Manage Azure Storage

Lab scenario

You need to evaluate the use of Azure storage for storing files residing currently in on-premises data stores. While majority of these files are not accessed frequently, there are some exceptions. You would like to minimize cost of storage by placing less frequently accessed files in lower-priced storage tiers. You also plan to explore different protection mechanisms that Azure Storage offers, including network access, authentication, authorization, and replication. Finally, you want to determine to what extent Azure Files service might be suitable for hosting your on-premises file shares.

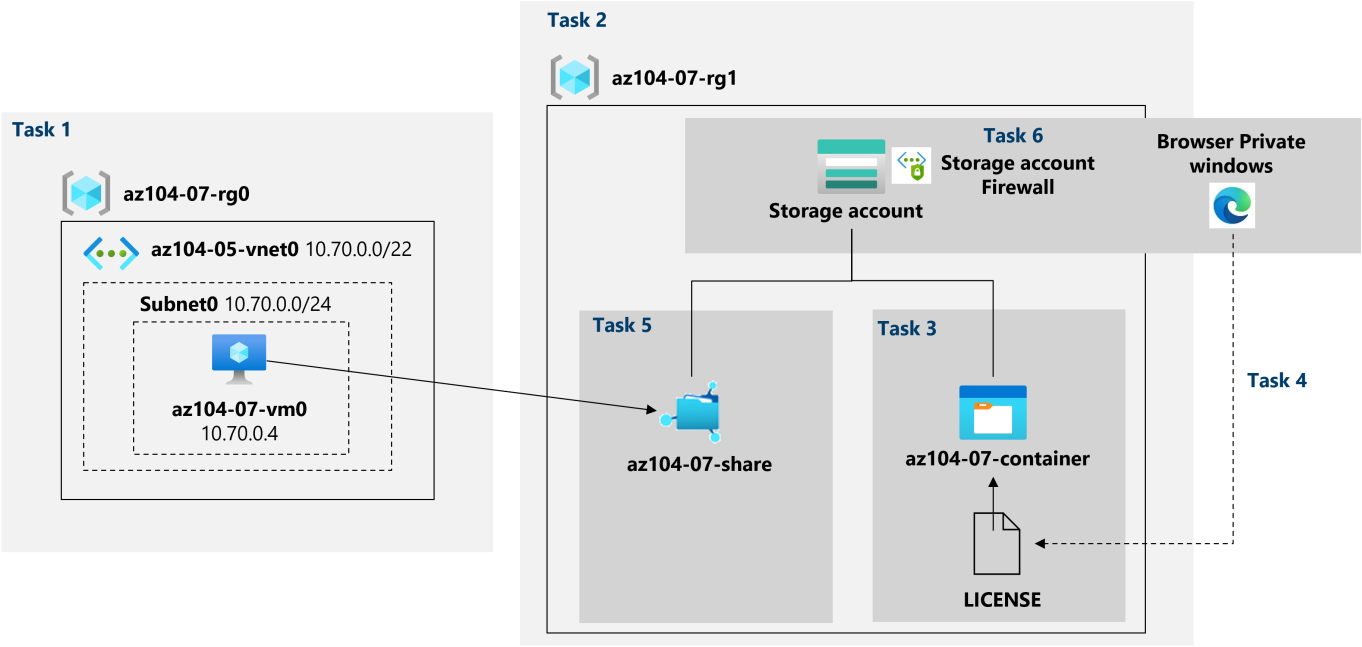
Objectives

In this lab, you will:

- Task 1: Provision the lab environment
- Task 2: Create and configure Azure Storage accounts
- Task 3: Manage blob storage
- Task 4: Manage authentication and authorization for Azure Storage
- Task 5: Create and configure an Azure Files shares
- Task 6: Manage network access for Azure Storage

Estimated timing: 40 minutes

Architecture diagram



Instructions

Exercise 1

Task 1: Provision the lab environment

In this task, you will deploy an Azure virtual machine that you will use later in this lab.

1. Sign in to the Azure portal.

2. In the Azure portal, open the **Azure Cloud Shell** by clicking on the icon in the top right of the Azure Portal.

3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

   > **Note**: If this is the first time you are starting **Cloud Shell** and you are presented with the **You have no storage mounted** message, select the subscription you are using in this lab, and click **Create storage**.

