POLITECNICO DI TORINO

Master of Science Degree
in Computer Engineering

Master of Science Thesis

Moving from local server to cloud service using Microsoft Azure

Supervisor:

prof. Gianpiero Cabodi

Candidate:

Khatereh Yamini

July 2019

Abstract

Nowadays, applications play an important role in our life. In particular, online applications are widely used through the Internet. Consequently, one of the issues that software developers are facing is to get an application online.

Application deployment is making an application available for use. In order to satisfy this purpose, a server is needed.

Cloud servers are one of the solutions which have lots of advantages comparing to physical servers. In the first place, hardware or capital expenses are not required. It is a good choice for smaller businesses that are outgrowing storage very fast. In addition, storage can be added as required. Therefore, you only pay for what you need. Finally, data can be backed up in the cloud as regularly as 15-minute intervals, minimizing data losses in disaster situations. Small data set recovery time is improved.

However, not all aspects of the cloud server are positive. There are some disadvantages as well. For example, if the Internet goes down on your side or on your cloud provider's side, you won't have access to your information.

In this thesis, we had in mind to move a project from local to the cloud using Microsoft Azure so that there would be no need for any physical server, DB, and so on, and everything would be done through web services. Cloud service is a new concept and we believe that this is the future of all servers. The reason for this decision was to get rid of all problems of getting the application online on local servers, reducing the costs and applying easily the new changes in online application.

The problem which had to be solved was how to move a running application with real data used by real users from local servers to cloud service without any down time and loss of data and functionalities.

To this end, we had to move from local servers to service. In order to use this service, we had to upgrade the project from .Net framework to .Net core. It was needed to change the project in a way that makes it possible to use Azure. We needed to upgrade the libraries to .NET Standard. In this way, we were able to continue using them from all our current projects, whether they targeted .NET Core or .NET Framework. Then we had to upgrade the projects to use frameworks that are compatible with .NET Core. Because we are using ASP.NET MVC and ASP.NET Web API, this involved migrating to ASP.NET Core. Finally, we had to upgrade the projects to use .NET Core. If there were some APIs that were unsupported, we needed to migrate to an alternative. In some cases, this was a different, newer API, which was already part of .NET Core. In other cases, we needed to use an external library that adds the missing functionality. Also, if there were some libraries that did not support .NET Core, we either needed to upgrade them, find replacements, or use them in compatibility mode.

The next step was to upgrade the database from SQL to Azure SQL database. In Azure, not all functions, synonymous or other things are permitted. If our DB was using them, we had to remove and change these parts.

After the upgrade process, some steps had to be taken. In the first place, it was essential to create a cloud service app in .NET. Then a SQL Database in Azure had to be created. Furthermore, we had to connect a .NET Core app to SQL Database. After that, we deployed the app to Azure. Sometimes it is needed to update the data model and redeploy the app. Then it is possible to get stream diagnostic logs from Azure. Finally, we could manage the app in the Azure portal.

We faced some problems during the process. In the first place, a solution had to be found for how to migrate the database without down time, which was solved by using a free server and restoring a backup of the database on that server. Furthermore, a solution had to be found for database migration. Data migration assistant was used to solve this problem. Then we had problem with lots of synonyms which were used in the databases which are not supported by Azure. This problem is solved by using external tables. Furthermore, the features and the types which are not supported by Azure had to be replaced. Most of these problems were about the migration of the database. For deploying the application in order to get it online, there was not any serious problem, which I take it as a big advantage to use cloud services to deploy application comparing with using in house servers or outsourcing.

Acknowledgements

I would like to express my deepest appreciation to my advisor Professor Gianpiero Cabodi, for his support, understanding and patience.

I would also like to thank E-gate Company: Paolo Borri, Daniele Carozzi and all my colleagues for their guidance and support during the process of this study.

Last but not least, I appreciate and thank my father for his constant help and his best suggestions, my mother for her long-distance energy and supports throughout doing this thesis, and my fiancés for providing me with unfailing support and continuous encouragement through the process of this study.

Contents

List of Tables

List of Figures

Chapter 2

Chapter 4

Chapter 5

Chapter 1: Introduction

Introduction

Nowadays, applications play an important role in our lives. In particular, online applications are widely used through the Internet. Consequently, one of the issues that software developers are facing is to launch an application online.

Application deployment is making an application available for use. For this purpose, a server is needed which requires a capital investment in hardware and infrastructure or outsourcing to an external web host. Both have costs and limitations such as capacity, space for equipment, and so on.

An In-house server needs space in your office for a rack or server room/closet, in addition to dedicated IT support. You will also have to take account of the cost of maintenance and software licenses.

Outsourcing also has some disadvantages. For instance, you do not have the same level of control over outsourced hosting support in contrast to in-house staff. The process of hiring and firing is not controlled by you, at all.

A remote connection is needed to connect to the server which is slow and time-consuming. Every single change needs to be published again which takes lots of time. Any changes in the database need a synchronization. These processes not only are time-consuming but also leads to human errors. Very often the application goes down because updating a small part is forgotten during the update process. This imposes some costs on the owner company, depending on which kind of application it is.

Bugs in software are unavoidable. Even after lots of testing and fixing, some of the bugs will be found while the application is online, with the real data and real users. There is no way but to upload the application and let the users use an old version until the final tests with online applications are done.

Another solution is using cloud servers. A cloud server has lots of advantages. In the first place, hardware or capital expenses are not required. It is well-suited to smaller companies which will outgrow storage too quickly. Furthermore, storage can be added as needed. Applications and solutions are often developed on-demand, so you only pay for what you need. Finally, the data can be backed up in the cloud as regularly as 15-minute intervals, minimizing data losses in disaster situations. Small data set recovery time is improved.

However, not all aspects of the cloud server are positive. There are some disadvantages as well. For example, if the Internet goes down on your side or on your cloud provider's side, you will not have access to your information.

What this thesis concerns is moving a project from local to the cloud using Microsoft Azure. We believe that this is the future of all servers.

A car-pooling project using C# and MVC has been done before. For this project, there is a mobile and a web application both of which have been done and are being used online. There is a web API project which both applications are using through web services. This web API is using a database and is connected to SQL Server. There is also another project for administrating and management of car-pooling application's data, users and payments.

In this thesis, we have in mind to move the project from local to the cloud using Microsoft Azure so that there would be no need for any physical server, DB, and so on, and everything would be done through web services. To this end, we should move from local servers to service.

In order to use this service, we have to upgrade the project from .Net framework to .Net core. It is needed to change the project in a way that makes it possible to use Azure. For example, it is not possible to use custom libraries anymore. Instead, we must use services. It is needed to upgrade the libraries to .NET Standard. In this way, we will be able to continue using them from all our current projects, whether they target .NET Core or .NET Framework. Then we must upgrade the projects to use frameworks that are compatible with .NET Core. Because we are using ASP.NET MVC and ASP.NET Web API, this involves migrating to ASP.NET Core. Finally, we should upgrade the projects to use .NET Core. If there are some APIs that are unsupported, we will need to migrate to an alternative. In some cases, this can be a different, newer API, that is already a part of .NET Core. In other cases, we will need to use an external library that adds the missing functionality. Also, if there are some libraries that do not support .NET Core, we will either need to upgrade them, find replacements, or use them in compatibility mode.

The next step is to upgrade the database from SQL to Azure SQL database. In Azure, not all functions, synonymous or other things are permitted. If our DB is using them, we have to remove and change these parts.

After the upgrade process, some steps must be taken. In the first place, it is essential to create a cloud service app in .NET. Then a SQL Database in Azure should be created. Furthermore, we must connect the .NET Core app to SQL Database. After that, we must deploy the app to Azure. Sometimes it is needed to update the data model and redeploy the app. Then it is possible to get stream diagnostic logs from Azure. Finally, we can manage the app in the Azure portal.

The thesis is organized as follows: Chapter Two reviews the concepts and definitions about cloud computing, cloud service providers and Microsoft Azure as a cloud provider. In Chapter Three the details about the project methodology is discussed. Chapter Four explains the project Implementation, the problems we faced with and the solutions we found for them. Finally, the thesis is concluded in Chapter Five, which provides a summary of the study.

Chapter 2: Background Information

2.1 Cloud computing

There are many different definitions of cloud computing. Simply we can say that it is the use of assorted services, like software development platforms, servers, storage and software, over the net, often referred to as the "cloud". Term of anything that involves delivering hosted service over the web, in general, is Cloud computing.

Cloud computing is the outsourcing of computer programs. Users can access software and applications from where they are by using cloud computing. The applications are being hosted by a third party and reside within the cloud. This suggests that users should not worry about things like storage and power, and they can easily use the end result.

> There seems to be no consensus on what a Cloud is. We add yet another definition to the already saturated list of definitions for Cloud Computing:
>
> A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet. [1]

Instead of keeping files on a hard drive or local storage device, cloud-based storage allows us to save them on a remote database. As long as you have access to the web by an electronic device, you have access to the data and software programs to run it.

Some regard cloud computing as an overused buzzword that has been blown out of proportion at large software firms by marketing departments [2]. A typical criticism is that cloud computing cannot succeed because it means organizations should lose control over their data, such as an email provider that stores data at multiple locations around the world. A large regulated company, like a bank, may need to store data within the U.S. While this is not an

insurmountable issue, it shows the kind of issue some companies may have with cloud computing.

The name cloud is taken from the cloud symbol which is usually used to show the Internet in flowcharts and diagrams. In addition, it is named Cloud computing because the information being accessed is found remotely within the cloud or a virtual space. Companies that give cloud services allow users to store files and applications on remote servers and so access all the info via the net. This suggests the user does not need to be in an exceedingly specific place to achieve access to that, permitting the user to figure remotely.

Traditional business applications are very complicated and high-priced. The quantity and type of hardware and software needed to run them are intimidating. A full team of experts to install, configure, test, run, secure, and update them is required. When you multiply this effort across hundreds of apps, it's straightforward to work out why the largest companies with the most effective IT departments are not obtaining the apps they need. Small businesses do not stand a chance.

Cloud computing proponents point out that it is a new paradigm in software development, where smaller organizations have access to processing power, storage and business processes that once were only available to large companies.

Cloud-based apps will be up and running in days or weeks with less price. With a cloud app, simply open a browser, log in, customise the app, and begin using it.

Three distinct characteristics of cloud service differentiate it from traditional web hosting. It is sold on demand, usually by the minute or the hour; Users can have as much or as very little of a service as they require at any given time, that means it is elastic; and also the service is absolutely managed by the supplier (the client needs only a pc and Internet access). Important

innovations in virtualization and distributed computing, additionally as improved access to the high-speed web, have accelerated interest in cloud computing.

According to “techopedia” [2], there are three cloud computing characteristics that are common among all cloud-computing vendors:

- A cloud vendor manages completely the application’s back-end (especially hardware).
- Users pay only for services they use (memory, processing time and bandwidth, etc.).
- Services are scalable

Cloud computing provides an easy way to use different services over the Internet. These services can be servers, storages, databases, applications and more. In other words, instead of using a local server or a personal computer, it is possible to use a network of remote servers hosted on the Internet to store, manage, and process data. A cloud services platform owns and maintains the network-connected hardware needed for these services while providing and using what you need through a web application. These services are divided into 3 categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

While an application which shares photos to lots of users is running, or a critical operation is executing, a cloud service provides fast access to low cost and flexible resources. With cloud computing large investment in hardware is not needed. However, it is possible to provision exactly the needed size and type of computing resources. You pay only for cloud services you use, although it is practicable to access to as many resources as you need. In this way, it helps lower your operating costs, runs your infrastructure more efficiently and scales as your business needs change.

Cloud computing takes all the work concerned in crunching and process data far from the device you carry around or sit and work at. It additionally moves all that work to very large computer clusters far away in cyberspace. The Internet becomes the cloud, and your data, work, and applications are offered from any device with that you'll be able to hook up with the Internet, anyplace within the world.

Cloud computing can be both public and private. Public cloud services offer their services over the Internet for a fee to anyone on the web. Amazon Web Services is currently the largest provider of public cloud services [3]. Private cloud services, on the other hand, only offer services to a specific variety of people. These services are a system of networks that provide hosted services. There is another possibility that combines elements of both the public and private services which are called the hybrid cloud.

Regardless of the type of service, cloud computing services offer users with a series of functions such as storage, backup, and knowledge retrieval, Email, creating and testing apps, data analysing, audio and video streaming, delivering software system on demand.

Although cloud computing is a new service it is being used by a variety of various organizations from huge corporations to tiny businesses, nonprofits to government agencies, and even individual consumers.

2.1.1 Pros and cons

The significant advances in science and technology have made it possible to use Cloud computing in IT resource field. Here are some benefits of using cloud computing services:

Cost

Cloud computing reduces costs. It eliminates a large amount of expense of buying equipment and lots of other costs, such as hardware, software, the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure and so on.

Rather than having to invest heavily in data centres and servers before you know how to use them, you can only pay when you use computing resources and solely pay for how much you use them.

Usage from hundreds of thousands of customers are aggregative within the cloud, suppliers can do higher economies of scale, resulting in lower pay as you go costs.

Capacity

Eliminate guess on your infrastructure capability desires. You often either end up sitting on overpriced idle resources or managing limited capacity when you make a capacity decision before deploying an application. These issues go away with cloud computing. With just a few minutes' notice, you can access as much or as little as you wish and scale up and down as needed.

Speed

Considering that most cloud computing services are provided self-service and on demand, we can easily predict that cloud computing can increase the speed. Therefore, huge amounts of computing resources can be provisioned in a glance, which in turn leads to a lot of flexibility and taking the pressure off capacity planning.

New IT resources are only one click away in a cloud computing environment, which results in reducing the time it takes to make those resources offered to your developers from weeks to minutes. This leads to a dramatic increase in agility for the organization due to the considerably lower cost and time it takes to experiment and develop.

Global scale

Cloud computing can scale elastically. In other words, delivering the right amount of IT resources is possible right when it is needed, and from the right geographic location.

Productivity

Cloud computing eliminates many time-consuming tasks which are required in On-site data centres. These tasks include a lot of "racking and stacking"—hardware set up, software patching, and other time-consuming IT management chores. Since there is no need for these tasks, IT teams can spend time on achieving more important business goals. Focus on projects that make your business different, not on infrastructure. Cloud computing allows you to focus not on heavy racking, stacking and powering servers, but on your own customers.

Performance

The biggest cloud computing services run on a worldwide network of secure data centres, that are frequently upgraded to the most recent generation of quick and economical computing hardware. This offers many edges over one company data centre, as well as reduced network latency for applications and larger economies of scale.

With simply some clicks, you can easily deploy your application in multiple regions around the world. This results in providing your customers with lower latency and better experience at a minimal cost.

Security

Cloud computing increases and strengthens security. Because many cloud providers offer a set of policies, technologies and controls that satisfy this issue. Consequently, it helps to protect data, apps, and infrastructure from potential threats.

Disadvantages of Cloud Computing

However, not all aspects of the cloud server are positive. There are also some disadvantages. For example, if the Internet goes down on your side or on your cloud provider's side, you will not have access to your information.

When it involves in particular sensitive medical records and financial information, security has always been a major concern with the cloud. While laws force cloud computing services to strengthen their compliance and security measures, it remains an in-progress issue. Encryption protects important information. However, the data will disappear if that encryption key is lost.

