REPORT – ON: -

Cloud Automation with Azure for creation of storage account and upload secret file Using UiPath

Name: Monu Chauhan
Reg. no: 12109067
Roll no: B52
Section: KO359

Submitted to
Mr. Vaibhav Chadha

Introduction: -

In the rapidly evolving landscape of cloud computing, automation plays a pivotal role in enhancing efficiency, security, and scalability. As a network engineer for AZ Network Corporation, leveraging cloud automation with Azure can significantly streamline your operations. We focus on the automation process for creating a storage account in Azure and securely uploading a secret file using UiPath, a leading Robotic Process Automation (RPA) tool. With Azure's robust cloud infrastructure and UiPath's intuitive automation capabilities, you can achieve a seamless integration that ensures secure handling and management of your data. The process involves creating a virtual network, deploying virtual machines for testing, setting up Azure Storage Account, and using UiPath to automate the upload of a secret file to the storage account. Through this automated approach, you not only enhance the security of your data but also ensure a scalable and efficient workflow that can be easily managed and monitored. This guide will walk you through the step-by-step procedure to harness the power of cloud automation with Azure and UiPath, ensuring that your distributed application runs smoothly and securely.

There are some points about storage account in azure.

1. Storage account: -

An Azure storage account is a fundamental building block for Azure Storage, providing a unique namespace for your data objects. It contains all your Azure Storage data objects, such as **blobs, files, queues, and tables**. Here are some key points about Azure storage accounts:

- **Namespace**: Each storage account provides a unique namespace that is accessible from anywhere in the world over HTTP or HTTPS.
- **Data Durability and Availability**: Data in your storage account is durable and highly available. Redundancy options like Locally Redundant Storage (LRS), Geo-Redundant Storage (GRS), and Zone-Redundant Storage (ZRS) ensure your data is safe and accessible even in the event of hardware failures or regional outages.
- **Security**: All data written to an Azure storage account is encrypted by the service, and you have fine-grained control over who has access to your data.
- **Scalability**: Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Access**: Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. You can use client libraries for various

programming languages, Azure PowerShell, Azure CLI, and the Azure portal to interact with your storage account

2. Binary Large Object: -

Azure Blob Storage is Microsoft's object storage solution for the cloud, designed to store massive amounts of unstructured data, such as text or binary data.
Here are some key points about Azure Blob Storage:

- **Unstructured Data**: Ideal for storing data that doesn't adhere to a specific data model, such as images, documents, videos, and backups.
- **Scalability**: Azure Blob Storage can store petabytes of data and is highly scalable to meet the needs of various applications.
- **Durability and Availability**: Data stored in Azure Blob Storage is durable and highly available, with options for redundancy like Locally Redundant Storage (LRS), Geo-Redundant Storage (GRS), and Zone-Redundant Storage (ZRS).
- **Security**: Data is encrypted at rest and in transit, and you have fine-grained control over access using Azure Active Directory and role-based access control (RBAC).
- **Access**: Data can be accessed from anywhere in the world over HTTP or HTTPS, and you can use client libraries for various programming languages, Azure PowerShell, Azure CLI, and the Azure portal1.
- **Types of Blobs**: There are three types of blobs: Block blobs (ideal for text and binary data), Append blobs (used for append operations), and Page blobs (used for random read/write operations).

3. Containers: -

In Azure, a **container** is a storage entity that organizes a set of blobs, similar to a directory in a file system. Here are some key points about containers in Azure Blob Storage:

- **Organizing Data**: Containers help you manage and organize your data by grouping related blobs together.
- **Unlimited Containers**: A single storage account can include an unlimited number of containers, and each container can store an unlimited number of blobs.
- **Access Control**: You can set access policies at the container level to control who can access the blobs within the container.
- **Naming Rules**: Container names must be lowercase and can include only letters, numbers, and the dash (-) character. They must start

with a letter or number and be between 3 and 63 characters long.

- **Anonymous Access**: You can configure containers to allow or deny anonymous access. The recommended level is Private (no anonymous access) to ensure data security.

4. SAS: -

A **Shared Access Signature (SAS)** is a URI that grants restricted access rights to Azure Storage resources. It allows you to delegate access to your storage resources without sharing your account key. Here are some key points about SAS:

- **Granular Access Control**: With SAS, you have control over what resources the client can access, what permissions they have, and how long the SAS is valid.
- **Types of SAS**: There are three types of shared access signatures:
  - **User Delegation SAS**: Secured with Microsoft Entra credentials, providing superior security.
  - **Service SAS**: Secured with the storage account key and delegates access to a specific resource in one of the Azure Storage services (Blob, Queue, Table, or File storage).
  - **Account SAS**: Secured with the storage account key and delegates access to resources in one or more of the storage services.
- **Ad Hoc SAS**: When you create an ad hoc SAS, the start time, expiry time, and permissions are specified in the SAS URI.
- **Stored Access Policy**: A stored access policy is defined on a resource container (blob container, table, queue, or file share) and can be used to manage constraints for one or more service shared access signatures

