WHIZLABS

Azure Blob Storage using CLI

Level: **Intermediate**

Azure Storage Account Azure CLI Azure

0h 37m 47s left

15 MIN

End Lab

Open Console

Validation

Lab Credentials

User Name

labuser_80425_41771100@instructorwhizlabs.onmicrosoft.com

Password

9d0QsBu4$eC!#*7v%

Resource Group

rg_eastus_80425_1_170568577881

Lab Resources

No Lab Resources Found

Support Documents

No Support Documents Found

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview | Lab Steps | Lab Validation

Azure Administrator Associate, Azure Data Scientist Associate, Azure Developer Associate, Cloud Developer, Data Analyst

Storage

Lab Steps

Task 1: Sign in to Azure Portal

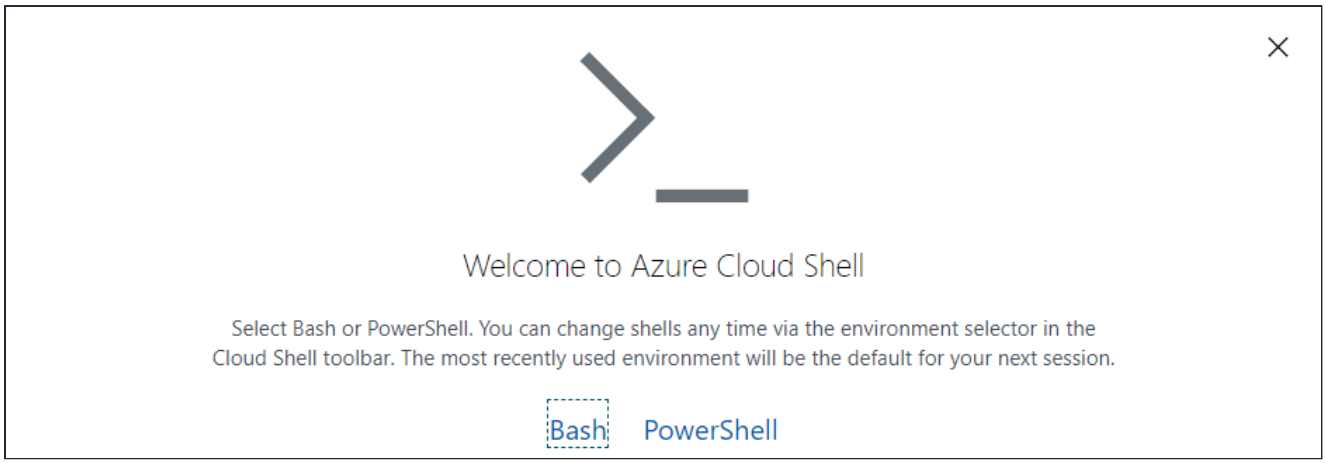
1. Go to the Azure portal by clicking on the **Open Console** button or by using URL https://portal.azure.com.
    - **Note**: It is recommended to use incognito mode to avoid Azure portal cache related issues.
2. If it automatically logs into any other azure account, please logout of it and clear cache.
3. Sign in with your given ***username*** and ***password*** on Azure portal.
4. If login is not working. Click on **End Lab** and start the lab again.

Task 2: Create a blob storage using CLI

In this task, we will create a Blob storage account using the Azure CLI and specify parameters such as account name, resource group, location, and SKU.

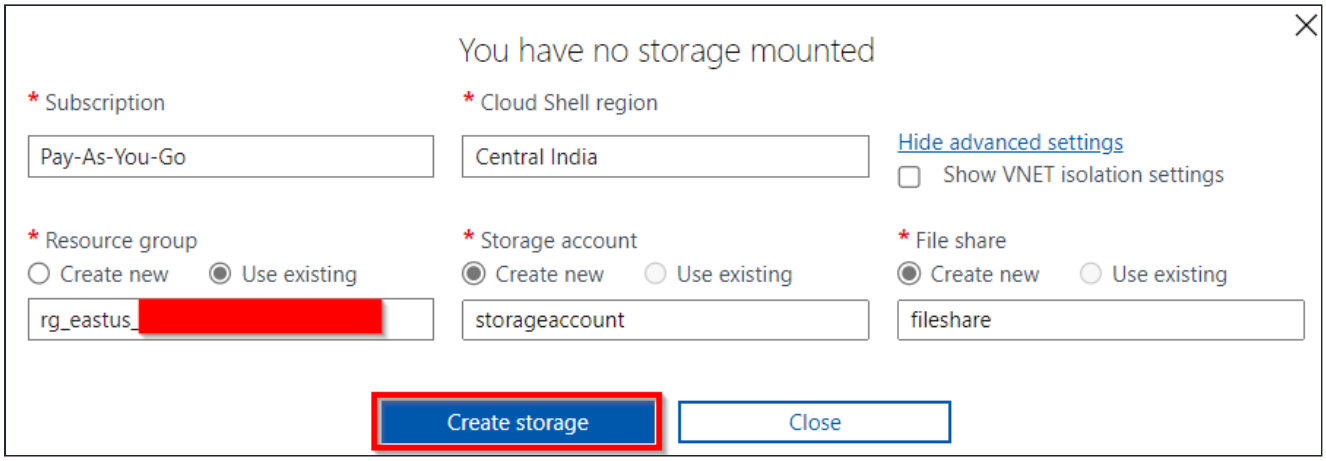
1. On the Azure portal click on the power shell icon at the top.

2. Now click on **Bash**



3. Now click on **show advanced settings**. Now, enter a unique **storage name** and **file share name** and then, click on **Create Storage.**

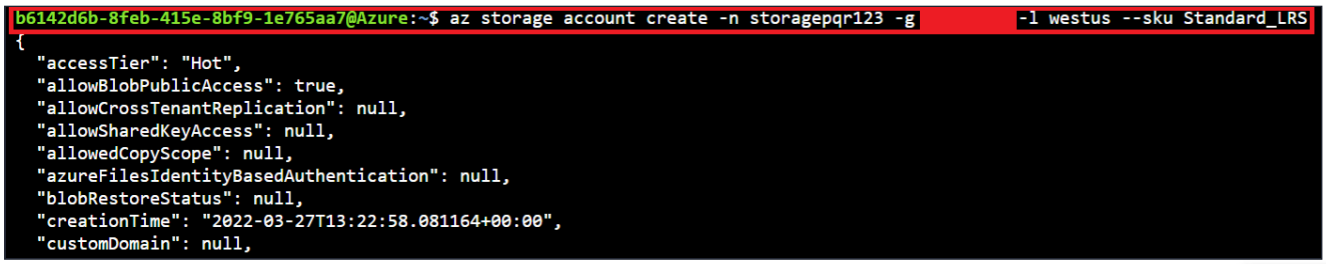
Note: Make sure to keep a copy the name of your resource group.



4. Now enter the following command to create a storage account.

```
az storage account create -n <storage account name> -g
<resource group name> -l westus --sku Standard_LRS
```

Note: Replace **<storage account name>** with an unique name and **<resource group name>** with your resource group name.



Do you Know?

Azure Blob Storage offers server-side encryption for automatic encryption of data at rest, enhancing data security.

Task 3: Create a Container using CLI

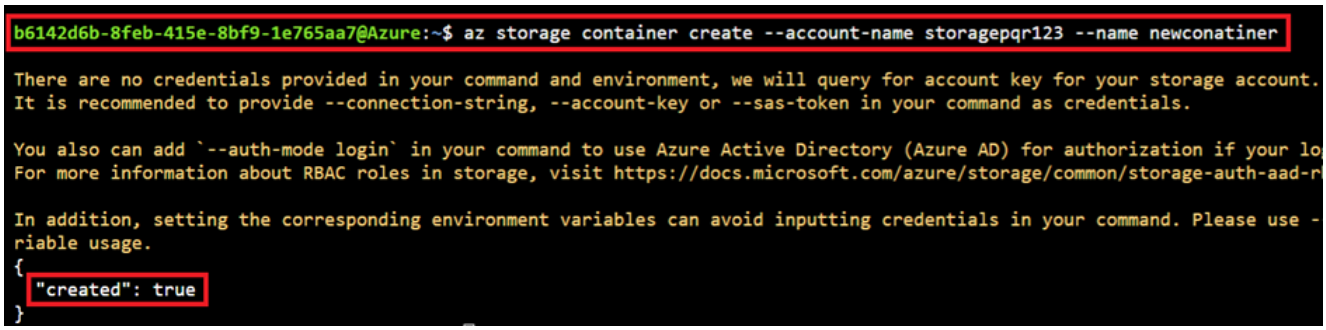
In this task, we will create a container in the Blob storage account using the Azure CLI and specify the storage account name and container name as parameters.

1. Open a **new session** in the **Bash** by clicking on the following icon.

2. Now type this command to create a container.

```
az storage container create --account-name <storage account name> --name <container name>
```

Note: Replace **<storage account name>** with the name of storage account you gave while creating storage account in Task 2 and replace **<container name>** with a name you want to give to your container.



Task 4: Upload a blob on to storage account

In this task, we will upload a sample HTML file to the Blob storage container using the Azure CLI and then check the file's accessibility via its URL in the Azure portal.