Servers maintained by cloud computing companies may also be victims of natural disasters, internal bugs, and power failures.

There is a learning curve for both managers and employees, as with any technology. However, with many people accessing and manipulating information through a single portal, accidental errors can be transferred across a system as a whole.

2.1.2 Types of services

There are various models and deployment strategies to access the technology service. However, 3 types of cloud computing are the most popular ones. Choosing the right type of cloud computing for your needs can help you achieve the correct control balance and avoid undifferentiated heavy lifting. Here are the definition and benefits of these types:

Platform as a service (PaaS)

PaaS is a branch of cloud computing which lets the applications to be developed, run and managed without having to urge wedged in code, storage, infrastructure and so on.

PaaS eliminates the necessity for organizations to manage the underlying infrastructure (usually hardware and OS) and permit you to focus on application deployment and management. Since you do not have to concern resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated work concerning running an application, you will be more efficient.

PaaS is taking into account the foremost complicated of the 3 layers of cloud-based computing. PaaS shares some similarities with SaaS, the main difference being that it is really a platform for creating software delivered over the Internet, rather than delivering online software. There are platforms such as Force.com and Heroku in this model.

Platform as a service provides a cloud-based environment with everything required to support the complete lifecycle of building and delivering web-based (cloud) applications — all without the cost and complexity of buying and managing the underlying hardware, software, provisioning and hosting.

Benefits of PaaS:

- Develop application and get to market faster
- Deploy new web applications to the cloud in minutes
- Reduce complexity with middleware as a service [4]

There are many varieties of PaaS. Each PaaS option is either public, private, or a hybrid mixture of the two. Public PaaS is hosted within the cloud, and its infrastructure is managed by the supplier. Private PaaS, on the opposite, is housed in onsite servers or private networks and is maintained by the user. Hybrid PaaS uses parts from each public and private and is able to execute applications from multiple cloud infrastructures.

Infrastructure as a service (IaaS)

A third-party hosts infrastructure element like hardware, software, servers, and storage, as well as providing backups, security, and maintenance. IaaS contains the basic building blocks for cloud IT which give access to networking features, computers and data storage space. IaaS offers the very best level of flexibility and management control over IT resources and is very similar to existing IT resources that are familiar to many IT departments and programmers these days.

IaaS includes a technique to deliver everything from operating systems to servers and storage through IP-based connectivity as part of an on-demand service. Clients can avoid the need to buy software or servers and provide these resources in an outsourced on-demand service instead. IBM Cloud and Microsoft Azure are popular examples of the IaaS system.

Infrastructure as a service provides companies with computing resources — including servers, networking, storage and data-center space — on a pay-per-use basis.

Benefits of IaaS:

- No need to invest in your own hardware

- Infrastructure scales on demand to support dynamic workloads
- Flexible, innovative services available on demand [4]

Software as a service (SaaS)

SaaS provides a completed product that the service provider runs and manages. In most cases, by software as a Service, people are referring to end-user applications. You do not have to think about how to maintain the service or how to manage the underlying infrastructure with a SaaS offering; you only have to consider however you may use that exact piece of software.

A common example of a SaaS application is a web-based email that allows you to send and receive emails whiteout having to manage feature additions to the email product or maintaining the servers and operating systems that run the email program.

SaaS includes software application's licenses to customers. Typically, licenses are provided via a pay-as-you-go model or on-demand model. You can find this kind of system in Microsoft Office 365.

Cloud-based applications — or software as a service — run on distant computers “in the cloud” that are owned and operated by others and that connect to users’ computers via the internet and (usually) a web browser.

Benefits of SaaS:

- You can sign up and rapidly start using innovative business apps
- Apps and data are accessible from any connected computer
- No data is lost if your computer breaks because the data is in the cloud
- The service is able to dynamically scale to usage needs [4]

2.1.3 Deployment options

There are different ways to deploy cloud resources. Deployment options can be categorized into 3 major types: Private, public and hybrid.

Private cloud

A private cloud also known as enterprise or internal cloud consists of computing resources used completely by one business or organization where a firewall is protecting all of the data. The private cloud can either be physically situated at your organization's on-site data centre or a third-party service supplier can host it. Private cloud technology is not sold to external customers as a service; rather, it is like gaining the advantages of the cloud system, whereas truly not leaving behind the management of maintaining your own data centre. In a private cloud, however, the services and infrastructure are invariably maintained on a secure private network which is not shared with other customers and also the hardware and software are dedicated entirely to your organization. Therefore, a private cloud will make it easier for a corporation to customise its resources to satisfy specific IT needs. Private cloud is customizable to satisfy the distinctive business and security desires of the organization. With bigger visibility and management into the infrastructure, organizations will operate compliance-sensitive IT workloads while not compromising on the safety and performance previously solely achieved with dedicated on-premise data centres.

Private cloud services are delivered to internal users from a business data centre. This model provides the cloud's flexibility and convenience while preserving local data centre management, control and security. Internal users may or may not be charged by IT chargeback for services.

Private clouds are the obvious choice when the main concern of a business is security. In cases, where the most important assets are data and applications and you wish to have control over them at the same time, private cloud can be mentioned. Another condition which is suitable for using private clouds is Having an oversized company that may effectively run a next generation cloud data centre and on its own as well. For companies that already have expensive data centres because they can use their existing infrastructure, this can be a good solution. Private clouds are typically utilized by government agencies, financial institutions, the other mid-size to large-scale organizations with business-critical operations seeking increased management over their setting.

However, not all aspects of private clouds are positive. There is some limitation in using them. In the first place, it is very expensive. Although, these clouds provide most security and control, at the same time the company must also purchase and maintain all software and infrastructure, thereby reducing the price savings. Furthermore, since there are high security measures, mobile users may have limited access to the cloud. The main disadvantage of private cloud is that it is the company's responsibility to maintain, manage and updating all data centres. Over time, your servers are expected to have to be replaced, which can be very expensive. On the opposite hand, private clouds provide an increased level of security and that they share only a few if any, resources with alternative organizations.

Private clouds' advantage:

- More flexibility: the organization will customize its cloud environment to satisfy specific business desires.
- Improved security: resources are not shared with others; thus, higher levels of control and security are attainable.

- High scalability: the scalability and potency of a public cloud are still available in private clouds.
- Compliance: Compliance with strict regulations as organizations can run protocols, configurations and measures in order to customize security based on workload needs.

AWS, VMware and OpenStack are common private cloud technologies and providers.

Public cloud

The most common method to deploy cloud computing is through public clouds. Public Cloud refers to the model of cloud computing that delivers IT services over the Internet. The service may be free, freemium, or a subscription-based offer charged on the basis of consumed computing resources. A third-party cloud service provider owns and operates the cloud resources (such as servers and storage) and is delivered over the Internet. With a public cloud, the cloud provider owns and manages all hardware, software, and other supporting infrastructure. The cloud vendor is responsible for developing, managing and maintaining the pool of shared computing resources across the network among multiple tenants. In a public cloud, you share with different organizations or cloud "tenants" the identical hardware, storage and network devices. You use a web browser to access services and manage your account.

The computing functionality can vary from common services like email, apps, and storage to the enterprise-grade operating system platform or software development and testing infrastructure environments. Public cloud deployments are often used to deliver web-based email, online office applications, storage, and environments for testing and development.

High elasticity and scalability for IT-enabled services delivered at a low-cost subscription-based price tier are the defining features of a public cloud solution. As the most popular cloud

computing services model, the public cloud offers huge solutions and computing resources to meet the growing requirements of organizations of all sizes and verticals.

Public cloud services are sold on demand, usually by minute or hour, although there are long-term commitments for many services. Customers are charged only for the CPU cycles, storage or bandwidth that they consume.

The main difference between private and public clouds is that you are not responsible for managing a public cloud hosting solution. Your data is keeping in the data centre of the provider and therefore the data centre is managed and maintained by the provider. Many companies appeal to this type of cloud environment because it leads to reducing lead times in testing and deploying new products. The drawback, however, is that with a public cloud, many companies feel security might be lacking. Although you do not manage a public cloud's security, all of your data remains distinct from others, and public cloud security breaches are rare.

In order to examine what is the best choice between public and private cloud, it should be noted that everything leads to control. A large company may select a private cloud, while a smaller company may select a public cloud.

Public clouds are a suitable solution when there are predictable computing requirements for a specific number of users, such as communication services. In addition, they are a good choice for apps and services needed to carry out business and IT operations. Furthermore, public clouds can be used for additional requirements for resources to meet different peak demands. Finally, they are the best choice for software development and test environments.

There are some limitations as well in using public clouds. For instance, for large-scale use, the total cost of ownership (TCO) can rise exponentially, specifically for mid-size to large companies.

Public clouds' advantages:

- Lower costs: You do not need to invest in buying hardware or software and only pay for the service you're using.
- No maintenance: Since the service provider offers the maintenance, complexity and requirements on IT expertise will be reduced.
- Near-unlimited scalability: Resources are available on demand to satisfy your business requirements, which results in high scalability and flexibility.
- High reliability: A large network of servers guarantees against failure.

Microsoft Azure, AWS, Google Cloud Platform, IBM's Blue Cloud and Sun Cloud are public cloud examples.

Hybrid cloud

Hybrid clouds often referred to as "the best of both worlds" mix on-premises infrastructure or private clouds with public clouds to enable organizations to reap the benefits of each. For greater flexibility and more deployment options, data and applications can move between private and public clouds in a hybrid cloud. For example, for sensitive, business-critical operations such as financial reporting, you can use the private cloud (or other on-site infrastructure) and the public cloud for high-volume, lower-security desires like web-based email. The environment itself is smoothly integrated to grantee maximum performance and scalability to changing business requirements. A hybrid cloud's goal is to create a unified,

automated, scalable environment that benefits from everything that a public cloud infrastructure can deliver while maintaining control over mission-critical data.

"Cloud bursting" is available too in this kind of clouds. This is when an application or resource runs in the private cloud until a spike in demand (like seasonal events such as online shopping or tax filing) occurs, at which point the organization can "burst through" to the public cloud to tap into extra computing resources.

Regarding the hybrid cloud limitations, it should be noted that It can be costly. In addition, strong compatibility and integration between cloud infrastructure traversing different locations and categories is required. Finally, infrastructure's complexity will rise up when organizations operate and manage an evolving mix of private and public cloud architecture.

Hybrid clouds' advantages:

- Control: The private infrastructure of sensitive resources can be maintained by the organization.
- Flexibility: Whenever you need them, you can enjoy the benefits of additional public cloud resources.
- Cost-effectiveness: You only pay for additional computing power if needed with the ability to expand to the public cloud.
- Ease: It doesn't need to be overwhelming to transition to the cloud since you can gradually migrate: workload phasing over time.

2.1.4 Business cloud computing's world

Cloud computing can be used by businesses in many ways. Some users keep all apps and data in the cloud, while others use a hybrid model, keeping some apps and data on private cloud servers and others on the cloud.

The big players in the providing services include:

- Google Cloud
- Amazon Web Services (AWS)
- Microsoft Azure
- IBM Cloud
- Aliyun

Amazon Web Services is 100% public and includes an outsourced model of the pay-as-you-go. You can sign up for apps and additional services once you're on the platform. Microsoft Azure permits customers to keep certain data on their own sites. In the meantime, Aliyun is an Alibaba Group subsidiary.

2.2 Microsoft Azure

Microsoft's public cloud computing platform is Azure. Microsoft Azure offers the best range of services and advantages such as computing, data analytics, high-quality storage and efficient networking. Azure users have the absolute freedom to choose and use the services they require to satisfy their edges. Users can select from these services to develop and scale new applications within the public cloud or run existing applications. Azure is a public based cloud computing website that everybody can utilize for their own purposes. With Azure, your business or

organization has the liberty to use their favoured tools and frameworks to build, manage and deploy applications on a huge, global network.

The Microsoft Azure website offers a directory of many completely different services that you can use, such as full virtual machines, databases, file storage, backups and web or mobile app's services. Both Service type, Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are offered by Microsoft Azure. Its creation was in October 2008 and the official release in February 2010. Previously this service was named "Windows Azure". But it is transformed into "Microsoft Azure" because it can handle more than simply Windows. You can run virtual machines for Windows or Linux on Azure.

2.2.1 Azure history [5]

Microsoft first announced its plans to launch the Windows Azure cloud computing service in 2008. The service's demo versions became available and developed, leading to its commercial launch at the beginning of 2010. Even though early iterations of Azure cloud services fell behind more known offerings like AWS, the business plan continued to develop and support a wider base of programming languages, frameworks and operating systems (including Linux). By early 2014, Microsoft acknowledged that cloud computing's implications extended far beyond Windows, rebranding the service as Microsoft Azure.

While internal development has been an essential part of Azure service maturity, Microsoft additionally focuses on mergers and acquisitions to make out its cloud portfolio. For instance, Microsoft developed Cloudyn in 2017 to increase the cost management capabilities of Azure

and gained Cycle Computing in the same year to bolster Azure's support for high-performance computing and big data tasks.

In early 2018, Microsoft developed Avere Systems to create out Azure's capabilities in high-performance storage with Network File Systems (NFSes) and Server Message Block (SMB) file-based storage for Linux and Windows systems.

2.2.2 Advantages

spinning virtual machines up in minutes, providing scalable and durable cloud storage, and backup and recovery solutions for businesses of all sizes are known as cloud's advantages.

All cloud solutions are not built in the same way. Here are four reasons why Microsoft Azure is chosen by more companies.

Security

Microsoft offers some of the market's foremost advanced security technology to ensure that your data is protected securely. Microsoft has taken significant actions to ensure high security levels within its cloud environment. With tools such as Threat Intelligence, Advanced Threat Analytics, Azure Information Protection, and Multi-Factor Authorization, the Azure environment is protected. These technologies enable you to examine threats in real time, identify and react to the network's suspicious user and device activity, and provide the highest level of security of access. Microsoft optimizes its years of experience in enterprise software to develop and continually strengthen security-aware software development, operational management, and threat mitigation practices that are important to secure your cloud environment.

Privacy

Azure makes it possible to own and control the collection, use, and distribution of your customer data. Microsoft offers detailed information about how your data will be handled. You know how your data will be managed, where it is located, who has access and under what conditions. Microsoft will not report to any government or law enforcement agency your customer data hosted on Azure unless as directed or required by law. Just a small fraction of a percentage of customer data has ever been subject to a request associated with criminal law or national security, from the government.

High Availability and Scalability

Unlike several other cloud service providers, due to Microsoft's massive global footprint, Azure provides high availability and redundancy. Azure can offer service level agreements with data centres located in all parts of the world guaranteeing 99.95 percent up time. That leads to less than five hours of downtime each year. Azure can scale up or down, as most cloud services do, to satisfy your business needs. With an easy mouse click, Azure makes it easy to change the computing power level as required.

Cost Effectiveness

Azure provides a pay-as-you-go payment plan that enables companies to have better control over their IT budgets as they only buy what they need. Azure's use of SaaS applications also lowers infrastructure, maintenance, and IT environment management costs.

2.2.3 Azure competition

Microsoft Azure is one of the many major providers of public cloud services operating on a global scale. Google Cloud Platform (GCP), Amazon Web Services (AWS) and IBM are other major providers.

Microsoft Azure is very much similar and works equally to Amazon Web Services (AWS) and also the Google Cloud Platform (GCP) and is in competition with Amazon and Google. Amazon Web Services is the leader within the field — ahead of Microsoft and Google.

As of now, there is an absence of standardization among cloud services or capabilities, which means no two cloud providers offer the identical service exactly in the same way, utilizing similar APIs or integrations. This makes it hard for a business to utilize over one public cloud provider to fulfil a multi-cloud strategy, although some of these challenges may be reduced by third party cloud management tools.

2.2.4 Statistics

Here are some statistics about Microsoft Azure:

Microsoft Azure has more than 600 user-friendly services [6]. According to "DYNTEK" in 2017 nearly two-thirds of Fortune 500 companies were using Microsoft Azure for their cloud computing solution and every day an average of 1,000 customers sign up [7]. In 2019, the number of companies that use Microsoft Azure has arrived at 996,990 [8].

A list of the top companies using Microsoft Azure is shown in Table 2-1 [8].

Table 2-1: Top companies using Azure

Company	Website	Country	Revenue	Company Size
MOSAIC TECHNOLOGIES GROUP	mosaicsgroup.com	United States	10M-50M	200-500
Oomf	oomf-recruit.com	Australia	1M-10M	10-50

SCREEN PILOT LLC	screenpilot.com	United States	1M-10M	10-50
PRIME TEAM PARTNERS, INC	primeteampartners.com	United States	1M-10M	10-50
Staffing Network LLC	staffingnetwork.com	United States	10M-50M	50-200

Monitoring the market share using the best scanning techniques combined with advanced data science is showing that in the Cloud Platforms & Services category, Microsoft Azure has a market share of about 28.2%. Other major and competing products in this category are shown in Figure 2-1 [8].

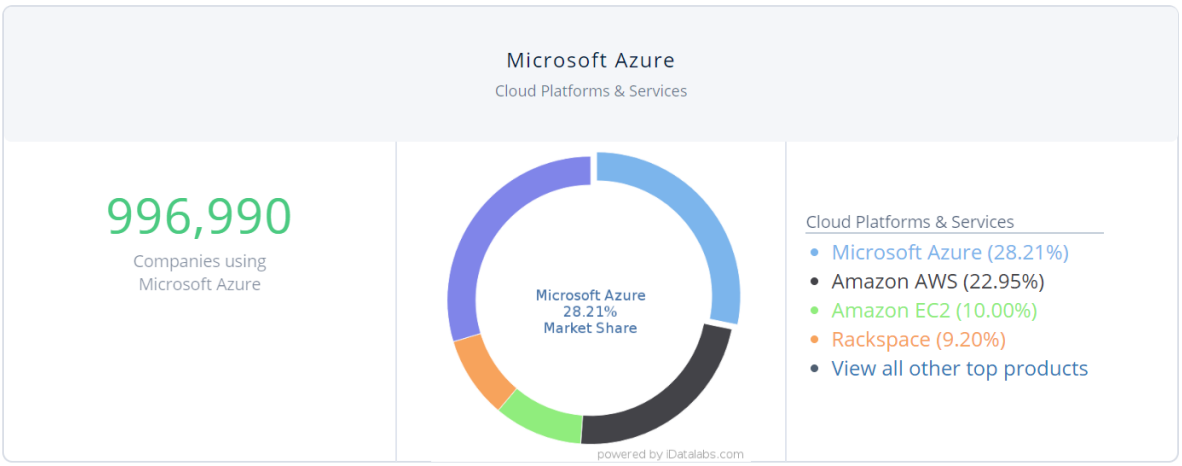

Figure 2 - 1 Market share in the Cloud Platforms & Services category

In terms of Microsoft Azure customers' number, 43% are in the United States and 10% are in the United Kingdom. The countries that use Microsoft Azure, are shown in Figure 2-2 [7].

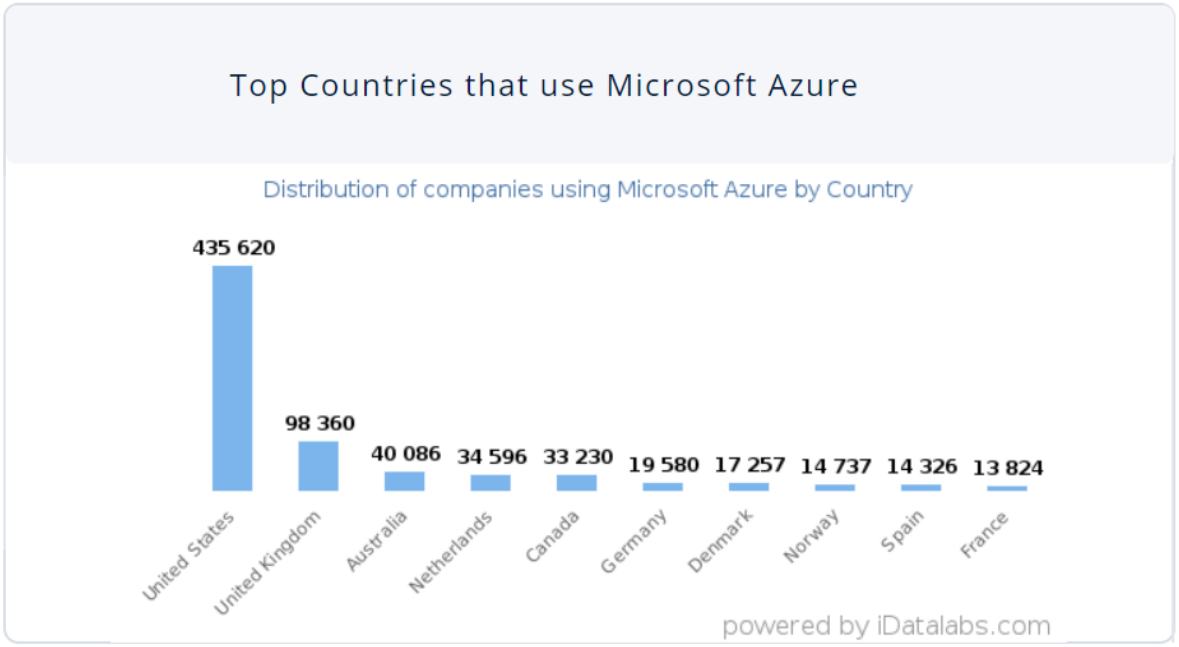

Figure 2 - 2 Top countries using Microsoft Azure

The number of companies by the country that use Microsoft Azure are shown in Table 2-2 [8].

Table 2 - 2: Companies that use Microsoft Azure, by country

Country	Number of companies
United States	435620
United Kingdom	98360
Australia	40086
Netherlands	34596
Canada	33230
Germany	19580
Denmark	17257
Norway	14737
Spain	14326
France	13824

2.2.5 Deployment models

The "classic" deployment model and the Azure Resource Manager are two deployment models that Microsoft Azure provides for cloud resource. Every Azure resource (virtual machine, SQL database, etc.) was individually managed in the classic model. The Azure Resource Manager, launched in 2014, allows users to create groups of related services in order to deploy, manage and carefully monitor linked resources together.

2.2.6 Azure products and services

Since July 2018 Microsoft divided Azur's services to 18 main categories [5]:

1. ***Compute:*** These services allow a user to deploy and manage virtual machines (VMs), containers and batch processing, and support remote access to applications.
2. ***Web:*** These services support web application development and deployment, and also provide search, content delivery, application programming interface (API) management, notification, and reporting features.
3. ***Data storage:*** This service category offers flexible cloud storage for structured and unstructured data as well as supporting large data projects, persistent storage (for containers) and deposit storage.
4. ***Analytics:*** These services offer many features such as distributed analytics and storage, real-time analytics, big data analytics, data lakes, machine learning, business intelligence (BI), IOT data streams and data warehousing.
5. ***Networking:*** This category contains virtual networks, dedicated connections and gateways, traffic management and diagnostics services, load balancing, DNS hosting and network protection from distributed denial-of-service (DDoS) attacks.

6. ***Media and content delivery network (CDN):*** This category contains streaming on-demand, digital rights protection, encoding and media playback and indexing.
7. ***Hybrid integration:*** For server backup, site recovery and connecting private and public clouds these services can be used.
8. ***Identity and access management (IAM):*** These offers guarantee that only authorized users can access Azure services and facilitate protecting the cloud's encryption keys and other sensitive information. Services include Azure Active Directory support and Multifactor Authentication (MFA).
9. ***Internet of things:*** These services allows users to capture, monitor and analyze IOT data from sensors and other devices. Services provide notifications, analytics, monitoring and coding and execution support.
10. ***Development:*** These services help developers of applications share code, test applications, and track future problems. Azure supports a variety of programming languages, such as JavaScript, Python, .NET and Node.js. Visual Studio support, software development kits (SDKs) and blockchain are also included in this category tools.
11. ***Security:*** These products are capable of identifying and responding to threats to cloud security, as well as managing encryption keys and other sensitive assets.
12. ***Artificial intelligence (AI) and machine learning:*** This is a good vary of services that a developer will use to infuse machine learning, AI and cognitive computing capabilities into applications and data sets.
13. ***Containers:*** These services help a company create, register, orchestrate and manage large volumes of Azure cloud containers using common platforms like Docker and Kubernetes.

14. ***Databases:*** This category includes Database as a Service (DBaaS) offerings for SQL and NoSQL, as well as other database instances like Azure Cosmos DB and PostgreSQL Azure Database. It additionally includes options for SQL Data Warehouse support, caching, and hybrid database integration and migration.
15. ***DevOps:*** This cluster offers project and collaboration tools that facilitate DevOps software development processes, such as Visual Studio Team Services. It also provides features for application diagnostics, DevOps tools integration, and test labs for test building and experimentation.
16. ***Migration:*** This tool package helps an organization estimate the cost of workload migration and perform real workload migration from local data centers to the Azure cloud.
17. ***Mobile:*** These products allow a developer to build cloud mobile application, provide notification services, support back-end tasks, build APIs tools, and the ability to couple geospatial (location) context with data.
18. ***Management:*** These services offer a range of tools for backup, recovery, compliance, automation, scheduling, and monitoring that can assist a cloud administrator manage an Azure deployment.

2.2.7 backup and disaster recovery (DR)

Due to its flexibility, advanced site recovery and built-in integration, Azure is a backup and disaster recovery dream tool. Azure is inherently flexible as a cloud-based solution – it can back up the data in about any language, on any operating system, and from anywhere. Additionally, you can define your backup schedule's frequency and extent (daily, weekly, monthly, etc.).

Tape backup incorporates time and place, but as a stand-alone backup and disaster recovery solution, it has limited abilities. Azure site recovery can improve your tape backup with offsite replication, reduced on-site maintenance, up to 99 years of data retention, marginal or no investment in capital, and low operating costs. Azure backup places three copies of your data in the data centre at three different locations, then three copies in a remote Azure data centre, therefore you never have to worry regarding data loss.

Azure's built-in integration for the extra backup will be a fast and painless solution if you are in a Windows virtual environment. Azure site recovery integrates with System Centre and HyperV architectures, making a strong and seamless Azure, System Centre, and HyperV cohesion.

Some companies use Azure for data backup and disaster recovery (DR) in the same way as they can with other public cloud platforms. Furthermore, some organizations are using Azure as an option to their own data centre. These companies choose to run some or all of their business applications in Azure instead of investing in local servers and storage.

Microsoft has situated Azure data centres around the world to guarantee availability. Microsoft Azure services are available in 54 regions, spread over 140 countries, as of July 2018 [5]. Since not all services are available in all localities, Azure clients must guarantee that all prevailing compliance requirements or other legislation are met by workload and data storage locations.

2.2.8 Host and Develop Web and Mobile Apps

Either you are searching for a platform to host, develop, or manage a web or mobile app, Azure makes these apps functional and adaptive with patch management, AutoScale and integration for an on-premise app.

It is possible to spend less time on your infrastructure's management and focus on developing the apps with automatic patch management for your virtual machines. Azure additionally comes with ongoing deployment support, enabling you to streamline continuous updates of code.

AutoScale is a feature developed into Azure Web Apps that automatically adjusts your resources based on customer web traffic therefore when traffic is high the resources you require are available and save money when you're not in peak times.

There is the possibility to connect a web app to an on-premise app seamlessly via Azure. Connecting apps at both locations allows both employees and partners to access resources securely within your firewall — resources which would otherwise be hard to access from outside.

2.2.9 How to use Azure

Microsoft Azure can be used by anybody. Simply by using the website of Azure, you will be able to sign up for a new account. Every account has a $ 200 credit that you simply will use over the first 30 days, so you can get started and try it in order to realize how Azure works for you. For the first year, a number of free services are offered, such as access to Linux virtual machines, Windows virtual machines, file storage, databases, and bandwidth.

Using Azure's pricing calculator, you can also get an idea of the cost (Figure 2-3).

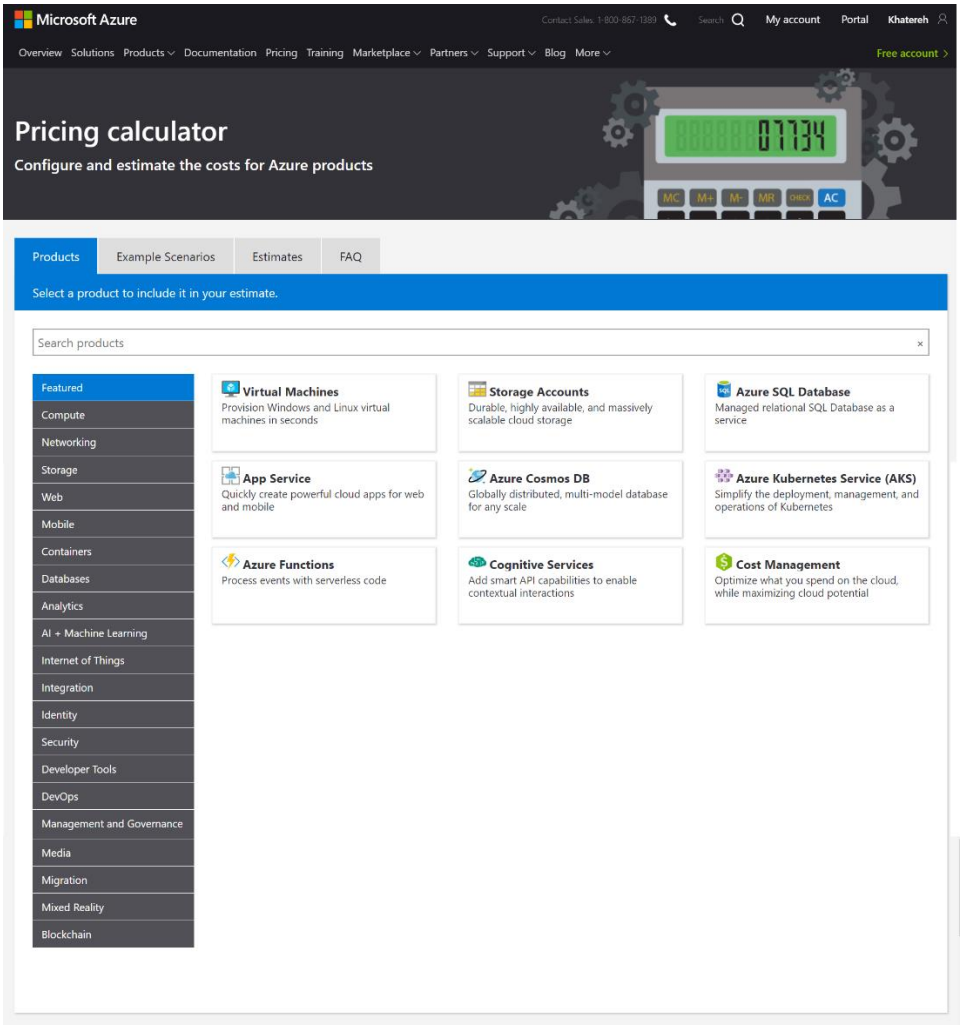

Figure 2 - 3 pricing calculator

Chapter 3: Project Description

As was already mentioned, this thesis focuses on moving a project from local to cloud using Microsoft Azure as a host for the application and managing it via the cloud.

