

# Microsoft Azure Migration: Microsoft's .NET Community Websites Customer Solution Case Study



## Migrating Microsoft's ASP.NET and IIS.NET Community Websites to Microsoft Azure

**Customer:** Microsoft Corporation  
**Website:** [www.microsoft.com](http://www.microsoft.com)  
**Customer Size:** Over 100K employees  
**Country or Region:** United States  
**Industry:** Software Engineering

### Customer Profile

From its headquarters in Redmond, WA, Microsoft employs thousands of employees across the globe to provide customers with a variety of software, devices, and services.

### Target Audience

This case study is designed to help IT decision-makers and IT professionals determine how best to migrate an enterprise-level website from traditional web hosting to Microsoft Azure.

### Software and Services

- Microsoft Azure
- Microsoft Azure Platform
- Windows Server 2012
- SQL Server 2012

### Benefits

- Capable and reliable application platform
- Advanced cloud computing technology
- Reliable worldwide data center services
- High availability network and good connectivity
- Scalable architecture and pay per use
- Strong technical support and service level guarantee

For more information about other Microsoft customer successes, please visit:

[www.microsoft.com/casestudies](http://www.microsoft.com/casestudies)

Microsoft is an industry leader in software and technology, and hosts a thriving user community of developers and IT professionals through its [www.asp.net](http://www.asp.net) and [www.iis.net](http://www.iis.net) websites. By migrating to Microsoft Azure, Microsoft was able to cut in half the number of physical servers that were required when these community websites were hosted in a traditional data center. In addition, Microsoft is now able to quickly deploy new servers and add those servers to their virtual infrastructure in hours instead of days.

### Business Needs

Microsoft maintains two large community websites for the ASP.NET and Internet Information Services (IIS) technologies, and the primary websites for each of these technologies are located at the respective URLs of [www.asp.net](http://www.asp.net) and [www.iis.net](http://www.iis.net). Each of these community presences consist of several different collections of content types: official documentation, tutorials, informational videos, blogs, and forums.

These two web communities receive a considerable amount of annual web traffic; for example, in the calendar year for 2013, the ASP.NET community websites received 188 million page views, and the IIS.NET community websites received 83 million pages views. With this volume of web traffic, it is imperative that these two web communities maintain a high level of availability, and provide the means to scale

to more servers during periods of increased traffic; for example, during new product launches.

Microsoft contracts two third-party companies, OrcsWeb and Neudesic, to manage its community websites. OrcsWeb maintained the physical hardware in a traditional hosting environment, and Neudesic managed the development, daily operations, and maintenance of the websites, databases, and content. Adding new servers to the existing farm required several steps: defining the specifications for a new physical server, going through the requisition and purchasing process, adding new hardware to the server racks, then installing and configuring all software.



With that in mind, there were specific goals that Microsoft was trying to accomplish by migrating to Microsoft Azure:

- Reduce Total Cost of Ownership (TCO) for the server farm
- Reduced deployment time when adding new servers to the farm
- Reduce the number of servers in the farm by selecting virtual machines that will handle the load appropriately
- Demonstrate how a farm of high-traffic and highly-available websites can be migrated to Microsoft Azure

## Solution

Based on Microsoft's web farm design and business needs, migrating to Microsoft Azure was a logical choice. Microsoft Azure's virtual machines offer a variety of virtualized hardware choices, and the virtualized hardware can be easily modified if business needs change in the future.

For example, the number of CPUs and the amount of RAM for a virtual machine can be changed by selecting a different virtual machine size through the Microsoft Azure Management Portal or through PowerShell. In addition, new virtual machines can be added to the farm through the Microsoft Azure Management Portal in a matter of minutes.

Because the different virtual machine sizes offer virtualized hardware choices which exceeded the physical hardware configuration which was used to host the .NET community websites, it became possible to reduce the number of servers across the entire farm by choosing virtual machine sizes that are capable of handling more server load than was possible with the physical hardware.

The combination of effortless deployment and virtualized hardware possibilities make it easy to scale-up or scale-out the server farm for the .NET community websites based on changing business requirements.

In addition, the Microsoft Azure Service Level Agreement (SLA) for Microsoft Azure virtual machines is at least 99.95%, thereby guaranteeing sustained uptime for the virtual servers hosting the .NET community websites.

For additional information about Microsoft Azure, see the following web pages:

- [Virtual Machine Sizes](#)
- [Virtual Machine Pricing Details](#)
- [Service Level Agreements](#)

## Design Decisions

There are several reasons why Microsoft and Neudesic chose to use Azure Virtual Machines instead of Azure Websites during this migration:

- Since the websites were already hosted on physical servers, the migration to Azure VMs seemed the shortest migration path.
- There are approximately a dozen or so individual websites that make up the .NET community presences. These websites are not hosted on standalone servers – there is a lot of cross-site cohesion, and these websites have dependencies on custom Windows services and scheduled tasks; it would have required a great deal of rewriting to accommodate migrating these websites to Microsoft Azure Websites.
- The .NET community websites utilize a large number of custom URL Rewrite

rules; by deploying the Application and Request Routing (ARR) module for IIS within the server farm, Microsoft could continue to use these complex sets of rules and server managers were able to log into the ARR servers and make changes.

- Microsoft uses the .NET community websites as a test platform for new operating systems; for example, the [www.iis.net](http://www.iis.net) websites are usually running on pre-release software, which allows Microsoft the opportunity to test upcoming operating systems in a live environment.

Another decision point was whether to host the web farm's databases in a virtual machine, or in SQL Azure, or to use an on-premises SQL server. The final decision was to host SQL on virtual machines, and Microsoft chose this design based on the following reasons:

- Several of the databases have legacy features that will not port directly into SQL Azure, which would have required redesigning databases.
- Using an on-premises SQL server could have been accomplished through a Virtual Private Network (VPN), but that would have reduced performance and introduced a possible point-of-failure.

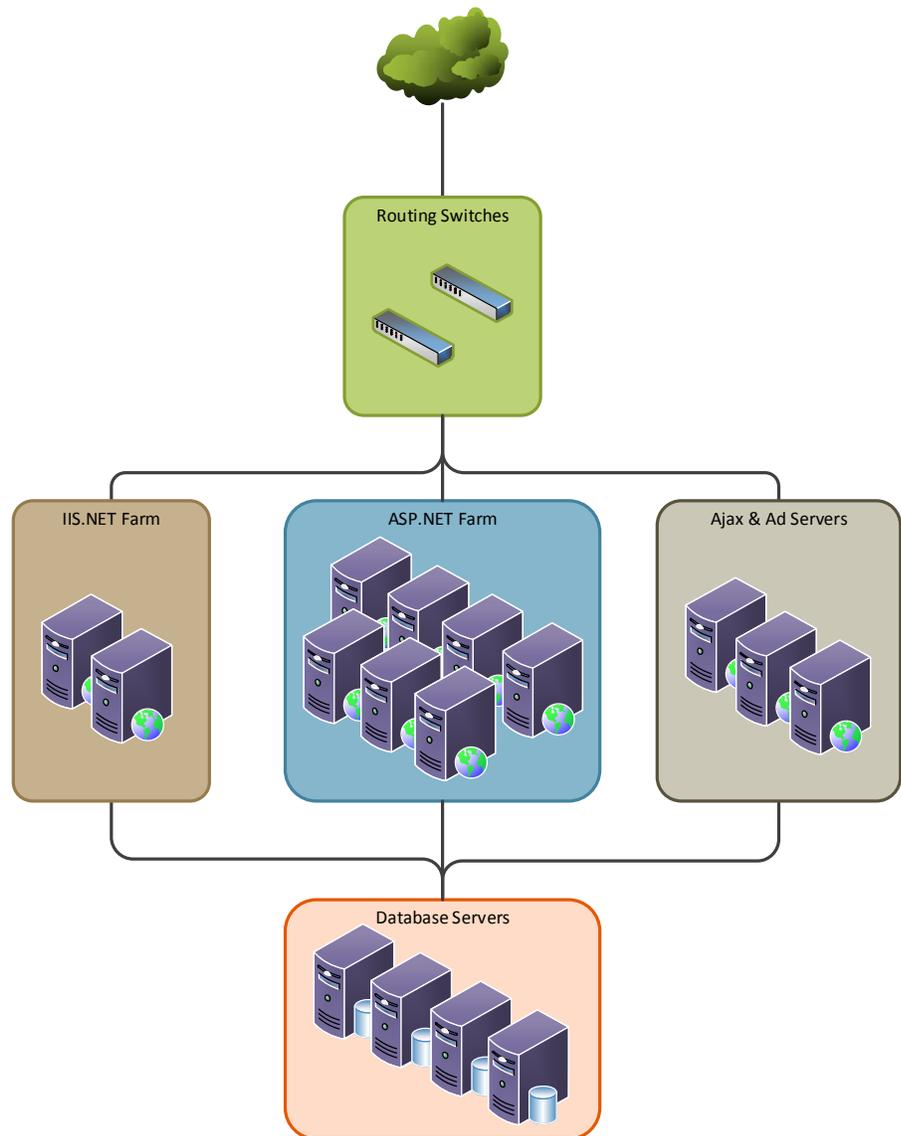
## Network Architecture Prior To Migration

Prior to the migration to Microsoft Azure, the network architecture for Microsoft's ASP.NET and IIS.NET community websites consisted of a large farm of 17 physical servers, two routing switches, and one virtual machine. These servers were organized into several smaller farms based on functionality and purpose.

In order to show the reduced number of servers that were required to host the ASP.NET and IIS.NET websites after the migration, the following diagrams and descriptions will describe the network design and hardware requirements for each of the servers.

### Server Farm Overview

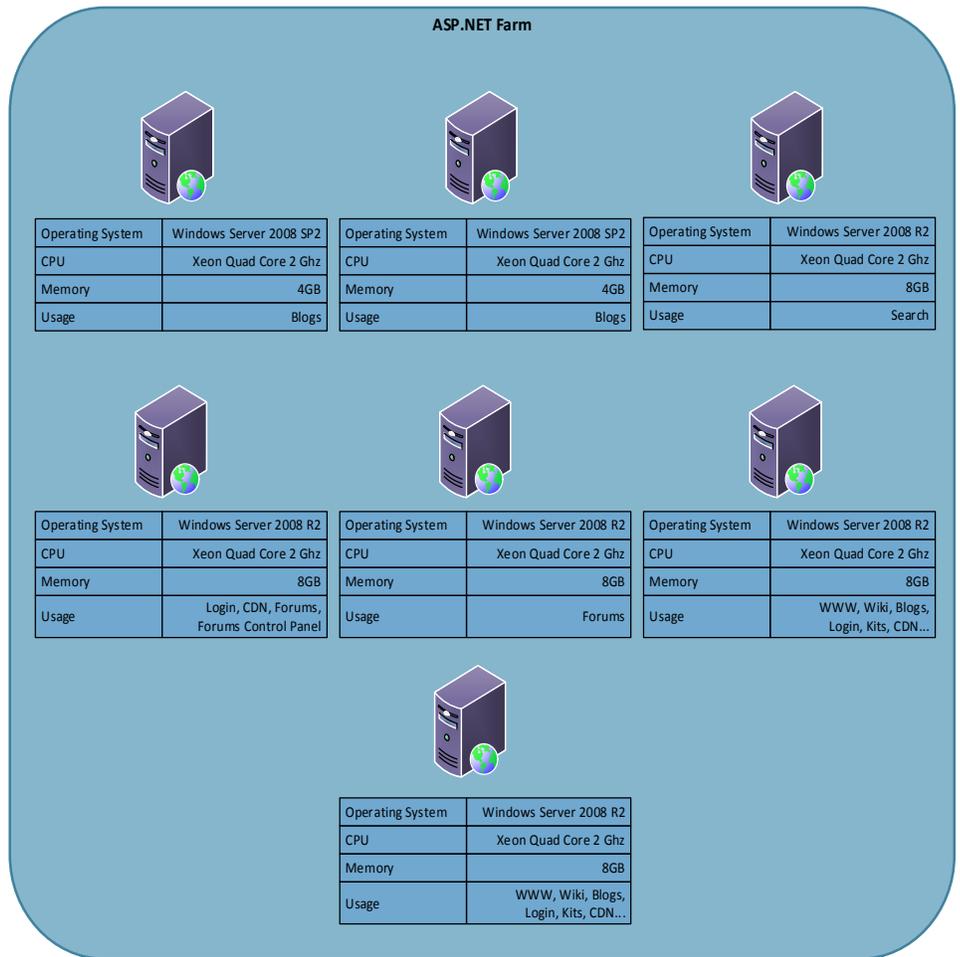
The overall topology of the server farm prior to the migration looked like the illustration on the right.



### ASP.NET Sub-Farm

The largest of the individual feature farms was for the ASP.NET community websites, and this sub-farm consisted of 7 physical servers.

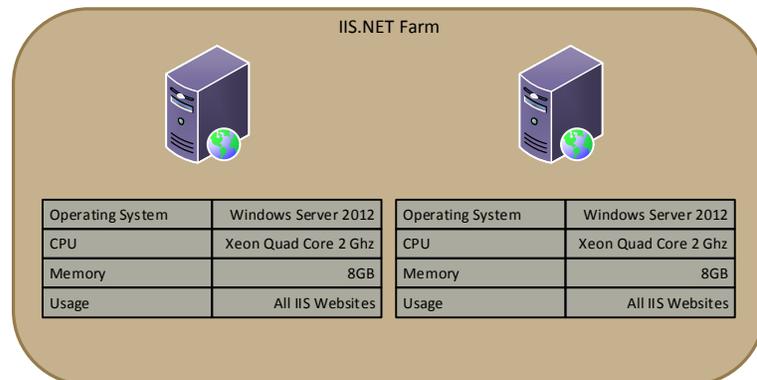
Each server played one or more roles within the ASP.NET farm as illustrated in the diagram on the right, and the hardware and software configuration for each server is listed.



### IIS.NET Sub-Farm

The smallest of the individual feature farms was for the IIS.NET community websites; this sub-farm consisted of 2 physical servers (for redundancy).

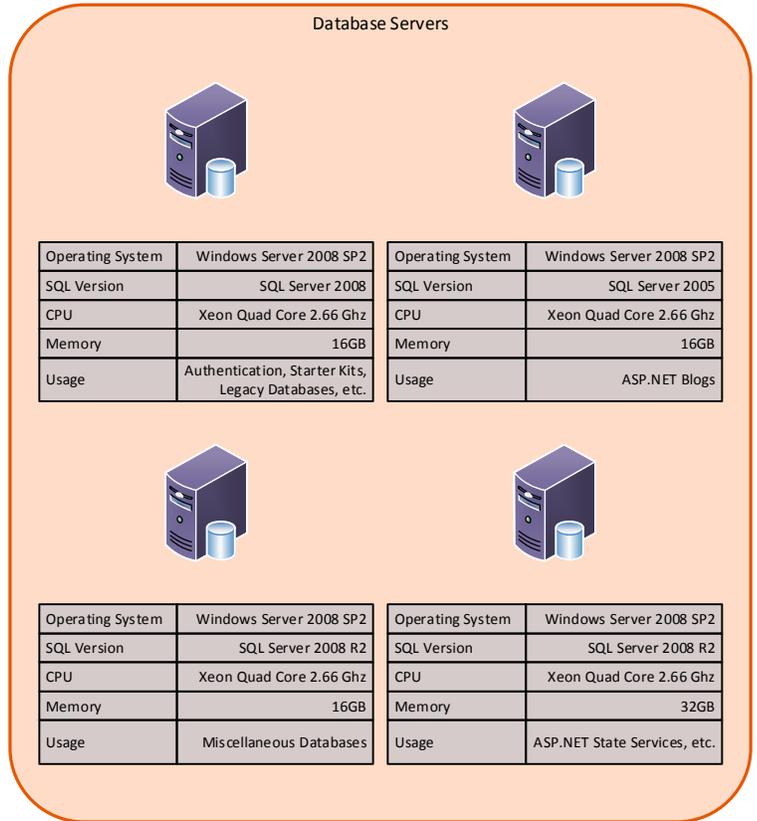
As illustrated in the diagram to the right, each server in the IIS.NET farm provided access to all of the community sites: www, forums, blogs, etc. As with the diagram for the ASP.NET sub-farm, this diagram lists the hardware and software configuration for each server.



### Database Server Sub-Farm

The ASP.NET and IIS.NET websites use several databases for a variety of purposes, and the sub-farm for these servers consisted of 4 physical servers.

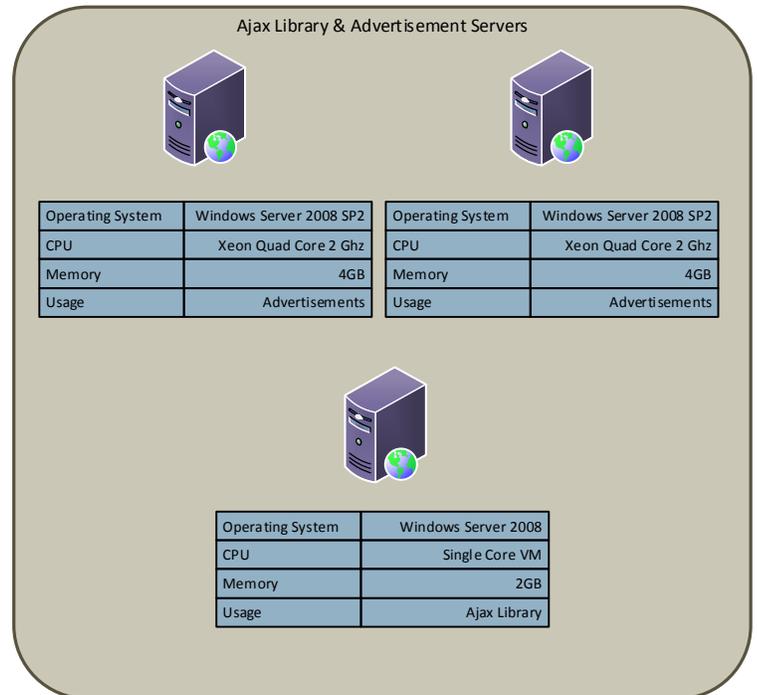
The illustration to the right lists the purpose for each server in the farm, as well as the hardware & software configuration for each server.



### Ajax Libraries and Advertising Sub-Farm

The remaining sub-farm contained the servers that are used for hosting the Ajax Libraries and advertising functionality for the .NET community websites.

As illustrated in the diagram to the right, this farm consisted of 2 physical servers and a single virtual machine.



## Migration Process

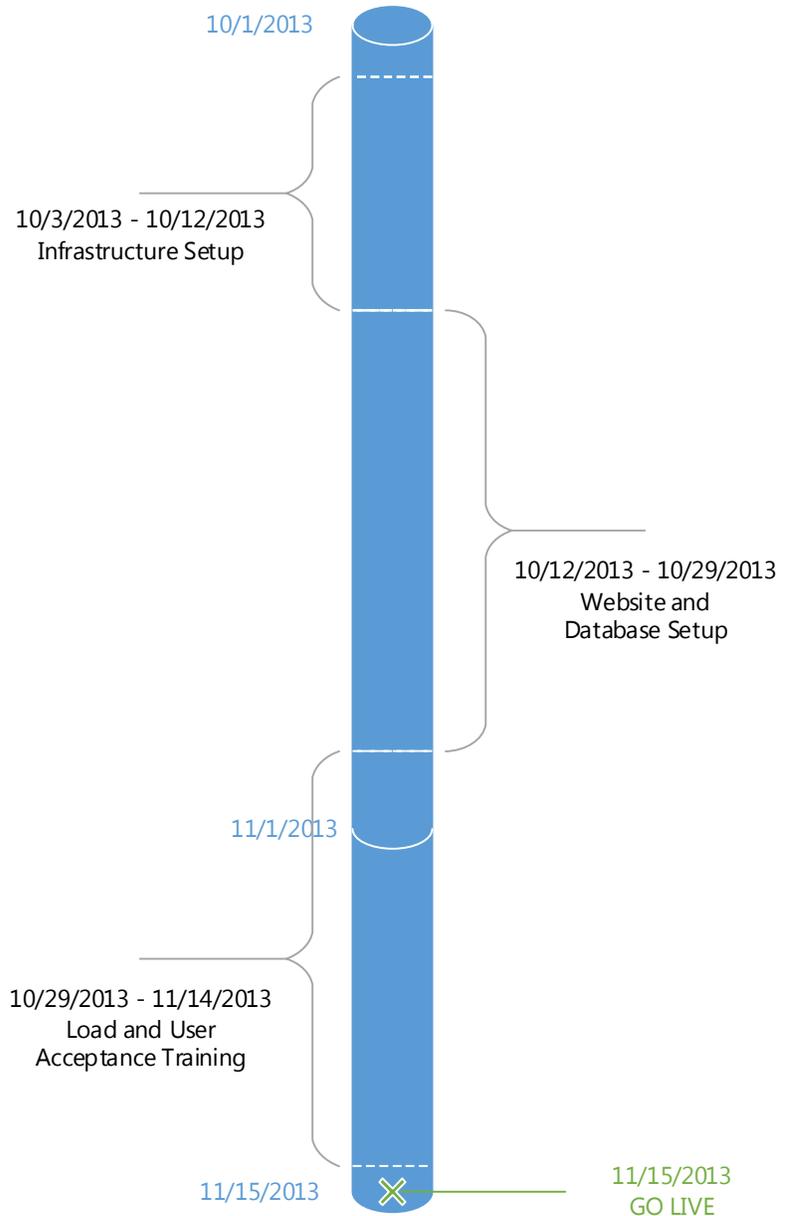
### Overview

The migration process from a traditional hosting environment to Microsoft Azure took place over a six-week period from October through November of 2013. All of the work was accomplished by two third-party companies: OrcsWeb and Neudesic.

The following list describes a basic, high-level overview of migration activity. (Details of migration activities will be covered later in this section.)

| Timespan  | Details                          |
|-----------|----------------------------------|
| 1.5 weeks | Infrastructure Setup             |
| 2.5 Weeks | Website and Database Setup       |
| 2 Weeks   | Load and User Acceptance Testing |

Once all of the websites were migrated to virtual machines in Microsoft Azure and sufficient testing had been completed, all that was required on the "Go Live" date was to update the DNS records for the .NET community websites to point to the Microsoft Azure virtual machines and to monitor the environment in case of any post-migration issues.



## Infrastructure Setup

The initial phase of the migration was to setup the infrastructure in Microsoft Azure, and the first part of this process was for OrcsWeb to configure the array of virtual machines. To do so, OrcsWeb began with a stock virtual machine from the Microsoft Azure catalog, and then they used their custom build scripts to configure the operating system. Once they had an initial system configured, they saved the virtual hard drive (VHD) file as a baseline image to set up the remaining servers throughout the farm.

The next step was to set up the individual servers and establish the network hierarchy, which is illustrated in the *Network Architecture After Migration* section of this case study. The roles for each server were identified during the preliminary planning phase, so all that needed done for each server was to install the necessary software for each virtual machine in the farm. For example, several of the servers were tasked as web servers, which required that Internet Information Services (IIS) be installed, other servers were tasked as database servers, which required that SQL Server be installed, and two servers were tasked for load-balancing, which required that IIS be installed with the Application and Request Routing (ARR) module.

The network topology consists of a single virtual network where all of the servers are deployed within a single Active Directory site with a single domain controller.

## Website and Database Setup

In the second phase of the migration process, Neudesic transferred the

configuration, databases, and web content to the new virtual machines.

The .NET community websites use several add-on modules for IIS, such as the Application and Request Routing (ARR) module and the URL Rewrite module. With that in mind, Neudesic installed and configured all of the required IIS add-on modules on the appropriate web server virtual machines. That being said, Neudesic did not use the original *applicationHost.config* files from the physical servers; since the roles for several of the servers machines were being combined into a smaller number of virtual machines, it made sense to create new configuration files from scratch.

Neudesic migrated the databases by backing up the existing SQL databases and jobs on the physical servers, transferring the backup files to the appropriate virtual machines over FTP, and restoring the backups into SQL Server on the virtual machines. Once that was accomplished, Neudesic created all of the necessary SQL logins and user accounts.

Migrating the content was a similar process to the SQL database migration; all content files were compressed into archive files, transferred to the appropriate virtual machines via FTP. Once the content was in place, changes were necessary to the website configuration in order to address several changes:

- The names of the SQL servers were changed, so all of connection strings for the community websites required modifications.
- The original physical servers were configured with two hard drives: one drive (C:\) was for the operating system and the other drive (D:\) was

for content. However, the virtual machines were configured to use a single hard drive for the operating system and content, therefore any paths which were specified in source code needed to be updated.

- IP addresses were changed as a result of the migration, so the bindings in the *applicationHost.config* files required modifications.

Once all of the SQL data, website content, and IIS configuration settings had been migrated, Neudesic set up ARR on two IIS servers to perform load-balancing for the server farm.

## Load and User Acceptance Testing

The third and last phase of the migration process was to test all of the websites thoroughly before deploying the new virtual machines into production. Testing the websites was a critical part of the migration process, because it was essential that there should be no performance degradation after migrating to Microsoft Azure.

The testing process was broken into two parts:

- Load Testing
- User Acceptance Testing

Neudesic used [Load Storm](#) for load testing, and they determined the amount of traffic based on the following logic: first the determined what the average level of normal traffic was across all websites, which was based on the number of requests per minute. Once they had determined the average, they multiplied that by 5 and used the higher traffic level to simulate web requests. Despite the

reduced number of web servers deployed in the farm, the virtual machines were able to handle the increased level of with no performance degradation.

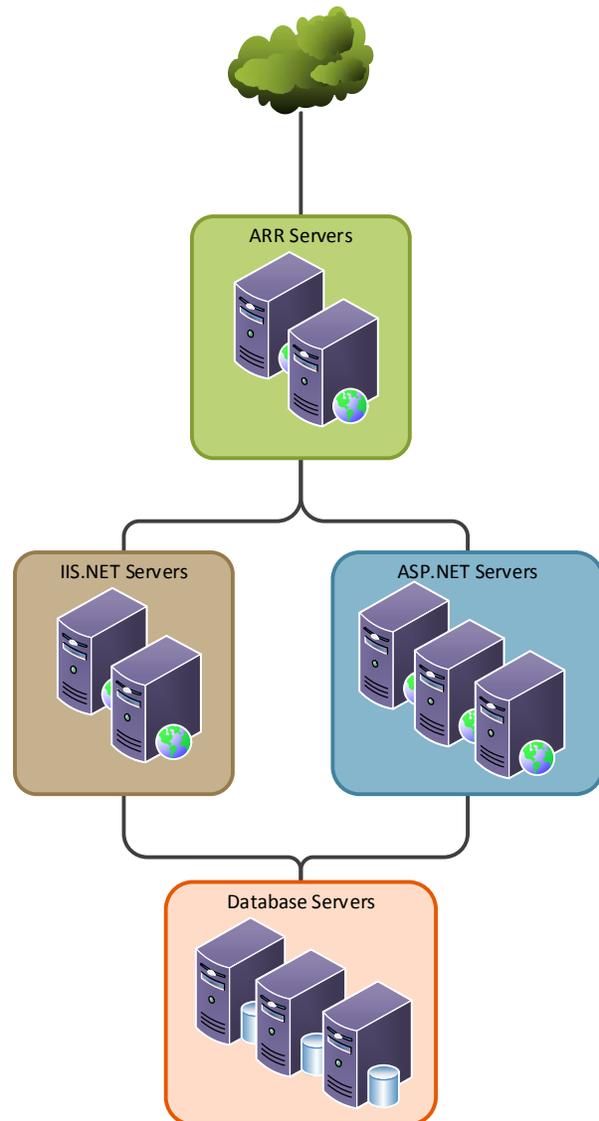
User Acceptance Testing was accomplished over several days by a virtual team of individuals from Neudesic and Microsoft. The team performed manual verification of all website functionality, and any bugs which were discovered by the team were fixed before the new websites were deployed into production.

## Network Architecture After Migration

After the migration to Microsoft Azure had been completed, the network architecture for Microsoft's ASP.NET and IIS.NET community websites consisted of a reduced farm of 10 virtual machines. As before, these servers were organized into several smaller farms based on functionality and purpose.

### Server Farm Overview

The topology of the server farm after the migration to Microsoft Azure now resembles the illustration on the right.

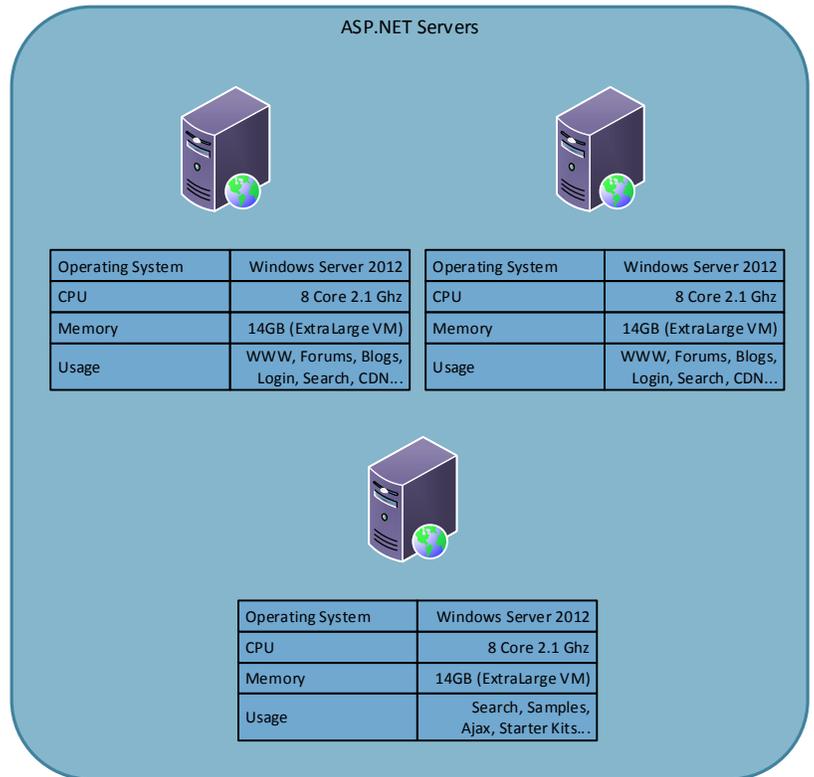


### ASP.NET Sub-Farm

One of the most-significant improvements in the post-migration server farm is the reduction of the number of servers that are required for the ASP.NET sub-farm; the number of servers was reduced from 7 physical servers to 3 virtual machines, and the need for the 3 additional servers for hosting the Ajax libraries and advertising features has been alleviated. In the end, 9 physical servers and 1 standalone virtual machine were replaced by 3 virtual machines in Microsoft Azure.

Each virtual server continues to function in one or more roles within the ASP.NET farm as illustrated in the diagram on the right, but the load is more-evenly distributed across the farm due to the hardware and software capabilities of each virtual machine.

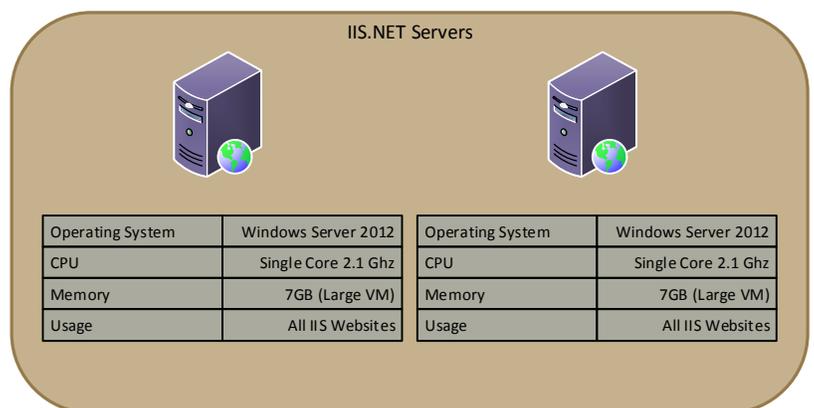
The illustration on the right describes the virtual machine size and operating system information for each of the virtual machines in the ASP.NET sub-farm.



### IIS.NET Sub-Farm

The IIS.NET community websites were hosted on two physical servers for redundancy prior to the migration, and after the migration they continue to be hosted on two virtual machines.

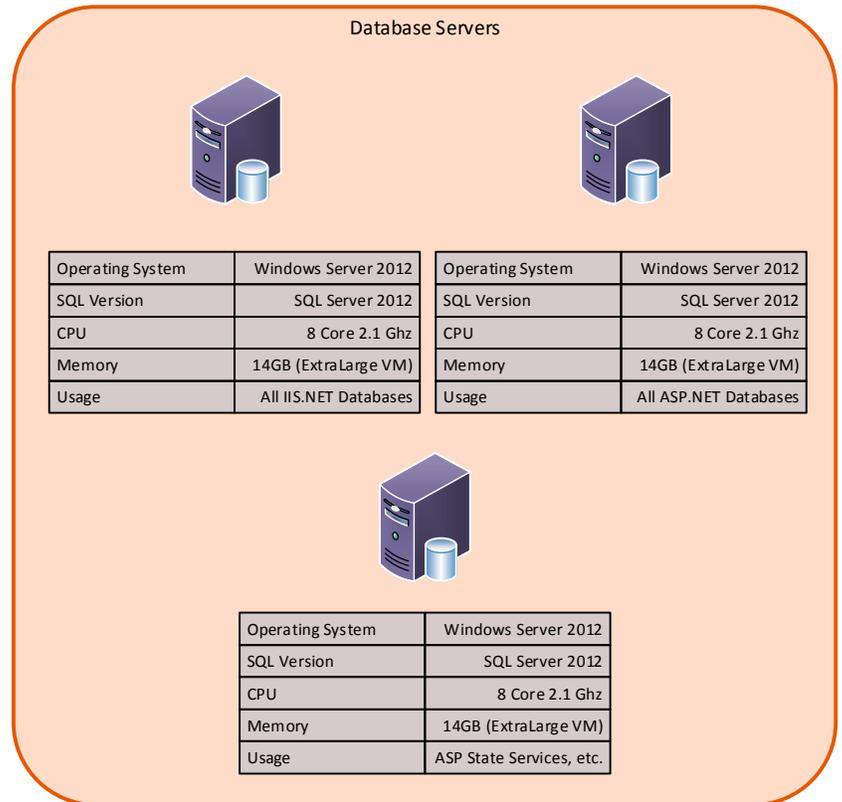
As illustrated in the diagram to the right, both virtual machines in the IIS.NET farm provide access to all of the community sites for IIS: www.iis.net, forums.iis.net, blogs.iis.net, etc. As with the diagram for the ASP.NET sub-farm, this illustration lists the virtual machine size and operating system information for each server.



### Database Sub-Farm

Continuing the resource savings from the ASP.NET sub-farm, the number of database servers in the database sub-farm was reduced from 4 physical servers to 3 virtual machines, and the roles for each database server are distributed more equitably across the farm.

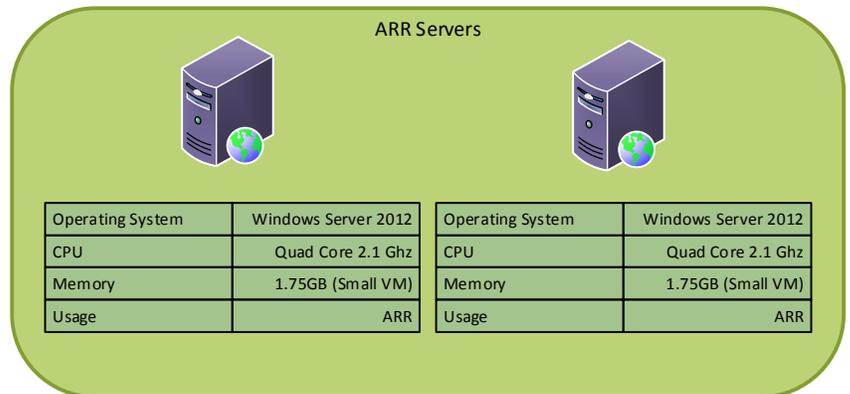
The illustration on the right lists the virtual machine size and operating system information for each virtual server in the database sub-farm.



### ARR Sub-Farm

Another significant difference between the network topology before and after the migration was to remove the two routing switches from the architecture, and then to replace those switches with virtual machines which are running Microsoft's Application and Request Routing (ARR) for dynamic load-balancing.

Microsoft's ARR provides a wealth of load balancing features for web farm administrators, and more information can be found on Microsoft's [ARR download page](#).



## Benefits

Migrating the .NET community websites to Microsoft Azure addressed all of the business needs that were identified by Microsoft.

### Reduce Total Cost of Ownership (TCO)

Microsoft was able to reduce the monthly hosting and operating costs by 45% to 50% as the total number of devices deployed across the server farm was significantly diminished. In addition, Microsoft was able to reduce some of the administration overhead by offloading the management of physical hardware for the server farm from one of the third-party companies to Microsoft Azure.

### Reduce the Number of Servers

Microsoft Azure's virtual machines provide powerful computing capabilities, which allowed Microsoft to cut number of servers nearly in half:

- Before the migration:
  - 15 physical servers
  - 2 routing switches
  - 1 virtual machine
- After the migration:
  - 10 virtual machines

### Reduce Deployment Time

Microsoft Azure makes it simple to add new servers to the farm. Instead of the traditional hassles of determining the hardware requirements for physical servers, Microsoft Azure provides a list of [Virtual Machine Sizes](#) from which to choose, and the virtual machine size can be adjusted as business needs change.

### Demonstrate the migration of a high-traffic/highly-available web farm to Microsoft Azure

The migration of Microsoft's .NET Community Websites to Microsoft Azure was beneficial for all of the business-related

Page 11 of 11

reasons which were listed earlier, but this migration also provided an excellent proof-of-concept for the processes and procedures that other large enterprises might follow when migrating their web farms from a traditional hosting scenario to Microsoft Azure. In addition, Microsoft Azure's Service Level Agreements guarantee at least 99.95% uptime, which ensures that Microsoft's .NET community websites will always be available to Microsoft's customers.

## Lessons Learned

There were several lessons learned throughout the migration process that might be of benefit to IT professionals who are looking at a similar migration:

- Good planning is essential before migration. Much of the migration process went smoothly because the individual server roles and naming conventions had been identified ahead of time. These details were extremely beneficial because the number of servers was being reduced instead of mirroring the original physical infrastructure.
- A time-consuming part of the migration was the network configuration. The virtual network could not be modified once it had been created, so any changes to the virtual network would require recreating any virtual machines on the network. Once again, good planning can alleviate this type of frustration.

## What's Next?

There are several changes that are still the planning phase for the future of Microsoft's .NET community websites:

- Some of the websites will be moved to [Microsoft Azure Websites](#), which provides the following built-in functionality which is easier to manage than using a virtual machine:
  - [Backup and Restore Features](#)
  - [Staged Publishing](#)
  - [Traffic Manager for Load Balancing](#)
- In addition, the databases will be moved to [Microsoft Azure SQL Databases](#) for similar reasons.

For more information about Microsoft Azure, see the following website: [azure.microsoft.com](http://azure.microsoft.com)

© 2014 Microsoft Corporation. All rights reserved. This case study is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. Microsoft Azure and Microsoft logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.