**Creating a Website on S3**

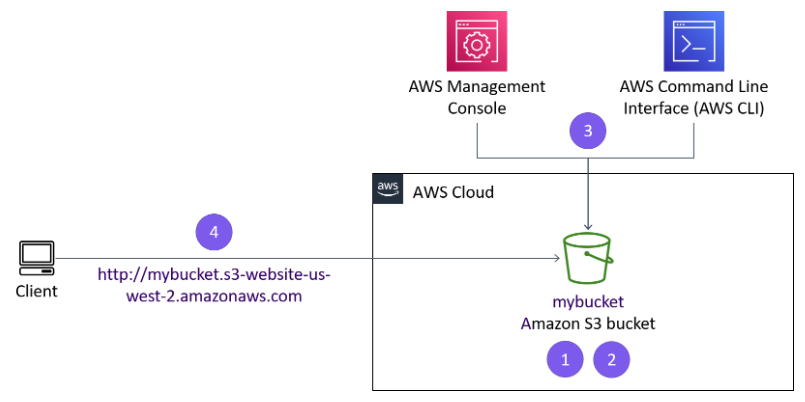
**We will use AWS Command Line Interface (AWS CLI) commands from an Amazon Elastic Compute Cloud (Amazon EC2) instance to:**

**• 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.**

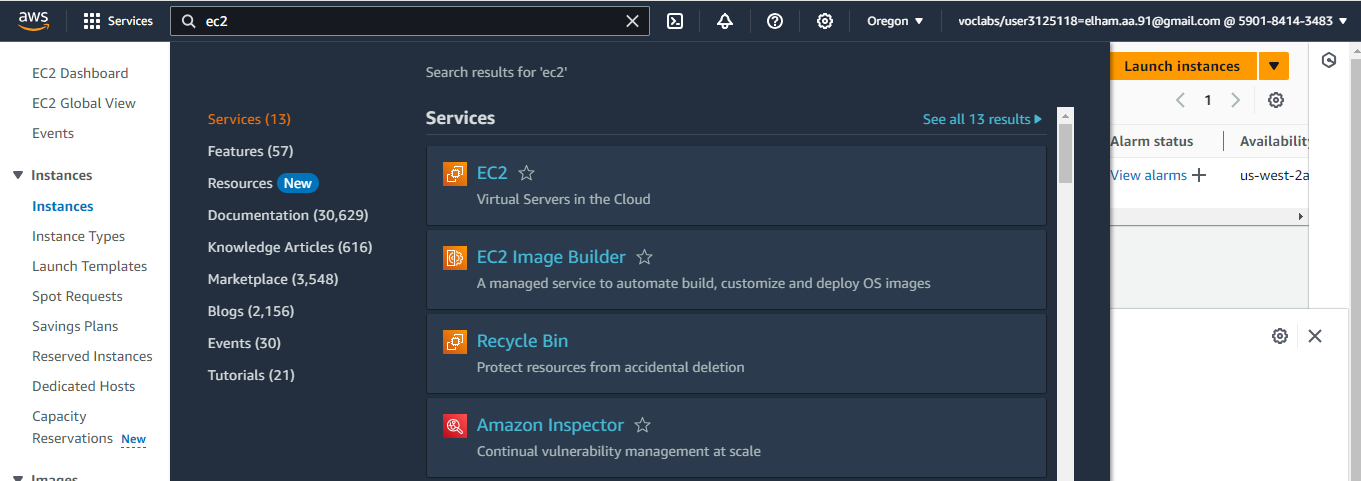
**• Upload files to Amazon S3 to host a simple website for the Café & Bakery.**

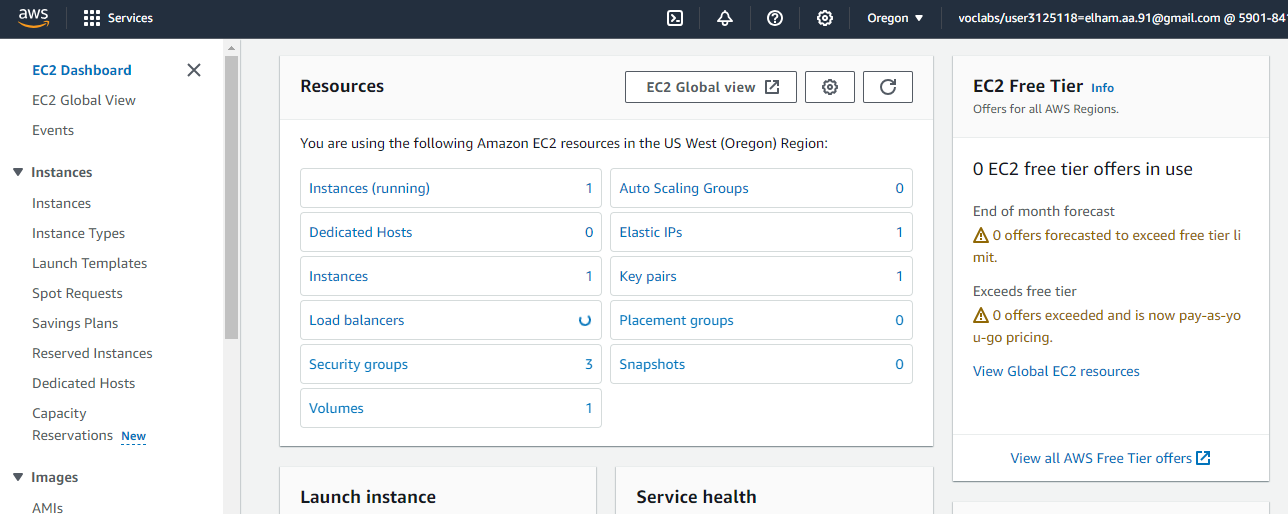
**• Create a batch file that can be used to update the static website when you change any of the website files locally.**