The project is a car-pooling project, known as BePooler, that has been done before using C# and MVC. BePooler is a car sharing application which has two kinds of users; driver and rider. Users can register on the app and choose the user type. Drivers have to mention the model of their car and the number of people that they can seat in the car. This application is made for users that have a fixed time for travel. For instance, the time they usually go out of home to work and come back home. Each user will fill the timesheet for their week. Two times schedules are available for all days of a week. Users will fill that according to their working time. They also have to set the address of home and work in the application. The riders can send a request for a ride for any date they want, and they can also mention they want to be alone or they want to share with other people. The application will find matches for their request according to the drivers and the riders' timesheet and addresses. If the trip is a shared one, the application will also consider other trips have already been created but are not done or cancelled yet to find the best matches. If the rider chooses one of the matches, the application will send a request to the driver and the driver has to accept the trip.

The most important part of the BePooler application is the trips. Application has to find the best matches between the users according to their paths, and also with special formulas the price of the trip will be calculated. The price depends on the length of the path and also on the number of people during the different parts of the path. This amount of money will be blocked from the riders' accounts. If the trip is cancelled by a rider or the driver, the money will go back to the riders' account. When the trip is done and finished, the price will be taken from the riders' accounts according to their share amount. This amount of money will be divided to three

parts according to the percentages which are defined. One part will go to BePooler account, one part will be paid to the payment platform and the rest will be transferred to the driver's account. Stripe Payment has been used for payment platform. Stripe manages the real bank accounts and credit cards, and money transfers. The payment part has been renewed and changed to use Stripe webservices recently, after the decision to move the project to the cloud. As a result, it is an ASP.NET Core project and uses standard libraries to be compatible with Microsoft Azure.

The goal of BePooler is to encourage people to share cars in rush hours to use less cars in the city which leads to less traffic and less pollution. It also costs less than taxis; therefore, it is more economical.

The project includes two main parts that should be moved to the cloud; the code part and the database.

For the code part, there is a mobile and a web application both of which have been done and are being used online. These two applications are used by BePooler users for registration, using the profile and managing their trips. There is a web API project which both applications are using through web services. This API contains webservices for registering the information of the users, finding the matches, calculating the prices and so on. This API is connected to the database and saves the data that comes from both applications or reads data from the database and sends it to the applications. It is also connected to .NET core payment project to send the amount of money to the payment platform and ask for blocking, unblocking or transferring the money. There is also another project for administrating and management of car-pooling application's data, users and payments.

The project is using five DBs that are connected to SQL Server. The first one is being used by the web API which is named as “BepoolerDB”. This is the project’s main DB and is the biggest and the heaviest one. “BepoolerDB” contains the data needed for the car-pooling application, such as user information, trip information, payments and so on. The web and mobile applications are storing to and reading data from this DB by calling the web API’s services. The web application is also using another DB which is called “BepoolerWeb”. The administrating application is connected to a DB whose name is “BepoolerIntranet”. There are also two other DBs which are called “BepoolerDBHistory” and “SocialCarDB”. All the 5 DBs are using each other’s data and they are connecting to each other by “Synonyms”.

A synonym is an object at the database level that enables you to define an alternative name for another object in the database. The aliased object may be in the same database, on the same server in another database, or on another server as well. The database objects that can be created with synonyms are tables, views, stored procedures, and user-defined functions. [9]

In this thesis, we have in mind to move the BePooler project from local to the cloud using Microsoft Azure so that there would be no need for any physical server, DB, and so on, and everything would be done through web services.

To this end, we should move from local servers to service. In order to use this service, we have to upgrade the project from .Net framework to .Net core. It is needed to change the project in a way that makes it possible to use Azure. For example, it is not possible to use custom libraries anymore. Instead, we must use services. It is needed to upgrade the libraries to .NET Standard. In this way, we will be able to continue using them from all our current projects, whether they target .NET Core or .NET Framework.

Then we must upgrade the projects to use frameworks that are compatible with .NET Core. Because we are using ASP.NET MVC and ASP.NET Web API, this involves migrating to ASP.NET Core.

Finally, we should upgrade the projects to use .NET Core. If there are some APIs that are unsupported, we will need to migrate to an alternative. In some cases, this can be a different, newer API, that is already a part of .NET Core. In other cases, we will need to use an external library that adds the missing functionality. Also, if there are some libraries that do not support .NET Core, we will either need to upgrade them, find replacements, or use them in compatibility mode.

The next step is to upgrade the database from SQL to Azure SQL database. In Azure, not all functions, synonymous or other things are permitted. If our DB is using them, we have to remove and change these parts.

After the upgrade process, some steps must be taken. In the first place, it is essential to create a cloud service app in .NET. Then an SQL Database in Azure should be created. Furthermore, we must connect the .NET Core app to SQL Database. After that, we must deploy the app to Azure. Sometimes it is needed to update the data model and redeploy the app. Then it is possible to get stream diagnostic logs from Azure. Finally, we can manage the app in the Azure portal.

Since the application was so big and it was already done, it was impossible to check all parts, all codes and all databases' structures to find which parts are not compatible with Azure and try to change them. Furthermore, there were lots of libraries used in the project and finding out which libraries are not supported by Azure was a very time-consuming process, although we had tried to use only standard libraries during the implementation of the application.

Consequently, it was decided to start the migration and application deployment process, and if an error occurred due to something which was not supported, then we tried to find a solution or change the unsupported feature.

The implementation and results, the problem we faced with during the process and the solutions found for the problems are explained in the next chapter.

Chapter 4: Implementation and Results

In order to deploy a project to cloud using Microsoft Azure, some steps must be taken. In the first place, it is needed to register on Azure in order to have access to the Azure portal (Figure 4-1). Simply by using the website of Azure, you will be able to sign up for a new account.

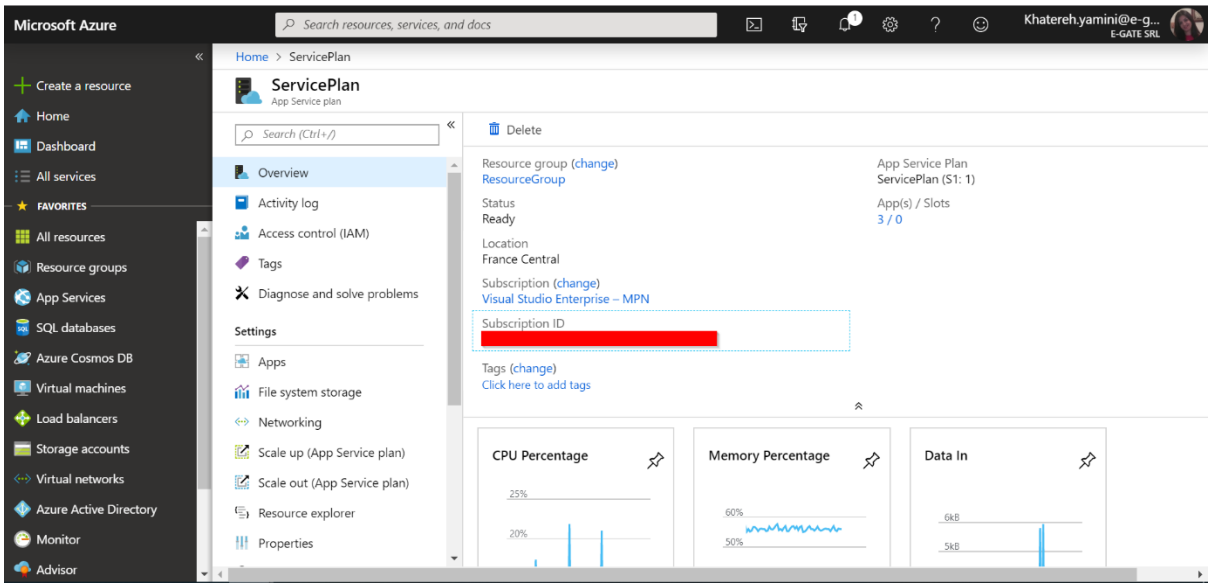

Figure 4 - 1: Azure Portal

> The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription using a graphical user interface. You can build, manage, and monitor everything from simple web apps to complex cloud deployments, create custom dashboards for an organized view of resources, and configure accessibility options for the best experience. [10]

Furthermore, a SQL Database in Azure should be created. Then we have to connect a .NET Core app to the SQL Database. The next step is to deploy the app to Azure. Finally, it is possible to manage the app in the Azure portal.

4.1 Database Migration

The very first step is creating a database in Azure. To this end, a resource group should be created in the Azure portal. A resource group is a container for an Azure solution that retains associated resources. The resource group involves the resources you want to manage as a group.

For this purpose, you can simply choose “Resource Group” (1) in the portal, select “Add” (2), choose a unique name (3) for resource group and create (4) it (Figure 4-2).

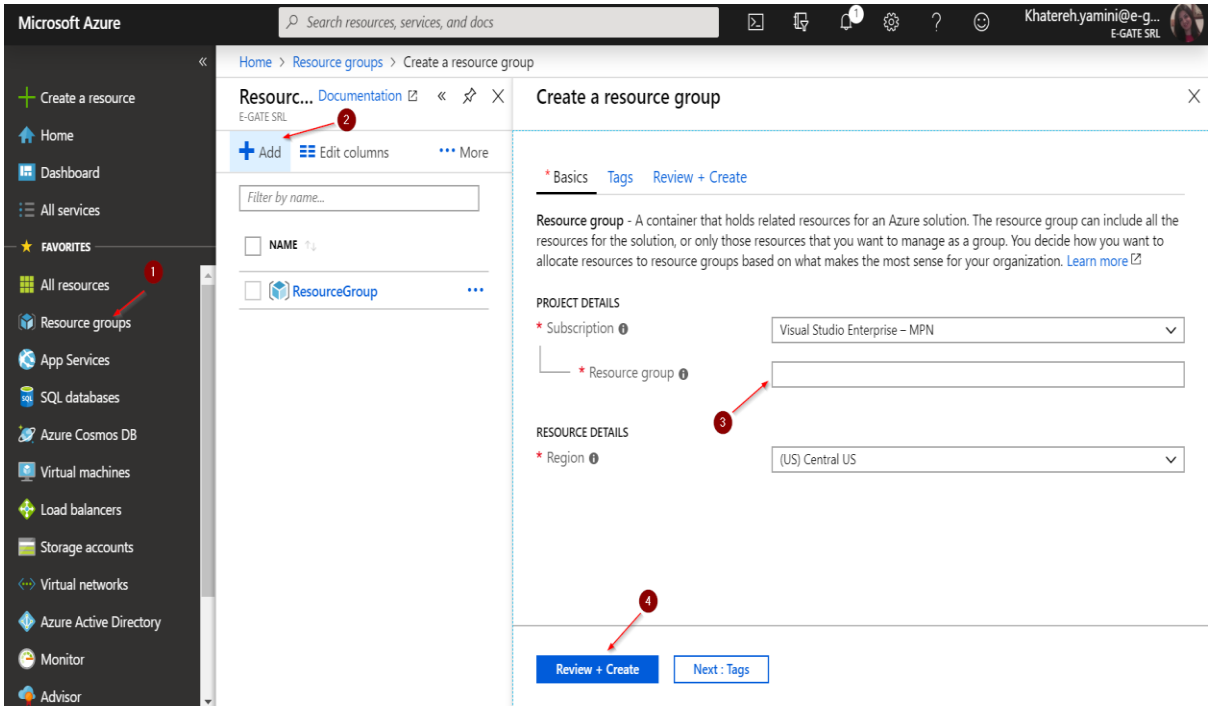

Figure 4 - 2: Create a resource group

Before creating a database, it is required to create a SQL server. It can be created as easy as a resource group. Server name should be a unique name, then a password and a resource group should be chosen for the server and finally create it. (Figure 4-3)

