**Assignment on azure**

**G. Vaishnavi**

**ACE12366**

**Scenario 1:** Your team needs to deploy a virtual machine in the Azure portal to test a new software application. The team has requested both Windows and Linux virtual machines.

**Question:** How would you set up these virtual machines? What considerations are needed for pricing and OS licensing?

**Answer:**

**Steps to create a virtual machine:**

1. Go to the **Microsoft Azure** portal, and in the left sidebar, select **Virtual Machines**.
2. Click on **+ Add** to create a new virtual machine.
3. Choose a **subscription**.
4. Select a **Resource Group**.
5. Configure the VM settings:
   * **VM name**: Provide a name for the virtual machine.
   * **Region**: Select the region.
   * **Availability Options**: Choose the appropriate availability option.
   * **Image**: Select the operating system (Windows, Linux, etc.).
   * **Size**: Pick the desired size (CPU, RAM).
   * **Authentication type**: Choose SSH key (for Linux) or password (for Windows).
   * **Inbound port rules**: Select the required ports (e.g., SSH for Linux, RDP for Windows).
6. **Storage Configuration**:
   * Azure VMs use **managed disks** for storage. Choose between Standard SSD, Premium SSD, or Standard HDD based on performance needs.
7. **Networking**:
   * Either choose an existing Virtual Network or create a new one.
8. **Review and Create**:
   * Review the configuration and click **Create** to deploy the VM.

**Pricing Considerations:**

* **VM size** (CPU, RAM) and type (general-purpose, compute-optimized, or memory-optimized).
* **Operating System**:
  + **Windows**: Charges apply for both the compute resource and the Windows license.
  + **Linux**: Generally free (no licensing costs).
* **Licensing models**: Options like **pay as you go**, **reserved instances**, and **Azure Hybrid Benefit**.
* **Storage Costs**: Costs depend on the disk type (Standard SSD, Premium SSD, or Standard HDD) and disk size (e.g., 128GB, 512GB).

**Scenario 2:** The IT security team has requested that sensitive data stored in an Azure storage account be encrypted to meet compliance requirements.

**Question:** How can you ensure the data stored in Azure storage is encrypted, and what encryption types are available?

**Answer:**

To ensure data stored in an Azure Storage Account is encrypted, you can use **Azure Storage Encryption (SSE)**, which automatically encrypts data at rest using **AES-256 encryption** by default. No extra configuration is required when using **Microsoft-managed keys**.

**Encryption Types:**

* **Microsoft-Managed Keys (MMK)**: Azure automatically manages encryption keys.
* **Customer-Managed Keys (CMK)**: You manage the encryption keys.
* **Client-Side Encryption (CSE)**: You encrypt the data before uploading it to Azure storage.
* **Infrastructure Encryption**: Adds an additional layer of encryption to the infrastructure.

**Scenario 3:** You are responsible for setting up a DevOps pipeline in Azure DevOps for your application. The pipeline must deploy code to an Azure app service and notify the team if the deployment fails.

**Question:** How would you configure this pipeline to meet this requirement?

**Answer:**

**Steps to configure the DevOps pipeline:**

1. Go to **Azure DevOps** and log in.
2. Click **New Project**, enter a project name, and select **Private/Public repo**, **Version Control**, and **Work Item**.
3. Click **Create** to create the project.
4. Navigate to your project, import the repository, and push the code.
5. Go to **Pipelines** and click **New Pipeline**.
6. Select the repository.
7. Choose **Starter Pipeline** (or “Existing YAML” if you already have a template).
8. Navigate to **Project Settings** in Azure DevOps.
9. Click **New Service Connection** and select **Azure Resource Manager**.
10. Choose **Service Principal** and select the **Subscription** and **App Service**.
11. Click **Save** to save the connection.
12. Go back to **Project Settings** and select **Notifications**.
13. Click **New Subscription**.
14. Choose **Build Completed** and set the condition to trigger on **failures** only.
15. Add the **team’s email addresses**.
16. Click **Save**.
17. Go to **Pipelines**, select your pipeline, and click **Run**.
18. Check the logs to monitor deployment.
19. If a failure occurs, the email notification will be sent to the team.

**Scenario 4:** Your organization is moving its on-premises SQL database to Azure. The database must remain accessible during migration with minimal downtime.

**Question:** Which Azure service could you use, and how could you perform the migration?

**Answer:**

**Azure Service: Azure Database Migration Service (DMS)**

**Steps to migrate the SQL database with minimal downtime:**

1. Ensure the on-premises SQL server is running and accessible.
2. Take a **backup** as a precaution before migration.
3. Enable **Transaction Log Backups** to ensure minimal downtime.
4. Choose **Azure SQL Database** as the destination.
5. Create an **Azure SQL server** and configure the network settings.
6. Deploy **Azure Database Migration Service (DMS)** in the Azure portal.
7. Choose the **Online Migration** option for minimal downtime.
8. Connect the source SQL server and the destination Azure SQL Database.
9. Start the migration process using DMS.
10. Monitor the progress of the migration through the Azure portal.
11. Once the migration is complete, validate the data integrity.