Michael Mstoi

Business Application Development In Azure

Cloud Service Introduction

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

25 April 2015

Author(s) Title Number of Pages Date	Michael Mstoi Business Application Development In Azure 52 pages 25 April 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Olli Hämäläinen, Senior Lecturer

The goal of this project was to study and develop an e-commerce web application with Microsoft's newest technology called ASP.NET. The development was done with Microsoft tools like Visual Studio 2013 and C# language. However, to develop the application, an entirely new approach was taken. Development environment and all development tools were deployed to Microsoft's Cloud Service called Azure. The main purpose of this project was to show that nowadays it is possible to develop and maintain business applications in Cloud Service. This approach opens new business opportunities for many small and medium-sized companies.

There are many companies in Finland and all over the world, which demand better organization and administration of their business intelligence data. Fluent communication, effective maintenance and sharing of various documents and other business objects of a company have become very critical aspects of a business nowadays.

During the implementation of the thesis work, the following technologies were learned and introduced: Microsoft Azure Cloud Service, Windows Server 2012, Microsoft SQL Server 2012, Microsoft IIS 8.0, Visual Studio 2013 and Microsoft Team Foundation Server. Then all those technologies were installed and integrated into the Microsoft Azure Cloud Service. After that an ASP.NET MVC sample application was developed and tested with Visual Studio 2013. Finally the developed application was deployed into a production environment to be accessed and used worldwide.

Keywords	Azure, ASP.NET, MVC, CSS, T-SQL, C#, IIS, Cloud Service

Contents

1 Introduction

There are many companies in Finland and all over the world, which demand better organization and administration of their business intelligence data. Fluent communication, effective maintenance and sharing of various documents and other business objects of a company have become very critical aspects of a business nowadays.

Microsoft SharePoint has made possible an administration and sharing of up-to-date documents and other objects of a company. In addition, a new phenomenon called Cloud Services provides new and better ways to conduct daily business tasks of an organization.

The aim of this thesis is to learn and combine different new technologies into a suitable package for an organization's business intelligence management. During the implementation of the thesis work, the following technologies will be learned, integrated and introduced to the reader:

- Microsoft Azure Cloud Service
- Windows Server 2012
- Microsoft SQL SERVER 2012
- Microsoft IIS 8.0
- Visual Studio 2013
- Microsoft Team Foundation Server
- ASP.NET MVC 4 application example

All the above technologies are studied and integrated together and then they will be installed into Microsoft's Cloud Service. Three environments will be created: Production Environment, Development Environment and Test Environment. The Production Environment will be installed in the Azure Cloud Service and will include MS IIS 8.0 (which holds ASP.NET MVC 4 application) and MS SQL Server. The Development Environment will also be installed in Azure Cloud Service and will include Visual Studio 2013, MS SQL server, MS Team Foundation Server. The third environment called the Test Environment will also be created in Azure Cloud Service and will be used for the application's alpha testing and for the other aspects of the project. After successful testing the project will be swapped into the Production Environment.

2 Introducing ASP.NET MVC 4 project and ASP.NET platform

2.1 Introducing the sample project

In this section an example project is introduced. This project is a web application and supports conducting e-commerce business in the final version. By developing and deploying the application, many tools and domains introduced in the previous chapter will be studied and integrated into a solidly functioning ensemble or package. The application is called SecondHandStore and it is developed by using Microsoft's ASP.NET MVC 4 application framework.

The application is developed gradually with tools installed in the virtual machine which resides in Microsoft Azure Cloud Service. This environment is called the Development Environment. Altogether there are four versions of the developed application:

1. Basic implementation
2. Adding navigation and chart to basic implementation
3. Creating shopping cart and adding items
4. Administration of the application

The application is developed with the Microsoft Visual Studio 2013 Community edition. All versions are added to the Microsoft Team Foundation Server (TFS) version control system. After the development is done and the application is tested, the released version will be deployed into another environment called the Production Environment. This environment also resides in Microsoft Azure cloud service. After deployment the application can be viewed and used from any computer in the world by the internet.

The SecondHandStore application contains an online product catalog of different outfits. Customers can browse products by page and category. They can add products or remove products into a shopping cart, then check out by entering their shipping details. In addition, the administration area of the application enables the administrator of the application to create, add, remove and read the catalog items.

2.2 Introducing ASP.NET MVC Application Platform

ASP.NET MVC was created by Microsoft Corporation. It is a framework for developing web applications. It combines the effectiveness of Model-View-Controller (MVC) architecture and best parts of ASP.NET platform. It delivers significant advantages for developing complex and non-trivial web development projects and is an alternative to ASP.NET Web Forms. ASP.NET MVC is lightweight framework and is highly testable with existing ASP.NET features. This framework is a part of .NET platform and can be found in System.Web.Mvc assembly.

ASP.NET MVC is based on ASP.NET and it allows the developers to build web applications based on three logic levels: Model, View, and Controller. The Model is the business level of the application. The View presents the display or user interface of the web application and the Controller is a layer which controls the application input and handles the interaction. In addition, the Controller layer also updates the Model in order to reflect a change in state of application and passes new information to the View. After accepting new information from the Controller, the View renders user interface to show updated information.

The phenomenon of Model – View – Controller is not new. It was used in Smalltalk applications in the late 1970's. It was conceived in Xerox Corporation to enforce a separation of concerns, which in other words meant decoupling the domain model and controller logic from the user interface. The most important part of the MVC application is the domain model. The domain model is created by identifying the real–world entities, rules and operations that exist in industry (domain) that the application must support. After identification the application developers create the domain model which is a software representation of the domain. In the case of this final year project the domain model will be created with C# classes, structs (types in C# jargon) etc.

Each request that comes to the application is handled by the Controller. In ASP.NET MVC framework controllers are .NET classes. There is no business or data storage logic in the controller. The role of the controller is to encapsulate the application logic. It processes incoming requests, performs operations on the domain model and selects views to render to the user.

The responsibility of the view is to render a user interface for the application user. The View of the MVC pattern is implemented through the view engines of ASP.NET MVC framework. The framework includes two built-in view engines that are full–featured and well tested. The first one is the legacy ASPX engine which is also known as the Web Forms view engine. This engine is useful for maintaining compatibility with older MVC applications. The second one is the Razor engine, which was introduced with the third version of the MVC. It has a simple and elegant syntax.

Because of the fact that ASP.NET MVC is based on .NET platform, the developer has flexibility to write code in any .NET language and access some API features of extensive .NET class library and huge ecosystem of third-party .NET libraries. Master pages, form authentication, membership, profiles, roles, and internationalization are ready-made ASP.NET platform futures. Those features can reduce the amount of code needed to develop and maintain a web application [1].

Next, Microsoft Azure Cloud Service will be introduced. The actual development of MVC application will be shown in the chapter 6 of this final year project.

3 Microsoft Azure Cloud Service and Subscription

It is very common that today’s business enterprises heavily rely on a datacenter. The enterprise’s datacenter has a very pivotal role in conducting daily business tasks successfully. The datacenter of the enterprise is created and maintained by the enterprise’s own Information Systems manager or some other accordant key role.

In order to create an on-premises datacenter, the system manager has to purchase and install the hardware, obtain and install an operating system and the required applications, configure the network and firewall and set up the storage for data. After the installation and configuration tasks have been done the responsibility for maintaining the datacenter through the lifecycle begins. The purchase and installation of this kind of datacenter comes with a huge impact on capital costs of the enterprise, which explains why new technology called Cloud Computing was emerged to provide alternative solutions to an on-premises datacenter.

Microsoft Azure is a public cloud. Cloud Computing provides a modern alternative for on-premises datacenters. The responsibility for maintaining hardware and software lies completely on the public cloud vendor, which provides a wide variety of platform services for different kind of enterprises. The enterprise leases the needed hardware and software services on as-needed bases. Cloud Computing also makes possible for the enterprise to lease such hardware or software which otherwise is too expensive to purchase. The payment for cloud services usually are billed to enterprises based on used time.

The subscribed user usually manages the provided cloud services through the portal. For example, in order to create a new virtual machine, the user configures the computing node size, the CPU, the size of storage, the operating system, network configuration and other parameters through the provided cloud portal. After that, the created and configured virtual machine is ready for usage within a very short time, contrary to creating a virtual machine on on-premises hardware, which could take more than a week to install, configure and use.

Besides the public cloud, there are two other kinds of clouds available – private cloud and hybrid cloud. In the case of a private cloud, the enterprise purchases hardware and creates its own cloud environment for its employees. The hybrid cloud on the other hand, integrates public and private clouds. For example, the enterprise may deploy and host a website on the public cloud but link it to a high-secure database which resides in the private cloud.

Microsoft Azure is deployed onto 19 datacenters across the globe. It makes possible for startups to start with very low cost and scale to rapidly gain customers. Azure is also very flexible to set up development and testing configurations, test new software versions without installing them on the hardware. For example the user may quickly install Windows Server 2012 R2 on cloud and test its existing application on this new platform without any complicated consequences [2].

Microsoft Azure provides the following services: Compute Services, Data Services, Application Services and Network Services. More information about Azure services is found in Table 1.

Table 1. Services provided by Microsoft Azure

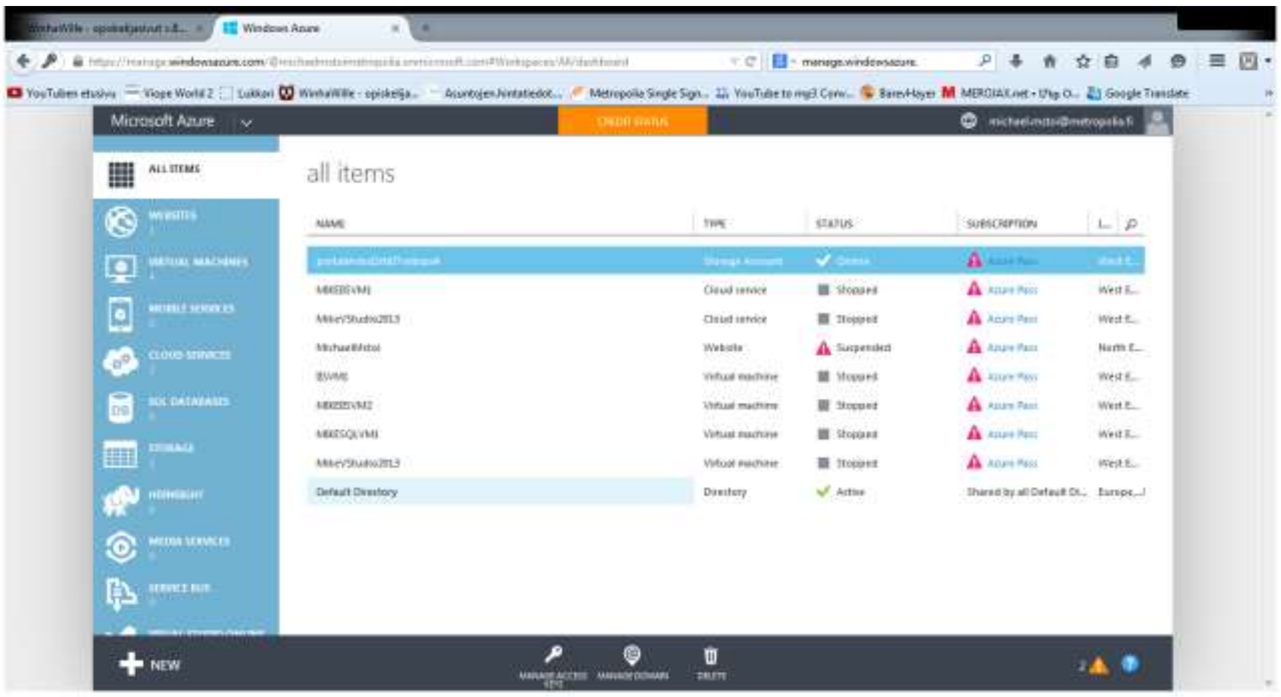
Compute Services	Includes Azure Cloud Services, Azure Virtual Machines, Azure Websites and Azure Mobile Services
Data Services	Includes Microsoft Azure Storage, Azure SQL Database and Redis Cache
Application Services	Includes services, which helps the user to build and operate his/her own applica-tions, such as Azure Active Directory, Service Bus, Azure Scheduler, Azure Me-dia Services and so on.
Network Services	Includes Azure features, such as Virtual Networks, Azure Traffic Manager.

For this final year project Compute and Data services were used. Next, the portal for accessing Azure cloud service will be introduced.

3.1 Accessing Microsoft Azure through the Portal

In order to manage and use services provided by Microsoft Azure, an online portal is used. The online portal is used to create and manage virtual machines, web pages, storage places and other services. Currently only two versions of portals are available for Microsoft Azure. The first version is called Azure Management Portal. The second portal is called Azure Preview Portal and is accessed through https://portal.azure.com page. Both portals have a completely different look and feel. In this document we will focus on Azure Management Portal.

Azure Management Portal is accessed through https://manage.windowsazure.com. The user may see all of the resources being used in the subscription. See Figure 1 for an example.

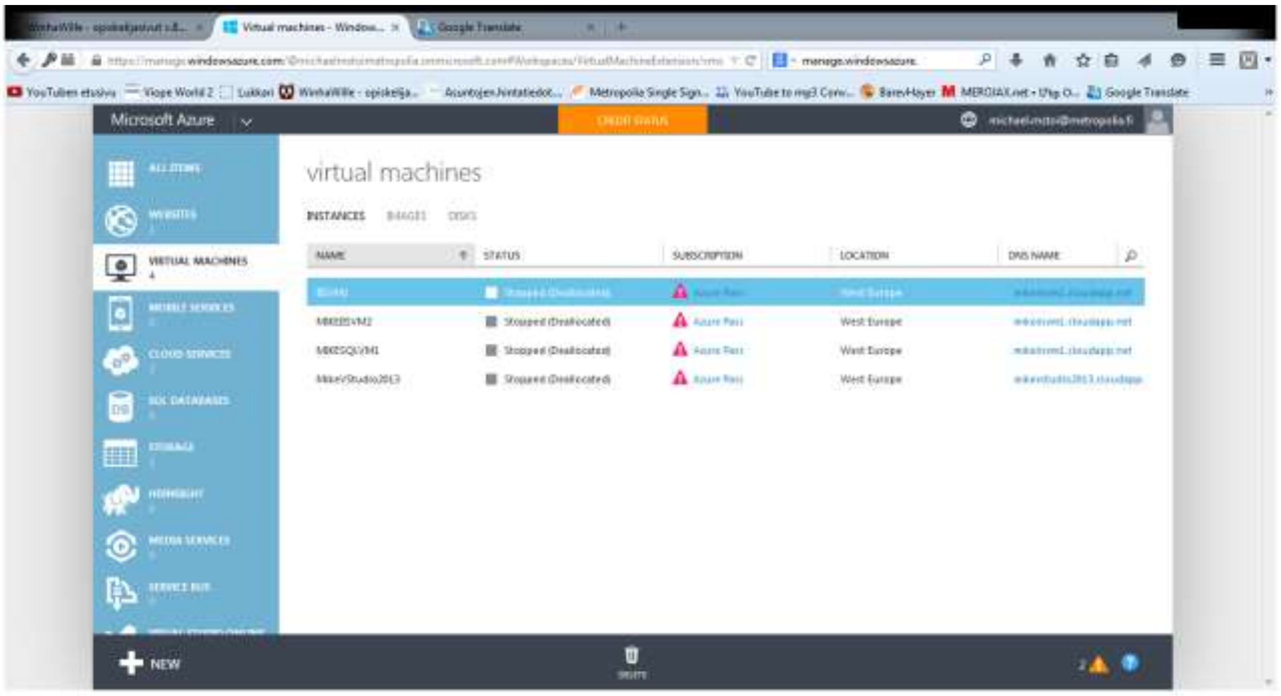


all items

NAME	TYPE	STATUS	SUBSCRIPTION	L...
[illegible]	Storage Account	✓ Online	Azure Pass	[illegible]
MIKEBSVM1	Cloud service	Stopped	Azure Pass	West E...
MikeVStudio2013	Cloud service	Stopped	Azure Pass	West E...
MichaelMitoi	Website	Suspended	Azure Pass	North E...
BSVM1	Virtual machine	Stopped	Azure Pass	West E...
MIKEBSVM2	Virtual machine	Stopped	Azure Pass	West E...
MIKESQLVM1	Virtual machine	Stopped	Azure Pass	West E...
MikeVStudio2013	Virtual machine	Stopped	Azure Pass	West E...
Default Directory	Directory	✓ Active	Shared by all Default Di...	Europe,...

Figure 1. All Items created by the user in Azure Management Portal.

As may be seen in Figure 1, all available items of the current subscription are shown on the left side of the screen. Besides looking at all items in one list, the user has possibility to select specific items by their resource type, for example virtual machines and mobile services. In Figure 2, the user has selected Virtual Machines and the list of all created virtual machines and their status is shown on the screen.



virtual machines

INSTANCES IMAGES DISKS

NAME	STATUS	SUBSCRIPTION	LOCATION	DNS NAME
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MIKEBSVM2	Stopped (Deallocated)	Azure Pass	West Europe	[illegible]
MIKESQLVM1	Stopped (Deallocated)	Azure Pass	West Europe	[illegible]
MikeVStudio2013	Stopped (Deallocated)	Azure Pass	West Europe	[illegible]

Figure 2. Created virtual machines and their status.

After selecting and clicking a specific virtual machine from the list of virtual machines, the Dashboard for the selected virtual machine will be opened. The menu options and Dashboard are not the same for all resource types. They change depending on the type of the resource. For example, menu options for a website contain different sets of tools which are not relevant to virtual machines. Figure 3 shows an example of available commands for virtual machine resources.

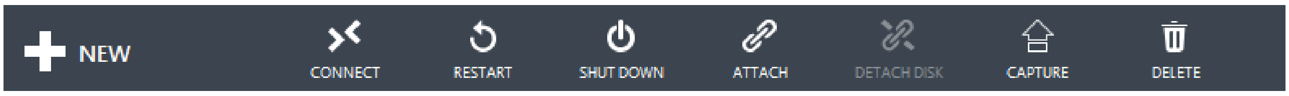


Figure 3. Available commands for virtual machine resource type.