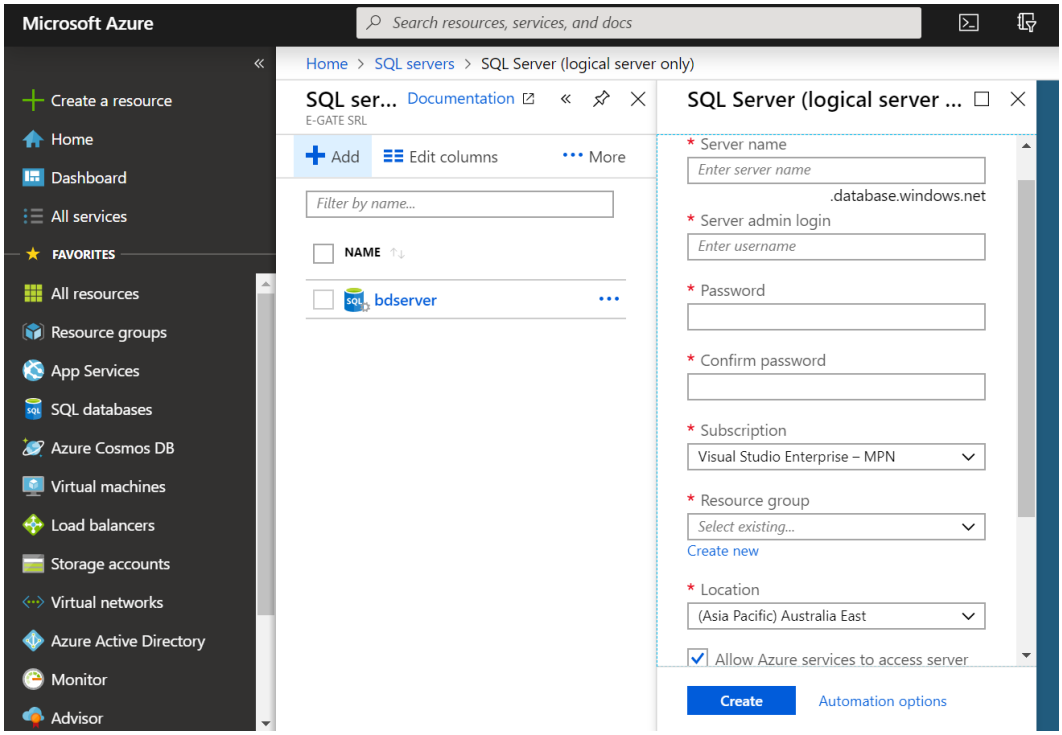

Figure 4 - 3: Create a server

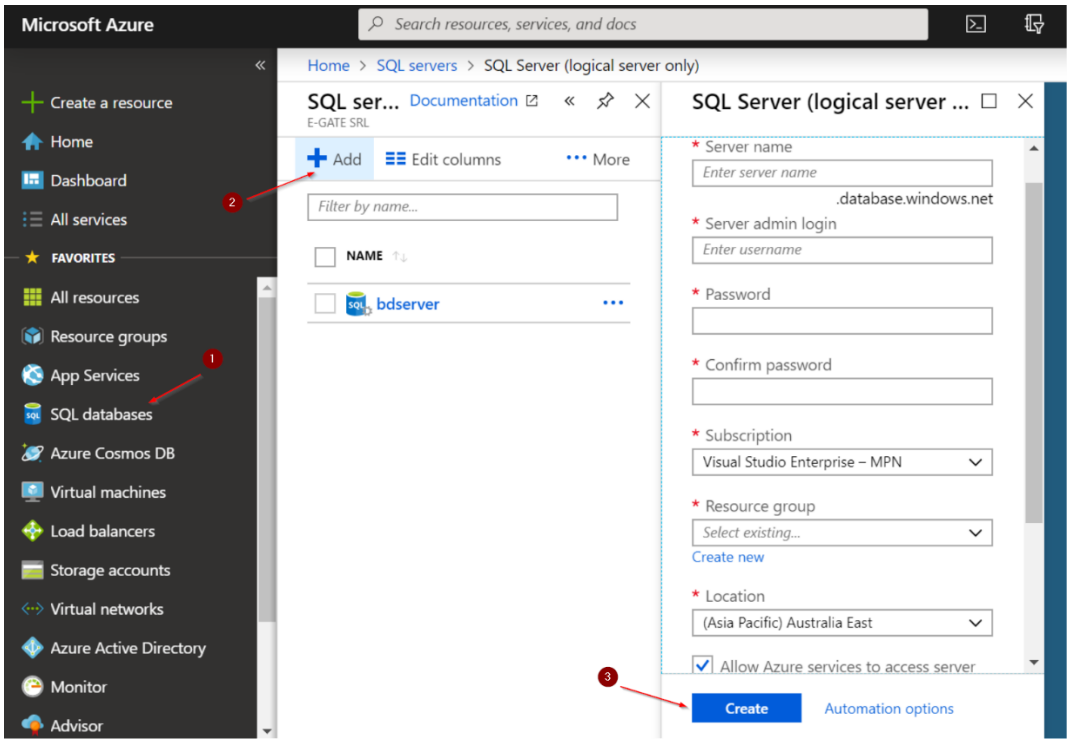

Figure 4 - 4: Create an Azure database

Having created the resource group and the server, it is time to create a database in Azure portal, by choosing “SQL databases” (1) from the menu, then select Add (2) and fill database information and create (3) it (Figure 4-4). All these 3 steps are simple and easy. Now there is an empty database and the problem is to migrate our database to the Azure database.

As was already mentioned, for BePooler project there are 5 databases; “BepoolerDB”, “BepoolerWeb”, “BepoolerIntranet”, “BepoolerDBHistory” and “SocialCarDB”. Therefore 5 Azure databases are created on the portal that can be seen in Figure 4-5.

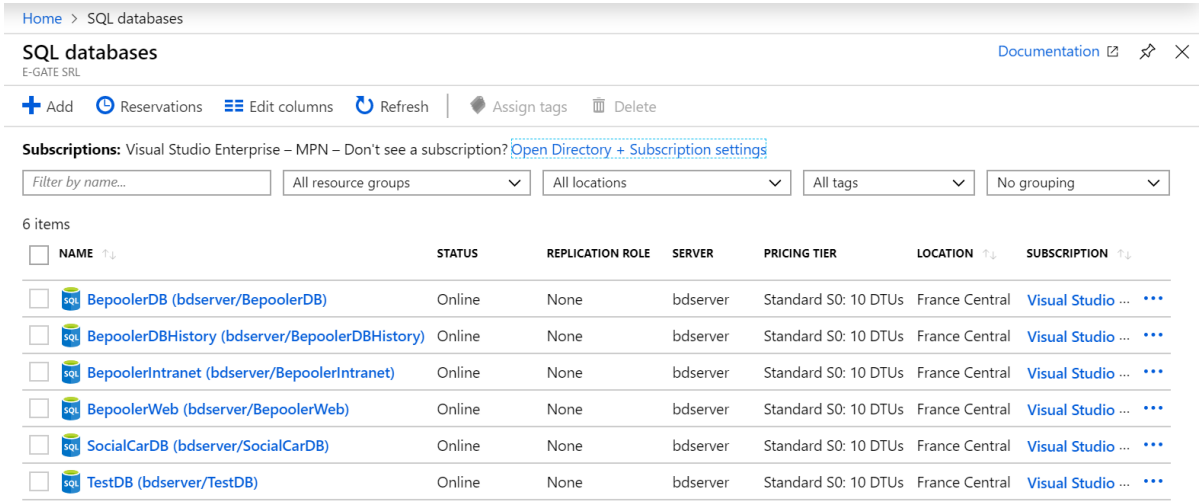

Figure 4 - 5: SQL Databases

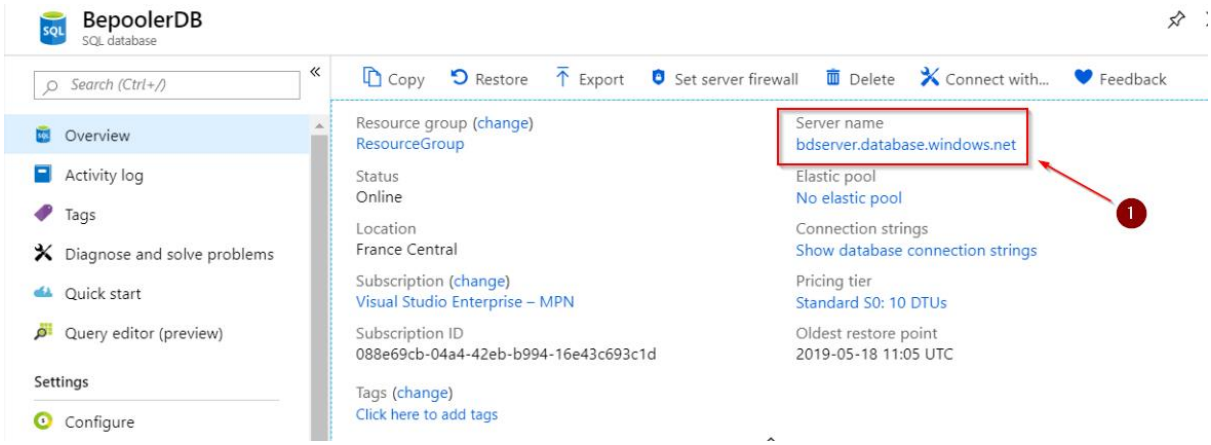

Figure 4 - 6: Azure database server name

It is possible to connect to the server created on Azure by SQL server management using the server name and the credential information is set while creating the server (Figure 4-6).

The problem that had to be solved was how to migrate the databases to Azure. In the first place, the project was online and was used by real users. It was not possible to do the migration from the production database, because if the migration were started the application would become offline and it would not be possible to have some downtime. Therefore, a new server was mounted in the company and nobody used it. A backup from each database was restored on the fresh SQL server of the new server. Now we could work on this database with real data and not worry about our users.

Furthermore, a solution should have been found for database migration. There are some ways to migrate the database to Azure. The first one is to use SQL Server Management on a local computer. It is possible to migrate data directly from SQL Server which is shown in Figure 4-7. But this solution has some limitation. If your database is small it will work for you, but this was not our case.

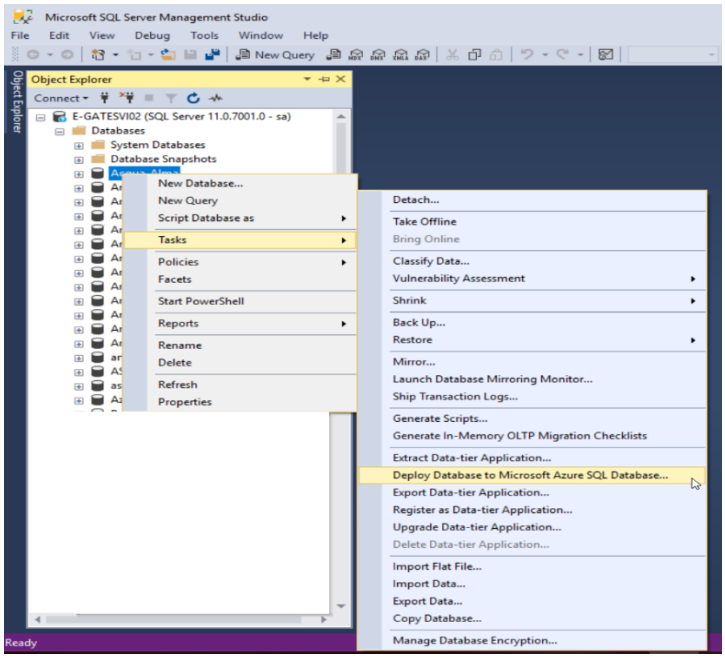

Figure 4 - 7: DB Migration using SQL server

Another solution, that is selected for BePooler project, is using “data migration assistant”. Data migration assistant is a Microsoft application that can be downloaded free.

> Data Migration Assistant (DMA) enables you to upgrade to a modern data platform by detecting compatibility issues that can impact database functionality on your new version of SQL Server. It recommends performance and reliability improvements for your target environment. It allows you to not only move your schema and data, but also uncontained objects from your source server to your target server [11].

In order to migrate database while using DMA, Source database (on local SQL server) and destination database (on Azure SQL server) should be selected. Analysing the database by DMA, a list will be shown with the elements of DB, mentioning that if the element will be transferred or there is an error and it will not be created on Azure. An error can be caused by, for instance, a wrong syntax in a stored procedure or a view, or because of an element which is not supported by Azure. To resole the errors, 2 strategies can be employed. The first one is to change the database element in order to solve the error and analyse the database by DMA one more time. Another strategy is to migrate the database without the element and add it later on the Azure database in a way it is supported.

As it is discussed, BePooler has 5 databases. Since some data in a database is needed in another database, there were lots of synonyms defined in the databases and many views were using these synonyms. Synonyms are alternative names that can be defined for database objects like tables, views, sequences, stored procedures, and other database objects. Synonyms can be used topically when is needed to provide an access to an object from another database.

Synonyms are not supported by Azure and they caused lots of errors while migrating the databases. In the first place, it was unavoidable to use them in BePooler databases. Furthermore, as they were not supported, they would not be migrated to Azure database. It was

decided not to migrate them and solve the problem later, on Azure database. It will be mentioned later how this problem was solved.

Another feature which is not supported in Azure is using three or four part names. For instance, if in a DB query there is "[BepoolerDB].[Table name].[Filed name]" it should be changed. Most of these kinds of errors for BePooler's database migration were caused by using synonyms. Therefore, they had to be solved later. The rest were solved one by one by checking the errors created by DMA.

Since BePooler project is a car-pooling application, it is unavoidable to use maps and locations. As a result, there were some fields in tables with the type of "geography". Geography type is not supported by Azure. The solution I found was to use "varbinary" type instead, which was suggested by Microsoft. Consequently, all the geography types in the database changed to varbinary type. This change caused some other changes to the code of the project. We had to find the codes which were using these fields and change them in a way which was compatible to use varbinary instead of geography.

There are some other types such as geometry, image, text, ntext and etc which are not supported by Azure. However, these types are not existing in BePooler's databases.

Having resolved the errors and ignoring those related to synonyms, the steps which mentioned were repeated and DMA reanalysed the databases. In the next step, DMA creates a script and then the migration will be started.

The migration started by BepoolerDB which is the main database and the heaviest one. Migration process lasted for about 40 hours (39 h 51m 22s). The result of the migration is shown in Figure 4-8.

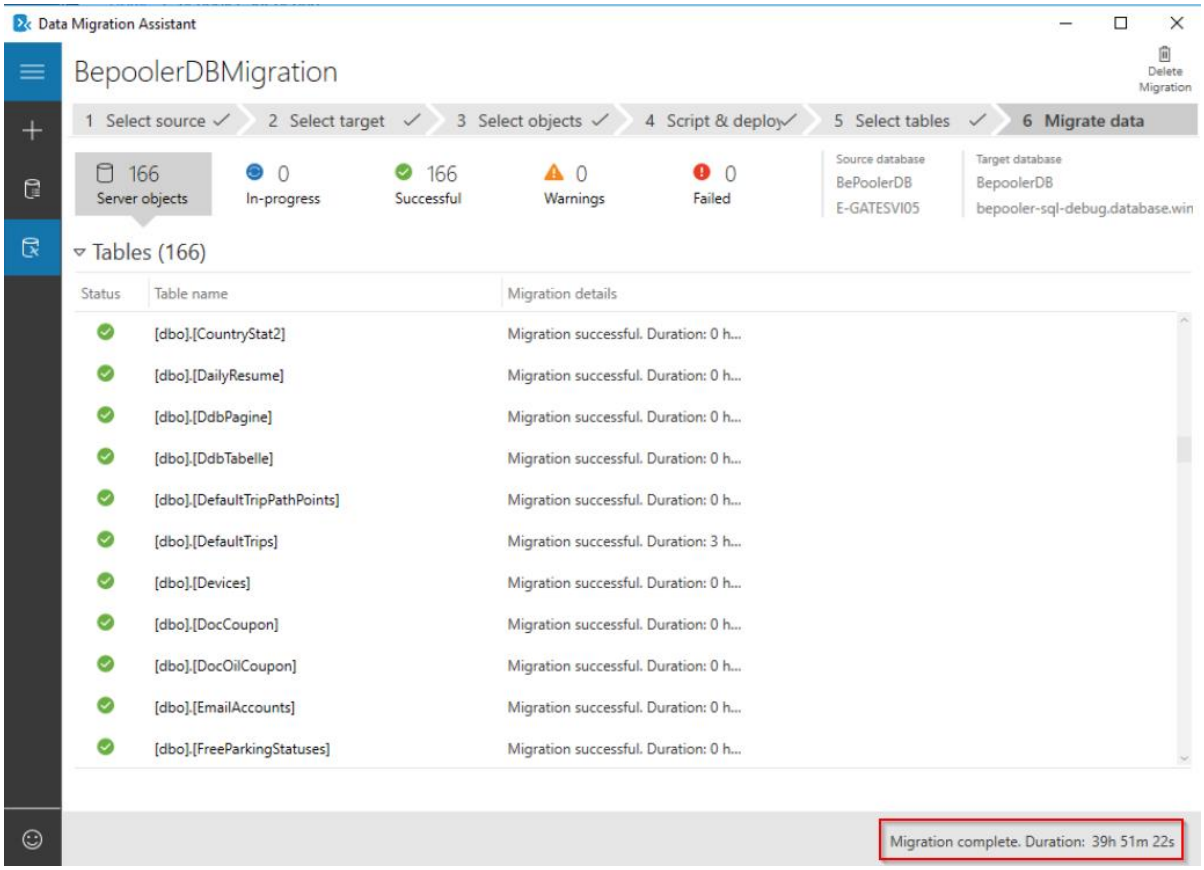

Figure 4 - 8: BepoolerDB Migration

The other 4 databases are light and migration duration was so short for them. DB migration lasted for "BepoolerWeb" 28 seconds, for "BepoolerIntranet" 20 seconds, for "BepoolerDBHistory" 30 seconds and for "SocialCarDB" 47 minutes and 52 seconds. The results are shown in Figure 4-9, Figure 4-10, Figure 4-11 and Figure 4-12.

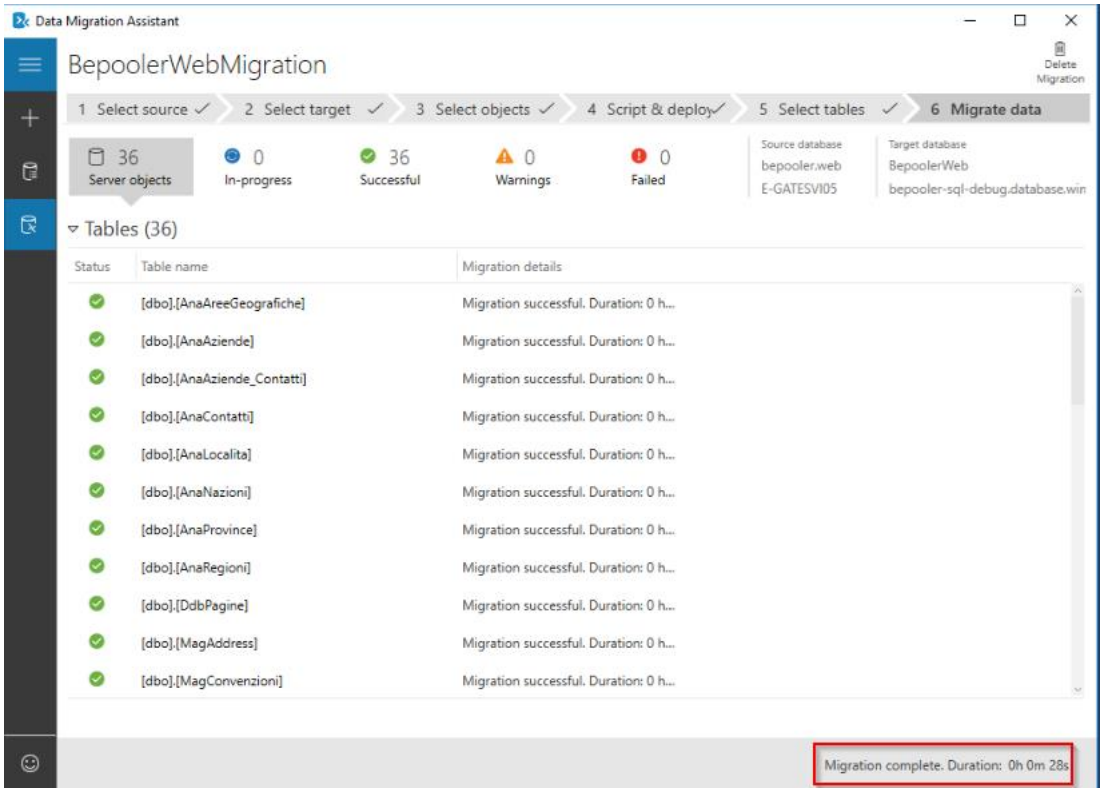

Figure 4 - 9: BepoolerWeb Migration

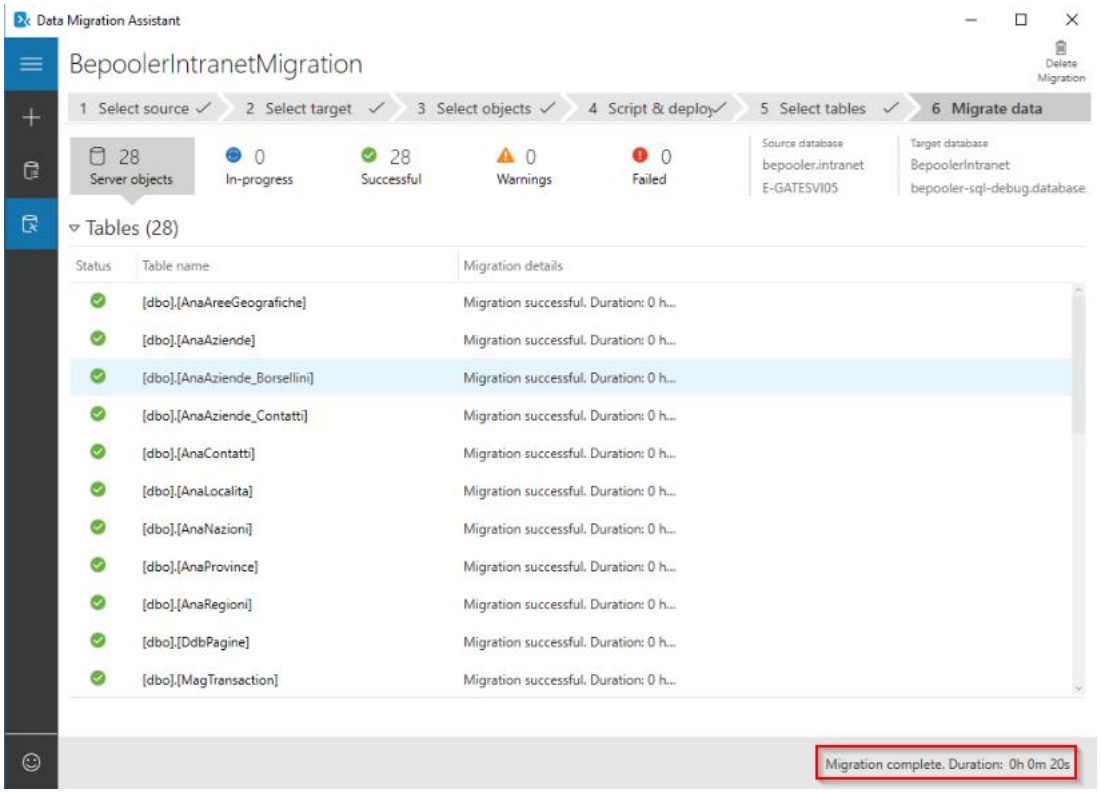

Figure 4 - 10: BepoolerIntranet Migration

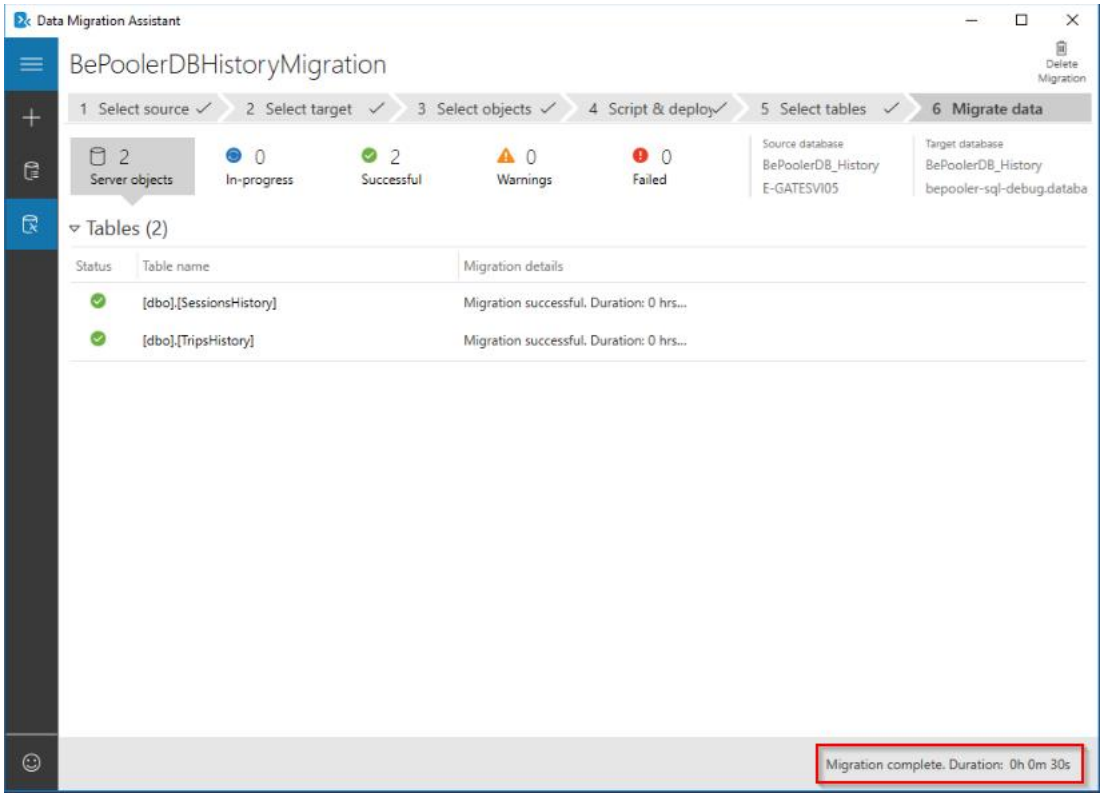

Figure 4 - 11: BepoolerDBHistory Migration

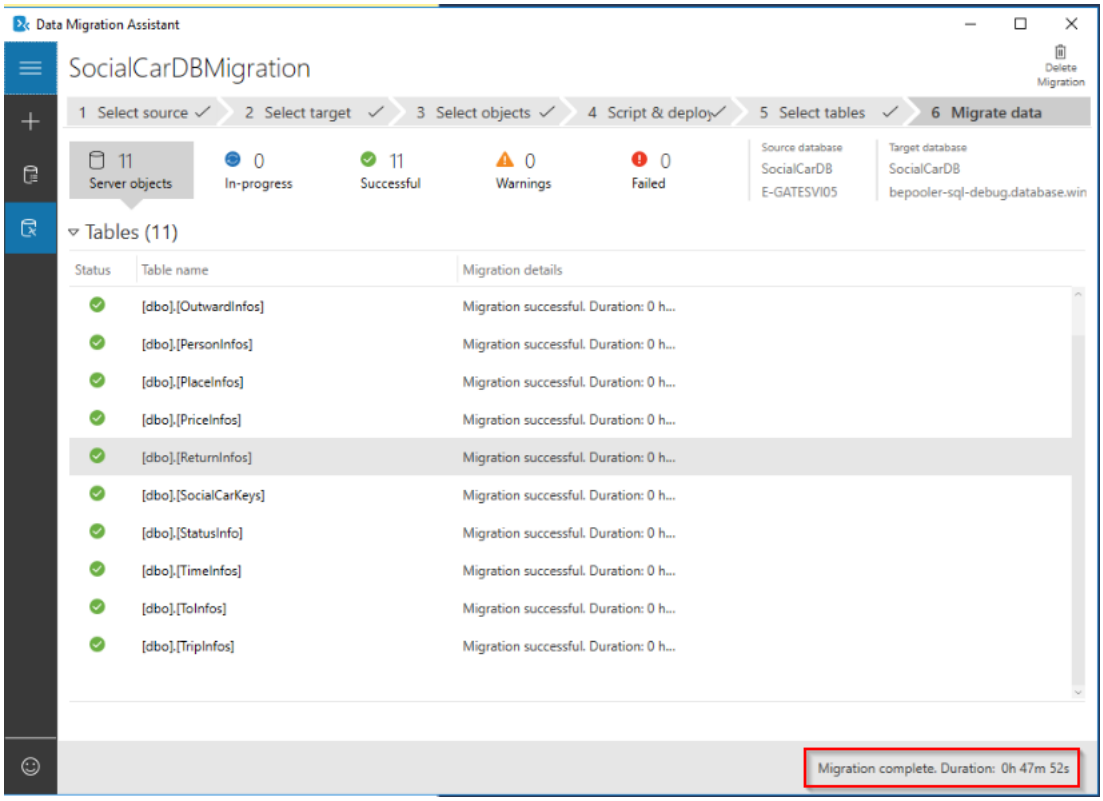

Figure 4 - 12: SocialCarDB Migration

Having migrated all databases, we should now turn to the possible solutions for the synonyms. The alternative solution that I found was creating external tables.

“External tables were first introduced in Oracle 9i. They provide easy access, by means of SQL, to data stored outside the database. The developer only has to create a table definition, which specifies the internal structure” [12].

To create an external table, it is enough to write a script and run it on the database. In the script, an external data source should be created for a remote database. Then an external table with the same schema of the synonym should be created which points to the external data source.

```
CREATE EXTERNAL DATA SOURCE RemoteBePoolerDB WITH
(
        TYPE = RDBMS,
        LOCATION = N' bdserver.database.windows.net',
        DATABASE_NAME = N'BepoolerDB',
        CREDENTIAL = …
);

SELECT * FROM AnaAziende_Syn
CREATE EXTERNAL TABLE [dbo].[AnaAziende_Syn](
        PrgAzienda      int      NOT NULL,
        …
)WITH(
        DATA_SOURCE = RemoteBePoolerDB,
        SCHEMA_NAME = 'dbo',
        OBJECT_NAME = 'AnaAziende'
);
```

For all synonyms an external table and then a view for the external table was created on the DB which were using the synonym before. This way all the objects that were not migrated to Azure DB were added after migration.

4.2 Connect the app to SQL Database

When the databases are created on Azure and the DB migration is done, it is time to connect the application to the new database. For this purpose, we only need to change the connection strings of the applications to the new database. The connection string of the database can be easily copied from the database's detail in the Azure portal (Figure 4-13) and pasted in web.config of the application (Figure 4-14).

