# 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

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 public:

A screenshot of a computer

Description automatically generated

De dns waarop deze accessible is:

<https://r0745616-lab1.s3.eu-west-1.amazonaws.com/apps/ap01/config.txt>

A screenshot of a computer

Description automatically generated

De andere files zijn niet available:

A screenshot of a computer

Description automatically generated

## Enable Versioning

A screenshot of a bucket versioning

Description automatically generated

Edit file door die te downloade, aan te passen, en terug te uploaden.

De versions zijn visible door de versions tab:

A screenshot of a computer

Description automatically generated

Om die te rollbacken download je de vorige versie en upload je deze opnieuw. Of je verwijdert de nieuwe versie zodat degene die je wilt de meest recente wordt.

Delete de file: A screenshot of a computer

Description automatically generated

Met show versions kunt ge de deleted files zienA screenshot of a computer

Description automatically generated

### 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

### 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')