

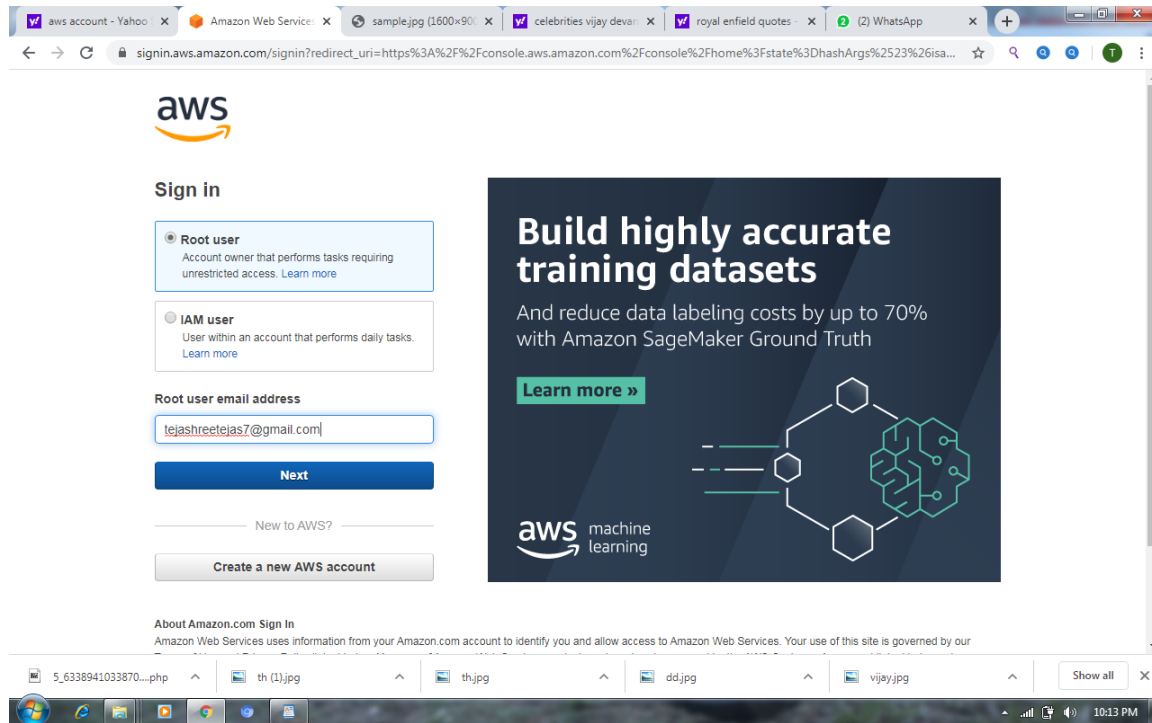
# FACE-DETECTION USING AWS



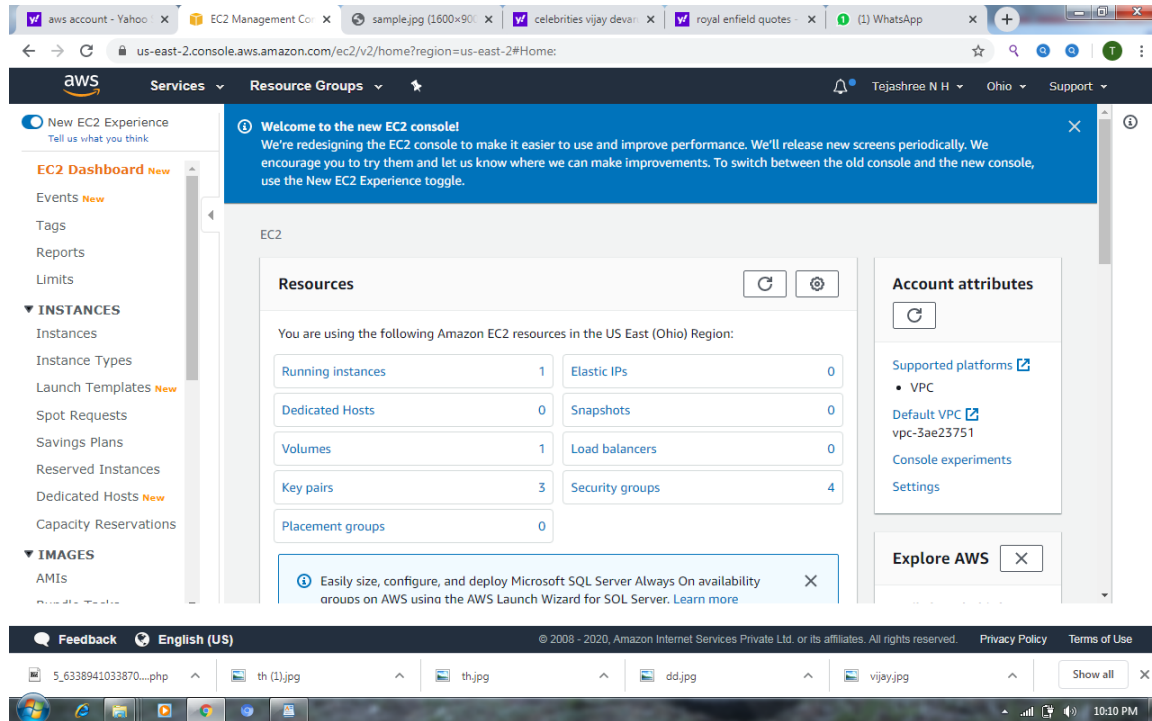
PROJECT BY:  
TEJASHREE N H

# Dashboards Screenshots

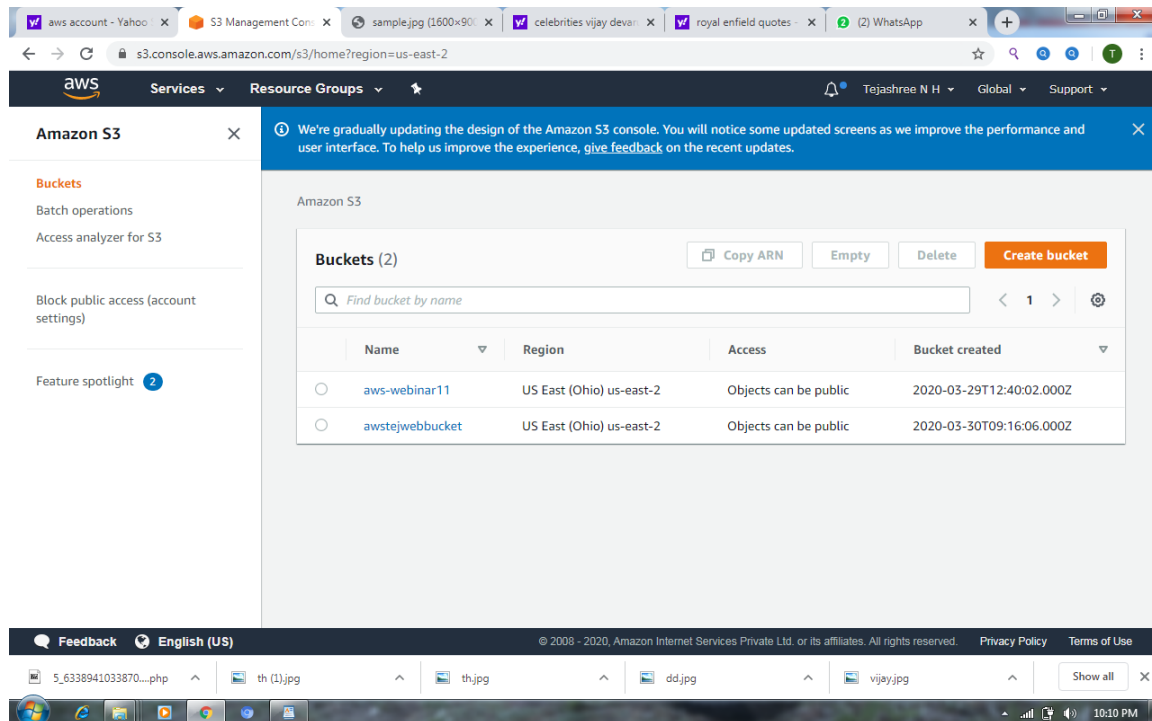
## 1. AWS Login Screen with username :



