Module 6

Planning and implementing Azure Storage

Module Overview

- Planning storage
- Implementing and managing Azure Storage
- Exploring Azure hybrid storage solutions
- Implementing Azure CDNs

Lesson 1: Planning storage

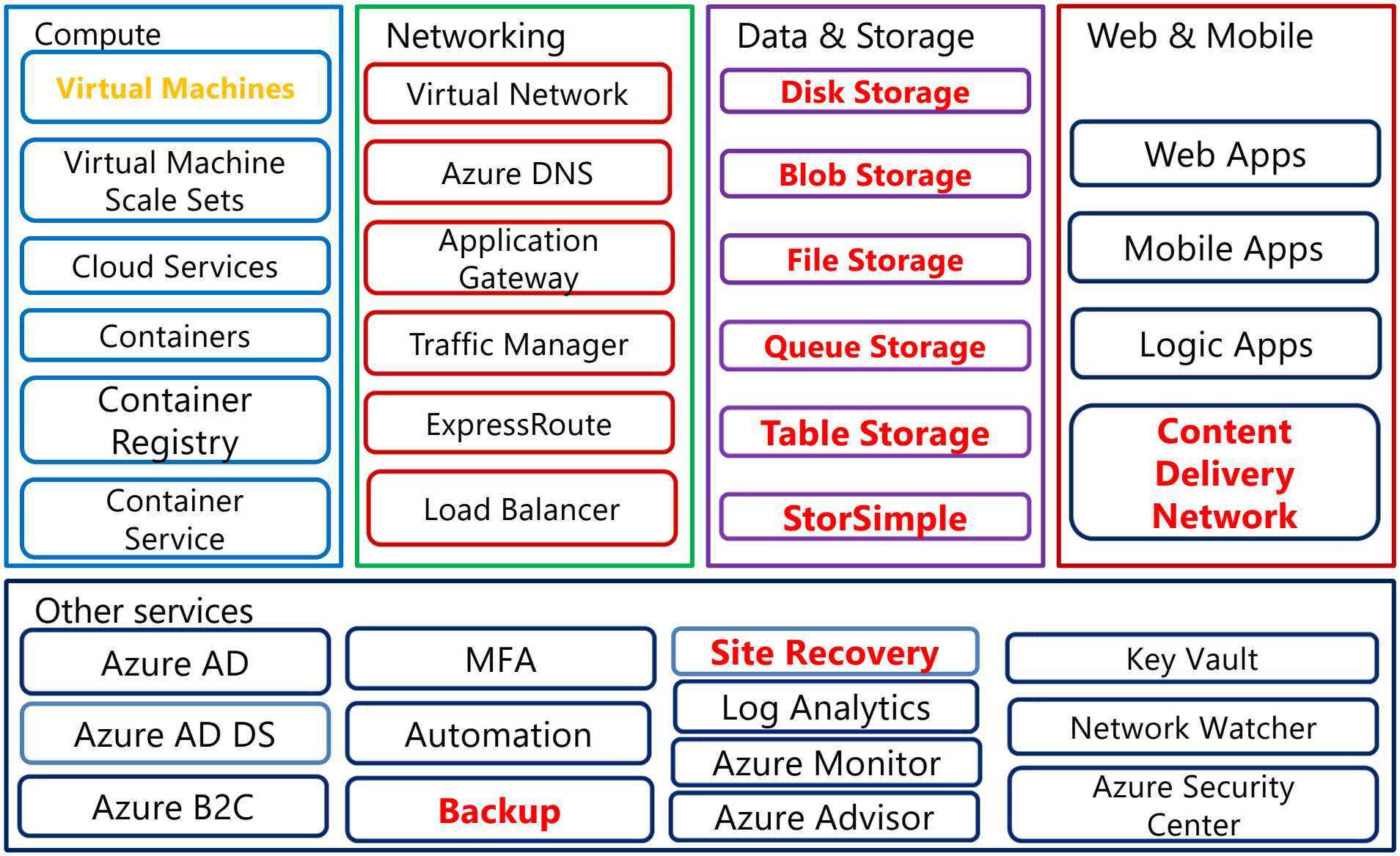
- Demonstration: Preparing the lab environment
- Role of Azure Storage in implementing Azure infrastructure solutions
- Overview of Azure Storage services
- Planning for Azure Storage standard services
- Planning for Azure Storage premium tier services
- Azure Storage pricing

