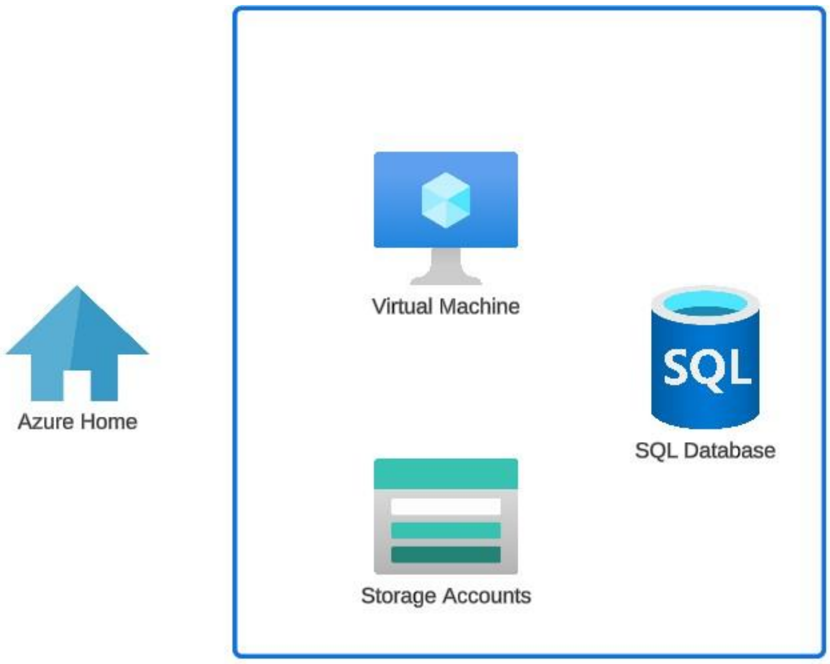
Implement Azure security

What is Microsoft Entra ID



So far we have been working with Azure resources with our Azure Admin Account.

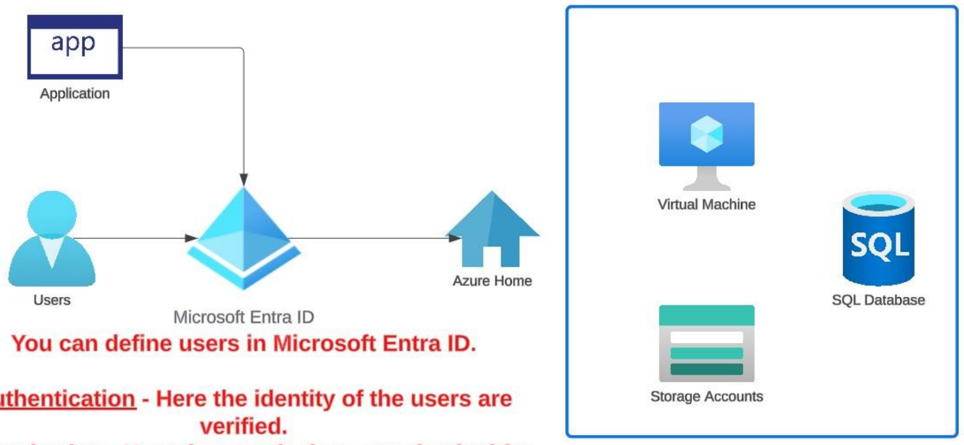
But in an organization, you want to have users who can access and manage resources.

Who has permission to create resources. Who has permission to access resources.

We need to create users and be able to assign permissions.

Microsoft Entra ID - This is a cloud-based identity and access management service. This identity service can be used for Azure, Microsoft 365 and even other Software-as-a-service applications.

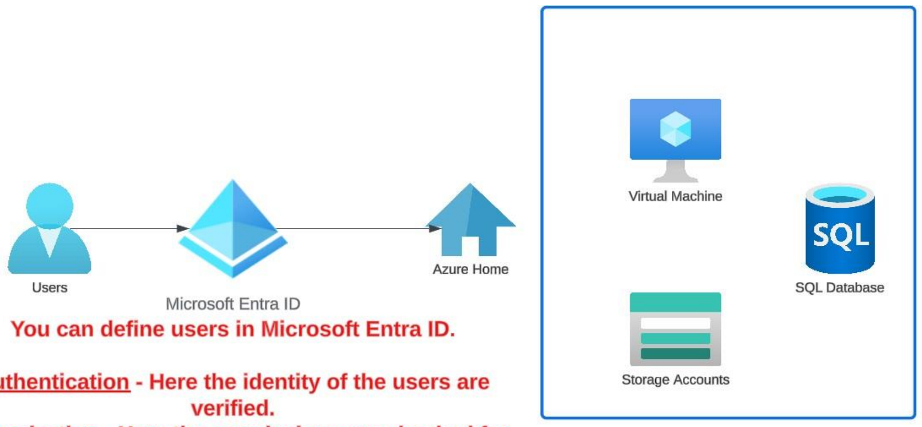
Even Applications can be linked to identities and be given access accordingly.



You can define users in Microsoft Entra ID.

Authentication - Here the identity of the users are verified.
Authorization - Here the permissions are checked for the users.

Lab - Role-based access control



You can define users in Microsoft Entra ID.

Authentication - Here the identity of the users are verified.
Authorization - Here the permissions are checked for the users.

Role

Role-based access control

We can assign different roles to a user.

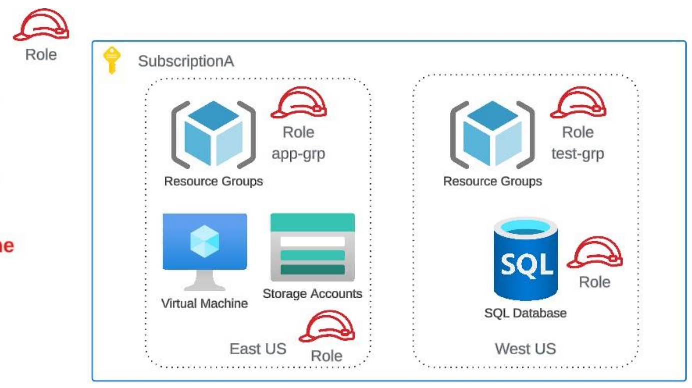
There are many in-built roles.

You can also define your own custom roles.

You can assign a role at the subscription level.

You can assign a role at the resource group level.

You can assign a role at the resource level.



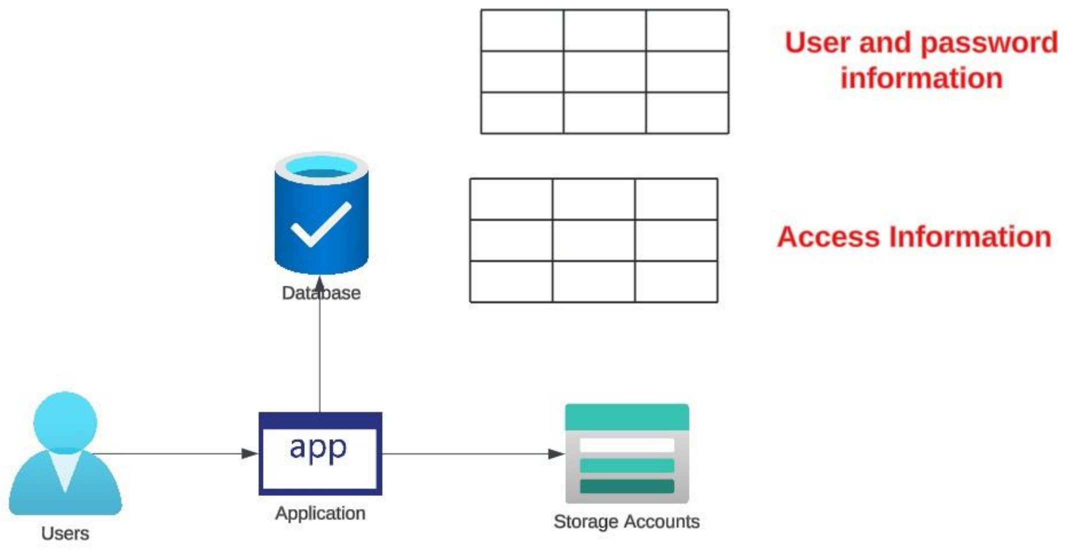
Owner Role
Here the user would have complete access and be able to manage the resources. The user can also delegate access to other users.

Contributor Role
Here the user would have complete access and be able to manage the resources.

User Access Administrator Role
Here the user would be able to delegate access to other users.

Reader Role
Here the user would be just be able to read the properties for the resources.

