BHADRINATH P S **21-JAN-2024** **INSURANCE**

AZURE - CASE STUDY QUESTIONS

1. Your team needs to deploy a VM in Azure portal or CLI to test a new software application. Here, the team has requested both Linux and Windows VMs.

To deploy both Linux and Windows VMs in Azure, we can easily set them up via Azure portal or CLI, selecting the appropriate operating system, VM size and configurations. The cost charges primarily depend on the VM type, storage and licensing requirements for Windows and Linux. It's very important for us to evaluate pricing models and licensing options to optimize costs based on our usage.

(I) How could you set up these VMs?

Linux VM:

Step 1: In the Azure Portal, Navigate to **Virtual Machines** in the left-hand menu.
Step 2: Click **+ Add** to create a new VM.
Step 3: Select the **Subscription** and **Resource Group** where we want the VM.
Step 4: Choose **Linux** as OS and select the distribution we want (Ubuntu, etc.).
Step 5: Provide a **VM name**, **Region**, and **Size** (e.g., Standard B1).
Step 6: Set up **Authentication** and Configure additional settings such as networking, storage, etc.
Step 7: Review our configuration and click **Create** to deploy the VM.

```
az vm create --resource-group --name --image UbuntuLTS --size Standard_B1s --admin-username --authentication-type ssh --ssh-key-value
```

Windows VM:

Step 1: In the Azure portal, click **+ Add** again to create a new VM.
Step 2: Select the **Subscription** and **Resource Group** for the Windows VM.
Step 3: Choose **Windows** as OS and select the desired version (e.g., Windows 7).
Step 4: Fill in the **VM name**, **Region**, and **Size** (e.g., Standard D2s).

Step 5: Set up **Authentication** and Configure additional settings like networking, etc.
Step 6: After reviewing the configuration, click **Create** to deploy the Windows VM.

```
az vm create --resource-group --name --image Win2019Datacenter --size Standard_D2s_v3 --admin-username --admin-password
```

(II) What considerations are needed for pricing and licensing?

Pricing Considerations:

- **VM Size and Type:** Smaller VMs like Standard_B1s are cheaper for testing, while larger VMs like Standard_D2s_v3 cost more.
- **Storage Costs:** Costs vary by disk type (Standard HDD, SSD, Premium SSD). Linux VMs usually have lower storage costs than Windows VMs.
- **Network Costs:** Data transfer across regions or out of Azure may require extra charges.
- **Regional Pricing:** Prices vary by region. Use the Azure Pricing Calculator for estimates.

License Considerations:

- **Windows License:** Windows VMs include the OS license. Use Azure Hybrid Benefit if we have existing licenses to reduce costs.
- **Linux License:** Most Linux distros are free, but commercial ones like RHEL or SUSE may have additional support fees.

---

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

To store sensitive data in Azure Storage and meet compliance requirements, we can enable encryption to ensure the data is securely protected. Azure provides several built-in encryption options to help safeguard data at rest and during transmission.

(I) How could you store data in Azure Storage as encrypted?

Azure Storage Encryption:

- **By default**, all data stored in Azure Storage (such as blobs, files, queues, and tables) is encrypted at rest using **Microsoft-managed keys**.
- For additional control, you can use **customer-managed keys** (CMK) to encrypt your data using your own keys, either stored in Azure Key Vault or provided via other means.

Steps to Enable Encryption:

- Azure Blob Storage: Go to your storage account in the Azure portal, select Encryption under Settings, and choose between Microsoft-managed keys or Customer-managed keys (if using your own keys).
- Azure File Storage: Enable encryption in the Encryption section of your storage account settings.

(II) What encryption types are available?

Encryption Types Available:

- Microsoft-managed keys
- Customer-managed keys (CMK)
- Azure Storage Service Encryption (SSE)
- Client-Side Encryption

---

3. You are responsible for setting up a DevOps pipeline in Azure DevOps for your application. The pipelines must deploy the code to an Azure App Service and notify the team if the deployment fails.

To configure an Azure DevOps pipeline for deploying code to an Azure App Service and notifying the team if the deployment fails, we need to set up the deployment task in the pipeline, enable failure notifications, and optionally configure approval stages. Azure DevOps provides built-in tasks for App Service deployment and customizable notification settings for failure alerts.

(I) How could you configure the pipelines to meet the requirement?

Create the Pipeline

- In Azure DevOps, navigate to Pipelines and create a new pipeline.
- Select the repository containing our application code and choose a pipeline template (or create a custom YAML pipeline).

Add Azure App Service Deployment Task

- Add the Azure App Service Deploy task to the pipeline.
- Configure the Azure subscription and the App Service name where the application will be deployed.
- Select the build artifact (e.g., a zip package) for deployment.

Enable Failure Notifications

- Set up failure notifications through Azure DevOps Notification settings.
- Configure email or service hooks (e.g., Slack or Microsoft Teams) to notify the team if the deployment fails.
- Use the failed() condition to trigger notifications only when the deployment fails.

---

4. Your organization is moving from an on-premises SQL database to Azure. The database must remain accessible during the migration with minimal downtime.

To migrate an on-premises SQL database to Azure with minimal downtime while ensuring the database remains accessible, you should use **Azure Database Migration Service (DMS)**. This service enables seamless database migration with minimal disruption to your operations.

(I) Which Azure service do you use?

- **Azure Database Migration Service (DMS):** This is the recommended service for migrating SQL databases from on-premises to Azure, ensuring minimal downtime during the process.
- DMS supports migrations from **SQL Server to Azure SQL Database** or **SQL Managed Instance**, providing tools to manage the entire migration process.

(II) How could you perform the migration?

Steps to Perform the Migration:

1. **Prepare the Environment**: Ensure the on-premises SQL database is supported and set up an Azure SQL Database or Managed Instance in the target region. Create a DMS instance.
2. **Configure the Migration**: In DMS, create a migration project, specify the source and target databases, and install the DMS agent on the on-premises server.
3. **Initial Data Migration**: Run the full migration to move the bulk of data while keeping the on-premises database accessible.
4. **Enable Continuous Data Replication**: Sync ongoing changes between the on-premises database and Azure, minimizing downtime.
5. **Cut Over to Azure**: Stop changes to the on-premises database, perform the final sync and switch to Azure.
6. **Post-Migration Testing**: Test the Azure database to ensure functionality and monitor performance.

Considerations for Minimal Downtime:

- **Continuous Replication**: Keeps the databases in sync, minimizing downtime.
- **Synchronization**: Ensure final data sync is complete before cutover.
- **Downtime Window**: The downtime during cutover is typically short, lasting only minutes to a couple of hours.
