Azure Identity and Access Management tasks

Practical Task 1: Introduction to Microsoft Entra ID

Create a basic Microsoft Entra ID setup for an organization to manage identity and access.

Requirements:

1. Create a new Microsoft Entra ID tenant.
2. Add at least two users to the directory.
3. Create two groups named **Developers** and **Admins**.
4. Assign the users to appropriate groups.
5. Assign the **Global Reader** role to the **Admins** group.
6. Assign the **Application Developer** role to the **Developers** group.
7. Verify that the role assignments function as expected for both groups.

Practical Task 2: Enabling Single Sign-On (SSO) and Multi-Factor Authentication (MFA)

Configure Single Sign-On (SSO) and Multi-Factor Authentication (MFA) for users in a Microsoft Entra ID directory to enhance identity and access security.

Requirements:

1. Enable Single Sign-On (SSO) for your Microsoft Entra ID tenant.
2. Enforce Multi-Factor Authentication (MFA) for all users in the directory.
3. Configure conditional access policies to require MFA for high-risk sign-ins.
4. Verify that SSO and MFA settings are correctly applied for the users.

Practical Task 3: Implementing Role-Based Access Control (RBAC)

Implement Role-Based Access Control (RBAC) in Azure to manage access to resources based on roles and ensure fine-grained access management.

Requirements:

1. Create a custom role named **Resource Viewer** with read-only permissions for a specific resource group.
2. Assign the **Resource Viewer** role to the **Developers** group created earlier.
3. Assign the built-in **Contributor** role to the **Admins** group for the same resource group.
4. Verify that members of the **Developers** group have only read access and members of the **Admins** group have full access to the resource group.

Practical Task 4: Securing Sensitive Information with Azure Key Vault

Set up Azure Key Vault to securely store and manage sensitive information such as keys, secrets, and certificates.

Requirements:

1. Create a new Azure Key Vault in your subscription.
2. Add a secret to the Key Vault (e.g., a database connection string).
3. Set access policies to grant the **Application Developer** role (assigned to the **Developers** group) permission to retrieve secrets from the Key Vault.
4. Verify that only members of the **Developers** group can access the stored secret.

Practical Task 5: Creating and Assigning Basic Azure Policies

Define and assign Azure Policies to enforce compliance with organizational standards for resource management.

Requirements:

1. Create an Azure Policy to enforce tagging for all newly created resources with a specific tag (e.g., Environment: Development).
2. Assign the policy to a resource group.
3. Verify that any new resource created in the resource group without the required tag is marked as non-compliant.
4. Review and document the compliance status of the resource group.

Practical Task 6: Using Policy Effects to Enforce Compliance

Configure Azure Policies with different policy effects to enforce compliance and manage resources according to organizational standards.

Requirements:

1. Create a policy with the **Audit** effect to monitor and log untagged resources within a resource group.
2. Create a policy with the **DeployIfNotExists** effect to automatically add a specific tag (Owner: IT) to any newly created resource.
3. Assign these policies to a resource group and verify their behavior by:
    - Creating a resource without a tag and checking the compliance logs.
    - Creating a resource to validate the automatic tag deployment.