Microsoft

Designing Microsoft Azure Infrastructure Solutions

Exam Ref AZ-305

Ashish Agrawal
Gurvinder Singh
Avinash Bhavsar
Mohamed Sabir Sopariwala

Microsoft

Exam Ref AZ-305 Designing Microsoft Azure Infrastructure Solutions

Ashish Agrawal
Gurvinder Singh
Avinash Bhavsar
Mohamed Sabir Sopariwala

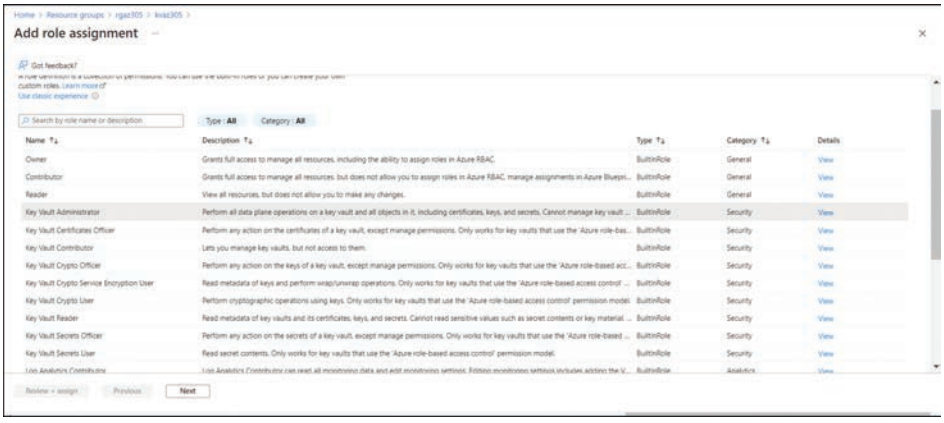

FIGURE 1-22 Role assignment for Azure Key Vault

An access policy helps provide authorization access to the data plane. Authorization can be done using Azure RBAC or a Key Vault access policy. It allows you to enable access of Azure services on Azure Key Vault and to specify a permission model for data-plane authorization.

Access can be enabled for Azure services as follows (see Figure 1-23):

- **Azure VM for deployment** VMs can retrieve certificates from secrets in a Key Vault.
- **Azure Resource Manager for template deployment** Azure Resource Manager can retrieve secrets from a Key Vault while deploying a template.
- **Azure Disk Encryption for volume encryption** The Azure Disk Encryption service can retrieve a key from a Key Vault and unwrap it, as required to encrypt disks.

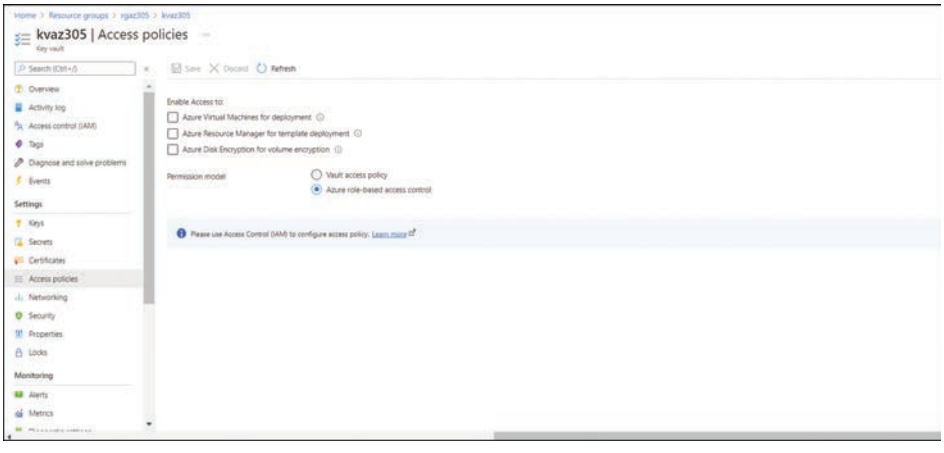

FIGURE 1-23 Access policies for Azure Key Vault

You can also configure vault access policies. A vault access policy is an alternative to Azure RBAC to provide permission on the Key Vault data plane. Vault access policies have a number of permission templates, as shown in Figure 1-24.

FIGURE 1-24 Permission templates for assigning permissions while configuring the Key Vault access policy

Each permission template provides a specific set of permissions for keys, secrets, and certificates. For example, selecting the Key Management permission template provides key management operations permissions and rotation policy operations permissions, as described in Table 1-3.