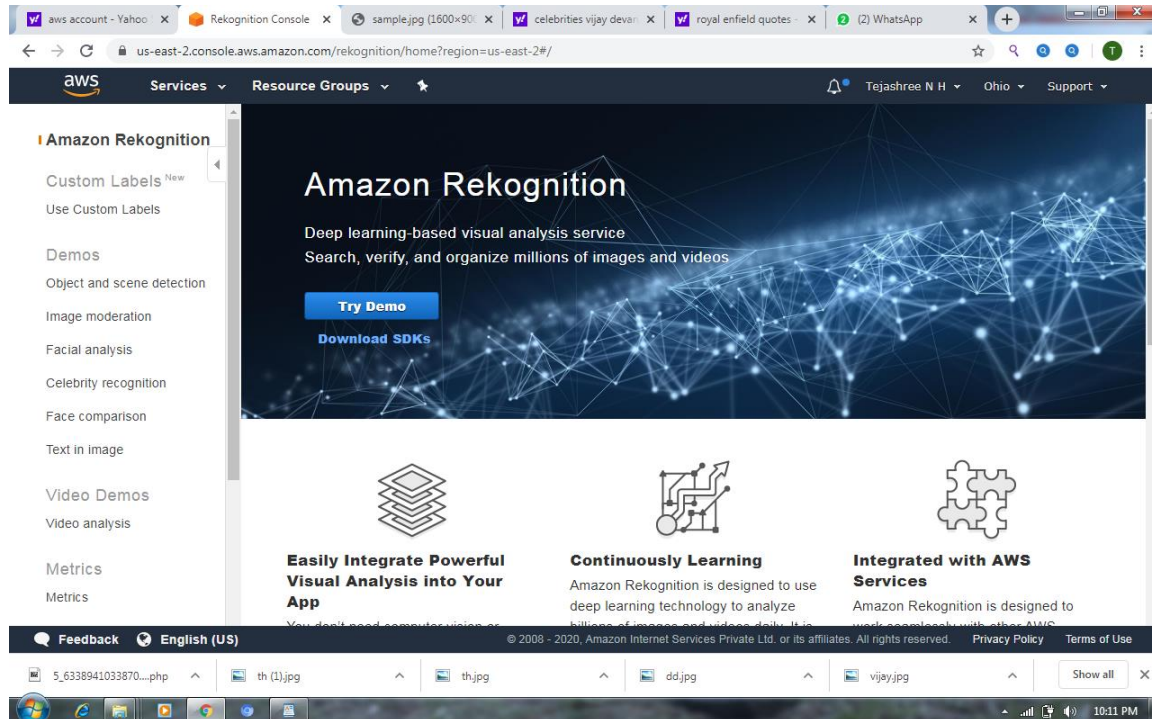
## 2. EC2 Dashboard :



### 3. S3 Dashboard :

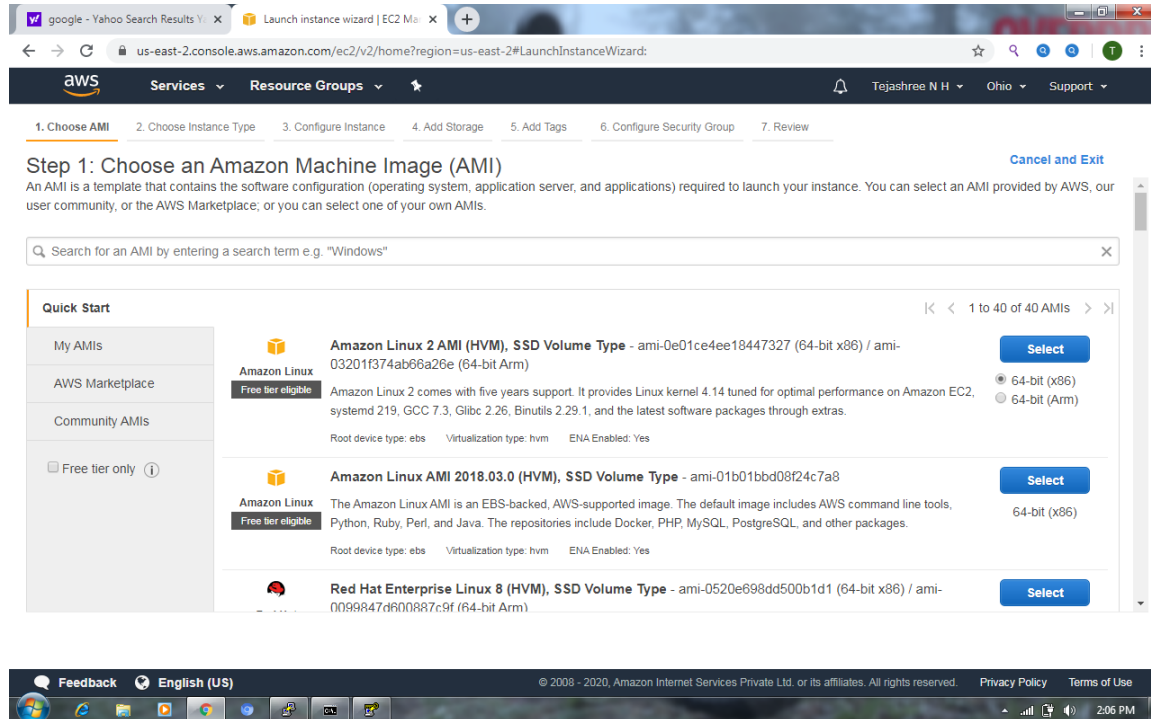


## 4. Rekognition Dashboard :



# EC2 Screenshots

## 1. Choosing an AMI :



## 2. Choosing an instance Type :

The screenshot shows the AWS Management Console interface for the 'Configure Instance Details' step of the EC2 instance launch wizard. The breadcrumb navigation at the top indicates the current step is '3. Configure Instance' out of seven steps. The main heading is 'Step 3: Configure Instance Details', followed by a descriptive paragraph. The configuration form includes several sections: 'Number of Instances' (set to 1), 'Purchasing option' (with a checkbox for 'Request Spot Instances'), 'Network' (selected VPC: vpc-3ae23751), 'Subnet' (selected: No preference), 'Auto-assign Public IP' (selected: Use subnet setting), 'Placement group' (checkbox for 'Add instance to placement group'), 'Capacity Reservation' (selected: Open), 'IAM role' (selected: None), 'Shutdown behavior' (selected: Stop), and 'Stop - Hibernate behavior' (checkbox for 'Enable hibernation'). At the bottom of the form are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'. The footer of the console shows the language set to English (US), copyright information, and system status icons.

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of Instances** ⓘ 1 [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot Instances

**Network** ⓘ vpc-3ae23751 (default) [Create new VPC](#)

**Subnet** ⓘ No preference (default subnet in any Availability Zone) [Create new subnet](#)

**Auto-assign Public IP** ⓘ Use subnet setting (Enable) ⓘ

**Placement group** ⓘ ☐ Add instance to placement group

**Capacity Reservation** ⓘ Open [Create new Capacity Reservation](#)

**IAM role** ⓘ None [Create new IAM role](#)

**Shutdown behavior** ⓘ Stop

**Stop - Hibernate behavior** ⓘ ☐ Enable hibernation as an additional stop behavior

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

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 2:07 PM

### 3. Adding Storage :

The screenshot shows the AWS Management Console interface for the 'Launch Instance Wizard'. The browser address bar shows the URL: `us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:`. The navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (Tejashree N H, Ohio, Support). The wizard progress bar shows steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (active), 5. Add Tags, 6. Configure Security Group, 7. Review.

