Module 6

Planning and implementing storage, backup, and recovery services

Module Overview

- Planning storage
- Implementing and managing Azure Storage
- Implementing Azure Content Delivery Networks
- Implementing Azure Backup
- Planning and implementing Azure Site Recovery

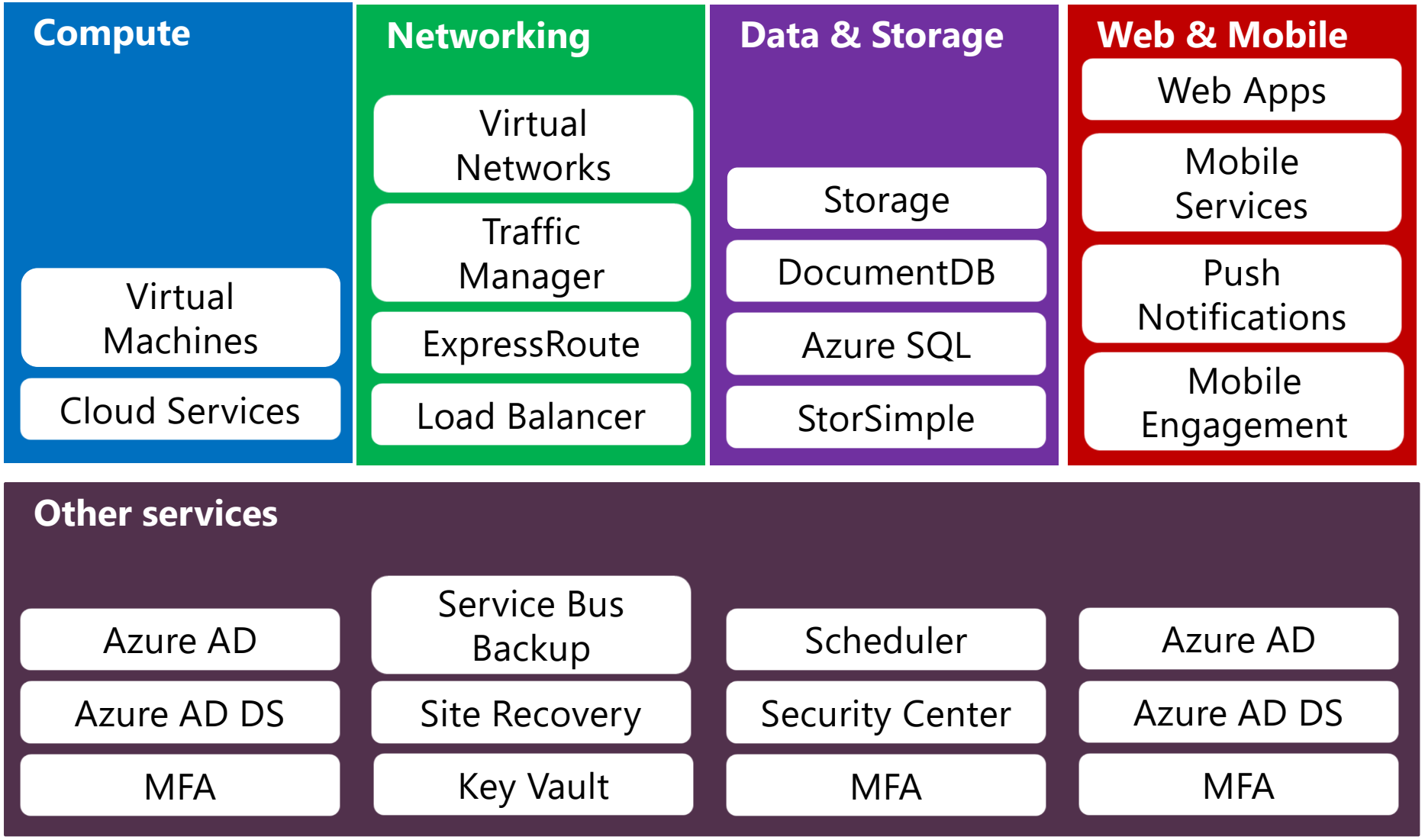
Lesson 1: Planning storage

- Demonstration: Preparing the environment for the lab and demos in this module
- Storage as an Azure component
- Overview of Azure Storage
- Planning for standard Azure Storage
- Planning for Azure Premium Storage