TABLE 1-3 Key Management permission template permissions

Key Management Operations Permissions	Rotation Policy Operations Permissions
Get	Rotate
List	Get Rotation Policy
Update	Set Rotation Policy
Create	
Import	
Delete	
Recover	
Backup	
Restore	

NOTE **Although permission templates are available, their use is optional. If you prefer, you can set permissions individually for keys, secrets, and certificates.**

NOTE **For new deployments, it is recommended to use the Azure RBAC model for data-plane operation authorization.**

MORE INFO **KEY VAULT ACCESS POLICY**

To see the step-by-step procedure for assigning Key Vault access policies, see the Microsoft documentation at *https://learn.microsoft.com/en-us/azure/key-vault/general/assign-access-policy?tabs=azure-portal*.

Azure Key Vault provides two types of storage for cryptographic keys: vault and managed hardware security module (HSM). Table 1-4 compares these two types of storage.

TABLE 1-4 Types of storage for cryptographic keys

Vault	Managed HSM
Supports software-protected keys and HSM-protected keys. HSM protection is available only in the Azure Key Vault Premium SKU.	Supports only HSM-protected keys.
Multitenant	Single tenant
Software-protected key: FIPS 140-2 Level 1 HSM-protected key: FIPS 140-2 Level 2	FIPS 140-2 Level 3
Used for low cost. Can be used where compliance requirements are less than FIPS 140-2 Level 3.	Used for high-value keys. Used when there is a specific requirement for FIPS 140-2 Level 3 compliance.

Recommend a solution for integrating applications into Azure Active Directory (Azure AD)

Applications, whether deployed in Azure, on-premises, on the edge, or in another public cloud, can rely on AAD for authenticating the users. You will learn in this section how to integrate applications with AAD for authenticating the users.

Application registration

We touched on application registration in the context of IDAM and the service principal earlier in this chapter. We will continue to discuss application registration in the context of the topic of this section.

Application developers can offload identity management and authentication functions to AAD. This requires the registration of the application in AAD. Registering an application creates a globally unique application object in the home tenant where the application is registered.

While registering an application in AAD, a developer can provide application access to the following (see Figure 1-25):

- Users belonging to the application's home AAD tenant
- Users belonging to any AAD tenant of any organization
- Users with a Microsoft account

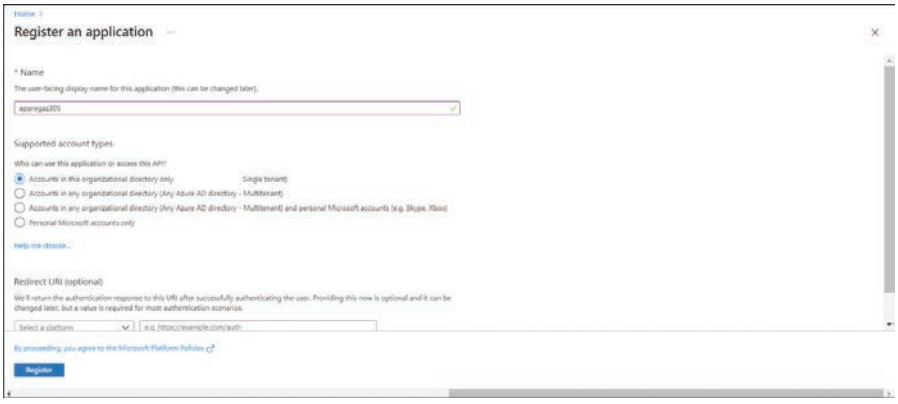

FIGURE 1-25 Application registration

Developers writing line of business (LOB) applications can use the single-tenant option, as only users within a specific organization are expected to use the application. If an application is meant for a B2B or B2C scenario, the multitenant option or the multitenant and personal Microsoft accounts option can be used. With a multitenant option, users with an account in another organization's AAD can be authenticated to consume the application. (You will learn more about this option in the next section.)

As discussed, when an application is registered through the Azure Portal, the application service principal (service principal object) is also created in the home AAD tenant. If, however, an application is registered through Microsoft Graph API (AAD API is part of Microsoft Graph API), the application service principal must be created separately.

Once an application is registered in AAD, various configuration options are made available for it. Some important configuration options are as follows:

- **Branding and organization properties** Use these options to supply a logo, home page URL, terms of service page URL, privacy statement page URL, the domain that users see on the consent screen, and so on.
- **Authentication** These settings enable you to specify additional settings based on the platform or device this application registration is targeting. You can also specify the logout URL and, importantly, the token that will be issued when the request is successfully authenticated. This can be an access token, an ID token, or both. In the case of an implicit grant flow, in which a single page application or web API is consumed by JavaScript, both an access token and an ID token can be sent as a response upon successful authentication. If the application is an ASP.net Core web app or any other web app using a hybrid authentication flow, only an ID token can be sent.

MORE INFO **UNDERSTAND IMPLICIT GRANT FLOW**

For more information about OAuth 2.0 implicit grant flow, see the Microsoft documentation at *https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-implicit-grant-flow?WT.mc_id=Portal-Microsoft_AAD_RegisteredApps*.

