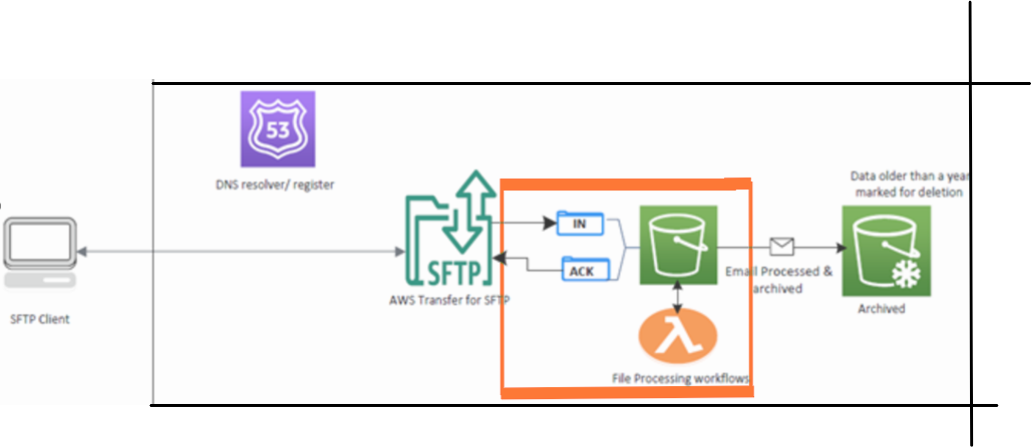
**AWS – Lambda Functions**

**Use case: File Transfer & archival solution for Test System**

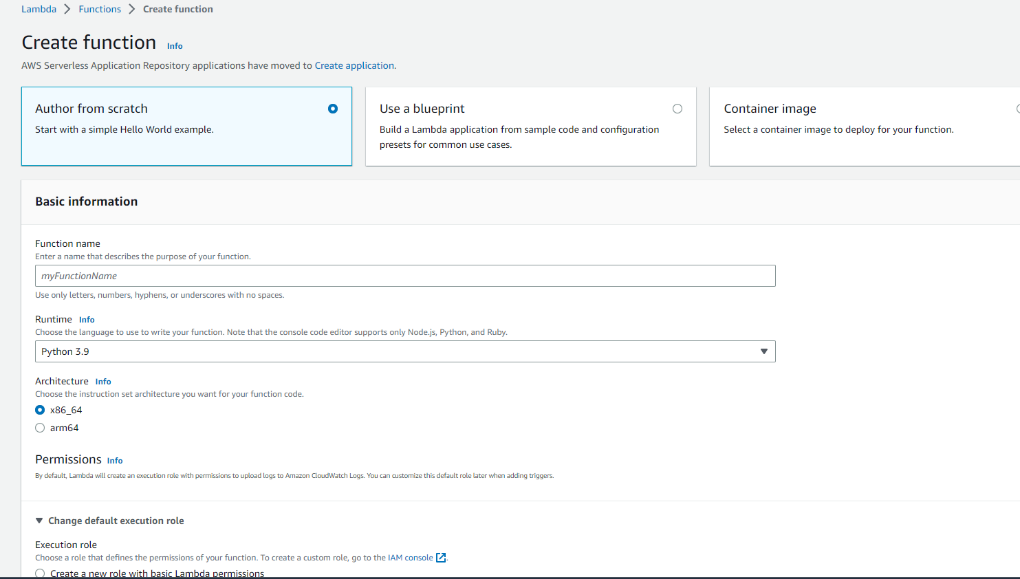


Architecture for the solution. (Deployment of services constituted inside the orange box)

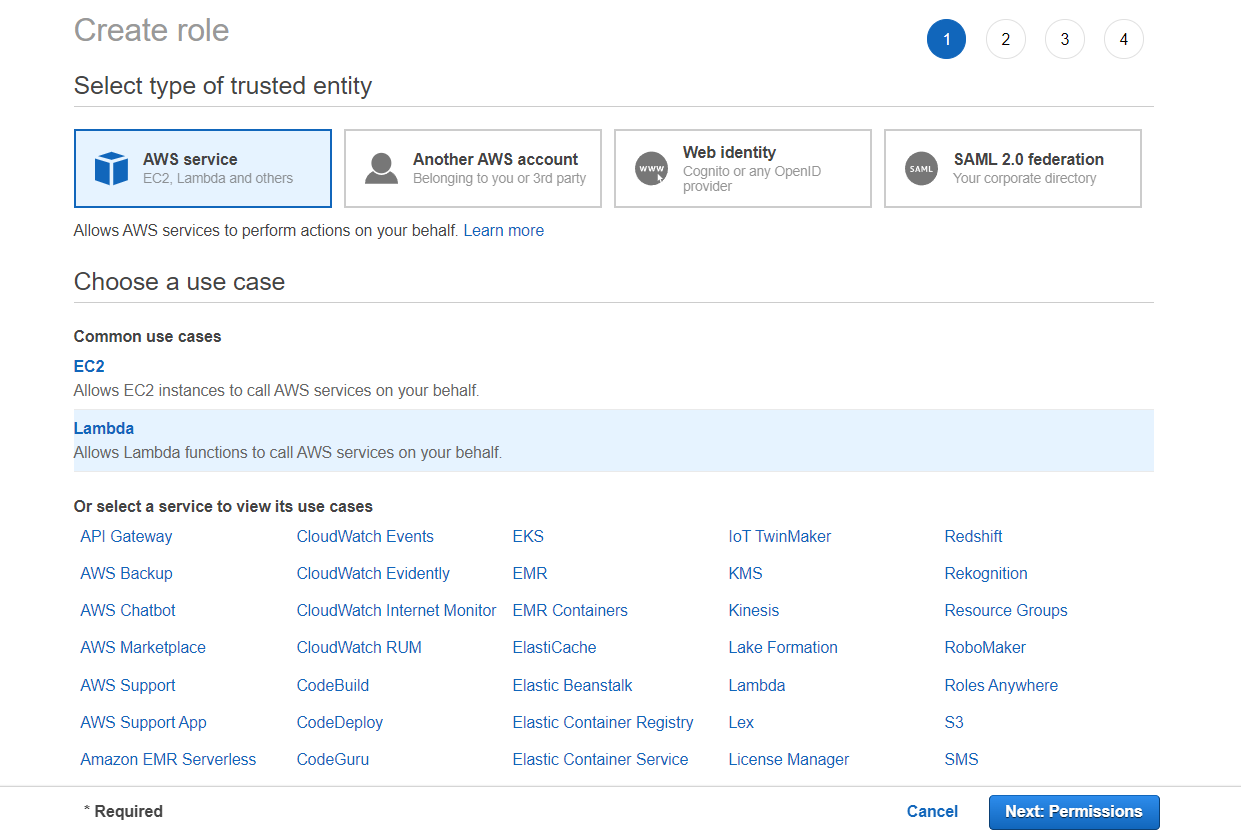
Deploying a Lambda Function for unzipping the uploaded zipped files (from the Client through AWS Transfer for SFTP) and converting those in XML format.

**Service Provisioning**

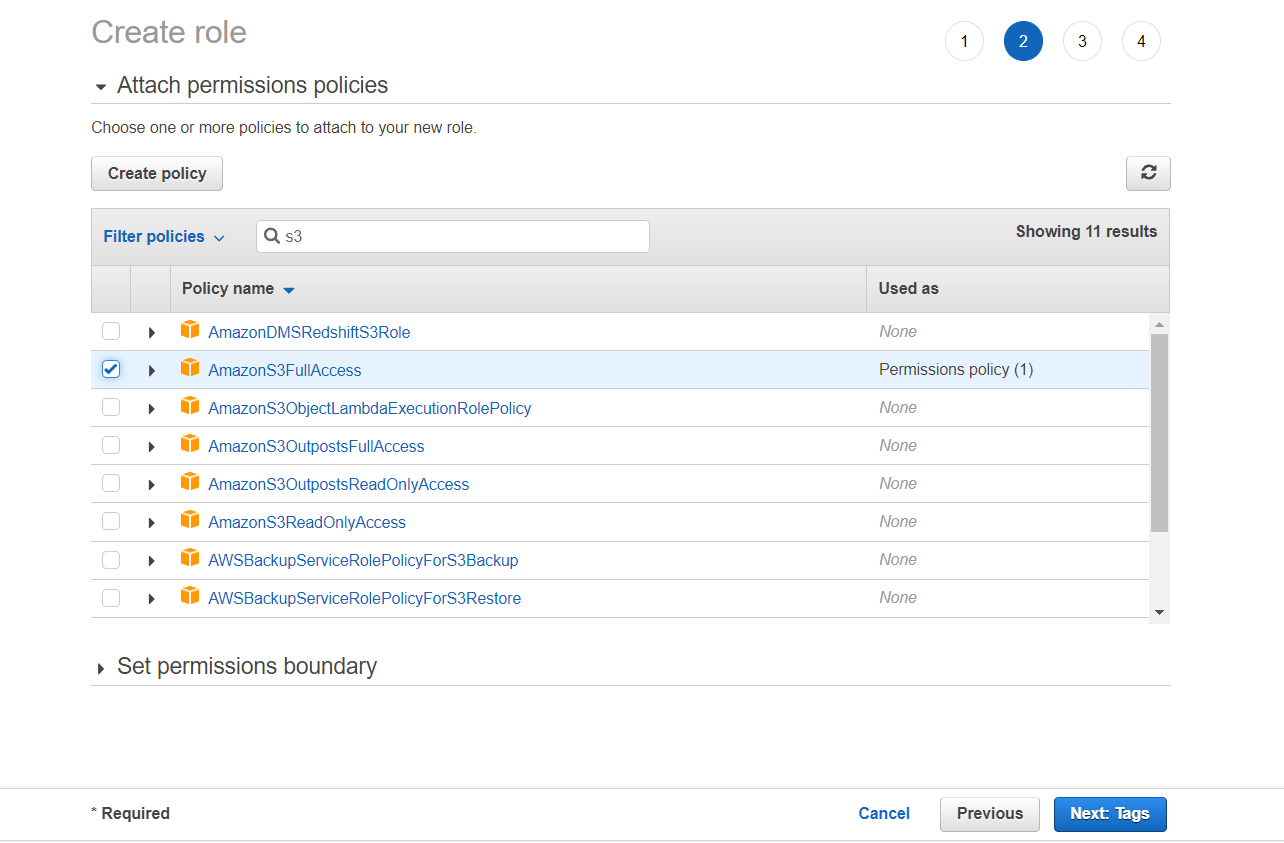
Creating a lambda function from scratch with Python 3.9 as Runtime language with a custom role with S3 access (r/w)