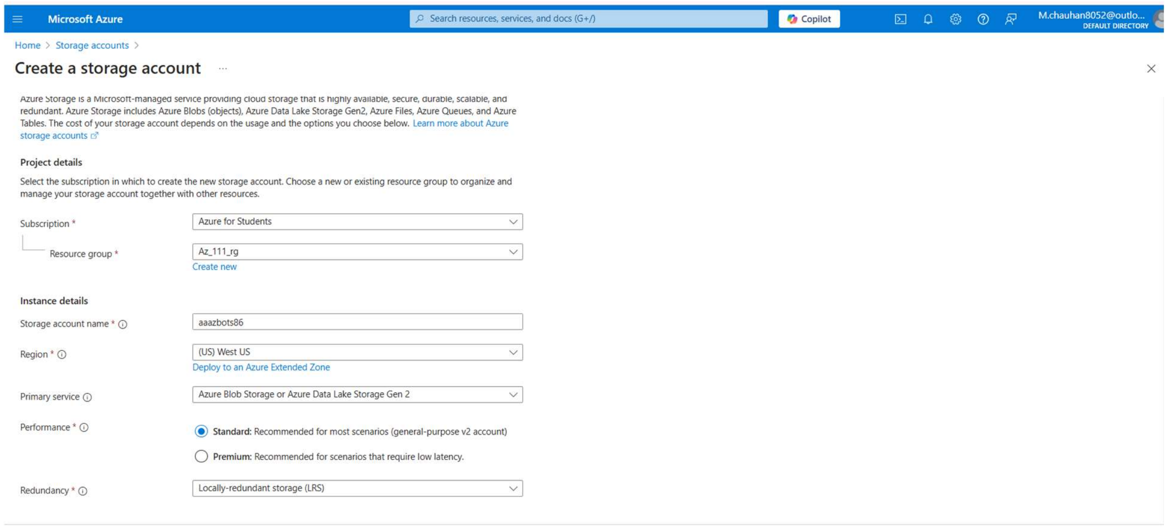
Working: -

Step1:

First of all using UiPath I have created a storage account in azure in storage account I have given resource group after given of resource group I have set zone, after setting of region, we setup Primary service after this setup LRS.

An Azure storage account is a fundamental building block for Azure Storage, providing a unique namespace for your data objects. It contains all your Azure Storage data objects, such as **blobs, files, queues, and**

tables.



Home > Storage accounts >

Create a storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts

Project details

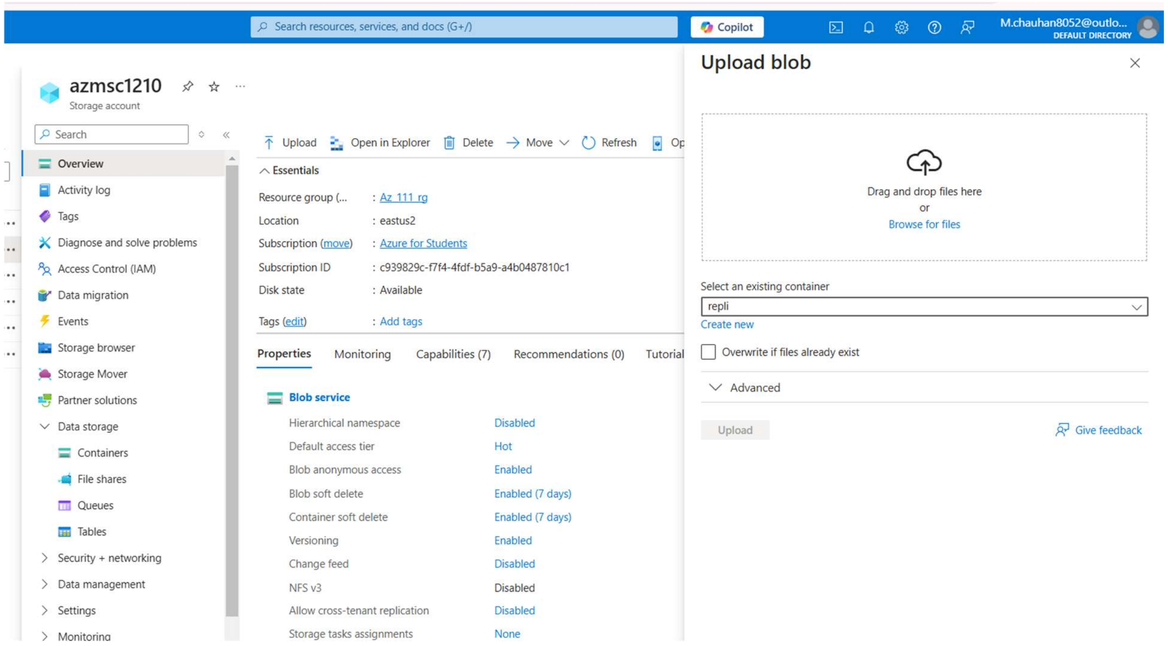
Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *	Azure for Students
Resource group *	Az_111_rg

Instance details

Storage account name *	aaazbots86
Region *	(US) West US
Primary service	Azure Blob Storage or Azure Data Lake Storage Gen 2
Performance *	[✓ **Standard:** Recommended for most scenarios (general-purpose v2 account)] **Premium:** Recommended for scenarios that require low latency.
Redundancy *	Locally-redundant storage (LRS)

Step2:
In this part after creation of storage account we upload blob (binary large object) and then we create container and upload it.
An Azure storage account is a fundamental building block for Azure Storage, providing a unique namespace for your data objects. It contains all your Azure Storage data objects, such as **blobs, files, queues, and tables**.



azmsc1210
Storage account

Essentials

Resource group (move)	Az 111 rg
Location	eastus2
Subscription (move)	Azure for Students
Subscription ID	c939829c-f7f4-4fdf-b5a9-a4b0487810c1
Disk state	Available
Tags (edit)	Add tags

Blob service

Hierarchical namespace	Disabled
Default access tier	Hot
Blob anonymous access	Enabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Enabled
Change feed	Disabled
NFS v3	Disabled
Allow cross-tenant replication	Disabled
Storage tasks assignments	None