- **Certificates and secrets** You can configure these settings to provide a higher level of assurance that the authentication request or the request to retrieve the tokens is from an authentic source.
- **Token configuration** These settings allow you to include additional claims in the token returned in response to a successful authentication request.
- **Expose API** A developer can use this setting to integrate their REST APIs with AAD to enable authorized users or client applications to access these APIs with delegated permissions. Multiple scopes can be defined for the API such that each one can be configured to require consent from the admin, the user, or both.
- **API permission** Use this setting to give the client app being registered access to other APIs, such as Microsoft APIs or any APIs within an organization exposed through the Expose API option in a separate application registration in AAD.

Enterprise applications

An application service principal is essentially a local representation of the globally unique application object. The service principal inherits certain configurations and properties from the global application object. In a sense, the application object serves as a template for the application's service principal object. Service principals of all the applications registered in a particular AAD tenant are available as Enterprise applications within the same AAD tenant.

Enterprises can integrate SaaS applications from the AAD application gallery, nongallery applications, and on-premises applications. Integration of on-premises applications requires the deployment of an Application Proxy connector in the on-premises environment on the Windows server. Integrating applications in AAD as enterprise applications creates a service principal of globally exposed SaaS applications (whether from the AAD application gallery or not) or an on-premises application.

Once the service principal is created for an application, the following settings can be applied to it:

- You can assign users and groups to access the application.
- You can configure Single Sign-On (SSO) for applications integrated from the AAD application gallery and even for nongallery applications. Note that SSO is not available for applications registered by the developer using the application-registration approach discussed in the preceding section. Developers can use OpenID Connect and OAuth to provide SSO features, however. SDKs are available for a number of programming languages as part of the Microsoft Authentication Library (MSAL) to easily enable the use of the Microsoft Graph API.
- You can configure automatic provisioning of accounts for all registered applications, whether they are gallery applications, nongallery applications, or on-premises applications. You can manage the identity lifecycle in AAD; the lifecycle of user accounts is automatically managed in the application. Note that this feature is not available for applications registered by the developer using the application-registration approach described in the preceding section.

- You can configure conditional access policies for registered applications, in a manner similar to the one described earlier in this chapter.
- You can enable self-service access requests for enterprise applications.

Recommend a user consent solution for applications

An application or service that tries to access organizational data or services should not be able to do so without proper consent in place. Applications and services should obtain consent in one of the following ways:

- By asking a user to use their identity to access organization data or services
- By having the administrator provide consent to the application on behalf of all users

Enterprise applications allow for consent-related configuration at the AAD tenant level for an organization.

Global administrators can configure user consent (see Figure 1-26) and group owner consent (see Figure 1-27) at the AAD tenant level for applications accessing organization data.

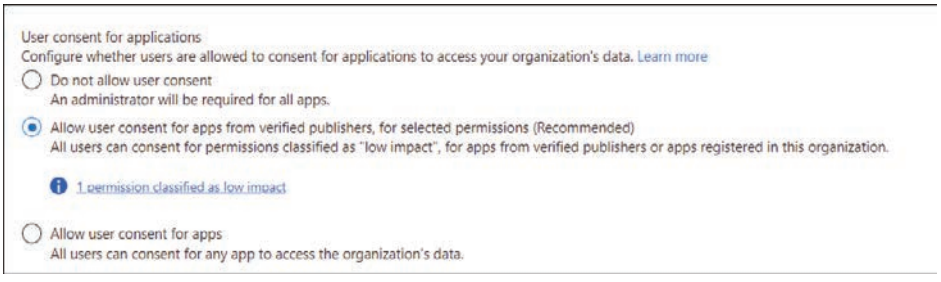

FIGURE 1-26 User consent for application configuration

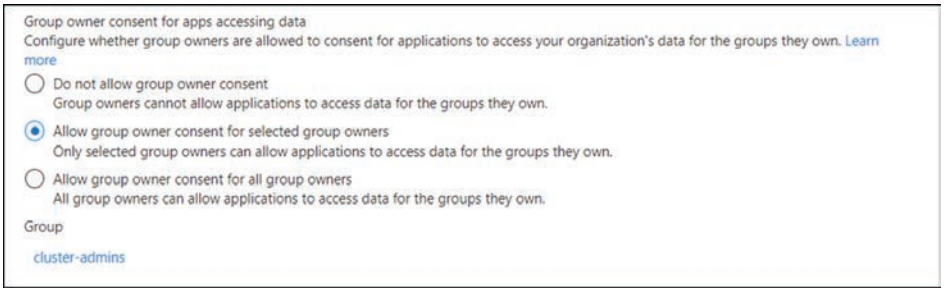

FIGURE 1-27 Group owner consent for application configuration

Global administrators, application administrators, and cloud application administrators can configure permission classifications for user consent for enterprise applications. Presently, only low-risk permission classification can be done for permissions. Permissions that do not require admin consent can be added to this classification. Examples of low-risk permissions include the following: