2023

Java Mainframe Tools - Zfile4aws

V1R4 BETA VERSION

USER GUIDE

Contents

Introduction

Zfile4aws is a java mainframe tool used to send and retrieve z/OS sequential file from an AWS S3 cloud type.

Data source could be any sequential file type or format (RECFM=PS).

Zfile4aws support tape files like DFDSS backup and LBI type dataset (Large physical block size).

With Zfile4aws sequential data could be compressed, encrypted and segmented before it is shipped to the cloud.

In theory, there is no limit in dataset size that could be send to the cloud. Limits depends on your environment and/or your cloud service provider.

Zfile4aws is running in batch mode (JZOS) only.

HFS, ZFS and OMVS file types are also supported.

Environment requirements

- JAVA V8 31 bits is required.
- At least z/OS 2.2.
- 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

There are four operations available to access data in the cloud with Zfile4aws.

The GET function is used to retrieve and download data from an S3 bucket.

The PUT function is used to upload and store data to an S3 bucket.

The LIST function is used to list objects information from an S3 bucket.

And the DELETE function used to delete cloud object(s) from an S3 bucket.

All configuration parameters required to run are specified in a single OMVS text file. This file is called Configuration File.

Before sending any z/OS dataset to cloud. Zfile4aws read and convert input data to an internal image format. This file image format support compression, encryption and segmentation. This internal image format is stored in a temporary work ZFS file. Internal image format is created by the PUT function.

User must predefine enough ZFS space to accommodate space requirement for his environment. This temporary ZFS space is called the Workspace.

Auto deletion of temporary file image format is controlled by the ‘-opt d’ option parameter (explain later in this document).

Access to cloud objects is controlled by validating RACF access to the corresponding z/OS dataset name.

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 have 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 request.

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 stand for 1 gig)

ZosKeyname: ICSF/ICRF key label to use to perform 3DES encryption.

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)

An Asterix (*) in Column1 stand 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.

Functions

GET | **GETR** is used to retrieve object from an AWS S3 bucket.

Parameters:

-cfg	Full file path name of the configuration file.
-fn	Full dsname or omvs file path name to download from cloud. By using //dsname… form you specify a full z/OS dataset name.
-dd	DDNAME used to store retrieved data.

Use function GETR to execute a GET with 'replace' if you want to replace any existing file name on the local z/OS system.

PUT | **PUTR** is used to store object to an AWS S3 bucket.

Parameters:

-cfg	Full file path name of the configuration file.
-fn	Full dsname or omvs file path name to upload to cloud. By using //dsname… form you specify a full z/OS dataset name.
-dd	DDNAME of source data.
-blk	Blocksize of the input source dataset. This value overrides ZosMaxblk config parameter.

Use function PUTR to execute a PUT with 'replace' if you want to replace any existing cloud object.

If **-dd** and **-fn** are both specified. DSN allocated by the DDNAME specified in the **-dd** parameter, override any DSN specified in the **-fn** parameter.

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

Parameters:

-cfg	Full file path name of the configuration file.
-fn | -fp	Specific object name to delete or objects prefix name for deleting multiples objects.

To delete an object representing a z/OS dataset (dsn). Current RACF userid must have RACF ALTER authority to the corresponding dsn.

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

Parameters:

