Scenario 1: Deploying Virtual Machines (VMs) in Azure

Windows VM Setup

1. **Log in to Azure Portal:**
    1. Go to Azure Portal and log in with your Azure credentials.
2. **Create a Resource:**
    1. Click on **Create a resource** on the left sidebar.
    2. Under **Compute**, select **Virtual Machine**.
3. **Basic Configuration:**
    1. **Subscription**: Choose the subscription you want to use.
    2. **Resource Group**: Select an existing resource group or create a new one.
    3. **VM Name**: Enter a name for your VM (e.g., "Blogs").
    4. **Region**: Select the region where the VM should be located.
    5. **Image**: Choose the appropriate Windows Server version.
    6. **Size**: Select a VM size. For testing, a smaller size like B1s can be sufficient.
    7. **Authentication Type**: Choose **Password**.
    8. **Username**: Enter a username (e.g., "user").
    9. **Password**: Set a strong password.
4. **Disks:**
    1. Select the OS disk type (Standard SSD, Premium SSD, or Standard HDD). Premium SSD is recommended for better performance.
5. **Networking:**
    1. Select a Virtual Network (VNet) and Subnet. If none exist, Azure will create them for you.
    2. Configure a public IP if needed for external access (or choose "None" if not required).
6. **Review and Create:**
    1. Review the configuration and click **Create** to deploy the VM.

   b. Once deployed, access the VM using Remote Desktop Protocol (RDP) with the public IP and credentials you set.

Linux VM Setup

1. **Log in to Azure Portal** and follow the same process to create a new VM.
2. **Create Virtual Machine:**
   a. Under **Compute**, select **Virtual Machine**.
   b. Choose a Linux distribution (e.g., Ubuntu) in the **Image** section.
3. **Configure the VM:**
   a. Set the **VM name**, **Region**, **Size**, and **Authentication**.
   b. For Linux, use **SSH public key** authentication. Enter or generate an SSH key pair.
4. **Disk:**
   a. Select your preferred disk type (Standard SSD, Premium SSD, etc.).
5. **Networking:**
   a. Set up **VNet**, **Subnet**, and public IP configuration (as per your requirements).
6. **Review and Create:**
   a. After reviewing the settings, click **Create** to deploy the VM.

Pricing and OS Licensing

1. **Pricing Considerations:**
   a. VM Size
   b. Storage options
   c. Operating System
   d. Networking costs
   e. Availability Zones
2. **OS Licensing:**
   a. **Windows VMs**: Licensing is included in the price.

    b. **Linux VMs**: Free, but costs may apply if you use any premium services.
    c. **BYOL (Bring Your Own License)**: If you have existing licenses, you can use them.

Scenario 2: Azure Storage Encryption

Understanding Azure Storage Encryption

Azure Storage uses encryption to protect your data both **at rest** and **in transit.**

1. **Encryption at Rest**: Protects data stored on Azure from unauthorized access.
2. **Encryption in Transit**: Ensures data is encrypted during transfer across the network.

Types of Encryption in Azure Storage

- **Server-Side Encryption (SSE)**:
    - **SSE with Microsoft-managed keys** (default)
    - **SSE with customer-managed keys (CMK)**
    - **SSE with customer-provided keys (CPK)**
- **Azure Storage Service Encryption for Data at Rest (SSE)** applies to:
    - **Azure Blob Storage**
    - **Azure File Storage**
- **Encryption in Transit**: Uses **TLS (Transport Layer Security)**.

Enable Encryption for Sensitive Data in Azure Storage

1. **Create a Storage Account**:
    a. Log in to the Azure Portal.
    b. Navigate to **Create a resource** > **Storage** > **Storage account**.

   c. Provide the necessary details (Subscription, Resource Group, Account Name, Region).
   d. Choose **StorageV2 (general-purpose v2)** as the performance and redundancy option.
   e. Click **Create** to deploy the storage account.

2. **Enable Server-Side Encryption (SSE)**:
   a. Go to your Storage Account and navigate to **Encryption Settings**.
   b. Choose your encryption option and save the settings.
3. **Use Azure Key Vault for Key Management** (for CMK):
   a. **Create a Key Vault** and add an encryption key.
   b. **Configure your storage account** to use CMK for enhanced security.

Scenario 3: Setting up Azure DevOps Pipeline

Prerequisites

- Azure DevOps account
- Azure Subscription
- Azure App Service
- Code repository

Set Up the Azure DevOps Pipeline

1. **Create a Project in Azure DevOps**:
   a. Log in to Azure DevOps at dev.azure.com.
   b. Create a new project (e.g., "MyApp CI/CD") with the desired visibility (Private or Public).
2. **Create a Pipeline**:
   a. Inside your project, go to **Pipelines** > **New Pipeline**.
   b. Select your repository (Azure Repos Git or GitHub).
   c. Configure the pipeline to build and deploy your code.

3. **Configure Deployment to Azure App Service**:
    a. Add a **Build Task** to build your application.
    b. Add a **Deploy Task** to deploy your code to Azure App Service.
    c. Set up necessary deployment settings (e.g., App Service name, Resource Group).
    d. Save and run the pipeline.
4. **Set Up Failure Notifications**:
    a. Go to **Project Settings** and configure **Email Notifications** for pipeline events (success, failure, etc.).

Scenario 4: Azure Database Migration Service (DMS)

Overview of Azure DMS

The Azure Database Migration Service (DMS) helps migrate databases from on-premises (or other cloud environments) to Azure with minimal downtime.

Steps to Migrate an On-Premises SQL Database to Azure

1. **Prepare Your Environment**:
    a. Ensure your **Azure Subscription** is active.
    b. Verify that the **on-premises SQL Server database** is operational and accessible.
    c. Create an **Azure SQL Database** or **Managed Instance** as the target.
2. **Set Up Azure Database Migration Service (DMS)**:
    a. Log in to the **Azure Portal.**
    b. Search for **Azure Database Migration Service** and click **Create.**
    c. Select your **Subscription**, **Resource Group**, and provide a **Migration Service Name**.
3. **Create a Migration Project in DMS**:

a. After the DMS service is created, navigate to it and click **New Migration Project**.
b. Name your project and select the **Source server type** (SQL Server).
c. Select the **Target server type** (Azure SQL Database or Managed Instance).

4. **Configure Source and Target Connections**:
   a. **Source Server**: Enter connection details for your on-premises SQL Server (e.g., username, password).
   b. **Target Server**: Enter the connection details for your Azure SQL Database or Managed Instance.
5. **Choose Migration Method**:
   a. **Offline Migration**: The database will be offline during the migration.
   b. **Online Migration**: Continuous data replication allows minimal downtime.
6. **Start the Migration**:
   a. Perform the **Initial Migration**.
   b. Enable **Continuous Data Replication** if using online migration.
7. **Switch Over to the Azure Database**:
   a. **Final Cutover**: Once the data is synchronized, switch to the Azure database.
   b. **Verify Migration**: Ensure the data has migrated successfully.

Additional Considerations for Minimal Downtime Migration

- **Test the Migration**: Run tests to ensure application compatibility.
- **Network Latency**: Monitor latency and optimize for better performance.
- **Backup and Restore**: Always take a backup before initiating migration.
- **Monitor Migration Progress**: Use Azure DMS tools to track migration status.