As one can see from Figure 3, the dashboard of a virtual machine shows all possible operations available for selected virtual machine.

3.2 Subscribing Azure and Billing Options

In order to use services provided by Microsoft Azure a subscription is needed. The user must have a Microsoft account or a work or school account to access the subscriptions. The most common subscriptions are found in Table 2.

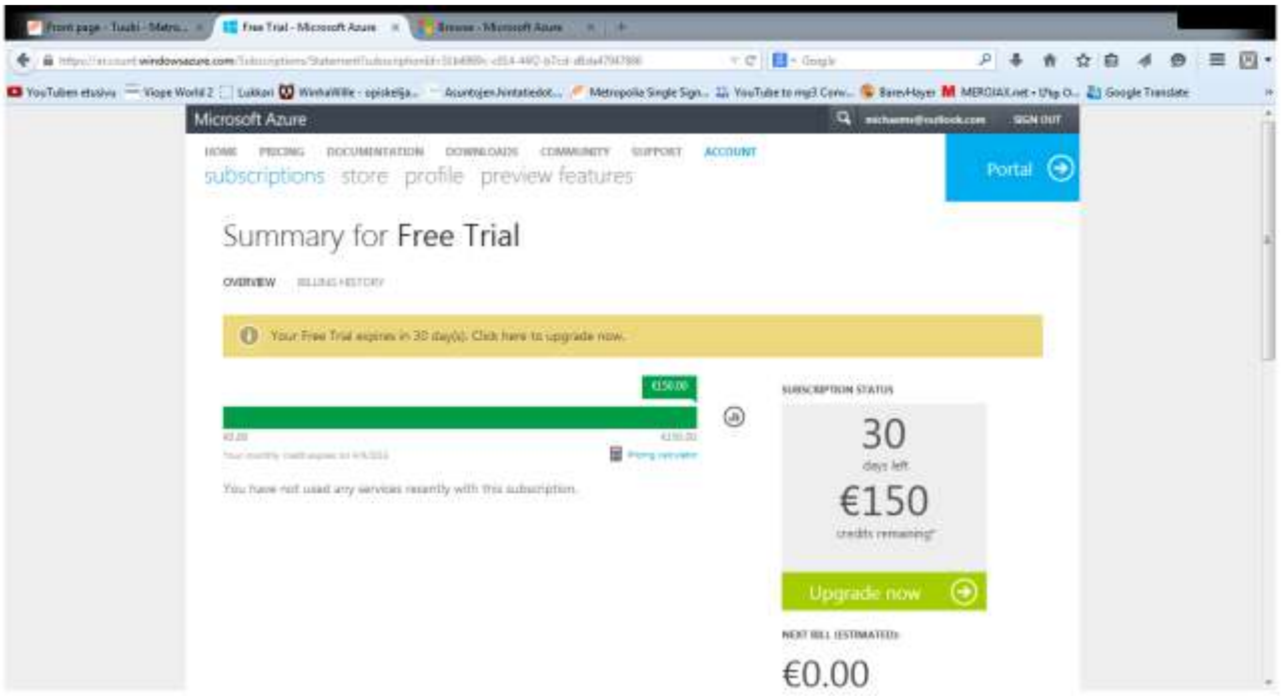
Table 2. Showing the most common subscriptions of Microsoft Azure.

Free Trial	Gives the user $150 credit and a month to try any combination of available resources in Azure. At the end of trial the services will be suspended and will no longer work. It is possible to upgrade the subscription to pay-as-you-go subscription to any time.
MSDN Subscription	If the user has MSDN subscription, he/she get specific amount of credit for every month.
BizSpark account	BizSpark program provides benefits for startups and the user gets $150 credit every month for each of those five MSDN accounts included in BizSpark program.
Pay-as-you-go	The user attaches a credit or debit card to this subscription and pays for what he/she uses.
Enterprise agreement	This subscription demands that the enterprise is committed to use a certain amount of services in Azure and payment is done ahead of time.

Table 2 shows that a user can create Free Trial subscription and later may easily update it to Pay-as-you-go or to another subscription.

3.3 Creating Azure Account for the Project

For this project, Azure Free Trial account is used. In order to create and access Microsoft Azure, Microsoft account is needed. I had no previous Microsoft account, so I created one. My Microsoft account's username is michaems@outlook.com. After inserting all required information and credit card number, the account was created and ready for the use. One may see the account creation summary in Figure 3.



Summary for Free Trial

Your Free Trial expires in 30 day(s). Click here to upgrade now.

€150.00

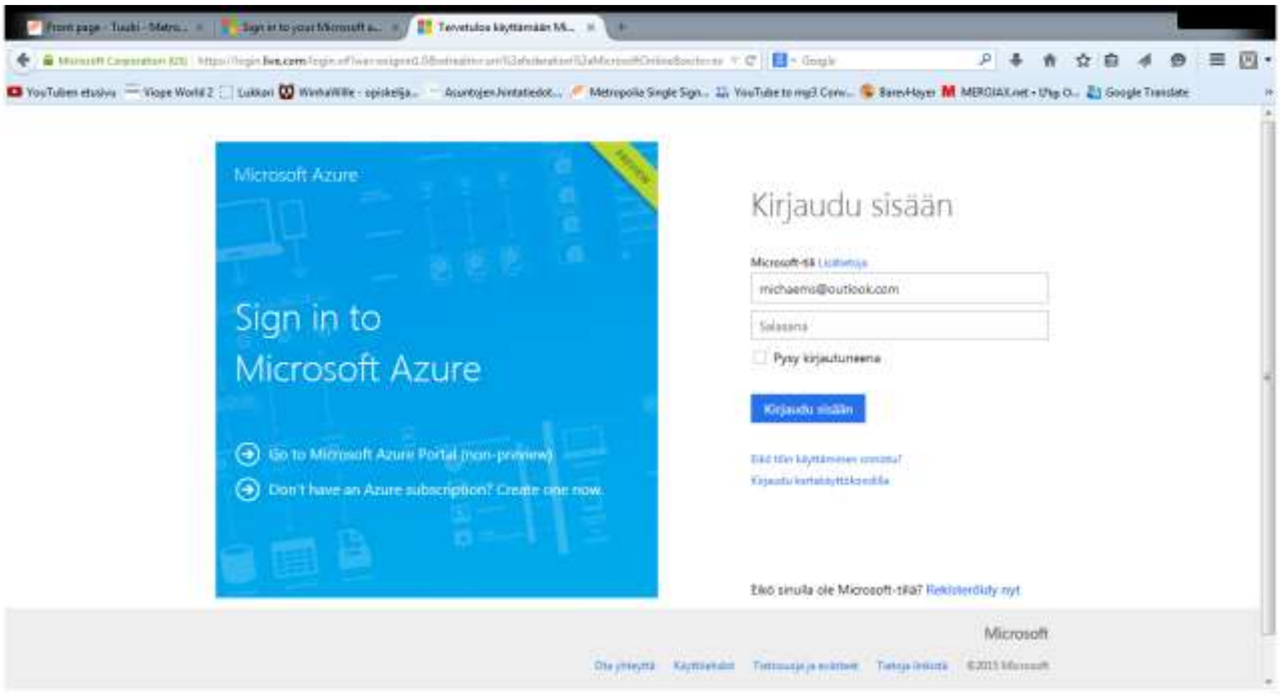
You have not used any services recently with this subscription.

30
days left
€150
credits remaining

€0.00

Figure 3. Microsoft Azure Free Trial account creation summary.

As may be seen in Figure 3, a free trial gives the user to use the Azure Cloud services for 30 days and with $150 credit amount. After the trial expiration, the user may extend the usage explicitly by allowing Microsoft to charge from the user's credit card.
As soon as the Azure Account is created, the user may sign in from http://login.live.com address. Figure 4 shows the interface for signing in.


Microsoft Azure
Sign in to
Microsoft Azure
Go to Microsoft Azure Portal (non-preview)
Don't have an Azure subscription? Create one now.
Kirjaudu sisään
Kirjaudu sisään
Microsoft

michaems@outlook.com
Salasana
Pysy kirjautuneena

Figure 4. Interface for signing in Microsoft Azure services.

By providing a username and password, the user signs in and may create and configure the required objects and services. See Figure 5.

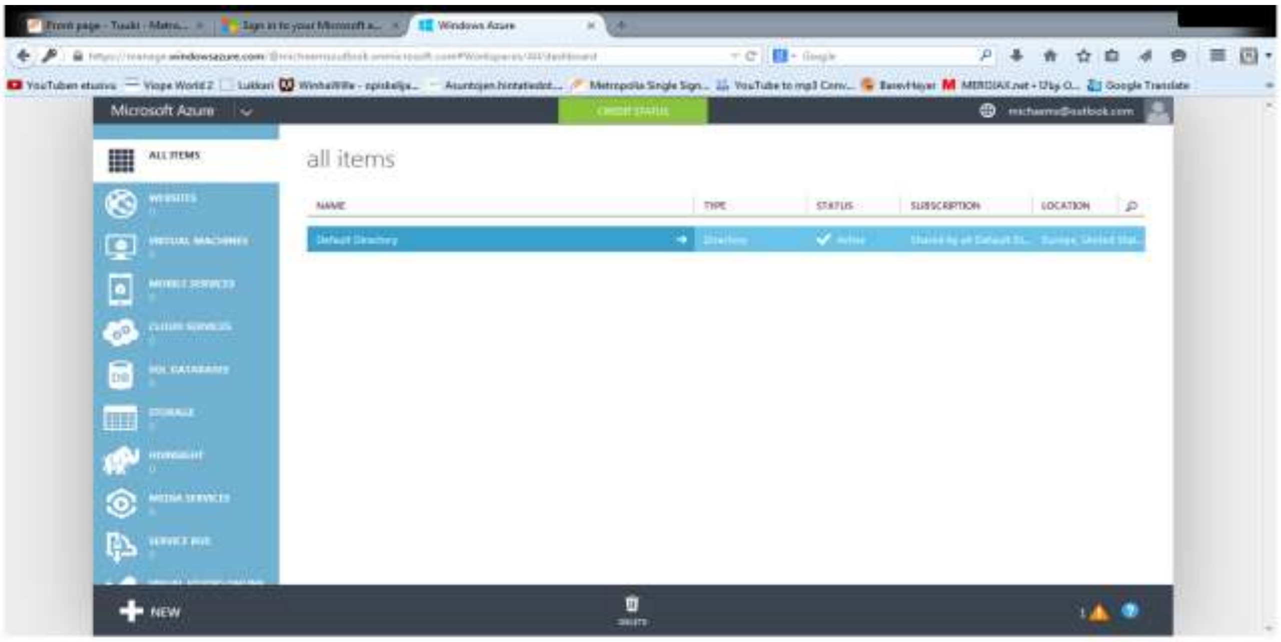


Figure 5. Microsoft Azure is signed in. No services created yet

Next, the Microsoft Server environment will be introduced. In that section, I will present key properties of Microsoft Windows Server 2012. A new virtual machine will be created for the final production environment. The operating system Microsoft Server 2012 will be installed and set up to the newly created virtual machine. Later this server will be used to host the developed ASP.NET application's final version. This will enable the users of that application to access it from anywhere in the world and use its provided operations.

4 Microsoft Server Environment

4.1 Introducing Windows Server 2012

Before going any further, let us define what a server is. A server is a computer program, running on operating system, which is capable to accept request from other applications and give a corresponding response. The other application may reside in the same computer or it may be running on another hardware (or computer), which resides anywhere in the world. The application sending request to the server application is called client application.

Both client and server applications operate within a client-server architecture. In order to create a connection between client and server applications, the TCP/IP protocol is used. TCP stands for Transmission Control Protocol and IP stands for Internet Protocol. In addition, in order to identify the request from the client and respond to it correctly, the server application runs on the background and continuously acts as a socket listener [3].

There are many server application types. In order to distribute and share different kind of resources to the users, the following server applications have been created and being used during the computing era:

- Web Server
- Database Server
- Mail Server
- Print Server
- Game Server
- Application Server

Hardware requirements for hosting server applications are different than for example hardware requirements for general desktop computers or computers for other purpose. Usually to implement a server environment, the speed of CPU is not a critical factor. Rather, many operating systems and applications used in server environment do not provide graphical user interface, which reduces the use of CPU resources.

Because of the fact that server computers are connected through the network, it is not mandatory that the computer, hosting the server environment, to have a monitor or to provide audio or USB interfaces. Rather, it is very critical that the server's hardware is able to provide fast network connections and high input / output throughout. Also, because the servers run for long periods without interruption, high availability of the server is a very important and critical factor.

Because servers are running for long periods without shutdown, they have more fans to prevent the hardware from overheating. In addition, the noise of server hardware and the fact that servers need stable power supply, good internet access and good security, often they are stored in dedicated server centers. The server casings are mostly flat and wide and are capable to store many devices next to each other in the server rack. Operating systems used in server environment usually are administered by system administrators and they are fine-tuned for stability and performance rather than for user-friendliness and ease of use.

In the market there are many vendors who provide Server Operating Systems. Some of those operating systems may be acquired free of charge based on the public GNU agreement, while others may cause a significant impact on the enterprise's investing costs. One may see five major server operating systems and their descriptions in Table 3.

Table 3. Five major server operating systems of the prevailing market.

Mac OS X server	This is Apple's UNIX server operating system. At its core it shares many components with FreeBSD. Mac OS X Server has the same architecture as Mac OS X operating system and includes additional services, applications, and administration tools for managing and deploying servers.
Microsoft Windows Server 2012 R2	Released By Microsoft company in October 2013. This is the sixth release of Windows Server family of operating systems. This product includes graphical user interface, but the user also has an option only

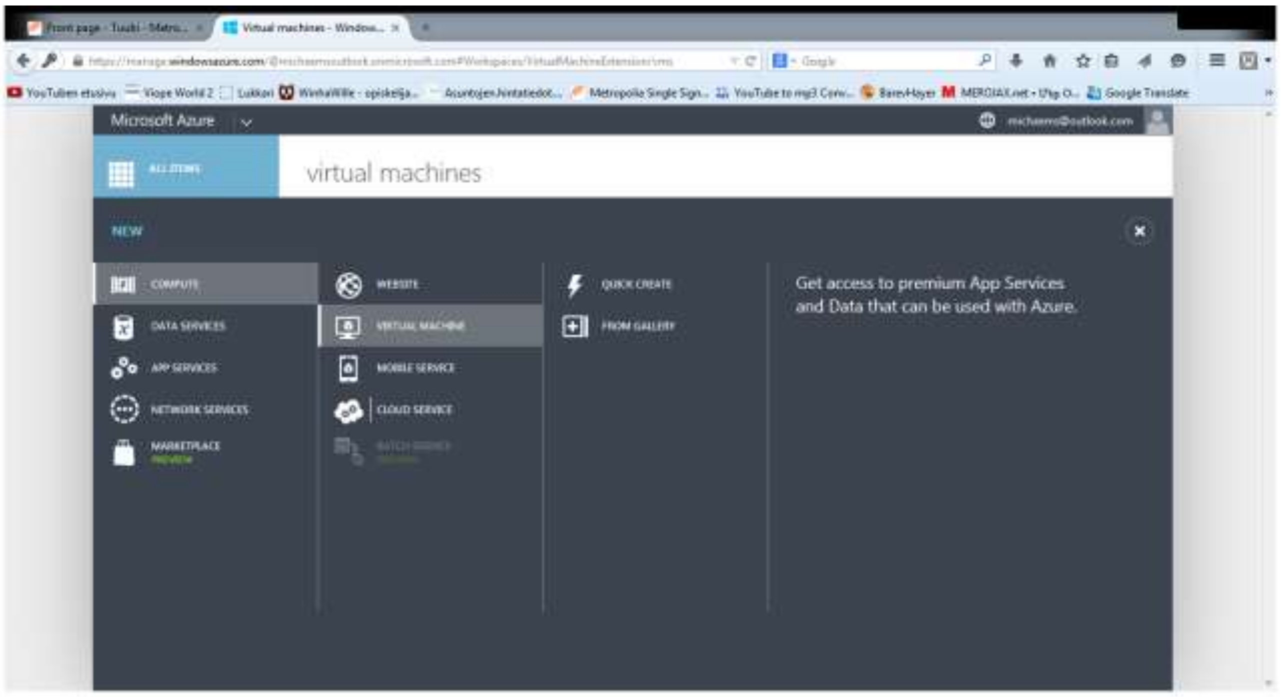
	to install “Server Core”.
Linux Operating System	The Server edition of Ubuntu Linux is free. This is very good choice for small companies, that want to minimize up-front costs but get professional support and service if needed.
FreeBSD	This operating system is derived from BSD and is UNIX version. It is very useful for high performance network applications and is easy to use.
Solaris	This operating system is created by Sun Microsystems. It is mainly used by company’s own 64 bit workstations and servers, which are based on AMD Opteron and Intel Xeon processors. Solaris also runs on systems manufactured by IBM, Intel, Dell etc.

Windows Server 2012 R2 was released in October 2013 by Microsoft Corporation. This was the sixth Server Operating System of Microsoft’s server family. There are four Windows Server 2012 R2 editions: Standard edition, Datacenter edition, Foundation edition and Essentials edition. The Standard edition is a very popular choice; it is feature rich and handles all general network needs. It provides only two virtualized instances. The Standard edition can be stripped down to its core for more security and better performance. The Datacenter edition is best used in highly virtualized environments and provides unlimited virtual instances. This is a very important and rich feature which is why this edition costs four times more than the Standard edition.

The Foundation edition includes most core features of found in above mentioned editions, but it has some significant limitations, for example, the maximum number of users is 15, the maximum number of Routing and Remote Access connections is 30, the maximum number of Internet Authentication Service is 10, only one CPU socket is allowed, it cannot host virtual machines, and some other limitations. The Essentials edition is used by very small enterprises with fewer than 25 users and fewer than 51 devices. This edition is very cost-effective to provide small-business networking. It includes the following features: Improved client deployment, allows to be installed as virtual machine, improved file history, System Restore, BranchCache and some other features.

4.2 Installing Windows Server 2012 R2 in Cloud Service

In order to install Windows Server 2012 R2 for the Production environment in the Azure Cloud, the first virtual machine shall be created. To create a new virtual machine, the user logs in the Azure Cloud Service and selects “Virtual Machines” option on the left pane of the user interface. Then the user presses the “New” button on the dashboard. A new window is opened, and the user selects the “Virtual Machine - > From Gallery” option. See Figure 6.

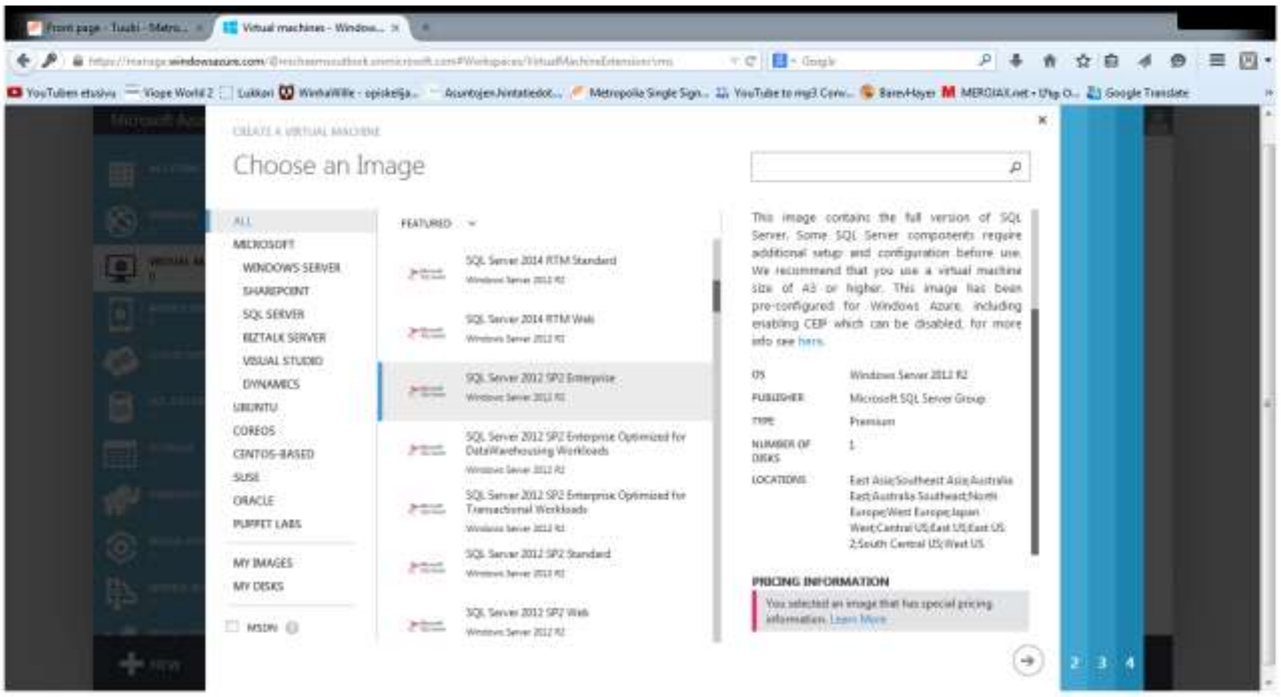


Get access to premium App Services and Data that can be used with Azure.

Figure 6. Creating new Virtual Machine in Azure Cloud Service.

After pressing the “From Gallery” option, another window is opened, and the user gets options to choose the desired operating system and/or configuration. In our case, we can select the first choice, which is Windows Server 2012 R2 Datacenter. However, Microsoft SQL Server 2012 is used for our Production Environment. In “Choose an Image” window, there is an option called SQL Server 2012 SP2 Enterprise.

Bellow the text of this option, the user may also see “Windows Server 2012 R2”. This means that by selecting this option, Azure Cloud Service will create virtual machine, install Windows Server 2012 R2 Datacenter edition on it and then in addition Microsoft SQL Server 2012 SP2 Enterprise edition will be installed on newly created Windows Server 2012 R2. So for our case we select this option and press “Next”. See Figure 7.



This image contains the full version of SQL Server. Some SQL Server components require additional setup and configuration before use. We recommend that you use a virtual machine size of A3 or higher. This image has been pre-configured for Windows Azure, including enabling CEIP which can be disabled, for more info see here.

OS	Windows Server 2012 R2
PUBLISHER	Microsoft SQL Server Group
TYPE	Premium
NUMBER OF DISKS	1
LOCATIONS	East Asia;Southeast Asia;Australia East;Australia Southeast;North Europe;West Europe;Japan West;Central US;East US;East US 2;South Central US;West US

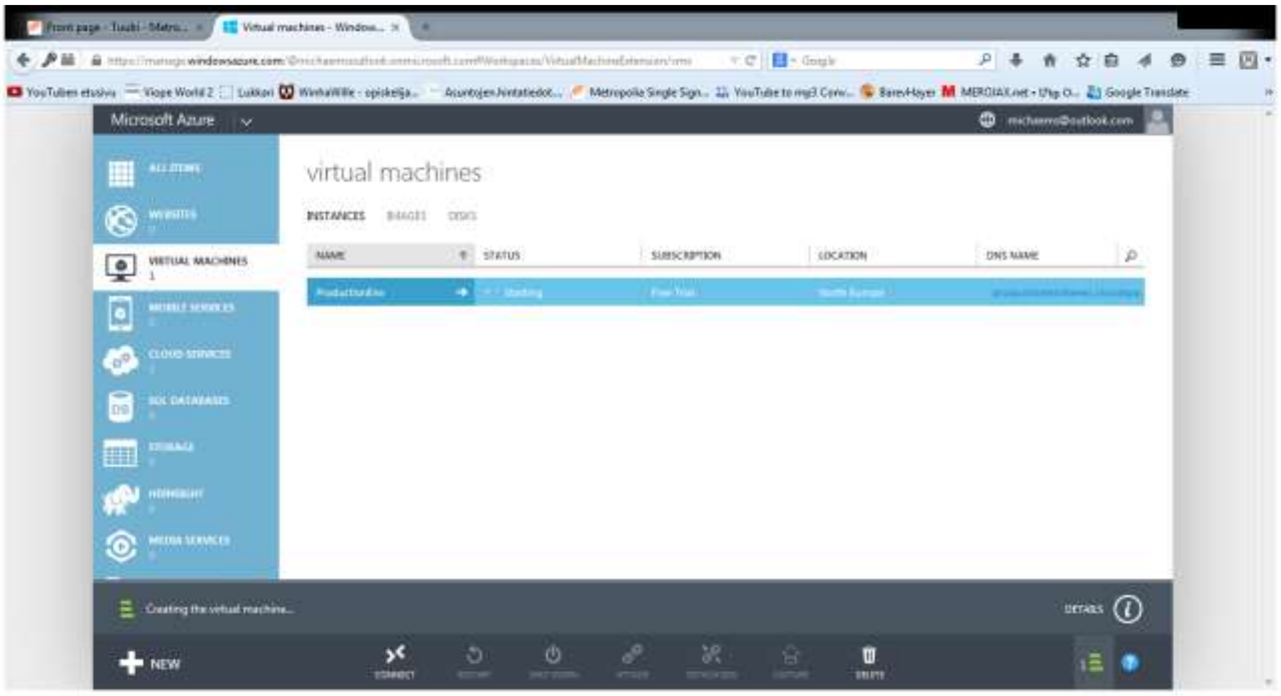
PRICING INFORMATION

You selected an image that has special pricing information. Learn More

Figure 7. Selecting the appropriate operating system image from the list of available images.

After pressing “Next”, the user enters the Virtual Machine’s name, new username and password for new user. In our case as the name for the virtual machine was selected “ProductionEnv” and new user name was entered as “michaems”.

The next to deal with is configuration of virtual machine. Cloud service DNS name was entered to be “ProdMichaems.cloudup.net”. All other options were left to default, and after pressing “Final” button, the creation of virtual machine with appropriate operating system and MS SQL Server 2012 R2 started. See Figure 8.



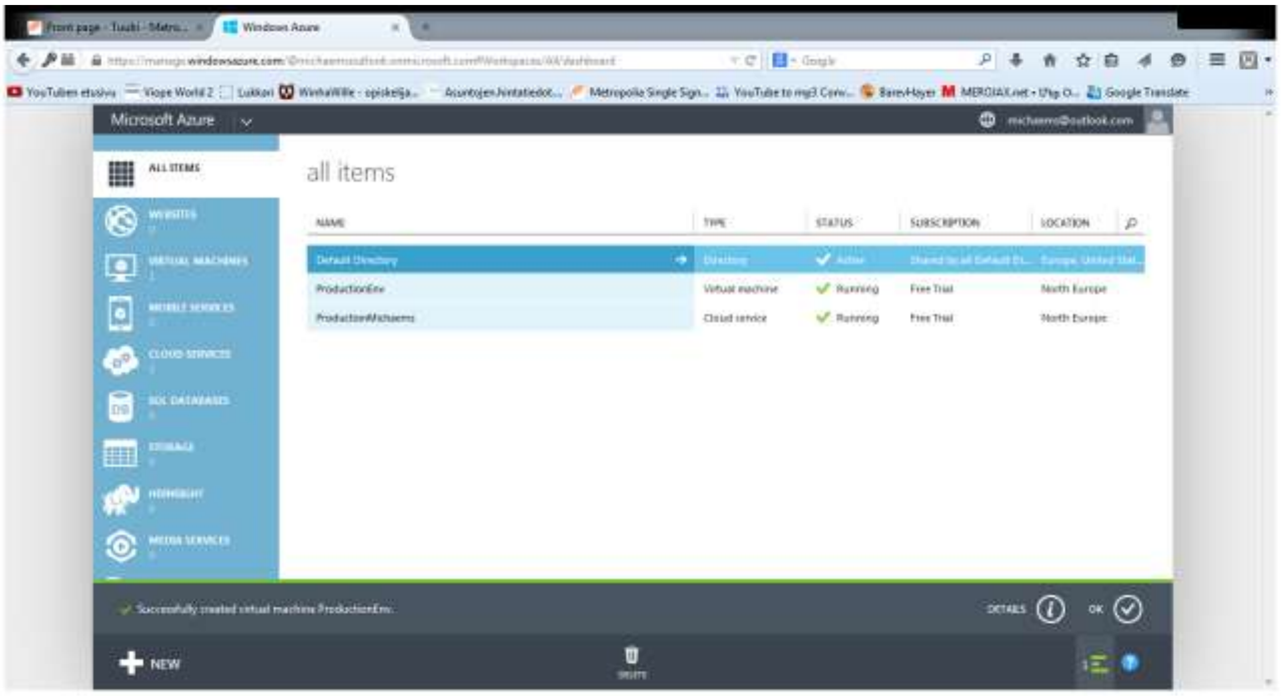
virtual machines

NAME	STATUS	SUBSCRIPTION	LOCATION	DNS NAME
ProductionEnv	Starting	Free Trial	North Europe	[illegible]

Creating the virtual machine...

Figure 8. Creating “ProductionEnv” virtual machine with appropriate operating system.

The created virtual machine and cloud service holding Windows Server 2012 R2 is seen in Figure 9.



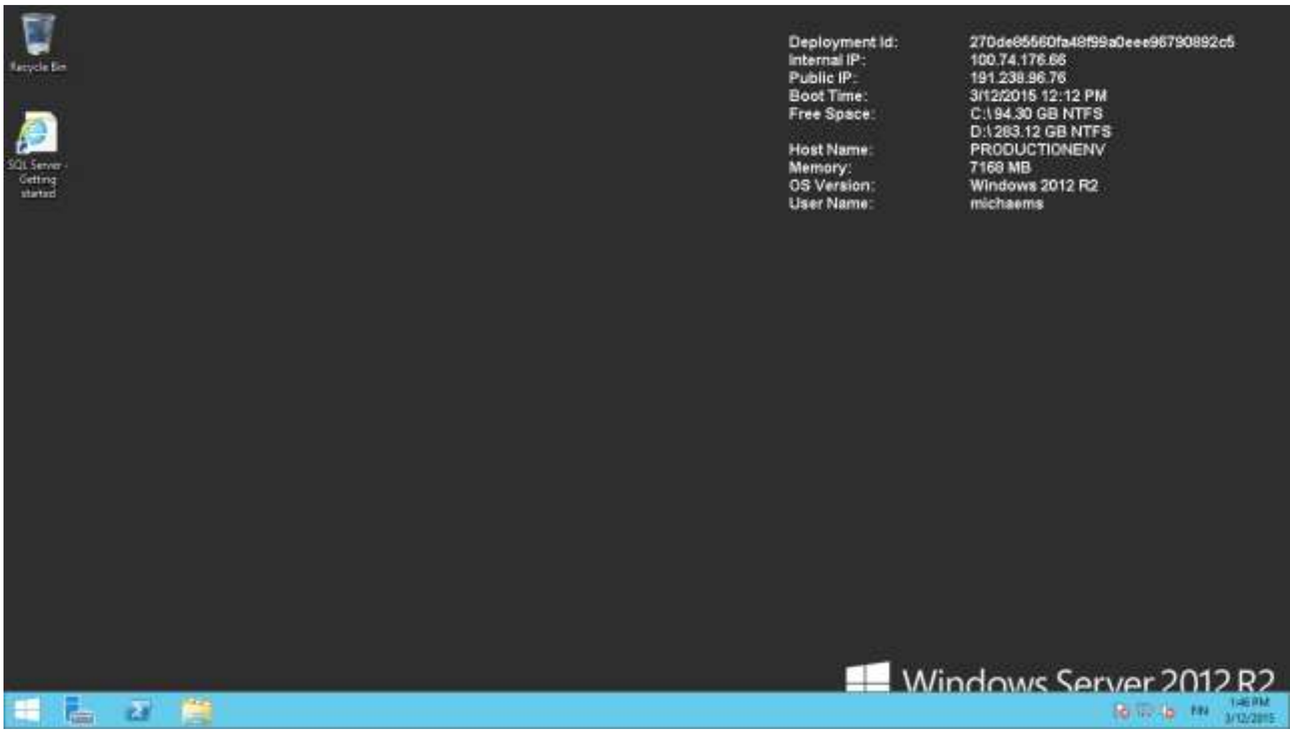
all items

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
Default Directory	Directory	✓ Active	Shared by all Default Di...	Europe, United Stat...
ProductionEnv	Virtual machine	✓ Running	Free Trial	North Europe
ProductionMichaems	Cloud service	✓ Running	Free Trial	North Europe

Successfully created virtual machine ProductionEnv.

Figure 9. The created virtual machine and cloud service holding Windows Server 2012 R2.

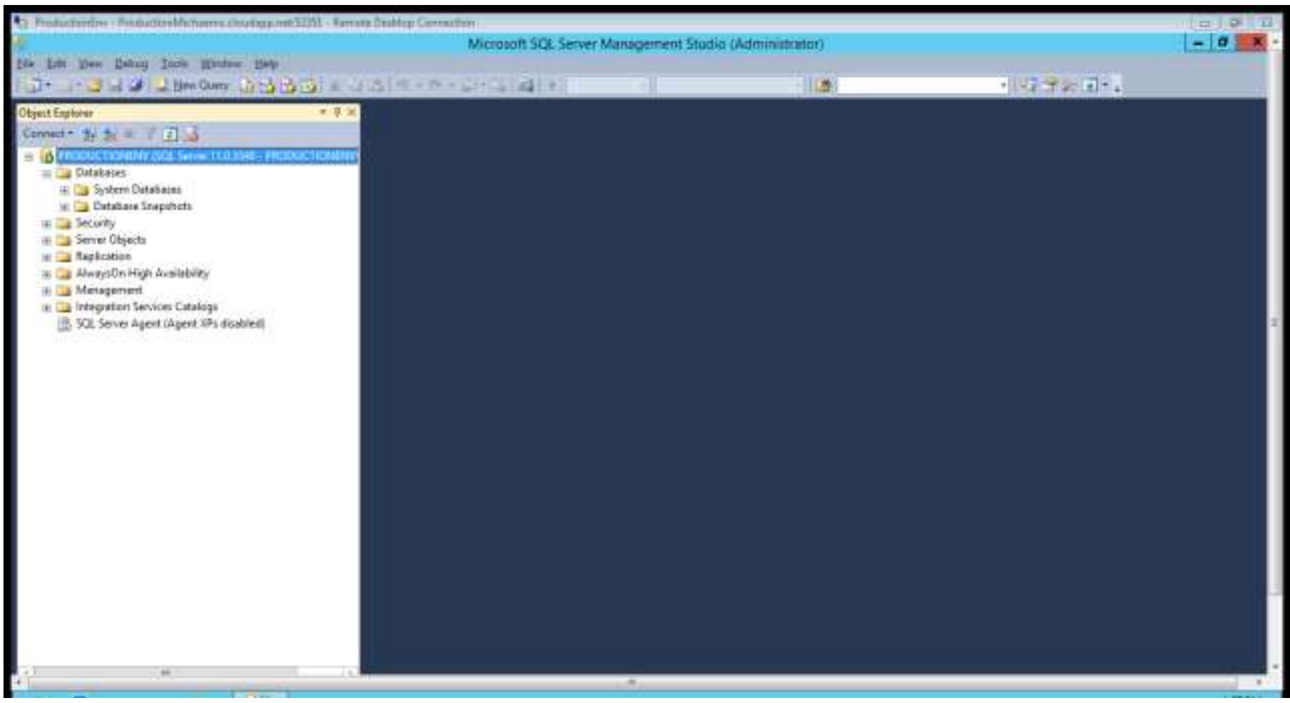
After the creation was done, the new virtual machine was connected by a remote desktop connection. See Figure 10.



```
Deployment Id:      270de85560fa48f99a0eee96790892c5
Internal IP:        100.74.176.66
Public IP:          191.238.96.76
Boot Time:          3/12/2015 12:12 PM
Free Space:         C:\ 94.30 GB NTFS
                    D:\ 283.12 GB NTFS
Host Name:          PRODUCTIONENV
Memory:             7168 MB
OS Version:         Windows 2012 R2
User Name:          michaems
```

Figure 10. “ProductionEnv” Windows Server 2012 R2 connected through remote desktop connection.

During the installation process of Windows Server 2012 R2, also MS SQL Server 2012 R2 was installed. One may see the interface of it in Figure 11.



Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
PRODUCTIONENV (SQL Server 11.0.3348 - PRODUCTIONENV
Databases
System Databases
Database Snapshots
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)

Figure 11. The user interface of newly created MS SQL Server 2012 R2.