1. Open a **new session** in the **Bash by** clicking on the following icon.

2. Enter the following command to **create** a sample html file.

```
echo This is a sample html file > sample.html
```

```
b6142d6b-8feb-415e-8bf9-1e765aa7@Azure:~$ echo This is a sample html file > sample.html
b6142d6b-8feb-415e-8bf9-1e765aa7@Azure:~$
```

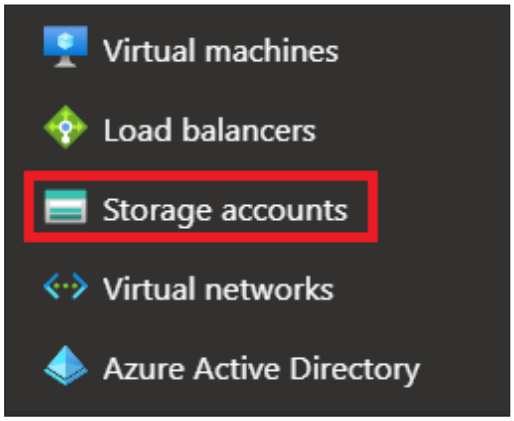
3. Now enter the following command to **upload a blob to the storage account** you created earlier.

```
az storage blob upload --account-name <storage account name> --container-name <container name> --name sample.html --file sample.html
```

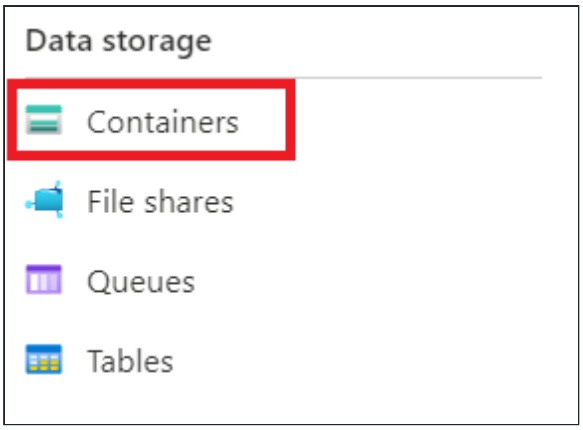
Note: Replace **<storage account name>** with the name of storage account you gave while creating in Task 2 and replace **<container name>** with a name you gave earlier to your container.

```
b6142d6b-8feb-415e-8bf9-1e765aa7@Azure:~$ az storage blob upload --account-name storagepqr123 --container-name newconatiner --name sample.html --file sample.html

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC rol
For more information about RBAC roles in storage, visit https://docs.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about envir
riable usage.
Finished[#############################################################]  100.0000%
{
  "client_request_id": "f9b5fbdc-add6-11ec-ae8c-0a580af4586b",
  "content_md5": "D6oU3ddBfGvFRX7GSgLULA==",
  "date": "2022-03-27T14:05:36+00:00",
  "encryption_key_sha256": null,
  "encryption_scope": null,
  "etag": "\"0x8DA0FFADE1514FE\"",
  "lastModified": "2022-03-27T14:05:36+00:00",
  "request_id": "e28b17b8-701e-004d-51e3-416f1b000000",
  "request_server_encrypted": true,
  "version": "2020-10-02",
  "version_id": null
}
```

4. Now **close** the bash terminal. Then, in the **portal menu** click on the **storage account**. And select the storage account you created.



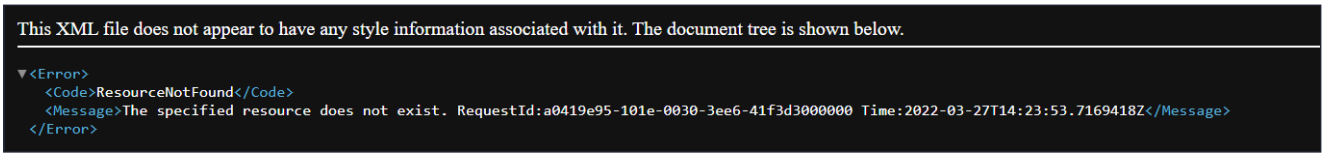
5. Now in the **blade** under the **data storage section** select **containers**. And **open** the container you created.



6. **Click** on the **sample html** file.

