Similar procedures

AWS	Azure	Description
s3_client = boto3.client('s3')	blob_service_client = BlobServiceClient.from_connection_string(connect_str)	Creates a low-level service client which is referenced to by name
s3_client.create_bucket(Bucket=bucket_name)	blob_service_client.create_container(container_name)	Creates the containers
s3.Bucket().put_object()	blob_client.upload_blob()	Uploads the file into the specified resource
boto3.resource().create_table()	TableService().create_table()	Creates the tables. For AWS you can provide a schema, name & definitions whereas the azure one takes the name of the table
table.put_item()	table_service.insert_entity()	Takes a formatted data row entry and adds it to the table.
bucket.objects.all()	container_client.list_blobs()	Lists all the objects in the containers
s3.Bucket()	blob_service_client.get_container_client()	Returns the container you want if name is input as a parameter
download_file(bucketName, objName, objName)	download_file.write(blob_client.download_blob().readall())	Downloads the object
boto3.resource().Table(table_name')	table_service = TableService(connection_string)	Retrieves the table

Differences:

Azure doesn’t seem to be able to list all blob containers, whereas aws does.

For filtering, aws gets the filter expression with a Key() function to get the key and a .equals() or .between() to filter on the key. Azure on the other hand, takes in a string similar to an sql query and does not use any functions.