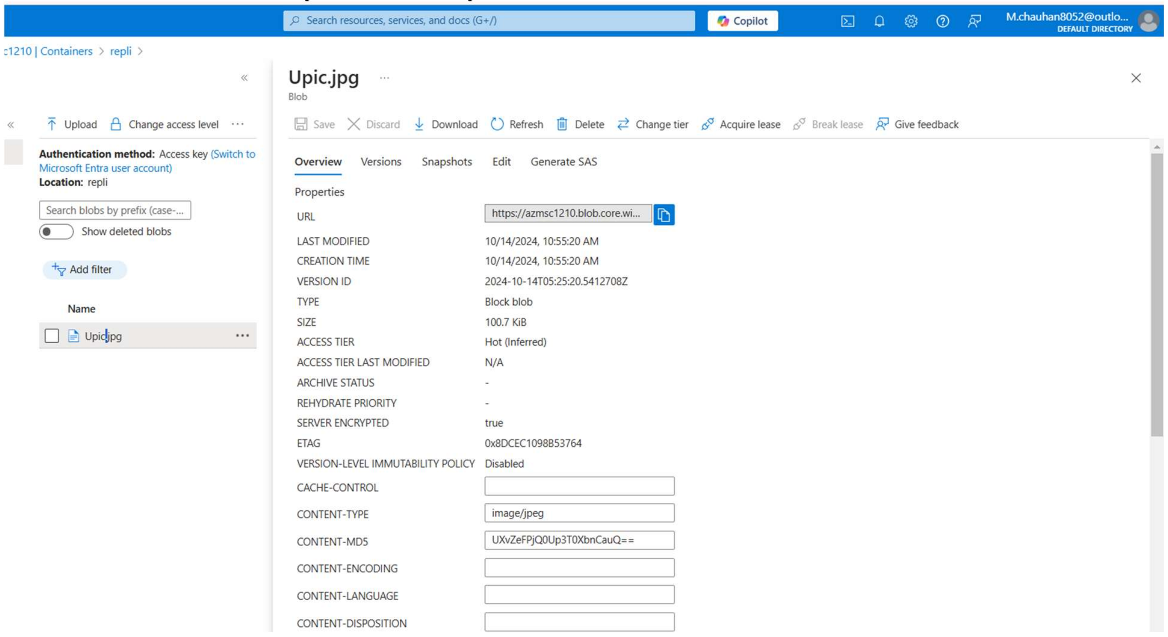
Upload blob

Select an existing container: repli

Overwrite if files already exist

Step 3:
In this part we after uploading our file then create container for access the uploaded file. In Azure, a **container** is a storage entity that organizes a set of blobs,

similar to a directory in a file system.



:1210 | Containers > repli >

Upic.jpg
Blob

Save | Discard | Download | Refresh | Delete | Change tier | Acquire lease | Break lease | Give feedback

Overview | Versions | Snapshots | Edit | Generate SAS

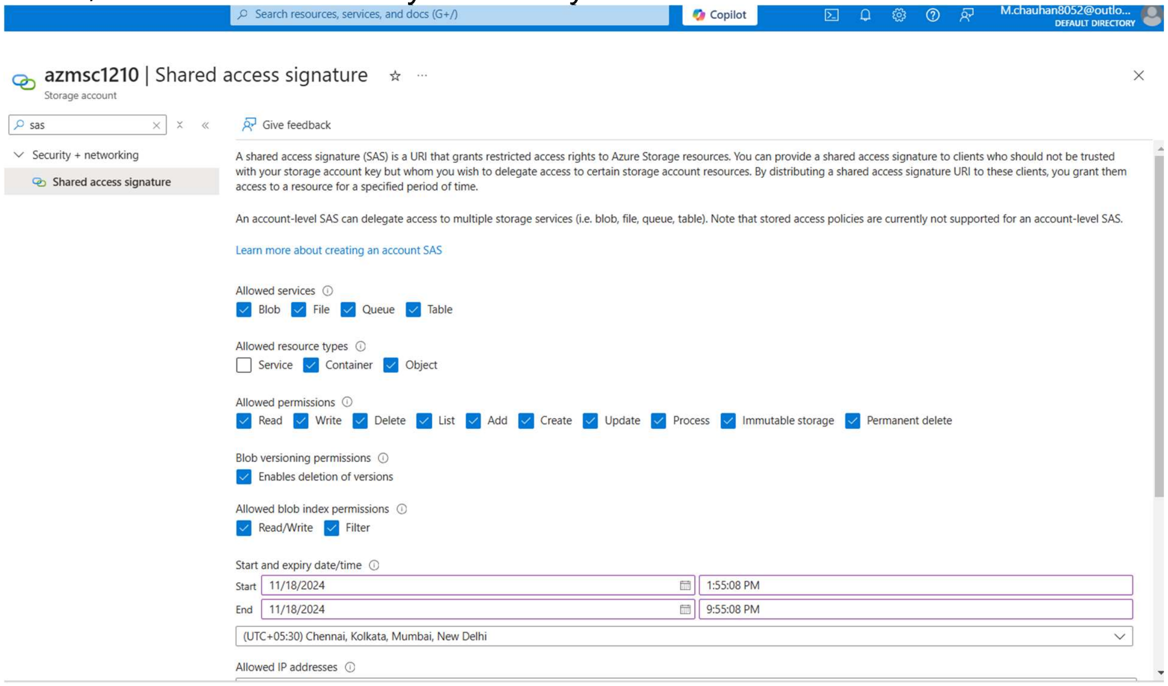
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: repli

Upic.jpg

Properties

URL	https://azmsc1210.blob.core.wi...
LAST MODIFIED	10/14/2024, 10:55:20 AM
CREATION TIME	10/14/2024, 10:55:20 AM
VERSION ID	2024-10-14T05:25:20.5412708Z
TYPE	Block blob
SIZE	100.7 KiB
ACCESS TIER	Hot (Inferred)
ACCESS TIER LAST MODIFIED	N/A
ARCHIVE STATUS	-
REHYDRATE PRIORITY	-
SERVER ENCRYPTED	true
ETAG	0x8DCEC1098B53764
VERSION-LEVEL IMMUTABILITY POLICY	Disabled
CACHE-CONTROL	
CONTENT-TYPE	image/jpeg
CONTENT-MD5	UXvZeFPjQ0Up3T0XbnCauQ==
CONTENT-ENCODING	
CONTENT-LANGUAGE	
CONTENT-DISPOSITION	

Step 4:
After copy of uploaded file link we will create SAS(shared access signature) for access the file. In Azure, a **container** is a storage entity that organizes a set of blobs, similar to a directory in a file system.



azmsc1210 | Shared access signature
Storage account

Security + networking
Shared access signature

A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

Learn more about creating an account SAS

Allowed services
✓ Blob ✓ File ✓ Queue ✓ Table

Allowed resource types
☐ Service ✓ Container ✓ Object

Allowed permissions
✓ Read ✓ Write ✓ Delete ✓ List ✓ Add ✓ Create ✓ Update ✓ Process ✓ Immutable storage ✓ Permanent delete

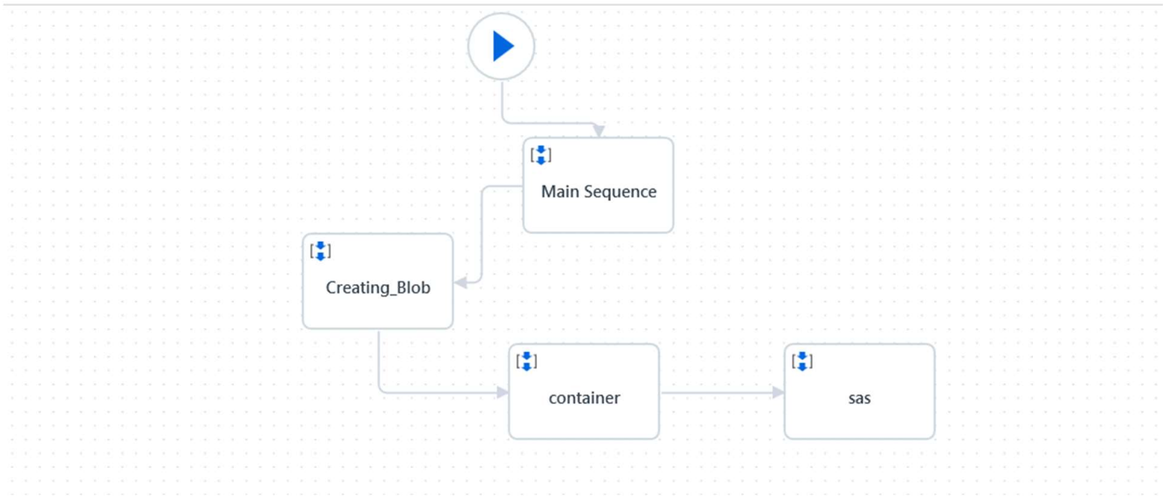
Blob versioning permissions
✓ Enables deletion of versions

Allowed blob index permissions
✓ Read/Write ✓ Filter

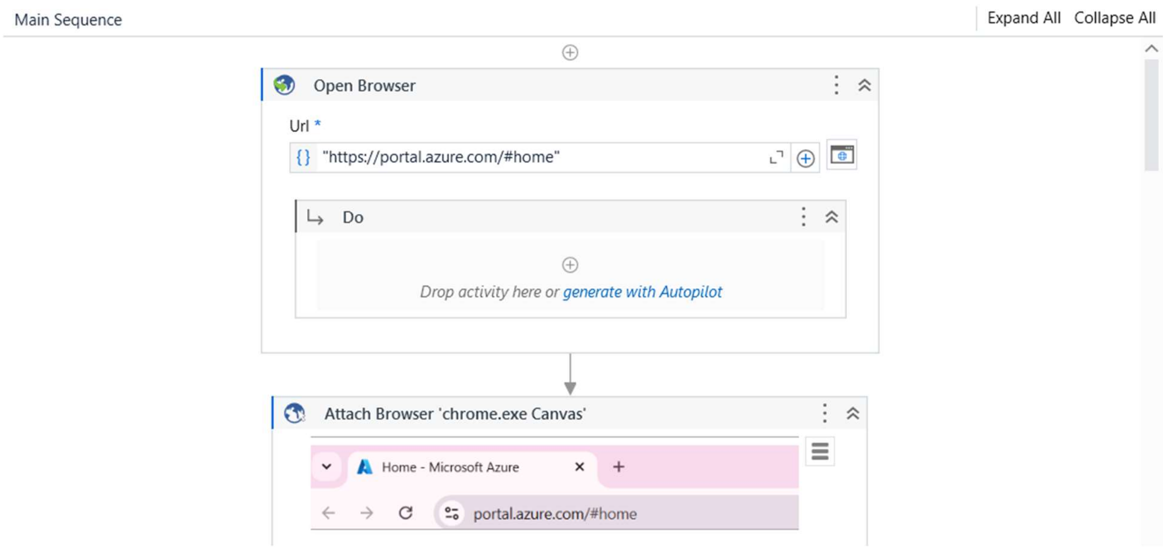
Start and expiry date/time
Start 11/18/2024 1:55:08 PM
End 11/18/2024 9:55:08 PM
(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses

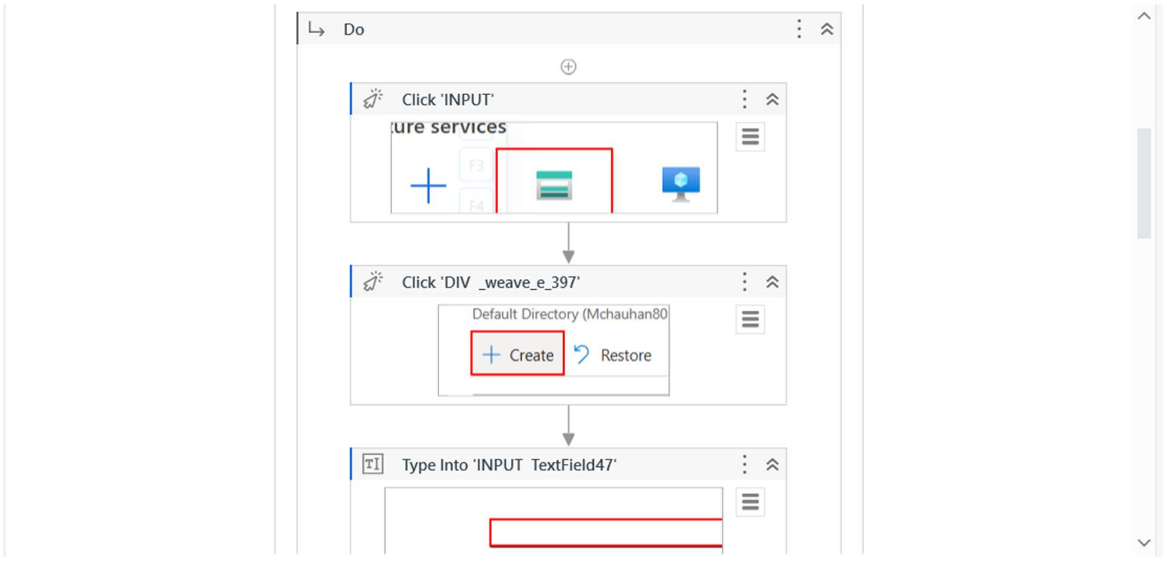
Flowchart:

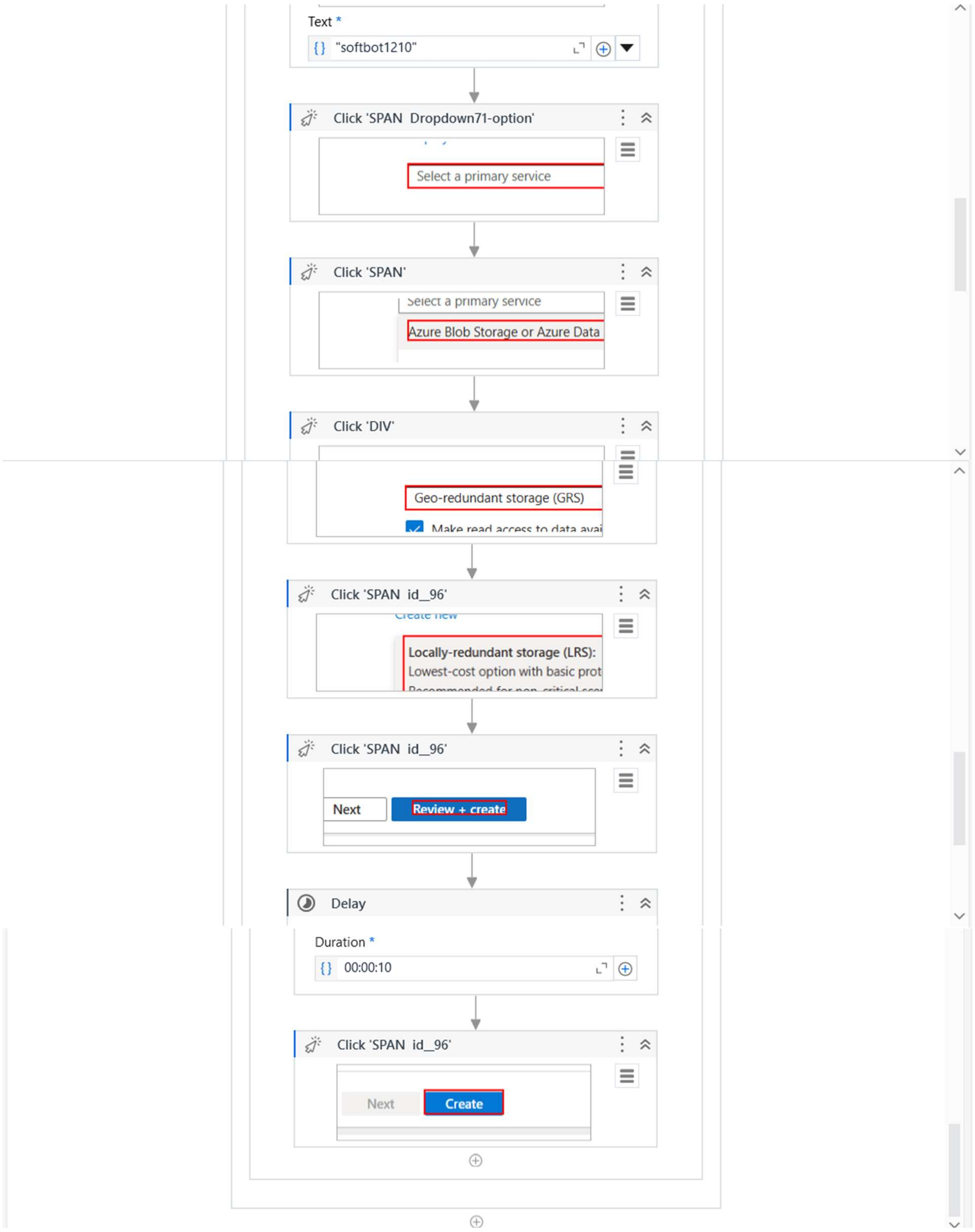


Main sequence:



Url: "https://portal.azure.com/#home"




Text *
"softbot1210"

