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.

In this lab you will learn about soft delete for Azure storage blobs.

#### **Learning Objective**

In this lab, you will learn how to do the following:

Log into the Azure CLI
Understand soft delete for Azure storage blobs
Enable soft delete for Azure storage blobs using CLI
Confirm the soft delete setting using Azure CLI
Check the soft delete settings in the portal
Clean up

az login -u \$username -p \$password

# To create a new Azure storage account:

az storage account create -n \$storageAccountName --resource-group \$resource --sku Standard\_LRS

# To create a new blob container under the above storage account:

az storage container create --name myfiles --account-name \$storageAccountName --account-key \$storageKey

### To upload a sample text file to the above container:

az storage blob upload --account-name \$storageAccountName --account-key \$storageKey --container-name myfiles --file blob.txt --name blob.txt

# **Understand Azure Storage Soft Delete for Blobs**

You can store different file types in Azure storage blobs. You need a way to protect these files against accidental deletion, so that the deleted files can be recovered if needed.

In this lab you will learn about soft delete for Azure storage blobs. This feature, if enabled, allows you to recover (undelete) deleted blob files within the configured retention period.

We already created a new storage account and uploaded a text file to a blob container. Use the following command to confirm that the account is present:

```
$ az storage blob list --account-name $storageAccountName --account-key $storageKey --container-name myfiles --query [].name

{
    "blob.txt"
    $ az storage blob list --account-name $storageAccountName --account-key $storageKey --container-name myfiles

{
        "container": "myfiles",
        "content": "",
        "deleted": null,
        "encryptedMetadata": null,
        "encryptionKeySha256": null,
        "encryptionScope": null,
        "hasLegalHold": null,
        "inmutabilityPolicy": {
        "expiryTime": null,
        "policyMode": null
    },
        "isAppendBlobSealed": null,
        "isCurrentVersion": null,
        "
```

In the next step, we will enable soft delete for this storage account and will delete and recover a text file.

# **Enable Soft Delete for Storage Blobs using Azure CLI**

Use the following command to enable the soft delete feature for your storage account:

az storage account blob-service-properties update --account-name \$storageAccountName --resource-group \$resource --enable-delete-retention true --delete-retention-days 14

Wait for the command to execute. Here are the command parameters:

- --resource-group: The parent resource group for the storage account
- --account-name: The storage account name
- --enable-delete-retention: Enable or disable soft delete
- --delete-retention-days: The soft delete retention period in days. The deleted files can be undeleted within this period.

az storage account blob-service-properties update is a command that can be used with the Azure CLI (Command Line Interface) to update the Blob service properties of a storage account in Azure.

Use the following command to confirm that the soft delete is enabled for your storage account blobs:

```
$ az storage account blob-service-properties show --account-name $storageAccountName --resource-group $resource --query deleteRetentionPolicy
{
    "allowPermanentDelete": false,
    "days": 14,
    "enabled": true
}

$ az storage account blob-service-properties show --account-name $storageAccountName --resource-group $resource {
    "automaticSnapshotPolicyEnabled": null,
    "containerDeleteRetentionPolicy": null,
    "containerDeleteRetentionPolicy": null,
    "corsRules": []
    "defaultServiceVersion": null,
    "deleteRetentionPolicy": {
        "allowPermanentDelete": false,
        "days": 14,
        "enabled": true
},
    "id": "/subscriptions/7075f968-4ed0-4eda-a971-4afd22b73b63/resourceGroups/user-jxtusihgdugn/providers/Microsoft.Storage/storageAccounts/storult",
    "isVersioningEnabled": null,
    "lasVersioningEnabled": null,
    "name": "default",
    "resourceGroup: "user-jxtusihgdugp",
    "resourceGroup: "user-jxtusihgdugp",
    "resourceGroup: "user-jxtusihgdugp",
    "resourceGroup: "user-jxtusihgdugp",
    "resourceGroup: "standard LRS",
    "tier": "Standard LRS",
    "tier": "Standard LRS",
    "tier": "Standard
},
    "type": "Microsoft.Storage/storageAccounts/blobServices"
}
```

In the next step you will recover a deleted blob file using the soft delete protection.

## **Confirm Soft Delete Settings using Azure CLI**

Soft delete is enabled for your storage account and the retention period is 14 days. This means you can recover any deleted blob file within 14 days of the deletion time.

Now, let's delete the blob.txt file from our container:

az storage blob delete --account-name \$storageAccountName --account-key \$storageKey --container-name myfiles --name blob.txt

Now use the following command to confirm that the blob file is deleted:

az storage blob list --account-name \$storageAccountName --account-key \$storageKey --container-name myfiles --query [].name

Now, use the following command to recover (undelete) your blob file:

az storage blob undelete --account-name \$storageAccountName --account-key \$storageKey --container-name myfiles --name blob.txt

```
$ az storage blob undelete --account-name $storageAccountName --account-key $storageKey --container-name myfiles --name blob.txt
{
    "undeleted": null
}
$ [
```

Important: The previous command only succeeds if you are within the configured retention period, in our case 14 days.

Now use the following command again to confirm the blob file is recovered:

```
$ az storage blob list --account-name $storageAccountName --account-key $storageKey --container-name myfiles --query [].name
[
    "blob.txt"
]
$
```

The command should return blob.txt as the only result.

In the next step, we will check the soft delete setting in the Azure portal.

# **Check Soft Delete Settings in the Azure Portal**

In the Overview page, confirm that you see Blob soft delete set to Enabled (14 days). 😙 Interactive Learning x | 😙 Interactive Scenario: Protecting A x | 😙 Interactive Scenario: Protecting A x | 🐧 Interactive Scenario: Protecting A x | A storage 12037700 - Microsoft As: x + - - - - - - - X ← → C a portal.azure.com/#@oreilly-cloudlabs.com/resource/subscriptions/7075/968-4ed0-4eda-a971-4afd22b/3b63/resourceGroups/user-jxtusihqduqp/providers/Microsoft-Storage/storage/c... 😉 🖻 🖈 🔭 🛽 🦠 Pocuments - OneD... K Recommended Job... MONSTER Shine.com - My Shi... I Indeed Resume G GLASSDOOR I JOBS\_PORTAL 10 udemy storage12037700 🖈 🖈 … Storage accounts + Create 7 Restore ··· Blob service Security Overview Filter for any field... Require secure transfer for REST API operations Hierarchical namespace Disabled Name ↑ Hot Storage account key access ■ storage12037700 ... ♦ Tags Enabled Version 1.0 Minimum TLS version Enabled (14 days) Infrastructure encryption Disabled Disabled Networking Disabled Events Allow access from NFS v3 Disabled Storage browser Number of private endpoint connections 0 Allow cross-tenant replication Enabled Network routing Microsoft network routing File service Access for trusted Microsoft services Containers Large file share Disabled Endpoint type File shares Not configured Active Directory Default share-level permissions Disabled Tables Soft delete Enabled (7 days) Security + networking Share capacity 5 TiB Queue service < Page 1 V of 1 > Azure CDN CMK support Q Search 🔯 🔎 🗩 🧰 🐸 🕫 🗲 🗘 🥅 🥰 

## Clean Up

Use the following command to clean up the storage accounts from your allocated resource group. Deleting a storage account will delete all the child blob containers, queues, file shares, and tables:

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

Use the following command to list all storage accounts in your allocated resource group:

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