

Java Mainframe Tools - Zfile4aws

V1R5 BETA VERSION

USER GUIDE

Contents

Introduction	2
Environment requirements	3
Usage	4
Configuration file	6
Configuration file Hints and tips:	8
Functions	9
GET is used to retrieve a file from a cloud object in an AWS S3 bucket.	9
PUT is used to store file as a cloud object in an AWS S3 bucket.	10
DELETE is used to remove object from an AWS S3 bucket.	11
LIST is used to list object(s) from an AWS S3 bucket.	12
ENC is used to encrypt configuration file.	13
Environment variables	14
Special -opt parameter (options)	14
Using remote mode, by using ' -opt r' on put and get functions.	15
Examples	16
Example 1a – Configuration file to a MINIO server	16
Example 1b – Configuration file to an AWS server	17
Example 1c – Remote mode configuration file to a MINIO server	18
Example 2 – PUT job (Upload z/OS file to a cloud bucket)	19
Example 3 – LIST job (List cloud objects)	20
Example 4 – GET job (download or restore cloud objects to z/OS ds)	21
Example 5 – DELETE job (delete object(s) from cloud bucket)	22
Example 6 – Running from linux (remote mode execution)	23
*** License usage Warning Notice ***	25
How to order a Zfile4aws BETA TEST COPY	25
Copyright Statement	25
Glossary of terms	26

Introduction

Zfile4aws serves as a Java mainframe utility designed for the transmission and retrieval of z/OS sequential files within an AWS S3 cloud environment. This tool accommodates various data sources sequential file types and formats.

One of Zfile4aws's notable features is its compatibility with tape files such as DFDSS backups and LBI-type datasets, which typically have large physical block sizes. When utilizing Zfile4aws, sequential data can undergo compression, encryption, and segmentation prior to its transfer to the cloud.

Theoretically, there is no imposed limitation on the dataset size that can be dispatched to the cloud. However, practical constraints may arise based on specific environmental factors and the policies of your chosen cloud service provider.

It's essential to note that Zfile4aws exclusively operates in batch mode (JZOS). Additionally, the utility extends support to various OMVS file types, like HFS and ZFS.

Environment requirements

- At least z/OS 2.2.
- At least JAVA V8.
- IBM Crypto hardware is recommended but not required.
- IBM Compression feature is recommended but not required.
- IBM CPACF crypto enablement is required.
- ZIIP processors are also recommended but not required.
- Basic knowledge over modern z/OS function capability is a good starting point.
- Be familiar on how to run JAVA on Z/OS and JCL basic knowledge are also mandatory.
- Be familiar with AWS S3 cloud service and terms.
- JES2 Option SYSSYM=ALLOW is required in the job class definition.

Usage

Zfile4aws provides five key operations for accessing data within the cloud:

1. GET Function:

- Purpose: Retrieve and download data from an S3 bucket. Object retrieved must have been created using the PUT function.

2. PUT Function:

- Purpose: Upload and store data to an S3 bucket.
- Process: Before sending a z/OS dataset to the cloud, Zfile4aws reads and converts the input data to an internal image format. This format supports compression, encryption, and segmentation. The internal image format is then stored in a temporary ZFS file, referred to as the Workspace. The creation of this internal image format is facilitated by the PUT function.

3. LIST Function:

- Purpose: List object information from an S3 bucket.

4. DELETE Function:

- Purpose: Delete cloud object(s) from an S3 bucket.

5. ENC Function:

- Purpose: Encrypt all parameters stored in the Configuration file.

To execute these operations, all necessary configuration parameters are consolidated in a single OMVS text file known as the Configuration File.

Before initiating the transfer of any z/OS dataset to the cloud, it's essential to allocate sufficient ZFS space to accommodate the workspace requirements for the given environment.

The Workspace, identified as the temporary ZFS space, is where the internal image format is stored during the PUT function process. It is crucial for users to predetermine an adequate amount of ZFS space to fulfill the temporary space requirements.

Access to cloud objects is regulated by validating RACF (Resource Access Control Facility) access to the corresponding z/OS dataset name. This ensures controlled and secure access to cloud objects.

By example, suppose you want to create an object by using PUT function to store a z/OS dataset to a S3 bucket. Access to do so is permitted by validating if the current RACF “userid” has ALTER authority to the corresponding dataset name.

In the same idea, to retrieve an object by using the GET function. The current user must have a RACF READ authority to restore and retrieve a z/OS dataset image from an S3 bucket.

Configuration file

Parameters are specified as follow in a single OMVS text file.

FilePath: Indicate full path name to the ZFS Workspace.

FileCert: Indicate full path name of the java key ring to use for https and ftps request.

FileCertPw: JAVA key ring password.

AwsAccKey: Indicate the AWS S3 access key to use.

AwsSecret: Indicate the AWS S3 secret key to use.

AwsRegion: Set the AWS region.

AwsBucket: Set the AWS S3 bucket name.

AwsHost: IP adr or end point name of the AWS S3 server.

AwsPort: Server IP port number.

Compress: Yes | No Used to Indicate if File image must be compressed.

Replace: Yes | No Used to replace any existing file or object by default.

ZOSSegsize: nnnn segment size in Megabytes (1024 = 1g).

ZosMaxblk: 1 | 2 | 3 Physical max blksize of input data source.

- 1 = 262144 bytes (LBI support 256k)
- 2 = 65535 64k
- 3 = 32760 32k (dasd device)

ZosHost: Z/OS FTP and RSHD server IP address (ref: opt -r).

ZosFtpPort: Z/OS FTP port number (ref: opt -r).

ZosUserid: Z/OS FTP userid and RSHD client userid (ref: opt -r).

ZosPassword: Z/OS userid password.

ZosFtpJobP: Remote jobname prefix name, 3 to 7 chars max (ref: opt -r).

JobOutPath: Full path dir. where to store remote jobs output (ref: opt -r).

KeyName: AES Key name to used to encrypt data.

KeyStore: Full path name of the PKCS12 encrypting key store.

KeyStorePw: Key store password.

OutAlloc: Used to set 'bpxwdyn' dynamic allocation parameters.

Configuration file Hints and tips:

- If you prefix the "AwsHost" parameter by 'https://', that means to use a secured HTTPS connection. The 'https://' prefix indicates that the communication between the client (in this case, Zfile4aws) and the server (AWS S3) is encrypted and secured.
- Prefixing "ZosHost" parameter by 'ftps://' means that z/OS ftp server use SSL secured connection.
- Specifying "KeyStore ICSF" means to Zfile4aws that key used to do encryption reside in ICSF CKDS.
- An Asterix (*) in Column1 stands for a comment line.
- User must set OMVS config file security attributes as is own desired access HFS security policies.
- Configuration file name must be specified by using the **-cfg** parameter when invoking a specific function. It also can be specified through the environment variable: **Zfile4aws_cfg**.
- If you enter a question mark (?) in the ZosPassword parameter, it indicates that you will be prompted to input password for the ZosUserid.
- Since remote shell exec function is used when using the remote option (-opt r). ZosHost parameter is used to determine the IP address of the z/OS FTP server as the z/OS RSHD server.
- To set the OutAlloc parameter, refer to the appropriate IBM documentation to get details about bpxwdyn function.

