# Assigment lab 1

## Creating bucket

Creating buckets can be done in the S3 tab.

A screenshot of a computer

Description automatically generated

Properly naming and picking the correct region. A screenshot of a bucket

Description automatically generated

## Creating files

Create the files and upload

A screenshot of a computer

Description automatically generated

## Making file publicly accessible

Enable ACLs on the bucket:

A screenshot of a computer

Description automatically generated

Change public access bucket settings: A screenshot of a computer

Description automatically generated

Make the object public:A screenshot of a computer

Description automatically generated

Check if the file is available:

A screenshot of a computer

Description automatically generated

Other files aren’t available:

A screenshot of a computer

Description automatically generatedEnable Versioning

A screenshot of a bucket versioning

Description automatically generated

Edit the file by downloading it, making changes and uploading it again.

The versions are visible through the versioning tab.

A screenshot of a computer

Description automatically generated

To roll back the version, download the latest version you want and reupload it, or delete the newer version until that one is the newest.

Delete the object:

A screenshot of a computer

Description automatically generated

With show version you can still see the deleted files:A screenshot of a computer

Description automatically generated

Delete file completely:

A screenshot of a computer

Description automatically generated

Versioning can not completely be turned off, only suspended:

A screenshot of a computer program

Description automatically generated

## Replicate to another region:

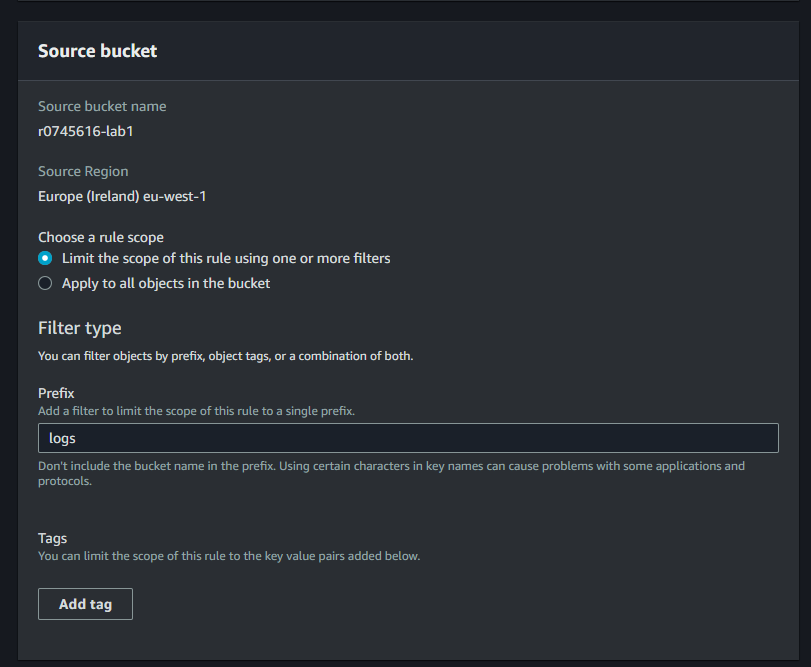
It is used to copy objects across Across S3 buckets in different AWS regions.

Create a bucket in a different region:

A screenshot of a bucket

Description automatically generated

Add replication rule only to specific folder:



Choose the newly created bucket:

A screenshot of a computer program

Description automatically generated

Existing files are not replicated automatically.

Old versions will not be replicated.

New versions will be replicated.

New files will be added automatically.

Will be deleted from both buckets (versioning will keep old versions).

Delete markers will also be replicated.

## Create lifecycle rule

Make new rule:

A screenshot of a computer

Description automatically generated

Define lifecycle rule actions:

A screenshot of a computer

Description automatically generated

The minimum time for the Standard-IA is 30 days and glacier 100 days.

## Encryption

Encryption is now on by default:

A screenshot of a computer

Description automatically generated

Yes this changes nothing about visibility.

## CLI copy

Can be done with a simple command:

aws s3 cp .\01.txt s3://r0745616-lab1/logs/mar/01.txt.



They are replicated automatically.

Overwrite the file:

aws s3 cp .\01.txt s3://r0745616-lab1/logs/jan/01.txt

The old versions will still be there.

### Batch script

Here is the example bash script to create a bucket.

aws s3api create-bucket --bucket $name --region us-east-1

aws s3api put-object --bucket test-bash-2 --key logs/jan/02.txt --body logs/jan/02.txt

aws s3api put-object --bucket test-bash-2 --key logs/jan/01.txt --body logs/jan/01.txt

aws s3api put-bucket-versioning --bucket $name --versioning-configuration Status=Enabled

## Static site

A static site can easily be configured on the new bucket: A screenshot of a web hosting

Description automatically generated

The files need to be set publicly accessible, as done earlier.

The static page:

A screen shot of a computer

Description automatically generated

The error page:

A screenshot of a computer

Description automatically generated

The URL of the site:  
<http://site-bucket-r0745616.s3-website-eu-west-1.amazonaws.com>

## Bash Static Site Script

aws s3api create-bucket --bucket %1 --create-bucket-configuration LocationConstraint=eu-west-1 --object-ownership BucketOwnerPreferred

aws s3api put-public-access-block --bucket %1 --public-access-block-configuration "BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=false,RestrictPublicBuckets=false"

aws s3api put-bucket-acl --bucket %1 --acl public-read

aws s3 cp index.html s3://%1

aws s3 cp error.html s3://%1

aws s3api put-object-acl --bucket %1 --key index.html --acl public-read

aws s3api put-object-acl --bucket %1 --key error.html --acl public-read

aws s3 website s3://%1 --index-document index.html --error-document error.html

echo Website address: http://%1.s3-website-us-east-1.amazonaws.com

## Python Script

A python script that logs processor usage, and hosts it as a static web site:

import datetime  
import boto3  
import psutil  
from botocore.exceptions import NoCredentialsError  
  
  
# Function to log CPU usage  
def log\_cpu\_usage():  
 cpu = []  
 for \_ in range(10):  
 cpu\_percent = psutil.cpu\_percent(interval=1)  
 cpu.append(cpu\_percent)  
 return cpu  
  
  
# Function to create an HTML file with CPU usage data  
def create\_html\_report(cpu):  
 with open('results.html', 'w') as f:  
 f.write("<html><body>")  
 f.write("<h1>CPU Usage Report</h1>")  
 f.write("<ul>")  
 for idx, usage in enumerate(cpu):  
 f.write(f"<li>Second {idx + 1}: {usage}%</li>")  
 f.write("</ul>")  
 f.write("</body></html>")  
  
  
# Function to upload the HTML file to S3  
def upload\_to\_s3(html\_file):  
 s3 = boto3.client('s3')  
 bucket\_name = 'r0745616-lab1'  
 key = 'results.html'  
  
 # Set the content type to text/html  
 extra\_args = {'ContentType': 'text/html'}  
  
 try:  
 s3.upload\_file(html\_file, bucket\_name, key, ExtraArgs=extra\_args)  
 print(f"Uploaded {html\_file} to S3 bucket: {bucket\_name}/{key}")  
 make\_public(s3, bucket\_name, key)  
 allow\_static\_site(s3, bucket\_name, key)  
 except FileNotFoundError:  
 print(f"The file {html\_file} was not found.")  
 except NoCredentialsError:  
 print("AWS credentials not found.")  
  
  
# Function that makes the file publicly accessible through a web browser  
def make\_public(s3, bucket\_name, key):  
 s3.put\_object\_acl(  
 Bucket=bucket\_name,  
 Key=key,  
 ACL='public-read'  
 )  
  
  
# Changes the bucket properties so it hosts results.html as a static site  
def allow\_static\_site(s3, bucket\_name, key):  
 s3.put\_bucket\_website(  
 Bucket=bucket\_name,  
 WebsiteConfiguration={  
 'IndexDocument': {'Suffix': key},  
 }  
 )  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 cpu\_usage = log\_cpu\_usage()  
 create\_html\_report(cpu\_usage)  
 upload\_to\_s3('results.html')