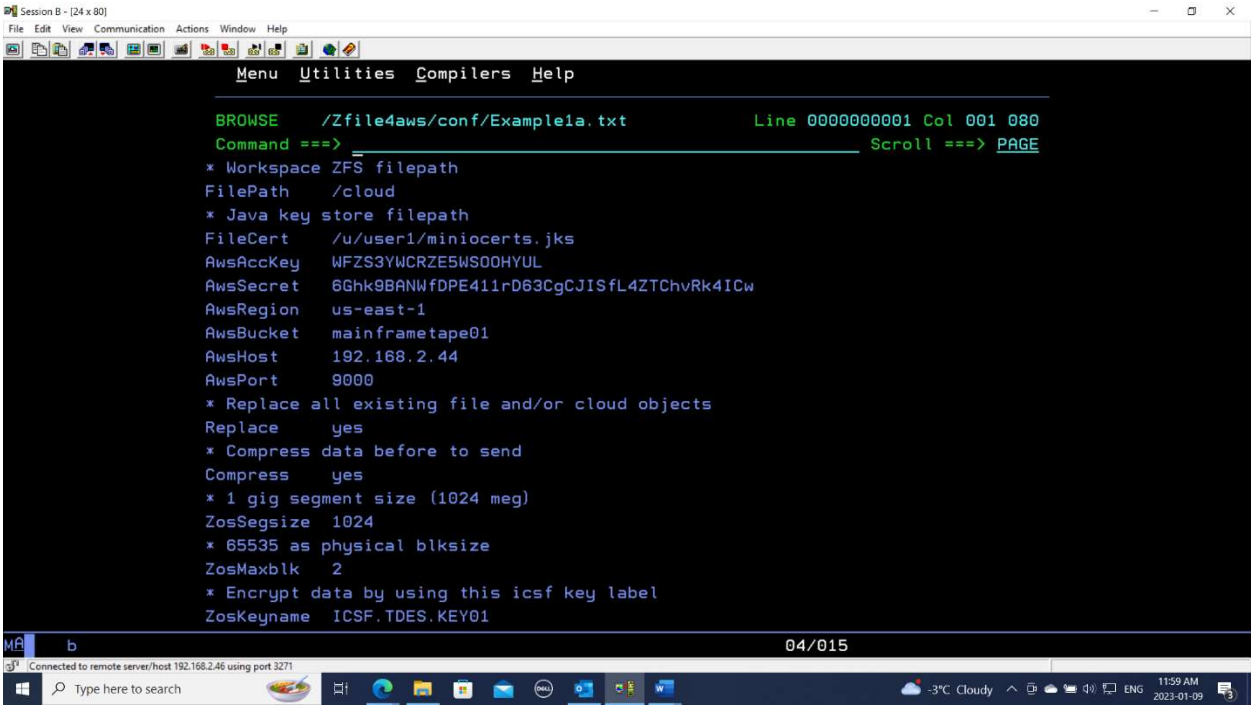
-cfg	Full file path name of the configuration file.
-fp	List all objects having name starting with this prefix name. This parameter is optional.

The **-opt** parameter may be set for GET and PUT functions.

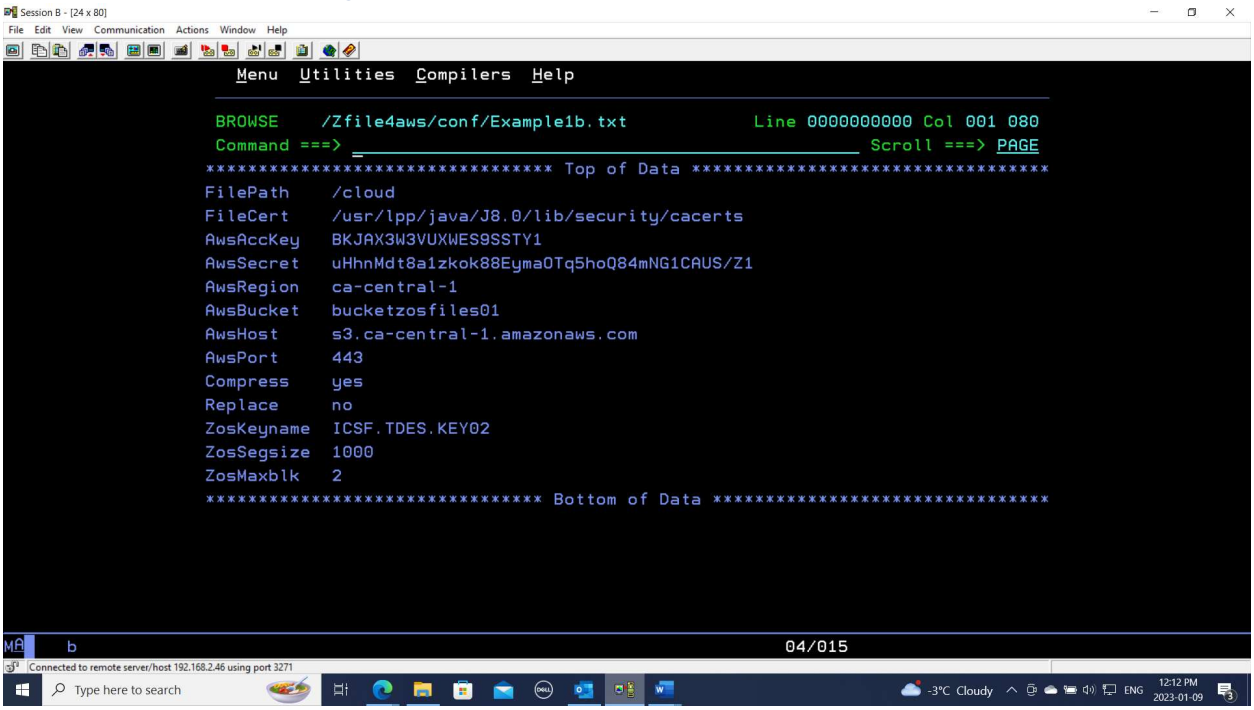
-opt v	Verbose option to get more messages with timestamp.
-opt d	Auto deletion of internal file image from the ZFS workspace.
-opt b	Bypass file image creation. Valid only for USS file type.

