2024

Java Mainframe Tools - Zfile4aws

V1R5 BETA VERSION

USER GUIDE

Contents

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 //*Dsname* | /*Omvsfile* [**-cfg** /*Config-filename*]
[**-opt** *Options*]

Parameters:

//*Dsname*:

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** /C*onfig-filename*]

Parameters:

Object-name-prefix:

All cloud objects having this name prefix will be listed.

/C*onfig-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:

/C*onfig-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 il 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 ll is possible to execute compressing and encrypting works on others 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 summited 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.

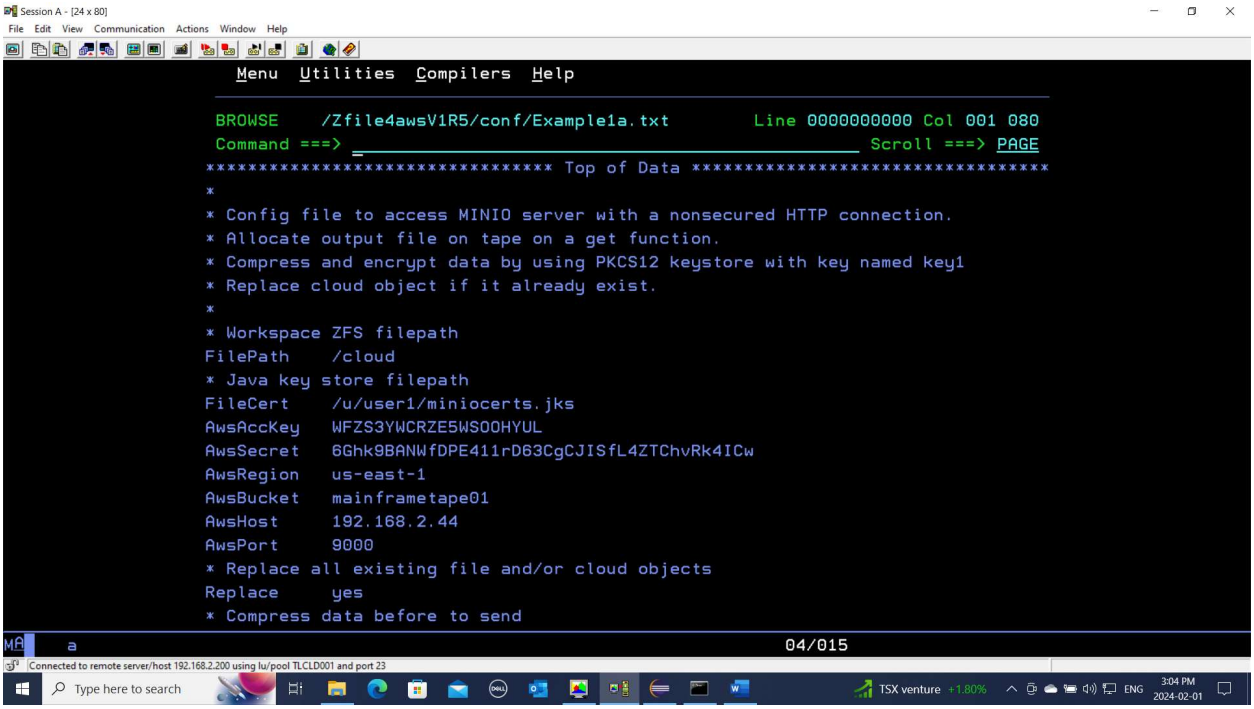
Since, mainframe data transport is made thru ftp only. You may use z/OS ftp SSL to implement a secure connection for data transport between z/OS and the running java program running on linux/windows platform.

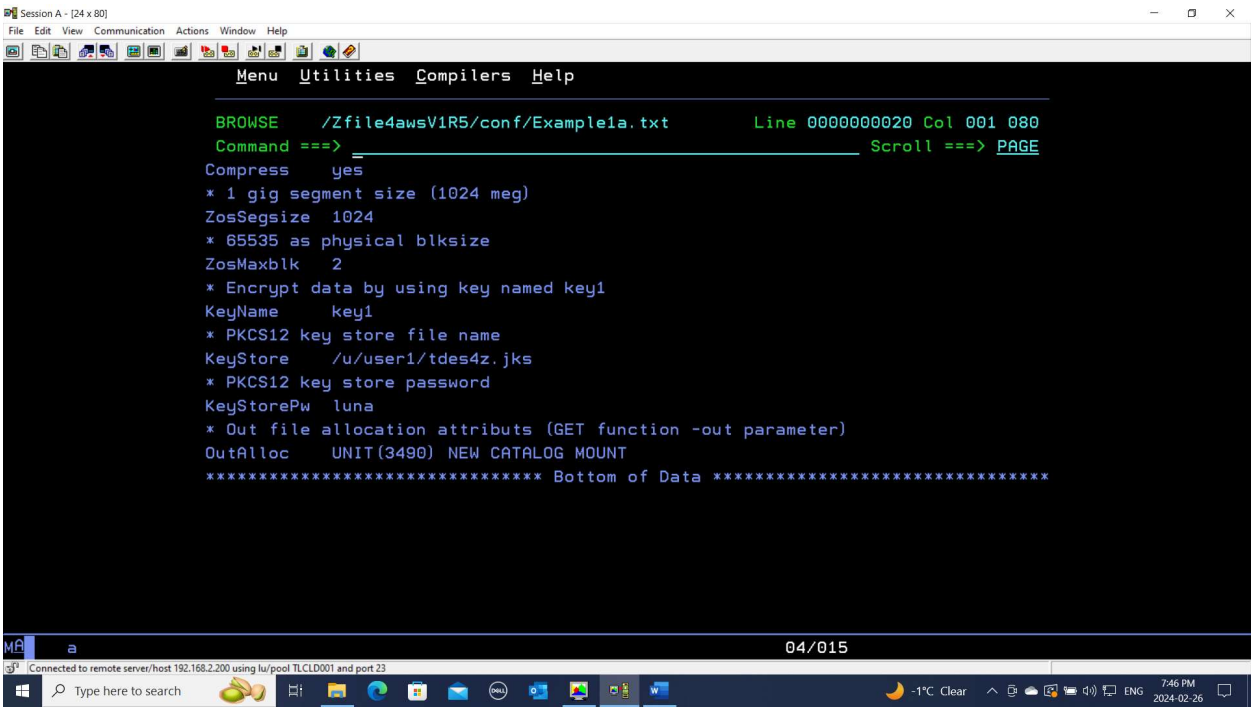
Remote mode required that ‘Remote shell exec server’ be implemented on z/OS (RSHD server).