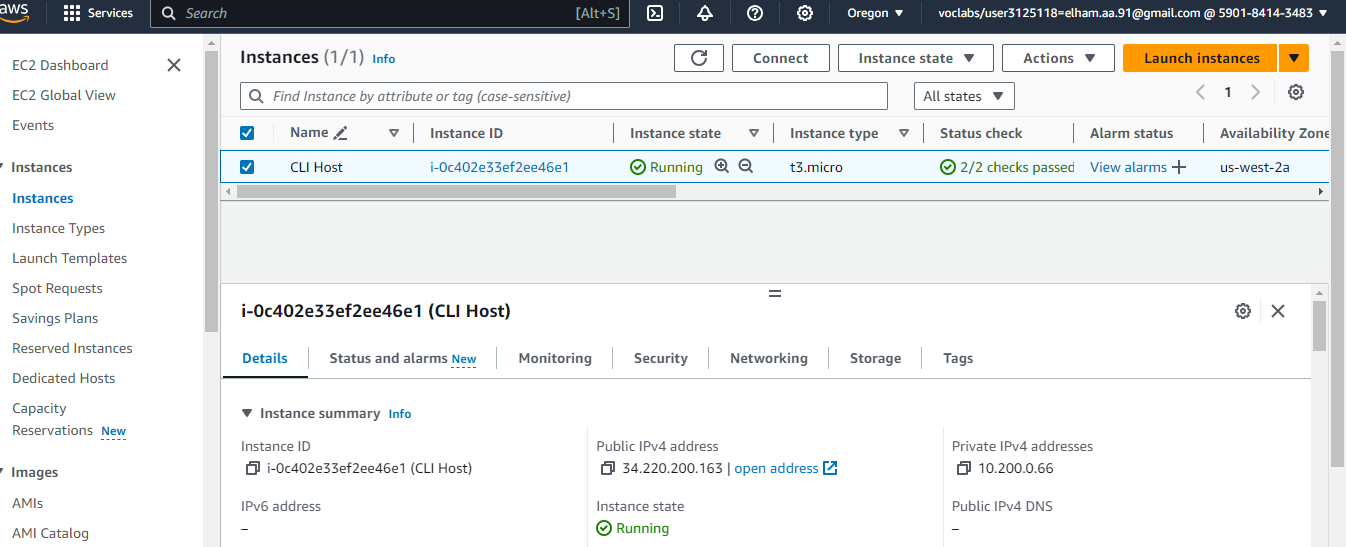
****

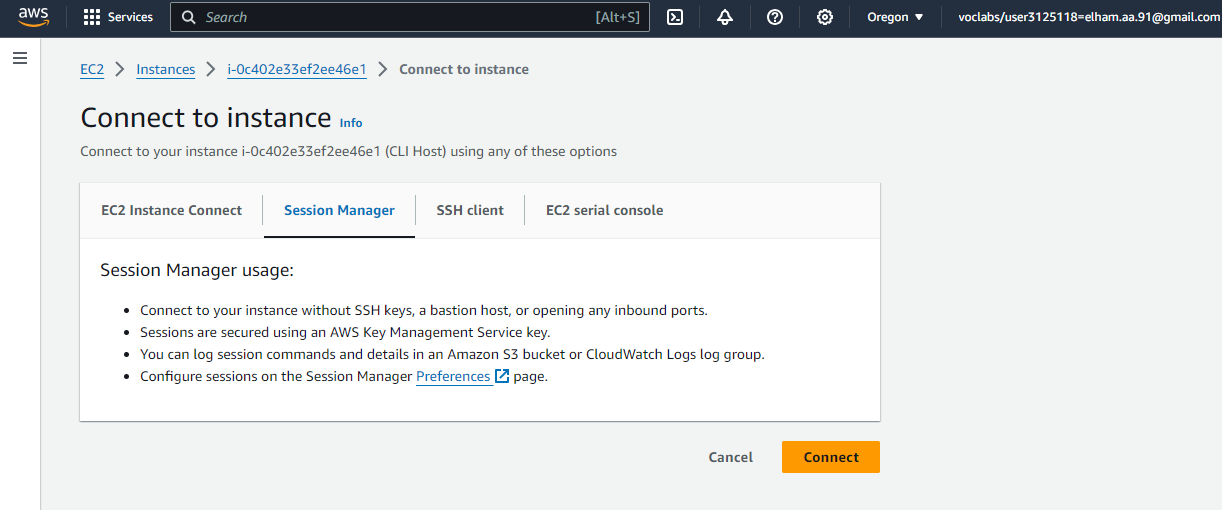
*1: Connecting to an Amazon Linux EC2 instance using SSM*

* connecting to the Amazon EC2 Instance using AWS Systems Manager Session Manager

**

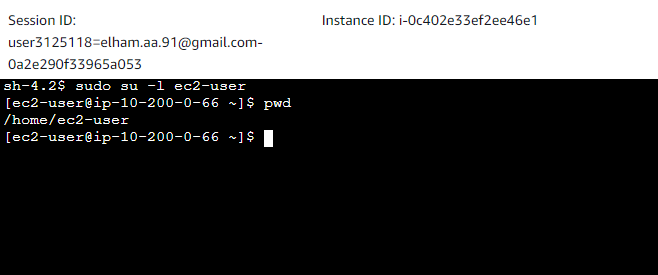


**

**

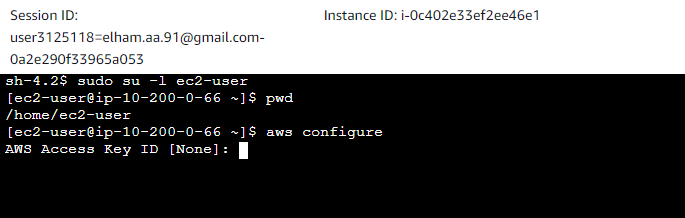
**

* changing the user and home directory:

**

*2: Configuring the AWS CLI*

* updating the AWS CLI software with credentials

**

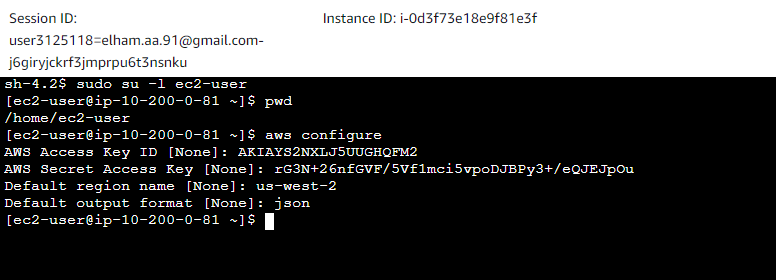
* configuring the following:

**AWS Access Key ID**

**AWS Secret Access Key**

**Default region name**

**Default output format**



*Task 3: Creating an S3 bucket using the AWS CLI*

When we create a new S3 bucket, the bucket must have a unique name, such as the combination of our first initial, last name, and three random numbers. By default, the S3 bucket is created in the us-east-1 Region.

