

## Azure Tasks

### Scenario 1:

#### Windows VM:

1. **Log in to Azure Portal:**

Go to <https://azure.com> and log in using your Azure account credentials.

2. **Create a Resource:**

On the left sidebar, click on "Create a resource".

Under the "Compute" category, click "**Virtual Machine**".

3. **Basic Configuration:**

**1.Subscription:** Choose your subscription.

**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 Azure region where you want the VM to reside.

**5.Image:** Choose **Windows Server** (select the version based on your needs)

**6.Size:** Select a VM size. For testing, a smaller size like **B1s** could be sufficient. You can change this later based on your performance needs.

**7.Authentication Type:** Choose **Password**

**Username:** Enter a username (e.g., "user").

**Password:** Set a strong password.

4. **Disks:**

Choose the OS disk type (Standard SSD, Premium SSD, or Standard HDD).

**Premium SSD** is recommended for better performance but is more expensive.

5. **Networking:**

Select a Virtual Network (VNet) and Subnet. If you don't have one, Azure will create it for you.

Set up a public IP if needed for external access to the VM (you can choose "None" if the VM doesn't need external access).

6. **Review and Create:**

Review your settings and click **Create**. The VM will be deployed and you can access it via Remote Desktop Protocol (RDP) using the public IP and credentials you set earlier.

#### **For Linux VM:**

1. **Log in to Azure Portal** and go through the same process to create a resource.

2. **Create a Virtual Machine:**

Under **Compute**, click on **Virtual Machine**.

In the **Image** section, select a Linux distribution (e.g., Ubuntu).

3. **Configure the VM:**

Set the **VM name**, **Region**, **Size**, and **Authentication**.

For Linux, you'll likely choose **SSH public key** for authentication.

Enter the **SSH public key** or generate one if you don't have it already.

4. **Disk:**

As with the Windows VM, choose your disk type (Standard SSD, Premium SSD, etc.).

5. **Networking:**

Same as with the Windows VM: set up the networking options like VNet, Subnet, and public IP

6. **Review and Create:**

After reviewing the settings, click **Create**. Once the VM is deployed.

#### **Pricing and OS Licensing**

1. **Pricing:**

VM Size

Storage

Operating System

Networking

Availability Zones

#### **Operating System Licensing:**

Windows VMs

Linux VMs

BYOL (Bring Your Own License)

## **Scenario 2:**

### **1. Understanding Azure Storage Encryption**

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

**Encryption at Rest**

**Encryption in Transit**

### **2. 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)**

Azure Blob Storage

Azure File Storage

#### **Encryption in Transit**

TLS (Transport Layer Security).

### **Enable Encryption for Sensitive Data in Azure Storage**

#### **Step 1: Create a Storage Account**

**Log in to Azure Portal:**

Go to <https://portal.azure.com> and log in with your credentials.

**Create a Storage Account:**

In the left-hand menu, click **Create a resource**.

Choose **Storage > Storage account**.

Fill in the details like Subscription, Resource Group, Storage Account Name, Region, etc.

Choose **StorageV2** (general-purpose v2) as the performance and redundancy options.

**Review and Create:**

Review your settings and click **Create** to deploy the storage account.

**Step 2: Enable Server-Side Encryption (SSE)**

Go to the Storage Account:

Navigate to Encryption Settings

**Choose the Encryption Option:**

Save the Settings

**Step 3: Use Azure Key Vault for Key Management (if using CMK)**

Create a Key Vault:

Add an Encryption Key to Key Vault:

Configure Your Storage Account to Use CMK

**Scenario:3**

**Prerequisites**

Azure DevOps account

Azure Subscription

Azure App Service

Code repository

## Set Up the Azure DevOps Pipeline

### Step 1: Create a Project in Azure DevOps

#### Log in to Azure DevOps:

Go to <https://dev.azure.com> and log in.

#### Create a New Project:

Click on **New Project**.

Name your project (e.g., "MyApp CI/CD").

Set the visibility to **Private** or **Public**, depending on your preference.

Click **Create** to set up the project.

### Step 2: Create a Pipeline

#### Navigate to Pipelines:

Inside your project, click on **Pipelines** in the left menu.

#### Create a New Pipeline:

Click **New Pipeline**.

Select your repository (Azure Repos Git, GitHub).

#### Configure Pipeline Source:

Select the repository that contains your application code.

### Step 3: Configure the Pipeline to Deploy Code to Azure App Service

Add a Task to Build the Application:

Configure Build Task

Add a Task to Deploy to Azure App Service

Configure Deployment Settings

Save the Pipeline

## Set Up Failure Notifications

### Step 1: Set Up Email Notifications

Go to Project Settings:

Configure Notifications  
Choose Pipeline Events

## Scenario 4:

### Azure Database Migration Service (DMS)

The **Azure Database Migration Service (DMS)** is specifically designed to help you migrate databases from on-premises (or other cloud environments) to Azure.

#### Key Features of DMS:

Online Migration  
Minimal Impact  
Support for SQL Server  
Schema and Data Migration

### Steps to Migrate an On-Premises SQL Database to Azure

#### Step 1: Prepare Your Environment

**Azure Subscription:** Ensure you have an Azure subscription set up and access to the Azure Portal.

**On-Premises SQL Server Database:** Ensure your on-premises SQL Server database is operational and accessible from the Azure environment.

**Azure SQL Database or Managed Instance:** Create an **Azure SQL Database**

#### Step 2: Set Up Azure Database Migration Service (DMS)

##### Log in to Azure Portal:

Go to the AZURE PORTAL log in using your Azure credentials.

##### Create the Migration Service:

In the Azure Portal, search for **Azure Database Migration Service**.

Click **Create** to set up a new migration project.

Select the **Subscription, Resource Group**, and provide a **Migration Service Name**.

### Step 3: Set Up the Migration Project in DMS

#### 1. Create a Migration Project:

After the service is created, navigate to **Azure Database Migration Service** in the Azure Portal.

Click on the DMS instance you created, then click on **+ New Migration Project**.

Name your project and choose the **Source server type** (select **SQL Server**). Choose the **Target server type** (select either **Azure SQL Database** or **Azure SQL Managed Instance**, depending on your target).

#### 2. Configure Source and Target:

**Source Server:** Provide connection details for your on-premises SQL Server database (username, password, etc.).

**Target Server:** Provide the connection details for your Azure SQL Database or Managed Instance.

### Step 4: Select the Migration Method

**Choose Migration Type:** DMS offers two types of migration:

Offline Migration

Online Migration

### Step 5: Start the Migration

Perform Initial Migration

Enable Continuous Data Replication

### Step 6: Switch Over to the Azure Database

Final Cutover

Verify the Migration

### Additional Considerations for Minimal Downtime Migration

Test the Migration

Network Latency

Backup and Restore

Monitor Migration Progress

Post-Migration Testing