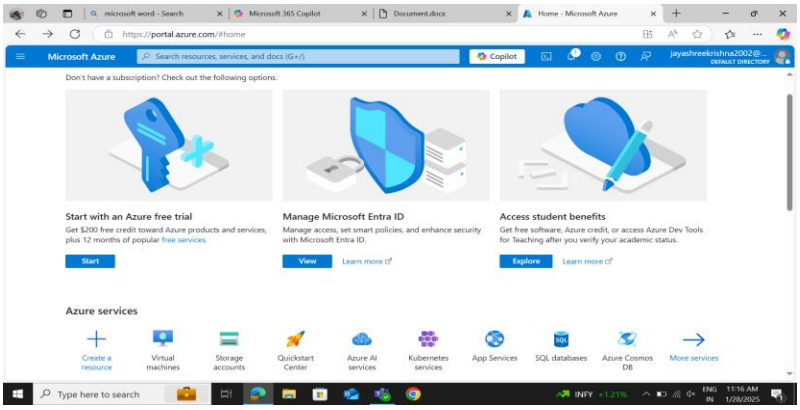
Name: Jayashree K
Date: 28/01/2025
Domain: Testing

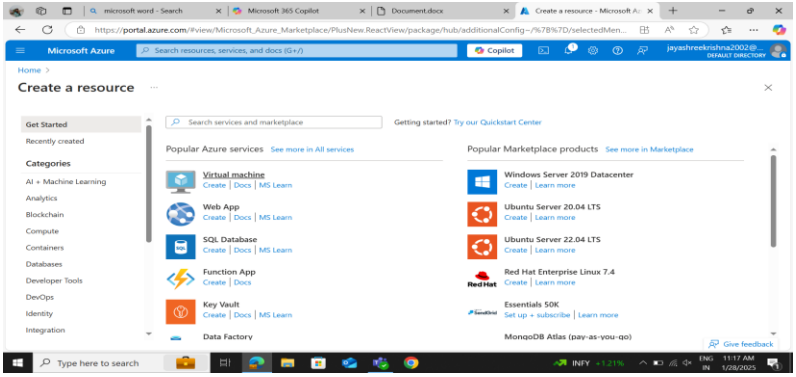
AZURE SCENARIO BASED QUESTIONS

Scenario 1: Your team needs to deploy a virtual machine in Azure portal or CLI to test a new software application. The team has requested both Windows and Linux machines.
Question: How could you set up these virtual machines? What considerations are needed for pricing and OS licensing?

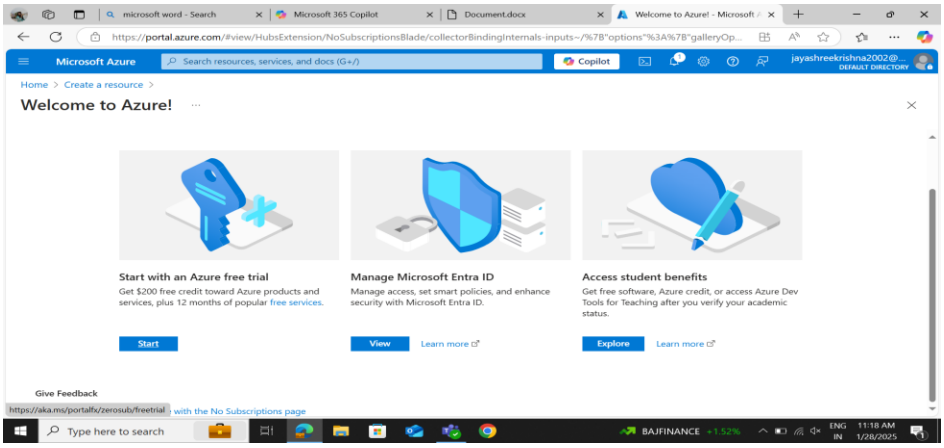
Answer (Windows):
Steps:
Step 1: Login to the Azure portal with valid credentials.
Step 2: Click on “Create resource” below Azure Services.



Step 3: Click on “Virtual machine”.



Step 4: Select the Azure subscription (Start with an Azure free trial)



Step 5: Setup the basic configurations: Resource Group, VM Name, Region, Size, Authentication.
Step 6: Choose SSD or HDD based on performance needs.
Step 7: Use the default “Virtual machine”.
Step 8: Click **Review + Create** and then **Create**.

Answer(Linux):
Steps: Repeat the same steps as Windows, but in Linux select image as Linux distribution, use SSH key for Authentication, instead of RDP use SSH for connection method.

Pricing and OS Licensing:

- Windows VMs: Higher cost as OS licensing is included in the VM pricing.
- Linux VMs: Generally cheaper; most distributions (e.g., Ubuntu) are free. Paid distributions (e.g., Red Hat, SUSE) incur extra charges.
- Key Difference: Windows is ideal for Windows-specific applications, while Linux suits open-source and web-based environments.

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

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

Answer:
To ensure the data stored in an Azure Storage Account is encrypted to meet compliance requirements, you can use Azure Storage Encryption. Azure automatically encrypts data at rest using Microsoft-managed keys by default.

Azure Storage Service Encryption (SSE) - Azure automatically encrypts all data at rest using AES-256 encryption. No additional configuration is required for Microsoft-managed keys.

Encryption Types: Microsoft-Managed Keys (MMK), Customer-Managed Keys (CMK), Client-Side Encryption (CSE), Infrastructure Encryption.

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 could you configure this pipeline to meet this requirement?

Answer:
Step 1: Go to Azure DevOps and sign in
Step 2: Click New Project and name it
Step 3: Select Private/Public repo, Version Control and Work Item.
Step 4: Click Create
Step 5: Navigate to your project, Import the repo and push the code
Step 6: Go to Pipelines and Click New Pipeline.
Step 7: Select the repository
Step 8: Choose “Starter Pipeline”, if existing then “Existing YAML”
Step 9: Navigate to Azure DevOps and select Project Settings.
Step 10: Click on New Service Connection and Select Azure Resource Manager.
Step 11: Choose Service Principle
Step 12: Select your Subscription and App Service
Step 13: Click Save
Step 14: Again, go to Project Settings and navigate to Notifications
Step 15: Click New Subscription
Step 16: Select Build Completed
Step 17: Set the condition to trigger only on failures

Step 18: Add team's email addresses
Step 19: Click Save
Step 20: Go to Pipelines and select the pipeline
Step 21: Click Run & Check logs
Step 22: If a failure occurs, email notification will be sent.

Scenario 4: Your organization is moving its 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: Use Azure Database Migration Service (DMS) to migrate the SQL database with minimal downtime.

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