Microsoft Official Course

AZ-301T02

Designing a Data Platform Solution

AZ-301T02

Designing a Data Platform Solution

Contents

Module 0 Welcome 1
Start Here 1

Module 1 Module Backing Azure Solutions with Azure Storage 3
Pricing 3
Blob Storage 10
Files 14
StorSimple 17
Review Questions 20

Module 2 Module Comparing Database Options in Azure 21
Relational Databases 21
NoSQL Services 24
Azure Cosmos DB 29
Data Storage & Integration 32
Data Analysis 35
Web Apps & SQL Case Study 38
Online Lab - Deploying Database Instances in Azure 48
Review Questions 59

Module 3 Module Monitoring & Automating Azure Solutions 61
Monitoring 61
Backup 67
Automation 71
Design - Business Continuity 75
Online Lab - Deploying Configuration Management solutions to Azure 82
Review Questions 91

Module 0 Welcome

Start Here

Welcome to Designing a Data Platform Solution

Welcome to *Designing a Data Platform Solution*. This course is part of a series of four courses to help students prepare for Microsoft's Azure Solutions Architect technical certification exam AZ-301: Microsoft Azure Architect Design. These courses are designed for IT professionals and developers with experience and knowledge across various aspects of IT operations, including networking, virtualization, identity, security, business continuity, disaster recovery, data management, budgeting, and governance.

This course contains the following three modules:

Module 1 - Backing Azure Solutions with Azure Storage

This module describes how many Azure services use the Azure Storage service as a backing store for other application solution in Azure. The module dives into critical considerations when using Azure Storage as a supplemental service for an all-up Azure solution.

After completing this module, students will be able to:

- Determine the ideal pricing option for Azure Storage based on a solution's requirements.
- Identify performance thresholds for the Azure Storage service.
- Determine the type of Storage blobs to use for specific solution components.
- Use the Azure Files service for SMB operations.
- Identify solutions that could benefit from the use of StorSimple physical or virtual devices.

Module 2 - Comparing Database Options in Azure

This module compares the various relational and non-relational data storage options available in Azure. Options are explored as groups such as relational databases (Azure SQL Database, MySQL, and PostgreSQL on Azure), non-relational (Azure Cosmos DB, Storage Tables), streaming (Stream Analytics) and storage (Data Factory, Data Warehouse, Data Lake).

After completing this module, students will be able to:

- Compare and contrast various database options on Azure.
- Identify data streaming options for large-scale data ingest.
- Identify longer-term data storage options.

Module 3 - Monitoring & Automating Azure Solutions

This module covers the monitoring and automation solutions available after an Azure solution has been architected, designed and possibly deployed. The module reviews services that are used to monitor individual applications, the Azure platform, and networked components. This module also covers automation and backup options to enable business-continuity scenarios for solutions hosted in Azure.

This module contains the hands-on lab entitled Deploying Configuration Management Solutions to Azure.

After completing this module, students will be able to:

- Compare and contrast monitoring services for applications, the Azure platform, and networking.
- Design an alert scheme for a solution hosted in Azure.
- Select the appropriate backup option for infrastructure and data hosted in Azure.
- Automate the deployment of future resources for backup recovery or scaling purposes.
- Determine the ideal pricing option for Azure Storage based on a solution's requirements.
- Identify performance thresholds for the Azure Storage service.
- Determine the type of Storage blobs to use for specific solution components.
- Use the Azure Files service for SMB operations.
- Identify solutions that could benefit from the use of StorSimple physical or virtual devices.
- Compare and contrast monitoring services for applications, the Azure platform, and networking.
- Design an alert scheme for a solution hosted in Azure.
- Select the appropriate backup option for infrastructure and data hosted in Azure.
- Automate the deployment of future resources for backup recovery or scaling purposes.

Module 1 Module Backing Azure Solutions with Azure Storage

Pricing

Azure Storage

This module describes how many Azure services use the Azure Storage service as a backing store for other application solution in Azure. The module dives into critical considerations when using Azure Storage as a supplemental service for an all-up Azure solution.

After completing this module, students will be able to:

- Determine the ideal pricing option for Azure Storage based on a solution's requirements.
- Identify performance thresholds for the Azure Storage service.
- Determine the type of Storage blobs to use for specific solution components.
- Use the Azure Files service for SMB operations.
- Identify solutions that could benefit from the use of StorSimple physical or virtual devices.

Azure storage has many options which affect performance and pricing. To be able to select the best storage for your infrastructure and applications requires a knowledge of what to store, how to store it, access is and pay for it. This lesson covers the essential elements of Azure Storage.

After completing this lesson, you will be able to:

- Describe the three Azure storage accounts and features.
- Understand the redundancy options available in Azure Storage.
- Chose which Azure storage account works best for your requirements.
- Understand how to access, share and delegate storage accounts.

Azure Storage

Azure Storage is massively scalable, so you can store and process hundreds of terabytes of data to support the big data scenarios required by scientific, financial analysis, and media applications. You can also store the small amounts of data required for a small business website as billing is calculated by usage, not capacity. Storage uses an auto-partitioning system that automatically load-balances your data based on traffic. Since storage is elastic and decoupled from your application, you can focus on your workload while relying on Storage's elastic capabilities to scale to meet the demand for your applications.

All Storage services can be accessed using a REST API. Client libraries are also available for popular languages and platforms such as:

- .NET
- Java/Android
- Node.js
- PHP
- Ruby
- Python
- PowerShell

Microsoft Azure provides a storage subsystem that allows users to create a variety of different types of storage. The storage can be provided in differing levels of performance.

Storage Services

Storage accounts are subdivided into available storage services all intended for different functions:

- Blobs: VHDs and large blocks of data (images or documents)
- Tables: structured data (a NoSQL store)
- Queues: for message passing in applications and for backlog processing
- Files: Fully managed SMB 3.0 file shares

To store and manage these storage services Azure uses a container called a Storage Account.

This section deals with the storage account; a subsequent lesson will deal with managed Disks for Virtual Machine disk storage without the need for a storage account.

Azure Storage Accounts

There are three types of Azure Storage accounts.

Type of Account	General Purpose Standard	General Purpose Premium	Blob Storage (hot and cool access tiers)
Services Supported	Blob, File, Queue services	Blob service	Blob service
Types of Blobs supported	Block blobs, Page blobs and Append blobs	Page blobs	Block blobs and Append blobs

There are also three types of entities in Azure Storage accounts:

Blobs: Blobs are available in three forms all intended for different purposes:

- Page blobs lend themselves to storing random access files up to 8TB in size, ideal for VHD storage for Virtual Machines.
- Block blobs are for images or other documents and files up to 4TB in size.

• Append blobs are similar in format to block blobs but allow append operations and are ideal for auditing and logging applications such as log or monitoring data from many sources.

Tables: "*Massive auto-scaling NoSQL store*."

Ideal for: User, device and service metadata, structured data. Features include:

• Schema-less entities with strong consistency.

• No limits on number of table rows or table size.

• Dynamic load balancing of table regions.

• Best for Key/value lookups on partition key and row key.

• Entity group transactions for atomic batching.

Queues: "*Reliable messaging system at scale for cloud services*."

Ideal for: Data sharing, Big Data, Backups. Functions include:

• Decouple components and scale them independently.

• Scheduling of asynchronous tasks.

• Building processes/work flows.

Features include:

• No limits on number of queues or messages.

• Message visibility timeout to protect from component issues.

• **UpdateMessage** to checkpoint progress part way through.

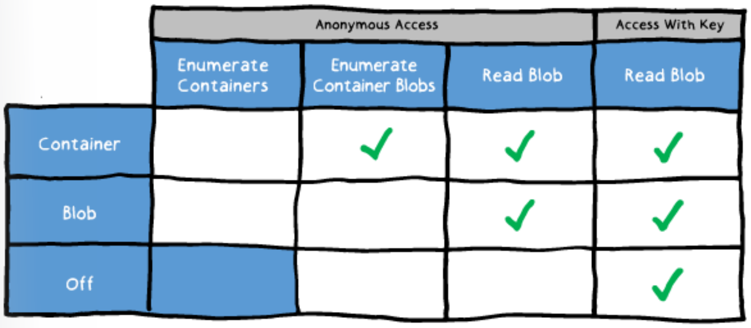
Storage Account Security

Access to storage accounts is controlled by Storage Account Keys (there are primary and secondary to allow for key refreshes whilst maintaining access). If you have the key, you have access to the account and all the data in the account. Container security is also provided for blob storage at the time of creation; the container can be public, private or read-only.

To provide greater security, Azure provides Azure AD RBAC to allow for management functions on the Storage account and the use of RBAC on the storage account keys to prevent incorrectly applied access to the data itself. There are also Shared Access Signatures and Stored Access Policies to provide more granular access at the container, blob, queue or table level. A shared access policy adds the ability to revoke, expire or extend access. The use of all of these security features is a design decision when building your infrastructure or application.

Container Security

Typically, only the owner of a storage account can access resources within that account. If your service or application needs to make these resources available to other clients, you have various options available. First, you can make the public access key generally available. This is not typically recommended as this key gives individuals full access to your entire storage account and its management operations. Another, more common option is to manage access to the entire container.



	Anonymous Access			Access With Key
	Enumerate Containers	Enumerate Container Blobs	Read Blob	Read Blob
Container		✓	✓	✓
Blob			✓	✓
Off				✓

The Public Read Access property controls what data is available anonymously for your container. You can select the following values for the Public Read Access setting:

- **Container:** Blobs in a container can be enumerated. The container metadata is also accessible. Individual blobs within this container and their properties can also be accessed with this setting.
- **Blob:** Only individual blobs and their properties in this container can be accessed. Blobs are not allowed to be enumerated.
- **Off:** With this setting, enumeration of blobs is not allowed. Individual blobs and their properties are also not accessible. You must use your access keys to access any data about this container or its blobs.

Shared Access Signatures

A shared access signature is a URI that grants restricted access rights to containers, blobs, queues, and tables. You can provide a shared access signature to clients who should not be trusted with your storage account key but to whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you can grant them access to a resource for a specified period of time, with a specified set of permissions.

A shared access signature can grant any of the following operations to a client that possesses the signature:

- Reading and writing page or block blob content, block lists, properties, and metadata
- Deleting, leasing, and creating a snapshot of a blob
- Listing the blobs within a container
- Adding, removing, updating, and deleting queue messages
- Getting queue metadata, including the message count
- Querying, adding, updating, deleting, and upserting table entities
- Copying to a blob from another blob within the same account

The shared access signature URI query parameters incorporate all of the information necessary to grant controlled access to a storage resource. The URI query parameters specify the time interval over which the shared access signature is valid, the permissions that it grants, the resource that is to be made available, and the signature that the storage services should use to authenticate the request.

For example, you may wish to have your client application access a resource (container: pictures, blob: profile.jpg) at this URI:

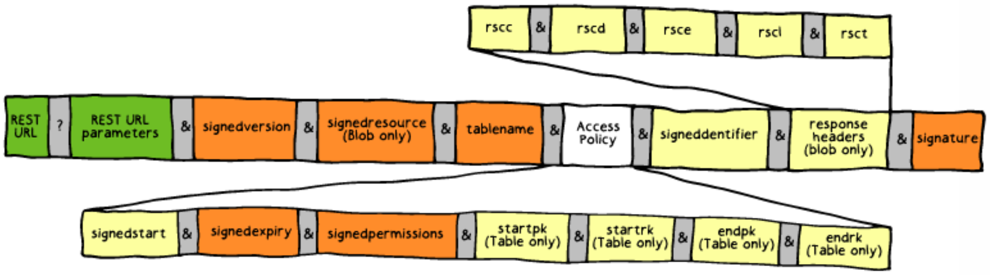
GET https://[account].blob.core.windows.net/pictures/profile.jpg

You could give the client application your access key to manage your entire subscription or make the pictures container anonymously accessible. Neither of these is ideal solutions, so you instead chose to generate a shared access signature. You simply append the generated signature to the end of your URI like this:

GET https://[account].blob.core.windows.net/pictures/profile.jpg?sv=2012-02-12&st=2009-02-09&se=2009-02-10&sr=c&sp=r&si=YWJjZGVmZw%3d%3d&sig=dD80ihBh5jfNpymO5Hg1IdiJIEvHcJpCMiCMnN%2fRnbI%3d

The signature is using version **2012-02-12** of the storage API; it allows read access is beginning from **02/09/09 to 02/10/09** to the container.

The structure of a SAS URI is the composition of multiple pieces.



Now the resource is temporarily accessible by this particular client.

Stored Access Policies

Azure SAS also supports server-stored access policies that can be associated with a specific resource such as a table or blob. This feature provides additional control and flexibility compared to application-generated SAS tokens and should be used whenever possible. Settings defined in a server-stored policy can be changed and are reflected in the token without requiring a new token to be issued, but settings defined in the token itself cannot be changed without issuing a new token. This approach also makes it possible to revoke a valid SAS token before it has expired.

A stored access policy provides an additional level of control over shared access signatures on the server side. Establishing a stored access policy serves to group shared access signatures and to provide additional restrictions for signatures that are bound by the policy. You can use a stored access policy to change the start time, expiry time, or permissions for a signature, or to revoke it after it has been issued.

New shared access signatures are then generated from an existing stored access policy. A maximum of five access policies may be set on a container, table, or queue at any given time. To revoke a stored access policy, you can either delete it or rename it by changing the signed identifier. Changing the signed identifier breaks the associations between any existing signatures and the stored access policy. Deleting or renaming the stored access policy immediately effects all of the shared access signatures associated with it.

Storage Account Replication

The data in your Microsoft Azure storage account is always replicated to ensure durability and high availability. At a minimum, your data is stored in triplicate. You may also choose extended replication options for scenarios where you require your data to be replicated across geography.

Options include:

• **Locally redundant storage (LRS):** Locally redundant storage maintains three copies of your data. LRS is replicated three times within a single facility in a single region. LRS protects your data from normal hardware failures, but not from the failure of a single facility. LRS is the minimum amount of replication.

• **Zone-redundant storage (ZRS):** Zone-redundant storage maintains three copies of your data. ZRS is replicated three times across two to three facilities, either within a single region or across two regions, providing higher durability than LRS. ZRS ensures that your data is durable within a single region.

• **Geo-redundant storage (GRS):** Geo-redundant storage is enabled for your storage account by default when you create it. GRS maintains six copies of your data. With GRS, your data is replicated three times within the primary region, and is also replicated three times in a secondary region hundreds of miles away from the primary region, providing the highest level of durability. In the event of a failure at the primary region, Azure Storage will failover to the secondary region. GRS ensures that your data is durable in two separate regions.

• **Read access geo-redundant storage (RA-GRS):** Read access geo-redundant storage replicates your data to a secondary geographic location, and also provides read access to your data in the secondary location. Read-access geo-redundant storage allows you to access your data from either the primary or the secondary location, in the event that one location becomes unavailable. RA-GRS is also used in scenarios where reporting and other read-only functions can easily be distributed to the replica instead of the primary therefore spreading application load across multiple instances.

Replication	LRS	ZRS	GRS	RA-GRS
Data stored in multiple datacenters	No	Yes	Yes	Yes
Data read from secondary & primary location	No	No	No	Yes
No of copies of data stored in separate nodes	3	3	6	6

Note: Geographically distributed replicas receive any replication asynchronously. This means that your replica is eventually consistent and could possibly have older data if you access the replica before the replication operation from the primary is complete.

Storage Performance and Pricing

Storage Accounts are available in Standard and Premium tiers. Azure Premium Storage delivers low-latency and high-performance disk support for virtual machines (VMs). Premium Storage stores data on solid-state drives (SSDs). Standard VM disks may be migrated to Premium Storage. Using multiple disks gives your applications up to 256 TB of VM storage. Premium Storage provides up to 80,000 I/O operations per second (IOPS) per VM, and a disk throughput of up to 2,000 megabytes per second (MB/s) per VM.

Premium Storage allows the lift-and-shift of demanding applications such as Dynamics AX, Dynamics CRM, SQL Server, Exchange Server and SharePoint farms to Azure. Applications that require consistent high performance and low latency such as SQL Server, Oracle, MongoDB, MySQL, and Redis can run happily in Azure Premium Storage. Premium Storage accounts can only be created with LRS redundancy. Not all Azure VM sizes can take advantage of Premium disk support. Plan carefully your disk and VM sizes to ensure maximum price and efficiency benefits. Currently Premium Storage is available on DS-series, DSv2-series, GS-series, Ls-series, and Fs-series VMs within Azure. Premium disks are always available for use. The storage is allocated on creation and is charged for the whole disk size rather than the data used. This makes premium disks more expensive.

Blob Storage

Blob Storage

The Blob Storage Service provides the infrastructure for Azure to store and manage the Block and Page blobs required to host Azure VM Disks and all other random-access Blobs as well as the block blob documents, images, and files. This lesson provides insight into the managed and unmanaged disk service as well as the blob service within a storage account.

After completing this section, you will be able to:

- Describe managed disks, snapshots, and disk exports.
- Decide whether to deploy VM disks as managed or unmanaged.
- Deploy a managed disk.
- Deploy data disks to a VM for a storage space.

Blob Storage

Azure Blob storage was introduced in the previous lesson as part of the overall Azure Storage platform. In this lesson, we cover the Azure Blob service from an Azure IaaS VM Disk perspective. Looking at VM disks in a managed and unmanaged form and the premium versus standard performance tier of both.

VM Disks

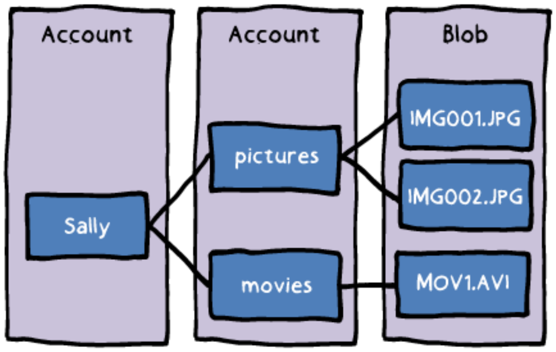
Azure IaaS VMs can attach OS and Data Disks. These disks can be premium storage (SSD based) or Standard storage (HDD). The performance of the disks depends upon the VM in which it is attached and the performance tier in which it is created. The management of VM data disks has been a challenging task in the past due to the performance limitations of a storage account. These were limited to a maximum of 20,000 IOPS per account. The Managed Disk service has provided a new feature in which Azure handles to underlying storage infrastructure and carries out all the redundancy and high availability tasks as well as ensuring that throughput is not throttled by the 20000 IOPS limit in storage accounts. Managed and un-managed disks have different feature sets and performance.

The choice of whether to use Azure Files, Azure Blobs or Azure Disks is one primarily dictated by the data to be stored and the use of that data.

Feature	Description	Suggested Use
Files	SMB 3.0 interface, client libraries and a REST interface access from anywhere to stored files	Application lift and shift using native file system API, Windows File Shares
Blobs	Client libraries, REST interface, for unstructured data stored in Block blobs	Streaming and random access, access application data from anywhere
Disks	Client libraries, REST interface for persistent data accessed from a VHD	Access to data inside a Virtual machine on a VHD, lift and shift file system API apps that write to persistent disks

Unmanaged Disks

Unmanaged disks are available in Standard and Premium tiers. To use an un-managed disk, it is first necessary to create a Storage account and a container within the blob service of that account.



Once created blobs (including VM disks) can then be stored in the container. Each storage account has a limit of 20000 IOPS throughput. To be able to provide maximum performance on a striped storage space with multiple disks inside a VM it would be necessary to spread those data disks across many storage accounts.

Any Azure VM size can have multiple standard disks attached. A storage account can only store one type of disk. The performance of the storage account is chosen at time of creation. Un-managed standard disks can have disk snapshots (using Azure CLI or PowerShell), and VM images can be created with disks included.

Managed Disks

Azure Managed Disks simplifies the creation and management of Azure IaaS VM Disks. When using the Managed Disk service, Azure manages the storage accounts in which the VM disks are stored. With un-managed disks, you need to create a storage account, a container and decide on Standard or Premium storage for the account. Then you need to ensure the VM disks are spread efficiently around several storage accounts. With Managed disks, the requirement is merely to choose the storage type and the disk size. Azure does the rest.

Managed disks provide several other benefits. First amongst these is the ability to upgrade a standard disk to a premium disk and downgrade a premium disk to a standard disk. Because there is no performance level for the underlying storage account, the Azure Managed disk architecture allows quick and straightforward performance tier changes. The only requirement is to detach the disk from a VM before it is upgraded or downloaded.

The architecture of Managed disks also provides:

• **Scalable and straightforward VM deployment**—no need to provision additional storage accounts when scaling the number of disks, the ceiling of 20000 IOPS does not exist. In addition, storing custom images on VHD files can now be accomplished with one storage account per region no need for additional storage accounts.

• **Managed Disks allows up to 10000 VM disks per subscription.** Using managed disks in VM Scale Sets allows up to 1000 VMs per scale set using a Marketplace image.

• **Better reliability for Availability Sets**—Managed disks use a different storage scale unit (or stamp) when using Availability Sets, this further ensures that all the disks in a VM are stored in the same scale unit, and more importantly each VM's disks will be stored in separate scale units ensuring the application relying on those disks remains active.

• **Highly durable and available managed disks are designed for 99.999% availability**. Overall Azure manages a ZERO% annualized failure rate.

• **Granular access control**—Azure AD RBAC allows specific permissions to be assigned to managed disks for one or more users or groups. The granularity allows only read access or to preventing export actions on a disk.

• **Sizing and Pricing** (See: **https://azure.microsoft.com/en-us/pricing/details/managed-disks**[1]**):**

--Premium Managed disks are available from P4 which is 32 GB up to P50 which is 4 TB in size.

--Standard Managed disks are available from S4 which is 32 GB again up to S50 which is 4 TB in size.

--You are billed for the number of transactions performed on a standard managed disk. Premium disks do not attract transaction charges.

• **Managed Disk Snapshots**—A Managed Snapshot is a full copy of a managed disk which is stored as a standard read-only managed disk. Snapshots allow backups at any time. These are stand-alone objects and can be used to create new disks. You are only billed for the used size of the data on the disk. So, if a 4 TB disk that holds 500 GB of data, the snapshot will be billed for 500 GB.

• **Images**—Managed Disks support creating a managed custom image. This captures in a single image all managed disks attached to a VM.

• **Images vs. snapshots**—An image is a copy of the VM and all disks attached. A snapshot is a single disk copy. For this reason, a snapshot is not a suitable choice for the scenario with multiple striped data disks. No snapshot co-ordination is available.

• **Managed Disks and Encryption**—By default Managed disks are encrypted by Azure Storage Service Encryption; this provides encryption at rest for disks, snapshots, and images. Azure Disk Encryption is also available at the VM level in Windows this uses BitLocker Drive Encryption. Azure Key Vault integration is included which allows users to bring their own disk encryption keys.

Deployment Considerations

Deployment of Azure VM disks is easy to achieve using any of the standard tools:

• **Azure Portal**

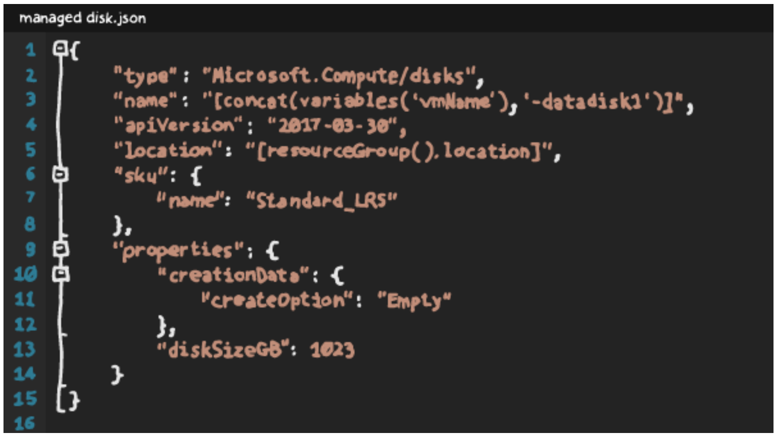
• **PowerShell**

• **Azure CLI**

JSON templates may be used to create and format managed disks as well as un-managed disks.

