Accessing Azure Resources Securely

We need to ensure that resources are protected from unintended exposure. We need to have full confidence in our processes that they expose information on need-to-know basis only.

Security is a big topic and many books have been written on it. Last time I counted there were at least 30 dimensions of security. But, in this context, we’ll only cover how best to access Azure resources securely.

To accomplish this, we need to identify what resources need to be protected; who can access them and whether the requestor has proper permissions. This is done via authentication (requestor is known) and authorization (allowed to access) the said resource.

There are several ways to authenticate and get authorization to access Azure resources. For example, user id, password, service principal, system-assigned Managed Identity and user-assigned managed identity.

It is important to understand the context and the nuances of the selected method to ensure that privacy and protection is fully maintained. And its long-term viability is understood. For example, system-assigned managed identity will die as soon as the resource for which it created is deleted.

One simple rule of thumb, if your resource is client facing (interacting with users), user id and password will be fine. But if it is an azure resource to resource communication, alternatives such as Service Principal and Managed Identity are more appropriate and should be considered.

The use cases for user id, password are pretty clear. However, you will have to understand the context, capabilities and nuances of service principal, user-assigned managed identity and system-assigned managed identity to be able to make the right choice.

On Azure platform, managed identities eliminate the need for developers having to manage credentials by providing an identity for the Azure resource in Azure AD and using it to obtain Azure Active Directory (Azure AD) tokens.

Benefits of using Managed identities:

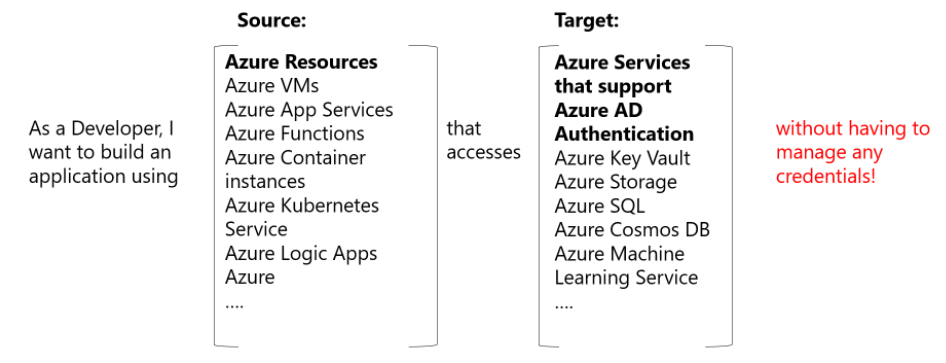
* You don't need to manage credentials.
* You can use managed identities to authenticate to any Azure service that supports Azure AD authentication including Azure Key Vault.
* A managed identity is managed by the Azure platform and does not require you to provision or rotate any secrets.
* Managed identities can be used **without any additional cost**.

Types of managed identities:

* System-assigned - You enable a managed identity directly on a service instance. When you enable a system-assigned managed identity, an identity is created in Azure AD that is **tied to the lifecycle of that service instance**. When the resource is deleted, Azure automatically deletes the identity. By design, **only that Azure resource can use this identity** to request tokens from Azure AD.
* User-assigned - User may also create a managed identity as a standalone Azure resource. You can [create a user-assigned managed identity](https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/how-to-manage-ua-identity-portal) and assign it to one or more instances of an Azure service. In the case of user-assigned managed identities, the **identity is managed independently of the resource that uses it**. A user-assigned managed identity can be used for multiple Azure services.

Differences between service principals and managed identities:

* The main difference between a managed identity and a service principal is that the managed identity manages the creation and automatic renewal of a service principal on your behalf.
* A service principal can be thought of as an application whose tokens can be used to authenticate and grant access to specific Azure resources from a user-app, service or automation tool. A service principal is the local representation, or application instance, of a global application object in a single tenant or directory.
* The service principal object defines what the app can actually do in the specific tenant, who can access the app, and what resources the app can access.
* It enables the principle of "least privilege" and accountability.
* A managed identity is a layer on top of a service principal, removing the need for you to manually create and manage service principals directly.
* Azure Service principals password/certificate expire after 1-2 years (defaults to 1 year, can be increased to 2 years).
* System-assigned managed identity is only active/alive as long as the Azure resource instance that created it lives. Once the creating instance is deleted, it is automatically deleted as well.
* User-assigned managed identity – has an independent lifecycle. You have to manually clean it up, when required.
* **Managed identities are now the preferred** approach over service principals to managing identities for apps and automation access.



Use cases where managed identities can be used

For a comprehensive list of Azure services supporting managed identities, please refer to: [Azure Services that support managed identities - Azure AD | Microsoft Docs](https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/services-support-managed-identities) article.

**Examples of managed identity usage:**

* Service Principal - [Access Secrets from Azure Key Vault using Service Principal.docx](Access%20Secrets%20from%20Azure%20Key%20Vault%20using%20Service%20Principal.docx)
* System-assigned managed identity - [Access Azure Blob Storage from Azure App Service using System Assigned Managed Identity.docx](Access%20Azure%20Blob%20Storage%20from%20Azure%20App%20Service%20using%20System%20Assigned%20Managed%20Identity.docx)
* User-assigned managed identity - [Access Azure Blob Storage from Azure App Service using User Assigned Managed Identity.docx](Access%20Azure%20Blob%20Storage%20from%20Azure%20App%20Service%20using%20User%20Assigned%20Managed%20Identity.docx)

**Note:** The examples presented herein are for demonstration purpose and are not limited to the resource types used in these examples. They can be used for a wide array of resource types.