

MIGRATION OF DATA FROM ONE STORAGE ACCOUNT TO ANOTHER.

Establish a secure SSH connection to the jumphost "lrchaosfp01.bankofamerica.com."

```
PS C:\Users\ZKITHYV> ssh lrchaosfp01.bankofamerica.com

This is a private computer system with access restricted to
those with proper authorization.  If you are not specifically
authorized to access data on this system, disconnect now.
All information and communications on this system are
subject to review, monitoring, and recording at any time
without notice or permission.  Unauthorized use or access
may be subject to prosecution or disciplinary action.

Password:
Last login: Tue Aug 13 09:48:22 2024 from 30.177.225.166
RHEL7 Build (rhel7-2024-07)
-sh-4.2$ _
```

Run a bash shell with root privileges within a controlled environment managed by the PowerBroker system.

Enter the id,password and you'll get a security code on authenticator.

pbrun <group> <user=shell>

```
-sh-4.2$ pbrun pbosfdev root=bash

Password:

Security Code: 515782

[root@lrchaosfp01 ~]# _
```

Navigate to /hosting/azure_cli/venv/bin where you'd find the activate script which sets up the environment variables and paths necessary to use the packages installed within the virtual environment. Run the script and configure the system to use a proxy server for HTTPS traffic.

```
[root@lrchaosfp01 ~]# cd /hosting/azure_cli/venv/bin
[root@lrchaosfp01 bin]# pwd
/hosting/azure_cli/venv/bin
[root@lrchaosfp01 bin]# ./activate
[root@lrchaosfp01 bin]# export https_proxy=http://ausdev:z7d8J4X7@appproxy3.bankofamerica.com:8080
[root@lrchaosfp01 bin]#
```

Establish an Azure CLI session using a service principal for authentication, allowing the user to interact with Azure resources using the provided credentials.

```
[root@lrchaosfp01 bin]#
[root@lrchaosfp01 bin]# ./az login --service-principal --username 9f6d92f2-1d8d-41eb-8ab9-3b0d5984a9be --password _..._
--tenant d5106b4b-92ce-4ea1-8f40-4fd0de0717bc
{
  "cloudName": "AzureCloud",
  "homeTenantId": "d5106b4b-92ce-4ea1-8f40-4fd0de0717bc",
  "id": "d8b3cf94-c685-482c-a41e-d9912aabb86d",
  "isDefault": true,
  "managedByTenants": [],
  "name": "sub-corp3-d-71148-apptest",
  "state": "Enabled",
  "tenantId": "d5106b4b-92ce-4ea1-8f40-4fd0de0717bc",
  "user": {
    "name": "9f6d92f2-1d8d-41eb-8ab9-3b0d5984a9be",
    "type": "servicePrincipal"
  }
}
```

Specify the Azure subscription to be used for subsequent operations. Verify the details of the selected subscription by running 'account show' command.

```
[root@lrchaosfp01 bin]# ./az account set --subscription "d8b3cf94-c685-482c-a41e-d9912aabb86d"
[root@lrchaosfp01 bin]# ./az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": "d5106b4b-92ce-4ea1-8f40-4fd0de0717bc",
  "id": "d8b3cf94-c685-482c-a41e-d9912aabb86d",
  "isDefault": true,
  "managedByTenants": [],
  "name": "sub-corp3-d-71148-apptest",
  "state": "Enabled",
  "tenantId": "d5106b4b-92ce-4ea1-8f40-4fd0de0717bc",
  "user": {
    "name": "9f6d92f2-1d8d-41eb-8ab9-3b0d5984a9be",
    "type": "servicePrincipal"
  }
}
[root@lrchaosfp01 bin]#
```

This Azure CLI command is used to list the blobs within a specific container in an Azure Storage account.

```
[root@lrchaosfp01 bin]# ./az storage blob list --container-name testmg --account-name testsaeuscsom1 --output table --auth-mode login
```