Click 'SPAN Dropdown71-option'
Select a primary service

Click 'SPAN'
Select a primary service
Azure Blob Storage or Azure Data

Click 'DIV'
Geo-redundant storage (GRS)
Make read access to data avai

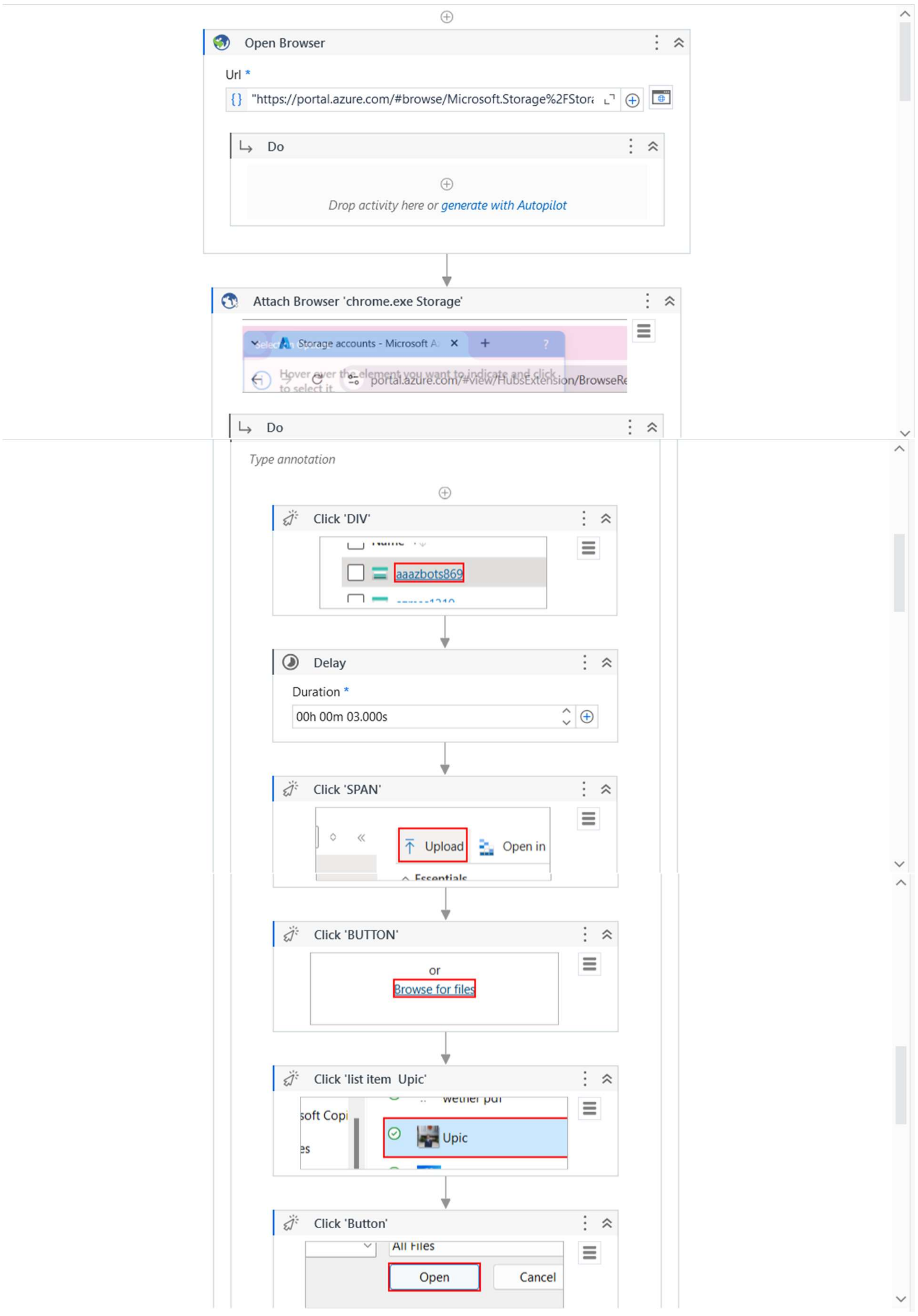
Click 'SPAN id__96'
Create new
Locally-redundant storage (LRS):
Lowest-cost option with basic prot

Click 'SPAN id__96'
Next
Review + create

Delay
Duration *
00:00:10

Click 'SPAN id__96'
Next
Create

Creation of Blob:


Open Browser

Url *

"https://portal.azure.com/#browse/Microsoft.Storage%2FStor

Do

Drop activity here or generate with Autopilot

Attach Browser 'chrome.exe Storage'

Storage accounts - Microsoft A

Hover over the element you want to indicate and click to select it

portal.azure.com/#view/HubsExtension/BrowseRe

Do

Type annotation

Click 'DIV'

aaazbots869

Delay

Duration *

00h 00m 03.000s

Click 'SPAN'

Upload

Open in

Essentials

Click 'BUTTON'

or

Browse for files

Click 'list item Upic'

wether pdf

soft Copi

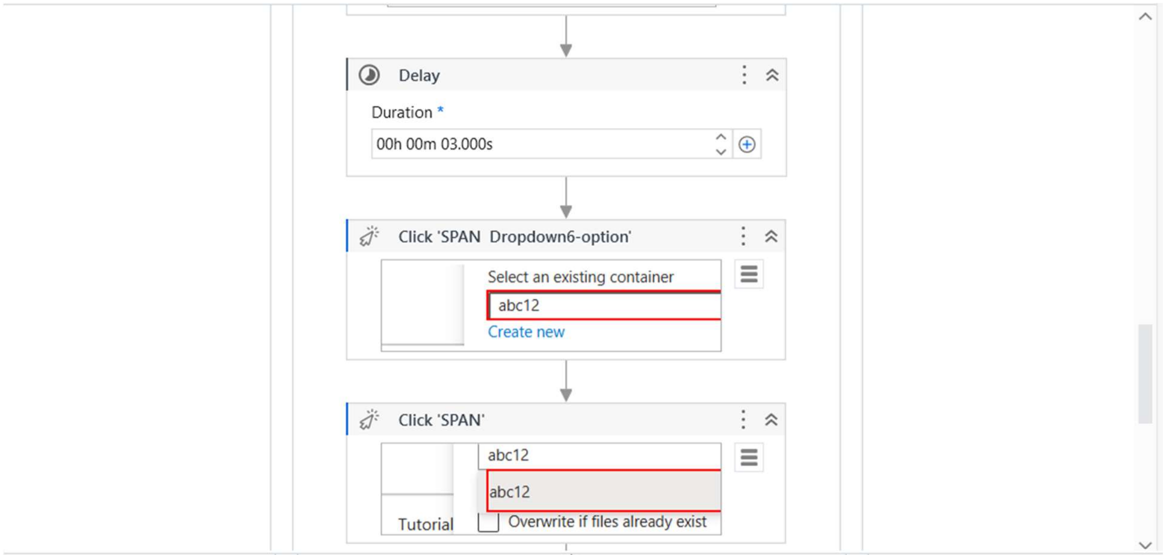
Upic

Click 'Button'

All Files

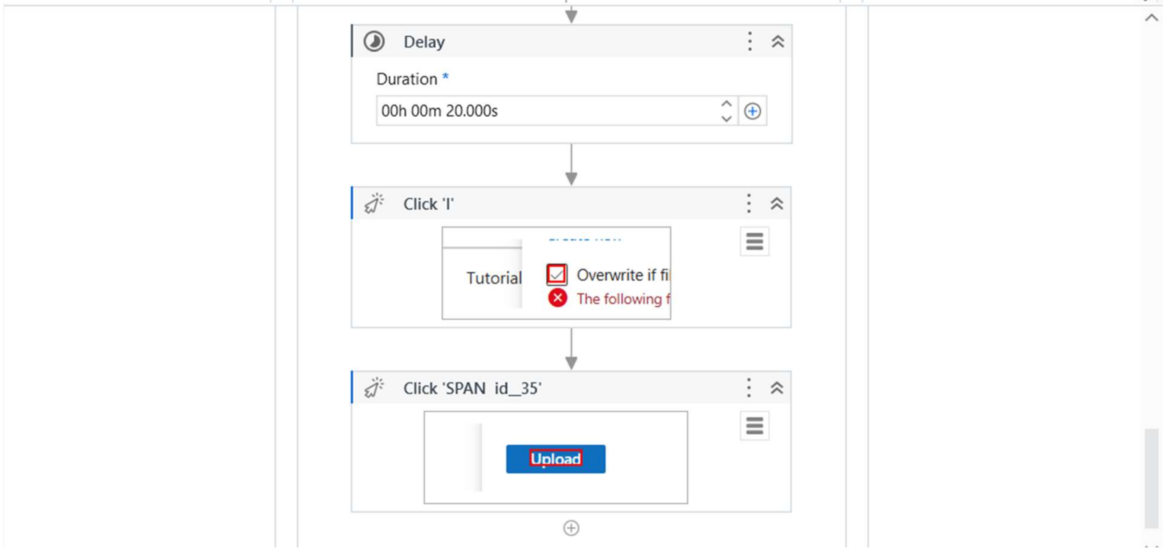
Open

Cancel


Delay
Duration *
00h 00m 03.000s

Click 'SPAN Dropdown6-option'
Select an existing container
abc12
Create new

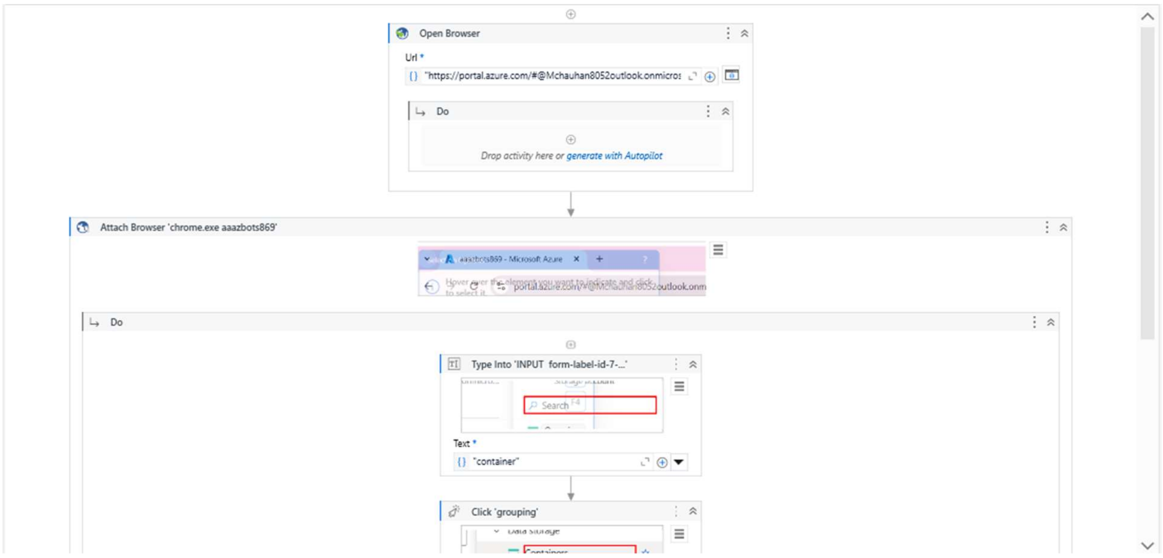
Click 'SPAN'
abc12
abc12
Tutorial
Overwrite if files already exist


Delay
Duration *
00h 00m 20.000s

Click 'I'
Tutorial
Overwrite if fi
The following f

Click 'SPAN id__35'
Upload

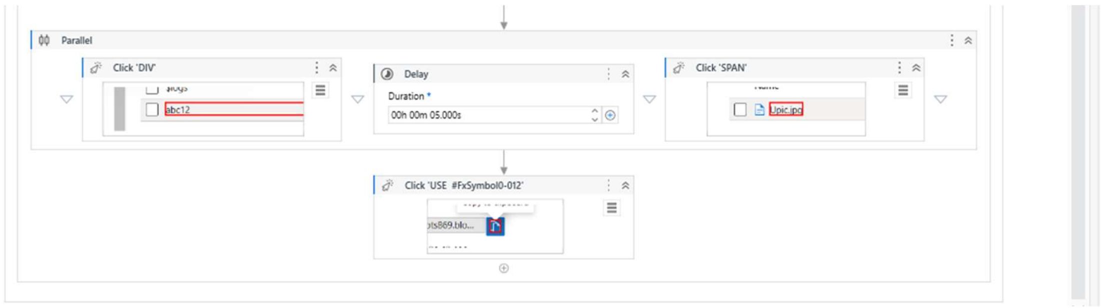
Create container:


Open Browser
Url *
"https://portal.azure.com/#@Mchauhan8052outlook.onmicros
Do
Drop activity here or generate with Autopilot

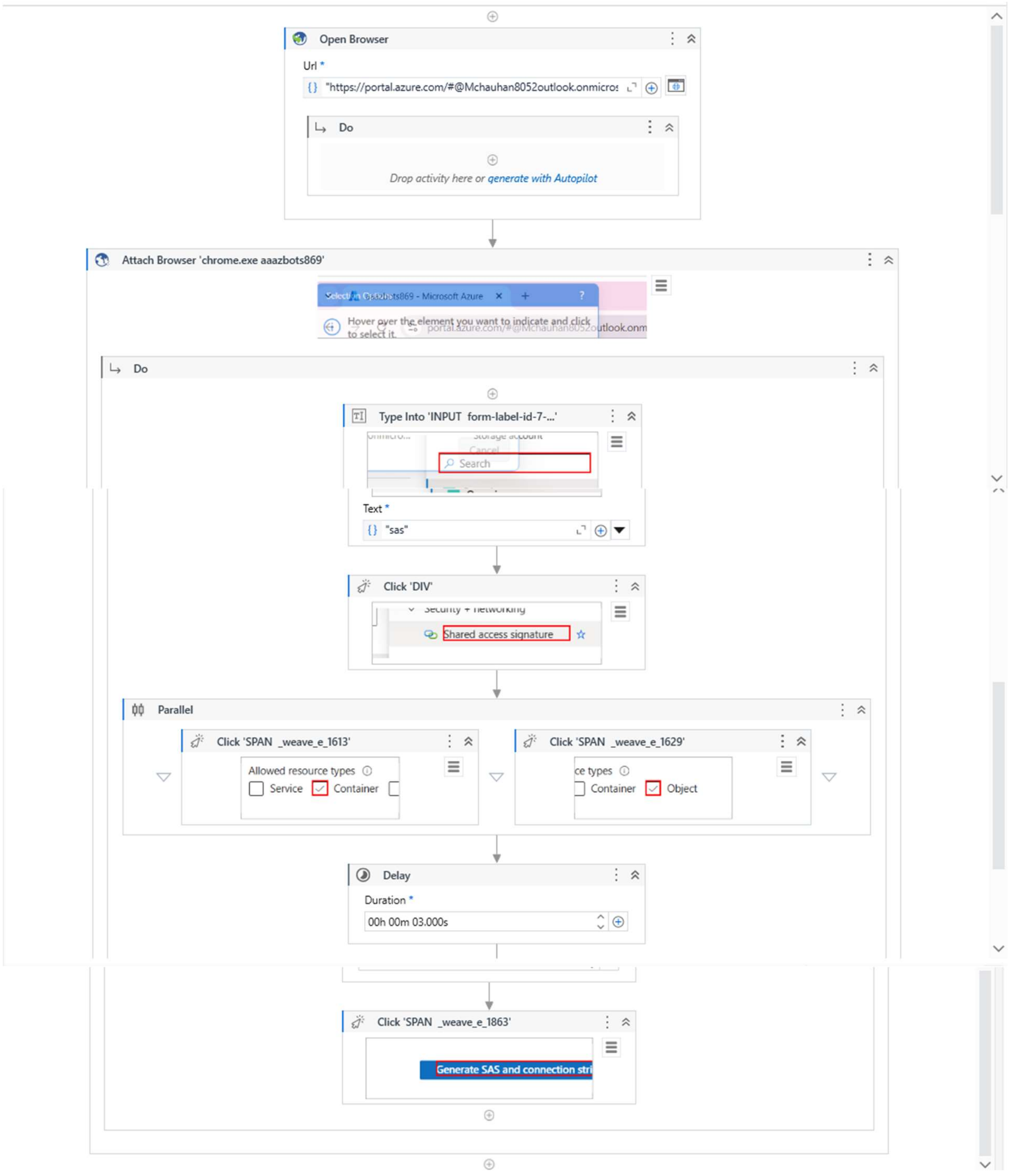
Attach Browser 'chrome.exe aaazbots869'
aaazbots869 - Microsoft Azure
portal.azure.com/#@Mchauhan8052outlook.onm
Do

Type Into 'INPUT form-label-id-7-...'
Search
Text *
"container"

Click 'grouping'
Containers



Create SAS:



Open Browser

Url *

"https://portal.azure.com/#@Mchauhan8052outlook.onmicros

Do

Drop activity here or generate with Autopilot

Attach Browser 'chrome.exe aaazbots869'

Do

Type Into 'INPUT form-label-id-7-...'

Text *

"sas"

Click 'DIV'

Shared access signature

Parallel

Click 'SPAN _weave_e_1613'

Allowed resource types

Service Container

Click 'SPAN _weave_e_1629'

Container Object

Delay

Duration *

00h 00m 03.000s

Click 'SPAN _weave_e_1863'

Generate SAS and connection stri

Conclusion: -

Creating a storage account in Azure using UiPath involves several steps, but it's a straightforward process once you get the hang of it. Here's a brief conclusion of the steps involved:

1. **Set Up Azure**: Ensure you have an Azure account and the necessary permissions to create and manage resources.
2. **Install UiPath Azure Activities**: Add the UiPath Azure Activities package to your UiPath Studio.
3. **Create App Registration in Azure**: Register an app in Azure Active Directory to authenticate your UiPath automation.
4. **Configure Storage Account**: Use the Create Storage Account activity in UiPath to create a new storage account or update an existing one.
5. **Manage Storage Resources**: Use other UiPath activities to manage blobs, containers, and other storage resources within your automation workflows.