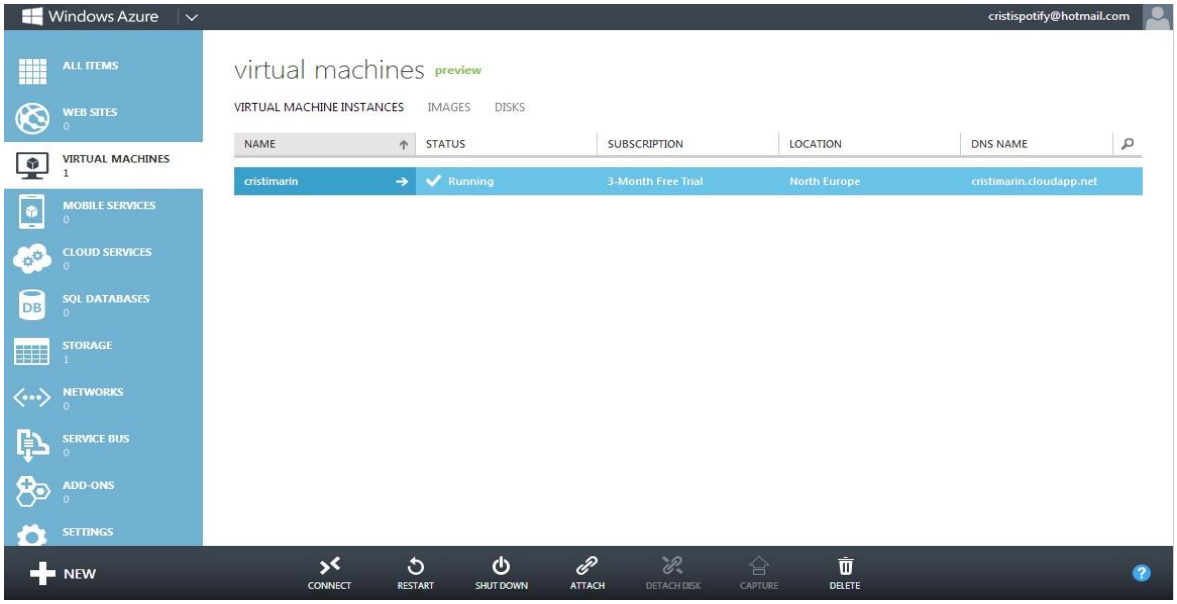

Figure 13. Running VM (Windowsazure.com 2012d)

Connecting to a VM is also an easy process. Selecting a VM that runs Windows OS and clicking the *Connect* button at the bottom of the page starts the download of a file with the .rdp extension; rdp stands for Remote Desktop Protocol. The .rdp file needs to be run and the developer must provide the credentials assigned during the VM creation process in order to complete the connection process. To connect to a VM running a Linux OS, a secure shell (henceforth SSH) application needs to be installed. Microsoft recommends PuTTY and OpenSSH for Windows and Linux systems, respectively. The connection details necessary to connect through the SSH application of choice can be found on the VM details page.

4.5 Storage services

The storage services in MWA provide the customer with three options. The storage options available are blobs, tables and queues, each with its own purpose. The variety of storage options available allows the customer to build highly scalable applications for low costs. Instead of storing data locally, applications can store huge amounts of data on the MWA servers. Though prices also increase as the size of data increases, the storage services of MWA can potentially offer lower overall costs than local storage. Furthermore, the storage services are essential for the functionality of applications that run in the MWA environment. By storing data using a storage service, MWA creates a VM in which the data is stored. Even if the physical location may not be the same, the VM is always accessible by connecting to the same endpoint. This way, data is also available at all times, regardless if the VM is running. Also, as a security measure, data is replicated at least three times (Dudley & Duchene 2010, 248; Chappell 2009, 7.)

Blob storage is the first storage service to be discussed. The name of the service is an acronym that stands for binary large object. Blobs are utilized to store large amounts of unstructured data that can be easily accessed by connection to the endpoint associated with the blob. They can be used to store any kind of data and have a maximum size of 1 terabyte (henceforth TB), depending on their type. The infrastructure of blobs is similar to the file system available on local systems, illustrated by Figure 14. A MWA storage account can contain multiple containers which can in turn contain multiple blobs, similar to how a hard drive can contain multiple directories and a directory can contain multiple files. There are two types of blobs, namely block and page blobs. Block blobs are utilized for storing streaming data such as video files and have a maximum size of 200 GB, whereas page blobs are designed for random access write and read actions and can store up to 1 TB of data. Moreover, a container can store an unlimited number of blobs and a storage account can store an unlimited number of containers. However, the maximum size of a storage account is 100TB. (Dudley & Duchene 2010, 77-78; Microsoft 2012o.)

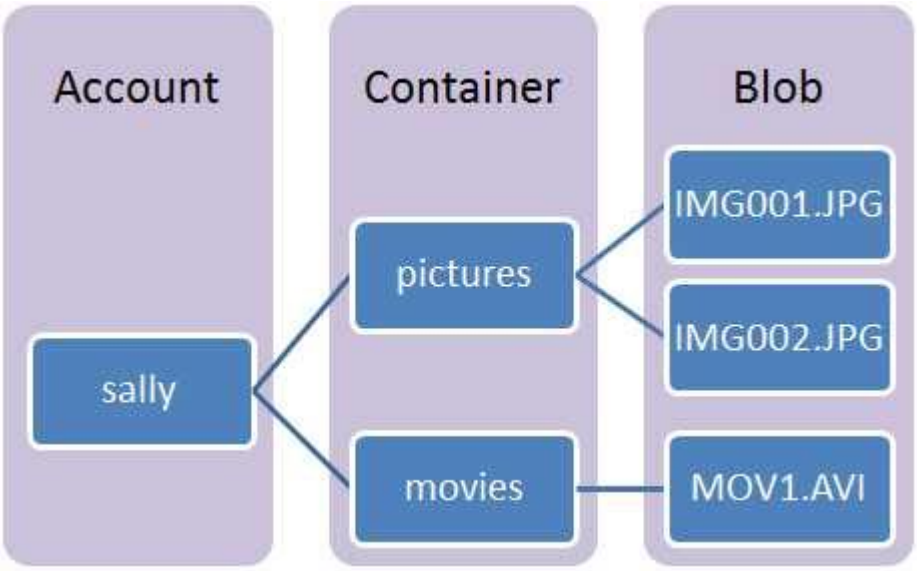

Figure 14. Infrastructure of a blob (Microsoft 2012o)

The process of creating a blob is performed through the management portal and it is a separate process for creating a blob, a container or a storage account. First, a storage account needs to be created, process illustrated in figure 15. Accessing the management portal and authenticating is required. Further, the Storage service category needs to be selected and the *New* button at the bottom of the page needs be clicked on. A name must be assigned to the storage account, name which also represents the endpoint through which the blobs, tables and queues associated with the storage account can be accessed for developmental purposes. An appropriate region/affinity group also needs to be selected and the Geo-Replication feature can be enabled. The *Create storage account* button completes the process of creating a storage account.

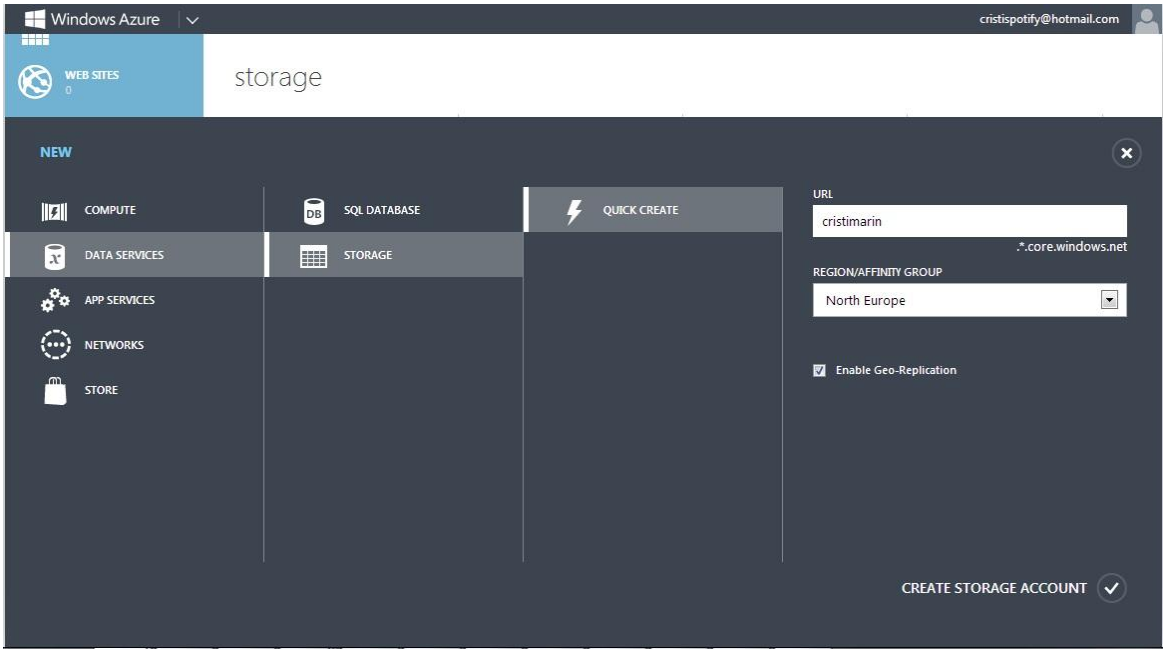

Figure 15. Creating a MWA storage account (Windowsazure.com 2012e)

Access to additional information about the account is granted when clicking on the name of the newly created storage account. The Dashboard menu presents an overview of the account. The Monitor menu is intended for monitoring and logging information regarding the storage account and can be activated by enabling various settings through the Configure menu. Once enabled, the Monitor menu is customizable through the *Add Metrics* button displayed at the bottom of the page. The last menu available here is called Containers. Selecting this menu offers the possibility to create new containers. If no containers exist, one can be created by selecting the *Create a blob container* option. The container must be assigned a name and an access type needs to be selected. Private access restricts the container to be accessed by users over the Internet, public container allows full read access to the container and blobs, and public blob only allows read access to the blob properties.

Deleting a storage account, a container or a blob can be performed by pressing the appropriate *Delete* button available on the bottom of the page. Further, uploading a blob is possible through .NET or Representational State Transfer (henceforth REST) programming (Microsoft 2012o).

Aside from managing blobs through .NET or REST programming, a third party solution is available. CloudBerry Explorer for Windows Azure (henceforth CBE) is a freeware

file manager software application that provides an alternative to programming for creating and managing blobs. CBE is developed by CloudBerry Lab and it can perform actions such as creating containers and blobs, uploading data from a local system to a blob or moving data between containers. CBE is available for download at http://www.cloudberrylab.com/free-microsoft-azure-explorer.aspx. In order to perform operations using blobs, the CBE application must be downloaded, installed and run. Figure 16 presents the CBE software default interface. (Cloudberrylab.com 2012.)

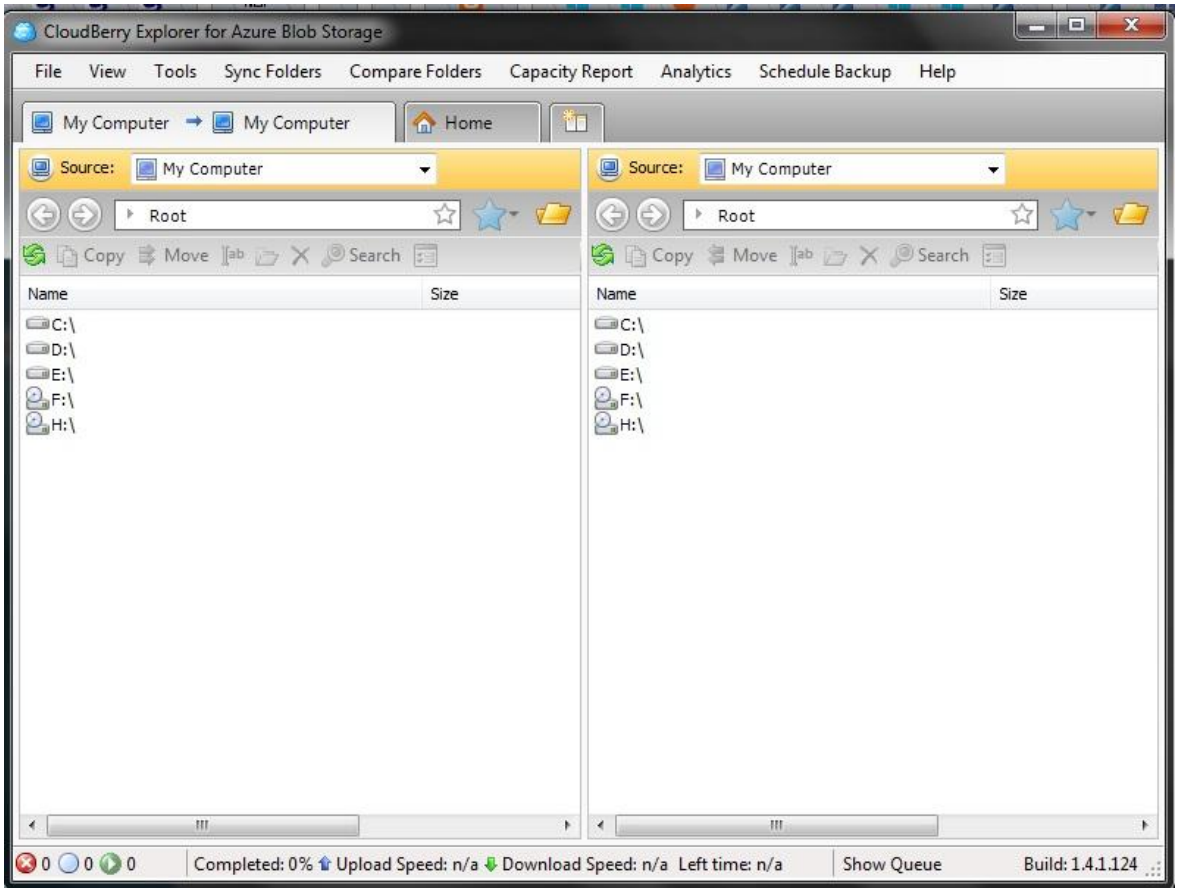

Figure 16. CBE Interface

Before proceeding, I recommend considering that utilizing the CBE application requires sharing access to the MWA storage account with the application itself. In order to upload data into a blob, first an MWA storage account needs to be associated in CBE. This can be done by opening the File menu and selecting Azure Blob Storage Accounts. In the new window, the *New Account* option has to be selected and the *Add* button has to be pressed. Then, several data are required, as shown in Figure 17.

Figure 17. Adding a storage account in CBE

The display name represents the name assigned to the storage account and it is only relevant for use within the CBE software. The account name needs to be identical with the MWA storage account name. The last input box requires a shared key to be entered. For this, the developer must log in into the MWA management portal, navigate to Storage and select an existing account or create a new one. At the bottom of the page, clicking on Manage Keys displays a window that contains the value of the Primary Access Key (henceforth PAK). The Manage Keys window is shown in Figure 18. Further, the PAK value needs to be entered into the Shared key input box in CBE. The MWA storage account is linked to the CBE software if the PAK is imported successfully.

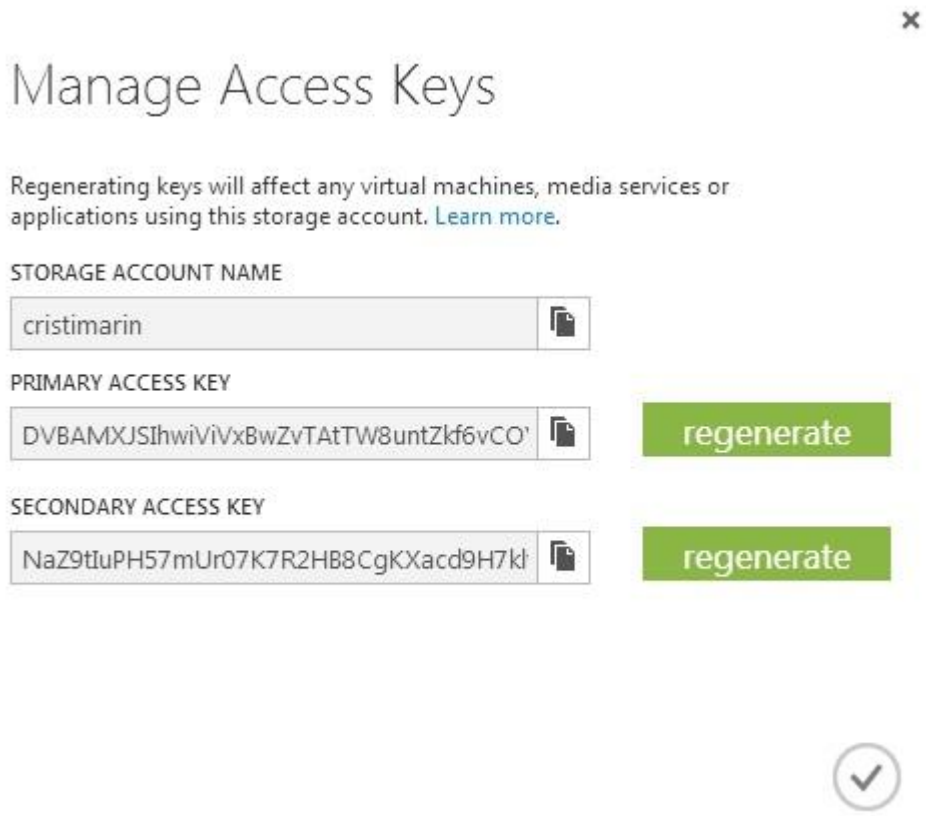

Figure 18. Manage Keys window in MWA (Windowsazure.com 2012e)

In order to upload data to a blob, one of the source systems in CBE needs to be set as the MWA storage account previously linked. If no containers exist, one can be created either through the management portal or CBE. Further, selecting a file or directory from the local system, right-clicking on the file and selecting Move performs the action of uploading the specified file into a blob. Once the file is uploaded, the blob containing the file is created and can be further managed through the management portal or CBE. For security reasons, I recommend regenerating the preshared key once CBE is no longer needed.

A second storage service available is table storage. Table storage is not a relational database nor is it a table. In fact, tables are collection of entities, in a form similar to a list, and are not required to follow a specific template which means that different tables can contain entities with different sets of properties. Figure 19 illustrates the infrastructure of a storage table. Tables do not have size limits but entities can have a maximum size of 1 MB. An entity can have a maximum of 255 properties. Similar to blobs, tables can only be created through .NET or REST programming. (Microsoft

2012p; Seroter et al. 2010, 123.) A good example that highlights the use of tables is using table storage for auto-complete suggestions, similar to Google's search box auto-complete features, because tables do not have size limits and are optimized for millions of entries (Dudley & Duchene 2010, 94).

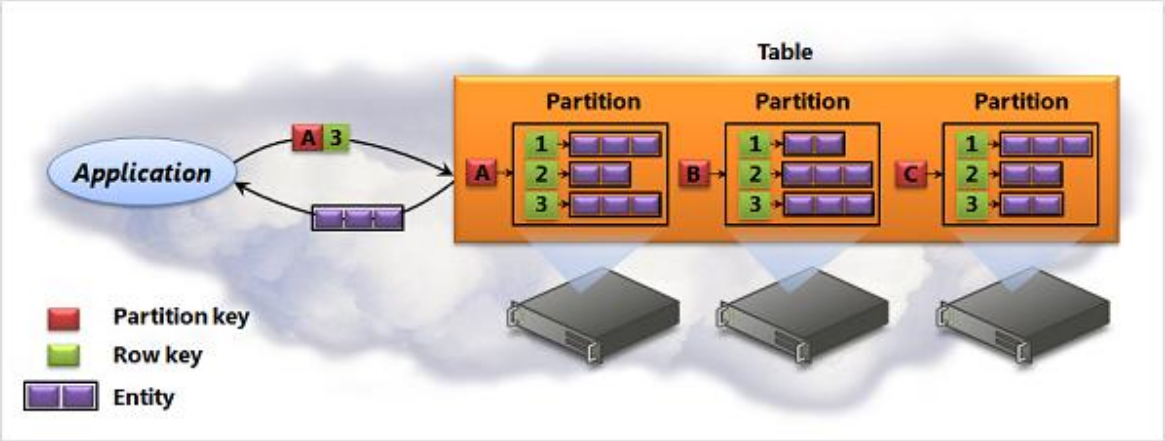

Figure 19. Infrastructure of a storage table (Microsoft 2012q)

Lastly, queue storage is not a traditional storage service. Queues represent areas for storing large numbers of messages in order to increase the performance and response times of worker roles. When several requests are received from a web role, a queue can be used to store the requests until a worker role instance is available to receive a request. By employing queues, worker roles are not overwhelmed with client requests. Furthermore, queues can be used to store requests failed to be processed as a result of one or more worker roles malfunctioning. The infrastructure of a queue contains the storage account and the queue itself. A queue can store countless messages but the messages can only be as large as 64 kilobytes (henceforth KB). Queues can be created using .NET and REST programming. (Microsoft 2012r; Dudley & Duchene 2010, 107.)

The configuration file of each storage type can be read by accessing the corresponding endpoint, available in the following format: http://storageaccount.servicetype.core.windows.net/, where storageaccount represents the name of the storage account and servicetype represents the type of service. As example, the correct URL for accessing a queue endpoint using a storage account named CristiMarin is http://cristimarin.queue.core.windows.net/. (How to use the Queue Storage Service 2012.) Service specific URLs are also available in the management

portal and can be found by selecting the appropriate storage account and service type. Additionally, local emulated storage services are available for testing purposes. The services are accessible through the following endpoints: http://127.0.0.1:10000/devstoreaccount1, http://127.0.0.1:10001/devstoreaccount1, http://127.0.0.1:10002/devstoreaccount1 for blobs, tables and queues, respectively (Microsoft 2011b).

In addition to blobs, tables and queues, MWA's latest storage service is the Azure Drive service. Azure Drive acts similar to a local NTFS file system. Data is stored in page-type blobs and data stored can be mounted as a virtual hard disk (henceforth VHD), therefore reducing the difficulty of migrating existing applications into the cloud. The size of VHDs ranges from 16MB to 1TB. VHDs can be mounted either as NTFS disks or as image disks, for installing operating systems other than the ones provided by MWA. Azure Drive is part of the VM service and, as of December 2012, it is still in beta stage. (Calder & Edwards 2010, 2-3; Microsoft 2012s.)

4.6 Azure SQL Database

Azure SQL Database (henceforth Azure SQL) is a cloud relational database and it is developed and maintained by Microsoft. Based on the technology of SQL Server 2008, Azure SQL is a highly scalable, highly available and secure database service. Unlike when deploying a local SQL server database, deploying an Azure SQL database only requires the developer to manage the logical server. Microsoft manages and maintains the physical resources necessary to run an Azure SQL database. Furthermore, because the developers do not have control over the physical resources, they cannot select the physical hard drive which stores the database files. However, the endpoint through which can be accessed remains the same. (Krishnaswamy 2010, 46-47.)

Azure SQL databases benefit from seamless automated resolution of hardware failures and automated data replication. In addition, Azure SQL is designed for increased performance and high scalability. Developers can create and deploy a database within

minutes and the size of the database is automatically scaled as the storage limit is reached, if the billing plan allows. (Seroter et al. 2010, 128.)

4.6.1 Creating and managing an Azure SQL database

An Azure SQL database can be created by accessing to the management portal, clicking *New*, selecting *Data Services* and selecting *SQL Database*. In the form that appears, an appropriate name needs to be assigned to the database and one of the two editions currently available needs to be selected. The Web edition database has a size limit of 1 or 5 GB, whereas the Business edition has a size limit of at least 10 GB but 150 GB at most. To finalize the creation process, a collation model and an SQL server need to be chosen. In the second form, appropriate credentials for the administrator account must be assigned. Furthermore, selecting the same region group as the one selected for the other MWA services that connect to the Azure SQL database can provide various advantages, such as low response times for processing client requests.

Designing or migrating an existing SQL database can be performed after accessing the SQL server previously created. From the management portal, developers can browse the SQL databases category and select Servers from inside the page. Then, the SQL management menu can be accessed by clicking on the server name and pressing the *Manage* button at the bottom of the page. Alternatively, the database server management portal can be accessed by visiting the public endpoint available in the format of https://servername.database.windows.net, where servername represents the name of the SQL server. In order to gain access to the server management portal, the database name and the account credentials assigned to the database administrator account need to be supplied. From the Azure SQL management portal, developers can design databases from scratch or import existing SQL databases. Figure 20 illustrates the default interface of the Azure SQL management portal.

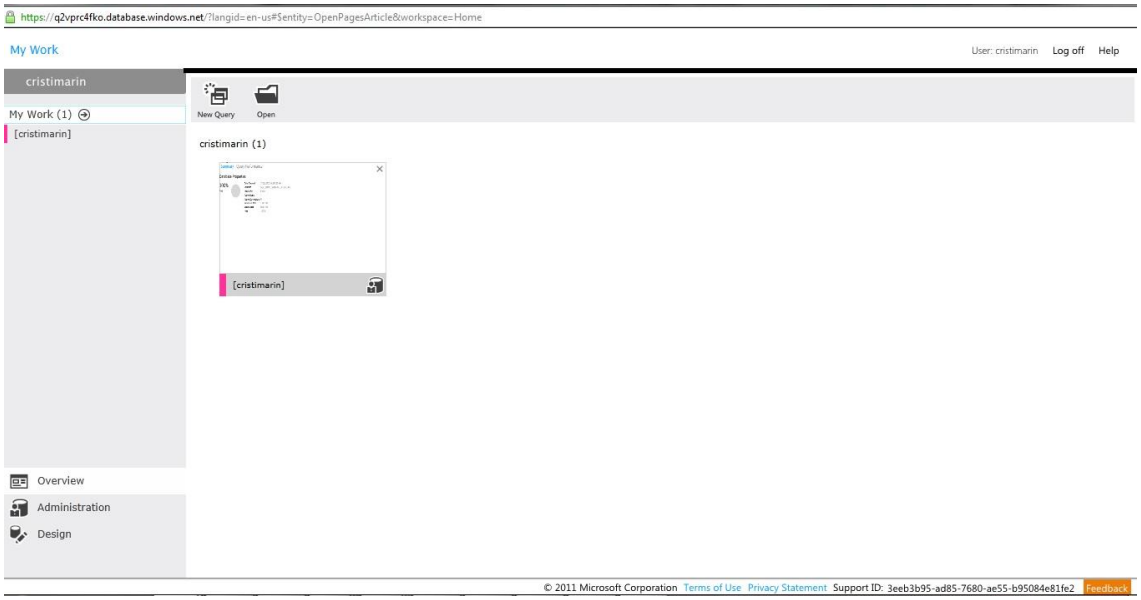

Figure 20. Azure SQL management portal

4.6.2 Azure SQL security aspects

The Azure SQL service is located behind a network firewall. By default, all connections to the Azure SQL server are disabled, with one exception. The IP address which was used to create the SQL server is allowed access, but only through port 1433. Developers must navigate to the Azure SQL database server configuration menu from the management portal and specify which IP addresses can access the database server. Furthermore, the database firewall allows developers to block MWA applications from accessing specific databases. (Microsoft 2012y.)

Another security feature is certificate validation. Azure SQL databases only accept communications encrypted and refuse connections from applications with invalid security certificates. Furthermore, established connections with Azure SQL are reset every hour, therefore developers are required to re-enter the database administrator account credentials. (Microsoft 2012y.)

Additionally, Microsoft recommends developers to use the latest version of tools, to prevent security vulnerabilities that have already been fixed. As an added security measure, developers are recommended to block inbound network connections on port

1433, as only outbound connections are required to communicate with Azure SQL. (Microsoft 2012y.)

4.7 Azure AppFabric

The Azure AppFabric is an important component of the MWA architecture and it provides services that separate the MWA platform from other cloud platforms. It represents the middle layer of the MWA platform, serving as a communication bridge between applications running on different systems. One benefit of utilizing AppFabric services is that it simplifies the application management process, thus increasing development productivity. The two main components of the AppFabric are Service Bus and Access Control. Developers can use the services provided by the AppFabric to securely connect application components together, i.e. different roles, and easily manage the access to resources. These two services are essential for most cloud applications and they rely on and complement each other, in order to achieve the best connectivity results. (Brunetti 2011, 161-162.)

4.7.1 Service Bus

The Service Bus provides the infrastructure necessary for building cloud applications that need to communicate with clients or other applications over the Internet. The messaging service provided by the Service Bus simplifies the application development process by allowing developers to take advantage of built-in features such as publicly exposed endpoints. As illustrated in Figure 21, the Service Bus is built around a centralized relay service. The Service Bus functions in such a way that applications send messages to the bus itself and then, any application with appropriate access right is able to connect to the bus and retrieve any message. By utilizing public endpoints as the destination for messages, applications are not engaged in direct communication with the client making the request. Instead, the Service Bus opens a uni- or bidirectional communication channel that provides the means to overcome common communication security challenges such as bypassing network firewalls or having to enable closed

communication ports. The Service Bus capabilities can provide advantages for any application but it can be especially useful for instant messaging, file sharing or similar applications. (Skonnard 2009, 2-8.; Microsoft 2012u.)

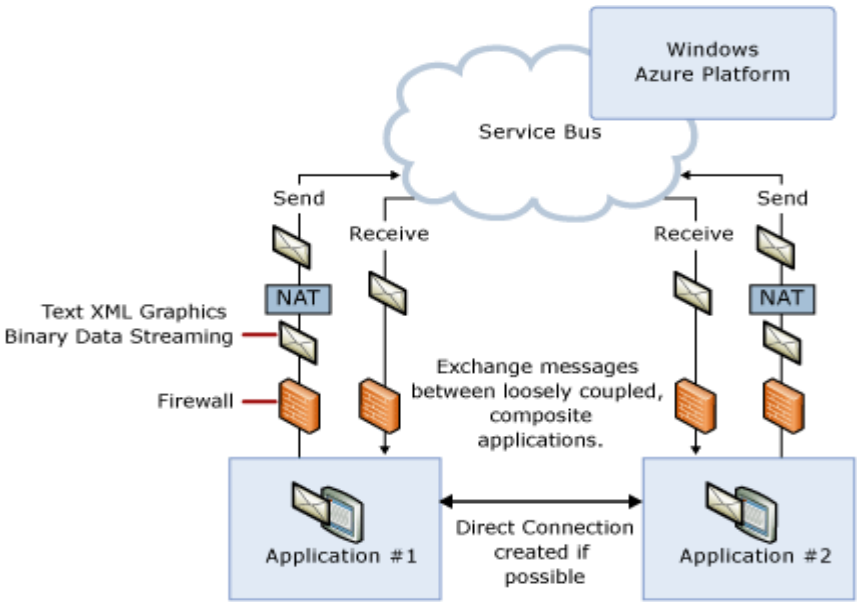

Figure 21. Service Bus architecture model (Microsoft 2012u)

There are two types of Service Bus messaging, namely relayed and brokered. The relayed messaging service acts on a request-response basis, providing unidirectional communication, whereas the brokered messaging service provides asynchronous, bidirectional communication. The latter type of Service Bus messaging can be particularly useful for applications that are not running simultaneously. Messages are stored in tables, queues or subscriptions, allowing applications to retrieve messages at any moment. (Microsoft 2012v.)

Service Bus messaging services can be created through the management portal, by pressing the *New* button at the bottom of the page and selecting the *App Services* category. Further, the *Service Bus Relay* category allows assigning an appropriate namespace name and region. New relays, queues and topics can be created by clicking on the assigned namespace name. To connect an application to a namespace, a connection string key which can be found from the management portal is required.

The Service Bus provides secure messaging infrastructure that supports a wide variety of network communication protocols. Though it offers great development possibilities, the Service Bus is only optimized for usage with .NET applications. (Skonnard 2009, 7-8.) Furthermore, I recommend exercising caution when employing the Service Bus services with applications that perform heavy network communication. The Service Bus service is billed based on the number of communication requests, which may increase the costs of running a MWA application.

4.7.2 Access Control

The Access Control service (henceforth ACS) provides developers with an easy method of authenticating and authorizing clients that require access to the cloud application. By employing ACSs, developers are no longer needed to implement authentication code into their applications. Furthermore, Access Control allows developers to implement open authentication protocols. This means that applications can support authentication with accounts other than the ones created for the application itself. Clients can use accounts provided by several popular websites, such as Google or Facebook, referred to as identity providers. (Microsoft 2011c.)

Access Control does not require developers to modify the application code in order to implement the ACSs. Instead, developers need to manage trust relationships with different identity providers. The features provided by Access Control are efficiently integrated with the Service Bus, thus significantly easing the difficult process of developing own security mechanisms. (Dudley & Duchene 2010, 179.)

4.8 Application deployment

Applications can be deployed in two different ways. However, both deployment processes require the use of Microsoft Visual Studio or other development environment supported by the MWA platform. Before starting the deployment process, developers need to consider the different options for selecting an affinity group. An affinity group represents the geographical location in which the application is hosted. Currently,

MWA features six data centres in various global regions, as illustrated in Figure 22. Selecting an identical affinity group for all the application roles and storage services provides benefits such as lower latency times and faster responses to client requests. Furthermore, Microsoft does not charge for data transfer between applications hosted in the same region. Another factor that needs to be considered when selecting an affinity group is the location of most of the users that access the application. (Dudley & Duchene 2010, 212.)

Figure 22. MWA data centres locations (Microsoft 2011b)

The first method of deploying an application is directly from the development environment, Visual Studio. Right-clicking on an open project's name and selecting *Publish* opens the publishing settings window. First, developers need to download the MWA subscription profile by clicking on the *Sign in to download credentials* link at the top of the *Publish* window. A website is opened and a file that contains the account credentials and information is automatically downloaded. Then, the recently downloaded configuration file must be imported. Secondly, the Settings configuration contains parameters for deploying the application to the cloud. An existing cloud service must be selected or a new one must be created, followed by setting the build and service type. For the purpose of building a sample application, I selected the *Release* build and the *Cloud* service type. Additionally, one of the two deployment environment types available needs to be selected. The staging deployment allows developers to deploy the application in an environment designed for testing purposes, whereas the

production deployment represents deploying the application to a live environment. The latter deployment type allows the application to be accessed through the public endpoint assigned to the application. If an application is deployed in the staging environment, it can be upgraded to production deployment through the management portal (Microsoft 2012w.) RPD connections can also be enabled from this Settings configuration page. After selecting all the appropriate parameters, a summary of the selections made previously is available by pressing the *Next* button. Pressing the *Publish* button deploys the application. Figure 23 shows the summary of the deployment parameters that I configured. If an application is deployed using the Publish menu from Visual Studio, I recommend waiting approximately five minutes after the publishing process has finished. Synchronization between the cloud and the local development environment might be delayed, depending on the speed of the local Internet connection or for configuration purposes.

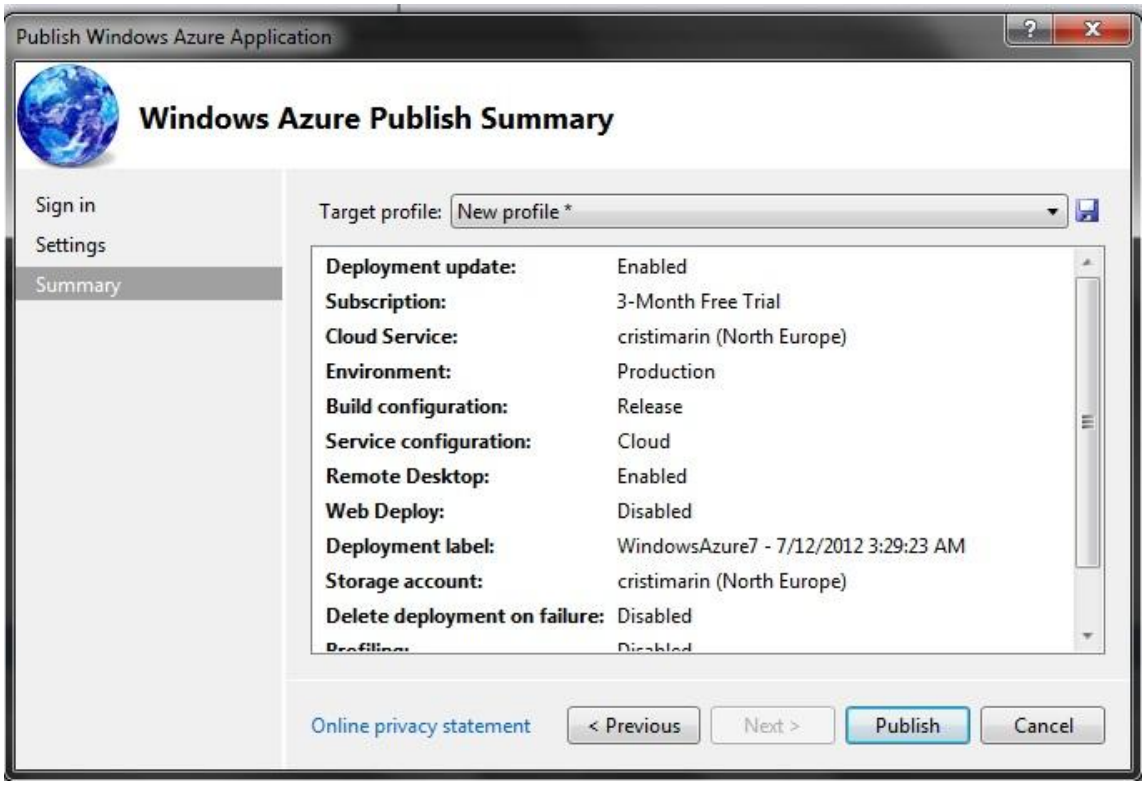

Figure 23. Deployment parameters summary

The second method of deploying an application is through the management portal, by selecting the Cloud Services category. After pressing the *New* button, developers must select *Custom Create* and assign an appropriate name for the cloud service. Further, an

affinity group must be selected and the box corresponding to the *Deploy a cloud service package now* option needs to be checked. In case a cloud service is already created, clicking on its name opens the menu which contains the deployment options. In the form required to be filled in order to complete the deployment process, two files are required and several other parameters need to be configured.