4. In the toolbar of the Cloud Shell pane, click the **Upload/Download files** icon, in the drop-down menu, click **Upload** and upload the files **\Labs\01\az104-07-vm-template.json** and **\Labs\01\az104-07-vm-parameters.json** into the Cloud Shell home directory.

5. Edit the **Parameters** file you just uploaded and change the password. If you need help editing the file in the Shell please ask your instructor for assistance. As a best practice, secrets, like passwords, should be more securely stored in the Key Vault.

6. From the Cloud Shell pane, run the following to create the resource group that will be hosting the virtual machine (replace the '[Azure_region]' placeholder with the name of an Azure region where you intend to deploy the Azure virtual machine)

   > **Note**: To list the names of Azure regions, run `(Get-AzLocation).Location` **Note**: Each command below should be typed separately

   ```
   $location = '[Azure_region]'
   ```

   ```
    $rgName = 'az104-07-rg0'
   ```

   ```
   New-AzResourceGroup -Name $rgName -Location $location
   ```

7. From the Cloud Shell pane, run the following to deploy the virtual machine by using the uploaded template and parameter files:

   ```
   New-AzResourceGroupDeployment `
      -ResourceGroupName $rgName `
      -TemplateFile $HOME/az104-07-vm-template.json `
      -TemplateParameterFile $HOME/az104-07-vm-parameters.json `
      -AsJob
   ```

   > **Note**: Do not wait for the deployments to complete, but proceed to the next task.

> **Note**: If you got an error stating the VM size is not available please ask your instructor for assistance and try these steps.
>
> 1. Click on the `{}` button in your CloudShell, select the **az104-07-vm-parameters.json** from the left hand side bar and take a note of the `vmSize` parameter value.
> 2. Check the location in which the 'az104-04-rg1' resource group is deployed. You can run `az group show -n az104-04-rg1 --query location` in your CloudShell to get it.
> 3. Run `az vm list-skus --location <Replace with your location> -o table --query "[? contains(name,'Standard_D2s')].name"` in your CloudShell.
> 4. Replace the value of `vmSize` parameter with one of the values returned by the command you just run.
> 5. Now redeploy your templates by running the `New-AzResourceGroupDeployment` command again. You can press the up button a few times which would bring the last executed command.

8. Close the Cloud Shell pane.

Task 2: Create and configure Azure Storage accounts

In this task, you will create and configure an Azure Storage account.

1. In the Azure portal, search for and select **Storage accounts**, and then click **+ Create**.

2. On the **Basics** tab of the **Create storage account** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a **new** resource group **az104-07-rg1**
Storage account name	any globally unique name between 3 and 24 in length consisting of letters and digits
Region	the name of an Azure region where you can create an Azure Storage account
Performance	**Standard**
Redundancy	**Geo-redundant storage (GRS)**

3. Click **Next: Advanced >**, on the **Advanced** tab of the **Create storage account** blade, review the available options, accept the defaults, and click **Next: Networking >**.

4. On the **Networking** tab of the **Create storage account** blade, review the available options, accept the default option **Enable public access from all networks** and click **Next: Data protection >**.

5. On the **Data protection** tab of the **Create storage account** blade, review the available options, accept the defaults, click **Review + Create**, wait for the validation process to complete and click **Create**.

> **Note**: Wait for the Storage account to be created. This should take about 2 minutes.

6. On the deployment blade, click **Go to resource** to display the Azure Storage account blade.

7. On the Storage account blade, in the **Data management** section, click **Redundancy** and note the secondary location.

8. In the **Redundancy** drop-down list select **Locally redundant storage (LRS)** and save the change. Note, at this point, the Storage account has only the primary location.

9. On the Storage account blade, in the **Settings** section, select **Configuration**. Set **Blob access tier (default)** to **Cool**, and save the change.

   > **Note**: The cool access tier is optimal for data which is not accessed frequently.

Task 3: Manage blob storage

In this task, you will create a blob container and upload a blob into it.

1. On the Storage account blade, in the **Data storage** section, click **Containers**.

2. Click **+ Container** and create a container with the following settings:

Setting	Value
Name	**az104-07-container**
Public access level	**Private (no anonymous access)**

3. In the list of containers, click **az104-07-container** and then click **Upload**.

4. Browse to **\Labs\01\LICENSE** on your lab computer and click **Open**.

5. On the **Upload blob** blade, expand the **Advanced** section and specify the following settings (leave others with their default values):

Setting	Value
Authentication type	**Account key**
Blob type	**Block blob**
Block size	**4 MB**
Access tier	**Hot**
Upload to folder	**licenses**

   > **Note**: Access tier can be set for individual blobs.