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

[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](#)

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 2:08 PM



## 4. Configuration Security Group :

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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](#)

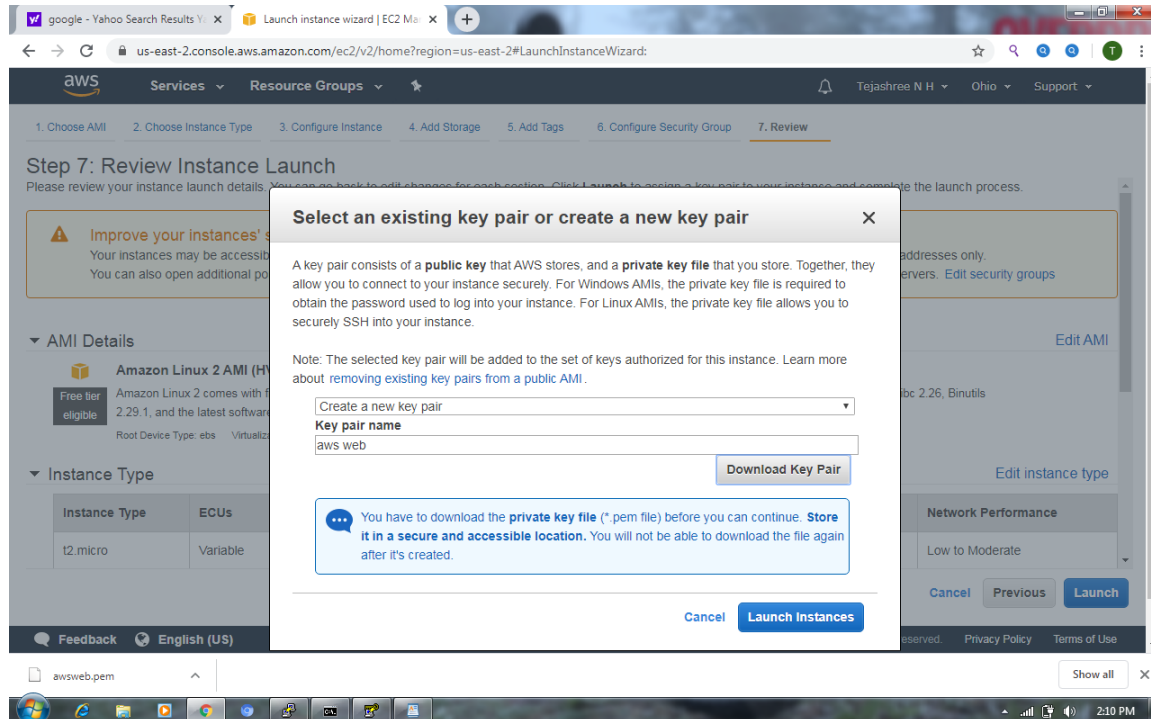
**Warning**

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.

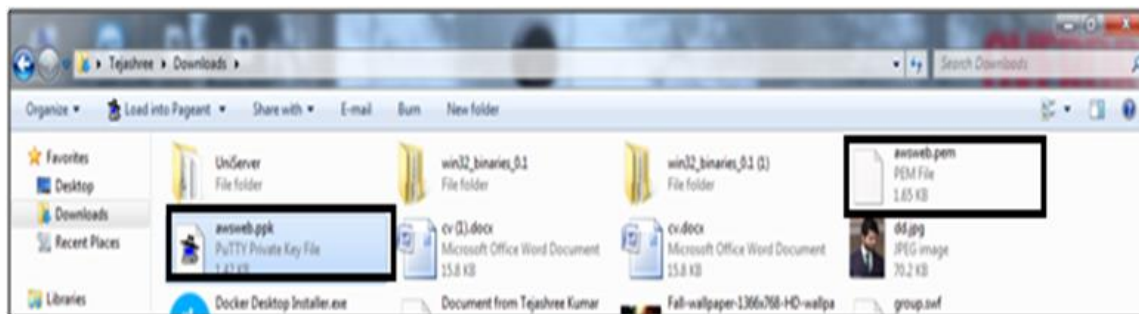
[Cancel](#) [Previous](#) [Review and Launch](#)

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 2:09 PM

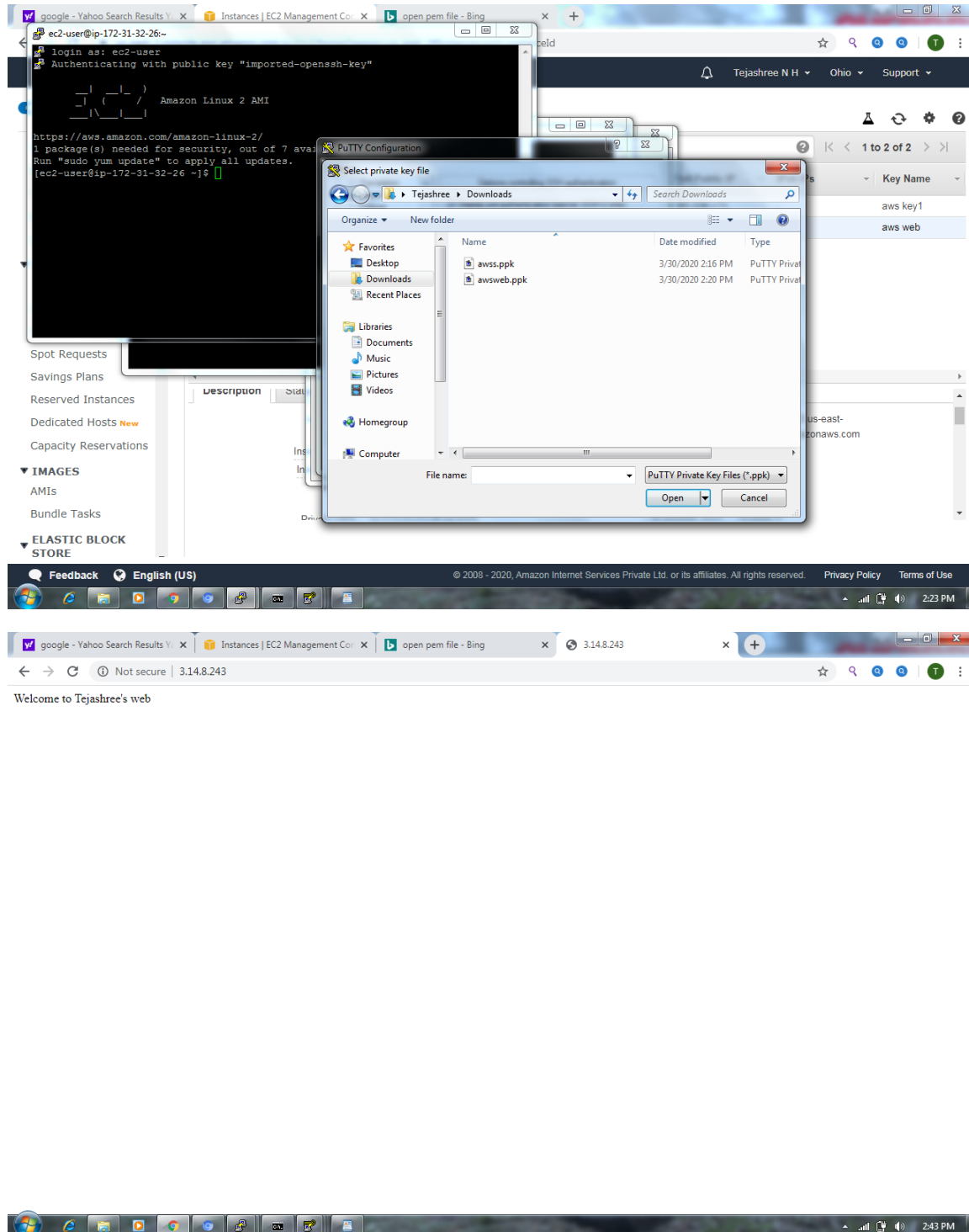
## 5. Key Pair Download :