Name	Blob Type	Blob Tier	Length	Content Type	Last Modified	Snapshot
monis.txt	BlockBlob	Hot		text/plain	2024-08-12T14:28:36+00:00	
myFile.txt	BlockBlob	Hot		text/plain	2024-08-12T14:26:11+00:00	

```
[root@lrchaosfp01 bin]#
```

This Azure CLI command is used to upload a local file to an Azure Storage Blob container.

```
[root@lrchaosfp01 bin]# ./az storage blob upload --account-name testsaeuscsom1 --container-name testmg --name new-test-file.txt --file new-test-file.txt --auth-mode login
```

```
{
  "client_request_id": "82389690-5992-11ef-a6f7-0090fa6cdce4",
  "content_md5": "1B2M2Y8AsgTpgAmY7PhCfG=",
  "date": "2024-08-13T16:38:45+00:00",
  "encryption_key_sha256": null,
  "encryption_scope": null,
  "etag": "\"0x8DCBB66692D5F5\"",
  "lastModified": "2024-08-13T16:38:45+00:00",
  "request_id": "36e7ebc9-301e-00bd-619f-edee01000000",
  "request_server_encrypted": true,
  "version": "2022-11-02",
  "version_id": "2024-08-13T16:38:45.9108853Z"
}
```

```
[root@lrchaosfp01 bin]#
```

This Azure CLI command is used to list the blobs within a specific container in an Azure Storage account. The output shows a list of blobs within the specified container.

```
[root@lrchaosfp01 bin]# ./az storage blob list --container-name testmg --account-name testsaeuscsom1 --output table --auth-mode login
```

Name	Blob Type	Blob Tier	Length	Content Type	Last Modified	Snapshot
monis.txt	BlockBlob	Hot		text/plain	2024-08-12T14:28:36+00:00	
myFile.txt	BlockBlob	Hot		text/plain	2024-08-12T14:26:11+00:00	
new-test-file.txt	BlockBlob	Hot		text/plain	2024-08-13T16:38:45+00:00	

```
[root@lrchaosfp01 bin]#
```

Script-

```
1 #!/bin/bash
2
3 # Set variables
4 SOURCE_STORAGE_ACCOUNT="monissal"
5 SOURCE_CONTAINER="srcdata"
6 DEST_STORAGE_ACCOUNT="monissa2"
7 DEST_CONTAINER="destdata"
8 FAILED_DIR="failed_uploads"
9
10 # Create a directory for failed uploads
11 mkdir -p $FAILED_DIR
12
13 # Get the list of blobs in the source container
14 blob_list=$(./az storage blob list --account-name $SOURCE_STORAGE_ACCOUNT --container-name $SOURCE_CONTAINER --query "[].name" -o tsv --auth-mode login)
15
16 # Loop through the list of blobs and download each one
17 for blob in $blob_list; do
18     echo "Downloading blob: $blob"
19     ./az storage blob download --account-name $SOURCE_STORAGE_ACCOUNT --container-name $SOURCE_CONTAINER --name $blob --file $blob --auth-mode login
20
21     echo "Uploading blob: $blob to destination storage account"
22     ./az storage blob upload --account-name $DEST_STORAGE_ACCOUNT --container-name $DEST_CONTAINER --name $blob --file $blob --auth-mode login
23
24     # Check if the upload was successful
25     if [ $? -ne 0 ]; then
26         echo "Failed to upload $blob, moving to $FAILED_DIR"
27         mv $blob $FAILED_DIR/
28     else
29         # Optionally delete the downloaded file to save space if upload was successful
30         rm $blob
31     fi
32 done
33
34 echo "All blobs have been transferred."
35
```

This Bash script transfers blobs from one Azure Storage container to another.

1) Sets up: Defines storage account names, container names, and a directory for failed uploads.

2) Gets blob list: Retrieves a list of blobs from the source container.

3) Loops through blobs:

- Downloads each blob to the local machine.
- Uploads the downloaded blob to the destination container.
- If upload fails, moves the blob to the failed uploads directory.
- If upload succeeds, optionally deletes the downloaded file.

4) Completes: Indicates all blobs have been transferred.