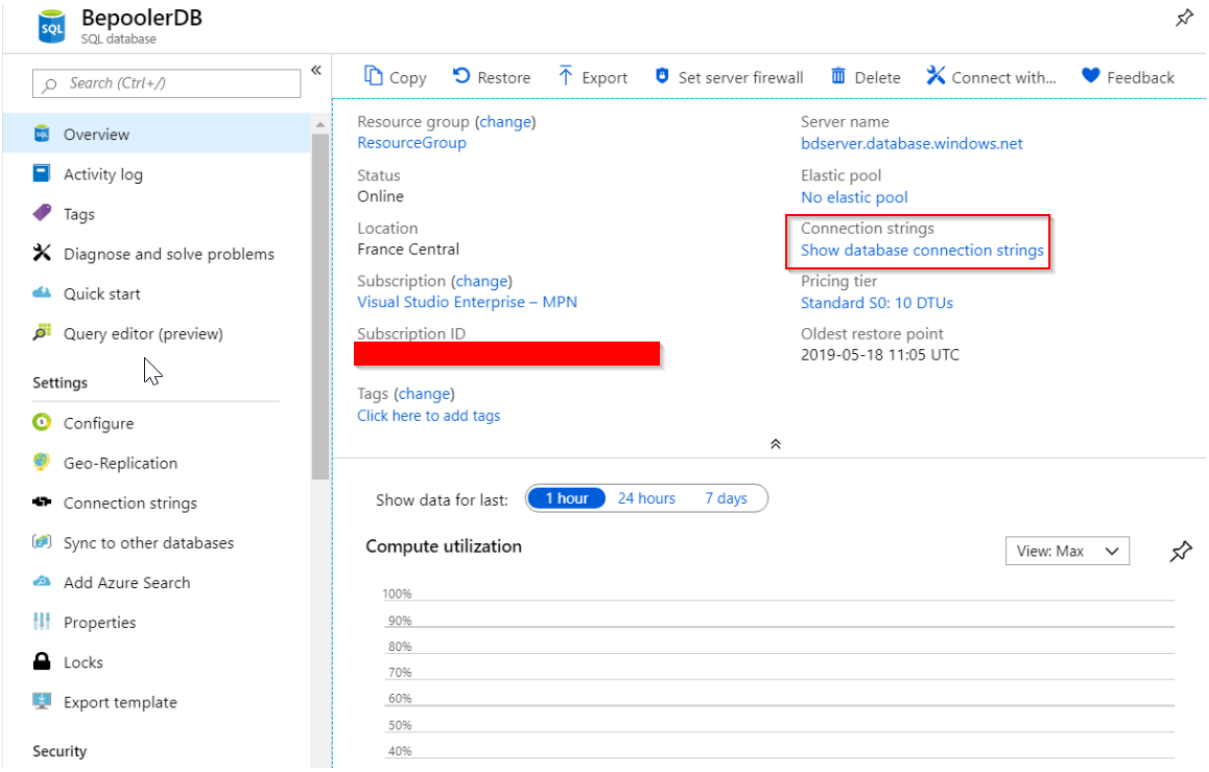

Figure 4 - 13: Connection string

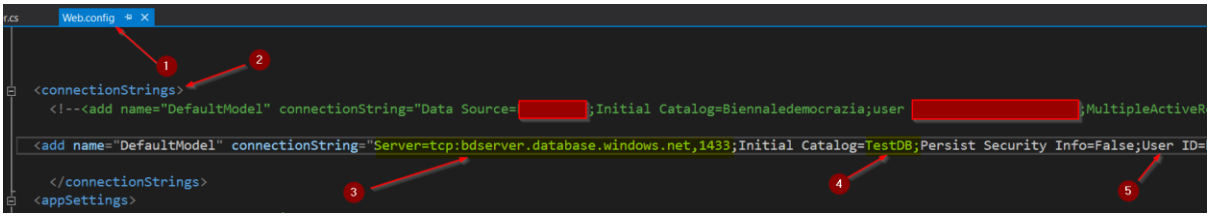

Figure 4 - 14: Web.config

4.3 Deploy the app to Azure

The last step is to deploy the application. Unlike a deployment on an in-house server or an outsourced host, deployment on Azure is very easy, fast and without human mistakes.

In the first place, an app service should be created on the Azure portal (Figure 4-15).

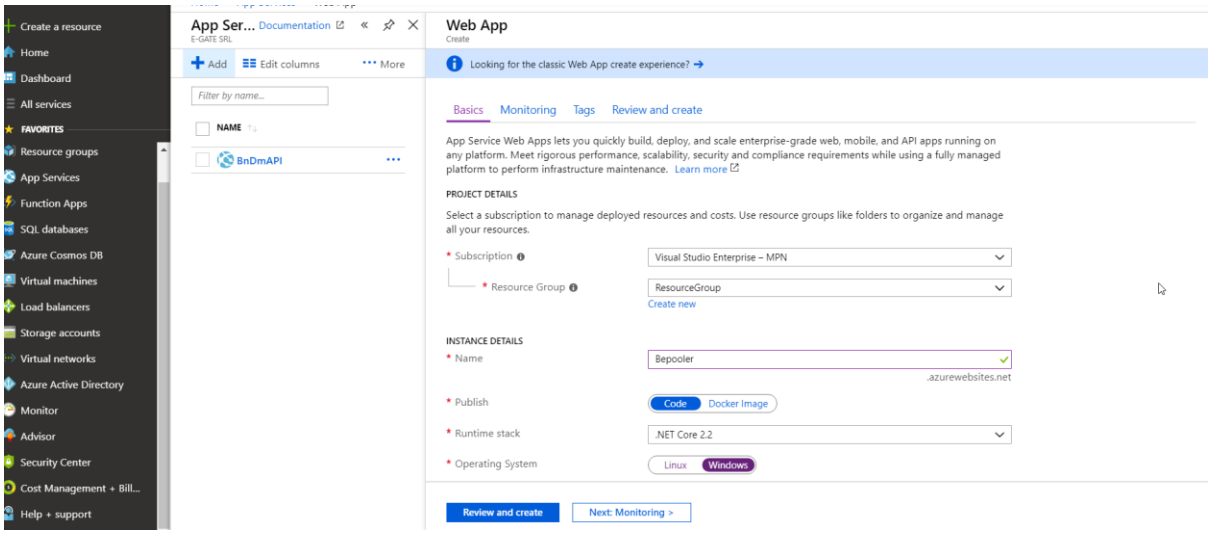

Figure 4 - 15: Create a web app

The name selected for the web app should be unique. Because ".azurewebsites.net" will be added to this name to create the link for the application (Figure 4-16).

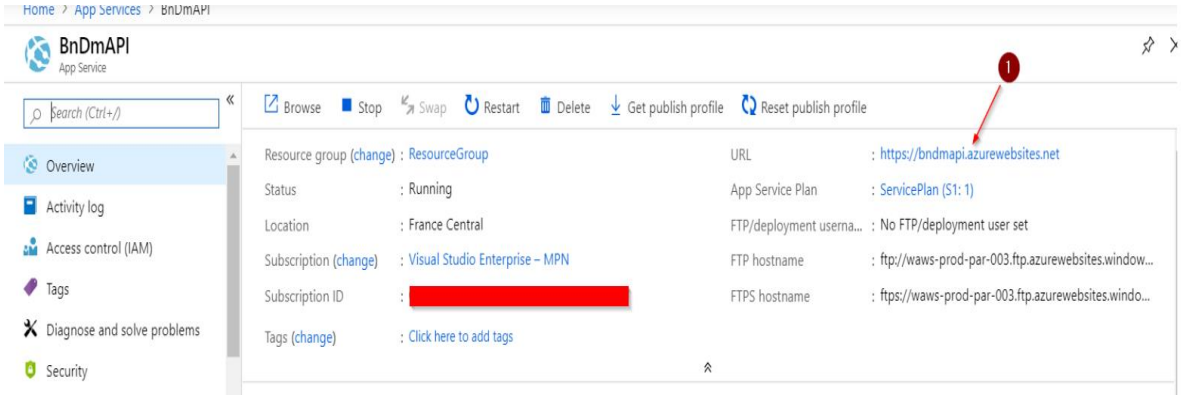

Figure 4 - 16: Web APP's URL

When the web app is created, the application can be deployed on that. Deployment can be done directly from visual studio. A right click on the project in solution explorer will open a menu. On the menu “Publish” has to be selected (Figure 4-17).

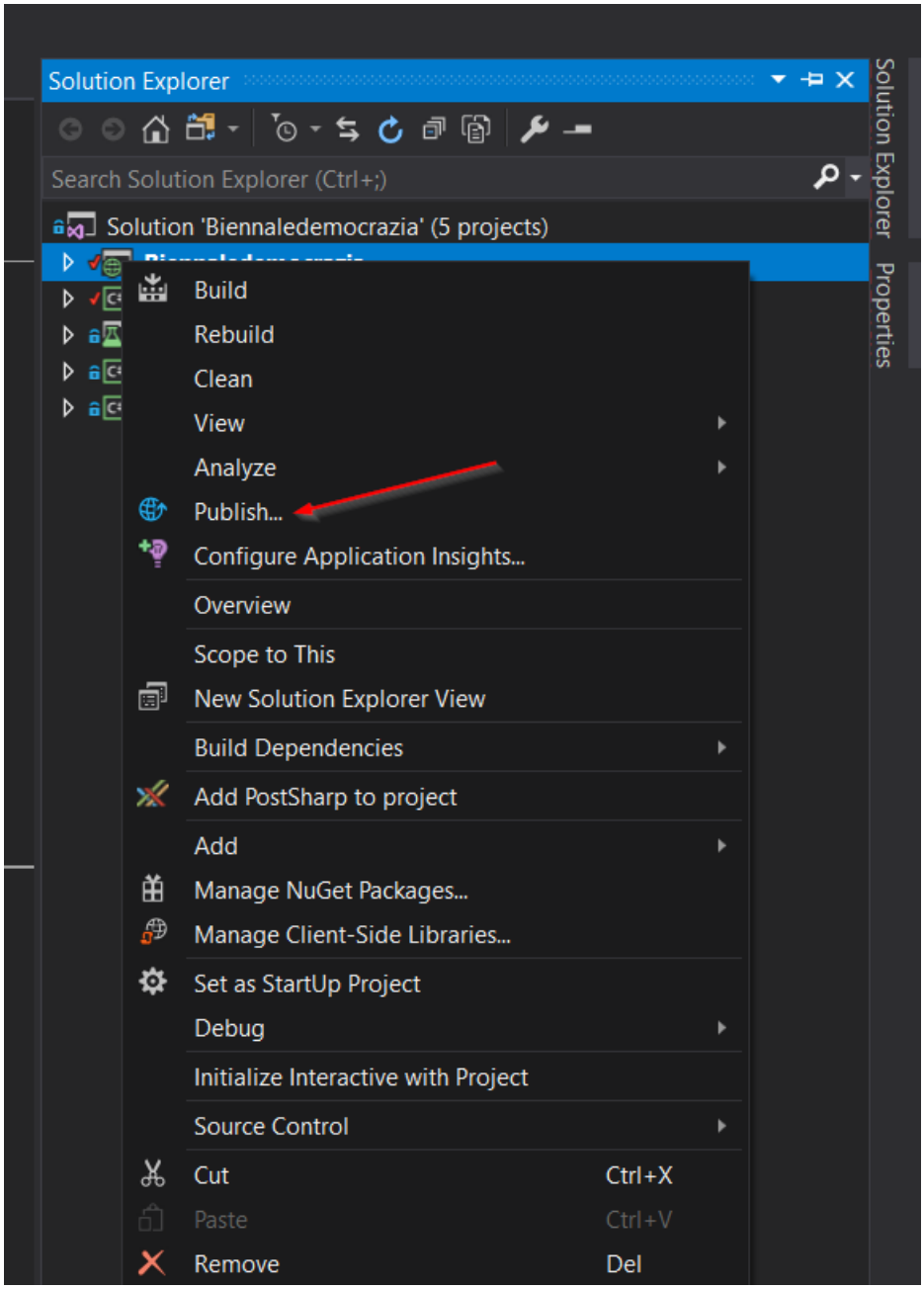

Figure 4 - 17: Publish

Then APP Service should be selected from the menu as a publish target (Figure 4-18). On the next step, if you are logged in on the Azure portal, the resource groups which are created on

the portal will be shown and then you can choose from the web app list existing in the resource group (Figure 4-19).

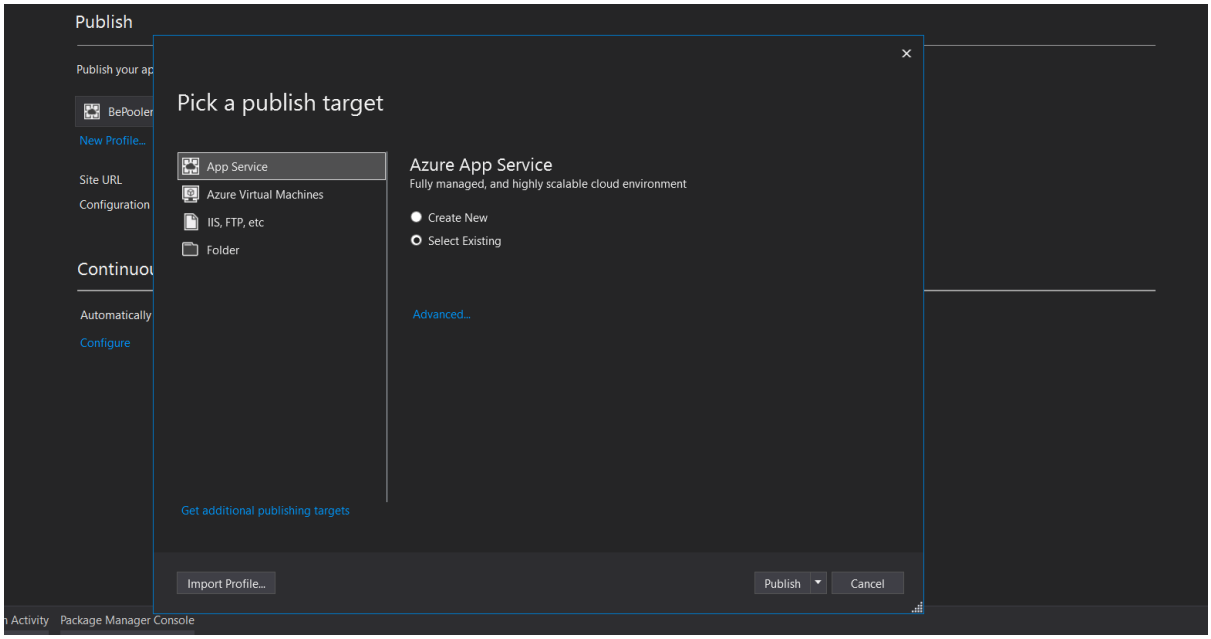

Figure 4 - 18: Publish target

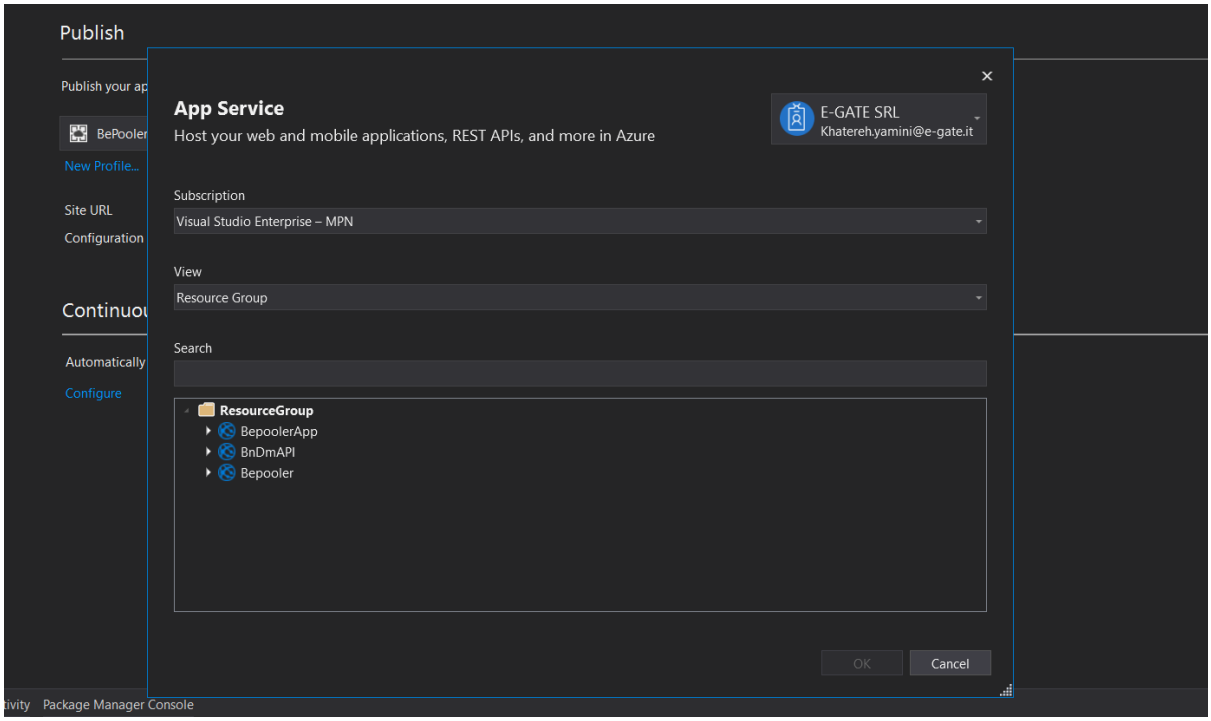

Figure 4 - 19: APP service selection

After selecting the app service and pressing OK, the deployment will be started. The time needed for deployment is only the time it takes for building the application. If there is no error in the code the application will be online (Figure 4-20).