## 6. Putty conversion from pem to ppk :

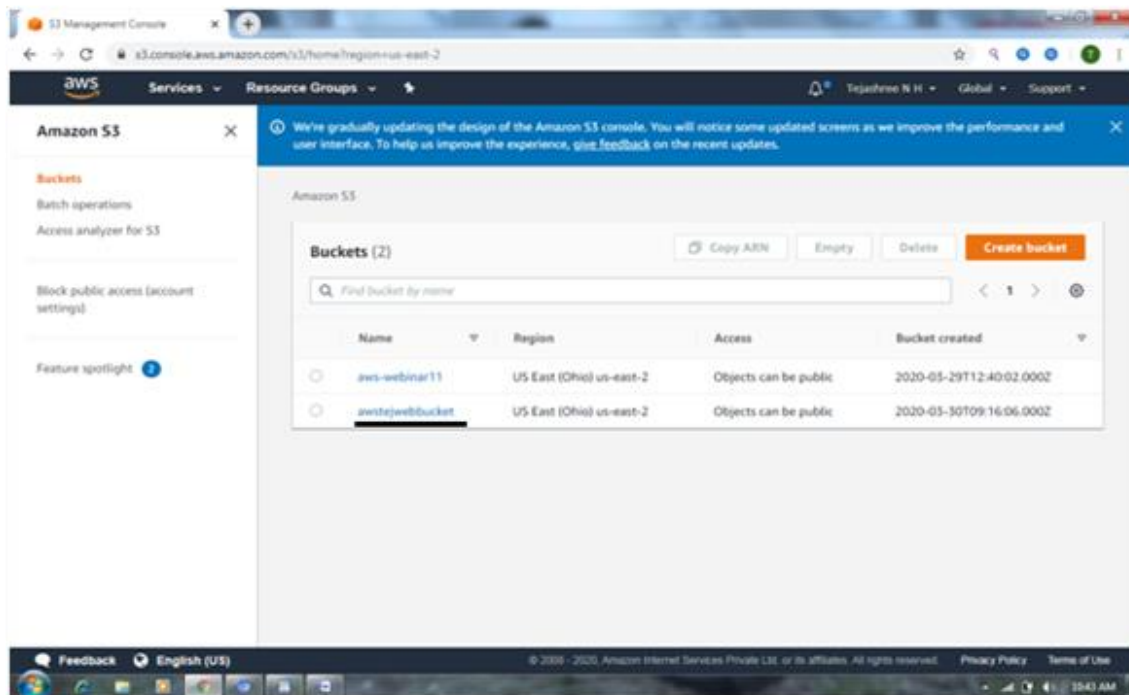


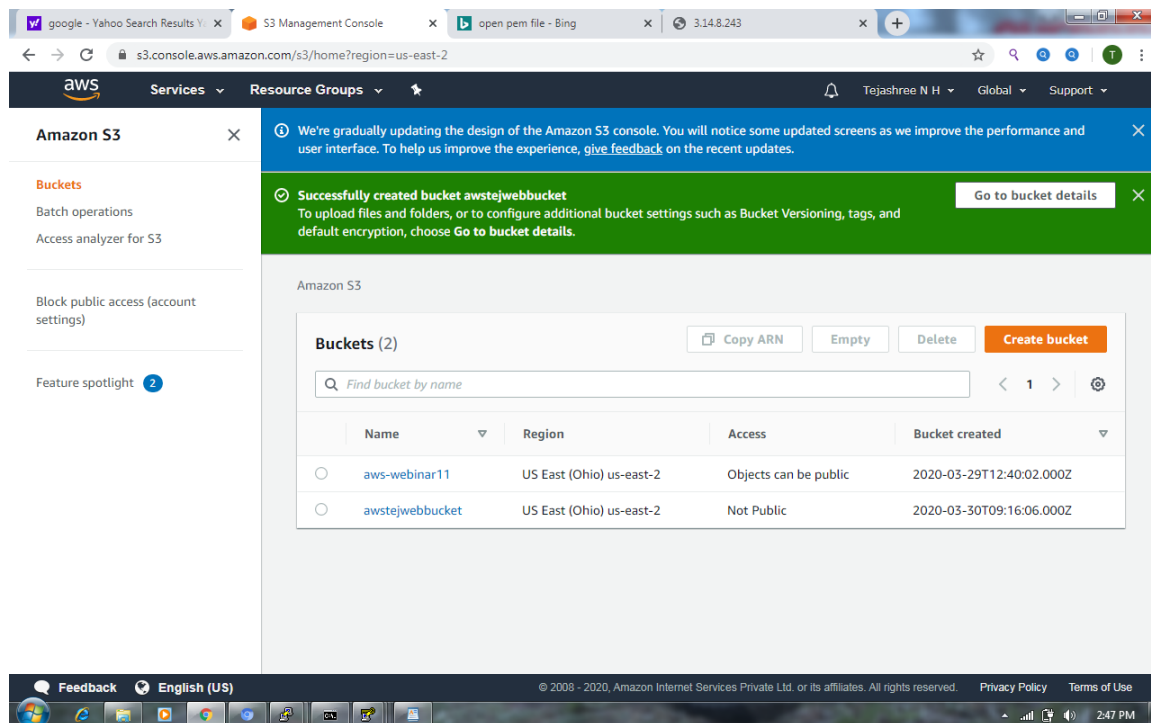
## 7. Logging in EC2 black screen :



