In this lab, you will learn how to work with file shares in an Azure storage account using the Azure CLI.

Microsoft Azure is a cloud computing service offered and operated by Microsoft. Use this service to host your data and applications in the cloud.

An Azure storage account is an Azure cloud service that contains all of your Azure storage data objects: blobs, file shares, queues, tables, and disks. You can host files, including images, videos, music and binary, as well as noSQL data and messages in the cloud using this service.

Azure Files offers fully managed file shares in the cloud that are accessible via the industry-standard Server Message Block (SMB) protocol or the Network file system (NFS) protocol.

Create a new file share in Azure Storage Account
Upload files to file shares and list them
Clean up the file share and storage account

You're now ready to start deploying Azure storage files (Azure Files).

Do you need to host your local SMB or NFS file shares in the cloud? You can do so by creating an Azure Files service.

Step 2: Confirm the New Storage Account Has Been Created
We have created an Azure storage account for this lab. You need this Account to work with Azure Files.

az storage account show -g $resource -n $storageAccountName

```
$
$ az storage account show -g $resource -n $storageAccountName
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": true,
  "allowCrossTenantReplication": null,
  "allowSharedKeyAccess": null,
  "allowedCopyScope": null,
```

Step 3: Create a New File Share
As noted before, Azure Files offers fully managed file shares in the cloud that are accessible via the industry-standard Server Message Block (SMB) protocol or the Network file system (NFS) protocol.

One of the methods to authenticate into a storage account is by using an account key. The following command fetches the primary mykatastorageaccount account key and stores it into the $storageKey environment variable:

storageKey=$(az storage account keys list -g $resource -n $storageAccountName --output json --query [0].value)

Next, run the following command to create a new file share, documents, in your storage account:

az storage share create --name documents --account-name $storageAccountName --account-key $storageKey

Confirm that the new file share was created in your storage account by running the following:

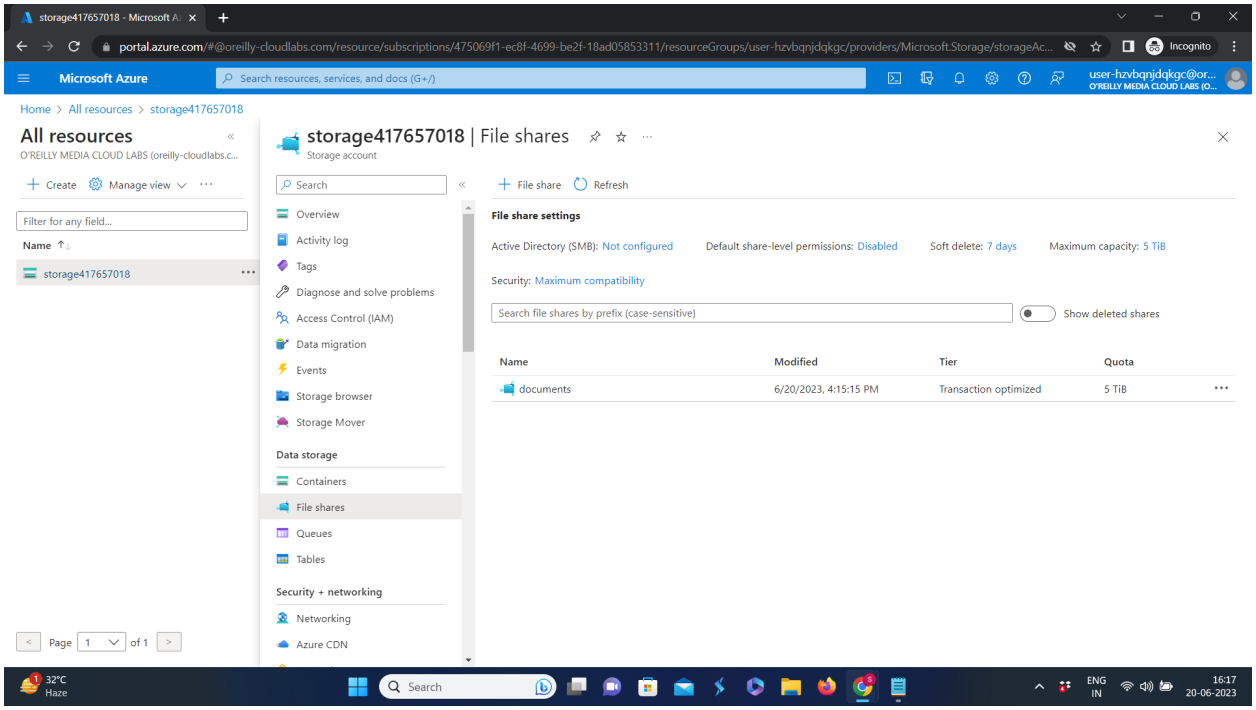
az storage share list --account-name $storageAccountName --account-key $storageKey

