

AWS

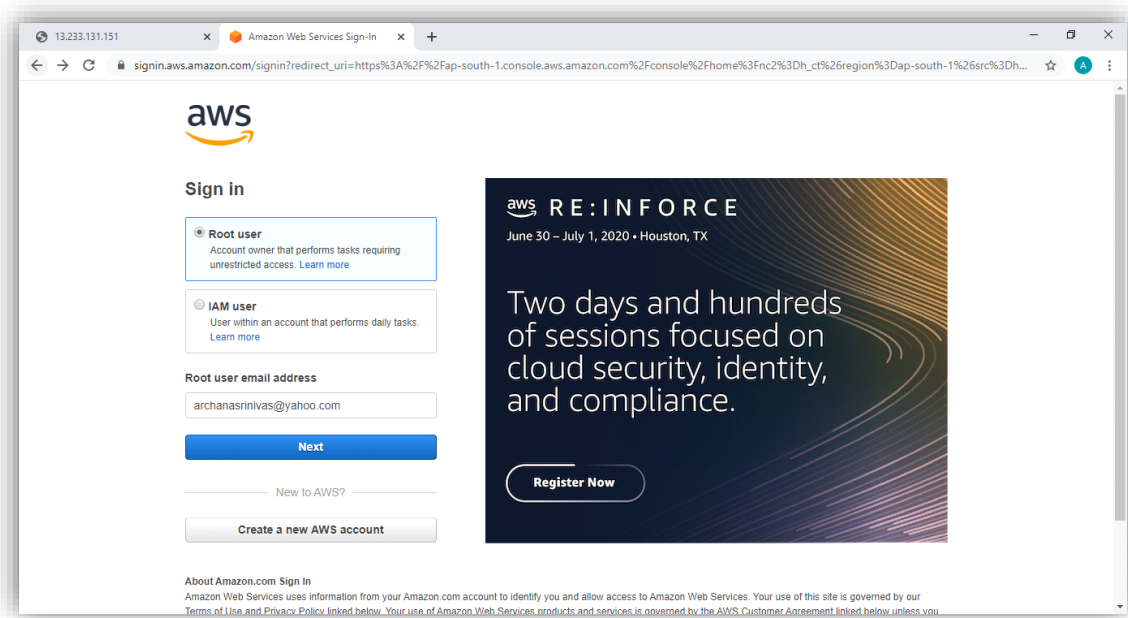
FACE DETECTION APP



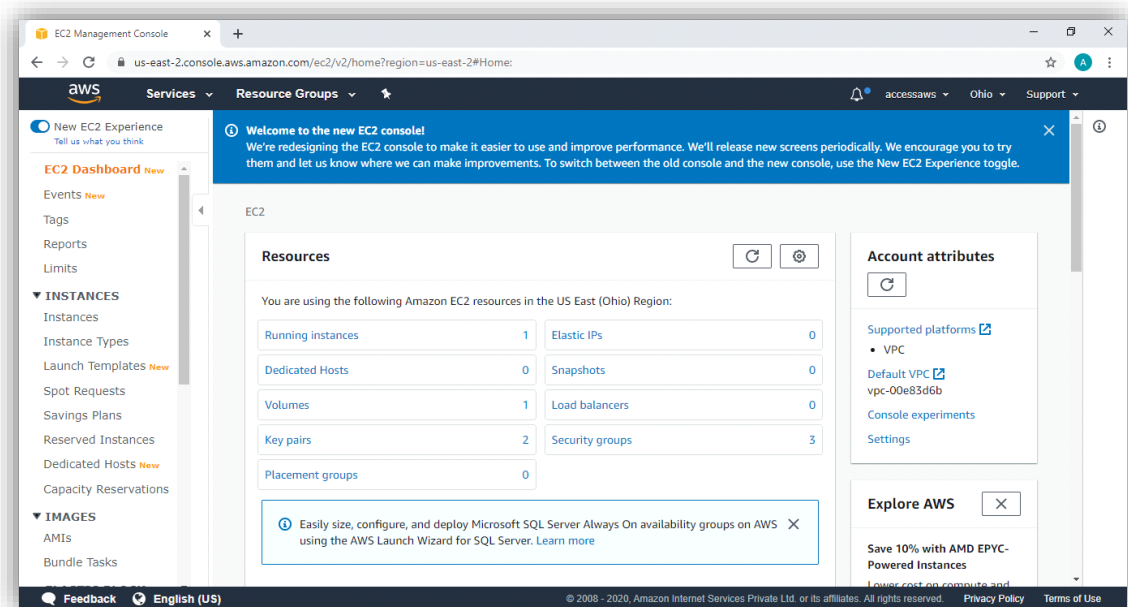
PROJECT SUBMITTED
BY
ARCHANA S

SCREENSHOTS OF DASHBOARDS

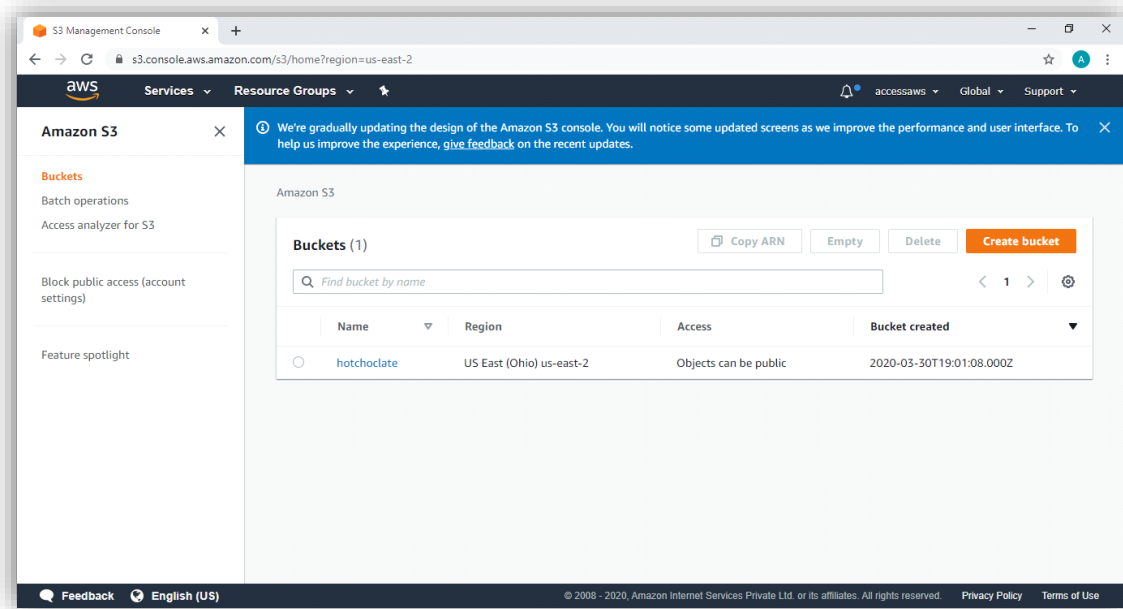
1. AWS LOGIN SCREEN WITH USERNAME



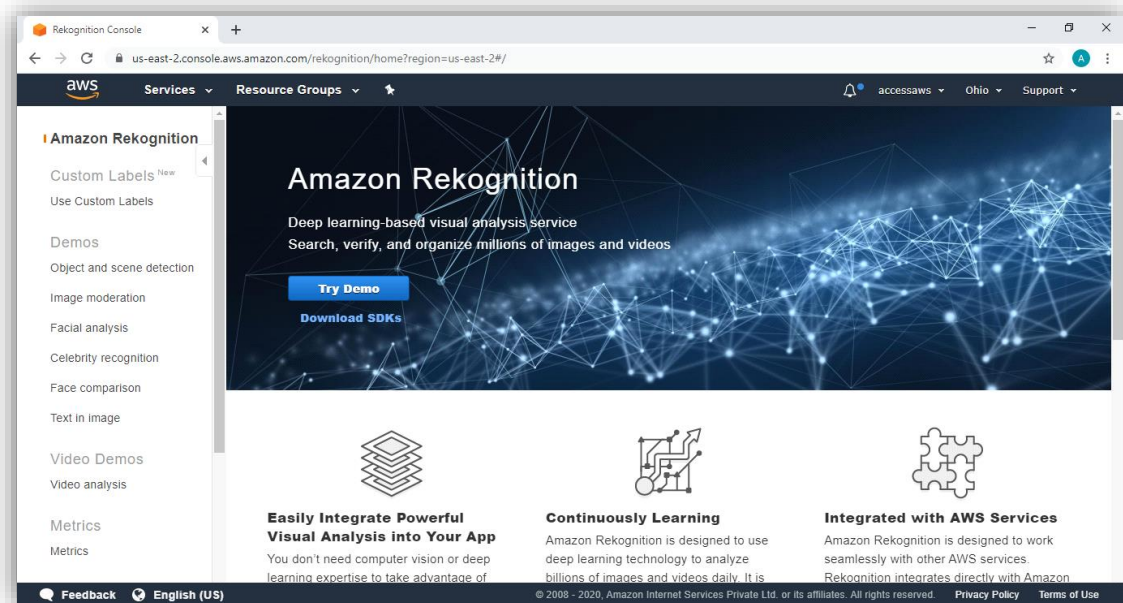
2. EC2 DASHBOARD



3. S3 DASHBOARD

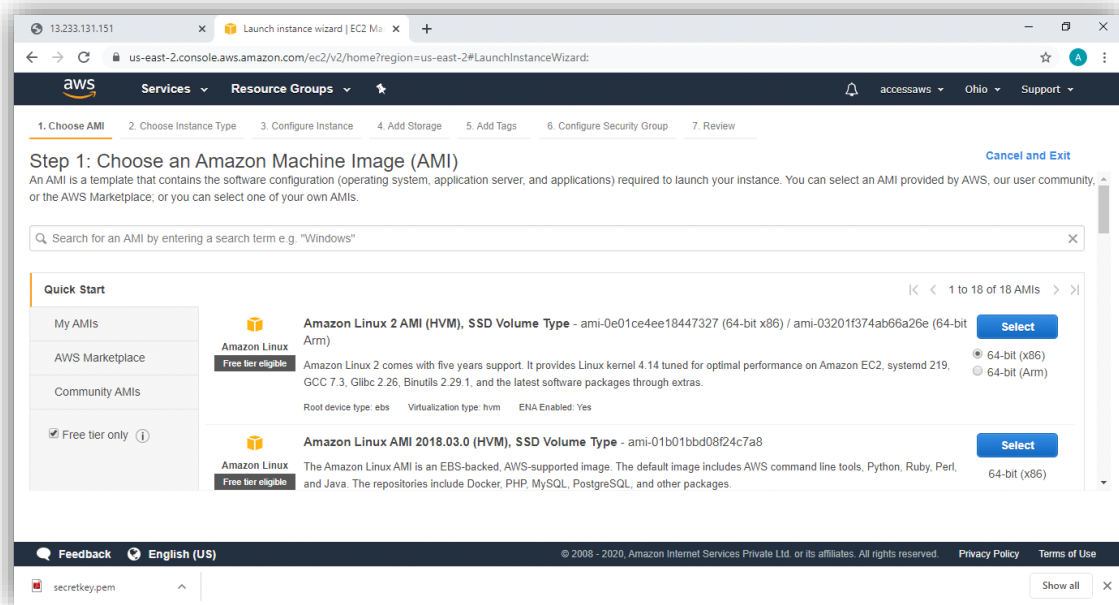


4. REKOGNITION DASHBOARD

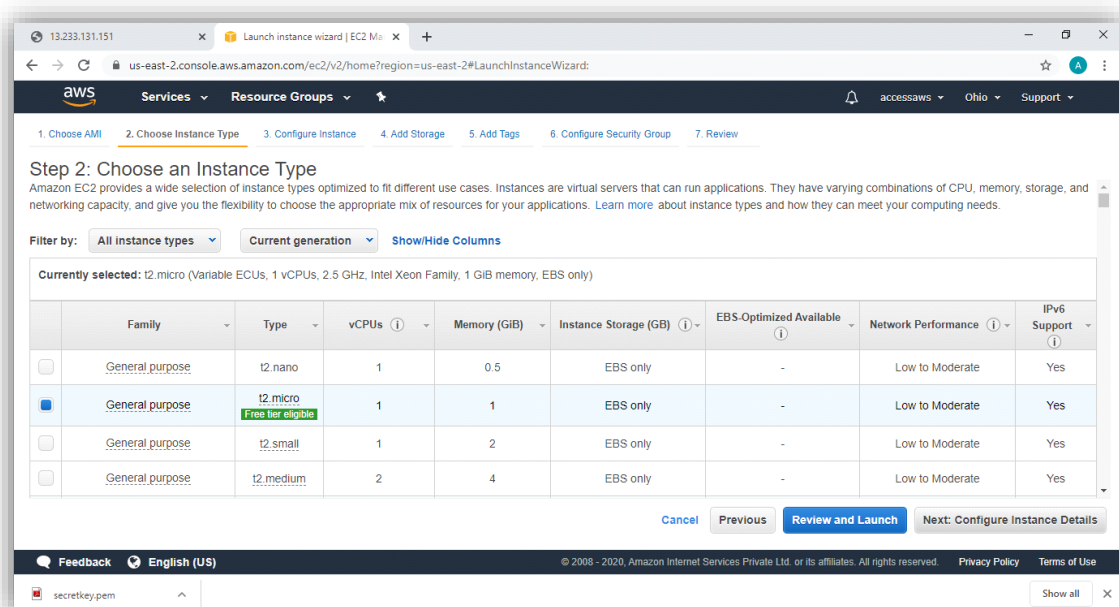


SCREENSHOTS OF EC2

1. CHOOSING AN AMI



2. CHOOSING AN INSTANCE TYPE



3. ADDING STORAGE

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0f54692056aaa4c20	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

4. CONFIGURING SECURITY GROUP

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

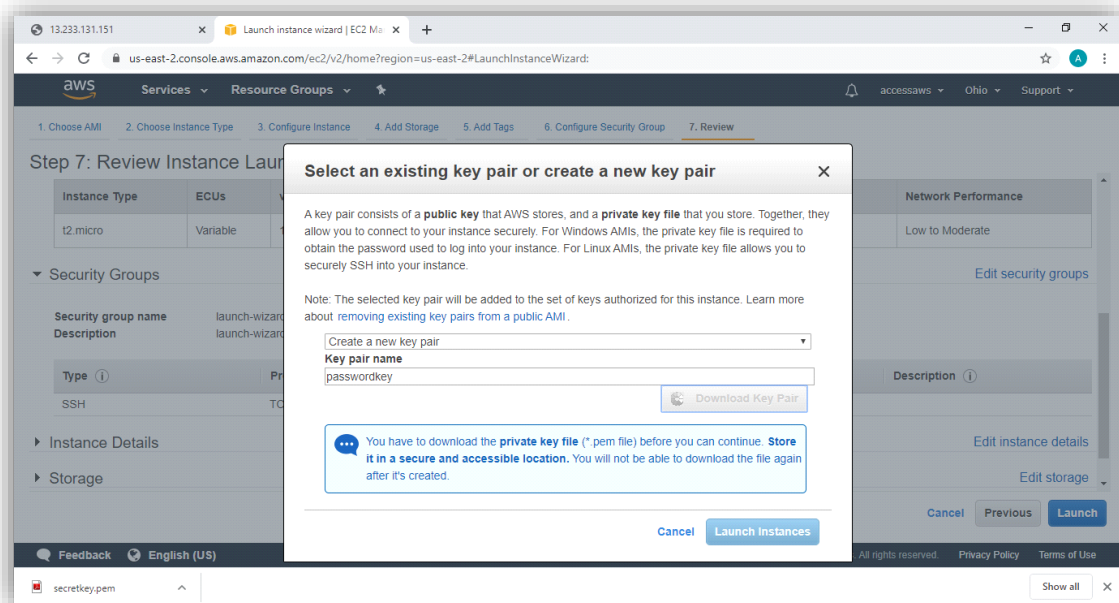
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

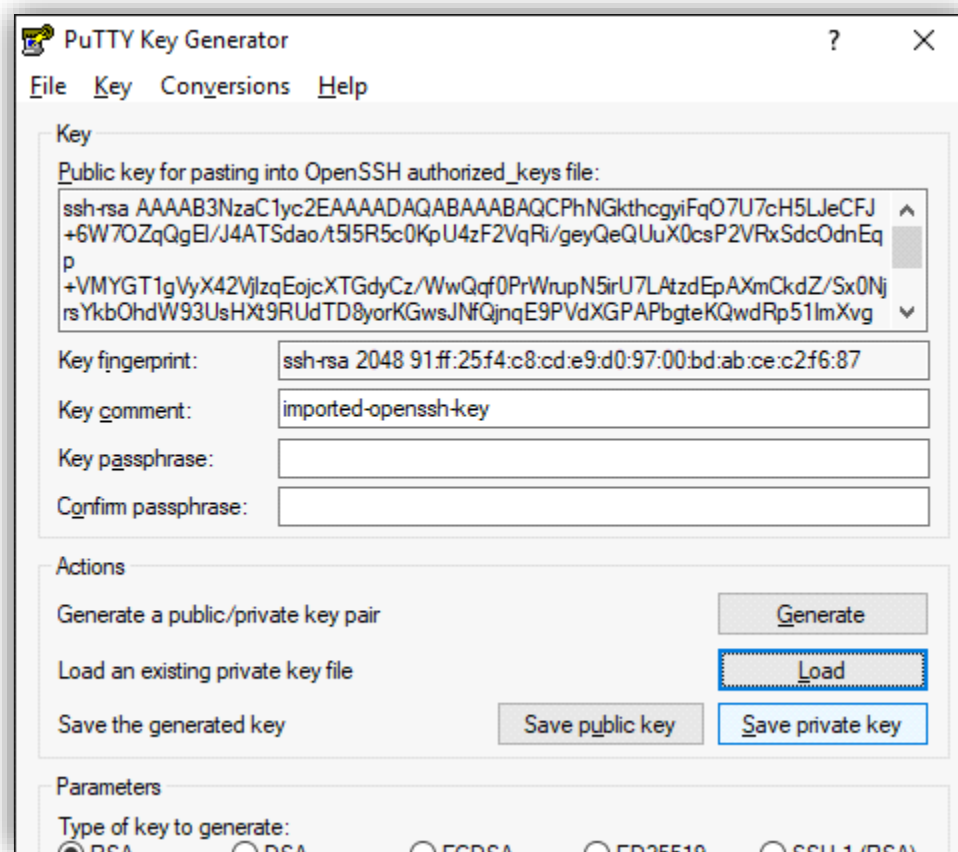
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

5. KEY PAIR DOWNLOAD



6. PUTTYGEN CONVERSION FROM PEM TO PPK

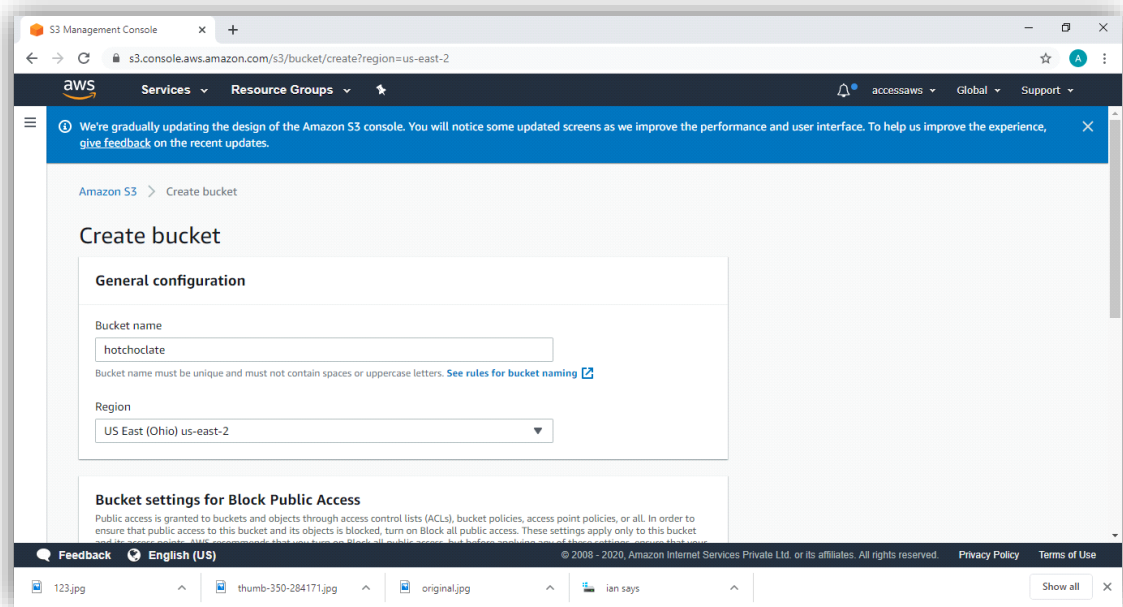


11. LOGGED IN EC2 BLACK SCREEN

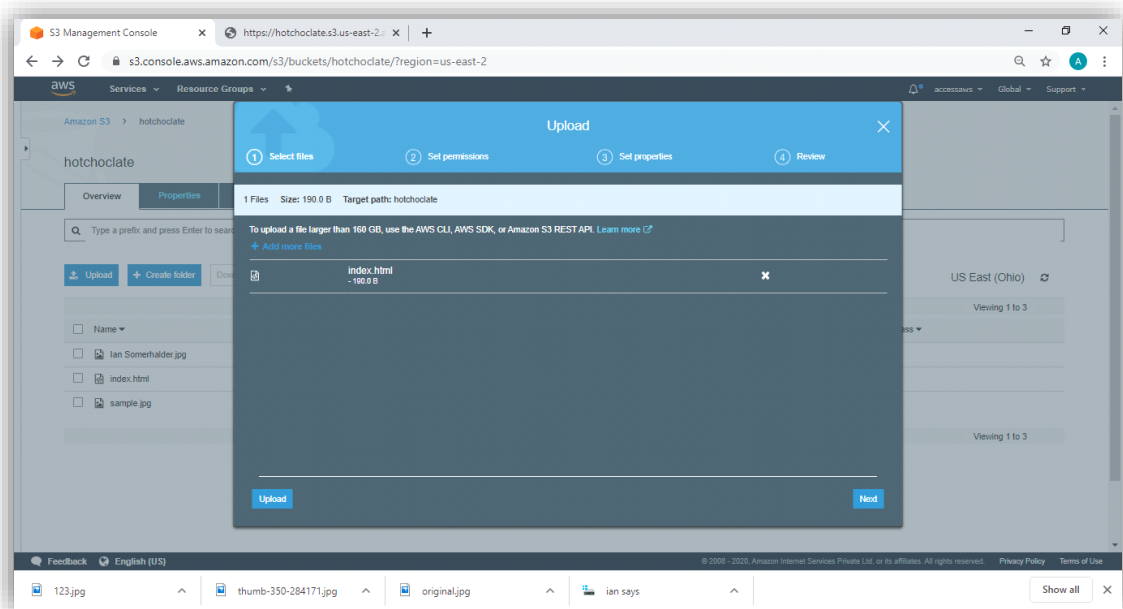
[illegible]

SCREENSHOTS OF S3

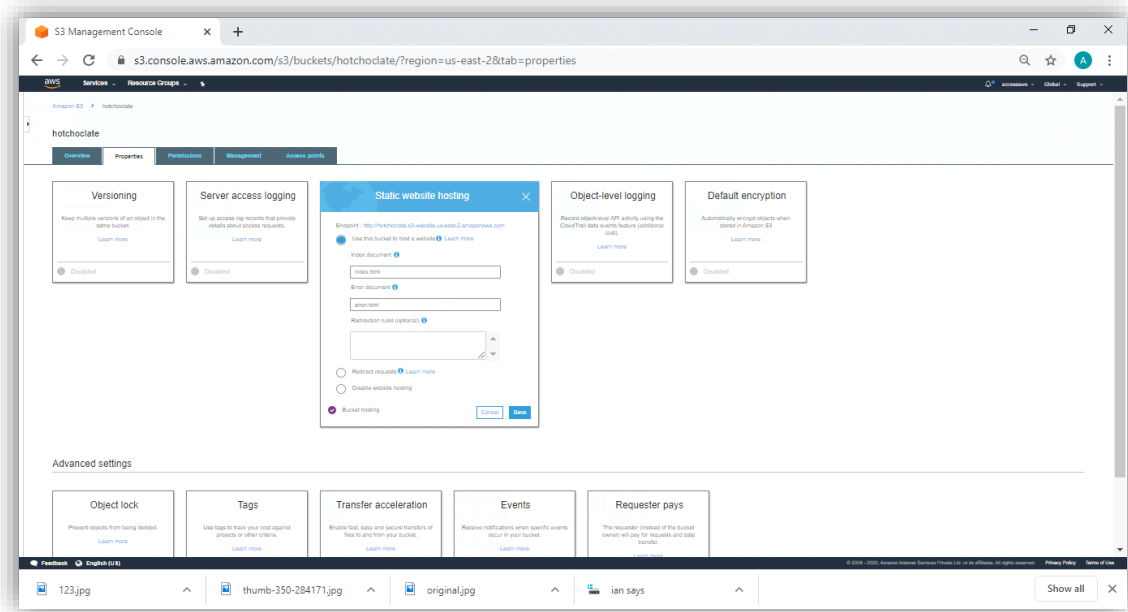
1. CREATING A BUCKET



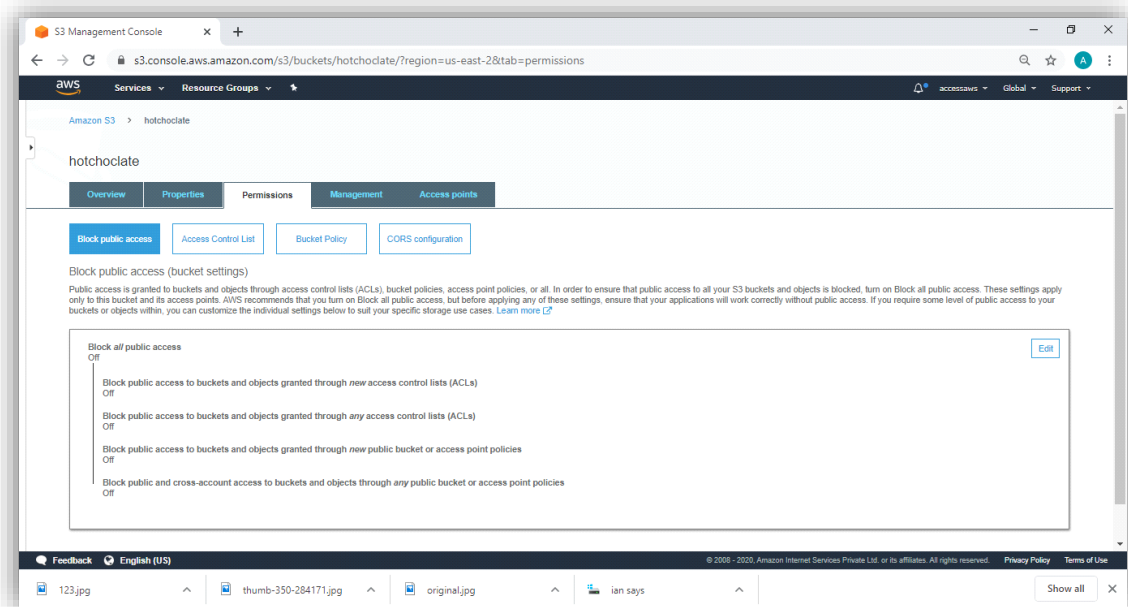
2.UPLOADING AN OBJECT



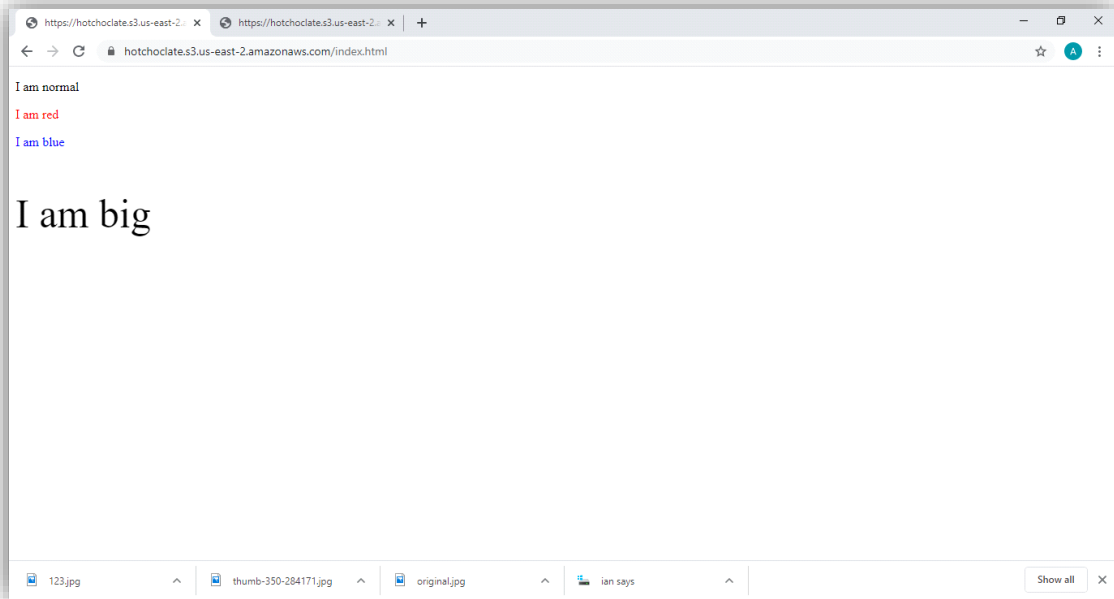
3. ENABLE STATIC WEBSITE



4. MAKING THE OBJECT PUBLIC

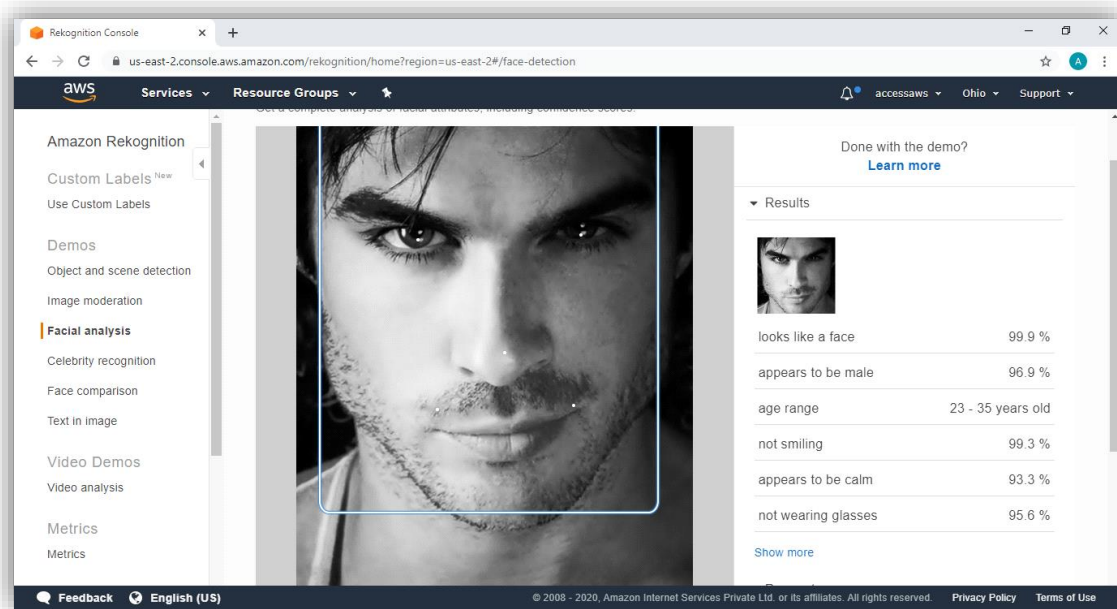


5. CHECKING THE S3 LINK ON THE BROWSER

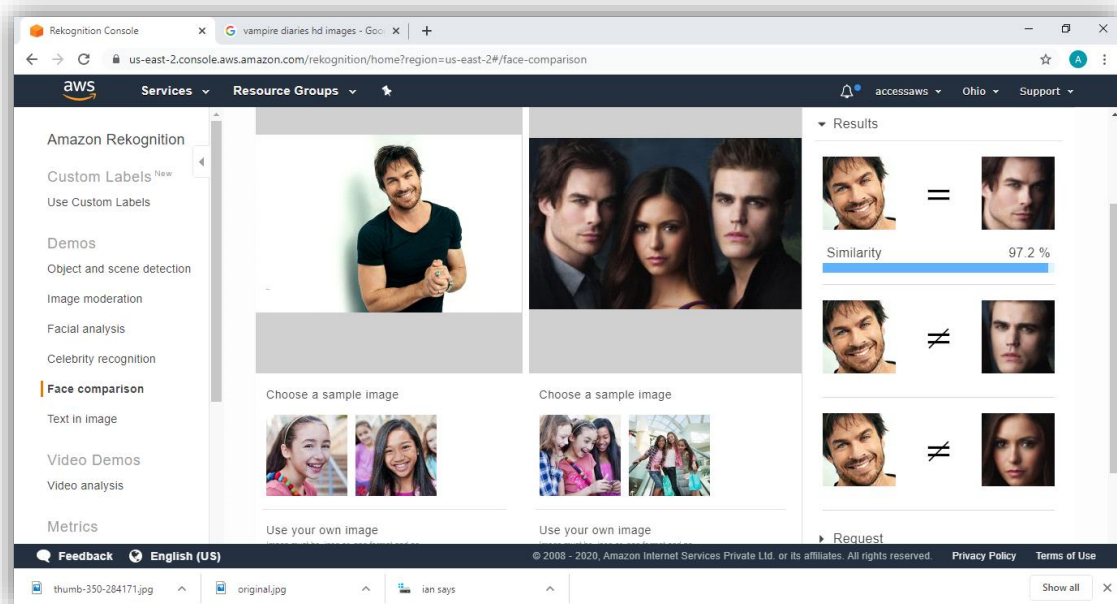


SCREENSHOTS OF FACE REKOGNITION

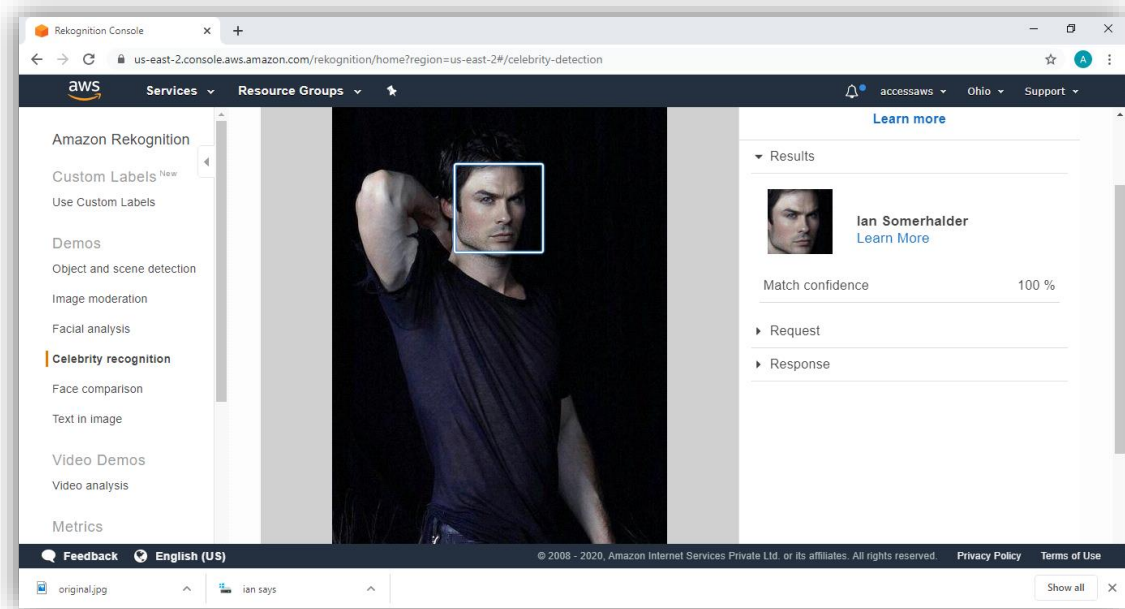
1. FACE DETECT



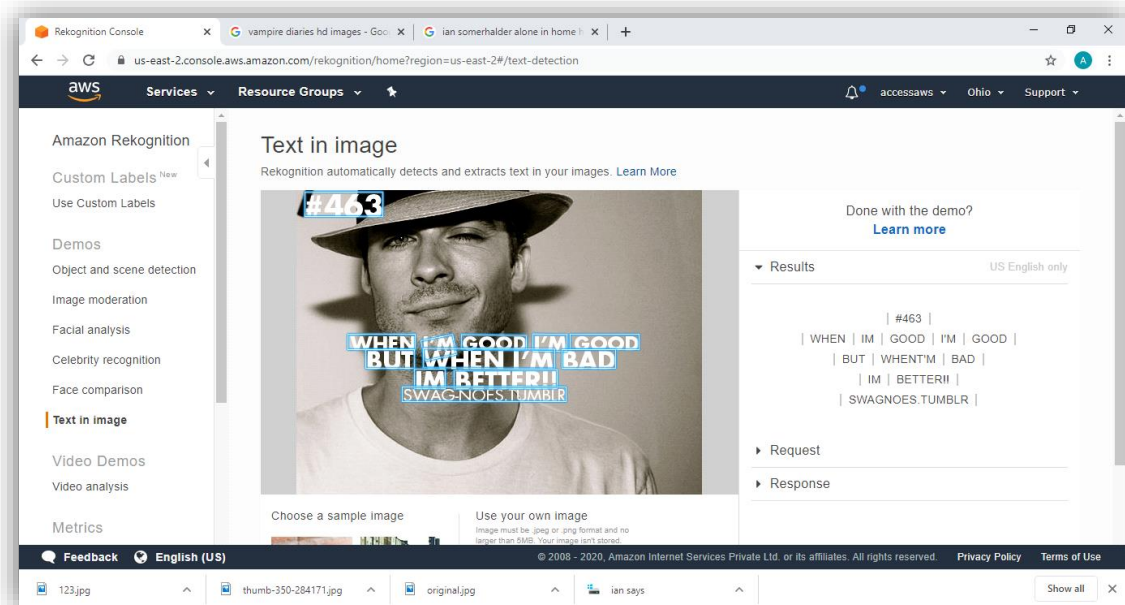
2. FACE COMPARE



3. CELEBRITY REKOGNITION



4. TEXT IN IMAGE



SCREENSHOTS OF EC2 AND S3

1. INSTALLING AWS-SDK

```
ec2-user@ip-172-31-19-42:/var/www/html/face
Package operations: 0 installs, 1 update, 0 removals
- Updating aws/aws-sdk-php (3.134.2 => 3.134.3): Downloading (100%)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/gu
zle instead.
Writing lock file
Generating autoload files
! package you are using is looking for funding.
Use the "composer fund" command to find out more!
[ec2-user@ip-172-31-19-42 face]$
[ec2-user@ip-172-31-19-42 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
--2020-04-04 16:48:30-- https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 23.210.196.239, 2600:1408:20:aa6:1931, 2600:1408:20:a89:1931, ...
Connecting to i.pinimg.com (i.pinimg.com)|23.210.196.239|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215551 (210K) [image/jpeg]
Saving to: 'b97ea33b5842c7894b804923c6c05580.jpg'

100%[=====] 215,551  --.-K/s  in 0.04s

2020-04-04 16:48:30 (5.65 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-19-42 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ ls
b97ea33b5842c7894b804923c6c05580.jpg  composer.json  composer.lock  index.php  sample.jpg  vendor
[ec2-user@ip-172-31-19-42 face]$ cd
[ec2-user@ip-172-31-19-42 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg image2.jpg
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  sample.jpg  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo rm sample.jpg
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo rm index.php
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.
com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ sudo php
index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws
```

2. INSTALLING PHP

```
ec2-user@ip-172-31-19-42:/var/www/html/face
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package Arch Version Repository Size
-----
Installing:
php-mbstring x86_64 7.2.28-1.amzn2 amzn2extra-php7.2 501 k
php-xml x86_64 7.2.28-1.amzn2 amzn2extra-php7.2 211 k
Transaction Summary
-----
Install 2 Packages

Total download size: 712 k
Installed size: 2.8 M
Is this ok [y/d/N]: y
Downloading packages:
(1/2): php-mbstring-7.2.28-1.amzn2.x86_64.rpm | 501 kB 00:00
(2/2): php-xml-7.2.28-1.amzn2.x86_64.rpm | 211 kB 00:00
-----
Total 4.1 MB/s | 712 kB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : php-mbstring-7.2.28-1.amzn2.x86_64 1/2
Installing : php-xml-7.2.28-1.amzn2.x86_64 2/2
Verifying : php-xml-7.2.28-1.amzn2.x86_64 1/2
Verifying : php-mbstring-7.2.28-1.amzn2.x86_64 2/2

Installed:
php-mbstring.x86_64 0:7.2.28-1.amzn2 php-xml.x86_64 0:7.2.28-1.amzn2

Complete!
[ec2-user@ip-172-31-19-42 ~]$ sudo yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.41-1.amzn2.0.1 will be installed
--> Processing Dependency: httpd-tools = 2.4.41-1.amzn2.0.1 for package: httpd-2
4.41-1.amzn2.0.1.x86_64
```

3. INDEX.PHP FILE CODE

```
ec2-user@ip-172-31-19-42:/var/www/html/face
sudo /sbin/mkswap /var/swap.1
sudo /sbin/swapon /var/swap.1

sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg

#Incase if you are getting any class NOT found error, follow these steps

sudo yum remove php*
sudo yum remove httpd*
sudo yum clean all
sudo yum upgrade -y
sudo amazon-linux-extras install php7.2
sudo yum install php-json php-xml php-cli php-mbstring
sudo yum install httpd

//
// error_reporting(0);

require_once(__DIR__ . '/vendor/autoload.php');

use Aws\S3\S3Client;
use Aws\Rekognition\RekognitionClient;

$bucket = 'hotchocolate';
$keyname = 'image2.jpg';

$s3 = new S3Client([
    'region'      => 'us-east-2',
    'version'     => '2006-03-01',
    'signature'   => 'v4'
]);

try {
    // Upload data
    $result = $s3->putObject([
        'Bucket'      => $bucket,
        'Key'         => $keyname,
        'SourceFile'  => __DIR__ . "/$keyname",
        'ACL'         => 'public-read-write'
    ]);
}
```

4. UPLOAD SUCCESS SCREENSHOT

```
ec2-user@ip-172-31-19-42:/var/www/html/face
HTTP request sent, awaiting response... 200 OK
Length: 215551 (210K) [image/jpeg]
Saving to: 'b97ea33b5842c7894b804923c6c05580.jpg'

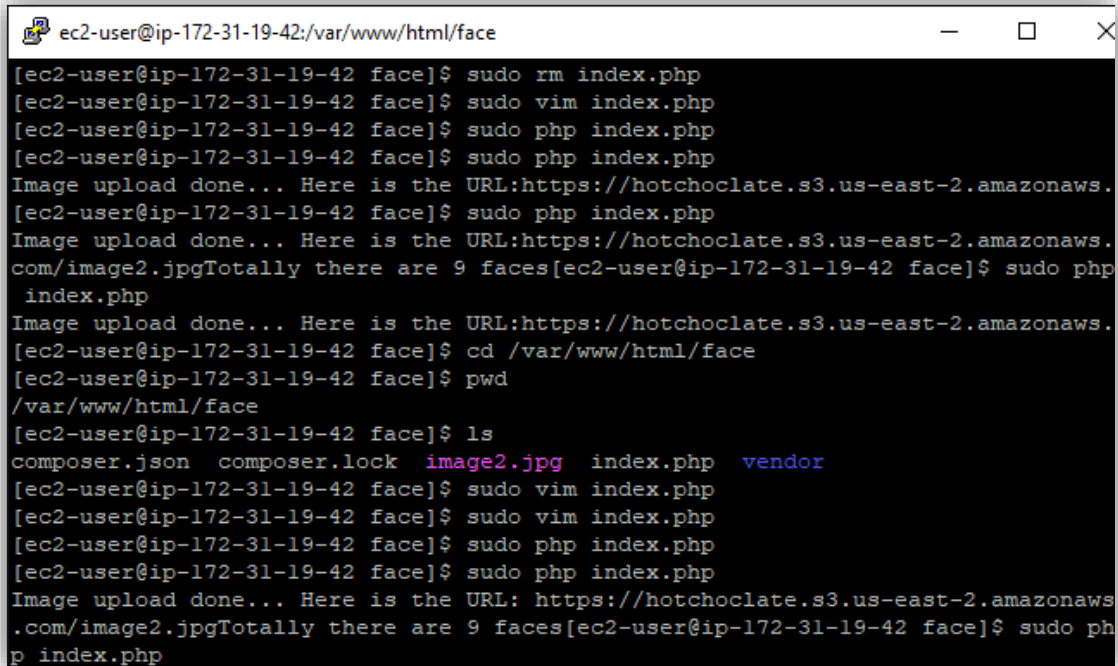
100%[=====] 215,551  --.-K/s  in 0.04s

2020-04-04 16:48:30 (5.65 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-19-42 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ ls
b97ea33b5842c7894b804923c6c05580.jpg  composer.json  composer.lock  index.php  sample.jpg  vendor
[ec2-user@ip-172-31-19-42 face]$ ^C
[ec2-user@ip-172-31-19-42 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg image2.jpg
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  sample.jpg  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo rm sample.jpg
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo rm index.php
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/
[ec2-user@ip-172-31-19-42 face]$ cd /var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL: https://hotchocolate.s3.us-east-2.amazonaws.com/
```

SCREENSHOTS OF EC2 AND REKOGNITION

1. FACE DETECT SUCCESS SCREENSHOT

A terminal window titled 'ec2-user@ip-172-31-19-42:/var/www/html/face' with standard window controls. The terminal shows a series of commands and their outputs. The user runs 'sudo rm index.php', 'sudo vim index.php', and 'sudo php index.php'. The output of the first 'php' command is 'Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces'. The user then runs 'sudo php index.php' again, and the output is identical. Next, the user runs 'cd /var/www/html/face', 'pwd' (outputting '/var/www/html/face'), and 'ls' (outputting 'composer.json composer.lock image2.jpg index.php vendor'). Finally, the user runs 'sudo vim index.php', 'sudo php index.php', and 'sudo php index.php'. The output of the last 'php' command is 'Image upload done... Here is the URL: https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces'.

```
ec2-user@ip-172-31-19-42:/var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ sudo rm index.php
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL:https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ cd /var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-19-42 face]$ ls
composer.json  composer.lock  image2.jpg  index.php  vendor
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
Image upload done... Here is the URL: https://hotchocolate.s3.us-east-2.amazonaws.com/image2.jpgTotally there are 9 faces[ec2-user@ip-172-31-19-42 face]$ sudo php index.php
```