The data disks can be added within the VM creation template or defined stand alone as a top-level disk resource; this would then be attached when the VM is created.

[1] https://azure.microsoft.com/en-us/pricing/details/managed-disks/



managed disk.json

```
{
    "type": "Microsoft.Compute/disks",
    "name": "[concat(variables('vmName'),'-datadisk1')]",
    "apiVersion": "2017-03-30",
    "location": "[resourceGroup().location]",
    "sku": {
        "name": "Standard_LRS"
    },
    "properties": {
        "creationData": {
            "createOption": "Empty"
        },
        "diskSizeGB": 1023
    }
}
```

Files

Azure Files

The ability to share files without the need to deploy the underlying server infrastructure provides several benefits when building an Azure based application. This lesson will describe the Azure File Service and the Azure File Sync service to allow the practitioner to choose which method of sharing files across an application or server infrastructure whether hybrid, on-premises or cloud-based.

After completing this lesson, you will be able to:

- Describe Azure Files service features.

- Decide which file sharing option suits and application best.

- Understand the benefits of Azure File Sync.

- Create an Azure File Sync service.

Azure Files

File storage offers shared storage for applications using the standard SMB 3.0 protocol. Microsoft Azure virtual machines and cloud services can share file data across application components via mounted shares, and on-premises applications can access file data in a share via the File storage API.

Applications running on Azure virtual machines or cloud services can mount a File storage share to access file data, just as a desktop application would mount a typical SMB share. Any number of Azure virtual machines or roles can mount and access the File storage share simultaneously.

Since a File storage share is a standard SMB 3.0 file share, applications running in Azure can access data in the share via file I/O APIs. Developers can, therefore, leverage their existing code and skills to migrate existing applications. IT Pros can use PowerShell cmdlets to create, mount, and manage File storage shares as part of the administration of Azure applications. This guide will show examples of both.

Typical uses of File storage include:

- Migrating on-premises applications that rely on file shares to run on Azure virtual machines or cloud services, without expensive rewrites.

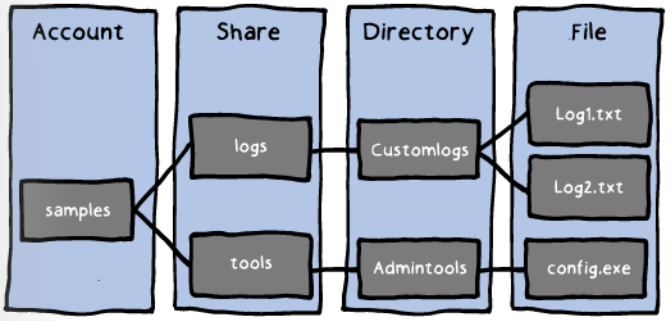
- Storing shared application settings, for example in configuration files.

- Storing diagnostic data such as logs, metrics, and crash dumps in a shared location.

- Storing tools and utilities needed for developing or administering Azure virtual machines or cloud services.

File Share Service Components

File storage contains the following components:



Components of File Share Service

- **Storage Account**: All access to Azure Storage is done through a storage account.

- **Share**: A File storage share is an SMB 3.0 file share in Azure. All directories and files must be created in a parent share. An account can contain an unlimited number of shares, and a share can store an unlimited number of files, up to the capacity limits of the storage account.

- **Directory**: An optional hierarchy of directories.

- **File**: A file in the share. A file may be up to 1 TB in size.

- **URL format:** Files are addressable using the following URL format:

https://[account].file.core.windows.net/<share>/<directory/directories>/<file>

The following example URL could be used to address one of the files in the diagram above:

HTTP://[account].file.core.windows.net/logs/CustomLogs/Log1.txt

Azure File Sync

The Azure File Sync Service is a new service that will allow you to centralize your file shares in Azure Files, whilst maintaining the compatibility of an on-premises file server with all the flexibility and performance benefits that provide. The Azure File Sync service turns your file server into a cache of the Azure-based file share. Any protocol installed on the Windows Server can access the file share, including SMB, NFS, and FTPS. The ability to have as many servers as you need spread across the globe turns the service into a useful way of providing all the benefits of a distributed file system without the infrastructure and maintenance requirements.

Azure File Sync Terminology

When planning an Azure File Sync deployment, there are several components required to deploy the service efficiently.

- **Storage Sync Service:** The Storage Sync Service is the Azure resource created to host the Azure File Sync service. The Storage Sync Service resource is deployed to a resource group and can be created and amended using JSON templates if required. This service is required since the service can sync between multiple storage accounts, hence an additional resource is required to manage this. A subscription can contain multiple storage sync services.

- **Sync Group:** A Sync Group defines and controls the hierarchy and topology of the files to be synced. The sync group will contain Cloud and Server endpoints. Async service can contain multiple sync groups.

- **Registered Server:** Before adding a server endpoint to a sync group, the server must be registered with a storage sync service. A server can only be registered to a single sync service. Async service can host as many registered servers as you need.

- **Azure File Sync Agent:** To register a server, you need to install the Azure File Sync Agent. This is a small downloadable MSI package comprising three components:

-- FileSyncSvc.exe: Monitors changes on Server Endpoints, and for initiating sync sessions to Azure.

-- StorageSync.sys: A file system filter, which handles tiering files to Azure Files.

-- PowerShell Management cmdLets: PowerShell cmdlets for the Microsoft.StorageSync Azure resource provider.

- **Server Endpoint:** Once registered, you can add a server to a Sync group, this then becomes a server endpoint. A server endpoint is synonymous with a folder or a volume on the server that will cache the contents of the Azure File Share. Cloud tiering is configured individually by server endpoint.
- **Cloud Endpoint:** When added to a sync group, an Azure File Share is a cloud endpoint. One Azure File Share can only be a member of one Cloud Endpoint and thereby can only be a member of one Sync Group. Any files that exist in a cloud endpoint or a server endpoint before they are added to the sync group, automatically become merged with all other files in the sync group.

Azure Files Features

Below is a list of features available for Azure Files:

- **Adding files to the File Share:** Azure File Sync supports adding and removing files directly within the Azure file share. These files will only sync once every 24 hours down to the server endpoints. This is due to the change detection job only being scheduled once every 24 hours.
- **Cloud tiering:** Cloud tiering is a feature of Azure File Sync which can save considerable space on a server endpoint. When a file is tiered, the sync system filter replaces the local file with a pointer to the location if the Azure file share. This is marked as offline in NTFS. When accessed locally the file is downloaded and opened for use. This is Hierarchical Storage Management (HSM).
- **Supported versions of Windows Server:** Currently, Azure File Sync is supported by all GUI editions of Windows Server 2012 R2 and Windows Server 2016 versions of Windows.
- **Access control lists (ACL):** Supported and enforced on files held on Server endpoints. Azure Files do not currently support ACLs.
- **NTFS compression:** Fully supported, and Sparse files are fully supported but are stored in the cloud as full files, and any cloud changes are synced as full files on server endpoints.
- **Failover Clustering:** Supported for File Server for General Use but not for Scale-out file server for application data. Cluster Shared Volumes are not supported. To function correctly, the sync agent must be installed on every node of a cluster.
- **Data Deduplication:** Fully supported for volumes that do not have cloud tiering enabled.
- **Encryption solutions:** Azure File Sync, is known to work with BitLocker Drive Encryption and Azure Rights Management Services. NTFS Encrypted File System does not work with Azure File Sync.

StorSimple

StorSimple

The use of Active Directory is widespread throughout the on-premises and cloud-based Windows infrastructure world. The advent of Azure AD brings many options for the Azure Architect to choose between. This lesson will examine the benefits of and differences between cloud only and hybrid solutions comprised of on-premises Active Directory Domain Services, Azure AD, and Azure AD Domain Services.

After completing this section you will be able to:

• Describe the StorSimple service and devices.

• Decide when to use StorSimple in your hybrid File storage solution.

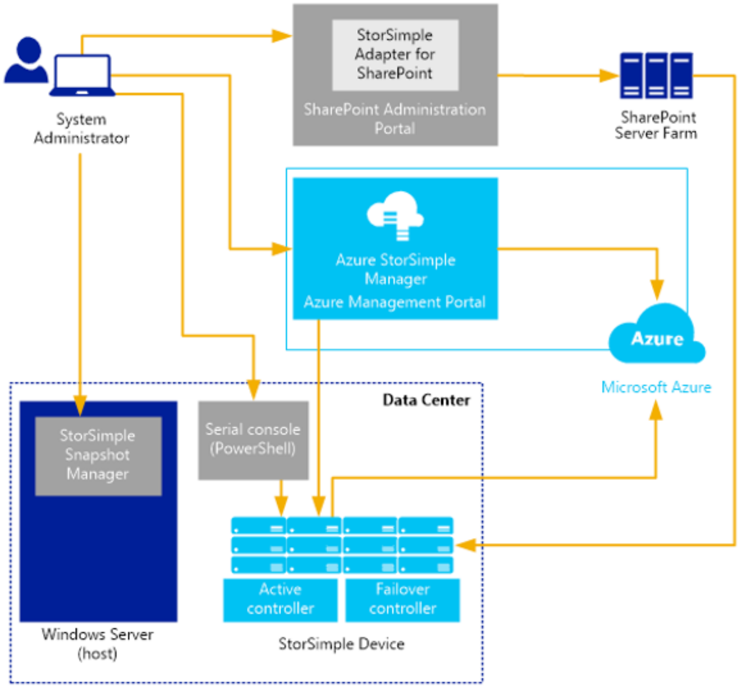
• Understand the StorSimple data tiering process.

StorSimple

StorSimple is the combination of a service, device and management tools that can create workflows for migrating data to a cloud storage center or back on premise.

The StorSimple device is an on-premises hybrid storage array that provides primary storage and iSCSI access to data stored on it. It manages communication with cloud storage and helps to ensure the security and confidentiality of all data that is stored on the StorSimple solution. The StorSimple device includes solid state drives (SSDs) and hard disk drives (HDDs), as well as support for clustering and automatic failover. It contains a shared processor, shared storage, and two mirrored controllers.

You can alternatively use StorSimple to create a virtual device that replicates the architecture and capabilities of the actual hybrid storage device. The StorSimple virtual device (also known as the StorSimple Virtual Appliance) runs on a single node in an Azure virtual machine.



StorSimple provides a web-based user interface (the StorSimple Manager service) that enables you to manage data center and cloud storage centrally. You can also use a Windows PowerShell-based, a command-line interface that includes dedicated cmdlets for managing your StorSimple device. Finally, you can interact with StorSimple using a Microsoft Management Console (MMC) snap-in that's used to configure and create consistent, point-in-time backup copies of local and cloud data.

Features

- **Transparent integration:** Microsoft Azure StorSimple uses the Internet Small Computer System Interface (iSCSI) protocol to invisibly link data storage facilities. This ensures that data stored in the cloud, in the data center, or on remote servers appears to be stored at a single location.

- **Reduced storage costs:** Microsoft Azure StorSimple allocates sufficient local or cloud storage to meet current demands and extends cloud storage only when necessary. It further reduces storage requirements and expense by eliminating redundant versions of the same data (deduplication) and by using compression.

- **Simplified storage management:** Microsoft Azure StorSimple provides system administration tools that you can use to configure and manage data stored on-premises, on a remote server, and in the cloud. Additionally, you can manage backup and restore functions from a Microsoft Management Console (MMC) snap-in. StorSimple provides a separate, optional interface that you can use to extend StorSimple management and data protection services to content stored on SharePoint servers.

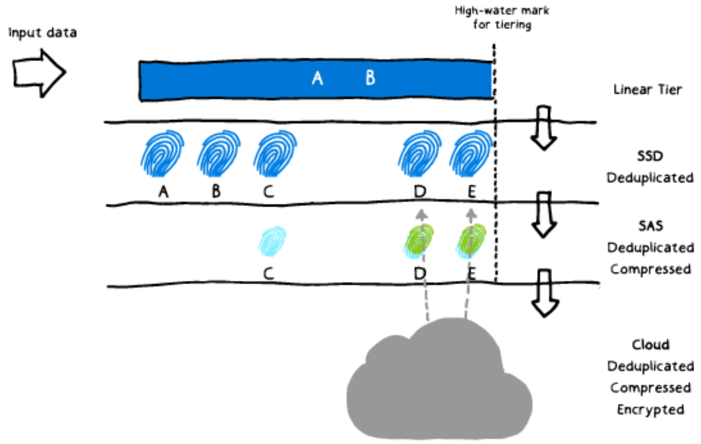
- **Improved disaster recovery and compliance:** Microsoft Azure StorSimple does not require extended recovery time. Instead, it restores data as it is needed. This means regular operations can continue with minimal disruption. Additionally, you can configure policies to specify backup schedules and data retention.

- **Data mobility:** Data uploaded to Microsoft Azure cloud services can be accessed from other sites for recovery and migration purposes. Additionally, you can use StorSimple to configure StorSimple virtual devices on virtual machines (VMs) running in Microsoft Azure. The VMs can then use virtual devices to access stored data for test or recovery purposes.

Data Tiering

StorSimple automatically tiers and classifies your data based on how often you access it. Data is always being shuffled between tiers as the mechanism learns about your usage patterns.

Data that is most active is stored locally, while less active and inactive data is automatically migrated to the cloud. To enable quick access, StorSimple stores very active data (hot data) on SSDs in the StorSimple device. It stores data that is occasionally used (warm data) on HDDs in the device or on servers at the datacenter. It moves inactive data, backup data, and data retained for archival or compliance purposes to the cloud. StorSimple adjusts and rearranges data and storage assignments as usage patterns change. For example, some information might become less active over time. As it becomes progressively less active, it is migrated from SSD to HDD and then to the cloud. If that same data becomes active again, it is migrated back to the storage device.


High-water mark for tiering
Input data
A B
Linear Tier
A B C D E
SSD Deduplicated
C D E
SAS Deduplicated Compressed
Cloud Deduplicated Compressed Encrypted

Review Questions

Module 1 Review Questions

Storage services

You are designing a solution for a company. The solution will use Azure Storage.

What storage solutions does Azure offer? Explain how these different storage solutions are managed.

Suggested Answer ↓

Azure offers four storage solutions designed for different purposes. These include Blobs, Tables, Queues, and Files. To store and manage services, Azure uses a container called a storage account.

Azure storage

You are designing a solution that uses Azure Storage.

What types of Blob disk storage are available? Which type allows quick and straightforward performance tier changes?

Suggested Answer ↓

Blob storage offers managed and unmanaged disks type. There are many benefits to using managed including the ability the quickly change performance tiers.

Azure storage

You are designing a solution for an IT security company. The company archives large sets of log files.

You need to design an Azure solution to store the log files.

Which Azure service should you use? What can you do to ensure your files are centralized?

Suggested Answer ↓

Azure Files provides the ability to store and share files without the need to deploy any underlying server infrastructure. The Azure File Sync Service is used to centralize file shares in Azure Files.

Module 2 Module Comparing Database Options in Azure

Relational Databases

Azure SQL Database

This module discusses the three managed database services in Azure; Azure SQL Database, Azure Database for MySQL, and Azure Database for PostgreSQL.

After completing this section you will be able to:

- Describe the difference between the three services for SQL, MySQL and PostgreSQL.

- Determine when to use advanced features of Azure SQL Database such as Elastic Database and Stretch Database.

Azure SQL Database

SQL Database is a relational database as-a-service offering that provides predictable performance and a high degree of compatibility with existing management tools.

- **Predictable Performance**: By using a consistent unit of measurement, such as Database Throughput Units, you can compare the expected service level for each performance tier that is offered in the SQL Database service. Consistent and predictable performance allows you to select a tier that very closely matches your application's real-world utilization.

- **High Compatibility:** A Tabular Data Stream (TDS) endpoint is provided for each logical server that is created in the SQL Database service. You can use existing SQL client applications and tools with SQL Database by using the TDS protocol.

- **Simple Management:** Additional tools are available in Azure to manage databases that are created by SQL Database. A portal for managing database objects is available in the Azure Management Portal, which you can access by clicking the Manage button. You also can manage SQL Database instances by using the portals, REST API, Windows PowerShell, or the cross-platform command-line interface (Xplat CLI).

Database Throughput Unit (DTU)

DTUs are used to describe the capacity for a specific tier and performance level. DTUs are designed to be relative so that you can directly compare the tiers and performance levels. For example, the Basic tier has a single performance level (B) that is rated at 5 DTU. The S2 performance level in the Standard tier is rated at 50 DTU. This means that you can expect ten times the power for a database at the S2 performance level than a database at the B performance level in the Basic tier.

Azure SQL Database Tiers

Azure SQL Database Tiers

The SQL Database service is offered in several tiers. You can select a tier that closely matches your application's intended or actual resource needs. The following is a list of SQL Database service tiers with the associated performance characteristics:

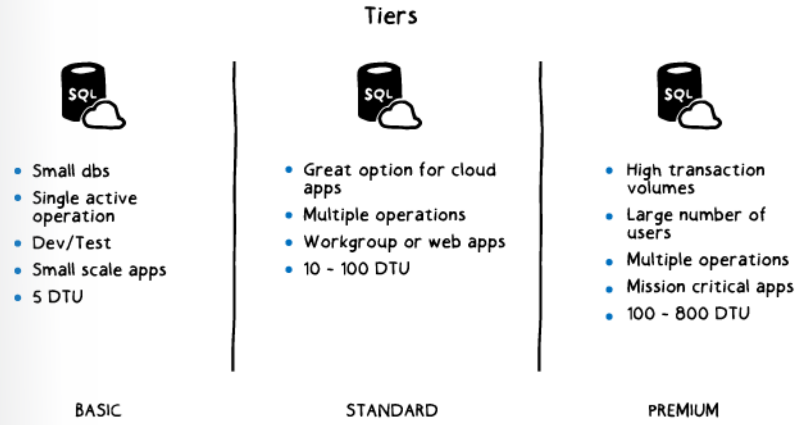
• **Basic**: Ideal for simple databases that requires only a single connection performing a single operation at a time.

• **Standard**: The most common option and is used for databases that require multiple concurrent connections and operations.

• **Premium:** Designed for applications that require large quantities of transactions at volume. These databases support a large quantity of concurrent connections and parallel operations.

These tiers are further separated into performance levels. Performance levels are very specific categories within a service tier that provides a specific level of service. For example, the P1 performance level in the Premium tier offers a maximum database size of 500 gigabyte (GB) and a benchmarked transaction rate of 105 transactions per second.

Every tier has one or more performance levels. In general, the performance levels in the Premium tier are rated higher than the performance levels in the Standard tier, which are again rated higher than the Basic tier. The following chart illustrates this distinction.



Azure SQL Database Elastic Scale

The new Elastic Scale capabilities simplify the process of scaling out (and in) a cloud application's data tier by streamlining development and management. Elastic Scale is composed of two main parts:

• An Elastic Scale library for client applications to configure shards and access shards.

• The Elastic Scale features in Azure SQL Database that implements any changes requested by your application.

Elastic Scale implements the database scaling strategy known as sharding. As a developer, you can establish a "contract" that defines a shard key and how shards should be partitioned across a collection of databases. The application, using the SDK, can then automatically direct transactions to the appropriate database (shard), perform queries across multiple shards or modify the service tier for existing shards. Elastic Scale also enables coordinated data movement between shards to split or merge ranges of data among different databases and satisfy common scenarios such as pulling a busy tenant into its own shard. The Split-Merge service is provided through a downloadable package that customers can deploy as an Azure cloud service into their own subscription.

Third-Party Databases in Azure

Azure provides two additional managed database options for applications running on Azure.

• **Azure Database for MySQL:** This is a managed MySQL instance running the MYSQL community version. The instance comes with tools preinstalled like mysql.exe and phpMyAdmin. You can run one or more databases with this instance.

• **Azure Database for PostgreSQL:** This is a managed PostgreSQL instance that can run one or more databases.

Both of the managed database services in Azure share a common set of features:

• Built-in high availability with no additional cost.

• Predictable performance, using inclusive pay-as-you-go pricing.

• Scale on the fly within seconds.

• Secured to protect sensitive data at-rest and in-motion.

• Automatic backups and point-in-time-restore for up to 35 days.

• Enterprise-grade security and compliance.

Other Options

Using Windows or Linux virtual machines, you can always install and run MySQL in the Azure environment. ClearDB also provides a managed MySQL instance that you can create from the Azure Marketplace.

NoSQL Services

Azure Storage

This Module talks about some of the NoSQL services available in Azure to store and perform analysis on massive data sets.

Lesson Objectives

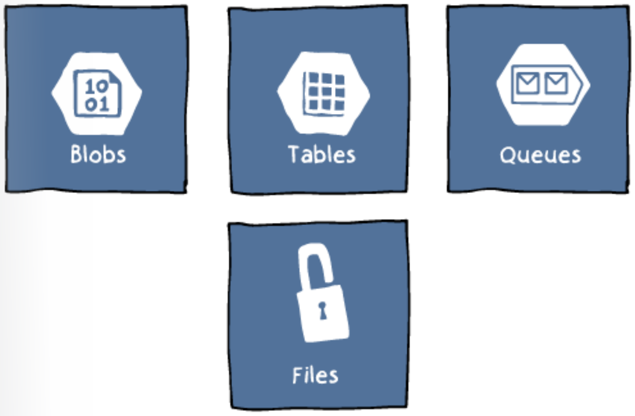
After completing this section you will be able to:

- Identify when to use Azure Storage Tables in a solution.

- Integrate Azure Search with existing data solutions.

Azure Storage

Azure Storage is massively scalable, so you can store and process hundreds of terabytes of data to support the big data scenarios required by scientific, financial analysis, and media applications. You can also store the small amounts of data required for a small business website as billing is calculated by usage, not capacity. Storage uses an auto-partitioning system that automatically load-balances your data based on traffic. Since storage is elastic and decoupled from your application, you can focus on your workload while relying on Storage's elastic capabilities to scale to meet demand for your applications.



All Storage services can be accessed using a REST API. Client libraries are also available for popular languages and platforms such as:

- .NET

- Java/Android

- Node.js

- PHP

- Ruby

- Python

- PowerShell

Finally, all resources in Storage can be protected from anonymous access and can be used in the Valet-Key pattern configuration discussed in previous modules.

Replication

The data in your Microsoft Azure storage account is always replicated to ensure durability and high availability. At a minimum, your data is stored in triplicate. You may also choose extended replication options for scenarios where you require your data to be replicated across geography.

- **Locally redundant storage (LRS):** Locally redundant storage maintains three copies of your data. LRS is replicated three times within a single facility in a single region. LRS protects your data from normal hardware failures, but not from the failure of a single facility. LRS is the minimum amount of replication.

- **Zone-redundant storage (ZRS):** Zone-redundant storage maintains three copies of your data. ZRS is replicated three times across two to three facilities, either within a single region or across two regions, providing higher durability than LRS. ZRS ensures that your data is durable within a single region.

- **Geo-redundant storage (GRS):** Geo-redundant storage is enabled for your storage account by default when you create it. GRS maintains six copies of your data. With GRS, your data is replicated three times within the primary region, and is also replicated three times in a secondary region hundreds of miles away from the primary region, providing the highest level of durability. In the event of a failure at the primary region, Azure Storage will failover to the secondary region. GRS ensures that your data is durable in two separate regions.

- **Read access geo-redundant storage (RA-GRS):** Read access geo-redundant storage replicates your data to a secondary geographic location, and also provides read access to your data in the secondary location. Read-access geo-redundant storage allows you to access your data from either the primary or the secondary location, in the event that one location becomes unavailable. RA-GRS is also used in scenarios where reporting and other read-only functions can easily be distributed to the replica instead of the primary therefore spreading application load across multiple instances.

Note: Geographically distributed replicas receive any replication asynchronously. This means that your replica is eventually consistent and could possibly have older data if you access the replica before the replication operation from the primary is complete.

Architecture

The Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency whitepaper that was released at the 2011 Association for Computing Machinery (ACM) Symposium on Operating Systems Principles.

Azure Storage Tables

The Azure Table storage service stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of the Table service include:

- Storing TBs of structured data capable of serving web scale applications.

- Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be de-normalized for fast access.

- Quickly querying data using a clustered index.

- Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries.

You can use the Table service to store and query huge sets of structured, non-relational data, and your tables will scale as demand increases.

The Table service contains the following components:

- **URL format:** Code addresses tables in an account using this address format:

http://<storage account>.table.core.windows.net/<table>

You can address Azure tables directly using this address with the OData protocol.

- **Storage Account:** All access to Azure Storage is done through a storage account.
- **Table:** A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties. The number of tables that a storage account can contain is limited only by the storage account capacity limit.
- **Entity:** An entity is a set of properties, similar to a database row. An entity can be up to 1 MB in size.
- **Properties:** A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has 3 system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.

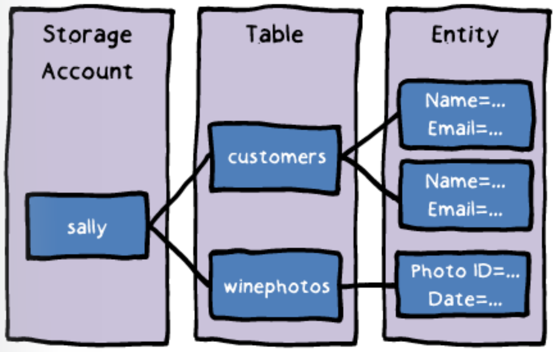
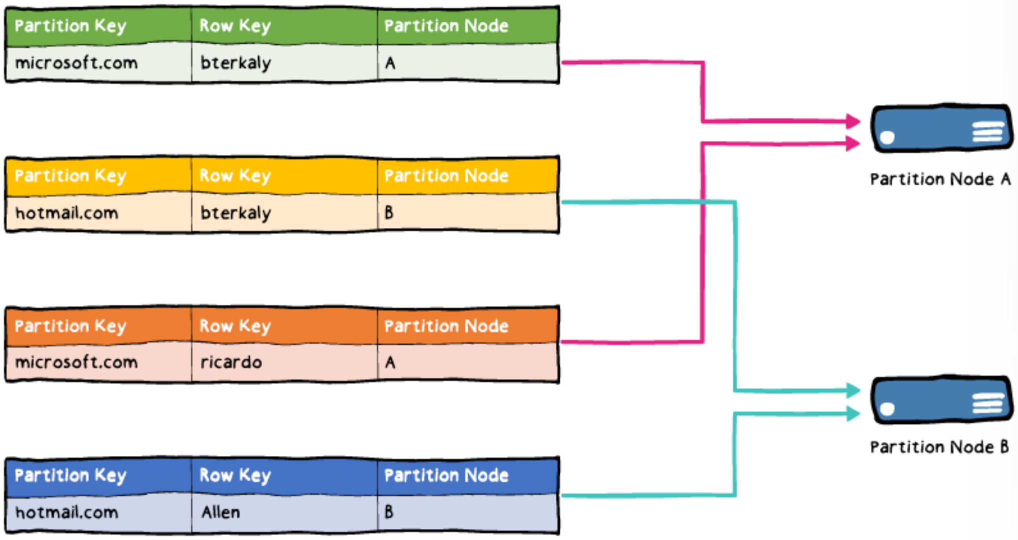


Table Partitioning

Partitions represent a collection of entities with the same PartitionKey values. Partitions are always served from one partition server and each partition server can serve one or more partitions. A partition server has a rate limit of the number of entities it can serve from one partition over time. Specifically, a partition has a scalability target of 500 entities per second. This throughput may be higher during minimal load on the storage node, but it will be throttled down when the node becomes hot or very active.

To better illustrate the concept of partitioning, the following figure illustrates a table that contains a small subset of data for users. It presents a conceptual view of partitioning where the PartitionKey contains different values:



Partition Key	Row Key	Partition Node
microsoft.com	bterkaly	A

Partition Key	Row Key	Partition Node
hotmail.com	bterkaly	B

Partition Key	Row Key	Partition Node
microsoft.com	ricardo	A

Partition Key	Row Key	Partition Node
hotmail.com	Allen	B

Storage Table Partioning

The primary key for an Azure entity consists of the combined PartitionKey and RowKey properties, forming a single clustered index within the table. The PartitionKey and RowKey properties can store up to 1 KB of string values. Empty strings are also permitted; however, null values are not. The clustered index sorts by the PartitionKey in ascending order and then by RowKey in ascending order. The sort order is observed in all query responses.

Because a partition is always served from a single partition server and each partition server can serve one or more partitions, the efficiency of serving entities is correlated with the health of the server. Servers that encounter high traffic for their partitions may not be able to sustain a high throughput. For example, if there are many requests for Partition B, server B may become too hot. To increase the throughput of the server, the storage system load-balances the partitions to other servers. The result is that the traffic is distributed across many other servers. For optimal load balancing of traffic, you should use more partitions, so that the Azure Table service can distribute the partitions to more partition servers.

The PartitionKey values you choose will dictate how a table will be partitioned and the type of queries that can be used. Storage operations, in particular inserts, can also affect your choice of PartitionKey values. The PartitionKey values can range from single values to unique values and also can be composed from multiple values. Entity properties can be composed to form the PartitionKey value. Additionally, the application can compute the value.

Azure Search

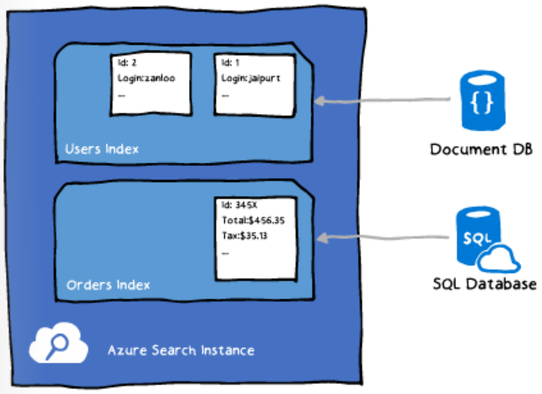
Azure Search is a fully managed cloud service that allows developers to build rich search applications using a .NET SDK or REST APIs. It includes full-text search scoped over your content, plus advanced search behaviors similar to those found in commercial web search engines, such as type-ahead query suggestions based on a partial term input, hit-highlighting, and faceted navigation. Natural language support is built-in, using the linguistic rules that are appropriate to the specified language.

You can scale your service based on the need for increased search or storage capacity. For example, retailers can increase capacity to meet the extra volumes associated with holiday shopping or promotional events. Azure Search is also an API-based service for developers and system integrators who know how to work with web services and HTTP. You can use existing platforms and frameworks since search only requires HTTP requests.

Azure Search is a PaaS service that delegates server and infrastructure management to Microsoft, leaving you with a ready-to-use service that you populate with search data, and then access from your application. Depending on how you configure the service, you'll use either the free service that is shared with other Azure Search subscribers, or the Standard pricing tier that offers dedicated resources used only by your service. Standard search is scalable, with options to meet increased demands for storage or query loads. Azure Search stores your data in an index that can be searched through full text queries. The schema of these indexes can either be created in the Azure Portal, or programmatically using the client library or REST APIs. The schema can also be auto-generated from an existing data source such as SQL Database or Document DB. Once the schema is defined, you can then upload your data to the Azure Search service where it is subsequently indexed.

NOTE: Search can index existing data stores including:

- CosmosDB DocumentDB API
- Azure SQL Database



Azure Cosmos DB

Azure Cosmos DB APIs

This section briefly introduces the Azure Cosmos DB NoSQL database service.

After completing this section you will be able to:

- Describe the general features available in the Azure Cosmos DB service.

- List the specific APIs and models available for Azure Cosmos DB client applications.

Cosmos DB

Azure Cosmos DB is Microsoft's new globally distributed, multi-model database service. Azure Cosmos DB offers a turn-key database service that allows you to create a database and distribute the data globally so that the data users access or in datacenters closer to them.
Azure Cosmos DB APIs.

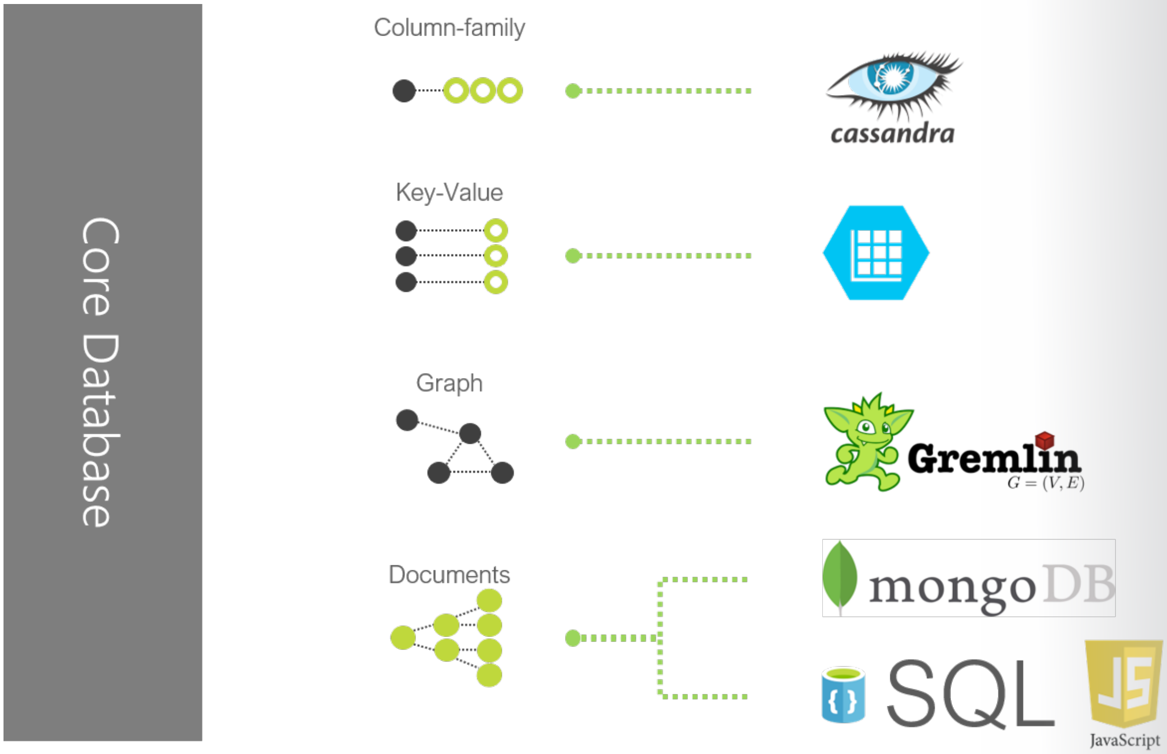
Today, Azure Cosmos DB can be accessed using four different APIs:

- **DocumentDB (SQL) API**

- **MongoDB API**

- **Graph (Gremlin) API**

- **Tables (Key/Value) API**



Over time, Azure Cosmos DB will be expanded to offer new APIs and data models that are relevant to the latest distributed applications.

Consistency Levels

When you create a set of data in Azure Cosmos DB, your data is transparently replicated to ensure high availability. To accomplish this, the Azure Cosmos DB service automatically creates partitions, behind the scenes, and distribute your data across these partitions.

In Azure Cosmos DB, a partition is a fixed amount of high-performance storage that contains your data. When your data grows beyond the capacity of a partition, the Azure Cosmos DB service automatically determines the quantity of partitions needed and how to distribute the
data across those partitions.

Additionally, you can specify a partition key to influence how your data is distributed. A partition key is a JSON path or property that is used by DocumentDB to ensure that related documents are stored in the same parition. This means that documents with the same partition key would be stored within the same partition. This also means that queries within a single partition perform better than queries that cross multiple partitions.

Consistency Levels

When you create a set of data in Azure Cosmos DB, your data is transparently replicated to ensure high availability. To accomplish this, the Azure Cosmos DB service automatically creates partitions, behind the scenes, and distribute your data across these partitions.

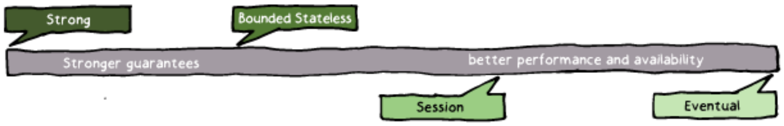
In Azure Cosmos DB, a partition is a fixed amount of high-performance storage that contains your data. When your data grows beyond the capacity of a partition, the Azure Cosmos DB service automatically determines the quantity of partitions needed and how to distribute the
data across those partitions.

Additionally, you can specify a partition key to influence how your data is distributed. A partition key is a JSON path or property that is used by DocumentDB to ensure that related documents are stored in the same parition. This means that documents with the same partition key would be stored within the same partition. This also means that queries within a single partition perform better than queries that cross multiple partitions.

DocumentDB allows you to specify one of four potential consistency levels per account. A consistency level specified at the database-level is applied automatically to all databases and collections within your account.
The consistency levels range from very strong consistency where reads are guaranteed to be visible across replicas before a write is fully committed across all replicas to an eventual consistency where writes are readable immediately and replicas are eventually consistent with the primary.

Consistency Level	Description
Strong	When a write operation is performed on your primary database, the write operation is replicated to the replica instances. The write operation is only committed (and visible) on the primary after it has been committed and confirmed by ALL replicas.
Bounded Stateless	This level is similar to the Strong level with the major difference is that you can configure how stale documents can be within replicas. Staleness refers to the quantity of time (or version count) a replica document can be behind the primary document.

Consistency Level	Description
Session	This level guarantees that all read and write operations are consistent within a user session. Within the user session, all reads and writes are monotonic and guaranteed to be consistent across primary and replica instances.
Eventual	This level is the loosest consistency and essentially commits any write operation against the primary immediately. Replica transactions are asynchronously handle and will eventually (over time) be consistent with the primary. This tier is the most performant as the primary database does not need to wait for replicas to commit to finalize its transactions.



Choosing a Consistency Strategy

There are two main things to consider when thinking about your consistency level. First a consistency level on the strong side of the list will ensure that your versions of documents in your replica do not lag behind the primary. If your application requires all replica documents to exactly match the primary at any point in time, this strategy makes a lot of sense. The downside is that the primary write operation will be a lot slower than usual because that operation must wait for every replica to confirm that the operation has been committed.

A consistency level on the eventual (loose) side will ensure that your database operates at peak efficiency. This occurs because operations against the primary database commit immediately and do not wait for the replicas to confirm that they are committed. This is useful for scenarios where you need the highest tier of performance. The downside here is that there is a potential for any read operations against a replica to be a couple of versions behind the primary and return inconsistent data.

Data Storage & Integration

Data Storage & Integration Options

This section is a brief overview of various Data Storage & Data Integration options available on Azure.

After completing this section you will be able to:

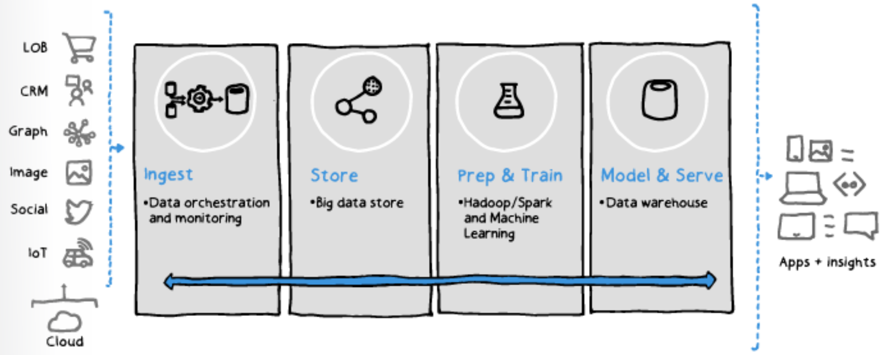
- Describe the SQL Data Warehouse and Data Lake services.
- Describe the Data Factory service.
- Decide when it is appropriate to use SQL Data Warehouse, Data Lake or Data Factory.
- Incorporate Data Factory into a design that uses either SQL Data Warehouse or Data Factory.

Data Storage & Integration Options

SQL Data Warehouse is a cloud-based Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data. Use SQL Data Warehouse as a key component of a big data solution. Import big data into SQL Data Warehouse with simple PolyBase T-SQL queries, and then use the power of MPP to run high-performance analytics. As you integrate and analyze, the data warehouse will become the single version of truth your business can count on for insights.

In a cloud data solution, data is ingested into big data stores from a variety of sources. Once in a big data store, Hadoop, Spark, and machine learning algorithms prepare and train the data. When the data is ready for complex analysis, SQL Data Warehouse uses PolyBase to query the big data stores. PolyBase uses standard T-SQL queries to bring the data into SQL Data Warehouse.

SQL Data Warehouse stores data into relational tables with columnar storage. This format significantly reduces the data storage costs, and improves query performance. Once data is stored in SQL Data Warehouse, you can run analytics at massive scale. Compared to traditional database systems, analysis queries finish in seconds instead of minutes, or hours instead of days.



The analysis results can go to worldwide reporting databases or applications. Business analysts can then gain insights to make well-informed business decisions.

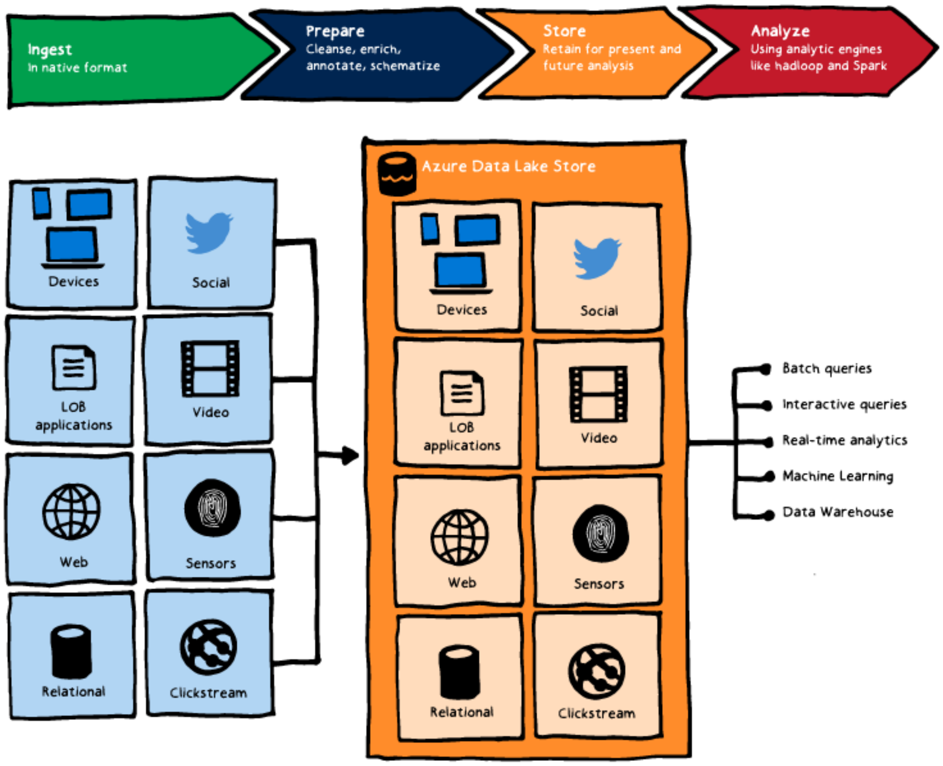
Azure Data Lake

Azure Data Lake Store is an enterprise-wide hyper-scale repository for big data analytic workloads. Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics.

Azure Data Lake Store can be accessed from Hadoop (available with HDInsight cluster) using the WebHD-

FS-compatible REST APIs. It is specifically designed to enable analytics on the stored data and is tuned for performance for data analytics scenarios. Out of the box, it includes all the enterprise-grade capabilities—security, manageability, scalability, reliability, and availability—essential for real-world enterprise use cases.

Some of the key capabilities of the Azure Data Lake include the following:



Azure Data Lake Analytics is an on-demand analytics job service to simplify big data analytics. You can focus on writing, running, and managing jobs rather than on operating distributed infrastructure. Instead of deploying, configuring, and tuning hardware, you write queries to transform your data and extract valuable insights. The analytics service can handle jobs of any scale instantly by setting the dial for how much power you need. You only pay for your job when it is running, making it cost-effective. The analytics service supports Azure Active Directory letting you manage access and roles, integrated with your on-premises identity system. It also includes U-SQL, a language that unifies the benefits of SQL with the expressive power of user code. U-SQL's scalable distributed runtime enables you to efficiently analyze data in the store and across SQL Servers in Azure, Azure SQL Database, and Azure SQL Data Warehouse.

Data Integration

Azure Data Factory

Azure Data Factory is a platform created to help refine big data, enormous stores of raw data, into actionable business insights. It is a managed cloud service that's built to handle extract-transfer-load, data integration, and hybrid extract-transfer-load projects. You can use it to schedule and create data-driven workflows, other words known as pipelines, that can ingest data from data stores. Azure Data Factory operates in conjuncture with services such as Azure Data Lake Analysts, Azure Machine learning, and Spark to process and transform data.

Azure Data Factory is the service that can help with scenarios of this kind. Azure Data Factory is a cloud-based data integration service. It allows users to create data-driven workflows in the cloud, as well as methods to automate data movement and transformation. You can also issue output data to data stores to consume. By using Azure Data Factory, raw data can be organized into meaningful data lakes to assist in business decisions.

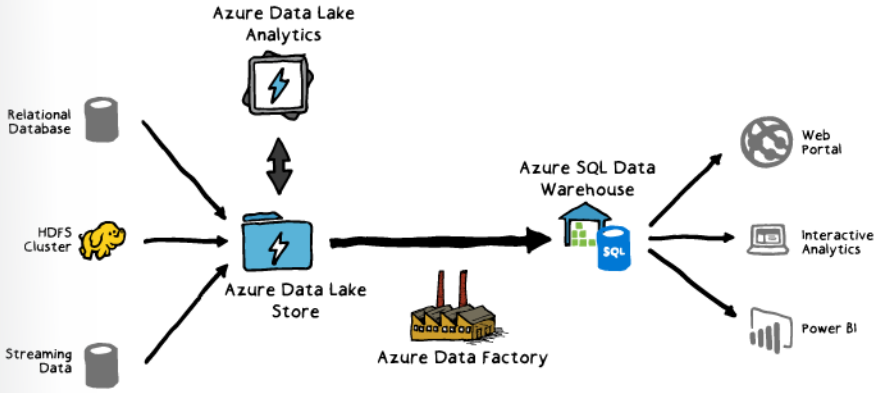
To give an example of the uses of Azure Data Factory imagine a social media company that collects petabytes of member information and logs that are produced by members in the cloud. The company would want to examine to gain insights on member preference, usage behavior, and demographics. The company would also want to discover opportunities to cross-sell or up-sell, generate new features, provide an excellent experience for the members, and drive business growth.

To examine this information the company needs to make use of various reference data. This data includes the member's information, marketing campaign information, and information on the social media platform which is all stored in an on-premises data store. The company could apply data from an additional log data from a cloud store, combining it with the on-premises data store.

When using Azure Data Factory, your data-driven workflows (better known as pipelines) typically performs the following four steps: Connect & Collect, Transform & Enrich, Publish, and Monitor. Data Factory connects to all required sources such as file shares, FTP web, databases, and software-as-a-service (SaaS) services. From here you can use Copy Activity within the data pipeline to move data to a centralized data store in the cloud for analysis.

After the raw data has been collected and moved to the centralized data store, Azure Data Factory processes the collected data using compute services such as Spark, Data Lake Analytics, and Machine Learning. The refined data, now in a business-ready consumable form, is loaded to an analytics engine such as Azure SQL database to be published to be used with your business intelligence tools. After building and deploying the data integration pipeline, you can monitor the scheduled activities and pipelines for failure and success rates.

As seen in the graphic below, Azure Data Factory is an option to migrate data from Azure Data Lake to Azure SQL Data Warehouse.



Data Analysis

Data Analysis Options

After completing this section you will be able to:

• Design a solution that uses Azure Analysis Services for end-user ad-hoc data set queries.

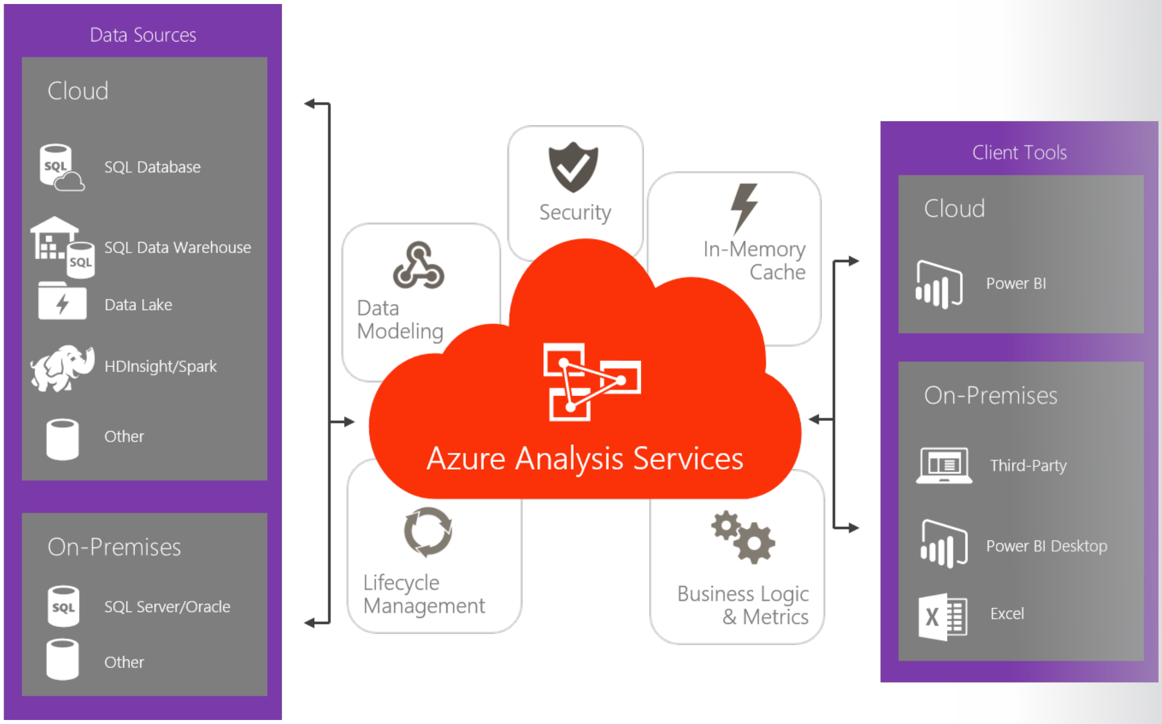
• Design a solution that surfaces "tribal knowledge" using Azure Data Catalog.

• Detail how Azure HDInsight can be integrated into solutions using various Azure Data storage, integration or analysis services.

Data Analysis Options

Azure Analysis Services provides enterprise-grade data modeling in the cloud. It is a fully managed platform as a service (PaaS), integrated with Azure data platform services.

With Analysis Services, you can mashup and combine data from multiple sources, define metrics, and secure your data in a single, trusted semantic data model. The data model provides an easier and faster way for your users to browse massive amounts of data with client applications like Power BI, Excel, Reporting Services, third-party, and custom apps.



Azure Analysis Services is compatible with many features already in SQL Server Analysis Services Enterprise Edition. Azure Analysis Services supports tabular models at the 1200 and 1400 compatibility levels. Partitions, row-level security, bi-directional relationships, and translations are all supported. In-memory and DirectQuery modes are also available for fast queries over massive and complex datasets.

For developers, tabular models include the Tabular Object Model (TOM) to describe model objects. TOM is exposed in JSON through the Tabular Model Scripting Language (TMSL) and the AMO data definition language.

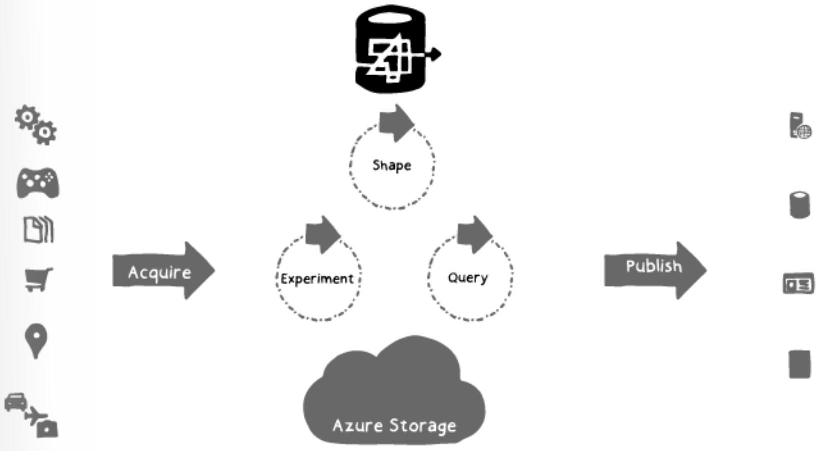
HDInsight

Apache Hadoop was the original open-source framework for distributed processing and analysis of big data sets on clusters. The Hadoop technology stack includes related software and utilities, including Apache Hive, HBase, Spark, Kafka, and many others. To see available Hadoop technology stack components on HDInsight, see Components and versions available with HDInsight. To read more about Hadoop in HDInsight, see the **Azure features page for HDInsight**[1].

Apache Spark is an open-source parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications.

Azure HDInsight is the Azure distribution of the Hadoop components from the Hortonworks Data Platform (HDP). Azure HDInsight makes it easy, fast, and cost-effective to process massive amounts of data. You can use the most popular open-source frameworks such as Hadoop, Spark, Hive, LLAP, Kafka, Storm, R, and more to enable a broad range of scenarios such as extract, transform, and load (ETL); data warehousing; machine learning; and IoT.

HDInsight is big data processing on top of Azure Storage.



Azure Data Catalog

Azure Data Catalog is a fully managed cloud service whose users can discover the data sources they need and understand the data sources they find. At the same time, Data Catalog helps organizations get more value from their existing investments.

With Data Catalog, any user (analyst, data scientist, or developer) can discover, understand, and consume data sources. Data Catalog includes a crowdsourcing model of metadata and annotations. It is a single, central place for all of an organization's users to contribute their knowledge and build a community and culture of data.

Data Catalog provides a cloud-based service into which a data source can be registered. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data-source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

After a data source has been registered, its metadata can then be enriched, either by the user who registered it or by other users in the enterprise. Any user can annotate a data source by providing descriptions, tags, or other metadata, such as documentation and processes for requesting data source

[1] https://azure.microsoft.com/en-us/services/hdinsight/

access. This descriptive metadata supplements the structural metadata (such as column names and data types) that's registered from the data source.

Discovering and understanding data sources and their use is the primary purpose of registering the sources. Enterprise users might need data for business intelligence, application development, data science, or any other task where the right data is required. They can use the Data Catalog discovery experience to quickly find data that matches their needs, understand the data to evaluate its fitness for the purpose, and consume the data by opening the data source in their tool of choice.

At the same time, users can contribute to the catalog by tagging, documenting, and annotating data sources that have already been registered. They can also register new data sources, which can then be discovered, understood, and consumed by the community of catalog users.

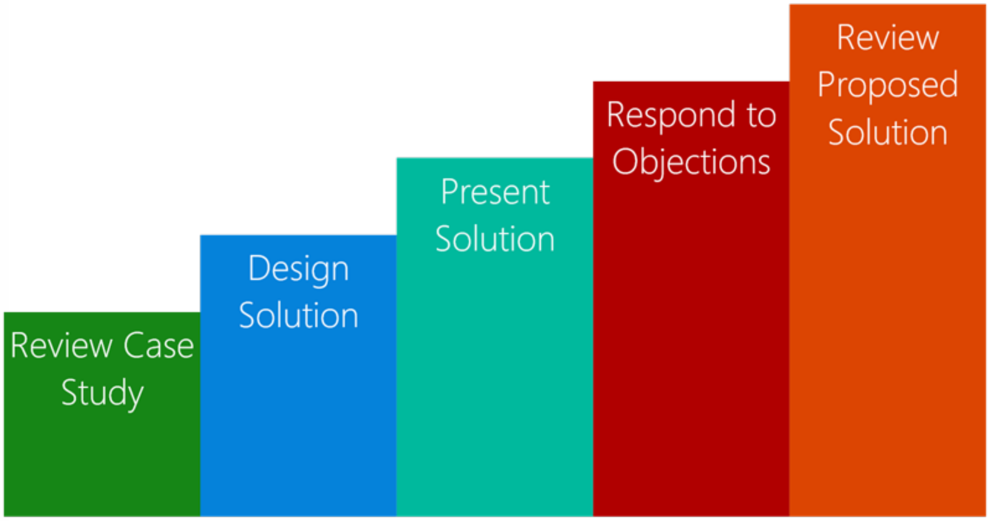
Web Apps & SQL Case Study

Case Study Overview

In this case study, we will look at a customer problem that requires an architectural recommendation.

After this case study, you should:

- Identify customer problems as they are related to networking.
- Design a solution that will meet the customer's objectives.
- Ensure your designed solution accounts for customer objections.



Who Is the Customer?

Founded in 1954, Adventure Works Cycles has grown from a boutique manufacturer of high-quality bicycles and parts into one of the world's largest makers of premium race and commuter bicycles. The Adventure Works mission has remained the same: to passionately pursue advanced, innovative technologies that help cyclists of all abilities find more enjoyment in the sport. Adventure Works Cycles Company manufactures and sells bicycles, bicycle parts, and bicycle accessories under the Adventure Works Cycles and Tailspin brands worldwide.

During the global recession of 2009, most of the company's IT infrastructure was located in the company's Provo, Utah, headquarters, but Adventure Works also had a sizable third-party colocation datacenter, costing US $30,000 to $40,000 a month, and other servers scattered around the United States.

Adventure Works knows that its datacenters are filled with dozens of smaller web servers and databases that run on underutilized hardware, and it has customer data scattered in multiple places. Faced with the prospect of a very large capital expenditure owing to the fact that the vast majority of their servers are now due for a hardware refresh, Adventure Works is looking for other options that eliminate the costly and high risk hardware refresh cycles.

What Does the Customer Already Have?

In reviewing all of Adventure Works' web applications, it was determined that there were three archetypical database backed web applications present:

• **Product Catalog:** Resellers and consumers in North America, Asia and Europe access the product catalog via a website, the data for which is currently stored in SQL Server. Currently, both web app and database are hosted in its Utah datacenter. The database is 50 GB in size, and is not expected to grow past 100 GB.

• **Inventory**: The inventory web application is a mission critical system primarily used by operations running in North America. Currently, both web app and SQL Server database are hosted in its Utah datacenter. This is Adventure Works' largest database at 3 TB in size, but it is not expected to grow beyond 5 TB.

• **Departmental**: These web applications support the regional offices only. Both web app and database are hosted in the Utah data center. While individually these have web applications have low demands and data sizes in the 500 MB-1 GB range, Adventure Works has 100's of SQL Server databases supporting the numerous web apps, and fully expects new databases to be created to support departmental efforts.

Most of Adventure Works developers are already trained in Microsoft tools as all of Adventure Works' web applications and services are built using .NET. Additionally, they have already deployed Active Directory and are currently using a replicated directory in support of departments within each regional office.

What Is the Customer's Goal?

Not only are all the servers expensive to acquire and maintain, but scaling the infrastructure takes significant time. "During and after a high-profile race, there's a great deal of interest in our product, and our web apps receive a lot of hits," says Hayley Leigh, Manager of Solution Development for Adventure Works Cycles. "It is difficult to scale our web hosting environment fast enough, and consumers and resellers could experience slow response times and even downtime." Another challenge: Adventure Works conducts weekly server hardware maintenance, which causes downtime for some of its global offices.

Adventure Works wants to move many consumer-facing web apps, enterprise databases, and enterprise web services to Azure. "By using Microsoft global datacenters, we're able to move infrastructure for key applications and web apps closer to the people who use them," Leigh says. A big problem for Adventure Works is resellers and consumers in Japan and China have to use applications that run in a Utah datacenter, and because of the distance, encounter performance problems. Adventure Works would like to resolve this without the difficulty, expense and time requirements incurred by setting up infrastructure on the other side of the world using Adventure Works-owned servers.

What Does the Customer Need?

In reviewing all of Adventure Works' web applications, it was determined that there were three archetypical database backed web applications present:

• **Product Catalog**: Resellers and consumers in North America, Asia and Europe access the product catalog via a website, the data for which is currently stored in SQL Server. Currently, both web app and database are hosted in its Utah datacenter. The database is 50 GB in size, and is not expected to grow past 100 GB.

• **Inventory**: The inventory web application is a mission critical system primarily used by operations running in North America. Currently, both web app and SQL Server database are hosted in its Utah datacenter. This is Adventure Works' largest database at 3 TB in size, but it is not expected to grow beyond 5 TB.

• **Departmental**: These web applications support the regional offices only. Both web app and database are hosted in the Utah data center. While individually these have web applications have low demands and data sizes in the 500 MB-1 GB range, Adventure Works has 100's of SQL Server databases supporting the numerous web apps, and fully expects new databases to be created to support departmental efforts.

Most of Adventure Works developers are already trained in Microsoft tools as all of Adventure Works' web applications and services are built using .NET. Additionally, they have already deployed Active Directory and are currently using a replicated directory in support of departments within each regional office.

What Things Worry the Customer?

Scale & Performance

- I do not want to have to make code changes (or re-deploy) in order to change the scale of a website.
- I hear Azure Web Apps is only useful for web apps with small amounts of traffic; will it really support the heavy traffic we receive?
- We would prefer to avoid performing a database migration (e.g., to another server) in order to scale the throughput of our database.
- We have heard SQL Database does not provide consistent performance, is this true?

Business Continuity

- How can we certain our data will survive in the event of a catastrophe in a certain part of the world?
- We need to be able to recover from mistakes made by administrators that accidentally delete production data (we know they happen, we would love an “undo”).
- Do we need to have multiple web server instances for each property to have a high SLA?

Tool Familiarity

- Will we need to learn new tools to develop for Azure Web Apps and SQL Database?
- What about diagnosing problems? Are there new tools we need purchase and learn?

Connectivity

- Some of our enterprise web services need to access data and other services located on-premises, is this supported?
- How can we ensure we are delivering the lowest latency possible to our website visitors?
- We need to ensure that if we have multiple web servers backing a given website, that no one web server gets all the traffic.

Management

- We would prefer not to have to manage patching of web servers and databases.
- With all of our web apps and databases around the world, how do we keep tabs on which is up and which is down and which is struggling?
- We need a simple solution to schedule and automate backup of the website and database.

Security

- Is it possible to allow our visitors to use a mix of legacy and modern browsers and still provide for secure transactions?
- What does Azure offer to help us with auditing access to our web servers and databases?

- Our staff is accustomed to accustomed to a single sign-on experience—will this still be possible?

Case Study Solution

Preferred Target Audience

Hayley Leigh, Manager of Solution Development for Adventure Works Cycles

The primary audience is the business decision makers and technology decision makers. From the case study scenario, this would include the Manager of Solution Development. Usually we talk to the Infrastructure Managers who report into the CIO's, or to application sponsors (like a VP LOB, CMO) or to those that represent the Business Unit IT or developers that report into application sponsors.

Preferred Solution

Adventure Works Cycles decided that cloud computing could solve just about all these problems. Moving some workloads into a cloud environment—where virtualized compute, storage, and network resources run in public datacenters, are shared by multiple parties, and are delivered over the internet—could reduce costs, improve scalability and business agility, and enable the desktop management team to manage remote computers.

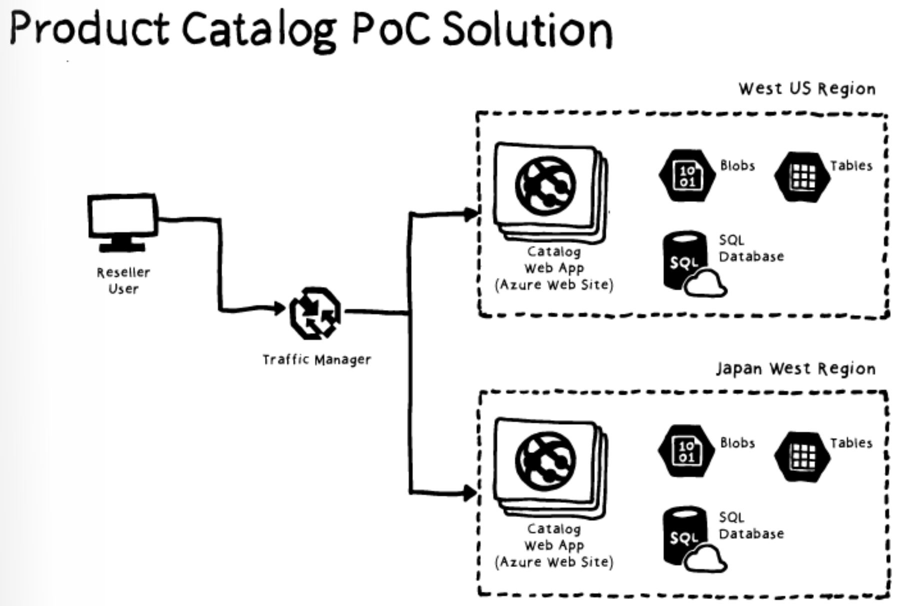
Cowles's team evaluated cloud offerings from multiple providers and selected Microsoft Azure, the Microsoft cloud platform that provides on-demand compute, storage, content delivery, and networking capabilities from Microsoft datacenters. "The other companies offered infrastructure-as-a-service but not software-as-a-service or platform-as-a-service as Microsoft did," Cowles says. "We wanted the whole spectrum of cloud options to fit a range of cloud needs. Also, most of our developers are trained in Microsoft tools, so it was much easier for us to connect our in-house systems to Microsoft Azure."

Adventure Works then moved many consumer-facing web apps, enterprise databases, and enterprise web services to Azure. "By using Microsoft global datacenters, we're able to move infrastructure for key applications and web apps closer to the people who use them," Cowles says. For example, resellers and consumers in Japan and China previously had to use applications that ran in a Utah datacenter, and because of the distance, encountered performance problems. In about an hour, Cowles's team set up the same services in a Microsoft datacenter in Asia and completely eliminated the performance lags. "It would have been extremely difficult, expensive, and time-consuming to set up this infrastructure on the other side of the world using Adventure Works-owned servers," Cowles says.

Proof of Concept (PoC)

The PoC scenario would be to address the latency issue faced by resellers around the world that currently rely on to access the Utah datacenter. The PoC would address showing how by using Azure Traffic Manager along with Azure Web Apps in multiple regions, one can ensure that these users are always being routed to the Web app that is "closest" to them in terms of minimizing network latency. The PoC could demonstrate this by provisioning a representative portion of the Product Catalog web application in an Azure Website, along with the storage required for it (e.g., Blob Storage, Table Storage and SQL Database) in the West US region, and then also provisioning the same infrastructure in the Japan West region. Traffic Manager would then be configured to route traffic using the Performance load balancing method.

The following diagram illustrates this:

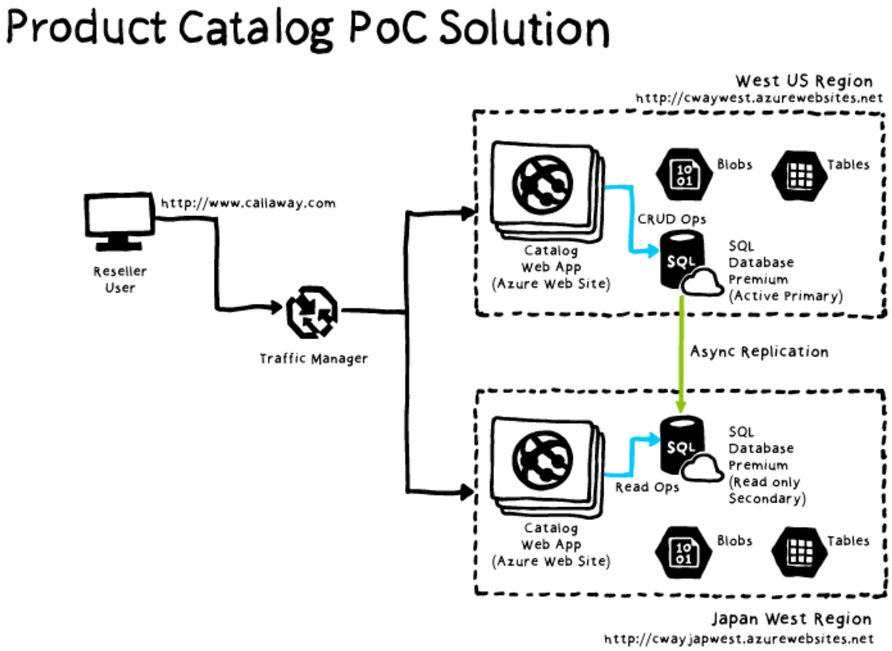


GEO-DISPERSE WEB APPLICATION

To complete the PoC, either Adventure Works staff or resellers located near each region could be asked to visit the PoC reseller web app, measure their experience (e.g., page load times, time to first paint, etc.) and report on the perceived performance (e.g., does it feel more responsive than the current site?).

On the database side, the PoC should be augmented to use the SQL Database Premium tier. The PoC would address showing using SQL Database Premium along with Azure Web Apps in multiple regions, whereby one configures SQL Database Premium to provide Active Geo-Replication across two regions (West US and Japan West as examples). SQL Database Premium could be configured with the primary in the West US Region, and a readable replica available in the Japan West Region.