Examples

Example 1a – Configuration file to a MINIO server

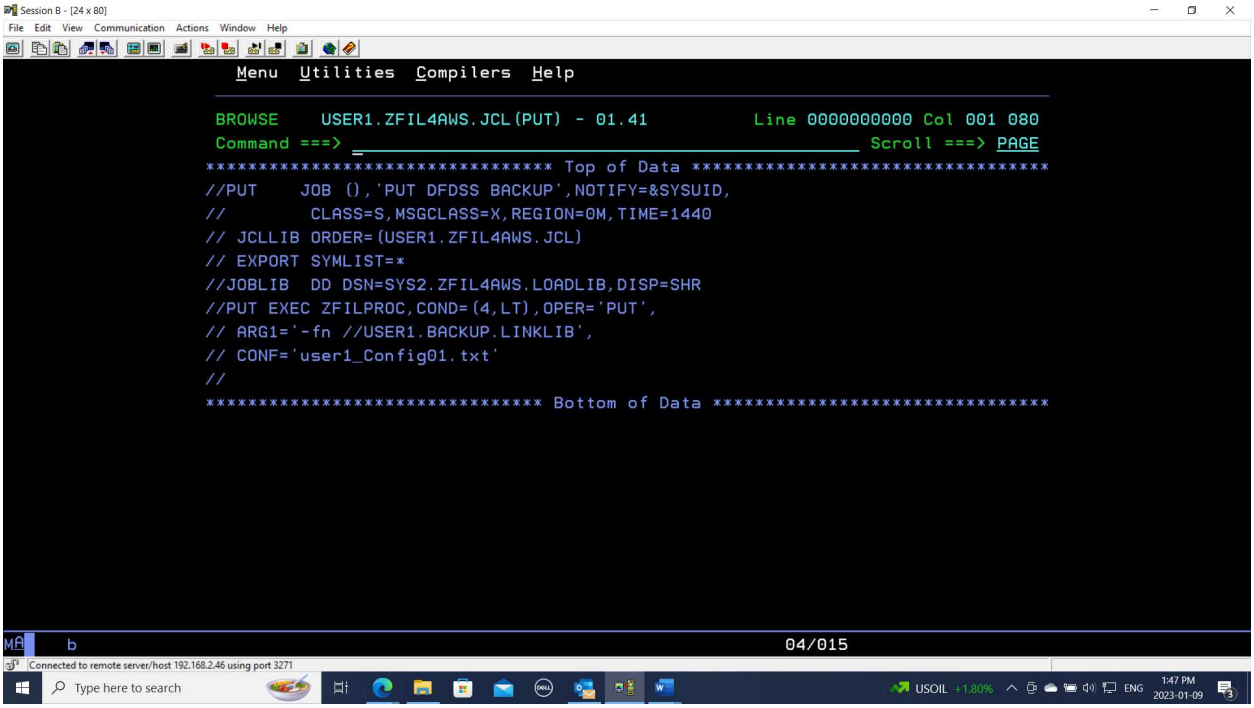


Example 1b – Configuration file to a AWS server

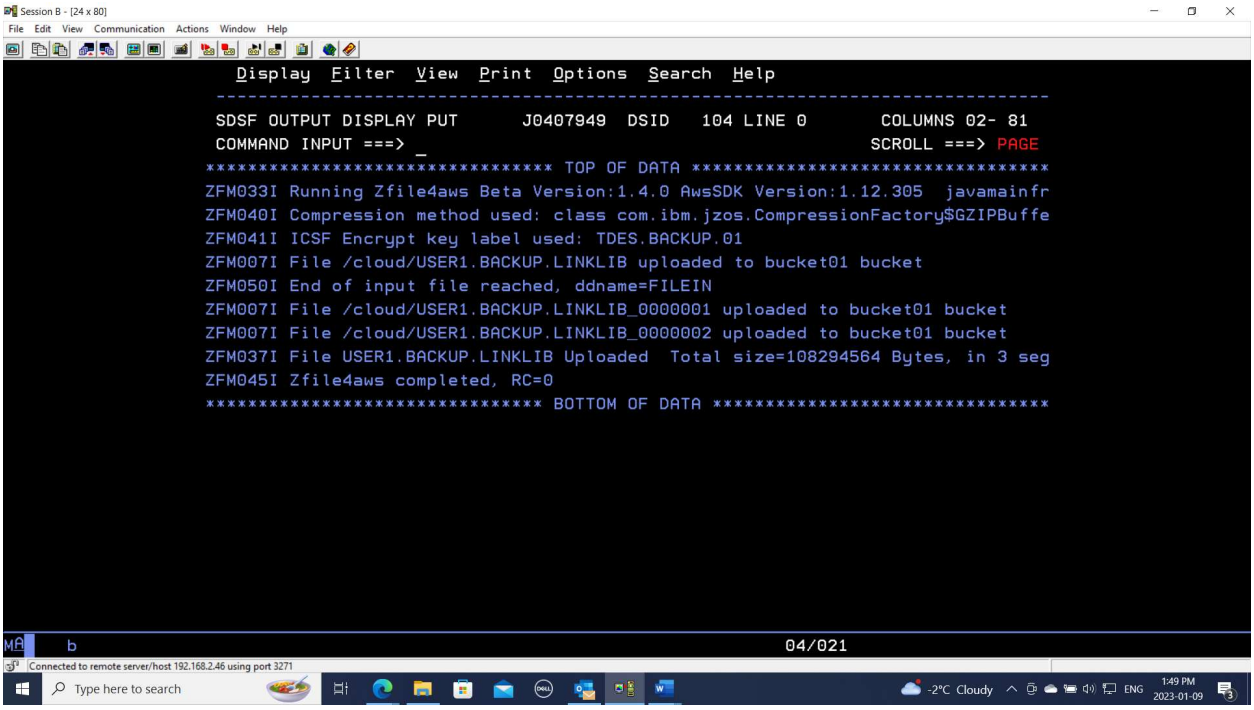


Example 2 – PUT job (Upload z/OS file to a cloud bucket)

JCL used to upload tape file ‘USER1.BACKUP.LINKLIB’:

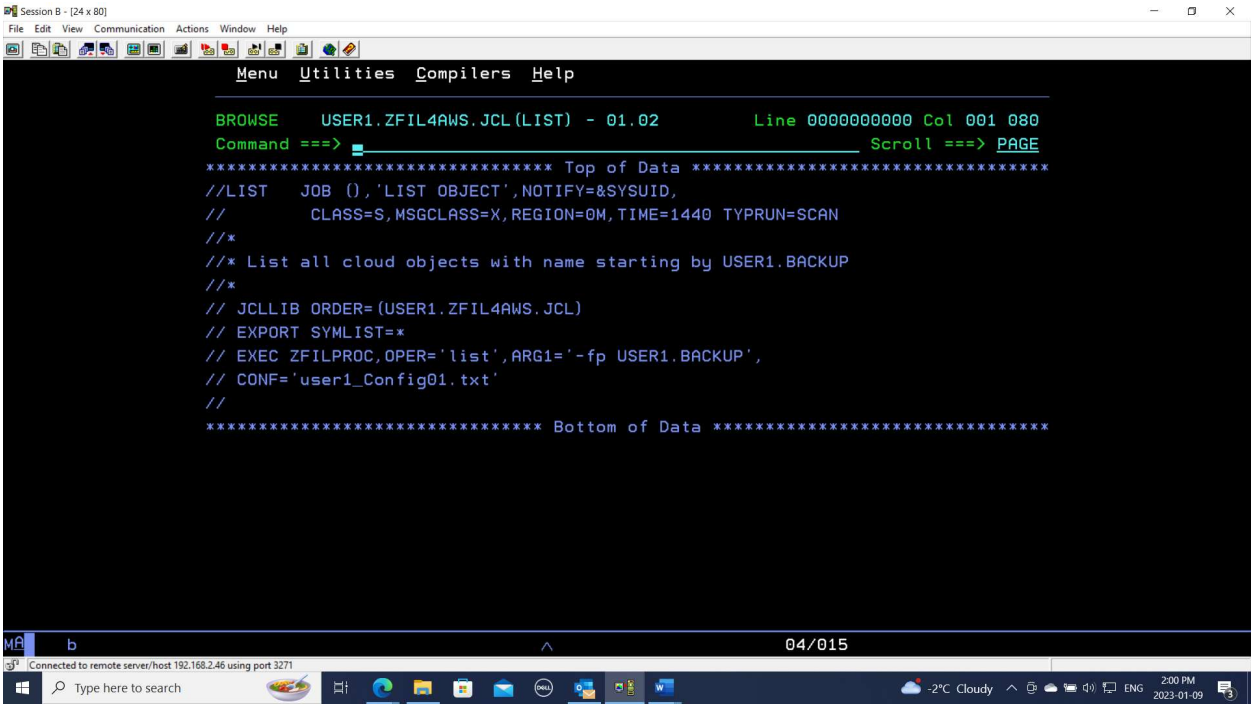


JOB output result:

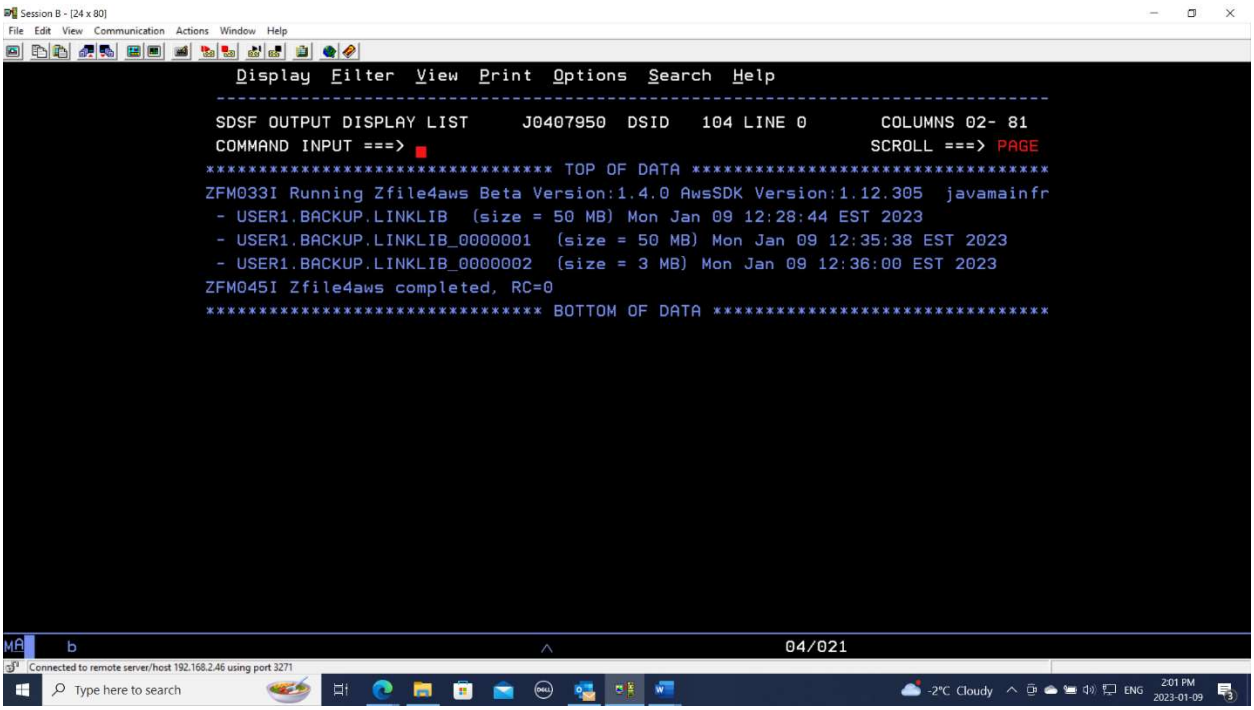


Example 3 – LIST job (List cloud objects)