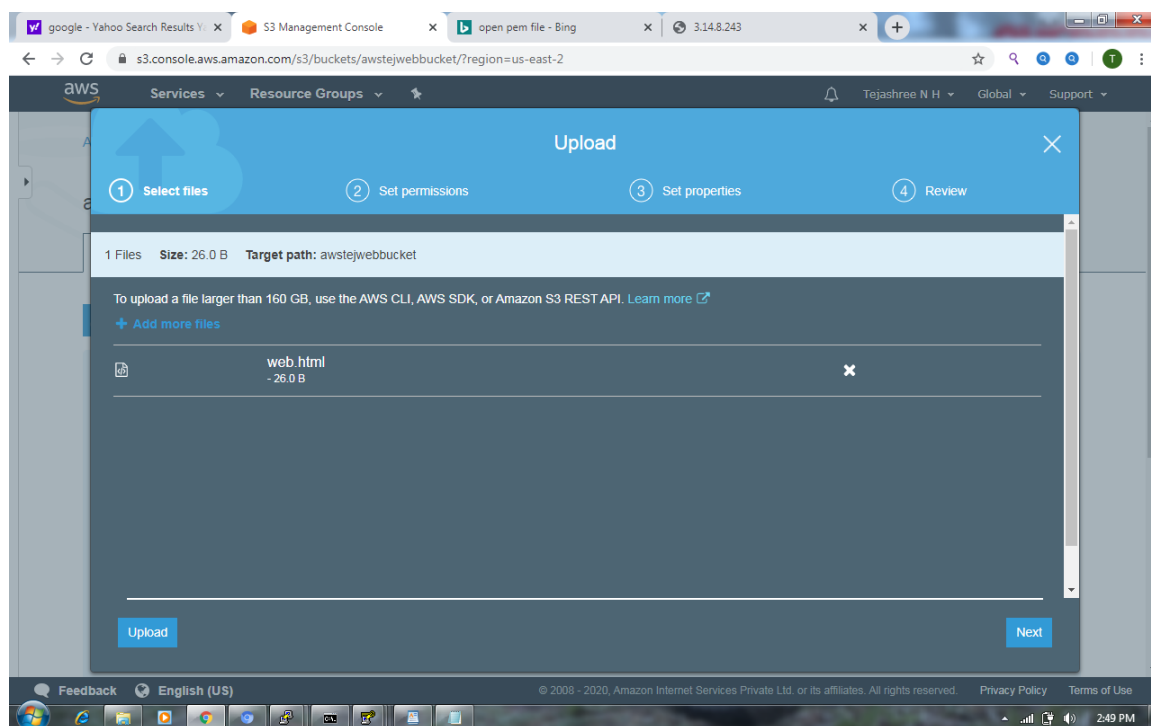
# S3 Screenshots

## 1. Creating a bucket :

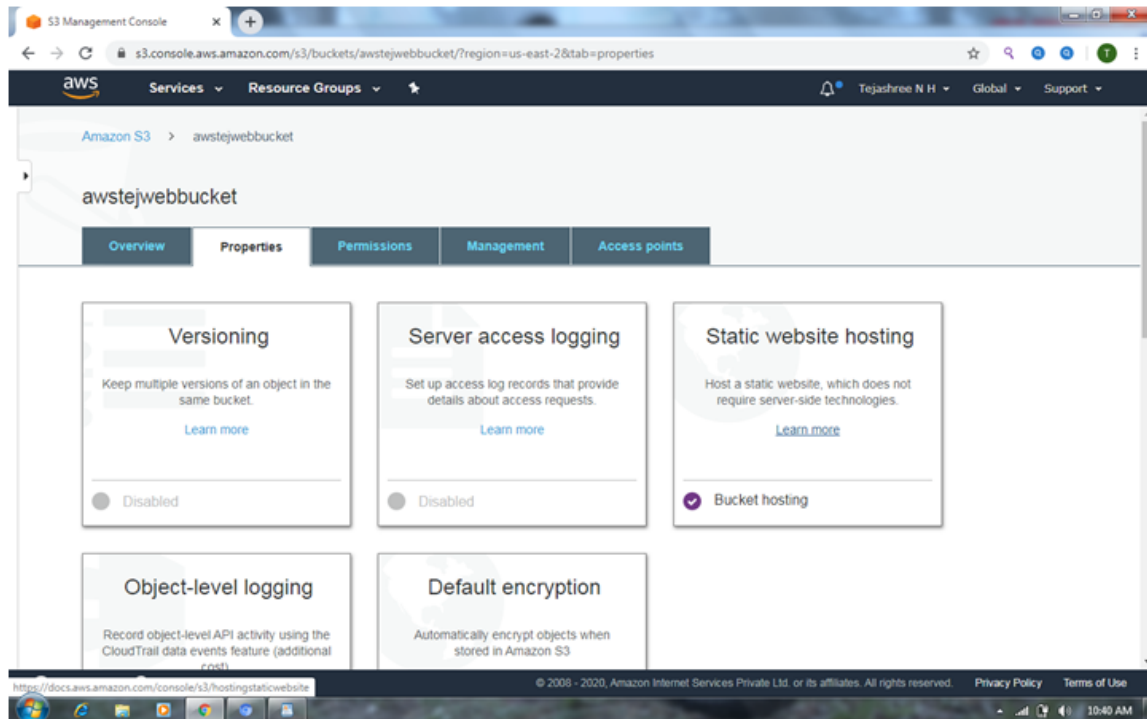




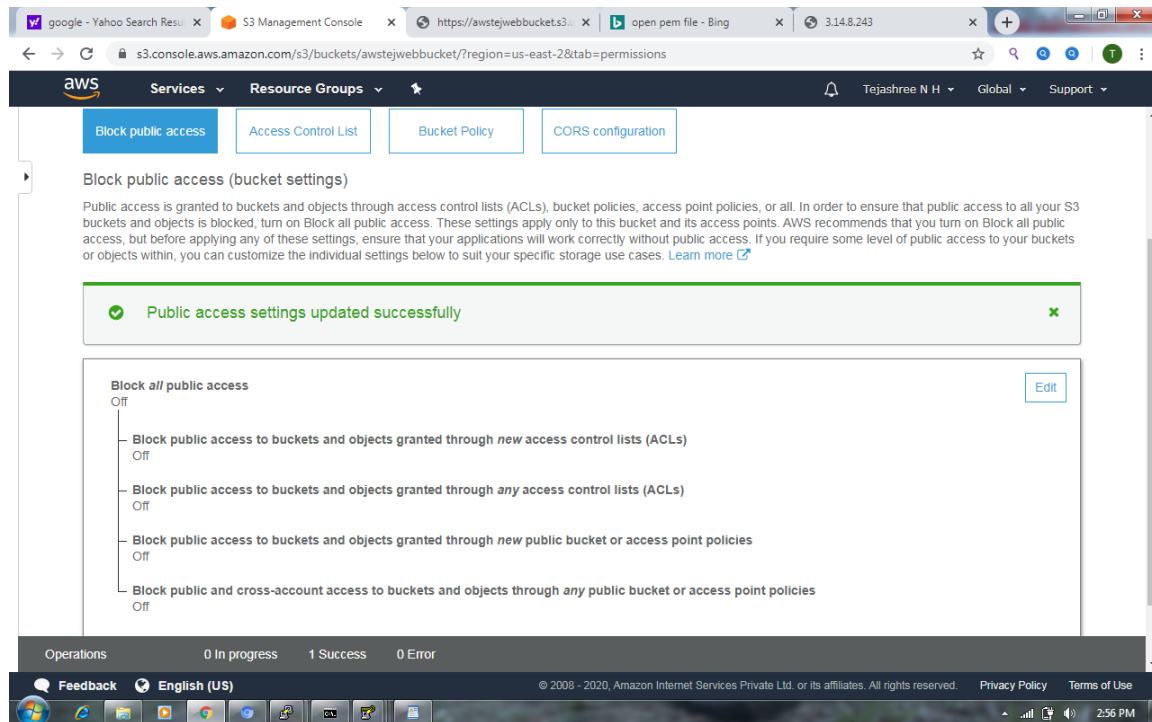
## 2. Uploading an Object :



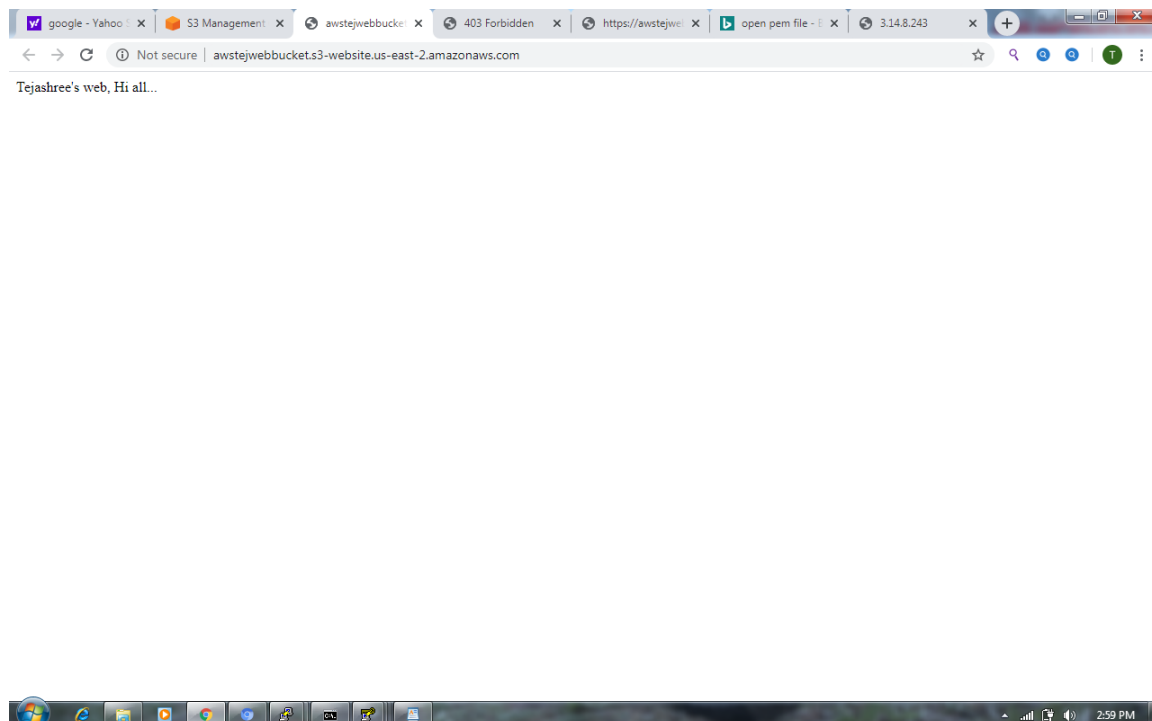
### 3. Enabling Static Website :