We have a new file share, now it's time to upload a new file to it! See the next step for details.

```
$ storageKey=$(az storage account keys list -g $resource -n $storageAccountName --output json --query [0].value)
$ az storage share create --name documents --account-name $storageAccountName --account-key $storageKey
{
  "created": true
}
$ az storage share list --account-name $storageAccountName --account-key $storageKey
[
  {
    "accessTier": "TransactionOptimized",
    "deleted": null,
    "deletedTime": null,
    "etag": "\"0x8DB717B6E80E5D4\"",
    "lastModified": "2023-06-20T10:45:15+00:00",
    "lease": {
      "duration": null,
      "state": "available",
      "status": "unlocked"
    },
    "metadata": null,
    "name": "documents",
    "nextAllowedQuotaDowngradeTime": null,
```

Step 4: Create a New File to Upload

```
$ echo "This is a document to be shared with the company employees!" > mydocument.txt
$ ls
mydocument.txt
$
```

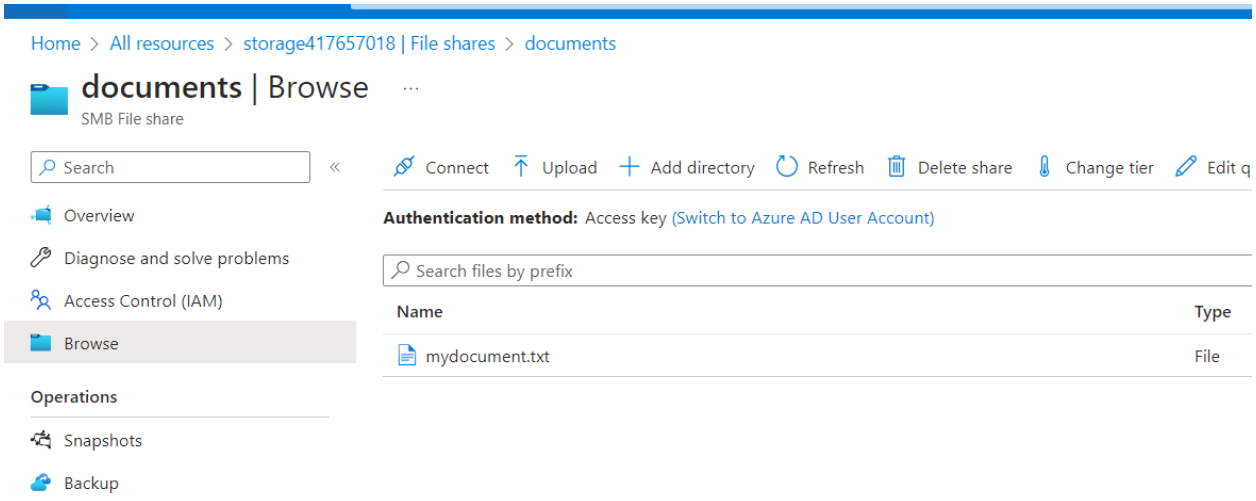


Step 5: Upload a New File to the File Share
Time to upload our local file to our documents file share:

az storage file upload --share-name documents --source ./mydocument.txt --account-name $storageAccountName --account-key $storageKey

You can confirm that the file was uploaded to the file share by running this command:

az storage file list --share-name documents --account-name $storageAccountName --account-key $storageKey



az storage file delete --share-name documents --path mydocument.txt --account-name $storageAccountName --account-key $storageKey

You can confirm that the file was deleted by running this command:

az storage file list --share-name documents --account-name $storageAccountName --account-key $storageKey

We can delete the file share as well, using the following:

az storage share delete --name documents --account-name $storageAccountName --account-key $storageKey

Use the following command to confirm that the new file share was deleted from your storage account.

az storage share list --account-name $storageAccountName --account-key $storageKey

You can also delete the parent storage account, which we'll do in the next step.

```
]
$ az storage file delete --share-name documents --path mydocument.txt --account-name $storageAccountName --account-key $storageKey
{
  "deleted": null
}
$ az storage file list --share-name documents --account-name $storageAccountName --account-key $storageKey
[]
$ az storage share delete --name documents --account-name $storageAccountName --account-key $storageKey
{
  "deleted": true
}
$ az storage share list --account-name $storageAccountName --account-key $storageKey
[]
$
```

Step 8: Delete the Storage Account

```
$ az storage account delete -n $storageAccountName -g $resource
Are you sure you want to perform this operation? (y/n): y
$ az storage account list -g $resource
[]
$
```