JCL used to list objects USER1.BACKUP.**

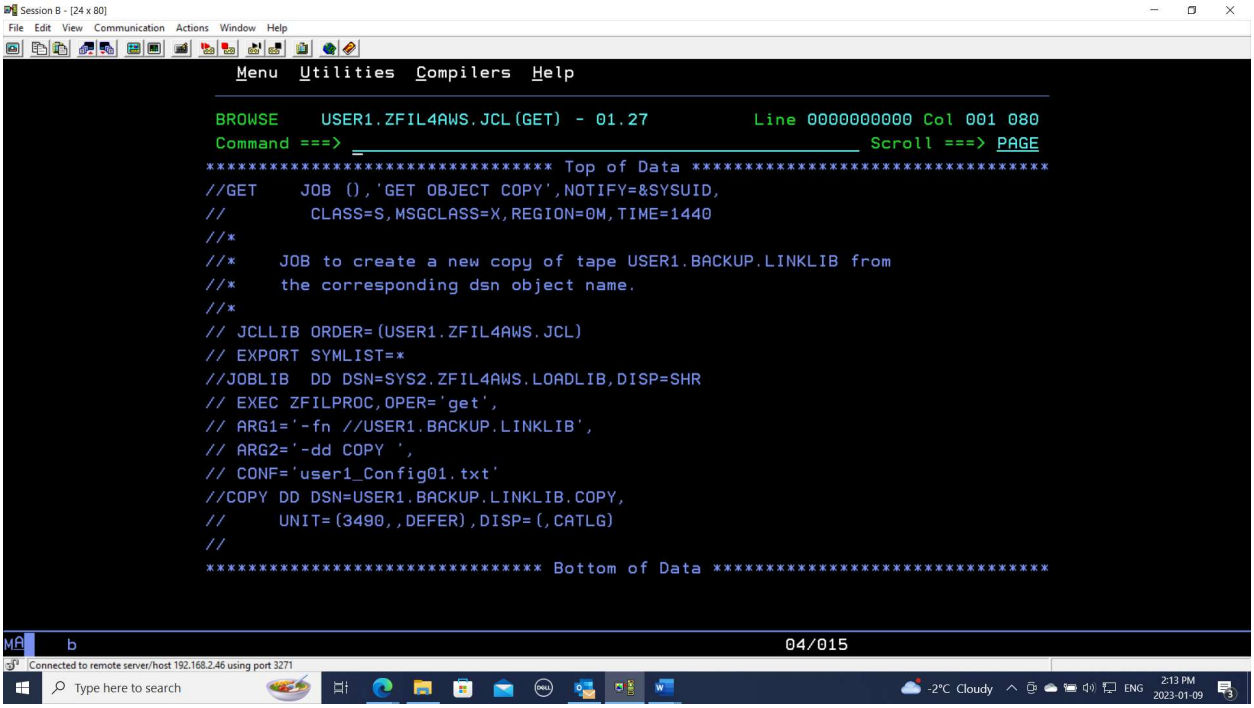


JOB output result:

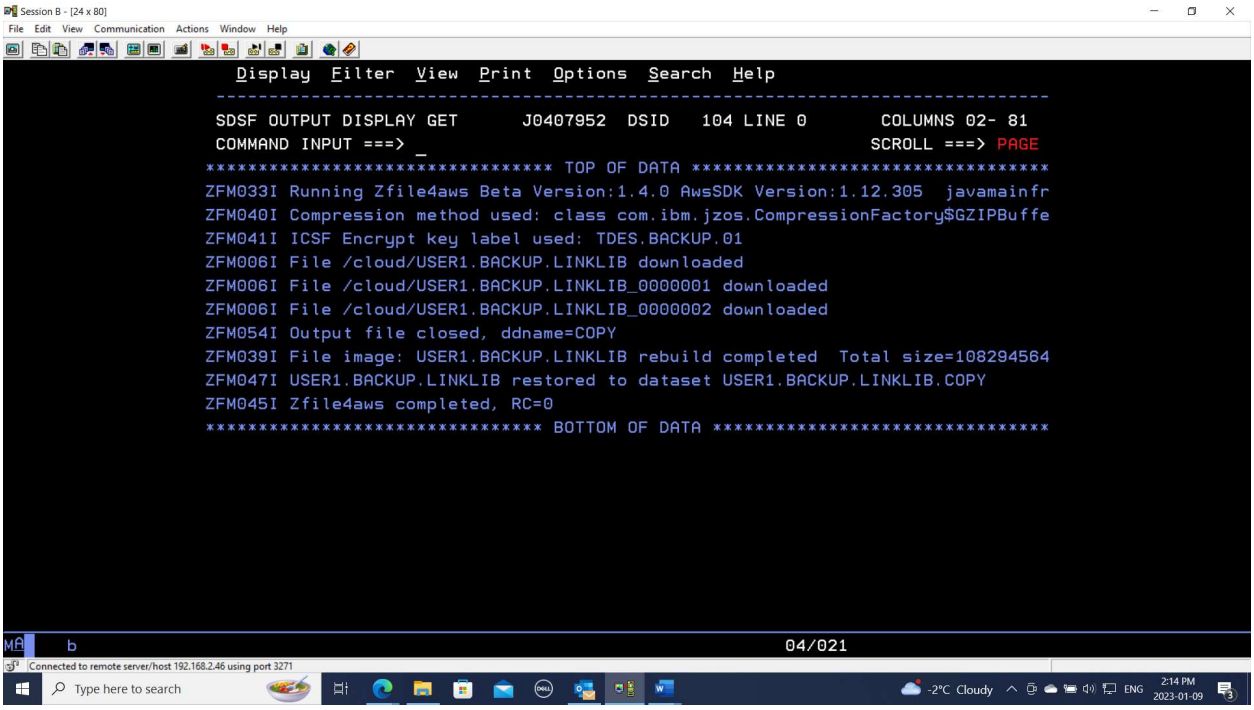


Example 4 – GET job (download or restore cloud objects to z/OS ds)

JCL used to get object copy of USER1.BACKUP.LINKLIB

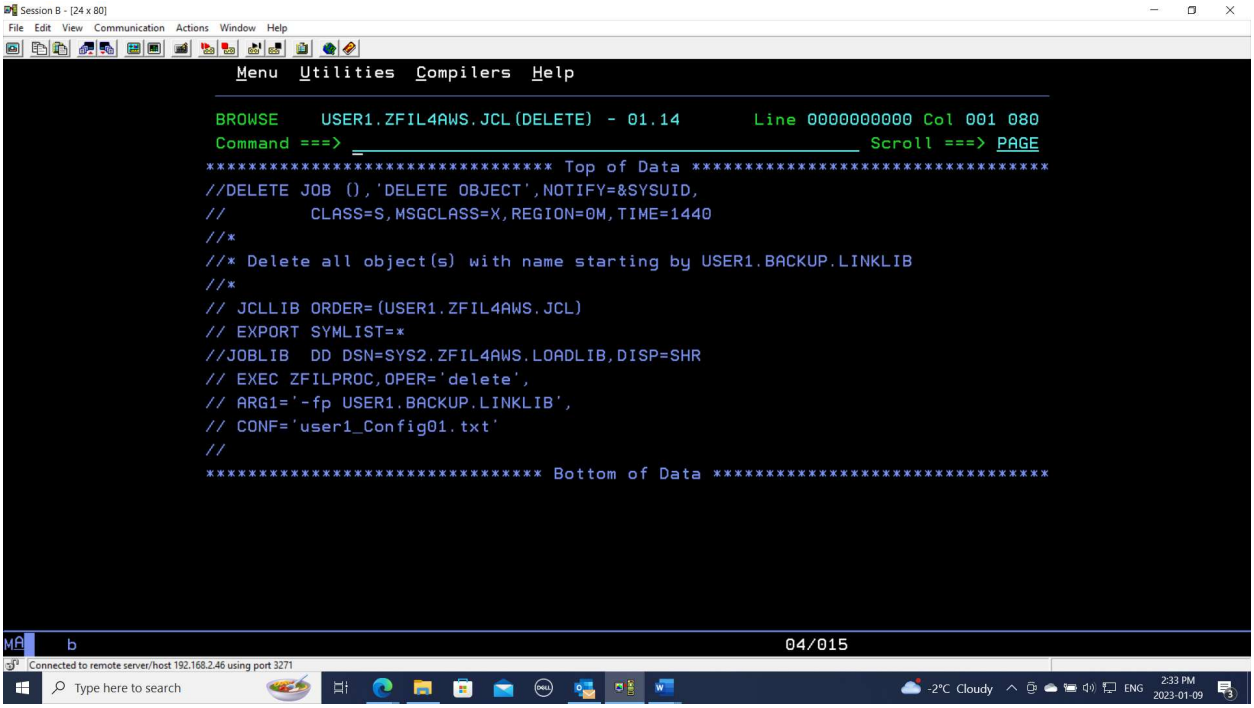


JOB output result:

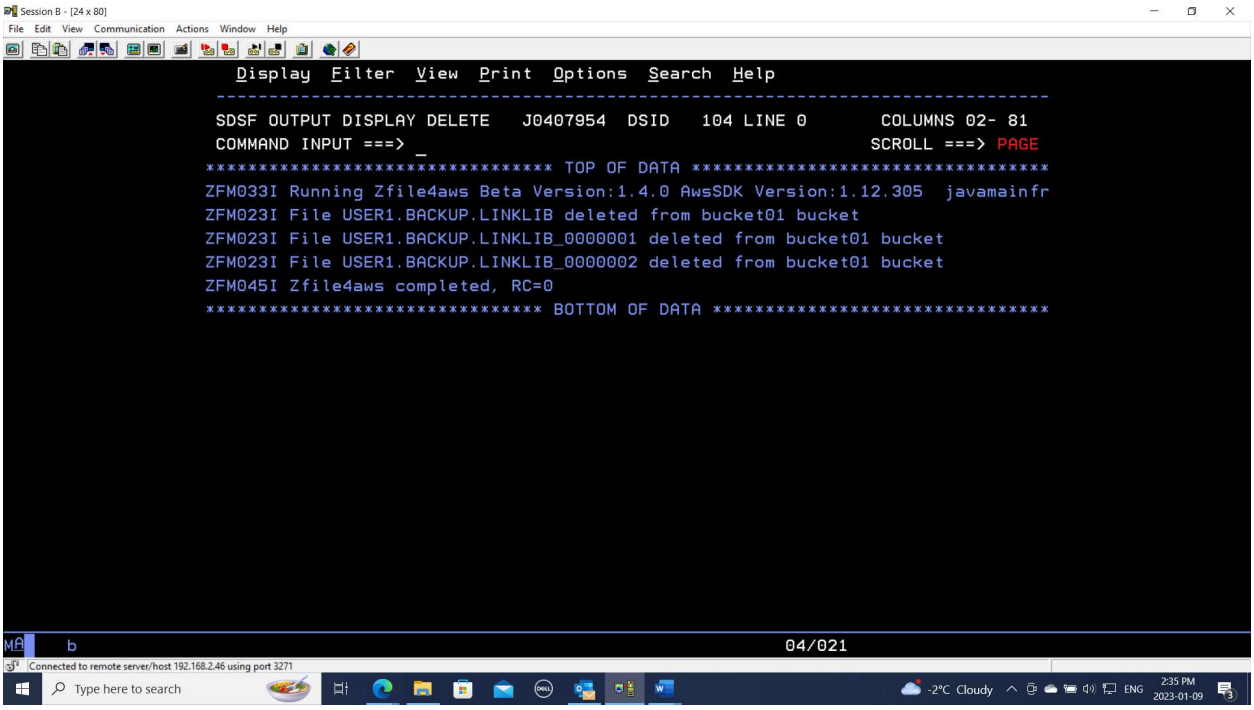


Example 5 – DELETE job (delete object(s) from cloud bucket)

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



JOB output result:



*** License usage Warning Notice ***

ZFIL4AWS Beta version is valid for about 100 days.

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.

3. Zfile4aws beta test copy is valid for about 100 days.

Questions and Comments can be sent to Email:

javamainframetools@gmail.com

How to order a Zfile4aws BETA TEST COPY

Send an Email to: javamainframetools@gmail,com

Or download it from Github: javamainframetools/Zfile4aws_V1R4_Beta

Email should be answered with an attached zip file. Unzip it on your workstation and see installation instruction in the /doc subdirectory.

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.