The following diagram illustrates this:


Product Catalog PoC Solution
West US Region
http://cwaywest.azurewebsites.net
http://www.callaway.com
Reseller User
Traffic Manager
Catalog Web App (Azure Web Site)
Blobs
Tables
CRUD Ops
SQL
SQL Database Premium (Active Primary)
Async Replication
SQL
SQL Database Premium (Read only Secondary)
Read Ops
Catalog Web App (Azure Web Site)
Blobs
Tables
Japan West Region
http://cwayjapwest.azurewebsites.net

SYNCING DATABASE SOURCES

By approaching it this way, the PoC could explore the benefits of accessing read-only data from the database nearest the Web app, while reserving data modifications operations for the active primary database instance.

To complete the PoC, Adventure Works could simulate a regional failure and switch to using the SQL Database in the Japan West region as the active primary. They could go one step further and stop the Web Apps in the West US region and thereby simulate a complete regional failure of the West US region. By having Traffic Manager in place, all traffic would flow to the Japan West Region, and the active database would be in that region as well.

Risks & Mitigation

By demonstrating the performance load balancing method, the PoC helps Adventure Works to immediately establish confidence in Azure's global capabilities to deliver a high performance experience to their customers and resellers around the world. By demonstrating the fail over between regions, the PoC helps Adventure Works to immediately establish confidence in Azure's global capabilities to deliver a highly available experience to their customers and resellers around the world, even in the face of a regional failure.

By migrating the Product Catalog web app, it can also help to mitigate the risk the components used by the reseller website are incompatible with Azure Web Apps (for example, if one of the components used by the web app application requires an MSI based install).

This PoC is a good opportunity to evaluate Adventure Works's use of "naked domains" (e.g., a domain with no "www" subdomains) for the product catalog, because Azure Traffic Manager cannot directly resolve requests against naked domains. If naked domains are required, Adventure Works would need to configure DNS forwarding with their DNS provider (if their DNS provider supports it) or leverage another DNS forwarding service to redirect naked domain requests to resource having a subdomain.

By migrating a representative portion of the Product Catalog Web App and its data, it can also help to mitigate the risk that SQL Server features used by the database are incompatible with SQL Database (for example, if they are using CLR stored procedures).

Success Criteria

The Product Catalog Web Application (or portion thereof) and supporting data can be successfully migrated to Azure. Users in each region are reporting improved performance with reduced latency. When a failover is simulated, the web app remains functional (albeit users may experience greater latency).

Implementation

Inventory Web App

For the inventory web app, whose database sees large database sizes (3-5 TB), only SQL Server in a VM can scale to that capacity. SQL Database is not a good recommendation here for two reasons:

• SQL Database has a maximum of 500 GB per database

• To grow beyond this using SQL Database would require sharding, which would require re-designing the web application. One of the customer needs is to avoid such re-architecting of the database structure and the making of large changes to the application.

Departmental Web App

For the departmental web apps, the database sizes were small (500 MB to 1 GB). In choosing between SQL Database and SQL in a VM, one should consider the ease of migrating brownfield apps. The customer stated having 100's of such databases, so moving these to SQL Database could prove overly expensive and time consuming on account of any incompatibilities that would need to be worked thru. Therefore, for the existing departmental apps, these databases should be created with SQL Server in a VM.

For new, greenfield, applications that do not have such incompatibilities, the departmental customer is very likely to appreciate the minimal IT involvement required: to quickly provision a SQL Database for their departmental application without IT support, to benefit from the Point-in-Time Restore should a departmental user make a mistake that results in data loss, and to have high availability within a single data center without the extra configuration that setting up a SQL Server cluster or Availability Group would require. Given the small database sizes, most departmental solutions could leverage the cost efficient SQL Database Basic tier.

Checklist of Preferred Objection Handling

Scale & Performance

I do not want to have to make code changes (or re-deploy) in order to change the scale of a website.

- With Azure Web Apps, you do not need to make changes or re-deploy in order to change the scale of a Web app.

I hear Azure Web Apps is only useful for web apps with small amounts of traffic; will it really support the heavy traffic we receive?

- Azure Web Apps is capable of supporting Web Apps with loads ranging from small amounts of traffic to large amounts of traffic—this is enabled by its ability to scale up the instance size and to scale out the number of instances in order to meet demand.

We would prefer to avoid performing a database migration (e.g., to another server) in order to scale the throughput of our database.

- You can move between SQL Database service tiers or performance levels using the Azure Portal or Azure PowerShell without having to migrate the database.

We have heard SQL Database does not provide consistent performance, is this true?

- It was true with Web and Business editions, but these are now in the process of being deprecated. The SQL Database Basic, Standard and Premium offerings allow you to purchase a database meeting specific performance criteria.

Business Continuity

How can we certain our data will survive in the event of a catastrophe in a certain part of the world?

- By deploying Web Apps to multiple regions and using Azure Traffic Manager to route between them for performance, you will still get the benefit failing over to another web app in another region should all of the web app endpoints in one region become unavailable. It is worth noting that such a failover may not be to the next "closest" region in such scenarios.
- With Active Geo-Replication, a feature of SQL Database Premium, you can create and maintain up to four readable secondary databases across geographic regions. All transactions applied to the primary database are replicated to each of the secondary databases. The secondary databases can be used for read workloads, database migration, and protection against data loss during application upgrade as a failback option.
- SQL Database Standard provides Standard Geo-Replication which enables you to have a single offline secondary in a pre-set region that is different from the active primary. This secondary is offline in the sense that, while it is synchronized with the primary, it will refuse any other connections until a failover happens and the datacenter hosting the primary becomes unavailable.
- Geo-Restore is a feature available to Basic, Standard and Premium SQL Database. It enables you to request a restore of your database using the latest weekly full backup plus differential backup, to any server in any Azure region. These backups are stored in geographically redundant storage.

We need to be able to recover from mistakes made by administrators that accidentally delete production data (we know they happen, we would love an "undo").

- Microsoft Azure SQL Database creates backups of your data, and gives you the ability to recover your data from unwanted deletions or modifications. With Point in Time Restore on SQL Database Premium, you can restore to a database state as far back as 35 days.

Do we need to have multiple web server instances for each property to have a high SLA?

- Unlike other Azure compute services, Web Apps provides a high availability SLA using only a single standard instance.

Tool Familiarity

Will we need to learn new tools to develop for Azure Web Apps?

- No. You can use familiar tools like Visual Studio to develop for Azure Web Apps.
- Visual Studio and SQL Server Management Studio can both be used to manage SQL Database.

What about diagnosing problems? Are there new tools we need purchase and learn?

- No. While there are new tool options, such as using the Azure Portal, Server Control Manager (Kudu) or examining logs stored in Blob or Table storage, when it comes to diagnosing problems you can still use familiar tools like Visual Studio.
- SQL Server Management Studio and Dynamic Management Views can be used to diagnose problems with SQL Database and SQL Server in a VM.

Connectivity

Some of our enterprise web services need to access data and other services located on-premises, is this supported?

- Yes. Using the Service Bus Relay, Virtual Networks VPN or ExpressRoute access to data and services located on-premises is made possible.

How can we ensure we are delivering the lowest latency possible to our website visitors?

- Use Traffic Manager with the Performance load balancing method and deploy your solution to Web Apps in multiple regions nearest to your web app visitor populations.

We need to ensure that if we have multiple web servers backing a given website, that no one web server gets all the traffic.

- Within a datacenter, the Azure load balancer automatically handles round-robin load balancing between instances.
- Outside the datacenter, Traffic manager can be used in the Round-Robin load balancing method to route requests across data centers.

Management

We would prefer not to manage patching of web servers and databases.

- With Azure Web Apps, patching of the underlying virtual machines is performed automatically and is transparent to you.
- With SQL Database, patching of the host OS and the database is handled for you.

With all of our web apps and databases around the world, how do we keep tabs on which is up and which is down and which is struggling?

- Azure Web Apps provides support for end point monitoring, which enables you to collect responsiveness metrics of a given Web app from multiple endpoints around the world.
- You can monitor these metrics and also configure the sending of alerts when certain thresholds are exceeded using the Azure Portal.
- You can acquire an overview of Database health for each of your SQL Databases from the Azure Portal.

We need a simple solution to schedule and automate backup of the website.

- With Web app Backup, you can create scheduled backups of your Web app. The offline copy is stored in Blob storage.

Security

Is it possible to allow our visitors to use a mix of legacy and modern browsers and still provide for secure transactions?

- Azure Web Apps provides support for both IP Based SSL and SNI SSL. The former is mechanism that should be used to support both legacy and modern browsers.

What does Azure offer to help us with auditing access to our web servers?

- Operations logs are available from the Azure portal that can be used to track various management operations. These logs can be retrieved by REST API for collecting them into more permanent storage.

Our staff is accustomed to accustomed to a single sign-on experience-- will this still be possible?

- Yes, this is possible using Azure Active Directory.

Potential Benefits

By integrating Microsoft cloud solutions into its datacenter strategy, Adventure Works Cycles has been able to reduce IT costs by more than $300,000 a year while gaining greater datacenter scalability and datacenter agility. The IT team can respond to server requests in hours instead of months and "turn off" servers when they are no longer needed. With servers running in Microsoft datacenters around the world, Adventure Works can provide better application performance and availability to offices and customers that are located far from the company's California datacenter.

• Improve Scalability and Agility

Adventure Works has gained a level of IT scalability and agility that it never before had.
Cowles's team can respond much faster to requests for IT resources. "With Azure, we can respond to business requests in hours versus the months that it took before," Cowles says. "It's especially valuable in responding to requests in non-US regions, because it's difficult to set up infrastructure in Carlsbad and provide remote access to those applications."

• Enhance Performance and Availability of Core Business Systems

Since moving important pieces of its infrastructure to Azure, Adventure Works has enjoyed higher overall availability of critical applications and web properties. "With Azure, we no longer have maintenance-related downtime that negatively affects our business around the world," Cowles says.
By migrating its business-to-business website to Microsoft Azure, Adventure Works will eliminate the cost of that on-premises infrastructure and also realize reliability and performance gains. "It's a lot easier to load-balance virtual machines in Azure than physical servers in our datacenter," Cowles says. "We can also mitigate the impact of server security updates. By moving the B2B infrastructure to Azure, we can apply updates only to the regional servers that need them without affecting uptime in all of our offices."

Online Lab - Deploying Database Instances in Azure

Lab Steps

Online Lab: Deploying Database Instances in Azure

NOTE: For the most recent version of this online lab, see: **https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign**.

Before we start

1. Ensure that you are logged in to your Windows 10 lab virtual machine using the following credentials:
   - Username: **Admin**
   - Password: **Pa55w.rd**
2. Review Taskbar located at the bottom of your Windows 10 desktop. The Taskbar contains the icons for the common applications you will use in the labs:
   - Microsoft Edge
   - File Explorer
   - **Visual Studio Code**[2]
   - **Microsoft Azure Storage Explorer**[3]
   - Bash on Ubuntu on Windows
   - Windows PowerShell
3. **Note**: You can also find shortcuts to these applications in the **Start Menu**.

[4]

Exercise 1: Deploy a Cosmos DB database

[5]

Task 1: Open the Azure Portal

1. On the Taskbar, click the **Microsoft Edge** icon.
2. In the open browser window, navigate to the **Azure Portal** (**https://portal.azure.com**).

[2] https://code.visualstudio.com/
[3] https://azure.microsoft.com/features/storage-explorer/
[4] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#exercise-1-deploy-a-cosmos-db-database
[5] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-1-open-the-azure-portal

3. If prompted, authenticate with the user account account that has the owner role in the Azure subscription you will be using in this lab.

[6]

Task 2: Create a Cosmos DB database and collection

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Cosmos DB** and press **Enter**.
3. On the **Everything** blade, in the search results, click **Azure Cosmos DB**.
4. On the **Azure Cosmos DB** blade, click the **Create** button.
5. On the new **Azure Cosmos DB** blade, perform the following tasks:
   - Leave the **Subscription** drop-down list entry set to its default value.
   - Resource group: ensure that the **Create new** option is selected and then, in the text box, type **AADesignLab0701-RG**.
   - In the **Account Name** text box, type a globally unique value.
   - In the **API** drop-down list, select the **Core (SQL)** option.
   - In the **Location** drop-down list, select the Azure region in which you want to deploy resources in this lab.
   - Leave all remaining settings with their default values.
   - Click the **Create** button.
6. Wait for the provisioning to complete before you proceed to the next step.
7. **Note**: The deployment should take less than 5 minutes.
8. At the top of the portal, click the **Cloud Shell** icon to open a new shell instance.
9. **Note**: The **Cloud Shell** icon is a symbol that is constructed of the combination of the *greater than* and *underscore* characters.
10. If this is your first time opening the **Cloud Shell** using your subscription, you will see a wizard to configure **Cloud Shell** for first-time usage. When prompted, in the **Welcome to Azure Cloud Shell** pane, click **Bash (Linux)**.
11. **Note**: If you do not see the configuration options for **Cloud Shell**, this is most likely because you are using an existing subscription with this course's labs. If so, proceed directly to the next task.
12. In the **You have no storage mounted** pane, click **Show advanced settings**, perform the following tasks:
    - Leave the **Subscription** drop-down list entry set to its default value.
    - In the **Cloud Shell region** drop-down list, select the Azure region matching or near the location that you selected earlier in this task.
    - In the **Resource group** section, select the **Use existing** option and then, in the drop-down list, select **AADesignLab0701-RG**.

[6] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-2-create-a-cosmos-db-database-and-collection

- In the **Storage account** section, ensure that the **Create new** option is selected and then, in the text box below, type a unique name consisting of a combination of between 3 and 24 characters and digits.
- In the **File share** section, ensure that the **Create new** option is selected and then, in the text box below, type **cloudshell**.
- Click the **Create storage** button.

13. Wait for the **Cloud Shell** to finish its first-time setup procedures before you proceed to the next task.
14. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the name of the resource group that contains the Azure Cosmos DB account you deployed earlier in this task:

```
RESOURCE_GROUP='AADesignLab0701-RG'
```

15. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the name of the CosmosDB account you created earlier in this task:

```
COSMOSDB_NAME=$(az cosmosdb list --resource-group $RESOURCE_GROUP --query
"[0].name" --output tsv)
```

16. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the primary key of the CosmosDB account you created earlier in this task:

```
PRIMARY_KEY=$(az cosmosdb list-keys --resource-group $RESOURCE_GROUP --name
$COSMOSDB_NAME | jq -r '.primaryMasterKey')
```

17. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the URI of the CosmosDB account you created earlier in this task:

```
URI="https://$COSMOSDB_NAME.documents.azure.com:443/"
```

18. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a new CosmosDB database named **FinancialClubDatabase**:

```
az cosmosdb database create --url-connection $URI --key $PRIMARY_KEY --db-
name 'FinancialClubDatabase'
```

19. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a fixed collection named **MemberCollection** in the newly created database:

```
az cosmosdb collection create --url-connection $URI --key $PRIMARY_KEY
--db-name 'FinancialClubDatabase' --collection-name 'MemberCollection'
--throughput 400
```

20. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to display the value of the PRIMARY_KEY variable:

```
echo $PRIMARY_KEY
```

21. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to display the value of the URI variable:

```
echo $URI
```

22. **Note**: Take a note of these values - you will need them in the next exercise.

[7]

Task 3: Create and query documents in Cosmos DB

1. On the left side of the Azure Cosmos DB account blade, click **Data Explorer**.
2. In the **Data Explorer** pane, click the **MemberCollection** child node of the **FinancialClubDatabase** node.
3. Click the **New SQL Query** button at the top of the **Data Explorer** pane.
4. In the **Query 1** tab that opened, view the default query:

```
SELECT * FROM c
```

5. Click the **Execute Query** button at the top of the query editor.
6. In the left pane of the Data Explorer, expand the **MemberCollection** node.
7. Click the **Documents** child node within the **MemberCollection** node.
8. In the new **Documents** tab that opened, click the **New Document** button at the top of the tab.
9. In the **Documents** tab, replace the existing document with the following document:

```
{
    "firstName": "Pennington",
    "lastName": "Oneal",
    "age": 26,
    "salary": 90000.00,
    "company": "Veraq",
    "isVested": false
}
```

10. Click the **Save** button at the top of the **Documents** tab.
11. In the **Documents** tab, click the **New Document** button at the top of the tab.
12. In the **Documents** tab, replace the existing document with the following document:

```
{
    "firstName": "Suzanne",
    "lastName": "Oneal",
    "company": "Veraq"
}
```

[7] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-3-create-and-query-documents-in-cosmos-db

13. Click the **Save** button at the top of the **Documents** tab.
14. Switch back to the **Query 1** tab, re-run the default query `SELECT * FROM c` by clicking the **Execute Query** button at the top of the query editor, and review the results.
15. In the query editor, replace the default query with the following query:

```
SELECT
    c.id,
    c.firstName,
    c.lastName,
    c.isVested,
    c.company
FROM
    c
WHERE
    IS_DEFINED(c.isVested)
```

16. Click the **Execute Query** button at the top of the query editor and review the results.
17. In the query editor, replace the existing query with the following query:

```
SELECT
    c.id,
    c.firstName,
    c.lastName,
    c.age
FROM
    c
WHERE
    c.age > 20
```

18. Click the **Execute Query** button at the top of the query editor and review the results.
19. In the query editor, replace the existing query with the following query:

```
SELECT VALUE
    c.id
FROM
    c
```

20. Click the **Execute Query** button at the top of the query editor and review the results.
21. In the query editor, replace the existing query with the following query:

```
SELECT VALUE {
    "badgeNumber": SUBSTRING(c.id, 0, 8),
    "company": c.company,
    "fullName": CONCAT(c.firstName, " ", c.lastName)
} FROM c
```

22. Click the **Execute Query** button at the top of the query editor and review the results.

Review: In this exercise, you created a new Cosmos DB account, database, and collection, added sample documents to the collection, and run sample queries targeting these documents.

[8]

Exercise 2: Deploy Application using Cosmos DB

[9]

Task 1: Deploy API App code using Azure Resource Manager templates and GitHub

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Template Deployment** and press **Enter**.
3. On the **Everything** blade, in the search results, click **Template Deployment**.
4. On the **Template deployment** blade, click the **Create** button.
5. On the **Custom deployment** blade, click the **Build your own template in the editor** link.
6. On the **Edit template** blade, click the **Load file** link.
7. In the **Open** file dialog that appears, navigate to the **\allfiles\AZ-301T02\Module_02\LabFiles\Starter\** folder.
8. Select the **api.json** file.
9. Click the **Open** button.
10. Back on the **Edit template** blade, click the **Save** button to persist the template.
11. Back on the **Custom deployment** blade, perform the following tasks:
    - Leave the **Subscription** drop-down list entry set to its default value.
    - In the **Resource group** section, select the **Use existing** option and then, in the drop-down list, select **AADesignLab0701-RG**.
    - In the **Terms and Conditions** section, click the **I agree to the terms and conditions stated above** checkbox.
    - Click the **Purchase** button.
12. Wait for the deployment to complete before you proceed to the next task.
13. **Note**: Deployment from source control can take up to 10 minutes.

[8] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#exercise-2-deploy-application-using-cosmos-db

[9] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-1-deploy-api-app-code-using-azure-resource-manager-templates-and-github

[10]

Task 2: Validate API App

1. In the hub menu in the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab0701-RG**.
3. On the **AADesignLab0701-RG** blade, click the entry representing the newly created App Service API app.
4. On the API app blade, click **Application Settings**.
5. On the Application Settings blade, scroll down to the **Application settings** section and perform the following tasks:
    - Set the value of the **CosmosDB:AuthorizationKey** setting to the value of the **PRIMARY KEY** setting of the **Cosmos DB** account you created earlier in this lab.
    - Update the value of the **CosmosDB:EndpointUrl** setting to the value of the **URI** setting of the **Cosmos DB** instance you created earlier in this lab.
    - Click the **Save** button at the top of the pane.
6. On the left-side of the API app blade, click **Overview**.
7. Click the **Restart** button at the top of the blade and, when prompted to confirm, click **Yes**.
8. Click the **Browse** button at the top of the blade. This will open a new browser tab displaying the **Swagger UI** homepage.
9. **Note**: If you click the **Browse** button before the API app has fully restarted, you may not be able to follow the remaining steps in this task. If this happens, refresh your browser until the API app is running again.
10. On the **Swagger UI** homepage, click **GET/Documents**.
11. Click the **Try it out!** button.
12. Review the results of the request.
13. Back on the **Swagger UI** homepage, click **POST/Populate**.
14. In the **Parameters** section, in the **Value** field for the **options** parameter, paste in the following JSON content:

    ```
    {
        "quantity": 50
    }
    ```

15. In the **Response Messages** section, click the **Try it out!** button.
16. Review the results of the request.
17. Back on the **Swagger UI** homepage, click **GET/Documents**.
18. Locate the **Response Messages** section. Click the **Try it out!** button.
19. Review the results of the request.

[10] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-2-validate-api-app

20. Close the new browser tab and return to the browser tab displaying the Azure portal.

Review: In this exercise, you created a new API App that uses the .NET Core DocumentDB SDK to connect to Azure Cosmos DB collection and manage its documents.

[11]

Exercise 3: Connect Cosmos DB to Azure Search

[12]

Task 1: Create Azure Search Instance

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Search** and press **Enter**.
3. On the **Everything** blade, in the search results, click **Azure Search**.
4. On the **Azure Search** blade, click the **Create** button.
5. On the **New Search Service** blade, perform the following tasks:
   - In the **URL** text box, enter a globally unique name. Record its value. You will use it later in this lab.
   - Leave the **Subscription** drop-down list entry set to its default value.
   - In the **Resource group** section, select the **Use existing** option and then, in the drop-down list, select **AADesignLab0701-RG**.
   - In the **Location** drop-down list, select the Azure region matching or near the location where you deployed Cosmos DB resource earlier in this labb
   - Click **Pricing tier**.
   - On the **Choose your pricing tier** blade, click **Free** and then click the **Select** button.
   - Click the **Create** button.
6. Wait for the provisioning to complete before you proceed to the next step.
7. In the hub menu in the Azure portal, click **Resource groups**.
8. On the **Resource groups** blade, click **AADesignLab0701-RG**.
9. On the **AADesignLab0701-RG** blade, click the entry representing the newly created Azure Search instance.
10. On the Search service blade, click **Keys**.
11. In the **Keys** pane, record the value of **PRIMARY ADMIN KEY**. You will use it later in this lab.

[11] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#exercise-3-connect-cosmos-db-to-azure-search

[12] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-1-create-azure-search-instance

[13]

Task 2: Index Cosmos DB Data in Azure Search

1. In the hub menu in the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab0701-RG**.
3. On the **AADesignLab0701-RG** blade, click the entry representing the Azure Cosmos DB account you created earlier in this lab.
4. On the Azure Cosmos DB account blade, click **Add Azure Search**.
5. On the **Import data** blade, click **Search service** and, on the **Search services** blade, click the newly created Azure Search Service instance.
6. **Note**: You will be presented with two blades: the **Data Source** blade with the **CosmosDB** option already selected and with the **New data source** blade.
7. On the **New data source** blade, perform the following tasks:
   - In the **Name** text box, type **cosmosdata**.
   - In the **Connection string** text box, accept the default entry.
   - In the **Database** drop-down list, select the **FinancialClubDatabase** entry.
   - in the **Collection** drop-down list, select the **MemberCollection** entry.
   - In the **Query** field, enter the following SQL query:
   - SELECT c.id, c.firstName, c.lastName, c.age, c.salary, c.company, c.isVested, c._ts FROM c WHERE c._ts >= @HighWaterMark ORDER BY c._ts
   - Ensure that the **Query results ordered by _ts** checkbox is selected.
   - Click the **OK** button.
8. On the **Cognitive Search** blade, click the **OK** button.
9. On the **Index** blade, perform the following tasks:
   - In the **Index name** text box, type **memberindex**.
   - In the **Key** drop-down list, ensure that the **id** entry is selected.
   - For the **id** field in the table, ensure that the **RETRIEVABLE**, **FILTERABLE**, and **SORTABLE** checkboxes are selected.
   - For the **firstName** field in the table, ensure that the **RETRIEVABLE**, **SORTABLE**, and **SEARCHABLE** options are selected.
   - For the **lastName** field in the table, ensure that the **RETRIEVABLE**, **SORTABLE**, and **SEARCHABLE** checkboxes are selected.
   - For the **age** field in the table, ensure that the **RETRIEVABLE**, **FILTERABLE**, **SORTABLE**, and **FACETABLE** checkboxes are selected.
   - For the **salary** field in the table, ensure that the **RETRIEVABLE**, **FILTERABLE**, **SORTABLE**, and **FACETABLE** checkboxes are selected.

