

LAB: Access S3 via CLI from EC2 with IAM Roles and Policies

You need:

- An AWS Account
- An S3 Bucket with Static Hosting Enabled

Duration of the Lab: 30 Minutes.

Difficulty: medium

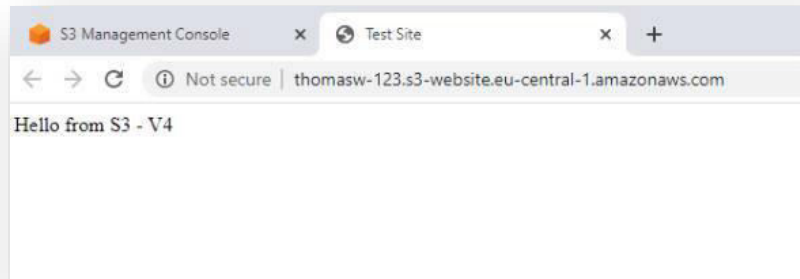
Contents

S3 CLI	2
Create a new Role.....	2
Launch an EC2 Instance	5
Add an Inline Policy	6
S3 CLI with inline Policy	9
Cleanup.....	11

S3 CLI

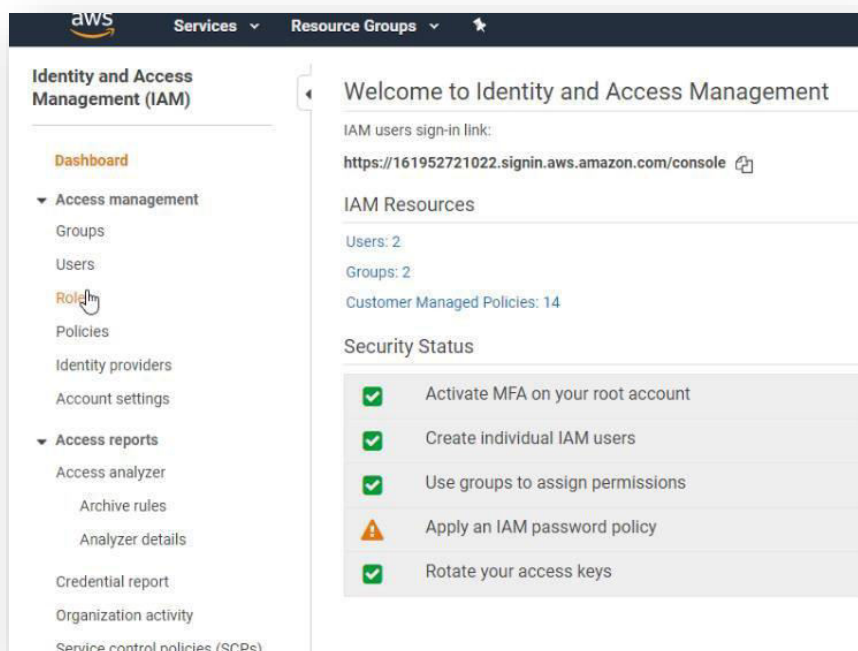
We will use an EC2 Instance with IAM permissions to access your S3 Bucket.

Make sure the S3 statically hosted Website is still here:

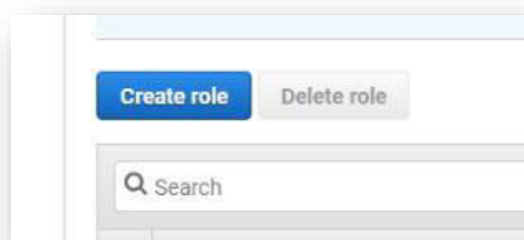


Create a new Role

First, let's create a new IAM Role. Go to the IAM Dashboard -> Roles:



Create a new role:



Choose AWS-Service and EC2 Role:

Create role

1234

Select type of trusted entity

AWS service
 EC2, Lambda and others

Another AWS account
 Belonging to you or 3rd party

Web identity
 Cognito or any OpenID provider

SAML 2.0 federation
 Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
 Allows EC2 instances to call AWS services on your behalf.

Lambda
 Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeDeploy	EMR	KMS	RoboMaker
AWS Backup	CodeGuru	ElastiCache	Kinesis	S3
AWS Chatbot	CodeStar Notifications	Elastic Beanstalk	Lambda	SMS
AWS Support	Comprehend	Elastic Container Service	Lex	SNS
Amplify	Config	Elastic Transcoder	License Manager	SWF
AppStream 2.0	Connect	ElasticLoadBalancing	Machine Learning	SageMaker
AppSync	DMS	Forecast	Macie	Security Hub
Application Auto Scaling	Data Lifecycle Manager	Global Accelerator	MediaConvert	Service Catalog
Application Discovery Service	Data Pipeline	Glue	Migration Hub	Step Functions
Batch	DataSync	Greengrass	OpsWorks	Storage Gateway
Chime	DeepLens	GuardDuty	Personalize	Textract
CloudFormation	Directory Service	Health Organizational View	QLDB	Transfer
	DynamoDB	IAM Access Analyzer	RAM	Trusted Advisor

* Required

Cancel

Next: Permissions

Don't attach any policy or tags:

Create role

1234

Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Showing 647 results

Policy name	Used as
AccessAnalyzerServicePolicy	None
AdministrativeAccess	Permissions policy (1)
AmazonBusinessDeviceSetup	None
AmazonBusinessDeviceAccess	None
AmazonBusinessGatewayConnection	None
AmazonBusinessNetworkProfileServicePolicy	None
AmazonBusinessProductCatalogAccessPolicy	None
AmazonBusinessProductOnlyAccess	None

Set permissions boundary

* Required

Cancel

Previous

Next: Tags

Create role

1234

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text"/>	<input type="text"/>	

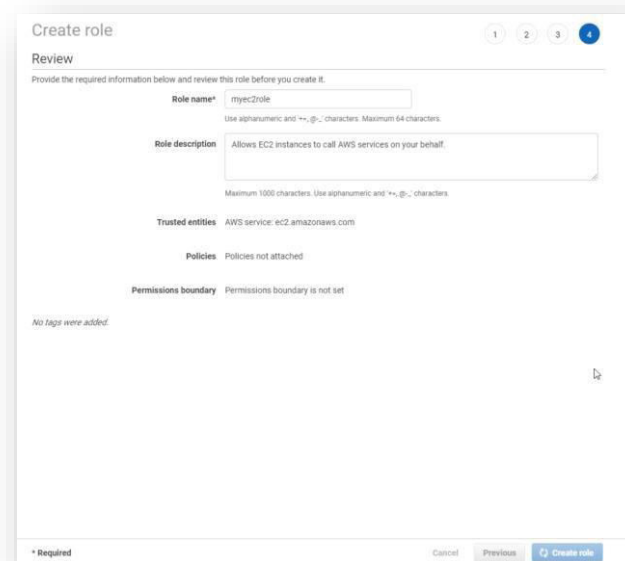
You can add 50 more tags.

Cancel

Previous

Next: Summary

Give the Role a name:



The screenshot shows the 'Create role' wizard in the AWS IAM console, specifically the 'Review' step. The role name is 'myec2role'. The role description is 'Allows EC2 instances to call AWS services on your behalf.' The trusted entities are 'AWS service: ec2.amazonaws.com'. The policies are 'Policies not attached'. The permissions boundary is 'Permissions boundary is not set'. There are no tags added. The bottom of the screen shows a 'Create role' button.

Create role

Review

Provide the required information below and review this role before you create it.

Role name* myec2role
Use alphanumeric and +, -, @, _ characters. Maximum 64 characters.

Role description
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and +, -, @, _ characters.

Trusted entities
AWS service: ec2.amazonaws.com

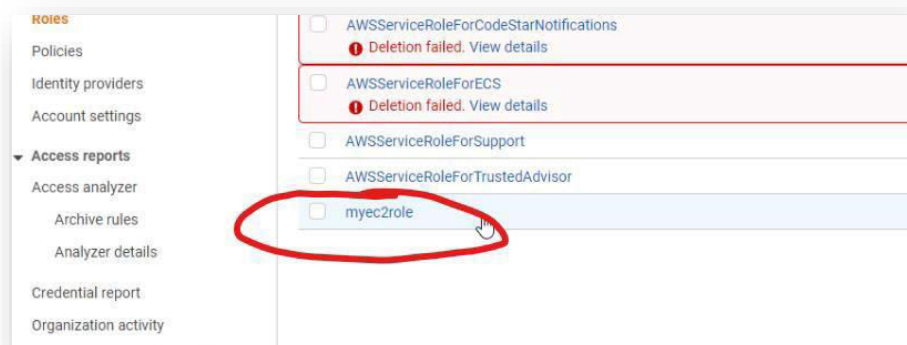
Policies
Policies not attached

Permissions boundary
Permissions boundary is not set

No tags were added.

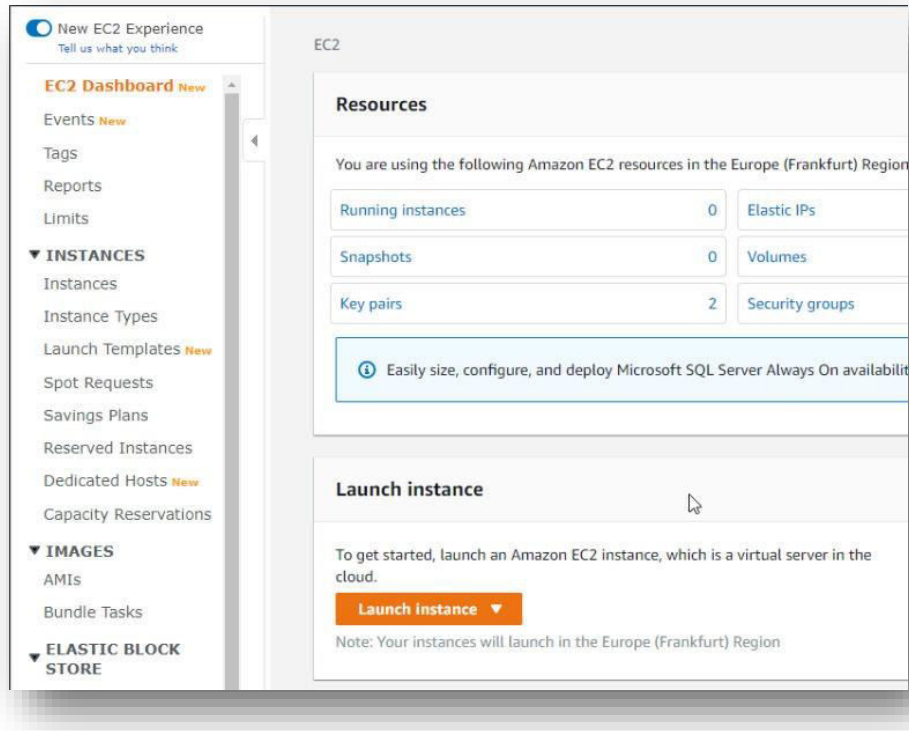
* Required Cancel Previous Create role

Open the new role:

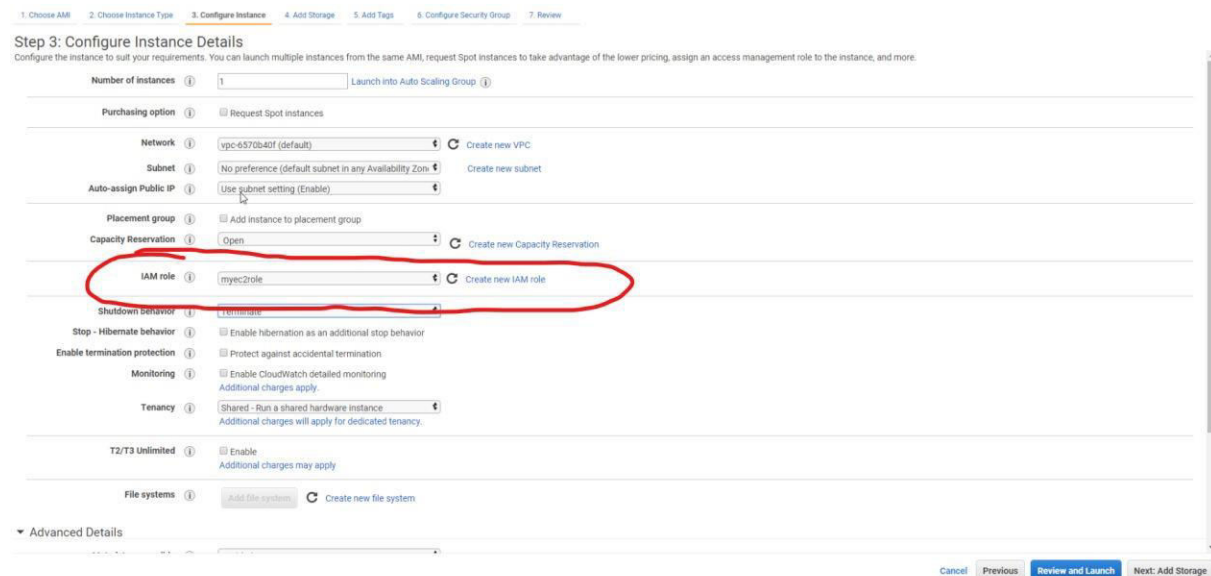


Launch an EC2 Instance

Before we add a new permission (an inline policy), let's see if we can get access to the S3 Bucket like this. Head over to the EC2 Dashboard and launch a new EC2 Instance.



Launch a new instance, choose Amazon Linux 2 AMI and t2.micro instance, but configure the instance to use the new role you created:



Launch the instance and then ssh into the instance:

```
Course 14 - Understanding Docker with AWS ECS and Fargate> ssh -i "my-keypair.pem" ec2-user@ec2-3-126-208-162.eu-central-1.compute.amazonaws.com
The authenticity of host 'ec2-3-126-208-162.eu-central-1.compute.amazonaws.com (3.126.208.162)' can't be established.
ECDSA key fingerprint is SHA256:A0gYP2X5glq0kFniIpbj8quUns051ZVK2tfg2v4r97I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-126-208-162.eu-central-1.compute.amazonaws.com,3.126.208.162' (ECDSA) to the list of known hosts.

 _ | _ | _ )
 _ | _ | _ ) Amazon Linux 2 AMI
 _ | _ | _ |

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-41-159 ~]$ aws
usage: aws [options] <command> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: too few arguments
[ec2-user@ip-172-31-41-159 ~]$
```

Add an Inline Policy

Now try to run the following command:

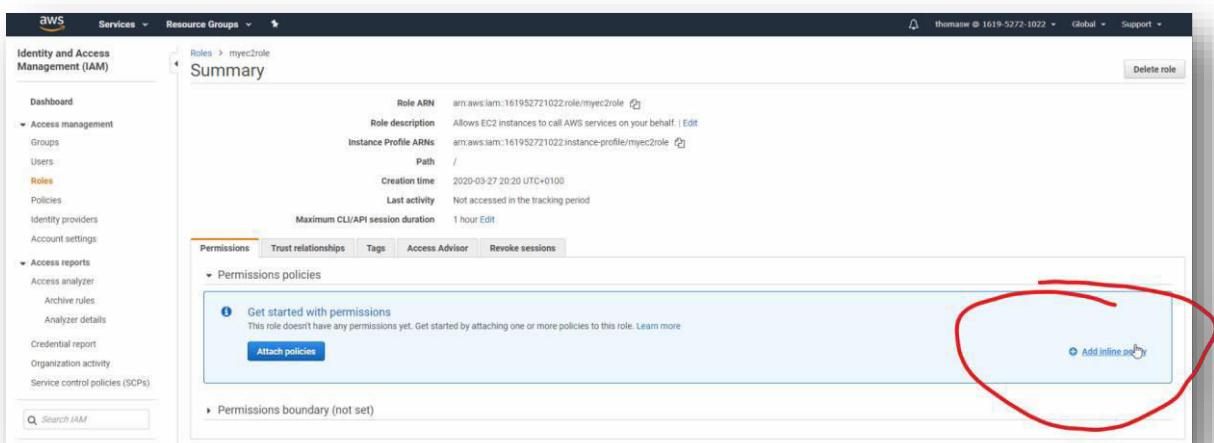
```
aws s3 ls
```

It will fail, because it doesn't have the right permissions:

```
aws: error: too few arguments
[ec2-user@ip-172-31-41-159 ~]$ aws s3 ls

An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
[ec2-user@ip-172-31-41-159 ~]$
```

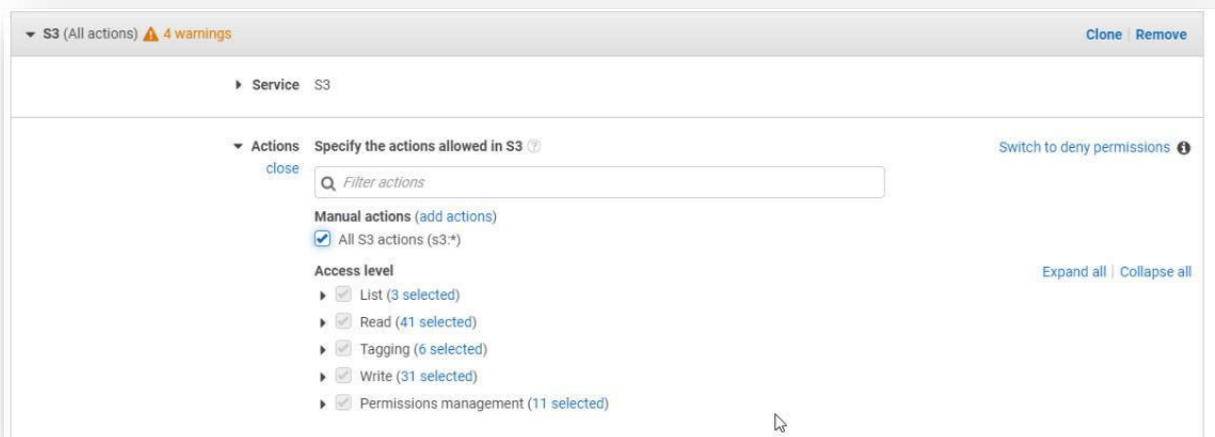
Go back to the IAM Dashboard -> roles and open the role your created. Let's add an inline policy:



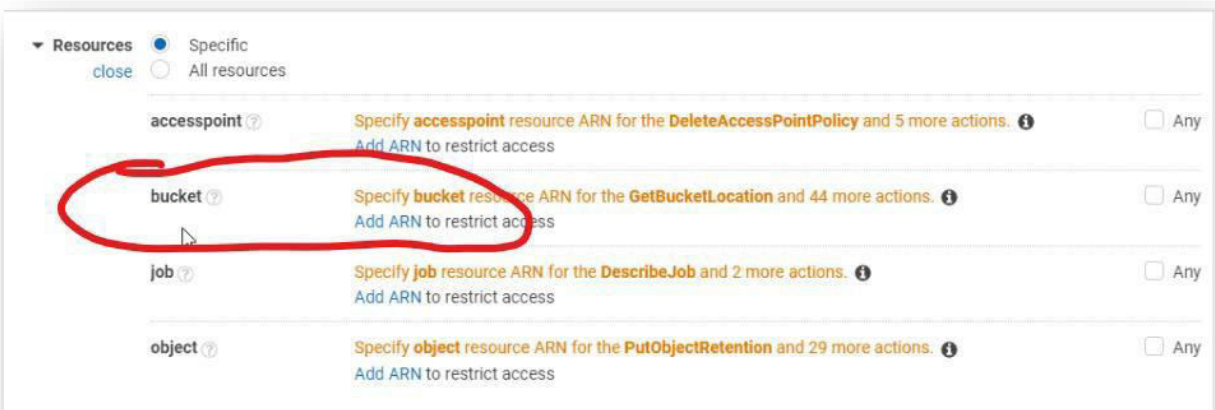
Service choose "S3":



Choose all S3 actions:



For Bucket name choose the bucket name you created earlier:



Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#)

Specify ARN for bucket [List ARNs manually](#)

arn:aws:s3::thomasw-123

Bucket name * ☐ Any

Cancel **Add**

For Object name choose “any” in the bucket you created earlier:

Add ARN(s)

Amazon Resource Names (ARNs) uniquely identify AWS resources. Resources are unique to each service. [Learn more](#)

Specify ARN for object [List ARNs manually](#)

arn:aws:s3::thomasw-123/*

Bucket name * ☐ Any

Object name * ☒ Any

Cancel **Add**

object

Review your policy:

S3 (All actions) 2 warnings [Clone](#) [Remove](#)

Service S3

Actions Manual actions

Resources ☒ Specific ☐ All resources

accesspoint Specify **accesspoint** resource ARN for the **DeleteAccessPointPolicy** and 5 more actions. Add ARN to restrict access ☐ Any

bucket arn:aws:s3::thomasw-123 [EDIT](#) ☐ Any

job Specify **job** resource ARN for the **DescribeJob** and 2 more actions. Add ARN to restrict access ☐ Any

object arn:aws:s3::thomasw-123/* [EDIT](#) ☐ Any

Request conditions [Specify request conditions \(optional\)](#)

Character count: 406 of 10240.
The current character count includes character for all inline policies in the role. [myec2role](#)

Cancel **Review policy**

Give your policy a name:

Create policy

1 2

Review policy

Before you create this policy, provide the required information and review this policy.

Name*

Maximum 128 characters. Use alphanumeric and '+, @, _' characters.

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining**. [Learn more](#)

Q Filter

Service	Access level	Resource	Request condition
Allow (1 of 224 services) Show remaining 223			
S3	Full: List, Tagging Limited: Read, Write, Permissions management	Multiple	None

* Required

Cancel Previous **Create policy**

S3 CLI with inline Policy

When you try to list the buckets now from your EC2 Instance then it should work:

```
ec2-user@ip-172-31-41-159:~
[ec2-user@ip-172-31-41-159 ~]$ aws s3 ls
2020-03-27 13:31:36 thomasm-123
[ec2-user@ip-172-31-41-159 ~]$
```

You can also list the contents of the bucket:

```
[ec2-user@ip-172-31-41-159 ~]$ aws s3 ls s3://thomasm-123
2020-03-27 15:31:35 105 index.html
[ec2-user@ip-172-31-41-159 ~]$
```

```
aws s3 ls s3://bucket-name
```

or download files:

```
[ec2-user@ip-172-31-41-159 ~]$ aws s3 cp s3://thomasw-123/index.html .
download: s3://thomasw-123/index.html to ./index.html
[ec2-user@ip-172-31-41-159 ~]$
```

```
aws s3 cp s3://bucket-name/index.html .
```

Then edit the file with nano and write some text between the body-tags:

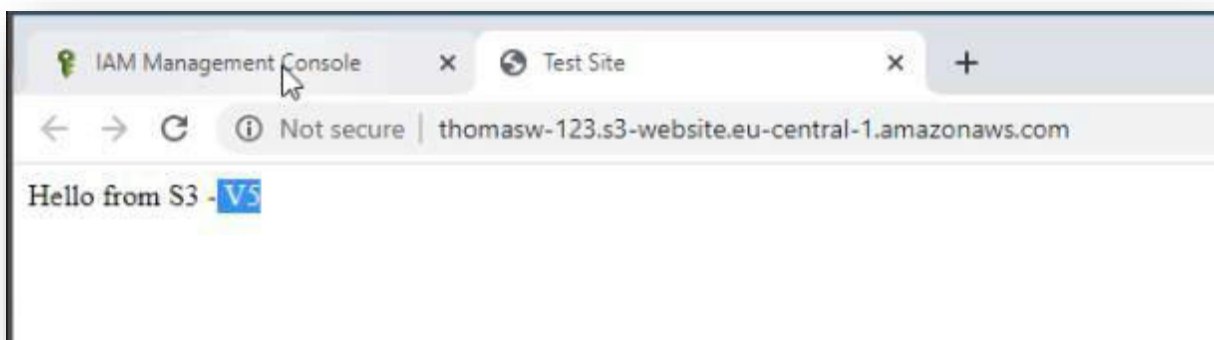
```
nano index.html
```

Note: Exit with Ctrl+x, you will be asked to save, hit return to save.

And re-upload the file:

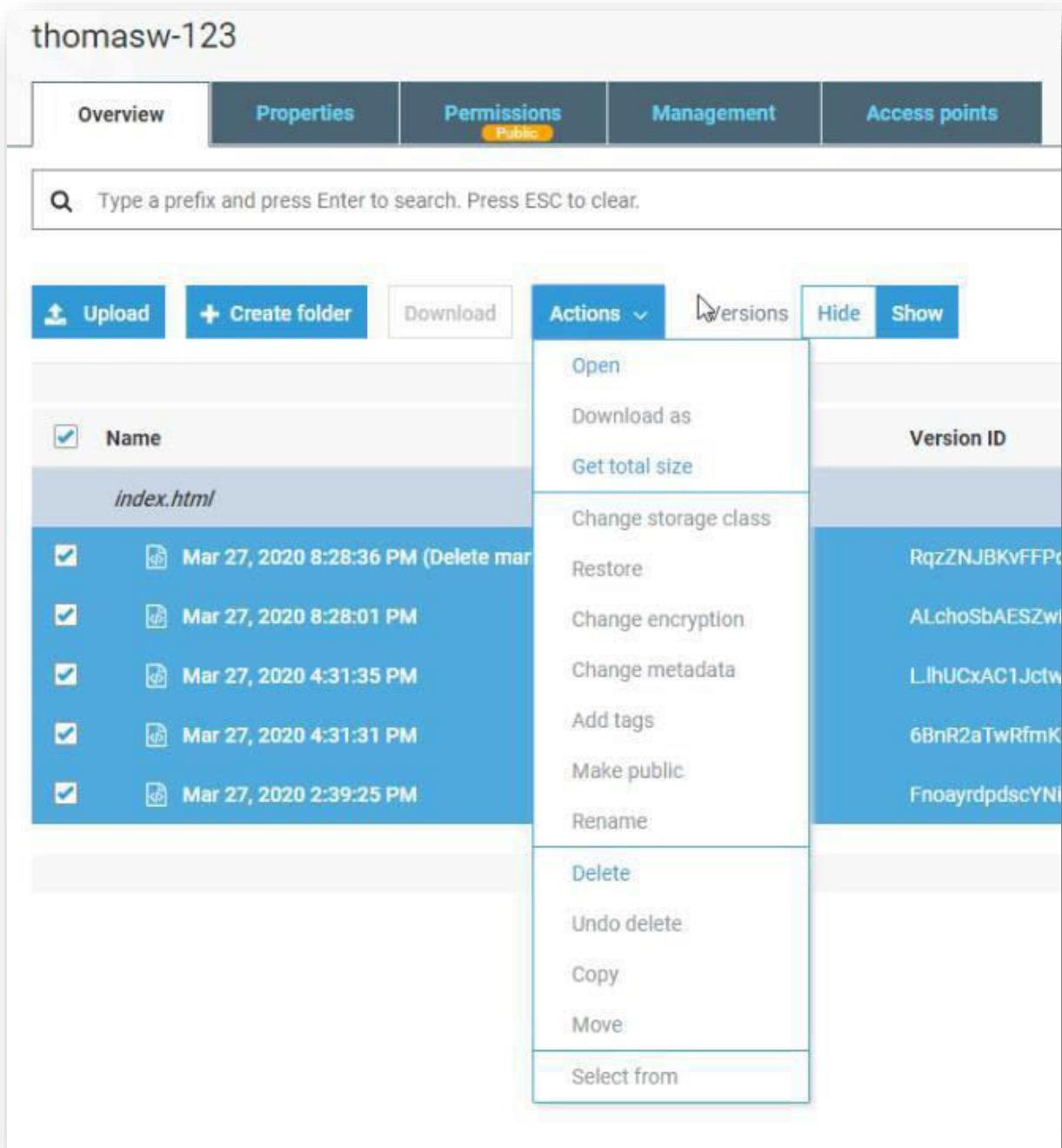
```
[ec2-user@ip-172-31-41-159 ~]$ cat index.html
<!DOCTYPE html>
<head>
  <title>Test Site</title>
</head>
<body>
  Hello from S3 - V4
</body>
[ec2-user@ip-172-31-41-159 ~]$ nano index.html
[ec2-user@ip-172-31-41-159 ~]$ cat index.html
<!DOCTYPE html>
<head>
  <title>Test Site</title>
</head>
<body>
  Hello from S3 - V5
</body>
[ec2-user@ip-172-31-41-159 ~]$ aws s3 cp index.html s3://thomasw-123/index.html
upload: ./index.html to s3://thomasw-123/index.html
[ec2-user@ip-172-31-41-159 ~]$
```

If you reload the website it should be updated immediately:



Cleanup

Delete all the versions in your S3 Bucket, otherwise you can't delete the bucket:



Then type in

```
aws s3 rb s3://bucket-name
```

```
delete you tried to delete is not empty. You must delete all  
[ec2-user@ip-172-31-41-159 ~]$ aws s3 rb s3://thomasw-123  
remove_bucket: thomasw-123  
[ec2-user@ip-172-31-41-159 ~]$
```

And terminate your EC2 instance, as well as remove the role you created.

Lab End
