Implementing and Managing Storage

Lab: Implementing Azure Storage access controls

Scenario

Adatum Corporation wants to protect content residing in Azure Storage

Objectives

After completing this lab, you will be able to:

- Create an Azure Storage account.
- Upload data to Azure Storage.
- Implement Azure Storage access controls

Lab Setup

Estimated Time: 30 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Creating and configuring an Azure Storage account

The main tasks for this exercise are as follows:

1. Create a storage account in Azure
2. View the properties of the storage account

Task 1: Create a storage account in Azure

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. From Azure Portal, create a new storage account with the following settings:
    - Subscription: the name of the target Azure subscription
    - Resource group: a new resource group named **az3000201-LabRG**
    - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
    - Location: the name of the Azure region that is available in your subscription and which is closest to the lab location
    - Performance: **Standard**
    - Account kind: **Storage (general purpose v1)**
    - Replication: **Locally-redundant storage (LRS)**
    - Secure transfer required: **Disabled**
    - Network connectivity: **Public endpoint (All networks)**
    - Blob soft delete: **Disabled**
    - Data Lake Storage Gen2: **Disabled**

3. Wait for the storage account to be provisioned. This will take about a minute.

Task 2: View the properties of the storage account

1. In Azure Portal, with your storage account blade open, review the **Overview** section, including the location, replication, and performance settings.
2. Display the **Access keys** blade. On the access keys blade, note that you have the option of copying the values of storage account names including key1 and key2. You also have the ability to regenerate both keys.
3. Display the **Configuration** blade.
4. On the **Configuration** blade, notice that you have the option of performing an upgrade to **General Purpose v2** account and changing the replication settings. However, you cannot change the performance setting (this can only be assigned when the storage account is created).

> Result: After you completed this exercise, you have created your Azure Storage and examined its properties.

Exercise 2: Creating and managing blobs

The main tasks for this exercise are as follows:

1. Create a container
2. Upload data to the container by using the Azure portal
3. Access content of Azure Storage account by using a SAS token

Task 1: Create a container

1. In the Azure portal, navigate to the blade displaying the properties of the storage account you created in the previous task.
2. From the storage account blade, create a new blob container with the following settings:
    - Name: **labcontainer**
    - Access type: **Private**

Task 2: Upload data to the container by using the Azure portal

1. In the Azure portal, navigate to the **labcontainer** blade.
2. From the **labcontainer** blade, upload the file: **C:\Windows\ImmersiveControlPanel\images\splashscreen.contrast-white_scale-400.png**.

Task 3: Access content of Azure Storage account by using a SAS token

1. From the **labcontainer** blade, identify the URL of the newly uploaded blob.
2. Start Microsoft Edge and navigate to that URL.
3. Note the **ResourceNotFound** error message. This is expected since the blob is residing in a private container, which requires authenticated access.
4. Switch to the Microsoft Edge window displaying the Azure portal and, on the **splashscreen.contrast-white_scale-400.png** blade, switch to the **Generate SAS** tab.

5. On the **Generate SAS** tab, enable the **HTTP** option and generate blob SAS token and the corresponding URL.
6. Open a new Microsoft Edge window and, in the navigate to the URL generated in the previous step.
7. Note that you can view the image. This is expected since this time you are authorized to access the blob based on the SAS token included in the URL.
8. Close the Microsoft Edge window displaying the image.

Task 4: Access content of Azure Storage account by using a SAS token and a stored access policy.

1. In the Azure portal, navigate to the **labcontainer** blade.
2. From the **labcontainer** blade, navigate to the **labcontainer - Access policy** blade.
3. Add a new policy with the following settings:
    - Identifier: **labcontainer-read**
    - Permissions: **Read**
    - Start time: current date and time
    - Expiry time: current date and time + 24 hours
4. In the Azure portal, in the Microsoft Edge window, start a **PowerShell** session within the **Cloud Shell**.
5. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:
    - Subsciption: the name of the target Azure subscription
    - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location
    - Resource group: **az3000201-LabRG**
    - Storage account: a name of a new storage account
    - File share: a name of a new file share
6. From the Cloud Shell pane, run the following to identify the storage account resource you created in the first exercise of this lab and store it in a variable:

```
$storageAccount = (Get-AzStorageAccount -ResourceGroupName az3000201-LabRG)
[0]
```

7. From the Cloud Shell pane, run the following to establish security context granting full control to the storage account:

```
$keyContext = $storageAccount.Context
```

8. From the Cloud Shell pane, run the following to create a blob-specific SAS token based on the access policy you created in the previous task:

```
$sasToken = New-AzStorageBlobSASToken -Container 'labcontainer' -Blob
'splashscreen.contrast-white_scale-400.png' -Policy labcontainer-read -
Context $keyContext
```

9. From the Cloud Shell pane, run the following to establish security context based on the newly created SAS token:

```
$sasContext = New-AzStorageContext $storageAccount.StorageAccountName -
SasToken $sasToken
```

10. From the Cloud Shell pane, run the following to retrieve properties of the blob:

```
Get-AzStorageBlob -Container 'labcontainer' -Blob 'splashscreen.contrast-white_scale-400.png' -Context $sasContext
```

11. Verify that you successfully accessed the blob.
12. Minimize the Cloud Shell pane.

Task 5: Invalidate a SAS token by modifying its access policy.

1. In the Azure portal, navigate to the **labcontainer - Access policy** blade.
2. Edit the existing policy **labcontainer-read** by setting its start and expiry time to yesterday's date.
3. Reopen the Cloud Shell pane.
4. From the Cloud Shell pane, re-run the following to attempt retrieving properties of the blob:

```
Get-AzStorageBlob -Container 'labcontainer' -Blob 'splashscreen.contrast-white_scale-400.png' -Context $sasContext
```

5. Verify that you no longer can access the blob.

> Result: After you completed this exercise, you have created a blob container, uploaded a file into it, and tested access control by using a SAS token and a stored access policy.

Exercise 3: Remove lab resources

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. If needed, switch to the Bash shell session by using the drop down list in the upper left corner of the Cloud Shell pane.
3. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

```
az group list --query "[?starts_with(name,'az30002')]".name --output tsv
```

4. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name,'az30002')]".name --output tsv | xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

> Result: In this exercise, you removed the resources used in this lab.