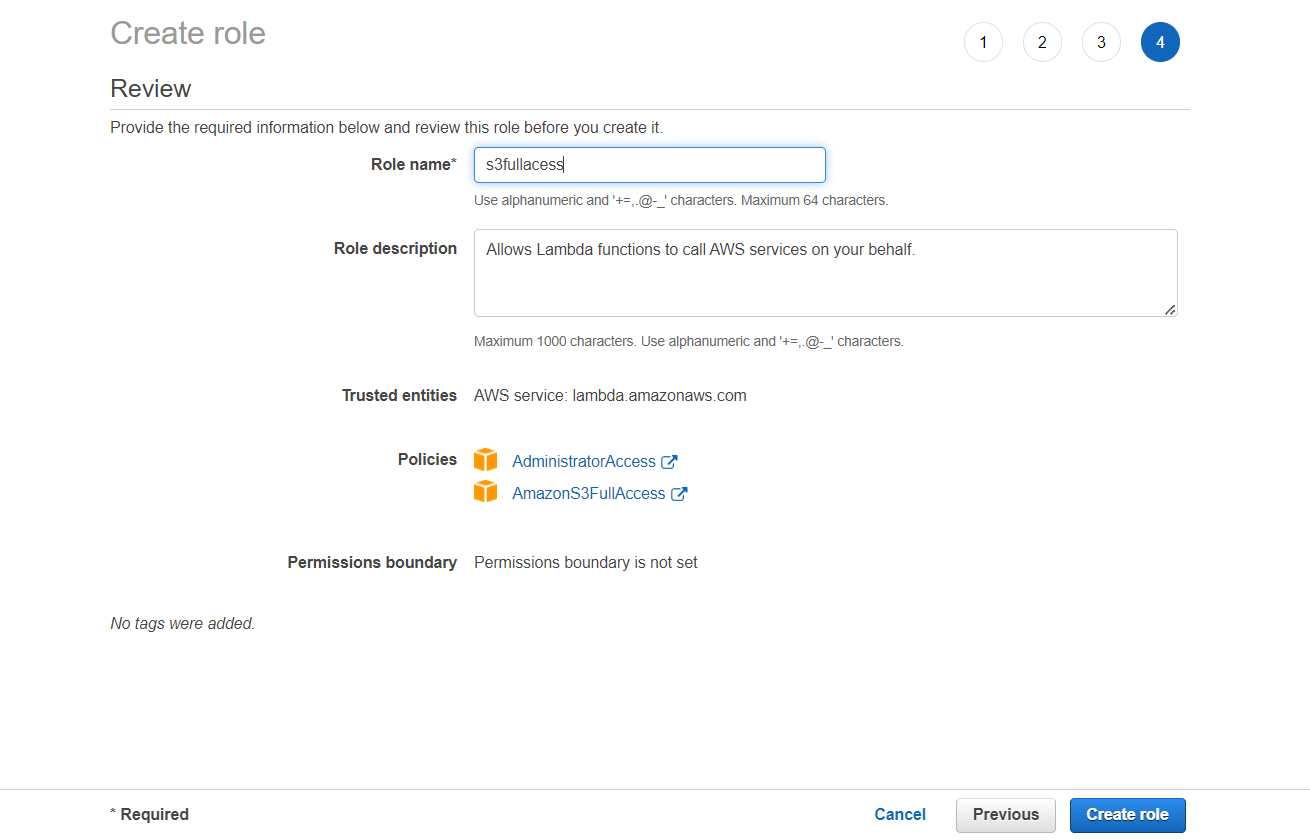
**Custom Role**



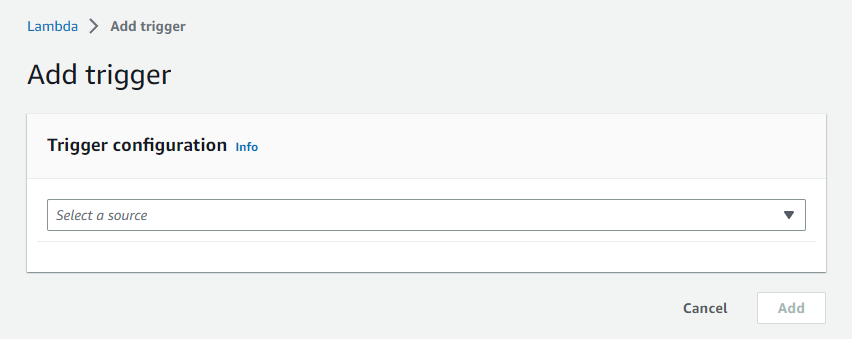
**Attaching Permissions**



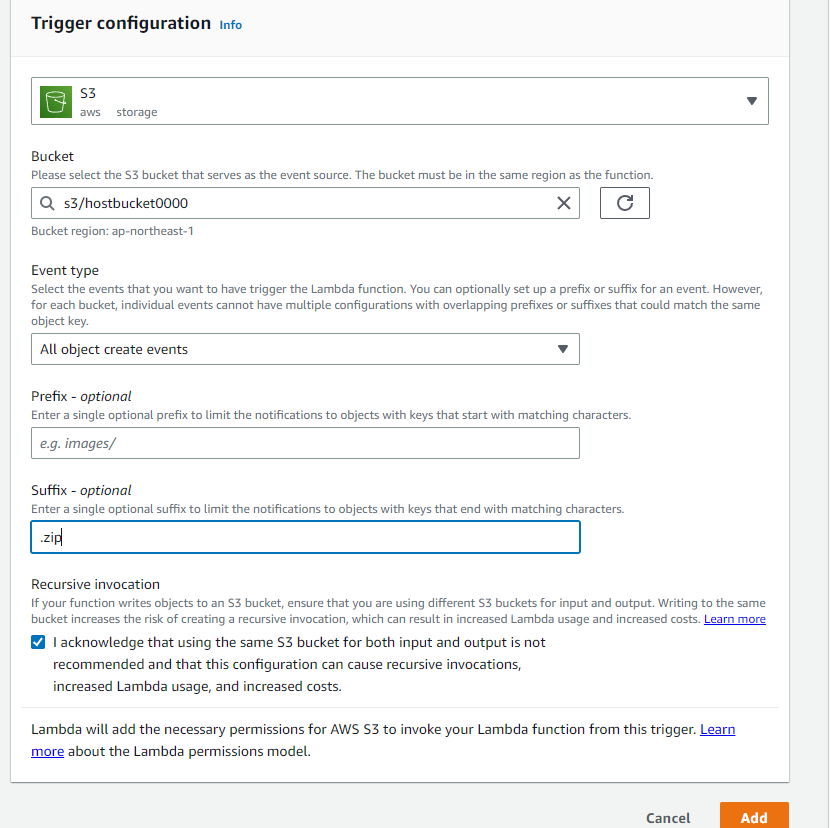
\*\*Note: Minimum access can be given (i.e., only read/write)



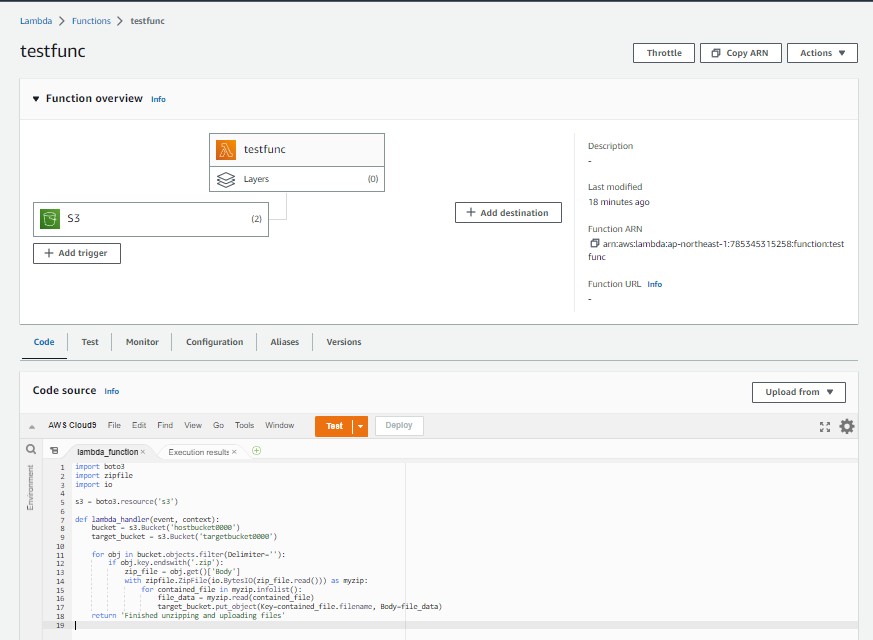
**Adding Trigger to the Function**



**Configuring Trigger**



**Testing the Code**



**Code**: Extracting zip file from Host s3 to Target s3 bucket

import boto3

import zipfile

import io

s3 = boto3.resource('s3')

def lambda\_handler(event, context):

bucket = s3.Bucket('hostbucket0000')

target\_bucket = s3.Bucket('targetbucket0000')

for obj in bucket.objects.filter(Delimiter=''):

if obj.key.endswith('.zip'):

zip\_file = obj.get()['Body']

with zipfile.ZipFile(io.BytesIO(zip\_file.read())) as myzip:

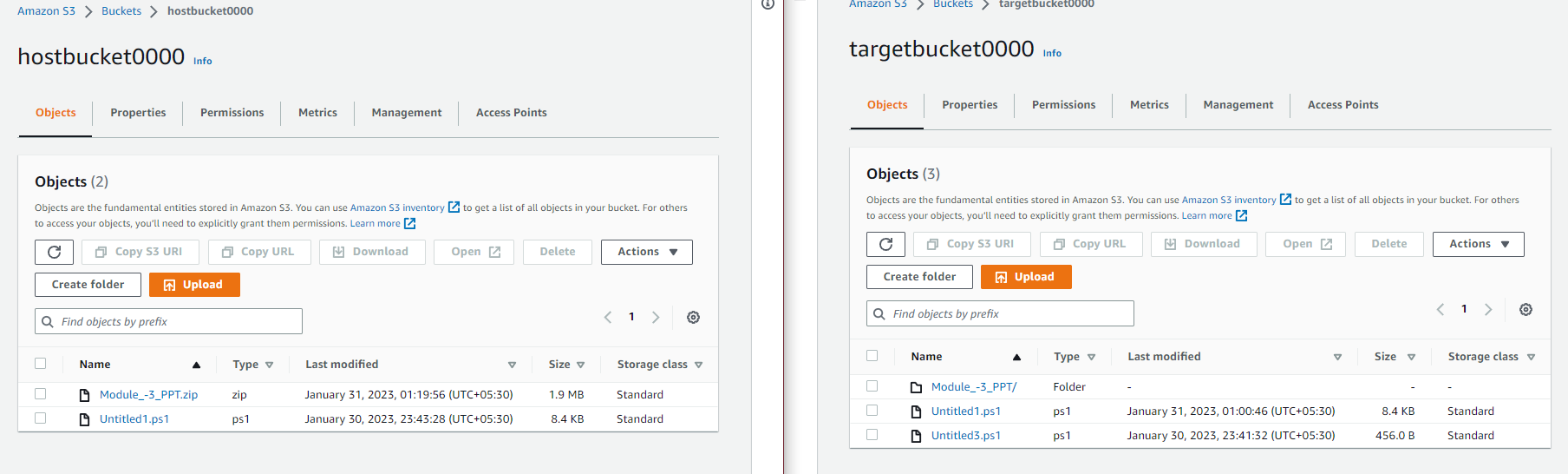
for contained\_file in myzip.infolist():

file\_data = myzip.read(contained\_file)

target\_bucket.put\_object(Key=contained\_file.filename, Body=file\_data)

return 'Finished unzipping and uploading files.’

**Output**

****

**Code:** Extracting and writing to File Type as XML

import boto3

import zipfile

import io

import xml.etree.ElementTree as ET

s3 = boto3.resource('s3')

def lambda\_handler(event, context):

bucket = s3.Bucket('hostbucket0000')

target\_bucket = s3.Bucket('targetbucket0000')

for obj in bucket.objects.filter(Delimiter=''):

if obj.key.endswith('.zip'):

zip\_file = obj.get()['Body']

with zipfile.ZipFile(io.BytesIO(zip\_file.read())) as myzip:

root = ET.Element("root")

files = ET.SubElement(root, "files")

for contained\_file in myzip.infolist():

ET.SubElement(files, "file", name=contained\_file.filename).text = contained\_file.filename

tree = ET.ElementTree(root)

xml\_data = ET.tostring(root).decode()

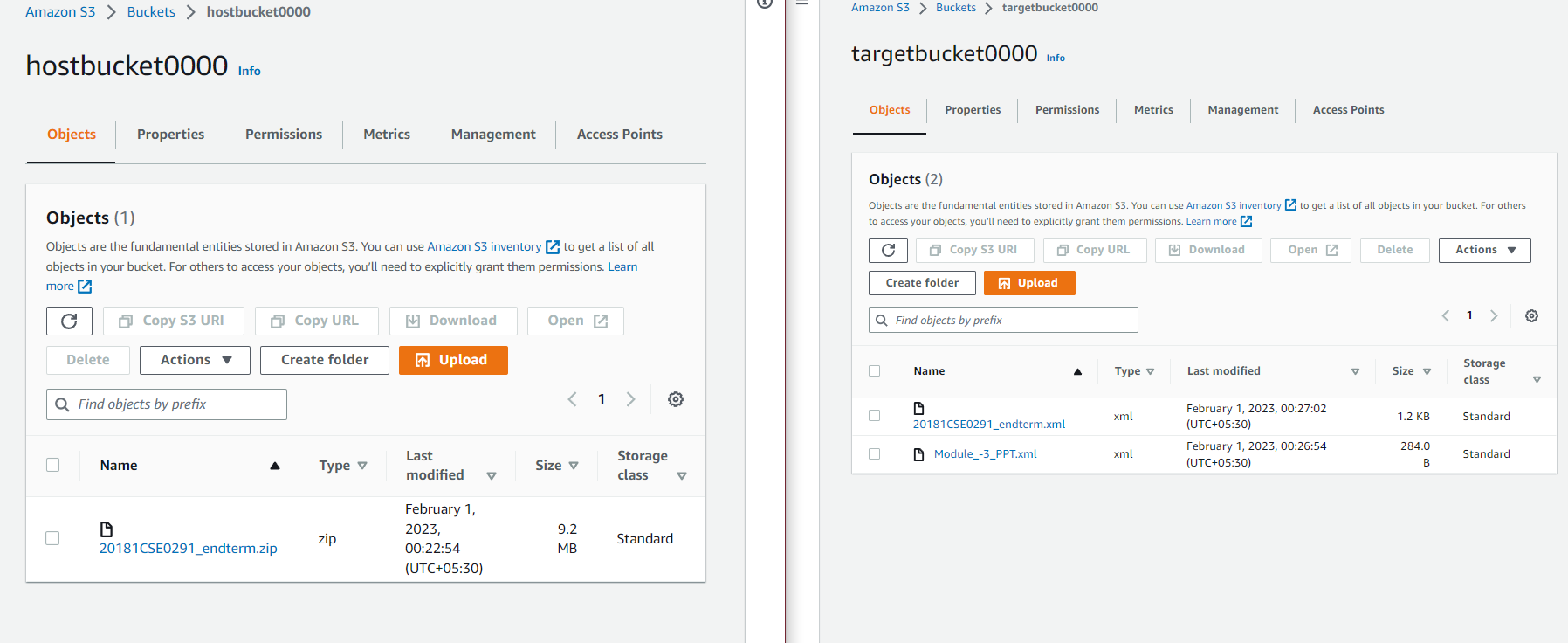
xml\_name = obj.key[:-4] + '.xml'

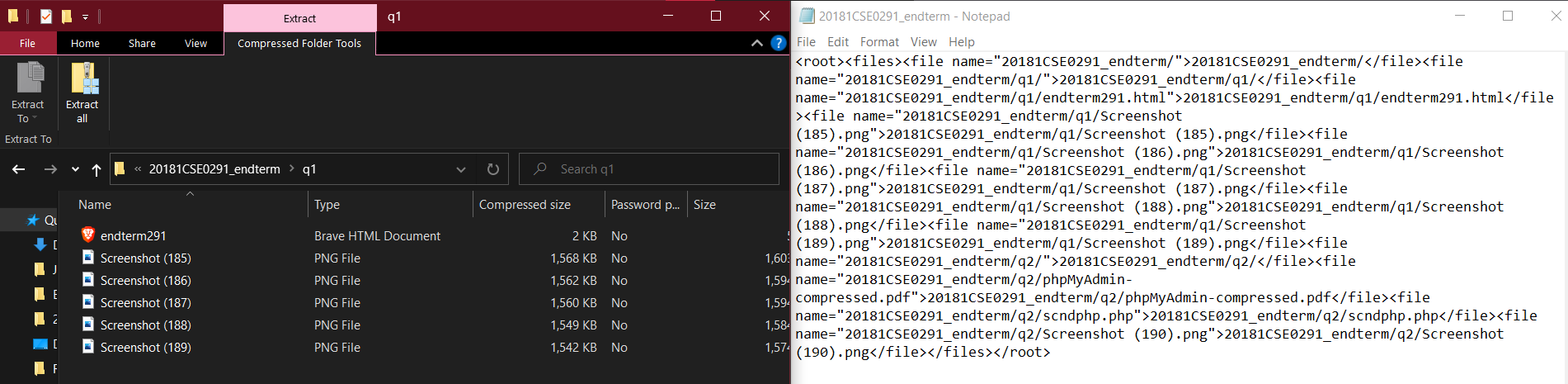
target\_bucket.put\_object(Key=xml\_name, Body=xml\_data)

return 'Finished writing names inside the zipped file to XML file'