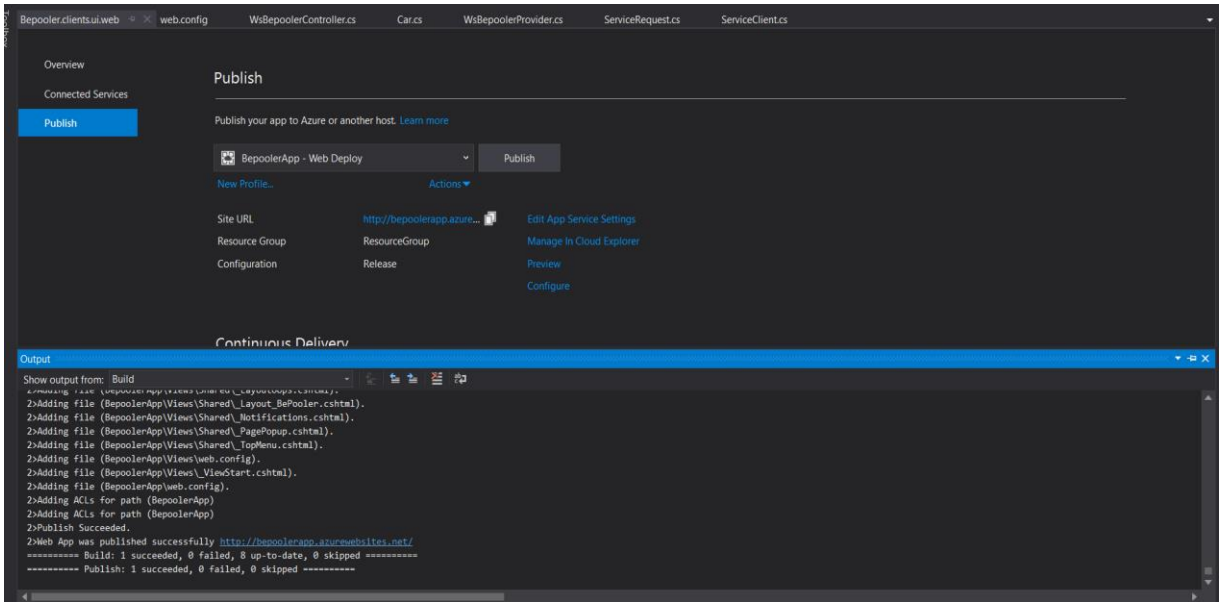

Figure 4 - 20: Deployment success message

4.5 Update and maintenance

Whenever something is changed in the code, or there is a new version, or a new part is added to the code it should be redeployed. Unlike publishing on in-house or outsource server, redeploying on Azure is very simple an easy. It is enough just to publish again as the first time (Figure 4-17). However, it is not needed anymore to set the target and app service again. When you click publish, all of them are already set (Figure 4-21). Because now the application is already connected to the cloud and the site URL is also accessible from visual studio (1).

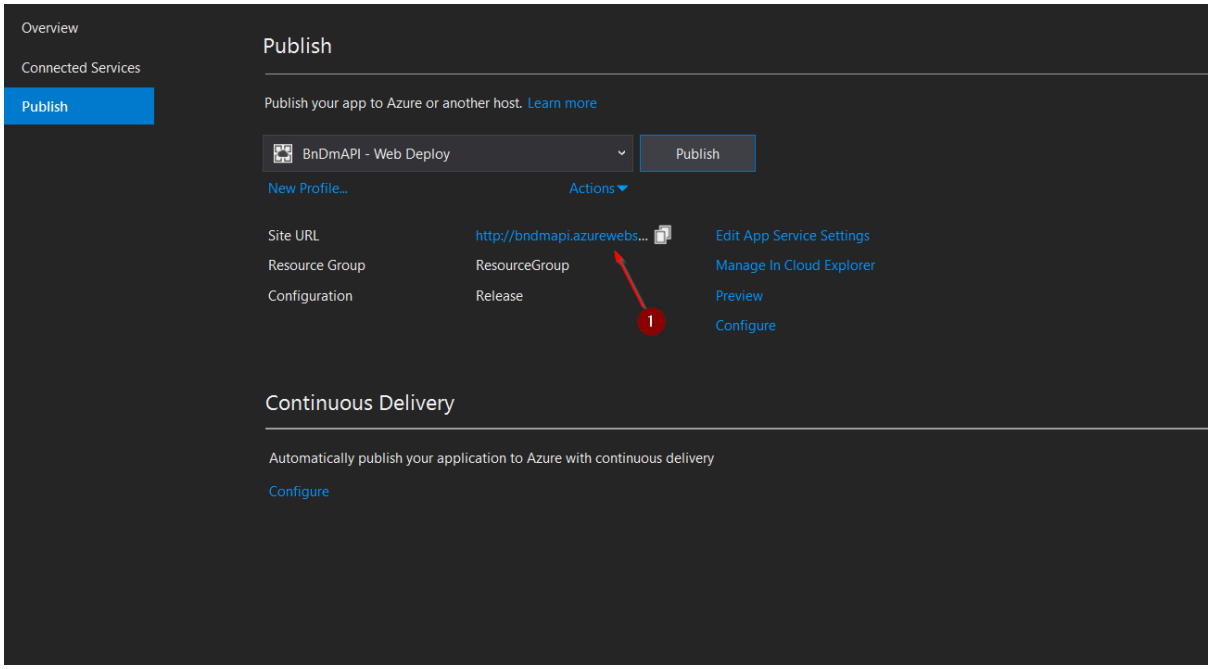

Figure 4 - 21: Redeployment

Updates and any changes in the database can be done directly on the cloud. From the query editor part (1) it is possible to access the database objects (2). We can edit directly the objects or run scripts in the query part (3). The query editor is shown in Figure 4-22.

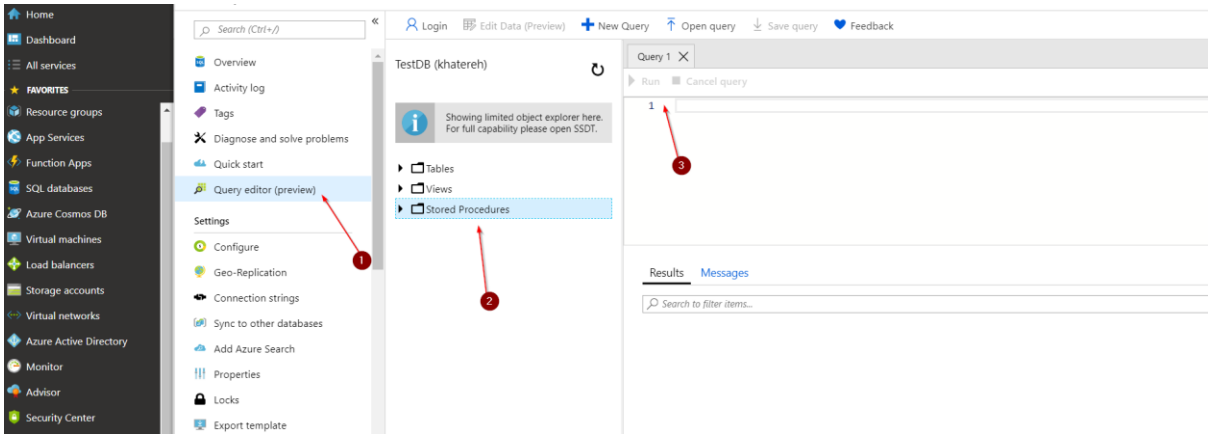

Figure 4 - 22: Query editor

Chapter 5: Conclusion

Nowadays, applications play an important role in our lives. In particular, online applications are widely used through the Internet. Consequently, one of the issues that software developers are facing, is to launch an application online.

Application deployment is making an application available for use. In order to satisfy this purpose, a server is needed.

This thesis focused on moving a deployed project from an in-house server to the cloud using Microsoft Azure. In conclusion, I hold the view that deployment to a cloud is much easier and faster. In addition, it leads to less human mistakes. In terms of database, using clouds reduces the risk of data loss.

In the first place, deployment on Azure is easier than publishing on a physical server. It is enough to only push the publish button. Everything will be managed by Azure itself. It is not needed to check which files are changed and have to be published again. There is no need to use remote access to the server which is time consuming and has many limitations, such as speed or the number of users that can connect at the same time to the server. Everything can be managed from visual studio and the Azure portal.

Furthermore, publishing on Azure is faster. As was already mentioned, using remote access to a server is time consuming, because it is slow. Deploying on Azure is fast, and it needs only the time for building the application on visual studio. In the case of remote access, you may face many errors whose solution waists lots of time. However, errors that occur during the deployment on Azure are only compiling errors which usually they are already solved while testing and coding, before the deployment. Although it was expected to face many errors because of the libraries that were used in the application and not supported codes by Azure, all

the errors that happened during the BePooler deployment were due to the absence of some files which were in the solution of the project. These files were mostly in temp folders for libraries we were using, like java script's libraries (Figure 5-1). They were solved simply by removing them from the project solution, which helped to find unnecessary files and make the solution lighter. I consider this as another advantage.

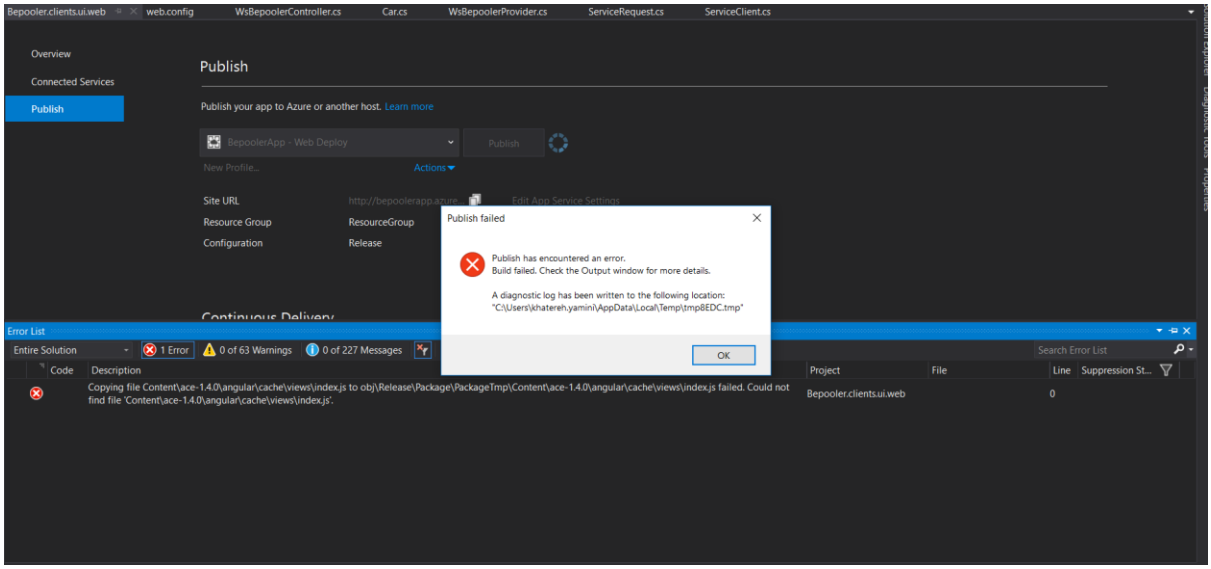

Figure 5 - 1: Deployment's error

Moreover, using the cloud reduces human mistakes. While publishing the applications using physical servers, you have to consider which files are changed so that you can publish them again. If some files or some changes are forgotten, it will lead to some downtime on the application, which happens a lot. However, all of the publishing processes on the cloud will be managed by Azure. Therefore, human mistakes will be reduced efficiently.

Finally, the database is protected on the cloud by high security. In addition, by taking backups regularly, the risk of data loss will fall down.

To sum up, I hold the view that using clouds will be the future of servers which makes life easier for application developers to deploy their applications.

For what concerns the future activities in this filed, different cloud services like Amazon, Google cloud, IBM cloud, etc. can be used for application deployments and they can be compared to find out which one is better, or which one is suitable for different kinds of applications.

References

[1] Ian Foster, Yong Zhao,Ioan Raicu,Shiyong Lu, "Cloud computing and grid computing 360-degree compared," *arXiv preprint arXiv:0901.0131,* (2008).

[2] "techopedia," [Online]. Available: https://www.techopedia.com/definition/2/cloud-computing.

[3] M. Rouse, "TechTarget," [Online]. Available: https://searchcloudcomputing.techtarget.com/definition/cloud-computing.

[4] "IBM," [Online]. Available: https://www.ibm.com/cloud/learn/cloud-computing.

[5] M. Rouse, "TechTarget," [Online]. Available: https://searchcloudcomputing.techtarget.com/definition/Windows-Azure.

[6] "TechinPost," [Online]. Available: https://www.techinpost.com/what-is-microsoft-azure/.

[7] S. Sanchez, "DYNTEK," 30 Nov 2017 . [Online]. Available: https://www.dyntek.com/blog/4-key-advantages-of-microsofts-azure-cloud.

[8] "Enlyft," 2019. [Online]. Available: https://enlyft.com/tech/products/microsoft-azure.

[9] R. Rehak, "TechTarget," November 2007. [Online]. Available: https://searchsqlserver.techtarget.com/tip/Configure-aliases-for-SQL-Server-databases-and-servers.

[10] Robert Lyon, Kate Follis, "Microsoft," 24 May 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/azure-portal/azure-portal-overview.

[11] "Microsoft," [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=53595.

[12] P. Kok, "AMIS TECHNOLOGY BLOG," 5 April 2005. [Online]. Available: https://technology.amis.nl/2005/04/05/updateable-external-tables/.

[13] L. Dignan, "Top cloud providers 2019," *https://www.zdnet.com,* 2019.