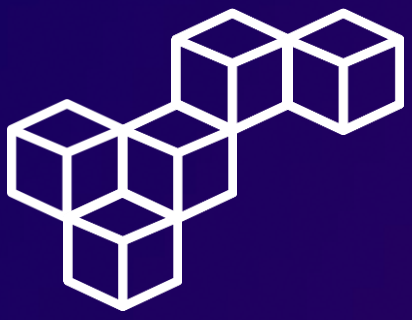
AWS re:Start LAB

Creating a Website on Amazon S3

WEEK 8

aws re/start



Overview

Creating a website on Amazon S3 is a straightforward and cost-effective way to host static web content. You'll start by setting up an Amazon S3 bucket to store all your web files, such as HTML, CSS, and JavaScript, making them accessible online. Configuring permissions is crucial; creating an IAM user with the necessary access to manage your S3 resources securely ensures that only authorized users can modify your website content.

After setting up the bucket and permissions, you can upload your website files to Amazon S3, making your site live quickly. To streamline updates, you can create a batch file that automates uploading changes to your S3 bucket whenever you modify local website files. This ensures your site remains current with minimal effort. By leveraging these AWS services, you can efficiently host and maintain a static website with robust security and scalability.

Topics covered

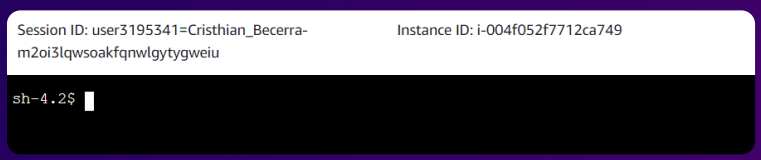
- Create an Amazon Simple Storage Service (Amazon S3) bucket.
- Create a new AWS Identity and Access Management (IAM) user that has full access to the Amazon S3 service.
- Upload files to Amazon S3 to host a simple website
- Create a batch file that can be used to update the static website when you change any of the website files locally.

Task 1

Connect to an Amazon Linux EC2 instance using SSM

Connect to your EC2 instance

Connect to your Amazon EC2 Instance using AWS Systems Manager Session Manager.



Change the user and home directory

Run the following commands to change the user and home directory.

```
sh-4.2$ sudo su -l ec2-user
[ec2-user@ip-10-200-0-19 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-200-0-19 ~]$
```

aws re/start

Task 2

Configure the AWS CLI

The configure command

In the session terminal window, run the aws configure command to update the AWS CLI software with credentials.

```
[ec2-user@ip-10-200-0-19 ~]$ aws configure
```

Set configuration variables

At the prompt, configure the following:

- AWS Access Key ID
- AWS Secret Access Key
- Default region name
- Default ouput format

```
[ec2-user@ip-10-200-0-19 ~]$ aws configure
AWS Access Key ID [None]: AKIA6GBMHO6NIXPFTNGO
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-200-0-19 ~]$
```

Task 3

Create an S3 bucket using the AWS CLI

Create a new S3 bucket

To create a bucket in Amazon S3, you use the aws s3api create-bucket command.

```
[ec2-user@ip-10-200-0-19 ~]$ aws s3api create-bucket --bucket twhitlock128 --region us-west-2
 --create-bucket-configuration LocationConstraint=us-west-2
```

Review command output

If the command is successful, you will get a JSON-formatted response with a **Location** name-value pair, where the value reflects the bucket name.

```
[ec2-user@ip-10-200-0-19 ~]$ aws s3api create-bucket --bucket twhitlock128 --region us-west-2
 --create-bucket-configuration LocationConstraint=us-west-2
{
    "Location": "http://twhitlock128.s3.amazonaws.com/"
}
[ec2-user@ip-10-200-0-19 ~]$
```

aws re/start

Task 4

Create a new IAM user that has full access to Amazon S3

Step 1: Create a new IAM user

Using the AWS CLI, create a new IAM user with the command aws iam create-user and username **awsS3user**.

```
[ec2-user@ip-10-200-0-19 ~]$ aws iam create-user --user-name awsS3user
{
    "User": {
        "UserName": "awsS3user",
        "Path": "/",
        "CreateDate": "2024-05-18T22:18:42Z",
        "UserId": "AIDA6GBMHO6NG7HY256N3",
        "Arn": "arn:aws:iam::975050340250:user/awsS3user"
    }
}
[ec2-user@ip-10-200-0-19 ~]$
```

Step 2: Create a login profile

Create a login profile for the new user by running the following aws iam create-login-profile command.