Demonstration: Preparing the lab environment

In this demonstration, you will learn how to prepare the lab environment

Note: To prepare the lab environment for this module, you must complete this task

Role of Azure Storage in implementing Azure infrastructure solutions



Overview of Azure Storage services

Storage types:

- Blob storage - Containers for data blobs. The three types of blobs are:
  - Page blobs:
    - Optimized for random access
    - Azure virtual machine disk files
  - Block blobs:
    - Optimized for sequential access
    - Ideal for media and backups
  - Append blobs:
    - Optimized for append operations only
    - Ideal for logging
- Table storage - Store for non-relational key/value entities
- Queue storage - Temporary store for asynchronous exchange of messages
- File storage - File sharing store through SMB 3.x and SMB 2.1

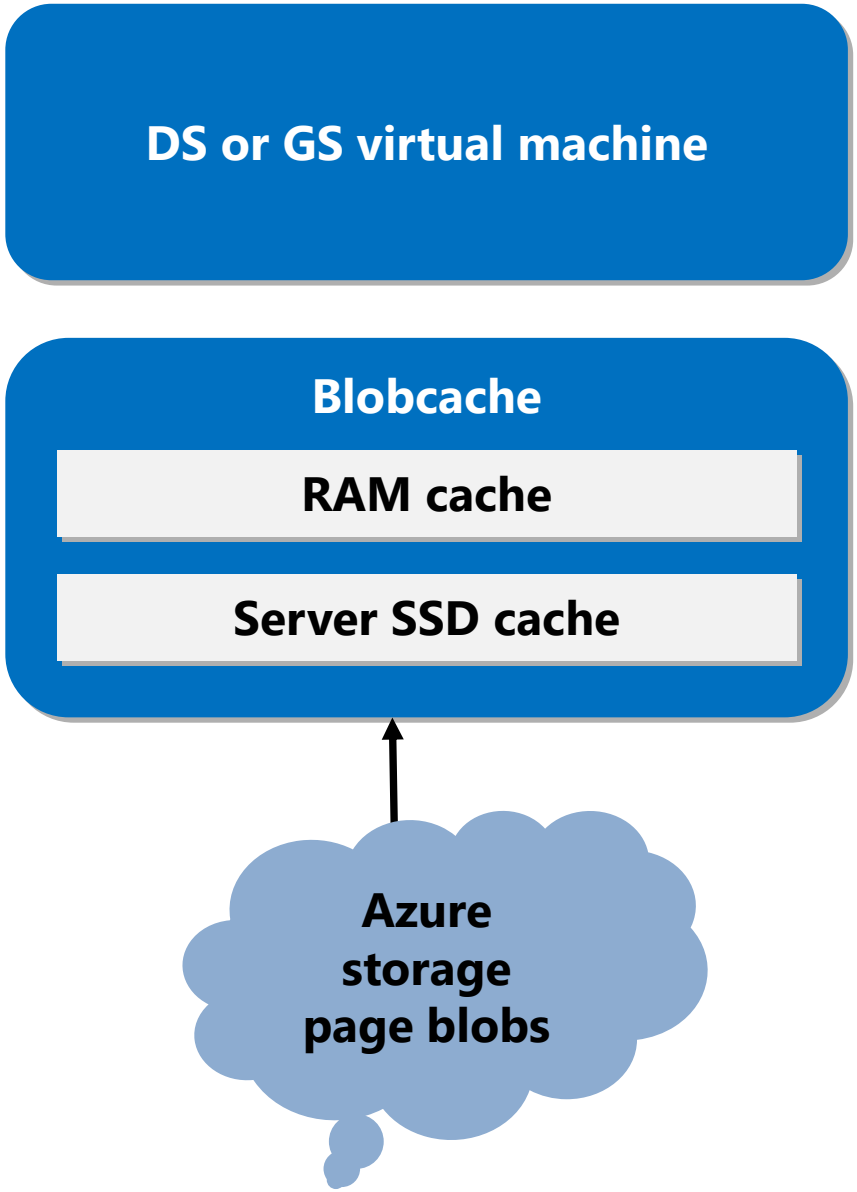
Overview of Azure Storage services

- General purpose v2 storage accounts:
  - Objects: Blobs (page, block, append), tables, queues, files
  - Performance: Standard or Premium (page blobs, LRS only)
  - Replication: LRS, ZRS (all objects except for VM disks), GRS, RA-GRS
  - Access tiers: Hot, cool, or archive
- General purpose v1 storage accounts:
  - Objects: Blobs (page, block, append), tables, queues, files
  - Performance: Standard or Premium (page blobs, LRS only)
  - Replication: LRS, ZRS classic (block blobs only), GRS, RA-GRS
- Blob storage accounts:
  - Objects: Block and append blobs only (optimized pricing)
  - Performance: Standard only
  - Replication: LRS, GRS, RA-GRS
  - Access tiers: Hot, cool, or archive