\*\*Note: Editing out the code for extraction of zip file and adding code to write the names inside the zipped folder to a XML file (using etree.ElementTree). [The new XML file name being kept as old file + .xml (obj.key[:-4] + '.xml')]

**Output** : Files are converted to XML

****

****

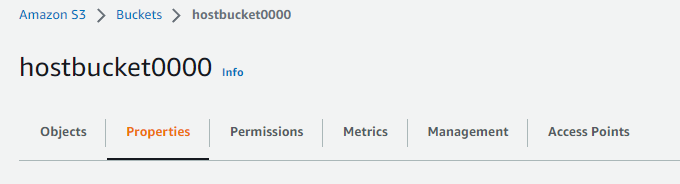
Additionally,

**Storing the Access logs**

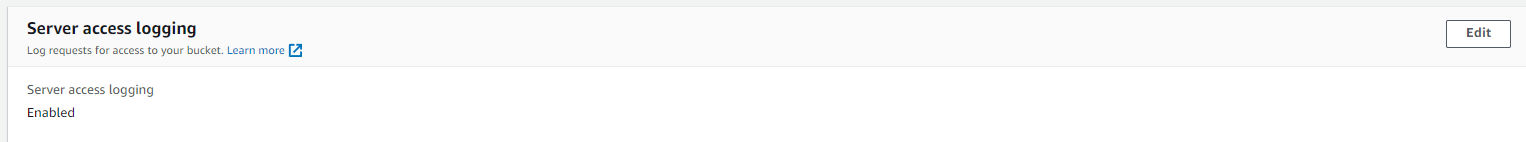
*Steps:*

S3

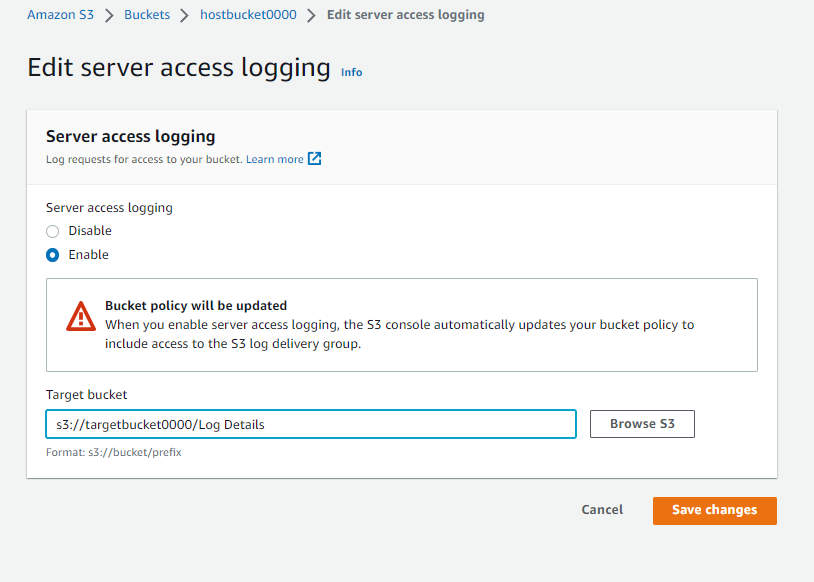
* Select Bucket
* Select Properties

****

* Server access Login

****

* Add the destination for logs

****

[ Workaround: Either a single S3 bucket can be configured [i.e... source\_bucket : source\_bucket] to get the desired outcome or data can be transferred to multiple buckets [i.e.. source\_bucket : dest\_bucket]. Provisioning of s3 buckets can be based according to Business requirement.]