6. Click **Upload**.

   > **Note**: Note that the upload automatically created a subfolder named **licenses**.

7. Back on the **az104-07-container** blade, click **licenses** and then click **LICENSE**.

8. On the **licenses/LICENSE** blade, review the available options.

> **Note**: You have the option to download the blob, change its access tier (it is currently set to **Hot**), acquire a lease, which would change its lease status to **Locked** (it is currently set to **Unlocked**) and protect the blob from being modified or deleted, as well as assign custom metadata (by specifying an arbitrary key and value pairs). You also have the ability to **Edit** the file directly within the Azure portal interface, without downloading it first. You can also create snapshots, as well as generate a SAS token (you will explore this option in the next task).

Task 4: Manage authentication and authorization for Azure Storage

In this task, you will configure authentication and authorization for Azure Storage.

1. On the **licenses/LICENSE** blade, on the **Overview** tab, click **Copy to clipboard** button next to the **URL** entry.

2. Open another browser window by using InPrivate mode and navigate to the URL you copied in the previous step.

3. You should be presented with an XML-formatted message stating **ResourceNotFound** or **PublicAccessNotPermitted**.

   > **Note**: This is expected, since the container you created has the public access level set to **Private (no anonymous access)**.

4. Close the InPrivate mode browser window, return to the browser window showing the **licenses/LICENSE** blade of the Azure Storage container, and switch to the the **Generate SAS** tab.

5. On the **Generate SAS** tab of the **licenses/LICENSE** blade, specify the following settings (leave others with their default values):

Setting	Value
Signing key	**Key 1**
Permissions	**Read**
Start date	yesterday's date
Start time	current time
Expiry date	tomorrow's date
Expiry time	current time
Allowed IP addresses	leave blank

6. Click **Generate SAS token and URL**.

7. Click **Copy to clipboard** button next to the **Blob SAS URL** entry.

8. Open another browser window by using InPrivate mode and navigate to the URL you copied in the previous step.

   > **Note**: If you are using Microsoft Edge, you should be presented with the **The MIT License (MIT)** page. If you are using Chrome, Microsoft Edge (Chromium) or Firefox, you should be able to

> view the content of the file by downloading it and opening it with Notepad.

> **Note**: This is expected, since now your access is authorized based on the newly generated the SAS token.

> **Note**: Save the blob SAS URL. You will need it later in this lab.

9. Close the InPrivate mode browser window, return to the browser window showing the **licenses/LICENSE** blade of the Azure Storage container, and from there, navigate back to the **az104-07-container** blade.

10. Click the **Switch to the Azure AD User Account** link next to the **Authentication method** label.

    > **Note**: You can see an error when you change the authentication method (the error is *"You do not have permissions to list the data using your user account with Azure AD"*). It is expected.

    > **Note**: At this point, you do not have permissions to change the Authentication method.

11. On the **az104-07-container** blade, click **Access Control (IAM)**.

12. On the **Check access** tab, click **Add role assignment**.

13. On the **Add role assignment** blade, specify the following settings:

Setting	Value
Role	**Storage Blob Data Owner**
Assign access to	**User, group, or service principal**
Members	the name of your user account

14. Click **Review + Assign** and then **Review + assign**, and return to the **Overview** blade of the **az104-07-container** container and verify that you can change the Authentication method to (Switch to Azure AD User Account).

    > **Note**: It might take about 5 minutes for the change to take effect.

Task 5: Create and configure an Azure Files shares

In this task, you will create and configure Azure Files shares.

> **Note**: Before you start this task, verify that the virtual machine you provisioned in the first task of this lab is running.

1. In the Azure portal, navigate back to the blade of the storage account you created in the first task of this lab and, in the **Data storage** section, click **File shares**.

2. Click **+ File share** and create a file share with the following settings:

Setting	Value
Name	**az104-07-share**

3. Click the newly created file share and click **Connect**.

4. On the **Connect** blade, ensure that the **Windows** tab is selected. Below you will find a button with the label **Show Script**. Click on the button and you will find grey textbox with a script, in the bottom right corner of that box hover over the pages icon and click **Copy to clipboard**.