[13] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-2-index-cosmos-db-data-in-azure-search

- For the **company** field in the table, ensure that the **RETRIEVABLE**, **FACETABLE**, and **SEARCHABLE** checkboxes are selected.
- For the **isVested** field in the table, ensure that the **RETRIEVABLE**, **FILTERABLE**, **SORTABLE**, **FACETABLE** checkboxes are selected.
- Click the **OK** button.

10. On the **Create an Indexer** blade, perform the following tasks:
    - In the **Name** text box, type **cosmosmemberindexer**.
    - In the **Schedule** section, select the **Custom** option.
    - In the **Interval (minutes)** text box, type **5**.
    - In the **Start time (UTC)** field, specify the current date and accept the default value of the time entry.
    - Click the **OK** button.
11. Back on the **Import data** blade, click the **OK** button.

[14]

Task 3: Validate API App

1. In the hub menu in the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab0701-RG**.
3. On the **AADesignLab0701-RG** blade, click the entry representing the App Service API app you created earlier in this lab.
4. On the API app blade, click **Application settings**.
5. On the Application settings blade, scroll down to the **Application settings** section and perform the following tasks:
    - Set the value of the **Search:AccountName** setting to the name of the Azure Search instance you created earlier in this lab.
    - Set the value of the **Search:QueryKey** setting to the value of the **PRIMARY KEY** of the Azure Search instance you created earlier in this lab.
    - Click the **Save** button at the top of the blade.
6. On the API app blade, click **Overview**.
7. Click the **Restart** button at the top of the blade and, when prompted to confirm, click **Yes**.
8. Click the **Browse** button at the top of the blade. This will open a new browser tab displaying the **Swagger UI** homepage.
9. **Note**: If you click the **Browse** button before the API app has fully restarted, you may not be able to follow the remaining steps in this task. If this happens, refresh your browser until the API app is running again.
10. On the **Swagger UI** homepage, click **Cosmos DB API v.1.0.0** at the top of the page and select the **Cosmos DB API v.2.0.0** option from the drop-down list.

[14] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-3-validate-api-app

11. Click **GET/Documents/search**.
12. In the **Parameters** section, in the **Value** text box of the **query** parameter, type the following text:

```
Oneal
```

13. In the **Response Messages** section, click the **Try it out!** button.
14. Review the results of the request.
15. In the **Parameters** section, in the **Value** text box of the **query** parameter, type the following text:

```
penn*
```

16. In the **Response Messages** section, click the **Try it out!** button.
17. Review the results of the request.
18. Close the new browser tab and return to the browser tab displaying the Azure portal.

Review: In this exercise, you created an Azure Search instance that uses an indexer to index the documents in Azure Cosmos DB.

15

Exercise 4: Remove lab resources

16

Task 1: Delete the resource group

1. In the hub menu in the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab0701-RG**.
3. On the **AADesignLab0701-RG** blade, click **Delete resource group**.
4. In the **Are you sure you want to delete "AADesignLab0701-RG"?** pane, in the **TYPE THE RESOURCE GROUP NAME** text box, type **AADesignLab0701-RG** and click **Delete**.

Review: In this exercise, you removed the resources used in this lab.

[15] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#exercise-4-remove-lab-resources

[16] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod02_Deploying%20Database%20Instances%20in%20Azure.md#task-1-delete-the-resource-group

Review Questions

Module 2 Review Questions

Azure SQL Database

You are designing mobile app for your company. You plan to use to Azure SQL Database to support the app.

What are the benefits of using Azure SQL Database? How does Azure SQL Database support scaling? What are some reasons that you would choose an Azure storage solution other than Azure SQL Database?

Suggested Answer ↓

Azure SQL Database offers predictable performance, high compatibility, and simple management. Azure SQL Database provides elastic scaling. This technology uses a strategy known as sharding to simplify scaling.

Azure storage services

You are designing mobile app for your company.

The data store that supports the app must store den-ormalized data to optimize read access.

Which Azure service should you use?

Suggested Answer ↓

The Azure Table Storage service stores large amounts of structured data and is great for storing normalized data for quick reads.

Distributed data storage

You are designing an Azure solution for your company.

The solution requires a globally distributed database solution that supports multiple database types.

Which Azure storage service should you use? What data store types does it support?

Suggested Answer ↓

Azure Cosmos DB is Microsoft's new globally distributed, multi-model database service. Azure Cosmos DB currently supports Document DB, MongoDB, Graph, and Tables.

Module 3 Module Monitoring & Automating Azure Solutions

Monitoring

Azure Network Watcher

This module covers the monitoring and automation solutions available after an Azure solution has been architected, designed and possibly deployed. The module reviews services that are used to monitor individual applications, the Azure platform, and networked components. This module also covers automation and backup options to enable business-continuity scenarios for solutions hosted in Azure.

After completing this module, students will be able to:

- Compare and contrast monitoring services for applications, the Azure platform, and networking.

- Design an alert scheme for a solution hosted in Azure.

- Select the appropriate backup option for infrastructure and data hosted in Azure.

- Automate the deployment of future resources for backup recovery or scaling purposes.

Monitoring

This lesson reviews a variety of services available in Azure to monitor your workloads and applications.

After completing this lesson, you will be able to:

- Select between OMS, Network Watcher, Security Center, Azure Monitor, Azure Service Health and Azure Advisors when determining which monitoring solution to use for workloads deployed to Azure.

- Query historical log data using Power BI.

- Integrate Application Insights into a custom software solution hosted on Azure.

Azure Network Watcher

Customers can build an end-to-end network in Azure by orchestrating and composing various individual network resources such as VNet, ExpressRoute, Application Gateway, Load balancers, and more. Monitoring is available on each of the network resources. We refer to this monitoring as resource level monitoring.

The end to end network can have complex configurations and interactions between resources, creating complex scenarios that need scenario-based monitoring through Network Watcher.

Network Watcher provides the following features:

- Topology
- Variable Packet Capture
- IP Flow Verify
- Next Hop
- Diagnostics Logging
- Security Group View
- NSG Flow Logging
- VPN Gateway Troubleshooting
- Network Subscription Limits
- Role Based Access Control
- Connectivity

Network Monitor

Operations performed as part of the configuration of networks are logged. These logs can be viewed in the Azure portal or retrieved using Microsoft tools such as Power BI or third-party tools. Audit logs are available through the portal, PowerShell, CLI, and Rest API.
Metrics are performance measurements and counters collected over a period of time. Metrics are currently available for Application Gateway. Metrics can be used to trigger alerts based on thresholds.

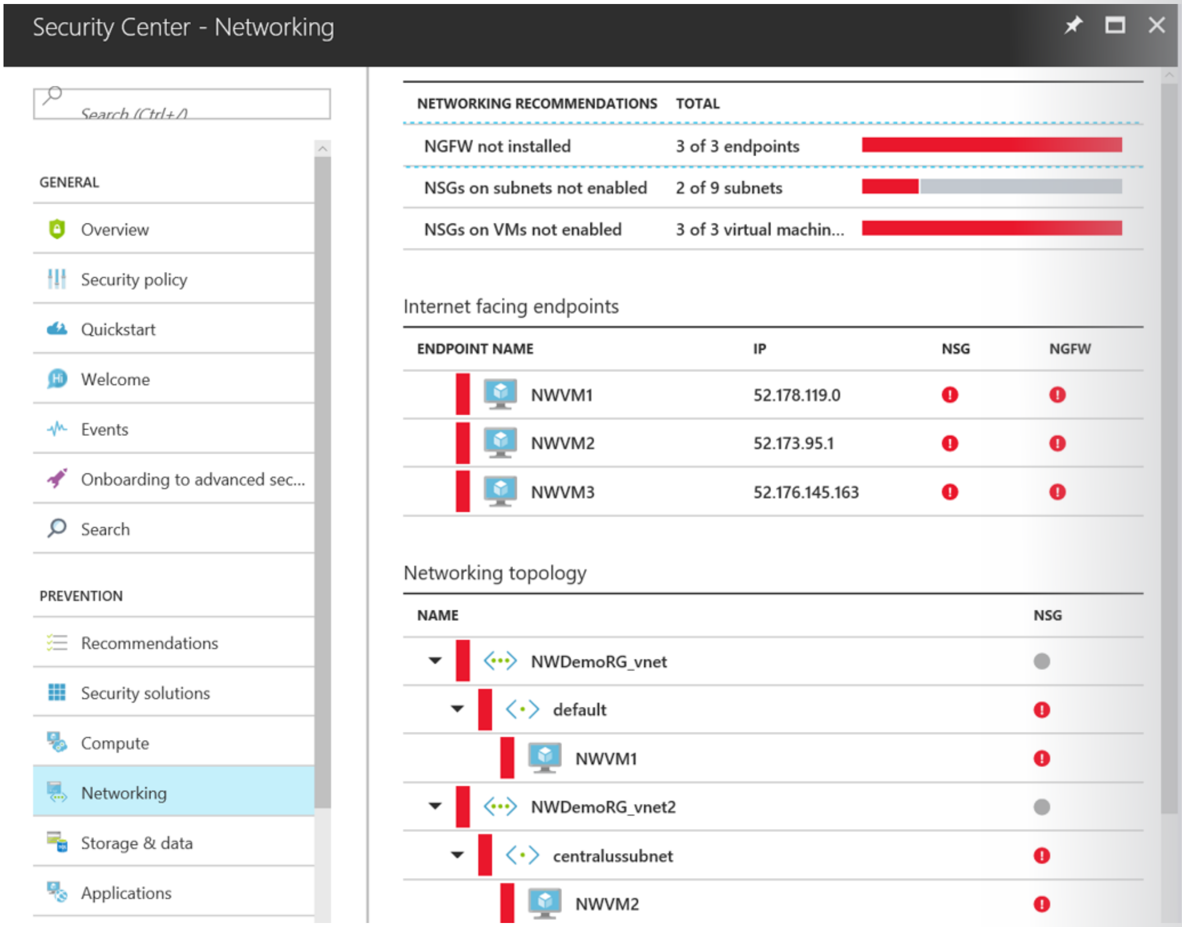
Periodic and spontaneous events are created by network resources and logged in storage accounts, sent to an Event Hub, or Log Analytics. These logs provide insights into the health of a resource. These logs can be viewed in tools such as Power BI and Log Analytics.

The troubleshooting blade, available in the portal, is provided on network resources today to diagnose common problems associated with an individual resource. This blade is available for the following network resources - ExpressRoute, VPN Gateway, Application Gateway, Network Security Logs, Routes, DNS, Load Balancer, and Traffic Manager.

Azure Security Center

Azure Security Center provides unified security management and advanced threat protection for workloads running in Azure, on-premises, and in other clouds. It delivers visibility and control over hybrid cloud workloads, active defenses that reduce your exposure to threats, and intelligent detection to help you keep pace with rapidly evolving cyberattacks.

The Security Center Overview provides a quick view into the security posture of your Azure and non-Azure workloads, enabling you to discover and assess the security of your workloads and to identify and mitigate risk.



NETWORKING RECOMMENDATIONS	TOTAL
NGFW not installed	3 of 3 endpoints
NSGs on subnets not enabled	2 of 9 subnets
NSGs on VMs not enabled	3 of 3 virtual machin...

Internet facing endpoints

ENDPOINT NAME	IP	NSG	NGFW
NWVM1	52.178.119.0	❗	❗
NWVM2	52.173.95.1	❗	❗
NWVM3	52.176.145.163	❗	❗

Networking topology

NAME	NSG
NWDemoRG_vnet	●
default	❗
NWVM1	❗
NWDemoRG_vnet2	●
centralussubnet	❗
NWVM2	❗

Azure Monitor & Diagnostics

Azure Monitor is part of Microsoft Azure's overall monitoring solution. Azure Monitor helps you track performance, maintain security, and identify trends. Learn how to audit, create alerts, and archive data with our quickstarts and tutorials.

Azure Monitor enables you to consume telemetry to gain visibility into the performance and health of your workloads on Azure. The most important type of Azure telemetry data is the metrics (also called performance counters) emitted by most Azure resources. Azure Monitor provides several ways to configure and consume these metrics for monitoring and troubleshooting.

What can you do with metrics?

Metrics are a valuable source of telemetry and enable you to do the following tasks:

- Track the performance of your resource (such as a VM, website, or logic app) by plotting its metrics on a portal chart and pinning that chart to a dashboard.
- Get notified of an issue that impacts the performance of your resource when a metric crosses a certain threshold.
- Configure automated actions, such as autoscaling a resource or firing a runbook when a metric crosses a certain threshold.
- Perform advanced analytics or reporting on performance or usage trends of your resource.

• Archive the performance or health history of your resource for compliance or auditing purposes.

Metrics

Azure Monitor enables users to obtain telemetry so that the user can gain visibility into the health and performance of workloads on Azure. Metrics is the essential type of Azure telemetry data that can be emitted by most Azure resources. Though Azure Monitor, a user has several ways to consume and configure these metrics for monitoring and troubleshooting.

Metrics have a set of characteristics you can use to identify it. Metrics become available immediately, meaning there is no need to set up additional diagnostics for metrics, nor opt-in for the data. Metrics also have a frequency of one minute. Users receive all metric values every minute from a resource, which allows for expanded visibility into the health and current state of your resource. Some metrics available can also have name-value pair attributes. These are known as dimensions which enable you to further segment and explore a metric. Moreover, Lastly, metrics allow users to access up to 30 days of history for each metric. By doing so, you can explore recent and monthly trends in the health and performance of your resource.

Users can use metrics to complete multiple tasks. These include tracking the performance of a resource by plotting its metrics on a chart. You can get notified of an issue if the issue impacts the performance of a resource. Since metrics have a frequency of one minute, this allows users to become aware of an issue on a near real-time basis. You can report on performance and usage trends on your resource to perform advanced analytics. After, users can choose to achieve health and performance history of a resource for compliance or auditing purposes.

Users can also choose to configure a metric alert rule that takes an automatic action, or even merely sends out a notification whenever the metric achieves a defined threshold. One of those automated actions is known as autoscale, which allows you to scale out your resource to meet incoming loads or requests. You can also route al metrics Log Analytics or Application Insights to enable instant Analytics. You can also choose to stream this information to an event hub. Doing so allows you to route them to Azure Stream Analytics for near-real-time analysis. You can choose to view all metrics, easily accessing them when you select a resource. You can also choose to achieve metrics to storage if you need to retrain them for longer than the archive period. You can choose o also route the metrics to an Azure Blob storage when you configure the settings for your resource.

Azure Advisors

Azure Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry. It then recommends solutions to help improve the performance, security, and high availability of your resources while looking for opportunities to reduce your overall Azure spend.

With Advisor, you can:

• Get proactive, actionable, and personalized best practices recommendations.

• Improve the performance, security, and high availability of your resources, as you identify opportunities to reduce your overall Azure spend.

• Get recommendations with proposed actions inline.

The Advisor dashboard displays personalized recommendations for all your subscriptions. You can apply filters to display recommendations for specific subscriptions and resource types. The recommendations are divided into four categories:

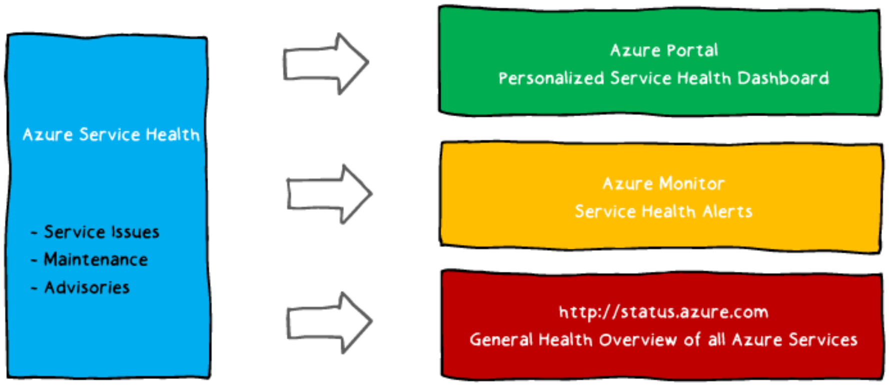
- **High Availability:** To ensure and improve the continuity of your business-critical applications.
- **Security:** To detect threats and vulnerabilities that might lead to security breaches.
- **Performance:** To improve the speed of your applications.
- **Cost:** To optimize and reduce your overall Azure spending.

Azure Service Health

Azure Service Health provides personalized guidance and support when issues in Azure services affect you, and helps you prepare for upcoming planned maintenance. Azure Service Health alerts you and your teams via targeted and flexible notifications.

Service Health tracks three types of health events that may impact your resources:

- Service issues - Problems in the Azure services that affect you right now.
- Planned maintenance - Upcoming maintenance that can affect the availability of your services in the future.
- Health advisories - Changes in Azure services that require your attention. Examples include when Azure features are deprecated or if you exceed a usage quota.



Operations Management Suite – Log Analytics

Log Analytics is part of Microsoft Azure's overall monitoring solution. Log Analytics monitors cloud and on-premises environments to maintain availability and performance. Get insight across workloads and systems to maintain availability and performance.

Log Analytics is a service in Operations Management Suite (OMS) that monitors your cloud and on-premises environments to maintain their availability and performance. It collects data generated by resources in your cloud and on-premises environments and from other monitoring tools to provide analysis across multiple sources. This article provides a brief discussion of the value that Log Analytics provides, an overview of how it operates, and links to more detailed content so you can dig further.

You can access Log Analytics through the OMS portal or the Azure portal which run in any browser and provide you with access to configuration settings and multiple tools to analyze and act on collected data. From the portal you can leverage log searches where you construct queries to analyze collected data, dashboards which you can customize with graphical views of your most valuable searches, and solutions which provide additional functionality and analysis tools.

Application Insights

Application Insights is an extensible Application Performance Management (APM) service for web developers building and managing apps on multiple platforms. Learn how to detect and diagnose issues and understand usage for your web apps and services.

Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live web application. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud. It integrates with your DevOps process, and has connection points to a variety of development tools. It can monitor and analyze telemetry from mobile apps by integrating with Visual Studio App Center and HockeyApp.

Power BI

With Azure services and Power BI, you can turn your data processing efforts into analytics and reports that provide real-time insights into your business. Whether your data processing is cloud-based or on-premises, straightforward or complex, single-sourced or massively scaled, warehoused or real-time, Azure and Power BI have the built-in connectivity and integration to bring your business intelligence efforts to life.

Power BI has a multitude of Azure connections available, and the business intelligence solutions you can create with those services are as unique as your business. You can connect as few as one Azure data source, or a handful, then shape and refine your data to build customized reports.

Azure SQL Database and Power BI

You can start with a straightforward connection to an Azure SQL Database, and create reports to monitor the progress of your business. Using the Power BI Desktop, you can create reports that identify trends and key performance indicators that move your business forward.
Do you have more complex data, and all sorts of sources? No problem. With Power BI Desktop and Azure services, connections are just a tap of the Get Data dialog away. Within the same Query you can connect to your Azure SQL Database, your Azure HDInsight data source, and your Azure Blob Storage (or Azure Table Storage), then select only the subsets within each that you need, and refine it from there.

There are all sorts of scenarios where Azure and Power BI can be combined - the possibilities and opportunities are as unique as your business. For more information about Azure services, check out this overview page, which describes Data Analytics Scenarios using Azure, and learn how to transform your data sources into intelligence that drives your business ahead.

Backup

Azure Backup

This lesson covers the many possible ways that Azure Backup can be integrated into an infrastructure solution hosted on Azure or in a hybrid-infrastructure scenario.

After completing this lesson, you will be able to:

- Detail the architectural components for deploying Azure Backup, using different topologies.
- Recognize the key use cases for implementing the different Azure Backup scenarios.
- Understand the built-in Security features that Azure Backup provides.
- Detail the architectural components for deploying Azure Site Recovery, using different topologies.
- Recognize the key use cases for implementing ASR, whether for DR or as a VM migration tool.

Azure Backup

Azure Backup is a simple and cost-effective backup as a service (BaaS) solution, that gives you trusted tools on-premises with rich and powerful tools in the cloud. It delivers strong protection for customer data wherever it resides—in your enterprise data center, remote and branch offices, or the public cloud—while being sensitive to the unique requirements these scenarios pose. Azure Backup, in a seamless portal experience with Azure Site Recovery, gives you cost-efficiency and minimal maintenance, consistent tools for offsite backups and operational recovery, and unified application availability and data protection.

Backup Options

There are three primary options for backing up to Azure Backup with different characteristics:

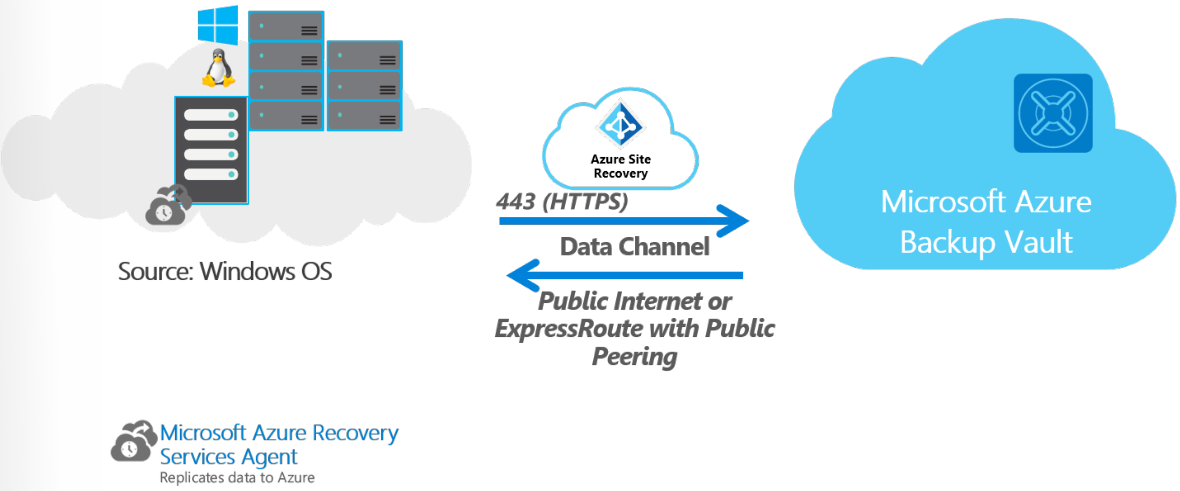
1. Azure Backup / Restore of On-Premises Files & Folders:
    - Ideal for backing up Files & Folders only
    - Deploy the Azure Backup Agent (Azure Recovery Services Agent) on the VM guests running on-premises Hyper-V
      / SCVMM / Vmware / Physical infrastructure.
    - Configure Azure Backup from within the VM guest.
    - Configure the integration with Azure Backup Vault.
    - Run the Backup job from within the VM guest.
    - Files & Folders backup will be stored in Azure Backup Vault, and can be restored from there.
2. Azure Backup / Restore of On-premises running full workloads (OS, Sysvol, and Applications):
    - Better for backing up full system workloads (OS, system state, applications – consistent)
    - Deploy the Azure Backup Server (or System Center DPM 2012 R2 or 2016) on the on-premises Hyper-V / SCVMM / Vmware / Physical infrastructure.
    - Configure Azure Backup Server backup policies, backup storage (2-tier) and deploy agents to your workloads.
    - Configure the integration with Azure Backup Vault.

- Run the Backup job from within the Azure Backup Server console.
- VM workloads (system state, OS, applications,…) backup will be stored in Azure Backup Vault, and can be restored from there.

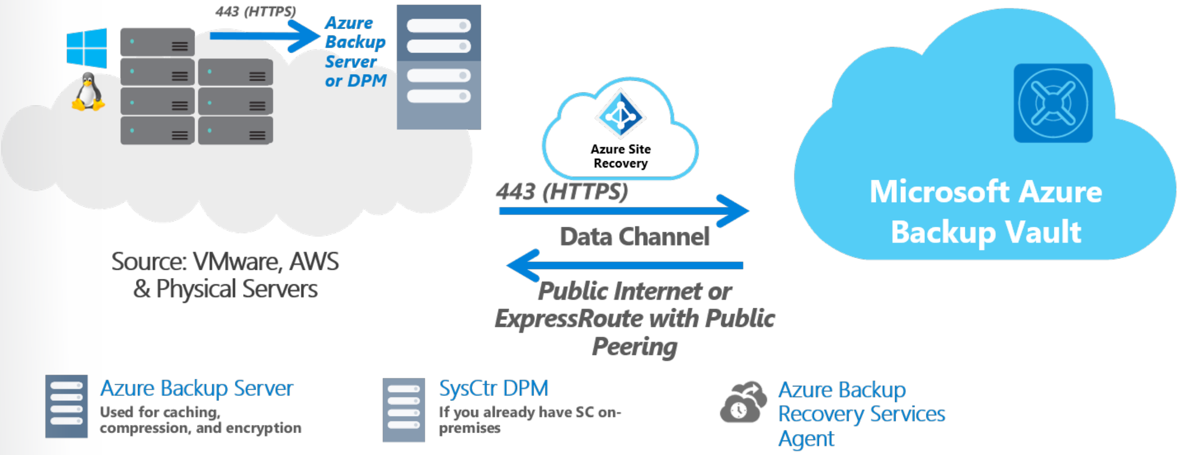
3. Azure VM Backup / Restore to Azure Backup Vault:

- Best for when you want to backup Azure VMs to Azure Backup Vault
- Deploy the Azure Backup Extension, or select Azure Backup in the VM configuration.
- Configure Azure Backup backup policies, in the Azure platform.
- Configure the integration with Azure Backup Vault.
- Run the Backup job from within the Azure Platform.
- Azure VMs will be backed up as full VM snapshots, and can be restored from within the Azure Portal.

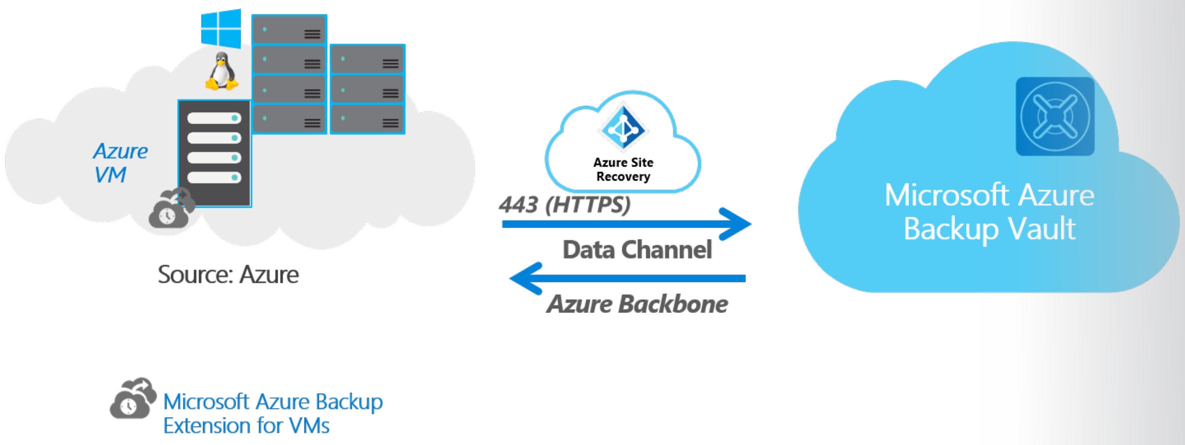
Azure Backup / Restore of On-Premises Files & Folders


Source: Windows OS
Azure Site Recovery
443 (HTTPS)
Data Channel
Public Internet or ExpressRoute with Public Peering
Microsoft Azure Backup Vault
Microsoft Azure Recovery Services Agent
Replicates data to Azure

Azure Backup and Restore of On-Premises Running Full Workloads (OS, Sysvol, and Applications)


443 (HTTPS)
Azure Backup Server or DPM
Source: VMware, AWS & Physical Servers
Azure Site Recovery
443 (HTTPS)
Data Channel
Public Internet or ExpressRoute with Public Peering
Microsoft Azure Backup Vault
Azure Backup Server
Used for caching, compression, and encryption
SysCtr DPM
If you already have SC on-premises
Azure Backup Recovery Services Agent

Azure VM Backup and Restore to Azure Backup Vault



Specialized Backup

Hybrid Backup Encryption

Security is an important aspect of any Public Cloud, and particularly in Microsoft Azure.

Azure Backup allows for end-to-end encryption of the backup platform:

1. It starts with a passphrase for the Azure Recovery Services Agent installation.

2. The next layer is the Backup Data itself, which gets encrypted in transit.

3. Once the data is stored in Azure Backup Vault, it gets encrypted at rest as well.

Azure Backup Monitoring with Log Analytics

- Azure Backup monitoring is possible from Log Analytics, part of Azure Operations Management Suite.
- Out of Log Analytics, one can get a detailed view on the backup statistics, the amount of data that is being consumed, successful and failed jobs and alike.

Azure Backup Reports with Power BI

- Besides Azure Monitoring capabilities in Operations Management Suite and Log Analytics, Azure Backup also allows for reporting integration with Microsoft Power BI.

Linux Application Consistent Azure Backup

- Taking backups of Azure VMs running Linux OS is fully supported, for Azure supported Linux Operating Systems.
- To allow for application consistent backups, you need to run a pre- and post- backup script. The VM Snapshot will be your VM Backup, which gets stored in the Backup Vault using an incremental update process.

Site Recovery

Azure Site Recovery is an Azure solution, initially built to provide a datacenter disaster recovery solution for your VM workloads. Whether they were running on-premises on Hyper-V hosts, VMware hosts, running as physical hosts, or as AWS VMs. Next to the core VM disaster recovery aspect of it, Azure Site Recovery is also an ideal tool for performing VM lift & Shift operations of your workloads.

Why Azure Site Recovery?

Azure as your Disaster/Recovery Datacenter Site:

- Replication-based failover to Azure Virtual Machines
- Near-zero downtime for your application workloads
- Application-consistent failover
- Failover & Failback
- DR for on-premises Hyper-V, VMware and physical Servers (*), as well as Azure VMs

Ideal as a Virtual Machine "Lift & Shift" migration tool:

- Full machine-state replication to an Azure VM
- Perfect for test/dev scenarios
- Zero-data loss during migration

Automation

Azure Automation

Azure Automation provides a way for users to automate the manual, long-running, error-prone, and frequently repeated tasks that are commonly performed in a cloud and enterprise environment. It saves time and increases the reliability of regular administrative tasks and even schedules them to be automatically performed at regular intervals. You can automate processes using runbooks or automate configuration management using Desired State Configuration. This lesson provides a brief overview of Azure Automation and answers some common questions.

After completing this section, you will be able to:

• Understand Azure Automation concepts and architecture.

• Recognize and describe Azure Automation capabilities.

• Interact with machines using Azure Automation and Desired State Configuration.

Azure Automation

Microsoft Azure Automation provides a way for users to automate the manual, long-running, error-prone, and frequently repeated tasks that are commonly performed in a cloud and enterprise environment. It saves time and increases the reliability of regular administrative tasks and even schedules them to be automatically performed at regular intervals. You can automate processes using runbooks or automate configuration management using Desired State Configuration.

Azure Automation is a software as a service (SaaS) application that provides a scalable and reliable, multi-tenant environment to automate processes with runbooks and manage configuration changes to Windows and Linux systems using Desired State Configuration (DSC) in Azure, other cloud services, or on-premises. Entities contained within your Automation account, such as runbooks, assets, Run As accounts are isolated from other Automation accounts within your subscription and other subscriptions.

Runbooks that you run in Azure are executed on Automation sandboxes, which are hosted in Azure platform as a service (PaaS) virtual machines. Automation sandboxes provide tenant isolation for all aspects of runbook execution – modules, storage, memory, network communication, job streams, etc. This role is managed by the service and is not accessible from your Azure or Azure Automation account for you to control.
To automate the deployment and management of resources in your local datacenter or other cloud services, after creating an Automation account, you can designate one or more machines to run the Hybrid Runbook Worker (HRW) role. Each HRW requires the Microsoft Management Agent with a connection to a Log Analytics workspace and an Automation account. Log Analytics is used to bootstrap the installation, maintain the Microsoft Management Agent, and monitor the functionality of the HRW. The delivery of runbooks and the instruction to run them are performed by Azure Automation.

Automation Flow

An Automation account is separate from the account you use to sign in to the portal to configure and use Azure resources. Automation resources included with an account are the following:

• **Certificates** - contains a certificate used for authentication from a runbook or DSC configuration or add them.

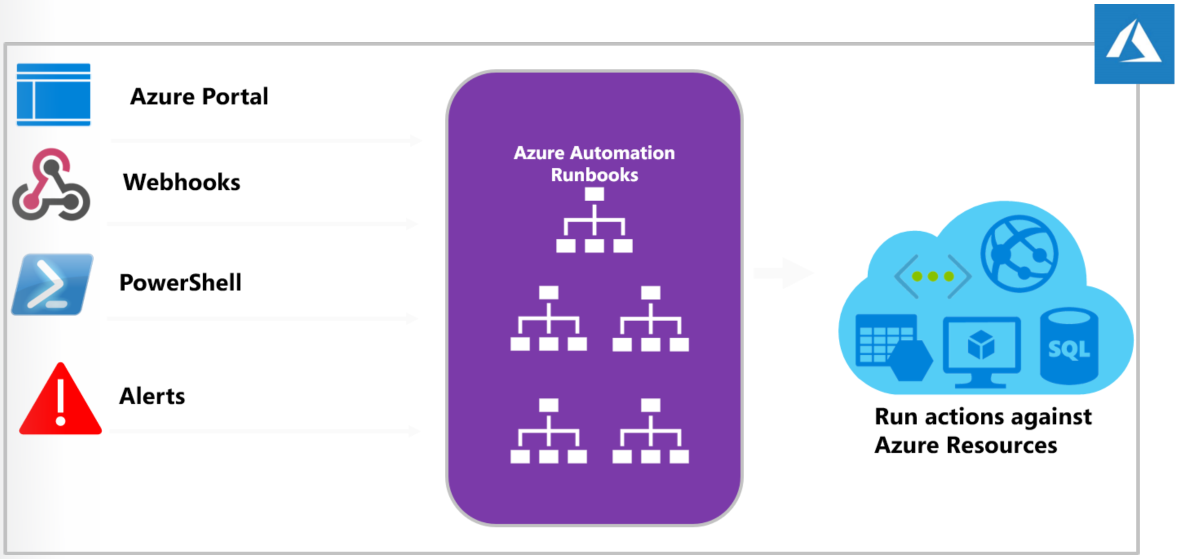
• **Connections** - contains authentication and configuration information required to connect to an external service or application from a runbook or DSC configuration.

- **Credentials** - is a PSCredential object which contains security credentials such as a username and password required to authenticate from a runbook or DSC configuration.
- **Integration modules** - are PowerShell modules included with an Azure Automation account to make use of cmdlets within runbooks and DSC configurations.
- **Schedules** - contains schedules that starts or stops a runbook at a specified time, including recurring frequencies.
- **Variables** - contain values that are available from a runbook or DSC configuration.
- **DSC Configurations** - are PowerShell scripts that describes how to configure an operating system feature or setting or install an application on a Windows or Linux computer.
- **Runbooks** - are a set of tasks that perform some automated process in Azure Automation based on Windows PowerShell.

When you create an Automation account in the Azure portal, you automatically create two authentication entities:

- **A Run As account**. This account creates a service principal in Azure Active Directory (Azure AD) and a certificate. It also assigns the Contributor role-based access control (RBAC), which manages Resource Manager resources by using runbooks.
- **A Classic Run As account**. This account uploads a management certificate, which is used to manage classic resources by using runbooks.

Role-based access control is available with Azure Resource Manager to grant permitted actions to an Azure AD user account and Run As account, and authenticate that service principal. Read Role-based access control in Azure Automation article for further information to help develop your model for managing Automation permissions.


Azure Portal
Webhooks
PowerShell
Alerts
Azure Automation Runbooks
SQL
Run actions against Azure Resources

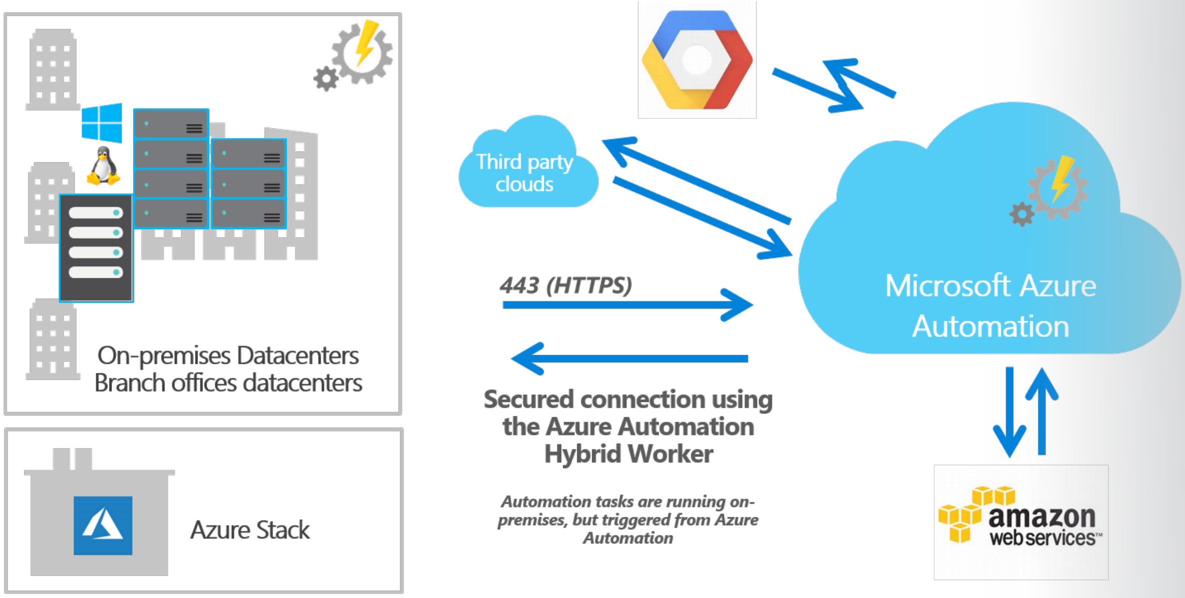
Cross-Cloud

Although Azure Automation lives in the Azure platform, it allows management and configuration of Azure systems, on-premises running systems, systems in AWS, Google Cloud, or any other 3rd party hosting data center.

The core component of Azure Automation is defined and configured in the Azure Platform (cross Regions). From there, you can establish hybrid automation capabilities, by using the Azure Automation Hybrid Worker.

This is similar to running like an Azure Automation agent on your non-Azure cloud platforms, where Amazon Web Services, Google Cloud, your own Private Cloud datacenter, or any third party hosted datacenter if you want, would be a good example.

Latest supportability added to the Azure Automation feature set, is integration with on-premises running Azure Stack.


On-premises Datacenters
Branch offices datacenters
Azure Stack
Third party clouds
443 (HTTPS)
Secured connection using the Azure Automation Hybrid Worker
Automation tasks are running on-premises, but triggered from Azure Automation
Microsoft Azure Automation
amazon web services

Configuration Management

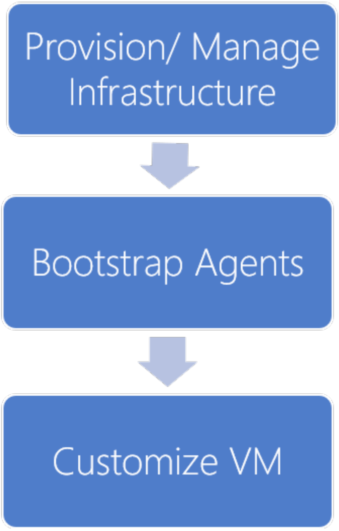
To create and manage Azure virtual machines (VMs) in a consistent manner at scale, some form of automation is typically desired. There are many tools and solutions that allow you to automate the complete Azure infrastructure deployment and management lifecycle. This article introduces some of the infrastructure automation tools that you can use in Azure.

Chef

Chef is an automation platform that helps define how your infrastructure is configured, deployed, and managed. Additional components included Chef Habitat for application lifecycle automation rather than the infrastructure, and Chef InSpec that helps automate compliance with security and policy requirements. Chef Clients are installed on target machines, with one or more central Chef Servers that store and manage the configurations.

Puppet

Puppet is an enterprise-ready automation platform that handles the application delivery and deployment process. Agents are installed on target machines to allow Puppet Master to run manifests that define the desired configuration of the Azure infrastructure and VMs. Puppet can integrate with other solutions such as Jenkins and GitHub for an improved devops workflow.



Provision/ Manage Infrastructure

Bootstrap Agents

Customize VM

CHEF

puppet

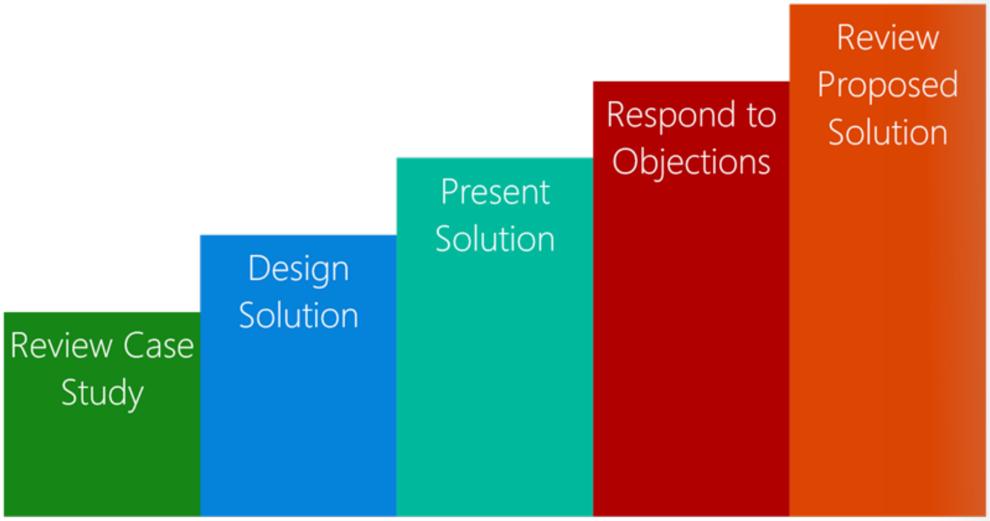
Design - Business Continuity

Who Is the Customer?

In this case study, we will look at a customer problem that requires an architectural recommendation.

After this case study, you should:

- Identify customer problems as they are related to networking.
- Design a solution that will meet the customer's objectives.
- Ensure your designed solution accounts for customer objections.



Who Is the Customer?

Fabrikam Publishing is a media and publishing company in Seattle, Washington, with approximately 5,000 employees.
What Does the Customer Already Have?

Fabrikam has a single data center that primarily runs Microsoft server software, including Active Directory Domain Services AD DS) and a number of AD-integrated services, including Exchange 2013, as well multi-tier, internal, AD-integrated IIS-based web applications with SQL Server 2014 as the database platform. The services are consumed by client systems hosted in three buildings located in adjacent areas of the city. Buildings are connected to the data center by using site-to-site VPN with the throughput of 100Mbps. Each building has also an independent connection to the internet.

Server backups are performed by using tape libraries with autoloaders. Tapes are periodically shipped for permanent storage to an offsite location.

If something catastrophic were to happen to the data center, the IT team would have to deploy replacement physical servers by reinstalling the operating system and restoring data from backups in the equipment rooms in small server rooms located in each building.

Fabrikam's IT staff likes to stay current with the latest offerings from Microsoft so that the department functions as cost-effectively as possible. In order to reduce costs, the IT staff has recently started planning the initiative to virtualize majority of its physical servers using the Hyper-V platform and to deploy System Center 2012 R2 Virtual Machine Manager (SCVMM) for managing the resulting virtualized environment.

What Is the Customer's Goal?

“We need to greatly improved disaster, server, and application recovery processes,” says Anthony Ciske, IT Director for Fabrikam. “We’ve had some near-disasters in the past that were a real pain to recover from. We needed a real disaster recovery solution for our critical workloads that was compatible with our budget—and our staffing bandwidth.” The team had explored building a secondary data center and employing commercial disaster recovery solutions in the past, but both turned out to be too expensive for serious consideration.

What Does the Customer Need?

The ability to perform data-center level recovery for critical workloads that can be executed in the event of a data center failure, with an automated and orderly recovery process so that different tiers of the application start in the correct order and remain in a consistent state without manual intervention.

- The ability to perform failback following restoring on-premises data center functionality that can be executed in the automated and orderly manner.
- The ability to perform multi-tier application and individual server-level recovery of critical workloads.
- Support for server-level and application-level high availability whenever possible.
- Quick testing and validation of recovery processes with minimal interruption to the production environment.
- Minimized capital and operational expenses.
- Optimized authentication for AD-integrated services and applications.
- Centralized management of backups and reduced or eliminated dependency on offsite tape storage.
- The level of security and privacy commensurate with highly sensitive and competitive nature of the business.

What Things Worry the Customer?

- Solution must significantly improve their current recovery point/time objectives (which today is a manual process).
- Overall cost of the solution.
- Protecting a diverse environment such as physical servers or other hypervisors.
- The management tools for the solution must be available in the event one of the data centers is unavailable.
- Protect data that is not hosted within a virtual hard disk (VHD/X).
- The protected data must be secure.
- Unsure about which workloads are supported on Azure.

Case Study Solution

Preferred Target Audience: Anthony Ciske, IT Director – Fabrikam Publishing

Network Administrator – Fabrikam Publishing

Application owners (Exchange, SQL, n-tier applications)

Preferred Solution

Fabrikam Publishing decided to complete their deployment of SCVMM and implement two Azure recovery solutions – Azure Site Recovery to provide failover capability of their virtual servers and Azure Backup to replace their existing backup solution and to provide longer term protection of their virtual and physical servers. Azure Site Recovery has been configured to use Microsoft Azure as a disaster recovery site, with protection enabled for virtual machines hosting servers critical from the business continuity standpoint.