## 4. Making the Object Public :

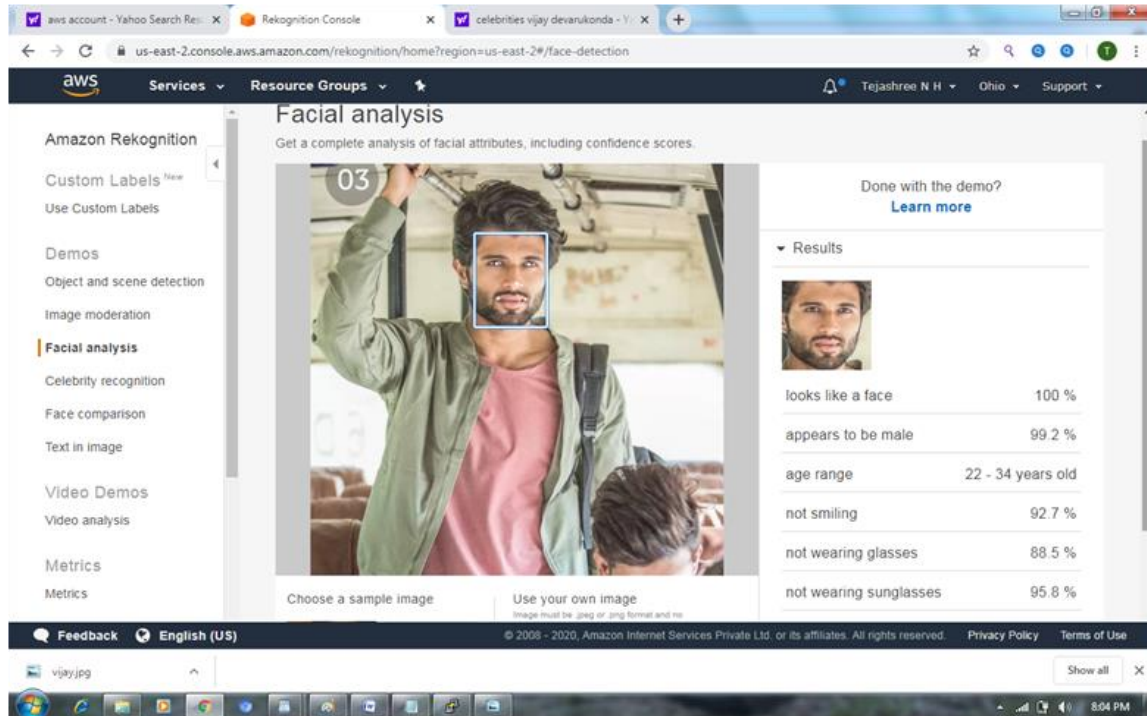


## 5. Checking the S3 link on the browser :



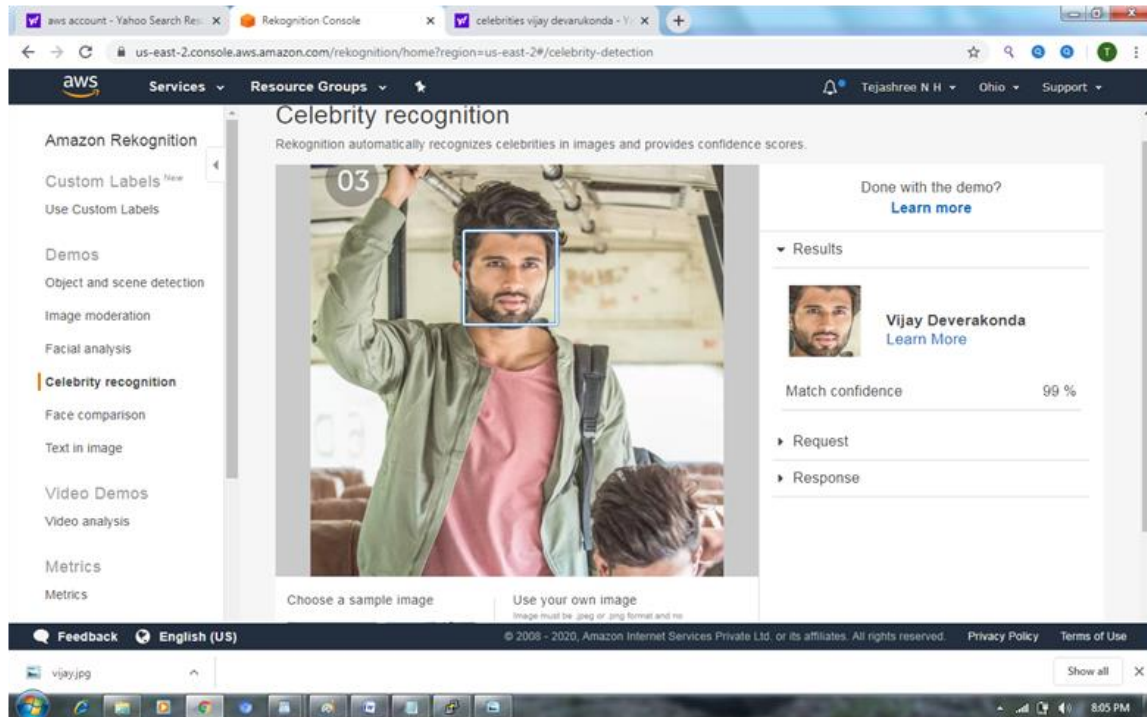
# Rekognition screenshots

## 1. Face detect :

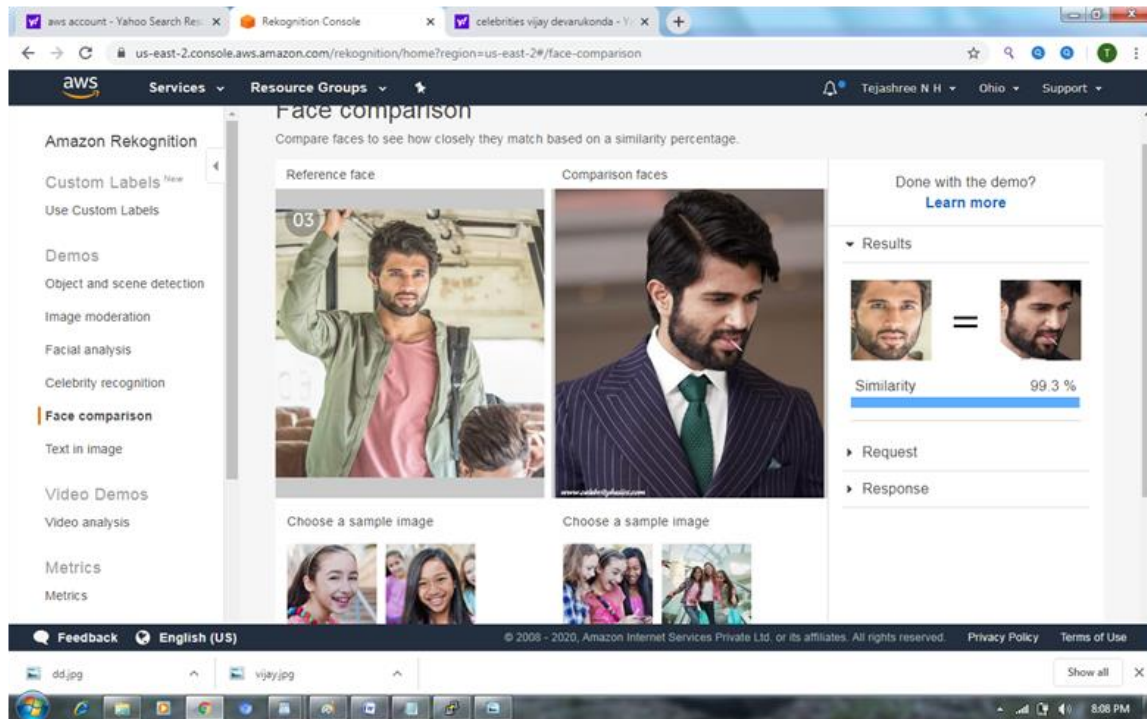




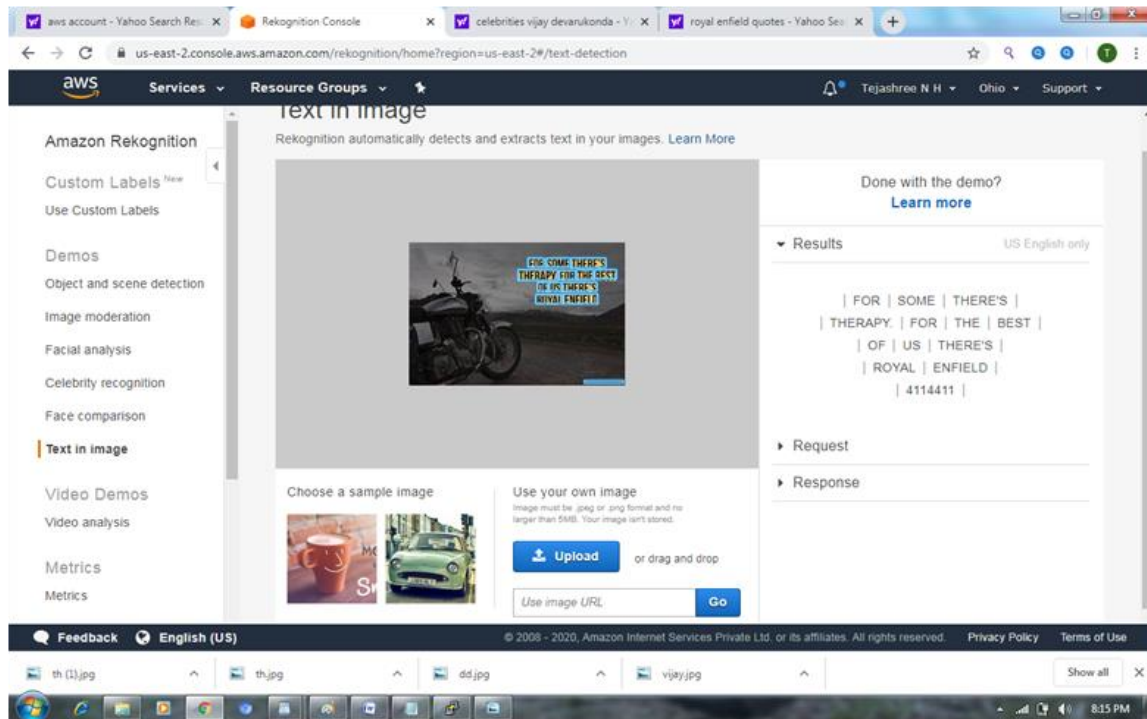
## 2. Celebrity recognition :