Demonstration: Preparing the environment for the lab and demos in this module

In this demonstration, you will see how to prepare the environment

Storage as an Azure component



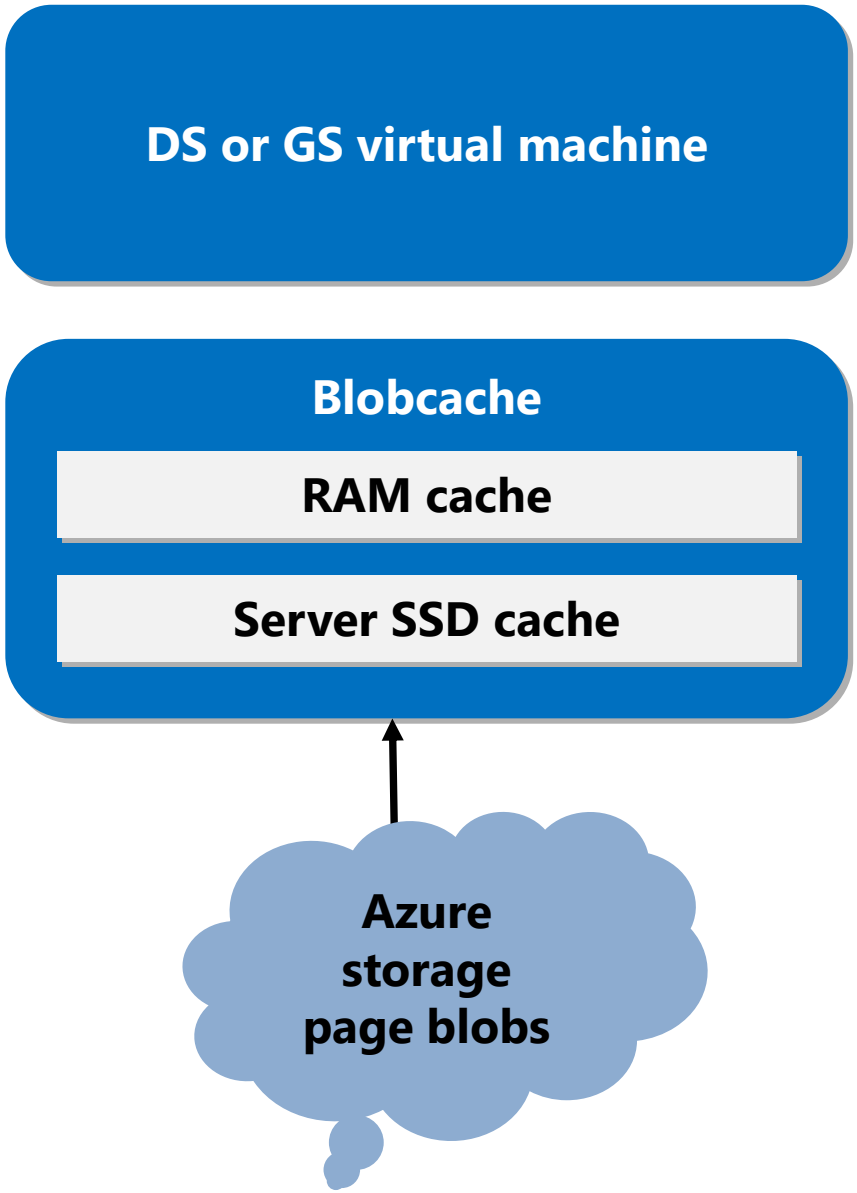
Overview of Azure Storage

- 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

Planning for standard Azure Storage

- 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 3.x file shares

Planning for Azure Premium Storage



Lesson 2: Implementing and managing Azure Storage

- Storage access tools
- Creating a storage account
- Implementing blobs
- Implementing Azure file storage
- Implementing Azure table and queue storage
- Controlling access to storage
- Monitoring storage
- Demonstration: Implementing storage

Storage access tools

- REST APIs and Client Libraries
- Azure PowerShell
- AzCopy
- Azure Storage Explorer (CodePlex)
- Server Explorer (Visual Studio 2015)

Creating a storage account

- General purpose storage accounts:
  - Blobs (page, block, append), tables, queues, files
  - Performance
    - Standard or Premium (page blobs, LRS only)
  - Replication
    - LRS, ZRS (block blobs only), GRS, RA-GRS
- Blob storage accounts:
  - Block and append blobs only (optimized pricing)
  - Access tiers (depending on frequency of data access):
    - Hot or cool
    - Switching between tiers is supported (consider cost implications)
  - Replication:
    - LRS, GRS, RA-GRS

Implementing blobs

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

Implementing Azure file storage

- Creating file shares:
  - Azure portal, Windows PowerShell, or REST API
- Accessing file shares:
  - Map a drive with the **net use** command:
    - 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 required)
  - Alternatively, use Windows PowerShell, AzCopy, or the REST API

Implementing Azure table and queue storage

- Create and update programmatically by applications
- Manage by using:
  - Azure Storage Explorer
  - Azure PowerShell:
    - 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
```

Controlling access to storage

- Storage account access keys:
  - Primary and secondary
  - Automatically generated but can be recycled
  - Provide full access to a storage account
- SAS:
  - Granular (container or resource level)
  - Time-limited
- Stored access policy:
  - Granular (container level)
  - Time-limited
  - You can easily revoke policy-linked SAS
- Role-based access control:
  - Default roles
  - Custom roles

Monitoring storage

- You can enable monitoring for a new or existing standard storage account:
  - Aggregate metrics
  - Per-API metrics
  - Logs
- Not supported for Azure Premium storage accounts
- Metrics and logs are stored in the same storage account
- Metrics can be displayed in the **Monitoring** lens
- Metric-based alerts:
  - Delivered through email
  - Routed to a Webhook

Demonstration: Implementing storage

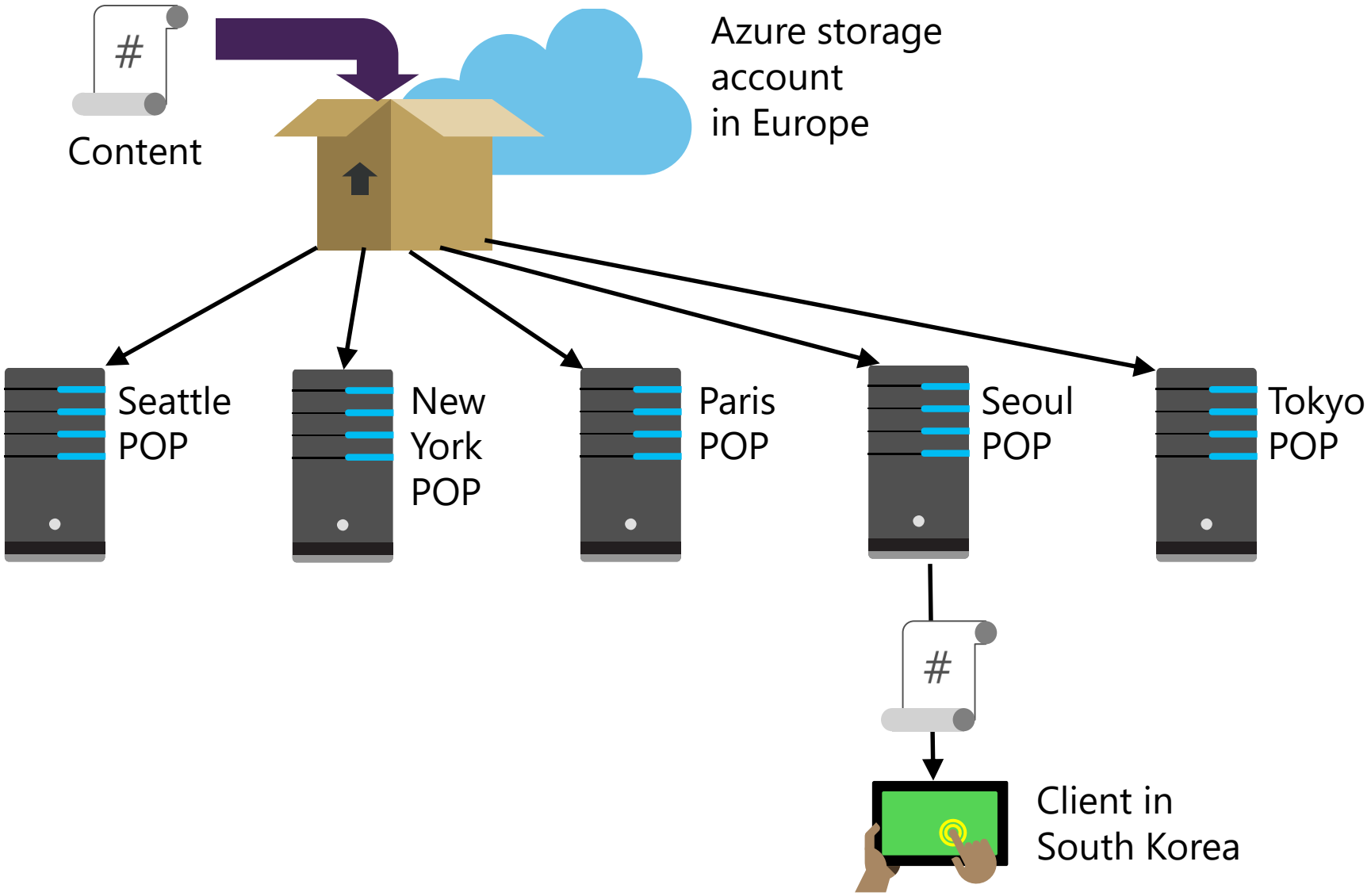
In this demonstration, you will see how to:

- Create a storage account
- Use Windows PowerShell to upload blobs
- View blob storage in Visual Studio
- Configure monitoring and logging
- View logged events

Lesson 3: Implementing Azure Content Delivery Networks

- Overview of CDNs
- CDN architecture
- Caching content from Azure blobs
- Caching content from cloud services and web apps
- Using custom domains to access CDNs

Overview of CDNs



CDN architecture

- CDN endpoints are globally distributed
- Data from Azure Storage is cached at each CDN endpoint
- Users access data from their closest CDN endpoint
- If data is not available at a CDN endpoint, Azure retrieves it from the origin and caches it at the CDN endpoint

Caching content from Azure blobs

- Azure can only cache publicly available blobs in CDN endpoint
- After a CDN is implemented, all publically available blobs in the container will be cached
- Cached content remains in the cache for the duration of TTL, which is 7 days by default

Caching content from cloud services and web apps

- You can cache PaaS cloud service or Azure Web App content in a CDN
- The content to cache should be static and must be accessible via HTTP on port 80
- The cloud service must be in the production deployment slot
- You can configure TTL:
  - At the site level (applicationHost.config)
  - At the web app level (web.config)
  - Programmatically (HttpResponse.Cache)

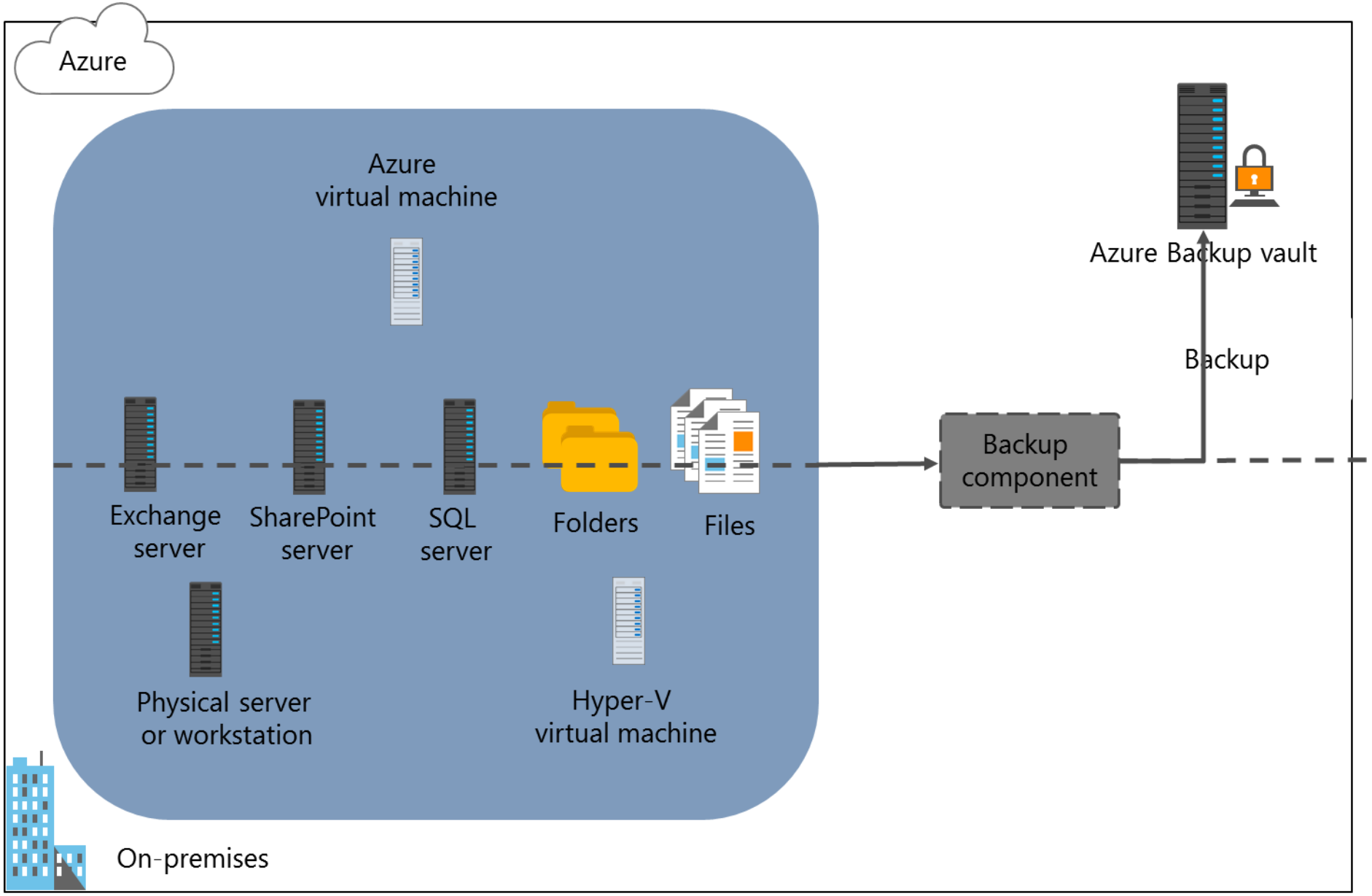
Using custom domains to access CDNs

- 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

Lesson 4: Implementing Azure Backup

- Overview of Azure Backup
- File and folder backups with the Azure Site Recovery agent
- VM-level backup by using the Azure Backup VM extension
- Integrating Azure Backup with Data Protection Manager and Microsoft Azure Backup Server
- Demonstration: Implementing Azure IaaS virtual machine backups

Overview of Azure Backup



File and folder backups with the Azure Site Recovery agent

1. Create an Azure Backup vault
2. Configure vault replication type
3. Specify the backup goal
   - Location of the workload: On-premises
   - The workload type: Files and folders
4. Download the vault credentials
5. Download and install the Site Recovery agent
6. Register the computer with the vault and set the passphrase
7. Configure the initial backup type, choose files and folders to back up, and create a backup schedule

VM-level backup by using the Azure Backup VM extension

1. Create an Azure Backup vault
2. Configure vault replication type
3. Specify the backup goal
   - Location of the workload: Azure
   - The workload type: Virtual machine
4. Choose the backup policy
5. Specify the virtual machines to back up

Integrating Azure Backup with Data Protection Manager and Microsoft Azure Backup Server

Feature	System Center 2016 DPM	Azure Backup Server
Application workloads	Yes	Yes
Tape backup	Yes	No
Integration with System Center suite	Yes	No
System Center licensing required	Yes	No
Deduplication support	Yes	Yes

Demonstration: Implementing Azure IaaS virtual machine backups

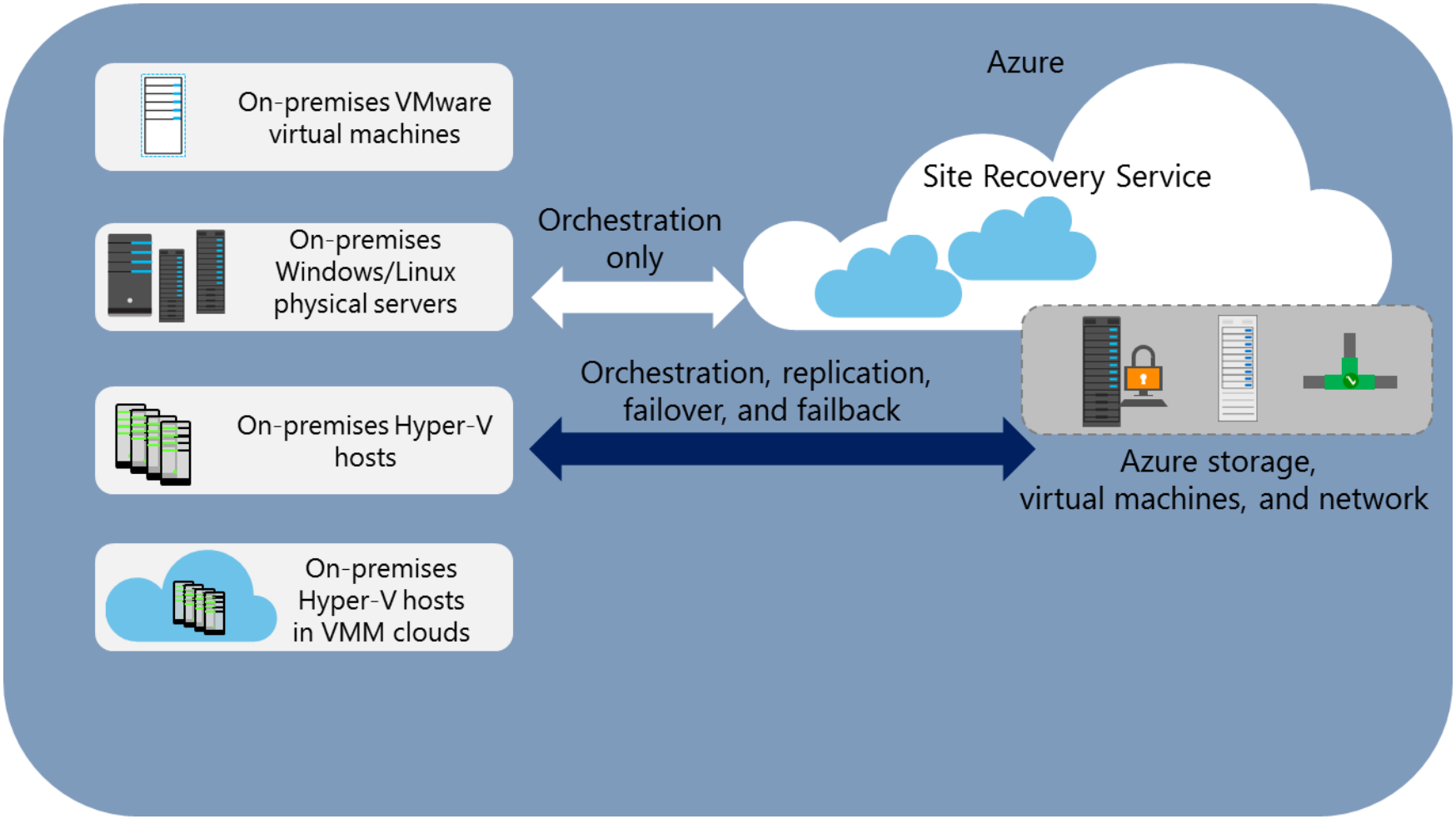
In this demonstration, you will see how to:

- Create an Azure Recovery Services vault
- Create a custom backup policy
- Register an Azure VM in the Recovery Services vault

Lesson 5: Planning and implementing Azure Site Recovery

- Overview of Azure Site Recovery
- Planning for Azure Site Recovery
- Implementing Azure Site Recovery

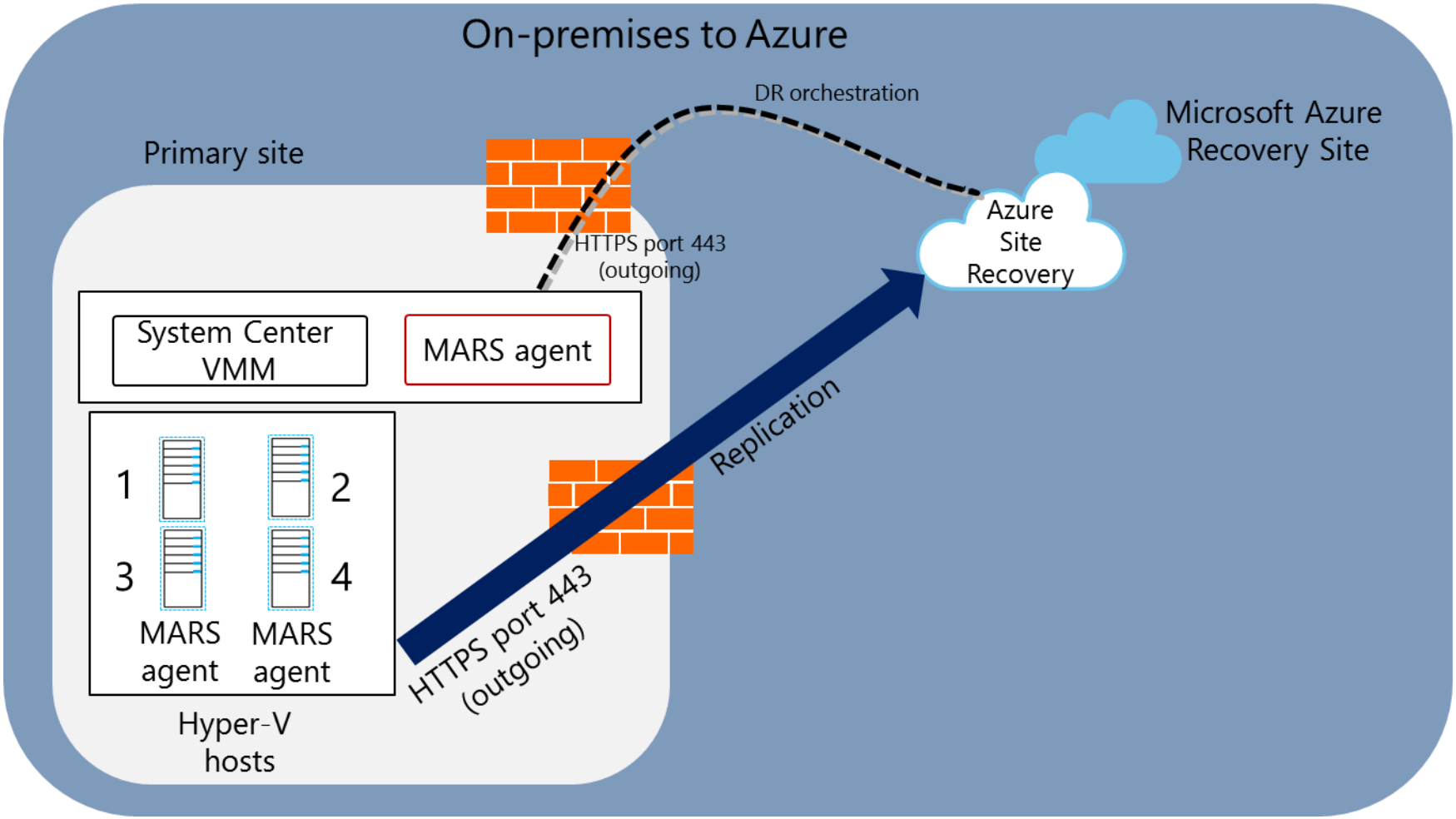
Overview of Azure Site Recovery



Planning for Azure Site Recovery

- Choose scenario:
  - Replicating Hyper-V VMs to Azure with VMM
  - Replicating Hyper-V VMs to Azure without VMM
  - Replicate VMware virtual machines and physical servers to Azure
  - Replicate Hyper-V VMs to a secondary datacenter
  - Replicate Hyper-V VMs to a secondary datacenter with SAN replication
  - Replicate between on-premises physical servers or VMware virtual machines in primary and secondary datacenters
- Plan capacity
  - Azure Site Recovery Capacity Planner
  - Hyper-V capacity planning tool
  - vSphere capacity planning appliance
- Identify supported workloads

Implementing Azure Site Recovery



Lab: Planning and implementing Azure Storage

- Exercise 1: Creating and configuring Azure Storage
- Exercise 2: Using Azure File storage
- Exercise 3: Protecting data with Azure Backup

Estimated Time: 60 minutes

Lab Scenario

The IT department at A. Datum 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 A. Datum'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. A. Datum 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. Additionally, A. Datum wants to evaluate the ability of Azure Backup to protect the content of on-premises computers and Azure IaaS virtual machines.

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

- Best Practices
- Review Question