For disaster recovery, Fabrikam implemented two separate Azure virtual networks. The first network was for planned/unplanned failover of their server workloads. The second virtual network was configured for testing failover in a non-disruptive manner. As part of the network configuration they implemented site-to-site connectivity between the failover virtual network and the on-premises sites that needed protecting.

Next, they deployed Active Directory in the failover virtual network and configured it to account for the local site to ensure localized authentication in recovery scenarios. In addition, DNS configuration on client systems as well as DNS settings on Azure virtual networks have been modified to ensure that name resolution continues to function during both test and planned/unplanned failover.

The final step for the disaster recovery solution was to address application level recovery on the application servers and high-availability concerns for Exchange 2013, SQL Server 2014, and other Active Directory-integrated applications.

Fabrikam implemented site recovery by configuring cloud recovery settings for the application servers.

Exchange 2013 is not currently supported within Azure IaaS. For an immediate cloud-based solution Exchange online is recommended.

For SQL Server deploying an AlwaysOn availability group on-premises with a replica in the failover network is the best solution. The replica should have async commit configured for replication.

Implementing separate recovery plans allows sequencing the failover of various tiers in a multi-tier application, hence, maintaining application consistency. What changes would you recommend to the existing SQL Server 2012 environment to facilitate high availability and recovery?

To address the data security concerns, encryption was enabled so any at rest data is automatically encrypted by Azure Site Recovery.

The second part of the preferred solution was to implement Azure Backup to remove the reliance on tape storage. Fabrikam created an Azure Backup Vault, downloaded and installed vault credentials on the servers that offload backups to tape and configured the Azure Backup Agent to protect the data in Azure instead of tape backup.

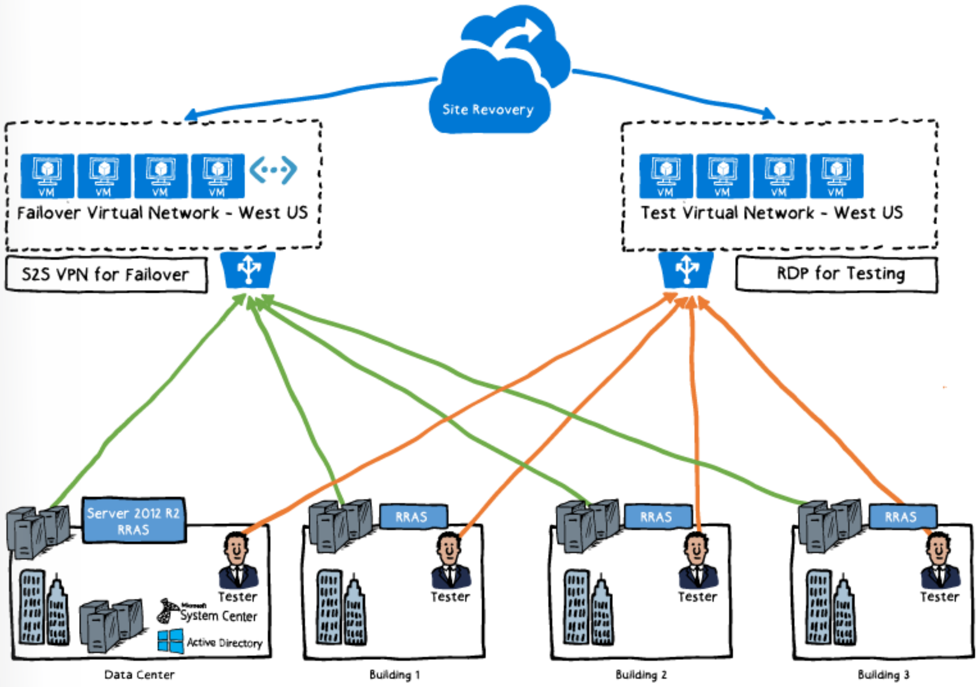
Potential Benefits

Fabrikam Publishing is using Microsoft Azure Site Recovery to implement their Disaster Recovery and Business Continuity strategy. By leveraging built-in features of Azure Site Recovery, they are able to accomplish their recovery and resiliency objectives efficiently and with a minimal cost, giving them a competitive advantage.

- **Recovery Objectives:** Azure Site Recovery with a recovery site in Azure offers the ability to perform planned and unplanned recovery and carry out testing with minimal disruption to the production environment. It allows for orchestrated, automated and orderly failover and failback processes, so different tiers of the application can start in the correct order and remain in a consistent state without manual intervention.
- **Cost Effective:** Azure Site Recovery eliminates capital expenses associated with implementing a secondary data center and provides predictable operational expenses. Azure Backup allows for centralized management of backups and eliminates the dependency on offsite tape storage.
- **Security and Privacy:** Data replicated to Azure is encrypted during transit and, if desired, it can be encrypted at rest, while residing in the storage account.
- **Non-Disruptive Testing of their DR Solution:** Site Recovery supports non-disruptive testing. Marquette can validate their disaster recovery solution as their environment changes without disrupting production.

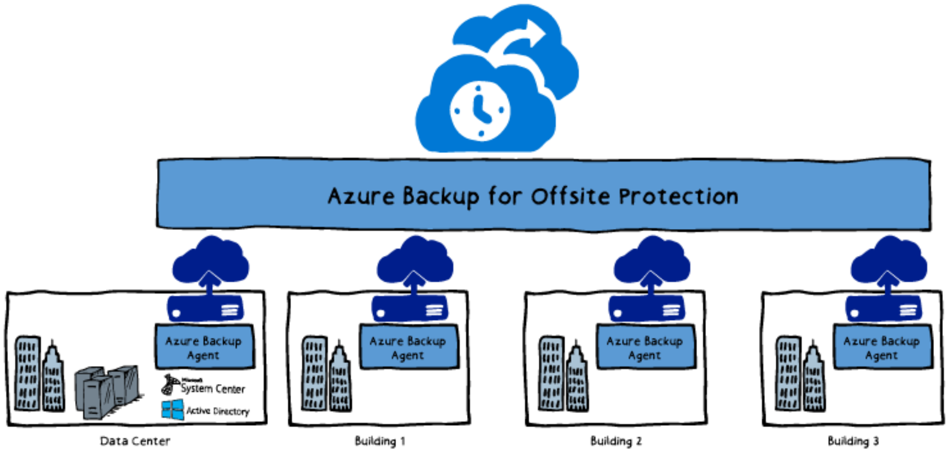
Architecture Example

Using Azure Site Recovery and multiple virtual networks for failover and non-disruptive failover testing:



ARCHITECTURE NETWORKING EXAMPLE

Using Azure Backup to protect servers' on-premises and store the data offsite in Azure:



ARCHITECTURE BACKUP EXAMPLE

Checklist of Potential Benefits

Solution must significantly improve their current recovery point/time objectives (which today is a manual process).

• With Azure Site Recovery the copy frequency can be configured to as low as 30 seconds. You can also configure additional recovery points to automatically be taken (the default is every hour). Additional recovery points contain one or more snapshots that enable you to recover a snapshot of a virtual machine from an earlier point in time.

Overall cost of the solution

• Protecting their servers by using Azure as a failover data center is significantly less expensive than implementing a failover solution with secondary data center and hardware.

Protecting a diverse environment such as physical servers or other hypervisors

• Currently, the Site Recovery service only allows replication to Azure for virtual machines based on Microsoft Hyper-V. Site Recovery can protect VMWare as well but only for on-premises to on-premises scenarios.

Protect data that is not hosted within a virtual hard disk (VHD/X)

• Azure Site Recovery works with Hyper-V based virtual machines and their attached virtual hard disks. In the scenario of restoring to Azure, the source virtual machines can be VHDX. In Azure they will be in the VHD format.

The management tools for the solution must be available if one of the data centers is unavailable.

• Azure Site Recovery is hosted in the cloud so in the event that one of the data centers is down the solution is still available to monitoring and to recover the servers.

Security of protected data

• Data replicated using Hyper-V Replica/Azure Site Recovery is encrypted while in transit.

• When replicating virtual machines to Azure replicated data can be encrypted even while at rest.

Supported Workloads

• Not all workloads are supported (or fully supported) with Hyper-V replica and Site Recovery.

• We are currently working with the various teams for key workloads such as SharePoint, SQL and Exchange to enable full support for these workloads in Hyper-V replica and site recovery.

Proof of Concept Checklist

Objectives:

- Identify service recovery point/time objectives for workloads in proof of concept.
- Identity manual steps needed for a recovery in secondary data center and script those steps as part of a recovery plan to ensure a fully automated recovery.
- Demonstrate that Azure Site Recovery and Hyper-V replication can replicate and recover virtual machines in the secondary data center within the RPO/RTO objectives.
- Enable backup protection to offsite storage.

Flow/Scope of the proof of concept (a list of 2-3 bullets):

Identify services (Virtual Machines) to configure for replication.
Implement two separate Azure virtual networks:

- One for planned/unplanned failover.
- The other for testing application and sever-level recovery.
- Configure network connectivity between the on-premises environment and the failover virtual network.

Create VMM cloud in the data center (if it does not exist).
Configure Active Directory and DNS.
Configure Azure Site Recovery with on-premises to Azure:

- Create an Azure Site Recovery Vault.
- Install the Provider application on the VMM server.
- If you don't have a storage account, create one.
- Install the Microsoft Azure Recovery Services agent on each Hyper-V host located in VMM clouds you want to protect.
- Configure cloud protection settings for VMM cloud.
- Configure network mapping to map source VM network to target Azure network.
- Enable protection for virtual machines located in protected VMM clouds.
- Failover using the test failover method.

Configure Azure Backup and protect on-premises server backups.

Conditions of satisfaction / success criteria for the proof of concept.

- Demonstrate that Azure Site Recovery does indeed fulfill the customer's recovery requirements of protecting the entire data center.

Resources / Bill of Materials that you would use:

- Online Documentation
- System Center 2012 R2 and Windows Server 2012 R2
- Azure Site Recovery and Azure Storage

- Virtual Private Networks
- Partner / MCS

Online Lab - Deploying Configuration Management solutions to Azure

Lab Steps

Online Lab: Deploying Configuration Management solutions to Azure

NOTE: For the most recent version of this online lab, see: **https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign**

Before we start

1. Ensure that you are logged in to your Windows 10 lab virtual machine using the following credentials:
   - Username: **Admin**
   - Password: **Pa55w.rd**
2. Review Taskbar located at the bottom of your Windows 10 desktop. The Taskbar contains the icons for the common applications you will use in the labs:
   - Microsoft Edge
   - File Explorer
   - **Visual Studio Code**[1]
   - **Microsoft Azure Storage Explorer**[2]
   - Bash on Ubuntu on Windows
   - Windows PowerShell
3. **Note**: You can also find shortcuts to these applications in the **Start Menu**.

[3]

Exercise 1: Deploy compute resources

[4]

Task 1: Open the Azure portal

1. On the Taskbar, click the **Microsoft Edge** icon.
2. In the open browser window, navigate to the **Azure Portal** (**https://portal.azure.com**).

[1] https://code.visualstudio.com/

[2] https://azure.microsoft.com/features/storage-explorer/

[3] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#exercise-1-deploy-compute-resources

[4] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-1-open-the-azure-portal

3. When prompted, authenticate with the user account account that has the owner role in the Azure subscription you will be using in this lab.

5

Task 2: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open a new shell instance.
2. **Note**: The **Cloud Shell** icon is a symbol that is constructed of the combination of the *greater than* and *underscore* characters.
3. If this is your first time opening the **Cloud Shell** using your subscription, you will see a wizard to configure **Cloud Shell** for first-time usage. When prompted, in the **Welcome to Azure Cloud Shell** pane, click **Bash (Linux)**.
4. **Note**: If you do not see the configuration options for **Cloud Shell**, this is most likely because you are using an existing subscription with this course's labs. If so, proceed directly to the next task.
5. In the **You have no storage mounted** pane, click **Show advanced settings**, perform the following tasks:
    - Leave the **Subscription** drop-down list entry set to its default value.
    - In the **Cloud Shell region** drop-down list, select the Azure region matching or near the location where you intend to deploy resources in this lab.
    - In the **Resource group** section, select the **Create New** option and then, in the text box, type **AADesignLab1201-RG**.
    - In the **Storage account** section, ensure that the **Create new** option is selected and then, in the text box below, type a unique name consisting of a combination of between 3 and 24 characters and digits.
    - In the **File share** section, ensure that the **Create new** option is selected and then, in the text box below, type **cloudshell**.
    - Click the **Create storage** button.
6. Wait for the **Cloud Shell** to finish its first-time setup procedures before you proceed to the next task.

6

Task 3: Deploy a Linux VM

1. At the top of the portal, click the **Cloud Shell** icon to open a new Clould Shell instance.
2. In the **Cloud Shell** pane, click the **Upload/Download files** icon and, in the drop-down menu, click **Upload**.
3. In the **Open** dialog box, navigate to the **\allfiles\AZ-301T02\Module_03\LabFiles\Starter\** folder, select the **linux-template.json** file, and click **Open**. The file contains the following template:

[5] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-2-open-cloud-shell

[6] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-3-deploy-a-linux-vm

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/
deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "userName": {
            "type": "string",
            "defaultValue": "Student"
        },
        "password": {
            "type": "securestring"
        }
    },
    "variables": {
        "vmName": "[concat('lvm', uniqueString(resourceGroup().id))]",
        "nicName": "[concat('nic', uniqueString(resourceGroup().id))]",
        "publicIPAddressName": "[concat('pip', uniqueString(resource-
Group().id))]",
        "virtualNetworkName": "[concat('vnt', uniqueString(resourceGroup().
id))]",
        "subnetName": "Linux",
        "imageReference": {
            "publisher": "suse",
            "offer": "opensuse-leap",
            "sku": "42.3",
            "version": "latest"
        }
    },
    "resources": [
        {
            "apiVersion": "2017-06-01",
            "type": "Microsoft.Network/publicIPAddresses",
            "name": "[variables('publicIPAddressName')]",
            "location": "[resourceGroup().location]",
            "properties": {
                "publicIPAllocationMethod": "Dynamic"
            }
        },
        {
            "apiVersion": "2017-06-01",
            "type": "Microsoft.Network/virtualNetworks",
            "name": "[variables('virtualNetworkName')]",
            "location": "[resourceGroup().location]",
            "properties": {
                "addressSpace": {
                    "addressPrefixes": [
                        "10.0.0.0/16"
                    ]
                },
                "subnets": [
                    {
                        "name": "[variables('subnetName')]",
                        "properties": {
                            "addressPrefix": "10.0.0.0/24"
                        }
                    }
                ]
            }
        },
        {
            "apiVersion": "2017-10-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('nicName')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[resourceId('Microsoft.Network/publicIPAddresses/', variables('publicIPAddressName'))]",
                "[resourceId('Microsoft.Network/virtualNetworks/', variables('virtualNetworkName'))]"
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Dynamic",
                            "publicIPAddress": {
                                "id": "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicIPAddressName'))]"
                            },
                            "subnet": {
                                "id": "[concat(resourceId('Microsoft.Network/virtualNetworks',variables('virtualNetworkName')), '/subnets/', variables('subnetName'))]"
                            }
                        }
                    }
                ]
            }
        },
        {
            "apiVersion": "2017-03-30",
            "type": "Microsoft.Compute/virtualMachines",
            "name": "[variables('vmName')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "Standard_A1_v2"
                },
                "osProfile": {
                    "computerName": "[variables('vmName')]",
                    "adminUsername": "[parameters('username')]",
                    "adminPassword": "[parameters('password')]"
                },
                "storageProfile": {
                    "imageReference": "[variables('imageReference')]",
                    "osDisk": {
                        "createOption": "FromImage"
                    }
                },
                "networkProfile": {
                    "networkInterfaces": [
                        {
                            "id": "[resourceId('Microsoft.Network/network-
Interfaces', variables('nicName'))]"
                        }
                    ]
                }
            }
        }
    ]
}
```

4. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the name of the resource group that will contain the hub virtual network:

   ```
   RESOURCE_GROUP='AADesignLab1202-RG'
   ```

5. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a variable which value designates the Azure region you will use for the deployment (replace the placeholder `<Azure region>` with the name of the Azure region to which you intend to deploy resources in this lab):

   ```
   LOCATION='<Azure region>'
   ```

6. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a new resource group:

   ```
   az group create --name $RESOURCE_GROUP --location $LOCATION
   ```

7. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to deploy the Azure Resource Manager template with the specified parameters file:

   ```
   az group deployment create --resource-group $RESOURCE_GROUP --template-file
   ~/linux-template.json --parameters password=Pa55w.rd1234
   ```

8. Do not wait for the deployment to complete before you proceed to the next task.

[7]

Task 4: Deploy an Azure Automation account

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Automation** and press **Enter**.
3. On the **Everything** blade, in the search results, click **Automation**.
4. On the **Automation** blade, click **Create**.
5. On the **Add Automation Account** blade, perform the following tasks:
    - In the **Name** text box, type **LinuxAutomation**.
    - Leave the **Subscription** drop-down list entry set to its default value.
    - In the **Resource group** section, select the **Create new** option and then, in the text box, type **AADesignLab1203-RG**.
    - In the **Location** drop-down list, select the Azure region matching or near the location where you deployed the Azure VM in the previous task.
    - In the **Create Azure Run As account** section, ensure that **Yes** option is selected.
    - Click the **Create** button.
6. Wait for the provisioning to complete before you proceed to the next task.

Review: In this exercise, you created a Linux VM using an Azure Resource Manager template and provisioned an Azure Automation account from the Azure portal.

[8]

Exercise 2: Configure Azure Automation DSC

[9]

Task 1: Import Linux PowerShell DSC modules

1. In the hub menu of the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab1203-RG**.
3. On the **AADesignLab1203-RG** blade, click the newly created Azure Automation account.
4. On the **LinuxAutomation** blade, in the **SHARED RESOURCES** section on the left side of the blade, click **Modules gallery**.

[7] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-4-deploy-an-azure-automation-account

[8] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#exercise-2-configure-azure-automation-dsc

[9] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-1-import-linux-powershell-dsc-modules

5. On the **LinuxAutomation - Modules gallery** blade, perform the following tasks:
   - In the **Search** text box, type **nx** and press **Enter**.
   - In the search results, click the **nx** module.
6. On the **nx** blade, click the **Import** button at the top of the blade.
7. On the **Import** blade, click the **OK** button.
8. Wait for the import process to finish before you proceed to the next task. A status message on the **nx Module** blade will indicate that the module was successfully imported.
9. **Note**: This process shoudl take about 2 minutes.

10

Task 2: Create Linux DSC Configuration

1. Navigate back to the **LinuxAutomation** blade.
2. Back on the **LinuxAutomation** blade, in the **CONFIGURATION MANAGEMENT** section, click **State configuration (DSC)**.
3. On the **LinuxAutomation - State configuration (DSC)** blade, click the **Configurations** link.
4. On the **LinuxAutomation - State configuration (DSC)** blade, click the **+ Add** button at the top of the pane.
5. On the **Import** blade, perform the following tasks:
   - Next to the **Configuration file** field, click the blue button with a folder icon.
   - In the **Choose File to Upload** dialog box, navigate to the **\allfiles\AZ-301T02\Module_02\LabFiles\Starter\** folder.
   - Select the **lampserver.ps1** file.
   - Click the **Open** button to close the dialog and return to the **Import** blade.
   - In the **Name** text box, accept the default entry **lampserver**.
   - In the **Description** text box, type **LAMP Server configuration using PHP and MySQL**.
   - Click the **OK** button.
6. Back in the **DSC configurations** pane, click **Refresh** and then click the newly created **lampserver** configuration.
7. On the **lampserver Configuration** blade, click the **Compile** button at the top of the blade. In the confirmation dialog box, click **Yes** to proceed with compiling the configuration.
8. Wait for the compilation task to finish. To determine the status of the compilation task, review the **STATUS** column of the **Compilation jobs** section of the **lampserver Configuration** blade.
9. **Note**: You may need to close and re-open the blade to see the latest compilation status. This blade does not refresh automatically.

[10] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-2-create-linux-dsc-configuration

[11]

Task 3: Onboard Linux VM

1. Navigate back to the **LinuxAutomation - State Configuration (DSC)** blade.
2. Back on the **LinuxAutomation - State Configuration (DSC)** blade, click the **Nodes** link.
3. On the **LinuxAutomation - State configuration (DSC)** blade, click the **+ Add** button at the top of the pane.
4. On the **Virtual Machines** blade, click the entry representing the Linux virtual machine you deployed in the previous exercise.
5. On the virtual machine blade, click **+ Connect**.
6. On the **Registration** blade, perform the following tasks:
   - Leave the **Registration key** setting with its default value.
   - In the **Node configuration name** drop-down list, select the **lampserver.localhost** entry.
   - Leave all remaining settings with their default values.
   - Click the **OK** button.
7. Wait for the connection process to complete before you proceed to the next step.
8. Navigate back to the **LinuxAutomation - State Configuration (DSC)** blade.
9. Back on the **LinuxAutomation - State Configuration (DSC)** blade, click the **Refresh** button.
10. In the list of DSC nodes, verify that the Linxu virtual machine has the **Compliant** status.

Review: In this exercise, you created a PowerShell DSC configuration and applied the configuration to a Linux virtual machine.

[12]

Exercise 3: Remove lab resources

[13]

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt at the bottom of the portal, type in the following command and press **Enter** to list all resource groups you created in this lab:

[11] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-3-onboard-linux-vm

[12] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#exercise-3-remove-lab-resources

[13] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-1-open-cloud-shell

```
az group list --query "[?starts_with(name,'AADesignLab12')]".name --output
tsv
```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

14

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name,'AADesignLab12')]".name --output
tsv | xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Review: In this exercise, you removed the resources used in this lab.

[14] https://github.com/MicrosoftLearning/AZ-301-MicrosoftAzureArchitectDesign/blob/master/Instructions/AZ-301T02_Lab_Mod03_Deploying%20Configuration%20Management%20solutions%20to%20Azure.md#task-2-delete-resource-groups

Review Questions

Module 3 Review Questions

Azure Network Watcher

You are designing a solution to monitor Azure-based networking resources.

You need to view the network topology, and relationships between the resources.

What Azure service should you recommend? What are the benefits and limitations of the service.

Suggested Answer ↓

Azure Network Watcher allows you to view your Azure network topology and the relationships between the resources. As an example, a virtual network contains subnets. Subnets contain resources, such as Azure Virtual Machines (VMs). VMs have one or more network interfaces. Each subnet can have a network security group and a route table associated to it. The topology capability of Azure Network Watcher enables you to view all of these resources in the virtual network and their relationships in a visual diagram.

Azure Application Insights

You are designing a solution that includes an Azure web app. The solution must detect and report performance anomalies in real time.

What Azure service should you recommend? What additional information can you collect?

Suggested Answer ↓

Azure Application Insights includes powerful analytics tools to help you diagnose issues and to understand what users do with your app. It is designed to help you continuously improve performance and usability.

Azure Application Insights works for apps on a wide variety of platforms including .NET, Node.js and J2EE. This includes apps that are hosted on-premises or in the cloud.

Azure Advisor

A company hires you to audit Azure resources. You must optimize resource usage and reduce costs. You must also examine security, availability, and performance of applications.

What Azure service should you use? What are the limitations of the service?

Suggested Answer ↓

Azure Advisor displays personalized recommendations for all your subscriptions. You can apply filters to display recommendations for specific subscriptions and resource types. The recommendations are across four categories that include high availability, security, performance and cost. Azure Advisor provides proactive, actionable, and personalized best practices recommendations to improve the performance, security and availability of your resources hosted on Azure. Azure Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments.