

# LAB: Host a static website in a versioned encrypted S3 bucket

## You need:

- An AWS Account

**Duration of the Lab:** 20 Minutes.

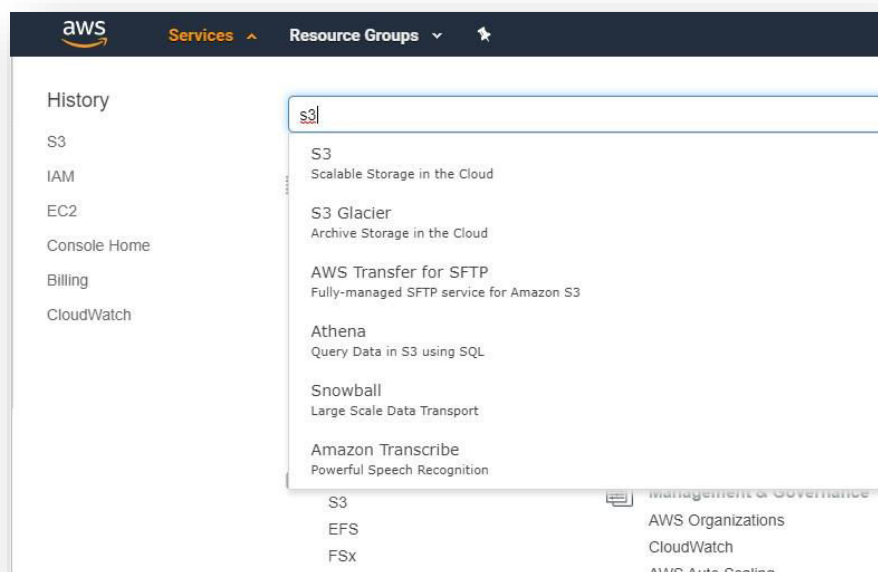
**Difficulty:** medium

## Contents

Create a Bucket.....	1
Versioning.....	3
Encryption.....	4
Static Website Hosting.....	4
Upload a file .....	5
Open the Website.....	8
Update the Version.....	9
Bucket Policy.....	13

## Create a Bucket

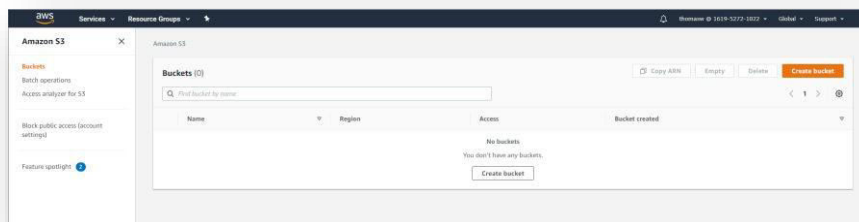
Select the S3 Service from your AWS Dashboard



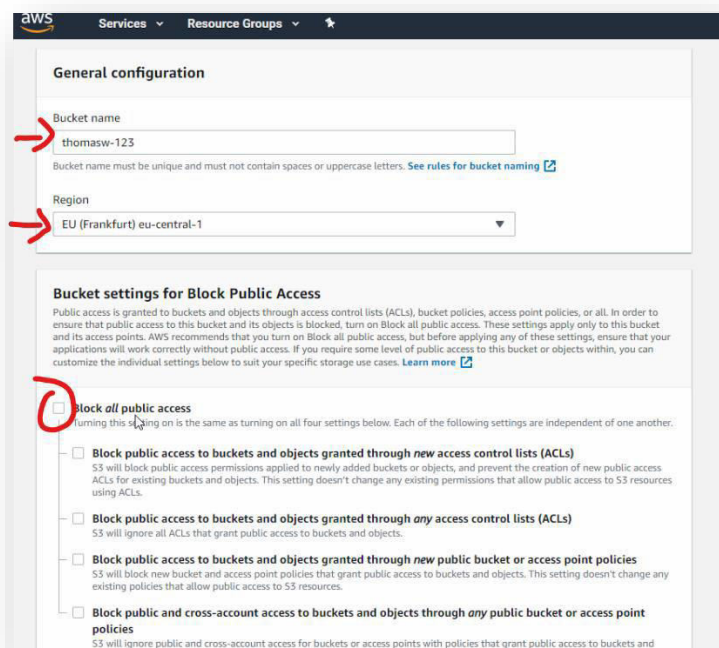
Mind the Region-Selector jumped from your region to “Global”. S3-Buckets are in a global scope.



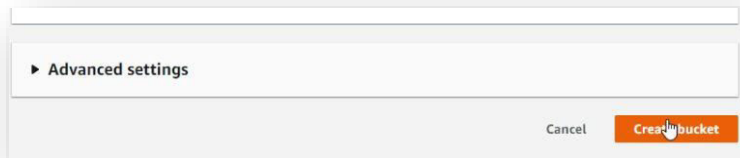
Create a new Bucket



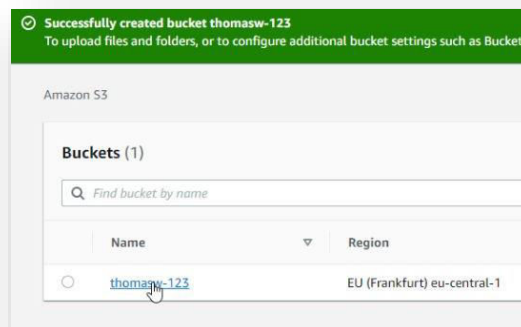
Give the bucket a name, select the region where your infrastructure is located at and de-select block public access:



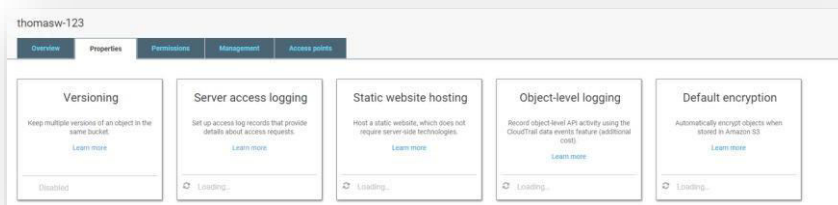
Click “Create Bucket”



You will be redirected to the Dashboard, open your new Bucket:

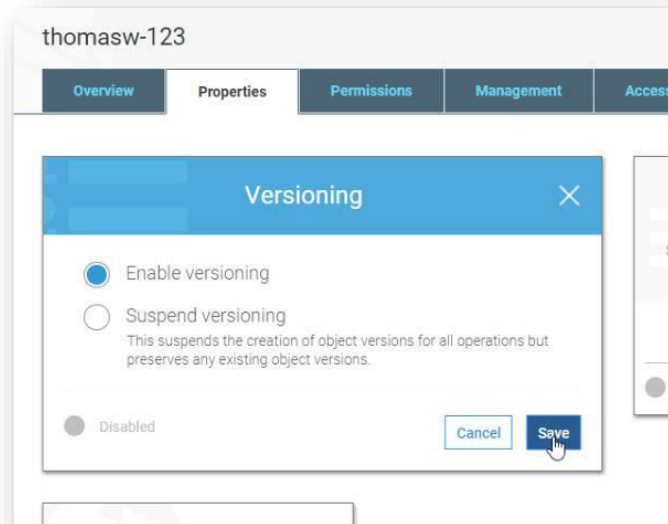


Head over to the Bucket Properties:



## Versioning

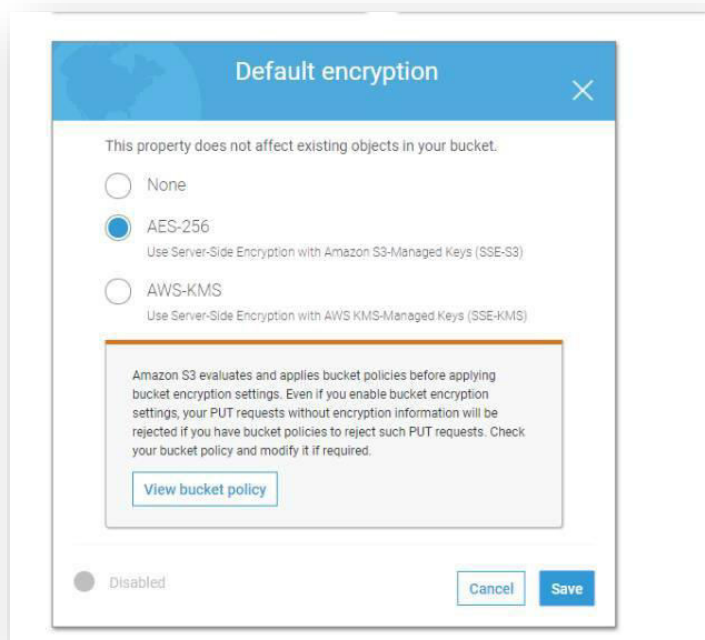
Turn on bucket versioning:



Versioning will give every new upload a version ID, generated by amazon. If you delete an object then it will be flagged and this will be a version as well.

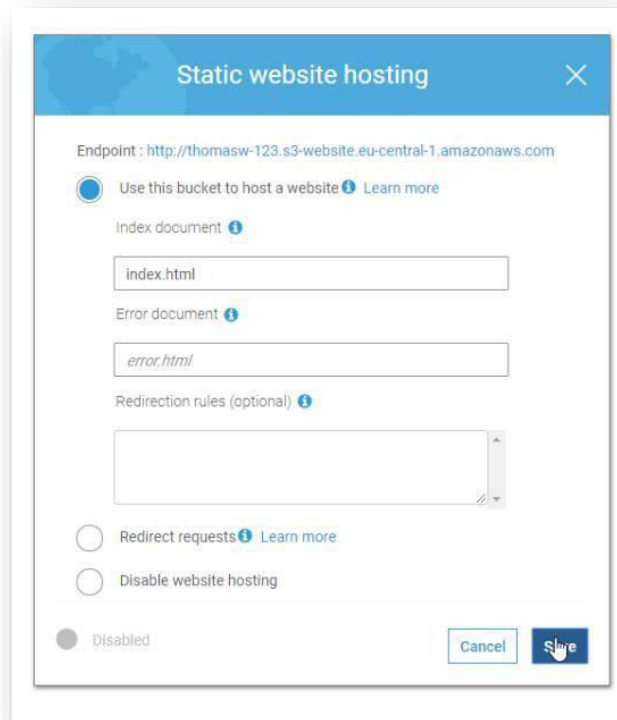
### Encryption

Turn on encryption. Although it is a bit far-fetched to encrypt a publicly accessible bucket, we still turn it on to get a feeling how that works:



### Static Website Hosting

Next, turn on static website hosting and enter index.html as a default file:

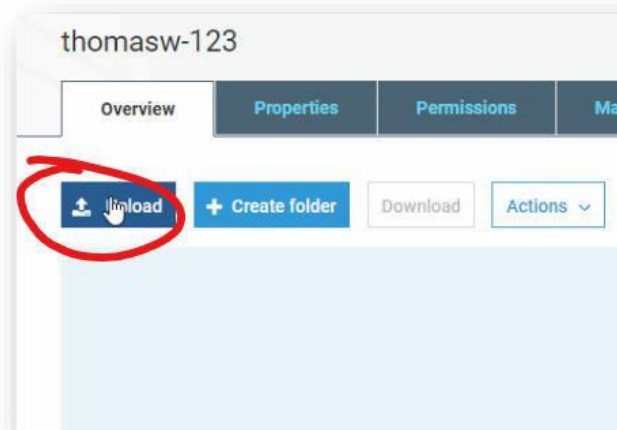


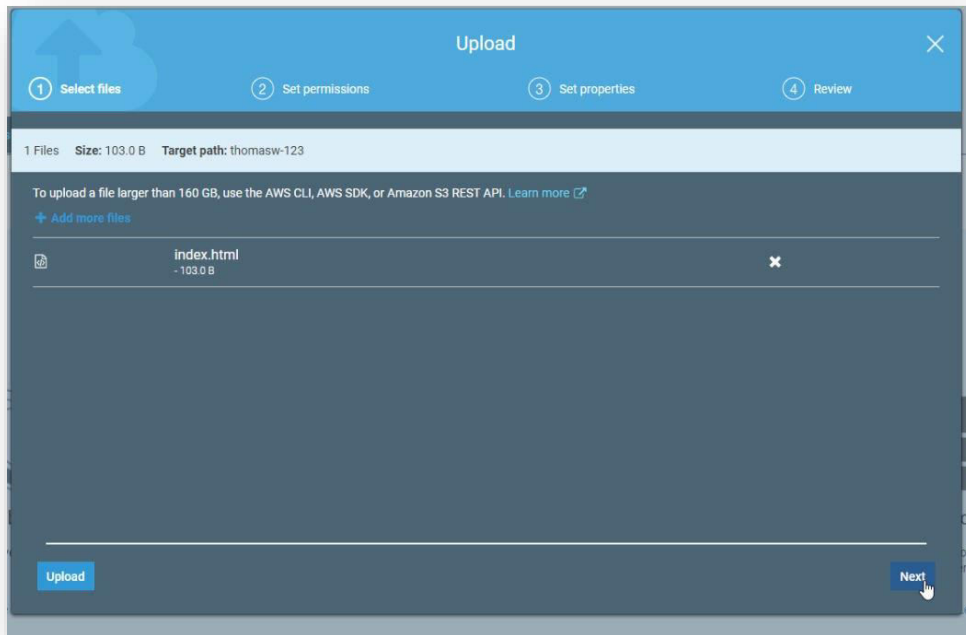
## Upload a file

Create a new file on your computer called "index.html" and fill it with the following content:

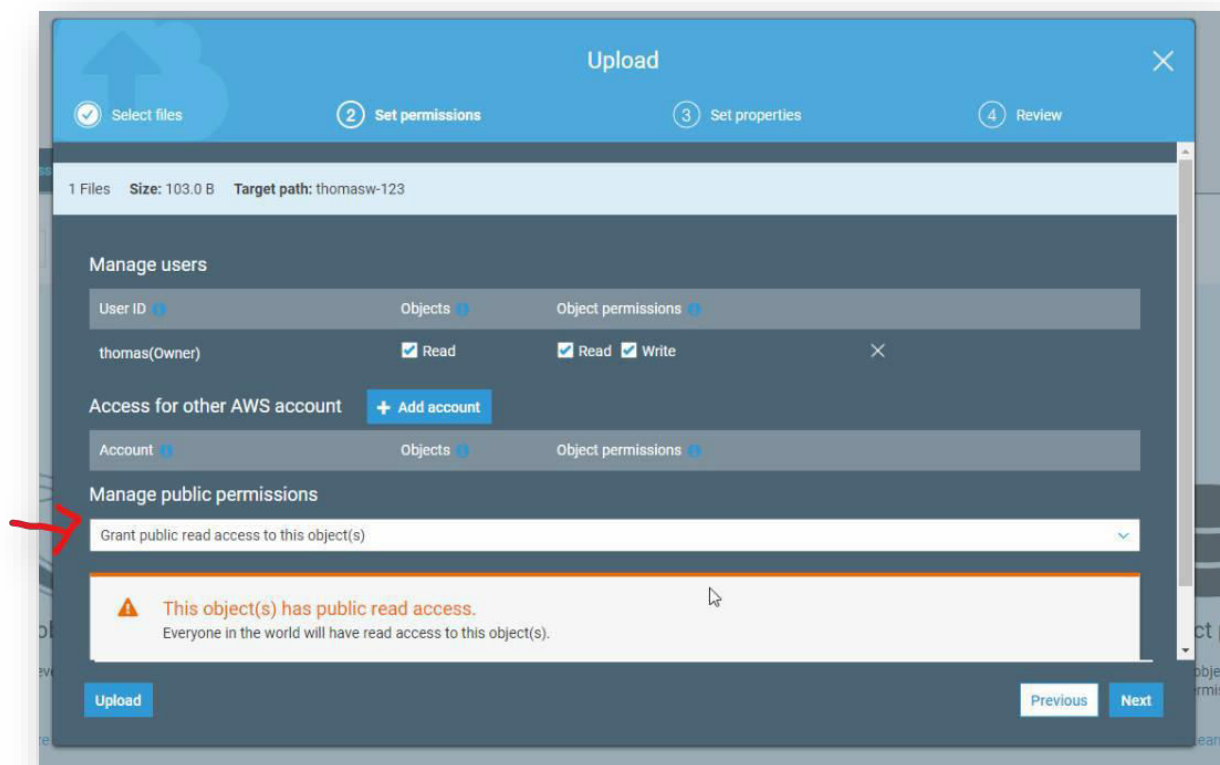
```
<!DOCTYPE html>
<head>
  <title>Test Site</title>
</head>
<body>
  Hello from S3 - V1
</body>
```

Then upload the file using the S3 Dashboard:

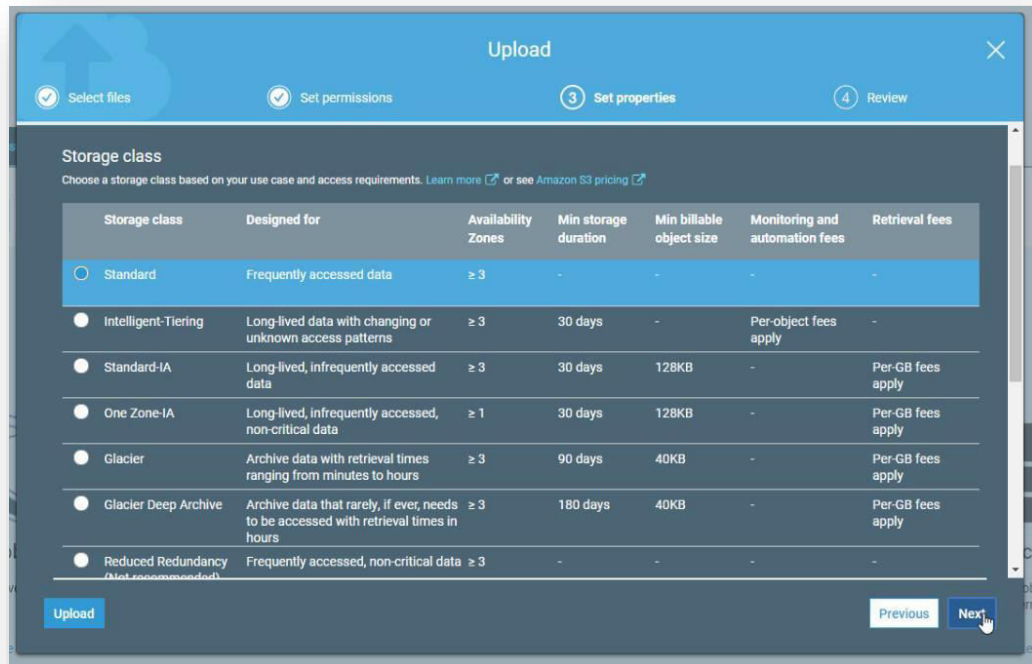




Give public read access:



Select the standard storage class:



**Upload**

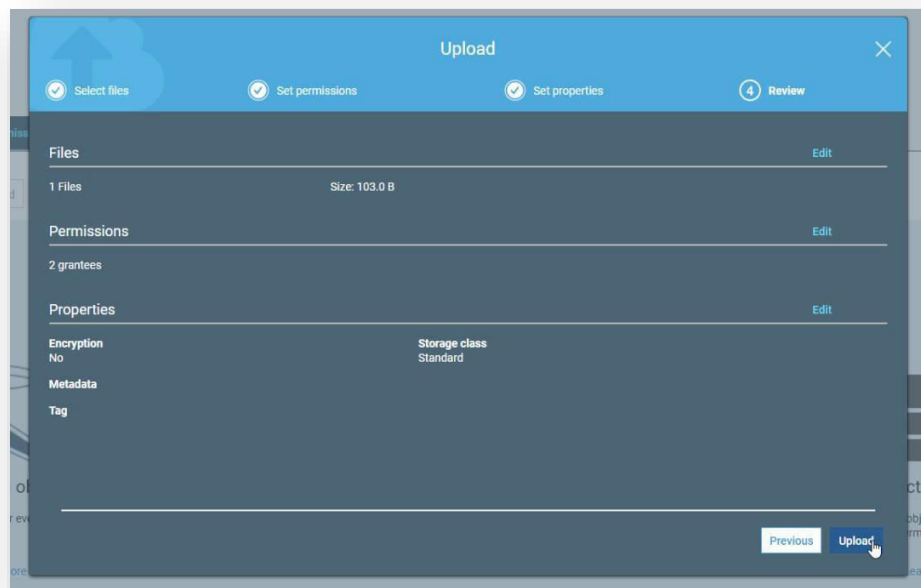
1 Select files 2 Set permissions 3 Set properties 4 Review

**Storage class**  
Choose a storage class based on your use case and access requirements. [Learn more](#) or see [Amazon S3 pricing](#)

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
<input type="radio"/> Standard	Frequently accessed data	≥ 3	-	-	-	-
<input type="radio"/> Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
<input type="radio"/> Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	40KB	-	Per-GB fees apply
<input type="radio"/> Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in hours	≥ 3	180 days	40KB	-	Per-GB fees apply
<input type="radio"/> Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data	≥ 3	-	-	-	-

**Upload** Previous **Next**

Review the upload the confirm:



**Upload**

1 Select files 2 Set permissions 3 Set properties 4 Review

**Files** Edit

1 Files Size: 103.0 B

**Permissions** Edit

2 grantees

**Properties** Edit

**Encryption**  
No

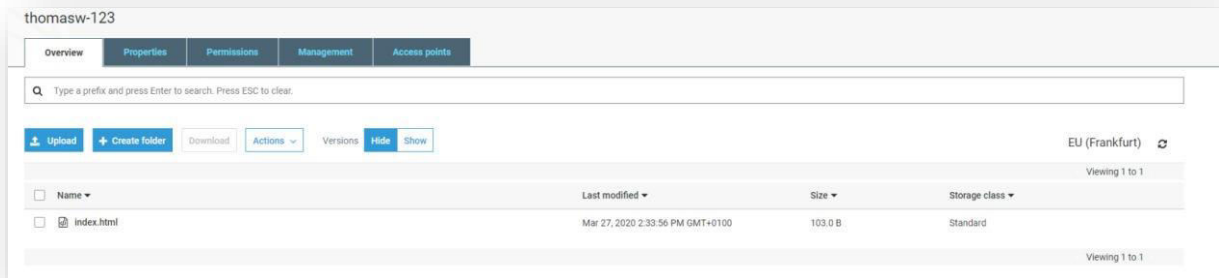
**Storage class**  
Standard

**Metadata**

**Tag**

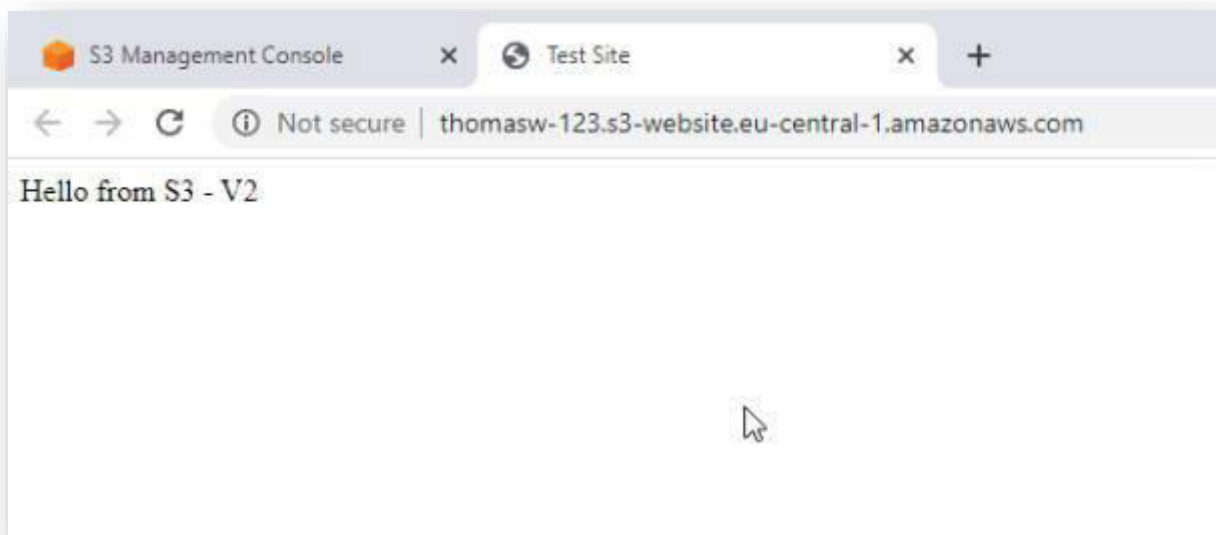
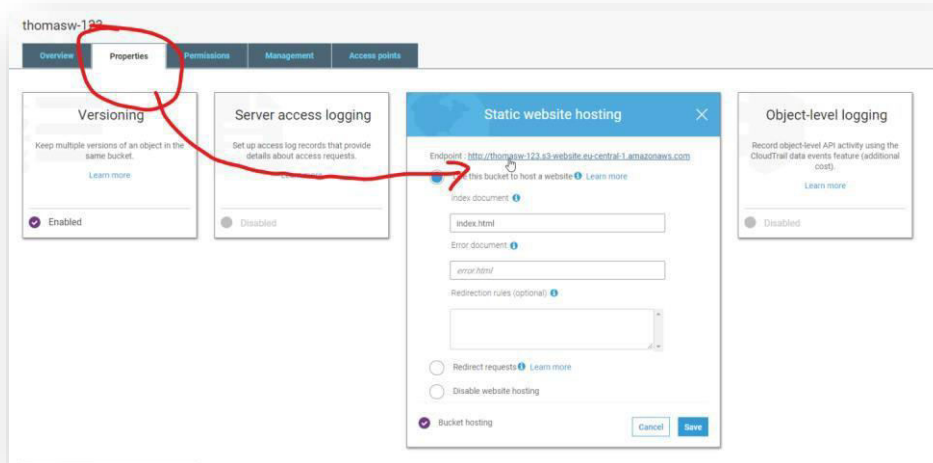
Previous **Upload**

The file should appear in the overview:



### Open the Website

Head over to the Properties and open the website:

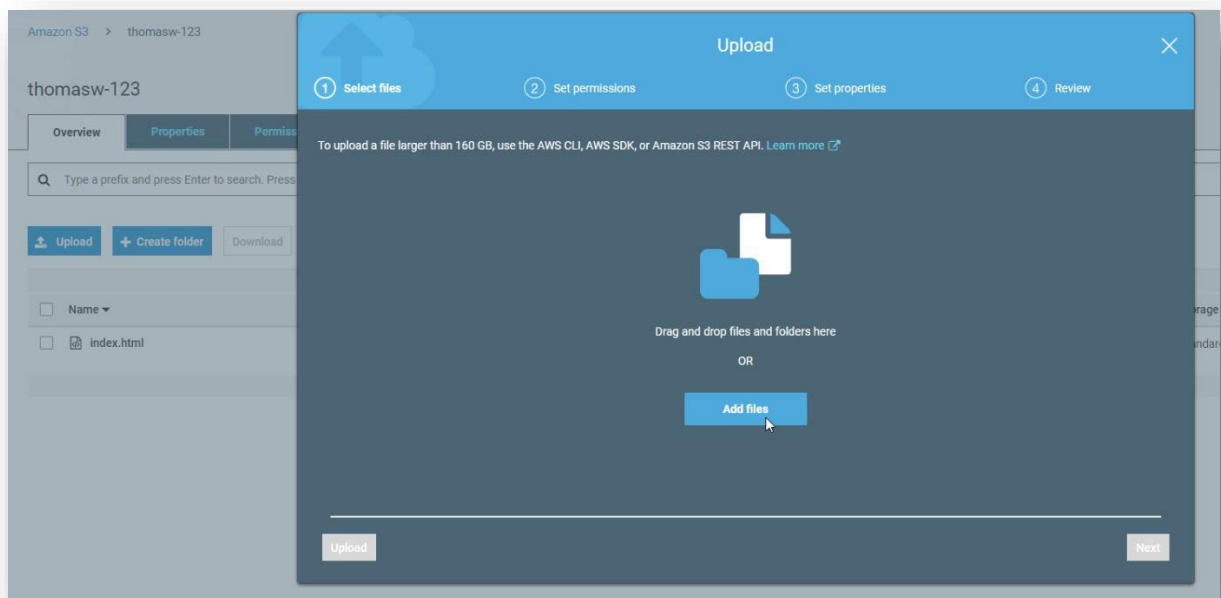




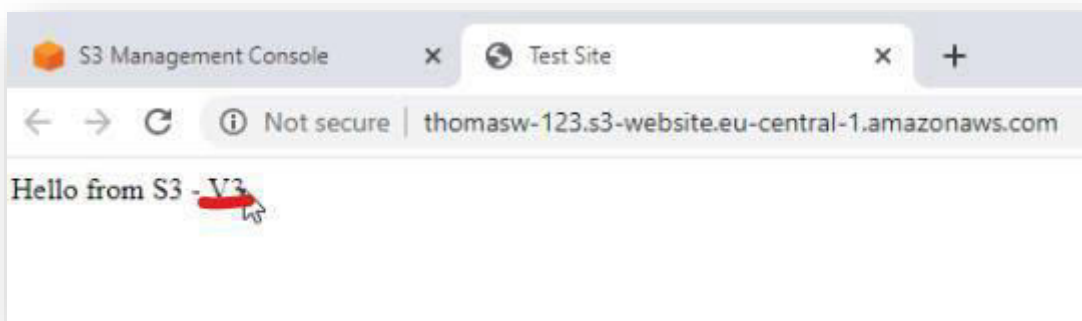
## Update the Version

```
<> index.html ●
<> index.html > body
1  <!DOCTYPE html>
2  <head>
3    <title>Test Site</title>
4  </head>
5  <body>
6    Hello from S3 - V3
7  </body>
```

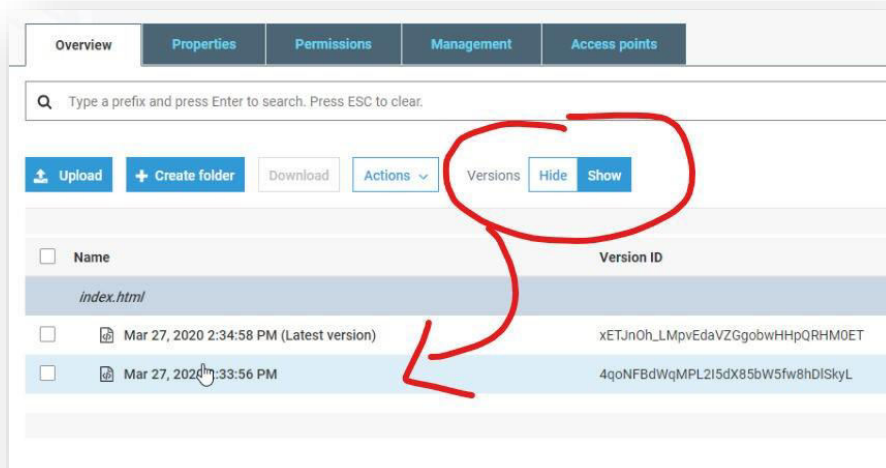
Upload the updated file again:



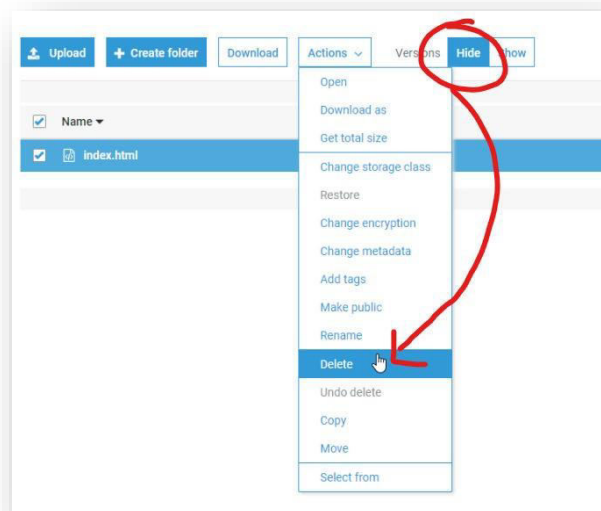
It immediately updates on the website too:



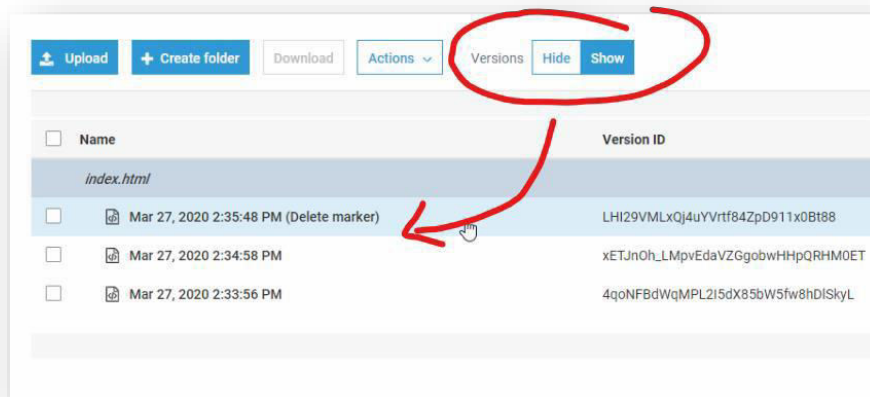
Show the versions:



Hide the versions and delete the file:



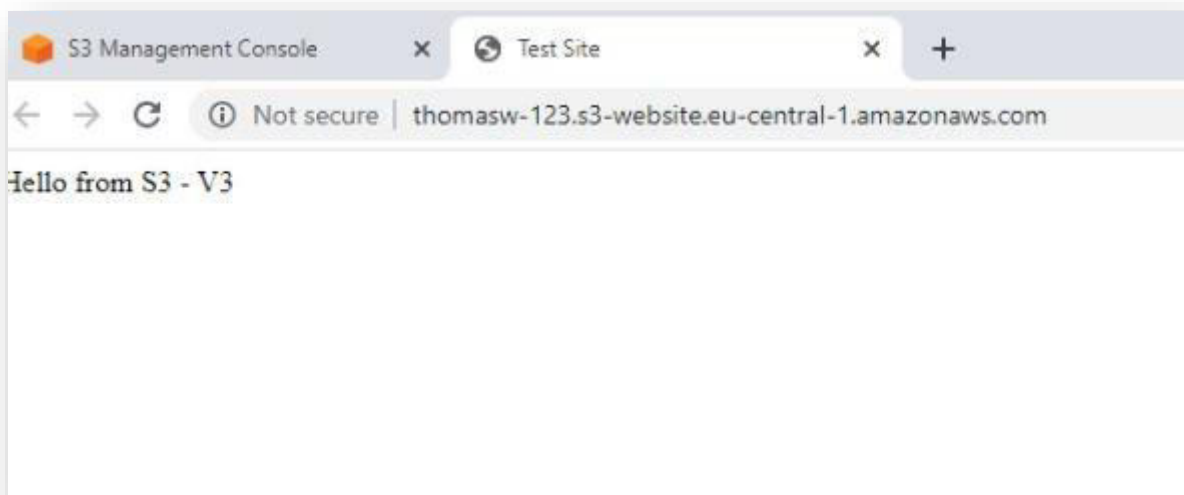
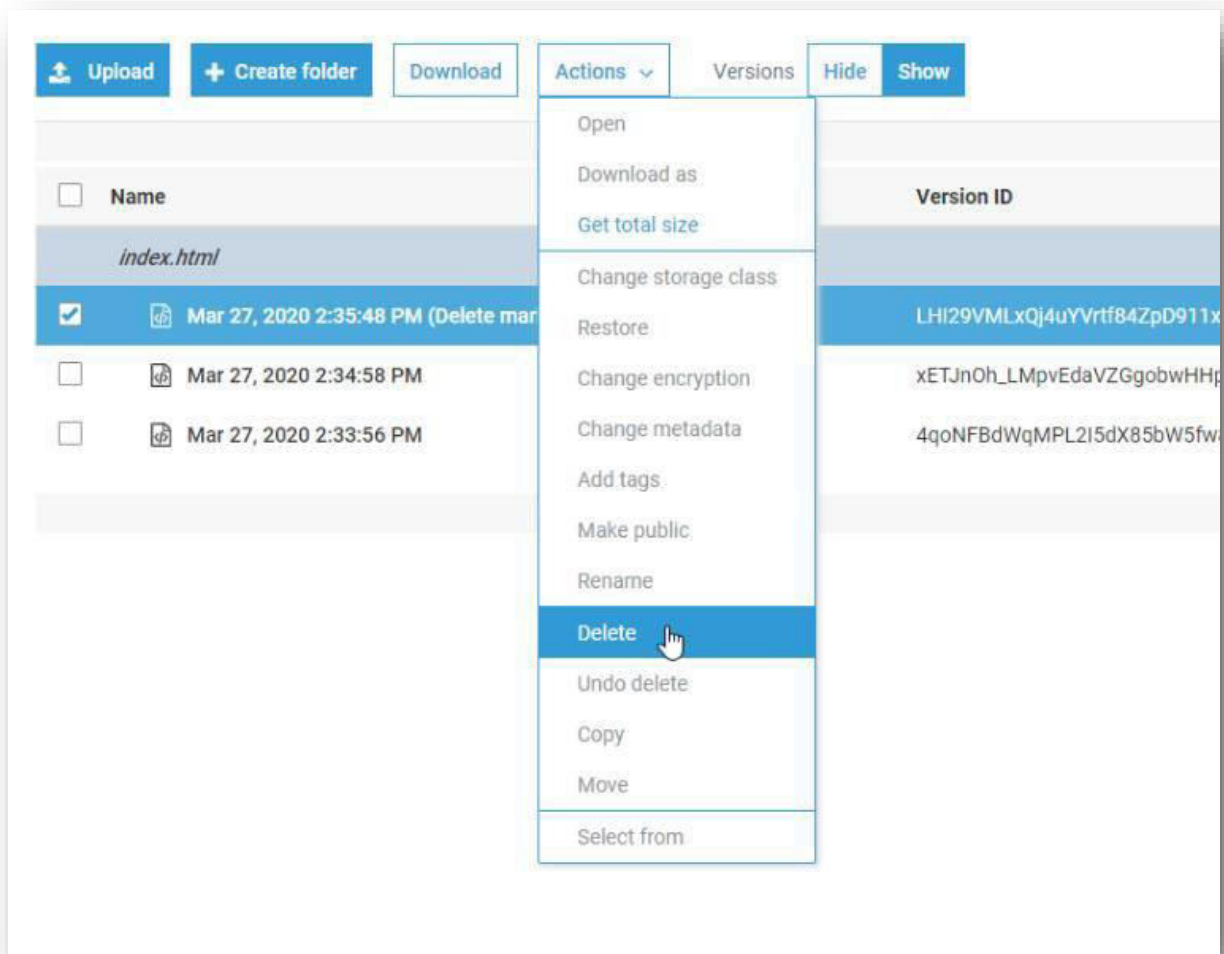
Show the versions again and see the file is still there, with a delete marker:



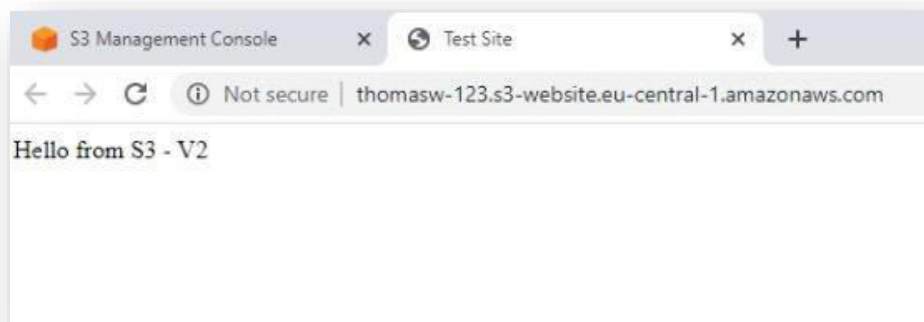
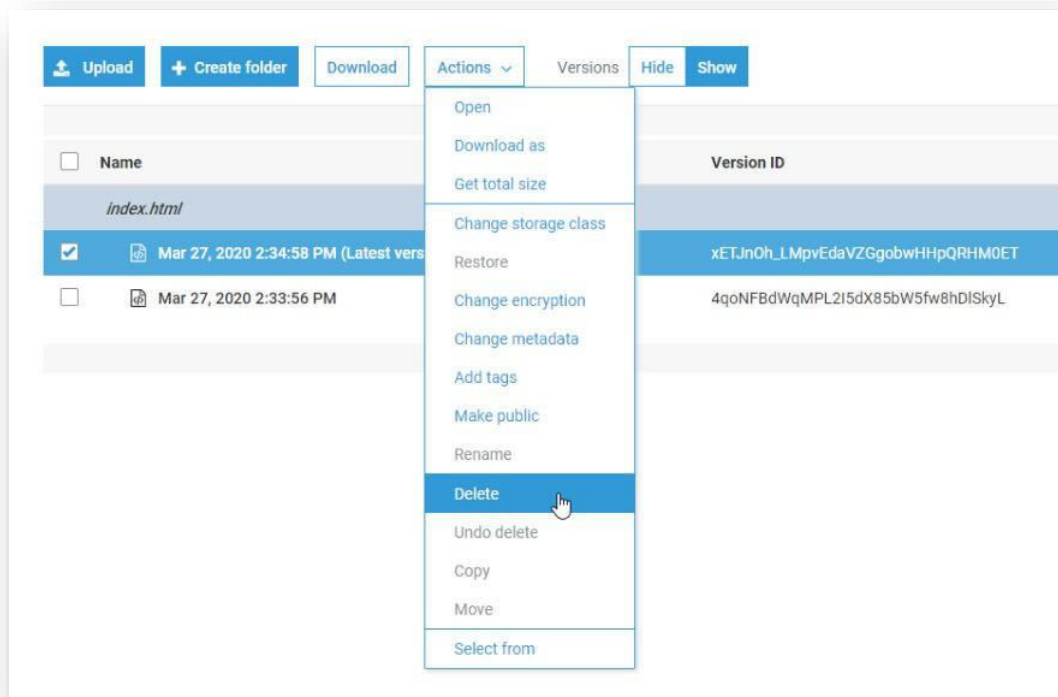
When you open the website, the file is really gone:



But when you delete the “delete marker” the file is back again:

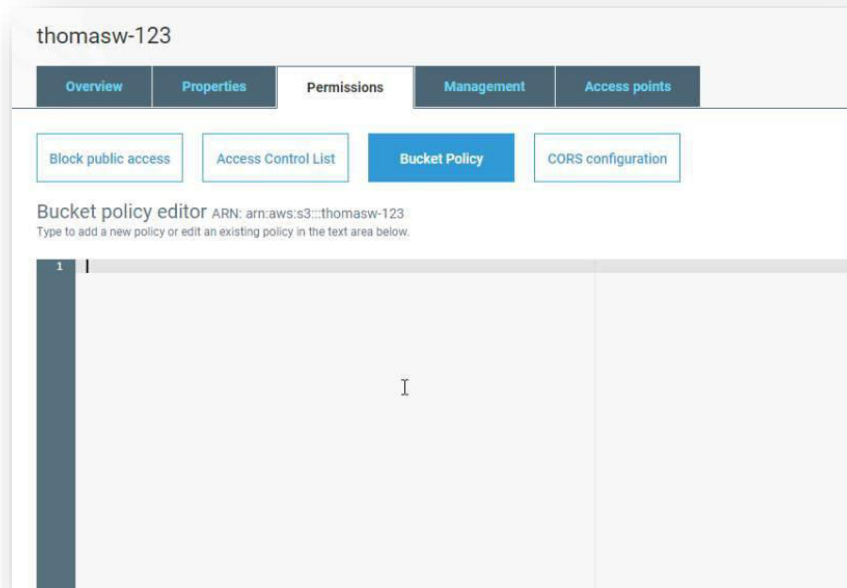


You can delete another version to go back to the first upload:



## Bucket Policy

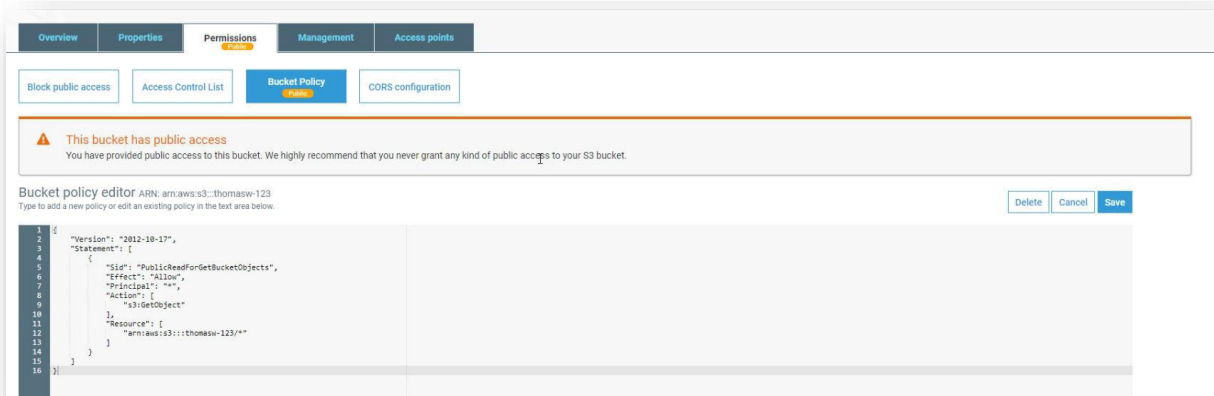
Uploading a single file and setting permissions to public for a single file is maybe not what you want for a static website. So, let's add a bucket policy that grants everyone read access to all objects in this bucket:



Add the following policy, but substitute your bucket name:

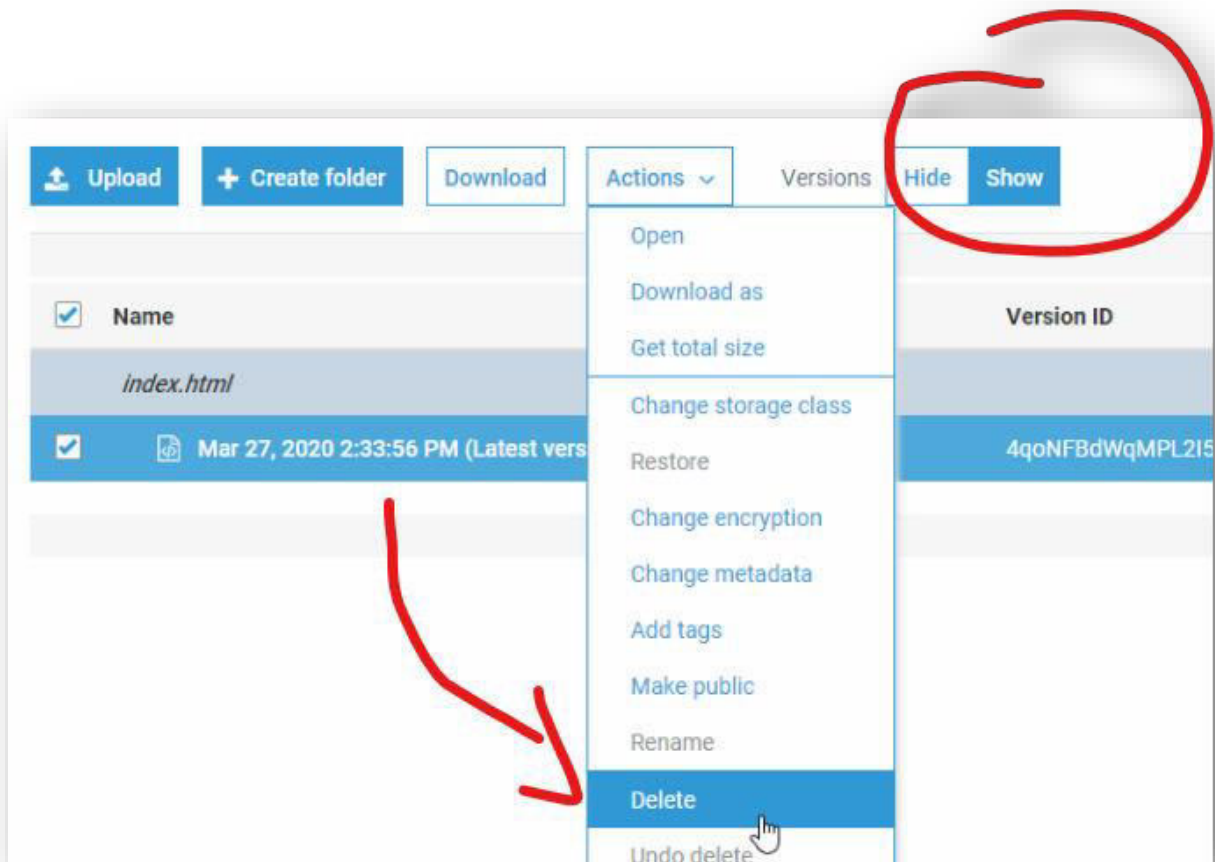
```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": "*",
    "Action": ["s3:GetObject"],
    "Resource": ["arn:aws:s3::: example-bucket / *"]
  } ]
}
```

AWS will show you that the bucket is public:

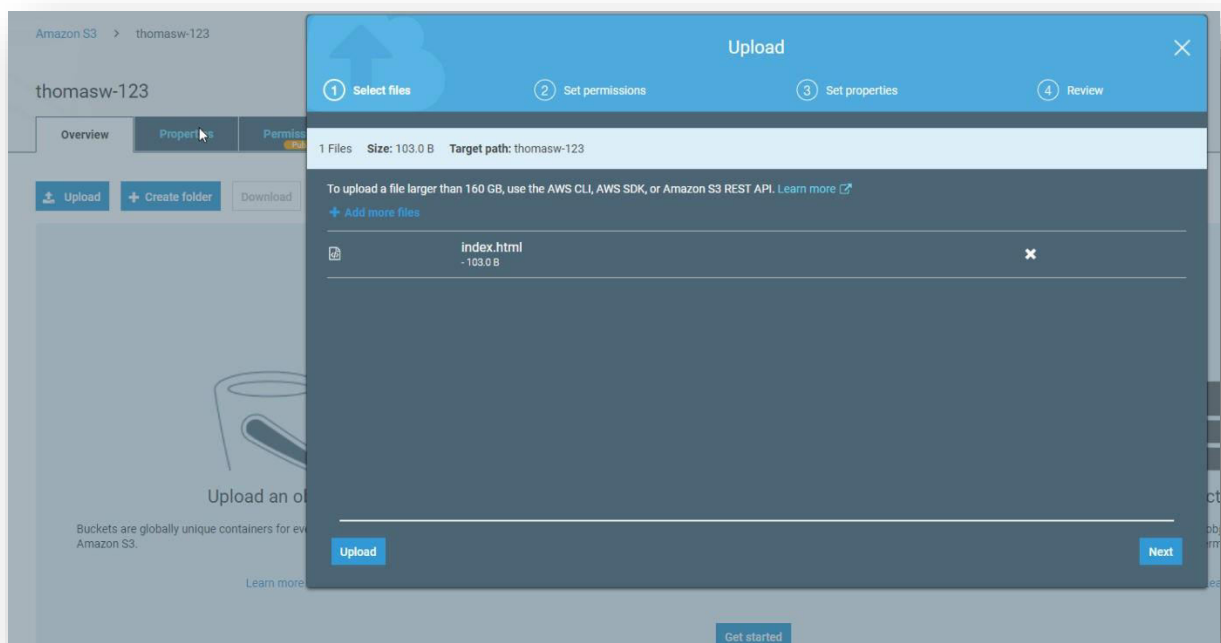


Let's test...

Now completely delete the file



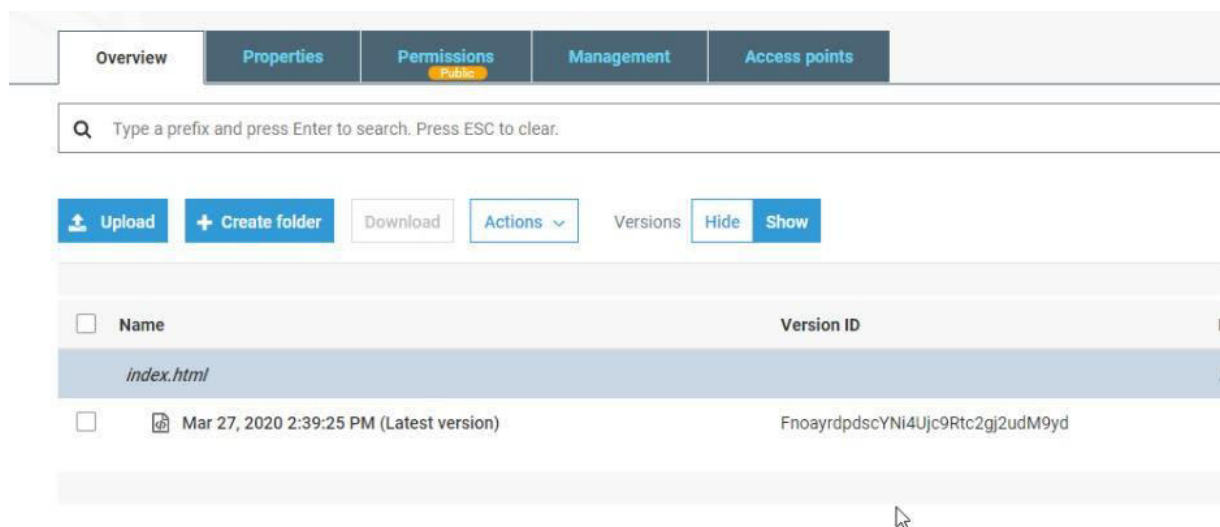
Re-Upload the index.html file:



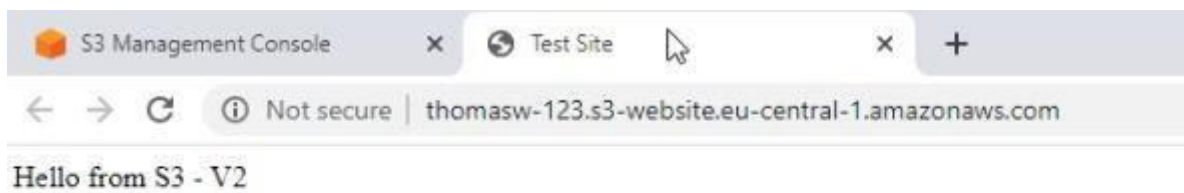
But do not grant any special permissions:



As soon as the file is uploaded:



Reload the browser:



It works just fine.

Before you delete everything wait for the next lab to remove everything via the CLI



---

*Lab End*

---