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

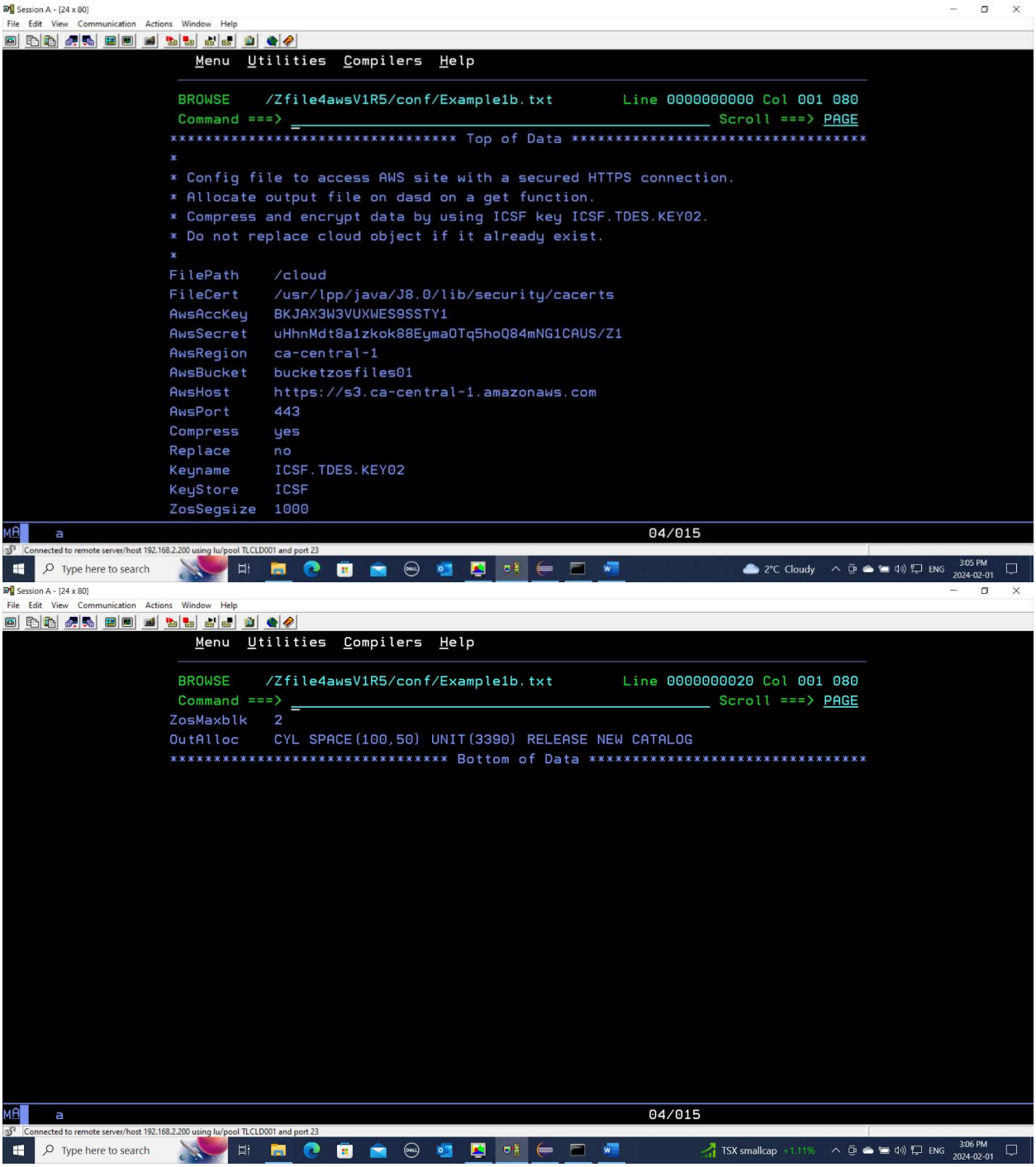
Examples

Example 1a – Configuration file to a MINIO server

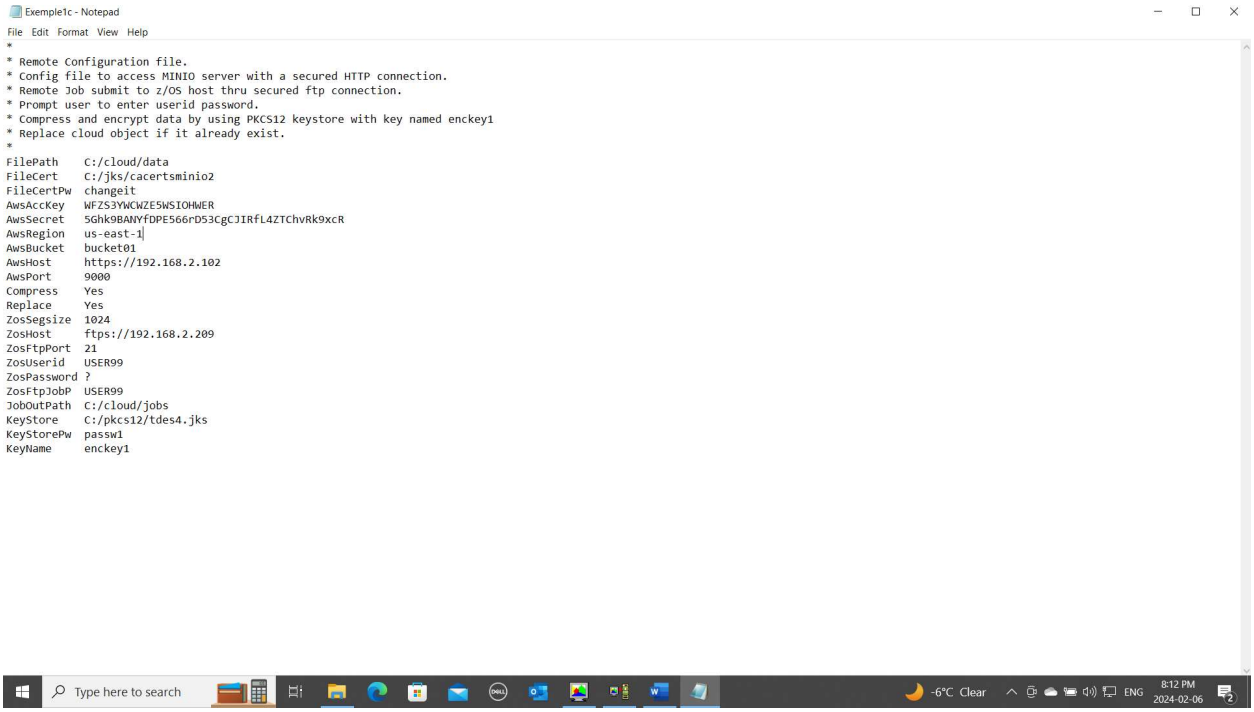