Introduction to Application Objects

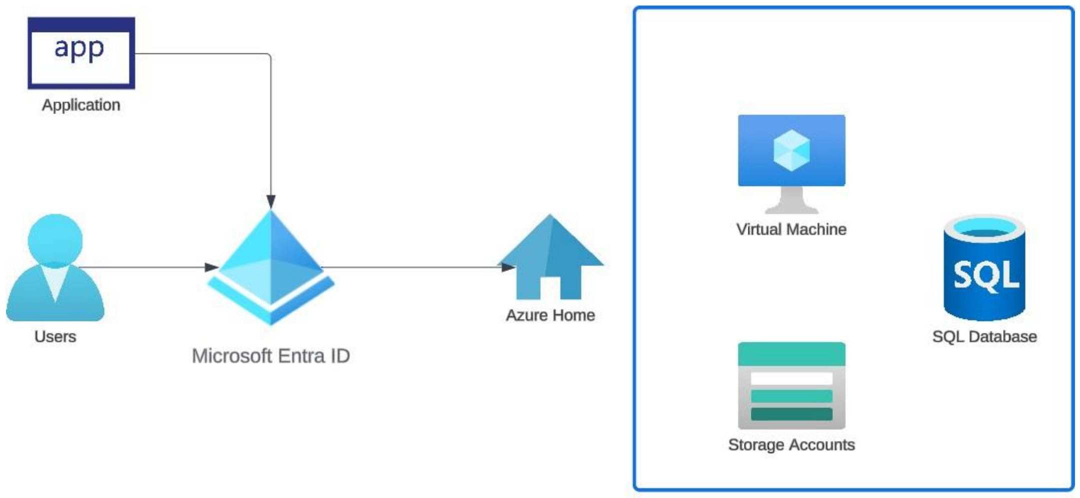


Consider an application that needs to authenticate users and authorize them to use resources on Azure.

The application would need to maintain a data store that has information for the different users and their access permissions.

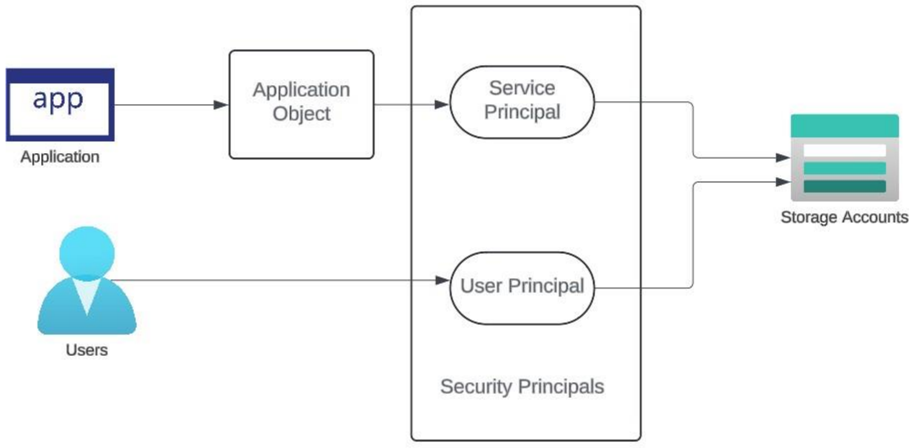
The company maintaining the application would need to maintain the database of credentials , their security and the required protocols used to authenticate and authorize users.

Instead we can make use of Microsoft Entra ID as the authentication and authorization provider. We delegate these tasks to Microsoft Entra ID.



In order for the Application to use Microsoft Entra ID, it needs to be registered in Entra ID. This is done by creating an Application Object.

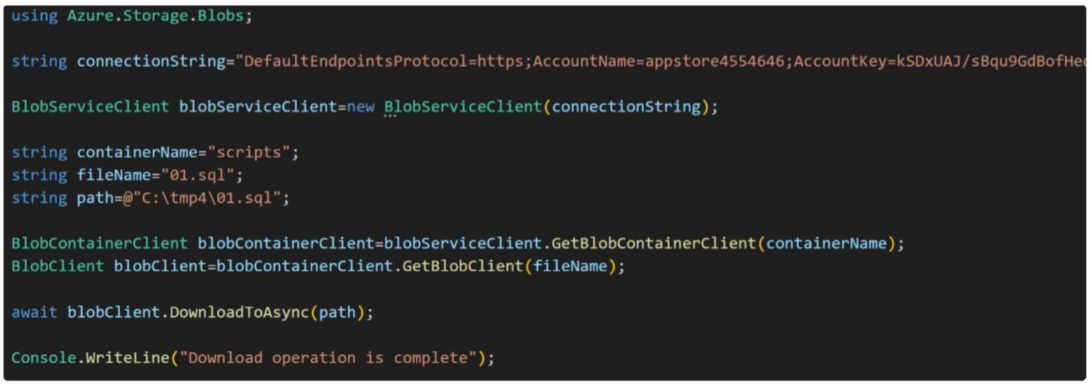
In order for the Application to use Microsoft Entra ID, it needs to be registered in Entra ID. This is done by creating an Application Object.



The Application Object is associated with a service principal. This principal is then given permissions to access resources.

Lab - Application Object - Blob objects

Revisit the program that was used to download a blob from a storage account in .NET



```
using Azure.Storage.Blobs;

string connectionString="DefaultEndpointsProtocol=https;AccountName=appstore4554646;AccountKey=kSDxUAJ/sBqu9GdBofHe

BlobServiceClient blobServiceClient=new BlobServiceClient(connectionString);

string containerName="scripts";
string fileName="01.sql";
string path=@"C:\tmp4\01.sql";

BlobContainerClient blobContainerClient=blobServiceClient.GetBlobContainerClient(containerName);
BlobClient blobClient=blobContainerClient.GetBlobClient(fileName);

await blobClient.DownloadToAsync(path);

Console.WriteLine("Download operation is complete");
```

At that point we used connection strings to connect to the storage account.

We now want to make use of an Application Object.

app
Application

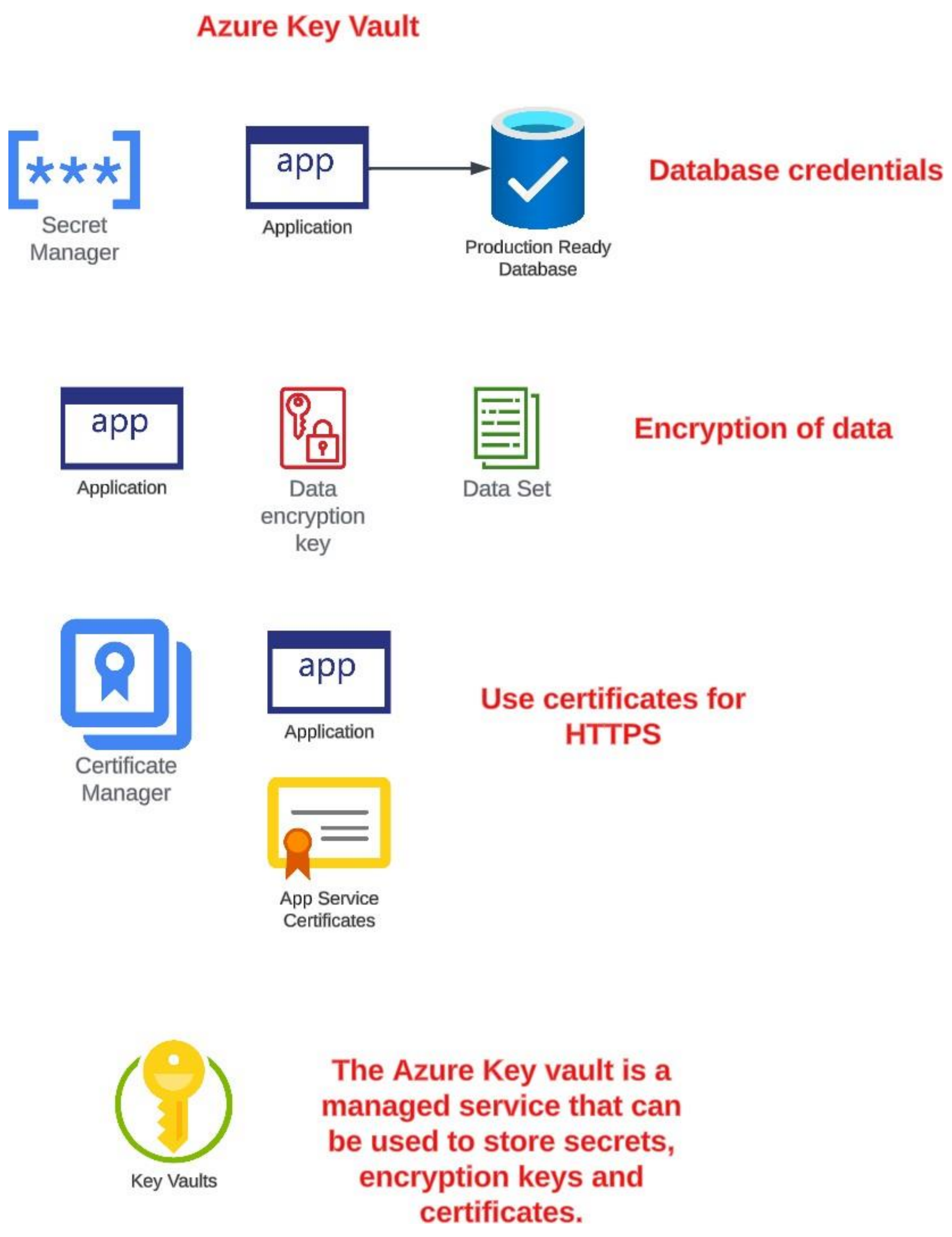
Microsoft Entra ID

In Microsoft Entra ID, we will first create an Application Object.

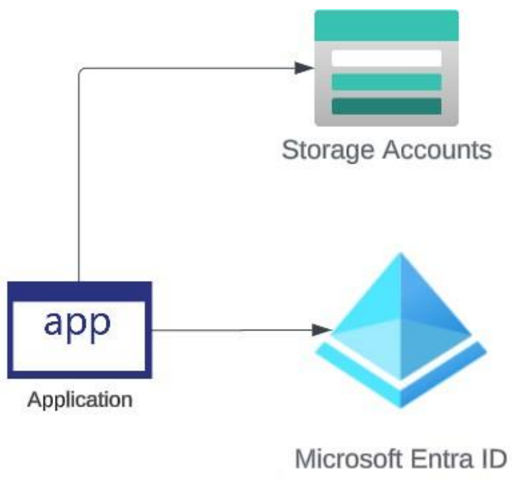
We wil give permissions for the Application Object to access our Storage Account - We will provide Role-based access to the service principal attached to the Application Ob

Then in our .NET code , we wil make the required changes to make use of the Application Object instead of the connection string.

Azure Key Vault



Managed Identities

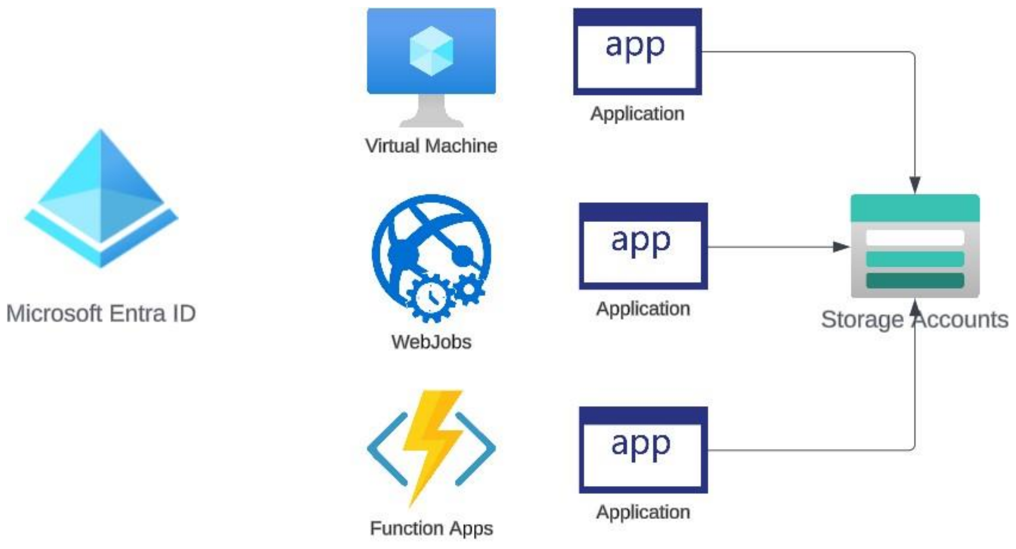


Application uses an Application Object to access an Azure Storage account

The application with the help of the in-built classes would get the required access tokens to access the storage account.

Even though we now make use of RBAC, we still need to embed the credentials of the Application Object in our code.

You can make use of Managed Identities. This gives a way for applications to authenticate to Azure resources without the need of embedding credentials.



Your application could be hosted on a service that supports managed identities.

The managed identity for the resource can be registered in Microsoft Entra ID. This would create a service principal for that resource.

You can then provide RBAC access for that service principal onto the resource. And in your code you don't embed any sort of credentials.

We will look into an example of having an application hosted on a Virtual Machine that is accessing the blob service.