Next IIS 8 will be introduced and configured for the user.

4.3 Introducing IIS 8

The Microsoft Internet Information Server (IIS) application has been existed for a long time, more than 15 years. Since the first release, it has been evolved all the time, from being a simple HTTP basic service provider to a fully functional and configurable application services platform. Currently it is fully integrated into Microsoft's client and server operating systems.

The first version of IIS was released with Windows NT 3.51. That time it was only a set of services providing functionalities such as HTTP, WAIS and Gopher. This release did not gain much popularity; most users preferred the third-party vendor's products, for example, Netscape's server. The second version of IIS came with the version Windows NT 4. Significant improvement of this version was close integration with Windows NT operating system. It took advantage of Windows security accounts and provided integrated administration through a management console. Microsoft Index Server began its existence with this second release of IIS. This service made it possible to introduce a web browser interface for management and context indexing.

IIS 3 came into existence with Windows NT Service pack 3. Microsoft's concept called Active Server Pages (ASP) was introduced with this version of IIS. Nowadays ASP is referenced as Classic ASP to be distinguished from currently wildly known and used ASP.NET. Classic ASP was a server-side scripting environment, which gave users a possibility to create dynamic web pages. Using VBScript or JScript, programmers got a very strong competitor to the Common Gateway Interface (CGI) and to other third-party scripting technologies.

Classic ASP was refined further into version 2.0 with the appearance of IIS 4. This version of ASP provided an object-oriented functionality and had six built-in objects to provide standardized functionality in ASP pages. After the forth version of IIS, it was not any more possible to download and install IIS outside the Windows operating system.

With the release of Windows 2000 IIS 5.0 was introduced, and with release of Windows XP came IIS 5.1. Those two versions of IIS were almost identical with only slight differences. The most significant difference of the IIS 5.0 with previous versions was full integration with the operating system. It became a service of the Windows operating system, meaning it became the base for other applications, especially for ASP applications. IIS now had a functionality to serve either static content (ISAPI) functions or ASP scripts. It used ASP file extensions to determine application and hand off the scripts directly to script engine, thus bypassing the static content handler. This architecture endured in all forthcoming versions.

IIS 6.0 no longer was installed by default, as it shipped with Windows Server 2003. Even after installation it served only HTML static pages by default. This minimized security holes of the operating system. Administrators and programmers could restrict account access and had to explicitly enable ASP and ASP.NET. Kerberos authentication allowed secure communication within an Active Directory domain and solved many remote permission issues.

In IIS 6.0 request processing was changed. IIS 6.0 used the HTTP.sys listener to accept the requests and then delegate them to worker processes. These workers were isolated in application pools, which made it possible to restrict the consequences of possible process failures within the pools. The separate memory pool created by an administrator for separate applications prevented a faulty application from crashing other applications outside its memory pool. This feature made it possible for the overall system stability to stay intact.

IIS 7.0 was fully rewritten from the base code of previous versions. It became the programmer's dream because of many significant new features. ASP.NET platform was fully integrated, and configuration was transferred into XML files. Also request tracking, diagnostics and new administration tools were welcomed by programmers and administrators. The modular structure of IIS 7.0 and IIS 7.5 allowed an easy implementation and installation of modules with custom functionality. A new unified IIS Manager combined all management functions of IIS and ASP.NET functions in one location. Microsoft's PowerShell was fully integrated to IIS 7.0 and Request Tracing could be configured at the server, site, or application level [4].

Currently the newest version of IIS is 8.0. This version is not a radical change from IIS 7.5, but due to some changes it embraces cloud and newest storage architectures. IIS 8.0 functions now integrate well also with non-Microsoft technologies, going towards open web environment. IIS 8.0 is fully integrated application environment. New IIS

Configuration Editor allows the administrator to change every configurable feature via the user interface, C# and JavaScript. PowerShell is becoming the default scripting language for administrators and allows to script IIS configuration in PowerShell.

4.4 Installing IIS 8 on Windows Server 2012

The installation of IIS 8 on Windows Server 2012 is done via Server Manager. With the basic installation the server is able to serve a static content. It is very common, that the user gets IIS 8 preinstalled with the MS Server operating system from the vendor. Later the user adds required features to existing installation. To perform default IIS 8 installation, the user opens the Server Manager in MS Server 2012 and from the dashboard selects Add Roles and Features option. See Figure 12.

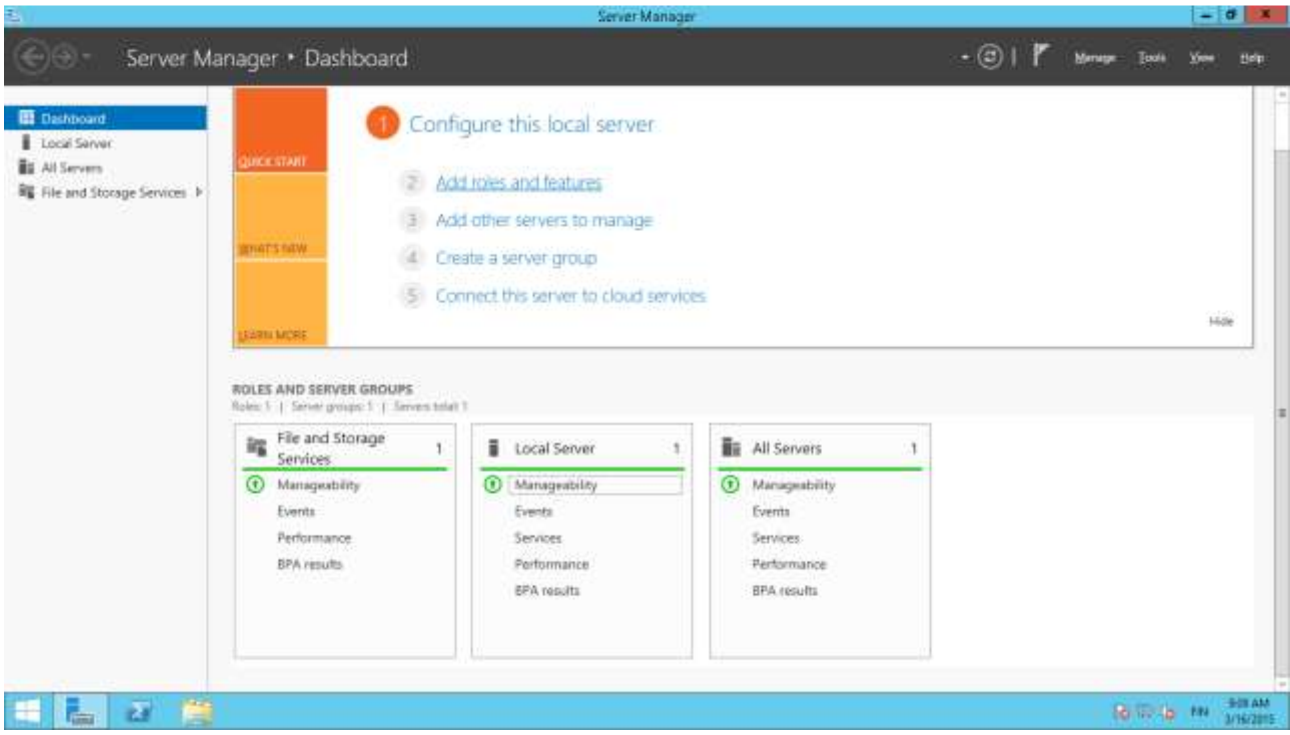
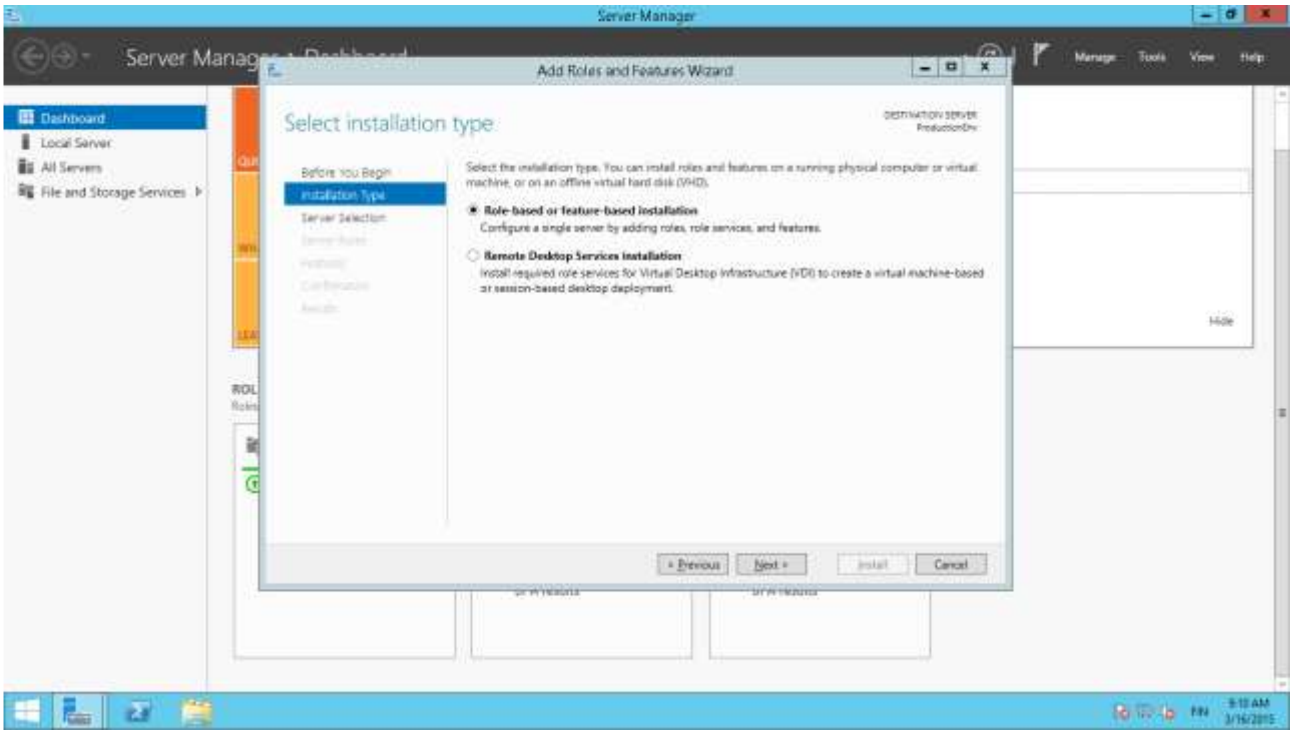


Figure 12. Installing IIS 8 from “Add roles and features” option.

An installation wizard is opened and the user selects, in this case, the role-based or future-based option. See Figure 13.



Select installation type

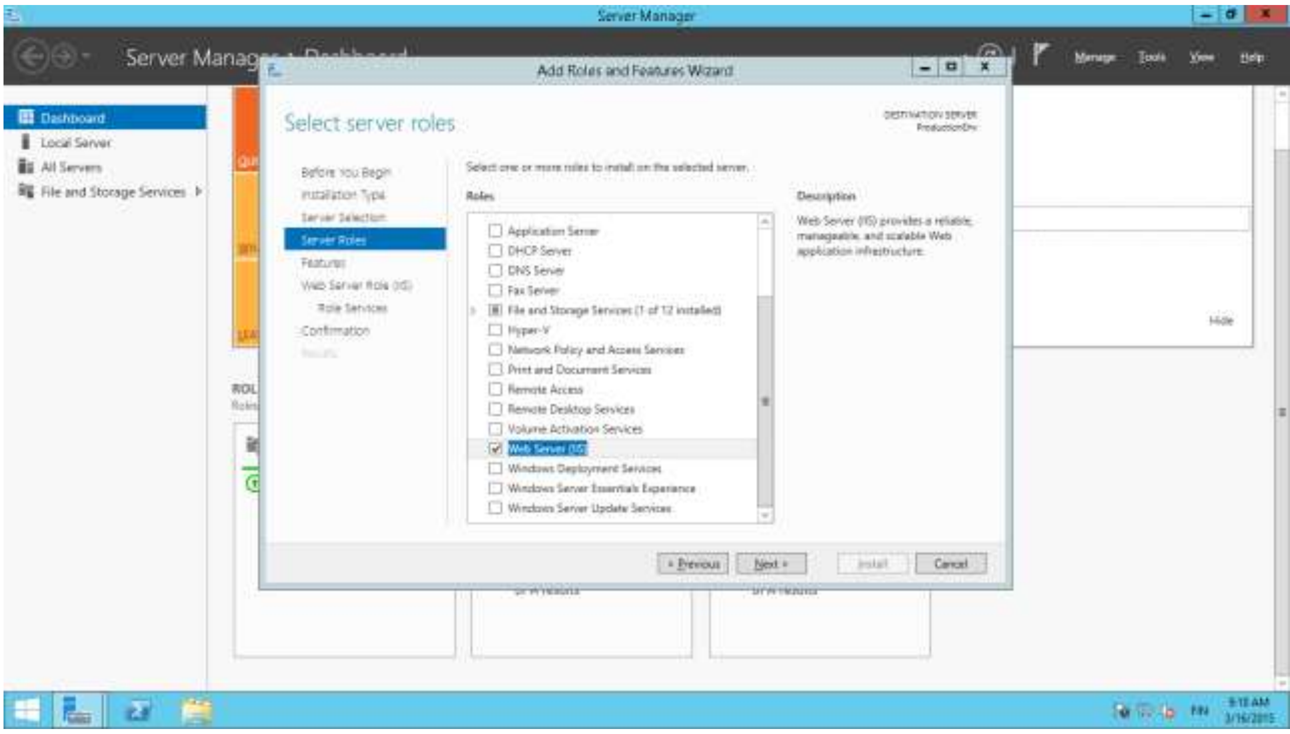
Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).

● **Role-based or feature-based installation**
Configure a single server by adding roles, role services, and features.

○ **Remote Desktop Services installation**
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.

Figure 13. Selecting the “Role Based or Future Based” installation.

The next step is to select the Web Server (IIS) option and finalize the installation. See Figure 14 and 15.



Select server roles

Select one or more roles to install on the selected server.

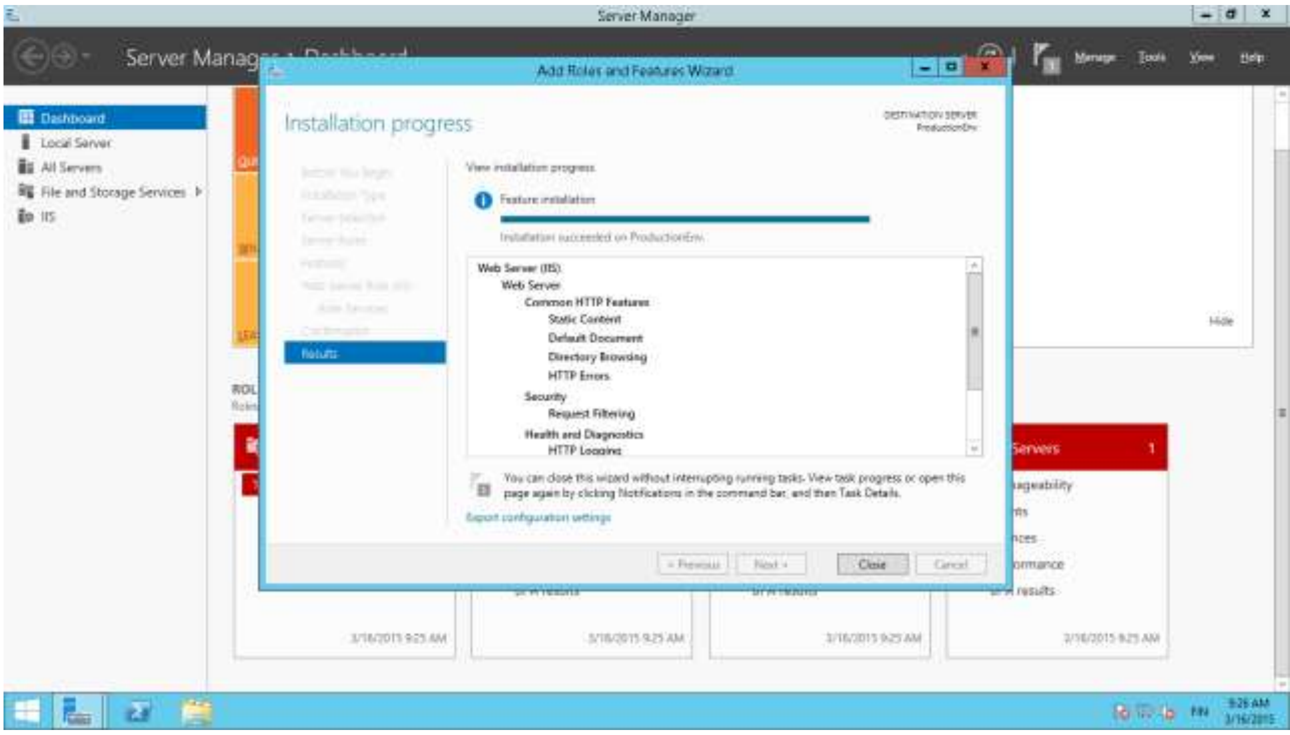
Roles

- ☐ Application Server
- ☐ DHCP Server
- ☐ DNS Server
- ☐ Fax Server
- ▣ File and Storage Services (1 of 12 installed)
- ☐ Hyper-V
- ☐ Network Policy and Access Services
- ☐ Print and Document Services
- ☐ Remote Access
- ☐ Remote Desktop Services
- ☐ Volume Activation Services
- ☑ Web Server (IIS)
- ☐ Windows Deployment Services
- ☐ Windows Server Essentials Experience
- ☐ Windows Server Update Services

Description

Web Server (IIS) provides a reliable, manageable, and scalable Web application infrastructure.

Figure 14. Selecting Web Server (IIS) for installation.



Installation progress

View installation progress

Feature installation

Installation succeeded on ProductionEnv.

```
Web Server (IIS)
    Web Server
        Common HTTP Features
            Static Content
            Default Document
            Directory Browsing
            HTTP Errors
        Security
            Request Filtering
        Health and Diagnostics
            HTTP Logging
```

You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Figure 15. The installation wizard finalizing IIS 8.0 basic installation.

The installed Internet Information Manager with a basic set of installation looks as in Figure 16.

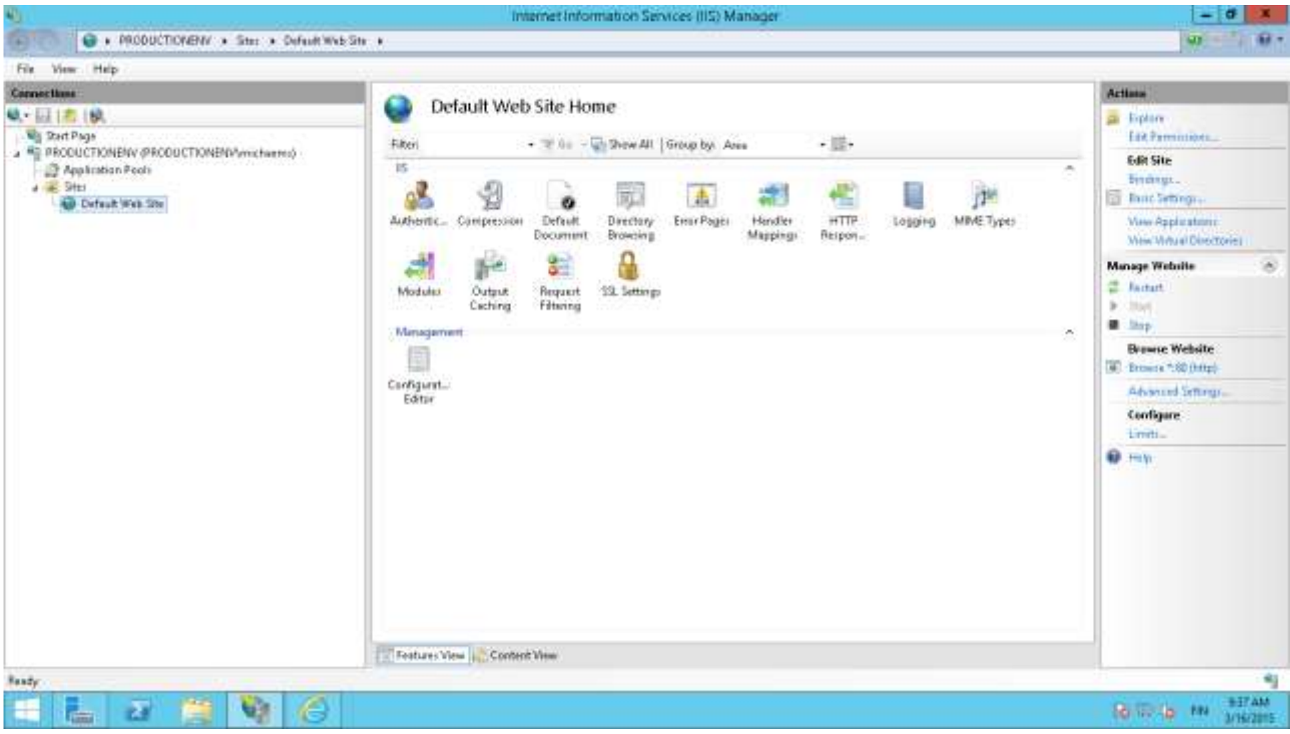


Figure 16. Internet Information Manager with basic installation.

To test the installed IIS 8 the following site is accessed: http://localhost/iisstart.htm from the browser. The result is shown in Figure 17, which indicates that IIS 8.0 basic installation was successful.

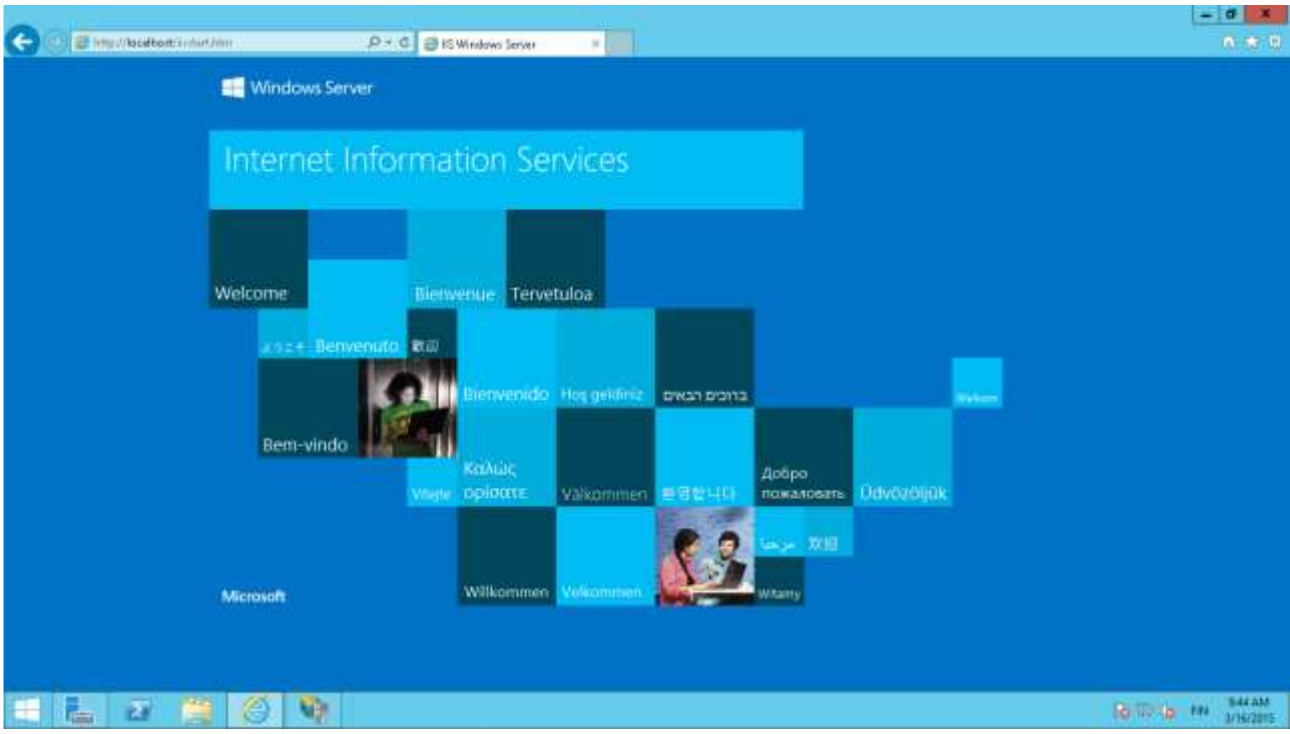


Figure 17. IIS 8.0 test interface showing IIS Windows Server as successfully installed and started.

4.5 Introducing MS SQL SEVER 2012

The first version of Microsoft SQL Server was released in 1989 for OS/2 operating system. The release was a result product of collaboration of three corporations: Microsoft, Sybase and Ashton Tate. Nowadays Microsoft Corporation fully owns this software product line. Microsoft SQL Server is a relational database management system (RDMS). It is database software, whose primary task is to save and retrieve data requested by other software applications. Those applications may reside in the same computer or alternatively the request may come from the software running in computer which resides in another corner of the globe.

There are five editions of MS SQL Server 2012. One may see the descriptions of those editions in Table 4.

Table 4. MS SQL Server 2012 editions.

Edition	Description
Standard	Standard edition Includes basic data management and Business Intelligence capabilities. It suits the needs of relatively smaller organization.
Business Intelligence	Business Intelligence edition allows an organization to build, deploy and manage high scalability solutions. The edition is new in the SQL Server family. Its main focus is on delivering Business Intelligence–focused solutions.
Enterprise	Enterprise edition is the premium edition of SQL Server. It contains all available functions found in other editions, and provides a comprehensive data center solution.
Developer	Developer edition is identical to Enterprise Edition, but its main purpose is to deliver license for developing and testing solu-

	tions. It can be easily updated to Enterprise Edition if needed.
Express	Express Edition is free and its main purpose is to deliver entry–level product. This edition is ideal for learning and developing small–scale applications.

Very often MS SQL Server 2012 as a product is divided roughly into two distinct categories: Database Engine and Business Intelligence (BI). Altogether, the product contains several components. Those components are Business Intelligence, Database Engine, T-SQL Programming Interface, Security Subsystem, Replication, SQL Server Agent, SQL Server Integration Services, and SQL Server Management Tools.

Business Intelligence of MS SQL Server is a mechanism or tool, whose primary task is to transform raw data into knowledge which can be used to make more informed and intelligent business decisions. For example, a company selling daily consumable stuff could use its data to identify sales trends and from that analysis the company could decide to focus sales efforts on a particular region or area, thus making more profit.

The Database Engine is the core component of MS SQL SERVER 2012. It operates as a service on the hardware and is often referred to as an instance of the SQL Server. In order to get data from the SQL Server, the application needs to connect to the Database Engine. After a connection is established, the client application will send T-SQL statements to the instance (or database engine). The instance in return will send data back to the client application. Security layer resides within the established connection and validates access to data as specified by security policy [5].

The storage component of the Database Engine takes care of storing data on the disk. It physically organizes tables, indexes and other related structures on a computer’s disk subsystem. The storage component also referred to as Storage Engine is a core component within Database Engine.

In order to retrieve data, SQL Server provides a rich programming language called Transact-SQL or T-SQL. The language is implemented in the T-SQL Programming Interface component, which resides in the Database Engine. Using T-SQL, a user may write data manipulation queries that enable the user to modify or access the data on demand. T-SQL also makes possible to write queries for creating database objects such as tables, views, indexes, stored procedures triggers and user-defined functions. .NET based programming languages such as C# or Visual Basic.NET can also send commands from client applications to Database Engine. T-SQL language has several enhancements in MS SQL Server 2012. Such enhancements are for example windows functions, error handling and a simpler form of paging.

The Security Subsystem of SQL SERVER 2012 is very robust and allows a user to control access via two modes of authentication: Windows and SQL. An administrator may configure SQL Server on multiple levels. The access can be controlled to a particular instance of SQL Server, to a specific database, to objects of that database or even to columns within a particular table.

Native and Transparent Data Encryption allow the administrator of the database to encrypt an entire database without affecting how clients access the data. In SQL SERVER 2012 it is also possible to create users within the database without requiring creating server login known as contained databases. This option was not available in the previous versions of SQL Server. An administrator was required to create a server login prior to granting access on the database level.

Replication (or Replication Component of the Database Engine) allows distributing data locally or to different locations, using the File Transfer Protocol, over the internet and to mobile users. It can be configured to merge data, pull data, and push data across wide area networks or local area networks. Snapshot replication is the simplest type of replication. It periodically takes a snapshot of the data and distributes it to subscribed servers. This kind of replication is used for moving data at longer intervals, for example nightly.

Another type of replication, called transactional replication is used if users demand higher throughout. Instead of snapshot distribution, transactional replication sends the data as soon as change happens. In transactional replication, server-to-server topology is used. Usually in this topology one server is the source of data and the other one is used for backup copy or for reporting [6].

The primary task of the SQL Server Agent is to execute scheduled tasks, for example, loading the data warehouse, backing up databases or rebuilding indexes. This component runs on a separate service. Each instance of SQL Server has a corresponding SQL Agent service. SQL Agent can be configured to send alerts or notifications when the executed job succeeds, fails or completes.

Another component residing in Database Engine is called SQL Server Integration Services (SSIS). This platform makes possible of creating high-performance frameworks for extraction, transformation and loading (ETL) of data warehouses. Using SSIS, the user may quickly import or export data to various sources such as Oracle, text file, Excel and so on.

To monitor, manage, maintain and develop SQL Server environment, a graphical user interface is included in SQL SERVER 2012. It is called SQL Server Management Studio (SSMS). SSMS allows the user to perform different actions against an instance of SQL Server. It is an integrated environment which includes a broad set of tools that simplify and the process of developing and configuring SQL Server instances [6]. One may see SSMS interface in Figure 18.

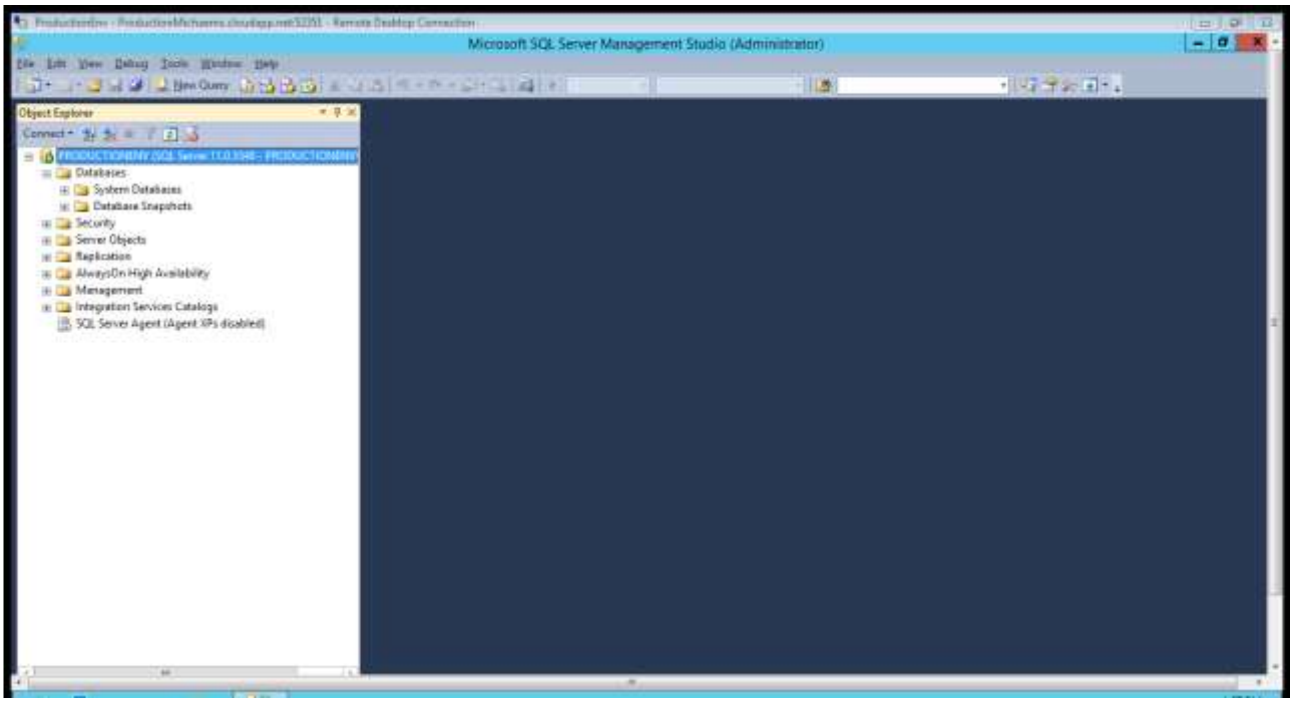

Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
Databases
System Databases
Database Snapshots
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)

Figure 18. User interface of SQL Server Management Studio.

5 Project Development Environment

To develop the ASP.NET application, an appropriate development environment was needed. The development environment for the project of this final year thesis was selected to run on Microsoft Azure Cloud service on a separate virtual machine. Windows Server 2012 R2 was selected as an operating system which hosted Microsoft .NET platform, development tool Microsoft Visual Studio 2013, Microsoft SQL Server 2012 R2, and a version controlling tool called Microsoft Team Foundation Server (TFS).

The creation of a new virtual machine for development environment was similar to the one previously created in section 4.2, which is why the process will not be repeated in this section. During the creation of new virtual machine for development named "DevelopmentEnv", an image with Visual Studio 2013 was selected, which installed also Microsoft Visual Studio 2013 on Windows Server 2012 operating system. See Figure 19.

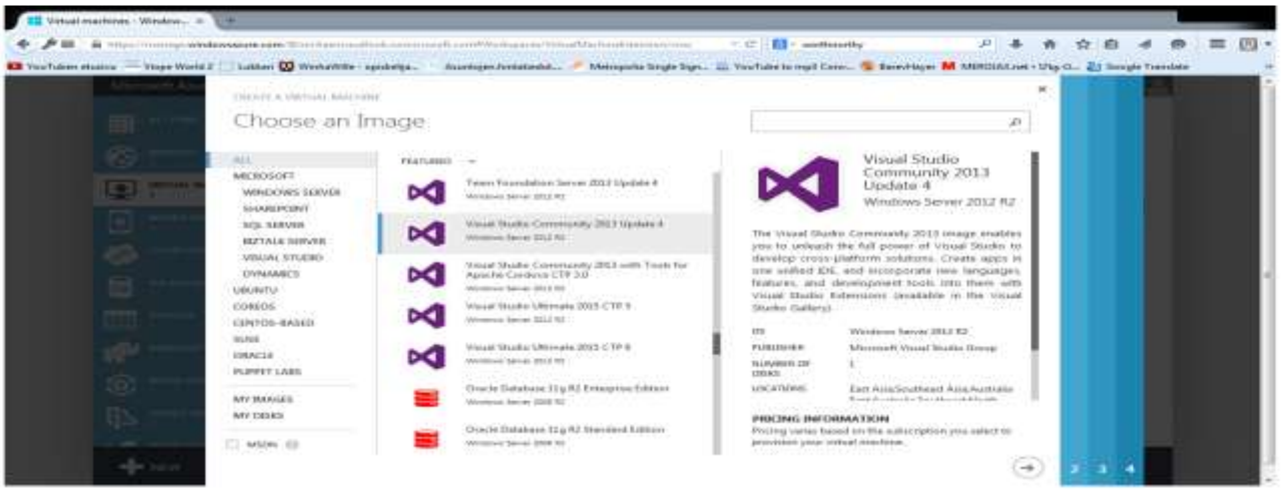


Figure 19. Creation of "DevelopmentEnv" virtual machine with Visual Studio 2013 development tool.

After a new virtual machine had been created and connected, Visual Studio 2013 was executed and was ready for creating new applications. See Figure 20.

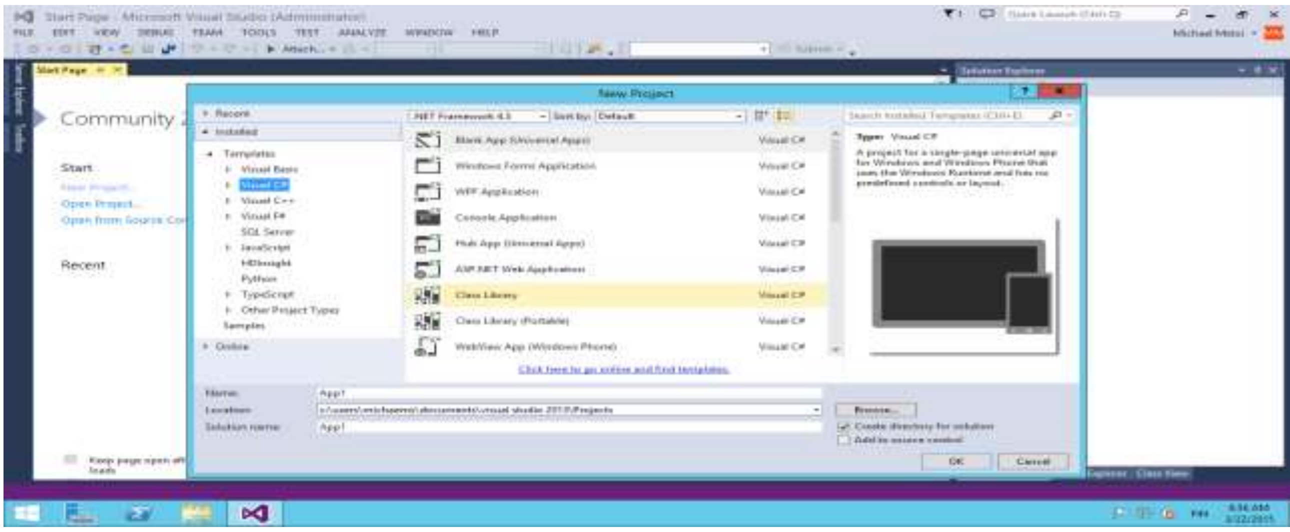


Figure 20. Visual Studio 2013 Community Edition on newly created virtual machine.

The next section will describe a software platform, which was used for the development of the application of the final year project.

5.1 Microsoft .NET Framework

The development of .NET framework began by Microsoft Corporation in late 1990's. It is a software platform that runs on Microsoft Windows. It provides Language Interoperability, which in other words means that programming languages supporting .NET framework can use the code written by other programming languages supporting the .NET framework. Another significant issue concerning .NET is that it provides a very large class library, which is called Framework Class Library (FCL). FCL provides numeric algorithms, network communications, web application development, user interface, data access, database connectivity and cryptography.

Applications created on .NET framework are executed by an application virtual machine which provides memory management, security, application hosting, execution and threads handling and some other services. This virtual machine is called Common Language Runtime (CLR). So CLR and FCL together constitute the core of .NET Software Framework.

Common Type System or CTS for short is another building block of .NET Framework. The role of CTS specification is to fully describe all possible programming constructs and data types supported by the runtime. It also details how they are represented in .NET metadata format and how those entities shall interact with each other. Another

specification, called Common Language Specification (CLS), defines common types and programming constructs that all .NET programming languages shall agree on. .NET programming languages are also called managed languages.
With the installation of Microsoft Visual Studio development tool, the user gets options for using following managed languages: C#, Visual Basic.NET, JScript.NET C++/CLI, and F#.

.NET-aware compiler creates a *.dll or *.exe binary files from the source code of the programmer. The created binary file (.dll or .exe) is termed as assembly. The assembly contains CIL (Common Intermediate Language) code. CIL is similar to Java bytecode in that it is not compiled to platform-specific instructions. The created assembly also includes metadata, which vividly describes in detail the characteristics of every entity (or type) resided in the binary. To describe the assembly itself, the assembly includes another metadata called manifest. The manifest of the assembly contains a list of referenced external assemblies needed to properly execute the code, version information of the assembly and culture information.

CIL or Common Intermediate Language provides language integration. Each .NET – aware compiler creates nearly identical CIL instructions. Before running the code, a CIL code needs to be compiled on the fly. That is why an entity called JIT compiler is used to compile CIL code into meaningful CPU instructions. Thus programmers can write a code that can be efficiently compiled with the JIT compiler and executed on hardware with a different architecture. Once CIL instructions are compiled into platform-specific code, they are retained in the memory for later use after the first call. Next time a function is called there is no need to compile the CIL code again.

Every developer understands the importance of code libraries. Libraries like MFC, Java EE, and ATL among others enable developers to leverage their code with well-defined set of existing code. However C# or other .NET – aware programming languages do not provide language – specific code library. Rather developers are using .NET – aware languages to leverage .NET libraries, which are language – neutral. That is why .NET uses extensively namespace concept, in order to keep all the types within the base class libraries well organized.

A namespace is a group of related types, found in the assembly. For example, System.Data namespace contains basic database types; mscorelib.dll assembly contains many namespaces, such as System.IO, System.Collections, and System.Diagnostics.

Let us take a small example for demonstrating assembly, namespace and type concept. Most developers have used the console application of Windows operating system. In order to use the windows console from the developed C# or Visual Basic.NET application, the .NET framework provides Console class (or type in .NET jargon).

To write some text to console, the developer calls for example WriteLine() method of Console type: Console.WriteLine(“Hello from C#”). The Console type resides in System namespace. System namespace on the other hand resides in mscorelib.dll assembly. So in order to use the Console type, the developer includes System namespace into source code by means of “using” keyword: “using System;” This action makes sure that crated executable has a link to mscorelib.dll assembly in its metadata, and mscorelib.dll assembly is loaded into memory during application execution [7].

.NET platform is considered platform–independent. Thus .NET assemblies may be used to develop and deploy applications on non-Microsoft platforms, such as Solaris, Mac OS X, Linux, and so on. The reader may see the names and the explanations of some namespaces of .NET Framework in Table 5.

Table 5. Some namespaces of Microsoft .NET Framework.

System	The namespace contains types for mathematical computations, environment variables, garbage collection, random number generation, and so on.
System.Collections	The namespace allows the developer to create customized collections. It contains base types, interfaces and container types.
System.IO System.IO.Ports	The namespaces are used for port manipulation, file I/O, compression of data, and so on.
System.Windows.Forms	The namespace contains .NET Frame-

	work's original user interface toolkit. Types defined in this namespace are used to create Windows desktop applications.
System.Web	The namespace allows the developer to create ASP.NET web applications.
System.Threading System.Threading.Tasks	The namespaces are used for multi-threaded applications, which are able to distribute workloads to multiple CPUs.
System.Data System.Data.Common System.Data.SqlClient	The namespaces are used to interact with relational databases using ADO.NET.
Microsoft.CSharp Microsoft.ManagementConsole Microsoft.Win32	The namespaces are called Microsoft root namespaces. They contain types that are used to interact with services unique to the Windows operating system.

Microsoft .NET platform and its namespaces are used by different application development tools which are provided by numerous vendors in the market. Next, a development tool used in this final year project will be introduced.

5.2 Microsoft Visual Studio

Microsoft Visual Studio is an IDE for developing .NET based applications running on Microsoft Windows operating system. Other types of applications created by Visual Studio are Web applications and Web Services. IDE means Integrated Development Environment. It is possible to create .NET application using freely available .NET SDK and notepad editor. But developers working for industrial enterprises seldom create applications in that way. Instead they use Microsoft Visual Studio for application development and get many advantages to create, test and deploy their applications to end users much faster.

MS Visual Studio can produce both managed code and unmanaged code. Intrinsically it does not support any programming language or tool, but instead it allows plugging codded functionality. The plugged functionality usually is codded and packed into VSPackage. After installation the functionality is available as a service. After installing

Visual Studio, the following built-in languages are available for immediate use: VB.NET, C#, F#, C, C++, and C++/CLI. It supports also other languages, provided that language-specific service exists. Via language services installed separately, Visual Studio supports M, XML/XSLT, HTML, CSS, Python, and Ruby languages.

Visual Studio integrates various Microsoft platforms to produce different types of applications. Those platforms are Windows API, Windows Forms, Windows Store, Windows Presentation Foundation (WPF) and Microsoft Silverlight. As for the features Visual Studio provides following: Code Editor, Debugger, Windows Forms Designer, Web Designer, Class and Data Designers, Object Browser, Solution Explorer, Data Explorer, Properties Editor and other features [8].

Visual Studio 2013 Community edition was used in this Final Year Project. This edition was launched in November 2014 and is going to take over traditional Visual Studio Express edition. Express editions in previous versions of Microsoft Visual Studio were free and were allowed to be used for commercial applications. But Express versions had many limitations and did not provide many features of other editions. In contrast of Express edition, Visual Studio 2013 Community edition has no any limitation for application development and is allowed to be used for commercial and non-commercial application development. It is intended for small-size enterprises with a maximum of five application developers. One may see other editions of Microsoft Visual Studio 2013 in Table 6.

Table 6. Microsoft Visual Studio 2013 editions.

Ultimate 2013 with MSDN	Reliably captures and reproduces bugs found during manual and exploratory testing to eliminate "no repro" bugs. Performs unlimited web performance and load testing. Collects and analyses runtime diagnostic data from production systems. Includes all remaining features in other editions.
Premium 2013 with MSDN	Improves code quality with a peer code review workflow within Visual Studio. Automates user interface tests to validate application UI. Finds and manages duplicate code in the code base to improve the application architecture. Determines how much code is being tested with code coverage analysis.
Test Professional 2013 with MSDN	Organizes and defines test plans with test case management and exploratory testing. Provisions and manages virtual lab environments for testing with consistent configurations.
Professional 2013 with MSDN	Gets access to Microsoft platforms and tools past and present, with new releases added all the time, including Visual Studio. Includes also capabilities of Online Editi-on.
Visual Studio Online Professional 2013	Works in the same IDE to create solutions for the web, desktop, cloud, server, and phone. Hosts team projects only in the cloud.

5.3 Team Foundation Server: Installation and Usage

Developing an application is not an easy task. An essential factor to successfully develop the software is how the development team members communicate with each other, as well as with people who have a significant stake in the software being developed.

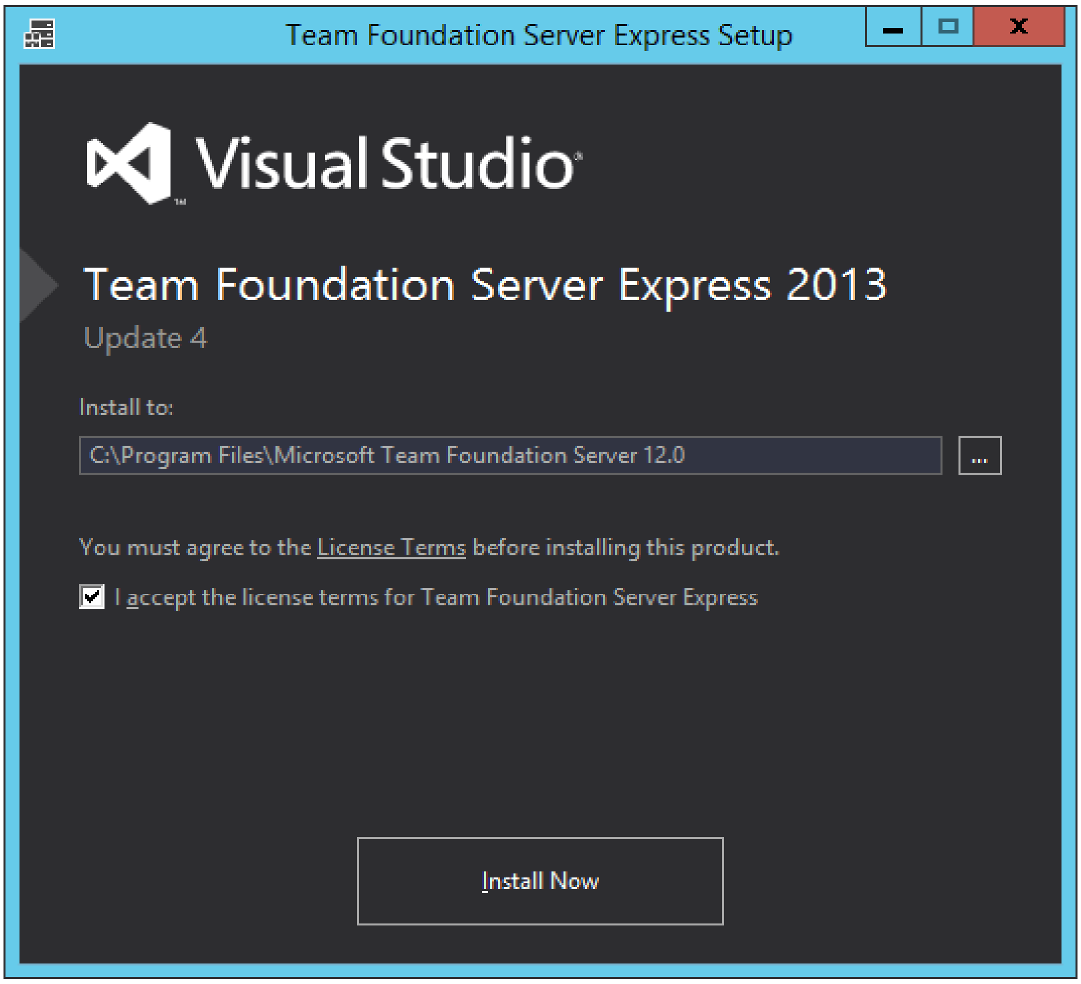
Team Foundation Server is a tightly integrated environment which provides a core collaboration functionality for software development teams. The core collaboration functionality includes the following components: Version Control, Project Management, WIT (Work Item Tracking), Test Case Management, Build Automation, Reporting and Feedback Management. Thus, different engineering teams, such as developers, testers, architects, business analysts, project managers and other groups contributing to the software being developed use Team Foundation Server to collaborate and communicate accurately.

The Team Foundation Server is made up of two tiers. Those tiers can be physically deployed across one or several computing nodes. The first tier is called Application Tier. This tier consists of a set of web services. The client computer communicates with this tier by using web service-based protocol. The tier also includes web access site for interacting with the server without using Visual Studio as a client. Second tier is called Data Tier. This tier is made up of a Microsoft SQL Server database, which contains the database logic for the Team Foundation Server application as well as the data for Team Foundation Server instance [9].

The Express edition of Team Foundation Server provides Version Control, Work Item Tracking and Build Automation. This set of functionality is fully enough for this Final Year Project. Express edition is perfect start for small teams up to five developers. Microsoft also provides 90-day trial edition which includes all missing functionalities of Express edition. After trial period is over, it is possible to apply the license product key to activate full functionality of TFS. Trial edition can be downloaded and installed from http://aka.ms/TFS2013Downloads site. For this Final Year Project the Express edition is downloaded and installed from https://www.visualstudio.com/downloads/download-visual-studio-vs#d-team-foundation-server-express site.

Setting up the Team Foundation Server is roughly divided into two distinct phases: Installation phase and configuration phase. The installation phase finalizes copying the components onto the hardware. During the configuration phase the administrator of Team Foundation Server decides which SQL Server instance to use, which accounts to use for permissions, which optional components to use and other similar options. Figure 21 shows the installation wizard of Team Foundation Server 2013 Express edition. Installation package was downloaded into Development Environment virtual machine (which hosts already Windows Server 2012 R2 operating system and Visual Studio 2013 Community edition).

After the required components have been copied onto hardware, the configuration wizard starts, may be seen in Figure 22. The configuration wizard provides a guided way of picking the best configuration options for the Team Foundation Server. In our case the configuration wizard installed Microsoft SQL Server 2012 Express edition into the Development Environment.


Team Foundation Server Express Setup

Visual Studio

Team Foundation Server Express 2013

Update 4

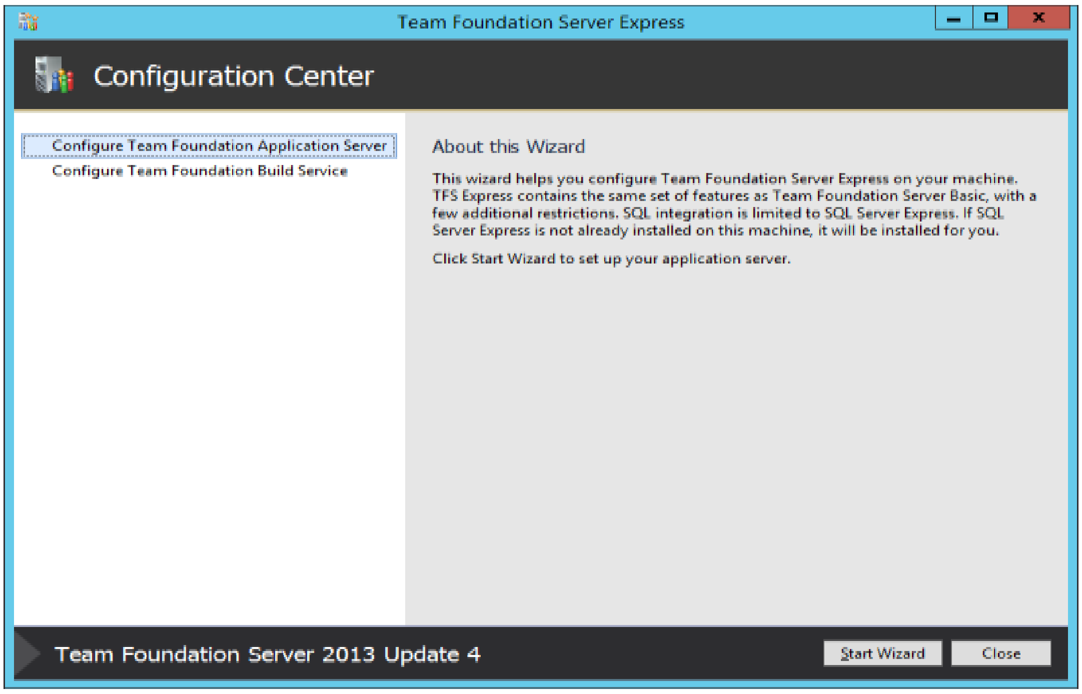
Install to:

C:\Program Files\Microsoft Team Foundation Server 12.0

You must agree to the License Terms before installing this product.

