Custom View Settings

Question #2 *Topic 3*

You have a new Azure subscription. You are developing an internal website for employees to view sensitive data. The website uses Azure Active Directory (Azure
AD) for authentication.
You need to implement multifactor authentication for the website.
Which two actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

A. Configure the website to use Azure AD B2C.

B. In Azure AD, create a new conditional access policy.

C. Upgrade to Azure AD Premium.

D. In Azure AD, enable application proxy.

E. In Azure AD conditional access, enable the baseline policy.

Correct Answer: *BC*
B: MFA Enabled by conditional access policy. It is the most flexible means to enable two-step verification for your users. Enabling using conditional access policy only works for Azure MFA in the cloud and is a premium feature of Azure AD.
C: Multi-Factor Authentication comes as part of the following offerings:
➪ Azure Active Directory Premium licenses - Full featured use of Azure Multi-Factor Authentication Service (Cloud) or Azure Multi-Factor Authentication Server
(On-premises).
➪ Multi-Factor Authentication for Office 365
➪ Azure Active Directory Global Administrators
Reference:
https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-getstarted

Question #3 *Topic 3*

You are developing a Java application that uses Cassandra to store key and value data. You plan to use a new Azure Cosmos DB resource and the Cassandra
API in the application. You create an Azure Active Directory (Azure AD) group named Cosmos DB Creators to enable provisioning of Azure Cosmos accounts, databases, and containers.
The Azure AD group must not be able to access the keys that are required to access the data.
You need to restrict access to the Azure AD group.
Which role-based access control should you use?

A. DocumentDB Accounts Contributor

B. Cosmos Backup Operator

C. Cosmos DB Operator

D. Cosmos DB Account Reader

Correct Answer: *C*
Azure Cosmos DB now provides a new RBAC role, Cosmos DB Operator. This new role lets you provision Azure Cosmos accounts, databases, and containers, but canλ€™t access the keys that are required to access the data. This role is intended for use in scenarios where the ability to grant access to Azure Active Directory service principals to manage deployment operations for Cosmos DB is needed, including the account, database, and containers.
Reference:
https://azure.microsoft.com/en-us/updates/azure-cosmos-db-operator-role-for-role-based-access-control-rbac-is-now-available/

Question #4 *Topic 3*

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.
You are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.
You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.
You need to configure authorization.
Solution: Configure the Azure Web App for the website to allow only authenticated requests and require Azure AD log on.
Does the solution meet the goal?

A. Yes

B. No

Correct Answer: *B*
Instead in the Azure AD applicationλ€™s manifest, set value of the groupMembershipClaims option to All.
Reference:
https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

Question #5 *Topic 3*

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.
You are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.
You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.
You need to configure authorization.
Solution:
☞ Create a new Azure AD application. In the application's manifest, set value of the groupMembershipClaims option to All.
☞ In the website, use the value of the groups claim from the JWT for the user to determine permissions.
Does the solution meet the goal?

A. Yes

B. No

Correct Answer: *A*
To configure Manifest to include Group Claims in Auth Token
1. Go to Azure Active Directory to configure the Manifest. Click on Azure Active Directory, and go to App registrations to find your application:
2. Click on your application (or search for it if you have a lot of apps) and edit the Manifest by clicking on it.
3. Locate the λ€groupMembershipClaimsλ€ setting. Set its value to either λ€SecurityGroupλ€ or λ€Allλ€. To help you decide which:
☞ λ€SecurityGroupλ€ - groups claim will contain the identifiers of all security groups of which the user is a member.
☞ λ€Allλ€ - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member
Now your application will include group claims in your manifest and you can use this fact in your code.
Reference:
https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

Question #6 *Topic 3*

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.
You are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.
You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.
You need to configure authorization.
Solution:
☞ Create a new Azure AD application. In the application's manifest, define application roles that match the required permission levels for the application.
☞ Assign the appropriate Azure AD group to each role. In the website, use the value of the roles claim from the JWT for the user to determine permissions.
Does the solution meet the goal?

A. Yes

B. No

Correct Answer: *B*
To configure Manifest to include Group Claims in Auth Token
1. Go to Azure Active Directory to configure the Manifest. Click on Azure Active Directory, and go to App registrations to find your application:
2. Click on your application (or search for it if you have a lot of apps) and edit the Manifest by clicking on it.
3. Locate the λ€groupMembershipClaimsλ€ setting. Set its value to either λ€SecurityGroupλ€ or λ€Allλ€. To help you decide which:
☞ λ€SecurityGroupλ€ - groups claim will contain the identifiers of all security groups of which the user is a member.
☞ λ€Allλ€ - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member
Now your application will include group claims in your manifest and you can use this fact in your code.
Reference:
https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

Previous Questions Next Questions