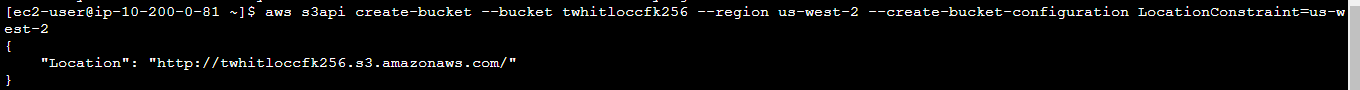
To create a bucket in Amazon S3, we use the aws s3api create-bucket command. When we use this command to create an S3 bucket, we also include the following:

* Specify --region us-west-2
* Add --create-bucket-configuration LocationConstraint=us-west-2 to the end of the command

The following is the command to create a new S3 bucket:

aws s3api create-bucket --bucket twhitloccfk256 --region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2

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

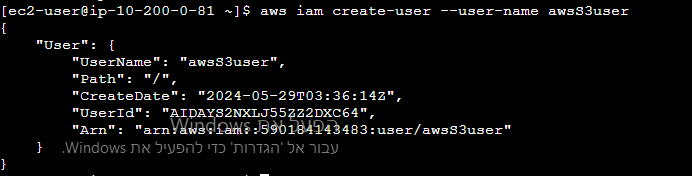


*Task 4: Creating a new IAM user that has full access to Amazon S3*

The AWS CLI command: aws iam create-user creates a new IAM user for our AWS account. The option --user-name is used to create the name of the user and must be unique within the account.

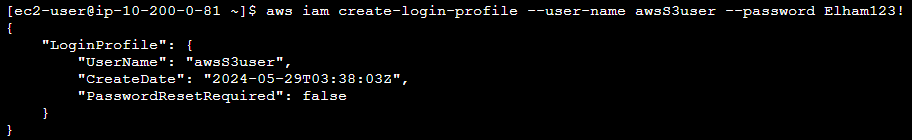
* Using the AWS CLI, we create a new IAM user with the command aws iam create-user and username awsS3user:

aws iam create-user --user-name awsS3user

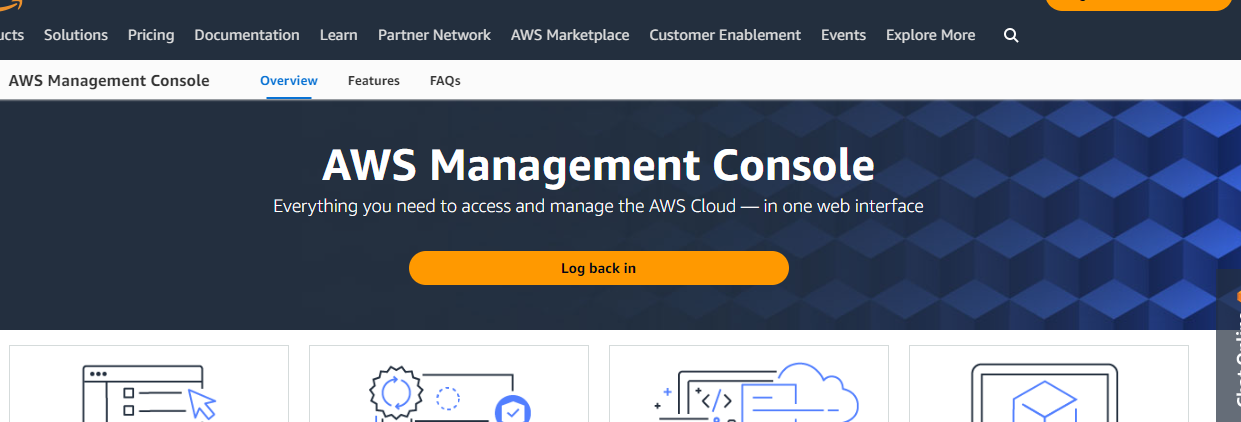


* Creating a login profile for the new user by using the following command:

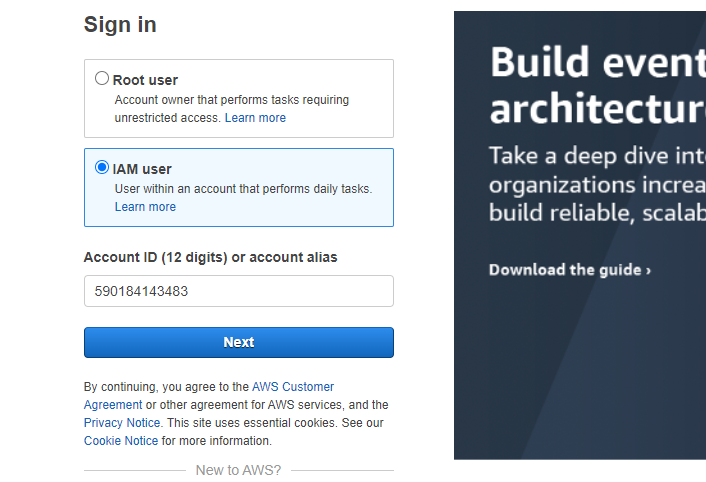
aws iam create-login-profile --user-name awsS3user --password Elham123!



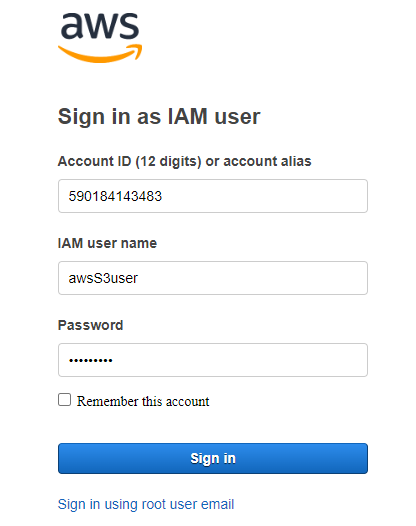
* Copying the AWS account number:
* We copy the 12 digit Account ID number: 590184143483
* In the current we drop down menu, choose Sign Out.
* Loging in to the AWS Management Console as the new awsS3user user:
* In the browser tab where wee just signed out of the AWS Management Console, we choose Log back in or Sign in to the Console.



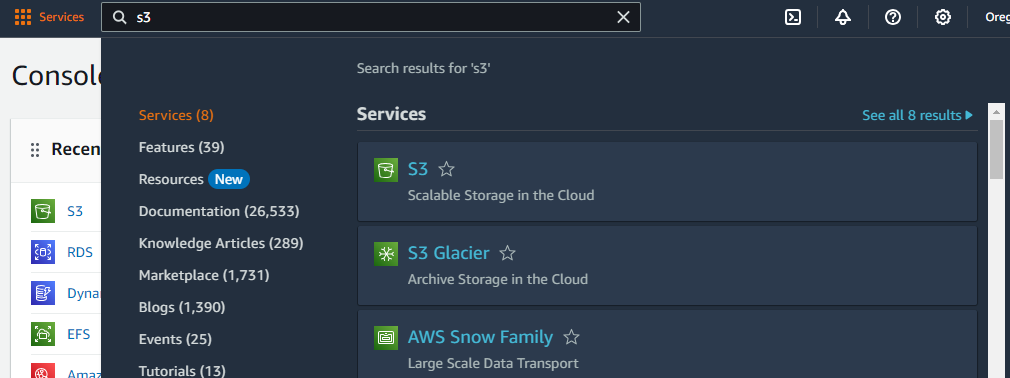
* In the sign-in screen, we choose the radio button IAM user.In the text field, we paste or enter the account ID with no dashes.Choose Next.