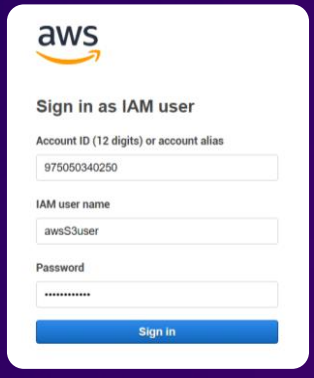
```
[ec2-user@ip-10-200-0-19 ~]$ aws iam create-login-profile --user-name awsS3user --password Tr
aining123!
{
    "LoginProfile": {
        "UserName": "awsS3user",
        "CreateDate": "2024-05-18T22:19:12Z",
        "PasswordResetRequired": false
    }
}
```

Task 4

Create a new IAM user that has full access to Amazon S3

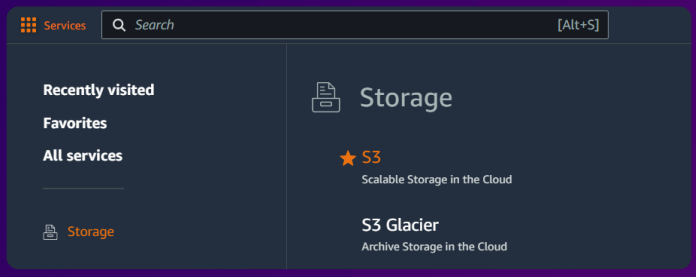
Step 3: Log in as the new IAM user

Log in to the AWS Management Console as the new **awsS3user** user.



Step 4: Access the S3 service

In the AWS Management Console, select the S3 service.



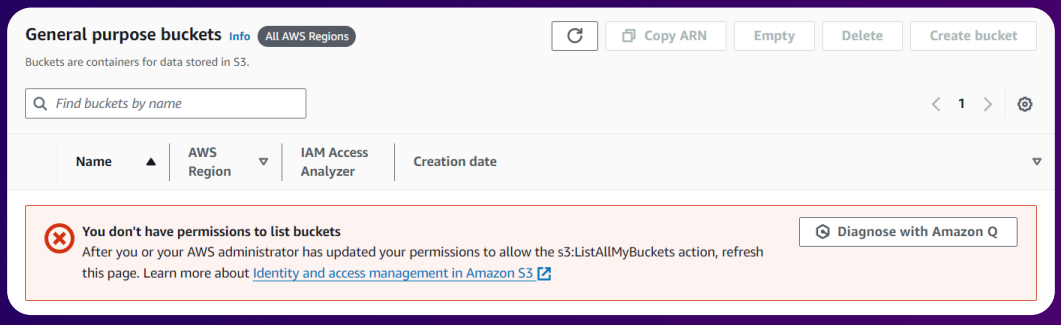
aws re/start

Task 4

Create a new IAM user that has full access to Amazon S3

Step 5: Review new bucket

Navigate to the **Buckets** section. The **awsS3user** user does not have Amazon S3 access to the bucket that you created.



Step 6: Review AWS managed policies

To find the AWS managed policy that grants full access to Amazon S3, run the following aws iam list-policies command.

```
[ec2-user@ip-10-200-0-19 ~]$ aws iam list-policies --query "Policies[?contains(PolicyName,'S3')]"
[
    {
        "PolicyName": "AmazonS3FullAccess",
        "PermissionsBoundaryUsageCount": 0,
        "CreateDate": "2015-02-06T18:40:58Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ANPAIFIR6V6BVTRAHWINE",
        "DefaultVersionId": "v2",
        "Path": "/",
        "Arn": "arn:aws:iam::aws:policy/AmazonS3FullAccess",
        "UpdateDate": "2021-09-27T20:16:37Z"
    },
    {
        "PolicyName": "AmazonS3ReadOnlyAccess",
```

aws re/start

Task 4

Create a new IAM user that has full access to Amazon S3

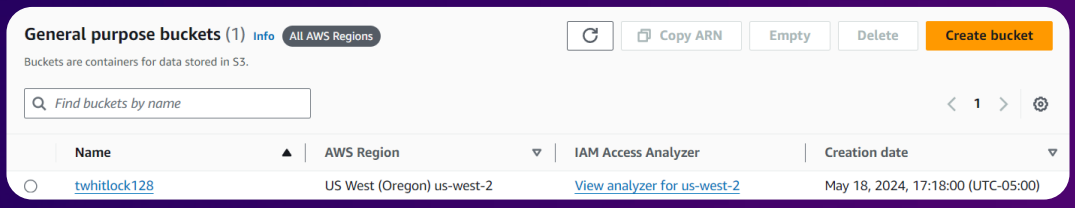
Step 7: Grant full access to S3

Run the following aws iam attach-user-policy command to grant the **awsS3user** user full access to Amazon S3.

```
[ec2-user@ip-10-200-0-19 ~]$ aws iam attach-user-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3FullAccess --user-name awsS3user
[ec2-user@ip-10-200-0-19 ~]$
```

Step 8: Access the bucket

The new IAM user **awsS3user** now has full access to all Amazon S3 buckets. Select your bucket name.



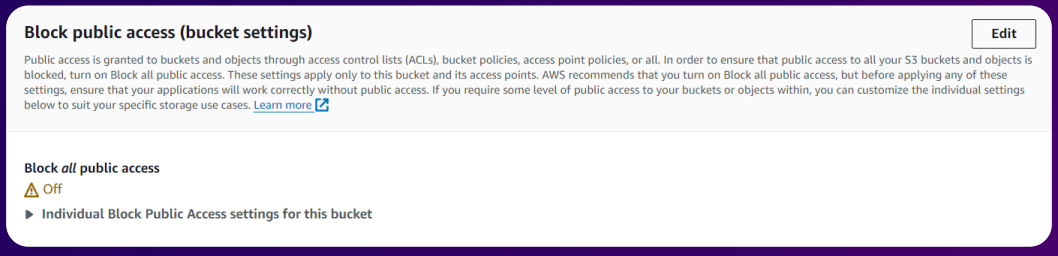
aws re/start

Task 5

Adjust S3 bucket permissions

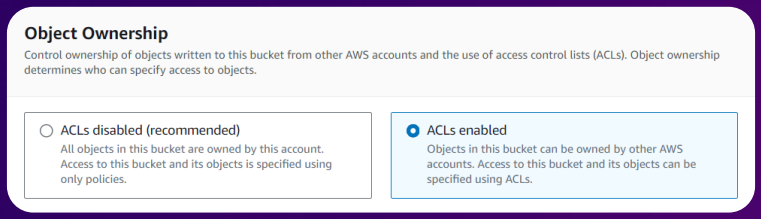
Edit Block public access

Go to permissions, under **Block public access (bucket settings)**, choose **Edit,** and UnSelect Block all public access.



Edit Object Ownership

Under **Object Ownership**, choose **Edit**, and choose ACLs enabled, and choose I acknowledge that ACLs will be restored.



aws re/start

Task 6

Extract the files that you need for this lab

Extract the website files

In the Session terminal, extract the files that you need for this lab by running the following commands.

```
[ec2-user@ip-10-200-0-19 ~]$ cd ~/sysops-activity-files
[ec2-user@ip-10-200-0-19 sysops-activity-files]$ tar xvzf static-website-v2.tar.gz
static-website/
static-website/css/
static-website/css/styles.css
static-website/images/
static-website/images/Cafe-Owners.png
static-website/images/Cake-Vitrine.png
static-website/images/Coffee-and-Pastries.png
static-website/images/Coffee-Shop.png
static-website/images/Cookies.png
static-website/images/Cup-of-Hot-Chocolate.png
static-website/images/Strawberry-&-Blueberry-Tarts.png
static-website/images/Strawberry-Tarts.png
static-website/index.html
[ec2-user@ip-10-200-0-19 sysops-activity-files]$
```

Review the extracted files

To confirm that the files were extracted correctly, run the following commands.

```
[ec2-user@ip-10-200-0-19 sysops-activity-files]$ cd static-website
[ec2-user@ip-10-200-0-19 static-website]$ ls
css  images  index.html
[ec2-user@ip-10-200-0-19 static-website]$
```

aws re/start

Task 7

Upload files to Amazon S3 by using the AWS CLI

Step 1: Enable Static website hosting

To ensure that the bucket can function as a website run the following aws s3 website command. Also, to upload the files to the bucket, run the following aws s3 cp command.

```
[ec2-user@ip-10-200-0-19 static-website]$ aws s3 website s3://twhitlock128/ --index-document
index.html
[ec2-user@ip-10-200-0-19 static-website]$ aws s3 cp /home/ec2-user/sysops-activity-files/stat
ic-website/ s3://twhitlock128/ --recursive --acl public-read
upload: css/styles.css to s3://twhitlock128/css/styles.css
upload: images/Coffee-Shop.png to s3://twhitlock128/images/Coffee-Shop.png
upload: ./index.html to s3://twhitlock128/index.html
upload: images/Cookies.png to s3://twhitlock128/images/Cookies.png
upload: images/Coffee-and-Pastries.png to s3://twhitlock128/images/Coffee-and-Pastries.png
upload: images/Cafe-Owners.png to s3://twhitlock128/images/Cafe-Owners.png
upload: images/Cake-Vitrine.png to s3://twhitlock128/images/Cake-Vitrine.png
upload: images/Cup-of-Hot-Chocolate.png to s3://twhitlock128/images/Cup-of-Hot-Chocolate.png
upload: images/Strawberry-&-Blueberry-Tarts.png to s3://twhitlock128/images/Strawberry-&-Blue
berry-Tarts.png
upload: images/Strawberry-Tarts.png to s3://twhitlock128/images/Strawberry-Tarts.png
[ec2-user@ip-10-200-0-19 static-website]$
```

Step 2: Review uploaded files

To verify that the files were uploaded, run the following aws s3 ls command.

```
[ec2-user@ip-10-200-0-19 static-website]$ aws s3 ls twhitlock128
                           PRE css/
                           PRE images/
2024-05-18 22:36:45       2980 index.html
[ec2-user@ip-10-200-0-19 static-website]$
```

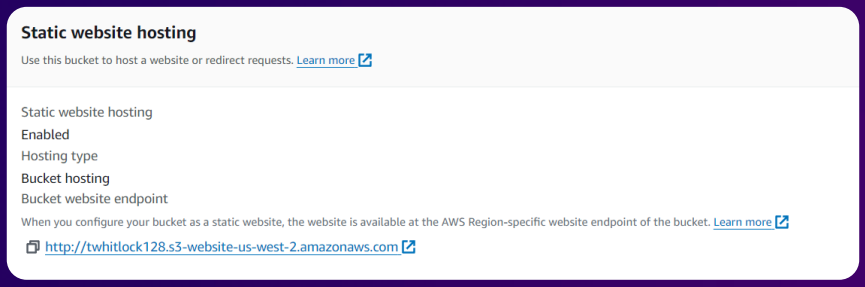
aws re/start

Task 7

Upload files to Amazon S3 by using the AWS CLI

Step 3: Review Static website hosting

To open the website URL, select the **Bucket website endpoint**.



Step 4: Review the website

The static website is now available to the public for viewing.

aws re/start

Task 8

Create a batch file to make updating the website repeatable

Step 1: Create a batch file

To create a repeatable deployment, use the vi editor to create a batch file, add the following aws s3 cp command, and make the batch file executable.

```
[ec2-user@ip-10-200-0-19 static-website]$ cd ~
[ec2-user@ip-10-200-0-19 ~]$ vi update-website.sh
[ec2-user@ip-10-200-0-19 ~]$
```

```
#!/bin/bash

aws s3 cp /home/ec2-user/sysops-activity-files/static-website/ s3://twhitlock128/ --recursive --acl public-read
~
```

```
[ec2-user@ip-10-200-0-19 ~]$ chmod +x update-website.sh
[ec2-user@ip-10-200-0-19 ~]$
```

Step 2: Edit a local file

Open the local copy of the **index.html** file in a text editor, and edit some HTML color codes.

```
[ec2-user@ip-10-200-0-19 ~]$ vi sysops-activity-files/static-website/index.html
[ec2-user@ip-10-200-0-19 ~]$
```

