**Managing resources in Azure: Project solution exemplar**

****Important disclaimer for the Azure UI****

Given the dynamic nature of Microsoft Cloud tools, you might experience changes in the Azure UI that have taken place after the development of this training content. As a result, the screenshots in the videos, readings, exercises, or project solution exemplars might not align exactly with how you currently experience the UI. However, please note that these changes do not impact the functionalities of the UI. Hence, you can still perform all the steps shown in the videos, readings, exercises, or project solution exemplars.

**Task: Managing resources in Azure**

|  |  |
| --- | --- |
| **Activity #** | **Steps** |
| **Activity 1:**  Create Azure AD App registration and implement Azure Authentication for a web application. | Step 1: Sign in to the Azure portal and create an Azure AD App registration. |
|  | Step 2: After the creation of Azure AD App registration, enable the authentication for your web app by going to the respective **Resource groups** and click Authentication. Then go to **Add an identity provider**.    Step 3: Verify the access to the web app. |
| **Activity 2:**  Implement RBAC for a web application. | Step 1: Browse to **Identity** > **Applications** > **App registrations,** and then select the application in which you want to define app roles.  Step 2: Under the **Manage** tab, select **App roles**, and then click **Create app role**.    Step 3: In the **Create app role** window, enter the settings for the role.    Step 4: Click **Apply** to save the changes. |
| **Activity 3:**  Set up authentication for your application using Azure Active Directory (AAD) and assign roles to the user. | Step 1: Create an Azure AD tenant on the **Create a tenant** window.    Step 2: In the newly created **Azure Active Directory (AAD)**, create one or more users for testing.    Step 3: Provide the information of the new user.    Step 4: In the search box,searchfor the scope or groups to which you want to grant access to the user. For example, **Management groups**, **Subscriptions**, **Resource groups**, or a specific resource.    Step 5: Click **Access control (IAM)**.    Step 6: Click the **Role assignments** tab to view the role assignments at this scope.  Step 7: Click **Add** > **Add role assignment**.    Step 8: Select the appropriate role.    Step 9: If you want to assign a privileged administrator role, select the **Privileged administrator roles** tab to select the role.    Step 10: In the **Details** column, click **View** to get more details about a role.    Step 11: On the **Members** tab, select **User, group,** or **Service Principal** to assign the selected role to one or more Microsoft Entra users, groups, or service principals (applications).    Step 12: On the **Review + assign** tab, review the role assignment settings.    Step 13: Click **Review + assign** to assign the role. |
| **Activity 4:**  Implement RBAC to control access based on user roles by deploying Azure App Service. | Step 1: Go to the management page for your **Azure App Service**.  Step 2: On the left pane, select **Deployment Center,** and then click **Settings**.  Step 3: In the **Source** box, select one of the **Continuous Deployment (CI/CD)** options.    Step 4: [GitHub Actions](https://learn.microsoft.com/en-us/azure/app-service/deploy-continuous-deployment%22%20/l%20%22how-does-the-github-actions-build-provider-work) is the default build provider. To change the provider, select **Change provider** > **App Service Build Service** (Kudu) > **OK**.  Step 5: If you’re deploying from GitHub for the first time, select **Authorize** and follow the authorization prompts. If you want to deploy from a different user’s repository, select **Change Account**.  Step 6: After authorizing your Azure account with GitHub, select the **Organization**, **Repository**, and **Branch** to configure the CI/CD.  Step 7: If you can’t find an organization or repository, you might need to enable more permissions on GitHub.  Step 8: In the **Preview** dialog box that opens, under **Authentication type**, select **User-assigned identity** for better security.  Step 9: When **GitHub Actions** is selected as the build provider, you can select the workflow file you want by using the **Runtime stack** and **Version** dropdown lists. Azure commits this workflow file into your selected GitHub repository to handle build and deploy tasks. To see the file before saving your changes, select **Preview file**.  Step 10: Select **Save**.  Step 11: New commits in the selected repository and branch now deploy continuously into your App Service app. You can track the commits and deployments on the **Logs** tab. |
| **Activity 5:**  Test whether the roles are assigned correctly to the users or not. | Step 1: Click **Access control (IAM)**.    Step 2: On the **Check access** tab, click the **Check access** button.  Step 3: In the **Check access** pane, click **User, group, or service principal**.  Step 4: In the search box, enter a string to search the directory for display names, email addresses, or object identifiers.    Step 5: Click the user to open the **assignments** pane.    Step 6: On the **Check access** tab, click **View my access**.    All the assigned roles and user list can be checked and verified from this tab. |
|  |  |