



MIGRATING FROM ON-PREMISES TO **WINDOWS AZURE CLOUD** -AN OVERVIEW

IT leaders and strategists are reasserting the value of information and communication technologies in enterprises with the power of Cloud. Windows Azure's ability to deploy scalable solutions rendered "**as a service**" is bringing in a paradigm shift to organizations that encourage Cloud.

INDEX

AN INTRODUCTION TO WINDOWS AZURE	01
WHY MIGRATE	
MIGRATION: AN OVERVIEW	
SPECIFIC MODELS OF AZURE MIGRATION	02
IAAS MIGRATION	
PAAS MIGRATION	
SOFTWARE AS A SERVICE (SAAS)	
DATABASE MIGRATION	
SHAREPOINT MIGRATION	
OFFICE MIGRATION	
KEY CONSIDERATIONS	04
PRE-REQUISITES FOR MIGRATION	
COMPATIBILITY ANALYSIS	
THE MIGRATION PROCESS	07
MIGRATING DATABASES TO AZURE	
CONCLUSION	
ABOUT SUYATI	09

AN INTRODUCTION TO WINDOWS AZURE

Enterprises would love to host their application or software on a platform that gives them the best ROI including high performance (speed), scalability, and affordability. Windows Azure is a Microsoft cloud platform that bestows you with these three critical success factors essential for the survival of any cloud program. Microsoft **Windows Azure** is a platform for the public cloud, used to build scalable cloud based web applications that store data in Microsoft datacenters.

Windows Azure offers a flexible and open platform which facilitates to quickly build, deploy and maintain the application universally. By using Azure, you can create a new application as well as migrate an existing application from on-premises to cloud. This White Paper analyzes the necessity of migrating to Windows Azure cloud. We will also discuss the key considerations behind this migration and try to provide some insights on some of the specific models of Azure migration like migrating to **IaaS, PaaS, SaaS**, database migration, SharePoint migration, and office migration.

WHY MIGRATE

The cloud offers several benefits, such as anytime availability, easy scale-up and scale-out in a pay-as-you-go model, and freedom from buying or maintaining the underlying infrastructure. Microsoft Windows Azure platform offers certain advantages over and above such general advantages.

With Windows Azure, developers may focus on their core task of development, without having to worry about the infrastructure. **Windows Azure offers SLA 99.95%** for the hosted applications. Developers having .NET skills can develop and migrate applications on Azure by learning Azure SDK.

Apart from leveraging the various cloud based benefits, there are technical reasons to consider migration. For instance, Windows Server 2003 has reached end of support, and developers who host their websites on an IIS server or Windows Server 2003 would anyway have to migrate, and migration to the cloud offers the best option.

MIGRATION: AN OVERVIEW

Migrating legacy .NET based on-premises application to Azure **App Service Web Apps** is easy. The Azure Web Apps Migration Assistant in fact automates the migration process. However, the process requires a methodological approach.

Migration to the Azure cloud mainly occurs **in two approaches**: the Infrastructure-as-a-service (IaaS) approach and the Platform-as-a-service (PaaS) approach. In addition, it is also possible to move the database alone to Azure cloud. Let us have a glimpse at some of the different models of Azure migration.

SPECIFIC MODELS OF AZURE MIGRATION

Azure migration can happen in a number of ways. Few of them are listed here:

IAAS MIGRATION

With the Infrastructure as a Service (IaaS) approach, Microsoft Azure provides the runtime infrastructure, such as a virtual server and network connectivity, and the subscriber installs their own OS, services and applications on such infrastructure. The subscriber runs their selected servers in Microsoft datacenters. Subscribers may still choose from a range of preinstalled OS such as Windows Server and Linux, offered by Azure, and also leverage peripheral services, such as Azure Active Directory authentication services, Traffic Manager global routing, Azure storage queues or Service Bus messaging service, and more. The selected Azure Virtual Machine (VM) may also come with optional SQL Server or MySQL database preinstalled.

The IaaS approach towards hosting applications essentially replaces the Ethernet cable in the datacenter with an Internet connection that links to Azure. This approach allows migrating VM to the cloud without changing application code, and all deployment, management, testing, and monitoring systems working just as before. The IaaS approach is also useful when the subscriber does not have access to the source code of any application, requires non-standard OS configuration or needs to set specific file permissions.

PAAS MIGRATION

The Platform-as-a-Service approach is a much more comprehensive approach to migration compared to IaaS. With this approach, Microsoft Azure manages everything including the OS, database, and runtime platform. The cloud user takes care of only the application code and its interaction with peripheral services.

The Azure hosting platform offers simple management and deployment capabilities, supports almost all programming languages in vogue, and integrates directly with many source control systems. The Cloud Services approach offers better control for the user over the platform, including the capability to run various types of roles and integrate caching directly. Users also gain a wide range of configuration options, and the option to deploy separate Web and worker roles.

Gartner recognizes Windows Azure as a leader in the Platform as a Service (PaaS) offering.

SOFTWARE AS A SERVICE (SAAS)

The simplest way to migrate a site or web application to Azure is to install it to Azure web sites. This approach might require to make a few changes to the application. The software code may be deployed at Microsoft Team Foundation Server (TFS), GitHub, or any other source code repository.

Based on your business needs and your hosting budget, you can decide whether to host on a shared Web server or on a reserved instance where you can ensure the performance and manage the number of occurrences to meet the demand.

DATABASE MIGRATION

Almost all business applications use data, stored in databases such as SQL Server or a Relational Database Management System (RDBMS).

Microsoft Azure Cloud offers several database migration possibilities.

Azure SQL server hosted database is capable of being used exactly the same way as on-premises SQL Server. It is best suited for generated data storage and migrating applications that use SQL server. The migrated applications work without any changes in code, unless the code requires advanced capabilities of the SQL Server, such as free text search, distributed queries, CLR programmability procedures, XML handling capabilities, and more.

Azure offers preconfigured VMs **containing SQL Server or MySQL**, which works best when migrating an application using relational database, and for PaaS migrations. Such database server hosted in a VM is an attractive solution for storing large volumes of data and deploying multiple databases. It offers the flexibility to choose the correct size of VM, and also allows us to make use of multiple VMs to implement database failover or a shared data store. It is possible to install almost any other type of data store that runs on Windows or Linux in a VM running in Azure.

Azure also offers schema-less storage options outside the relational SQL offerings.

- Azure HDInsight service is based on open source Hadoop technologies
- Azure storage tables allow you to store application data
- Offers storage queues to pass messages between application components
- Storage drives rather act like a traditional disk-based filing system
- Azure storage blobs are best suited to store unstructured data such as documents, binary data, XML files and images.

Azure storage options are comparatively cheaper than using a VM-hosted database server or SQL Database. But migrating to such databases requires to rewrite the data-access code.

Azure also offers the option of keeping the database on-premises and communicating with the database server over a virtual network or through messaging. Connection to the cloud-hosted database from the cloud-hosted server is just as if the database and application were located in your own datacenter.

SHAREPOINT MIGRATION

The Azure cloud offers a good environment to host SharePoint Server solution. Hosting SharePoint 2013 on Windows Azure Virtual Machines facilitates rapid deployment and infuses scalability and reliability to the application.

Migrating SharePoint 2013 to Windows Azure is straightforward, for SharePoint Server 2013 prerequisites come preinstalled. Simply sign-in to the **Windows Azure Portal**, and navigate to the SharePoint Server 2013 Trial option, to select an appropriate virtual machine size, such as A6, A7, or Extra Large, to use with SharePoint server. Convert the trial version to a permanent one by entering the license key.

OFFICE MIGRATION

It is possible to migrate on-premises Office files to the Azure cloud using Office 365 import services.

Access the storage account key and URL from the IMPORT page of the Office 365 admin center, and transfer the data into Office 365 by either using the in-house network to upload files into Azure blob storage, or in case of very large volumes, physically ship drives to Microsoft for manual transfer.

KEY CONSIDERATIONS

There are several pre-requisites to consider before moving an application to Windows Azure.

Cost:

At the onset, it is important to do a cost-benefit analysis. Migration to the Azure cloud brings in a host of benefits, but it also comes at a price. Abandoning the migration half way owing to resource crunch can lead to a disaster. By doing a proper cost-benefit analysis, organizations can plan on how to proceed by supporting the cost and more importantly how the perceived benefits exceed the costs.

Application compatibility:

Windows Azure comes with architectural demands, and it is important to check whether the application to be migrated satisfies such architectural requirements.

External / internal dependencies:

If the application to migrate has external dependencies, not all such dependencies may be accessible through Azure. By analyzing and identifying those which are not accessible through Azure cloud, we can create workarounds for such dependencies.

Application integration:

Similar to external and internal dependencies, if the application for migration is integrated with other on-premise applications and shared services, appropriate provisioning has to be made for such integrations post-migration.

Third-Party Framework and Class Libraries:

Azure does not support all third-party framework and class libraries. If any such framework or class library is in use, the library would require some modifications or may even have to be rewritten. However, if the migration is in the IaaS model, code changes are not required. Rather, the server image in entirety will be migrated to the Windows Azure virtual machine.

Compliance requirements:

The prevalent enterprise compliance and regulations that govern important aspects, such as location of data, whether the data can be moved or stored outside the enterprise's realm and more may be disrupted when the application is migrated to the cloud. Evaluate all such requirements, and make sure all compliance requirements will be fulfilled even after migration.

Scalability / Elasticity:

Azure supports scalability, whereas most on-premises infrastructure do not do so. Identify whether the application design of the app being migrated supports scalability, to leverage the scalability feature offered by Azure.

Application class:

Evaluate the importance or classification of the application under migration. Critical line-of-business applications demand high availability, and appropriate provisions have to be made in Azure.

UI analysis:

It may not always be possible to migrate the UI interface directly to Azure using the PaaS model. Some amount of re-engineering works may be required to modify existing web application code to use Azure SDK, and some non-web applications may have to be kept on-premises as well.

Database compatibility:

Analyze whether the existing database is the best fit post-migration. If not, the data would have to be migrated to a new database in the Azure cloud.

Security:

Security is an important causality when applications are migrated. Make sure to provision for the same level of security as before, and enhance the security levels to make it more suitable for the cloud based environment after migration. Consider data security, authentication, authorization, and all other applicable security levels.

Authentication:

Analyze the authentication mechanism in the legacy application, and evaluate whether the same will be compatible with the Azure cloud. Azure Web App supports Anonymous Authentication by default and Forms Authentication, but does not support basic authentication.

Application maintenance / management:

It is also important to identify how logs are maintained and where they are stored.

IMPORTANCE OF COMPATABILITY ANALYSIS

Having decided on the **broad parameters** of migration, sufficient attention needs to be paid to the technical aspects of migration, to ensure technical competency. This can be attained by performing a compatibility analysis.

Here are some of the common issues which often flare up:

Port Bindings: Azure Web App supports only Port 80 for HTTP and Port 443 for HTTPS traffic. It ignores other port configurations and routes to 80 or 443, regardless of the specification in the legacy on-premises app.

Authentication issues: Azure Web App supports only Anonymous Authentication by default and Forms Authentication. It is possible to use Windows Authentication only by integrating with Azure Active Directory or ADFS. The Azure cloud does not support any other form of authentication, including Basic Authentication. A workaround to developing a new authentication mechanism is to authenticate users with Windows Azure Active Directory Access Control (ACS), when they gain access to a web application from Facebook, Microsoft, Google, or Yahoo.

Global Assembly Cache (GAC): Azure Web Forms do not support GAC. If the application is being migrated, references usually deploy assemblies to the GAC, and it becomes necessary to deploy to the application bin folder in Web Apps.

Application Pools: Azure Web Apps run each parent site and its child applications using the same application pool. The migration throws an error if the application under migration is having multiple child applications utilizing several application pools. To migrate, it is necessary to either consolidate all child applications into one application pool with common settings or migrate each application to a separate web app.

Component Object Model (COM) Components: Azure Web Apps do not allow the registration of COM Components. Any COM Components in the legacy application need to be re-written in managed code and deployed with the website or application, before migration takes place.

Internet Server Application Programming Interface (ISAPI) Filters: Azure Web Apps support the use of ISAPI Filters. But to migrate, it is necessary to deploy the DLLs (Dynamic Link Library) with the web app, and register the DLLs using Web.config.

Configuration changes: When the application being migrated accesses information from physical storage, it needs to be migrated to Azure storage / Windows Azure SQL /CDL database. The process displays an error if hard-coded physical disk paths are present in the applications. Azure applications do not support hard-coded physical disk or network access values.

Session and Identity provider: Make sure to include the session and identify provider (ADFS / ACS information) in the web configuration file before starting the migration process.

Session state management: In Azure, each web role occurrence operates on its own VMserver and is configured behind a load balancer. The ASP.NET session state is not automatically shared across cases in this load balanced environment. There are various approaches to address this incompatibility.

- **Use a Table storage session provider:**

- A custom provider compiled into class library (.dll file)
 - Enables developers to store session state inside Windows Azure table storage
 - Relatively affordable, well tested and ready for consumption
 - In most cases, no re-engineering work is required

- **Use the default session state management- Inproc:**

- Inproc session state functions only for a single instance set in load balanced environments such as Windows Azure
 - If more than one instance is used, it might result in inconsistency

- **Use Windows Azure SQL database session provider:**

- Essentially a subset of SQL server
 - It can also be used as storage for session state
 - Can be derived from SQL server session provider with a few modifications
 - Cost effective when used with the existing database

Third-Party Library References: If the third-party library or content references are not included as part of the Azure package, the migration report generates an error. Make sure to include the same before the start of the process.

THE MIGRATION PROCESS

Let us take a look at the steps involved in the migration to Azure cloud:

1. A linked server communicates with other enterprise databases to fetch data that resides external to the application.
2. Azure performs the readiness analysis and provides the assessment report, which lists out the issues that require resolution.
3. After resolving **all the required items**, click “Begin Migration.”
4. On being redirected to Azure, you can log in with the account that has the active Azure account running.
5. Choose the tenant account, Azure subscription and region to use for the migrated Azure web apps and databases.
6. On being prompted, choose to migrate from the local server, or enter the credentials of the remote server. The latter mostly comes into play when migrating from an IIS server.

It is essential to purchase an Azure account upfront before starting the migration process. However, it is also possible to sign up for a free trial before investing in a full blown subscription.

7. At the backend, the migration tool inspects the server configuration, such as Sites, Applications, Application Pools, and dependencies, to identify candid websites for migration, and effects the migration.
8. The next screen offers the option to make changes to the default migration settings.

The major decisions to be taken at this stage include:

- Select the websites to migrate. The migration tool would list all available websites.
 - Whether to use an existing Azure SQL Database or create a new Azure SQL Database, and configure its credentials
 - Define names for the Azure web apps and their linked SQL databases
 - Customize the global settings and site-level settings
9. The Enable Azure Active Directory checkbox in custom settings integrates the Azure web app with Azure Active Directory (the Default Directory).
 10. After making the desired changes, clicking on “Create” would start the migration process. The migration tool will create the Azure SQL Database and Azure web app, and publish the website content and databases.
 11. The webpage displays the **progress of the migration**, and at the end, generates a summary screen that details the sites migrated, success, and links to the newly-created Azure web apps. The migrate web apps may be controlled by logging in to the Azure app service.