aws re/start

Task 8

Create a batch file to make updating the website repeatable

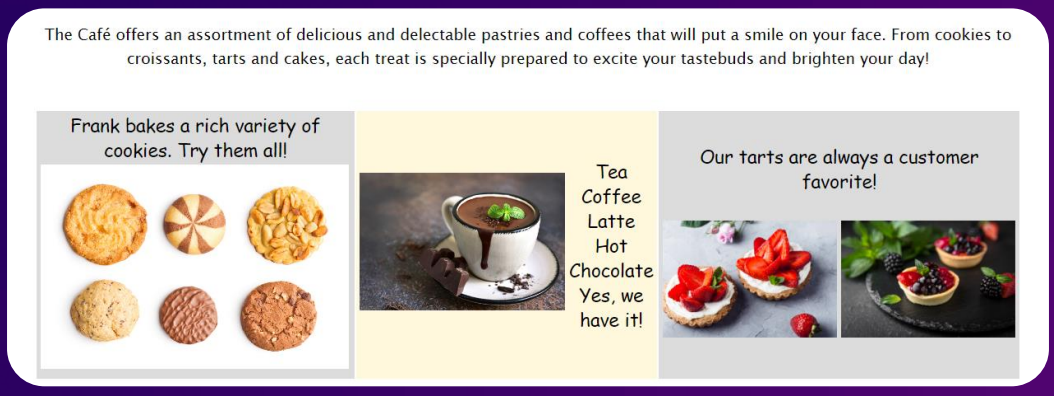
Step 3: Update the website

To update the website, run the batch file.

```
[ec2-user@ip-10-200-0-19 ~]$ ./update-website.sh
upload: sysops-activity-files/static-website/css/styles.css to s3://twhitlock128/css/styles.css
upload: sysops-activity-files/static-website/images/Coffee-Shop.png to s3://twhitlock128/images/Coffee-Shop.png
upload: sysops-activity-files/static-website/index.html to s3://twhitlock128/index.html
upload: sysops-activity-files/static-website/images/Cake-Vitrine.png to s3://twhitlock128/images/Cake-Vitrine.png
upload: sysops-activity-files/static-website/images/Cafe-Owners.png to s3://twhitlock128/images/Cafe-Owners.png
upload: sysops-activity-files/static-website/images/Strawberry-&-Blueberry-Tarts.png to s3://twhitlock128/images/Strawberry-&-Blueberry-Tarts.png
upload: sysops-activity-files/static-website/images/Cookies.png to s3://twhitlock128/images/Cookies.png
upload: sysops-activity-files/static-website/images/Cup-of-Hot-Chocolate.png to s3://twhitlock128/images/Cup-of-Hot-Chocolate.png
upload: sysops-activity-files/static-website/images/Strawberry-Tarts.png to s3://twhitlock128/images/Strawberry-Tarts.png
upload: sysops-activity-files/static-website/images/Coffee-and-Pastries.png to s3://twhitlock128/images/Coffee-and-Pastries.png
[ec2-user@ip-10-200-0-19 ~]$
```

Step 4: Review the update

Review the changes in the website page.



aws re/start

Task 9

Optional challenge

The aws s3 sync command

Notice that the batch file uploads every file to Amazon S3 every time you run it even when most of the files have no changes to them. To help make the script more efficient, replace the aws s3 cp command that you've been using with the following aws s3 sync command.

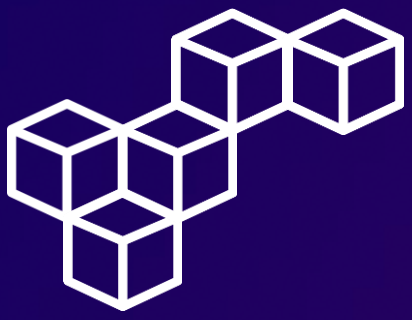
```
#!/bin/bash

aws s3 sync /home/ec2-user/sysops-activity-files/static-website/ s3://twhitlock128/ --acl public-read
~
~
```

Run the batch file again

Make some minor changes to the local copy of the **index.html** file. Then, to update the website, run the batch file again. Notice that this time, only the modified file is uploaded.

```
[ec2-user@ip-10-200-0-19 ~]$ vi sysops-activity-files/static-website/index.html
[ec2-user@ip-10-200-0-19 ~]$ ./update-website.sh
upload: sysops-activity-files/static-website/index.html to s3://twhitlock128/index.html
[ec2-user@ip-10-200-0-19 ~]$
```

aws re/start



Conclusions

The aws configure command
The aws configure command simplifies setting up AWS CLI credentials and default configurations for seamless interaction with AWS services.

The aws iam commands
The aws iam commands enable secure management of AWS IAM resources, allowing precise control over user permissions and access.

The aws s3api commands
The aws s3api commands provide detailed control over Amazon S3 operations, enabling advanced configuration and management of S3 buckets and objects.

The aws s3 commands
The aws s3 commands facilitate easy file operations like upload, download, and synchronization between local storage and S3 buckets.

Static website hosting
Static website hosting utilizes Amazon S3 to serve static websites, offering a scalable, reliable, and cost-effective solution for web hosting.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru