5.5 Azure Storage Services



This section will guide you to:

* Create an Azure Storage account and manage the storage using Azure Storage Explorer

**Development Environment:**

* Windows 10
* Windows Azure cloud

This guide has four subsections, namely:

5.5.1 Creating an Azure Storage account

5.5.2 Managing the storage using Azure Storage Explorer

5.5.3 Securing the storage using SAS

5.5.4 Pushing the code to GitHub repositories

**Step 5.5.1:** Creating an Azure Storage account

* An Azure account has been enabled in your practice lab. Refer DotNet Lab guide: Phase 4 to learn how to use the practice lab.
* Login to the Azure Portal with the Azure/Microsoft credentials provided in the lab.
* From the portal dashboard, click on **More Services.**
* In the next page click **Storage Accounts.**
* On the Storage Accounts page, click **Add** at the top.
* For **Resource Group,** either select an existing one or click **Create New.**
* For **Storage Account Name** choose a globally unique name.
* Choose **Locally-redundant storage(LRS)** in the **Replication** dropdown.
* Click **Next:Networking.**
* Click **Next:Advanced.**
* Click **Next:Tags.**
* Click **Next:Review+Create.**
* Click **Create** after validation tests are passed.
* Wait till deployment is complete.
* Click on **Go to Resource** to view the storage.
* From the left bar, click on **Access Keys.** This will display the keys you need for accessing the storage using Storage Explorer.

**Step 5.5.2:** Managing the storage using Azure Storage Explorer

* Go to <https://azure.microsoft.com/en-in/features/storage-explorer/> to download the Azure Storage Explorer. Choose Windows as the **Operating System.**
* Double click the downloaded installer and complete the installation process.
* Start the Azure Storage Explorer application.
* From the left bar, right click **Storage Accounts** and click **Connect to Azure Storage.**
* From the displayed options, choose **Use a Storage Account name and key.**
* In the next screen, select **Use a Storage Account name and key.**
* For **Display Name,** use the same name as the Storage account.
* For **Account Name,** use the same name as the Storage account.
* For **Account Key,** copy the value in **Key1 Key** in the Azure Storage account screen.
* Click **Next.**
* In the next screen click **Connect.**
* This will display the contents of the storage account in the left bar of the Storage Explorer.

**Step 5.5.3:** Securing the storage using SAS

* On the Storage Accounts page, click the storage account you want to add SAS.
* In the Storage Account overview page, look for **Shared Access Signature** in the left bar and click on it.
* You can change the settings as per your requirement and click **Generate SAS and connection string.**
* This will generate and display the URLs connection string and SAS token for the storage account.
* To test the SAS access, open Azure Storage Explorer.
* From the left bar, right click **Storage Accounts** and click **Connect to Azure Storage.**
* From the displayed options, choose **Use a Shared Access Signature (SAS) URI.**
* Click **Next.**
* For **Display Name** choose a name of your choice.
* In **URI,** paste the value from the **Blob Service SAS Url** in the overview page.
* Click **Next.**
* In the **Summary** screen, click **Connect.**
* The new connection should be successfully listed in the left bar now.

**Step 5.5.4:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master