5. In the Azure portal, search for and select **Virtual machines**, and, in the list of virtual machines, click **az104-07-vm0**.

6. On the **az104-07-vm0** blade, in the **Operations** section, click **Run command**.

7. On the **az104-07-vm0 - Run command** blade, click **RunPowerShellScript**.

8. On the **Run Command Script** blade, paste the script you copied earlier in this task into the **PowerShell Script** pane and click **Run**.

9. Verify that the script completed successfully.

10. Replace the content of the **PowerShell Script** pane with the following script and click **Run**:

    ```
    New-Item -Type Directory -Path 'Z:\az104-07-folder'

    New-Item -Type File -Path 'Z:\az104-07-folder\az-104-07-file.txt'
    ```

11. Verify that the script completed successfully.

12. Navigate back to the **az104-07-share** file share blade, click **Refresh**, and verify that **az104-07-folder** appears in the list of folders.

13. Click **az104-07-folder** and verify that **az104-07-file.txt** appears in the list of files.

Task 6: Manage network access for Azure Storage

In this task, you will configure network access for Azure Storage.

1. In the Azure portal, navigate back to the blade of the storage account you created in the first task of this lab and, in the **Security + Networking** section, click **Networking** and then click **Firewalls and virtual networks**.

2. Click the **Enabled from selected virtual networks and IP addresses** option and review the configuration settings that become available once this option is enabled.

    > **Note**: You can use these settings to configure direct connectivity between Azure virtual machines on designated subnets of virtual networks and the storage account by using service endpoints.

3. Click the checkbox **Add your client IP address** and save the change.

4. Open another browser window by using InPrivate mode and navigate to the blob SAS URL you generated in the previous task.

> **Note**: If you did not record the SAS URL from task 4, you should generate a new one with the same configuration. Use Task 4 steps 4-6 as a guide for generating a new blob SAS URL.

5. You should be presented with the content of **The MIT License (MIT)** page.

   > **Note**: This is expected, since you are connecting from your client IP address.

6. Close the InPrivate mode browser window, return to the browser window showing the **Networking** blade of the Azure Storage account.

7. In the Azure portal, open the **Azure Cloud Shell** by clicking on the icon in the top right of the Azure Portal.

8. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

9. From the Cloud Shell pane, run the following to attempt downloading of the LICENSE blob from the **az104-07-container** container of the storage account (replace the `[blob SAS URL]` placeholder with the blob SAS URL you generated in the previous task):

   ```
   Invoke-WebRequest -URI '[blob SAS URL]'
   ```

10. Verify that the download attempt failed.

    > **Note**: You should receive the message stating **AuthorizationFailure: This request is not authorized to perform this operation**. This is expected, since you are connecting from the IP address assigned to an Azure VM hosting the Cloud Shell instance.

11. Close the Cloud Shell pane.

Clean up resources

> **Note**: Remember to remove any newly created Azure resources that you no longer use. Removing unused resources ensures you will not see unexpected charges.

> **Note**: Don't worry if the lab resources cannot be immediately removed. Sometimes resources have dependencies and take a long time to delete. It is a common Administrator task to monitor resource usage, so just periodically review your resources in the Portal to see how the cleanup is going. You might also try to delete the Resource Group where the resources reside. That is a quick Administrator shortcut. If you have concerns speak to your instructor.

1. In the Azure portal, open the **PowerShell** session within the **Cloud Shell** pane.

2. List all resource groups created throughout the labs of this module by running the following command:

   ```
   Get-AzResourceGroup -Name 'az104-07*'
   ```

3. Delete all resource groups you created throughout the labs of this module by running the following command:

```
Get-AzResourceGroup -Name 'az104-07*' | Remove-AzResourceGroup -Force -AsJob
```

> **Note**: The command executes asynchronously (as determined by the -AsJob parameter), so while you will be able to run another PowerShell command immediately afterwards within the same PowerShell session, it will take a few minutes before the resource groups are actually removed.

Review

In this lab, you have:

- Provisioned the lab environment
- Created and configured Azure Storage accounts
- Managed blob storage
- Managed authentication and authorization for Azure Storage
- Created and configured an Azure Files shares
- Managed network access for Azure Storage