### 3. Face Compare :

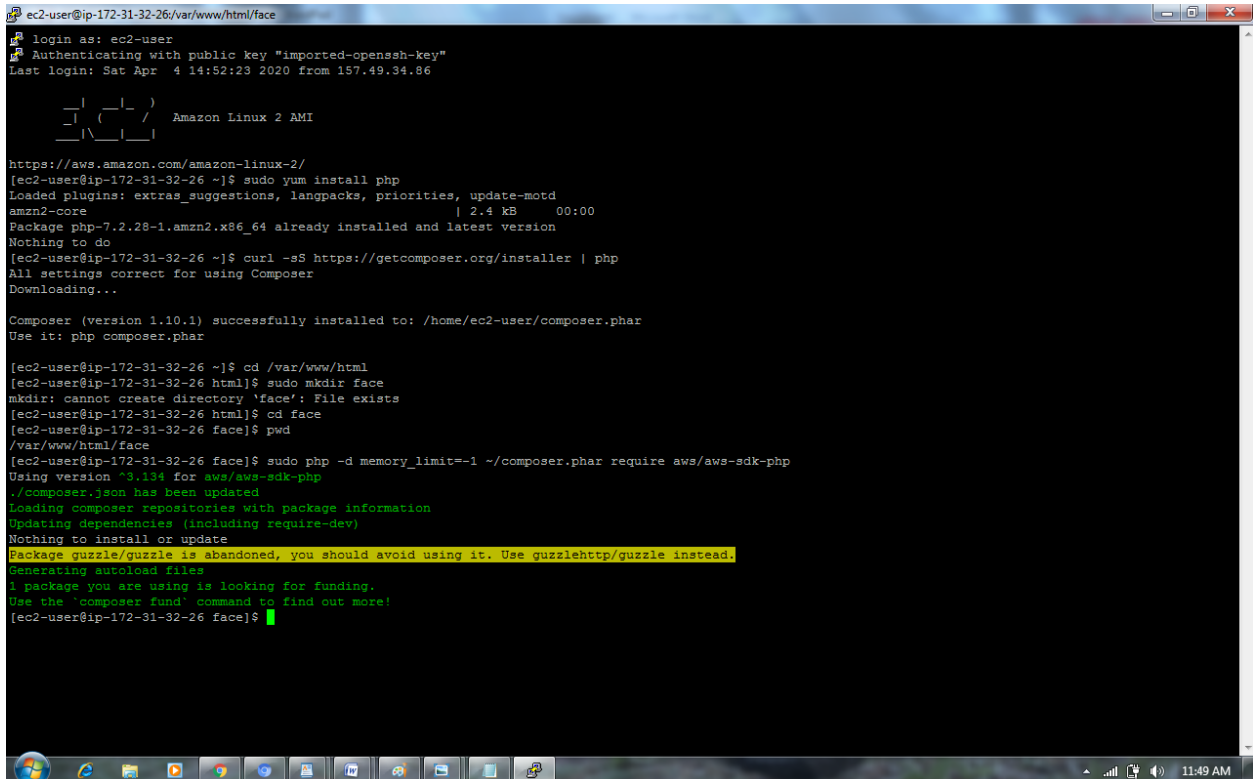


## 4. Text in Image :



# EC2 & S3 Screenshots

## 1. Installing aws-sdk :



```
ec2-user@ip-172-31-32-26:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sat Apr  4 14:52:23 2020 from 157.49.34.86

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-32-26 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                | 2.4 kB    00:00
Package php-7.2.28-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-32-26 ~]$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 1.10.1) successfully installed to: /home/ec2-user/composer.phar
Use it: php composer.phar

[ec2-user@ip-172-31-32-26 ~]$ cd /var/www/html
[ec2-user@ip-172-31-32-26 html]$ sudo mkdir face
mkdir: cannot create directory 'face': File exists
[ec2-user@ip-172-31-32-26 html]$ cd face
[ec2-user@ip-172-31-32-26 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-32-26 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version ^3.134 for aws/aws-sdk-php
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Generating autoload files
1 package you are using is looking for funding.
Use the 'composer fund' command to find out more!
[ec2-user@ip-172-31-32-26 face]$
```

## 2.installing php :

```
ec2-user@ip-172-31-34-189:~$ sudo yum install php
Installing:
php x86_64 5.4.16-46.amzn2.0.2 amzn2-core 1.4 M
Installing for dependencies:
libzip010-compat x86_64 0.10.1-9.amzn2.0.5 amzn2-core 30 k
php-cli x86_64 5.4.16-46.amzn2.0.2 amzn2-core 2.8 M
php-common x86_64 5.4.16-46.amzn2.0.2 amzn2-core 563 k

Transaction Summary
-----
Install 1 Package (+3 Dependent packages)

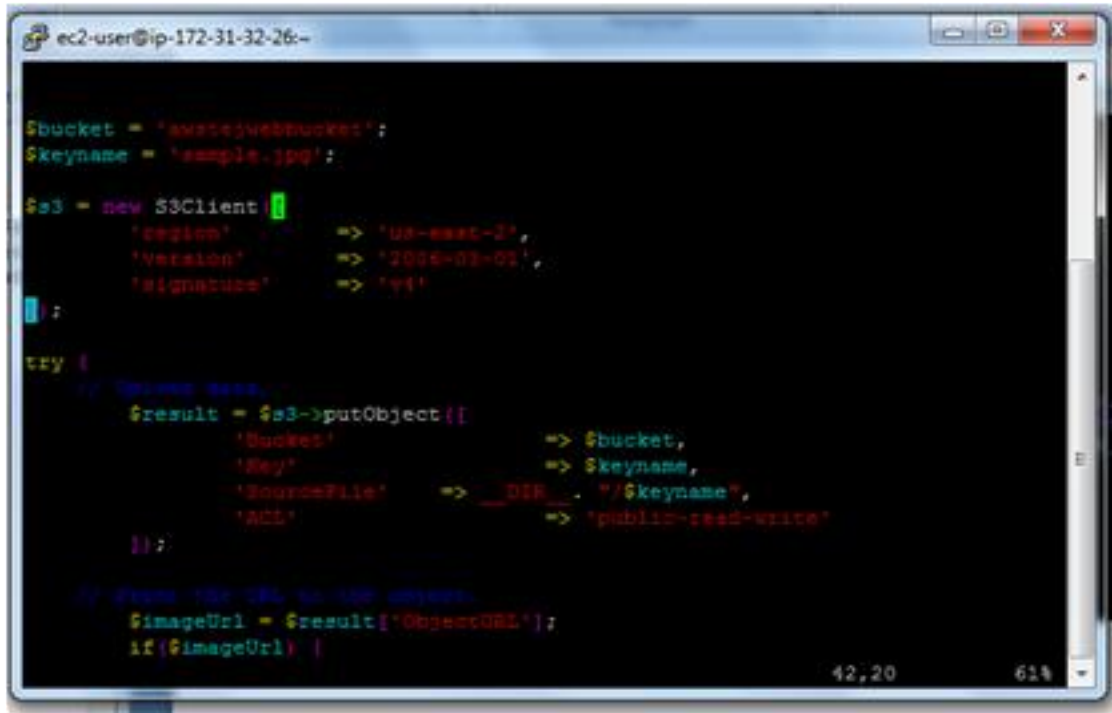
Total download size: 4.7 M
Installed size: 17 M
Is this ok [y/d/N]: y
Downloading packages:
(1/4): libzip010-compat-0.10.1-9.amzn2.0.5.x86_64.rpm | 30 kB 00:00
(2/4): php-5.4.16-46.amzn2.0.2.x86_64.rpm | 1.4 MB 00:00
(3/4): php-common-5.4.16-46.amzn2.0.2.x86_64.rpm | 563 kB 00:00
(4/4): php-cli-5.4.16-46.amzn2.0.2.x86_64.rpm | 2.8 MB 00:00
-----
Total 19 MB/s | 4.7 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libzip010-compat-0.10.1-9.amzn2.0.5.x86_64 1/4
Installing : php-common-5.4.16-46.amzn2.0.2.x86_64 2/4
Installing : php-cli-5.4.16-46.amzn2.0.2.x86_64 3/4
Installing : php-5.4.16-46.amzn2.0.2.x86_64 4/4
Verifying : php-5.4.16-46.amzn2.0.2.x86_64 1/4
Verifying : libzip010-compat-0.10.1-9.amzn2.0.5.x86_64 2/4
Verifying : php-cli-5.4.16-46.amzn2.0.2.x86_64 3/4
Verifying : php-common-5.4.16-46.amzn2.0.2.x86_64 4/4

Installed:
php.x86_64 0:5.4.16-46.amzn2.0.2

Dependency Installed:
libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5
php-cli.x86_64 0:5.4.16-46.amzn2.0.2
php-common.x86_64 0:5.4.16-46.amzn2.0.2

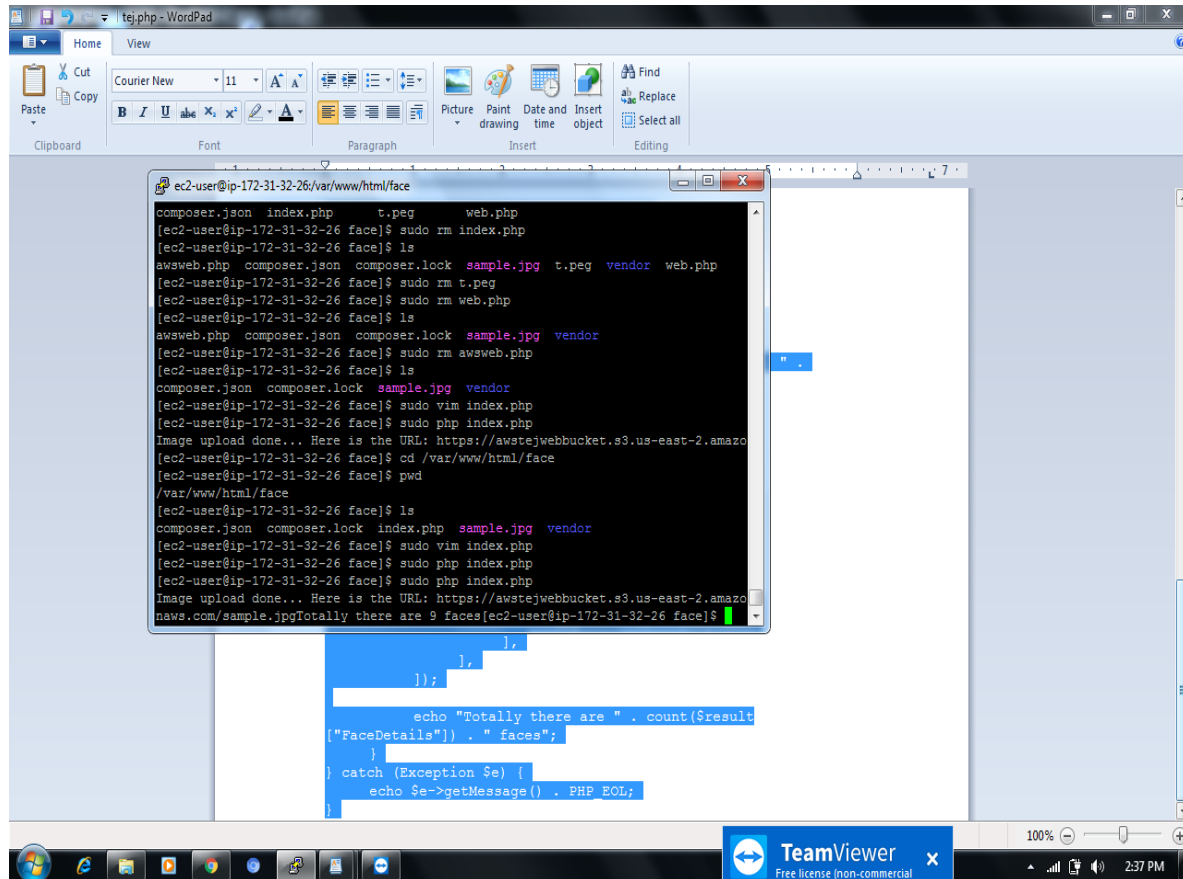
Complete!
[ec2-user@ip-172-31-34-189 ~]$
```

### 3.index.php file code :



```
ec2-user@ip-172-31-32-26:~$  
  
$bucket = 'awsfacebucket';  
$keyname = 'sample.jpg';  
  
$s3 = new S3Client([  
    'region'      => 'us-east-2',  
    'version'     => '2006-03-01',  
    'signature'   => 'v4'  
]);  
  
try {  
    // Upload some.  
    $result = $s3->putObject([  
        'Bucket'      => $bucket,  
        'Key'         => $keyname,  
        'SourceFile'  => __DIR__ . "/" . $keyname,  
        'ACL'         => 'public-read-write'  
    ]);  
  
    // From the URL to the object.  
    $imageUrl = $result['ObjectURL'];  
    if($imageUrl) {  
        echo "Image uploaded successfully: " . $imageUrl . "  
    }  
}
```

## 4.Success Screenshot :

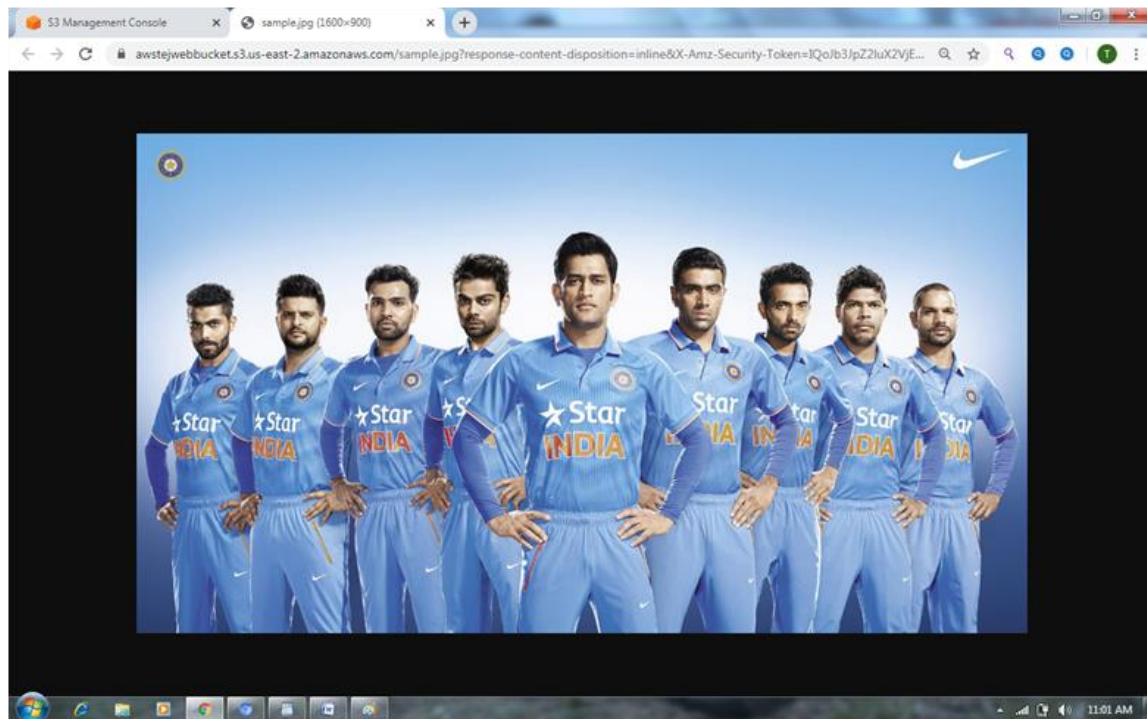


The screenshot shows a Windows WordPad window titled 'tej.php - WordPad'. The main text area contains PHP code for a face detection application. A terminal window is overlaid on the WordPad, showing the execution of the code on an AWS EC2 instance. The terminal output shows the successful upload of images and the detection of 9 faces.

```
composer.json index.php t.peg web.php
[ec2-user@ip-172-31-32-26 face]$ sudo rm index.php
[ec2-user@ip-172-31-32-26 face]$ ls
awsweb.php composer.json composer.lock sample.jpg t.peg vendor web.php
[ec2-user@ip-172-31-32-26 face]$ sudo rm t.peg
[ec2-user@ip-172-31-32-26 face]$ sudo rm web.php
[ec2-user@ip-172-31-32-26 face]$ ls
awsweb.php composer.json composer.lock sample.jpg vendor
[ec2-user@ip-172-31-32-26 face]$ sudo rm awsweb.php
[ec2-user@ip-172-31-32-26 face]$ ls
composer.json composer.lock sample.jpg vendor
[ec2-user@ip-172-31-32-26 face]$ sudo vim index.php
[ec2-user@ip-172-31-32-26 face]$ sudo php index.php
Image upload done... Here is the URL: https://awstejwebbucket.s3.us-east-2.amazo
[ec2-user@ip-172-31-32-26 face]$ cd /var/www/html/face
[ec2-user@ip-172-31-32-26 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-32-26 face]$ ls
composer.json composer.lock index.php sample.jpg vendor
[ec2-user@ip-172-31-32-26 face]$ sudo vim index.php
[ec2-user@ip-172-31-32-26 face]$ sudo php index.php
[ec2-user@ip-172-31-32-26 face]$ sudo php index.php
Image upload done... Here is the URL: https://awstejwebbucket.s3.us-east-2.amazo
naws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-32-26 face]$
```

```
}
    },
    });
    echo "Totally there are " . count($result["FaceDetails"]) . " faces";
}
} catch (Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```





Face Detection success screenshot :