Deploying an application using this method requires the developer to package the application utilizing the development environment of choice, Microsoft Visual Studio 2010 in this case. After having developed the application itself, developers must open the project in Visual Studio and right-click on the project name. Selecting Package opens the menu that allows creating the files needed to deploy the application. The appropriate service and build configuration types need to be selected. Additionally, RDP connections can be enabled at this stage, if needed. Pressing the *Package* button starts the packaging process which results in two files being created. The package and configuration files have the extensions .cspkg and .cscfg, respectively. The files are located in the My Documents directory on the local system, under the following path: \Documents\Visual Studio 2010\Projects\ \projectname\solutionname\bin\ReleaseOrDebug\app.publish, where accountname represents the name of the project and solutionname represents the name assigned to the solution when the project was created. Having successfully created the application package, the package and configuration file must be selected in the management portal, as shown in Figure 24.

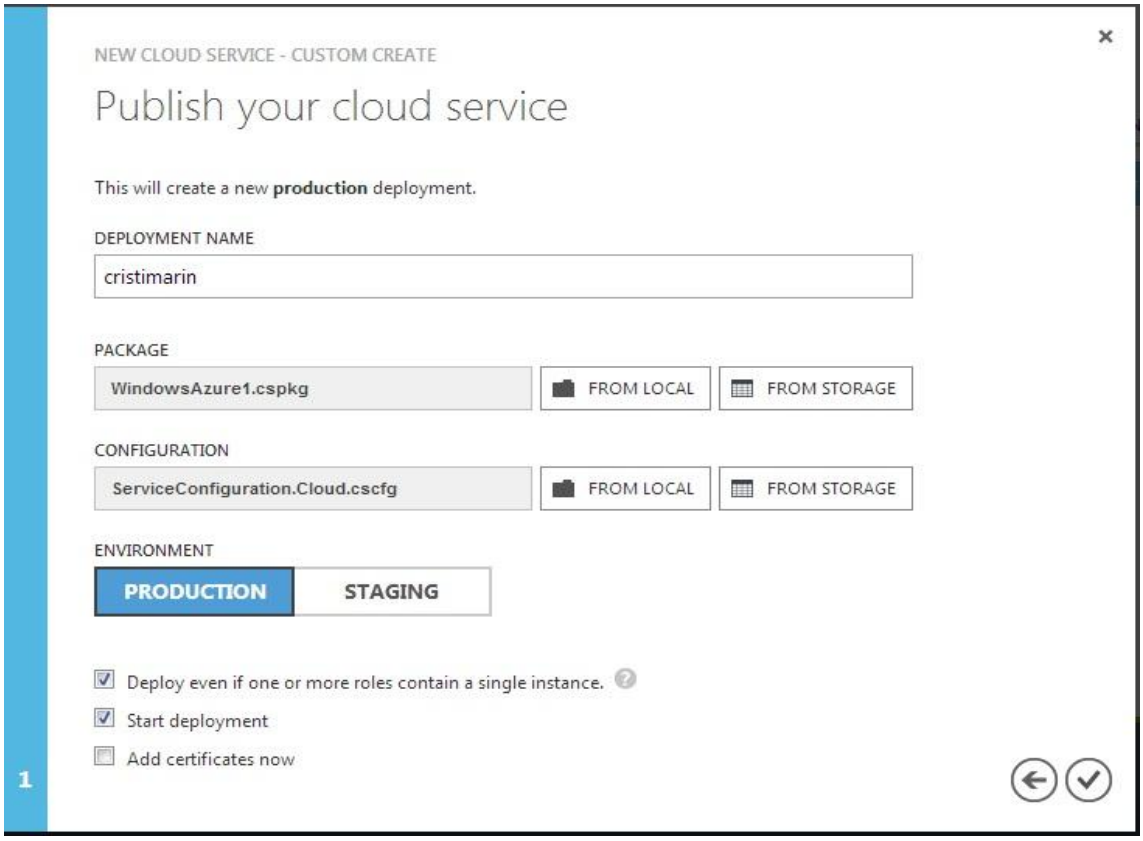

Figure 24. Application deployment from the management portal (Windowsazure.com 2012f)

Further, a deployment type needs to be selected. Another parameter that needs to be considered when deploying the application is the number of instances each application role runs in. If any of the roles are intended to run a single instance, the "Deploy even if one or more roles contain a single instance" option needs to be checked. Microsoft recommends running at least two instances of any role, to avoid performance issues in case one instance malfunctions. (Windowsazure 2012f.) The *Complete* button finalizes the application deployment process. The application can be accessed through the Internet by accessing the Internet address available in the following format http://cloudservicename.cloudapp.net, where cloudservicename represents the name assigned to the cloud service. The sample application that I built and deployed for the purpose of this research is available by accessing http://cristimarin.cloudapp.net.

If RDP connections have been enabled during the deployment process, developers are able to connect to the VM each role instance is running in. In order to connect to a VM, developers must select the appropriate cloud service and open the Instances menu.

Then, an instance needs to be selected and pressing the *Connect* button at the bottom of the page starts the download of a configuration file. The connection is established by opening the file and entering the credentials that have been set up during the deployment process and a successful connection should be established.

Updating an application can easily be performed without having to recreate the cloud service. First, the application needs to be repackaged using Visual Studio. Secondly, the new .cspkg and .cscfg files have to be uploaded from the management portal using the *Update* button at the bottom of the page corresponding to the cloud service of choice. Lastly, the application is automatically restarted to apply the changes made.

4.9 Add-ons Store

The add-ons store is a recent addition to the already wide array of services offered by the MWA platform. As of October 2012, the Windows Azure store provides developers with third party application services and data add-ons that improve and allow for customization of the core MWA services. Though only available in beta stage, the store can be accessed through the management portal. (Zhiming 2012.)

The store contains a wide variety of add-ons which are suitable for various purposes and can significantly improve the developers' experience. Among other add-ons that are currently featured on the add-ons store website, applications such as ClearDB and MongoLab, offered as database services, are cloud versions of the popular database solutions available under the same name. Other application services available are SendGrid, providing cloud email services, and AppDynamics, providing monitoring and scaling services. To complement the add-ons collection, data add-ons such as BingSearch and Microsoft Translator provide the means to integrate search engine results and automated text translation, respectively, into a cloud application. (Microsoft 2012t; Zhiming 2012.)

The add-ons store is implemented in beta stage and it is only available to legal residents of the United States of America (USA) (Microsoft 2012t). Despite the current regional limitations, as the add-ons selection becomes more and more varied, the use of third

party add-ons provide multiple development possibilities and research is worth conducting regarding these possibilities.

5 SECURITY OF THE AZURE CLOUD ENVIRONMENT

The security of the MWA platform is the focus of this chapter. First, an overview of the security services implemented into MWA is presented and common security issues are discussed. Then, monitoring and diagnostics tools are explored. Finally, a security development process proposed by Microsoft is discussed.

5.1 Microsoft Windows Azure security services overview

MWA is designed to reduce the infrastructure management responsibilities, allowing developers to focus on developing applications. Microsoft aims to provide high CIA standards, thus several security mechanisms are implemented in the MWA platform. First, data confidentiality is enhanced through authenticating mechanisms such as secure authentication on the management portal and unique, randomly generated storage account keys. Moreover, data is physically or logically divided on Microsoft's servers, offering increased security in case the security of one server is compromised. Developers might also opt to encrypt their data using strong encryption algorithms. Secondly, the primary mechanism for maintaining the integrity of data is the way data is replicated on three VHDs. Data stored on two of the three VHDs cannot be delete or overwritten, while the data on the primary VHD is secured through the configuration file uploaded by the developer when deploying the application. Developers can restrict access to specific application roles through this configuration file. Lastly, the availability of data is ensured through automated data replication mechanisms. (Kaufman & Venkatapathy 2010, 7-14.)

At platform and infrastructure layers, the MWA platform automatically resolves several security threats, without requiring any action from the developer. The first security measure employed at these levels is a firewall that blocks any communication ports that are not specifically open is enabled on each VM that runs an application role. Another severe security threat, denial of service (henceforth DoS) attacks are handled by MWA particularly well. Due to automated load balancing services and high scalability by increasing the number of role instances, MWA can handle excessive network traffic with ease. (Marshall et al. 2010, 11-12.)

The service layer is susceptible to common Internet attacks. As applications are publicly exposed to clients through Internet protocols, Microsoft can only ensure the security of the cloud environment. Therefore, developers are required to adopt best practices when writing code for applications. At the service layer level, the most common security measures that developers can employ are securing application data, auditing and logging events, validating input data, utilizing valid SSL certificates and encrypting stored data.

The recommendations that follow are drawn from the referenced sources listed accordingly, as well as from my experience and knowledge. First, application data can be secured by applying ACLs that restrict access to the application itself. Only the developers that require direct access to the application code and data should be given access rights. Furthermore, developers are recommended to only allow the application to communicate with users through specific ports and only through HTTPS protocols, when possible. (Marshall & Howard & Bugher & Harden 2010, 8-10.)

Secondly, auditing and logging are security measures that developers should enable at every levels of the application code where authentication and authorization mechanisms function. This allows the developers to verify that the application functions correctly and to diagnose problems, such as poor performance issues or external attacks over the Internet. Furthermore, developers are recommended to store event logs using the MWA storage service. More specifically, table storage provides the best solution for storing large amounts of event logs. Storing event logs on the VM that the application role runs in is not a viable practice. As the VM physical location might change when the VM is restarted, the developer loses access to the event log files. Developers are recommended to periodically analyze all the log files, as this allows them to recognize abnormal usage levels. (Marshall et al. 2010, 10.)

Thirdly, developers are recommended to consider all input data received from clients as invalid or malicious. All data should be validated before the application processes it. Common practices for validating input data include validating the length, type and format in which data is entered, and restricting the range of characters which can be used when data is sent from a client. (Marshall et al. 2010, 10-11.)

Lastly, developers are recommended to avoid storing sensitive data such as passwords and storage PAKs. Furthermore, developers should encrypt all data stored and all data transferred over the Internet by utilizing valid SSL certificates or by encrypting the data prior to transferring. To increase the security of data, storage PAKs, passwords and other similar sensitive data should be periodically renewed. (Marshall et al. 2010, 10.)

At platform and infrastructure levels, MWA implements automated security prevention and mitigation measures. The MWA platform is protected against common Internet threats such as spoofing, DoS and eavesdropping. The first security measure implemented by MWA is restricting communication over all the ports that are not explicitly enabled. Developers are recommended to enable only the ports that are required for the application to function. This security measure minimizes the methods through which potential attackers might gain access to application data. (Marshall et al. 2010, 11.)

Further, spoofing represents attacks in which the attacker attempts to gain access to an application by providing a false identity, which can be acquired by hijacking client sessions and stealing valid client credentials. The MWA platform features multiple firewalls that are filtering communication at various architecture levels. In addition to protection from spoofing attacks, several protection mechanisms against DoS attacks are implemented in the MWA platform. The automated load balancing service performed by the AppFabric partially mitigates DoS attacks by distributing incoming client requests to all application role instances running. To complement the MWA security measures against DoS attacks, developers are recommended to run more than one instance of any application role. Moreover, the MWA automated security services block any communication from clients that are detected as being malicious. ((Marshall et al. 2010, 11.)

Appendix 1 contains a list of security threats that concern the MWA cloud environment. Moreover, the list specifies which layer of the MWA architecture is susceptible to a specific type of threat and also provides recommended actions that the developers might need to perform in order to mitigate or avoid the threat. To summarize the contents of Appendix 1, the most relevant threats can be categorized as spoofing, information disclosure, denial of service, and elevation of privilege threats.

5.2 Azure monitoring and diagnostics

The MWA platform provides performance monitoring tools for all the cloud services deployed into the cloud. Several performance-monitoring metrics can be enabled without implementing any custom code into the application, but only CPU usage percentage is monitored by default. Monitoring statistics can be viewed by accessing the management portal, selecting a cloud service and opening the Monitor menu. Additional monitoring metrics can be added by pressing the *Add Metrics* button at the bottom of the page. Other metrics include Disk Read, Disk Writes, Network In and Network Out, as illustrated in Figure 25. (Microsoft 2012x.)

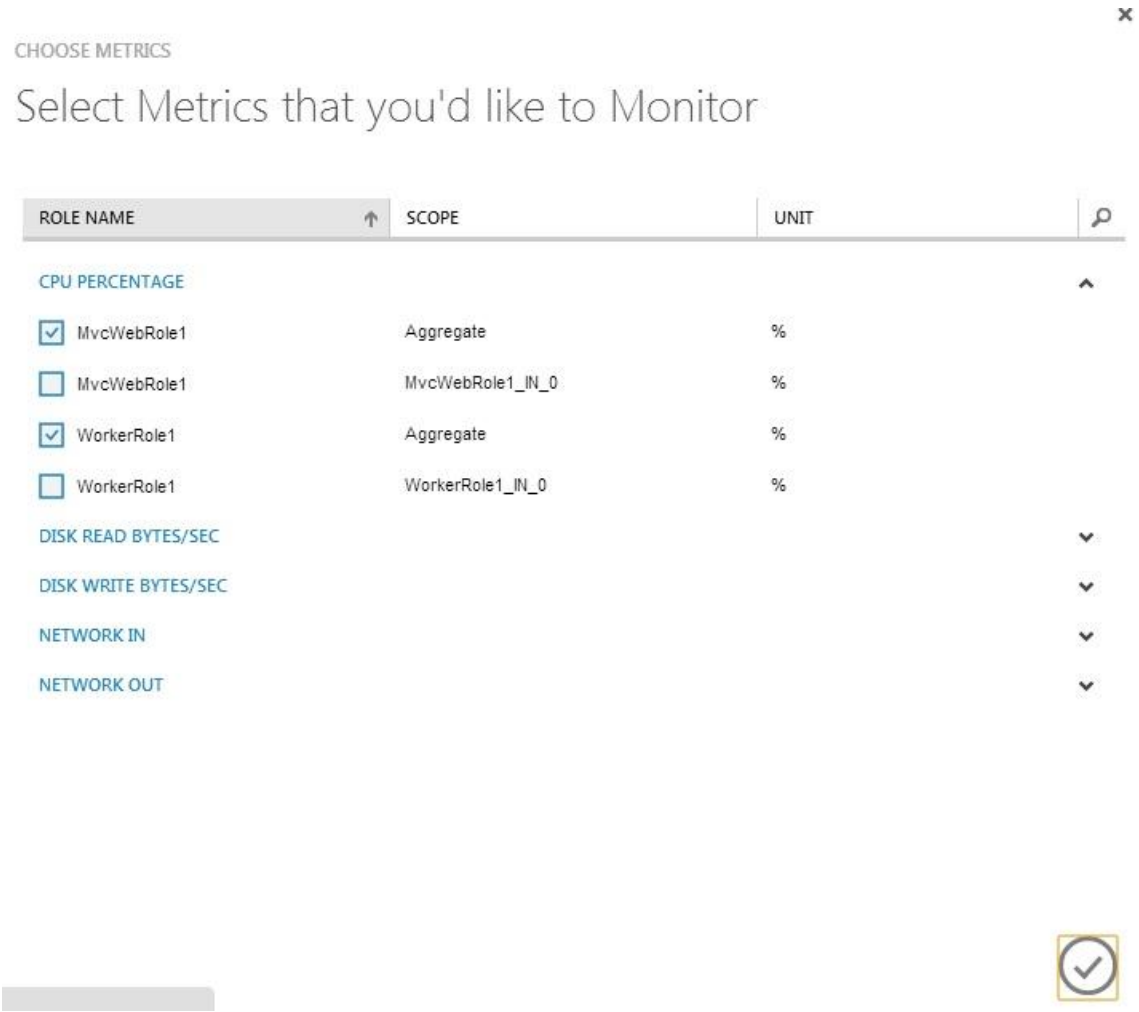

Figure 25. Monitoring metrics (Windowsazure 2012g)

In addition to the default monitoring features, diagnostics services can be enabled by adding custom code to the application. The features provided by MWA Diagnostics allow developers to gather data which can be analyzed to improve the performance of

the application or to save logs concerning various malfunctions that the application might experience. (Seroter et al. 2010, 126.)

For an overview of the availability of the MWA services, the MWA dashboard is available at https://www.windowsazure.com/en-us/support/service-dashboard/. Through the dashboard, developers can monitor the status of various MWA services and their corresponding region. Even though the MWA platform is highly available and does not have a lot of downtime, specific services in specific regions are constantly affected miscellaneous issues that cause total or partial service interruptions. During the months of October and November 2012, certain services have been interrupted regionally or globally. However, the services are quickly restored, the issue resolve time being one hour on average.

5.3 Microsoft Secure Development Lifecycle (SDL)

SDL is a software development model developed by Microsoft. It aims to decrease the number of vulnerabilities in software applications by following security and privacy best practices throughout the development process. Adopted by Microsoft themselves, the SDL is a seven-step process that can be applied to cloud applications, among other types of applications. To provide efficient results, the SDL activities must be performed together, as a set. Furthermore, developers should focus on achieving complete and high quality results instead of hastening the SDL process. (Microsoft 2010b, 3-5.)

Microsoft has made a simplified version of the SDL publicly available, free of charge. Prior to implementing the simplified SDL activities, illustrated in Figure 26, the development team must be properly informed regarding the current security trends. Basic training should cover topics such as secure design, threat modelling, secure coding, secure testing and privacy. (Microsoft 2010b, 6.)

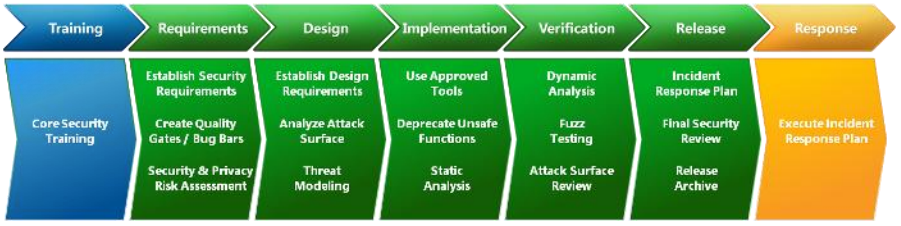

Figure 26. SDL process (Microsoft 2010b, 6)

While the SDL process proves to be beneficial for some applications, it may not be suitable for any application. Microsoft recommends developers to use the SDL as a guide and does not guarantee solid results. Furthermore, the SDL activities are designed for implementation with any OS or development platform. Through SDL, Microsoft managed to create a development methodology with wide applicability that lowers development costs, in addition to increasing the security of applications. (Microsoft 2012b, 13-15.)