Functions

GET is used to retrieve a file from a cloud object in an AWS S3 bucket.

```
-oper get -fn //Object-dsn | /Object-omvsfile [-cfg /Config-filename]  
[-opt Options] -out //Output-dsn | /Output-omvsfile
```

Parameters:

//Object-dsn:

Specific cloud object name to retrieve, cloud object is a z/OS file image.

To retrieve a cloud object, RACF READ access is required to the corresponding z/OS file.

/Object-omvsfile:

Specific cloud object name to retrieve, cloud object is a OMVS file image.

To retrieve a cloud object, READ access to the corresponding OMVS file full path name is required.

/Config-filename:

Full path name of the OMVS file containing all configuration parameters.

Look at the section: *Environment variables*.

//Output-dsn:

Zos dataset name where the retrieved data is stored. To specify a ddname instead of a dsname, use the format //DD:ddname.

/Output-omvsfile:

Full path name of the output OMVS file where the retrieved data is stored.

Options:

Look at the section: *Special -opt parameter*.

PUT is used to store file as a cloud object in an AWS S3 bucket.

-oper put | putr -fn //Dsnname | /Omvsfile [-cfg /Config-filename]
[-opt Options]

Parameters:

//Dsnname:

Zos dataset name to store to the cloud. To specify a ddname instead of dsname, use the format *//DD:ddname*. User must have WRITE access to the input dsn to be able to store object with the same name.

/Omvsfile:

Full path name of the OMVS file to store to the cloud. User must have WRITE access to the input file to be able to store object with the same name.

/Config-filename:

Full path name of the OMVS file containing all configuration parameters.

Look at the section: *Environment variables*.

Options:

Look at the section: *Special -opt parameter*.

Notes:

Use function PUTR to execute a PUT with 'replace' if you want to replace any existing cloud object. Go to see 'Replace' configuration parameters.

DELETE is used to remove object from an AWS S3 bucket.

-oper delete -fn *Objectname* [-cfg */Config-filename*]

or

-oper delete -fp *Objectname-prefix* [-cfg */Config-filename*]

Parameters:

Objectname:

Specific cloud object name to delete.

Objectname-prefix:

All cloud objects having this name prefix will be deleted.

/Config-filename:

Full path name of the OMVS file containing all configuration parameters.

Look at the section: *Environment variables*.

Notes:

To delete an object, it is required that the user have ALTER or WRITE access authority to the corresponding file or dataset.

LIST is used to list object(s) from an AWS S3 bucket.

-oper list [-fp Object-name-prefix] [-cfg /Config-filename]

Parameters:

Object-name-prefix:

All cloud objects having this name prefix will be listed.

/Config-filename:

Full path name of the OMVS file containing all configuration parameters.

Look at the section: *Environment variables*.

ENC is used to encrypt configuration file.

-oper enc -fn /Encrypted-config-filename -cfg /Config-filename

Parameters:

/Config-filename:

Full path name of the OMVS file containing all configuration parameters.

Look at the section: *Environment variables*.

/Encrypted-config-filename:

Full path name of the OMVS output file containing all configuration parameters in an encrypted form.

Notes:

Since configuration file contains many sensitive information it might be required to run with an encrypted version of this file. This will secure installation where the config file is shared among many users.

When using the encrypted form of a configuration file it is required to set option as -opt e.

Environment variables.

Zfile4aws_cfg

Set this variable to the full path name of the configuration file.

Zfile4aws_opt

Set this variable to the default desired options.

-Dcom.ibm.jzos.compression.type

JVM system variable used by JZOS to set specific compression method.

Look to appropriate IBM documentation to set desired value.

Default method set is GZIP.

Special **-opt** parameter (options).

- opt e Indicate that the config file is encrypted (ref: enc function).
- opt k No auto deletion of the file image stored in the ZFS workspace.
- opt r Activate the remote mode. Go to see special section about this mode.
- opt v Verbose option to get more messages with timestamp.

Using remote mode, by using '-opt r' on put and get functions.

With remote mode it is possible to execute compressing and encrypting works on other platform than z/OS. You may want to execute those CPU intensive tasks on a Linux or Windows instead of consuming mainframe CPU resource.

When remote mode request starts on Linux/Windows a batch job is submitted to z/OS, this job will oversee reading or writing data only and transmit/receive data thru FTP to the Linux/Windows process. All other tasks like compress and encrypt data are executed on the running remote Java program on Linux/Windows platform.

Sending and receiving data from the cloud is also executed on Linux/Windows platform.

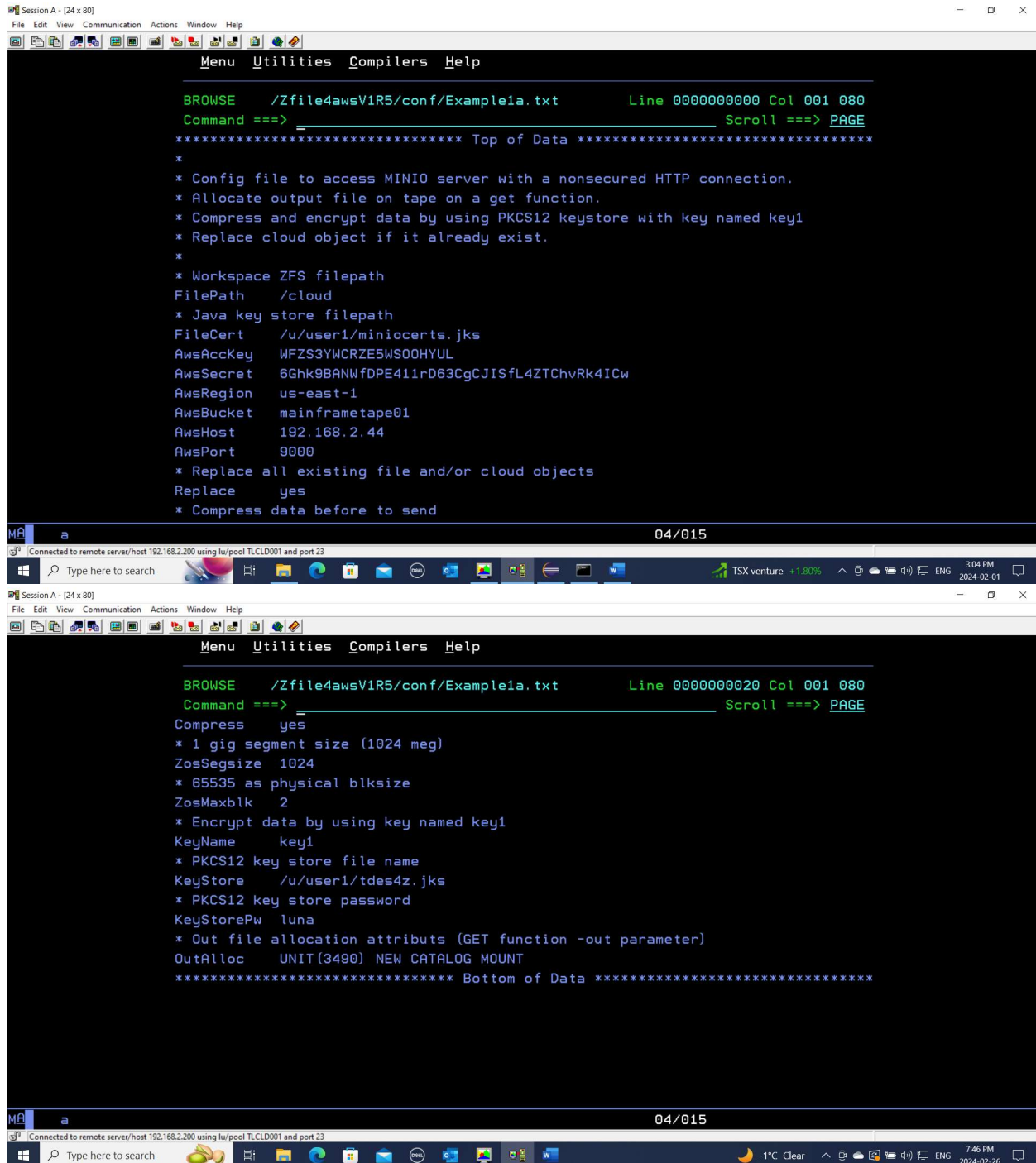
Since, mainframe data transport is made thru FTP only. You may use z/OS FTP SSL to implement a secure connection for data transport between z/OS and the running Java program running on Linux/Windows platform.

Remote mode requires that 'Remote shell exec server' be implemented on z/OS (RSHD server).

JOB submitted by the remote shell process are executed using ZosUserid config parm as the remote shell client userid.

Examples

Example 1a – Configuration file to a MINIO server



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

Menu Utilities Compilers Help

BROWSE /Zfile4awsV1R5/conf/Example1a.txt Line 0000000000 Col 001 080
Command ==> Scroll ==> PAGE

***** Top of Data *****
*
* Config file to access MINIO server with a nonsecured HTTP connection.
* Allocate output file on tape on a get function.
* Compress and encrypt data by using PKCS12 keystore with key named key1
* Replace cloud object if it already exist.
*
* Workspace ZFS filepath
FilePath /cloud
* Java key store filepath
FileCert /u/user1/miniocerts.jks
AwsAccKey WFS3YVCRZE5WS00HYUL
AwsSecret 6Ghk9BANWfdPE411rD63CgCJISfL4ZTChvRk4ICw
AwsRegion us-east-1
AwsBucket mainframetape01
AwsHost 192.168.2.44
AwsPort 9000
* Replace all existing file and/or cloud objects
Replace yes
* Compress data before to send

04/015
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD001 and port 23
Type here to search TSX venture +1.80% 3:04 PM 2024-02-01

Session A - [24 x 80]
File Edit View Communication Actions Window Help

Menu Utilities Compilers Help

BROWSE /Zfile4awsV1R5/conf/Example1a.txt Line 0000000020 Col 001 080
Command ==> Scroll ==> PAGE
Compress yes
* 1 gig segment size (1024 meg)
ZosSegsize 1024
* 65535 as physical blksize
ZosMaxblk 2
* Encrypt data by using key named key1
KeyName key1
* PKCS12 key store file name
KeyStore /u/user1/tdes4z.jks
* PKCS12 key store password
KeyStorePw luna
* Out file allocation attributs (GET function -out parameter)
OutAlloc UNIT(3490) NEW CATALOG MOUNT
***** Bottom of Data *****

04/015
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD001 and port 23
Type here to search -1°C Clear 7:46 PM 2024-02-26
```

Example 1b – Configuration file to an AWS server

The image displays two screenshots of a z/OS console session, likely from a mainframe environment, showing the configuration of an AWS connection file. The session is titled "Session A - [24 x 80]" and shows a menu with options: Menu, Utilities, Compilers, and Help.

Top Screenshot: The user has browsed to the file `/Zfile4awsV1R5/conf/Example1b.txt`. The command prompt shows `Command ==>` and `Scroll ==> PAGE`. The file content is displayed, starting with a header line: `***** Top of Data *****`. The configuration details are as follows:

- * Config file to access AWS site with a secured HTTPS connection.
- * Allocate output file on dasd on a get function.
- * Compress and encrypt data by using ICSF key ICSF.TDES.KEY02.
- * Do not replace cloud object if it already exist.
- *

The configuration parameters are listed below:

Parameter	Value
FilePath	/cloud
FileCert	/usr/lpp/java/J8.0/lib/security/cacerts
AwsAccKey	BKJAX3W3VUXWES9SSTY1
AwsSecret	uHhnMdt8a1zkok88Eyma0Tq5hoQ84mNG1CAUS/Z1
AwsRegion	ca-central-1
AwsBucket	bucketzosfiles01
AwsHost	https://s3.ca-central-1.amazonaws.com
AwsPort	443
Compress	yes
Replace	no
Keyname	ICSF.TDES.KEY02
KeyStore	ICSF
ZosSegsize	1000

The bottom of the console shows the status bar with `04/015`.

Bottom Screenshot: The user has browsed to the file `/Zfile4awsV1R5/conf/Example1b.txt`. The command prompt shows `Command ==>` and `Scroll ==> PAGE`. The file content is displayed, starting with a header line: `***** Bottom of Data *****`. The configuration details are as follows:

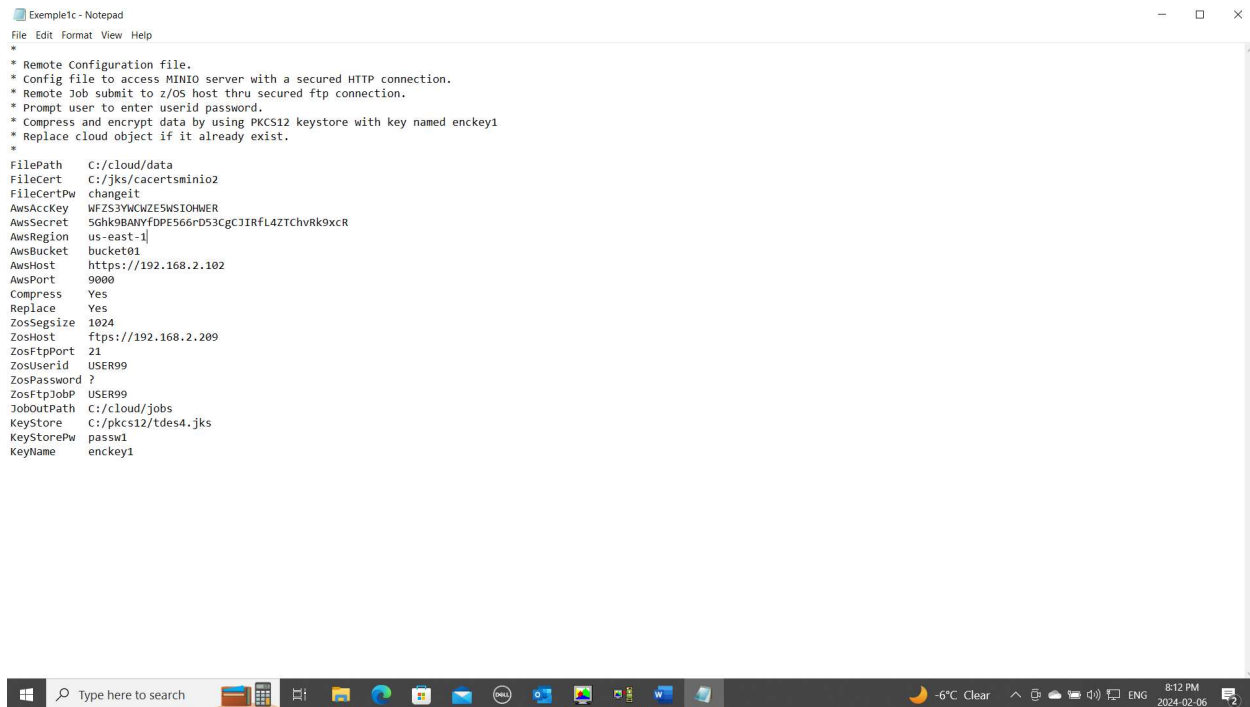
- *

The configuration parameters are listed below:

Parameter	Value
ZosMaxblk	2
OutAlloc	CYL SPACE(100,50) UNIT(3390) RELEASE NEW CATALOG

The bottom of the console shows the status bar with `04/015`.

Example 1c – Remote mode configuration file to a MINIO server



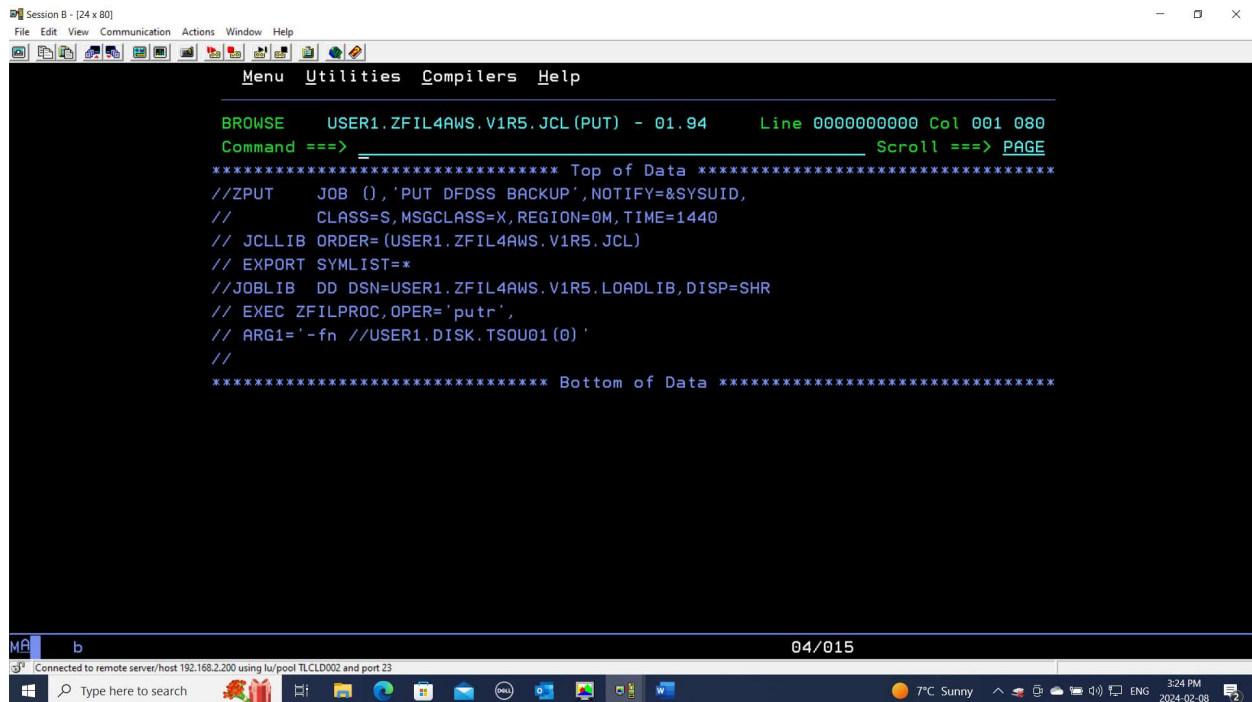
```
Exemple1c - Notepad
File Edit Format View Help

*
* Remote Configuration file.
* Config file to access MINIO server with a secured HTTP connection.
* Remote job submit to z/OS host thru secured ftp connection.
* Prompt user to enter userid password.
* Compress and encrypt data by using PKCS12 keystore with key named enckey1
* Replace cloud object if it already exist.
*
FilePath      C:/cloud/data
FileCert      C:/jks/cacertsminio2
FileCertPw    changeit
AwsAccKey     WfZ53YmCWZE5WSIOHMER
AwsSecret     5Ghks9BANYfDPE566rD53cgCJIRfL42TChvRk9xcR
AwsRegion     us-east-1
AwsBucket     bucket01
AwsHost       https://192.168.2.102
AwsPort       9000
Compress      Yes
Replace       Yes
ZosSegsize    1024
ZosHost       ftps://192.168.2.209
ZosFtpPort    21
ZosUserId     USER99
ZosPassword   ?
ZosFtpJobP    USER99
JobOutPath    C:/cloud/jobs
KeyStore      C:/pkcs12/tdes4.jks
KeyStorePw    passw1
KeyName       enckey1
```

Windows taskbar at the bottom shows the search bar, taskbar icons, system tray with weather (-6°C Clear), and date/time (8:12 PM, 2024-02-06).

Example 2 – PUT job (Upload z/OS file to a cloud bucket)

JCL used to upload DFDDS tape file 'USER1.DISK.TSOU01(0)':

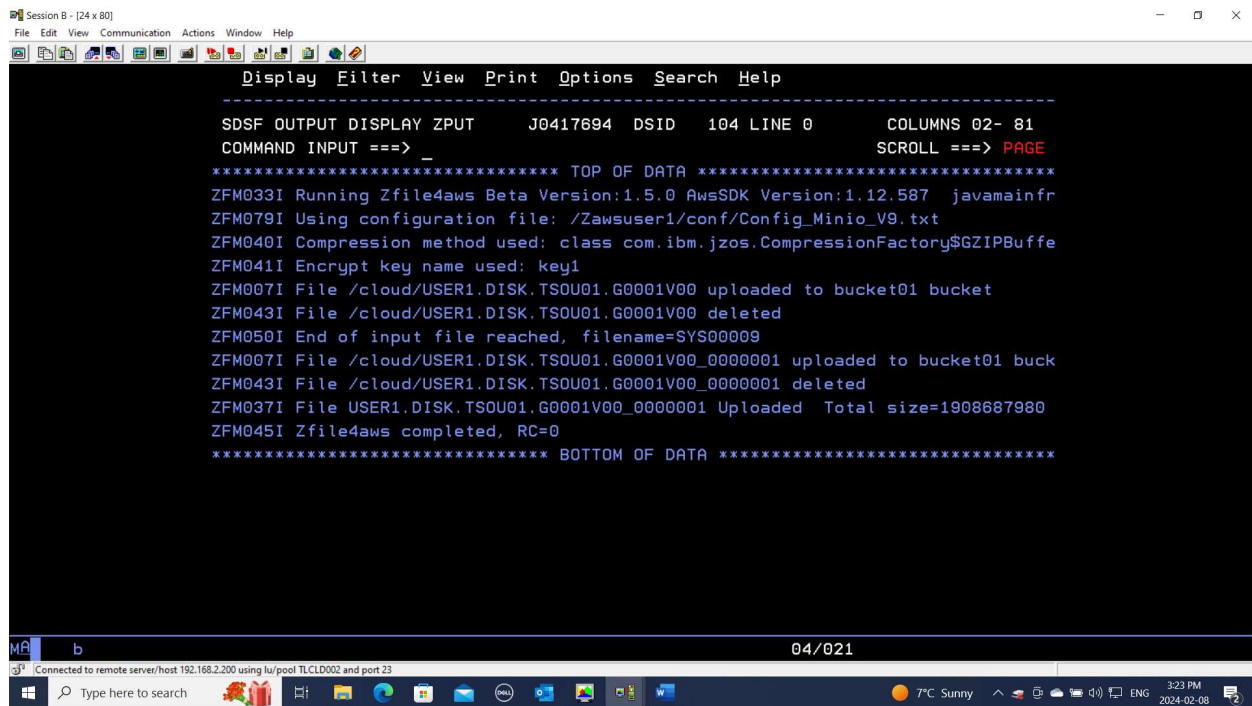


The screenshot shows a z/OS JCL editor window titled 'Session B - [24 x 80]'. The menu bar includes 'Menu', 'Utilities', 'Compilers', and 'Help'. The main text area displays a JCL job definition for a PUT operation. The job is named 'USER1.ZFIL4AWS.V1R5.JCL (PUT)' and is on line 0000000000, column 001, row 080. The command is 'BROWSE'. The JCL code is as follows:

```
BROWSE      USER1.ZFIL4AWS.V1R5.JCL (PUT)  - 01.94      Line 0000000000 Col 001 080
Command ==> _____ Scroll ==> PAGE
***** Top of Data *****
//ZPUT      JOB (), 'PUT DFDDS BACKUP', NOTIFY=&SYSUID,
//          CLASS=S, MSGCLASS=X, REGION=0M, TIME=1440
// JCLLIB   ORDER= (USER1.ZFIL4AWS.V1R5.JCL)
// EXPORT   SYMLIST=*
//JOBLIB    DD DSN=USER1.ZFIL4AWS.V1R5.LOADLIB, DISP=SHR
// EXEC     ZFILPROC, OPER='putr',
// ARG1='-fn //USER1.DISK.TSOU01(0) '
//
***** Bottom of Data *****
```

The status bar at the bottom shows '04/015' and a connection message: 'Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23'.

JOB output result:



The screenshot shows a z/OS JCL editor window titled 'Session B - [24 x 80]'. The menu bar includes 'Display', 'Filter', 'View', 'Print', 'Options', 'Search', and 'Help'. The main text area displays the output of the PUT job. The output is as follows:

```
-----
SDSF OUTPUT DISPLAY ZPUT      J0417694  DSID   104 LINE 0      COLUMNS 02- 81
COMMAND INPUT ==> _____ SCROLL ==> PAGE
***** TOP OF DATA *****
ZFM033I Running Zfile4aws Beta Version:1.5.0 AwsSDK Version:1.12.587 javamainfr
ZFM079I Using configuration file: /Zawsuser1/conf/Config_Minio_V9.txt
ZFM040I Compression method used: class com.ibm.jzos.CompressionFactory$GZIPBuffe
ZFM041I Encrypt key name used: key1
ZFM007I File /cloud/USER1.DISK.TSOU01.G0001V00 uploaded to bucket01 bucket
ZFM043I File /cloud/USER1.DISK.TSOU01.G0001V00 deleted
ZFM050I End of input file reached, filename=SYS00009
ZFM007I File /cloud/USER1.DISK.TSOU01.G0001V00_0000001 uploaded to bucket01 buck
ZFM043I File /cloud/USER1.DISK.TSOU01.G0001V00_0000001 deleted
ZFM037I File USER1.DISK.TSOU01.G0001V00_0000001 Uploaded Total size=1908687980
ZFM045I Zfile4aws completed, RC=0
***** BOTTOM OF DATA *****
```

The status bar at the bottom shows '04/021' and a connection message: 'Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23'.

Example 3 – LIST job (List cloud objects)

JCL used to list objects USER1.DISK.**

```
Session B - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

BROWSE USER1.ZFIL4AWS.V1R5.JCL (LIST) - 01.13 Line 0000000000 Col 001 080
Command ==> Scroll ==> PAGE
***** Top of Data *****
//ZLIST JOB (), 'LIST OBJECT', NOTIFY=&SYSUID, 00010000
// CLASS=S, MSGCLASS=X, REGION=0M, TIME=1440 TYPRUN=SCAN 00020000
// *
// JCLLIB ORDER=(USER1.ZFIL4AWS.V1R5.JCL)
// EXPORT SYMLIST=*
//JOB LIB DD DSN=USER1.ZFIL4AWS.V1R5.LOADLIB, DISP=SHR
// EXEC ZFILPROC, OPER='list -fp USER1.DISK' 00040000
//
***** Bottom of Data *****

04/015
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23
Type here to search 7°C Sunny 3:31 PM 2024-02-08
```

JOB output result:

```
Session B - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY ZLIST J0417757 DSID 104 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> SCROLL ==> PAGE
***** TOP OF DATA *****
ZFM033I Running Zfile4aws Beta Version:1.5.0 AwsSDK Version:1.12.587 javamainfr
ZFM079I Using configuration file: /Zawsuser1/conf/Config_Minio_V9.txt
- USER1.DISK.TSOU01.G0001V00 (size = 1 GB) Wed Feb 07 22:33:13 EST 2024
- USER1.DISK.TSOU01.G0001V00_0000001 (size = 796 MB) Wed Feb 07 23:16:33 EST 2
ZFM045I Zfile4aws completed, RC=0
***** BOTTOM OF DATA *****

04/021
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23
Type here to search 7°C Sunny 3:36 PM 2024-02-08
```

Example 4 – GET job (download or restore cloud objects to z/OS ds)

JCL used to get object copy of USER1.DISK.TSOU01.G0001V00

```
Session B - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

BROWSE USER1.ZFIL4AWS.V1R5.JCL(GET) - 01.82 Line 0000000000 Col 001 080
Command ==> _ Scroll ==> PAGE
***** Top of Data *****
//ZGET JOB (), 'GET OBJECT', NOTIFY=&SYSUID, 00010000
// CLASS=S, MSGCLASS=X, REGION=0M, TIME=1440 00020000
// JCLLIB ORDER=(SYS2.ZFIL4AWS.V1R5.JCL)
// EXPORT SYMLIST=*
//JOB LIB DD DSN=SYS2.ZFIL4AWS.V1R5.LOADLIB, DISP=SHR
// EXEC ZFILPROC, OPER='get', 00040000
// ARG1='-fn //USER1.DISK.TSOU01.G0001V00', 00040000
// ARG2='-out //USER1.DRCOPY.TSOU01(+1)' 00040000
// 00040000
***** Bottom of Data *****

04/015
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23
Type here to search Earnings upcoming 7:10 PM 2024-02-08
```

JOB output result:

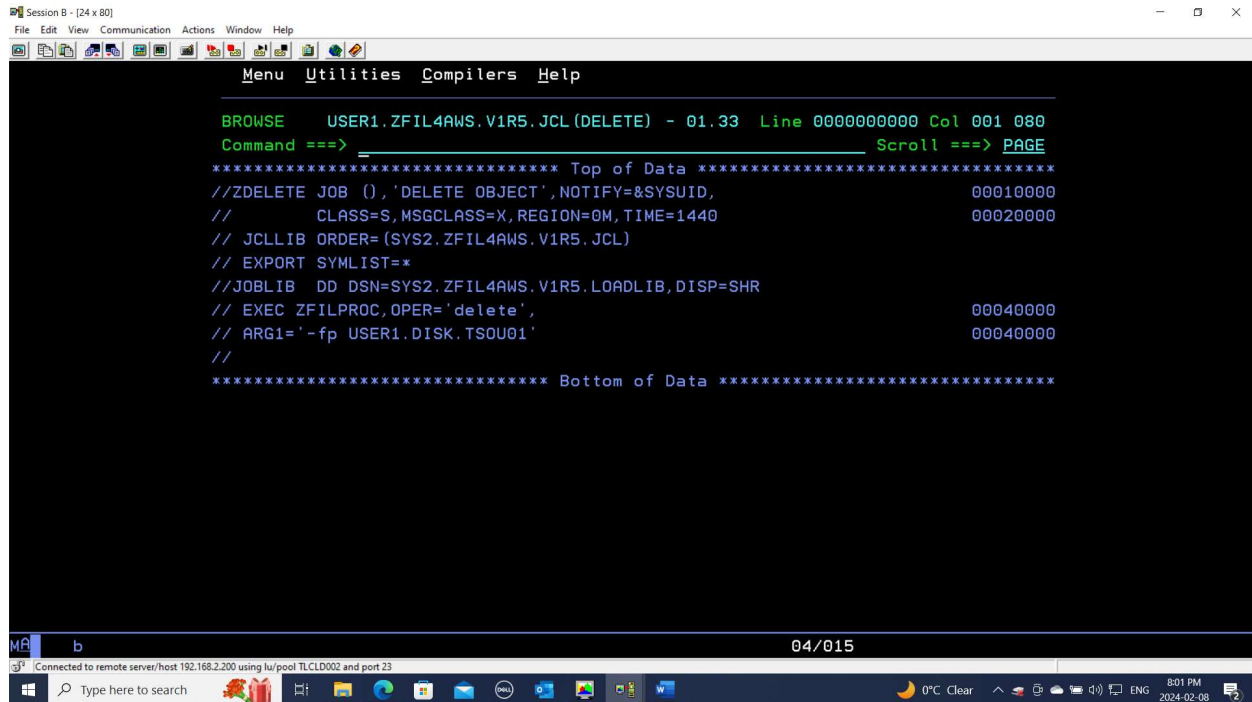
```
Session B - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY ZGET J0417760 DSID 104 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****
ZFM033I Running Zfile4aws Beta Version:1.5.0 AwsSDK Version:1.12.305 javamainfr
ZFM079I Using configuration file: /Zfile4awsV1R5/conf/Config_Minio_V9.txt
ZFM040I Compression method used: class com.ibm.jzos.CompressionFactory$GZIPBuffer
ZFM041I Encrypt key name used: key1
ZFM006I File /cloud/USER1.DISK.TSOU01.G0001V00 downloaded from bucket01 bucket
ZFM006I File /cloud/USER1.DISK.TSOU01.G0001V00_00000001 downloaded from bucket01
ZFM043I File /cloud/USER1.DISK.TSOU01.G0001V00 deleted
ZFM043I File /cloud/USER1.DISK.TSOU01.G0001V00_00000001 deleted
ZFM054I Output file closed, filename=SYS00009
ZFM039I File image: USER1.DISK.TSOU01.G0001V00 rebuild completed Total size=190
ZFM047I Object USER1.DISK.TSOU01.G0001V00 restored to file USER1.DRCOPY.TSOU01.G
ZFM045I Zfile4aws completed, RC=0
***** BOTTOM OF DATA *****

04/021
Connected to remote server/host 192.168.2.200 using lu/pool TLCLD002 and port 23
Type here to search Cold weather 7:12 PM 2024-02-08
```


Example 5 – DELETE job (delete object(s) from cloud bucket)

JCL used to delete objects USER1.BACKUP.LINKLIB.**

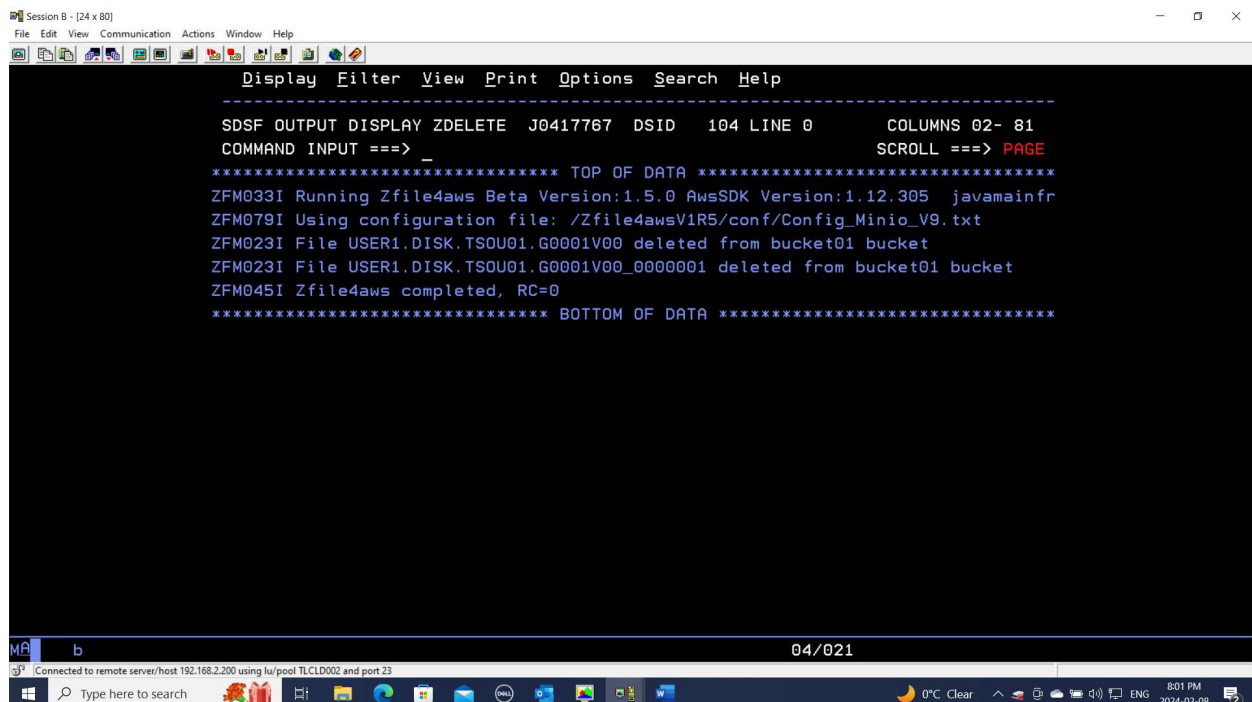


The screenshot shows a terminal window titled "Session B - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main display area shows the following JCL code:

```
BROWSE      USER1.ZFIL4AWS.V1R5.JCL (DELETE) - 01.33  Line 0000000000 Col 001 080
Command ==> _____ Scroll ==> PAGE
***** Top of Data *****
//ZDELETE JOB (), 'DELETE OBJECT', NOTIFY=&SYSUID,          00010000
//          CLASS=S, MSGCLASS=X, REGION=0M, TIME=1440      00020000
// JCLLIB ORDER=(SYS2.ZFIL4AWS.V1R5.JCL)
// EXPORT SYMLIST=*
//JOB LIB DD DSN=SYS2.ZFIL4AWS.V1R5.LOADLIB, DISP=SHR
// EXEC ZFILPROC, OPER='delete',                          00040000
// ARG1='-fp USER1.DISK.TSOU01'                          00040000
//
***** Bottom of Data *****
```

The terminal window has a status bar at the bottom showing "04/015" and a connection status "Connected to remote server/host 192.168.2.200 using lu/pool TLCD002 and port 23".

JOB output result:



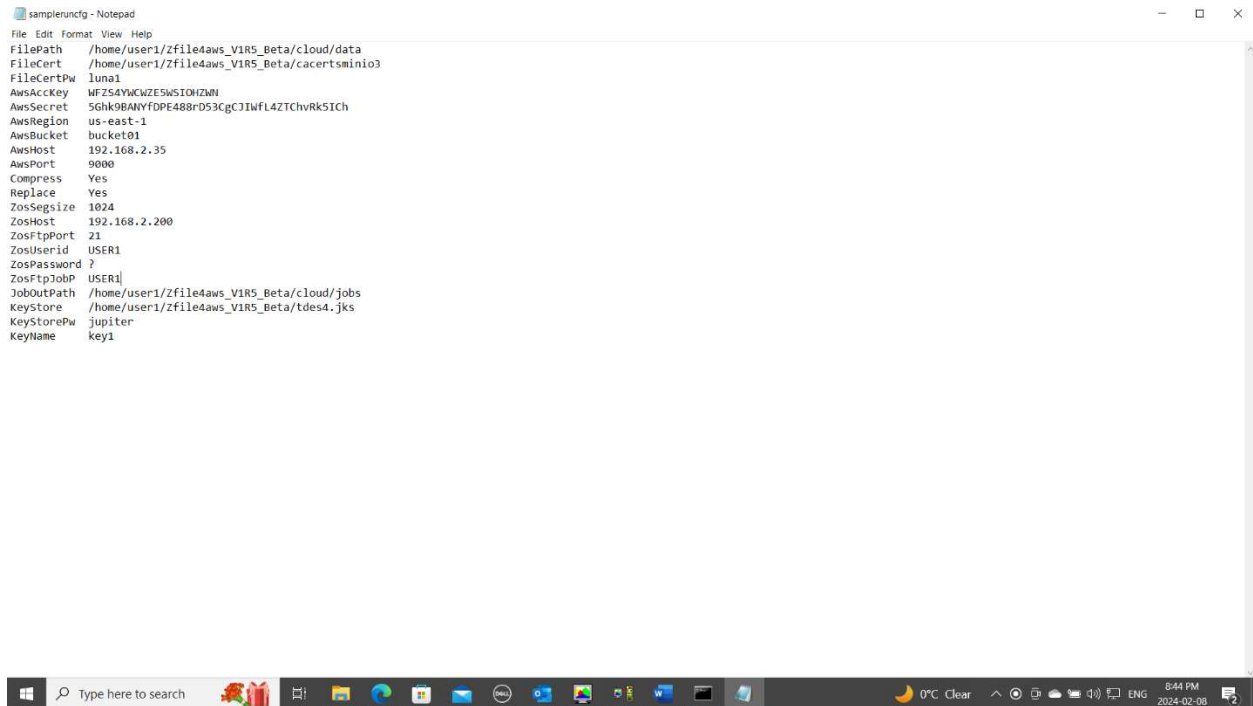
The screenshot shows a terminal window titled "Session B - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main display area shows the following output:

```
Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY ZDELETE J0417767 DSID 104 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> _____ SCROLL ==> PAGE
***** TOP OF DATA *****
ZFM033I Running Zfile4aws Beta Version:1.5.0 AwsSDK Version:1.12.305 javamainfr
ZFM079I Using configuration file: /Zfile4awsV1R5/conf/Config_Minio_V9.txt
ZFM023I File USER1.DISK.TSOU01.G0001V00 deleted from bucket01 bucket
ZFM023I File USER1.DISK.TSOU01.G0001V00_0000001 deleted from bucket01 bucket
ZFM045I Zfile4aws completed, RC=0
***** BOTTOM OF DATA *****
```

The terminal window has a status bar at the bottom showing "04/021" and a connection status "Connected to remote server/host 192.168.2.200 using lu/pool TLCD002 and port 23".

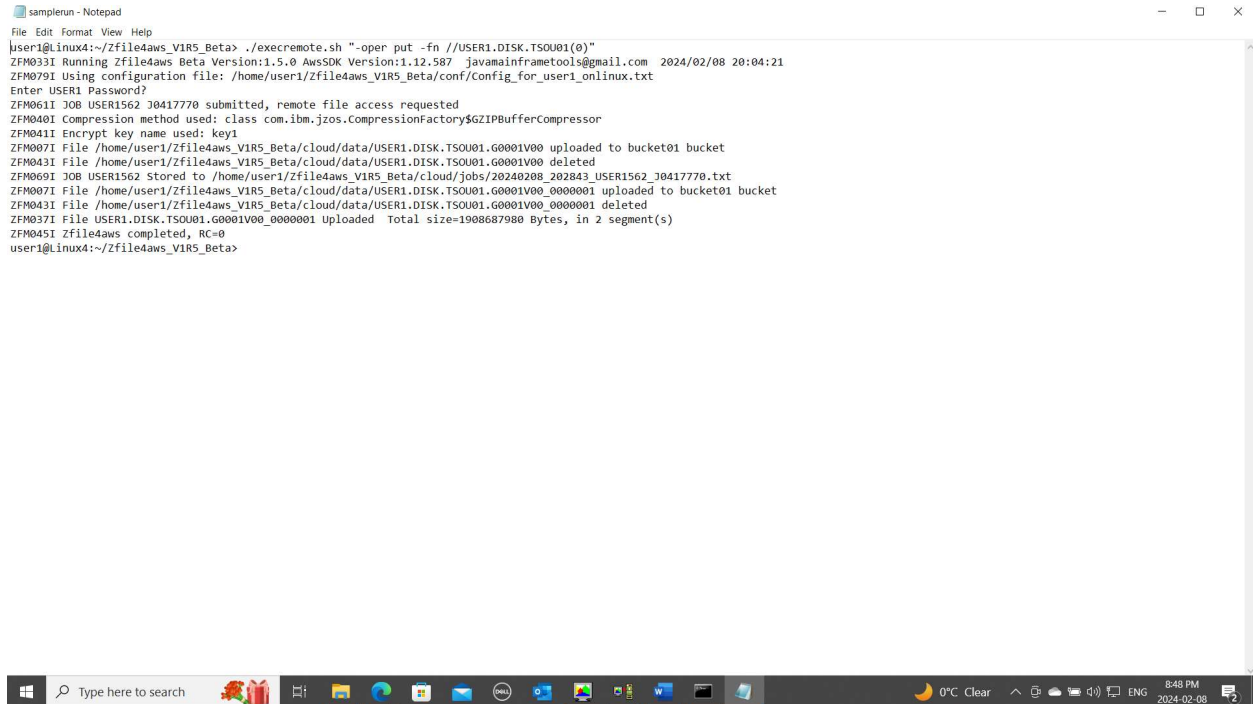
Example 6 – Running from linux (remote mode execution)

Config file used on linux:



```
samplefuncfg - Notepad
File Edit Format View Help
FilePath /home/user1/zfile4aws_V1R5_Beta/cloud/data
FileCert /home/user1/zfile4aws_V1R5_Beta/cacertsminio3
FileCertPw luna1
AwsAccKey MFZ54YMCWZE5WSIOHZWN
AwsSecret 5Ghk9BANYFDPE488rD53CgCJIWfL4ZTChvRk5TCh
AwsRegion us-east-1
AwsBucket bucket01
AwsHost 192.168.2.35
AwsPort 9000
Compress Yes
Replace Yes
ZosSegsSize 1024
ZosHost 192.168.2.200
ZosFtpPort 21
ZosUserId USER1
ZosPassword ?
ZosFtpJobP USER1
JobOutPath /home/user1/zfile4aws_V1R5_Beta/cloud/jobs
KeyStore /home/user1/zfile4aws_V1R5_Beta/tdes4.jks
KeyStorePw jupiter
KeyName key1
```


Result of bash script excremote.sh on linux:



```
samplerun - Notepad
File Edit Format View Help
user1@linux4:~/Zfile4aws_V1R5_Beta> ./excremote.sh "-oper put -fn //USER1.DISK.TSOU01(0)"
ZFM0331 Running Zfile4aws Beta Version:1.5.0 AwsSDK Version:1.12.587 javamainframetools@gmail.com 2024/02/08 20:04:21
ZFM0791 Using configuration file: /home/user1/Zfile4aws_V1R5_Beta/conf/Config_for_user1_onlinux.txt
Enter USER1 Password?
ZFM0611 JOB USER1562 J0417770 submitted, remote file access requested
ZFM0401 Compression method used: class com.ibm.jzos.CompressionFactory$GZIPBufferCompressor
ZFM0411 Encrypt key name used: key1
ZFM0071 File /home/user1/Zfile4aws_V1R5_Beta/cloud/data/USER1.DISK.TSOU01.G0001V00 uploaded to bucket01 bucket
ZFM0431 File /home/user1/Zfile4aws_V1R5_Beta/cloud/data/USER1.DISK.TSOU01.G0001V00 deleted
ZFM0691 JOB USER1562 Stored to /home/user1/Zfile4aws_V1R5_Beta/cloud/jobs/20240208_202843_USER1562_J0417770.txt
ZFM0071 File /home/user1/Zfile4aws_V1R5_Beta/cloud/data/USER1.DISK.TSOU01.G0001V00_00000001 uploaded to bucket01 bucket
ZFM0431 File /home/user1/Zfile4aws_V1R5_Beta/cloud/data/USER1.DISK.TSOU01.G0001V00_00000001 deleted
ZFM0371 File USER1.DISK.TSOU01.G0001V00_00000001 Uploaded Total size=1908687980 Bytes, in 2 segment(s)
ZFM0451 Zfile4aws completed, RC=0
user1@linux4:~/Zfile4aws_V1R5_Beta>
```

Notes:

Result of batch job submitted to z/OS is stored in the 'JobOutPath' config parameter directory.

With remote mode, all cpu cycles required to compress data and encrypting works are made on linux/windows server.

Access to the cloud server is also performed from linux/windows server after data compression and encryption was completed.

*** License usage Warning Notice ***

1. USE OF ZFILE4AWS IS AT YOUR SOLE RISK!
2. ALL MATERIALS, INFORMATION, PRODUCTS, SOFTWARE, PROGRAMS, ARE PROVIDED 'AS IS' WITH NO WARRANTIES OR GUARANTEES.

Questions and Comments can be sent to Email:

javamainframetools@gmail.com

How to order a Zfile4aws BETA TEST COPY

Download it from Github: [javamainframetools/Zfile4aws_V1R5_Beta](https://github.com/javamainframetools/Zfile4aws_V1R5_Beta)

Copyright Statement

- Copyright © 2022-2024 Zfile4aws. All rights reserved.
- Java is registered trademarks of Sun Microsystems, Inc.
- Amazon S3 is a registered trademark of **Amazon Simple Storage Service**.
- Windows are registered trademarks of Microsoft Corporation.
- Z/OS is registered trademarks of International Business Machines Corp.
- All other trademarks and copyrights referred to are the property of their respective owners.

Glossary of terms

JAVA

Java® is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

Z/OS

z/OS®, a widely used mainframe operating system, is designed to offer a stable, secure, and continuously available environment for applications running on the mainframe. z/OS today is the result of decades of technological advancement.

JES2

A z/os subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system.

JCL

Job Control Language is required to run job process in batch mode on z/OS.

JZOS

Utility required to run JAVA standalone application in batch mode.

AWS S3

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps.pwd