

Web Services Deployment in Microsoft Azure Cloud Computing Platform

Emanuela Srbinovska¹ and Pece Mitrevski²

Abstract – Data processing in the cloud is based on shared resources and services used by clients. Microsoft (Windows) Azure cloud computing platform can be taken as a considerable option as hosting environment for web services. In this paper we present how to use Azure for modern web service deployments, through a Windows 8.1 application that uses WCF web services deployed in Worker Role instance of a Cloud Service.

Keywords – Cloud computing, Microsoft Azure, web service, deployment.

I. INTRODUCTION

Cloud as a concept is built on the basis of established IT trends that are intended to increase productivity, efficiency and scalability through dynamic utilization and virtualization of resources, improve accessibility and flexibility of services with maximum utilization of existing facilities, achieve higher performance using fewer resources, introduce globalization and standardization of services, achieve higher degree of control, improved monitoring processes and minimization of need for software licensing, and thus introduce low cost labor and reduce overall capital and infrastructure costs.

Today, there is a wide spectrum of services available in the cloud, including collaborative and messaging solutions, storage solutions, identity management solutions, customer relationship management etc. Major vendors have also released cloud services based on their widely used on-premises software products. Businesses generally consider moving their applications to the cloud for one or more of the following reasons: speed, scale and economics.

Microsoft (Windows) Azure is a public cloud that is running on Microsoft data centers and offers a variety of services from provisioning and load balancing to health monitoring for continuous availability of services. At the core of the Azure platform is its ability to execute applications running in the cloud.

Azure is one of the largest cloud computing platforms that offer different types of services in the cloud. The benefits of using Azure are significant for all participants. Service providers can expand their services in areas where they have existing infrastructure and add new services without major infrastructure investments. Software engineers can use Azure to create, configure and manage web applications and web services without major capital expenditures and prepare the

resources needed quickly and efficiently.

Microsoft has identified ten different solution categories where Azure can bring significant benefits to users: infrastructure, mobile, web, media, integration, identity and access management, big data, development and testing, data storage, backup and recovery, data manipulation and data management.

Up until a few years ago web services were hosted in server clusters distributed across geographic regions and networks all over the world. Nowadays, hosting web services in the cloud becomes an extremely fast growing trend and widely accepted solution. Developers need not be concerned about overprovisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or underprovisioning for service that becomes widely popular, thus missing potential customers and revenue [1].

Azure currently provides four different models that comprise the compute services section of the Azure platform: Web Sites, Virtual Machines, Cloud Services and Mobile Services. All these features can either be used separately or combined together to build solutions that can meet specific needs. The number of different services available in Azure is expanded with new services continually being added and existing services enhanced as the Azure platform continues to develop and expand.

Azure also provides improvements for hosting web services including keeping up at least three redundant and independent copies of the service that will ensure high level of availability as well as ability for scaling. According to the service demand, the idea in scaling up (vertical scaling) is to increase capacity of individual nodes through hardware improvements while scaling out (horizontal scaling) is intended to increase capacity by adding entire nodes. Scaling down is the reduction of capacities, performed if service load is lower than expected [2].

II. DEPLOYING WEB SERVICES IN MICROSOFT AZURE

Azure offers several ways for deployment and communication of web services using machines with variety of Windows and Linux configurations. Because Azure is a multiplatform environment, Azure SDK or suitable set of tools for appropriate development platform is needed. Therefore, Azure offers appropriate SDKs for several languages as well as different tools for corresponding development platforms that provide shared resources which can be used in all segments of application development lifecycle. At the end, this will contribute to final setting of software solutions into service in the cloud.

¹Emanuela Srbinovska is with the Faculty of Technical Sciences at University of St. Clement Ohridski, Ivo Lola Ribar, Bitola 7000, Republic of Macedonia, E-mail: emanuela.srbinovska@yahoo.com

²Pece Mitrevski is with the Faculty of Technical Sciences at University of St. Clement Ohridski, Ivo Lola Ribar, Bitola 7000, Republic of Macedonia, E-mail: pece.mitrevski@uklo.edu.mk

Azure development tools are available for following platforms: .NET, Java, PHP, Node.js, Python, Ruby, mobile applications and media [3]. Azure packages for .NET include palette of integration tools for Microsoft Visual Studio. These tools are provided to enable development, packaging, operating and debugging of scalable web applications and web services on Microsoft Azure by using some of the .NET based programming languages.

In this paper all steps and procedures are performed with the usage of Azure SDK for .NET and Visual Studio as a development tool.

A. Deploying Web Services in Azure Web Site

One option for deploying web services on Azure is within Azure Web Site. Azure Web Sites are flexible, scalable and secure platform components that can be used for building web applications that run client application and web services. Azure Web Sites own an easy usable self-service portal with a gallery of different templates for web solutions that can be used without having to create and update virtual machines, install and configure Internet Information Server (IIS) etc [4].

The procedure for hosting service on web site consists of several steps and in continuation we will review the procedure for hosting .NET based web service, developed in C# programming language. First, Azure web site needs to be created from Azure Management portal or from Azure management tool integrated in Visual Studio. Then, in the client project that will use the web service, ASP.NET web site should be added to serve as a host for web services. The references are added to the existing services in the solution project itself. The next step is to publish web service on previously created Web Site component within Azure. In Fig. 1 all Visual Studio publishing settings for web deployment are presented:

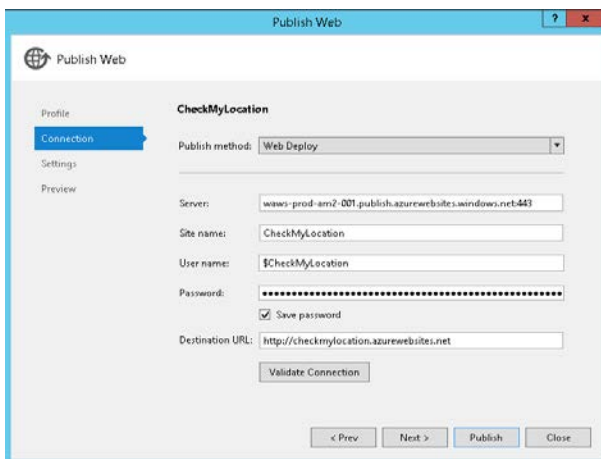


Fig. 1. Publish web service from Visual Studio solution on Azure Web Site

B. Deploying Web Services in Azure Cloud Service

Second option for deploying web services on Azure is by using Azure Cloud Services. Azure Cloud Services offer an

opportunity to set up and work in the cloud application level where Azure takes care of all details including security, load balancing and monitoring. All these features in combination provide continuous availability of applications. In other words, Azure Cloud Services are containers of hosted applications and web services [5].

From a security perspective, it is important to point out that any failure or error of a virtual machine does not cause the user to lose data and/or services, as they are stored in Azure database. If an error occurs, the controller sets the Azure services and hosted applications where necessary. In addition, if there is a need, Azure configures a new machine and all user files and hosted services are transferred to the new machine. Thus user data and machines are fully secured and there is no risk of data loss.

Physically, Azure Cloud Services are sets of virtual machines. Virtual machines can belong to one of two different types: Web Role or Worker Role. A role instance is the virtual machine on which application code runs. Each role can have one or several instances that are defined in the Cloud Service configuration file (.csdef) which determines the service model for the application. The Cloud Service configuration file (.cscfg) specifies configuration settings for the service and its preconfigured roles, including the number of role instances. Finally, the service package file (.cspkg) contains the actual application code along with the cloud service definition file.

Hence, Azure Cloud Service role that consists of application files and XML configuration files can be either a Web Role or a Worker Role.

A Web Role is a cloud service running on Windows Server with preconfigured Internet Information Server (IIS) and is typically used for hosting front-end web applications or middleware tier service layers. Worker Role, on the other hand, is a cloud service running on Windows Server without Internet Information Server (IIS). Worker Roles offer host for applications that can run asynchronously and which are generally used to perform long running data processing tasks that are independent of user interaction.

When setting up services in the Azure Cloud Service component user has more configuration options available and according to that, higher responsibility. Thus, the user can make the decision whether comparative would use a physical machine with or without IIS. Virtual machines of Web and Worker Roles use architecture based on Platform as a Service (PaaS) concept and can work with Windows Server 2012 R2, 2012 or 2008. The user does not need to install or maintain operating system or anything else on the machine because all these functionalities are under the competence of Azure controller.

Since the Web Role essentially provides features and functionalities to host front end web applications in the cloud, Microsoft Azure SDK comes with tools that let developers build, test and deploy web-based applications on Web Role.

Hosting web service in a Web Role instance in the cloud is performed by IIS. The main endpoint for the service is located on main port, usually port 80, and the firewall must be open to that port. To configure Azure Cloud Service based on a Web Role, Web Role component should be added in the solution project. In Web Role project, there is a host with the proper

name, as well as additional project that looks like a classic ASP.NET web application with several additional references that are pointing to Azure. Then, ASP.NET web site should be added into solution. This web site will add references to existing services in the application. The next step is setting up a Cloud Service in Microsoft Azure. This can be done through Visual Studio or service can already be created and configured from Microsoft Azure Management portal. Once the service is configured it needs to be published. With publishing operation, a preconfigured virtual machine with installed operating system and IIS is created in the background, where web service is deployed and accessed by its clients.

Although Web Role is ideal for hosting web applications, it does not provide features to execute long running tasks like business workflows or complex calculations with lengthy processing required. To decouple front-end and back-end operations, Azure Worker Role is available. The core function of the Worker Role is to process tasks that are considered too costly for the Web Role. Commonly, Web Role delegates processing to a Worker Role and focuses only on front-end related activities, like hosting web sites that provide a user interface.

Web services deployment in Worker Roles is more complex than deploying a web service in Web Role. Because of the absence of IIS, there are no fixed endpoints as in the Web Roles and therefore additional software using scripts should be installed. The user is required to create the endpoints for Azure, endpoints for the web service in the service code and configure service host.

The first step is adding a new Worker Role project into solution. For the newly created role configuration settings in terms of instances, diagnosis and domain needs to be added. Then, appropriate endpoint port that will listen using the TCP protocol should be configured. When all configuration options for Worker Role are completed, the next step is the configuration of settings for the web service. Given that, service host needs to be set up to take the appropriate domain, as well as service endpoint and corresponding port. Next, the service needs to be published. Fig. 2 presents Windows 8.1 application that uses web services deployed in Worker Role instance: net.tcp://checklocation.cloudapp.net:8081/mex.

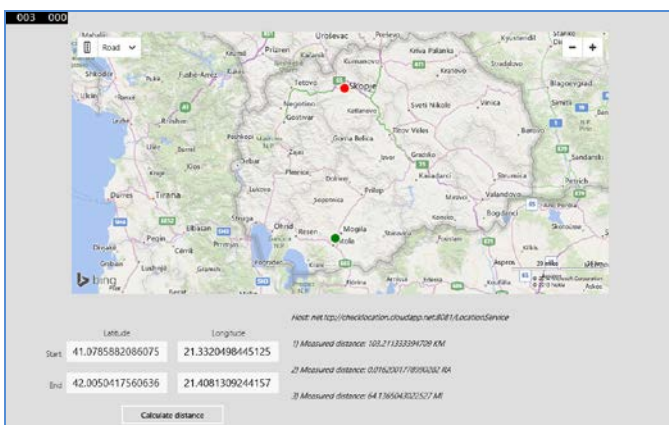


Fig. 2. Windows 8.1 application that uses WCF web services deployed in Worker Role instance of a Cloud Service

Microsoft Azure Management Portal is often used for monitor the health and availability of applications running on Cloud Services. From Azure Management Portal, alerts can be configured so that user can be notified in real time in case of a service error or interruption.

Within the Cloud Service published and running in Azure, there are many settings available in order to provide to examination and scaling of resources, increasing the number of service instances, scheduling service and many other options. These settings are presented in Fig. 3.

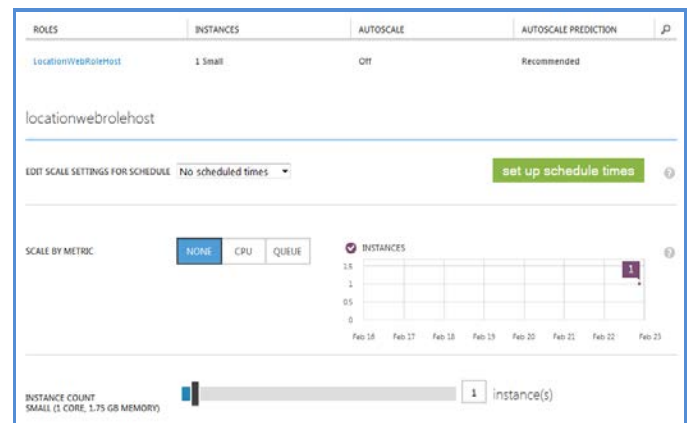


Fig. 3. Web Role configuration settings from Azure Management Portal

III. COMPARISON BETWEEN DIFFERENT DEPLOYMENT OPTIONS IN MICROSOFT AZURE

Azure Web Sites and Cloud Services Web Roles rely on a Platform as a Service pattern. Even though both approaches offer scalable platform for web services, they provide different levels of abstraction.

Web Site have a high degree of abstraction, because it basically provides IIS as the driver for the web services that are running in virtual machines. Virtual machines can be used by single user or shared by multiple users. If user asks for its own virtual machine instance, Microsoft Azure takes a virtual machine and only provisions the machine for the particular user.

In such conditions users do not have direct access to any of the other layers except for the IIS server. Because of this design pattern, users do not create new virtual machines every time when a new Web Site instance is needed. Out of the box, Azure runs a pool of virtual machines dedicated for the Web Sites service. Each virtual machine is packed with multiple IIS instances and when a new Web Site is required by user, Azure allocates one of the IIS instances in one of the virtual machines to run hosted application or web service. When resource scaling is needed, Azure just replicates user applications and web services to additional IIS instances.

Because Azure provides the platform at the IIS level, it can achieve much higher management and resource efficiency than if it provided the platform at the virtual machine level. Hence, Azure Web Sites come with low unit price for the services.

It is also worth mentioning that in their basis, Azure Web Sites are tremendous web application built on top of Cloud Services Web Role instances, in addition to many other components.

Specifically, IIS from the Azure Web Sites, as the application server runtime, includes support for the most popular development frameworks. Application based on the abovementioned platforms can run within the Azure Web Site. For dedicated instances, user can upload an SSL certificate, choose IIS mode (32-bit or 64-bit) and map a custom domain name for the application or web service hosted on the Web Site.

Additionally, Azure Web Sites offers a gallery of some of the most popular open source ASP.NET and PHP website templates including Django, WordPress, DotNetNuke, Orchard, Drupal and many more. User can simply select a template and Azure will quickly deploy selected template into website.

Because of the specific operation model, Azure Web Site can scale almost instantly because it is not waiting on a new virtual operating system being provisioned and configured. Azure simply allocates additional IIS or virtual machine instances from an existing collection of offered resources and thus reduces the time needed for process completion.

Additionally, Azure Web Site can integrate and work together with versioning and source control platforms, such as Team Foundation Server. Also, a user can connect to external databases through public ports exposed by the database server, but Microsoft Azure Web Sites cannot be provisioned inside a Virtual Private Network incorporated into Azure.

To conclude, high cost efficiency with Azure Web Sites can be achieved if users do not necessary need full control and flexibility on the underlying resources.

In comparison, the clearest usage of Azure Web Roles and Worker Roles in a multitenant application is in order to provide additional compute resources or reduce them on demand in the service of the tenants, with the notion that a single role instance serves multiple tenants [6].

Each time when a user demands a Web Role instance, Azure sets up a new virtual machine with the Windows Server operating system and IIS installed and properly configured. Then, cloud application packages from the user are deployed into the new virtual machine. Thus, users have full access to the virtual machine operating system during the deployment process as well at the runtime. Also, a user can run scripts for installing additional components outside of the IIS as well as perform custom configurations, install certificates and caching mechanisms, set up ports for public or internal usage and perform other adjustments.

Web Role IIS is configured with ASP.NET in a new instance, but the user is not limited to use just ASP.NET. Hence, one can install any other framework that leverages IIS and configure another web server along with the IIS. As long as Azure Fabric Controller can execute startup script without human intervention, the user can configure the virtual machines whenever it is needed.

Users can also create new Web Role instances as part of a multitier application into Azure Virtual Network, spanning multiple Cloud Services, including virtual machines based on

Infrastructure as a Service (IaaS). Each application tier can be configured with its own independent scalability settings according to the load. Remote Desktop Protocol (RDP) can be used for remote access into the virtual machines. Also, user can perform actions from the front-end of a cloud application inside of a network.

All available Web Role flexibility comes at a higher cost. Also, every time when a user instantiates a new Web Role or scales up or down an existing one, new virtual machines are created and this action takes longer time than creating or scaling Azure Web Site.

Worker Role, as previously mentioned, offers a clean Windows Server with no running services. Thus, the primary reason to use a Worker Role is when running background, long running processes or processes that need to run periodically to perform some action. Long running processes run outside of a web server because they cannot be limited by timeouts imposed by the server and should not rely on active connection with the client. Web applications can process requests that take a long time to a Worker Role process.

In addition, a startup task can be used to install and configure application components and variety of services that user applications may need. This makes the Worker Role suitable when web server is other than IIS or framework is other than .NET. Azure applications are not restricted to the .NET Framework, but it is the only framework available by default.

IV. CONCLUSION

A detailed overview of different options for web services deployment in Azure as a cloud computing platform is given in this paper. Microsoft Azure offers a wide range of different options and services, including quick and easy creation, publishing and management of web services as well as all the other software solutions within the cloud. The deployment of web services in Azure can be performed in different ways using different options and settings from the service provider within the cloud, and the choice depends on the specific needs of the particular web services.

REFERENCES

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, *A view of cloud computing*, Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.
- [2] B. Wilder, *Cloud architecture patterns*, Sebastopol, O'Reilly, 2012.
- [3] T.Redkar, T. Guidici, *Windows Azure Platform*, New York, Apress, 2010
- [4] T. Dykstra, R. Anderson, M. Wasson, *Building Real World Cloud Apps With Windows Azure*, Microsoft, E-book, 2014.
- [5] R. Jennings, *Cloud Computing with the Windows Azure Platform*, Indianapolis, Wrox, 2009.
- [6] L. Franks, S. Higa, S. Machiraju, C. Martinez, V. Mizonov, W. Myers, R. Ramani, J. Roth, M. Simms, P. Salvatori, A. Skewgar, R. Squillace, P. Wickline, T. Young, *Windows Azure Prescriptive Guidance*, Microsoft, E-book, 2012.