Example 1b – Configuration file to an AWS server

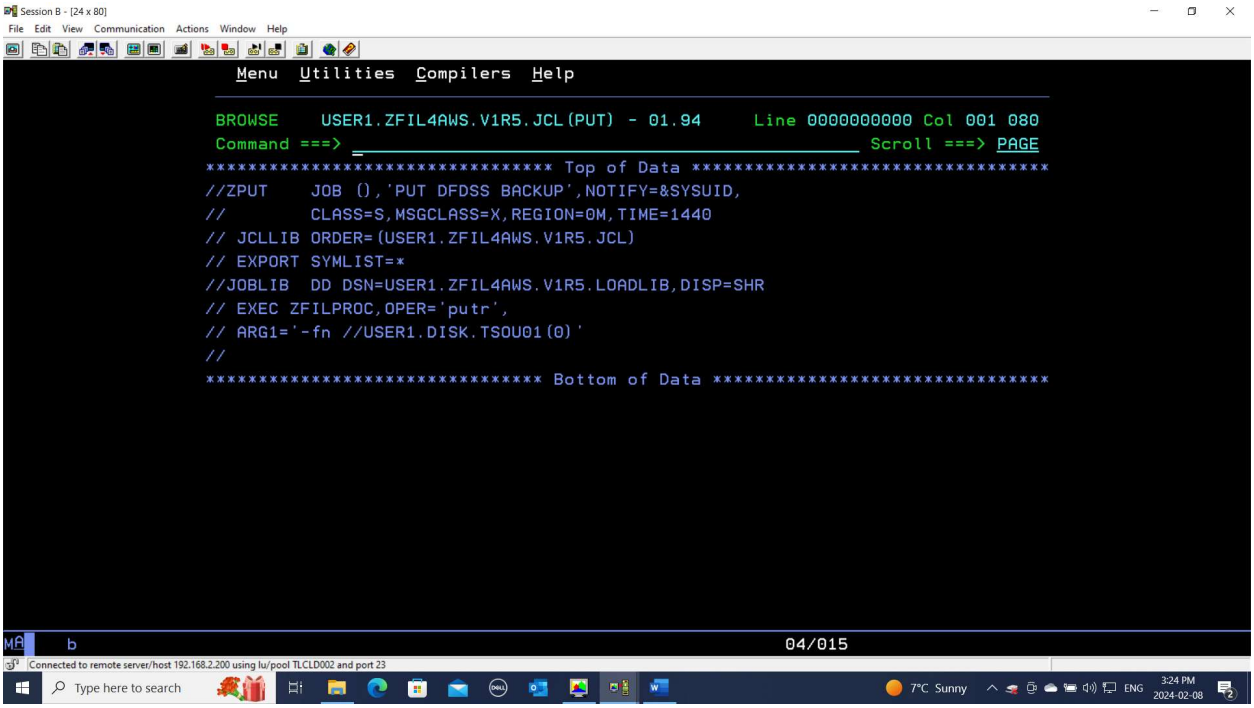


Example 1c – Remote mode configuration file to a MINIO server

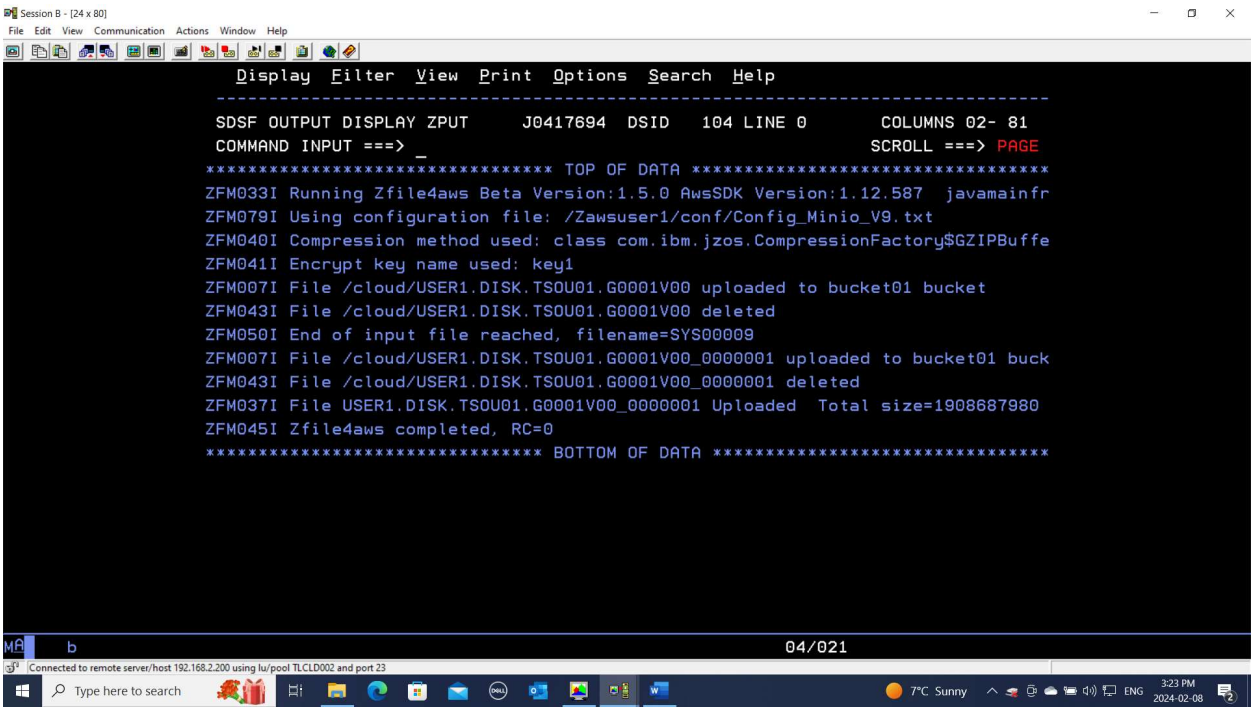


Example 2 – PUT job (Upload z/OS file to a cloud bucket)

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

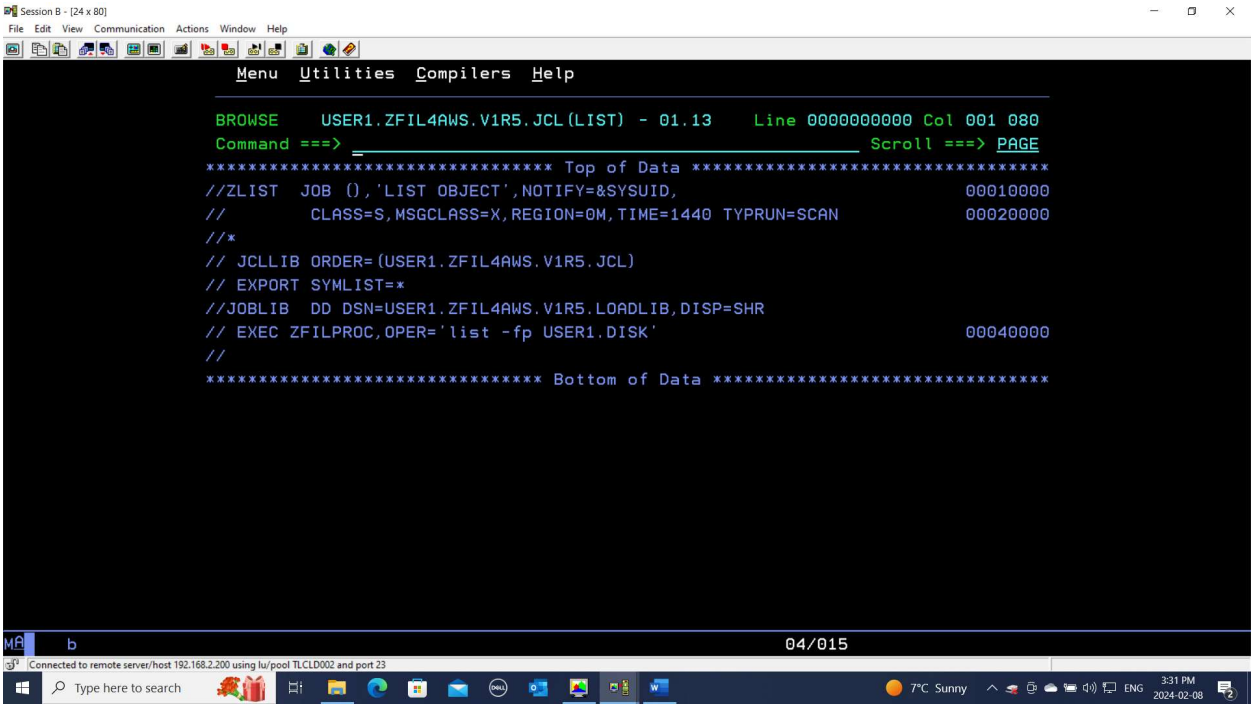


JOB output result:

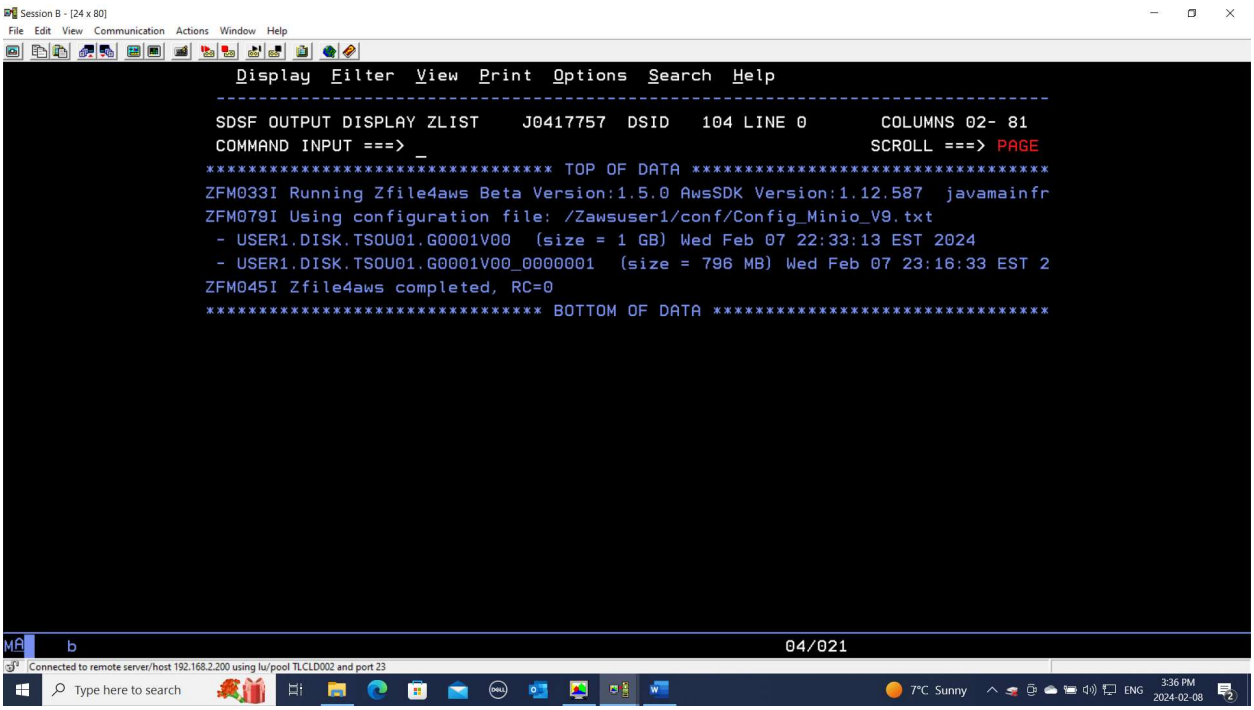


Example 3 – LIST job (List cloud objects)

JCL used to list objects USER1.DISK.**

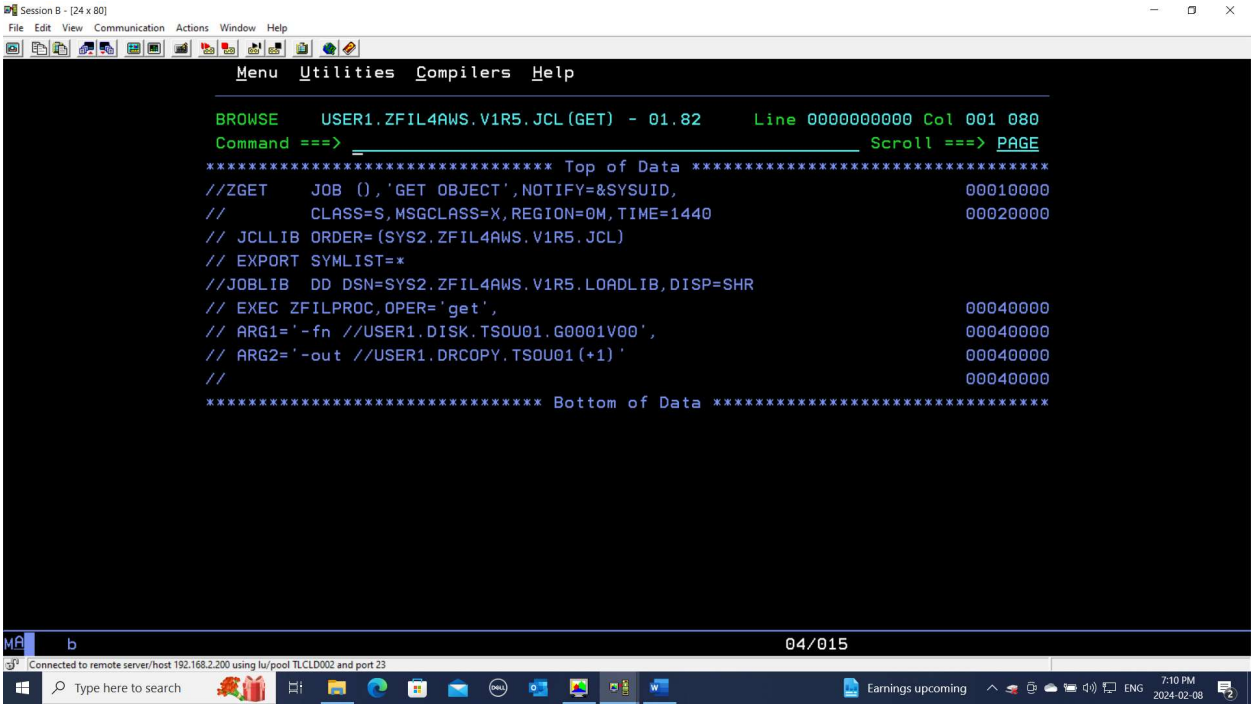


JOB output result:

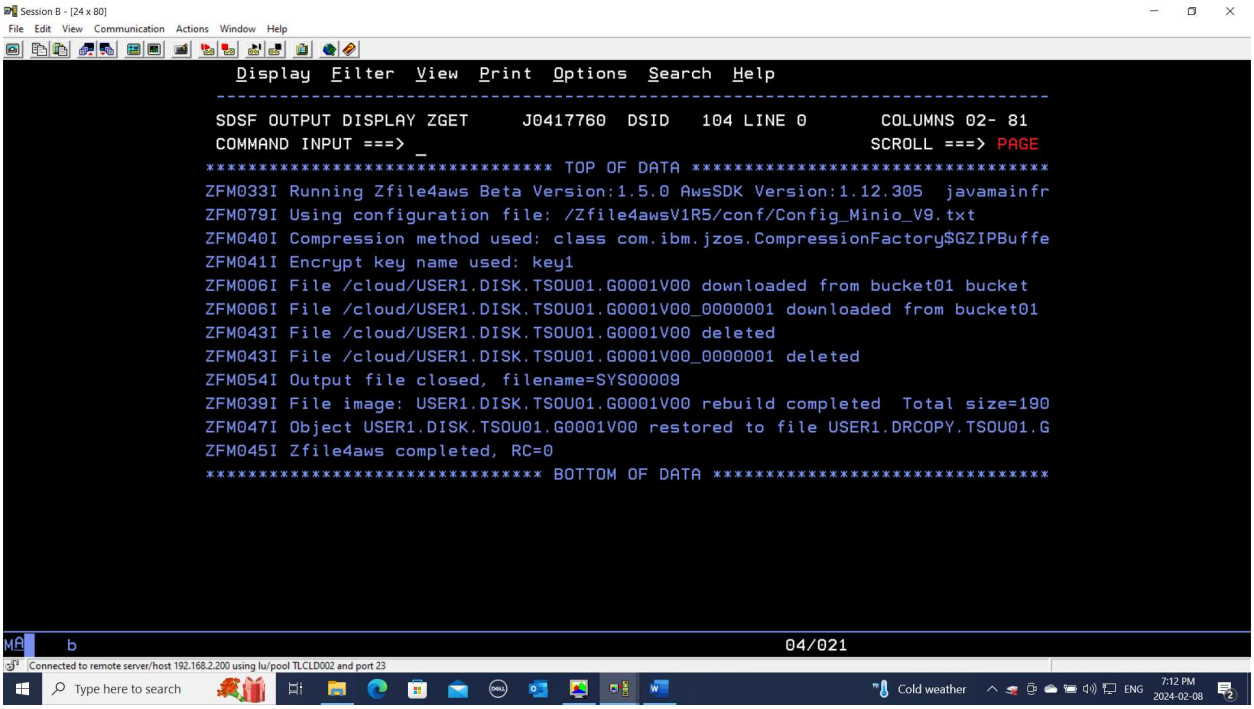


Example 4 – GET job (download or restore cloud objects to z/OS ds)

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

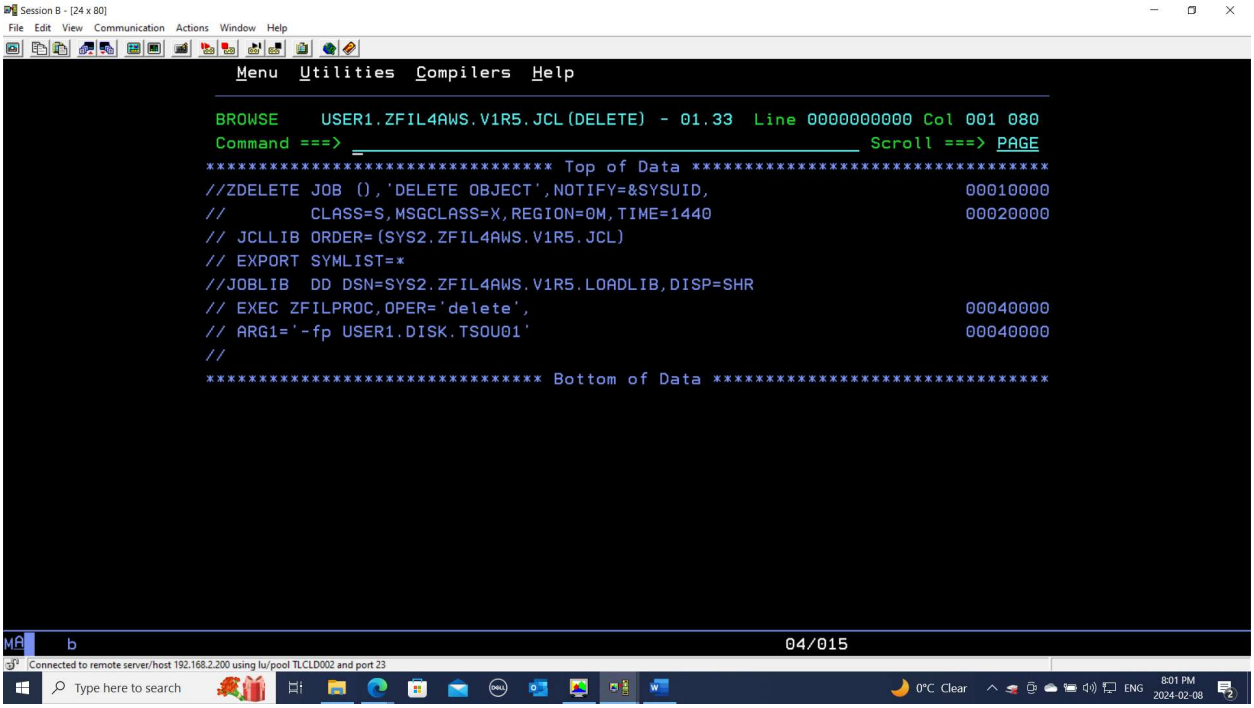


JOB output result:

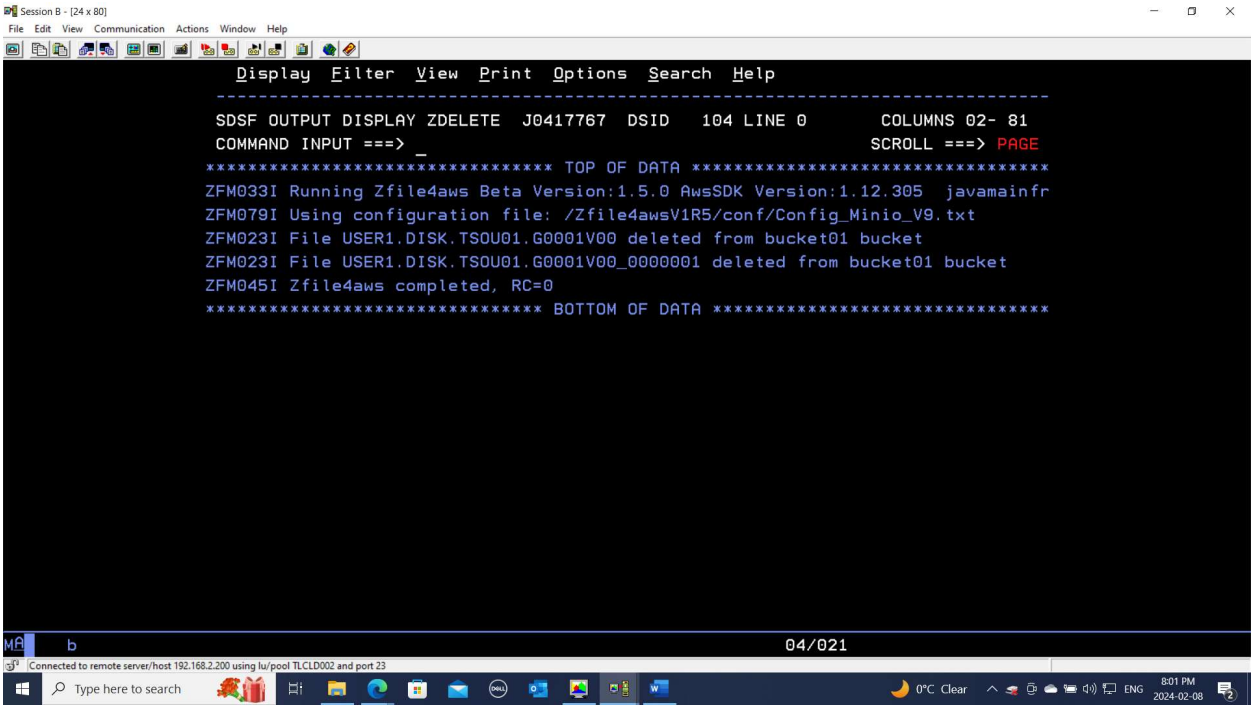


Example 5 – DELETE job (delete object(s) from cloud bucket)

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

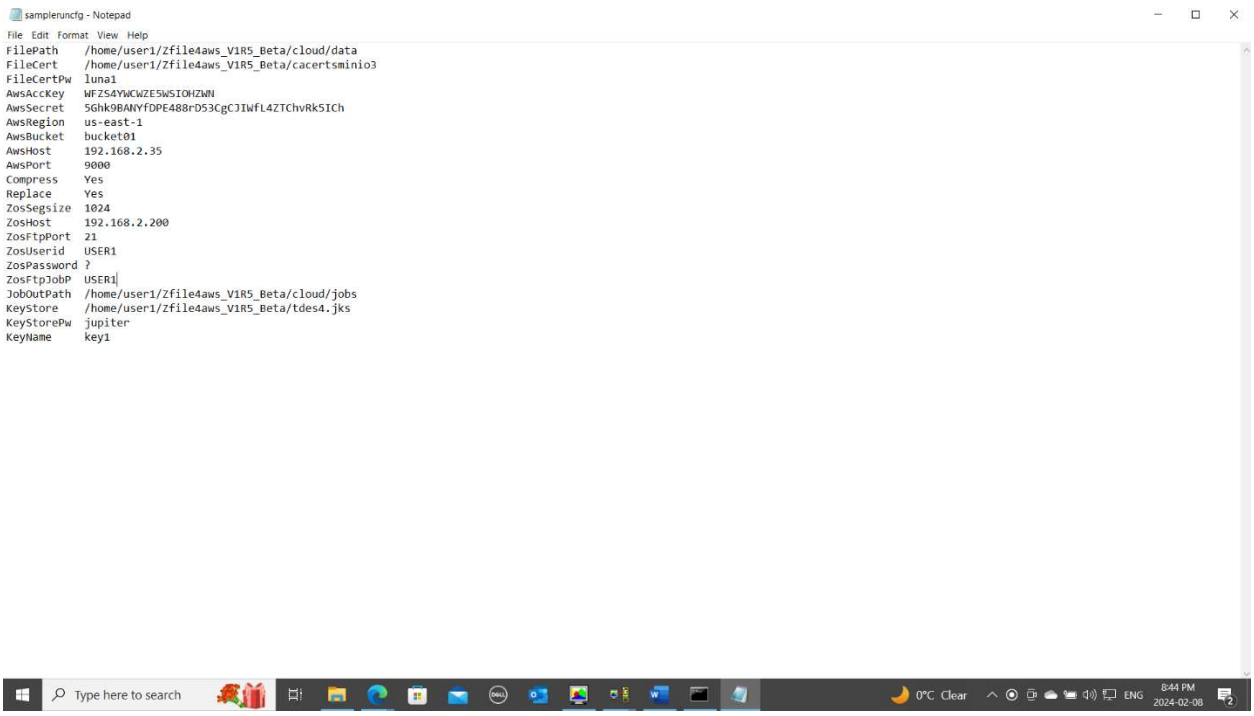


JOB output result:

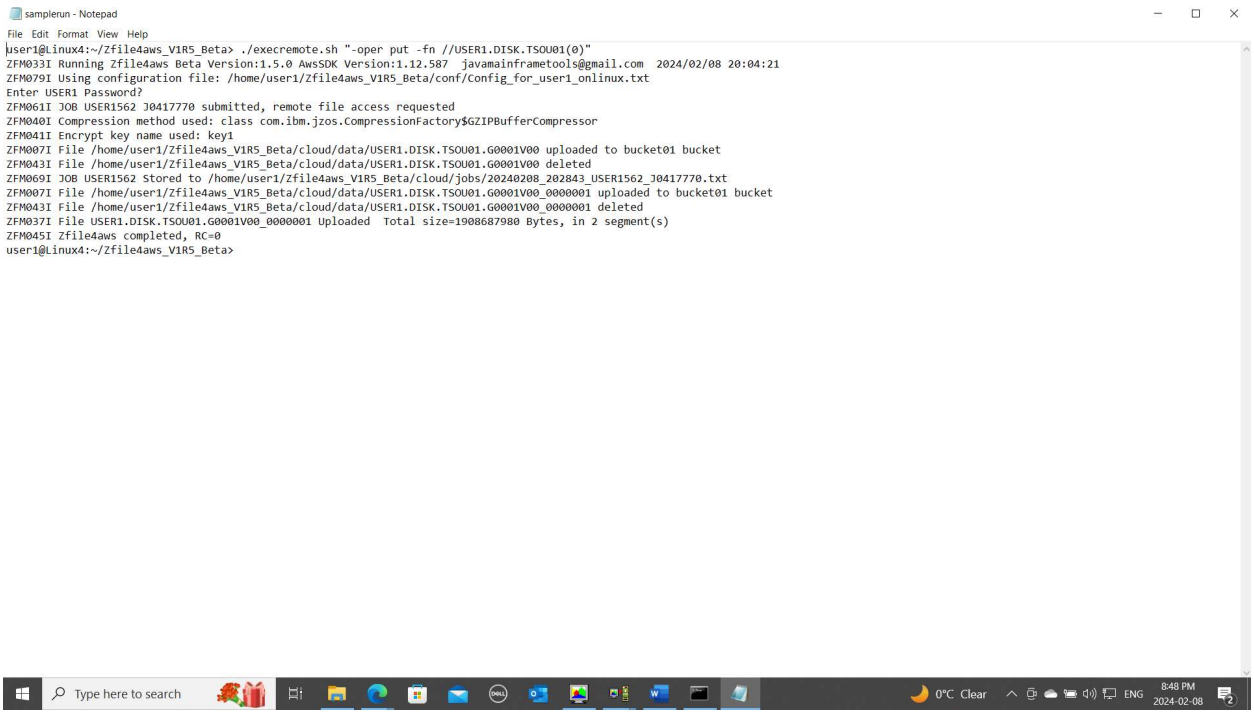


Example 6 – Running from linux (remote mode execution)

Config file used on linux:



Result of bash script execremote.sh on linux:



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

Copyright Statement

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