Planning for Azure Storage standard services

- Blob storage:
  - Block blobs - variable-sized blocks, optimized for large blobs
  - Page blob - 512-byte pages, optimized for random read/write
  - Append blob - specifically for append operations
- Table storage:
  - Stores data as key/value pairs in rows
  - Row entities or 252 custom properties (columns)
  - Single clustered index
- Queue storage:
  - Stores inter-application messages
- File storage:
  - SMB 2.1 and SMB 3.x file shares

Planning for Azure Storage premium tier services



Azure Storage pricing

- Storage account kind (general purpose v1, v2, or blob)
- Performance tier (standard or premium)
- Access tier (blob and general purpose v2 storage accounts)
- Replication type (LRS, ZRS, GRS, RA-GRS)
- Volume of storage transactions (Premium excluded)
- Volume of egress traffic
- Storage space:
  - In use (standard performance level, excluding managed disks)
  - Provisioned (premium performance tier)
- Volume of data reads and writes (cool and archive access tier)
- Type of storage objects (for general purpose v1 and v2 storage accounts)
- Early deletion for cool and archive blobs (general purpose v2 storage accounts only)

Lesson 2: Implementing and managing Azure Storage

- Azure Storage tools
- Creating an Azure Storage account
- Managing Azure blob storage
- Managing Azure file storage
- Managing Azure table and queue storage
- Managing access to Azure Storage
- Monitoring Azure Storage with Azure Storage Analytics
- Demonstration: Using Azure Storage

Azure Storage tools

- REST APIs and client libraries
- Azure PowerShell
  - Windows, Linux, Mac OS
- Azure CLI
  - Windows, Linux, Mac OS
- AzCopy
  - Windows and Linux
- Microsoft Azure Storage Explorer
  - Windows, Linux, Mac OS
- Visual Studio
  - Windows
  - Server Explorer and Cloud Explorer (Azure SDK 2.7+)

Creating an Azure Storage account

- Azure portal
  - Name: unique within the core.windows.net domain
  - Deployment model: Resource manager or Classic
  - Account kind: Storage (general purpose v1), StorageV2 (general purpose v2) or Blob storage
  - Performance: Premium or Standard
  - Replication: Locally-redundant storage (LRS), Geo-redundant storage (GRS), Read-access geo-redundant storage (RA-GRS), or Zone-redundant storage (ZRS)
  - Require secure transfer: Disabled or Enabled
  - Resource group: new or existing
  - Location: an Azure region
  - Virtual networks: exclusive network access from designated virtual network subnets
- Azure PowerShell
  - New-AzureRmStorageAccount
- Azure CLI 2.0
  - az storage account create

Managing Azure blob storage

- Create a container in a storage account
- Specify the container access level:
  - Private
  - Public Blob
  - Public Container
- Manage blobs by using:
  - AzCopy
  - Azure Storage Explorer
  - Azure PowerShell
  - Azure CLI
  - Azure portal

Managing Azure file storage

- Creating file shares:
  - Azure portal, Windows PowerShell, Azure CLI, or REST API
- Accessing file shares:
  - On Windows, create a drive mapping with the **net use**
  - On Linux, create a mountpoint and run **mount –t cifs**
  - Provide an account name and a key
  - Run from the same region (SMB 2.1 or SMB 3.x)
  - Run from another Azure region or from any on-premises location (SMB 3.x on Windows)
  - Alternatively, use Windows PowerShell, Azure CLI, AzCopy, or the REST API from any location

Managing Azure table and queue storage

- Create and update programmatically
- Manage by using:
  - Azure Storage Explorer
  - Azure portal
  - Azure PowerShell and Azure CLI:
    - Tables:

```
New-AzureStorageTable -Name $tabName -Context $context
Get-AzureStorageTable -Name $tabName -Context $context
Remove-AzureStorageTable -Name $tabName -Context $context
```

    - Queues:

```
New-AzureStorageQueue -Name $qName -Context $context
Get-AzureStorageQueue -Name $qName -Context $context
Remove-AzureStorageQueue -Name $qName -Context $context
```

Managing access to Azure Storage

- Storage account access keys:
  - Provide full access to a storage account
  - Primary and secondary, automatically generated but can be recycled
- SAS:
  - Granular (container or object-level)
  - Limited to a time window and an IP address range
  - Cannot be easily revoked (require regenerating the signing key)
- Stored access policy:
  - Granular (container level)
  - Limited to a time window
  - Can be easily revoked
- Azure firewall and virtual network access
- Role-based access control:
  - Default roles
  - Custom roles

Monitoring Azure Storage with Azure Storage Analytics

- You can enable monitoring for a new or existing standard storage account:
  - Aggregate metrics
  - Per-API metrics
  - Logs
- Metrics and logs are stored in the monitored storage account
- Metrics can be displayed in the **Monitoring** section of the storage account blade in the Azure portal
- Metric-based alerts can:
  - Be delivered through email
  - Be routed to a Webhook
  - Trigger an Azure logic app

Demonstration: Using Azure Storage

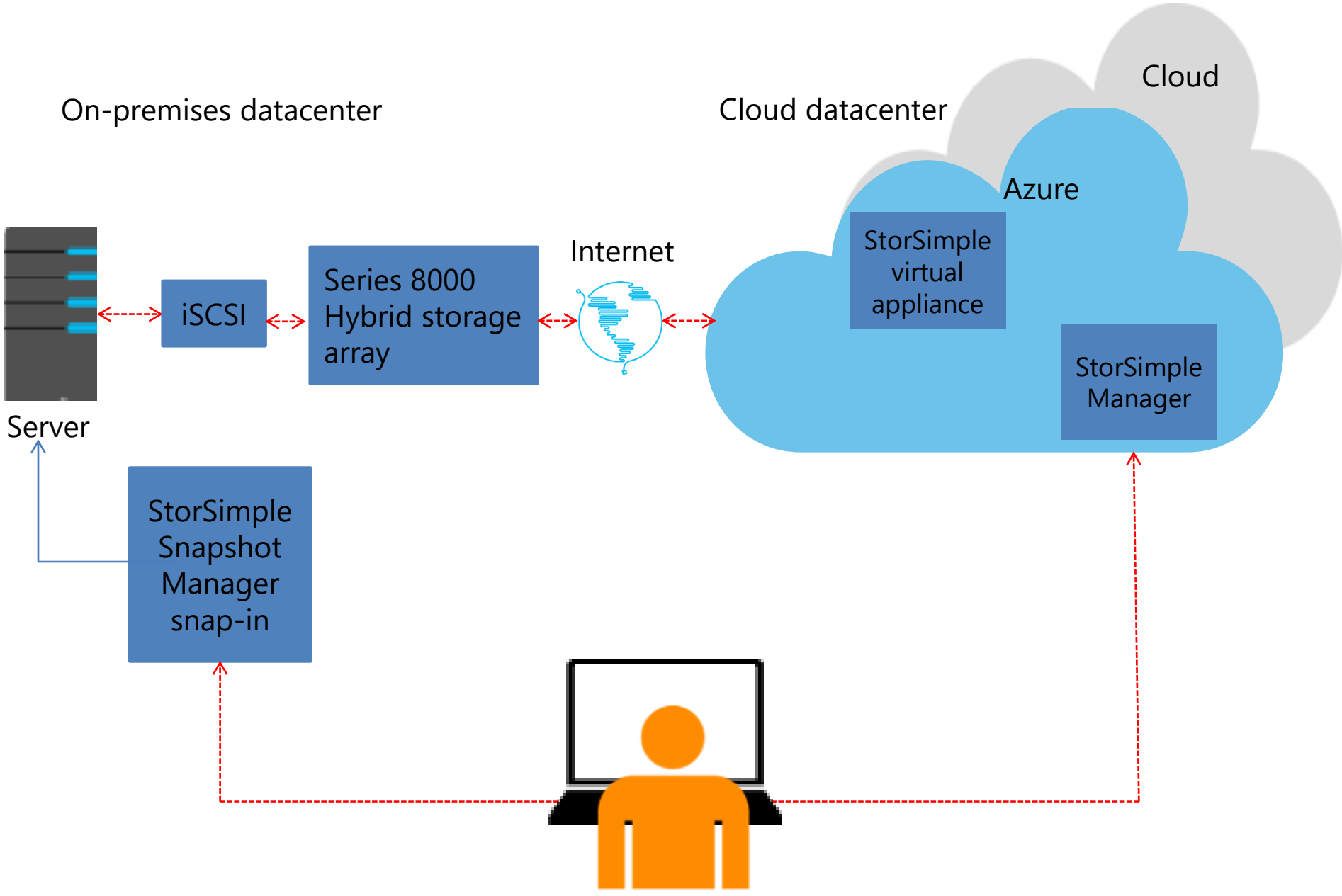
In this demonstration, you will see how to:

- Create an Azure storage account
- Create an Azure Files share
- Mount an Azure file share on an Azure VM

Lesson 3: Exploring Azure hybrid storage solutions

- Hybrid storage capabilities of StorSimple
- Cross-premises data transfer with the Azure Import/Export service and Azure Data Box
- Hybrid file services with Azure File Sync

Hybrid storage capabilities of StorSimple



Cross-premises data transfer with the Azure Import/Export service and Azure Data Box

- Import/Export Service:
  - Transfers data from and to Azure Storage via physical disks (SATA II/III HDDs and SSDs):
    - Exporting block, page, and append blobs from Azure blob and general purpose v1 storage accounts
    - Importing data into block, page, and append blobs in Azure blob and general purpose v1 storage accounts
    - Importing data into Azure files in Azure general purpose v1 storage accounts
- Azure Data Box:
  - Transfers data to Azure via a physical appliance (100-TB capacity)

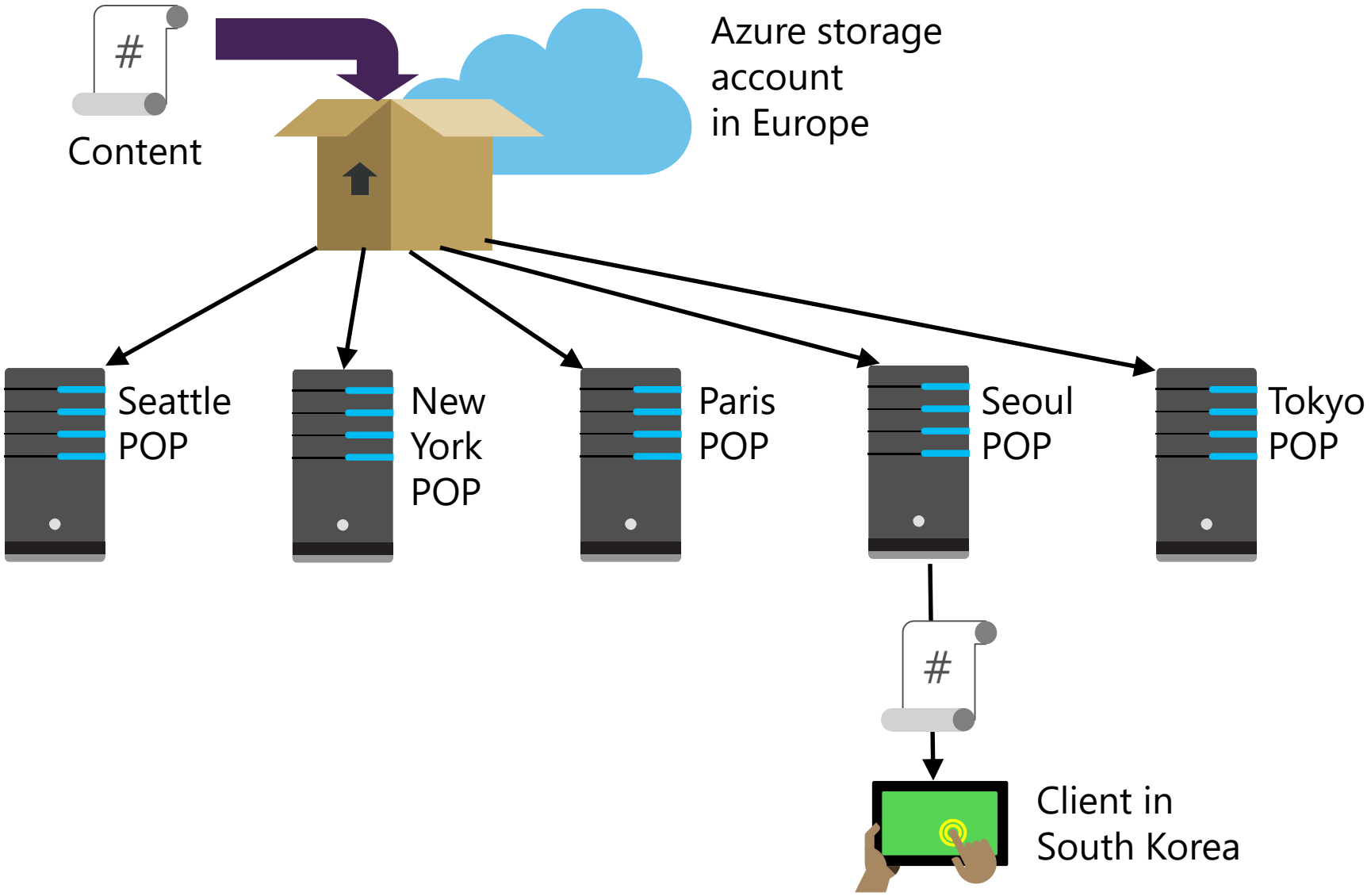
Hybrid file services with Azure File Sync

- Azure File Sync benefits:
  - Synchronization of Windows Server file shares
  - Windows Server file share data tiering
  - Centralized Windows Server file share backup
- Azure File Sync architecture:
  - Azure Storage Sync Service
  - Sync group consists of a cloud endpoint and server endpoint (agent based)
- Azure File Sync implementation:
  1. Create an Azure Storage account and an Azure File share
  2. Create an Azure Storage Sync Service instance
  3. Create a sync group
  4. Add cloud endpoint to the sync group
  5. Install Azure File Sync agent on a Windows file server
  6. Register the agent and add the server endpoints to the sync group
  7. Wait for the synchronization to complete
  8. Add other server endpoints

Lesson 4: Implementing Azure CDNs

- Introduction to CDNs
- Overview of CDN architecture
- Using CDN to cache content from Azure blobs, Azure Web Apps, and Azure Cloud Services
- Using custom domains to provide access to CDNs

Introduction to CDNs



Overview of CDN architecture

- CDN pricing tiers:
  - Azure CDN Standard from Akamai
  - Azure CDN Standard from Verizon
  - Azure CDN Premium from Verizon
- CDN profile:
  - An administrative and billing unit
  - Up to 25 per subscription
- CDN endpoints:
  - Default limit of 10 endpoints per profile (25 maximum limit)
    - Azure Storage blob
    - Azure Web app (Standard or Premium App Service plan)
    - Azure Cloud Service
    - Azure Media Services streaming endpoint.
    - Custom origin - any location accessible via HTTP or HTTPS

Using CDN to cache content from Azure blobs, Azure Web Apps, and Azure Cloud Services

- Storage CDN endpoint:
  - References the origin hostname (storage account)
  - Can reference a container (origin path)
  - The container must allow anonymous access
- Caching TTL:
  - Default of 7 days
  - To customize for storage blobs:
    - Modify cache-control on the blob level:
    - Use Azure PowerShell, Azure CLI, Storage Client Libraries, REST API
  - To customize for Web apps and cloud services:
    - Modify applicationHost.config (site level)
    - Modify web.config (web app level)
    - Manage HttpResponse.Cache programmatically

Using custom domains to provide access to CDNs

- You can map the fully qualified domain name of a CDN endpoint to a custom subdomain
- Create an alias (CNAME) record that automatically points all traffic to the corresponding CDN endpoint
- Use asverify to avoid downtime when assigning CNAME to an existing CDN-based implementation

Lab: Planning and implementing Azure Storage

- Exercise 1: Creating and configuring Azure Storage
- Exercise 2: Using Azure File storage

Logon Information

Virtual machine:	**20533E-MIA-CL1**
User name:	**Student**
Password:	**Pa55w.rd**

Estimated Time: 50 minutes

Lab Scenario

The IT department at Adatum Corporation uses an asset management application to track IT assets such as computer hardware and peripherals. The application stores images of asset types and invoices for purchases of specific assets. As part of Adatum's evaluation of Azure, you need to test Azure storage features as part of your plan to migrate the storage of these images and invoice documents to Azure. Adatum also wants to evaluate Azure File storage for providing SMB 3.0 shared access to installation media for the asset management application client software. Currently, corporate file servers host the media.

Lab Review

- The asset management application stores images of hardware components as blobs and invoices as files. If the application also needed to search the location of each asset by using an asset type, a unique asset number, and a text description of the location, what storage options should you consider?

Module Review and Takeaways

- Review Question
- Best Practices