Azure Scenario

Name : Ram Prasath TJ

ACEID :12528

Scenario 1:

your team needs to deploy a virtual machine in azure portal to test the new software application.
here the team has requested both windows and linux vm.

question:

- how could u setup these vm? and what considerations are needed for pricing an OS license? To deploy both **Windows** and **Linux** VMs in Azure for testing a new software application:

1. Setting up VMs:

- **Windows VM**:
  - Go to Azure Portal → **Create a Resource** → **Windows Server** or **Windows 10/11 VM**.
  - Choose size, region, network settings, and configure an admin username/password.
- **Linux VM**:
  - Go to Azure Portal → **Create a Resource** → **Linux** (choose from popular distributions like Ubuntu, CentOS, etc.).
  - Choose size, region, network settings, and configure an SSH key for secure access.

2. Considerations for Pricing & OS License:

- **Windows VM**:
  - Pricing includes the cost of the **Windows Server license**, which is bundled in the hourly VM cost.
  - Consider using **Azure Hybrid Benefit** if you have existing Windows Server licenses with Software Assurance for discounted pricing.

- **Linux VM**:
    - No additional licensing cost for Linux (as most distros are free).
    - Choose a distribution with support if required, as some offer paid support (e.g., Red Hat, SUSE).

- ## 3. Other Considerations:

- **VM Size**: Choose appropriate VM sizes based on resource needs (CPU, memory).
- **Region**: Select a region closest to your location or where testing will occur for lower latency.
- **Networking**: Configure virtual networks, public IPs, and security groups for proper communication between VMs.

2. **Second Scenario** :

The IT security team has requested the sensitive data stored in a azure storage account the encrypted to meet complaince requirements

question:

How could you ensure the data stored in azure storage is encrypted , and what encryption types are available.

To ensure sensitive data stored in an Azure Storage account is encrypted:

1. Encryption Methods:

- **Azure Storage Encryption (by default)**:
    - Data is encrypted at rest using **Storage Service Encryption (SSE)** with **Microsoft-managed keys** by default.
- **Customer-Managed Keys (CMK)**:
    - For more control, use **Azure Key Vault** to manage encryption keys for **Azure Storage** with **customer-managed keys** (SSE with customer-managed keys).

2. Encryption Types Available:

- **Encryption at Rest**:
  - **SSE (Microsoft-managed keys)**: Default encryption for data at rest.
  - **SSE with Customer-managed Keys (CMK)**: For full control over encryption keys.
- **Encryption in Transit**:
  - Data is encrypted using **SSL/TLS** during transfer to and from Azure Storage.

3. Considerations:

- Enable **Azure Storage Firewall** and **Virtual Networks** for additional security.
- Use **Azure Key Vault** for managing and rotating encryption keys securely.

3. Third Scenario :

Your responsible for setting up devops pipeline in azure devops for your application .
The pipeline must deploy code to an azure app servie and notify the team if the deployment fails

Question : How could you configure this pipeline to meet this requirements?

If the deployment fails, notifying the team and providing actionable insights is critical. Here's how you can configure the pipeline to notify the team effectively while providing relevant failure details:

1. Enable Built-In Azure DevOps Notifications

1. Go to your **Azure DevOps Project** → **Project Settings** → **Notifications**.
2. Create a **New Notification Subscription**:
    a. **Trigger:** Pipeline Fails
    b. **Scope:** Select the specific pipeline(s) where this applies.
    c. **Recipients:** Add team members, group emails, or service accounts for integrations with Slack, Microsoft Teams, etc.
    d. **Details:** Include failure information in the notification.

2.Integrate Application Insights Alerts:

1. Configure **Azure Monitor** with **Application Insights** to detect runtime failures or performance issues after deployment.
2. Set up alert rules for error codes, latency, or availability issues.
3. Send notifications to the team via email, SMS, or integration with tools like PagerDuty, Slack, or Teams.

4. Fourth scenario :

your organization in on permises SQL DAtabase to azure.
the database must remain accesible during migaration with minimal downtime.

Question :Which azure service would you use , and how could you perform the migration?

To migrate an on-premises SQL Server database to Azure with minimal downtime:

1. Use **Azure Database Migration Service (DMS)** for migration.
2. Set up DMS with **Online Migration** for minimal downtime.
3. Migrate the schema and initial data to Azure SQL Database or Managed Instance.
4. DMS will continuously replicate transactional changes during the migration.
5. Ensure network connectivity between on-premises and Azure environments.
6. After initial migration, synchronize ongoing changes from on-premises to Azure.
7. Perform a **cutover** once the databases are fully synchronized.
8. Redirect application traffic to the Azure SQL Database.
9. Validate data integrity and test applications post-migration.
10. Optionally, decommission the on-premises SQL Server if no longer needed.

This approach ensures minimal downtime and a smooth transition to Azure.