7. Copy the **URL** and paste it in the **new tab**.

8. You will see the below screen, that means your file **does not have public access.**

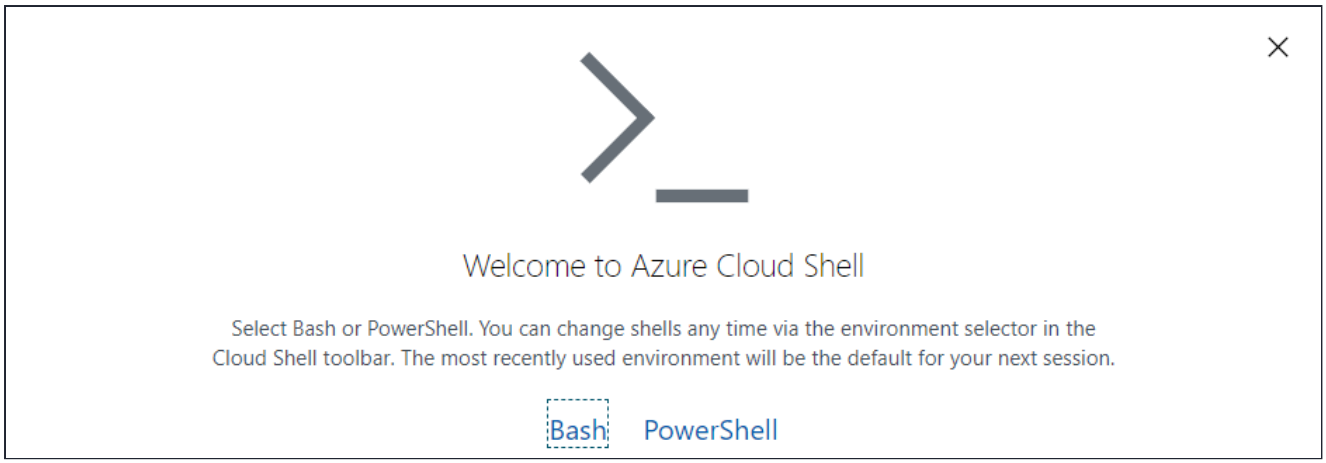


Task 5: Give public access to the blob file

In this task, we will use the Azure CLI to grant public access to the blob file in the Blob storage container, allowing access to the file's content via its URL in the Azure portal.

1. On the Azure portal **click** on the **power shell icon** at the top.

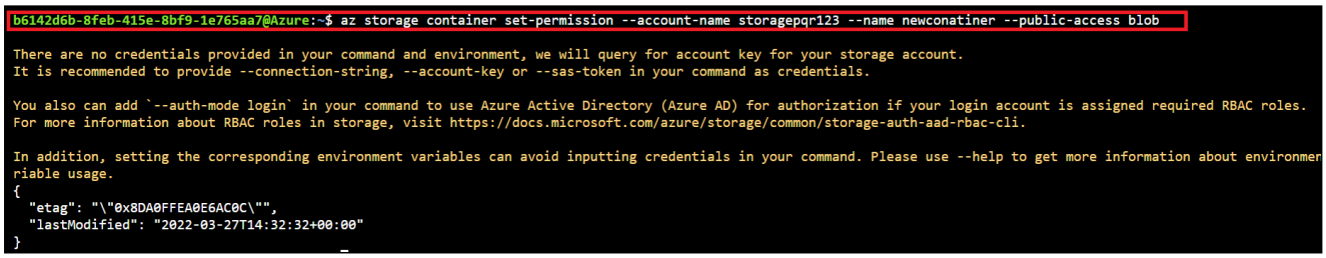
2. Now click on **Bash**



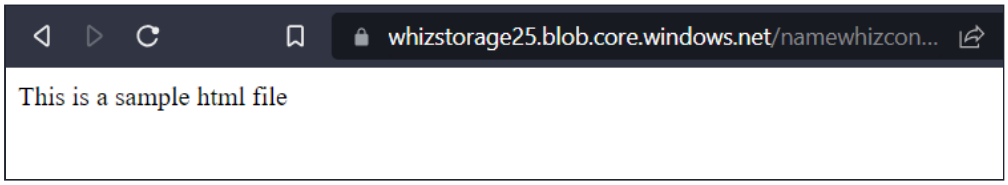
3. Enter the following command to give public access to the blob.

```
az storage container set-permission --account-name <storage account name> --name <container name> --public-access blob
```

Note: Replace **<storage account name>** with the name of storage account you gave while creating in **task 2** and replace **<container name>** with a name you gave earlier to your container.



4. Now again go to your sample html file, copy the URL and paste it in a new tab. You will see "**This is a sample html file**" written on the screen.



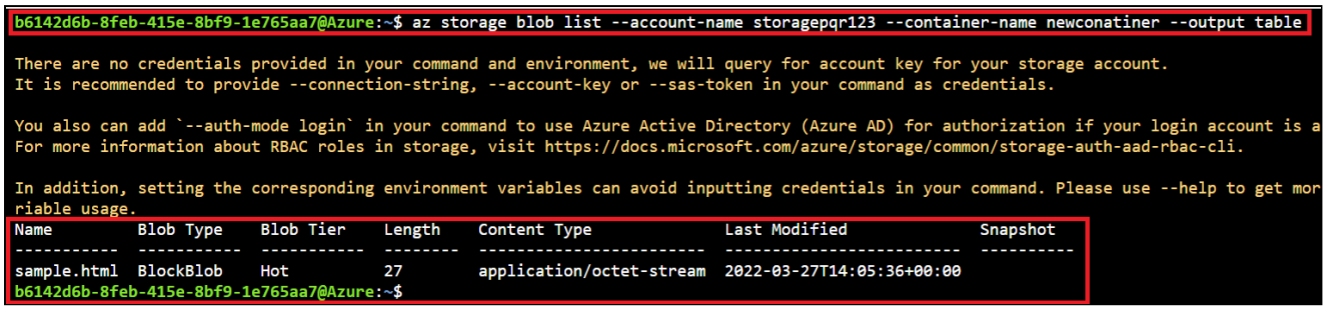
Task 6: List and download the blob using CLI

