Lab 3: Azure Managed Identity

Objective

1. Create a asp.net core webapp and publish it on Azure App Service
2. Change the code to connect to Azure Key Vault
3. Provision a managed identity
4. Give access control through Azure Key Vault

Note:

1. All the steps are to be done within VS Code and Azure Portal.
2. There will be breakout rooms assigned and each room will have a group number [1-5]
3. Login into Azure Portal
    a. Go to https://portal.azure.com
    b. Login with the supplied credentials (username and password).
        i. Each group has a unique integer for their login [1-5] eg. usergroup[1-5]
        ii. For example, group number 5 will have
            1. Username: usergroup5@makecloudwork.com
            2. Password: will be provided during the class.
    c. You will then see the landing Azure homepage. Dismiss any popups/message boxes

---

It's important that you enter all the resource names same as mentioned.

Section 1: Create a webapp and publish it on Azure

Steps

1. Open VS Code and install Azure extensions (Azure Account, Azure App Service, Azure Functions , Azure Resources, Azure CLI Tools, Azure Virtual Machines, Azure API Management)
2. Ensure you have .net 6 installed.

    dotnet --version

3. Create a new webapp through Terminal

    dotnet new webapp -n identitywebappsb[1-5]

    Add your group number as suffix e.g. if your group number is 4, name the app as **"identitywebappsb4"**

    cd identitywebappsb[1-5]

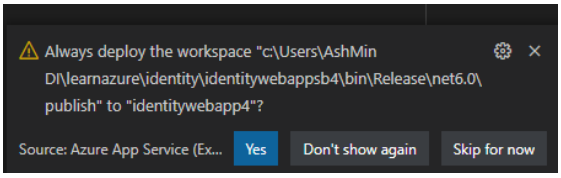
code . -r

dotnet build

dotnet run

Open web browser and follow the mentioned localhost link with the mentioned random port to see the welcome ASP.Net page.

4. On VS Code, select Azure extension. Under Resources, click on connect to Azure and provide your Azure username/password.
5. You will see the subscription **"Learn Azure"** and the App Service that you created in the last lab amongst other things.
6. Click on "Create Resource" next to the RESOURCES section and select "Create App Service Web App" from the drop down shown in the command palette.
7. Create new web app: Enter **"identitywebappsp[1-5]"**
8. Select a runtime stack: **".NET 6 (LTS)"**
9. Select a pricing tier: **"Basic (B1)"**
10. The process will start and will give warning that you do not have permission to create a resource group. Click on **"Select Existing"** and select the resource group shown in the command palette.
11. Give it few minutes and once finished, you may see a box asking whether you want to deploy. Click on "Deploy"
12. Deployment process will start. You may see a box sasking "Always deploy the workspace", click **"Yes"**



13. Once deployment is finished, you can click on **"Browse Website"**
14. Go back to Azure portal and under your existing resource group, you will see that 3 resources have be created. App Service Plan, App Service and Application Insights.

Section 2: Change the code to connect to Azure Key Vault

Steps

1. Go back to VS Code and go to explorer. We will now add code to access the secret under the Key Vault that we have created in the previous lab.
2. Run the following through Terminal

dotnet add package Azure.Security.KeyVault.Secrets

dotnet add package Azure.Identity

3. Expand the folder "Pages" and open "Index.cshtml.cs" file. Add the following code

On Top

```
using System;
using Azure.Identity;
using Azure.Security.KeyVault.Secrets;
```

Add 2 string variables

```
public string name { get; set; } = "SecretName";
public string value { get; set; } = "SecretValue";
```

Under function public void OnGet()

```
  string kvUri = Environment.GetEnvironmentVariable("KEY_VAULT_NAME");
   var client = new SecretClient(new Uri(kvUri), new DefaultAzureCredential());
   KeyVaultSecret kvs = client.GetSecret("secretname");
   name = kvs.Name;
   value = kvs.Value;
```

4. Open “Index.cshtml” file. Replace the entire code with the following code

```
@page
@model IndexModel
@{
   ViewData["Title"] = "Home page";
}

<div class="text-center">
        <h1 class="display-4">Welcome</h1>
        <p>@Model.name: @Model.value</p>
</div>
```

5. Open “Error.cshtml.cs” file. Add the following code

On Top

```
using Microsoft.AspNetCore.Diagnostics;
```

Add 1 string variable

```
public string? Msg {get; set;}
```

Under function public void OnGet()

```
RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier;
Msg =" This is a generic exception message";
var exceptionHandlerPathFeature =
HttpContext.Features.Get<IExceptionHandlerPathFeature>();
Msg = exceptionHandlerPathFeature?.Error.Message;
```

6. Open “Error.cshtml” file. Replace the entire code with the following code

```
@page
@model ErrorModel
@{
  ViewData["Title"] = "Error";
}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)
{
  <p>
    <strong>Request ID:</strong> <code>@Model.RequestId</code>
```

```
        </p>
        <p>
          <strong>Message :</strong> <code>@Model.Msg</code>

        </p>
    }
```

7. On VS Code, select Azure extension. Under Workspace, click on Deploy and "Deploy to Web App"
8. Select **"identitywebappsb[1-5]"** from the drop down in command palette.
9. Click on "Deploy" on the confirmation message box.
10. Once deployed, click on "Browse Website" and you will see an error. Note the error message

Section 3: Provision a managed identity

Steps

1. Login into Azure Portal
2. Type **"App service"** on the search bar and select **"App Services"** from the drop down. You will be redirected to **"App Services"** page. Select the newly created **"identitywebappsb[1-5]"** app service.
3. Within your App Service, under Settings, click on **"Identity"**
    a. Under System assigned, change the Status to **"On"** and click on **"Save".**
    b. Click **"Yes"** on the confirmation box.
4. Within your App Service, under Settings, click on **"Configuration"**
    a. Click on **"New application setting"**
        i. Name: **"KEY_VAULT_NAME"**
        ii. Value: Enter the Vault URI **"http://<keyvaultname>.vault.azure.net"** You will find the name under the overview section of your key vault
    b. Select "OK" and click on "Save".
    c. Open the webapp on the browser and you will get a different error this time.

Section 4: Give access control through Azure Key Vault

Steps

1. Login into Azure Portal

2. Type **“Key Vault”** on the search bar and select **“Key Vaults”** from the drop down. You will be redirected to **“Key Vaults”** page.
3. Select your Key Vault
4. Within your Key Vault, under Settings, click on **“Access policies”**
5. Click on **“+Add Access Policy”**
    a. Secret permissions: **Select All**
    b. Select principal: Click on “None selected”
        i. Start typing *“identity..”* and you will see the name of your App Service along with its Managed identity id.
        ii. Select the name and click on **“Select”**
    c. Click on **“Add”**

6. Click on **“Save”.**
7. Open Browser and enter the URL for the Web App. You will see both the name and value of the secret

End of Lab.