6 CONCLUSIONS

Cloud computing is an important field in IT and the range of services available is extremely wide. Adopting a cloud solution for deploying applications proves to be beneficial for both ISD and companies specialized in software development, in contrast with running applications locally. As a result of the rapid evolution of cloud computing environments, several IT companies developed their own cloud solutions. A wide variety of cloud services are currently available, however no cloud service provider manages to satisfy the needs of all software developers. Therefore, the main objective of this thesis is to examine and explore the MWA platform for developing cloud applications. Furthermore, this thesis discusses the process of developing cloud applications and builds a basic sample application, to serve as a demonstration of the capabilities of the MWA platform. The application I deployed in the MWA environment is available at http://cristimarin.cloudapp.net/ until the end of January 2013. Finally, discussions are made regarding the security implications of deploying cloud applications inside the MWA cloud environment.

To summarize the findings of the research, MWA is a cloud platform created and maintained by Microsoft. Fundamentally, cloud platforms provide compute and storage services. The services provided by the MWA platform are delivered as PaaS and IaaS services and provide for developers the means to build, deploy and manage applications. In contrast with traditional application development, employing the services offered by MWA can be beneficial for software developers. The technology integrated in MWA allows for a fast deployment process and provides an efficient method of developing cloud applications, by implementing automated services.

This research concludes that the MWA platform provides a diverse development environment for developing applications. Unlike the limited cloud services offered by two of the most important competitors against Microsoft, namely Amazon and Google, the MWA platform allows developers to use a large selection of programming languages and OSs. Furthermore, the development tools and support differentiate the MWA platform from other cloud platforms and create a more managed experience. Important aspects of developing cloud applications are emphasized through the programming model that MWA adopts. In MWA, applications are built from the

premise that any part of the application can be assigned one of the three different roles available. Applications should be optimized to run multiple instances of each role in order for the other instances to successfully handle the remainder of the tasks, in the eventually one instance malfunctions. Therefore, software developers can build highly scalable, highly available and easily manageable applications.

While the MWA platform offers a plethora of benefits for software developers, an important finding is that the MWA platform is not suitable for developing small-sized applications. The services provided by the MWA platform support developing applications of any size and satisfy the needs of many software developers, from technical and functional points of view. However, the pricing model of the MWA services does not reflect Microsoft's intentions of attracting ISDs. The monthly costs of running a small sized application in MWA can reach or exceed values close to 100€, whereas Google offers enough free resources for developers to run an application for unlimited periods of time. Moreover, poor performance issues might arise if the applications are not optimized to follow the MWA recommended programming model. Additionally, this research emphasizes that the Microsoft Visual Studio development environment is highly recommended for developing applications for MWA. Complete and premium versions of Visual Studio are available for purchase from Microsoft, though an express version is also available free of charge.

Further, this research demonstrates that MWA is a secure cloud environment by highlighting crucial security features. The data centres managed by Microsoft provide a stable and secure cloud environment, as Microsoft employs various globally recognized security certifications and security best practices. In addition, Microsoft provides support for developers on how to increase the security and performance of cloud applications.

The MWA platform represents Microsoft's solution for cloud services. As the platform is frequently updated, further research possibilities arise regarding newly implemented or future technologies integrated into the platform. Moreover, additional research can be conducted on other cloud platforms and on the implications of employing specific cloud solutions.

REFERENCES

Amazon 2012. Amazon Elastic Compute Cloud. Downloaded October, 2012. <http://aws.amazon.com/ec2/>

Brunetti, Roberto 2011. Windows Azure™ Step by Step. USA: O'Reilly Media, Inc.

Calder, Brad & Edwards, Andrew. Windows Azure Drive. Downloaded November, 2012. <http://go.microsoft.com/?linkid=9710117&clcid=0x409>

Chappell, David 2010. Introducing Windows Azure. Downloaded October, 2012. <http://go.microsoft.com/?linkid=9682907&clcid=0x409>

Chappell, David 2009. Windows Azure and ISVs – A Guide for Decision Makers. Downloaded October, 2012. <http://download.microsoft.com/download/F/9/E/F9EAD956-18D1-42D8-AB1C-7F119856ABBF/Windows%20Azure%20for%20ISVs,%20v1.2--Chappell.pdf>

Cloudberrylab.com 2012. Downloaded November, 2012. <http://www.cloudberrylab.com/free-microsoft-azure-explorer.aspx>

Cloudsleuth.com 2012. Downloaded December, 2012. <https://cloudsleuth.net/global-provider-view>

Cunningham, Sean R 2010. Cloud Optimization – Expanding Capabilities, while Aligning Computing and Business Needs. Downloaded November, 2012. <http://go.microsoft.com/?linkid=9751401&clcid=0x409>

Dudley, Richard J. & Duchene, Nathan A. 2010. Microsoft Azure: Enterprise Application Development. Great Britain: Packt Publishing Ltd.

Google 2012. What is Google App Engine? Downloaded October, 2012. <https://developers.google.com/appengine/docs/whatisgoogleappengine>

Hurwitz, Judith & Bloor, Robin & Kaufman, Marcia 2009. USA: Cloud Computing For Dummies.

Kaufman, Charlie & Venkatapathy, Ramanathan 2010. Windows Azure Security Overview. Downloaded November, 2012. <http://download.microsoft.com/download/6/0/2/6028B1AE-4AEE-46CE-9187-641DA97FC1EE/Windows%20Azure%20Security%20Overview%20v1.01.pdf>

Krishnaswamy, Jayaram 2010. Microsoft SQL Azure Enterprise Application Development. Great Britain: Packt Publishing Ltd.

Krishnaswami, O.R. & Satyaprasad, B.G. 2010. Business Research Methods. India: Global Media.

Krutz, Ronald L. & Vines, Russell Dean & Brunette, Glenn 2010. Cloud Security: A Comprehensive Guide to Secure Cloud Computing. USA: Wiley.

Laing, Bill 2012. Announcing New Windows Azure Services to Deliver "Hybrid Cloud". Downloaded November, 2012. <http://blogs.msdn.com/b/windowsazure/archive/2012/06/06/announcing-new-windows-azure-services-to-deliver-hybrid-cloud.aspx>

Lu, Jie & Zhang, Guangquan & Ruan, Da 2007. Multi-Objective Group Decision Making: Methods Software and Applications with Fuzzy Set Techniques. Singapore: Imperial College Press.

Marks, Eric A. & Lozano, Bob 2010. Cloud Computing. USA: Wiley.

Marshall, Andrews & Howard, Michael & Bugher, Grant & Harden, Brian 2010. Security Best Practices for Developing Windows Azure Applications. Downloaded November, 2012. <http://go.microsoft.com/?linkid=9751405&clcid=0x409>

McDonald, Kevin 2010. Above the Clouds: Managing Risk in the World of Cloud Computing. Great Britain: IT Governance.

Microsoft 2010a. Configuring Connection Strings. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/ee758697.aspx>

Microsoft 2010b. Simplified Implementation of the SDL. Downloaded December, 2012. <http://download.microsoft.com/download/F/7/D/F7D6B14F-0149-4FE8-A00F-0B9858404D85/Simplified%20Implementation%20of%20the%20SDL.doc>

Microsoft 2011a. How to Configure Local Storage Resources. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/ee758708.aspx>

Microsoft 2011b. Storage Infrastructure 2011. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/hh508989.aspx>

Microsoft 2011c. Azure Services 2.0. Downloaded December, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/gg429786.aspx>

Microsoft 2012a. Windows Azure vs. AWS - What's Different and What's the Impact on Your Startup. Downloaded October, 2012. <http://www.microsoft.com/BizSpark/Azure/AzureAWS.aspx>

Microsoft 2012b. What's New in Windows Azure. Downloaded October, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/gg441573.aspx>

Microsoft 2012c. Introducing Windows Azure. Downloaded October, 2012. <https://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/>

Microsoft 2012d. Pricing details 2012. Downloaded October, 2012. <https://www.windowsazure.com/en-us/pricing/details/?currency-locale=de-de>

Microsoft 2012e. Pay-As-You-Go 2012. Downloaded November, 2012. <https://www.windowsazure.com/en-us/offers/ms-azr-0003p?currency-locale=de-de>

Microsoft 2012f. 6-Month Plan. Downloaded November, 2012. <https://www.windowsazure.com/en-us/offers/ms-azr-0037p?currency-locale=de-de>

Microsoft 2012g. 12-Month Plan. Downloaded November, 2012. <https://www.windowsazure.com/en-us/offers/ms-azr-0039p?currency-locale=de-de>

Microsoft 2012h. 6-Month Plan (Prepaid). Downloaded November, 2012. <https://www.windowsazure.com/en-us/offers/ms-azr-0038p?currency-locale=de-de>

Microsoft 2012i. 12-Month Plan (Prepaid). Downloaded November, 2012. <https://www.windowsazure.com/en-us/offers/ms-azr-0040p?currency-locale=de-de>

Microsoft 2012j. Windows Azure Support. Downloaded November, 2012. <https://www.windowsazure.com/en-us/support/plans/>

Microsoft 2012k. Unit Testing in ASP.NET MVC Applications. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/gg416510(VS.98).aspx>

Microsoft 2012l. Configuring a Windows Azure Project. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/ee405486.aspx>

Microsoft 2012m. How to Use: Windows Azure Caching. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/jj131262.aspx>

Microsoft 2012n. Virtual Machines. Downloaded October, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/jj156003.aspx>

Microsoft 2012o. How to Use the Windows Azure Blob Service in .NET. Downloaded November, 2012. <https://www.windowsazure.com/en-us/develop/net/how-to-guides/blob-storage/#header-11>

Microsoft 2012p. How to Use the Table Storage Service. Downloaded November, 2012. <https://www.windowsazure.com/en-us/develop/net/how-to-guides/table-services/>

Microsoft 2012q. Data Management and Business Analytics. Downloaded November, 2012. <https://www.windowsazure.com/en-us/develop/net/fundamentals/cloud-storage/>

Microsoft 2012r. How to Use the Queue Storage Service. Downloaded November, 2012. <https://www.windowsazure.com/en-us/develop/net/how-to-guides/queue-service/>

Microsoft 2012s. Using the Windows Azure Storage Services 2012. Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/ee924681.aspx>

Microsoft 2012t. Windows Azure Store. Downloaded December, 2012. <https://www.windowsazure.com/en-us/store/overview/>

Microsoft 2012u. About the Windows Azure Service Bus. Downloaded December, 2012. <http://msdn.microsoft.com/en-us/library/hh690929.aspx>

Microsoft 2012v. Windows Azure Service Bus. Downloaded December, 2012. <http://msdn.microsoft.com/en-us/library/ee732537.aspx>

Microsoft 2012w. Staging an application in Windows Azure. Downloaded December, 2012. <https://www.windowsazure.com/en-us/develop/other/common-tasks/enable-staging-deployment/>

Microsoft 2012x. How to Monitor Cloud Services. Downloaded December, 2012. <https://www.windowsazure.com/en-us/manage/services/cloud-services/how-to-monitor-a-cloud-service/>

Microsoft 2012y. Security Guidelines and Limitations (Windows Azure SQL Database). Downloaded November, 2012. <http://msdn.microsoft.com/en-us/library/windowsazure/ff394108.aspx>

Mell, Peter & Grance, Timothy 2011. Computer Security. USA: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. Downloaded May, 2012. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

Robinson, Neil & Valeri, Lorenzo & Cave, Jonathan 2011. Cloud. USA: RAND Corporation.

Sachdeva, J.K. 2009. Business Research Methodology. India: Global Media.

Sarna, David E.Y. 2011. Implementing and Developing Cloud Computing Applications. USA: Auerbach Publications. Downloaded August, 2012. <http://ca.chitkara.edu.in/cloudsecurity/r-iadcc.pdf>

Seroter, Richard & Fairweather, Ewan & Ramani, Rama 2010. Applied Architecture Patterns on the Microsoft Platform. Great Britain: Packt Publishing Ltd.

Skonnard, Aaron 2009. A Developer's Guide to Service Bus in Windows Azure platform AppFabric. Downloaded December, 2012. <http://go.microsoft.com/?linkid=9751403&clcid=0x409>

Standards for Security Categorization of Federal Information and Information Systems 2004. USA: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. Downloaded August, 2012. <http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf>

Stoneburner, Gary & Goguen, Alice & Feringa, Alexis 2002. Risk Management Guide for Information Technology Systems. USA: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. Downloaded November, 2012. <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>

Windowsazure.com 2012a. Downloaded October, 2012. <https://www.windowsazure.com/en-us/pricing/free-trial/>

Windowsazure.com 2012b. Downloaded October, 2012. Requires sign up. <https://account.windowsazure.com/signup?offer=ms-azr-0018p&wa=wsignin1.0.>

Windowsazure.com 2012c. Downloaded October, 2012. Requires sign up. <https://manage.windowsazure.com/>

Windowsazure.com 2012d. Downloaded November, 2012. Requires sign up. <https://manage.windowsazure.com/?whr=live.com#Workspaces/VirtualMachineExtension/vms>

Windowsazure.com 2012e. Downloaded November, 2012. Requires sign up. <https://manage.windowsazure.com/?whr=live.com#Workspace/StorageExtension/storage>

Windowsazure.com 2012f. Downloaded December, 2012. Requires sign up. <https://manage.windowsazure.com/#Workspace/CloudServicesExtension/list>

Windowsazure.com 2012g. Downloaded December, 2012. Requires sign up. <https://manage.windowsazure.com/?whr=azure.com#Workspaces/CloudServicesExtension/CloudService/cristimarin/monitor>

Zhiming, Xue 2012. Windows Azure Store Preview. Downloaded December, 2012. <http://blogs.msdn.com/b/zxue/archive/2012/10/29/windows-azure-store-preview.aspx>

APPENDIX 1 Appendix 1

1(5)

Threat	**Layer where mitigation is implemented**	**Nature of mitigation provided (if specific to Windows Azure)**	**Application/Service-layer mitigation required**	**Is this issue higher risk or more complex in cloud deployments?**
Spoofing				
ARP Flooding	Platform	VM Switch	None Required	No
IP address spoofing	Infrastructure & Platform	Top-of-rack switches restrict which IP and MAC addresses VMs use	None Required	No
DNS spoofing	Infrastructure	Microsoft Live DNS Services	None Required	No
Tampering				
Packet tampering/interception on VM Bus	Platform	Trusted Channel between Hypervisor and VM tenants, VM switch has additional packet filters imposed	None Required	Yes
Windows Azure OS binary tampering	Platform	Binaries are Microsoft-signed and managed assemblies are strong named	Verify the signature of Windows Azure SDK binaries referenced in the application code.	Yes
Local Filesystem/Registry Tampering by compromised web services	Platform	Web Roles run as non-admin; strong ACLs on file system/registry enforced by runtime	None Required	Yes

Table 1. Security threats and mitigation measures (Marshall et al. 2010, 21-22)

2(5)

(Continued)

Tampering/disclosure of credentials or other sensitive application data	Web Role		Use Windows Identity Foundation and HTTPS mutual authentication for SSL connections	No
Tampering with customer configuration data, encryption keys and intellectual property during web role provisioning	Infrastructure & Platform	VLANs, IP ACLs, Mutual SSL authentication in use between fabric controller and root/guest nodes.	None Required	No
Repudiation				
Audit log collection, storage and analysis	Platform and Web Role	Windows Azure Monitoring and Diagnostics APIs	Use monitoring and diagnostic APIs as needed; transfer logs to Storage private blob/table storage over HTTPS	Yes
Information Disclosure				
Footprinting or enumeration of services & applications in the VM	Platform	VLANs, HW & SW firewalls, VM switch filters, IP filtering in VM Guest	None Required	Yes
Side-channel attacks against VM Guests on the same physical host	Platform	1 VM per core, no communications between different tenants	None Required	Yes
Disclosure of data in transit between client and server	Platform and Web Role		Use HTTPS in place of HTTP where sensitive data is transferred	Yes

Table 2. Security threats and mitigation measures (Marshall et al. 2010, 22-23)

3(5)

(Continued)

Disclosure of SSL Certificates/keys used by Web Roles	Platform and Web Role	Secure provisioning via Windows Azure Certificate Store	Use the certificate store for client and server SSL certificate storage	Yes
Disclosure of arbitrary secrets in blob/table/queue storage	Web Role/Client		Pre-encrypt secret data prior to uploading. Do not store decryption keys in Windows Azure Storage	Yes
Disclosure of Shared Access Signatures	Web Role/Client		Use HTTPS to securely transfer Shared Access Signatures to intended recipients and set appropriate permissions on containers.	Yes
Physical theft of storage account information, code or other intellectual property	Infrastructure	Physical Security and Operations Policies	None Required	No
Encrypted Storage of Arbitrary Secrets in Windows Azure Storage	Platform		Will require API calls similar to DPAPI	Yes
Denial of Service				
Denial of Service attacks via network bandwidth saturation (packet flooding)	Platform	Load balancing & throttling in network infrastructure	None Required	No

Table 3. Security threats and mitigation measures (Marshall et al. 2010, 23)

Appendix 1

(Continued)

Identification of botnets and malicious network traffic	Infrastructure	Windows Azure Live Services monitors and investigates	None Required	Yes
Deep packet inspection for network attacks with known signatures	Platform		None Required	Yes
Flooding of Web Role local storage or blob/table storage	Platform	Quotas, ACLs, Reduced privilege execution and flood monitoring protection	None Required	Yes
Request flooding at the customer code/app level	Web Role		Implement application-level request throttling if necessary	No
Elevation of Privilege				
Anti-virus scanning of VM Guests/Hosts	Platform		None Required	Yes
Misconfiguration of Service/Application settings	Web Role		Must scope all cookies and the document.domain property to the service subdomain (eg. http://contoso.cloudapp.net) and NOT to *.cloudapp.net	Yes
Cross-site Request Forgery Attacks against the web role	Web Role		Use ASP.NET defenses	No
Cross-site Scripting Attacks against the web role	Web Role		Use the Anti-XSS Library	No

Table 4. Security threats and mitigation measures (Marshall et al. 2010, 24)

(Continued)

API fuzzing attacks on interfaces exposed by the web role	Web Role		Fuzz all interfaces and endpoints unique to code exposed to the web (or any other services)	No
Network packet fuzzing attacks against network protocols used by web role	Platform	IPFilter driver customizations fuzzed to SDL requirements, external penetration testing performed	None Required	No
File Fuzzing attacks against file parsers which are part of Windows Server 2008	Platform	Existing parsers fuzzed to SDL requirements, external penetration testing performed	None Required	No
File Fuzzing attacks against custom, application-provided file parsers	Web Role		Fuzz test all proprietary network protocol or file format parsers	No
Patching of security vulnerabilities at the Web Role/customer code level	Web Role		Have a security response and updating plan in place	No
API Fuzzing attacks against the Hypervisor root	Platform	APIs fuzzed to SDL requirements, external penetration testing performed	None Required	Yes

Table 5. Security threats and mitigation measures (Marshall et al. 2010, 24-25)