☑ I accept the license terms for Team Foundation Server Express

Figure 21. Wizard for Installing Team Foundation Server 2013 Express edition.


Team Foundation Server Express
Configuration Center
Configure Team Foundation Application Server
Configure Team Foundation Build Service
Team Foundation Server 2013 Update 4
Start Wizard
Close

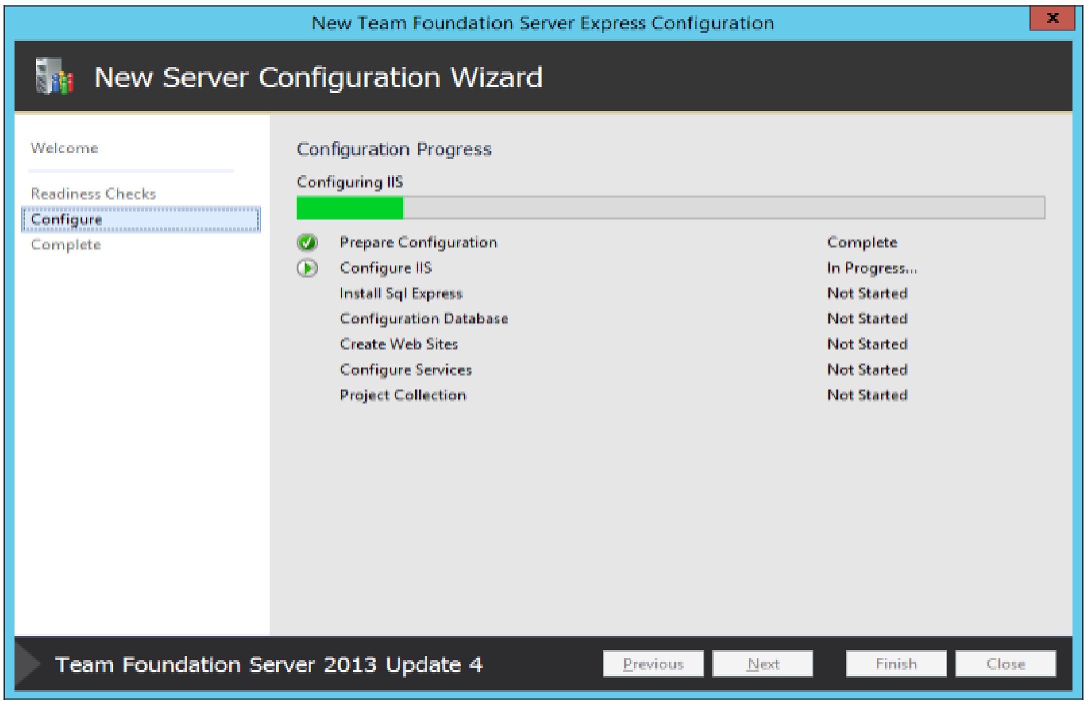
About this Wizard

This wizard helps you configure Team Foundation Server Express on your machine. TFS Express contains the same set of features as Team Foundation Server Basic, with a few additional restrictions. SQL integration is limited to SQL Server Express. If SQL Server Express is not already installed on this machine, it will be installed for you.

Click Start Wizard to set up your application server.

Figure 22. Configuration Center of Team Foundation Server 2013 Express edition.

During the configuration wizard, the user may observe how the configuration progress evolves forward. See Figure 23.


New Team Foundation Server Express Configuration
New Server Configuration Wizard
Welcome
Readiness Checks
Configure
Complete
Team Foundation Server 2013 Update 4
Previous
Next
Finish
Close

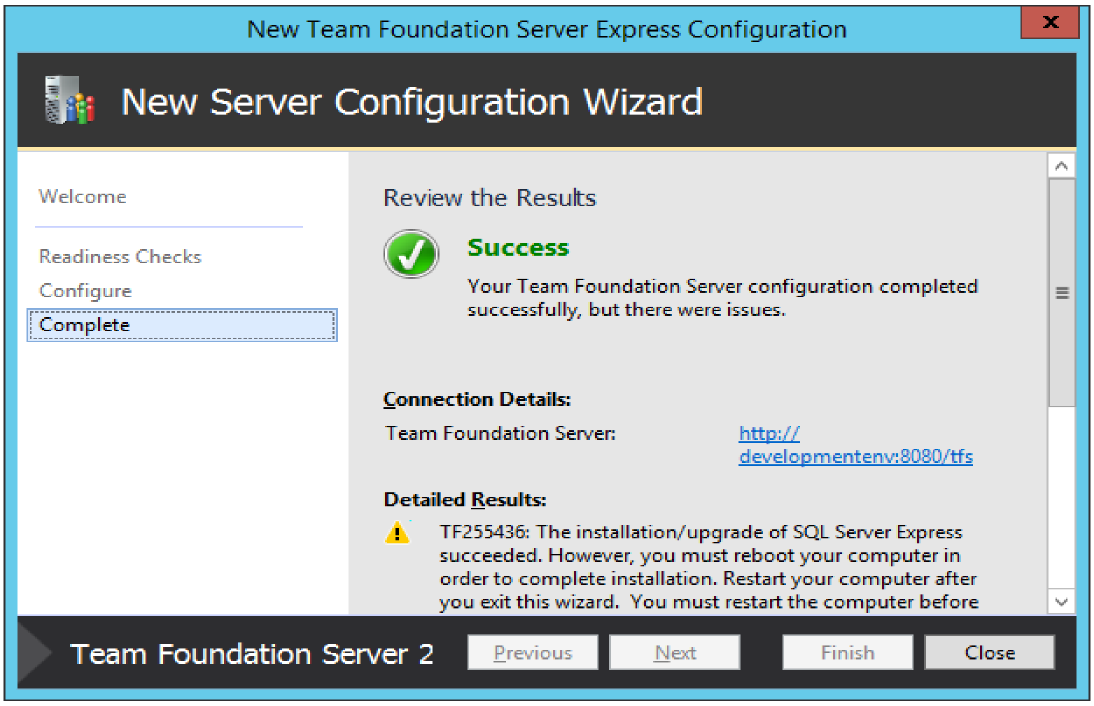
Configuration Progress

Configuring IIS

Prepare Configuration	Complete
Configure IIS	In Progress...
Install Sql Express	Not Started
Configuration Database	Not Started
Create Web Sites	Not Started
Configure Services	Not Started
Project Collection	Not Started

Figure 23. Configuration of different components of Team Foundation Server 2013 Express edition.

Finally, when all the components are configured, the user will be informed about the results. In our case, configuration progressed smoothly, as can be seen in Figure 24.



Review the Results

Success

Your Team Foundation Server configuration completed successfully, but there were issues.

Connection Details:

Team Foundation Server: http://developmentenv:8080/tfs

Detailed Results:

TF255436: The installation/upgrade of SQL Server Express succeeded. However, you must reboot your computer in order to complete installation. Restart your computer after you exit this wizard. You must restart the computer before

Figure 24. Final wizard shows the configuration results of Team Foundation Server 2013 Express edition.

Before saving any source code and using source control for further application development, a team project must be created. A team project is the basic container of the work used by TFS (Team Foundation Server). The Team project is created through Team Explorer, which is part of Visual Studio 2013. From Team Explorer the Connect icon is pressed and TFS running on local host is selected. See Figure 25.

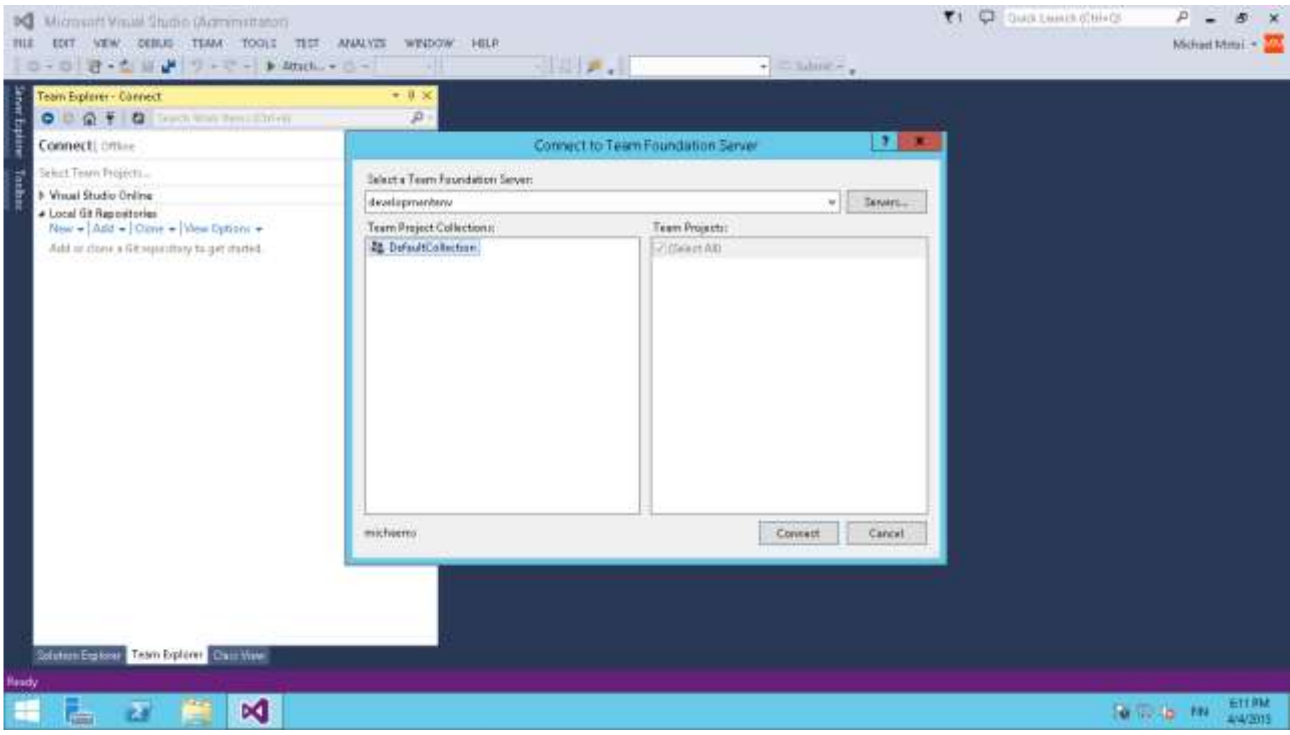
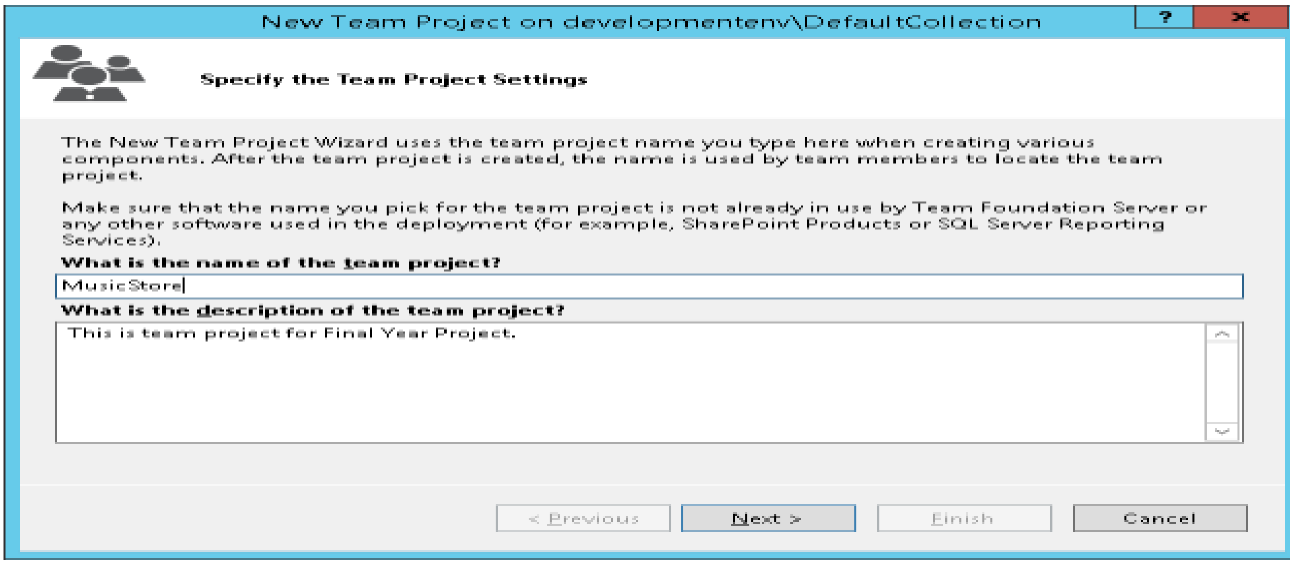

Microsoft Visual Studio (Administrator)
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Team Explorer - Connect
Connect | Offline
Select Team Projects...
Visual Studio Online
Local Git Repositories
New | Add | Clone | View Options
Add or clone a Git repository to get started.
Connect to Team Foundation Server
Select a Team Foundation Server:
developmentenv
Servers...
Team Project Collections:
DefaultCollection
Team Projects:
(Select All)
michaemo
Connect
Cancel
Solution Explorer
Team Explorer
Class View
Ready

Figure 25. Connecting Team Foundation Server from Visual Studio’s Team Explorer.

After selecting DefaultCollection and pressing the Connect button, Team Foundation Server will be connected, and the user can create Team Project. It is created by selecting File -> Team Project from the menu of Visual Studio 2013. The Wizard is opened and the user enters name and description for new Team Project. All other options are leaved by default. See Figure 26.


New Team Project on developmentenv\DefaultCollection

Specify the Team Project Settings

The New Team Project Wizard uses the team project name you type here when creating various components. After the team project is created, the name is used by team members to locate the team project.

Make sure that the name you pick for the team project is not already in use by Team Foundation Server or any other software used in the deployment (for example, SharePoint Products or SQL Server Reporting Services).

What is the name of the team project?

MusicStore

What is the description of the team project?

This is team project for Final Year Project.

< Previous | Next > | Finish | Cancel

Figure 26. Create new Team Project from File->Team Project from the menu of Visual Studio 2013.

As soon as the new team project is created, it may be seen in the source control explorer by selecting the Source Control Explorer option in the Team Explorer. See Figure 27.

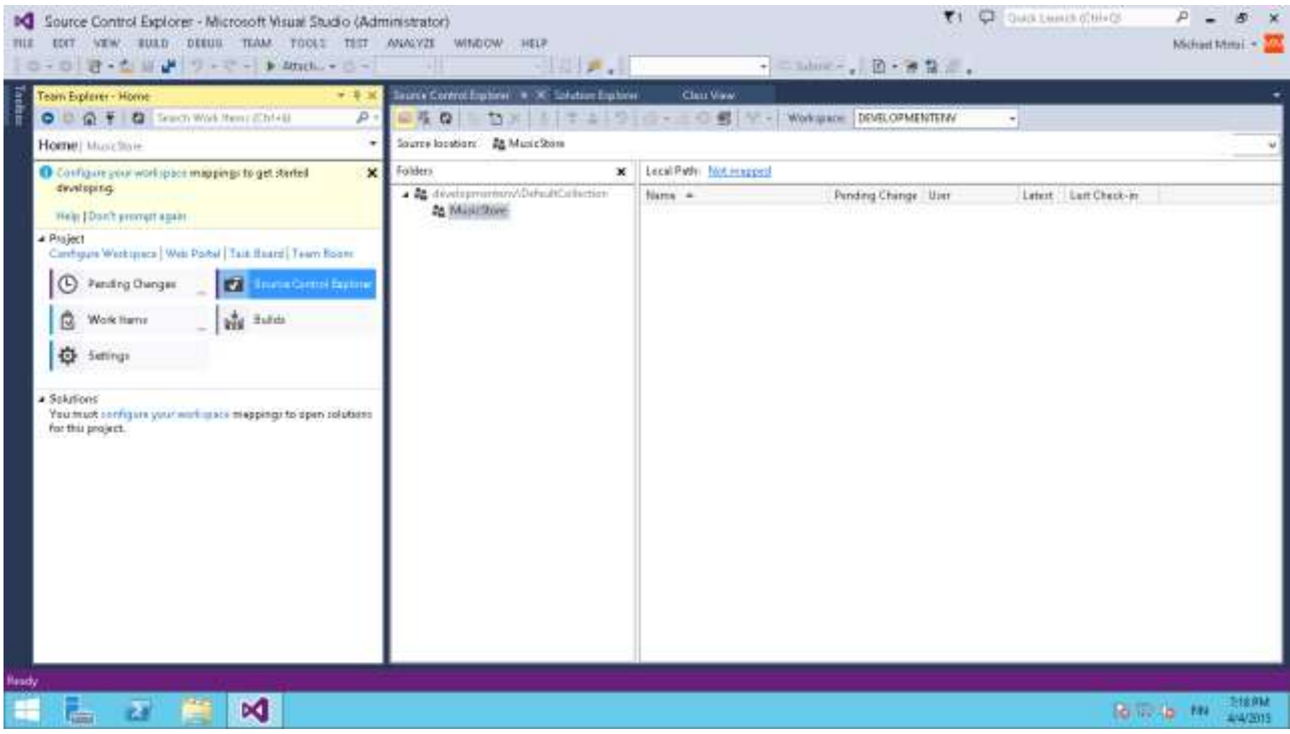


Figure 27. Newly created Team Project in Source Control Explorer.

Finally, Team Foundation Server may be configured to be connected from any computer in the world, and it can be given a friendly DNS name for easier connection from users across the world. In our case, there is no need to be connected from the outside of the development environment, thus no friendly DNS shall be given to the installed TFS. Finally some version control concepts will be explained. Version control is the single most important tool used during development of software. That is why it is a big advantage to understand the purpose, core concepts and benefits of version control before using it for the production.

There are many names used for version control. For example, revision control, source control and SCM (Software Configuration Management) are used to refer to the same set of functionality. Generally speaking, version control provides following capabilities: a) a place to store source code, images, build scripts, and other artifacts related to software product, b) the ability to track the history of those files, and c) mechanisms to work in parallel with team of software developers. If one of the developers makes mistake during source code editing or accidentally deletes some files, a version control system lets the developer to roll back time and get back the state before editing or accidentally removing important files.