On encountering errors during the migration process, the summary page will indicate the failure and rollback the changes.

The best practice is to migrate components of an application in an incremental, independent fashion, and ensuring that the application works with no internal (application) or external (network) issues at each stage.

Only websites and databases would migrate from the remote server to Azure. Components such as SharePoint, front page server extensions (FPSE), FTP, SSL certificates will not be migrated.

MIGRATING DATABASES TO AZURE

The data for applications hosted in Azure may be stored in three ways:

- In Windows Azure SQL Database, a cloud based relational database built on SQL Server technologies.
- As static content to Windows Azure Storage (Blob) or XDrive.
- As message queues to Windows Azure queue storage / service bus queue.

When a desktop or on-premises application is migrated to Windows Azure, any SQL Server databases used by the application are migrated to SQL Database. However, developers and IT administrators still need to migrate data from SQL Server to Windows Azure SQL Database.

The SQL Database Import/Export Service tool facilitates high-volume asynchronous transfer of data between a SQL Database in the cloud and a BACPAC file stored in Windows Azure blob storage.

The **developer may generate** a BACPAC file from an on-premises SQL Server database, and use the Windows Azure Management Tool (MMC) or Azure Storage Explorer to copy it to a container in a Windows Azure storage account. The developer may finally use the import/export service to import that BACPAC into SQL Database.

To conclude, moving to Azure will help you minimize costs, reduce maintenance requirements, and enhance performance. Moving to Azure can be relatively faster and easier if you take into account your business requirements and properly explore the features that Azure has in it for you.

ABOUT SUYATI

Suyati provides marketing technology and integration services for companies that wish to combine the best of breed solutions and create a unified approach to customer acquisition. This unified digital marketing approach requires system integration between various CMS and CRM platforms, and a slew of ecommerce, Marketing Automation, Social Media Listening, email and social marketing, and customer service systems. Our specialized knowledge in Salesforce, open source and .Net based systems enables us to build effective custom integrated solutions for our clients.

Suyati's custom technology solutions have been deployed in companies in the US, Western Europe and Australia, and have helped many enterprises leverage the web/cloud/mobile technologies to acquire customers through integrated digital marketing. Suyati is based in Chicago with product engineering capability out of the US and India.

More at: www.suyati.com Get in touch: services@suyati.com

REFERENCES:

- <http://www.mindtree.com/sites/default/files/mindtree-whitepaper-migrating-an-existing-on-premise-application-to-windows-azure-cloud.pdf>
- <https://azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure/>
- <http://cloudinteropelements.cloudapp.net/Migrate-MySQL-Apps-and-Data-to-Windows-Azure.aspx>
- <https://azure.microsoft.com/en-us/blog/migrate-on-premise-virtualized-workloads-to-azure-using-azure-site-recovery/>
- <https://msdn.microsoft.com/en-in/magazine/jj991979.aspx>
- <http://blogs.technet.com/b/wbaer/archive/2013/04/16/get-started-with-sharepoint-server-2013-fast-on-windows-azure.aspx>
- <https://technet.microsoft.com/en-us/library/mt210445.aspx>
- <http://blogs.technet.com/b/haroldwong/archive/2013/02/18/migration-and-deployment-windows-azure-as-a-paas.aspx>
- <http://blogs.technet.com/b/tothesharepoint/archive/2013/10/26/sharepoint-gaining-momentum-on-windows-azure.aspx>
- <https://msdn.microsoft.com/en-in/magazine/jj991979.aspx>
- <https://azure.microsoft.com/en-in/marketplace/partners/microsoft/cloudservice/>
- <http://www.gartner.com/it-glossary/infrastructure-as-a-service-iaas/>
- <http://www.gartner.com/it-glossary/platform-as-a-service-paas/>
- <http://www.gartner.com/it-glossary/software-as-a-service-saas/>