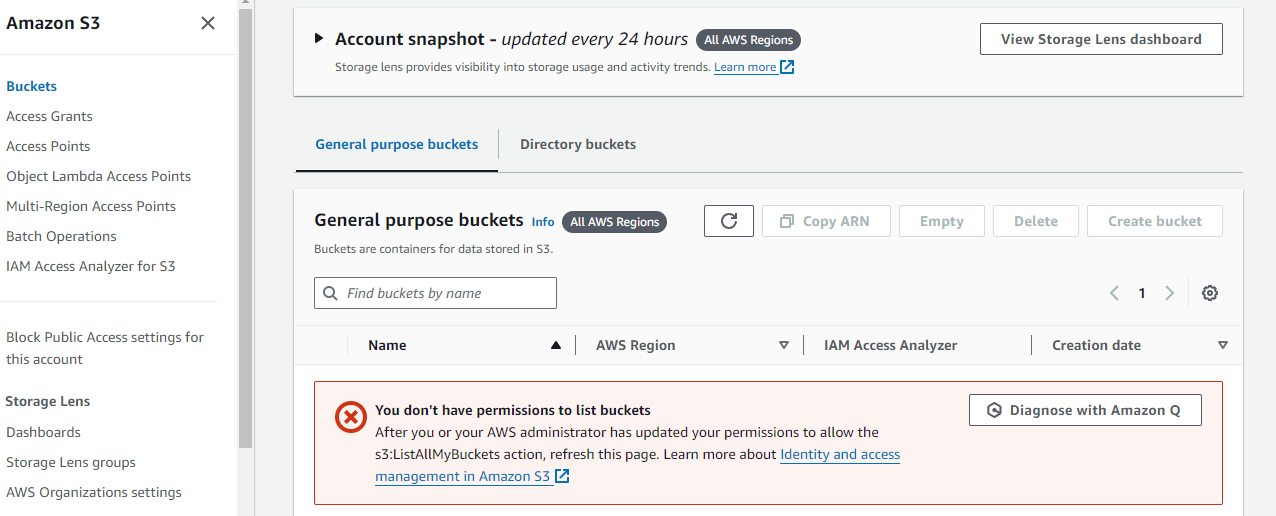
* A new login screen with Sign in as IAM user field will show. The account ID will be filled in from the previous screen.
  + - For IAM user name, we enter awsS3user
    - For Password, enter Elham123!
    - Choosing Sign In



* On the AWS Management Console, in the Search box, we enter S3 and choose S3. This option takes us to the Amazon S3 console page.



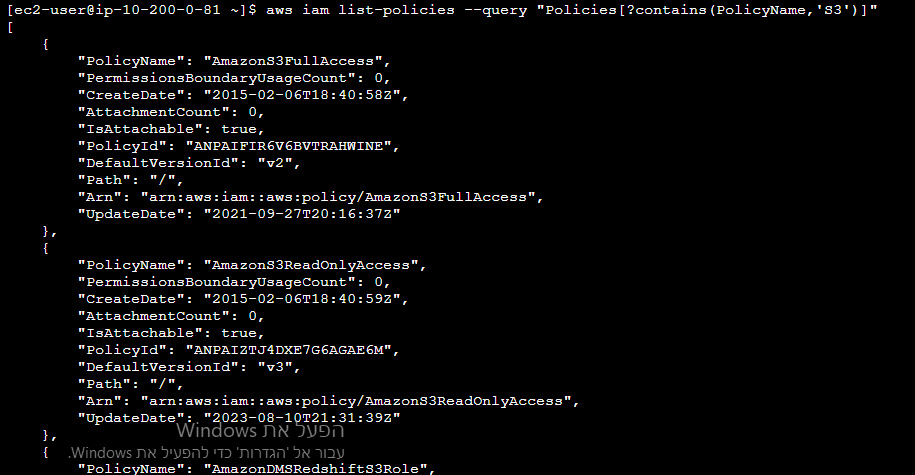
* Refreshing the Amazon S3 console page to see if it appears. The awsS3user user does not have Amazon S3 access to the bucket that we created, so we might see an error for Access to this bucket.



* In the terminal window, to find the AWS managed policy that grants full access to Amazon S3, we run the following command:

aws iam list-policies --query "Policies[?contains(PolicyName,'S3')]"

The result displays policies that have a PolicyName attribute containing the term S3. Locate the policy that grants full access to Amazon S3. we use this policy in the next step.



* To grant the awsS3user user full access to the S3 bucket, we replace <policyYouFound> in following command with the appropriate PolicyName from the results, and we run the adjusted command:

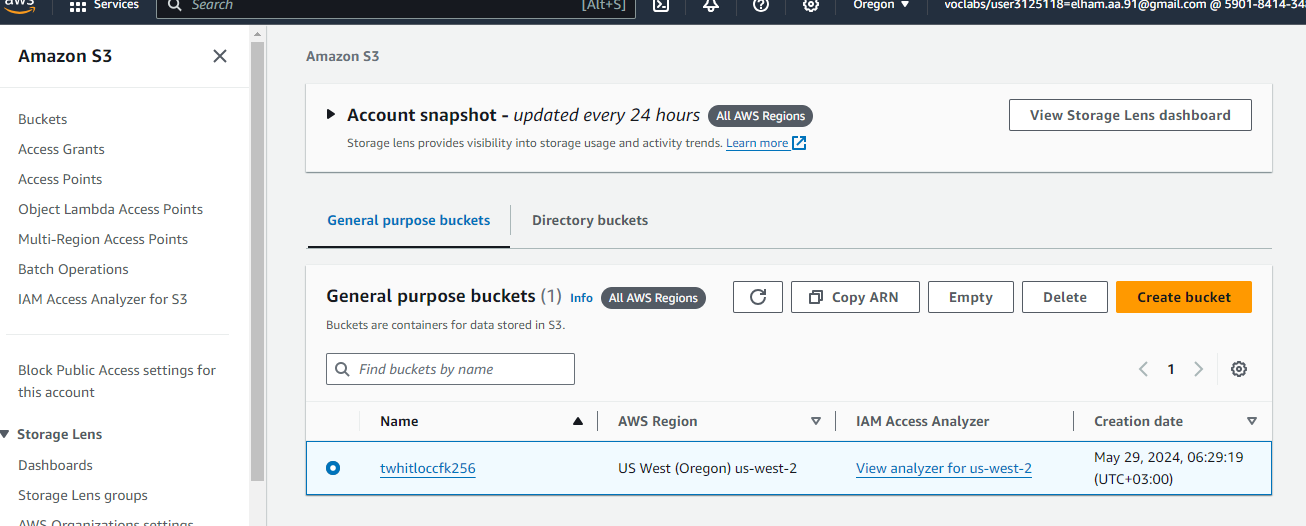
aws iam attach-user-policy --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess --user-name awsS3user



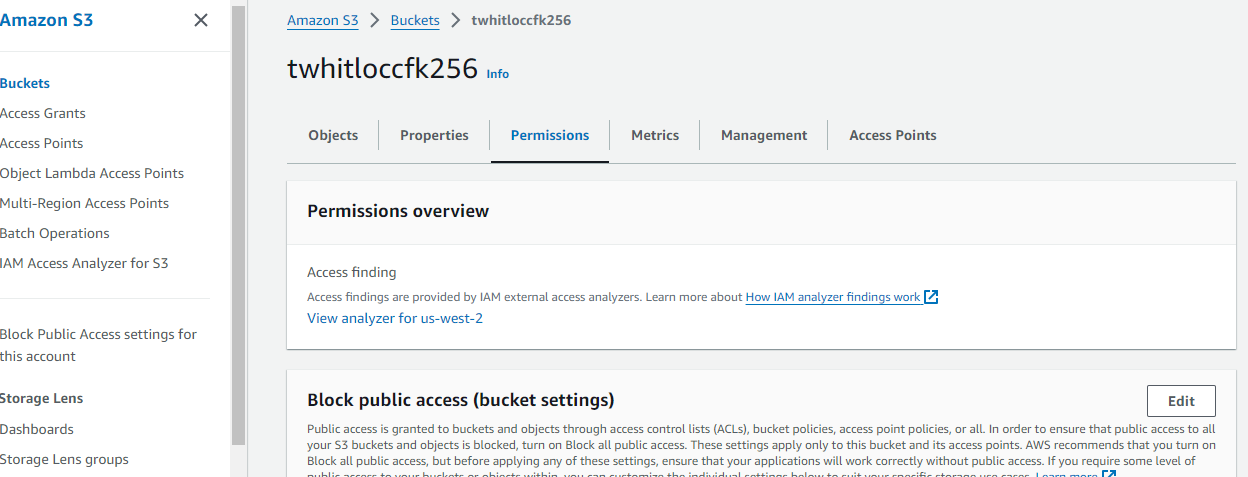
* Returning to the AWS Management Console, and refreshing the browser tab.

*Task 5: Adjusting S3 bucket permissions*

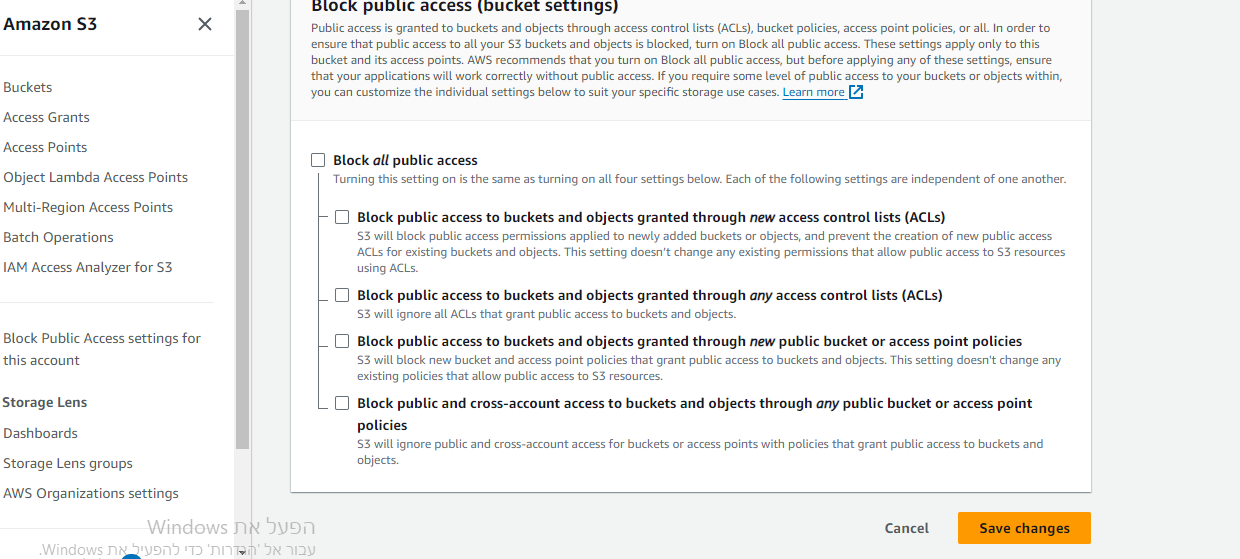
* On the AWS Management Console, on the Amazon S3 console, we choose our bucket name.



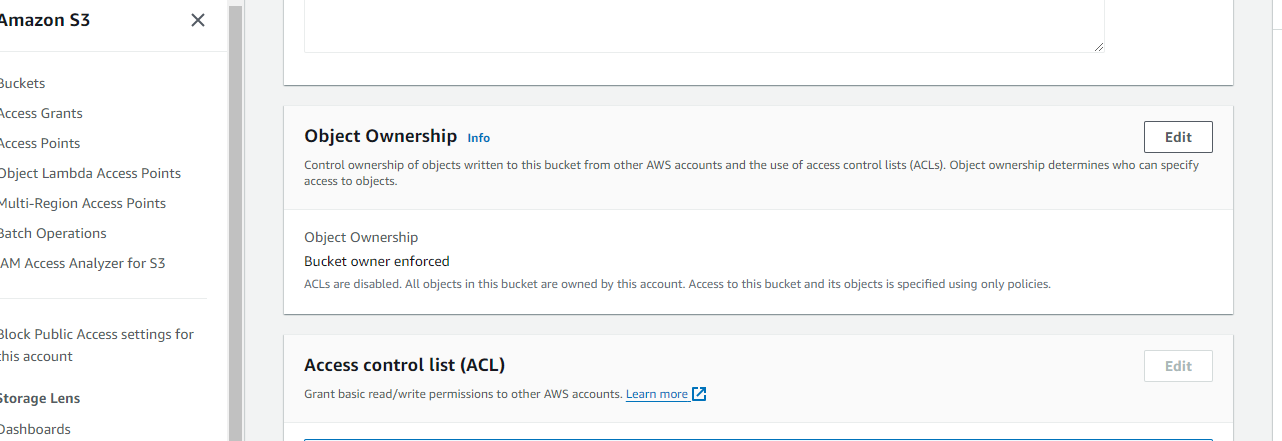
* Going to permissions, under Block public access (bucket settings), choosing Edit



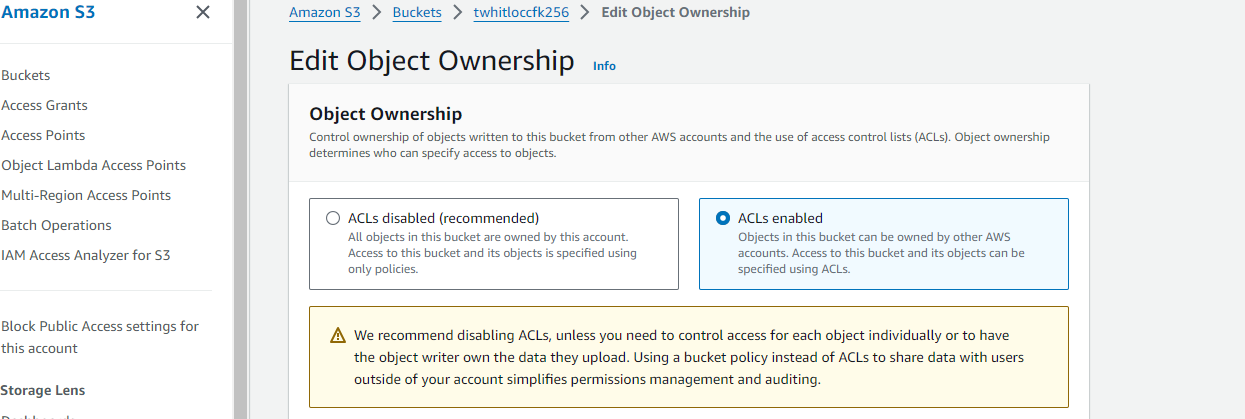
* We Unselect Block all public access ,and then Save changes (confirm on the prompt)



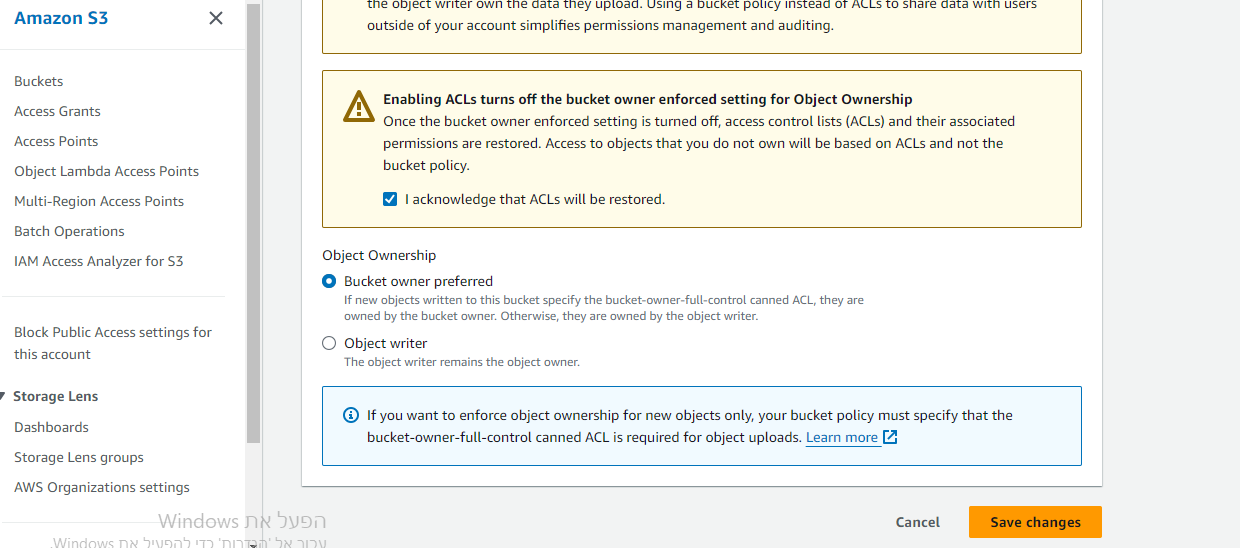
* On to permissions tab, under Object Ownership, we choose Edit



* Choosing ACLs enabled



* Choosing I acknowledge that ACLs will be restored, then Save changes



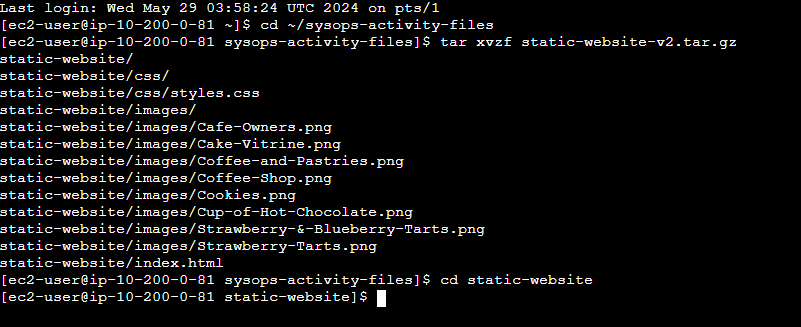
*Task 6: Extracting the files that we need for this lab*

* A file containing the static-website contents for the Amazon S3 bucket will need to be extracted in the following step. We extract the files that we need by running the following commands:

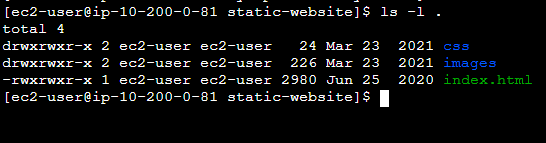
cd ~/sysops-activity-files

tar xvzf static-website-v2.tar.gz

cd static-website



* To confirm that the files were extracted correctly, we run the ls command. we should see a file named index.html and directories named css and images.



*Task 7: Uploading files to Amazon S3 by using the AWS CLI*

Once the files are extracted, we upload the contents of the file to Amazon S3. These files include what we explored when we run the ls command.

* So that the bucket can function as a website, run the following command with our bucket name, and run the adjusted command.

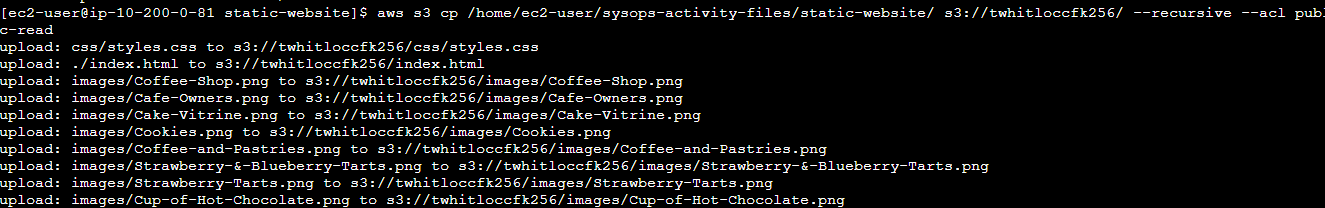
aws s3 website s3://twhitloccfk256/ --index-document index.html



This process helps ensure that the index.html file will be known as the index document.

* To upload the files to the bucket, we run the following command with our bucket name, and run the adjusted command:

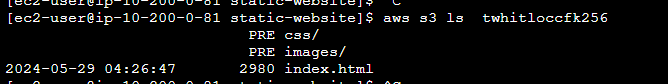
aws s3 cp /home/ec2-user/sysops-activity-files/static-website/ s3://twhitloccfk256/ --recursive --acl public-read



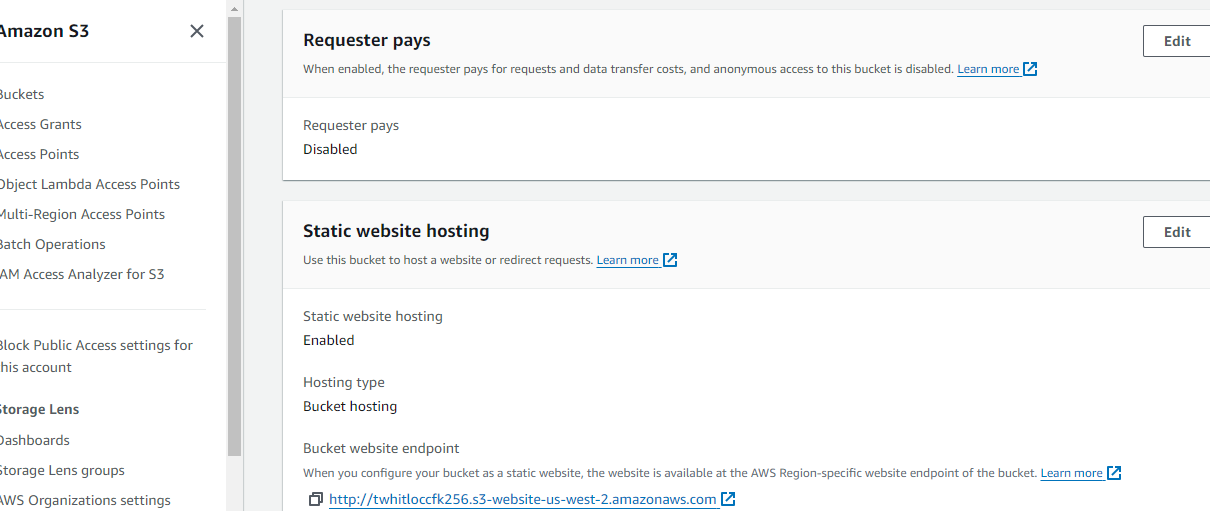
Notice that the upload command includes an access control list (ACL) parameter. This parameter specifies that the uploaded files have public read access. It also includes the recursive parameter, which indicates that all files in the current directory on your machine should be uploaded.

* To verify that the files were uploaded, we run the following command with our bucket name, and run the adjusted command:

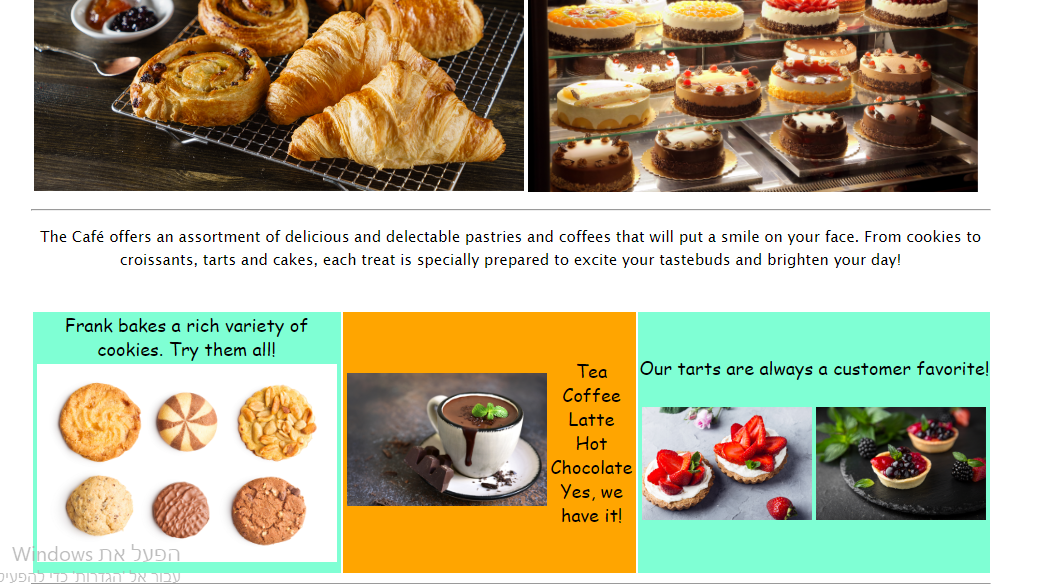
aws s3 ls twhitloccfk256



* On the AWS Management Console, on the Amazon S3 console, we choose our bucket name. Choose the Properties tab. At the bottom of the this tab, note that Static website hosting is Enabled. Running the aws s3 website AWS CLI command turns on the static website hosting for an Amazon S3 bucket. This option is usually turned off by default.
* To open the URL on a new page, we choose the Bucket website endpoint URL that displays.



* We have created a static website that is available to the public for viewing!



*Task 8: Creating a batch file to make updating the website repeatable*

To create a repeatable deployment, we create a batch file by using the VI editor.

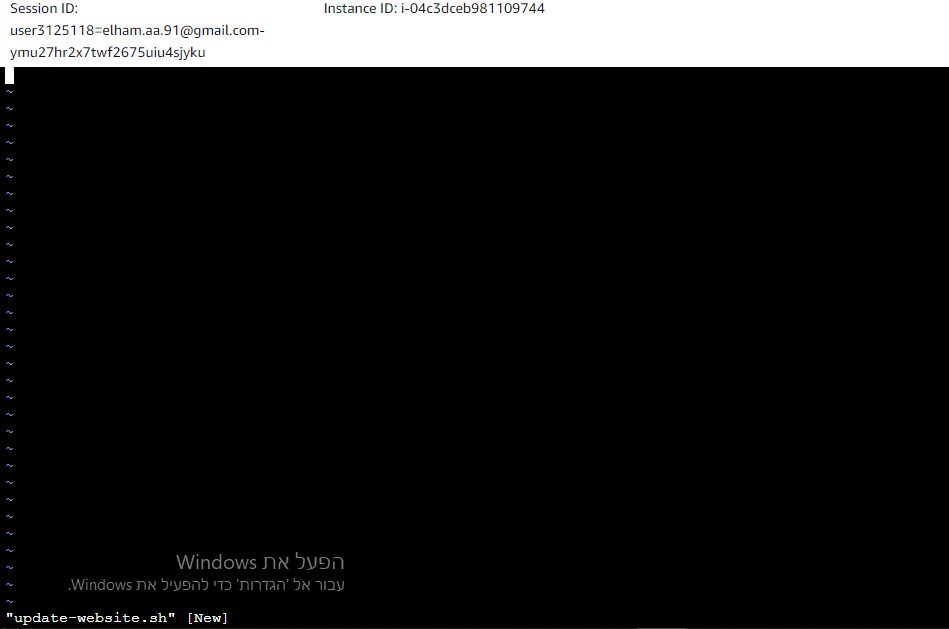
* To change directories and create an empty file, we run the following command in the SSH terminal session:

cd ~

touch update-website.sh

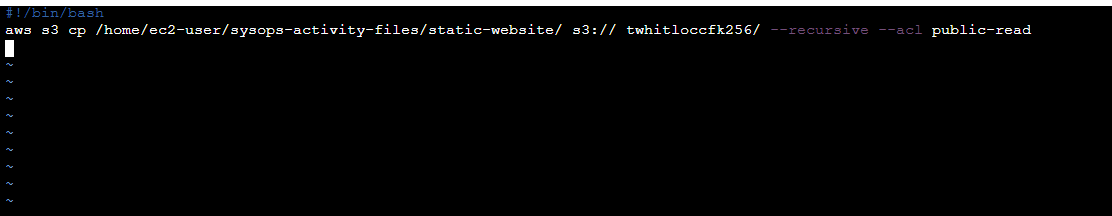
* To open the empty file in the VI editor, we run the following command:

vi update-website.sh



* To enter edit mode in the VI editor, we press i
* Next, we add the following lines into the file:

#!/bin/bash  
aws s3 cp /home/ec2-user/sysops-activity-files/static-website/ s3://twhitloccfk256/ --recursive --acl public-read



* To write the changes and quit the file, we press Esc, enter :wq and then press Enter.
* To make the file an executable batch file, we run the following command:

chmod +x update-website.sh



* To open the local copy of the index.html file in a text editor, we run the following command:

vi sysops-activity-files/static-website/index.html

* To enter edit mode in the VI editor, we press i and modify the file as follows:

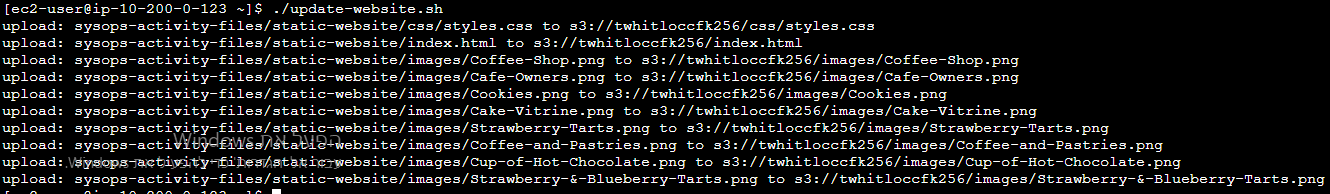
1. Locating the first line that has the HTML code bgcolor="aquamarine" and changing this code to bgcolor="gainsboro"
2. Locating the line that has the HTML code bgcolor="orange" and changing this code to bgcolor="cornsilk"
3. Locating the second line that has the HTML code bgcolor="aquamarine" and changing this code to bgcolor="gainsboro"



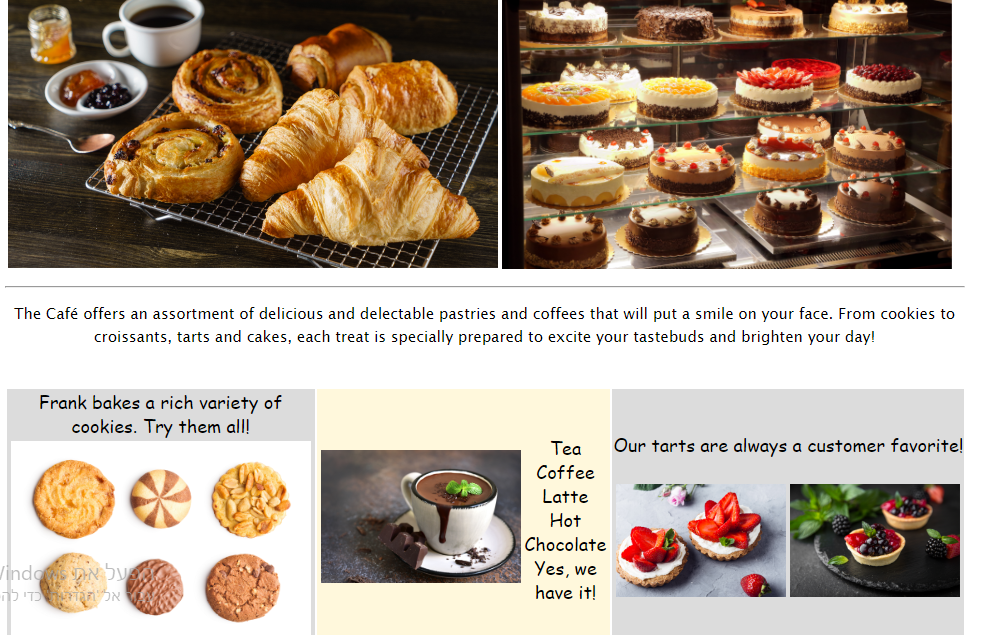
* To write the changes and quit the file,we press Esc, enter :wq and then Enter.
* To update the website, we run our batch file.

./update-website.sh

Note: The command line output should show that the files were copied to Amazon S3.



* To see the changes to the website, we return to the browser and refresh the Café and Bakery page.



*Task 9: challenge*

We notice that our batch file uploads every file to Amazon S3 every time we run it even when most of the files have no changes to them?

To help make the script more efficient, we replace the aws s3 cp command that we've been using with the aws s3 sync command from this document.

aws s3 sync /home/ec2-user/sysops-activity-files/static-website/<s3:// twhitloccfk256/ --acl public-read