There are 10 core concepts of version control system: Repository, Working Copy, Working Folder Mappings, Get, Add, Check-Out, ChangeSet, Check-In, Branching and Merging, and History.

The Repository of the version control system is a place where the source code of the software is saved. It is represented as a tree of files, similar to regular directory structures. However, the repository of the version control system is able to track the changes of those files, which makes it possible to know the state of any file at any given time. The Team Foundation Server saves the initial version of the file, and then stores the changes between each version for subsequent changes.

Before making changes to the existing version of software, the developer downloads the source code from the repository of the version control system to its local computer. This copy of the software is called a working copy. This local version or a working copy is also known as workspace in Team Foundation Sever. It can also be called a sandbox. After editing the code and making the changes to the working copy, the developer commits it to repository, and thus other developers can access to new version of software. Using a working copy makes it possible for developers to work in parallel.

A Working Folder Mapping is the link between working copy of source code in local system of the developer and the place in repository, where all files are stored. Once the working folder mapping has been set up, the developer must download the files from repository to its local machine (or local system). In Team Foundation Server this process is known as Get. In some other version control systems, for instance in SVN, this process is called check-out.

As the developers begin to develop the software, there is no file in the repository. That is why they need to Add created files to repository. They select which files and folders to take from the local machine and add to repository, so those may be shared to the rest of the development team. Naturally there is no need to share all files of the project. For example, compiled binaries derived from the source code usually are not shared. Team Foundation Server provides tools to developers for selecting files to add to repository.

Check-Out means a slightly different thing in other version control systems. For example in SVN Check-Out is the same as Get (download files to local machine) in Team Foundation Server. In Team Foundation Server check-out means that developer is currently working on working copy and a notification is sent through TFS to other developers. However, in latest version of TFS, there is no need for developer to explicitly check-out files in order to work on therm. This new mode of TFS is called Local Workspace. With Local Workspace the developer can edit any downloaded file immediately.

As files are checked out and edited, the developer needs to apply generated changes back to repository. The set of changes to be applied is called ChangeSet. The ChangeSet contains all the changes to be done with the files. As soon as the ChangeSet is applied, the process is called Check-In in Team Foundation Server. In some other version control systems Check-in is called Commit. In TFS Check-in is performed as a single atomic transaction. This means, that if for some reason Check-in is not applicable, the whole ChangeSet is rolled back. In TFS it is also possible to provide some additional data with Check-in, for example comments about changes and so on.

Branching and Merging is a process, which makes possible for different development teams to work in parallel without interfering with other teams. A branch is a set of files in different part of the repository. Every team may work on its own branch. In some point in the future, however, there will be need to combine all sets of the code into one set. This process is called merging. Branching and merging strategy adaptation must be approached with care, as there are many different strategies to conduct the process.

A version control system works as file system, that stores all the changes made to it. However it adds also an extra layer or dimension called History. The version control system can tell everything what happened to particular file or folder over time. This feature makes version control system to be useful for application development and provides capabilities to know what code actually was shipped to particular customer at any given time. It also makes it possible to get back in time and understand which kind of changes the particular file or folder has been undergone, who has worked on the file, which kind of changes he/she has done and why. Team Foundation Sever provides for all of this history functionality. In addition it also makes easy to see what changes occurred to a file before it had been renamed.

6 Developing the SecondHandStore application

The SecondHandStore application will follow a classic approach taken by any online store in the world. The application is going to be a solid and realistic and demonstrate best practices. Using the MVC pattern provides the developer with maintainable, extensible and well-structured code. An online product catalog will be created so that customers can browse by category and page. Also shopping cart and checkout option will be added, where customers can add or remove products, and enter their shipping details. Finally administration area will be created, which includes create, read, update, and delete facilities for managing the catalog.

6.1 Basic Implementation

In order to create the first version of the application, a blank solution is created in Visual Studio 2013. It is named “SecondHandStore”. A Visual Studio solution is a container for one or more projects. Two projects are added to that blank solution. First project will contain the application’s domain model and is named “SecondHandStore.Domain”. The second project will contain the application’s MVC model and is named “SecondHandStore.WebUI”. See Figure 28.

.

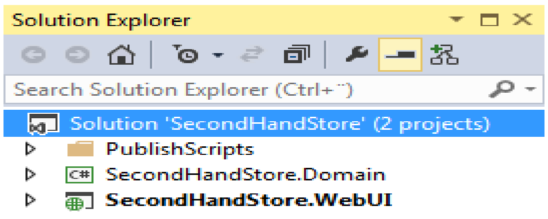


Figure 28. SecondHandStore application’s projects in Visual Studio’s Solution Explorer.

Since this is an e-commerce application, the most obvious domain entity is a product. A new folder called Entities is created in the domain project and a C# class called Product is added. See Listing 1.

```
Namespace SecondHandStore.Domain.Entities
{
      public class Product
       {
            public int ProductID { get; set; }
            public string Name { get; set; }
            public string Description { get; set; }
            public decimal Price { get; set; }
            public string Category { get; set; }
      }
}
```

Listing 1. The C# code of Product class.

Product information will be saved in the database. In order to get product information from the database, a new folder called Abstract is created and IProductsRepository is added to that folder. See Listing 2.

```
using System.Linq;
using SecondHandStore.Domain.Entities;

namespace SecondHandStore.Domain.Abstract
{
      public interface IProductRepository
       {
            IQueryable<Product> Products { get; }
      }
}
```

Listing 2. IProductsRepository interface.

A class that uses IProductsRepository interface can obtain Product objects without knowing their real residing place. For the obtaining class it does not matter, if Product objects come from the database or some other source.

The next step is to create Controller and View in SecondHandStore.WebUI project. There is already a folder in this project called Controllers, so the developer adds a new controller by selecting “Add Controller” from the right-click menu. The new Controller is named ProductController. The controller is coded to contain an IProductsRepository private member which lets the controller to retrieve all product data from the database. In addition, the ProductController contains an action method called List, which renders a View showing a complete list of products. See Listing 3 for the View code.

```
@model IEnumerable<SecondHandStore.Domain.Entities.Product>

@
{
      ViewBag.Title = "Products";
}

@foreach (var p in Model)
{
      <div class="item">
      <h3>@p.Name</h3>
      @p.Description
      <h4>@p.Price.ToString("c")</h4>
      </div>
}
```

Listing 3. The code of List view.

The next step is to create a database and populate it with data. If an SQL Server instance is installed on the development environment, it will be possible to create a database from Visual Studio’s Server Explorer. SQL Server is connected from Server explorer and the “SecondHandStore” is created. A table called “dbo.Products” is created in the “SecondHandStore” database and the data is added manually. See Listing 4 for creating a Products table.

```
CREATE TABLE Products
(
      [ProductID] INT NOT NULL PRIMARY KEY IDENTITY,
      [Name] NVARCHAR(100) NOT NULL,
      [Description] NVARCHAR(500) NOT NULL,
      [Category] NVARCHAR(50) NOT NULL,
      [Price] DECIMAL(16, 2) NOT NULL
)
```

Listing 4. Creating Products table in SecondHandStore database.

In order to connect from the ASP.NET MVC application to the database and retrieve product information transparently, an Entity Framework will be used. Entity Framework is widely used in the web application development community and it can be downloaded and installed through the NuGet Package Manager. After installing the framework, a context class, which associates the model with the database, is created. A new folder called Concrete is created in SecondHandStore.Domain project and a new class called EFDbContext is added to that folder. See Listing 5.

```
using SecondHandStore.Domain.Entities;
using System.Data.Entity;

namespace SecondHandStore.Domain.Concrete
 {
         public class EFDbContext : DbContext
         {
                 public DbSet<Product> Products { get; set; }
         }
}
```

Listing 5. Creating EFDbContext class.

At the end of development a basic infrastructure of the application is ready and the appearance of the user interface is finalized with CSS styles. CSS stands for Cascading Style Sheets. It is a style sheet language for describing the look and formatting of any document written in markup language. The main purpose of CSS is the separation of document presentation from document content.

In order to create the desired look and feel, the default layout was modified. There is a default layout file called _Layout.cshtml in Views/Shared folder of SecondHandStore.WebUI project. The contents of that file were modified according to Listing 6.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>@ViewBag.Title</title>
<link href="~/Content/Site.css" type="text/css" rel="stylesheet"
/>
</head>
<body>
<div id="header">
<div class="title">SECONDHAND STORE</div>
</div>
<div id="content">
@RenderBody()
</div>
</body>
</html>
```

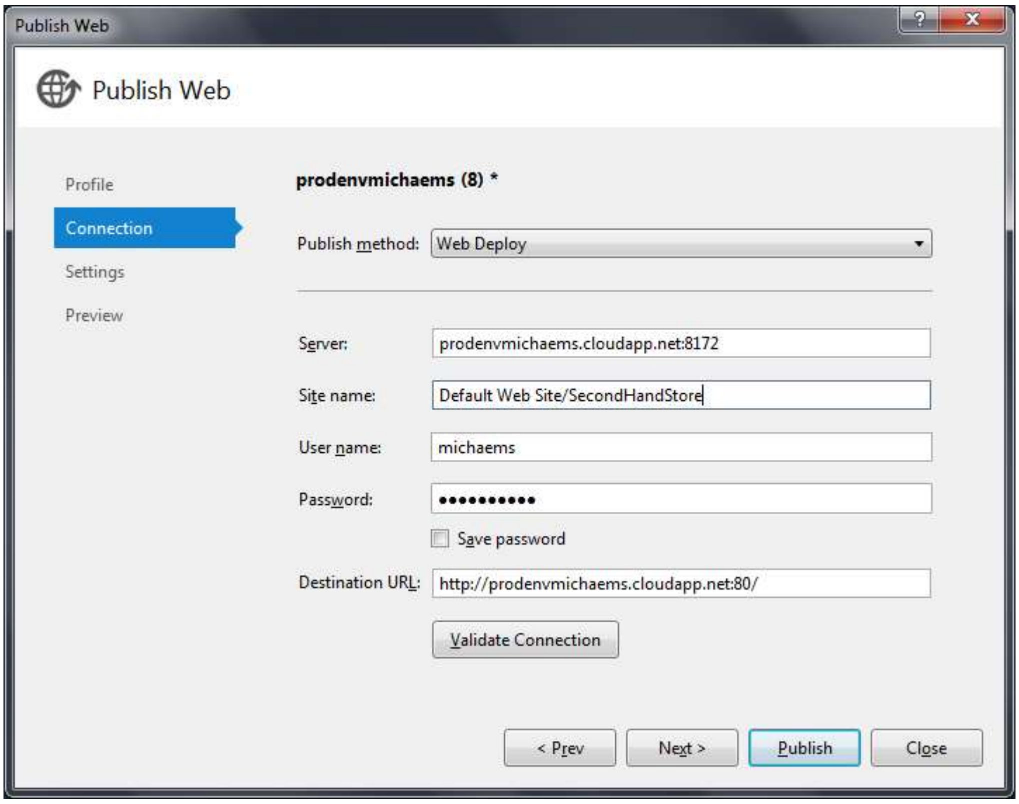
Listing 6. _Layout.cshtml contents were modified to present a desired user interface.

Visual Studio created a CSS file automatically when ASP.NET MVC project was created. The created file is called Site.css and it can be found in the Content folder of the SecondHandStore.WebUI project. As can be seen in Listing 6, _Layout.cshtml default layout file has a link to Site.css file. In order to create the desired look and feel, the CSS code was added to the end of Site.css code. Listing 7 shows a fragment of added code.

```
BODY { font-family: Cambria, Georgia, "Times New Roman"; margin:
0; }
DIV#header DIV.title, DIV.item H3, DIV.item H4, DIV.pager A {
font: bold 1em "Arial Narrow", "Franklin Gothic Medium", Arial;
}
DIV#header { background-color: #444; border-bottom: 2px solid
#111; color: White; }
DIV#header DIV.title { font-size: 2em; padding: .6em; }
DIV#content { border-left: 2px solid gray; margin-left: 9em;
padding: 1em; }
DIV#categories { float: left; width: 8em; padding: .3em; }
```

Listing 7. A fragment of CSS code added to Site.css file.

After testing the developed application, it was deployed to Microsoft Azure Production Environment. Deployment was executed through the right-click menu of Visual Studio. After pressing the Publish option, a publishing form was opened and the user entered the required data for publishing the application. See Figure 29.



Publish Web

prodenvmichaems (8) *

Publish method: Web Deploy

Server: prodenvmichaems.cloudapp.net:8172

Site name: Default Web Site/SecondHandStore

User name: michaems

Password: ●●●●●●●●●●●

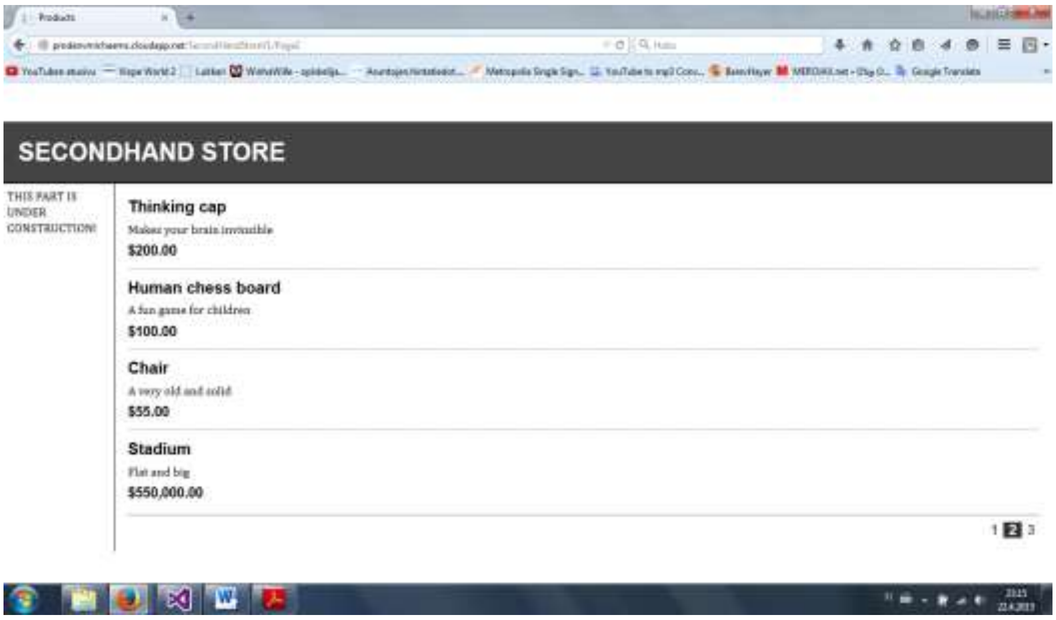
☐ Save password

Destination URL: http://prodenvmichaems.cloudapp.net:80/

Validate Connection

Figure 29. Publishing the ASP.NET MVC application to production environment.

The final application can be seen and used from any web browser in the world. See Figure 30.



SECONDHAND STORE

THIS PART IS UNDER CONSTRUCTION!

Thinking cap
Makes your brain invisible
$200.00

Human chess board
A fun game for children
$100.00

Chair
A very old and solid
$55.00

Stadium
Flat and big
$550,000.00

1 2 3

Figure 30. Final application deployed to MS Azure virtual machine.

6.2 Further Development of the Application

In chapter 6.1 the development and deployment of first version of e-commerce SecondHandStore application was presented. In the next versions the application will be developed further and will include navigation and chart, shopping cart, an option select and remove items into and from the shopping cart, enter shipping details and checking out, and an area for administrating the application. The administrator of the application will be provided with options to add and remove shopping items and execute other administrative tasks.

Development of the e-commerce application demands very controlled attention and many lines of development code. In addition it demands extensive testing. In order to gain on understanding of the development process and milestones, the reader is required to possess technical knowledge of modern development tools and processes. To keep this final year project simple and understandable, I have decided not to include further development of the application in this thesis.

7 Conclusion

The goal of this final year project was to study many modern technologies and integrate them to develop a web application in the Microsoft Azure cloud service. The task was challenging and interesting .Though I was familiar with some technologies used in this paper, the majority of them still needed in-depth studying and examination.

The project started at the end of February 2015 and was finalized at the end of April 2015. I studied all the needed technologies and integrated them for developing the application. The biggest challenge was Microsoft Azure cloud service and Windows Server environment, because I had almost non-existent experience with those technologies. Also .NET library was challenging due to its vast number of different assemblies for different kind of technologies.

During implementation of this project I learned that Microsoft Azure is a public cloud and provides many services to the companies of different sizes. Especially small and middle-size companies gain great benefits by avoiding on-premises solutions and subscribing the service and using different kind of applications and environments. Microsoft Azure is deployed onto 19 datacenters across the globe and makes it possible for companies to start with very low costs and rapidly gain customers.

Another useful skill learned was setting up a Windows Server environment in the virtual machine of the Azure cloud service. Computers with a server operating system run around the clock and provide the users across the globe to access the hosted applications. However in order to host an application, the sever computer contains an IIS server application, which is the real host and configurator of a web application.

ASP.NET MVC technology was learned and used for developing the web application. This technology is very powerful, is based on Microsoft's .NET framework and is used by the majority of the web application development community.

Altogether the project was successfully implemented and I acquired many useful skills for my future career.

References

1. Freeman A. Pro ASP.NET MVC 4. USA: New York, Apress Media, 2012.

2. Fundamentals of Azure: Microsoft Azure Essentials. USA: Microsoft Corporation, 2015.

3. Windows Server Administration Fundamentals. Microsoft Official Academic Course. USA: Hoboken, John Willey & Sons, 2011.

4. Schaefer K, Cochran J, Forsyth S. Professional Microsoft IIS 8. USA: Indianapolis, John Willy & Sons, 2013.

5. Simmons K, Carstarphen S. Pro SQL Server Administration. USA: New York, Apress Media, 2012.

6. LeBlanc P. SQL Server Step by Step. USA: California, O'Reilly Media Inc., 2013.

7. Overview of the .NET Framework [online].
   URL: https://msdn.microsoft.com/en-us/library/zw4w595w.aspx.
   Accessed: 10 April 2015.

8. Troelsen A. C# 2010 and the .NET 4 Platform. USA: New York, Apress Media, 2010.

9. Blankenship Ed, Woodward M, Holliday G, Keller B. Professional Team Foundation Server. USA: Indianapolis, John Willy & Sons, 2013.