In this task, we will use the Azure CLI to list all the blobs present in the Blob storage container and then download a specific blob to the local file system.

1. Enter the following command to list all the blobs present in your container in a table format.

```
az storage blob list --account-name <storage account name> --container-name <container name> --output table
```

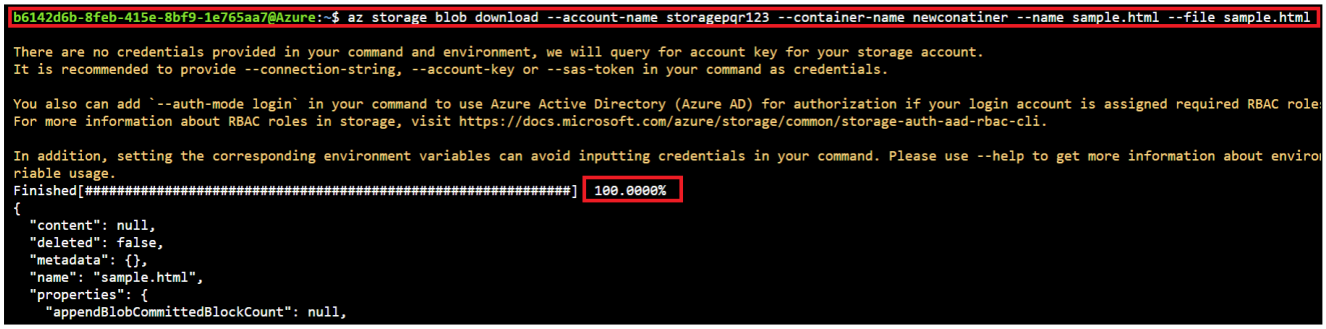
Note: Replace **<storage account name>** with the name of storage account you gave while creating in Task 2 and replace **<container name>** with a name you gave earlier to your container.



2. Enter the below command to download the present blob in your container.

```
az storage blob download --account-name <storage account
name> --container-name <container name> --name sample.html --
file sample2.html
```

Note: Replace **<storage account name>** with the name of storage account you gave while creating in Task 2 and replace **<container name>** with a name you gave earlier to your container.

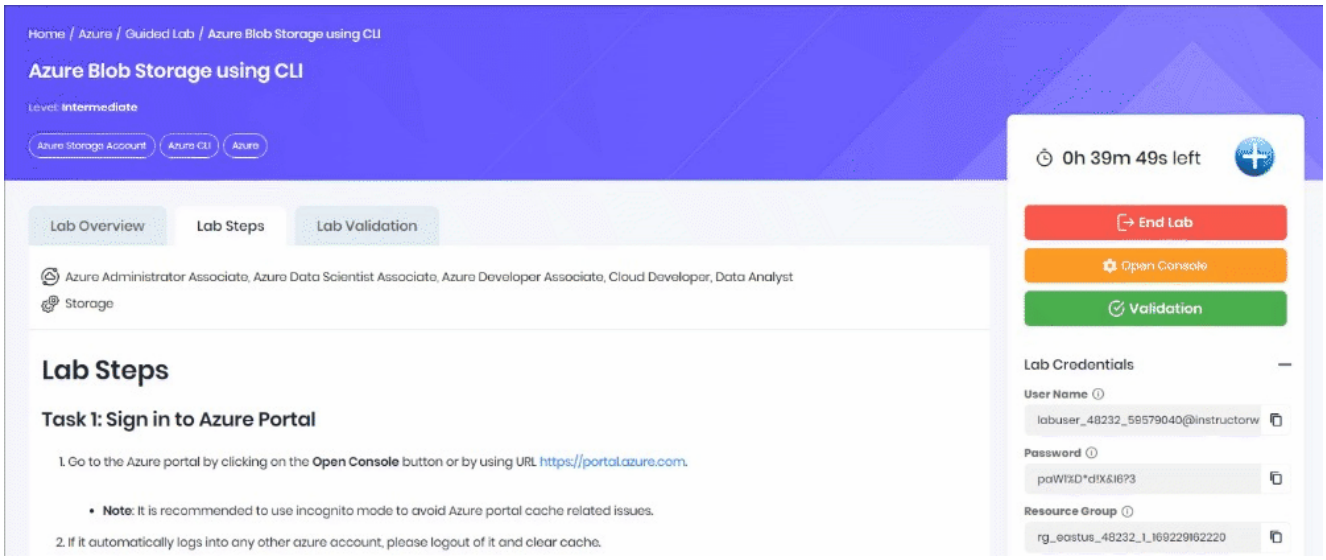


Task 7: Validation test

1. Once the lab steps are completed, click on **Validation** button or go to **Lab Validation** section.

2. Click on **Validate My Lab** button.

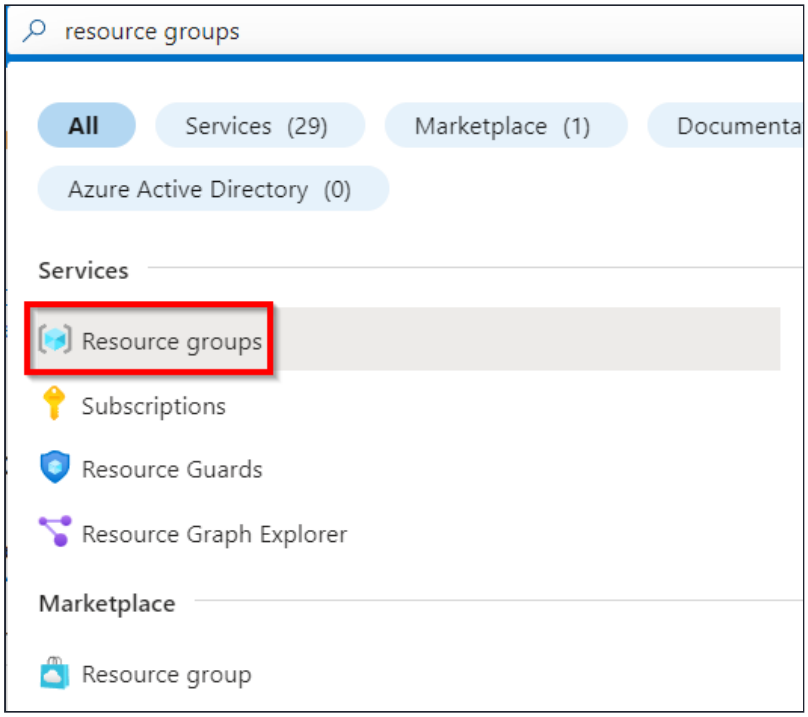
3. You will get the **"Lab Overall Status"** which will indicate whether or not you have completed the lab successfully.



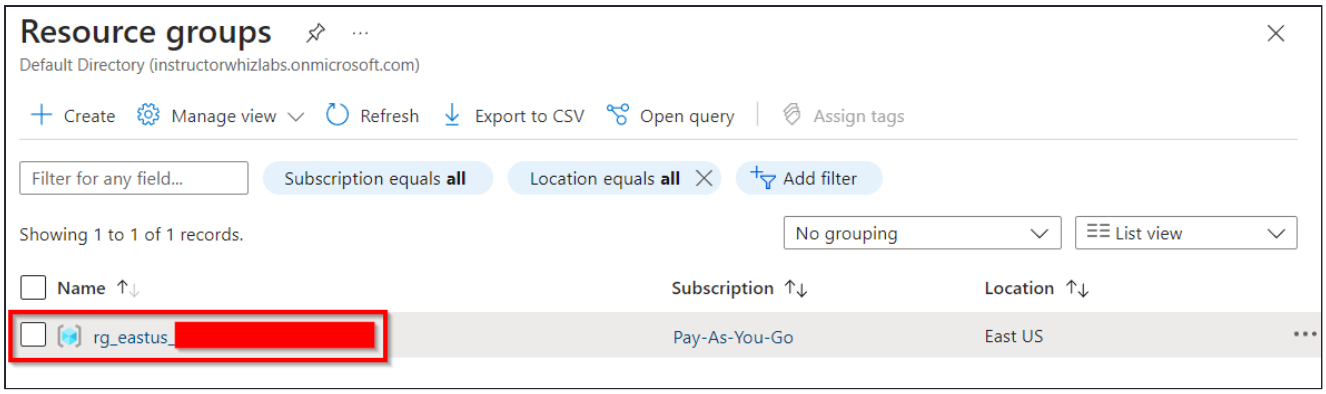
Task 8: Delete the Resources

In this task, we will delete all the resources.

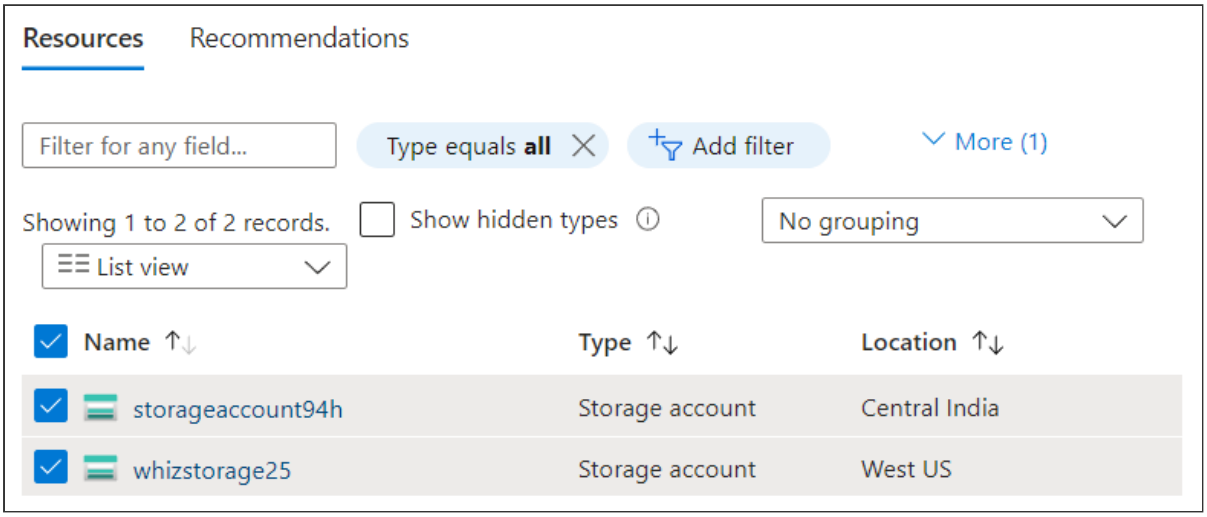
1. In the search box at the top of the Azure portal, enter **Resource groups**. Select **Resource groups** from the search results.



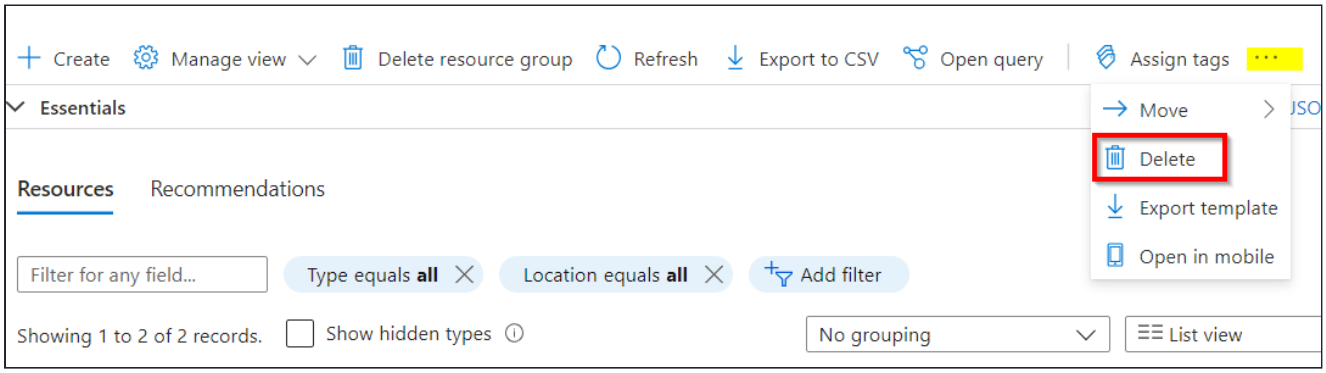
2. Click on the name of the **Resource groups**.



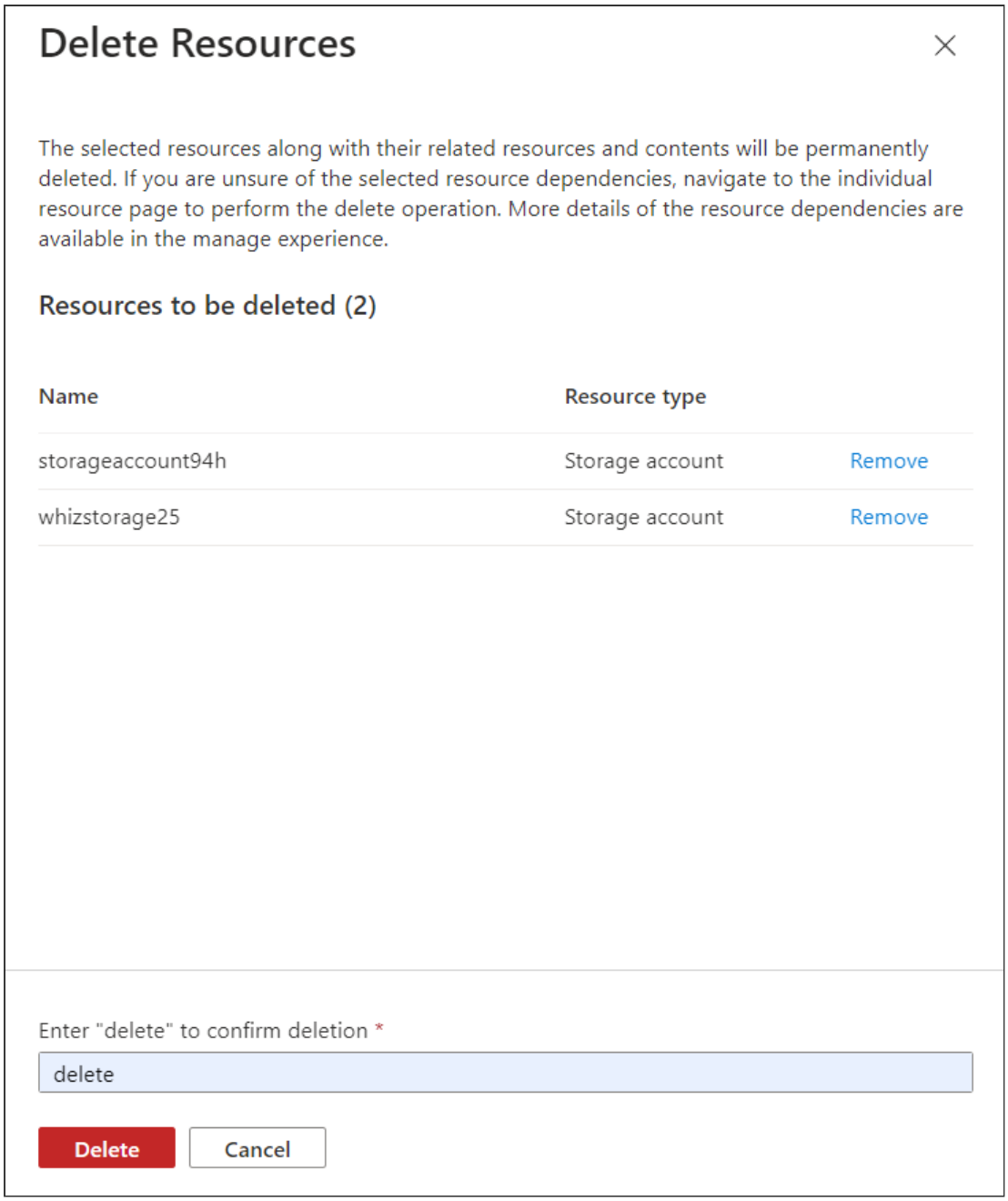
3. Select all the Resources in that **Resource groups.**



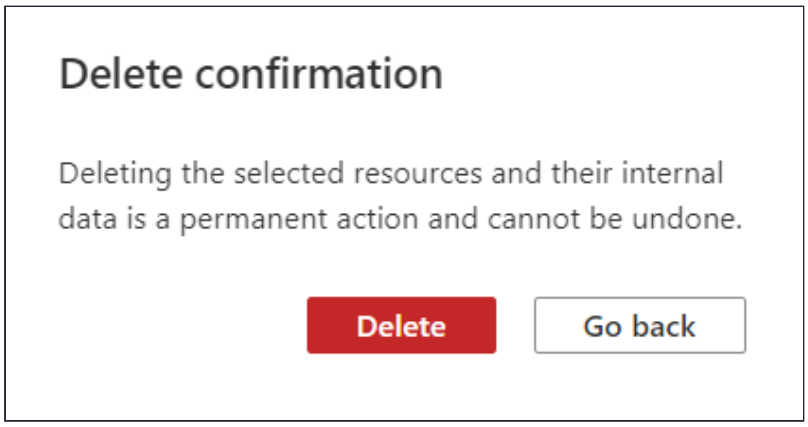
4. Go to three dots to the right and then click **Delete** button.



5. Now type **delete** in the box present at the bottom.



6. Click on **Delete** to confirm deletion of resources.



Completion and Conclusions

1. You have successfully logged into Azure Portal.
2. You have successfully created a storage account
3. You have successfully Created a container
4. You have successfully uploaded a blob to the container.
5. You have successfully given public access permission to your blob.
6. You have successfully listed and downloaded the present blob in your container.
7. You have successfully Validated the lab.
8. You have successfully deleted the resources.

End Lab

1. You have successfully completed this lab.
2. Click on **Sign out** in Azure Portal by clicking on the logout button in the top right corner inside Azure Profile.
3. Click on **End Lab** once you have completed the Lab.