Custom View Settings

Question #22 *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 develop Azure solutions.
You must grant a virtual machine (VM) access to specific resource groups in Azure Resource Manager.
You need to obtain an Azure Resource Manager access token.
Solution: Run the Invoke-RestMethod cmdlet to make a request to the local managed identity for Azure resources endpoint.
Does the solution meet the goal?

A. Yes

B. No

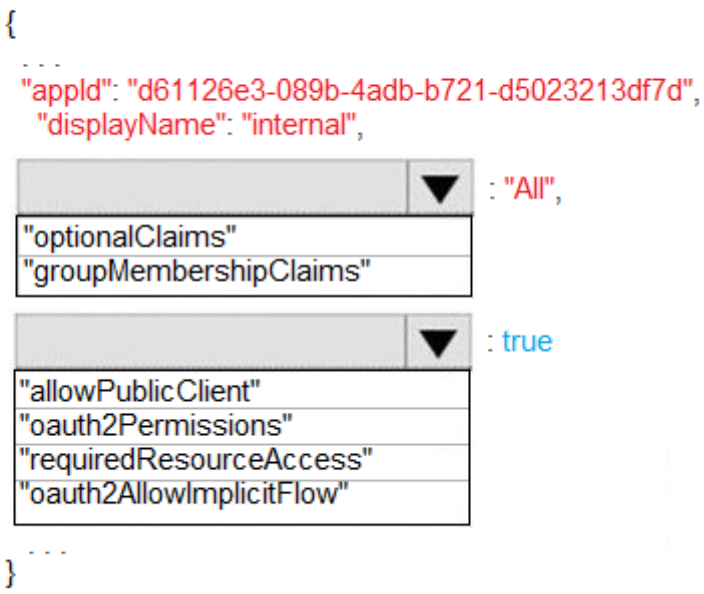
Correct Answer: *A*
Get an access token using the VM's system-assigned managed identity and use it to call Azure Resource Manager
You will need to use PowerShell in this portion.
1. In the portal, navigate to Virtual Machines and go to your Windows virtual machine and in the Overview, click Connect.
2. Enter in your Username and Password for which you added when you created the Windows VM.
3. Now that you have created a Remote Desktop Connection with the virtual machine, open PowerShell in the remote session.
4. Using the Invoke-WebRequest cmdlet, make a request to the local managed identity for Azure resources endpoint to get an access token for Azure Resource
Manager.
Example:
$response = Invoke-WebRequest -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https:// management.azure.com/' -Method GET -Headers @{Metadata="true"}
Reference:
https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/tutorial-windows-vm-access-arm

Question #23 *Topic 3*

HOTSPOT -
You are building a website to access project data related to teams within your organization. The website does not allow anonymous access. Authentication is performed using an Azure Active Directory (Azure AD) app named internal.
The website has the following authentication requirements:
- Azure AD users must be able to login to the website.
- Personalization of the website must be based on membership in Active Directory groups.

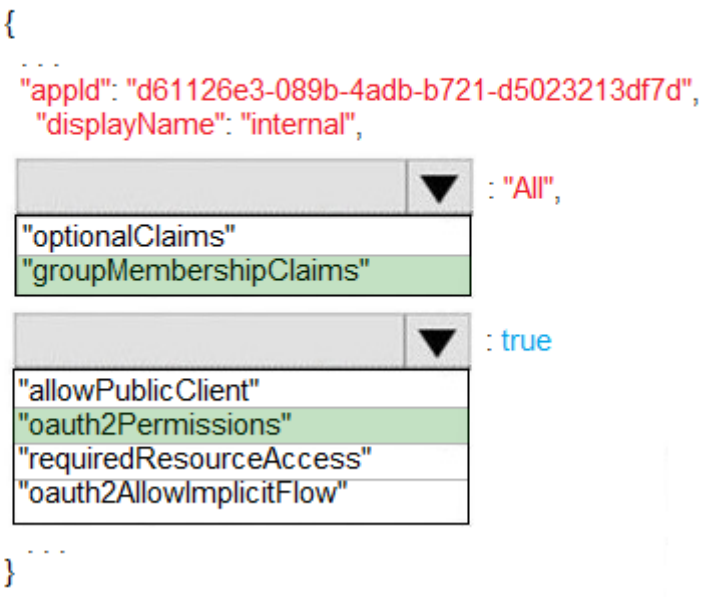
You need to configure the application's manifest to meet the authentication requirements.
How should you configure the manifest? To answer, select the appropriate configuration in the answer area.
NOTE: Each correct selection is worth one point.
Hot Area:

Answer Area



Correct Answer:

Answer Area



Box 1: groupMembershipClaims -
Scenario: Personalization of the website must be based on membership in Active Directory groups.
Group claims can also be configured in the Optional Claims section of the Application Manifest.
Enable group membership claims by changing the groupMembershipClaim
The valid values are:
"All"
"SecurityGroup"
"DistributionList"
"DirectoryRole"

Box 2: oauth2Permissions -
Scenario: Azure AD users must be able to login to the website. oauth2Permissions specifies the collection of OAuth 2.0 permission scopes that the web API (resource) app exposes to client apps. These permission scopes may be granted to client apps during consent.
Incorrect Answers:
oauth2AllowImplicitFlow. oauth2AllowImplicitFlow specifies whether this web app can request OAuth2.0 implicit flow access

tokens. The default is false. This flag is used for browser-based apps, like Javascript single-page apps.
Reference:
https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-fed-group-claims

Question #24 *Topic 3*

You develop an app that allows users to upload photos and videos to Azure storage. The app uses a storage REST API call to upload the media to a blob storage account named Account1. You have blob storage containers named Container1 and Container2.
Uploading of videos occurs on an irregular basis.
You need to copy specific blobs from Container1 to Container2 when a new video is uploaded.
What should you do?

A. Copy blobs to Container2 by using the Put Blob operation of the Blob Service REST API

B. Create an Event Grid topic that uses the Start-AzureStorageBlobCopy cmdlet

C. Use AzCopy with the Snapshot switch to copy blobs to Container2

D. Download the blob to a virtual machine and then upload the blob to Container2

Correct Answer: *B*
The Start-AzureStorageBlobCopy cmdlet starts to copy a blob.

Example 1: Copy a named blob -
C:\PS>Start-AzureStorageBlobCopy -SrcBlob "ContosoPlanning2015" -DestContainer "ContosoArchives" -SrcContainer "ContosoUploads"
This command starts the copy operation of the blob named ContosoPlanning2015 from the container named ContosoUploads to the container named
ContosoArchives.
Reference:
https://docs.microsoft.com/en-us/powershell/module/azure.storage/start-azurestorageblobcopy?view=azurermps-6.13.0

Question #25 *Topic 3*

You are developing an ASP.NET Core website that uses Azure FrontDoor. The website is used to build custom weather data sets for researchers. Data sets are downloaded by users as Comma Separated Value (CSV) files. The data is refreshed every 10 hours.
Specific files must be purged from the FrontDoor cache based upon Response Header values.
You need to purge individual assets from the Front Door cache.
Which type of cache purge should you use?

A. single path

B. wildcard

C. root domain

Correct Answer: *A*
These formats are supported in the lists of paths to purge:
☞ Single path purge: Purge individual assets by specifying the full path of the asset (without the protocol and domain), with the file extension, for example, /
[1]
☞ Wildcard purge: Asterisk (*) may be used as a wildcard. Purge all folders, subfolders, and files under an endpoint with /* in the path or purge all subfolders and files under a specific folder by specifying the folder followed by /*, for example, /pictures/*.
☞ Root domain purge: Purge the root of the endpoint with "/" in the path.
Reference:
https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching

Question #26 *Topic 3*

Your company is developing an Azure API.
You need to implement authentication for the Azure API. You have the following requirements:
All API calls must be secure.
▪
☞ Callers to the API must not send credentials to the API.
Which authentication mechanism should you use?

A. Basic

B. Anonymous

C. Managed identity

D. Client certificate

Correct Answer: *C*
Use the authentication-managed-identity policy to authenticate with a backend service using the managed identity of the API Management service. This policy essentially uses the managed identity to obtain an access token from Azure Active Directory for accessing the specified resource. After successfully obtaining the token, the policy will set the value of the token in the Authorization header using the Bearer scheme.
Reference:
https://docs.microsoft.com/bs-cyrl-ba/azure/api-management/api-management-authentication-policies

Previous Questions Next Questions